Phishing Prevention: A Multi-Layered Approach

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der
RWTH Aachen University zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Vincent Jakob Drury, M.Sc. RWTH
aus Herdecke

Berichter: Univ.-Prof. Dr. Ulrike Meyer
Univ.-Prof. Dr. Sascha Fahl

Tag der mündlichen Prüfung: 16. Juni 2023

Abstract. Phishing attacks have been a relevant and ongoing threat for several decades, resulting in monetary loss for private users and serving as a first step in attacks on larger organizations. Despite decades of research into automated detection, education, and design interventions to prevent phishing attacks, current solutions fall short of providing adequate and general protection to users and businesses against the evolving threat. In particular, we identify research gaps for anti-phishing learning games, which currently lack personalization and mainly offer rather simple game mechanics, the user interface (UI) design of email clients, which do not highlight information that is relevant to phishing email detection, and automated phishing website classification, where the focus is on detection after the attack was executed, instead of aiming to disrupt attacks before they reach potential victims. This thesis addresses these gaps and showcases the effectiveness of human-centered and technical phishing-prevention techniques based on a categorization of phishing URLs and a kill chain model, thus contributing to a multi-layered defense strategy where each layer addresses different attacks.

Our main contributions in the research area of anti-phishing education are the evaluation of baseline detection capabilities for a new categorization of phishing URLs, as well as the comparative evaluation of four new learning games in two user studies. The result of these user studies motivate, which categories of URLs to focus on to optimize future educational interventions, and give insights into the effect of different game mechanics and personalization on the classification capabilities of users who played the learning games. We further demonstrate, that accurately reflecting the diversity of phishing attacks and measuring service familiarity in user studies is essential to ensure representative results.

In the area of design interventions, we evaluate the effect of using Reverse Domain Name (RDN) notation for URLs and changing the UI of email clients on the classification performance of untrained users. Both studies reveal advantages of the proposed changes over the baseline and motivate further studies of the interventions' effect on awareness outside of a lab setting.

For automated phishing detection, we evaluate the detection of phishing websites on certificate transparency (CT) logs, which are publicly available stores of certificates. We show, how data cleaning and class imbalance during training, and the inclusion of additional certificate information in the classification task can have an effect on classification performance, resulting in classifiers that approach acceptable levels of false positives to be practical in the real world.

In all, our results exemplify the advantages and disadvantages of several broader approaches to phishing prevention, and demonstrate how combining these approaches can provide a more comprehensive defense than each of its parts taken by itself.

Zusammenfassung. Phishing Angriffe stellen schon seit langer Zeit ein Risiko für NutzerInnen und Unternehmen dar, da sie zu finanziellen Verlusten führen und als erster Schritt in komplexen Angriffen dienen können. Trotz jahrzehntelanger Forschung in den Bereichen der automatisierten Erkennung, Bildung, und Design Interventionen um Phishing zu verhindern, existieren derzeit noch keine adäquaten und generellen Lösungen um NutzerInnen und Unternehmen vor der sich kontinuierlich entwickelnden Bedrohung zu schützen. Diese Doktorarbeit identifiziert Forschungslücken im Bereich von Anti-Phishing Lernspielen, welche derzeit keine Personalisierung anbieten und größtenteils auf relativ einfachen Spielmechanismen basieren, den Bedienoberflächen (UIs) von Email Client Anwendungen, in denen keine für Phishing relevanten Informationen hervorgehoben werden, und der automatisierten Klassifizierung von Phishing Webseiten, welche sich derzeit auf die Erkennung von Angriffen fokussieren nachdem diese bereits gestartet wurden, statt Angriffe zu verhindern bevor sie ein potentielles Opfer erreichen können. Diese Arbeit befasst sich mit den Forschungslücken und präsentiert die Effektivität von didaktischen und automatischen Präventionsmaßnahmen basierend auf einer Kategorisierung von Phishing URLs sowie eines Killchain Models, und steuert so zu einer mehrschichtigen Verteidigungsstrategie bei in der jede Schicht gegen verschiedene Angriffe vorgeht.

Der Hauptbeitrag im Forschungsbereich der Anti-Phishing Bildung sind die Evaluation von Referenzwerten der Erkennungsfähigkeit von NutzerInnen für eine neue Kategorisierung von Phishing URLs, sowie eine vergleichende Evaluation von vier neuen Anti-Phishing Lernspielen, welche in zwei Studien erhoben werden. Die Ergebnisse der Studien können angewandt werden um zu bestimmen, welche URL Kategorien in neuen Interventionen fokussiert werden sollten um deren Effizienz zu maximieren, und schaffen neue Erkenntnisse bezüglich der Auswirkung von Personalisierung und verschiedenen Spielmechanismen auf die Klassifizierungsfähigkeit von NutzerInnen in Lernspielen. Außerdem demonstrieren sie, dass das Einbinden von diversen Angriffen und des Bekanntheitsgrades von Diensten in Studien notwendig ist um representative Ergebnisse zu liefern.

Im Bereich der Design Interventionen werden die Auswirkung von Reverse Domain Name (RDN) Notation für URLs und von Änderungen im UI für Email Client Anwendungen auf die Klassifizierungsfähigkeit von untrainierten NutzerInnen getestet. Beide Studien offenbaren Vorteile der vorgeschlagenen Änderungen über die Ausgangssituation, und motivieren weitere Studien um die Effektivität der Interventionen in realistischeren Szenarien zu ermitteln.

Für automatisierte Phishing Erkennung wird die Erkennung auf Certificate Transparency (CT) Aufzeichnungen evaluiert, bei denen es sich um öffentlich zugänglichen Quellen von Zertifikaten handelt. Diese Arbeit evaluiert die Auswirkung von verschiedenen Filterungstechniken und Class Imbalance für Trainingsdaten, sowie der Einbindung von zusätzlichen Informationen aus Zertifikaten im Klassifizierungsschritt, was insgesamt zu Klassifikatoren führt, welche in der echten Welt Einsatz finden können.

Insgesamt veranschaulicht die Doktorarbeit, wie die Vor- und Nachteile von verschiedenen allgemeineren Ansätzen zur Phishingprävention zusammenspielen, und demonstriert wie die Kombination dieser Ansätze einen umfassenderen Schutz bieten kann als jegliches Einzelteil für sich.

Acknowledgments

This dissertation is the result of several years of work, and would not have been possible without the generous support of the following people.

First, I am extremely grateful to my supervisor, Prof. Dr. Ulrike Meyer, whose guidance and advice have been invaluable during my time as a PhD student, and whose feedback to this thesis and my work in general have been a constant source for inspiration and improvement. I am also thankful to Prof. Dr. Sascha Fahl and my dissertation committee, who made organizing the defense a smooth experience by keeping to the proposed deadlines despite opposing circumstances.

Special thanks are due to my colleagues at the Research Group IT Security, particularly my long-standing fellow PhD students Andreas Klinger, Arthur Drichel, and Malte Breuer, who offered valuable discussions, kept me motivated by sharing interesting problems and ideas, and created a working environment that made me feel welcome and appreciated. I am also particularly thankful to René Röpke from the Learning Technologies Research Group, who generously shared his experiences and helpful advice, in addition to being a great and knowledgeable collaborator on several publications. Thanks should also go to the members of the NERD research group, who provided feedback and ideas, and helped direct me when I was still starting to plan my dissertation.

Many thanks are due to the student assistants who supported the realization of the approaches presented in this thesis, but also helped shape the core ideas with their insights and discussions. I would additionally like to mention the bachelor and master students who worked with me in the context of their thesis projects, and who helped me explore the research area more efficiently than I could have on my own.

Lastly, I would like to express my deepest gratitude to my friends and family, and my parents in particular, whose encouragement gave me the fortitude to continue this project, even when motivation was low or deadlines too close.

Previously Published

Parts of this thesis are based on previously published work, which is listed below. Explanations of the authors' contributions to these publications can be found in Appendix B.

[1] Arthur Drichel, Vincent Drury, Justus von Brandt, and Ulrike Meyer. "Finding phish in a haystack: A pipeline for phishing classification on certificate transparency logs". In: *Proceedings of the 16th International Conference on Availability, Reliability and Security*. ACM, 2021.

[2] Vincent Drury, Luisa Lux, and Ulrike Meyer. "Dating Phish: An Analysis of the Life Cycles of Phishing Attacks and Campaigns". In: *Proceedings of the 17th International Conference on Availability, Reliability and Security*. ACM, 2022.

[3] Vincent Drury and Ulrike Meyer. "Certified phishing: taking a look at public key certificates of phishing websites". In: *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. USENIX, 2019.

[4] Vincent Drury and Ulrike Meyer. "No Phishing With the Wrong Bait: Reducing the Phishing Risk by Address Separation". In: *2020 European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2020.

[5] Vincent Drury, Rene Roepke, Ulrik Schroeder, and Ulrike Meyer. "Analyzing and creating malicious URLs: a comparative study on anti-phishing learning games". In: *Proceedings of the Workshop on Usable Security and Privacy (USEC 2022)*. Internet Society, 2022.

[6] Rene Roepke, Vincent Drury, Ulrike Meyer, and Ulrik Schroeder. "Exploring Different Game Mechanics for Anti-Phishing Learning Games". In: *Games and Learning Alliance (GaLA '21)*. Springer, 2021.

[7] Rene Roepke, Vincent Drury, Ulrike Meyer, and Ulrik Schroeder. "Exploring and evaluating different game mechanics for anti-phishing learning games". In: *International Journal of Serious Games* 9.3 (2022).

[8] Rene Roepke, Vincent Drury, Philipp Peess, Tobias Johnen, Ulrike Meyer, and Ulrik Schroeder. "More Than Meets the Eye - An Anti-Phishing Learning Game with a Focus on Phishing Emails". In: *Games and Learning Alliance (GaLA '22)*. Springer, 2022.

[9] Rene Roepke, Vincent Drury, Ulrik Schroeder, and Ulrike Meyer. "A modular architecture for personalized learning content in anti-phishing learning games". In: *Software Engineering 2021 Satellite Events (SE-SE 2021)*. CEUR, 2021.

[10] Rene Roepke, Vincent Jakob Drury, Ulrike Meyer, and Ulrik Schroeder. "Better the Phish You Know : Evaluating Personalization in Anti-Phishing Learning Games". In: *Proceedings of the 14th International Conference on Computer Supported Education (CSEDU 2022)*. SciTePress, 2022.

[11] Rene Roepke, Klemens Koehler, Vincent Drury, Ulrik Schroeder, Martin R. Wolf, and Ulrike Meyer. "A Pond Full of Phishing Games - Analysis of Learning Games for Anti-Phishing Education". In: *Model-Driven Simulation and Training Environments for Cybersecurity*. Springer, 2020.

[12] Rene Roepke, Ulrik Schroeder, Vincent Drury, and Ulrike Meyer. “Towards personalized game-based learning in anti-phishing education”. In: *20th International Conference on Advanced Learning Technologies (ICALT)*. IEEE, 2020.

Contents

Chapter 1

Introduction

Phishing attacks are “a scalable act of deception whereby impersonation is used to elicit an exploitable action from a victim” (modified from [Las14]). Their active focus on the human element, coupled with an aptitute for evolving alongside current technology, have cemented phishing attacks as a consistent threat, even with a history of attacks and defenses dating back several decades.

The APWG reports on the number of phishing attacks per quarter of the year, where they found an all-time high in the third quarter of 2022 with over one million unique observed attacks [APW22]. Similarly, AV vendor Kaspersky reports the number of emails that included malicious attachments as almost 150,000,000 across all of their users in 2021, and blocked more than 250,000,000 phishing links, more than 300,000 of which were sent via messenger apps [Kas22]. These numbers translate into actual consequences for general users of the Internet, as can for example be seen in the annual threat report by the FBI, who reported more than 300,000 victims in the year 2021, resulting in almost 45,000,000 USD of loss (more than 2 billion USD if counting business email compromise and other related attacks) [Inv21]. Phishing attacks are also encountered in different contexts, as is demonstrated by Verizon’s data breach investigation report, where the human element was present in more than 80% of attacks, and phishing attacks were among the most popular attack vectors [Ver23]. These findings imply, that phishing attacks, in particular those that are highly targeted, are a powerful tool of attackers to gain a first foothold in an organization.

In this thesis, we argue that the great diversity of phishing attacks is one of the problems why phishing detection is a complex task. Phishing attacks can come in a variety of different configurations, with different delivery methods, goals, and levels of sophistication by attackers. This makes it complicated to provide general protections against phishing attacks, without compromising on the freedom provided by the current ecosystem surrounding the Internet.

The current state of phishing prevention in practice is mainly based on blocklists, which are for example integrated into popular browsers. These blocklists ensure an extremely low rate of false alarms, as only known phishing websites are added and entries can be removed after the malicious website was taken down or cleaned, but are also reactive in nature. This leads to a short timespan in which users are unprotected, which can be abused by attackers. As a consequence, previous research

has found that phishing attacks typically only have a very short lifespan of several hours or days (see Chapter 3 for more details), where attackers attempt to defraud as many victims as possible in the short timespan that is available to them.

To improve upon reactive blocklists, several research directions have been proposed and explored. One prominent example is automated phishing detection, which aims to detect attacks without requiring human intervention and in real-time by applying machine learning or heuristic techniques. However, current methods seem to be unable to fulfill the requirements to be integrated more widely in the real world. In particular, they seem to struggle to offer a high level of protection while guaranteeing low numbers of false alarms in real-world settings. They might consequently be avoided by vendors of anti-phishing products who fear users losing trust in their products or even legal consequences, for example when erroneously flagging a popular legitimate website [She+09]. It is also somewhat unclear how the detection approaches can be integrated into the greater phishing prevention workflow, as it is unlikely that users without technical inclination would, for example, install a browser add-on that includes an automated URL classifier. Instead, centralized approaches based on blocklists seem to currently be the preferred method of anti-phishing product vendors.

A second, complementary approach to phishing prevention is user education. User actions are always a vital part of phishing attacks (see the definition in Chapter 2), and preventing users from taking that action therefore disrupts the attack completely. In particular, game-based anti-phishing education has emerged as a scalable and motivational resource, with promising results in previous studies. Existing games, however, mostly incorporate binary classification mechanisms, which do not provide details on the decision processes of players and leaves room for random guessing to advance in the games. Furthermore, user studies that evaluate anti-phishing learning games have largely ignored the effect of familiarity with the services that appear as examples in the games and tests, which might lead to potential biases and complicates reproduction studies.

Next to the active approach of education, design interventions that aim to highlight or change the displayed information to increase awareness or improve detection rates have been studied as well. A recent overview paper on educational interventions found this area to be promising in providing more general methods to phishing prevention, but also with less focus from the research community compared to the active educational approaches [Fra+21]. Existing examples of design interventions include the highlighting of relevant information in the URL bar in browsers to emphasize where the URL leads, as well as highlighting information in email clients to make users focus on sender information.

Other approaches to prevent phishing attacks, such as strong authentication, are not widely deployed or understood, or offer fallback methods that can be abused in phishing attacks. As such, the combination of technical and human-centered interventions is currently necessary to combat the risk of phishing attacks.

In this thesis, we approach the complex problem of phishing prevention from different perspectives, and evaluate four defenses which can be deployed independently: an automated detection approach using TLS certificates, the usage of anti-phishing learning games as educational interventions, as well as two design interventions focusing on URL notation and email clients. Each defensive approach operates at different steps of the phishing attack process, thus providing different protections

and focusing on different attacks. The result is a layered defense that offers a more comprehensive protection than any of the four approaches taken by itself. We first define impostor URLs as a subset of phishing URLs that include a reference to a target and are of particular interest in this thesis, as they offer more context than random URLs or URLs that do not include a reference to a target. Then, we present a categorization of impostor URLs based on the general structure of URLs, which we use as a basis to evaluate which preventive measures thwart which types of attacks. The first interventions we evaluate are anti-phishing learning games that make use of more complex game mechanics compared to previous games and incorporate personalization options. Next, two design interventions are presented and evaluated, based on URL notation and email clients, that do not require user education and instead aim to highlight or rearrange information to nudge users towards more secure behavior. Finally, we present TLS certificates as a source for automated and manual detection of phishing websites, and evaluate machine learning classifiers that offer the potential for early and centralized detection of phishing websites based on their certificates.

1.1. Contributions

In this thesis, we present and evaluate several preventive measures against phishing attacks, which include technical and human-centered approaches. We introduce the main contributions of the thesis in this section, followed by an outline in the next section. Details on the contributions for collaborative work are available in Appendix B.

Impostor URLs

Our first contribution is a categorization of phishing URLs based on the concept of impostor URLs, which are a subset of phishing URLs that include a reference to a target, thus offering more context for automated detection and education. It extends previous work that studies URL reading capabilities by consolidating categories from different studies directly with the structure of URLs and unambiguously matching URLs to their respective categories. The proposed categorization therefore offers a more detailed look at phishing URLs overall, since it includes more atomic sub-categories. The baseline detection capabilities of untrained users for benign and phishing URLs from different categories were evaluated in a study where 45 participants classified URLs as either benign or phishing. The study shows, that users perform worst when classifying URLs that start similarly to the benign target URL (i.e., when reading the URLs from left to right they are similar or identical at first), where they perform comparable to random guessing. Parts of this work are submitted for publication.

Education

Our next contribution is the analysis of anti-phishing learning games and their game mechanics and learning contents. Here, we find that existing games mostly include a binary decision as their main game mechanic, which could be extended to provide additional benefits, such as better feedback during gameplay. Furthermore,

existing games do not include personalization options, which might restrict how well the learned skills and knowledge translate into the real world and everyday life of players. To address these research gaps, we present four new game prototypes, to directly compare the effect of more complex game mechanics, as well as the effect of personalization and how users interact with URLs of services they are not familiar with during gameplay. The results of a user study with 182 participants indicate, that the usage of more complex game mechanics or personalization did not result in immediate improvements over the baseline, but do offer additional insights into the players decision processes. A retention study with 82 participants performed three months after the post-test further indicates, that knowledge was retained, as the classification performance remained significantly higher compared to the pre-test. In part, this work has been published in [5; 7; 10] and [11].

Design Interventions

The third contribution of this thesis is the analysis of two design interventions as a complementary approach to user education. Here, we propose Reverse Domain Name (RDN) notation as an alternative to the current URL notation, since RDN notation has the potential advantage of moving important information for phishing detection to the beginning of the URL, thus better representing the apparent parsing process of users found in the previous study. The usage of RDN notation to improve the readability of URLs has, to our knowledge, not been tested previously, and complements previous work on the effect of URL highlighting techniques. A user study that was performed to compare RDN to normal notation with 47 participants indicates, that a category of URLs that was previously not detected well indeed improves significantly using the new notation, and that the time taken to classify URLs was significantly lower for the new notation, thus potentially making the classification task less taxing in the real world where security is typically not the main concern. Parts of this work are submitted for publication.

For the second study of design interventions, we present and compare four User Interfaces (UIs) for email clients that were created to highlight different information on the sender of emails, thus potentially supporting users in an email classification task. The UIs extend previous research on the effect of email UI changes on classification performance, and potentially awareness, by comparing more atomic changes, and presenting completely new highlighting options. We furthermore evaluate the UIs for six clearly defined categories of phishing and benign emails, which correspond to different attack and benign usage scenarios representing advanced attacks and situations that users are likely to encounter in their everyday online activities. A user study with 74 participants reveals significant improvements for newly proposed UIs, but with significant differences between the UIs as well, indicating a potential trade-off between better classification results for phishing or benign emails. The time taken to classify emails also decreased substantially, and might indicate that users mainly focused on the UI changes for classification, ignoring other indicators. This did, however, not result in worse classification results. We further find, that the simulated scenarios representing spear-phishing and lateral phishing represent a serious threat, as the classification performance for these categories was at best close to random guessing for the baseline UI, even in the lab-study setting.

Service Familiarity

All user studies in this thesis also include a questionnaire that asks for familiarity with the services which appear in the classification tasks, to correct for the effect of unknown services. While differences between known and unknown services have been hypothesized in the past (see, e.g., [Gop+21; SUM17; WLR16]), previous studies seem to be inconclusive about the effect and do not correct for potential differences. We confirm significant differences between unknown and known services in all user studies presented in this thesis, leading us to remove the responses for unknown services as a source of unwanted bias. In this way, our results validate and extend previous work which does not correct for these differences. Based on the observed differences, we furthermore recommend that future studies should correct for service familiarity in classification tasks whenever possible.

Automated Detection

The next contribution of this thesis is concerned with the automated detection of phishing websites as a complementary approach to both education and design interventions. The automated detection technique presented in this thesis is based on the TLS certificates of phishing websites, and focuses on the classification of certificates in certificate transparency (CT) logs. These logs are public stores of certificates which potentially enable the detection of phishing websites before the attack is executed, thus reducing the amount of time that victims are susceptible to attacks compared to traditional prevention methods, while still enabling the centralized detection and verification of phishing websites. The evaluation presented in this thesis extends previous work on detecting the certificates of phishing websites in CT logs by employing different classifiers and performing all evaluations on real-world CT log data. To this end, we present a classification pipeline for CT logs, which can be used to train and evaluate automated classifiers on CT log data. Based on an analysis and comparison of the certificates of benign and phishing websites, we test the effect of including different features extracted from certificates, in addition to the effect of different training data filtering techniques and class imbalance. We find, that training data cleaning and introducing class imbalance both have a positive effect on accuracy when focusing on extremely low false alarm rates in the order of 10^{-5} to 10^{-6}. Finally, we find that classifiers that only receive information about domain names in the certificate perform equally well to those that achieve additional input. In part, this work has been published in [1] and [3].

Datasets

Finally, several datasets were genereated for the studies and analyses presented in this thesis, which we make publicly available. In particular, we provide the datasets of phishing and impostor URLs presented in Chapter 4, as well as the certificates of phishing websites used in the evaluation in Chapter 10 [DD23]. Furthermore, the results of the user studies to compare anti-phishing learning games from Chapter 6, as well as phishing URL categories and RDN notation from Chapter 7 are made available [RD23; Dru23]. We hope, that providing these datasets makes it possible to replicate or confirm our results, and enables future research into improving or extending the anti-phishing defenses presented in this thesis.

Overall, we evaluate anti-phishing methods based on different principles that can be employed independently, but also complement each other in their capabilities and shortcomings, thus highlighting the multi-layered approach that is currently used to reduce the risk of phishing attacks.

1.2. Outline

The remainder of this thesis is split into four main parts: *Foundations*, *Human Factors*, *Certificates of Phishing Websites*, and the *Conclusion*. The foundations include preliminaries, related work, as well as a categorization of impostor domains, which introduce definitions and notation that is required for the following chapters and should therefore be read first. The following parts on human factors in phishing attacks and certificates of phishing websites can be read out of order, though we recommend adhering to the proposed order for the intended reading experience.

Chapter 2 starts with the preliminaries in the **Foundations** part, where we provide a definition of phishing, an overview of the attacks that are of most relevance in this thesis, and a model of the phishing attack process. The preliminaries also include definitions of the URL structure and related terminology used throughout the thesis, as well as an overview of the methods of statistical testing we utilize in our user studies. Additionally, it introduces the main concepts of email security and education, and provides an overview of TLS certificates and their contents.

The preliminaries are followed by related work in Chapter 3, where we present an overview of phishing attacks with an intuition on why the prevention of phishing attacks in general is a complex task. We then focus on the current state of phishing prevention and highlight the particular research gaps addressed in the thesis.

Chapter 4 continues with an overview and analysis of a dataset of phishing URLs collected from various threat intelligence sources, which is used to argue about the content of educational interventions and to train automated classifiers in later chapters. Here, we also define the concept of impostor URLs, which are a subset of phishing URLs that include a reference to a target, thus offering more context for the human-centered and automated prevention techniques presented in following chapters. In addition, we present a simple rule-based classifier to extract impostor domains from our existing dataset, which we then analyze in more detail.

Chapter 5 concludes the foundations part by proposing a new categorization of impostor URLs based on the general structure of URLs. The categorization is evaluated in a user study, where a comparison of the classification accuracies of different subsets of URLs that correspond to different URL categories provides a baseline on the classification abilities of untrained users, and motivates the focus chosen in the educational interventions presented in the following chapters of this thesis.

The second part of the thesis is about the **human factor** in phishing attack. Here, Chapter 6 starts with game-based anti-phishing education, where we compare four different learning games in a user study to evaluate the effect of game mechanics and personalization on classification results. To this end, the chapter begins with a comparison of existing learning games, from which we derive requirements for the new prototypes which are presented and analyzed in the remainder of the chapter. The subsequent analysis of the games in a user study with 182 participants confirms, that

all games were successful in improving the players' URL classification performances, but also reveals that not all URL categories were detected equally well even after playing either one of the games.

To complement the results of the learning games, the study presented in Chapter 7 therefore focuses on URL categories that were not detected well even after playing the games. To this end, we present RDN notation as an alternative URL syntax, which places relevant information for phishing detection closer to the beginning of the URL. We then present the setup and results of a user study with 47 participants to evaluate the differences between RDN and normal URL notation for different URL categories.

A similar design intervention that does not require active user education is the research focus of Chapter 8. Here, we present a study on the effect of four proposed UI changes in email clients on the detection accuracy in an email classification task with 74 participants. Since the chapter focuses on emails instead of URLs, it does not make use of URL categories, and instead presents a categorization of different email phishing attacks. After introducing the newly proposed UIs and email attack scenarios, we present the setup and results of a user study with 74 participants where the proposed UIs are compared.

The final research topic presented in this thesis is concerned with the **TLS certificates of phishing websites**. Here, we begin in Chapter 9 with a detailed analysis of the certificates of phishing websites in two different scenarios. The first scenario represents an automated detection engine or classifier that classifies certificates as either benign or phishing based only on the information available in the certificates (e.g., domain names or organization of the subject), while the second scenario represents a classification with more context information, for example when a user is presented with a certificate and has to decide, if the certificate belongs to a given target, which the user is familiar with.

Based on the findings of the certificate comparison, we take a closer look at the classification of certificates in Chapter 10. The chapter first presents a detection pipeline for classifying certificates on CT logs, followed by two evaluations: the first to test the pipeline capabilities and provide a baseline, and the second evaluation to answer several specific research questions in order to improve classification capabilities.

Finally, the last part concludes the thesis with Chapter 11.

Part I.

Foundations

Chapter 2

Preliminaries

While phishing attacks, the focus of this thesis, are a common type of online attack and therefore relatively well known, they still include several lesser known aspects that are necessary to understand the remainder of this thesis. Therefore, this chapter presents several details and definitions for phishing attacks and additional foundational topics that are of relevance for this thesis in particular. We first present an introduction to phishing attacks in general, including a kill chain model and two specific attack scenarios, followed by the URL syntax and an overview of statistical testing used throughout the thesis, a short introduction to educational concepts used mainly in Chapter 6, basic knowledge on email security required for Chapter 8, and an overview of TLS certificates for Part III.

Parts of this chapter were adapted from [3; 4] and [5].

2.1. Phishing Attacks

Phishing can be defined as "a scalable act of deception whereby impersonation is used to elicit an exploitable action from a victim" (modified from [Las14]). This definition includes several key concepts that are relevant for the methods of phishing detection presented in this thesis. First, phishing is an act of deception, which means it is an attack that is performed with malicious intentions. Here, the usage of impersonation is a key aspect, which implies the existence of a third entity that is not necessarily involved in the attack, but whose credibility is abused by the attacker. Impersonation can be used either of a known or public entity (e.g., a popular banking service, online shop, or company), or an unknown or even fictional entity (e.g., a copyright lawyer). Finally, the goal of the attack is to elicit an action from a user, often to make them enter their credentials in a fake website or execute malware. As such, user action is always an important part of phishing attacks, and we do not consider attacks as phishing that do not require any user action. Phishing attacks can take different forms, including the typical example of a phishing email containing a link to a fake website where account credentials are harvested, but also include attacks using different communication media (e.g., SMS, phone, or social media), or that do not make use of websites (e.g., by distributing malware directly in email attachments).

The actors that always appear in a phishing attack according to the above definition

are the **attacker** (also sometimes called phisher) and **victim**, as well as the entity that is impersonated, which we call the **target** (of impersonation). Since attacks rarely take case in a vacuum, there are often more actors that can influence the success of an attack, including but not limited to AV vendors and other creators of blocklists, browser- and email-client vendors, social media platforms, email or cellular service providers, as well as domain registrars and website hosting providers.

We also use the term **user** throughout this thesis to refer to general users of the Internet, who can for example be potential victims or the target audience for education or design interventions. While we do not further restrict the term *user*, we typically assume a degree of autonomy, which might include the ability to read and understand educational material, or the possession of online accounts that might be compromised in phishing attacks.

The above definition and overview already imply some of the problems in defeating phishing attacks in general: there is a wide diversity of attacks, and even specific attacks can be realized in a multitude of different ways (e.g., hosting a website using compromised infrastructure, a hosting provider, or attacker-owned infrastructure). Similarly, interpreting what constitutes impersonation is a complex problem, a solution to which has thus far eluded the research community (i.e., there are currently no automated classifiers that solve the problem completely).

In the next subsections, we first present a process model for phishing attacks, to provide further details on phishing attacks in general and context to the detection approaches presented in the following chapters in particular, followed by a short overview of the specific attacks that are of importance to this thesis.

2.1.1. Phishing Attack Process Based on Kill Chain Model

Using a kill chain model to visualize the phishing process offers several advantages. In particular, splitting the attack into several distinct steps, with different goals and requirements, makes it possible to consider these steps separately. Since each step in the model is assumed to be mandatory for a successful attack, defenders can use the model to consider strategies to disrupt the chain, thus creating effective protection methods.

The aim in creating the kill chain model presented in the following was to make it as general as possible, without removing any steps that are usually required in a successful attack. While related work on the phishing process exists (see Chapter 3 for more details), we did not find an agreed upon process definition, nor any papers explicitly aiming to create one (see Section 3.1.1). This is also true for the non-academic models of the phishing process, which share similar problems in that they are either too specific or miss steps we assume to be important in the context of this thesis. As such, we present a new process model that aims to remedy these problems and fit the required context.

The proposed phishing kill chain consists of five steps:

1. **Information gathering**: Collection of information on the victim (e.g., method of contact, additional context for spear phishing, ...) and target (e.g., cloning website, understanding login process, ...)

2. **Attack preparation**: Setting up the require infrastructure (e.g., setting up a server and website, requesting TLS certificates, ...)

3. **Attack delivery**: Executing the attack by delivering the payload to the victim (e.g., sending emails or SMS)
4. **User action**: Action of the user that is necessary to complete the attacker's goals (e.g., clicking on a link and entering credentials, executing malware, ...)
5. **Exploitation**: Making use of the results of the action of the user (e.g., stealing account, selling credentials, deleting data on victim machine, ...)

Interestingly, it has been our experience over the last few years, that defenses focus on only few of the steps in the kill chain, in particular preventing the attack delivery and user action, with some work on authentication to reduce the impacts of an attack in step (5). Less work can be found in preventing the earlier steps, even though there is potential (e.g., email address separation to prevent information gathering, see [4]). We also note, that different perspectives on the model can lead to different approaches, e.g., browsers that save a history of visited websites could potentially warn users that visited a phishing website retroactively to lessen the effect of compromise. In this thesis, we mainly focus on steps (2) and (4), aiming to either detect attacks before they are executed, or on empowering users and thereby preventing the user action.

Different steps, depending on the specifics of the attack, can produce different artifacts that are available for analysis. In particular, creating a website, sending out emails with URLs, and requesting certificates are actions that create artifacts of interest to this thesis (URLs, emails, and TLS certificates). We next take a closer look at the two corresponding attack scenarios: website-based phishing and email phishing.

2.1.2. Website-based Phishing Attacks

In this type of attack, the attacker clones the website of a target and sends a link to a user of the target, the victim. In particular, we do not restrict the delivery method to email, other methods might also be possible. The victim then clicks on the link, opens the attacker's website, and interacts with the phishing website as if it was the website of the target. This interaction typically includes entering the victim's username, password, or other private information into the fake website, thus enabling the attacker to impersonate the user to the target or other entities in the future. In this thesis, we are not concerned about specifics of the attack (e.g., circumvention of two-factor authentication, website cloning techniques, etc.), as long as a fake website is involved. Due to the usage of websites, typical artifacts of this attack are an URL (i.e., the link that was sent to the victim), as well as all artifacts that were created for the website, including the domain name (e.g., WHOIS [Dai04] information), TLS certificates, and other resources (e.g., html documents, or phishing kits). We focus on this type of attack in the educational games of Chapter 6, the design intervention presented in Chapter 7, as well as the certificate analyses in Part III.

2.1.3. Email phishing

In this scenario, we consider phishing attacks that utilize emails as delivery method. Emails can, for example, be used in attacks to deliver malware directly as an attachment, or to send a link to a phishing website, as explained above. In these

attacks, the emails are artifacts that can be analyzed in more detail. Here, the sender identities and all checked security mechanisms are of interest, in addition to the content of the email.

We further divide email phishing into the scenarios of mass phishing in contrast to spear phishing. Both scenarios are included in our analysis of email UIs in Chapter 8.

Mass Phishing

Mass phishing, in the context of this thesis, describes phishing where the attacker is not aware of the specific recipients of the email as *bait*. Instead, attacks are prepared using a generic email, for example impersonating a commonly used or well known service, and sending the email to a large number of participants. Though attacks might be targeted, they are also automated, and the attacker typically does not interact with the potential victims directly. This category includes the most basic type of phishing, where attackers send emails to a large amount of possible victims to make them enter their credentials on a fake website or open a malicious attachment.

Spear Phishing

In spear phishing, the attacker tailors the messages to the recipient, thus increasing the chance of success by presenting a more believable email fitting the context expected by the recipient. Here, attackers often make use of public information, for example the email addresses used to initially contact the victims. Furthermore, attacks might consist of several stages, or warm-up phases where the attackers do not include any malicious information to build a first rapport with the victim and make it less likely that the actual attack email will be filtered by the victim's email service provider. Consequently, these attacks typically include less cues that can be used to easily recognize the attack. There are several types of attacks that can follow this pattern, including attacks where attackers, pretending to be interested in job offers, send malware infected files to human resources divisions [Mal16]. For the analysis in Chapter 8, we are also interested in **lateral phishing** (see, e.g., [Ho+19]). In this specific type of spear phishing attack, compromised accounts are used to attack additional victims, typically in the same organizational context as the compromised account. This makes the detection of attacks even more complicated, as attackers are able to remove several common cues of phishing emails which are based on the sender address.

2.2. URL Structure and Domain Names

In this thesis, there is a focus on website phishing attacks, where URLs play an important role. As such, an understanding of the URL structure and a definition of several terms is necessary for most of the thesis.

We base the terminology on the living standard by the whatwg[1] (see Figure 2.1 for an overview). URLs (technically "valid URL strings", we use the term *URL* for simplicity) start with a **scheme**, also sometimes called protocol. In this thesis, we mostly use the `https` scheme, which stands for *Hypertext Transfer Protocol Secure* and indicates a `http` connection over `TLS`. Other popular schemes include `http` and

[1] `https://url.spec.whatwg.org/`, accessed 2022-12-22

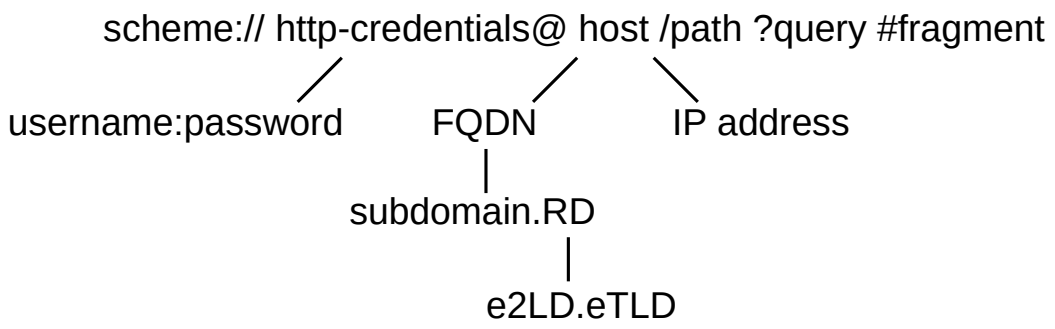


Figure 2.1.: Overview of URL structure and notation.

`ftp`. Note, that while all URLs technically have to start with a scheme, browsers often include functionality to guess the scheme if it is not included in a given URL, for example when users only enter a domain name. The scheme is followed by a colon, and usually two slashes (e.g., `https://`), which we also collectively refer to as the scheme, as the distinction is not relevant in this context. The scheme is optionally followed by **credentials** (we also call them **http credentials** to add context), consisting of a username and optional password separated by a colon, and followed by an `@` symbol which is the main indicator that credentials are included in the URL. The next part of the URL is the **host** specification, which is typically either an **IP address** or a **domain name**. The domain name typically consists of several **labels**, which are separated by dots. We use the term Fully Qualified Domain Name (**FQDN**) to refer to all labels that make up the host's domain name. Due to the underlying hierarchy of the DNS, domain names should be resolved from right to left, with the right-most label called Top Level Domain (**TLD**). Typically, domain names are requested from registrars, which are responsible for specific TLDs. In some cases, the domain name is not registered under the TLD directly, and instead under a previously defined combination of labels (e.g., `co.uk`). In these cases, we refer to the fixed labels as effective TLD (**eTLD**), as they serve a similar purpose to TLDs but are technically not the same due to the larger number of labels. The eTLD and TLD are equal if the eTLD consists of only one label. The first label to the left of the eTLD, which can typically be chosen freely when registering a domain, is called effective second level domain (**e2LD**). We also define the registrable domain (**RD**) as the combination of e2LD and eTLD as one of the most important parts of the URL for phishing detection, as it conceptually defines the *destination* of the URL, i.e., where the URL leads. Since domain names are parsed from right to left, it is possible for domain owners to add labels to the left of a domain in their control. We call all labels to the left of the RD, which can typically be chosen freely, **subdomains**. Note, that our definition of subdomains does not include the RD, and might therefore differ slightly from other common definitions. The host ends with the first slash, after which follows an optional **path**, then **query** and **fragments**. Similar to the FQDN, a path can have several components, which are divided by slashes `/`, and is typically conceptualized as a file path on the host. The query starts with a question mark `?`, and typically consists of key value pairs that serve as additional parameters in evoking the resource at the given path. Finally, fragments are separated from the query by a hash symbol `#` and often serve the purpose of defining a specific location

on the website referenced by path and query. Since the distinction between path, query, and fragment is often not necessary in this thesis, we refer to them collectively as **full path**, even if one or several components are missing (e.g., a URL with only a path but no query or fragments still has a *full path* consisting only of the path). Note, that due to the restriction on symbols in different components of the URL (e.g., the path can not contain a question mark as it would denote the beginning of the query), it is possible to encode characters using what we call **URL encoding** (also sometimes called percent encoding). This encoding makes use of the percent sign `%` followed by a hexadecimal value to define the encoded characters, and while it is typically only used in the full path, it can also be used to, for example, encode an FQDN. A second technique to encode characters in URLs which focuses on domain names are Internationalized Domain Names (**IDN**) using a representation called `punycode`, which was created to allow for non-ASCII characters in the FQDN. While there are several attacks that focus on IDN (see, e.g., [ES18; Hu+21]), these are not discussed in the context of this thesis.

We refer to services, where it is possible to host content (e.g., websites) on the owners infrastructure as **hosting services**. The websites hosted on these providers often share their RD, which we also refer to as the **apex domain** of the hosting provider to emphasize, that it belongs to the provider.

Finally, we refer to the website that is returned for a domain name without specifying a path component as the **homepage** for that domain.

2.3. Statistical Testing and Classification Metrics

The typical process for the studies presented in this thesis is to formulate research questions (RQs) or research objectivese (ROs), which in turn define groups or categories that can be compared to each other. We use the term *groups* for between-subject comparisons (e.g., comparing two groups that played different learning games) and *categories* for repeated-measure (also called within-subject) comparisons (e.g., when comparing pre- to post- test after playing a learning game). To find out, whether the groups or categories differ, we typically first report and compare the means of the metric that is to be compared, followed by null-hypothesis significance testing (NHST, also sometimes "null-hypothesis statistical testing") to test for significance of these differences. While a comprehensive explanation of all methods of NHST we use in this thesis is out of scope for this work, we still present an overview of the general processes, methods, **notation** and convention in this section.

We often report the sample sizes (Notation: N), as well as mean differences (Notation: MD) and standard deviations (Notation: SD) for comparisons and descriptive statistics (see Table 2.1 for an overview of the used notation). In all cases, we attempt to perform parametric tests if possible (e.g., all preconditions are met), and only default to non-parametric tests when parametric testing is not possible. For direct comparisons of two groups (independent) or categories (paired), we make use of Student's t-tests (Notation: t), testing for normality using a Shapiro-Wilk test and using Cohen's d (Notation: d) for effect sizes. If a deviation from normality is detected or other preconditions are not met, Wilcoxon signed-rank tests (Notation: W) are used for repeated measures, with the rank biserial correlation (Notation: r) for effect sizes. No non-parametric comparisons of two groups were performed in this

Table 2.1.: Notation for statistical tests

Variable	Meaning
N	Sample size
M	Mean
MD	Mean difference
SD	Standard deviation
α	Significance level
p	Significance value of statistical test
t	Test statistics for Student's t-test
W	Test statistics for Wilcoxon signed-rank test
F	Test statistics for ANOVA
χ^2	Test statistics for Friedman's test
ϵ_G	Greenhouse-Geisser estimate
d	Effect size by Cohen
r	Rank-biserial correlation coefficient
η_p^2	Partial η^2 estimates of effect size for ANOVA
W	Effect size by Kendall for Friedman's test

thesis.

For more complex comparisons, we use different types of ANOVA (Notation: F), using partial η^2 as effect sizes (Notation: η_p^2). Here, Mauchly's test of sphericity is used, and we apply Greenhouse-Geisser corrections for degrees of freedom if necessary (we use ϵ_G to report these corrections). If the ANOVA confirms significant differences, we perform post-hoc tests using Holm's corrections and again report effect sizes using Cohen's d. For the non-parametric variant of the ANOVA, we use a Friedman's test (Notation: χ^2) in Chapter 8 with Kendall's W (Notation: W) as effect size and Conover's post-hoc tests using Holm's corrections. While the notation for Kendall's W and Wilcoxon signed-rank tests is the same, it is always clear from context whether we refer to the test or effect size.

For all testing, we use $\alpha = .05$ as cut-off value for significance. Note, that we do not always report on assumption checks (e.g., Shapiro-Wilk test) if their outcomes are clear from the context to improve readability, so non-reporting of any checks indicates that they were not violated. All statistical testing in this thesis was performed using Jasp[2].

When comparing classification performances in either human and automated classification tasks, we use the phishing class as positive class and legitimate websites as negative class. We often use the number of False Positives (**FPs**) and True Positives (**TPs**) for comparisons, which denote the number of legitimate websites classified as phishing, and the number of phishing websites classified correctly. False negatives (**FNs**) and true negatives (**TNs**) describe the number of phishing websites classified as legitimate and correctly classified legitimate websites, respectively. We further make use of the False Positive Rate (**FPR**), which is defined as the number of FPs divided by the sum of FPs and TNs: $FPR = \frac{FP}{FP+TN}$, **precision** $= \frac{TP}{TP+FP}$, **recall** $= \frac{TP}{TP+FN}$, and **accuracy** as $\frac{TP+TN}{TP+FP+TN+FN}$. Note, that we sometimes use accuracy

[2] `https://jasp-stats.org/` online, accessed 2023-01-09

when refering to a subset of the sample consisting only of one class, e.g., only phishing URLs. In this case, the definition still holds, but both TN and FP are zero. We also use the term **performance scores** to refer to the accuracy of classifications in the user studies.

Note, that our institution did not have an ethics committee that could have approved the studies performed in this thesis at the time the studies were conducted. Instead, the studies were designed similarly to existing studies with ethical approval. In particular, we were open about the context and goals of the studies, and provided additional information as well as a contact email address for participants in case they had questions or concerns after the study. The studies also comply with data protection policies as discussed with the data protection officer of our institution, by limiting the collection of identifiable information and replacing identifiable information like names and email addresses of participants with unique random tokens before the analysis.

2.4. Education

The human factor, and user education in particular, are key aspects of the research discussed in this thesis, with anti-phishing education making up the major topic of Chapter 6. In the following, we therefore define several terms related to user education, and shortly present a framework that we used to categorize learning objectives for existing and newly proposed games.

There is currently no unified definition of the differences between awareness, education, and training in the security context [ALK14]. In this thesis, we use **awareness** to describe the state of mind that is necessary to detect attacks, i.e. a user who is not aware does not look for an attack in the first place. This stands in contrast to **knowledge** and **skills**, which correspond to the knowledge, respectively skills, that are necessary to accurately detect an attack when looking for it. As an example, a user who is not aware of phishing attacks at all is unlikely to look for cues of phishing emails, and thus likely to fall victim to the attack. Even with awareness of the attack, if users lack the knowledge or skills to differentiate legitimate from phishing emails, they will not be able to detect attacks either. While this example implies, that both awareness and knowledge are necessary in the context of anti-phishing education, we argue that awareness cannot be accurately measured in lab studies, which are the only studies performed in this thesis. It is, however, likely that teaching knowledge and skills also has an effect on the awareness, for example by introducing new cues of potential attacks to the learners (see, e.g., [Cuc+19; Kum+08; SL20]).

We further use **education** to describe interventions that aim to increase the awareness, or teach the knowledge and skills required to detect an attack. This definition also includes **training**, which is education that involves any kind of hands-on practice. We are particularly interested in **game-based education**, which is education that makes use of games. We use the term **learning game** to refer to games that have an educational context, in particular to describe the games in Chapter 6. Similarly, the term **serious game** can also be used to describe games that have a goal outside of an entertainment context. One of the games presented in this thesis offers **personalization** as an option, which we describe informally as changing the game depending on the player (more rigurous definitions exist, e.g., [SS16]).

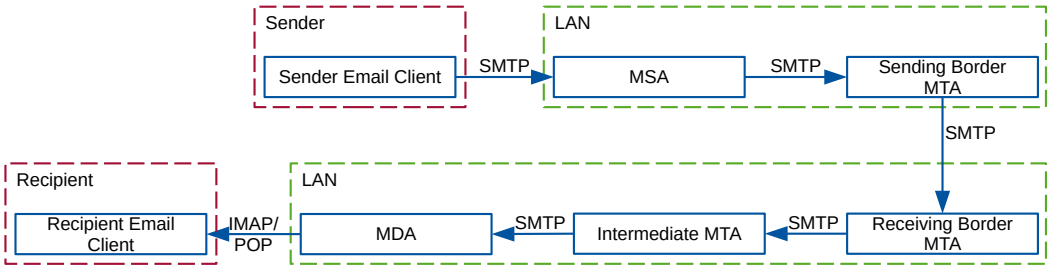


Figure 2.2.: Overview of parties involved in typical email transfer.

A key concept used for the games presented in this thesis is Bloom's Revised Taxonomy (**BRT**), a framework aiming to classify the *learning objectives* of educational interventions [Kra02]. The framework defines two dimensions, knowledge and cognitive process, which together represent the learning objectives. We are mainly interested in the cognitive process dimension, which defines six categories: (1) Remember, (2) Understand, (3) Apply, (4) Analyze, (5) Evaluate, and (6) Create. The knowledge dimension consists of four levels: (1) Factual, (2) Conceptual, (3) Procedural and (4) Metacognitive Knowledge. A general assumption of the BRT is, that higher-order processes implicitly require processes of lower dimensionality to be solved by learners. As such, a focus on higher dimensions is hypothesized to require a better understanding, or at least define a threshold which has to be reached to translate the skills into the real world effetively (see, e.g., [WW13]). We make use of the BRT in Chapter 6, in the analysis of existing games and the resulting requirements for the new prototypes.

2.5. Email Security

This section includes a short overview of the email delivery process using the Simple Mail Transfer Protocol (SMTP) [Kle08], and how sender identities can be verified using various security mechanisms. Note, that the process and mechanisms are presented using several simplifications to focus only on the information that is relevant to this thesis. More information on all methods can be found in the corresponding standard documents.

While emails are typically sent by a **sender** to a **recipient**, these are not the only parties involved in the sending process (see, e.g., [Cro09] for details). In the example depicted in Figure 2.2, the sender first sends the email to a Message Submission Agent (MSA), which in turn first submits it to a Message Transfer Agent (MTA) that sits at the border of the Local Area Network (LAN) of the sending server. This sending border MTA (**sending server**), then passes the message on to a receiving border MTA (**receiving server**), which might in turn forward it to intermediate MTAs in its local network until it reaches a Message Delivery Agent (MDA), from which the message can be downloaded by the recipient, for example using the Internet Message Access Protocol (IMAP) or Post Office Protocol (POP).

A simplified exchange of the SMTP between a sending server (C for client) and receiving server (S for server) can be seen in Listing 2.1 (adapted from [Kle08]). The client first identifies itself using the `EHLO` or `HELO` command, and initiates the sending of an email by specifying the sender in the `MAIL FROM` command. Next, it specifies

```
1  S: 220 foo.com Simple Mail Transfer Service Ready
2  C: EHLO bar.com
3  S: 250-foo.com greets bar.com
4  C: MAIL FROM:<Smith@bar.com>
5  S: 250 OK
6  C: RCPT TO:<Jones@foo.com>
7  S: 250 OK
8  C: DATA
9  S: 354 Start mail input; end with <CRLF>.<CRLF>
10 C: From: "Jones" <Jones@foo.com>
11 C: More content...
12 C: .
13 S: 250 OK
14 C: QUIT
15 S: 221 foo.com Service closing transmission channel
```

Listing 2.1: SMTP example adapted from [Kle08] with highlighted sender identities and SMTP commands.

the recipient of the message using `RCPT TO`, followed by the content of the email after the `DATA` keyword. In this example, the message starts with the `From:` header, which consists of a display name and an email address. In analogy to analog mail, we call the first part of the protocol the **envelope**, as it is only relevant for message delivery, and the DATA part between lines 10 and 12 in the example the **message**. The message typically starts with several **headers**, usually including the `From:`, `To:`, `Subject:` and `Date:` headers.

As can be seen, the sending mail server provides several identities to the receiving mail server in the example below: an **EHLO** identity, the **MAIL FROM** identity, and the **message `From:`** identity. As a distinction between `EHLO` and `MAIL FROM` identities is typically not required in this thesis, we refer to them collectively as **envelope `From`** identity. Email clients typically display the message `From:` identity, either using the display name, email address, or both. In SMTP, none of the identities are verified, making sender spoofing possible when senders simply claim a different existing identity (e.g., setting the message `From:` header to the name and email address of a popular commercial website).

To prevent sender spoofing, several mechanisms can be implemented by the receiving server to verify, that the sending server is authorized to use its claimed identity (see Figure 2.3). Common security methods are DKIM, SPF and DMARC, all of which make use of DNS records to attest the sender's identity. DomainKeys Identified Mail (**DKIM**) makes it possible for sending servers to sign outgoing emails using a signature scheme where the public key is published via DNS [CHK11]. DKIM allows specifying which parts of the message are signed (i.e., only specific headers), and adds a new header to the message that includes the signature, as well as the domain where the public key can be retrieved (`d=` in line 7 in Figure 2.3). The receiving server or recipient can then retrieve the public key using DNS, and verify the signature, thus confirming that the message was indeed sent by the domain claimed in the DKIM header. Note, that the domain used for verification has to match neither the envelope nor the message `From` identities. Sender Policy Framework (**SPF**) allows sending servers to specify which IP addresses should be allowed to send messages on their behalf [Kit14]. Here, the receiving server queries a DNS

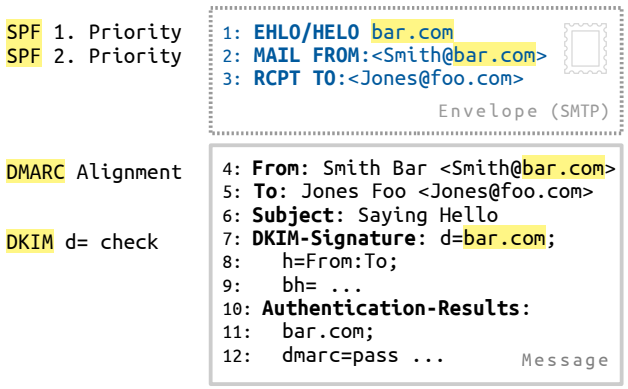


Figure 2.3.: Overview of security mechanisms in emails.

record from the domain which is claimed in the `EHLO/HELO` or `MAIL FROM` domain sent by a sending server, and retrieves a list of IP addresses that are allowed to send emails in the name of this domain. The receiving server can then confirm, that the sending server it is currently communicating with is included in the list, thus verifying that the message was sent using an authorized sending server. As such, SPF can verify the domains appearing in the envelope sender identities, but does not offer any protections for the message `From` header either. To this end, Domain-based Message Authentication, Reporting, and Conformance (**DMARC**) introduces the notion of *identifier alignment*, which refers to the state when at least one domain name verified by SPF or DKIM matches the domain appearing in the email address of the message `From:` header [KZ15]. To make the outcomes of these verification methods available to the recipient (in particular SPF, which cannot be confirmed by the recipient), the **Authentication-Results** header can be added to the email as a part of the message [Kuc19]. This header starts with an identifier of the server that performed the checks, followed by `method=outcome` pairs that describe which verification methods were performed and what the outcome of the methods was (e.g., `spf=pass` for a successful SPF verification).

Apart from verifying the sending server, it is also possible to verify the sender directly. Here, S/MIME and PGP are popular options that allow senders to digitally sign their messages directly, thus providing proof of their identity without relying on the receiving server to perform any checks. Secure/Multipurpose Internet Mail Extensions (**S/MIME**) allows the usage of X.509 certificates to bind a public key to an email address [Ram04]. As with the certificates used in TLS (see Section 2.6), a hierarchy of CAs can be used to verify a given certificate. By embedding the certificate and signature in the email message, the sender's identity can thus be verified by the recipient directly. Similarly, **PGP** (or **OpenPGP**) offers end-to-end encryption and signatures of message content using a separate public key infrastructure [Cal+07]. It is not based on a hierarchy of CAs and instead relies on users to perform the initial verification of public key to email address themselves. Once trust is established, the recipient can verify that a message was authored by the sender by verifying signatures included in the message using knowledge about the sender's public key.

2.6. Certificates

Https allows web servers to authenticate themselves to users based on X.509 certificates using TLS [Res00]. Such certificates are issued by Certification Authorities (CAs) and bind the public key of a web server to the identity of the web server. We use the terms **certificate** and **TLS certificate** interchangibly to refer to the certificates presented by websites using TLS in this thesis. Table 2.2 illustrates how the identity of the web server and its issuing CA are represented in an X.509 certificate and which other fields included in a certificate are of interest in the context of this thesis. Other fields that commonly appear in certificates include, for example, information about the subject's public key and the issuer's signature, as well as a serial number that must be unique per issuer. Note, that the web server's domain name is included in the certificate either in the subject `CN` field or as an entry in the `SAN` extension field.

The CAs used on the web today are ordered in a hierarchy, where CAs on higher levels issue certificates for CAs on lower levels, and the CAs at the lowest level issue leaf certificates for the individual web servers. The certificates of the CAs at the highest level, the root CAs' certificates, are shipped in web browsers and are thus readily available on the client side. The (simplified) verification process of a certificate starts, when a client connects to the web server with https, at which point the web server presents a chain of certificates to the client. The client can then validate the certificates in the chain, starting with checking that the last certificate in the chain was issued by one of the pre-established root CAs and thus obtaining a trusted public key to check the next certificate in the chain. While certificates certainly help in validating public keys, the mere fact that a website is able to present a valid chain of certificates is not a guarantee that the website itself is trustworthy as CAs may follow different policies while issuing certificates. Thus, it is possible that a request for a certificate, e.g., for an intentionally misleading domain name, is indeed signed by a CA if the policy used by the CA to validate the identity of the requester does not include a corresponding check. Indeed, the currently most popular CA *Let's Encrypt* explicitly states[3], that it only performs validation of domain ownership, arguing that it is not responsible to filter out potentially malicious domains. We further find, that it is not the only CA that has issued certificates to phishing websites in Chapter 9.

2.6.1. Types of Validation

There are several levels of vetting a CA can perform before signing a certificate for a subject, that can also influence the amount of information included in the certificate. These validation types represent different levels of trust or effort by the CAs and are briefly introduced in the following. We use the CA/Browser Forum's (CAB) guidelines as reference for the different validation levels [For19].

According to these guidelines, all CAs have to ensure certain qualities regardless of the type of validation, that include basic employee vetting as well as logging and auditing requirements. The CAs also have to ensure that all information that is included in a certificate was verified, taking reasonable steps to ensure correctness. In the context of phishing, it is worth mentioning that CAs are required to maintain a database of "high-risk" names, that are at risk for phishing or other fraudulent

[3] `https://letsencrypt.org/2015/10/29/phishing-and-malware.html` online, accessed 2023-01-25

Table 2.2.: Certificate fields and shortnames used in this thesis.

	Field
Subject:	`Common Name` (`CN`) `Organization` (`O`) `Organizational Unit` (`OU`) `Locality` (`L`) `Country of Residence` (`C`) `Business Category`
Issuer:	`Common Name` (`CN`) `Organization` (`O`) `Country of Residence` (`C`)
Validity:	`Valid From` `Valid To`
Extensions:	`Subject Alternative Name` (`SAN`) `Certificate Policies`

usage. This database has to be checked for each certificate that is issued, and if a high-risk name is found, additional scrutiny on the part of the CA is expected to make sure that the certificate is issued to a valid entity. There are, however, no specific requirements on how "high-risk" names are to be handled in the CAB documents.

Domain Validation

Domain Validation (DV) is the most basic form of validation. Here, the CA only checks that the certificate signing request is valid and that the subject has control over the domain in question (indicated in the `CN` or `SAN` field of the certificate). This might include a challenge, e.g., setting a specific DNS entry or uploading a file with some predefined content. Since no further review is required to validate control over a domain, this process can be automated, e.g., as is the case with the CA *Let's Encrypt*[4].

Organization Validation

Certificates where the CA has asserted the validity of the subject's organization identity are called Organization Validated (OV) certificates. In the CAB documents [For19], this requires more rigorous validation of the subject, beyond simple control of the domain. Verification of the organization identity means, that the issuing CA has to verify name and address of the organization entity, e.g., via consulting the government agency in the jurisdiction of the organization, or a site visit. Additionally, the CA has to verify the authenticity of the certificate applicant, e.g., via a reliable method of communication.

As a result of the organization validation, the CA is able to add organization information (i.e., the `O`, `OU`, `C`, `L` fields) to the certificate. They might also include the

[4] `https://letsencrypt.org/` online, accessed 2023-01-25

CAB policy ID for OV certificates (2.23.140.1.2.2) in the `Certificate Policies` field of the certificate, and must then include the subject field `O` as well as location information (i.e., country and state or province).

Extended Validation

The most thorough validation level is called Extended Validation (EV) and is used to validate the legal entity that controls a website [For18]. Preventing phishing is explicitly mentioned as a secondary purpose, a consequence of the more reliable information included in the certificate. The main difference to OV certificates is, that the process for issuing EV certificates is defined in much more detail and adds some additional requirements. In theory, a CA could therefore issue a non-EV certificate using the EV validation processes.

The guidelines also introduce additional constraints to EV certificates, including the prohibition of wildcard certificates. CAs will also have to look out for high risk certificates, that include websites with the risk of fraud (e.g., websites with an IDN that looks similar to an existing business). An EV certificate must include several fields:

- The subject organization name.
- The subject business category (e.g., private organization or government entity).
- The subject jurisdiction of incorporation or registration.
- The subject registration number (identifying the subject in the registration agency at the jurisdiction of the subject).
- An EV policy identifier that confirms the CA's compliance to the CAB EV documents. This can be specific to each CA.

All steps of the issuance process have to be documented and reviewed before granting the certificate request, all discrepancies have to be resolved.

A client, for example a browser, checking the validity of an EV certificate, has to check for the corresponding policies in the certificate and confirm, that the issuing CA is valid and known to adhere to the EV guidelines.

While EV certificates were supposed to increase the trust of visitors in the owners of a website and phishing prevention was mentioned as a secondary goal, they have somewhat fallen out of favor in recent years, as most popular browsers no longer display any additional information for websites with an EV certificate[5]. This design decision seems to be based on the fact, that users did not notice the absence of the EV indicators (e.g., when visiting a phishing website), thus making them less useful for their designed purpose[6].

[5] `https://groups.google.com/g/firefox-dev/c/6wAg_PpnlY4/m/ygv01NyEAgAJ` online, accessed 2023-01-25

[6] `https://chromium.googlesource.com/chromium/src/+/HEAD/docs/security/ev-to-page-info.md` online, accessed 2023-01-25

2.6.2. Certificate Transparency

Certificate Transparency (CT) is a project designed to lessen the impact of mis-issuance of certificates [LLK13]. CAs have a high level of trust in the hierarchy explained above, and trusting a root certificate implicitly trusts all subsidiary CAs as well, which can lead to a large number of entities that have the capability of issuing trusted certificates. This trust turned out to not always be well founded in the past, due to, for example, the compromise of CA private keys, or mistakenly issued certificates. As such, the goal of CT is to make the issuance of certificates more transparent by publishing all issued certificates publicly, thus making it possible to monitor for illicitly issued certificates.

To this end, certificates are published in **CT logs**, which can be monitored to detect problematic certificates. To encourage CAs and domain owners to publish certificates in CT logs, popular browsers today require the inclusion of the certificates in one or several logs for them to be considered valid[7]. Since the logs are public, append-only, and have some guarantees of being tamper-proof, they can be used to monitor the ecosystem of TLS certificates at a given point in time, or be monitored continually for newly added certificates of websites that are about to be published. We make use of CT logs as a source of early information on phishing websites, since domains using https and therefore a TLS certificate are likely to be included in the logs.

[7] e.g., `https://support.apple.com/en-us/HT205280` and `https://chromium.github.io/ct-policy/ct_policy.html` online, accessed 2021-01-06

Chapter 3

Related Work

The research area concerned with phishing attacks is well established, with a large number of publications on different detection and prevention approaches, analyses of phishing attack artifacts, and studies aiming to understand the human factor. In this chapter, we provide an overview of the topics that are of relevance to this work, thus setting the context for the thesis, with a particular focus on research gaps. We start with analyses about phishing attacks in general, with a focus on the phishing process and artifacts produced in website based phishing attacks. Next, we present work that aims to understand the human factor of phishing attacks. In Section 3.2, we continue with phishing attack prevention and detection approaches, with a particular focus on anti-phishing education and automated detection using the certificates of phishing websites.

The part about anti-phishing learning games in Section 3.2.2, as well as Section 3.2.5 about the automated detection based on certificates were previously published and adapted from [1] and [5].

3.1. Phishing Attack Analyses

Phishing attacks have been studied for several decades, resulting in papers that document the changes in attacker behavior and techniques over time. In this section, we attempt to give an overview of the state of the art on phishing attack analyses by presenting related work on the technical details, process and artifacts of phishing attacks, thus providing a foundation for our work and a comparison to the kill chain model presented in Chapter 2. Subsequently, we present related work that focuses on the human factor in phishing attacks, including the URL parsing capabilities of potential victims in website phishing attacks.

3.1.1. Taxonomy, Process and Life Cycle

The definition of phishing attacks, as well as the general attack process described in Chapter 2 deliberately omit specifics regarding the attack, in particular about the delivery method, payload, and goal of the attacker. Here, several taxonomies of phishing attacks exist that attempt to provide comprehensive overviews on specific or general attacks. For example, Aleroud et al. determine seven different delivery

methods, including the typical phishing email, but also instant messaging, social networks, mobile devices, and VOIP [AZ17]. They further differentiate different attack techniques by, for example, distinguishing general email or website phishing attacks from spear-phishing. There are also several attempts to create a general model for phishing attacks, for example by Alkhalil et al., who define a process of four steps which differs from our process model mainly in that it does not include the user action as a distinct step [Alk+21]. Similar process models are provided by other authors as well, in that the attack is split into three to four main phases (the first two steps are sometimes combined), including reconnaissance and setup, followed by attack delivery and exploitation (e.g., [Abr+21; MX22]). We used these process definitions as a foundation for our own model (see Section 2.1.1), but added the user action as an additional step, as it more closely matches our definition of phishing attacks, and showcases the importance of user education in preventing attacks. These taxonomies also highlight the diversity of the phishing ecosystem, and show that even when focusing on popular methods (e.g., email or website phishing) and providing adequate preventive measures, attackers have been able to circumvent these measures in different ways. As such, while we aim to provide robust indicators for currently common phishing attacks in this thesis, we do not cover all possible attacks, nor do we claim to provide general protection measures that will remain relevant even in the future.

An interesting result of previous work regarding the life cycle of phishing attacks is the duration of attacks, which can be measured using different methodologies and events, but generally describes the time frame in which a phishing website is available and poses a potential danger to users. Here, Oest et al. found an average attack duration of 21 hours in 2020 based on the time difference between the first and last victim of the phishing website, while it took an average of 8 hours for the website to appear in a blocklist [Oes+20a]. These results indicate a change from previous research, where Han et al. found a duration of approximately 8 days in 2016 using honeypot servers to gain detailed insights into attacker behaviors [HKB16], and a duration of around 3 days reported by Moore et al. in 2009 based on the difference between the inclusion of a website on a blocklist to its takedown [MCS09]. In general, while these short time spans indicate fast responses to newly created phishing websites, they also motivate the usage of automated and timely interventions against phishing attacks, as even short delays in a response can leave users unprotected. More recently, we measured the duration of attacks and complete phishing campaigns using timing information extracted from the metadata (e.g., certificates), and content (e.g., images) of phishing websites [2]. We found, that almost 28% of certificates were requested within 24 hours of a website appearing on a blocklist, though the average time-to-blocklist was much higher which can most likely be attributed to compromised or benign infrastructure. The analysis of phishing campaigns indicates, that patterns exist in the domain names of some phishing websites which could be used to detect newly created websites belonging to a known campaign for an average of 12 days after the first attack was observed. These results motivate the usage of automated detection approaches, including the approach presented in Chapter 10, where potential phishing websites might be detected and verified based on similar previously seen attacks.

Attackers have been found to employ several strategies in the past that makes the detection of phishing websites more complicated. Apart from diversifying the

attacks, for example based on hosting infrastructure, they have also been found to employ website cloaking, a technique that aims to prevent security researchers and automated detection tools from reaching the malicious phishing website. Cloaking is usually performed depending on information such as the IP address, user agent, or JavaScript capabilities of the client, and is employed to present a different, benign view to potential anti-phishing bots or researchers (see, e.g., [Zha+21; Oes+19; Oes+18]). In an analysis of these common cloaking techniques, Oest et al. found that the evasion was effective, in that it significantly decreased the likelihood of websites being taken down or included in blocklists [Oes+19]. Interestingly, these evasion strategies might also be used to prevent phishing attacks, as was shown by Zhang et al., who mutate http requests to make them look like anti-phishing bots, thus triggering the cloaking mechanism of the website and removing any malicious content [Zha+22]. The techniques discussed in this thesis aim to avoid the effect of website cloaking, as they either focus on user action, where cloaking is usually not employed or results in removing malicious content, or work on URLs or domain names directly.

3.1.2. Phishing URLs and Domain Names

The majority of approaches presented in this thesis focus on phishing URLs or domain names, which have been analyzed in the past. Due to the large amount of papers on URL-based automated phishing detection, there are several descriptions of phishing URL datasets based on the features that can be extracted from URLs. We present an overview of these analyses in the following, and take a closer look at phishing URL categorizations based on different manipulation techniques in Section 3.1.4.

Common lexical features in URL classification tasks include the length, number of occurrences of several characters (e.g., dots, or slashes), inclusion of pre-defined keywords, and whether the host is an IP address [Das+19]. For example, Jeeva et al. compare the occurrence of features in benign and phishing URLs on a comparatively small dataset of 1400 URLs and find that they differ significantly, making a rule-based detection approach feasible [JR16]. More recently, Sánchez-Paniagua et al. compare phishing URLs to legitimate homepages and login URLs, and find that the two legitimate collections differ in their distribution of features, leading to drops in performance when classifying login URLs using classifiers trained on homepage URLs [Sán+22]. These findings indicate, that choices concerning the datasets can lead to significant differences in feature occurrences, making the decision about representative collection sources an important first step in analysis and classification tasks. We therefore aim to choose samples that are representative for the corresponding settings throughout this thesis, and use common features to describe our dataset of phishing URLs in Chapter 4 to note potential biases.

Previous research has also looked at the occurrence of a target name in phishing URLs. Here, McGrath et al. found that target names appeared in more than 50% of PhishTank URLs, and more than 75% of URLs collected from MarkMonitor in 2007 and 2008 [MG08]. For PhishTank, they report a larger number of URLs with the name in the full path than FQDN, typically by including the complete RD instead of only the e2LD. Oest et al. define four categories of deceptive URLs, differentiating between the target or a misleading keyword appearing in the path, subdomain, or RD [Oes+18]. They found, that most of their URLs collected from the first two

quarters of 2017 did not fit any of the categories (61.65%), followed by RD (21.41%), path (10.57%) and subdomain (6.27%). This indicates a change in trend from Garera et al. in 2007, who based their data on URLs collected from Google Safe Browsing (GSB) and found inclusion of the target in the path, particularly while using IP addresses as host, to be the most popular (50.63%), followed by inclusion in the FQDN (46.46%), and only a small minority (2.92%) of URLs with no reference to the target at all [Gar+07]. These results are related to our definition and extraction process of impostor URLs in Chapter 4, which also include a reference to a target.

Another point of diversity in the phishing ecosystem is the infrastructure used to host malicious websites. Here, different methods are available for attackers, including self-hosted, compromised, or public, and even free, web hosting infrastructure. To differentiate between the different types of infrastructure, several different automated approaches have been developed in the past. Le Page et al. designed a classifier to distinguish between maliciously created and compromised infrastructure based on features extracted from the domain and meta data including whether it was archived in the past, domain rank and DNS [Le +19]. Using the classifier to analyze phishing attacks over a period of three years starting from 2016, they found that the majority of 73% of websites were compromised, with the remainder being created by attackers. Similar studies were performed with different classifiers and datasets, for example by Maroofi et al., who focused on publicly available feature and datasets and found a larger amount of maliciously created domain names of 58% in their manual labeling process [Mar+20]. De Silva et al. further include a category for public hosting, resulting in three labels, and analyze malicious domains in general. They find that public hosting makes up a large amount of malicious domains (46.5%), with the remainder being more often compromised (65.5%) than registered by attackers (34.4%) [De +21]. Taken together, these studies indicate that the creation of ground-truth labeling of compromised domain names is a complex task, and that the results can greatly vary based on these labels. We attempt to avoid this problem by introducing impostor domains in Chapter 4, which are easier to label based on only the information in the domain name, and additionally more accurately reflect the challenges we attempt to solve in this thesis.

3.1.3. The Human Factor in Phishing Attacks

Due to the active role of the victim in phishing attacks, researchers have previously set out to better understand why users are susceptible to phishing attacks, and which factors influence this susceptibility. In previous studies, several hypothesis have been tested, that aim to provide a basis to understand and ultimately prevent phishing susceptibility. Here, Dhamija et al. presented a user study in 2006, where their participants classified 20 websites as either phishing or legitimate [DTH06]. They found, that few users actually made use of robust indicators in the browser, like the URL in the URL bar, instead they put more focus on website content. Similarly, Downs et al. conducted a role-playing study with 232 participants, who were not primed on phishing and asked to interact with five emails and four URLs, followed by a short quiz about general security-related knowledge and a questionnaire about perceived consequences of phishing attacks [DHC07]. Their results indicate, that higher degrees of security-related knowledge were correlated with better scores when detecting phishing emails and URLs, while perceived consequences did not predict

the outcome. These results motivate the application of anti-phishing education, which aims to introduce and explain robust indicators that users can focus on to detect phishing websites, thus providing the necessary knowledge.

Other studies also include psychological aspects of phishing attacks, e.g., using the principles of persuasion by Cialdini [Cia06]. As an example, Oliveira et al. studied the differences of susceptibility between younger and older participants and found that older people were generally more likely to click on links in phishing emails, and that different persuasion techniques had different click-rates for the different age groups [Oli+17]. A model of the decision process in email-based phishing attacks is provided by Wash, who interviewed 21 IT experts about their experiences when detecting phishing emails, and identified a three-stages process of sensemaking, suspicion, and decision that was applied by the majority of experts [Was20]. Wash identified the transition between the first and second stages as of particular importance when detecting phishing attacks, and identified nine cues that triggered this shift in the experts. The study therefore provides a model for awareness as noticing cues when making sense of an email that trigger a shift to suspicion.

Simulated phishing attacks can give insights into how well users detect phishing attacks in a more realistic context in practice (i.e., by requiring awareness, similar to a real attack). Here, Williams et al., present an analysis of phishing emails impersonating fictional organizations according to different principles of persuasion sent to more than 60,000 participants, and found that click rates varied between 6% and 35% and was higher for emails that induced a sense of urgency or made an appeal to authority [WHJ18]. Note, that the usage of fictional organizations as targets might have influenced the study. A closer look at spear-phishing is performed by Burns et al., who first define different targeting levels, and then tested different interventions in a user study with 260 participants [BJC19]. They found, that their highly targeted attacks had an initial hit rate of 70%, which decreased in a second round, in particular for users who received training highlighting their individual loss. This research emphasizes the importance of awareness as a first step to detect phishing attacks in the real world. Since we did not test the interventions presented in this thesis in the real world, for example by using a simulated phishing attack, the effect of the proposed interventions on awareness is open for future work.

A further concern about the human factor we address in the studies presented in this thesis is the difference between known and unknown services in classification tests. It stands to reason, that participants behave differently for services they are not familiar with, leading to differences between study and real life (e.g., users would likely simply ignore or instantly delete emails about services they are not familiar with) as well as to differences between the services they know or do not know in user studies (e.g., users are more likely to recognize the benign URLs of services they use). Here, several previous studies have analyzed the effect of service familiarity on the classification outcome, with varying results. Gopavaram et al., found that familiarity had an effect on classification accuracy for legitimate websites, but not for phishing [Gop+21]. Similarly, Wang et al., found that familiarity had a significant effect on classification confidence, leading to overconfidence, but not on accuracy [WLR16]. We found the differences between known and unknown services to be significant in all of the studies presented in this thesis. As such, ignoring or failing to report on the familiarity of services in classification tasks might make the reproduction of previous studies more complex, as services appearing in the test are

typically selected for a certain population (e.g., the Bank of America is relatively unknown in Europe). We furthermore found in a related study, that players of learning games interact differently with URLs of services they are not familiar with, indicating potential differences during the general interaction with interventions as well (see Chapter 6). As such, we include a questionnaire on service familiarity in all of our studies, and attempt to correct for unknown services when evaluating the results of the classification tests where possible.

3.1.4. URL Categories and User Classification Results

Categories of phishing URLs can be created with different goals in mind, for example a focus on detection, evasion, or education. Previous work has defined different URL categories based on the manipulation techniques that were applied, or whether and where the target appears in the URL (see, e.g., [Rey+20; Oes+18; Rob+19]). We collect and combine these methods in an attempt to provide a more comprehensive categorization in Chapter 5.

Previous work has also set out to study how different categories of URLs differ in their detection rates by users. A general study of URL reading capabilities by Albakry et al. compared URLs of four categories, and finds that participants often assume the URL leads to any recognizable entity that is referred to in the URL [AVW20]. The work by Reynolds et al. defines 13 different categories of phishing URLs, though some of them overlap or are not clearly defined, and tested differences between them in a user study with 94 participants [Rey+20]. They found significant differences, with long subdomain URLs being the most difficult to detect, while participants had high accuracies classifying typosquatting URLs. Taking a closer look at subcategories of typosquatting, Spaulding et al. define 7 categories based on omission, swap, replace, and insert operations, and tested how the category effects classification accuracy in a user study with 34 participants classifying 200 URLs [SUM17]. They found, that character-omission typos and random substitutions were particularly difficult to detect accurately in their study. However, they also note that these results were likely influenced by the participants' familiarity with the modified domain name, which they did not correct for. The study presented in Chapter 5 of this thesis aims to reproduce and extend the results above by evaluating a new categorization of phishing URLs in a user study where service familiarity is measured and corrected for.

Taking a more detailed look at a specific manipulation technique, Roberts et al. analyzed different subcategories for phishing URLs that change the TLD in a user study with 249 participants [Rob+]. They compared generic TLDs with country-code and common TLDs and found, that generic TLDs which were chosen to semantically fit the target decreased the detection accuracy for target embedding URLs compared to country-code or common TLDs. In a second step, they found that users were unable to differentiate domain names using a gTLD that was controlled by the target from randomly chosen different gTLDs. These results indicate, that phishing URLs with generic TLDs that semantically fit the context should potentially be regarded separately from common TLDs when categorizing phishing URLs. Note, that we did not make this distinction in this thesis, however, in order to avoid having to automatically decide when a gTLD fits the context for randomly generated URLs.

An additional category of URL manipulation are IDN homograph attacks, where

characters in the URL are replaced by visually similar or even identical characters using IDN. Here, Thao et al., analyzed how the degree of visual similarity between the original and replaced character affects classification performance in a user study with 2,067 participants, and found that, somewhat unsurprisingly, performance decreases with increasing visual similarity [Tha+19]. We do not include IDN homograph attacks in this thesis, as support for IDN domains has been restricted by several popular browsers (see, e.g., [Hu+21]), and their occurrence in phishing attacks seems to be low (see Chapter 4).

3.2. Phishing Prevention

The previous section defined several of the technical details, which translate into the challenges that phishing prevention techniques have to address. In this section, we take a closer look at those prevention techniques, starting with a general overview of promising approaches. Next, we present past research on anti-phishing education, and provide a first comparison to the learning games presented in Chapter 6. Afterwards, we include related work for the proposed changes in URL syntax in Chapter 7 and email UIs in Chapter 8. Finally, automated phishing detection techniques are compared in more detail, with a focus on the current state of detection using TLS certificates.

3.2.1. General Approaches

The larger research area that is concerned with phishing prevention in general can be split into smaller areas based on the type of attack that is prevented and the prevention mechanism itself. Preventive measures can focus on the automated or reactive detection of phishing attacks, preventing the user action, or even preventing the abuse of stolen credentials.

Possibly the most commonly employed preventive measure against phishing websites are blocklists. Examples of blocklists are Google Safe Browsing (GSB)[1], which is integrated into several popular browsers, or PhishTank[2], a community-curated repository of phishing URLs. While blocklists offer low false-positive rates, an important requirement for any real-world detection system, their reactive nature also comes with several shortcomings. Here, Oest et al. found that the duration between attack execution and inclusion in a blocklist currently leaves a *window of opportunity* of up to several hours, where victims are unprotected [Oes+20b]. The existence of this window of opportunity has been known for a relatively long time, as demonstrated by previous studies [She+09]. Research has therefore turned towards alternative methods that can be used to provide immediate protection methods, particularly in the form of automated detection (see Section 3.2.4).

One approach to prevent phishing is the usage of scoped credentials in authentication, as is for example the case with FIDO2[3] (previously called U2F). Here, the origin of a given website is relayed to an authenticator device, which then selects credentials corresponding to this origin, thus preventing the user from entering their credentials on an unrelated website. While the idea is promising and has the potential

[1] `https://developers.google.com/safe-browsing/v4` online, accessed 2023-01-24
[2] `https://www.phishtank.com/` online, accessed 2023-01-24
[3] `https://fidoalliance.org/fido2/` online, accessed 2023-01-24

of preventing a large amount of attacks, U2F is currently not widely available (see, e.g., [Alq+20]), nor well understood by users (see, e.g., [Las+21; Lya+20]). It is also unclear how to handle fallback methods (see, e.g., [Kun+21; Lya+20]), which can reduce or completely remove the benefits of strong authentication in practice. Wiefling et al., analyzed how risk-based authentication (RBA) can be used to prevent the abuse of stolen credentials [WDL21]. They found that RBA can be successful in preventing credential abuse in phishing attacks, but success rates depend strongly on the features used by the system.

Less common are approaches focusing on other steps of the kill chain. Here, we proposed email address separation as a measure to prevent attackers from gaining the information needed to attack victims (i.e., prevents attack in the first step of the kill chain) [4]. The proposed scheme requires users to make use of different email addresses for different services, which makes leakage of addresses associated with higher value accounts less likely, thus reducing the chance of a phishing attack being sent to the correct address.

As a complementary approach to the above, awareness and education training can have a positive effect, which we discuss in more detail in the next section.

3.2.2. Anti-Phishing Education

In the security domain, phishing attacks are uniquely qualified for educational research, as they include a user action to be successful per definition. Preventing this user action by teaching the necessary knowledge and raising awareness is therefore the goal of many different approaches and interventions, which we present shortly in this section.

An overview of educational material is provided by Franz et al., who identified 64 publications in a literature review and distributed them into the four categories *education*, *training*, *awareness-raising*, and *design* [Fra+21]. They find a lack of interventions that require no or little effort by users but are still effective in increasing the classification accuracy of their users. The fact that friction between users' intended action and security is created by current approaches has also been explicitly criticized in the past. For example, Sasse argues that usable security should focus on the users' goals, and reduce friction towards that goal when implementing security measures [Sas15]. Similarly, Cranor presents a framework that can be applied to understand the success and failure of humans in security-critical tasks, but notes that the ultimate goal should be to remove the possibility of failure wherever possible[Cra08]. We attempt to work towards that goal by supplying relevant information to users without impeding their primary tasks or making decisions for them with the interventions presented in Chapters 7 and 8.

Still, as long as technical measures are insufficient at removing phishing attacks completely, education can serve as an important part in a holistic prevention strategy, as it has been shown to be effective in improving the phishing detection capabilities of users in lab studies as well as simulated attacks. Notable examples for anti-phishing education include the study by Kumaraguru et al., who tested an embedded email training method called PhishGuru with more than 500 participants [Kum+09]. In their field study with a simulated phishing attack, participants that received training performed significantly better than the control group, and the improvement remained after roughly one month even without retraining. Similarly, Silic et al. compared

gamified training to non-gamified training in a field study with 488 participants, and found that gamification significantly improved detection rates compared to both normal training and the control group [SL20]. Both studies demonstrate, that education can improve users' stance towards phishing attacks in practice.

Of particular interest to this thesis are educational approaches using game-based learning to improve learning and foster motivation and engagement. Games provide consequence-free environments where learners can experiment and make mistakes, and thus, games allow for graceful failure and active learning [PHK15]. Different topics for anti-phishing learning games are possible, each with potential advantages, as researchers struggle to address the evolving threat. Here, URL classification is a common topic for the games [11], as URLs can not be chosen freely by a phisher and are thus a robust proof of a website's origin, are generally available for phishing attacks that use websites, and are further a common element users encounter when using the Internet. Anti-phishing Phil is an early example of an anti-phishing game, that teaches conceptual and procedural knowledge, followed by levels where players have to classify URLs into benign and phishing URLs [She+07]. The game was evaluated using a pre- and post-test setup, where 20 websites (ten benign and ten phishing) had to be classified and the confidence about each classification had to be rated. Sheng et al. compared the game to other types of existing educational materials and found, that participants had higher scores and confidences after playing the game. No Phish is a second notable approach to game-based anti-phishing education [Can+15], which requires binary classification (phishing or benign) similar to other games. However it also includes a different type of level where users are instructed to select the RD of URLs, which requires a different cognitive process and makes guessing more difficult. In Chapter 6, we present a more in-depth analysis of related work on game-based anti-phishing education, and derive design goals for new prototypes to extend this idea by testing the effect of more complex game mechanics, as well as personalization.

3.2.3. Design Interventions

Since education has been shown to have a positive effect on phishing detection, but has not yet yielded a solution to the phishing problem in general, a different but related field of research focuses on supporting the user to make better decisions, and raise awareness by changing the design of client software. Here, deciding which information is presented to users in websites or emails, and how the information is presented has been shown to have significant effects on classification performance.

Active warnings have been proposed and studied in the past to deter users from visiting unsafe websites. For example, Felt et al. tested alternative browser designs to warn about unsafe SSL connections, and found that an opinionated design that strongly suggests a safe action prevented more users from visiting the unsafe website than any of the other warnings that focused on explaining the threat and associated consequences [Fel+15]. Deterring users in general has, however, been criticized, as it does not make a system more usable, and enforces a technical opinion by preventing users from achieving their primary goals [Sas15]. Furthermore, the effect of active warnings has been shown to diminish over time due to warning fatigue [And+15]. Instead, design interventions that highlight important information and *nudge* users into secure behavior might provide an acceptable compromise for real-world systems.

One such example is the URL and how it is presented in the browser, where

previous studies have mainly focused on highlighting relevant parts of the domain (i.e., the RD). Lin et al. found as early as 2011, that highlighting by itself is not effective, since more than half of the participants did not focus on information in the URL bar even with highlighting, and most users did not understand the URL when explicitly told to look at it [Lin+11]. Similarly, Thompson et al. tested the effect of several browser UI alterations on a single phishing website, and found that none of the methods, including URL highlighting and only showing the RD, significantly improved the detection accuracy of users [Tho+19]. In a user study with 411 participants who classified 16 website screenshots twice, Volkamer et al. found, that instructing users to look at the URL bar significantly increased their accuracies, and that pruning the URL to only show the RD resulted in better performances than URL highlighting in this case [VRG16]. In Chapter 7, we propose and evaluate URL rewriting as an alternative highlighting method, that might offer advantages when users look at the URL bar.

As for email UIs, several studies have set out to design alternative client UIs. Nicholson et al., investigated the effect of "nudges", which include the highlighting of sender name and address on phishing classification outcomes [NCB17]. They found that sender highlighting in particular was successful, in that it significantly improved the detection of phishing emails without introducing a significant bias towards legitimate emails. In Chapter 8, we extend this study by introducing additional information, a different focus for sender highlighting, as well as a set of well-defined categories of phishing emails corresponding to different attack scenarios. Highlighting information about encryption and digital signatures is more common (see, e.g., [TL20]), and similar to browser warnings about security indicators. While we include the highlighting of security information in Chapter 8, our focus is not on the inclusion or state of these mechanisms, but on how their statements about the validity of an email's sender influence the participants in their classification task. An interesting approach by Dwyer et al. highlights the path the email took until arriving in the recipients inbox by highlighting the geographical locations of servers on the path on a map [DD10]. While they found that 82% of 100 randomly sampled phishing and spam emails might have been detected due to suspicious paths, they did not evaluate the UI in a user study. A different study by Volkamer et al. integrated URL highlighting technique into emails, and found that it had a significant positive effect on classifications of emails with links [Vol+17]. This technique does, however, not prevent phishing attacks that make use of benign (e.g., hosting) services for attacks, nor does it help in attacks that do not use website phishing. As such, we present an additional complementary approach in Chapter 8, where we discuss different options to highlight the sender of an email and the information used to validate this sender.

3.2.4. Automated Phishing Detection

To improve upon the reactive nature of blocklists, researchers have been working on automated methods to detect phishing attacks. These generally aim to classify URLs, emails, or websites actively as soon as they are seen, thus removing the window of opportunity of blocklists. An overview of automated phishing detection is provided by Das et al., who defined several constraints for automated detection in the security domain, reviewed literature on phishing URL, website, and email detection, and analyzed the used features, datasets, and learning methods [Das+19]. They

found, that few studies address their proposed constraints, in particular regarding the applicability of proposed detection methods regarding low FPRs, real-time capabilities, and active attackers in the real world.

A notable example for large-scale URL classification is the study by Whittaker et al., who evaluate a classifier based on URL and website feature and trained on a comprehensive dataset of millions of URLs, which was integrated into GSB to automatically update the blocklist [WRN10]. Their experiment demonstrates, that classification at scale is generally possible but does result in FPs and cannot prevent all attacks as even automation still leaves a window of opportunity for attackers. Other approaches include features about the hosting infrastructure of websites, as is the case with Kim et al., who combine features from the URL with its domain name, resolving IP address, and name server [Kim+22]. They show, that their network-based inference approach using the proposed combination of features makes the classifier less susceptible to evasion techniques that change parts of the phishing URL to make them seem benign.

Compared to URL detection, the detection of phishing emails profits from the context provided by the email content and sender information, in addition to the inclusion of URLs or attachments. To this end, natural language processing techniques can be applied, as shown by a recent survey by Salloum et al. [Sal+22]. Ho et al. propose a classifier to detect lateral phishing, an attack where compromised email accounts are used to attack additional victims, thus lending the sent email credibility [Ho+19]. Their proposed classifier uses features based on the recipients and sender history, inclusion of pre-defined keywords, as well as URLs included in the email, and achieved a recall of 87.3% while retaining a low FPR of $3.6 \cdot 10^{-6}$. While we do not focus on automated email detection in this thesis, the above results indicate the diversity of attacks in this area, and how even sophisticated systems trained on large datasets struggle to provide comprehensive protections at low FPRs.

In all, while many automated phishing detection approaches were presented and evaluated, few seem to be deployed in practice, which might be due to differences between the evaluation and real-world settings.

3.2.5. Phishing Detection Using Certificates

In the following, we take a closer look at automated detection approaches that make use of information from the certificates of the website. For example, Mohammad et al. [MTM14] include the usage of https, as well as information about the certificate's issuer in their feature set. The corresponding dataset has also been made public and been used in several additional studies.

Compared to the classification of URLs, it is likely that certificates offer less information. Even though there are several additional fields in certificates, we found that they are often very similar or identical for certificates issued by the same issuer (see Chapter 9). Still, the domain names embedded in certificates remain as a promising factor for phishing detection. However, the domains in certificates include much less information than a complete URL, as they do not include path information, and sometimes even do not include all subdomains when wildcard certificates are used. Even more problematic is the case of phishing websites hosted on compromised or benign hosting infrastructure, where the certificate was not requested with malicious intent. As such, we analyze the impact of focusing on a subset of domain names,

which were likely created with malicious intentions, on the classification performance in Chapter 10.

Still, several approaches that focus only on information from certificates for phishing detection exist. In 2015, Dong et al. [Don+15] proposed a number of certificate features of phishing websites, including the existence, length, and relationship of different fields in certificates. They compared a number of classifiers, trained on certificates collected directly from known phishing and benign websites between late 2012 and 2015, and found that random forest (RF) classifiers achieved the highest precision. To our knowledge, the first proof of concept for using CT logs as basis for phishing website classification is the Phishing Catcher[4], available at a GitHub repository. As for peer-reviewed research, Scheitle et al. [Sch+18a] noted in 2018 that the CT logs might offer a new perspective for phishing detection. In a preliminary look at the logs utilizing regular expressions, they found a large number of certificates (more than 125,000) that likely impersonated a small number of popular services, but did not include an in-depth analysis. Torroledo et al. [TCB18] trained a phishing classifier on certificates only, aiming to detect differences in the legitimacy of phishing and non-phishing certificates. Using a highly imbalanced dataset, they achieved a precision of around 90%, which we were not able to reproduce or verify in our experiments. Fasllija et al. [FEP19] trained a classifier on domain names extracted from full URLs, arguing that it would be able to perform classification on CT logs as well. However, they did not perform such an evaluation. Recently, Sakurai et al. [Sak+20] proposed a classifier that is specifically created for the task of CT log classification. The classifier is based on regular expressions extracted from known phishing websites, and achieved promising results on certificates collected from Censys[5]. However, the static logic based on regular expressions is neither able to detect new phishing campaigns with unknown domain name patterns, nor is it suitable for detecting spear-phishing campaigns, which do not create large amounts of similar domain names in the first place.

A fundamentally different approach was presented by Lin et al., who present a classification pipeline that first detects prominent logos on a given website and compares them to a set of possible target logos [Lin+21]. The authors demonstrate how it can be used to classify certificates in CT logs where it achieves a high precision of 93.63%, but also that manual verification of positives was necessary, as more than half of the true positives identified by the authors did not appear in their ground truth labeling. While the visual approach has the advantage of providing additional context compared to using only information from the certificate, it also comes with several drawbacks, as it requires access to the phishing website to be effective, thus making it susceptible to cloaking techniques or websites that require a path component to display the phishing website, and requires more download bandwidth and processing power compared to looking at only certificates.

In Chapter 10, we present a pipeline which makes it possible to evaluate new and existing classifiers on actual CT log data, including the possibility of classifying certificates as soon as they are added to the logs. We also evaluate a number of feature-based and deep learning classifiers that only make use of information included in the TLS certificate as an alternative to the existing approaches. Finally, we present

[4] `https://github.com/x0rz/phishing_catcher` online, accessed 2023-01-25
[5] `https://censys.io/` online, accessed 2023-01-25

evidence that a focus on particular sets of certificates in the training phase can have an impact on the classification performance of the classifiers.

Chapter 4

Phishing and Impostor URLs

As a comparatively robust factor to detect phishing websites, URLs take up a central role in many of the following chapters. This chapter presents our dataset of phishing URLs (**DS-Phish**), which was used as a foundation for the topics taught in the games presented in Chapter 6, as well as the classification task explained in Chapter 10. We present the collection process and an analysis of the URLs in the dataset, including different obfuscation techniques in addition to the general structure of phishing URLs. Next, we compare the phishing URLs to a sample of benign URLs collected from popular websites to highlight similarities, but also common differences between the two classes. Finally, we focus on a subset of the phishing URLs with the *impostor* property, which describes URLs that include a reference to a benign target in the URL, thus adding some measure of misdirection to the URL itself, and extract a second dataset of impostor domain names (**DS-Impostor**).

Contributions: The main contributions of this chapter are the creation and analysis of the **DS-Phish** dataset, as well as the definition and rule-based classification of impostor URLs. The **DS-Phish** dataset of phishing URLs was created based on two methodologies, one of which, resulting in 3,069,231 URLs, was created in collaboration with Arthur Drichel and Justus von Brandt and has been previously published in [1]. The analysis of phishing URLs and comparison to benign URLs is a new contribution of this thesis. For impostor URLs, the definition and baseline classification of impostor domains were joint work with Tobias Johnen and formalized in his master thesis [Joh22]. Both definition and the source code for the rule-based classification were slightly adapted for this thesis. The analysis of impostor domain names resulting from the rule-based classification of the **DS-Phish** dataset is a new contribution of this thesis.

4.1. A Short Analysis of Phishing URLs

We begin by presenting details on our dataset of phishing URLs to provide an overview of potential biases in the dataset, and compare the phishing URLs to a set of benign login URLs, with a focus on highlighting the similarities between the two classes. To this end, this section presents general details about phishing URLs, including their structure and estimates about the underlying hosting infrastructure, followed by a similar analysis and comparison for the benign URLs.

Table 4.1.: Comparison of phishing and benign URL features

	Phishing	Benign (Login)	Benign (Random)
Length (M)	66.26	60.01	58.46
Subdomains	49.45%	68.18%	62.06%
Full path components	87.90%	99.53%	98.29%
Port	0.28%	1.35%	0.01%
IP address	2.24%	0.11%	< 0.01%
Https	52.68%	96.39%	95.73%
Http credentials	< 0.01%	0.00%	< 0.01%
URL encoding	< 0.01%	0.00%	< 0.01%
IDN	0.13%	< 0.01%	< 0.01%
Subdomain labels (M)	0.82	0.69	0.62

4.1.1. Phishing URLs

To create the **DS-Phish** dataset of phishing URLs used in this thesis, we continuously downloaded phishing URLs from the three threat intelligence feeds PhishTank[1], OpenPhish[2], and PhishStats[3] over a period of several years. In detail, we created a dataset of 3,515,858 URLs, by combining 446,627 URLs collected daily from PhishTank between December 2018 and March 2021, with 3,069,231 URLs collected hourly from the sources OpenPhish, PhishTank, and PhishStats from March 2021 to October 2022. After normalizing the URLs by converting all letters to their lowercase equivalent and removing exact duplicates, the **DS-Phish** dataset includes 1,520,114 URLs.

We begin the analysis of phishing URLs with a general look at the URL structure of the URLs in the dataset. To this end, we parsed all URLs using the Python `urllib` module[4], and extracted eTLD and RD based on the public suffix list[5]. An overview of analysis results is depicted in Table 4.1.

We first take a look at the **length** of the URL, which is for example often used as a feature in phishing detection classifiers. In our dataset, the mean length is 66.26($SD = 86.91$), with a median of 47, however we observe a long tail of longer URLs up to several thousand characters, while 1,334,708 URLs (87.80%) consist of no more than 100 characters (see Figure 4.1).

As for the URLs' **composition**, 751,647 (49.45%) have subdomains, 1,335,589 (87.86%) paths, 209,882 (13.81%) queries, and 9,860 (0.65%) fragments, with 1,336,239 (87.90%) having at least one full path component (i.e., either path, query, or fragment, see Section 2.2 for details). Only 4,308 URLs (0.28%) include a port specification. As for the usage of IP addresses as host, a technique that can be used to hide the destination of a URL, 34,101 (2.24%) phishing URLs make use of the technique. In summary, this indicates, that path components are very common, as is the usage of at least one subdomain, while only few URLs include a port specification or an IP

[1] `https://www.phishtank.com/` online, accessed 2023-01-24
[2] `https://openphish.com/` online, accessed 2023-01-24
[3] `https://phishstats.info/` online, accessed 2023-01-24
[4] `https://docs.python.org/3/library/urllib.parse.html` online, accessed 2023-01-25
[5] `https://publicsuffix.org/list/` online, accessed 2023-01-25

address as host.

Next, we take a closer look at the different parts of the URLs, beginning with the **scheme**. Here, the APWG has reported a rising trend of websites using https instead of http, increasing from about 30% in 2017 to more than 80% in 2021 [APW21], which we confirm by checking the scheme of the URLs in the dataset directly. The most commonly used scheme in the dataset is https (800,857 URLs, 52.68%), followed closely by http (719,253 URLs, 47.32%), and the only remaining four URLs using ftp. Since the protocol specification in the URL does not directly indicate, whether or not the website supports https (in fact, many URLs with an http scheme do support https, as can for example be seen in Chapter 9), this already confirms, that https is becoming more relevant for phishing URLs.

Following along with the syntactical structure of URLs, we next take a look at two additional **obfuscation** techniques. While it is possible to use http credentials in URLs, this technique is used by only 116 (less than .01%) examples in our dataset. Of the URLs that include http credentials, 15 define username and password, the remaining only a username. A second technique that could be used by attackers to obfuscate where a URL leads is URL encoding. While its usage is relatively common in the path components of the URL, we did not find a large amount of URLs that use it in the host: Comparing the URL encoded host to the decoded one only results in differences for 68 URLs. Similarly, we counted URLs that include non-ascii characters or the punycode indicator `xn--` in their domain name, and found that only 1,922 URLs (0.13%) make use of IDN as a potential obfuscation technique. These results already indicate that advanced techniques, such as http credentials or URL encoding in the host to obfuscate URLs, are not a common occurrence.

Next, we take a look at the **hosts** of phishing URLs, where we are particularly interested in the RD and subdomains. The most common RDs in the phishing dataset are those of hosting providers (see Table 4.2). However, the ten most common RDs do not only include web hosting apex domains, but also one domain that generated a large amount of unique threat intelligence feed entries (e.g., due to using different subdomains for different victims or campaigns), while one RD also belongs to a link shortening service. Of the 506,620 unique RDs in the phishing dataset, 246,494 (48.65%) appear in at least two URLs, with 260,126 (51.35%) appearing only once. The twenty most common RDs cover 125,664 URLs (8.27% of all phishing URLs in the dataset). In all, we note that the analysis of RDs reveals, that many phishing websites in our dataset are not hosted on attacker-owned infrastructure. Almost 10% of URLs share the same 10 RDs, even though more than half of the RDs appear in only a single URL.

Finally, determining the number of **domain labels** in subdomains by simply counting the number of dots reveals the distribution seen in Figure 4.2. Interestingly, the distribution of the number of labels has a long tail, as while most URLs (768,467 URLs, 50.55%) have no, or between 1 and 3 labels in the subdomain (728,432 URLs, 47.02%), one URL even includes 39 labels.

In all, our dataset is dominated by URLs using https, with most URLs being *complex* in that they include subdomain and path components. Both, and the usage of subdomains in particular, can be an effective technique to confuse potential victims about the destination of the URL (as we see in Chapter 5). While the usage of advanced techniques like URL encoding or http credentials is uncommon, URLs with these techniques do appear in the dataset. Finally, we note that the analysis of

Table 4.2.: The ten most common RDs in the phishing URL dataset

Registrable Domain	Assumed Type	Occurrences
000webhostapp.com	Hosting	36633
weebly.com	Hosting	19852
tmweb.ru	Hosting	7418
swtest.ru	Hosting	6746
qwo231sdx.club	Unknown	6319
google.com	Hosting	6168
fleek.co	Hosting	5774
co.vu	Hosting	4691
justns.ru	Hosting	4083
bit.ly	Link Shortening	3558

RDs already reveals, that many phishing websites in our dataset are not hosted on attacker-owned infrastructure. The usage of hosting services, next to compromised infrastructure, makes the detection of phishing websites a complex task, as it hinders the usage of blocklists or even known-benign lists, and poses problems to automated detection based on website and domain meta information (e.g., the URL, WHOIS information, or information extracted from a certificate).

4.1.2. Comparison to Benign URLs

While the analysis of phishing URLs on its own already gives insights into the structure, composition, and even underlying infrastructure of phishing URLs, a comparison to benign URLs can further highlight similarities and differences which should be avoided or might be used in automated detection or user education. In this subsection, we therefore endeavor to provide such a comparison by collecting two sample datasets consisting of (1) a dataset with a particular focus on benign login URLs (as URLs which ask for username and password are often replicated by and therefore similar to typical phishing websites), and (2) random benign URLs. Note, that the goal of this section is not to create representative samples of benign URLs, as this would be a more complex task (see Section 4.3), but rather to create a baseline for comparison, with a particular focus on similarities between benign and phishing URLs, thus highlighting the complexity of the phishing classification task.

To **create the datasets** of benign URLs, we downloaded the Tranco[6] list of popular websites on October 10, 2022, and processed the top 250,000 domain names by adding the prefix `https://` to all domain names. We then crawled the websites using Python `requests`[7], parsed the returned html code using the package beautifulsoups[8], and extracted all `href` attributes from `<a>` tags to obtain more diverse URLs compared to simply using the homepages of the crawled websites. The `href` attribute can contain relative URLs[9], which start with a slash and describe a location relative to the document root (e.g., `/login/`), which we resolve by appending relative URLs

[6] `https://tranco-list.eu/` online, accessed 2022-12-15
[7] `https://requests.readthedocs.io/en/latest/` online, accessed 2022-12-15
[8] `https://pypi.org/project/beautifulsoup4/`, online, accessed 2022-12-15
[9] `https://url.spec.whatwg.org/` online, accessed 2022-12-22

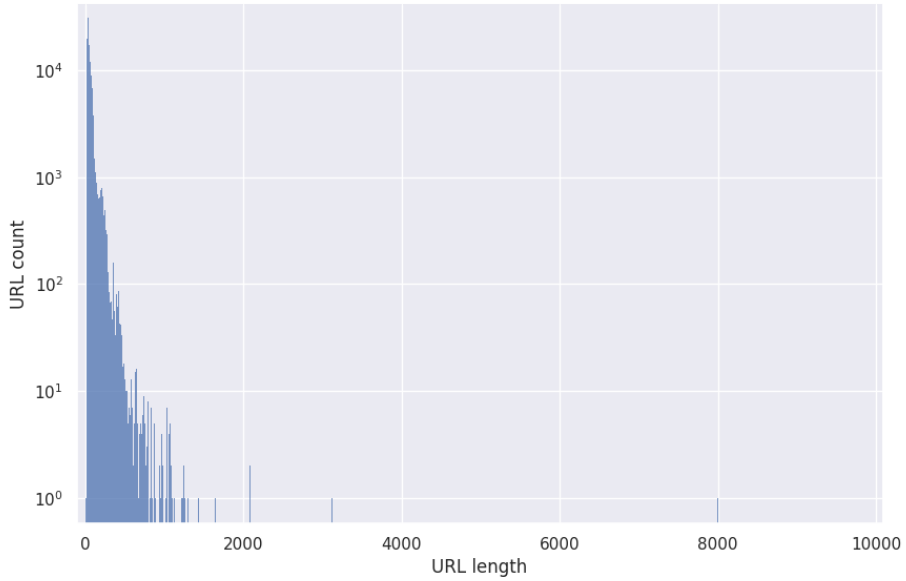


Figure 4.1.: Histogram of URL lengths of phishing URLs up to 10,000 characters in log scale.

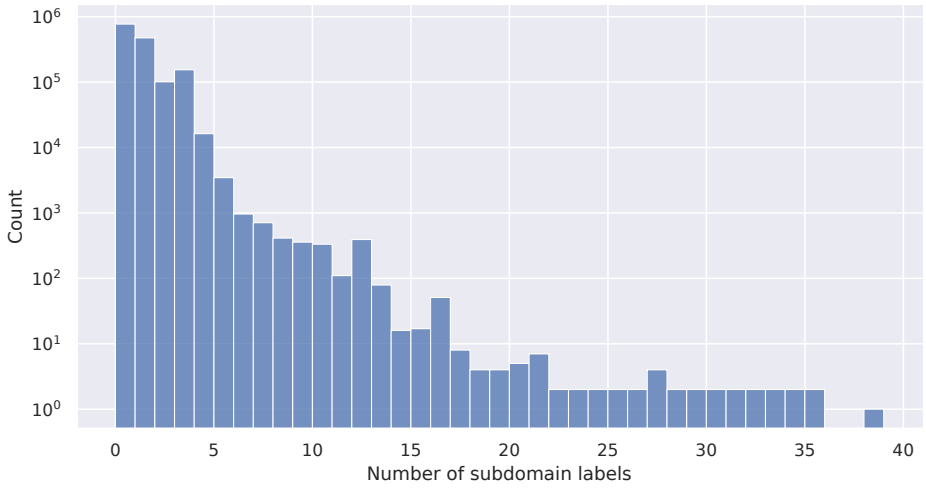


Figure 4.2.: Histogram of number of subdomain labels for phishing URLs.

Table 4.3.: The ten most common RDs in the benign login URL dataset

Registrable Domain	Occurrences
vigilafrica.com	1235
google.com	338
admissionlogin.in	282
voxmedia.com	282
medium.com	211
facebook.com	157
force.com	154
amazon.com	151
travelandleisure.com	149
loginradius.com	141

to the current base URL. After resolving all relative URLs, we removed duplicates, resulting in a set of 13,796,620 URLs, which we assume to be mostly benign due to being directly reachable from popular websites. From these we first extracted login-related URLs, as we assume them to be the most likely to be targeted in phishing attacks, by selecting URLs that include the keywords `login` or `signin` (i.e., by matching against the regular expressions `log.?in` and `sign.?in`). This results in 67,155 (0.49%) login related URLs. We also created a second benign dataset of similar size to the set of phishing URLs by randomly sampling 1.5 million URLs from the remaining URLs. Both benign datasets were sanitized by only including URLs starting with the http or https schemes (this was necessary as the collection process led to several strings that could not be parsed as URLs at all).

Repeating the previous analysis of phishing URLs for the new datasets, we find commonalities but also several differences (see Table 4.1). First, for the login-related URLs, the **length** of the URLs is similar to those of phishing URLs, with a mean of 60.01 (SD=57.39) and a median of 46 (see Figure 4.3 for a comparison of the cumulative distribution of lengths between benign and phishing URLs). Here, 90.04% of URLs consisted of 100 or less characters. Another similarity is the general **composition**, since 45,784 (68.18%) URLs had subdomains, 66,661 (99.26%) paths, 20,541 URLs (30.59%) included queries, and 2,467 (3.67%) fragments, with 66,841 (99.53%) including at least one component in the full path. Again, only few URLs specified a port (908, 1.35%), or had an IP address as host (75, 0.11%).

As for **schemes**, this dataset consists almost exclusively of https URLs (64,731, 96.39%) with only 2,424 (3.61%) http URLs (other protocols were filtered in the normalization step). This is likely due to the decision to use https as the scheme when crawling the websites, but might also be influenced by the fact that more popular websites are more likely to employ https. In contrast to the examples of **obfuscation** in phishing URLs, however, we did not encounter http credentials at all in the benign login URLs, nor any URL encoded hosts. Taking a closer look at the **host**, we first note that the ten most common RDs (see Table 4.3) already indicate a potential bias in our dataset, due to the inclusion of several lesser known services that may have simply generated many URLs in the collection step.

Of the 40,952 RDs, 9,176 (22.41%) appear in at least two URLs, resulting in 31,776

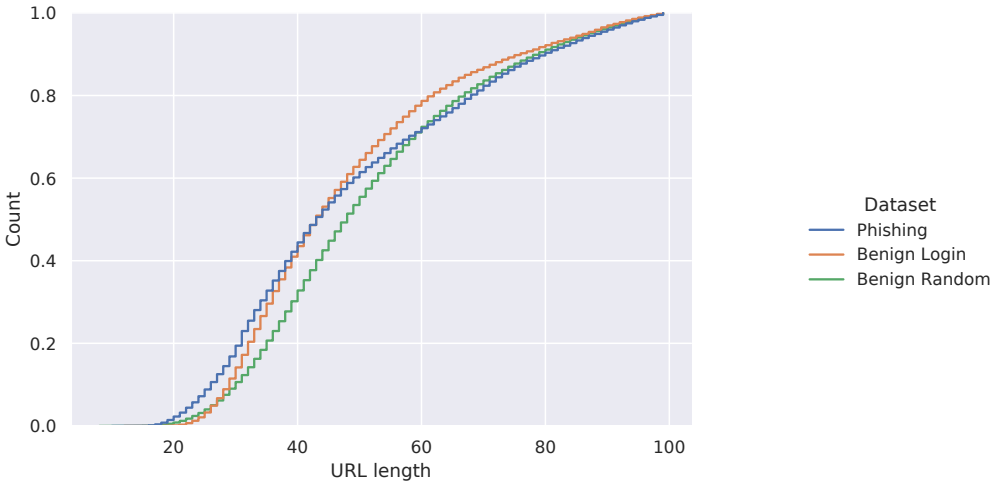


Figure 4.3.: Cumulative distribution of URL lengths for phishing, benign login, and random benign URLs.

(77.59%) RDs that only appear once, indicating more diversity than the phishing dataset. The opposite holds for the number of **domain labels**, which is restricted to three or less: 21,371 (31.82%) had no subdomains, and the majority exactly one (45,074 URLs, 67.12%).

While the dataset of **random benign URLs** is comparable to both previous sets, it does show somewhat different properties compared to the login-related URLs, as it includes 9 URLs with http credentials (though we note that some might be email addresses that were incorrectly parsed as URLs), and 12 URLs with URL encoding in the host. Length and composition are both similar to the other benign dataset, however some differences appear in the host. Here, of 168,365 unique RDs, the majority of 114,611 (68.07%) appear in at least two URLs, leaving the remaining 53,754 (31.93%) to appear only once. The ten most common RDs also changed, as can be seen in Table 4.4. As for subdomains, while most URLs again had only one subdomain label (925,809), with only few up to four labels, we note that we did observe several URLs with more subdomain labels in the complete benign dataset, just not in this random selection.

To summarize, we found that neither phishing nor benign URLs in our datasets make extensive use of http credentials, URL encoding in the host, or IDN. The composition of URLs is mostly similar between phishing and benign URLs, but phishing URLs have a more varied distribution of subdomain label counts, and are slightly longer on average than URLs from either of the sample benign data sets. Finally, we note that neither the usage of IP addresses or the specification of ports, nor using http over https automatically indicates a phishing websites, as URLs with these attributes appear in both datasets.

4.2. Impostor URLs

In this section, we concentrate on a subset of phishing URLs that is of particular interst to the topics discussed in this thesis. We call these URLs *impostor URLs*,

Table 4.4.: The ten most common RDs in the benign random URL dataset

Registrable Domain	Occurrences
twitter.com	11773
facebook.com	10662
youtube.com	8259
instagram.com	6522
linkedin.com	4526
vigilafrica.com	3875
worldairlinenews.com	3612
google.com	3090
otziv-otziv.ru	1811
lasdocas.cl	1753

as they impersonate a different website by include a reference to their target, even though they do not lead to this target website. In the following, we define this class of phishing URLs and present notes on their construction, followed by a closer look at impostor domains and their detection.

4.2.1. Definitions

Intuitively, we define impostor URLs as *Phishing URLs, that include a reference to a target name, or domain, while leading to a different domain.* Similarly to the case of the whole URL, we define *impostor domains* as FQDNs with this impostor property. There are two main aspects to this definition: first, we are only interested in phishing URLs, in contrast to benign URLs which include a reference to a known entity. Second, the URLs have to reference a *target*, which is the main aspect that separates impostor phishing URLs from non-impostor phishing URLs, as the latter do not include such a misdirection. Note, that this definition includes a semantic aspect: the inclusion of the target with an intend to mislead can not be captured using syntactic analysis alone (e.g., the URL `https://login-now.awazonaws.com` is hosted using amazonaws, thus including the brand name `amazon`, which is not necessarily the target of the phishing attack, or `https://some-website.co.uk/wordpress/wp-content?page=10` includes wordpress in the path, which is unlikely to be the target). As such, the classification of impostor domains depends on context, and while it can in fact be relatively easy to manually classify most impostor domains (provided one is familiar with the target), the automated detection, lacking this context, is a complex task.

Note, that the definition of impostor domains does not include all purposely misleading phishing URLs, as we do not include generic keywords as a target. As such, a phishing URL like `https://account-login.com/information/` does not fit the definition, even though the usage of login related keywords might also invoke misconceptions about the purpose of the URL in potential victims. As a second note, benign URLs, which are never impostor URLs by definition, can also include references to known entities. Obviously, companies often include their own name in the URL (`paypal.com` includes `paypal`), but it is also common to encounter URLs that include a different entity, for example when redirecting or referencing (e.g., `https://paypal.my-shop.de` might be a redirection to PayPal by a legitimate

shop).

4.2.2. Rule-based Impostor URL and Domain Detection

While it is not trivial to detect impostor URLs due to the dependence on context, we attempt to define several rules that, when applied to a set of known phishing URLs, are likely to return only impostor URLs. To this end, we first split the set of impostor URLs into different categories based on the manipulation technique which was applied to the original target name or target domain name when the URL was constructed. Here, the URLs can be differentiated by whether the target domain appears completely (*full domain embedding*) or only partly (e.g., when combining the target name with a keyword for *combosquatting* or including the target name in a subdomain for *subdomain posing*), and whether it appears with or without changes (e.g., characters can be replaced, inserted, removed or swapped for *typosquatting*).

In the following, we present a number of rules that we utilized with the aim of separating impostor URLs from other phishing URLs. We applied these techniques to domain names, and make use of the resulting set of impostor domains in a general analysis of the employed manipulation techniques, and later in the certificate classification task described in Chapter 10.

4.2.3. Process of Rule-based Domain Classification

Due to the focus on domain names from certificates in Part III of this thesis, we create a dataset consisting only of impostor domains, i.e. FQDNs with an impostor property. To analyze these domains and create impostor classifiers, we first created a rule-based baseline classifier to label the existing dataset of FQDNs (extracted from the dataset of URLs described above). The rule-based classifier is based on simple principles to detect URLs that are similar to known entities. The source code and rule-generation process were adapted from [Joh22]. Note, that the goal of this labeling process is not to generate a perfect ground truth, as this would be a very complex process, due to both false positives and false negatives resulting from the problems explained above, but rather to create a first baseline that we use to argue about the structure of impostor URLs and domains.

In order to detect impostor URLs, two main considerations are necessary: Collecting known benign entities (i.e., targets), and matching them to the set of potential impostor domains. In this thesis, the benign entities are collected similarly to previous work on phishing URL detection, where potential targets are selected based on domain rankings (see, e.g., [Rob+19] for examples). We therefore utilize a list of the top one million domain names according to the Tranco list as benign targets. While the whole list is used to filter out known benign domains, we do not perform a search for impostor domains using all benign domain names. This is mainly due to two reasons: performance constraints, as the filtering process takes longer the more benign domains are searched for, and false positives, since the more domains and names, and therefore keywords to match are added, the more FPs will be generated. In the analysis below, we therefore restrict the filtering process to 5,000 benign domains when looking for impostor domains. Note, that this restriction seems to be less severe than it may seem, as an analysis of the number of found impostor domains reveals a trend of diminishing returns when adding less popular targets. We discuss

this and other limitations of our approach in more detail in Section 4.3 below.

We match phishing URLs to different rules to detect different types of impostor domains (we assume a target with the RD `target.com` for the following examples):

1. RD embedding: the complete target RD, followed by either a '`.`' or '`-`', is included in the impostor domain. Example: `target.com.phishing.ml`.

2. e2LD embedding: the target e2LD is included in the impostor FQDN, but their RDs are not equal. This includes combosquatting, as well as URLs where the TLD was replaced, and the target e2LD appears in a subdomain. Examples: `target-online.com`, `target.ml`, `target.login-phishing.co.uk`.

3. Strict typosquatting: here we use the Damerau-Levenshtein distance [Dam64] to detect deliberate misspellings (i.e., typosquatting domains) on the complete phishing and benign e2LDs. The strict rule only matches impostor domains that have an e2LD with a Damerau-Levenshtein distance of one (i.e., there was only one manipulation) to the target e2LD. Example: `terget.ml`.

4. Relaxed typosquatting: includes combinations of typosquatting with e2LD embedding, i.e., includes domains that include the target e2LD with an edit distance of one in their FQDN. Example: `terget-online.com`.

We further restrict the search for relaxed typosquatting, the most resource- intensive operation, to targets that appeared at least 10 times using the other methods, as determined in a preliminary run. From these we further filter seven domains that had more than 100 relaxed typo hits, but were determined to be mostly false positive by a manual review of the most common targets using the relaxed typosquatting rule (e.g., the benign e2LD '`t-online`', which matches all URLs including '`-online`').

4.2.4. Impostor Domain Dataset

We utilize the baseline classifier on our **DS-Phish** dataset of phising URLs described above, which contains 804,664 unique FQDNs. Applying the rule-based classifier for the Top 5,000 domains in the Tranco list as target domains, after filtering the phishing URLs by extracting their RDs and filtering them against all known-benign RDs on the list to remove known false positives, results in 54,422 impostor domains (6.76%, see Table A.1 in Appendix A.1 for examples for each rule). We call the resulting dataset of impostor domains **DS-Impostor**. In the following, we describe the rules used to generate the dataset and its resulting FQDNs in more detail.

First, taking a closer look at the distribution of the rules which match the phishing domains reveals, that e2LD embedding is the most common method (35,313 FQDNs, 64.89%), followed by RD-embedding (9,435 FQDNs, 17.34%), relaxed typosquatting (8,764 FQDNs, 16.10%), and strict typosquatting as the least common method with 910 FQDNs (1.67%). As the e2LD rule matches different categories of phishing URLs, we investigate further, and find that 625 were TLD replacements (most of these seem to be FPs due to benign hosting providers which use several TLDs that were not present in our benign filter list), 16,978 where the target e2LD is included in the subdomain, and the remaining impostor domains (17,710) are combosquatting domains where the target e2LD is included together with additional keywords in the phishing e2LD. The most commonly targeted benign domains are shown in Tables 4.5

Table 4.5.: The ten most common targets of impostor domains

RD Embedding		Strict Typo		e2LD Embedding	
Target	N	Target	N	Target	N
olx.pl	1459	steamcommunity.com	153	amazon.com	5624
amazon.co.jp	1433	amazon.com	104	facebook.com	5253
rakuten.co.jp	1033	instagram.com	69	whatsapp.com	4882
apple.com	662	facebook.com	68	paypal.com	3161
facebook.com	613	paypal.com	43	rakuten.co.jp	1633
icloud.com	395	opensea.io	38	wellsfargo.com	1069
blogspot.com	355	mercari.com	36	instagram.com	1064
allegro.pl	299	geocities.com	35	microsoft.com	1012
paypal.com	297	rakuten.co.jp	30	donmai.us	988
irs.gov	172	netflix.com	29	netflix.com	856

Table 4.6.: The ten most common targets for the relaxed typosquatting rule

Target	Count
amazon.com	2743
whatsapp.com	1117
rakuten.co.jp	921
pochta.ru	688
paypal.com	670
mercari.com	257
facebook.com	250
docomo.ne.jp	238
instagram.com	232
telekom.de	198

and 4.6. Note, that the target domain is ambiguous for rules that focus on the e2LD when two or more benign domains differ only in their eTLD. In these cases, the first matching domain is utilized in the analysis, e.g., `telekom.de` is the first telekom provider (e2LD is extracted) that was found, other benign RDs that differ only in the eTLD (e.g., `telekom.hu`) match as well.

Next, we take a closer look at strict typo domains, and analyze which characters were used for these domains. This gives insights into whether attackers prefer particular sets of characters, for example look-alike characters when replacing letters, or vowels for the deletion manipulation. As can be seen in Table 4.7, there seem to indeed be some biases towards characters sets. For character replacements, the most common choices include look-alike characters (e.g., (i,l), (0, o)). For insertions, the hyphen was most popular, which makes sense as it can be used in composite target names (e.g., `steam-community.com`). We counted the number of consonants and vowels that were inserted or deleted, and found some differences: in both cases, consonants were more likely to be used, with 222 consonants and 100 vowels used for insertions and 78 consonants and 44 vowels for deletions. Finally, only few domains were created by swapping two characters, where the most common occurrence is

Table 4.7.: Characters used in typosquatting domains

Replacements		Insertions		Deletions	
Characters	N	Characters	N	Characters	N
i-l	46	-	35	s	23
a-e	36	l	33	a	16
g-n	35	i	33	l	12
0-o	24	n	25	h	10
m-n	21	s	21	o	10
a-o	11	o	21	e	8
o-u	10	c	20	i	8
l-y	8	e	19	m	6
c-o	7	r	17	n	4
a-u	7	a	16	t	4

again a look-alike operation (f,t) with 5 occurrences, followed by (a,z), (i,t), and (e,n) with only three occurrences each.

4.3. Discussion

In this chapter, we defined and described a set of phishing URLs, and compared it to a set of benign URLs. While the phishing URLs often included some complexity in the form of subdomain or path components, and were more likely to make use of obfuscation techniques, there were no discerning characteristics that immediately separate benign from phishing URLs. Taking a closer look at impostor URLs and domains, we created a dataset based on rule-based classification, and analyzed common manipulation techniques by attackers. Here, inclusion of the target e2LD or RD in the phishing domain were the most common techniques in our dataset. The less common typosquatting domains were analyzed for manipulation techniques as well, where we found that look-alike substitutions were the most common.

In the following, we discuss limitations of these results, followed by an interpretation of how they impact the following chapters of this thesis, and how they might transfer to the broader research area of phishing prevention.

4.3.1. Limitations

In the analysis of phishing URLs, we created a dataset based on several popular threat intelligence sources. While the collection of URLs took place over several months, it is still likely that the dataset is not representative of all phishing websites globally. It is furthermore possible, that URLs were edited before they were submitted to the intelligence feeds, for example by removing http credentials, or path components. We therefore note, that the analysis is only valid for the subset of URLs that are represented in this dataset.

Similarly, the benign dataset of URLs is not representative, as it only includes URLs extracted from links on the homepages of popular websites. We argue, that the decision to use popular websites as a substitute for benign websites in general was sufficient to achieve the goal of this chapter, which was to showcase similarities

and differences between phishing and benign URLs and introduce impostor URLs. Still, we recommend using datasets for both phishing and benign samples in real-world applications (e.g., an automated phishing URL detection tool) that are as close to the actual application scenario as possible, thus improving the validity of computed metrics on the training data or classification tests. Note, that the creation of representative datasets of benign URLs is a complex task, as it might dependent on the context (e.g., organization or private), region, and user group.

It is further possible, that some phishing campaigns and benign websites were over- or underrepresented, since we did not apply processes to remove duplicate campaigns. While we did remove duplicate URLs (and domain names in the impostor domain analysis), it is still possible that some campaigns or single benign websites resulted in large numbers of distinct but similar URLs, that may have introduced a bias in our analysis. Analyzing the most common RDs in the URL datasets seems to confirm that this might have taken place, since they make up significant parts of the overall number of URLs. Still, the majority of RDs in both benign login and phishing URLs were only used in one URL, which we argue indicates sufficient variety in the datasets to generalize the implications mentioned below.

As for impostor URLs and the analysis of impostor domains, one limitation is the usage of a rule-based classifier without complete manual verification of the results. It is likely, that the final dataset of impostor domains includes false positives, and that the remaining non-impostor domains still include false negatives. False negatives remain due to the decision to include only a subset of possible targets (to avoid false positives), in addition to the restriction for typosquatting domains to have an edit distance of at most one, which does not include all manipulations. It is therefore possible that typosquatting is more common than we found in our analysis. In Chapter 10, we use the dataset of impostor domains in a classification task, and find that several classifiers were able to detect impostor domains including more than one typo as well.

4.3.2. Implications

The analysis of phishing URLs revealed several trends that can further our understanding of phishing URLs. For one, most phishing URLs utilize complexity, by including subdomain or path components. It is likely that this complexity is intended to mislead users, as the additional parts offer more opportunity for misdirections.

Next, the usage of obfuscation techniques like URL encoding, http credentials or even IP addresses seems to be unpopular, as they only appeared in a small fraction of URLs. As such, we do not focus on advanced techniques in the games used for user education (see Chapter 6), though we do include some examples in several of the user studies presented in this thesis to test, how users handle unusual techniques.

In the comparison to benign URLs, the main difference between the two sets, apart from obfuscation techniques, was the larger numbers of subdomain labels for phishing websites. While this might simplify the automated detection of phishing URLs in some cases, our analysis in the next chapter also indicates that users struggle with these URLs when the target domain appears at the beginning of the subdomain. We therefore explore an alternative URL notation that aims to make this type of attack less effective in Chapter 7. Furthermore, the usage of long or numerous subdomains in benign domain names is likely more common depending on different usage scenarios,

which are not covered by our sample benign datasets. Other differences between the two datasets were less pronounced, and while even small differences might be used in automated phishing URL detection, they are unlikely to be helpful in educational contexts, and again depend on the benign datasets being representative.

The analysis of RDs of phishing websites revealed, that phishing websites often make use of benign hosting infrastructure. In these cases, the attackers typically lose some control over the RD of the phishing URL, but gain several advantages due to the ease of setting up the website, in addition to the positive domain reputation that is associated with services that also serve benign websites. Previous work also indicates that the usage of compromised infrastructure to host phishing websites is a common occurrence, with estimates ranging from 63.5% to 78% of all phishing websites [Le +19; De +21]. Both cases make the automated detection of phishing websites a complex problem, as there is currently no feasible way to determine whether a previously benign website was compromised and is now used in a phishing attack. We therefore focus on impostor domains in the classification task in Chapter 10, and analyze several different methods to reduce the success of using non-impostor domains in Part II, thus providing a multi-layered defense against phishing.

The definition of impostor URLs constitutes one of the threads that recurs throughout this thesis, as it is the basis for several analyses in following chapters. Still, impostor domains made up only 6.76% of all FQDNs in the URL dataset, which is lower than all estimates on attacker-owned domains discussed in related work (see Chapter 3). This is unsurprising, as we defined impostor domain as only a subset of domains that were registered by attackers, and the rule-based classification approach is furthermore unlikely to have labeled all impostor domains correctly. Consequently, the amount of domains available for classification tasks based on only impostor domains is reduced compared to the inclusion of all phishing domains (see, e.g., Chapter 10 for the effect of focusing on impostor domains in classification tasks).

In our analysis of impostor domains, we found that embeddings of e2LD and RD occurred most often in our dataset, followed by relaxed typosquatting, and only comparatively few strict typosquatting domains. We still include all categories in our educational efforts and the automated detection task in later chapters. Taking a closer look at typosquatting domains, we found that substitutions with look-alike characters were more common than random replacements, while inclusion of the hyphen was the most popular choice when adding characters. The choices for omissions and character swaps were less obvious. While it might be possible to normalize typosquatting URLs by performing replacements for look-alike character, thus including domains with an edit distance of two or more in the rule-based classification, the diversity of characters used by attackers, in addition to the potential of including additional false positives, might complicate this approach. Instead, we refer to [Joh22] and Chapter 10, where the usage of Deep Learning (DL) classifiers was shown to somewhat transfer to more complex typo domains, as a promising future direction in impostor domain detection.

In all, we defined a class of phishing URLs that include a reference to a specific target (impostor URLs), and presented the creation of a dataset of impostor domains based on this definition. A short analysis revealed clear biases in the dataset, concerning targets, manipulation techniques, and even on a character level. The insights from this chapter appear throughout the thesis, including in the URL categorization presented in the next chapter, and Chapter 10, where the **DS-Impostor** dataset is used as training data to detect phishing domains in the wild.

Chapter 5

Categorization of Impostor URLs

Throughout this thesis, a number of different URL categorizations are used for different analyses and studies. We have already seen in the previous chapter, that the impostor property can be realized by different means (i.e., different manipulation techniques). In this chapter, we refine this observation by categorizing impostor URLs based on the URL structure. We also provide the results of a study that focuses on differences in detection difficulty between the presented categories in isolation (as opposed to presenting the URLs as part of, for example, a website screenshot showing an URL bar), thus providing a baseline for the following chapters. We therefore first present a detailed categorization of phishing URLs, and an in-depth analysis of the categories based on a user study, followed by a discussion.

Contributions: The main contributions of this chapter are a categorization of impostor URLs and a user study that tests the URL classification capabilities of untrained users. The categorization is based on the general structure of URLs and consolidates URL categories from previous work, resulting in a more fine-grained categorization that removes several ambiguities and implied assumptions of previous work. It is adapted from a categorization which was created in collaboration with Jakob Drees and was later formalized in his master thesis [Dre22]. Compared to the master thesis, the categories presented in this thesis were mainly renamed and rearranged according to three main properties of impostor URLs. The user study to test the classification complexities of different URL categories gives insights into the classification capabilities of untrained users and extends previous work by including URLs from a broader set of URL categories, which focus on more atomic changes, and by considering the effect of familiarity with the services included in the URLs. While the study was designed together with and conducted exclusively by Jakob Drees for his master thesis [Dre22], the evaluation was adapted and newly performed in this thesis. In particular, while the main research questions of the study were only slightly modified, we exclude the URLs of unknown services from the evaluation in this thesis and change the hypothesis testing methodology accordingly.

5.1. Detailed Categorization of Impostor URLs

Intuitively, we propose categorizing URLs based on three properties: the inclusion and position of a target name in the URL, whether or how the name was manipulated,

Table 5.1.: Impostor URL e2LD modification categories and sub-categories

Category	Sub	Example
Combosquatting	Keyword before	`secure-target.com`
	Keyword after	`target-secure.com`
Typosquatting	Add random	`tarqget.com`
	Duplicate letter	`taarget.com`
	Replace random	`targzt.com`
	Replace similar	`tarqet.com`
	Remove consonant	`taget.com`
	Remove vowel	`targt.com`
	Swap adjacent	`taregt.com`
IDN		`tärget.com`

Table 5.2.: Impostor URL target placement categories and sub-categories

Category	Sub	Example
Http Credentials		`target.com@random.com`
Subdomain Posing	First	`target.com.long.subdomain.random.com`
Subdomain Posing	Middle	`long.sub.target.com.domain.random.com`
Subdomain Posing	Last	`long.subdomain.target.com.random.com`
Subdomain Posing	Only	`target.com.random.com`
RD		`tarqet.com`
Path Posing		`random.com/target`
Query Posing		`random.com?target=1`

and how the registrable domain (RD) was constructed. The categorization focuses on phishing URLs that include a reference to a target, and aims to provide a basis for an estimation on how well users detect a given phishing URL based on the manipulation techniques it employs. It therefore extends previous work by removing ambiguities and implicit assumptions about phishing URLs, and further specifying categories where the estimations of detection rates of different URLs in the same category might otherwise vary largely. Section 4.2 already included several references to this categorization in the selection of rules for impostor domain detection, which we will now formalize to offer a structured approach to all following analyses.

A complete overview with examples of the three properties making up the URL categorization is shown in Tables 5.1, 5.2 and 5.3. Since the focus is on impostor URLs, which are a subset of phishing URLs, the categorization does not include benign URLs. We still include differences between different benign URLs in the analyses presented in this thesis, and discuss the categorization of benign URLs in more detail in Section 5.4.

The categorization was created by consolidating related work on categories of phishing URLs with the structure of URLs presented in Section 2.2. In detail, we used the categories proposed by Reynolds et al. as basis (see Table A.2 in Appendix A.2 for details on these categories), who offer the most complete categorization to our knowledge [Rey+20]. They differentiate typosquatting, two types of subdomain

Table 5.3.: Impostor URL RD base categories and sub-categories

Category	Sub	Example
Generic		`secure-account.com`
Random		`tmpovpe3sa.com`
URL Encoding	With dot	`target.com.%70%65%33%73%61%2e%63%6f%6d`
	Without dot	`target.com%2e%70%65%33%73%61%2e%63%6f%6d`
IP Address		`5.155.13.79`
Modified TLD		`target.website`

posing depending on the length of the subdomain, usage of IP addresses as host, Internationalized Domain Name (IDN) homographs, placing a target in http credentials, random URLs without a reference to a target, combosquatting, the usage of ambiguous delimiters, using an unfamiliar Top Level Domain (TLD), using URL encoding, query and fragment posing, and path posing. While the categories by Reynolds already include the different placements and modifications of our proposed categorization, they cannot be combined, thus leading to ambiguities, e.g., when a URL includes the target in a subdomain and also employs URL encoding and typosquatting. The categorization furthermore includes implicit assumptions about some of the categories, e.g., that the target always appears first in subdomain posing, which might lead to significant differences in classification performance if violated. Similarly, Reynolds et al. do not differentiate different types of typosquatting. Our categorization was therefore created by removing ambiguities from the categories by Reynolds et al. by aligning them with the three properties (placement, modification, and RD bases), and introducing new sub-categories where we assumed this to be necessary to remove large deviations in classification performance for a single category (e.g., for the subdomain placement). We further incorporated additional details on typosquatting techniques (see, e.g., [SUM17; Szu+14]) and clarified the resulting categories to remove ambiguities. Next, our analysis of impostor URLs in Chapter 4 revealed, that attackers often combine manipulation methods (as demonstrated by the relaxed typosquatting rule in Section 4.2), and we therefore added this possibility to the categorization. Finally, we attempted to remove overhead resulting from duplicated manipulation techniques by splitting the categories according to the three properties explained below.

The resulting categorization defines three properties for impostor URLs: (1) e2LD modifications (see Table 5.1), which can be applied to the e2LD of the target to either directly create new RDs or obfuscate the impostor URL by not including the exact target RD, (2) target placements (see Table 5.2), which define placements for the reference to the target in the impostor URL based on the general structure of URLs, and (3) RD bases (see Table 5.3), which define different strategies to create RDs that can be applied when the reference to the target is not included in the RD of the impostor URL. Impostor URLs can then be categorized based on the three properties by determining whether and which RD modification, target placement, and RD base were applied.

The **e2LD modifications** category is further divided into *typosquatting*, *combosquatting*, where a keyword is added before or after the e2LD of the target, and *IDN*

homographs, where characters in the target's e2LD are replaced by lookalike IDN characters. Similar to previous work (e.g., [SUM17; Szu+14]), we define typosquatting domains based on the Damerau-Levenshtein distance between the target's e2LD and the reference to the target in the URL. This distance corresponds to the minimal number of operations (adding, removing, or replacing one character, or one transposition of adjacent characters) required to change the target e2LD into the included reference [Dam64]. Based on the hypothesis that impostor URLs that use similar-looking characters when adding or replacing characters are more complicated to detect for users, we further subdivide typosquatting into subcategories depending on whether the operations are performed using look-alike or random characters. We additionally include subcategories for removing vowels or consonants, as consonants have been attributed a greater importance for reading tasks in different contexts by previous work [NAN08]. For combosquatting, we differentiate two subcategories based on whether the keyword is inserted before or after the target e2LD.

The **target placements** consist of *http credentials*, *subdomain posing*, *RD*, or *path posing* where the target RD or e2LD is included as http credentials, in a subdomain, the RD, or a path component, respectively. Path posing can be further divided depending on whether the target is included in the path, query, or fragment of the full path, however we only differentiate path- and query posing in this thesis, as previous studies found little differences between inclusion of the target in the query or fragment [Rey+20].

For the **RD bases**, we differentiate *IP addresses*, where the host consists of an IP address, *random identities*, where the e2LD consists of completely random characters, *generic identities*, where the e2LD consists of generic keywords, *URL encoded* hosts, where the RD is rewritten using URL encoding, and *TLD modifications*, where only the TLD is changed. To ensure that URLs can be categorized without ambiguities, the URL encoding category has precedence over the other categories, as it is possible to obfuscate any host using this method. Similarly, the TLD modifications category is only applicable when the target is placed in the RD (i.e., otherwise one of the other RD bases already covers modified TLDs).

Note, that URLs with a target placement outside the RD always require an RD base to ensure, that URLs can be uniquely mapped to a corresponding category. Otherwise, URLs that, for example, include the target in the subdomain do not specify how the RD of the URL should be constructed. This led to ambiguities in previous work (e.g., [Rey+20]), where the target reference could for example be included in the subdomain and the RD obfuscated with URL encoding or an unusual TLD, leading to several possible categories for a single URL.

We refer to URLs that do not include a reference to a target as *random* URLs, as they appear without context compared to impostor URLs. Note, that we focus on URLs where a reference to the target name or domain appears only once in this thesis. While it is possible that different categories are combined when the target is included multiple times, we did not test how such a combination of techniques would influence the classification performance of users.

As the focus of the categorization is on phishing domains, we do not generally make a distinction between different categories of benign URLs. While it might, for example, be possible to differentiate benign URLs based on perceived complexity, which includes its length or the existence of subdomains and path or query components, we usually generalize the benign URL class in this thesis.

The next section describes a user study that was performed to create a baseline in classification performances for the different categories, and whether the chosen categories actually lead to observable differences in classification performances, or whether it is possible to combine categories in analyses for simplicity.

5.2. User Study Setup

The user study to test the categorization follows a simple URL quiz format, where 45 participants classified 99 URLs in an online survey. The main research objective of the survey is to evaluate, whether there are differences in classification performance between the different categories and subcategories, thus resulting in a baseline for educational content, while also testing whether differentiating the sub-categories is necessary in practice (e.g., in URL tests in user studies). In particular, we were interested in differences between:

- URLs with known and unknown services as target
- different RD base classes
- URLs with or without path and query components
- different subdomain subcategories, with a focus on the position of the target domain in the subdomain
- using the target domain in http credentials and the beginning of the subdomain
- different typosquatting subcategories
- different manipulations where the target reference is placed in the RD

The user study therefore extends related work by including the effect of service familiarity on classification outcomes, in particular in relation to typosquatting URLs where service familiarity likely had a large effect in a previous study [SUM17]. It furthermore explicitly tests for differences between RD bases, which have to our knowledge not been considered in isolation in previous studies. Finally, the study aims to reveal differences in the classification accuracy of different URL categories, with a focus on the complexity of the URL (e.g., whether the URL has a path or query) as well as the position of the reference to the target in the URL (e.g., at the beginning or end of a subdomain).

5.2.1. Participants

The participants for the study were recruited via two methods: (1) an online survey exchange platform[1] was utilized to recruit participants and (2) additional participants were recruited by inviting family members and friends of the supervisor of the study (see [Dre22] for details). We discuss the possible effect of this participant recruiting choice in the discussion below. As the study was conducted in German, all participants had to satisfy a sufficient level of language proficiency to participate. The choice to use German as a language was made as it was (1) the native language of the survey

[1] `https://surveyswap.io` online, accessed 2023-01-25

supervisor, thus making recruitment easier and (2) in the hope that it increases the likelihood that participants were familiar with the services used in the selected URLs, as they were based on German top websites (the next section explains the URL selection in more detail).

Note that no personal data was collected from or about the participants, including demographics or recruitment method. While this greatly simplifies the data protection aspect of the study, it also makes it impossible to report on potential biases due to the sample not being representative. We note, however, that the sample is likely to be skewed towards students, due to the recruitment methods.

In all, 48 participants took part in the survey, however not all of them classified all URLs. As it is possible to infer most of the research questions even for smaller numbers of classified URLs, we only removed 3 participants who classified less than 10 URLs overall. Of the remaining 45 participants, 37 completed the survey by classifying all 99 URLs.

5.2.2. Apparatus and Materials

The study was conducted as an online study consisting of an online questionnaire created and provided via the LIME[2] survey software. Apart from a general introduction of the survey and phishing attacks in the beginning (see Figure A.1 in Appendix A.2 for screenshots), the survey consists of two main parts:

- Familiarity of service questionnaire: As it has been previously found, that familiarity with a service can have a significant impact on the classification performance (see Section 3.1.3), we asked participants for their familiarity with the services used in the study as either *unknown*, or *known* (see Figure A.2 in Appendix A.2). This questionnaire was included to (1) confirm results of previous studies where familiarity had an effect on the classification, and (2) to be able to correct for this difference and remove potential biases if necessary.

- URL test: The URL test includes a test of 99 URLs (40 benign, 59 malicious). Each URL was presented on a new page in the survey, together with a screenshot of the corresponding website (without an URL bar) and the two answer possibilities *legitimate* or *phishing* (see Figure A.3 in Appendix A.2). The URLs were selected to provide answers to the research questions presented above, with a focus on analyzing the proposed categorization. All participants were presented with the same URLs, but in randomized order.

Participants were directed to the survey via a link (either from the survey swap platform or directly), were presented with the general introduction first, followed by the familiarity questionnaire and the URL test, after which they received feedback on the number of legitimate and phishing URLs they classified correctly (see Figure A.4 in Appendix A.2). In all, the survey was estimated to take about 15 minutes to complete.

[2] `https://www.limesurvey.org/` online, accessed 2023-01-25

URL Selection

The benign and phishing URLs in the URL test were generated from the benign RDs of the 30 most visited non-adult websites in Germany according to Semrush[3]. All URLs use the https scheme.

Phishing URLs were generated according to rule-based manipulation techniques, which correspond to the (sub-)categories defined above. Due to the large amount of possible URL categories, we restrict the study presented in this paper to a subset of all possible categories. First, we evaluate the three properties (placement, modification, and RD bases) separately, by including URLs that only differ in one of the properties in the test. Next, we did not test IDN or IP addresses, as they were already evaluated in previous work [Tha+19; Rey+20]. Applying the remaining manipulation techniques to a benign RD results in a pool of different, randomly generated URLs per (sub-)category, depending on the target and the specifics in applying the rule (e.g., usage of different random or generic strings). The URLs for the test were then selected uniformly at random from these pools, using a distribution that balances the trade-off between including too many URLs on the one hand, thus potentially leading to fatigue and deterring participants, and including sufficiently many URLs to make meaningful statements about differences between categories (as opposed to differences between single URLs) on the other hand. In detail, we selected 8 URLs to test differences between the RD base categories directly by selecting 2 URLs of the subdomain-only category for each of the tested base classes (generic, random, and URL encoding with or without dot), thus enabling a direct comparison. To test for differences between target placements in path or query, 4 URLs were selected with a random RD, two each for both placements. Next, we selected 6 URLs to test for differences between the subdomain posing sub-categories by selecting 1 URL for each position (first, middle, end) for random and URL encoded RDs. To test for differences between placements in http credentials and subdomains, we next selected 2 URLs with the target placed in http credentials for each of the random and URL encoded bases. Due to the high variance of different URLs applying the same typosquatting techniques in previous studies [SUM17], we selected 4 URLs for each of the seven sub-categories, resulting in an additional 28 URLs. Finally, to compare typosquatting to other RD modification categories, 9 additional URLs were selected, 3 for each sub-category of combosquatting (keyword before or after the target e2LD) and 3 for TLD modification where the target e2LD is identical to the impostor e2LD.

For benign URLs, all RDs were extended by the common '`www.`' prefix and either presented without any additional complexity, or with a randomly generated query component. We selected 40 benign URLs in all, 28 with query and 12 without, as a trade-off between keeping the overall number of URLs as low as possible, and achieving a balanced distribution of phishing and benign URLs.

The final selection of URLs and their categories, as well as their mean performance scores, can be found in Tables A.3 and A.4 in Appendix A.2.

[3] `https://de.semrush.com/blog/top-der-meistbesuchten-webseiten/` online, accessed 2023-01-25

Table 5.4.: Differences between levels of familiarity in URL category study

	N	Mean	SD
Unknown	39	0.701	0.167
Known	45	0.805	0.126

5.3. User Study Results

In the following, we present the evaluation of the research questions, which is based on the classification outcomes of the URLs in the URL test. The performance scores reported in the following are computed as relative scores, i.e. the number of correctly classified URLs divided by the number of all URLs, and measured using an interval scale. We use a significance level $\alpha = .05$.

5.3.1. Service Familiarity

We begin the analysis with the effect of service familiarity on classification performance. To this end, we label each URL as either *known* or *unknown* for a given participant by extracting the service that is used in the URL (i.e., the target for phishing URLs, and the actual benign service for benign URLs) and looking up the answer of the *familiarity of service questionnaire* for this service. If the participant was familiar with the service included in the URL (i.e., if they rated the service as *known*), we label the URL as known as well, otherwise we label it *unknown*. As can be seen in Table 5.4, despite the inclusion of website screenshots to provide context to the participants, unknown services were generally detected with less accuracy than known services. The mean performance difference between URLs of known and unknown services is .104, and is also present when dividing the URLs based on their categories (i.e., the difference does not arise due to more complicated categories including more URLs of unknown services). A paired samples t-test confirms these differences with a large effect size: As a deviation from normality was detected (Shapiro-Wilk test, $p < .001$), a two-tailed Wilcoxon signed-rank test was performed, leading us to accept the hypothesis that there are significant differences between unknown and known services: $W = 140, z = -3.489, p < .001, r = .641$.

As such, we decided to remove the answers to the URL quiz for all URLs of services that participants were not familiar with (i.e., they selected *unknown* for the service in the survey). This is possible in this case, as we did not find the exclusion of URLs of unknown services to significantly reduce the number of valid samples for the analyses below, while completely removing the effect of this potential bias on the evaluation outcomes. We provide more arguments for this decision in Section 5.4 below.

5.3.2. Different RD Bases

For the three tested *RD base* categories where the target is not placed in the RD (generic, random, URL encoding), the study includes an exemplary comparison for all three classes directly using the subdomain-only placement.

As shown in Table 5.5, the generic class leads to higher accuracies than the random and URL encoding classes in a direct comparison, however with large standard

Table 5.5.: Differences between the three RD bases for subdomain-only URLs

	N	Mean	SD
Generic	41	0.622	0.430
Random	39	0.513	0.480
URL Encoding	42	0.442	0.431

Table 5.6.: Differences between URL encoding and random RD bases for shared modifiers

	URL Encoding			Random		
Modifier	N	Mean	SD	N	Mean	SD
Http credentials	41	0.695	0.401	42	0.619	0.466
Subdomain-first	38	0.711	0.460	18	0.556	0.511
Subdomain-middle	37	1.000	0.000	39	0.974	0.160
Subdomain-end	43	0.977	0.152	38	1.000	0.000
Subdomain-only	42	0.442	0.431	39	0.513	0.480

deviations for all three classes. Consequently, a repeated-measures ANOVA (N=37) with the three base classes as categories is inconclusive: $F(2, 72) = 2.767, p = .070, \eta_p^2 = .071$. Comparing the five common placement categories of the random and URL encoding base classes (see Table 5.6) does not reveal notable differences either.

Since there are no significant differences between the RD bases, and the mean differences are relatively small with large standard deviations, we do not differentiate between RD bases in the following analyses.

5.3.3. Relevance of Path and Query

The next research question regarding the categorization of URLs is whether participants had different classification performances for URLs that include a path or query. For benign URLs, the study contains URLs that do not include any path or query components, and URLs that only have a query but no path (e.g., `https://www.example.com?query=1`). The large mean differences between plain URLs and URLs with query ($MD = .252$, see Table 5.7) indicate significant differences between the two categories, which are confirmed by a paired samples two-tailed t-test. As a deviation from normality was detected (Shapiro-Wilk test, $p < .001$), a Wilcoxon signed-rank test was performed: $W = 734.000, z = 4.801, p < .001, r = .882$.

While the first study did not include URLs to test the difference between URLs with a query but no path, and URLs that do include a path (indicated by whether the host is succeeded by a question mark or forward slash), the follow-up study described in Chapter 7 does. Here, we found only minimal mean differences between the two categories (MD=.002), which were not significant. As such, we note the differences between benign URLs that do have a path or query component and those that do not, but do not generally divide them further into subcategories.

Next, this study explicitly includes phishing URLs using the random RD base to

Table 5.7.: Differences between benign URLs with and without query

	Valid	Mean	SD
Plain	45	0.962	0.066
Query	45	0.710	0.343

Table 5.8.: Differences between subdomain subcategories

	Valid	Mean	SD
Subdomain-first	40	0.625	0.463
Subdomain-middle	42	0.988	0.077
Subdomain-end	43	0.977	0.152
Subdomain-only	43	0.520	0.363

test, whether there are differences between path and query posing here, however the results again indicate no significant differences ($MD = .012$, $p = 1$ for a Wilcoxon signed-rank test, as a deviation from normality was detected).

Taking a look at the effect of including a path component in phishing URLs in general, i.e., the difference between phishing URLs without any path component and those with a path or query, we again did not find the existing differences ($MD = .009$) to be significant according to a paired-samples t-test.

5.3.4. Subdomain Posing

The fourth research objective is to analyze, whether the position of the target domain in the subdomain when using subdomain posing makes a difference. In particular, subdomain URLs can be split into four categories: (1) long subdomain with the target domain appearing first, (2) long subdomain with the target domain appearing in the middle, (3) long subdomain with the target domain appearing right before the actual RD, and (4) target domain is the only subdomain.

The results of the user study seem to suggest, that the position does have an effect on classification performance, as URLs with the target appearing first or as the only subdomain were detected less well than those with the target domain in the middle or end of the subdomains (MD$\geq$ .352, see Table 5.8). These differences are confirmed as significant by a repeated-measures ANOVA with the four subcategories as repeated-measures factor (N=39, degrees of freedom were corrected using Greenhouse-Geisser $\epsilon = .563$): $F(1.689, 64.190) = 32.517, p < .001, \eta_p^2 = .461$. Post-hoc tests using Holm's correction indeed confirm significant differences between the above mentioned sub-categories, with large effect sizes ($d \geq 1.167$). While the mean differences between categories (1) and (4) are still relatively large (MD=.105), they are not significant according to the post-hoc tests ($p = .109$), nor are differences between (2) and (4).

To summarize, subdomain posing was significantly more complicated to detect in our study when the target domain makes up the left-most part of the FQDN. We therefore combine the subdomain-first and subdomain-only subcategories (subdomain-first) as well as the subdomain-middle and subdomain-end categories (subdomain-end) in our following analyses in this chapter.

Table 5.9.: Differences between Http credentials and subdomains

	Valid	Mean	SD
Subdomain-end	44	0.972	0.155
Subdomain-first	44	0.532	0.359
Http credentials	43	0.672	0.364

5.3.5. Http Credentials and Subdomain Posing

While we did not encounter large amounts of URLs utilizing http credentials in our previous analysis (see Section 4.1.1), determining whether they make a difference in classification performance is still useful for our categorization. As such, the third research objective is to determine, whether there are differences between using http credentials to include the target in the URL compared to using subdomains, since both techniques insert the target at the beginning of the URL.

The study confirms, that http credentials were easier to detect than subdomain posing where the target is at the beginning of the subdomain (see Table 5.9). This difference (MD=.140) is significant in a paired samples two-tailed student's t-test ($t(41) = 2.581, p = .014, d = .398$) with a moderate effect size.

We therefore differentiate http credentials from the other categories in our overall comparison below, but did not include them in the anti-phishing learning games described in Chapter 6 due to their low prevalence in the collected phishing URL dataset.

5.3.6. Typosquatting

This research objective is about differences between the subcategories of typosquatting URLs. The in-depth categorization defines seven different subcategories, with the goal of offering a complete view on which attributes might have an effect on classification performance. The subcategories include differences between different operations (e.g., replacing or adding a character), random and targeted operations (e.g., replacing a character with a different random character, or replacing with a visually similar character), and differences between vowels and consonants (e.g., removing a vowel or a consonant from the target name).

As can be seen in Table 5.10, the mean differences for all typosquatting categories in this study fall into a range of .130, which is the difference between *remove consonant* and *replace similar*. Null-hypothesis testing confirms significant differences between some of the subcategories, as the RM-ANOVA (N=40) over the seven subcategories indicates significant differences ($\epsilon_G = .752$, $F(4.511, 175.926) = 3.119, p = .013, \eta_p^2 = .074$). Post-hoc tests using Holm's correction are only significant for the comparison of *replace similar* and *remove consonant* ($p = .033, d = .687$).

Our results therefore stand in contrast to the findings of the previous study by Spaulding et al., who found that replacing similar characters and character omissions both resulted in the lowest accuracies [SUM17]. Additionally, classification performances for the typosquatting URLs in the tests in Chapter 6 were generally lower compared to this study. We discuss these differences in more detail in the discussion in Section 5.4.

Table 5.10.: Differences between typosquatting techniques

	N	Mean	SD
Add random	43	0.853	0.264
Duplicate letter	42	0.907	0.230
Replace random	44	0.941	0.180
Remove consonant	45	0.841	0.219
Remove vowel	43	0.895	0.257
Replace similar	43	0.971	0.098
Swap adjacent	42	0.905	0.162

Table 5.11.: Differences between the three RD placement categories

	N	Mean	SD
Combosquatting	43	0.756	0.286
TLD change	39	0.774	0.292
Typosquatting	45	0.882	0.164

5.3.7. Registrable Domain

Having found only minor differences between the different typosquatting subcategories, we merge them into one and focus on the three different categories for impostor URLs where the target is placed in the RD next. Here, the research objective is to analyze the different methods that focus on the RD of the URL for significant differences: combosquatting, modified TLD, and typosquatting.

Looking at the classification means (see Table 5.11), we find similar values for modified TLD and combosquatting, which are both lower than the detection rate for typosquatting URLs, and in particular even lower than the least-well detected typosquatting sub-category *remove consonant*. A repeated-measures ANOVA ($N = 39$) with the three categories as factors confirms significant differences between them: $F(2, 76) = 7.763, p < .001, \eta_p^2 = .170$. The post-hoc tests using Holm's correction confirm the expected differences: typosquatting domains were detected significantly better than both combosquatting ($p = .003, d = .640$), and modified TLD ($p = .003, d = .624$) URLs, with no significant differences between the two less well detected categories ($p = .929, d = .017$).

5.3.8. Additional Results

In addition to the main study results presented above, we also followed a number of minor questions about specific categories and the study in general.

We found no significant differences between using URL encoding to encode the last dot of the subdomain or not when using the *URL encoding* category, even though the two URLs with encoded dots were detected with lower accuracy (two-tailed Wilcoxon signed-rank test due to a deviation from normality, $W = 54, z = 1.177, p = .232, r = .385$). Interestingly, adding a keyword before the target in combosquatting attacks seems to be significantly easier to detect than adding it after the target (two-tailed Wilcoxon signed-rank test due to a deviation from normality,

Table 5.12.: Differences in performance scores between simplified URL categories

		N	Mean (SD)
Benign	Path	45	0.710 (0.343)
	Plain	45	0.962 (0.066)
Phishing	Http credentials	43	0.672 (0.364)
	Path	43	0.948 (0.215)
	RD	43	0.757 (0.269)
	Subdomain-end	44	0.972 (0.155)
	Subdomain-first	44	0.532 (0.359)
	Typosquatting	45	0.882 (0.164)

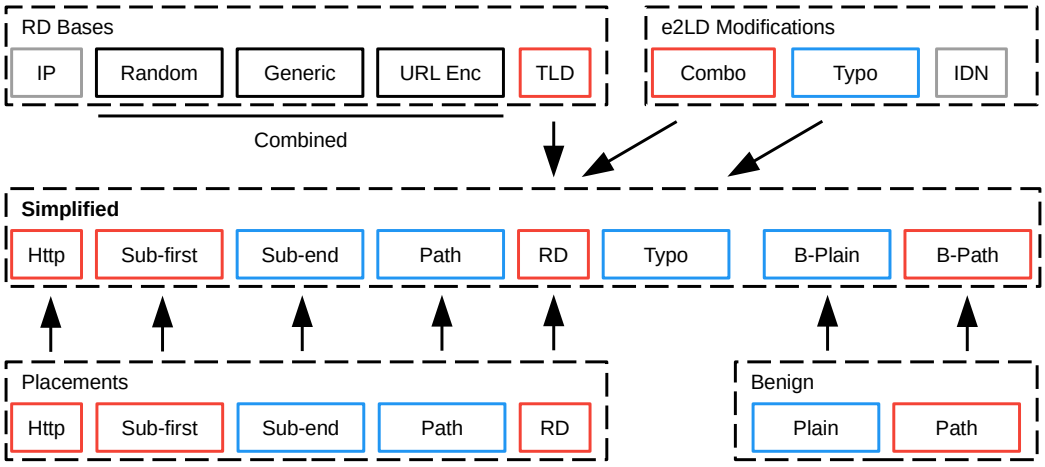


Figure 5.1.: Mapping of URL categories to simplifications. Colors indicate categories that were detected well (blue), not well (red) or not tested (gray).

$W = 31.000, z = 2.575, p = 0.10, r = .674$).

Finally, we perform an overall comparison between the merged categories where we found no or only minor differences above, resulting in eight categories (see Figure 5.1 for the merging process and Table 5.12 for resulting mean performance scores): (1) plain benign, (2) benign with query, (3) phishing using http credentials, (4) phishing with the target in a full path component, (5) phishing with the unmodified target e2LD in the RD (i.e, combosquatting and modified TLD), (6) phishing using subdomain posing where the target does not appear first, (7) phishing using subdomain posing where the target does appear first, and (8) phishing using typosquatting. A repeated-measures ANOVA ($N = 40$) comparing the resulting simplified categories reveals significant differences: Greenhouse-Geisser corrections ($\epsilon = .403$) were applied to the degrees of freedom, $F(2.824, 110.114) = 17.381, p < .001, \eta_p^2 = .308$. Post-hoc tests using Holm's correction confirm several differences splitting the phishing categories into those that were detected well (path posing, subdomain-end, and typosquatting) and those where participants had troubles (http credentials, RD, and subdomain-first), as well as the two benign categories (URLs with path were more complicated to classify than those without).

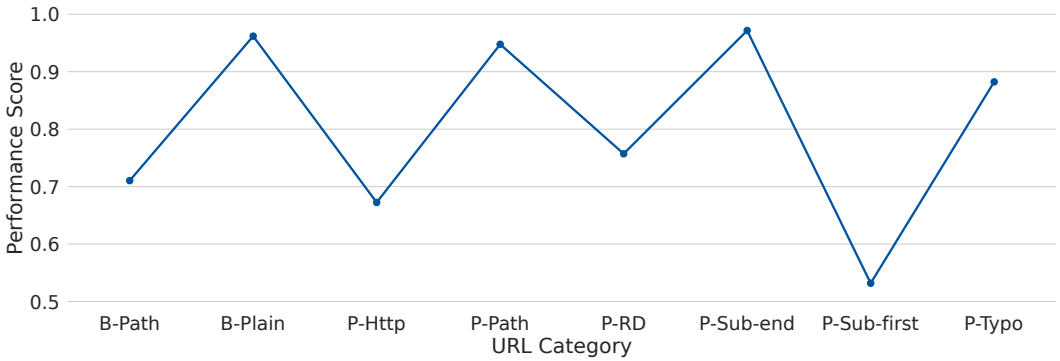


Figure 5.2.: Differences between the mean performances of simplified URL categories.

In particular, it seems that any type of complexity affected the classification performance for benign URLs. For phishing URLs, a general trend seems to indicate that users mostly focus on the start of the URL, as categories with the target in the beginning were generally detected worse. An exception is the typosquatting category, which was detected well in this study, but resulted in different performance when compared to related work and the study presented in Chapter 6. For an overview of the simplified categories in comparison, see Figure 5.2.

5.4. Discussion

The goal of this chapter was to define a categorization for phishing URLs, with a focus on impostor URLs. A user study performed to assess the proposed categorization confirmed, that there are significant differences between different categories of URLs, thus motivating a particular focus on URLs that are harder to detect in user education. In the following, we discuss potential shortcomings of the study, followed by an overview of its implications.

5.4.1. Limitations

There are several threats to the validity of the user study based on the sample. First, the sample size is relatively small and unlikely to be representative. While the repeated-measures design makes a comparison between different categories possible without the need of large sample sizes, it is still possible that the results of the found differences only generalize to a specific user group. Some of the participants were also recruited by the supervisor of the study directly, thus potentially introducing a bias. However, this is unlikely to have had a significant effect on the differences between categories, due to the usage of repeated measures in the analysis, and the fact that there was no priming that may have led to an unconscious bias towards any category, as the aim was to test the categorization in general. We further note, that a follow-up study (see Chapter 7) that made use of a subset of the URLs from this study resulted in similar results for all categories, and that the previous study by Reynolds et al. results in similar differences between the URL categories that are similar in their study [Rey+20].

The categorization itself, even though it was carefully created to capture as many general different phishing URLs as possible, is also not complete. For example,

the categorization generally does not include URLs where a target is referenced more than once. Moreover, the user study did not include URLs where a modified target (e.g., similar to typosquatting) appears in a subdomain or in combination with combosquatting. These cases correspond to the *relaxed typosquatting* rule presented in Section 4.2, which matched 16.09% of all impostor domains. While we argue, that this combination of rules is likely to make the URLs easier to detect than using either one of the compounding rules, this effect might be studied in more detail in the future. Furthermore, we found that typosquatting URLs had a much higher success rate in a different user study (see Chapter 6), which might indicate the existence of additional, currently hidden variables that influence the classification performance and are not captured by the current categorization. This would additionally explain the differences to the study by Spaulding et al. [SUM17], who also observed lower accuracies overall, and a different order when ranking the resulting performance scores for the sub-categories. One hypothesis that may be tested in future work is the effect of font choice on classification performance, as it is one of the factors that might influence typosquatting domains in particular. As such, while the categorization might stand as a starting point and general reference, it is not complete and unable to explain the effects of all manipulation techniques on classification accuracy.

After finding significant differences between the URLs including known and unknown services, we decided to remove all URLs of unknown services from the analyses. While this likely lessened the influence of familiarity with a service on the classification performance, it is possible that this decision introduced different biases. For example, it is possible that the decision favors users who are more knowledgeable about computer science education, as they are more likely to be familiar with the services used in the study, and might also have a higher performance overall due to this knowledge. However, we argue that this bias is less severe than the inclusion of unknown services, due to the repeated measures design of the study.

As the focus of the study was on categories of phishing URLs, the results for benign URLs are less comprehensive. At this point, the study did not include benign URLs with subdomains other than `www`, nor URLs with a path. While we did not find differences between path and query URLs in the study presented in Chapter 7, we did find significant differences when using less common subdomains in Chapter 6. We also note, that the benign URLs used in the test were randomly generated and thus often did not result in the actual login URL of a given service. This might have influenced users of the service in question, who might be familiar with the exact login URL and noticed differences, which led them to classify the URLs as phishing. However, we did not find indications of this being the case in our analysis, as the complexity of the URL seems to be the main factor when predicting classification outcomes. Furthermore, it is likely that collecting a representative dataset of benign URLs, which could then be used for a benign categorization, is a complex task, as it requires accurately simulating or capturing the browsing behaviors or users. Still, creating such a dataset and categorization, and using it in user education might be an interesting approach for future work.

Finally, we note that the tests might not be a good basis for making decisions on simplifications for categories, as they are not proof that there are no differences between similar categories (i.e., absence of evidence is not proof of absence). The simplifications used throughout the study and discussed below might therefore be improved in the future, even though we still make use of them to simplify the analyses

in the following chapters.

5.4.2. Implications and Generalizations

The study results presented in this chapter indicate a pattern in how users parse URL, where they focus mostly the left-most part of any given benign or phishing URL. These results might be used to improve educational resources by focusing on the specific URL categories that were not classified with high accuracies, or by teaching a more robust URL reading process in general. We focus on several of the problematic categories in the approaches presented in the next part of this thesis.

Due to the large number of possible subcategories and modifier combinations, the categories used in tests and analyses in later chapters use several simplifications. One of the most common simplifications for phishing URLs in this thesis is to focus only on the placement of the target. This results in the categories *http credentials*, *subdomain*, *RD*, and *path*, as well as a category for *random* URLs without a reference to a target. This is the basis for the categorization taught in the games, with only slight modifications (i.e., no http credentials, but adding IP addresses as a separate category, see Chapter 6 for details on this decision, and Table A.11 in Appendix A.3 for details on how the categories of this chapter are presented in the games). It is based on the assumption, that the categories that are merged for these approximations are sufficiently similar that grouping makes sense, which we confirmed for the URLs used in the study of anti-phishing educational games in Chapter 6, as well as those studied in this chapter.

This means ignoring RD base categories, where we typically use random or generic bases in the games, differences between path, query, and fragment, which did not differ significantly in this study, as well as differences between subdomain subcategories, where we typically only use URLs where the target appears first in later chapters. For differences between e2LD modifications, typosquatting was easier to detect in this test, however we did not confirm this for the study in Chapter 6, where the category was the most complicated to detect.

Next, our results indicate significant differences between URLs of unknown and known services, where URLs of unknown services were classified with worse performances. This opens a number of questions regarding the validity of test scores of phishing susceptibility tests in general if services in the test are unknown, as they might no longer accurately reflect the participants' abilities to detect phishing URLs. We therefore generally recommend including a survey for service familiarity when testing phishing classification ability, that at least differentiates used or known from unknown services.

For user studies with URL tests in general, the results of this study implicate that some care has to be taken when selecting URLs for the tests. While not all categories and sub-categories differ significantly, researchers that aim to cover a wide range of possible phishing and benign URLs in classification tests might consider including URLs from all categories, e.g., several different typo manipulations and subdomain posing where the target does and does not appear first. Even for smaller tests, we recommend including several categories of benign URLs, as plain URLs were easier to detect here, and phishing URLs where the target does or does not appear first, as this appeared to be the best indicator for detection performance in this study.

In all, we found that the position of the target seems to be a strong indicator

of the expected classification performance for impostor URLs, as categories with the target appearing first were generally harder to detect in the study than others. While there were some differences between subcategories targeting similar parts of the URL or using similar manipulation techniques, we note that it seems sensible to use simplified categorizations to reduce the overhead of comparisons and analyses. As such, simplifications are used in the analyses of the following chapters. The observed differences in performance scores between the categories imply, that URL classification tests, for example to determine the effectiveness of educational interventions, should include a diverse set of URLs that cover a large variety of categories to ensure that the results of the study cover a wide range of possible phishing attacks. On the other hand, differences to previous studies for similar URL categories might indicate additional hidden variables which determine the outcome of a given URL in a classification task, and which might be explored in more detail in the future.

Part II.

Human Factors

Chapter 6

Game-based Anti-Phishing Education

The first approach to phishing prevention discussed in this thesis is user education. In the scope of this thesis, we look at anti-phishing learning games as an educational intervention aiming at providing a motivational and safe environment to learn and train phishing classification.

In this chapter, we are concerned with the question, how learning games can be utilized to impart knowledge about phishing URLs. To this end, we analyzed the content of several learning games and derived directions for the design of new learning games, which were then implemented and evaluated. The results of the evaluation give insights into the effect of anti-phishing learning games as educational interventions on the URL classification performance in an ideal setting.

Parts of this chapter were previously published in and adapted from [5; 7; 10] and [11].

Contributions: The main contributions of this chapter are the analysis of existing learning games, and the design, implementation, and evaluation of four new anti-phishing learning game prototypes. The analysis of existing learning games was performed in collaboration with René Röpke and Klemens Köhler, and was previously published in [11]. Designing and developing the learning games, as well as the design and evaluation of the user studies to test them, was joint work with René Röpke and previously published in [5; 7] and [10]. The evaluation of study results was adapted from the previously published versions by removing responses where participants were unfamiliar with the service of the classification tasks, as well as testing for differences between different categories of benign URLs, and is therefore a new contribution of this thesis adapted from the collaborative work with René Röpke.

6.1. Systematic Literature Review and Research Objectives

To understand the state of the art of anti-phishing learning games, and derive research directions for new prototypes, we conducted an analysis of existing games and their publications (see [11] for additional details). The analysis includes aspects from several different perspectives, ranging from learning content to game mechanics. Based on these findings, we identified several goals for designing and evaluating new game prototypes, which were partly implemented by the prototypes presented and evaluated in the next section.

6.1.1. Review Methodology

The analysis of existing games is based on a systematic literature review of published papers on game-based anti-phishing education performed in 2020. To this end, the digital libraries ACM Digital Library[1], Google Scholar[2], and IEEE Xplore[3] were queried using the keyword `phishing` in combination with one of the keywords `educational game`, `serious game`, `learning game`, `game based learning`, or `competence developing game`. After removing duplicates and publications that are not written in English, this resulted in 282 publications, which were then further reduced by examining the titles and abstracts for relevance. In all, the final dataset of relevant publications on game-based anti-phishing education (called *Publications on Games* or *POG* dataset for short) contains 54 entries (see Table A.5 in Appendix A.3).

In the following, we present an overview of the analysis of the POG dataset, first for game mechanics based on Bloom's Revised Taxonomy (BRT) (see Section 2.4), followed by an in-depth analysis of the games' content for all games freely available for play.

6.1.2. Learning Goals and Game Mechanics

First, we analyzed the skills required to progress in the game based on the BRT. This is based on the assumption, that the knowledge transfer from a game to reality requires a certain level of alignment between the tasks or mechanics in the game and the task that is to be performed in the real world. In the case of phishing detection, this task is usually a classification task, which is based on some previously defined criteria. For example, to classify a given URL, the user would have to apply a URL parsing technique, and use the results to determine where the URL leads. They would then have to judge, whether this location is legitimate. Due to the complexity of this classification task, a level on the cognitive axis of the BRT of at least *apply* is preferable, as it has been hypothesized that this is the first activity that results in a *"significant impact to the external environment of the acting person"* [WW13], thus ensuring that the learned skills translate into a different, real-world environment.

A manual analysis of the publications in the *POG*, by applying the BRT to the described game mechanics that are required to advance in the games and comparing them to skills required in the real world to detect phishing attacks revealed, that most games focus on lower levels of the BRT, both in the conceptual as well as the cognitive axes. In particular, no game with a focus on anti-phishing education required evaluating or creating knowledge, thus posing the question whether these teaching goals might be helpful in conveying the content more effectively.

6.1.3. Game Content

The results of the previous section show that many games require factual and conceptual remembering and understanding for in-game advancement. In this section, we add an additional dimension to this analysis by examining the specific topics the games present and teach during a typical playing session. This reveals which subjects are popular or missing in current anti-phishing games, the level of detail with which

[1] `https://dl.acm.org/` online, accessed 2023-01-25
[2] `https://scholar.google.com/` online, accessed 2023-01-25
[3] `https://ieeexplore.ieee.org/` online, accessed 2023-01-25

they are presented and whether specific topics are missing even if a broader subject is included in the game. To this end, this section takes a closer look at the content of available digital learning games and presents an analysis of their content based on a number of subjects and topics. We analyzed the 9 available games in the *POG* data set (cf. Table A.5 in Appendix A.3), and also extended our collection by 4 reference games without academic publications that were found using the Google search engine[4] and represent games about phishing emails, URLs, and websites as offered by several companies ($9 + 4 = 13$ games, see Table 6.1).

After playing the games, we found that they can be broadly categorized by their main subject, which was *URLs and websites* for 4 games, *Emails* for 4 games, and *Other/Various* for 5 games (see Table 6.1). The games in the *Other/Various* category do not focus on phishing, and instead include learning content about phishing as well as several other topics of online security.

In the next step, we defined more specific topics for each subject, again based on the actual content of the games, while also including topics based on the advanced phishing attack techniques described in Chapter 4. We argue that covering a larger number of specific topics results in a more complete knowledge of a subject, making it less likely that users fall victim to a class of phishing attacks.

For URLs, we defined topics based on (1) the structure of URLs, (2) manipulation techniques, and (3) Others. The manipulation techniques (2) are based on the modification categories from Chapter 5, and include advanced techniques like abusing Internationalized Domain Names (IDNs, see, e.g., [ES18]). For (3), we particularly looked at a selection of URLs that users are likely to encounter in their daily browsing or in specific phishing attacks, namely redirections, and link-shortening.

For emails, we looked for (a) specific traits, (b) sender spoofing, and (c) email attachments. Specific traits (a), e.g., the lack of a personalized greeting, spelling mistakes, or urgent requests, are a common topic in games, even though they can sometimes be easily avoided by attackers. There are also several types of sender spoofing (b), with differences in the display-name and sender address [Res08]. For email attachments, most games with emails as subject included at least examples of how malware can be attached to phishing emails. We also looked for information on the email structure, e.g., on the existence of email headers or on different sender identities, which can often be used to detect anomalies in emails [HW18], but was not explained in any of the analyzed games.

Next, we include an analysis on advanced topics in Table 6.1, which includes services that host user-generated content (e.g., Dropbox[5]), usage of hex-digits to obfuscate IP addresses in URLs, as well as usage of pop-ups in phishing attacks (which could, for example, be used in picture-in-picture attacks [Jac+07]).

Lastly, we looked at various auxiliary topics, including different message delivery methods (e.g., SMS, social media), advanced protection strategies (e.g., Multi-Factor Authentication (MFA)), and common traits of the body of phishing websites.

The games were processed as follows: We first downloaded and set up all available digital games. In this step, we only looked at games that were generally available online, requiring no payment or membership. We then analyzed the games from a player's perspective while keeping note of the subjects and specific topics appearing

[4] `https://www.google.com/` online, accessed 2023-01-25
[5] `https://www.dropbox.com/`, last accessed on 2020-04-27

in the game. The games were rated for each specific topic according to four classes: 0 - *does not appear*, 1 - *does appear in game elements or examples*, 2 - *mentioned but not fully explained*, 3 - *fully explained*. These distinctions are based on the assumption that detailed explanations (class 3) are more likely to convey an understanding of the actual detection or protection strategy, while shallow descriptions (class 2) can be confusing and might lead to misunderstandings (e.g., [She+07]).

Still, we argue that these aforementioned instances of classes (2) or (3) are more likely to be actively considered by players than information that may be hidden in examples or game elements (class 1). An example for the different classes would be: (1) showing an email with a potentially malicious attachment in an example without any further explanation, (2) the instruction to examine email attachments carefully for malicious content without further explanations what to look for, (3) an explanation of how email attachments can be abused to attack users, including an explanation of different file endings and their corresponding attack surfaces.

Note that we always focused on phishing when presented with a choice in the games during our analysis. We do not claim to have encountered all examples or exhausted all possible selections, successes, and failures, though we did try to cover all content that is related to phishing, or, if the focus of the game is phishing, completed at least one gaming session.

Table 6.1.: Overview of analysis results regarding anti-phishing learning game content

Subject	URLs & Websites				Emails				Other				
	Anti-Phishing Phil [She+07]	codecanyon[a]	NoPhish [BC14]	OpenDNS[b]	Birds Life [WJZ18]	Sophos[c]	whatdothack [Wen+19]	WithGoogle [d]	ATMSG/CSAG [Huy+17]	cyberaware [GKG15]	CyberCraft [Lu18]	GHOST [KW18]	whatthehack [Gey19]
URL structure	3	1	3	2	0	0	1	2	1	0	0	0	0
URL manipulations	2	1	3	1	0	0	1	2	1	0	0	0	0
Other (URL)	2	1	3	2	0	0	1	2	1	0	0	0	0
Email traits	0	0	0	0	2	1	2	2	1	0	1	0	0
Sender spoofing	0	0	0	0	2	1	1	1	1	0	0	0	0
Attachments	0	0	2	0	1	0	2	2	1	2	0	0	0
Advanced	2[a]	0	0	1[b]	0	0	2[b]	0	2[c]	0	0	0	0

[a] Hex IP addresses, [b] Hosting service abuse, [c] Pop-ups

[a] https://codecanyon.net/item/anti-phishing-awareness-game/20935555, last accessed on 2020-04-16

[b] https://www.opendns.com/phishing-quiz/, last accessed on 2020-04-16

[c] https://www.sophos.com/en-us/lp/games/play-spot-the-phish.aspx?cmp=35375, last accessed on 2020-04-16

[d] https://phishingquiz.withgoogle.com/, last accessed on 2020-04-16

Results

In all; we were able to obtain 13 games, which includes the 4 reference games that were found using a search engine and 9 games from the *POG* data set (cf. Table A.5 in Appendix A.3).

Our main finding (see Table 6.1) is, that few games include detailed explanations of conceptual knowledge. There are only two games (NoPhish and Anti-Phishing Phil, both of which focus on URLs), which provide in-depth explanations on how to determine the RD of a URL. NoPhish goes a step further and includes explanations for the complete structure of URLs, however, it is only available in German. Anti-Phishing Phil also includes detailed explanations on how to locate the RD, however, it misses some details compared to NoPhish, including information on subdomains and the separation of different domain labels.

For games in the *Other/Various* category, it is interesting to note, that they tend to have a more developed story (ATMSG, Cybercraft, and GHOST), while other games have no or only few story elements, with a focus on education and exercises. Apart from that, games in this category typically do not offer detailed explanations about phishing or anti-phishing protection strategies.

The remaining games typically cover a specific subject (e.g., URLs), and present an example for classification followed by revealing the correct answer and an explanation on how to detect the malicious parts. Note, that none of these games ask the user for the reasons for their classification decisions. As such, users that guess or use a sub-optimal detection strategy might still be rewarded with positive results, which might lead to misconceptions or confusion.

In general, we did not find any games that offer detailed conceptual knowledge about emails; none of the games explain email headers or how to verify email authenticity. Though sender spoofing is a common theme, the only way to really verify an email's authenticity as presented by the analyzed games is to contact the help desk or IT support, which is usually not a viable option for private end-users. The lack of detailed explanations might be due to the game creators' perception of the email structure being more complicated, email sender `From:` spoofing being uncommon, or email headers being less reliable in identifying phishing emails than other indicators. As for attachments, seven games include information on malware in attachments or links in emails. Two games also address other message types beyond email (e.g., instant messaging).

An additional finding is the lack of advanced phishing techniques in available games. None of the games include information on IDN or percent-encoding in URLs, and only two include hosting service abuse. This might be due to the low expected occurrence of this type of attacks in phishing, or due to the fact that the emergence of advanced techniques are comparatively recent developments. Note that users, depending on their locality, might be confronted with techniques such as IDN in their benign browsing activities as well.

On the other hand, six games include examples of benign websites with uncommon characteristics. Examples here include the use of `www3` as subdomain instead of `www` or information on unexpected domain names (e.g., `dropboxmail.com` instead of `dropbox.com`). We argue that including these examples can potentially demonstrate to users, that benign websites might also exhibit uncommon behavior, thus reducing false positives in their decisions. Many games also include hints on additional

protection strategies, like using a search engine to determine the authenticity of a website (3 games) or hovering over a link to display the actual destination (3 games). Using MFA was recommended less often, only one game mentioned it as a method to protect against phishing.

6.1.4. Requirements for New Prototypes

To summarize, our analysis of existing game-based anti-phishing education interventions revealed, that the focus of the games is often spread over several topics. The games furthermore mainly cover the lower dimensions of BRT, indicating that remembering and understanding factual and conceptual knowledge are the main skills required to advance in the games. We therefore identified a research gap in developing anti-phishing games that focus on higher cognitive processes. To this end, the new game prototypes presented in the following sections, as well as an additional game about email-based phishing attacks [8], explicitly require higher-order cognitive processes (analyze and create) to advance in the games.

As we previously found (see Section 5.3), that familiarity with a service can affect the classification performance, we were also interested in whether this fact was incorporated in any of the games. As we did not find games making this distinction, however, we added a personalization option to one of the prototypes to test, whether differences in familiarity also affect the gameplay.

As for the content of the games, we found that when phishing is covered in more detail, the games either focus on phishing websites and URLs, or on phishing emails as a typical delivery method, and seldomly both. The only games offering in-depth explanations were about phishing URLs, teaching the structure of URLs in detail and how this knowledge can be applied to detect URLs of phishing websites. We decided to use URLs as the subject for the game prototypes as well, as the URL is a robust indicator that appears in all website-based phishing attacks. Few games, however, include advanced techniques, such as the usage of hosting services to host a malicious website, or using URL encoding. We decided to follow the example of previous games and focus on conveying basic knowledge about URLs first, i.e., the URL structure and manipulation techniques, in our prototypes. The main reasons for this choice are that the advanced attacks rarely appear in our dataset of phishing URLs, thus making them currently less relevant for detecting phishing URLs. Second, we still include an unusual attack (using URL encoding) in the evaluation of the new prototypes, even though URL encoding does not occur in the games. In this way, we can provide some evidence on whether knowledge can be transferred to concepts that were not explicitly taught in the games, and would thus offer some protection against previously unknown types of attacks.

In all, we created four game prototypes: three games that only differ in a single aspect to directly compare complex and binary decisions as game mechanics, as well as the effect of personalization, and one game that requires players to create URLs to advance in the game. The next section describes the prototypes in more detail, followed by the user study that was conducted to test their effectiveness and differences between their mechanics.

6.2. Game Prototypes

In order to answer the questions that were posed by the analysis of existing games, we developed four learning game prototypes. We started by creating two games with new game mechanics, called *All sorts of Phish* and *A phisher's bag of tricks*[6]. *All sorts of Phish* requires players to classify a given URL into several different buckets (benign, 5 different malicious categories) instead of a binary classification. The game *A phisher's bag of tricks* offers a more constructive approach which involves the application of manipulation techniques to create malicious URLs. As such, we refer to the first game as the *analysis game* and the second game as the *creation game* (see Figure 6.1 for screenshots). For evaluating a baseline, existing games were either not available as open source or were not available in the language spoken in our country of origin, or were not implemented as browser games for desktop devices. Thus, they were not adaptable to be compared to our prototypes in a coherent setup in our study. We therefore implemented our own baseline, which is an almost exact clone of the analysis game, differing only in the main game mechanic, which is changed to a binary decision scheme. We refer to the baseline game as the *decision game*. Finally, we also created a version of the analysis game that includes personalization options, which we call the *personalized game* (see [9; 10] and [12] for more information on the personalized game). Personalization is realized by offering players of the game a choice of services before the game starts, where they can select the services they are familiar with. The game automatically creates URLs based on this choice, but is otherwise identical to the analysis game. Note, that the personalized game explicitly includes both known and unknown services, to facilitate the analysis of differences between known and unknown services during gameplay.

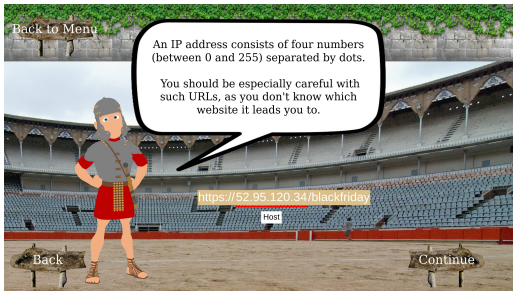
6.2.1. Learning Goals

Throughout each of the four games, players learn about the structure of URLs and the method of URL parsing, i.e. reading a URL and identifying the different components and their purpose. Players are then introduced to a set of manipulation techniques showcasing how benign URLs can be manipulated to become malicious URLs that still look trustworthy.

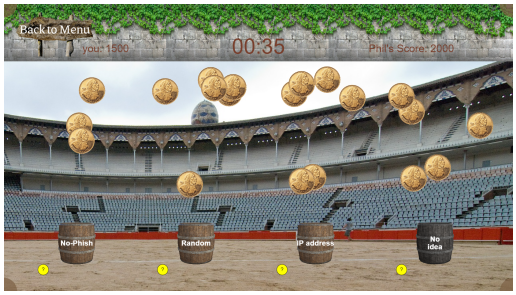
While the overall learning goal is for end-users to check the URL and classify it as either benign or phishing before clicking on it or before submitting sensitive data on a website, more fine-grained learning goals are defined for the four games and matched with the cognitive process categories in the BRT (see Section 2.4). Table A.6 in Appendix A.3 provides a complete overview of all learning goals and the mapping to each game. The learning goals of the games overlap when it comes to factual and conceptual knowledge in the lower-order cognitive processes, i.e. *remember* and *understand* in BRT. However, the goals explicitly differ for higher-order cognitive processes: In the analysis game and its derivatives, the learning goals address the cognitive processes of *analyze* and *evaluate*, while the goals in the creation game are focused on *apply* and *create*. The learning goals of the decision game, personalized game, and the analysis game are identical.

Since we used BRT as a design aid for game development, the current prototypes focus only on knowledge about phishing, and do not incorporate situational

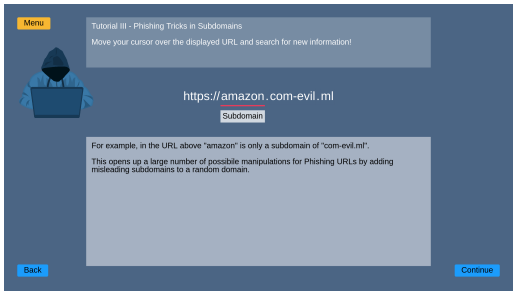
[6]Both games are available at `https://erbse.elearn.rwth-aachen.de/en/`



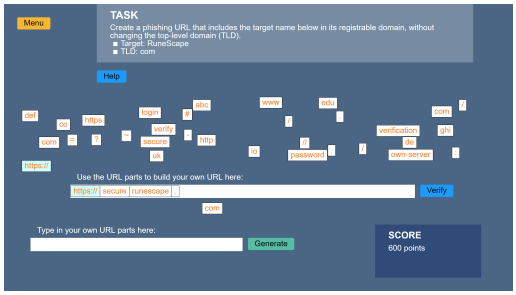
(a) Example tutorial section in the analysis game. Players can hover over the URL to reveal additional information.



(b) First level of the analysis game. Players have to classify given URLs, which are hidden behind coins.



(c) Example tutorial section in the creation game. Players can hover over the URL to reveal additional information.



(d) Level of the creation game. Players create malicious URLs by combining URL parts via drag-and-drop.

Figure 6.1.: Screenshots from the creation and analysis games.

awareness (further discussed in Section 6.6). Therefore, the games are limited to URL-based phishing and might not make up a comprehensive phishing education without additional information.

6.2.2. Game Content

All four games consist of tutorials to impart knowledge, followed by levels that challenge the players' understanding of the topic. In the tutorials, each part of the URL structure is introduced together with illustrative phishing URLs, that contain suspicious keywords in the corresponding part of the URL. The games teach the general structure of URLs, with a focus on the three main URL parts *subdomains*, *RD*, and *path*. After playing the game, players should therefore be able to identify the RD, analyze it for occurring manipulation techniques, and base their classification decision on the outcome of this process. We used a simplified URL categorization in the games (see Figure 6.2 for the merging process and Table A.11 in Appendix A.3 for an exact mapping from the detailed categories to the simplified categories used in each of the games), so as not to confuse players with overly detailed descriptions of the different (sub-) categories. This results in the following categories: *No-Phish* (for benign URLs), *IP*, *Random*, *Subdomain*, *RD*, and *Path*, which are introduced successively as players advance in the game. We furthermore added URLs with URL encoded RDs in the user studies, to test how users react to URL categories which were not part of the games. The resulting categories are similar to the simplified categorization presented in Table 5.12 in Section 5.3.8, but we differentiate phishing

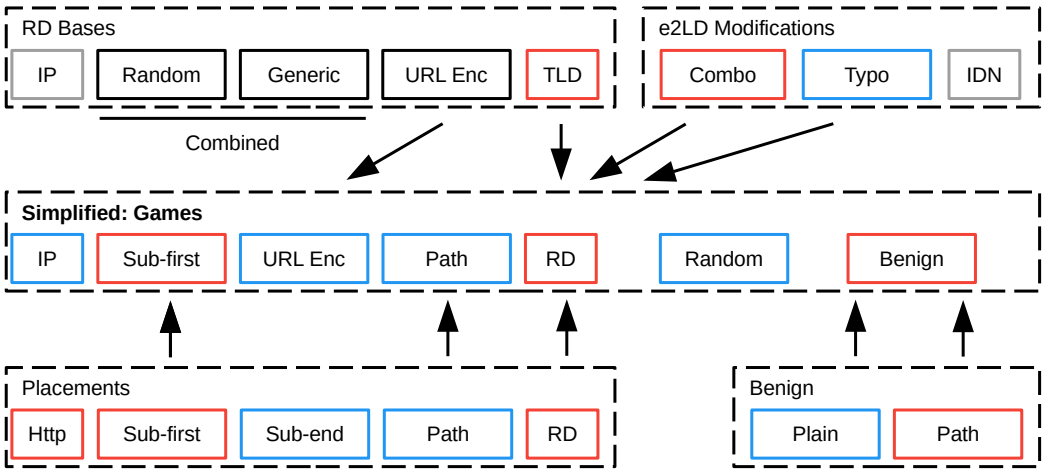


Figure 6.2.: Mapping of URL categories to simplifications used for the games. Colors indicate categories that were detected well (blue), not well (red) or not tested (gray).

URLs with IP addresses or URL encoding as RD bases, since they only appear in one or none of the games, and merge all benign URLs, as well as all RD placements, into one category to simplify the comparison. Furthermore, preliminary testing showed that the creation game in particular takes a long time to complete and thus, not all categories were included in all games (see Table 6.2 for a mapping of categories to games). The resulting content for the games still covers the URL categories that were not detected well in the previous study from Chapter 5, since both games focus on determining the RD in the presence of subdomains, thus teaching users to avoid impostor URLs of the subdomain-first and RD categories. An exception is the http credentials category, which was not included in the games to reduce the time required to play the games, and since http credentials were not common in our analysis of impostor and benign URLs in Chapter 4. We describe the creation process for URLs in the games, as well as which URL categories were used to test the games in the URL classification tests in more detail in Section 6.3.

6.2.3. Game Design

The objective of the analysis game and its derivatives is for players to classify multiple URLs by sorting them into the different URL categories. While the tutorials successively introduce the URL categories defined below (Section 6.3), levels allow players to practice their knowledge. A level in the analysis game is time-bound (using 60-seconds timer) and challenges players to beat an adaptable score in order to advance in the game. The adaptable timer and score allow for difficulty adaptation, which is not yet exploited in the scope of the study presented in the following. In the level, players are presented with multiple URLs (depicted as draggable coins which flip when clicked to reveal a URL) and a set of different buckets, each representing a specific URL category. The game offers a bucket for benign URLs labeled *No-Phish* as well as one bucket for each already introduced phishing URL category (e.g., *Random*, and *IP address* in the first level, depicted in Figure 6.1). An additional bucket

labeled *No idea* is available, allowing players to discard URLs they are not able to classify confidently. Players can sort URLs into categories by dragging coins or URLs into buckets. When a coin is dropped into a bucket, players receive immediate feedback about the classification result via a colored aura over the bucket and their scores are updated, i.e., increased for correct decisions and decreased for incorrect decisions (discarding URLs does not change the score). The level is finished when the timer runs out. Next, feedback is presented by a review of exemplary correct and incorrect decisions the players made during the level. For each incorrect decision, feedback regarding the correct URL category is given. The levels and tutorials of the personalized game are identical to the analysis game, and the decision game is structured equivalently, but instead of providing multiple buckets for different phishing URL categories, only one bucket labeled *Phishing* is available.

In contrast, the objective of the creation game is for players to apply manipulation techniques to a given benign URL to create their own malicious URLs. A level in the creation game consists of two to three different tasks called *presets*. Each preset poses a challenge for players to create a malicious but syntactically valid URL (e.g., '`https://ebay.com-signin.ml`' for a subdomain posing URL) based on the full URL, RD, or name of a given target (e.g., the RD '`ebay.com`'). Players are given a set of URL parts, e.g., single characters like "`.`" or "`/`" but also strings like "`.com`" or "`signin`". To complete a given task, players have to drag different URL parts into an initially empty URL bar while making sure, that the created URL follows a valid structure. In addition to the pre-defined URL parts, players can create custom URL parts or even complete URLs using a text input field and the *Generate* button. When the created URL is ready for submission, players have to click on the button labeled *Verify*. Then, a set of automated checks is performed, that test whether the URL is syntactically correct and fits the requirements of the task, and players receive feedback on the successful and failed checks in a pop-up window. If any check fails, players have to revise their created URL and resubmit it. Compared to the analysis game, levels are not time-bound and players have unlimited attempts. For further details regarding the game design of the two games, we refer to [6].

The target audience for all games are general users of the Internet. We did not add any requirements to the games that restrict the target audience, except for the ability to read and understand the explanations and tasks. This makes the games less suitable for young players or players with disabilities, which might be improved in the future by considering accessibility options for the games. While the games are currently only available in English and German, additional languages can easily be added if required.

The new game prototypes were created to test, whether the more complex game mechanics lead to better performance when classifying URLs compared to existing games using a binary decision scheme. This binary decision scheme does not allow for fine-grained assessment and feedback, and has a higher probability of guessing correctly. The aim of the more fine-grained assessment is to reduce the probability of guessing, as the number of possible solutions is higher, while also making the analysis of players' in-game data more powerful, as it is possible to better interpret choices regarding the categories of URLs. Furthermore, the URL parsing that is expected in the analysis game reflects a structured approach, which has been shown to be beneficial in identifying where a URL leads [Rey+20] and might facilitate the detection of phishing URLs. The design of the creation game is based on the

Table 6.2.: Explanation of URL categories and coverage in Analysis, decision and personalized (A), All (All) or None (None) of the Games

Category*	Explanation	Games
Benign	URLs with unaltered registrable domains	All
IP address	Host is IP address, target in path	A
Path	Random domain, target appears in path	All
Random	Domain and path are random, no target appears	A
RD	Misleading part included in registrable domain	All
Subdomain	Target appears as a subdomain	All
URL encoding	Parts of domain are URL encoded	None

*URLs of all categories appear in pre- and post-tests.

observation, that none of the existing games allowed for the creation of phishing URLs by manipulating benign URLs (see Section 6.1). Although users are not supposed to construct malicious URLs in a real-world scenario, the knowledge on manipulation techniques and the URL structure could be useful in recognizing malicious URLs. Furthermore, the user's active role in the learning process may lead to a deeper understanding compared to the other propose approaches.

To reduce the cognitive load per level and avoid fatigue by overly lengthy explanations, the explanation of URL structure was split into smaller parts, which can be introduced independently in different levels of the games. In the following, we present the URL creation and selection process which we used to create URLs from different categories that are presented in the game and user study.

6.3. URL Selection

Similar to the study presented in Chapter 5, we make use of a URL categorization to analyze the study results presented in the following. As noted in Section 6.2.2 above, we differentiate a total of seven categories of URLs, consisting of one benign and six malicious categories (see Table 6.2). This categorization enables a more detailed analysis of the results of the URL classification test compared to a simple comparison of benign and phishing URLs, as differences between the categories might indicate classes of URLs that are inherently more complicated to detect for users in our study, even after playing one of the games. We can also use the categories in Table 6.2 to enable a fairer comparison of the games, as not all categories appear in all games. Compared to the detailed categorization from Chapter 5, using simplified categories simplifies the analysis, and shifts the focus of the evaluation to reflect the structure of URLs, which was also used as a guide in the tutorials of the games.

Note, that not all sub-categories of the URL categories in Table 6.2 are presented and taught in the games, to avoid confusion and reduce the amount of time that is required (see Section 6.2.2). For example, while the creation game includes a detailed explanation of how to determine the RD in general, it only includes examples of combosquatting and URLs where the TLD was replaced. The pre- and post-tests, on the other hand, require users to classify URLs from all categories (see Section 6.4).

The games, as well as the URL classification tests used in our study, require example

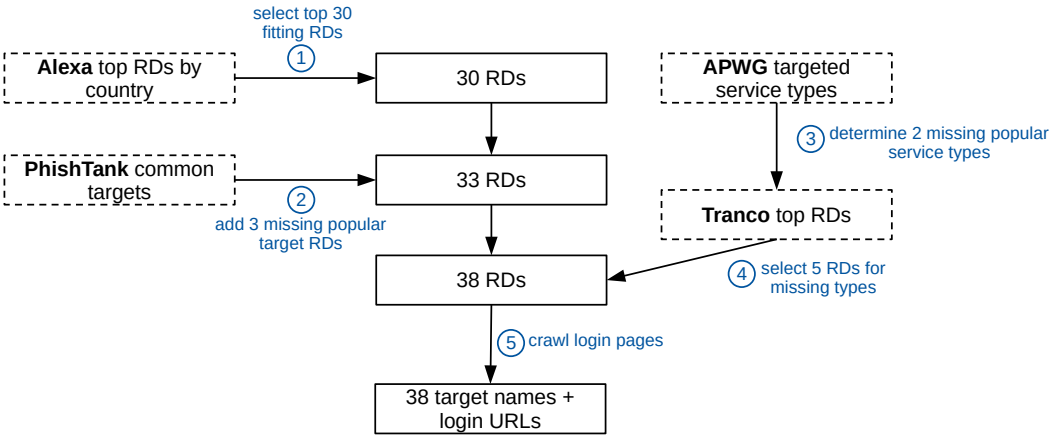


Figure 6.3.: Selection process for benign target names and login URLs.

URLs, which were selected from a pool of benign and phishing URLs that was created as follows. The pool was constructed with the goal to create a representative set of phishing and benign URLs. To this end, websites that were popular in Germany were collected, classified by the type of service they offer to ensure that commonly phished industry sectors are present, and then used to generate the pool of benign and phishing URLs (see Figure 6.3 for an overview of the process for benign URLs).

In detail, we started by constructing a set of relevant domain names by selecting services of various **types** (e.g., shopping). To this end, the 50 most popular websites in our country of origin were selected (according to Alexa[7], as this distinction was not available in the Tranco[8] list). Through manual review, we removed 20 websites as they were either adult websites or websites whose landing page was not displayed in German or English. The commonly used **names** of the remaining websites were extracted (e.g., name `google` for RD `google.com`) and the targets categorized by the type of service they offer. These types of service were then compared to the 5 most commonly targeted industries according to the APWG [APW21] and the 10 most commonly phished targets in PhishTank (as determined from more than 250,000 entries). Service types that were included among common phishing targets but not the Alexa list were added by choosing the highest-ranking websites from the Tranco list which fit the type of service and our country of origin. In all, this resulted in 38 target names and their corresponding RDs, which we expect to have been relatively well known in our country of origin at the time. The target information was further extended by a **login URL** that points to a login form, as determined by manually visiting the website of each service.

The targets were then used to generate the URLs that appear in the four games as well as the tests in the user study. We automatically generated a pool of benign URLs and one pool each for the categories of malicious URLs (e.g., `ebay-service.com` - *RD*, `pvyq5h4bmj.com/qgxfcvpacj` - *Random*) from the set of target names and RDs. This automatic generation was based on simple rule-based modifications of the input URL, and also resulted in *Benign* URLs that are recognizable by their benign RD but might not actually exist in the real world. We then selected a set of

[7] `https://www.alexa.com/topsites/countries` online, accessed 2021-02-16
[8] `https://tranco-list.eu/` online, accessed 2021-02-16

URLs for the pre- and post-tests (see Section 6.4.3) and removed these URLs from the pools. Note, that we only selected actually existing benign login URLs for the classification tests, so as to avoid confusing participants who know the exact login URLs of websites they use. The analysis, decision, and personalized games randomly select examples from the remaining URLs and present them to players of the games for classification. Since the creation game only requires benign reference domain names, we use the RDs of the targets there.

To assess differences between URL categories, we created a URL classification test which is used in the pre-, post-, and retention-tests. The test consists of a binary classification task, with URLs selected based on the categories described above. One additional constraint is added, as only URLs of actually existing login pages were selected from the benign URLs. The URLs for the pre-, post-, and retention-tests were selected uniformly at random from the pool of available URLs for each URL category (see Tables A.7 and A.8 in Appendix A.3). The pre-test consists of 13 malicious URLs, which were selected by choosing example URLs from all categories, while also ensuring that broader categories also include URLs from a representative range. For example, URLs from the *RD* category included combosquatting, changed-TLD and typosquatting, with different possible manipulations of characters for typosquatting (e.g., replacing or adding a character). The 7 benign URLs were selected by first choosing URLs of differing complexities (i.e., existence of subdomain or path) and then randomly selecting URLs to obtain 20 pre-test URLs in total. Ten additional URLs (3 benign and 7 malicious) are included in the post-test to test for learning bias and were chosen at random to get to a total of 30 URLs (20 malicious and 10 benign). This procedure was repeated for the retention test, which also extends the 20 URLs of the pre-test by 3 randomly chosen benign and 7 randomly chosen malicious URLs, resulting 20 malicious and 10 benign URLs in total. While the content of the URL classification test was equal for all participants, the order of items in the questionnaire was randomized between participants to reduce the influence of potential learning bias of the test items. We decided to only include URLs in the test, not complete website screenshots, as the games focus on URLs, and previous studies have shown, that users sometimes completely ignore this information when classifying websites (see, e.g., [AAC15]). We discuss potential problems of this approach in Section 6.6.

6.4. User Study Setup

In order to gain insights into the effectiveness of the games and the different game mechanics, a user study was conducted which is presented in the following. The study uses a four-group (four games), three-phase (pre-test, post-test, retention-test) design with A/B testing, a type of between-group design with four experimental groups and no control group. The four games (see Section 6.2) serve as independent variables, with the test performances in pre-, post-, and retention-test as dependent variables.

6.4.1. Research Questions

Based on the identified research gap, we developed the new game prototypes and designed a user study to evaluate them using the following research questions:

- **RQ-1**: Do the games have a positive influence on the participants' performance in classifying URLs?
- **RQ-2**: Do participants perform better in classifying URLs of services they know or use?
- **RQ-3**: Are there differences in the participants' performances between the four games? Are there advantages to using the newly proposed game mechanics?
- **RQ-4**: Are there performance differences in classifying different URL categories?
- **RQ-5**: Is knowledge retained three months after playing the games?

For the evaluation of our four games and the particular aspects regarding familiarity of services and possible differences between selected URL categories, we designed a user study with four groups of participants, where each group played one game. The objectives of our research are to evaluate the games in a pre-/post/retention-test study setup focused on URL classification knowledge. We aim to compare the learning outcomes of all games by comparing the players' performances before and after playing either game. We further analyze the effect of familiarity of services on classification performance. Finally, we compare the initial state, improvements and in-game behavior for classifying different URL categories.

6.4.2. Participants

The study was conducted in two parts, with 88 participants in November 2020 who played the analysis ($N_A = 40$) and creation ($N_C = 48$) games, and 94 participants in May 2021 who played the decision ($N_D = 45$) and personalized ($N_P = 49$) games. Recruiting was done online by posting information about the study in different social network groups of universities as well as distributing it via university mailing lists. Recruitment was focused on people with a general interest in playfully learning about IT security, regular online activities and little to no prior knowledge in IT security and Computer Science. Since the study required active participation for 60-70 minutes, a financial incentive of 15€ (approximately 18 USD) was offered to each participant. Among the participants, 111 identified as female (60.99%) and 71 as male (39.01%). The majority of participants was between 20-29 years old (76.37%). Due to the methods of recruiting, the majority of participants were students, with a high number of participants reporting their highest degree to be either a Bachelor's degree or high school diploma (82.42%). The remaining participants had mainly either completed their studies with a Master's degree (10.99%) or completed vocational training (2.74%). Besides the 182 participants, an additional eight participants were excluded for various reasons: one participant was excluded due to an unrealistic completion time, and seven participants had to be excluded due to technical problems during the online survey.

The retention tests were conducted over a period of two weeks three months after the original session. As the participants were already payed for the first part, we offered an additional incentive in the form of a lottery, offering a chance at winning four times 10€ for each of the two parts. We still experienced a dropout of 54.95% percent, leading to a response rate of only 82 participants ($N_A = 17, N_C = 25, N_D = 21, N_P = 19$). The process used to retain participants for the retention study is likely to have led to

a selection bias, where only participants that were originally interested in the games (and thus more likely to remember the topics) took part. We discuss this problem in more detail in Section 6.6.

6.4.3. Apparatus and Materials

The first session of the study was conducted as a remote, online lab study using a video conferencing software and a web browser, with the retention test only consisting of an online survey. For all test phases, an online survey containing the following questionnaires and tests was used:

- **URL classification test**: This test measures the performance in classifying a set of URLs (see Section 6.3). For each URL, participants had to decide whether it was benign or phishing. It was utilized in the pre- and post-tests, as well as the retention test, with the aim of answering **RQ-1** to **RQ-5**. A list of the used URLs can be found in Tables A.7 and A.8 in Appendix A.3.
- **Recognition of Services**: This questionnaire contains a list of services that were targets in the URLs of the URL classification test. Participants were asked whether they use the service, do not use but know the service or do not know the service (in response to **RQ-3**). It was used in the post-test to cover services of pre- and post-test URLs, as well as the retention test to update the list with services that only appeared there (see Table A.9 in Appendix A.3 for the complete list of services).
- **Demographics**: This questionnaire contains questions regarding gender, age and educational background. It was included in the post-test (see Table A.10 in Appendix A.3) and was used to report on potential biases among the participants.

6.4.4. Procedure

The first phase of the study began with a briefing phase, where the procedure of the study as well as the requirements for participation were explained. To give more contextual information and establish a shared understanding, a definition of phishing including an example was presented. The decision of which game was to be played by which participant was done uniformly at random by the survey system when each participant started the pre-test phase. Participants were redirected to the game from the survey platform, and did not know that different games were tested in the survey, nor to which group they were randomly assigned. After all participants finished the post-test, the instructors explained the purpose of the study and answered questions of participants in a debriefing before closing the session.

As no supervision was necessary for the retention test, the participants were sent the URL to the retention survey three months after the post-test and were asked to complete it in their own time.

6.5. User Study Results

Based on our research questions described in Section 6.4.1, we conducted a series of analyses and tests. For each test, we consider four groups depending on which

game the participants played: the *creation game group*, the *analysis game group*, the *decision game group*, and the *personalized game group*. As in the previous study (see Chapter 5), the performance scores are computed as relative scores and measured using an interval scale. We use a significance level $\alpha = .05$. The indices *pre*, *post*, and *ret* are used to distinguish pre-, post-, and retention-tests respectively and hyphenated suffixes in the indices are used to distinguish post and retention-test scores on the URLs also used in the pre-test (e.g., *post-pre*) or newly added URLs (e.g., *post-new*). Furthermore, indices **A** (for analysis), **C** (for creation), **D** (for decision), and **P** (for personalized) are used to indicate the respective game group. Due to the large dropout for the retention test, we split the analysis into two parts, beginning with the comparison of pre- and post-test results, with a higher number of participants, followed by the retention test, with only those participants that remained for all three tests.

We first check for a potential learning bias by comparing the performance means in the post-test (see Table 6.4). Instead of a higher mean performance for URLs also used in the pre-test ($M_{\text{post-pre}}$), the mean performance for new URLs in the post-test ($M_{\text{post-new}}$) is higher. As such, we argue that the effect of learning bias is negligible. We therefore compare only those URLs that were part of both pre- and post-test for **RQ-1**, where we look at general improvements, while including all post-test URLs in subsequent sections.

We answer the first four research questions using the pre- and post-test results, as there is more data available due to the higher number of participants, and only include the results of the retention test in the last research question.

6.5.1. Immediate Effectiveness of Games

The first RQ described in Section 6.4.1 focuses only on the general effectiveness of the games. For **RQ-1** we derive the following hypothesis: The participants' performance in classifying URLs increased after playing either one of the games. As shown in Table 6.4, the mean performance score did indeed improve in the post-test for all four games. To test for significance of these improvements, a one-tailed Student's t-test was performed for each game, comparing the results of the classification task on pre-test URLs between pre- and post-test. A non-parametric Wilcoxon signed-ranked test was performed if a deviation from normality was detected (Shapiro-Wilk test, cut-off value $\alpha < 0.05$, marked with an asterisk). The results (see Table 6.3) indicate, that the participants' performances increased significantly for all four games. There are, however, differences in effect sizes and mean differences, which are larger for the

Table 6.3.: Results of t-tests comparing relative scores in pre- and post-test for all three games

Game	Test statistic	p-value	effect size
Creation	$t(47) = -3.459$	$p < .001$	$d = -0.499$
Analysis	$t(39) = -6.404$	$p < .001$	$d = -1.013$
Decision*	$W = 24.500$	$p < .001$	$r = -0.946$
Personalized*	$W = 128.000$	$p < .001$	$r = -0.717$

*Deviation from normality detected.

Table 6.4.: Means (and standard deviations) for performance in pre- and post-test including means on partial URL sets

Game	N	M_{pre} (SD)	$M_{\text{post-pre}}$ (SD)	M_{post} (SD)	$M_{\text{post-new}}$ (SD)
Creation	48	0.702 (0.122)	0.755 (0.122)	0.782 (0.129)	0.835 (0.173)
Analysis	40	0.695 (0.098)	0.828 (0.115)	0.840 (0.095)	0.865 (0.121)
Decision	45	0.701 (0.097)	0.818 (0.091)	0.831 (0.097)	0.858 (0.137)
Personalized	49	0.726 (0.114)	0.811 (0.110)	0.823 (0.104)	0.847 (0.128)

Table 6.5.: Performance scores (and standard deviations) per service familiarity

Game	Used	Known	Unknown
Pre-test	0.694 (0.186)	0.721 (0.177)	0.576 (0.252)
Creation	0.809 (0.167)	0.790 (0.140)	0.702 (0.239)
Analysis	0.835 (0.148)	0.831 (0.133)	0.720 (0.277)
Decision	0.858 (0.135)	0.807 (0.140)	0.702 (0.235)
Personalized	0.841 (0.118)	0.817 (0.144)	0.686 (0.247)

analysis, decision, and personalized games than for the creation game.

6.5.2. Differences Between Used, Known and Unknown Services

For the difference between used, known and unknown services (**RQ-2**), we test the following hypothesis: The participants' performance in classifying URLs of services they use or know is better than for services they do not know.

Differences During Testing

Descriptive results seem to indicate significantly higher performance scores for used and known services (see Table 6.5). To test for significance in performance scores, we performed a factorial repeated-measure ANOVA, with the tests (pre, post) and service familiarity (unknown, known, used) as factors and the games as between-subject factor. Note, that some participants did not select any services as unknown or known, thus reducing the number of valid entries to $N = 153$. As Mauchly's test for sphericity was significant for levels of familiarity ($p < .001$), degrees of freedom were corrected using Greenhouse-Geisser estimates ($\epsilon = .850$). The results of the ANOVA confirm, that familiarity had a significant effect on the performance ($F(1.701, 253.386) = 38.677, p < .001, \eta_p^2 = .206$). Post-hoc tests using Holm's correction confirm significant differences between unknown and known or used services: unknown and known ($p < .001, d = -.648$), unknown and used ($p < .001, d = -.645$), but not for known and used ($p = .970, d = .003$). None of the interactions (familiarity × game, familiarity × test, familiarity × test × game) were significant ($p >= .052$). Overall, familiarity with a service had a significant effect on the participants' performance in pre- and post-tests, with significant differences for unknown URLs compared to known and used URLs. As such, we remove unknown URLs from the following analyses where possible to reduce the potential effect of this bias.

Table 6.6.: In-game mean (SD).

Familiarity	N	Correct	Incorrect	Not classified	Time (sec)
Used	49	0.680 (0.170)	0.186 (0.108)	0.133 (0.117)	4.13 (1.39)
Known	49	0.665 (0.180)	0.192 (0.142)	0.143 (0.115)	4.11 (1.69)
Unknown	49	0.597 (0.221)	0.250 (0.187)	0.154 (0.187)	4.27 (1.62)

Differences During Gameplay

We also present an overview of differences between the levels of familiarity during gameplay, as this information can be extracted from the log data of the personalized game. Python scripts were used to parse the in-game log data and extract different event sequences (e.g., opening a coin, then dragging the URL into a bucket), including timing information as well as the outcomes of classification events. For each aspect regarded in the following, mean values are first computed per player and then analyzed, e.g., as the average of all players.

We start by taking a look at the sorting outcomes for URLs of different levels of familiarity (see Table 6.6). We can observe clear differences in the relative classification outcomes of players, with URLs of unknown services being classified with the least accuracy with a mean difference of .068 to known and .083 to used services.

Next, we determine whether URLs of unknown services are discarded or ignored more often and whether there are time differences for the levels of familiarity. Indeed, it seems that even though the differences are small (mean difference $> .012$), URLs of unknown services were either actively skipped or opened but not classified more often than those of known or used services. Similar results can be observed for the timing data, where players seem to take longer for URLs of unknown services (mean difference $\geq .14$ seconds).

In all, the detailed analysis that is available in the personalized game seems to indicate similar results to the tests, in that URLs of unknown services are classified with less accuracy and discarded more often within the game than URLs with services of the other familiarity levels.

6.5.3. Differences Between the Four Games

To address **RQ-3**, we test the following hypothesis: The participants' performance in classifying URLs in the post-test differs between the four games. Mean values in Table 6.4 seem to suggest, that players of the creation game performed worse in the post-test than players of the analysis game and its derivatives, who performed similarly well. To test the hypothesis, we compared the performance scores in the post-test of URLs in URL categories that were part of all four games (i.e., excluding IP addresses, URL encoding and random URLs, see Table 6.2). An ANCOVA was performed, with the games as between group factor, performance in the post-test as dependent variable, and performance in the pre-test as covariate. Levene's test for equality of variances is not significant ($F(3, 178) = 0.917, p = 0.434$). The ANCOVA does not return significant results for the four games as between-subject factor ($F(3, 177) = 1.278, p = 0.284, \eta_p^2 = .021$), only for the pre-test score as covariate

Table 6.7.: Mean pre- and post-test relative scores for all URL categories differentiated in the tests

Category	Pre	Post_C	Post_A	Post_D	Post_P
Benign	0.728	0.779	0.853	0.882	0.798
IP	0.830	0.875	0.950	1.000	0.980
Path	0.948	0.938	1.000	1.000	0.978
Random	0.874	0.667	1.000	0.978	1.000
RD	0.602	0.738	0.770	0.723	0.765
Subdomain	0.712	0.880	0.871	0.870	0.893
URL encoding	0.909	0.915	0.912	0.898	0.948

($F(1, 177) = 50.397, p < 0.001, \eta_p^2 = .222$). We therefore retain the null hypothesis, that the differences in post-test performances between the games are not significant. Of particular note is the fact, that the more complex sorting mechanism included in the analysis game did not result in significant differences to the decision game in our study.

6.5.4. Differences Between URL Categories

For **RQ-4**, we take a look at URL categories, guided by the hypothesis: There are differences in the participants' performance in classifying different URL categories.

Simplified Categories

Table 6.7 shows the average scores for the URL categories in the pre-test, as well as the post-test (including post-only URLs) by game. These statistics seem to suggest, that similar to the study in Chapter 5, although there are general improvements after playing the games, some categories are less well classified (e.g., RD, Subdomain) while others are generally recognized well (e.g., Path, IP). To test for significant differences in performance scores, two repeated-measure ANOVA using the URL categories as repeated-measures factor were performed: the first for URL categories in the pre-test (see Figure 6.4), and the second for URL categories in the post-test with the game as between-group factor (see Figure 6.5). For both tests, Mauchly's test for sphericity was significant ($p < 0.001$), and degrees of freedom were corrected using Greenhouse-Geisser estimates ($\epsilon_{pre} = 0.804, \epsilon_{post} = 0.881$). The first ANOVA ($N = 160$) returns significant results, indicating that there are already differences in participant's performance for different URL categories in the pre-test ($F(4.827, 767.465) = 29.337, p < .001, \eta_p^2 = .156$). Post-hoc tests using Holm's correction confirm several significant differences, mainly including the generally well-detected Path and URL encoding URLs, as well as the RD category, which had very low average detection rates.

Similarly, the second ANOVA ($N = 172$) confirms that differences are still present in the post-test ($F(5.283, 887.616) = 28.762, p < .001, \eta_p^2 = .146$). Post-hoc tests (Holm, averaged over four games) again include significant differences for Path, IP, Random, and URL encoding (high detection rates), as well as RD (low detection rates) URLs. Random URLs were well detected only in the games they were presented

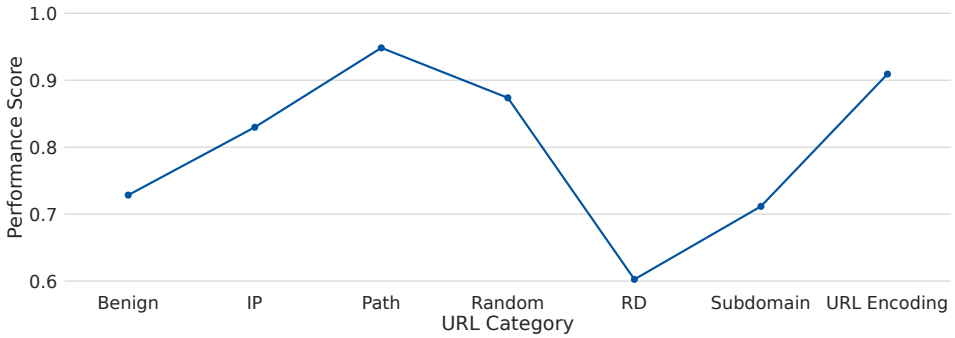


Figure 6.4.: Performance scores in the pre-test over different URL categories.

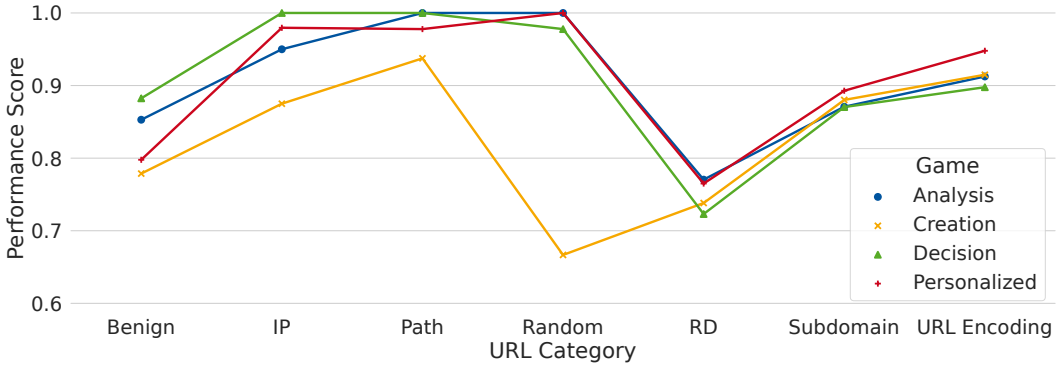


Figure 6.5.: Performance scores in the post-test over different URL categories.

in, but were still detected significantly better than most categories when averaged over the four games, while the remaining Benign and Subdomain categories were detected with neither particularly high nor low performances. In conclusion, results indicate that some URL categories (e.g., RD) were significantly more complicated to detect in our tests than others (e.g., Path), in both pre- and post-test.

Subdomains of Benign URLs

Since the URL test for the games includes benign URLs with different subdomains, which might have an effect on their perceived complexity and thus performance scores, this subsection present a short analysis of possible subcategories for these benign URLs. The tests include three different subdomain constructions: (1) plain URLs without subdomains (2) URLs with only the common `www` as subdomain and (3) URLs with a different subdomain (e.g., `login`). Here, descriptive statistics (see Table 6.8) seem to imply differences for URLs that use uncommon subdomains, which we again confirm in two repeated-measures ANOVA, one for the pre-test and one for the post-test with the games as between-subject factor. The test for sphericity is significant for the first ANOVA ($N = 42$), so we apply Greenhouse-Geisser corrections with $\epsilon = .807$. The results indicate significant differences between the sub-categories of benign subdomains: $F(1.613, 66.148) = 14.770, p < .001, \eta_p^2 = .265$). Post-hoc tests using Holm's correction confirm the expected differences between *plain* and *other* ($p < .001, d = 1.003$), as well as *www* and *other* ($p < .001, d = .766$), but not *plain* and *www* ($p = .223, d = .237$). Note, that the number of participants is lower

Table 6.8.: Differences between benign subdomain categories

		N	Mean	SD
Pre	Plain	42	0.905	0.297
	www	182	0.811	0.203
	Other	182	0.558	0.361
Post	Plain	160	0.938	0.223
	www	182	0.866	0.203
	Other	182	0.670	0.381

in this test, as it includes a subcategory with only one URL in the pre-test, which was not well known.

While all three categories see mean improvements in the post-test, the *other* category still stands out as it is detected less well than the other two. For the post-test ANOVA ($N = 160$), degrees of freedom were corrected with $\epsilon = .796$, resulting in $F(1.592, 248.390) = 45.748, p < .001, \eta_p^2 = .227$. Here, post-hoc tests again confirm significant differences between *plain* and *other* URLs ($p < .001, d = .903$), as well as *www* and *other* ($p < .001, d = .714$), but not the remaining comparison ($p = .059, d = .189$).

As such, we confirm that there are significant differences between the different usages of subdomains in benign URLs in our classification tests.

6.5.5. Analysis of Knowledge Retention

In response to **RQ-5**, we assessed the participant's retention three months after playing the games.

Differences between the three Tests

As can be seen in Table 6.9, the mean differences are higher in post- and retention-tests compared to the pre-test, with the post-test score being the highest. To test for significance of these differences, we performed a repeated-measures ANOVA using the relative scores of the three tests as repeated-measures factor and the games as between-group factor. The test was performed on the set of URLs that were originally present in the pre-test, to avoid biases that might arise due to different difficulties of the additionally added URLs. The ANOVA confirms that there are differences between the three tests: $F(2, 156) = 40.737, p < .001, \eta_p^2 = .343$. Post-hoc tests using Holm's correction reveal differences between all three tests: Both post-test ($p < .001, d = 1.106$) and retention-test ($p < .001, d = .812$) performances were significantly higher than pre-test scores, and post-test performance was still significantly higher than in the retention-test ($p = .022, d = .294$). The interaction test $\times$ game is not significant ($p = .292, \eta_p^2 = .045$).

Differences between URL categories

Taking a closer look at the different URL categories in the retention-test (see Table 6.10, Figure 6.6), we again observe the pattern found previously, where the categories with the target in the path (i.e., Path, IP) were significantly easier to

Table 6.9.: Comparison of Test Outcomes for Participants in Retention Test

	Pre	Post_C	Post_D	Post_A	Post_P	Retn_C	Retn_D	Retn_A	Retn_P
Valid	82	25	21	17	19	25	21	17	19
Mean	0.696	0.765	0.844	0.864	0.830	0.760	0.794	0.820	0.793
SD	0.126	0.116	0.095	0.095	0.125	0.125	0.124	0.079	0.125

Table 6.10.: Mean relative scores for all URL categories in the retention test for the four different games

Category	Retn_C	Retn_A	Retn_D	Retn_P
Benign	0.747	0.847	0.846	0.770
IP	0.920	0.882	0.905	1.000
Path	0.960	1.000	0.952	0.941
Random	0.820	0.941	0.976	0.921
RD	0.651	0.694	0.679	0.709
Subdomain	0.813	0.848	0.683	0.855
URL encoding	0.880	0.853	0.905	0.972

detect than those with information in the subdomain or RD. This difference remains significant, as confirmed by a repeated-measures ANOVA for the URL categories with the games as between-subject factor: $\epsilon_G = .837, F(5.021, 381.607) = 15.433, p < .001, \eta_p^2 = .169$). It is noteworthy, that the performances for the subdomain category in the decision game were noticeably lower than for the other games, even though there were few differences between the games otherwise. These differences are not significant, however, according to post-hoc tests using Holm's corrections.

6.6. Discussion

In the previous section, we presented the results of our user study to answer the research questions described in Section 6.4.1. We found, that (**RQ-1**) performances improved significantly after playing either one of the games, (**RQ-2**) players were significantly better in classifying URLs of services they are familiar with, (**RQ-3**) there are no differences in performance between the four games, (**RQ-4**) there are significant differences in classification performances for different URL categories, (**RQ-5**) knowledge was retained for a duration of three months. In the following, we discuss the setup and results of our study.

6.6.1. Limitations

For our setup, a general look at the participants of our study reveals a deviation from the general population. Even though we did not recruit participants of a specific age group or occupation, the advertisements for the study were mainly distributed in online social groups for students. As a result, our test population consists mainly of students and does not represent the general population, which might lead to problems in generalizing our findings. In particular, it is possible that these younger

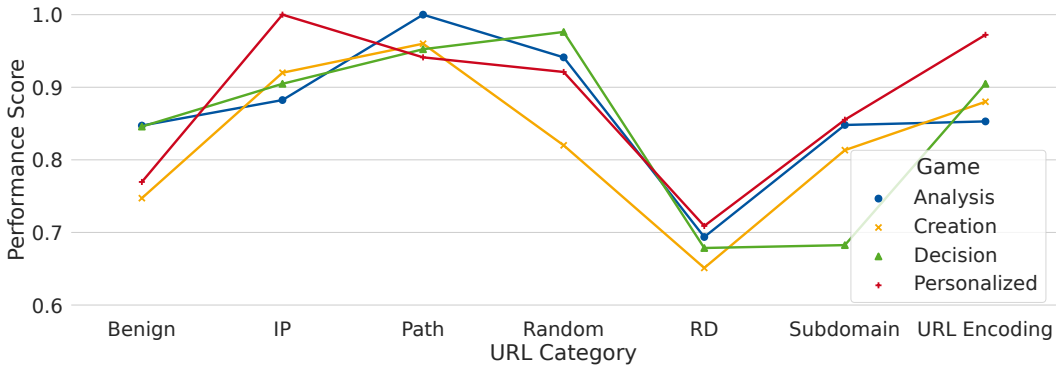


Figure 6.6.: Performance scores in the retention-test over different URL categories.

people have different states of minds concerning online risks (see, e.g., [Oli+17]), or have more experience in reading URLs than the general population. However, we argue that the results might be generalized to the population of students between 20-30 years old, which could be substantiated by additional user studies providing supporting evidence. Note, that we did not study the effect of gender and other demographics on the participants' classification performance in detail, as this was not the goal of this study. For gender in particular, we tested its effect on the ANCOVA in the comparison of the four games (**RQ-3**) to eliminate potential biases, and did not find a significant difference between female and male participants. All other statistical tests performed in this study are based on repeated measures. To support further generalization, we suggest replicating the study on a more representative group of participants.

Even though we performed a remote online study, where participants utilized their own, familiar devices, our study design was a lab study, and did not test, how participants would respond to actual phishing attacks in a more realistic setting. The focus of the games is to impart the knowledge required to detect phishing URLs, and the study shows how well this knowledge can be applied in an optimal setting where participants were fully aware of the task. In particular, we do not claim that the games raise situational awareness and help avoiding phishing attacks in a more realistic setting, since they do not convey the knowledge and awareness of how and when phishers lure potential victims into disclosing personal information. While we did find, that performance scores were retained over a period of three months, we did not test how well the knowledge translates to awareness against actual attacks in the real world. Here, a simulated phishing attack could provide further insights. We also note, that our study was performed during the time of the COVID-19 pandemic, which might have had an impact on participants' state of mind. We did not test for the effect of the pandemic on the participants, and assume that it did not have a more significant effect on our results than other possible limiting but unknown factors.

In the URL tests of our user study, we asked participants to classify URLs, not screenshots of websites. This is due to the fact, that it has previously been shown that users do not usually look at the URL bar, even in phishing classification tasks (see, e.g., [AAC15]). As the focus of the games is knowledge about URLs in general and phishing URLs in particular, and how this knowledge can be utilized to detect

phishing websites, we decided to only include URLs in the tests. Furthermore, knowledge about URLs can be applied in several different contexts, as users can analyze URLs before clicking on them, or use the knowledge about domain names to better understand browser URL highlighting, the sender domain in emails, or TLS certificates [3]. As such, we argue that the chosen URL classification task maps the requirements of our study more precisely than a website classification task. It might, however also possibly amplify the effect of unknown services, as screenshots would offer more context information. However, we argue that the crafted URLs always include a reference to the original target name, and additional information in the website would therefore not have made a significant difference. The only exception are random URLs, which do not include recognizable target names, but were also not included in the evaluation of **RQ-2** and **RQ-3**. We furthermore found the same effect regarding service familiarity in the previous study presented in Chapter 5, which showed screenshots along with URLs, thus demonstrating that the results seem to generalize to different study setups.

Next, we note that the tests also include more malicious URLs than benign URLs, which does not realistically reflect the real world situation and might have led to bias in our results. Still, we argue that the improved results for benign URLs (see Table 6.7) in all four games indicate, that players were not choosing *Phishing* more often, and did instead utilize their understanding of URLs to classify URLs more effectively. In addition, we also observed higher confidences in the post-test (see [5] for details), which might indicate that participants also felt like they were now able to apply their knowledge more effectively, thus deciding more confidently.

Finally, the URLs that were presented in the tests were constructed based on popular websites in Germany, either using the login URLs directly or by applying rule-based modification for phishing URLs, which might not have resulted in a representative set of URLs. For phishing URLs, the selected URLs do not include all sub-categories defined in Chapter 5. Even when we did not find significant differences between sub-categories in the previous study, this might reduce the external validity of the results. Similarly, the benign URLs represent actual login URLs, but this is not the only type of URL that users are likely to encounter in their daily browsing. As such, it is possible that the improvements seen in the post- and retention-tests do not translate directly into classification capabilities in the real world when users mainly encounter different types of URLs.

6.6.2. Implications

As described in Section 6.5, we found significant increases in performance scores from pre- to post-test in all four games (**RQ-1**). Taking a closer look at the differences between the four games (**RQ-3**), we found that none of the performances for either game significantly deviated from the others, even though the post-test scores for players of the creation game were overall lower than those of other games. It is noteworthy, that participants who played the creation game usually took more time and asked more questions during the study, as some of them had problems advancing through the game. One possible explanation for the differences is that the requirement to create URLs by themselves posed a higher difficulty and complexity, which resulted in confusion for players who did not really understand the learning content. This might indicate, that the creation mechanic is less suited for self-learning and should

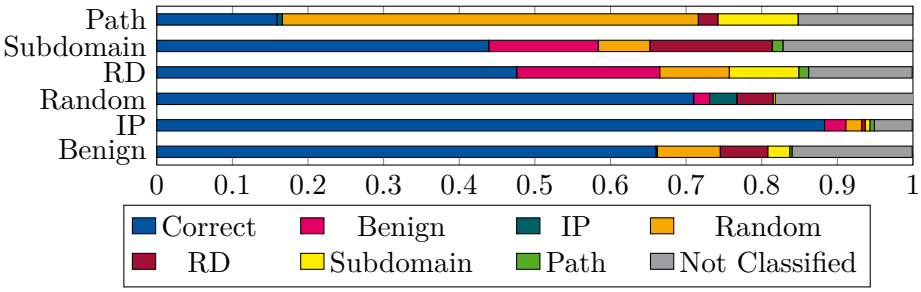


Figure 6.7.: Relative sorting outcomes for URL categories in the analysis game.

be avoided in such contexts. We note, however, that the creation game differs from the analysis (and therefore decision and personalized) game in the included URL categories, the tutorials and the levels. As such, the results of this comparison are less significant than the direct comparison between the analysis, decision, and personalized games.

While we did not find significant differences between the analysis game and the decision game, we still argue that the analysis game offers several advantages. In particular, when performing an analysis of in-game data for both games, the analysis game is able to offer more insights into players decision processes. This is due to the fact, that players' mistakes when URLs are sorted into buckets of different URL categories, instead of making a binary decision, offer a better understanding of players' misconceptions. Figure 6.7 shows the sorting outcomes (i.e. the percentage of URLs sorted into different buckets per URL category, where the outcome *not classified* includes all discarded or opened but not classified URLs) of the analysis game. Even though general trends are visible in the decision game as well, the choices and confusions of players are more evident in the analysis game. In particular, we note that many players had difficulties with path URLs, often classifying them as random URLs. This trend is not visible in the decision game, as it does not make this distinction, and indicates that players focus mainly on the domain name for classification, while mostly ignoring path information. Furthermore, the analysis game is better suited to understand misconceptions in the basic parsing abilities of URLs. This can, for example, be seen in the case of URLs in the subdomain category, which were often confused with RD URLs. Since the more complex mechanic also did not lead to drops in performance, we recommend its use if more information on the decision process of players is required in games, as might for example be the case in adaptive games.

For **RQ-2**, we looked at differences between known and unknown services in the URL tests, and found significant performance differences in all games. We note, however, that there were generally less URLs with services that participants did not know than those they did know (3.26 unknown on average), with a few players even knowing all of the services, which might have introduced a bias in the comparison. Similarly to the previous study in Chapter 5, we therefore removed unknown URLs from the remaining analyses to avoid this bias. Using log-data analysis of the personalized game, we confirmed similar differences between levels of familiarity during gameplay as well. Together, these results might implicate, that educational material should aim to cover the context that users experience in their day-to-day

lives as closely as possible. We argue, that it is likely that information, examples, and recommendations that fit the context (e.g., the setup of a specific institution) can be better understood and incorporated by users than information that is more general or even for a completely different context. More thorough analyses of personalization in educational interventions, in particular for anti-phishing education, might therefore be a promising approach for future work.

When focusing on different categories of URLs, we found that there are significant differences for performance scores in both pre- and post-test (**RQ-4**), as well as the retention test (**RQ-5**). This is consistent with prior work, that showed significant differences in test scores in the context of anti-phishing learning games as well [Can+15; Rey+20]. Of particular note are the high scores in URLs that include malicious keywords in the path, especially after playing the analysis, decision, or personalized games. This seems to indicate, that the distinction between domain name and path is relatively easy to grasp for players of the games, and that phishers might have to create benign-looking domain names to lure educated users. As for subdomains, we saw a significant increase in performance, but participants still seemed to have more trouble recognizing them compared to the *Path* category. The most troubling results are for URLs that manipulate the RD, as these were often detected less accurately, even in the lab setting of our study. This raises the question, whether users can be relied upon to detect these categories of phishing URLs at all, as the URLs cannot be simplified by URL highlighting or looking at domain names in a certificate. We argue that this drawback might be generalized to other educational games and resources as well, since determining the exact RD was a substantial part of all four games, by including conceptual and procedural knowledge in addition to the potential to test this knowledge in the levels. Lastly, URL categories that were not part of the game did not improve to the same extend as included categories, which might imply that knowledge was not transferred and retraining might therefore be necessary for newly emerging categories of phishing URLs. Comparing our results with those in the study in Chapter 5, NoPhish [Kun+16] or the study by Reynolds et al. [Rey+20], pre-test results already differ for our URL categories. These differences might indicate that there is a different measure of difficulty of a phishing URL that is not necessarily connected to the categories used in this paper, or might be due to differences in the study setup. Note, that the number of URLs per category is low in our setup, which might amplify this kind of hidden bias. For more reliable results we would recommend to include more URLs per category in the tests, which might on the other hand increase the average study duration and hence the chance for fatigue among participants.

The closer look at categories of benign URLs, with a focus on different subdomains, revealed that users had more troubles classifying URLs with uncommon subdomains. Note, that the benign login URLs were collected by visiting real websites, so the differences might indicate, that users would benefit if benign services were to use plain subdomains for their login process.

For **RQ-5**, we evaluated differences between the three test phases with a focus on performance in the retention test. While the performances seemed to have declined overall, they remained above the baseline of the pre-test, thus demonstrating that knowledge was retained over the period of time. In particular, performance scores for phishing URLs with the target in the path remained high for all games, while URLs with subdomain posing or the target in the RD dropped further (.187 difference for

subdomain in decision game, .087 for RD in creation game). Of particular note is the lower score in the subdomain category for the decision game, which might indicate that the more complex game mechanics in the other games might have led to better retention for this category. This effect should, however, be confirmed with a larger sample size in the future.

In all, we found that the games presented in this chapter were effective, in that the classification performance of players improved for all four games. While some categories were detected particularly well, performances were varied across different URL categories, where manipulation techniques targeting the RD or subdomains led to lower accuracy than those targeting the path. Future work might test, whether different tutorial content or levels can improve the accuracy for these categories. Testing the participants again after three months demonstrated, that participants retained their knowledge at least partly, as they still performed significantly better than in the pre-test, even though their performances had already degraded compared to the post-test.

While we do not claim that the results, in particular regarding the shortcomings of the games, translate to educational interventions in general, we argue that some trends are likely to generalize. For example, it is unlikely that educational approaches will be successful in teaching knowledge about URLs to an extend that all URLs in our tests would achieve the high scores of path URLs. Even then, we did not test how the knowledge of URLs transfers to awareness, i.e. whether the players are able to apply this knowledge in the correct situations. Finally, it is not clear whether knowledge about URLs alone is able to prevent all possible attacks, for example in malware phishing attacks, or when content is hosted on benign infrastructure. As such, while education is currently an important building block in a holistic defense against phishing attacks, it is unlikely to prevent all attacks on its own. In the next chapters, we therefore aim to augment and supplement the benefits of education with other approaches, that can be used in tandem to improve detection rates overall.

Chapter 7

Reverse Domain Name Notation

The previous chapter presented an overview of the successes and limitations of anti-phishing education based on the example of four anti-phishing learning games. While the overall performances increased after playing the games, and knowledge was retained after three months, there were significant differences in classification performances for different URL categories. One of the categories of URLs that was not detected with high accuracy even after playing the games was the *subdomain* category. This finding is also supported by the study on URL categories in Chapter 5, where we found that phishing URLs were harder to detect when a reference to the target appears at the beginning of the URL.

In this chapter, we test whether this problem with subdomains can be alleviated by using a different notation for URLs: Reverse Domain Name (RDN) notation. To this end, we first present RDN notation, followed by a user study that was performed to compare the normal URL notation with RDN notation.

Contributions: The main contribution of this chapter is the design and evaluation of a user study to research the effect of using RDN notation on URL classification accuracy. RDN notation potentially reduces the risk of phishing URLs where the target appears in a subdomain, as the reference to the target can no longer be placed at the beginning of the URL, and has, to the best of our knowledge, not been evaluated as a potential design intervention for phishing prevention before. These results are new contributions of this thesis and currently submitted for publication.

7.1. URL Notations and Research Questions

We previously found, that users have trouble discerning phishing URLs that start with the target name or RD when read from the left, indicating that they mainly focus on the beginning of the URL. This is in contrast to the fact, that FQDNs should be read from right to left, as the most important parts of the hierarchy, particularly the combination of eTLD and e2LD, appear at the end. In the following, we present RDN notation as a potential solution to this problem, and define a number of research questions for the user study following in the next section.

7.1.1. RDN Notation

The hosts of URLs used to visit common websites are typically made up of FQDNs containing several domain labels (e.g., `www`, `example`, `com` in `https://www.example.com/`). In normal URL notation, the DNS hierarchy that these labels refer to becomes more specific the further left the label appears, in other words a resolver would start by processing the right-most labels (e.g., TLDs) and work towards the labels further to the left (e.g., subdomains) in order to resolve the FQDN. It is therefore possible for domain owners to add almost arbitrary labels to the left of the domain name they registered, i.e., they can add potentially misleading labels to the left of the RD. Consequently, the RD part of the FQDN is the part of the URL that typically most closely describes the legal entity or service behind the URL and is thus the most important part of the FQDN in the context of phishing detection. This poses a problem to users who read the URL from left to right, since the first labels they encounter are potentially crafted by an attacker.

Contrarily, in RDN notation, the order of labels in the FQDN is reversed, thus placing more general labels in the DNS hierarchy closer to the beginning of the URL. As an example, the FQDN

`www.example.co.uk`

would be rewritten as

`uk.co.example.www`

in RDN notation. This makes it possible to read the URL from left to right without encountering a completely attacker-controlled part of the FQDN first, thus potentially reducing the risk of subdomain embedding URLs. In the URL test of the following user study, we retain the scheme of the original URLs, and do not change the order of path components or http credentials. As such, the more complex URL

`https://user:pass@sub.example.tld/some/path?some=query#fragment`

would be rewritten as

`https://user:pass@tld.example.sub/some/path?some=query#fragment`

in RDN notation.

In the following, we refer to the original URL notation as the *normal* notation and to the reversed notation as *RDN* notation.

7.1.2. Research Questions

This chapter aims to answer the question, whether this difference in notation has an impact on classification performance of phishing URLs, and in particular if it is able to improve the accuracy for the *subdomain* category. We therefore formulate the following research questions:

- **RQ-1** Are there general differences in classification performance between the two URL notations?
- **RQ-1-a** Are there differences in the time it takes users to classify URLs between the URL notations?

- **RQ-2** Are there differences in classification performance for specific URL categories?
- **RQ-3** Are there differences in classification performance compared to the related study presented in Chapter 5?

We attempt to answer the questions in a user study, where participants classify URLs in the two different notations. Both notations include URLs from the same URL categories, thus enabling a comparison of the effect of the notation on the different categories. We include the third research question, as the URL classification test is based on the test included in Chapter 5, making it possible to check for biases in the study by comparing both notations to the classification performance of the corresponding URL in the previous test. Finally, we take a look at the differences in time taken to classify URLs between the two URL notations, as it is possible that changing the order of FQDNs leads, for example, to participants only focusing on the beginning of the URL, which would decrease the time needed to judge a given URL.

7.2. User Study Setup

The study follows a simple repeated-measures design, where participants first classified URLs in the normal notation, followed by URLs in RDN notation. It was conducted as an online study, consisting only of a LIME survey to be completed. The study design is based on the study presented in Chapter 5, using the same URLs and categories as basis, thus simplifying the setup process and enabling a comparison between the two studies.

7.2.1. Participants

In all, 50 participants were recruited for the study using Prolific[1], an online recruitment platform. We published the survey with a conservative estimation on completion time of 25 minutes (the actual median time taken was 18:03 minutes), and rewarded 3 GBP to all participants. The participants were requested as a balanced sample based on gender, i.e., the number of male and female participants was equal. Participants were required to be fluent in German, as the survey is in German, otherwise no screening requirements were added. After reviewing the results of the study, three participants were removed since they always answered with *phishing* for URLs in RDN notation. We discuss this decision in Section 7.4. As a result, the sample is no longer balanced, with 23 participants identifying as female, and 24 as male. While most participants were between 20 and 40 years of age, with a mean age of 33.255 years ($SD = 11.387$), the overall range included ages 19 to 70, with three participants over the age of 55. As for the employment status of the participants, 19 reported a student status, and 27 participants specified full-time employment.

7.2.2. Apparatus and Materials

The study setup is based on the study used to analyze URL categories in Chapter 5, and uses the same texts and URLs as basis. Consequently, the study was conducted

[1] `https://www.prolific.co/`, accessed 2022-10-25

as an online study consisting only of an online survey, with the recommendation to complete the survey on a PC or tablet (not on mobile devices).

Apart from a general introduction (similar to the pre-study, see Figure A.5 in Appendix A.4), the survey consists of three main parts:

- Familiarity of service questionnaire: As we previously found significant differences for different familiarities of services, we asked participants for their familiarities with the services used in the study as either *unknown*, *known*, or *used*.
- URL test for normal URL notation: The first URL test included a short introduction to the normal URL notation (see Figure A.7 in Appendix A.4), followed by a test of 50 URLs (20 benign, 30 malicious). Each URL was presented on a new page in the survey, together with a screenshot of the corresponding website and the two answer possibilities *legitimate* and *phishing* (see Figure A.9 in Appendix A.4 for an example).
- URL test for RDN notation: The second URL test was structured and designed identically to the first one, with an introduction to RDN notation (see Figure A.8 in Appendix A.4) followed by 50 URLs in RDN notation for classification, shown together with a screenshot of the corresponding website.

The URL test in RDN notation was created by rewriting half of the URLs from the study on URL categories to RDN notation (see Tables A.12 and A.13 in Appendix A.4 for an overview of all URLs in the test and their mean performance scores). The selection on which URLs to select for RDN notation was performed by first splitting all URLs into their categories and subcategories, even if we did not find significant differences between subcategories during statistical testing. We then sorted the URLs by their mean detection rate in the preliminary test (including URLs of unknown services), and alternatingly selected URLs for the RDN and normal notation from each category. This results in two sets of URLs with very similar mean detection rates in the study described in Chapter 5, while also ensuring that all categories appear in both tests. We further selected one additional impostor URL for RDN notation, since the number of URLs in the previous study was odd. Finally, we replaced the query component in benign URLs with a path component for three URLs in both notations, as differences between path and query components in benign URLs were not part of the first study.

As in the preliminary study, the participants received a short overview of their performances after completing the main survey (see Figure A.6 in Appendix A.4). In this study, performances for the two notations were separated in addition to the differentiation of benign and phishing URLs, resulting in four overall scores.

7.3. User Study Results

As before, we performed statistical null-hypothesis testing in an attempt to answer the research questions. Performance scores are again computed as relative scores, i.e. number of correctly classified URLs divided by the overall number of URLs, and measured using interval scales. We use the indices **N** for normal notation and **R** for RDN notation.

Table 7.1.: Overall performance scores for the two URL notations.

Class	Mean_N	SD_N	Mean_R	SD_R
Phishing	0.830	0.126	0.821	0.139
Benign	0.850	0.206	0.803	0.239
Overall	0.839	0.090	0.814	0.103

Table 7.2.: Statistics on time taken to classify URLs for the two notations.

Notation	Median	Mean	SD	Minimum	Maximum
Normal	8.259	9.546	3.768	5.501	21.815
RDN	7.007	7.571	2.585	4.196	15.305

Similar to previous studies, we confirmed significant differences between unknown and known URLs in this survey. As such, we again decided not to include unknown URLs in the following analyses to avoid potential biases.

7.3.1. General Differences

We start the analysis with the first research question, which is concerned with general differences between the two URL notations.

Performance

As can be seen in Table 7.1, the two notations are very similar in overall classification performance, generally and when looking at phishing and benign URLs separately. While some differences can be observed, and the performances for URLs in normal notation are generally superior to the RDN notation, the mean differences are relatively small ($MD = .047$ for benign URLs, $MD = .025$ overall) with standard deviations between .090 and .239.

A two-tailed student's t-test comparing the overall performance does not return significant differences between the two notations: $t(46) = 1.931, p = .060, d = .282$. Nor were there significant differences for phishing or benign URLs, where a deviation from normality was detected in both cases, and Wilcoxon signed-rank tests yielded results with $p \geq .101, r \leq .319$.

Timing

In addition to performance, we also looked at the differences in time needed for classifications for the two notations. Here, participants took less time classifying URLs in RDN notation (see Table 7.2), with a mean difference of 1.975 seconds per classified URL.

A Wilcoxon signed-rank test, which was performed due to deviation from normality (Shapiro-Wilk, $p < .001$), confirms that the time taken for URLs in the normal notation was significantly higher than for RDN URLs: $W = 1033, z = 4.963, p < .001, r = .832$.

Table 7.3.: Performances for normal and RDN notation for different URL categories.

Category	N	Mean_N	SD_N	Mean_R	SD_R
B-Plain	47	0.966	0.090	0.929	0.177
B-Path	47	0.788	0.312	0.754	0.343
P-Http credentials	47	0.649	0.416	0.500	0.489
P-Path	47	1.000	0.000	0.894	0.254
P-RD	47	0.744	0.310	0.736	0.341
P-Subdomain-end	47	0.968	0.162	1.000	0.000
P-Subdomain-first	47	0.593	0.394	0.838	0.266
P-Typo	47	0.922	0.106	0.865	0.113

As there were only small differences in classification performance overall, we take a closer look at the classification performances for different categories next.

7.3.2. Differences per URL Category

For the second research question, we look at differences in URL categories between the two notations, making use of the eight simplified categories presented in Table 5.12 in Section 5.3.8. The original hypothesis of using RDN was, that subdomain categories are easier to detect using this notation, as the malicious part no longer makes up the left-most part of the FQDN (thus making URL reading easier). Here, there seem to indeed be differences between the two notations (see Figure 7.1, Table 7.3), as mean differences, particularly for the subdomain category, are higher in favor of the RDN notation. On the other hand, URLs in the *RD*, *http credentials* and even *path* categories were generally detected worse with RDN notation. Similarly, the *benign* categories were detected with better accuracy in the normal notation.

We performed a factorial repeated-measures ANOVA, using the notations as first, and the eight simplified URL categories as second factor, to test for significance of these differences. The ANOVA ($N = 47$) confirms significant differences for the interaction between notation and URL category (Greenhouse-Geisser correction $\epsilon = .561$): $F(3.925, 180.546) = 7.830, p < .001, \eta_p^2 = .145$. Post-hoc tests using Holm's correction confirm differences between the two notations only for two URL categories: http credentials ($p = .036, d = .539$), where the normal notation has the higher scores, and subdomain first ($p < .001, d = .889$) where the RDN notation performs better.

In all, we accept the hypothesis that there are differences between the notations when looking at specific categories, and particularly note the differences for the subdomain and http credentials categories.

7.3.3. Comparison to Previous URL Study

As the setup for this study is based on the study on URL categories in Chapter 5, and the base URLs were mostly the same, we also compare the two studies for **RQ-3**. Note, that the URLs in the study are not exactly the same, as we replaced the query component with a path component in three benign URLs for each notation in this study, to make this comparison available, and added one URL overall to achieve a

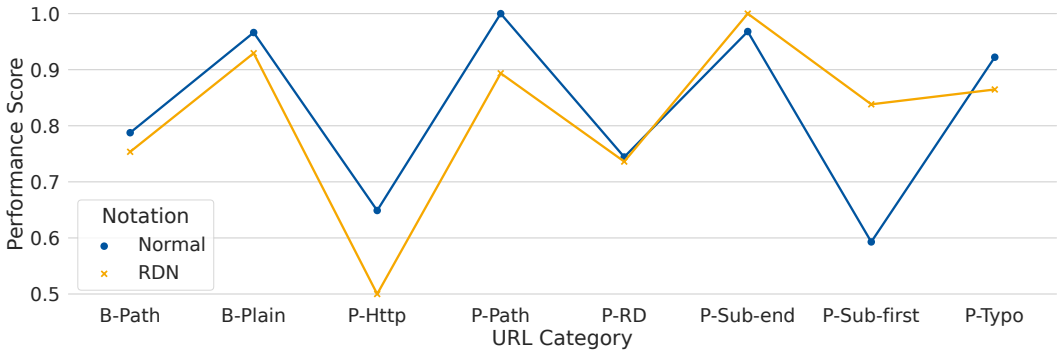


Figure 7.1.: Differences between normal and RDN notations for the different categories of URLs.

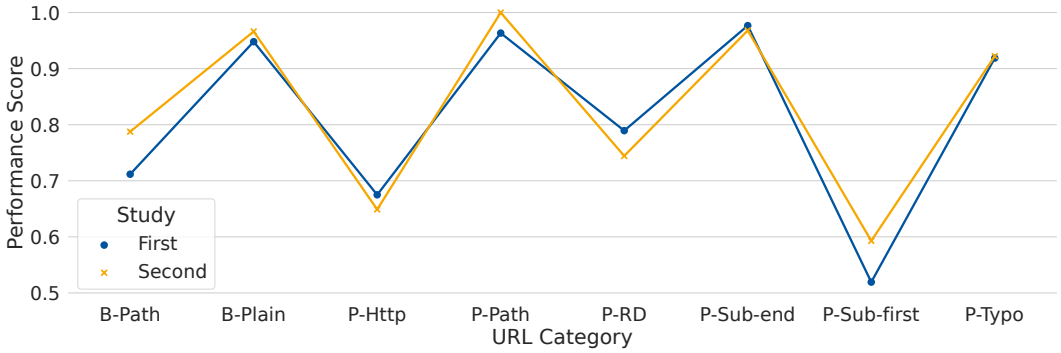


Figure 7.2.: Differences between first and second study for URLs in normal notation.

total of 100 URLs (which is divisible by two).

Comparing the results of the first and second studies, we did not find significant differences for the URL in normal notation (see Figure 7.2), with only slight differences in benign path URLs ($MD = .092$), which were detected with higher accuracy in the second study (even for benign URLs with only a query component). A repeated-measures ANOVA with the categories as repeated measures and the study as between-subjects factor ($N_{first} = 38, N_{second} = 47$) is inconclusive for the between-subject factor, as well as the Study x Category interactions, with $p \geq .526, \eta_p^2 \leq .009$.

This changes for the URLs in RDN notation (see Figure 7.3), where there are notable differences, especially in the subdomain category where the target appears first (MD=.263), but also the http credentials category (MD=.197). All other mean differences are at or below .080, which is the difference for the phishing *path* category. The second RM ANOVA for RDN URLs ($N_{first} = 38, N_{second} = 47$) is inconclusive for the study interaction as well ($p = .862, \eta_p^2 < .001$), however it yields significant differences for the Study x Category interaction: With a Greenhouse-Geisser correction of $\epsilon = .464$, $F(3.249, 269.676) = 4.557, p = .003, \eta_p^2 = .052$. Here, post-hoc tests using Holm's correction indeed confirm significant differences between the first and second study, but only for the *subdomain-first* category ($p = .002, d = 0.931, MD = .263$).

In all, these results seem to confirm the differences for the subdomain category (and to a lesser extend the http credentials), as we did not observe a potential bias

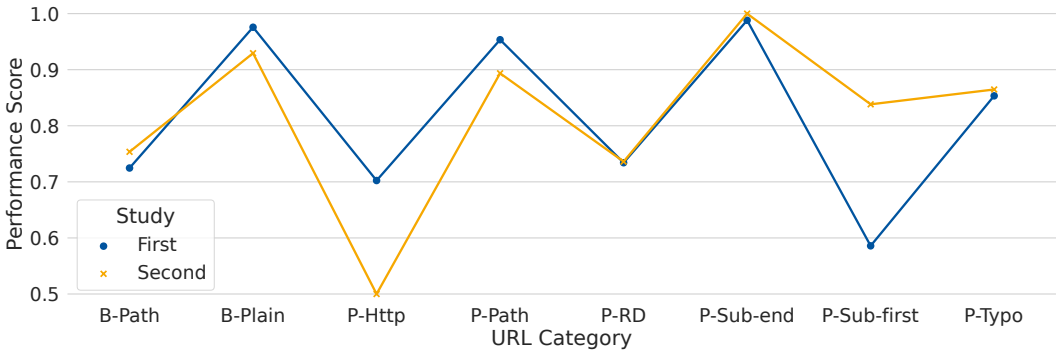


Figure 7.3.: Differences between URL categories in first (normal notation) and second (RDN notation) study.

between first and second study for URLs in normal notation, but did find large differences for the RDN notation.

7.4. Discussion

In this chapter, we presented a user study to evaluate the effects of changing the notation of URLs on the classification performance of users. While we found no differences in classification performance overall, a closer look at different URL categories revealed advantages for RDN notation in subdomain posing, at the cost of a lower performance for http credentials. In the following, we discuss limitations and implications of the study and its results.

7.4.1. Limitations

As with the previous studies, we note that the population of the sample does not represent the general public. While the distribution of age includes some variety, most participants were aged between 20-30, which excludes large parts of the population. In addition, the sample size of 47 participants might not be large enough, even when using the repeated measures design. As such, future work might replicate the study using a larger and more representative sample. Still, we argue that the effects of RDN notation on performances are likely to generalize, since the notation moves subdomains away from the front of the URL, which was consistently shown to be a problem for users in URL classification tests (see, e.g., Chapters 5 and 6). Like the previous studies, the study was also performed in a lab setting, since we did not perform a simulated phishing attack to test how the notation affects decisions in a more realistic setting. Here, awareness of attacks is important as well, since the notation is unlikely to make a difference if users are not looking at the URL in the first place. We argue, however, that the shorter time taken to classify URLs in RDN notation is an indicator, that the notation might be successful in a realistic setting as well, as it seemed to remove cognitive load when parsing the URL.

As for the setup of the study, the order of the two notations was fixed, with RDN notation always appearing after the normal notation. While we did not observe a learning bias in previous studies, it is possible that the order had an effect on classification performances. This is somewhat addressed by the comparison to the

first study, which completely randomized the order and had very similar scores to the normal notation, but is still a possible limitation of the setup. We further note, that the introduction text of the notations, in particular the RDN notation which explicitly states that users only have to look at the beginning of the URL now, may have influenced study results, in particular concerning the time taken to classify. We note, however, that even if this was the case, the results for most categories were similar to normal notation, thus RDN notation offered no clear disadvantages while decreasing the required classification time.

Finally, we did not test the effect of more complex subdomains in benign URLs, which we showed can be significant in the previous chapter. We furthermore did not include uncommon TLDs or FQDNs that are particularly crafted to be confusing in RDN (e.g., '`office.microsoft.com`', which uses the '`office`' TLD). A more complete test might indicate, whether RDN could improve detection results of the less common subdomains, as they no longer appear first, or uncommon TLDs as well.

7.4.2. Implications

Overall, we did not find significant differences between the two notations in classification performance. Only a closer look at different categories of URLs revealed differences, with RDN notation being the superior option for subdomain posing, but lacking in the http credentials category. The novel notation also generally performed somewhat worse than the normal one in the remaining categories. It is possible, that these overall differences are due to the lack of familiarity of the participants with the new notation, and that the effects vanish if RDN notation was used over a longer period of time. Still, we note that the positive effect on subdomain posing URLs already warrants a discussion about the benefits of using RDN in general, as the category was detected with the lowest accuracy in the previous study, and is potentially harder to detect using automated methods that rely on information in certificates or TLD zones, due to the lack of information on subdomains in these sources. This is also supported by the fact that http credentials seem to be far less commonly used compared to subdomains (see Chapter 4), and their risk could be reduced by, for example, not displaying them in the browser URL bar. RDN also offers advantages over highlighting of, e.g., the registrable domain, as is done by several popular browsers, as it is not dependent on the context (i.e., a URL in RDN can be used in other contexts, for example shared via email), though the two methods could also be combined to offer both of their benefits.

In all, we argue that, based on the results of the study presented in this chapter, RDN notation could potentially decrease the risk of phishing attacks, in particular when using subdomain posing. In contrast to URL pruning, it does not remove any information from the URL, and resulted in faster responses overall compared to the original notation. It does, however, require future work to confirm these findings in a real-world setting and on a different set of URLs, in addition to the effort that would be required to introduce the notation to replace the normal notation.

Chapter 8

Anti-Phishing Design Interventions for Email Clients

Up to this point, the focus of this thesis has been on URLs as an indicator for phishing attacks. This chapter deviates from this trend, and introduces several approaches that enable users to better detect and protect against phishing emails. Even with a shift to different methods like SMS or voice-based phishing (also called *SMishing* or *Vishing*), phishing emails are still a common delivery method: More than 150 million emails with malicious attachments were reported by Kaspersky in 2021 [Kas22], and emails were the second most common attack vector in data breaches and incidents in 2021 according to Verizon [Ver23]. Detecting malicious emails can also be relevant even with knowledge about phishing URLs, since it is possible to directly attach potentially malicious content to emails without including phishing URLs. Furthermore, phishing email detection aims to disrupt the phishing kill chain during delivery of the phishing message, and thus focuses on the attack delivery phase of the kill chain instead of the later user action. In other words, the results presented in this chapter are orthogonal to the learning-games presented in Chapter 6, which focus on education to prevent the user action, RDN notation from Chapter 7, which also provides protection at the user action phase but does not depend on education, as well as the automated detection approach presented in Chapter 10, which is applied in the preparation phase of an attack.

The approach to email detection analyzed in this chapter is about the User Interface (UI) of email clients, where we evaluate, whether the UI can be altered to help users detect phishing emails. In the following, we first describe the new UIs, before presenting the setup and results of a user study comparing them, followed by a discussion of the findings of this chapter.

Contributions: The main contribution of this chapter is the design and evaluation of a user study to research the effect of different email client UI designs on email classification accuracy. These design interventions aim to raise awareness and focus the attention of users on relevant information in real-world situations, thus potentially nudging users towards more secure behavior without the time and resource investment that is required for active education. The email client UIs compared in this study extend previous work by their focus on different aspects of email sender identities and how the conversation history and security mechanisms of a given email are

tied to these identities. The presented study furthermore includes clearly defined categories of phishing and benign emails, that correspond to different scenarios and give additional insights into the victim's behavior in spear phishing attacks and other real-world situations. The design of email clients and the creation of example emails were supported by the student assistants Luisa Lux and Tilbe Ugurel. The results of this chapter are new contributions of this thesis and currently unpublished.

8.1. Email UIs

When end-users interact with emails in their daily life, they typically do so via a specific UI, like a web or desktop email client (e.g., Thunderbird[1]). These clients offer advantages over interacting with the email in a purely textual form, such as rendering HTML content and the extraction of relevant information from the email headers. However this ease of use comes at the price of removing some of the information contained in a typical email from the UI (e.g., '`received:`' and other headers). As noted in the preliminaries (see Chapter 2), emails have several sender identities, that are not necessarily aligned, and can differ even in the claimed sending domain. If these identities differ, and only a fake identity is presented to the user, this opens an opportunity for attackers to spoof the sender, thus potentially increasing their chance of success in a phishing attack.

In this section, we present four different UIs that extract or highlight different information from the email header or from previous emails, thus extending the information that is typically presented by common email clients, with the aim of comparing them in a user study. The compared UIs are based on the original Thunderbird UI, and consist of a *plain* UI, *history* UI, *highlighting* UI and *spoofing* UI.

The proposed UIs are based on the Thunderbird email client for desktop PCs. We found, that the information shown in this UI is similar to that of other popular desktop and web email clients[2] (i.e., Outlook[3], Gmail[4], Apple Mail[5], Yahoo! Mail[6]), which also show the message `From:` identity, sometimes reducing it to the display name for known contacts. We discuss the decision to use the Thunderbird desktop UI as basis for the study in Section 8.4 below.

8.1.1. Plain

We use the Thunderbird UI as starting point for the UI generation, which is used without modifications for the first UI, which we call the *plain* UI for simplicity. It serves as a baseline and is meant to represent a typical UI in the way it is presented to users of desktop PCs nowadays. As such, it focuses on presenting information about sender and email content, but does not show additional security indicators.

An example email using the first UI is depicted in Figure 8.1. As can be seen, the default Thunderbird UI includes information on the sender (as determined by

[1] `https://www.thunderbird.net/` online, accessed 2023-02-13
[2] `https://www.litmus.com/email-client-market-share/` online, accessed 2023-02-13
[3] `https://outlook.live.com/owa/` online, accessed 2023-02-13
[4] `https://workspace.google.com/products/gmail/`, online, accessed 2023-02-13
[5] `https://support.apple.com/guide/mail/welcome/mac` online, accessed 2023-02-13
[6] `https://mail.yahoo.com/` online, accessed 2023-02-13

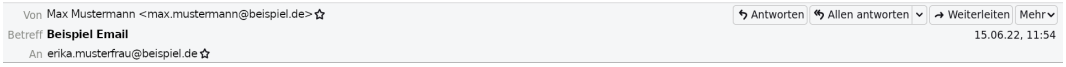


Figure 8.1.: Relevant part of the *plain* email UI.

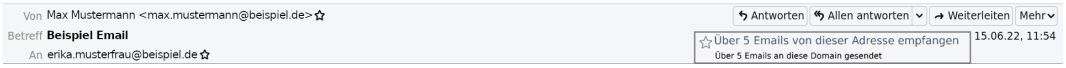


Figure 8.2.: Relevant part of the *history* email UI.

the message `From:` header), and the receiver, and also displays information on attachments (name and size), and the target of a link upon hovering over it. It additionally includes a blue star for contacts that are part of the user's address book, however we did not make use of this feature for the current study (i.e., no contacts had the blue star in the plain UI) to have a better comparison to the second UI which focuses on the relationship to the sender. Nor did we include indicators of PGP or S/MIME encrypted or signed messages, which is indicated by a letter icon in Thunderbird, as this information is the focus of the fourth UI.

8.1.2. History

The second UI, called *history* UI enhances the plain UI by adding an additional information box (see Figure 8.2). It shows, how many emails were received from the sender of the currently displayed email in the past, thus highlighting whether the receiver interacted with the sender in the past. Additionally, the second line includes information about how many emails the receiver has sent to the sender's domain in the past. The second line was included to combine information on whether the recipient has ever communicated with the sender in the past (one-way or two-way conversation), as well as to add details on the sender domain, not only the address. The UI is thereby focused on the relation between the receiver and the sender, in particular with regards to their correspondence history. The newly added information box does not conceal any existing information, thus purely adding content to the UI. The design is meant to offer information without forcing an opinion, and therefore refrains from using strong signaling colors (e.g., green if address is known). This is also the case for all other proposed UIs. It does, however, show a star symbol if emails have been received from this address previously, similar to the star in Thunderbird that marks previous correspondents, while a *not available* symbol (⊘) is shown if no emails have been received previously.

Since we did not implement the UI in practice, the design does not specify which sender identity should be used when compiling the history. However, we assumed for the study that the message `From:` header is utilized, and include a category of phishing emails where the sender identity is benign (e.g., due to account compromise), thus also testing the scenario where an attacker was able to spoof the message `From:` header (see Section 8.2.4 for details).

8.1.3. Highlighting

The third UI, called *highlighting* UI, also adds one feature compared to the plain UI, in that it highlights the RD of the sending domain of the message `From:` header. As

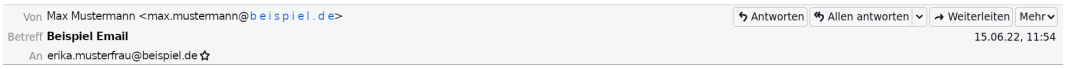


Figure 8.3.: Relevant part of the *sender highlighting* email UI.

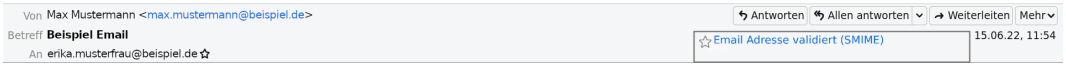


Figure 8.4.: Relevant part of the *spoofing* email UI.

can be seen in Figure 8.3, this is achieved by changing the font color and spacing in the sender display. We decided to increase the space between the characters of the highlighted RD as it has been previously proposed, that this modification improves typosquatting detection by increasing the readability of separate letters (see [Vol+17]). The sender domain with highlighting replaces the sender domain without highlighting in the `From` field in the UI (`Von:` in Figure 8.3), thus not adding superfluous content in the form of two senders. Again, we assume that the message `From:` header is used for this UI, as is the case in the plain UI, resulting in the highlight as the only difference. Note, that this UI does not include any positive or negative indications about the sender legitimacy, and is only used to highlight the RD to make users focus on this information.

8.1.4. Spoofing

Finally, the fourth UI, called *spoofing* UI, is the only UI that includes information from the envelope. Similar to the highlighting UI, it highlights information in the address, however this time, the highlight includes all information that was verified using an email security mechanism, instead of only the RD of the sending domain. As some mechanisms (e.g., PGP, S/MIME) cover the whole email address, it is possible that the complete address is highlighted.

Similar to the history UI, the spoofing UI also adds an additional information box to the plain UI (see Figure 8.4). Situated in the same position as the previous history feature, it displays information on the sender identity, as well as identity verification results of the current email. In this scenario, we assume that the receiving mail server adds the outcomes of all verification methods it performs to the email message (e.g., using an '`Authentication-Results`' header), such that the outcomes are available to the email client. Based on the outcome of the verification methods, the UI adds different highlights based on the following decision process (see Figure 8.5 for an overview of the process): First, if none of the verification methods are successful, e.g., when none of the methods is available or the email was not actually sent by the claimed sender, the color of the sender domain is changed to gray, and the information box states, that verification was not possible and, if applicable, which domain was actually used in the envelope `From`. If, on the other hand, at least one verification method was successful, we differentiate three additional cases: (1) the whole address in the message `From:` header could be verified, (2) only the domain of the message `From:` header could be verified, or (3) none of the verified information matches email address or domain in the message `From:` header. The first case, which for example occurs when the whole address was verified using S/MIME, results in a blue border for the information box and a blue highlight of the sender address

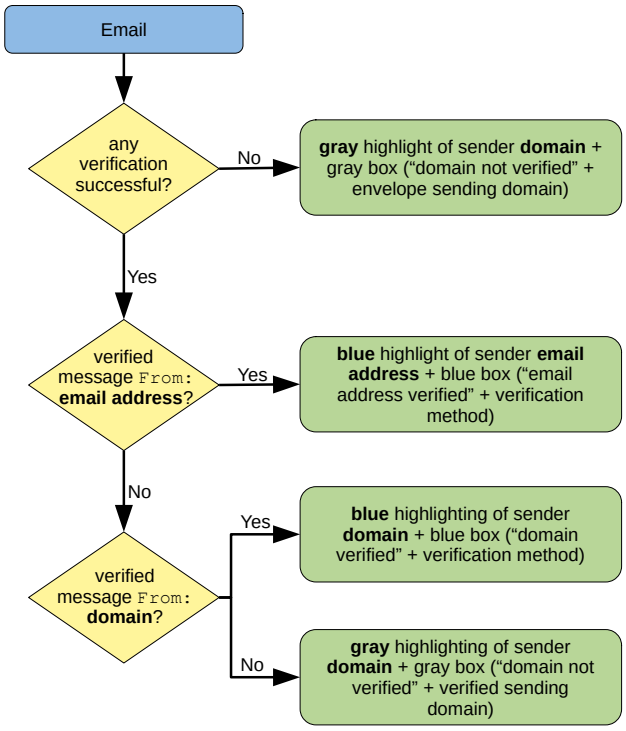


Figure 8.5.: Flow chart of the decision process for the *spoofing* UI.

(depicted in Figure 8.4). Similarly, the second case, which might for example be the result of a successful SPF or DKIM verification, results in a blue border for the information box but only highlights the domain of the sender address. The third case, on the other hand, again changes the color of box and sender domain to gray, and the information box states, that verification was not possible and which domain was actually used and verified in the envelope `From`. The box therefore incorporates the envelope identity if it was used for verification or the information differs from the claimed sender.

As we again did not implement the UI, we tested for several combinations of potential verification outcomes, including perfectly benign headers in phishing emails (e.g., due to account compromise), but also the case that the message `From:` was spoofed but mechanisms such as SPF, DKIM, or DMARC returned failures for the verification (see Section 8.2.4 for details). Note, that some of the verification methods cannot generally be performed by the user or email client, thus requiring the cooperation of the receiving MTA for this UI to be used in practice, for example by providing results via the '`Authentication-Results`' header (e.g., SPF, DMARC require information about the sending MTA, and DKIM verification might no longer be possible if the public key of the sending server was changed after the email was sent).

8.2. User Study Setup

Having created the UI prototypes, this section explains the setup of the user study we conducted to compare them.

8.2.1. Research Objectives

The main research objective is evaluating, whether existing UIs, that offer the information of the plain UI, can be improved, such that the detection of phishing emails becomes easier without offering unnecessary information to the user.

We therefore defined the following research questions:

- **RQ-1**: Are there differences in email classification performance between the four UIs?
  - **RQ-1-a**: Are there differences in the time taken to classify emails between the UIs?
  - **RQ-1-b**: Are there differences in classification performance between different categories of phishing and benign emails?
- **RQ-2**: Does the level of familiarity with the services used in the emails have an effect on classification performance?
- **RQ-3**: Does the feedback on the UIs by the participants reveal preferences?
- **RQ-4**: Are there differences between the two user groups (i.e., CS students compared to a sample with more representative demographics)?

The first research question is directly related to the main research objective, with a focus on time taken to classify in **RQ-1-a**. **RQ-1-b**, on the other hand, is meant to verify that it makes sense to differentiate between the selected email categories, and can give insights on the detection difficulty of different phishing attack scenarios, and how they differ between the UIs. With the second question, we aim to verify whether we have to differentiate between levels of familiarity in the tests, and whether the differences we observed for phishing URLs in previous studies (e.g., the URL categories study, URL notation study) are present for emails and this study setup as well. Next, we compare the outcomes of the feedback that participants gave after the classification task in the third research question. Finally, we also compare two different user groups, as the study was first tested with Computer Science (CS) students before being repeated with a more representative sample. This makes it possible to study differences between users with and without a background in CS regarding classification performance, as well as the perceived usefulness of the proposed UIs.

8.2.2. Process and Material

The study consists of an online survey, which we created using the LIME survey software. Participants were directed to the survey via a link, and could work on the survey in their own pace without any supervision.

The survey consists of:

- A **general introduction**, where the scenario of the study is introduced, together with a short introduction to email identities and phishing attacks (see Figure A.17 in Appendix A.5).

- The main part, where the four UIs are compared in a **classification test**. Here, UIs are first introduced (see Figure A.18 in Appendix A.5 for an example), followed by a classification task where users select either *phishing* or *legitimate* for the shown emails (see Figure A.19 in Appendix A.5 for an example). The emails were included as screenshots, together with a short explanation of the context the email was received in (e.g., explaining that the receiver of the email uses the service that is targeted in the email, to provide context that would otherwise be lacking for a classification task). In all, the test contains 30 emails: 6 for UI1, 8 for UI2, 8 for UI3 and 8 for UI4 (see Section 8.2.4 for details on the emails used in the test). The categories were always balanced (same amount of phishing and benign). The UIs were included in the study in the order they were presented above (plain, history, highlighting, spoofing), the order of emails per UI was randomized per participant.

- A survey that asks the participants for their **opinion on the UIs**. Questions were focused on the perceived usefulness in detecting phishing and benign emails, inclusion of unnecessary or exclusion of essential information, whether the information was understandable, as well as whether the participants could imagine using the UIs in their day-to-day email interactions (see Table A.15 in Appendix A.5 for the exact questions). It was included to collect the subjective impressions of participants on the usefulness and usability of the UIs, and compare it to the outcomes of the classification task. The answer options for each question were presented as a 6-point Likert scale ranging from 1 - "do not agree at all" to 6 - "agree completely".

- A **familiarity of service** questionnaire, which is similar to previous studies in that it asks users whether they *use*, *know* or *do not know* the services presented in the email classification test.

- A question about **previous knowledge in IT-security**, included to report on potential bias.

- A **conclusion**, with feedback (number of correctly classified phishing and benign emails) and additional information for interested participants to learn about phishing attacks, followed by an option to provide anonymous feedback.

After completing the survey, the participants were presented with a method to reach out to the study supervisors in case of questions or concerns, in addition to the anonymous feedback option.

As noted previously, our institution does not offer ethical reviews for general studies. In order to ensure high ethical standards, we therefore made sure to follow the setup of previous studies, offered participants additional information in case they were unsure about their online behavior after the study, and did not collect personal information.

8.2.3. Scenario

The study defines a scenario, that was used in the classification task to present additional context. Participants of the study were asked to act as if they were "Camila Antolin", a fictional character working at "Good Corp". In this way, we

were able to integrate a specific corporate setting into the study without having to pre-screen participants or personalize the questionnaire. This makes it possible to analyze the response of participants to several categories of phishing emails that are more commonly encountered in a corporate setting, including spear phishing and lateral phishing. To this end, the fictional character Camila was shortly introduced to participants before the start of the study, which included information about the email domain of her company (`good-corp.com`, with Camila's address being `antolin@good-corp.com`) from which all work-related emails were expected, according to the setting (see Figure A.17 in Appendix A.5). Participants were also told to expect both work-related and private emails on her email account.

The introduction also includes a definition of phishing attacks, together with a short overview on email security and sender identities. While the different sender identities of an email might not be common knowledge, the distinction was required to understand the spoofing UI, which incorporates information from message and envelope identities. As such, we decided to include it in the beginning, with the aim of avoiding biases when participants only learn about these identities for the last UI, thus potentially changing their detection approach due to the new information.

8.2.4. Emails and Email Categories

The main metric to compare the UIs is the classification performance of the participants in the email classification test. The selection of emails is therefore an important task, and has to ensure that the differences between the UIs are as small as possible, without repeating any emails or providing information that could lead to a learning bias. We therefore set out to define different categories of phishing and benign emails that were meant to have little variance in classification difficulty per category, but define different scenarios that might lead to a high variance between different categories - similar to the URL categories in Chapter 5.

Email Categories

We first define three different categories each for benign and phishing emails meant to cover a large amount of possible scenarios.

For **benign** emails, we differentiate the categories: (**b-1**) subscription services (e.g., emails from services such as PayPal, or Spotify), (**b-2**) fictional smaller services (e.g., a lesser known online shop), (**b-3**) company email (e.g., an invitation to a meeting by a colleague). We included (**b-1**) as we assume it to be a common kind of email that private users encounter in their day-to-day lives, and in particular the most likely target of mass-phishing attacks. The senders of emails in this category are popular legitimate services and all included URLs and email header information is benign and without ambiguities (i.e., the RD of included links and the sending domain match the service's RD). Category (**b-2**) was included to test how users deal when confronted with benign services that have misaligned or unexpected information in the headers (e.g., using an email sending service, meant to test how users interpret information on the proposed UIs that marks benign emails as suspicious). For emails in this category, the RD of an included URL or the sender address does not match the RD of the service (e.g., by using `amazonses.com` or similar), which might for example be the case for emails from smaller businesses. The last benign category,

(**b-3**), was included to provide a benign counterpart to the phishing emails that only make sense in the company setting. Emails from this category were sent from the company domain `good-corp.com`, and all links in the emails are benign.

For **phishing** emails we similarly differentiate (**p-1**) mass phishing, (**p-2**) spear phishing, and (**p-3**) lateral phishing. The mass phishing emails of (**p-1**) target known services and exhibit typical indicators of phishing emails, such as generic greetings, or email addresses with obvious modifications (e.g., only the display name is related to the targeted service). In comparison, emails from category (**p-2**) are customized and targeted and do not include obvious indicators, however while the sender email addresses do include references to the target, they still include modified RDs which can be detected by users (e.g., using a `gmail.com` RD for company email). For (**p-3**), we assume a compromised account, such that all email headers are legitimate and match the message `From:` address, and the email address is that of a legitimate colleague. We also added a fourth phishing category that only appears in the spoofing UI, where the correct domain is displayed for the sender but the UI indicates an attack where the envelope `From` differs from the message `From:` header. We include the categories (**p-2**) and (**p-3**), because targeted attacks have become more common, as has lateral phishing, both of which can be evaluated using these categories. Furthermore, the fourth category for the spoofing UI was included to see, if the participants actually look at the information in the UI to make decisions (i.e. in order to compare performances of category (**p-4**) to those of category (**p-3**)). Note, that all phishing emails always include a clue that they are malicious, either in the email address or an included URL. Emails of categories (**p-3**) and (**p-4**) therefore include malicious links, as it would otherwise be impossible or ambiguous whether the emails are malicious.

Email Creation

Having defined the different categories, we created the 30 emails for the classification test as follows: The goal was to find emails that represent the categories well and do not include any identifiable information. The emails for the classification test should also include a link or an attachment, with a request to perform some action (i.e., a reason to click on the link, or open the attachment). Furthermore, emails in the same category should be of similar complexity, i.e., they should result in similar accuracies when included in the classification test. We therefore started by collecting emails to be used in the survey from three main sources: the inboxes of the author and colleagues, spear phishing email collections on the Internet[7], as well as the Enron[8] dataset for benign emails. In detail, the benign and phishing emails collected from the inboxes of the author and colleagues were curated and used mainly for categories (**b-1**), (**b-2**), (**b-3**), and (**p-1**). We randomly sampled the emails from the Enron dataset and added several emails with a link and request to perform an action to the corpus for category (**b-3**). The spear phishing emails were mainly used in categories (**p-2**), after changing the context to fit the scenario of the study, but also for inspiration in creating emails for (**p-3**), as the content of the emails is similar for both categories. As such, emails for the categories (**p-3**) and (**p-4**) were mainly created manually using spear phishing emails as basis, by changing header

[7] `https://targetedemailattacks.tumblr.com/` online, accessed 2023-01-25
[8] `https://pages.aueb.gr/users/ion/data/enron-spam/` online, accessed 2023-01-25

information and context.

All emails were curated to replace the actual receiver with the fictional character Camila Antolin, URLs and other personal information were replaced, and the target company and sender of the emails were adjusted where it was necessary to fit the scenario. We discuss potential shortcomings of this creation process in the discussion in Section 8.4.

In all, we created 24 benign emails (8 per category) and 27 phishing emails (8 per category except (**p-4**)). To create screenshots of the emails making use of the different UIs for the user study, we first created screenshots of the emails in the original Thunderbird UI for the plain UI, and then used image manipulation software to create images of the newly proposed UIs. For the history UI, we attempted to create numbers of previous messages from the same sender that are realistic for the different scenarios, resulting in at least 3 previous messages for (**b-1**), 1 for (**b-2**), at least 5 for (**b-3**), 0 for (**p-1**) and (**p-2**), and at least 5 for (**p-3**). In the spoofing UI, emails from (**b-1**), (**b-3**), (**p-2**), and (**p-3**) were shown with successful verification results, while the remaining categories were shown with unsuccessful results. Examples of emails from all categories can be found in Appendix A.5.

8.2.5. Email Selection

In all, 30 emails were selected for the classification test, with 8 emails per UI except for the plain UI with 6 emails.

The spoofing UI includes emails from all seven categories, and two emails from (**b-1**) to ensure the same number of benign and phishing emails. All other UIs were shown with emails from all categories except (**p-4**), which was designed and only makes sense for the spoofing UI. The newly proposed UIs each include two emails for a given category: the highlighting UI contains two (**p-1**) and (**b-1**) emails, the history two (**p-3**) and (**b-3**) emails. We chose to include 8 emails for all of the newly proposed UIs such that all proposed UIs are shown with the same number of emails overall. This also enables us to study the variance per category for the categories where at least two emails were included for a given UI. The plain UI only includes one email per category, resulting in 6 emails overall. We discuss this decision in Section 8.4.

We selected the emails uniformly at random from the set of available emails by numbering the emails per category, and for each UI and category generating random numbers to indicate which email to include. We generated random numbers per UI and category until we found the required amount of emails that were not previously chosen for a different UI, thus ensuring that the emails per UI are distinct. In the classification test, the order of UIs was fixed, and the same emails were shown per UI for every participant, but the order of emails per UI was randomized per participant.

8.2.6. Participants

The study was performed in two phases with two different demographic groups. For the first phase, CS students were invited to participate in the study via an announcement in an online course room on computer networking. The invitation email included a short summary of the setting and goal of the study, and a note on the estimated completion time of roughly 30 minutes. The first phase was meant

as a preliminary study to test, whether the questions and answer options were comprehensible and unambiguous, or whether there were problems with the selected emails or explanation texts. However, since we only changed minor details after this phase, we still compare the results to that of the second phase, which was performed by a more representative demographic. Note, that it is likely that the first phase includes a selection bias of interested students, since we did not offer any rewards for participation, and also had a high number of incomplete responses.

For the second phase, recruiting and handling of participants were performed via Prolific[9]. Similar to our previous study using this platform, we added a pre-screening option to only include participants with fluency in the German language, as the survey was completely in German. We further asked potential participants not to take part in the survey if they were only interacting with emails on Apple devices, as we previously found that the UI design is significantly different in this case (e.g., URLs are displayed differently when hovering over links). The participants were again requested as a balanced sample based on gender. Participants were awarded 6 GBP for an estimated 45 minutes completion time, though the actual median time taken was 28:29 minutes.

In all, 24 CS students completed the survey in the first phase, with an additional 50 participants in the second phase. For the first study, we collected demographics in the questionnaire, as it was not provided via the platform as was the case for the second phase. Of the CS students, the majority identified as male (83.33%), and the remaining four either as female or chose not to answer. For the age distribution, one participant was younger than 18 years, the majority (87.5%) between 18 and 24 years, and two participants between 25 and 34 years of age. For the self-reported level of previous knowledge in computer science, the group tended towards the higher levels, with four reporting *average* knowledge, nine *high* and eleven *very high* previous knowledge.

As for the 50 participants in the second phase recruited via Prolific, we had an exact split of 50% between participants identifying as male and female due to the requested balanced sample. Here, the age distribution is more diverse, however the majority of participants was still between 20 and 29 years of age (66%), but with a long tail of up to 67 years, with 18% being 40 years or older. As for current occupations, 40% were currently students, with 36% being full-time and 24% part-time employed. In the self-reported previous knowledge on computer science, the answers were more varied, as 8 participants selected *low* and 1 *very low* previous knowledge, 19 *average*, 15 *high* and 7 *very high*.

In this study, we did not reject any participants who completed the survey. While there were several outliers in survey completion time, we found similar distributions for the classification performances of faster and slower participants. Since we furthermore argue, that the inclusion of less attentive participants can be a more accurate representation of the real-world scenario, where judging an email's authenticity is a secondary task at best, we therefore included all complete responses. We discuss this decision further in Section 8.4.

[9] `https://www.prolific.co/` online, accessed 2023-01-25

Table 8.1.: Mean performance scores of the four UIs for different email categories.

Category	Plain	History	Highlighting	Spoofing
B-1	0.640	0.740	0.900	**0.910**
B-2	0.640	**0.840**	0.440	0.680
B-3	0.620	0.880	0.700	**0.920**
P-1	0.780	0.740	**0.900**	0.700
P-2	0.520	0.540	**0.780**	0.440
P-3	0.100	**0.120**	0.100	0.040
P-4	na.	na.	na.	0.340
Overall	0.550	0.608	**0.703**	0.618

8.3. User Study Results

In the following, we again use statistical testing to answer the research questions of this chapter (see Section 8.2.1). As before, we compute performance scores as relative scores, measured in an interval scale, and use a significance level of $\alpha = .05$. The first four research questions are answered using responses from the second phase of the study with the more representative sample.

Note, that we did not remove emails with unknown services from this study. While we did find differences between levels of familiarity, and in particular unknown services compared to the other levels, removing these emails from the study would have greatly reduced the number of valid answers for the more complex analyses (e.g., by more than 50% for the comparison of UIs averaged over the email categories). We discuss this decision in more detail in Section 8.4.

8.3.1. Comparison of the UIs

For the first research question, we are interested in general differences between the four UIs (see Table A.14 in Appendix A.5 for an overview of all emails and their mean performance scores). As can be seen in Table 8.1, the overall scores of all three proposed UIs are superior to the baseline plain UI. However, while the history and spoofing UI performed well for benign categories, they lack behind the plain UI in the phishing categories, thus reducing their advantage overall (see Figure 8.6).

Since the distribution of categories was not equal for all UIs (e.g., the plain UI only includes one (**p-1**) email, while the history UI contains two, see Section 8.2.4), and the differences between categories are significant, we compare the UIs in statistical testing while averaging over the different categories. This is achieved via a factorial repeated measures ANOVA, where the four UIs serve as the first factor, and the categories as second factor. The Anova is significant for both factors, UIs ($F(3, 147) = 3.581, p = .015, \eta_p^2 = .068$) and categories, where Mauchly's test of sphericity was significant and degrees of freedom were corrected with $\epsilon = .741$ ($F(3.704, 181.478) = 57.233, p < .001, \eta_p^2 = .539$). Post-hoc tests using Holm's correction only confirm significant differences between the plain and history UIs ($p = .023, d = .233$) and the pain and highlighting UIs ($p = .036, d = .216$), but not for the spoofing UI ($p = .172, d = .162$). Comparing any of the newly proposed UIs does not result in

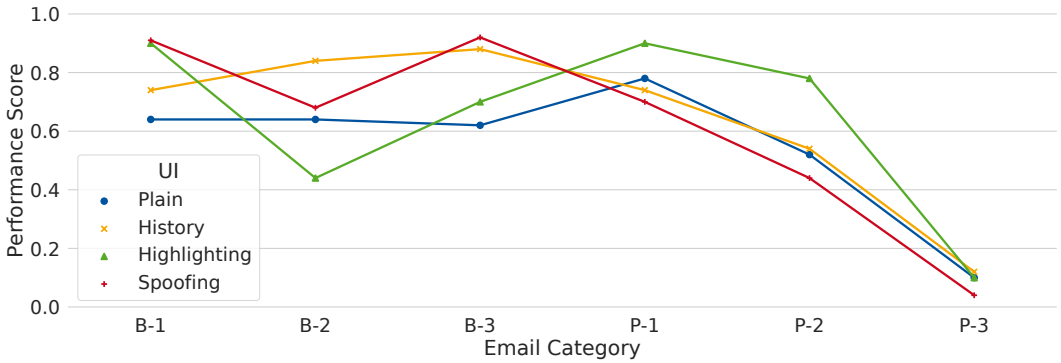


Figure 8.6.: Differences between the four UIs for the six common email categories in the second phase (Prolific).

Table 8.2.: Mean and percentiles of time taken in seconds per UI

	Mean	SD	25 percent	50 percent	75 percent
Plain	64.550	53.761	30.805	45.343	75.051
History	35.322	26.978	19.658	28.029	38.188
Sender Highlighting	32.264	20.658	18.206	23.456	36.150
Spoofing	23.949	14.505	15.923	19.304	30.389

significant differences ($p = 1, MD \leq .028, d \leq .071$). The post-hoc tests for categories revealed significant differences only between (**b-1**) and (**b-2**) for the benign categories ($p = .027, d = .368$), while all three phishing categories significantly differ from each other ($p < .001, d \geq .523$).

Time Taken

Next, we take a look at the time taken to classify emails and differences between the UIs. As can be seen in Table 8.2, there is a clear trend where the time taken to classify the emails decreased for later UIs. The most prominent differences are between the first (plain) and second (history) UI, however the step from third (highlighting) to fourth (spoofing) UI is notable as well.

A repeated measures ANOVA with the timings as factor confirms, that there are significant differences: $\epsilon_G = .441, F(1.324, 64.864) = 22.599, p < .001, \eta_p^2 = .316$. Post-hoc tests confirm significant differences between the baseline UI and all of the proposed UIs ($p < .001, d \geq .896$), but not among the proposed UIs ($p \geq .097$).

Extra Category in Spoofing UI

Finally, we compare the category of phishing emails that was specifically added for the spoofing UI, the only UI that differentiates message and envelope identities, to categories **p-2** and **p-3**. This comparison was added to determine, whether the information of the UI was used by participants in their classification decision, as the corresponding email was based on category **p-3**, differing only in a warning that the message sender could not be verified and that the envelope sender differed (see Figure A.16 in Appendix A.5). Descriptive statistics seem to indicate, that

Table 8.3.: Differences between levels of familiarity in the second phase of the email study

Familiarity	N	Mean	SD
Unknown	29	0.434	0.405
Known	29	0.623	0.295
Used	29	0.652	0.153
Fictional	29	0.578	0.300
Company	29	0.672	0.193

(**p-4**) is comparable to (**p-2**), both of which were classified with a higher accuracy than (**p-3**) (see Table 8.1). A repeated-measures ANOVA confirms these differences ($\epsilon_G = .841, F(1.682, 82.439) = 14.814, p < .001, \eta_p^2 = .232$), as post-hoc tests reveal significant differences for categories (**p-2**) and (**p-4**) over (**p-3**) ($p < .001, d \geq .721$).

In all, it seems as though all newly proposed UIs offer an advantage over the baseline in the overall comparison, however there seem to be trade-offs between improving the detection of phishing or benign emails. The time taken to classify emails decreased significantly after the plain UI, and while this trend held on for the following UIs the differences were no longer significant.

8.3.2. Service Familiarity

Even though we decided against removing unknown services from this analysis, we still report on the differences between different levels of familiarity for the second research question. Since the scenario includes fictional entities, in the form of fictional services in category (**b-2**), as well as emails from the fictional company "Good Corp" in categories (**b-3**), (**p-2**), (**p-3**) and (**p-4**), which do not have an equivalent in the real world that participants could be familiar with, we differentiate five different levels of familiarity in this study: the familiar *unknown*, *known*, and *used*, as well as the new *fictional* and *company*, representing the fictional services and corporate emails respectively. Interestingly, it seems as though the performances for the fictional services are comparable to the known or used services, not the unknown ones, as might be expected (see Table 8.3, which includes only the 29 participants who selected at least one service for each level of familiarity).

A repeated-measures ANOVA comparing the five levels of familiarity ($N = 29$) is significant: After adjusting degrees of freedom using Greenhouse-Geisser corrections ($\epsilon = .692$), the ANOVA confirms differences between the levels ($F(2.767, 77.476) = 3.399, p = .025, \eta_p^2 = .108$). Post-hoc tests (Holm) are significant for the comparisons of unknown to used ($p = .031, d = .769$), as well as unknown and company ($p = .015, d = .840$). None of the other comparisons are significant ($p \geq .086, d \leq .667$), in particular for comparisons that do not include the unknown level ($p = 1, d \leq .335$). Note, that we did not compute results for the familiarity levels over different email categories, as this would have removed too many valid entries to retain a sensible comparison. As such, the results still include certain biases, as the *fictional* category includes exactly the emails from category (**b-2**), while *company* includes all emails from (**b-3**), and only few additional ones from (**p-2**), (**p-3**), and (**p-4**).

Table 8.4.: Median agreement to feedback options for the four UIs

	Plain	History	Highlighting	Spoofing
Helps detecting phishing	2.0	**5.0**	4.0	**5.0**
Helps detecting benign	2.0	**5.0**	4.0	**5.0**
Includes unnecessary information	**1.0**	2.0	2.0	2.0
Missing information	4.0	**2.0**	3.0	**2.0**
Easy to understand	**5.0**	**5.0**	**5.0**	**5.0**
Would use UI	3.5	**5.0**	4.0	**5.0**

8.3.3. Perceived Differences

For the third research question, we focus on the results of the comparison between the four UIs as perceived by the participants of the study. In all, this questionnaire includes six questions, regarding the perceived support of the UIs in discerning phishing and benign emails, the amount of information present in the UI, and whether this information is easy to understand, as well as a verdict on whether they would use the UI for their emails (see Table A.15 in Appendix A.5 for the exact questions). As previously noted, the answer options range from 1 - "do not agree at all" to 6 - "agree completely". We present the responses to the six questions as a series of non-parametric Friedman's tests, and report the results of significant Conover's post-hoc tests using Holm's corrections for more details (see Table 8.4 for median and mean responses).

The first question asked participants whether they found that the given UI simplifies the detection of phishing emails. The Friedman's test for this comparison is significant with $\chi^2(3) = 81.916, p < .001, W = .546$. Post-hoc tests indicate, that participants preferred the history and spoofing UIs equally, followed by highlighting, with the baseline being the least popular choice with a significant margin.

The results for the corresponding question about legitimate emails are almost identical ($\chi^2(3) = 79.272, p < .001, W = .528$), with the same significant differences between UIs (history and spoofing similar and preferred to highlighting, which is still preferred to the plain UI).

The third questions asked the participants on their opinion regarding the inclusion of unnecessary information in the UIs. Here, the results are reversed from the previous two questions: Friedman's test indicate significant differences, but with lower agreement between participants $\chi^2(3) = 13.009, p = .005, W = .0.087$. Post-hoc tests confirm differences between the baseline and all three proposed UIs, with the new UIs being more likely to include information perceived as superfluous. There are no significant differences between the proposed UIs.

The next question, about whether vital information is missing in the UIs, again favors the newly proposed UIs. Here, Friedman's test is again significant with $\chi^2(3) = 56.582, p < .001, W = .377$, and post-hoc comparisons are similar to the first two questions, where all new UIs are preferred to the baseline, with the history' and spoofing UIs slightly outperforming the highlighting UI.

When asked whether the information in the UIs is easy to understand, the responses did not differ significantly according to the test ($\chi^2(3) = 2.849, p = .415, W = .019$). Note, that the responses in this case were generally favorable for all UIs, with means

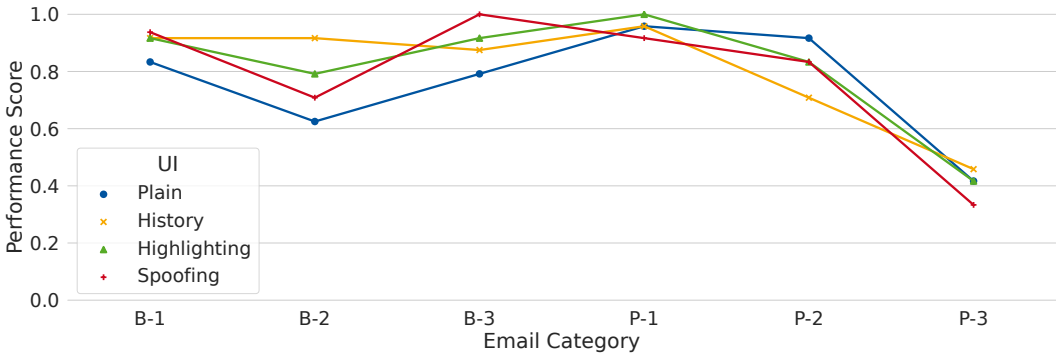


Figure 8.7.: Differences between the four UIs for the six common email categories for CS students.

between 4.94 and 5.02, indicating agreement with the statement.

Finally, for the verdict on whether participants would use the UI in practice, the differences are again significant ($\chi^2(3) = 46.707, p < .001, W = .311$), and participants preferred the history and spoofing UIs over the alternatives.

In all, participants seemed to overall prefer the history and spoofing UIs, as they were generally rated higher in comparisons regarding useful information, detection capabilities, and the final decision on whether they would use it for their emails.

8.3.4. Comparison to First Phase

While we do not go into too much detail on the first phase of the study due to the bias in demographics, we do note a number of interesting differences between the CS students and the sample provided via prolific. First, the CS students were, unsurprisingly, generally better at the classification task (see Figure 8.7). While there are still trends visible for differences between the UIs, they are not as pronounced as in the second phase of the study. In particular, the CS students achieved very high scores for almost all categories, with the notable exception of category (**p-3**). The higher variance and slightly lower performance in categories (**b-2**) and (**p-2**) might be due to missing explanation texts, which were only added after the preliminary study.

As for the perceived differences between the UIs, the results for the CS students was very similar for the questions concerning support in classifying legitimate and phishing emails, whether the information presented in the UIs were easy to understand, and the final verdict, where they clearly preferred the spoofing UI. Differences occurred for the question regarding unnecessary information, where the CS students preferred the baseline and spoofing UI over the highlighting and history UIs, with similar but reversed results for the exclusion of important information.

In all, the CS students had higher classification performances overall, though they still seemed to struggle with emails from the compromised category. While they agreed with the participants of the second phase on which UIs were helpful in the categorization task, they disagreed on the usefulness of the information presented in the highlighting and history UIs.

8.4. Discussion

This chapter presented and evaluation of the approach to utilize design interventions to support users in detecting phishing emails. A user study was performed, where the usage of different UIs for email clients had an effect on classification performances, with the proposed UIs generally leading to higher overall scores. In the following, we discuss limitations of the user study, followed by an interpretation of the results' impacts.

8.4.1. Limitations

The emails that were used in the email classification task of the user study presented in this chapter were selected according to two main objectives: enabling a fair comparison, and minimizing the amount of emails required, as we found that the classification task takes far more time than the URL classification tasks performed in previous chapters. While we attempted to create categories of emails that had similar properties and content, it is likely that there were still differences for different emails in the same category that have an unforeseen influence on how easy it is for users to classify them correctly. Together with the fact, that only few emails were shown per category, and the emails were not randomized between participants, i.e., for each UI, each participant was shown the same emails, it is possible that the comparison of UIs is biased due to individual differences of emails in a category. For example, previous studies that have attempted to categorize phishing emails based on psychological cues have found some differences that might have influenced the participants of this study differently [Oli+17]. To obtain an intuition on differences for emails of the same category, we compared the mean performances for the emails of the same category and the same UI (e.g., the two emails of category (**b-3**) in the history UI, see Section 8.2.4 for details), and found that mean differences were at most .12, which does imply some variation but less than the variation between categories. As such, while we argue that the persistently higher scores of the history and spoofing UIs for benign categories and the highlighting UI for phishing URLs are unlikely due to chance, the findings of this study should be replicated in the future with a different set of emails or a different distribution of emails to UIs.

Furthermore, the number of emails shown per UI differed, with the plain UI including fewer emails than the other UIs. While we attempted to minimize the required amount of emails overall, including two additional emails in the plain UI, that could also be used to test for variance in the remaining two categories (**b-2**) and (**p-2**), might have been a better decision in hindsight.

The order of UIs was also fixed without randomization between participants, which might have facilitated a learning bias affecting the UI comparison. This particular order was chosen to minimize the potential effect of such a learning bias, as additional information, which could have been interpreted as hints on which parts of the email are important, were introduced in an order that extends the previous UIs. On the other hand, the study required concentrating on the email classification task for roughly 40 minutes, which might have led to some fatigue in the later UIs. This might also explain the lower times taken to classify emails in later UIs.

We furthermore note, that the sample again consisted of only 50 participants for the second phase, and 24 CS students in the first phase. We also excluded Apple users

from the participants, since the UI on their devices is often significantly different (e.g., differences when hovering over a link). While the evaluation using repeated measures can have statistical power for smaller numbers of participants, a more representative and larger sample might offer additional insights or support of the study outcomes.

As for the decision to include all participants, even those with very short completion times, we argue that inclusion of somewhat inattentive participants might actually be more realistic than only including highly attentive users. The reason for this is, that security is often viewed as a second priority in real-world tasks, and it has been previously argued that including less attentive users in email phishing studies is more realistic and might therefore better reflect the real world [Mat+21].

Evaluating the perceived differences between the four UIs, it is likely that the responses include at least a certain amount of novelty bias, favoring the newly presented UIs. Interestingly, this would not explain differences between the newly presented UIs, in particular the preference of the history and spoofing UIs over the highlighting UI. As such, we argue that the effect of this bias might be relatively small, but still recommend more thorough usability analyses in the future if the UIs were to be implemented in practice.

Different to the studies presented in previous chapters, we did not take familiarity of services into account in this study. While we found, that emails including unknown services were classified less accurately in this study as well, we found that removing unknown services would have reduced the number of valid samples, and thus the predictive power and validity of the tests significantly. Still, the added context due to the scenario and explanation texts for each email might have lessened this effect somewhat. We further note, that the trends we found in the UI comparison still remained when removing unknown services.

In addition, we decided to base the UIs presented in this study on the Thunderbird email client for desktop PCs, which might limit how the results transfer to other email clients. While a look at the UIs of other popular desktop, web, and mobile email clients indicates, that they currently show similar information concerning sender identities (i.e., based on the message `From:` header, without further highlighting of the email address), general differences in design might lead to a different focus that might in turn influence the classification accuracy. In particular, most email clients (e.g., Outlook, Gmail, Apple Mail, and Yahoo Mail) offer the functionality to only show the display name (instead of an email address) for known contacts, which differs from the setup in our study where the plain UI always displays the sender's email address. While we argue, that this change is unlikely to have had a large effect on classification performance, a similar design might be analyzed in more detail in the future. Note, that link hovering does not work on mobile devices, which might make the emails in category (**p-3**) more complicated to detect for users of mobile devices. In all, while we argue that the results of this study are likely transferable to other UIs that do not particularly highlight sender identities, and are thus similar to the Thunderbird UI, different UIs as basis might be evaluated in the future.

Finally, we note that none of the UIs have been implemented and evaluated in real-world settings, which is also a potential direction for future research. An evaluation in a more realistic setting might, for example, consist of a simulated phishing attack, where participants are required to demonstrate awareness in addition to the knowledge required to detect phishing emails. Such a study might also reveal the potential

effect of any novelty biases which occurred in our study, or *warning* fatigue where users no longer pay attention to highlighted information when they are presented with the proposed UIs in their daily lives over a longer period of time.

8.4.2. Impacts

In this chapter, we found that changing the UI of email clients can have a significant effect on email classification performance. While all of the proposed UIs, which highlight different information about the sender and sending domain, led to improvements over the baseline overall, we found differences between benign and phishing emails.

In particular, the performance in the study suggests, that the highlighting UI was best suited to improve phishing detection, while the history and spoofing UIs improved performances for benign emails. It is possible, that the reason for this difference is the reliance of participants on positive and negative indicators in the UIs. For example, the history and spoofing UIs both contain indicators, whether emails have been received from the sender before and whether the security mechanisms of the email are aligned, that might be interpreted as a decision on the validity of the email. If users ignore other indicators of the email and base their decision only on these indicators, this might explain how benign emails were classified with higher accuracy, while emails from the phishing categories that made use of targeted attacks or compromised accounts, resulting in legitimate indicators in the proposed UIs, would be classified with less precision. Since the highlighting UI does not have positive or negative indicators, participants had to rely on their own decisions on whether the emails were legitimate, thus resulting in the observed performances. Analyzing this effect more thoroughly in the future could result in design decisions for security indicators in emails in general. We note, however, that the decrease in time taken to classify emails after the baseline UI seems to indicate, that users did indeed focus on only specific parts of the emails for the proposed UIs.

Interestingly, the participants of both studies preferred the spoofing and history UIs to the highlighting UI for detecting both phishing and legitimate emails, as is indicated by their feedback in **RQ-3**. This perception stands in opposition to the results of the classification task, in particular for phishing emails, where the highlighting UI resulted in higher accuracies. The results of the perceived differences therefore seem to support the finding, that participants prefer UIs that can be interpreted as offering a binary decision about the validity of an email, thus indicating that choosing a neutral design might lead to improved detection capabilities for well-crafted phishing emails, where opinionated UIs would indicate a benign email. Whether a neutral design indeed reduces a false sense of security and whether it is possible to highlight additional information without introducing a bias in users' decisions could be interesting questions for future work.

Still, even though the performances improved, they were still far from sufficient to protect against some of the proposed attacks in practice, in particular emails in categories (**p-3**) and (**p-2**). Here, the UIs proposed in this chapter all focus on sender information, which makes it possible to combine them with other proposed methods. In particular, it has been previously shown that highlighting information on links in emails can have a positive effect on classification performance [Vol+17]. Other approaches, like email address separation [4], automated malware detection, or

no-attachment policies could therefore be combined to offer a more holistic protection against several categories of attacks, with different traits. These might also offer some protection against attacks where the email does not contain any decisive clues on its legitimacy, for example compromised accounts being used to spread malware via attachments or legitimate file-sharing platforms. Future work might include studies with simulated phishing attacks, where these combinations are covered in more detail.

Interestingly, there were differences to a previous study on the effect of sender highlighting on phishing detection. Nicholson et al. found much larger mean differences when introducing sender highlighting (similar to the highlighting UI) for phishing emails [NCB17]. It is possible, that the additional information highlighted in their design was essential (i.e., date and display name), however we argue that this is unlikely as this information is often easy to manipulate by attackers and therefore does not offer robust indicators for the phishing classification task. It is therefore likely, that the difference stems from a different study setup, sample populations, or differences in the chosen emails and email categories. In particular, the introduction in our study included an explanation of email sender identities to avoid learning bias that might have occurred if this information was only included later for a specific UI. This might have led participants to be more aware of the sender and thus resulted in better classification performances overall compared to the study by Nicholson et al. It might be possible to remove or shorten the introduction when repeating the study in the future, thus introducing less focus on sender identities.

As for different categories of benign and phishing emails, our study included examples corresponding to several situations, thus giving some insights into the decision processes of potential victims in these situations. Here, we found that the emails in category (**p-3**), that simulated a compromised account, were very convincing, as their detection accuracy was below .120 for all UIs, even though they always included a recognizably malicious URL as indicator. While the emails in the spear phishing category (**p-2**) were detected with higher accuracy than (**p-3**), it still signifies a significant threat, with accuracies between .440 and .780. These results offer some evidence, how vulnerable users can be against spear phishing and lateral phishing attacks if the attackers are able to craft a convincing context, for example by employing Open Source INTelligence (OSINT) techniques to obtain additional information on their targets. Recently, more targeted attacks, even combined with artificial intelligence to, for example, create more plausible emails[10], have made spear phishing more common, which in turn motivates better detection methods. The low performances in our user study for targeted attacks, together with the prevalence of social engineering in reported incidences (see [Ver23]), therefore motivate the need to research and implement effective protection techniques against phishing emails in the future. We further note, that the clear differences between the categories, including the benign categories, underlines the importance of including a diverse set of emails in user studies when testing classification performance.

[10] see, e.g., `https://labs.withsecure.com/content/dam/labs/docs/WithSecure-Creatively-malicious-prompt-engineering.pdf` online, accessed 2023-01-18

Part III.

Certificates of Phishing Websites

Chapter 9

Analyzing Certificates of Phishing Websites

URLs and emails are not the only artifacts of phishing attacks that can be utilized for phishing detection. The tendency of attackers to follow current trends to make their websites more resistant against detection has led to an increase in the usage of https for phishing websites, which also results in additional information that is available for detection, for example in the form of TLS certificates. In this chapter and the next, we take a closer look at these certificates of phishing websites. Certificates could potentially be integrated into the defense against phishing in several ways, including education (as the information in certificates might be less complex to parse than URLs) and automated detection. In this chapter, we present an overview of the information that is included in phishing and benign certificates, with a focus on impersonation in different certificate fields. We therefore collected and analyzed both phishing and benign certificates, and compared them based on general certificate features, trying to uncover features for automated detection, but also to determine whether certificates may be used for anti-phishing education.

Parts of this chapter were previously published in [3].

Contributions: The main contribution of this chapter is an analysis of the certificates of phishing websites, which was previously published and adapted from [3]. The findings were partly reproduced on a larger dataset, which is a new contribution of this thesis and currently unpublished.

9.1. Certificate Collection

This section describes the process and resulting datasets of our certificate collection efforts in detail. The objective of the collection is to provide a dataset used to determine, whether there are general differences between the certificates of phishing and benign websites, as well as differences between the certificates of popular targets and their corresponding phishing websites. To achieve these goals, we collected certificate information from benign and phishing websites, extracted relevant features, and compared them for phishing and benign certificates.

9.1.1. Data Collection

For our analysis we retrieved 39 478 benign and 9 479 phishing certificates. In the following, we first describe how we collected benign and phishing domains, followed by an explanation of how we retrieved certificates from these domains. Note, that the phishing URLs used to collect certificates in the first part of this chapter are only a subset of the dataset presented in Chapter 4, as the collection was performed at an earlier time. We extend the set by including all available certificates in Section 9.3.

Data sources and preprocessing

In order to collect popular benign domains, we used the Alexa Top million list and crawled the top 50,000 entries in January 2019. The Alexa list does not include subdomains, which might result in differences between regularly browsing to a website and our automated process, as some domains present different certificates depending on the existence of subdomains. A prominent example for this is PayPal, the most common target for phishing campaigns in our dataset. In this case, querying '`paypal.com`' led to a certificate that differs from the one returned when querying '`www.paypal.com`'. In order to mirror the experience of users more closely, we therefore applied a pre-processing step and queried all benign websites using curl[1] to follow auto-redirects. We then used the resulting domain names for all further steps.

The phishing dataset was obtained from PhishTank, a website that collects phishing websites collaboratively. Users can submit potential phishing websites and verify the submissions of other collaborators, resulting in a peer-reviewed dataset of phishing websites. However, this dataset is not completely free of false positives: We did encounter several false positives when looking at specific certificates. We assume that this is due to one of the following reasons:

- The websites has been cleared and phishing content removed, but is still shown as "online and valid" by PhishTank.
- The websites were falsely flagged and the peer-reviewed verification was wrong.

Either way, these cases seem to be rare in comparison to the dataset of true phishing websites (we found less than ten cases in our detailed analysis in Section 9.2.3). We queried the PhishTank database for online and valid (i.e., verified by other users) phishing websites once daily over a period of 54 days between December 12, 2018 and February 1, 2019 (one day was missed due to technical problems). In this time, we collected 31 264 unique PhishTank entries.

Certificate Collection

We used the following process to retrieve certificates from benign and phishing websites (see Figure 9.1): For phishing websites, since we do not want to download certificates that have already been considered on a previous day, we merged the new datasets with a list of previously visited websites. We thus reduced the queried websites from several thousands to several hundred new phishing domains per day. This is not necessary for the larger benign dataset, as these domains could be queried all at once.

[1] `https://curl.se/` online, accessed 2023-01-25

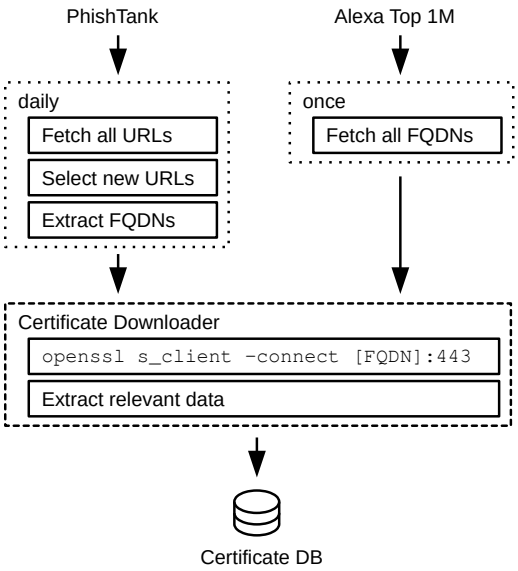


Figure 9.1.: Certificate downloading process from phishing and benign websites.

After acquiring the websites to be queried, we started the crawling process using OpenSSL[2]. OpenSSL is an open source toolkit for the TLS and SSL protocols. We used the *s_client* component of OpenSSL (version "OpenSSL 1.1.1a FIPS 20 Nov 2018") to query websites and extract certificate information[3]. As root certificates we made use of the Mozilla CA Certificate Store, which is, among others, also used by the Firefox browser[4]. We used *s_client* to connect to the specified domains on port 443 and retrieve a certificate, if possible. Note, that we did not follow redirects at this step, which might lead to differences between browsing to the website and using the automated process, but lessens the potential effect of website cloaking. Website cloaking (see Section 3.1.1) is a method to evade detection by showing different versions of a website depending on, e.g., the geolocation of the request. By requiring only the lower level TLS connection for the certificate download we minimize the effect of possible cloaking attempts. The certificates and additional information about the connection were saved on success for further analysis.

All in all, we were able to obtain 25 777 certificates from the 31 264 phishing domains. From these, we removed 11 712 duplicate certificates with respect to domain names in order to avoid polluting our dataset with several entries for a single phishing campaign. To be precise, we created a database that only contains unique domain names and for each domain name exactly one certificate. This resulted in 14 065 certificates, but introduces a bias in our dataset, which now includes phishing campaigns using different subdomains, but disregards campaigns using different URL paths. Note, that removing duplicates on a domain name basis does not imply that all of the remaining certificates are unique as well, as it is still possible that several different domain names are included in the same certificate. Next, we also

[2] `https://www.openssl.org/` online, accessed 2023-01-25

[3] `https://www.openssl.org/docs/man1.1.1/man1/openssl-s_client.html` online, accessed 2019-02-24

[4] `https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/` online, accessed 2023-01-25

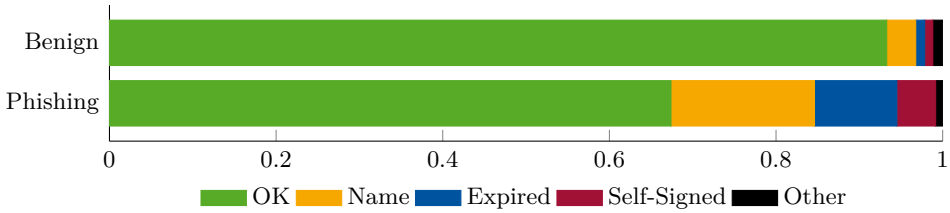


Figure 9.2.: Validity status of certificates from phishing and benign websites.

decided only to look at certificates that were valid (as recognized by OpenSSL), since browsing to websites with invalid certificates generates a visible error in all major browsers to warn users. An overview of the validity status of phishing and benign certificates can be seen in Figure 9.2. *Name mismatch* errors (the domain name of the website does not match the subject `CN` or `SAN` of the certificate) were the most common, followed by *expired* certificates (validity period is in the past) and *self-signed* certificates. Overall, phishing websites were more likely to present an invalid certificate than the popular benign websites. Our final dataset of valid phishing certificates contains 9 479 entries.

For benign websites, we removed 698 certificates with duplicate domain names and 2 842 invalid certificates and ended up with a dataset containing 39 478 benign certificates.

9.1.2. Analysis and Feature Extraction

The analysis started in a second pass, after all certificates were downloaded. Here, we scanned all certificates, extracted features of interest (see Table 9.1) and saved them in a database. The features are divided into three groups:

- **Subject Information**: This group contains the subject `Organization` as well as validity and EV information. These are usually easily available to users and directly correspond to the websites a user might expect to be on.
- **Issuance Information**: This group contains the issuer and root `CN`, as well as the validity period. These are features that go beyond subject information, but are typically still easily available to users. We use the `CN` of the issuer rather than the `O` information, as it is usually more detailed in our dataset.
- **URL information**: This group includes the subject `CN` and `SAN`, as well as the domain name of the website from which the certificate was extracted. We include this information to determine, whether looking at the certificate can be more effective than looking at the URL of a phishing website.

We disregard other fields commonly found in certificates (see Section 2.6) for several reasons: Some fields are very similar (or the same) for all certificates issued by the same issuer (e.g., signature algorithm, policies). Thus, they only differ for certificates issued by different issuers, which is a distinction we already consider. Other fields consist only of long strings of numbers, that would be impractical to deal with in the context of user education and are unlikely to be usable in the context of automated phishing detection as well (e.g., public key, serial number). Lastly, some fields simply do not offer much variation at all (e.g., key usage, basic constraints).

Table 9.1.: Selected certificate features

	Field
Subject:	`CN`
	`Organization`
CA:	Issuer `CN`
	Root `CN`
Validity:	Validity Period
	isValid
Extensions:	`SAN`
	Extended Validation

9.2. Results of Feature Extraction and Certificate Comparison

In the following, we present the results of the comparison of benign and phishing certificates. We first define the research objectives for the evaluation of certificates, followed by a comparison of benign and phishing certificates in general. We then take a closer look at the differences between certificates of phishing websites and the certificates of their targets.

9.2.1. Research Objectives

In this chapter, we evaluate the discriminatory power of certificate information when it comes to distinguishing between phishing and benign websites. Specifically we envision the following two scenarios:

1. In the first scenario, some agent (e.g., a user or automated classifier) has to determine whether a given website is a phishing website or is a benign website based solely on the content of the website's certificate.

2. In the second scenario, an agent (e.g., a user) with some prior knowledge about the benign website has to determine whether a given website is this benign website or is a phishing website targeting the benign website.

Both scenarios assume that the agent has access to the information included in certificates. Thus we investigate the following two research questions:

- **RQ-1** Are there general differences between the certificates of phishing websites compared to those of benign websites and if so which ones?

- **RQ-2** Are there differences between the certificate of a phishing website and the certificate of the corresponding targeted benign website and if so which ones?

The first research question corresponds to the first scenario, which also includes the detection of phishing websites based only on the information available in TLS certificates, which we perform in the next chapter. Here, we are mainly interested

in whether the features in a certificate apart from CN and SANs are likely to give an advantage in classification performance. In the second research question, on the other hand, we evaluate whether fields that can be chosen by the certificate subject (e.g., `O`, `OU`, ...) apart from CN and SANs are abused by attackers to impersonate their target.

9.2.2. General Information in Phishing and Benign Certificates

To address the first research question, we look at the distribution of features for benign and phishing certificates in general and try to find out how well they separate benign and phishing websites.

As described in Section 2.6, some CAs offer different levels of validation. It stands to reason, that the more complex types of validation, i.e., organization and extended validation, make it harder for attackers to present a corresponding certificate. Still, we found that 1 444 certificates, about 15 % of all phishing certificates in our dataset, included an `Organization` in their subject fields. We used the subject `O` field to determine if a certificate is OV, assuming that CAs follow best practices and do not include unverified information in the certificates they issue. For benign certificates, 13 852 or about 35 % of websites had an `Organization` in their subject fields. We assume that this difference is particularly pronounced for the higher ranks in the Alexa list, as these companies, with high user counts, have more incentives to buy organization validation or extended validation certificates. Taking this distribution into account, organization validation is not a deciding factor for differentiating phishing and benign websites, and would lead to many false positives if it were to be used as such.

EV certificates on the other hand are a more interesting matter. To decide if a certificate is EV, we used the subject `business category` field (OID: 2.5.4.15). This field is a requirement for EV certificates [For18], checking for it is therefore an over approximation if we assume compliance to the requirements. We found, that this approximation works quite well for otherwise valid certificates, as we did not find any false classifications, even after working with and randomly sampling our dataset. Using this method, we identified only 39 phishing websites, that is about 0.4 %, with a valid EV certificate. These consist of compromised servers, as well as websites that were abused to host malicious content (e.g., `dropbox.com`, `jsfiddle.net`, `medium.com`), but also include several false positives (e.g., `paypal-notice.com`, a website used by PayPal to inform about updates and API changes). As such, it seems that extended validation was less likely to be available to phishing websites, even though possible (social engineering) attacks to obtain malicious EV certificates have previously been demonstrated (e.g., [Jac+07]). Still, it is likely more complicated to correctly fake an existing organization, including business registration details, as required for extended validation. On the other hand, only about 7 % (2 746) of the benign websites used an extended validation certificate. Even among the top ten ranks, none protected their homepage with an extended validation certificate. This shows, that even though an EV certificate (if it is valid and has the correct organization displayed) can be a good indicator that a website is legitimate, it does not provide a robust method to detect phishing websites. We further found, that some OV and EV certificates were used for phishing in connection with services that allow users to host content on their platforms. This includes Tumblr, Dropbox, Heroku

Table 9.2.: Percentages of benign and phishing certificates issued by the ten most popular issuers of phishing certificates

Issuer `CN`	Phishing	Benign
Let's Encrypt Authority X3	34.4 %	17.4 %
cPanel, Inc. Certification Authority	22.2 %	1.6 %
RapidSSL TLS RSA CA G1	9.1 %	0.2 %
COMODO RSA Domain Validation Secure Server CA	5.3 %	10.2 %
COMODO ECC Domain Validation Secure Server CA 2	5.2 %	18.2 %
CloudFlare Inc ECC CA-2	5.0 %	6.5 %
DigiCert SHA2 Secure Server CA	3.4 %	4.4 %
Go Daddy Secure Certificate Authority - G2	2.9 %	4.4 %
Google Internet Authority G3	2.0 %	0.5 %
RapidSSL RSA CA 2018	1.4 %	2.6 %

and Medium. The interesting part of this phenomenon is, that these organizations have at least organization validated certificates. As such, a user that expects to be on `bankingsite.com` might open the certificate, look at the `Organization` and realize they are in fact on `somehostingsite.com`, which might awake suspicion.

The most common issuers for benign and phishing websites are shown in Table 9.2. Again, we did not find any distinct features for phishing: the 10 most popular issuers, making up for 8 598($\approx 90.7\,\%$) of all phishing certificates, were also popular among benign websites (26 046 certificates $\approx 66\,\%$). As such, issuer information alone is not enough to separate benign from phishing domains. More detailed numbers for common issuers for benign and phishing websites can be found in Tables A.16 and A.17 in Appendix A.6.

Similar to the issuers, we also found only slight differences in other certificate details. The validity period for benign websites was on average longer than that of phishing websites (about 252 days for phishing and about 412 days for benign websites). We assume this is mainly due to the distribution of issuers: phishing websites more often used issuers with short validity periods like *Let's Encrypt* (90 days on average for both phishing and benign) and *cPanel*(average validity period of ≈ 93 days for phishing, ≈ 98 days for benign).

All in all, we did not find simple indicators for whether a certificate originates from a benign website or a phishing website. Attackers that set up their own websites have restrictions similar to benign administrators, resulting in similar choices for issuers and in similar certificates. Even though we found that phishing certificates often did not include an organization in the respective field, we found that this is also the case for many benign websites, even popular ones. The similarity in certificates was even more prominent if a benign website was used to host an attacker's content, which can be the case for compromised websites or when using hosting services.

9.2.3. Popular Target Websites

Next, we attempt to answer research question **RQ-2**, i.e., the question whether the certificates of phishing websites differ significantly when comparing them to their target's certificate. For this, we looked at the 15 most popular target websites of

Table 9.3.: Certificate and URL similarities for popular phishing targets. False positives we found were removed. Entries marked with an asterisk are hosted on the target's own infrastructure.

Target name	Target RD	Number of phishing websites	Similar Organization	Same Issuer	Similar Issuer	Target in URL FQDN	Target matches wildcard
PayPal	paypal.com	1169	0	1	24	84	12
Facebook	facebook.com	571	0	4	221	32	31
Microsoft	live.com	297	47*	0	58	10	9
ABSA Bank	absa.co.za	214	0	0	0	5	0
RuneScape	runescape.com	87	0	0	1	74	0
eBay	ebay.com	67	0	1	0	5	0
MyEtherWallet	myetherwallet.com	62	0	1	2	15	0
Blockchain	blockchain.com	46	0	1	1	0	0
Allegro	allegro.pl	44	0	0	0	35	0
Apple	apple.com	42	0	0	2	8	3
Steam	steampowered.com	39	0	0	0	6	0
Dropbox	dropbox.com	37	1*	1*	0	2	1
Binance	binance.com	34	0	0	0	3	1
Google	google.com	33	1*[a]	1*	0	1	0
ASB Bank Limited	asb.co.nz	29	0	0	0	4	0

[a]No text input, refers to different website

phishing attacks, as determined by the target labels provided by the PhishTank community. These 15 targets correspond to 2,771 (84.61%) of 3,275 valid certificates of phishing websites with a target label in our database. To extract the relevant information from the certificates of these 15 targets, we manually visited the login pages of the targets and downloaded the certificates presented at their login pages. We then tried to find out if and how well the phishing attacks were able to mimic their targets' certificates by comparing the 15 target certificates with the 2,771 certificates of phishing websites imitating these targets. The complete results can be found in Table 9.3. Note, that all entries greater than one indicate unique domain names, that might still host several phishing websites on different URL paths.

First, we looked at target organizations, and whether the phishing certificates included an organization field similar to the one included in the original certificate (see *Similar Organization* in Table 9.3). To determine organization similarity, we used the Python `difflib.SequenceMatcher` module[5], which uses a variant of the gestalt pattern matching algorithm by Ratcliff and Obershelp to compute the similarity between two text strings. We manually verified all matches with a ratio of more than 0.3, which roughly corresponds to one third of the compared strings being identical. This ratio was chosen through experimentation, as lower ratios resulted in large numbers of false positives, i.e. matching strings that are not visually similar to each other. We found no evidence of any phishing website obtaining a certificate with the same or a visually similar organization name as the one in the target's certificate (even beyond the targets listed in Table 9.3). All entries in the table with a similar organization were hosted on the target's own infrastructure. For example, Microsoft offers several cloud services (e.g., Azure, SharePoint and OneDrive), that allow users to host content on domains owned by Microsoft. These domains are protected by Microsoft's own certificates and therefore match the target's `Organization`. We

[5]`https://docs.python.org/3/library/difflib.html` online, accessed 2023-01-25

discuss this type of attack in more detail later on in this section.

Next, we looked at issuers used in the phishing certificates and compared them to the issuer used in the corresponding certificate of the target's login page. The column *similar issuer* lists the number of phishing websites that use the same CA organization as issuer as the corresponding target's certificate (e.g., *DigiCert High Assurance* is similar to *DigiCert Extended Validation*). We found, that many popular targets had few or no exact matches for the issuing CAs of phishing websites. Disregarding false positives and misclassifications again, only seven targets' issuers were replicated by phishing websites, and these cases were very rare (only one case for six targets, four for Facebook). Still, issuers seem to be a less precise metric than organizations as described above. This is also supported by the fact that there were many phishing websites with a similar issuer. It is also notable that among the 15 most popular targets we analyzed in detail, 9 were using EV certificates for their login pages. These require a thorough investigation of the entity requesting the certificate (see Section 2.6.1), making it less likely that organization information is spoofed.

As with organizations before, looking at the details for similar and identical issuers reveals an interesting finding: Most of these entries came from websites that host user content, protecting it with their own certificate. In many such cases, users might still be able to recognize that they are not on the website they expect if they look at organization information. However, this is not the case if an attacker targets the service they are hosting their website on. We found this to be the case for Microsoft, as well as Google and Dropbox. To prevent such attacks, user content could be protected with a different certificate from the one used to login.

Lastly, we looked at URL similarities. We labeled a URL used by a phishing website as similar to the URL of its target if it embeds the organization or original domain name (i.e., it embeds the target's exact e2LD or service name if it differs). We found that the number of websites with similar URLs was far higher than either organizations or issuers for most targets. As shown in the preceding chapters, complex phishing URLs can be difficult to detect even for users that were previously educated on the subject. Interestingly, we find that attackers seem to be able to add the target name to the domain name in many cases (see Table 9.3). Therefore, even though many browsers offer a reduction in complexity by only showing the domain name when looking at certificates, this part can still lead to users mistaking a phishing site for a benign site. As an aside, our dataset does not include a single valid certificate for a URL where the host is an IP address, making this type of URL obfuscation less relevant when evaluating certificates.

9.3. Reproducing the Results

Since the analysis of certificates above makes use of a relatively small dataset collected in late 2018 and early 2019, this section extends the analysis by including phishing URLs and certificates from the dataset described in Chapter 4. In particular, we focus on the `O` and `OU` fields of phishing certificates, as we previously found this information to be the most reliable indicator of benign certificates (i.e., general differences between certificates from the same issuer are small, and the issuer of certificates was less reliable than information about the organization).

Here, we focus on the 9,618 certificates of impostor domains, which were collected

by downloading the certificates from the domain names included in the **DS-Impostor** dataset described in Chapter 4. Focusing only on impostor domains greatly reduces the amount of certificates, thus enabling the use of manual analysis techniques, while also focusing on the certificates of domains that were already found to impersonate a target. We argue that it is unlikely, that attackers went through the effort of including spoofed information in the certificate `O` field (see Section 2.6), which is supposed to be checked in more rigorous processes, but not create an impostor domain name, where no or fewer checks are performed.

We therefore extracted all `O` and `OU` fields (see Section 2.6) from the certificates of impostor domains and analyzed their contents manually. To this end, we grouped certificates by their organization (or `OU` respectively) and reviewed all entries manually, comparing them to the target name or domain where necessary. First, it is notable that the most common extracted entries were information about hosting providers in the `O` field, and information about the validation level (*Domain Control Validated*) in the `OU` field. In this, we confirm the previous results of the smaller dataset.

There were, however, some organizations included in certificates that did not belong to hosting providers, but were similar to potential targets. While we where not able to verify the legitimacy of all websites, as some were no longer available or might have already been cleaned of malicious content, the remaining entries seemed to be either compromised benign domains or false positives in the dataset, as they corresponded to legitimate websites on closer investigation. We did not encounter any manipulation techniques that were applied to the organization, e.g., variants of combo- or typosquatting. We note, however, that a number of domains with certificates of hosting providers (e.g., Microsoft) were used by phishing domains targeting that provider. Of particular note in these cases is the fact, that it can sometimes be complicated to determine the valid RDs of these providers, thus making the verification of the phishing domains more complicated.

In all, we found no strong indications of attacker including malicious information in the `O` or `OU` fields of certificates in the dataset of impostor domains collected over a longer period of time. While some certificates were ambiguous, in that we could not determine their legitimacy, we were also able to use the information in certificates to verify if a website does in fact belong to the benign target in several cases. Abusing existing infrastructure to create phishing websites targeting the hosting provider, however, seems to still be a possibility and a relevant problem.

9.4. Discussion

In this chapter, we compared the certificates of phishing websites to those of popular benign websites. While we did not find immediate general differences between the two classes, we also did not encounter phishing websites that purposely include misleading information in their certificates either. In the following, we discuss the limitations of this research, followed by the most important implications.

9.4.1. Limitations

For the sake of completeness, we include some considerations that might have influenced our collection process or the implications of our results. Firstly, the collection of certificates was performed from Germany, which might have influenced

the results. This is more likely the case for popular websites that use content distribution and serve different content to users from different countries. Both phishing and benign websites were likely influenced by this, as attackers can use larger services to host their websites (see Section 9.2.3). This bias, however, is hard to remove and still represents a large amount of users that would have been served similar results.

We furthermore based our benign certificate dataset on popular websites extracted from the Alexa list. This is very likely to have led to a bias, as popular website are not necessarily representative of all legitimate websites. The selection of popular websites mainly concerns the results of the general differences between certificates of **RQ-1**, which we evaluate in more details in the next chapter. We argue, however, that the distribution and configuration of certificates of less popular websites is actually more likely to resemble that of phishing websites due to similar limitations and choices in infrastructure, which we confirmed in a smaller comparison of the certificates of less popular websites. It is therefore likely, that the results for the first research questions can be transferred to less popular websites.

Note, that the first part of this study was performed in 2019, after which several changes were implemented in popular browsers, that might have influenced the certificate ecosystem in general. For example, indicators of EV certificates were removed from most browsers, making this type of validation less relevant (see Section 2.6.2). The large amount of domains using free certificates, on the other hand, is unlikely to have changed, as the CA Let's Encrypt has remained relevant.

When attempting to reproduce the previous results in Section 9.3, we focused only on the certificates of impostor domains, and the `O` and `OU` fields. We argue, that this decision is unlikely to have significantly restricted the analysis, as it is improbable that attackers create a fake organization that can be verified for the certificate, but do not create an impostor domain as well. Furthermore, the decision to focus on the specified fields is motivated by the previous results, where these fields were the most robust in determining the validity of a website. Still, it is possible that the inclusion of purposely misleading information in certificates is more common and was not part of the dataset or not detected by our analysis process.

Finally, it is possible that attackers noticed the crawling efforts and added our client to a blocklist at some point (see, e.g., [Oes+18]). In this case, we would no longer be able to capture some of the attackers' methods, which might include more sophisticated techniques. We currently do not have any indications of this being the case.

9.4.2. Implications

In our analysis, we found that, unsurprisingly, there are no straightforward features extractable from certificates that instantly separate certificates of phishing and benign websites. While it is possible, that a combination of features can be learned by a heuristic or classifier, it is unlikely that this process will be practicable for human users, leading us to answer **RQ-1**, introduced in Section 9.2.1, in the negative, in that there are no general differences between the certificates of phishing and benign websites. On the other hand, we found that the analyzed phishing websites did not seem to recreate the information included in the certificates of popular targets. So, as for **RQ-2**, we found that there were often differences between the certificates of

a phishing website and the certificate of its target. We particularly found that the subject `O` and issuer `CN` seemed to not be actively replicated.

However, it remains an open research question, whether it would be possible to expose the differences we observed to users in a way such that it would help them to detect phishing websites. Here, a previous analysis found, that it typically takes several clicks to access certificate information in popular browsers [3], making it unlikely that users will incorporate the process into their daily browsing. In addition, the lack of organization validation even among benign websites, a fact that might become even more prominent in the future due to the lack of browsers which display this information more prominently than just in the corresponding certificate field, makes the usage of certificates in classification decisions less applicable overall.

Furthermore, while some information was not replicated, it is an open question how robust these findings are, i.e., how difficult it would be for an attacker to replicate the information of a target's certificate in their own. Since organization validated certificates do not require the same level of vetting that EV certificates do, it is possible that attackers might get a fraudulent OV certificate without the risk of compromising their operations. It is also possible that a CA is compromised or misses a spoofed or fake organization in a certificate request. While certificate transparency logs are supposed to prevent this kind of misuse, it is possible that certificates are still issued and not detected by monitors, which focus mainly on detecting instances of wrongly issued certificates. Replicating the issuer of a website is generally less complicated, as it does not require the attacker to spoof any information. As such, we conclude that it is not a robust feature to consider when analyzing a website.

In our analysis of popular target websites, we found that phishing websites with certificates that were similar to their target's certificate were often using hosting services and were not self-hosted. If the user content on such hosting services is protected by, for example, a wildcard certificate that includes information about the hosting service, it might still be possible to recognize this type of attack by looking at the certificate. However, this is not the case if the service provider itself is the target, or if a benign workflow is abused to host malicious content. These types of attacks therefore require completely different detection approaches, as there is nothing inherently malicious to the websites and their certificates.

In all, the results of this chapter motivate the usage of certificates in phishing classification in general, due to their high availability, and an apparent absence of malicious manipulation of some certificate fields. However, the information in certificates is not sufficient to prevent all types of attacks, as is is often not sufficient (e.g., when using path posing, or when comparing a legitimate website with a DV certificate), or the scenario impossible to detect using certificate information (e.g., when a legitimate domain is misused in the attack). We still take a closer look at certificates as an early warning system against malicious websites in the next chapter.

Chapter 10

Automated Phishing Detection Using CT Log Analysis

The previous chapter motivated certificates as a means of automated and educational phishing detection in two different scenarios (see Section 9.2.1). In this chapter, we focus on the first scenario: automated classification of phishing certificates using only information from certificates. While the certificate does not contain all the information that can be extracted from, e.g., a URL, using certificates also has several advantages. One advantage is, that Certificate Transparency (CT) logs can contain certificates before the phishing attack is executed, thus potentially enabling the detection of phishing attacks before they reach the user, already in the second step in the kill chain introduced in Section 2.1.1. Furthermore, we found that users still struggle with RD manipulation URLs, even after education (see Chapter 6) or when using RDN notation for URLs (see Chapter 7). Certificates usually contain information on the RD of the website they belong to, thus potentially enabling the automated detection of phishing URLs with RD manipulations, adding yet another layer to the phishing defense process. In the following, we first motivate and describe the setting of CT log detection in more detail, followed by an introduction to a pipeline that was developed to enable training and evaluation on CT log data. Finally, we present the results of an evaluation of several classifiers, examining the differences in training data, as well as the effect of including certificate information apart from the CN and SANs on classification performance.

Parts of this chapter were previously published in [1].

Contributions: The main contributions of this chapter are the design and implementation of a pipeline for classification tasks on CT logs, as well as two evaluations of automated phishing detection classifiers on real-world CT log data. The evaluations compare newly proposed classifiers with classifiers from previous work, and demonstrate the effect of training data cleaning, class imbalance, and the inclusion of different features from certificates on the classification performance. The CT-log classification pipeline and its first evaluation presented in Sections 10.2 and 10.4 were created and performed in collaboration with Arthur Drichel and Justus von Brandt and previously published in [1]. The second evaluation presented in Section 10.5 is a new contribution of this thesis and currently unpublished. Here, the dataset cleaning method of time-based filtering was previously proposed and implemented in a joint

work with Moritz Thiele and Arthur Drichel and presented in Moritz Thiele's master thesis [Thi22].

10.1. Setting

Recently, a vast majority of more than 80% of phishing websites have started to present valid https certificates to their users [APW21]. Even though this trend leads to more legitimate-looking websites, it also opens an opportunity for researchers to detect phishing websites earlier in the process of a typical attack. This is due to the fact, that when major browsers check a certificate's validity, they also require it to be present in CT logs [LLK13; Sch+18a]. These public logs therefore offer a view of all certificates that are intended to be used generally by users, which includes the certificates of phishing websites. Certificates need to appear in at least two logs to be trusted by current major browsers[1] and are often submitted directly by the issuer of certificates when they are created[2]. Logs are operated by several entities, including Google, Cloudflare and DigiCert, and can be monitored by anyone to make sure the log operators are working as expected. Consequently, by monitoring these CT logs, it is possible to detect phishing websites while they are still being prepared by attackers. This early detection can help shorten the gap between the start of a phishing attack and its inclusion in commonly used blocklists. In the following, we are therefore interested in the first usage scenario of certificate information of the previous chapter, where a certificate is presented to an automated classifier, which has to decide whether the certificate belongs to a benign or phishing website.

We therefore envision a setting where a central entity, for example a research institute, company, or even government, continually monitors and classifies all public CT logs. The classification results can then be used to maintain entries in a global blocklist, block access to potential phishing websites from local networks, or monitor ongoing attacks and campaigns before they are executed.

There are, however, several problems with this detection approach. First, there is a large amount of certificates published in CT logs (in the order of several million certificates per day, see, e.g., [Li+19]), making low numbers of false positives in potential classifiers an important requirement. Next, for the vast majority of certificates in the CT logs, there is no ground truth available in the CT logs, since there is no complete set of benign or malicious certificates that could be used to label the logs. This makes it harder to train or test classifiers on the actual logs, as datasets are likely to be noisy and questions arise about the validity of performance measurements.

Finally, we already found in previous chapters, that attackers often make use of compromised or benign infrastructure to host their attacks. This has two detrimental effects on the classification process: for one it is unlikely, that a classifier would be able to detect certificates of websites that are about to be compromised, as they were created with no malicious intent. Similarly, benign hosting infrastructure that is used by attackers is often protected by a benign certificate, which should not be classified as phishing. Second, the inclusion of potentially benign domain names in

[1] e.g., `https://support.apple.com/en-us/HT205280` and `https://chromium.github.io/ct-policy/ct_policy.html` online, accessed 2021-01-06

[2] `https://letsencrypt.org/docs/ct-logs/` online, accessed 2021-01-06

the phishing datasets makes it more complicated to compile a suitable ground-truth when training classifiers for the detection task, as we have to filter out certificates that were originally requested for benign websites.

In this chapter, we present a modular pipeline that can be utilized to perform classifier analysis on CT logs. The pipeline offers functionality from dataset creation and training of classifiers up to real-world evaluation with variable verification sources to provide ground truth labeling. The pipeline therefore offers a first step towards the evaluation of certificate classifiers on real CT log data, including the real-time detection of the certificates of phishing websites.

We use this pipeline in a comparison of several classifiers, and show that training data cleaning can have a significant effect on classifier performance, in particular in settings that require low false positives. We further provide indications that the domain names included in the certificate are the most important features for the classification task, thus somewhat confirming the findings of the previous chapter that certificate information apart from subject information is very similar for benign and phishing certificates.

10.2. CT-Log Classification Pipeline

In this section, we present the detection pipeline which was designed to enable comparatively evaluating phishing certificate detection classifiers. The main target group for the pipeline are researchers, who can profit from an accelerated development process for new detection methods and from the possibility to assess and compare different classifiers in a unified setting. The pipeline is also able to perform retrospective analyses in addition to live classifications of certificates published in the CT logs. All complex tasks, beginning from data acquisition (collecting certificates from CT logs as well as by crawling from well-known phishing websites), over data pre-processing (filtering and sanitizing), data labeling (as benign or phishing), classifier training, the classification itself, evaluating the classifiers' performance, up to the preparation of the final results are covered by this approach.

In the following, we give a short overview over the main modules and tasks handled by the pipeline. The source code is written in Python and publicly available[3].

10.2.1. Pipeline Overview

The classification pipeline consists of three main modules, focusing on dataset creation, classifier training, and classifier evaluation, as well as a fourth intelligence module that can be used to further process evaluation results (see Figure 10.1 for an abstract illustration).

Database Module

The database module was created to support the collection, normalization, labeling, and storing of data obtained from various data sources in a unified database. This provides the basis for the training and evaluation of different types of machine learning models, as well as the validation of evaluation results in both retrospective and live classifications. We visualize this module in the upper part of Figure 10.1.

[3] `https://gitlab.com/rwth-itsec/ctl-pipeline` online, accessed 2023-01-24

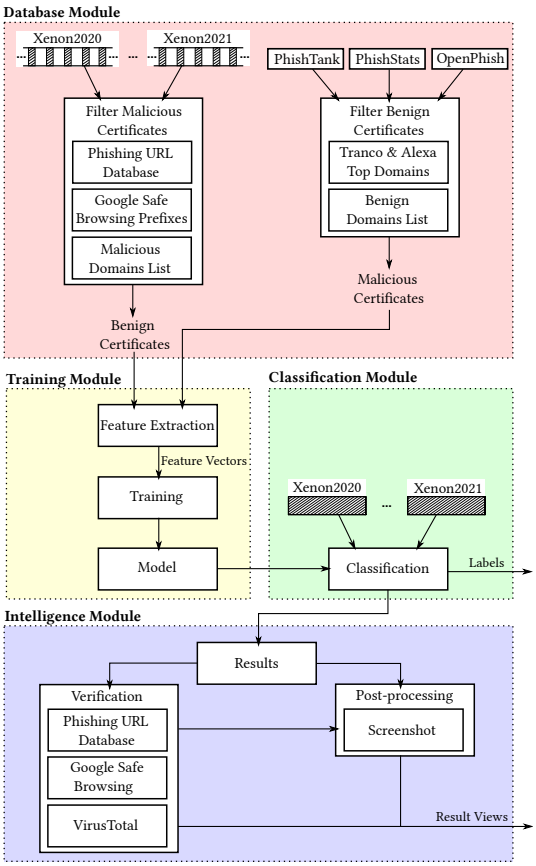


Figure 10.1.: Abstract illustration of the architecture and pipeline operation.

The outcome of the database module, in consideration to the pipeline flow, is a labeled dataset that can be used for classifier training.

Data for the labeled dataset can be collected from various sources. For benign data, certificates can be downloaded from the CT-logs in chunks (represented by hatched boxes in the figure) to retrieve a more representative sample that contains certificates published at different times of the day and different days of the week including workdays and weekends. Since several major browsers require that a certificate has to be published in at least two different logs in order to be displayed without a warning, the download process might yield duplicate certificates. The database module thus includes a data sanitization process which removes duplicates. Additionally, certificates can be filtered against a phishing URL database that includes known phishing URLs of the threat intelligence feeds of PhishTank, PhishStats, and OpenPhish, as well as Google Safe Browsing (GSB). Lastly, the pipeline permits the filtering of obtained certificates against a malicious domains list that is adjustable in order to enable the filtering of further known malicious certificates that are not (yet) included in the observed threat intelligence feeds. We label the set of certificates left in our collection after filtering against malicious domains as benign certificates and use them as benign data for the training of supervised machine learning classifiers.

For malicious labeled data, the certificates of phishing URLs that are contained in a phishing URL database are downloaded using the process explained in Chapter 9 using OpenSSL. As before, we did not follow redirects, as this approach is less likely

to be affected by website cloaking. To remove known benign certificates from the set of malicious samples, a filter list of benign domains can be provided (e.g., the Tranco list, or a curated list which contains common URL shorteners as well as common web hosting services).

As in the previous chapter, the benign training data is generated only once, while the gathering of threat intelligence (e.g., for the phishing URL database) and the download of phishing certificates is a continuous process. The database module can also be extended by incorporating further filtering or by including additional threat intelligence feeds. The generated labeled training data serves as input to the training module which is described in the following.

Training Module

The training module is responsible for providing a trained model that is ready for live classification or retrospective analysis. The module supports training classifiers using supervised machine learning and the labeled datasets from the database module, but also enables the integration of unsupervised models or rule-based classifiers. Further, it is possible to train feature-based (e.g., random forests (RFs)) as well as feature-less (e.g., recurrent (RNNs) or convolutional neural networks (CNNs)) models.

The result of the training module is a classifier which is ready to be used in the following steps of the pipeline. To this end, the classifiers can be serialized and stored in order to enable result reproduction and the sharing of trained classifiers.

Classification Module

The classification module uses the trained models provided by the training module for classifying the certificates published in CT logs. Here, it is possible to manually select a time span that should be taken into account for retrospective analysis, or whether to perform live classification. Similar to the generation of the benign labeled training data in the database module, the respective CT logs that should be investigated can be selected freely. The output of the classification module are the predicted labels of the certificates published in the CT logs, either as aggregated results for the retrospective analysis or as real-time output during live classification. These classification results can then be used by the last module to create comparisons of the classifiers or perform further post-processing steps. Note, that our implementation is highly parallelized and all classification methods can be scaled with either more GPUs or with a greater number of CPUs (or CPU cores in general).

Intelligence Module

The final module, called intelligence module, receives the results passed by the classification module, attempts to validate them, optionally performs post-processing, and processes the results into informative result views.

One of the main tasks of this module is the verification of the results obtained from the classification module. Since the goal of the pipeline is to try and detect phishing websites even before they are activated, the classification results can typically not immediately be verified due to missing ground truth (i.e. when a phishing certificate published in the CT logs in real-time is detected, it is unlikely that the corresponding phishing URL is already included in any threat intelligence feed). For this reason,

the pipeline enables a retrospective verification in order to be able to inspect the classification performance of a given classifier. As such, the live classification results can be stored and later validated when new phishing URLs are added to the threat intelligence feeds. Verification itself is performed via the usage of an offline phishing URL database, the GSB service, and VirusTotal[4].

The intelligence module also provides an interface which allows for easy post-processing of classification results. The current version supports taking screenshots of potential phishing websites, and can be extended to, e.g., collect additional information about the domain, add a second classification step, or periodically check whether the domain is included in blocklists or taken down. Finally, the intelligence module calculates several metrics based on the results obtained from the classification module and presents them in aggregated result views.

10.3. Certificate Classifiers

In this section, we introduce the new and existing classifiers that were used in the evaluations performed in this chapter. Apart from feature-based RF classifiers, we analyze Deep Learning (DL) classifiers, as well as two state-of-the-art approaches for domain name-based phishing certificate detection.

10.3.1. Classifying Domains

Since certificates can contain several domain names in their `SAN` field in addition to the `CN`, we generally distinguish the classification of certificates and domain names. Furthermore, as the majority of the features used for classification are extracted per domain name, we have to combine these separate predictions into a unified classification score for a given certificate. Therefore, features are extracted once for the certificate fields (*certificate features*), and then again for each domain name in the certificate (*domain features*) (see Figure 10.2). We then create a new feature vector for each domain name that combines both certificate and domain features, and use it for classifier training and predictions. The results per domain can then be combined using several different meta classifiers to achieve a combined score for the classifier. In the evaluations presented in the following, we use the simple aggregation functions minimum, maximum, average, or median as meta classifiers. We denote the combination of domain classifiers with a maximum, minimum, average, and median meta classifier by appending *-max*, *-min*, *-avg*, and *-med*, respectively, to the actual classification method where relevant.

10.3.2. Feature Engineering and Selection

To create a suitable feature set, we analyzed existing features from related work on the one hand (i.e., [Bah+17; Don+15; FEP19; Sch+18b; TCB18]) and developed novel features based on our analysis of benign and phishing certificates on the other hand. In our evaluations, we only use context-less features, i.e. features that can be extracted from a single certificate, and omit features that are not relevant to our use-case (e.g., validity status of certificates, as CT logs only contain valid certificates). We furthermore did not include any features extracted from other sources (such

[4] `https://www.virustotal.com/` online, accessed 2021-01-06

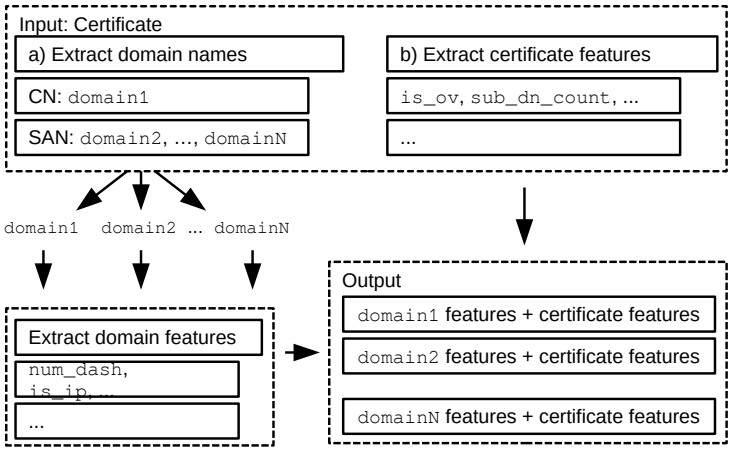


Figure 10.2.: Feature extraction process, resulting in domain and certificate features.

as from WHOIS) in order to be independent of third party services and to ensure real-time classification capabilities.

Overall, the features we investigate can be split into three categories: (1) certificate features, (2) domain features and (3) keyword features. The first category contains features that are extracted from the certificate itself ignoring included domain names. Instead, domain name specific features are included in the second category, except for the occurrence of specific keywords in the domain name, which are included in the third category. We distinguish these feature sets as the list of keywords needs to be updated and since keywords are more context specific compared to the more general domain features. Suspicious keywords were extracted from URLs collected from PhishTank by analyzing the labels in the domain name of URLs after removing the eTLD (see [1] for more details on the process).

In total, we extract 126 features: 22 certificate features, 55 domain features, and 49 keyword features. The full list of keyword features, as well as all certificate and domain features can be found in Tables A.18, A.19, and A.20 in Appendix A.7.

After feature engineering, we define two different feature sets. The first feature set contains *all* engineered features from all three categories and serves as baseline. The second feature set is created by performing feature selection on the features of the first two categories to ensure that only features that are actually relevant to the classification process are included. Additionally, reducing the number of features also reduces the required time for feature extraction and can improve a classifier by making it more robust to noise. We are limiting this feature set to a subset of features from the first two categories, thereby removing all keyword-based features. We argue, that this makes it more generally applicable, as it does not contain suspicious keywords, which are biased in favor of specific targets and languages. For comparison, we evaluate classifiers using both feature sets, i.e. classifiers that make use of all 126 engineered features including keyword-based features and classifiers that only use a subset of domain- and certificate-features.

Feature selection was performed by training an RF-based classifier, ranking its features by their importance according to the mean decrease in impurity (MDI) [Lou+13], and excluding features that were either not important or had high variances. This results in a total of 50 features which belong to the first two feature categories. The

selected features are marked in Tables A.19 and A.20 in Appendix A.7. We denote which feature set was used for a given classifier with an index as either *all* or *sel* for selected features.

10.3.3. Random Forest-based Classifiers

The first batch of classifiers we investigate in either of the two evaluations are random forest (RF) classifiers. In the past, it has been shown that RF classifiers are well suited for the phishing website and URL classification problems (e.g., [Bah+17; Don+15; Sah+19; Sub+17]). In our evaluations, we use both feature sets (*all* and *selected*), as well as all four meta classifiers with RF classifiers. For all RF-based classifier we use the default hyperparameters (set by scikit-learn[5]) but increase the number of estimators to 200 as this led to improvements in classification accuracy in preliminary experiments.

10.3.4. Deep Learning-based Classifiers

To compare the feature-based approaches above to DL models, we create and compare three classifiers based on neural networks (NNs). While feature engineering and selection are not necessary for NNs, information that is relevant for classification has to be encoded and provided to the classifier. Here, it is unlikely that encoding the complete certificate and providing it to the classifier makes sense, as information like the public key, signature, or serial number are more likely to create unwanted biases in the classifier than to be helpful in the classification task. We thus utilize all engineered features and provide this information to the model directly, and additionally provide the raw domain names using characterwise integer encoding. By choosing this approach, the NN can (1) learn to extract relevant information from a domain name and (2) select the relevant information from all provided features.

For the first NN classifier we choose an architecture based on long short-term memory (LSTM) networks in which the domain name and the features are consumed separately. LSTMs have been shown to be suitable for URL classification tasks in several domains before (e.g., [Bah+17; TCB18; Woo+16]). We present the architecture of our LSTM-based classifier including the input and output dimensions in Figure 10.3. The input data is processed by distinct hidden layers and afterwards combined by concatenation for further processing. The final prediction is output by a fully connected layer. In detail, the LSTM-based approach uses one unidirectional LSTM layer for the domain name and one bidirectional LSTM layer to process the features. Before a domain name is fed into the model, we convert every included character to a unique integer and pad the result with zeros from the left side to the maximal domain length of 253 characters [Moc87] as proposed in [Dri+20]. This ensures that the model is able to process domain names at any length while using batch learning. The resulting encoded domain is processed by an embedding layer that adds additional information about the relationships between characters to the encoding. We choose an embedding dimension of 128 and thus project every character to a unique 128-dimensional vector.

The second NN classifier is based on a residual neural network (ResNet) architecture, which we use in the second evaluation. It uses several residual layers each of which

[5] `https://scikit-learn.org/stable/` online, accessed 2023-01-24

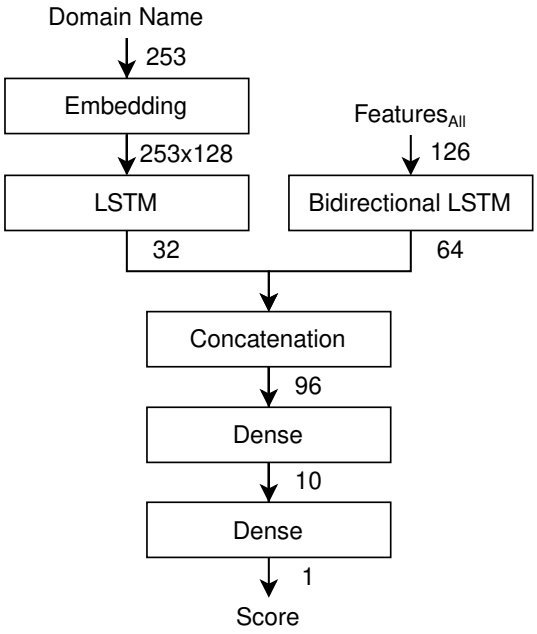


Figure 10.3.: Network architecture of the LSTM-based classifier with certificate and domain name features.

consists of two one-dimensional convolutional layers with an added skip connection. As with the LSTM, domain names and features are provided as input separately, and both inputs are first processed by three residual layers each. The different inputs are then concatenated, followed by two dense layers. The domain name embedding is identical to the LSTM network. We also create a second version of the ResNet architecture that is identical to the first one except that it does not include the certificate features as input and therefore only works on domain names (denoted $\texttt{ResNet}_{\texttt{san}}$). We use it to evaluate the effect of including certificate features in the second evaluation.

Finally, we evaluate a third classifier that combines convolutional neural network (CNN) and LSTM layers (denoted `CL`) in the second evaluation. It only takes domain names as input, which are first processed by an embedding layer with an output dimension of 32, but that is otherwise identical to the previous embedding. The result is then forwarded to three CNN layers on the one hand, which are concatenated followed by two dense layers, and a bi-directional LSTM layer on the other hand, which is followed by a second bi-directional LSTM layer and two dense layers. Both CNN and LSTM paths are then combined and again processed by two dense layers.

We optimized the NN architectures iteratively using only data from the given training set. Similar to the feature-based approach, they also make use of the meta classifiers based on simple aggregate functions to combine the predictions per domain name into a unified score for the certificate.

10.3.5. State-of-the-Art Domain Classifiers from Related Work

In addition to the classifiers we newly developed, we evaluate two state-of-the-art approaches for phishing certificate detection proposed in related work. We adapted

both approaches slightly to comply with the interfaces defined in the training module.

Template Classifier

The first classifier, by Sakurai et al. [Sak+20], uses only phishing domain names as input, and automatically creates regular expressions that match malicious domains following the same pattern. This classifier does not return a classification score between 0 and 1, it only returns information about expressions that match the given domain. We therefore modify the output of the template classifier to return the highest entropy reduction rate among all matching expressions (see [Sak+20]) as classification score. Since the classifier is designed to find malicious domains, not certificates, we return the highest matching domain's score when combining the classification scores for the individual domains to a classification score of the certificate as a whole, which corresponds to using the maximum meta classifier.

Phishing Catcher

Lastly, we evaluate the *Phishing Catcher*[6], a rule-based *certificate* classifier, where each rule potentially increases the classification score. Rules include, for example, inclusion of suspicious keywords in any of the domains, or the usage of suspicious top-level domains. The only non domain-specific feature indicates, whether the issuer of the certificate matches the popular free certificate authority *Let's Encrypt*. The Phishing Catcher is unique compared to the other classifiers in that it does not require any training, as its heuristics are purely based on pre-defined rules that were derived using domain knowledge. Each rule is associated with a numeric value, and the values of matching rules for a given domain name are summed up to result in a classification score for the domain, from which the maximum value is selected as overall classification score of the certificate. We modify this classifier slightly to return its classification score instead of printing messages for suspicious domains.

The classifiers were evaluated in two phases, where the goal of the first phase was to evaluate the functionality of the proposed pipeline in general and create a baseline for CT log classification, while the second phase extends on the baseline with an analysis of the effects of training data filtering and the inclusion of certificate features apart from the CN and SANs.

10.4. First Evaluation: Pipeline Functionality and Baseline

In the first evaluation, we test the basic pipeline functionality, and compare the different meta classifiers as well as the effect of feature selection. To this end, we first present the datasets that were used in the evaluation, followed by the evaluation method and results.

10.4.1. Datasets

In the following, we describe the training and evaluation datasets that were used to evaluate the pipeline functionality and baseline classifiers.

[6] `https://github.com/x0rz/phishing_catcher` online, accessed 2023-01-25

Malicious labeled training data

We used the database module (see Section 10.2.1) to generate malicious labeled training data. The used phishing URLs are a subset of the **DS-Phish** dataset presented in Section 4.1.1 which was created by only selecting URLs that appeared in one of the URL feeds up to but not including May 2020. The corresponding set of certificates crawled from these domains was further filtered to remove benign hosting infrastructure by comparing the domain names included in the certificates to a list of 177 known benign services, which was created based on an analysis of common domain names in the dataset and by using domain knowledge. Additionally, we removed all duplicates and filtered the CN as well as every SAN included in the malicious certificates using lists that include very popular domains. Certificates that contained at least one domain name that was determined to be benign were removed from the dataset. In detail, we filtered against the top 1,000 Alexa and top 1,000 Tranco domains, as it is unlikely that a phishing website is hosted on such a popular domain. After removing duplicates and filtering the set of certificates as described we ended up with 56,479 unique malicious certificates overall, which we utilize in classifier training.

Benign labeled training data

The benign labeled training data was obtained via the database module as well, and downloaded from the Google Xenon [Goo23] logs from April 2020. The downloaded certificates were filtered using the phishing URLs downloaded from threat intelligence feeds to remove known malicious domains. In all, we downloaded 70,889 unique certificates that we labeled as benign. From these, we randomly selected 56,479 certificates (same number as malicious) for the training process.

We combined the collected benign and malicious certificates into a balanced training set that includes 112,958 certificates in total. The effect of class imbalance is explored in more depth in the second evaluation.

Moreover, we experimented with different sources for obtaining benign data. For instance, we downloaded certificates from popular websites according to Cisco Umbrella[7]. However, classifiers trained on this data generally performed worse on a separate validation set containing actual CT log data. As such, we do not include them in our evaluation.

CT log test data

We chose the Google Xenon [Goo23] logs for our comparative evaluation. As these logs are scoped, i.e. certificates that are included in Xenon2020 have an expiration date in 2020, we analyzed all certificates published in the first week of May 2020 in Xenon2020, Xenon2021, Xenon2022, and Xenon2023. In total, these logs contain approximately 22.5 million unique certificates for the period under investigation. By selecting the first week of May as test data, we guarantee data chronology and disjoint training/testing data as would be the case in live-classification.

[7] `https://umbrella.cisco.com/blog/cisco-umbrella-1-million` online, accessed 2021-01-06

10.4.2. First Evaluation Overview

We trained classifiers using the training dataset and evaluated them on the real-world CT log test data. The evaluation, and subsequent validation, was performed in December 2020, to ensure enough time had passed for malicious websites to be added to malicious domain feeds. Note, that our evaluation approach is basically equivalent to performing the classifications in real time and verifying the results later, as the CT logs are append-only and therefore contain the same certificates in both settings.

In detail, we split a small portion from the training data for a validation set that is used either during the training in case of the DL based approaches, or after the training in case of RF-based approaches for model assessing purposes. We trained the DL based models until there were no further improvements on the validation set for at least three consecutive epochs.

The template classifiers use only domain names as input and automatically create regular expressions that match malicious domains following the same pattern. We therefore use only one domain name from each certificate for training, namely the one that matches most closely the original URL from the malicious URL feed. As the classifier distinguishes groups of domains by the number of dots present, we split our training set accordingly, and use up to 2,000 domains per group. We train on all possible domain names for groups for which less than 2,000 samples are available. For all other configuration options, we use the settings recommended in the original paper.

The Phishing Catcher classifier does not need any training at all since its classification relies on predefined rules.

We choose the false positive rate (FPR) and the true positive rate (TPR) as our evaluation metrics which are suitable measures especially for highly imbalanced data [DG06]. Since there is a far larger amount of benign certificates in the logs than phishing certificates, and the amount of certificates is large in general, we argue that a low FPR is the most important attribute of a suitable classifier. In addition, we observe the TPR which is a proxy for determining the amount of detected phishing certificates.

Note, the TPRs which we present only provide a lower bound on what our classifiers achieve, as the classifiers may be able to identify certificates that at the time of verification are not included in any of the Phishing URL feeds yet. We can thus only present a lower bound of the actual TPR because not every malicious certificate which is flagged positive by a classifier is verifiable via our verification process in the intelligence module. We discuss this effect in more detail in Section 10.6.

As an example for model validation, Figure 10.4 displays the receiver operating characteristic (ROC) curve obtained after training the domain $\mathrm{RF}_{\mathrm{all}}$ classifier (i.e. the RF-based domain classifier using all engineered features) for each of the four meta classifiers. The ROC curve enables us to select an operating point by balancing the trade-off between low FPR and high TPR. In this diagram, the x-axis is zoomed in to only include low FPRs, as we argue that a low number of FPs is the most important attribute of a classifier for our use-case. The approaches which utilize the minimum and average meta classifier are promising and achieve a TPR of over 10% at a very low FPR. The figure also displays a baseline which corresponds to random guessing as a dashed line at the bottom of the diagram, and which is significantly lower than all four approaches. While a TPR of around 10% seems to be rather low, we argue

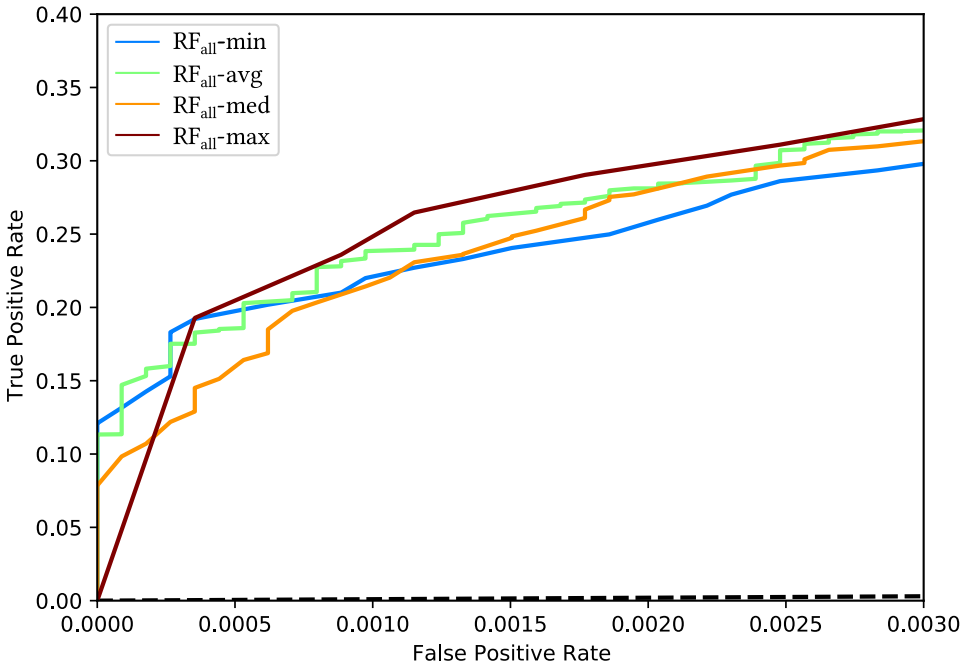


Figure 10.4.: Model validation results: ROC curve of the RF_{all} domain classifier using all four meta classifiers.

that a classifier set at such an operating point can detect several phishing certificates without generating too many false positives. Note, that these characteristics are only model validation results, and that the actual results of the comparative evaluation on real-world CT log data are presented in the next subsection.

10.4.3. Pipeline and Baseline Evaluation Results

We present the results of the comparative evaluation in Table 10.1. Here, we display the total amount of detected phishing certificates and the estimated TPRs at fixed FPRs of 10^{-3} and 10^{-4} which correspond to a total of 22,510 and 2,251 false positives, respectively, for classifying the Google Xenon logs for a full week.

At the fixed FPR of 10^{-3}, the rule-based classifier, Phishing Catcher, achieves by far the best results. At an FPR of 10^{-4}, the classifier of Sakurai et al. detects the most phishing certificates followed by Phishing Catcher. Our developed RF-based and LSTM-based approaches achieve worse results. We find that the proposed classifiers typically achieve better results using the minimum and average meta classifiers than by using the maximum or the medium meta classifiers. Furthermore, the RF-based approaches which make use of all features (including keyword-based features) perform generally better than the classifiers that utilize the selected feature set.

We reckon the worse results obtained by the RF-based and LSTM-based approaches compared to the classifier of Sakurai et al. and Phishing Catcher to be caused by noisy training data. Since we obtained malicious labeled samples by downloading the certificates from URLs included in the various threat intelligence feeds, we also obtained benign certificates that were used on phishing websites hosted on compromised server infrastructure. Although we filtered these certificates against our benign domains list, it is possible that a significant amount of benign certificates is

Table 10.1.: Evaluation results showing TPs at fixed FPRs in the first evaluation

Classifier	**FPR**=10^{-3}		**FPR**=10^{-4}	
	#TP	**TPR**	**#TP**	**TPR**
RF_{all}-min	272	0.00418	55	0.00086
RF_{all}-avg	243	0.00374	55	0.00086
RF_{all}-med	232	0.00357	40	0.00062
RF_{all}-max	244	0.00375	-	-
RF_{sel}-min	216	0.00332	39	0.00060
RF_{sel}-avg	208	0.00320	36	0.00055
RF_{sel}-med	189	0.00291	29	0.00045
RF_{sel}-max	148	0.00228	-	-
LSTM-min	352	0.00541	35	0.00054
LSTM-avg	366	0.00563	36	0.00056
LSTM-med	347	0.00533	34	0.00053
LSTM-max	206	0.00317	20	0.00032
Sakuarai et al.	242	0.00373	**117**	**0.00180**
Phishing Catcher	**1102**	**0.01692**	110	0.00170

falsely labeled as malicious as we only filtered against known redirecting and hosting services.

The fact that the RF_{all} classifiers achieve better results than the more general $RF_{selected}$ classifiers can be explained by the absence of keyword-based features within the selected feature set and by the fact that the selection was performed on the noisy training data.

With this evaluation, we were able to show that different types of machine learning classifiers can be used within the pipeline to detect phishing certificates. In the second evaluation, we therefore use the pipeline to answer several more specific research questions about the classification of certificates from CT logs.

10.5. Second Evaluation: Improving CT-Log Classifiers

In the first evaluation, we found that none of the proposed classifiers achieved satisfactory results, as the number of false positives outweighed those of true positives by at least an order of magnitude in all cases. In this section, we therefore evaluate several approaches to improve on these results with a focus on cleaning the training datasets by strategically selecting subsets of phishing websites. To this end, we first describe the datasets, followed by a description of newly trained classifiers and the results of the evaluation.

10.5.1. Datasets and Data Cleaning

For this evaluation, we extend the set of training certificates to include all valid certificates that were collected before March 2022, which results in 131,455 phishing certificates. Instead of using a manually curated list of known benign domain names, we remove all certificates that include a domain name that exactly matches a domain

Table 10.2.: Datasets used in the second evaluation

Shortname	Filtering	Phishing	Benign
n-02	Naive	10000	40000
n-05	Naive	129604	129604
t-02	Time (24h)	24938	99752
t-05	Time (24h)	24938	24938
t2-05	Time (48h)	46534	46534
i-01	Impostor	9618	86562
i-02	Impostor	9618	38472
i-05	Impostor	9618	9618

name which is present on the Tranco list in the first 100,000 entries in this evaluation. This results in a dataset of 129,604 certificates that is *naively* cleaned only of popular benign domain names, similar to the dataset in the first evaluation.

From this naive dataset, we then create two new datasets based on different filtering approaches: time-based filtering and impostor domain filtering (see Table 10.2 for an overview). The intuition of time-based filtering is, that due to the short lifespan of typical phishing attacks, it is likely that the websites are detected and added to a blocklist relatively shortly after their creation. If the website is hosted on compromised or benign hosting infrastructure, on the other hand, it is more likely that the domain and corresponding certificate have already existed before hosting the malicious content, resulting in time differences between requesting a certificate and the inclusion of the website in a blocklist between the two cases. For time-based filtering, we therefore include only certificates where the time difference between creation and blocklist inclusion is short, thus aiming to only include certificates of phishing websites that were created by the attacker and are therefore more likely to exhibit detectable malicious attributes. To this end, we determine the inclusion date of the domain names in our phishing dataset in any blocklist, and compare it to the `valid from` date of the corresponding certificate. Only when the time difference is lower than a given threshold is the certificate added to the time-filtered dataset. In this evaluation, we compare 24 and 48 hours as thresholds for the allowed time difference. In this way, we created two new sets of malicious certificates: one which only includes certificates with a domain name that was blocked in less than 24 hours, which results in 24,938 certificates, and one with a threshold of 48 hours, resulting in 46,534 certificates. Note, that this filtering method is only possible because a relation from certificate `valid from` date to an inclusion date in a blocklist exists, and therefore requires both certificate and blocklist information to be applied.

For impostor domain filtering, we simply select all certificates where at least one included domain name matches our dataset of impostor domains **DS-Impostor** created in Chapter 4, resulting in 9,618 certificates. The goal of this filtering is two-fold: for one, we argue that impostor domains are a good representation of the certificates we are actually interested in finding, as they include a reference to a target and are thus likely created with malicious intentions and at the same time more likely to mislead users. Secondly, it is also likely that the filtering can improve the classifier performance, as focusing on certificates that have a semantic similarity, and are

furthermore less likely to include domain names of compromised infrastructure, might make it easier for the classifiers to differentiate malicious from benign domain names and certificates. An additional advantage of using impostor domains as a filtering method is, that the domain names used in training do not require a relation to certificates, thus making it possible to train domain-only classifiers on all available phishing domain names, instead of only those with a corresponding TLS certificate. On the other hand, the low number of resulting certificates already implicates that impostor domains are rarer compared to the naive or time-filtering methods presented above.

Since both new filtering methods result in datasets of comparatively small size, we also evaluate the usage of imbalanced datasets during training by adding more benign certificates. In all, we compare eight datasets (see Table 10.2): naive filtering with ratios 0.2 (n-02) and 0.5 (n-05), time-based filtering of 24 hours with ratios 0.2 (t-02) and 0.5 (t-05), time-based filtering of 48 hours with ratio 0.5 (t2-05), and impostor filtering with ratios 0.1 (i-01), 0.2 (i-02) and 0.5 (i-05).

Benign data for this evaluation was obtained from the Google Xenon 2022 logs and includes certificates that were logged between January 1, 2022 and February 28, 2022. The resulting certificates were then filtered by removing certificates where at least one domain name matches a domain name from threat intelligence feeds that was observed before March 2022, but is not part of the top 100,000 Tranco domain names. In all, this results in 136,928 benign certificates, from which we randomly sample to create the desired ratios for the training datasets.

10.5.2. Classifiers

We begin the evaluation with simple RF classifiers, with a focus on the comparison of different training datasets. Based on the results of the first evaluation, we only include classifiers that include all features, instead of the subset of features from feature selection. We furthermore decided to only include classifiers with the `min` meta classifier, as it generally performed well in the first evaluation, and restricting the meta-classifier results in significantly fewer classifiers to compare. We already observe improved TPRs for the low FPRs required in the real-world evaluation in the training step, as can be seen on the example of a classifier trained on the unbalanced impostor domain dataset in Figure 10.5. As with the first evaluation, however, these improvements do not necessarily translate into significant performance improvements in the final evaluation.

We also included a new set of DL classifiers, to make use of the semantic information of the dataset filtering by learning domain features. Here, we used the two versions of the `ResNet` model described in Section 10.3.4: one with both domain and certificate features (denoted with `all`), and one that does not include any certificate features, and instead only works on domain names (denoted with `san`). In addition, we include the classifier based on combined CNN and LSTM inputs, architectures which were successful in previous work on phishing URL detection, denoted `CL`. This is a domain-only classifier as well, and therefore does not have any additional certificate features as input. Since we found the RF classifiers to perform better on unbalanced datasets (see Section 10.5.3), we only trained classifiers on the datasets: i-01, i-02, t-02, and n-02.

As in the previous study, we include the template classifier by Sakurai et al.

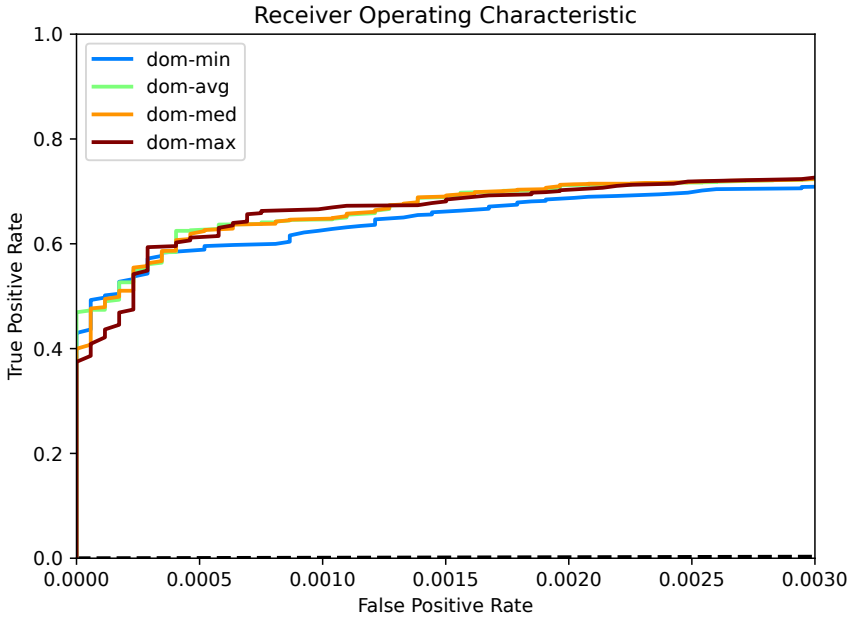


Figure 10.5.: ROC curve of a RF_{all} domain classifier trained on the i-01 dataset using all four meta classifiers.

(denoted `Template`), and the phishing catcher (denoted `Phishing Catcher`) as baselines. For this evaluation, the template classifier was trained either on all impostor domains (suffix `i`) or a random selection of 10,000 domain names from the naive dataset (suffix `n`).

10.5.3. Evaluation and Comparison Results

For this evaluation, we changed the general process compared to the first evaluation, by converting it into an offline evaluation. To this end, we downloaded 14,080,601 certificates from March 2022 from the Google Xenon logs into a local database to be used to evaluate the classifiers, and performed all operations on this offline dataset. This has the advantages of guaranteeing reproducible results and making ground-truth validation easier. For ground-truth validation, we used our dataset of phishing domain names presented in Chapter 4 to label malicious certificates, in addition to verification using GSB with online lookups performed in January 2023. In all, this labels 13,850 certificates and 29,121 domain names as phishing, which corresponds to approximately 0.098% of all certificates. From these validated phishing domain names, we further extract all impostor domains using the same process and targets as in Section 4.2. This results in only 782 domain names, and 468 certificates, with the impostor property. Since impostor domains are a major part of the second evaluation, we extend the classifier comparison by adding a column for each FPR threshold that shows, how many of the certificates that could be validated as TPs also contain at least one impostor domain in `SAN` or `CN` (denoted $\texttt{TP}_\texttt{i}$).

The evaluation was performed by first training the classifiers using the corresponding training datasets, and using the trained models to obtain predictions for the

Table 10.3.: Evaluation results showing TPs and TPs including impostor domains (TPi) for different FPR thresholds.

Classifier	**FPR**=10^{-3}		**FPR**=10^{-4}		**FPR**=10^{-5}		**FPR**=10^{-6}	
	#TP	**TPi**	**#TP**	**TPi**	**#TP**	**TPi**	**#TP**	**TPi**
RF-n-02$_{\text{all}}$	412.67	50.00	152.33	18.33	22.67	7.33	2.00	1.33
RF-n-05$_{\text{all}}$	460.00	44.33	122.00	14.33	0	0	0	0
RF-t-02$_{\text{all}}$	409.67	58.00	**162.33**	24.00	24.67	6.67	0	0
RF-t-05$_{\text{all}}$	314.00	45.00	103.33	16.67	0	0	0	0
RF-t2-05$_{\text{all}}$	400.00	53.33	130.00	16.67	0	0	0	0
RF-i-01$_{\text{all}}$	299.00	80.33	129.00	45.00	**31.67**	**15.67**	**5.33**	**3.67**
RF-i-02$_{\text{all}}$	285.00	74.33	127.33	42.33	25.33	13.67	2.67	1.67
RF-i-05$_{\text{all}}$	229.33	61.33	78.67	28.67	13.67	8.00	0	0
ResNet-n-02$_{\text{all}}$	398	59	0	0	0	0	0	0
ResNet-n-02$_{\text{san}}$	544	113	0	0	0	0	0	0
ResNet-t-02$_{\text{all}}$	**750**	130	0	0	0	0	0	0
ResNet-t-02$_{\text{san}}$	723	129	0	0	0	0	0	0
ResNet-i-01$_{\text{all}}$	363	216	0	0	0	0	0	0
ResNet-i-01$_{\text{san}}$	320	**220**	100	**90**	0	0	0	0
CL-i-01$_{\text{san}}$	518	192	114	84	0	0	0	0
ResNet-i-02$_{\text{all}}$	0	0	0	0	0	0	0	0
ResNet-i-02$_{\text{san}}$	315	201	00	0	0	0	0	0
Template-i	91	59	0	0	0	0	0	0
Template-n	57	20	10	0	10	0	4	0
Phishing Catcher	203	131	70	51	17	6	2	2

14,080,601 evaluation certificates. We then computed classification metrics based on the ground truth verification to compare the classifiers.

For the RF classifiers, where we directly compare the effects of dataset cleaning and class imbalances, we performed each training step (from dataset creation to classifier training and evaluation) three times and present the averaged results (see Table 10.3 for an overview). Due to the low variances observed across the three runs in practice, we did not extend this process for the DL classifiers to reduce the computational resources required for the evaluation.

We begin the analysis with the results and differences between the RF classifiers. Interestingly, the naive classifier performs best among the RF classifiers for the FPR 10^{-3} case, however it falls behind the other classifiers in the lower cases. Here, the time-based filtering has the best results for the 10^{-4} case, but the impostor domain classifiers are in the lead for the extremely low FPR cases (10^{-6} indicates no more than 14 FPs on the whole dataset). The same trend does not hold for certificates that include impostor domains, however, where the RF classifiers trained on impostor-filtered datasets were clearly able to detect more certificates for all FPR thresholds. Differences between time-based and naive filtering, on the other hand, are small for TP_i. Comparing the two different time-thresholds for the time-based

filtering method, we find that the 48 hours-threshold performs slightly better than the 24 hours-threshold when using a balanced dataset.

Furthermore, we see a general trend that for the same dataset, including more benign certificates, thus resulting in unbalanced datasets, has a positive effect on classification performance. This trend holds even when the imbalance is introduced by removing phishing certificates instead of adding additional benign certificates, thus reducing the overall size of the dataset, as is the case with the n-02 dataset.

Next, we take a closer look at the classifiers' FPs, particularly those with high confidence, to obtain an intuition of the biases of the different classifiers. Here, a general look at the distribution of confidence scores reveals, that the number of FPs at the maximum confidence of 1.0 for classifiers trained on imbalanced datasets are decreasing with higher ratios of imbalance. A closer review of the FPs with the highest confidence for the different classifiers further reveals, that apparent impostor domains are present for all RF classifiers, but with the highest ratio for those trained on impostor training data. Here, an exemplary manual review of the FPs returned by the RF-i01-all classifier reveals a group of similar domain names that likely belong to a phishing campaign that was not detected by our ground truth, as we were able to confirm a subset of the domains as being malicious by consulting a search engine which returned warnings about a potential scam making use of the domains. While similar domains were also returned with high confidence by classifiers trained on other training datasets, they also include domain names without a discernible reference to a target, thus making the classification outcome less explainable. Both time-based and naive filtering include such FPs, which among others include domains consisting only of numbers, or with high character entropy.

For the DL classifiers, the ResNet trained on time-filtering data performs best for the high FPR case, but falls behind the other filtering methods for all other cases. Here, the new combined architecture of CNN and LSTM performs best, but does still not match the RF classifiers in performance. In general, including features does not seem to improve performance for low FPR, as the performances are similar for FPRs below 10^{-4}. A comparison of TP_i for the DL classifiers mirrors the results of the RF classifiers, in that the classifiers trained on impostor-filtered datasets were better able to detect certificates that include impostor domains, regardless of whether they were trained only on domain names or included additional features from the certificates. Note, that the highest number of detected TP_i, for the ResNet-i-01-san at the 10^{-3} FPR threshold, corresponds to 47.01% of all certificates with impostor domains detected by the rule-based classifier of Chapter 4.

An additional finding is that the DL classifiers were not able to compete with RFs on the extremely low FPR cases. However, a closer look at the FPs with the highest confidences reveals that the classifiers trained on impostor domains and, to a lesser extend, time-filtered domains seem to detect domain names that include recognizable target names in nearly all cases, while the classifiers trained on naively filtered data include domain names without a reference to a target more frequently. A manual comparison of the ResNet classifier with all features to a domain-only ResNet for the case of the i-01 dataset indicates, that the domain-only classifier was more likely to predict domain names that include a recognizable target.

Interestingly, the DL classifiers mainly classified longer domain names with many domain labels with the highest confidence, while the RF classifiers trained on similar data found different patterns of typosquatting domains that were returned with the

highest confidence. Still, all classifiers include domain names that might have been missed by our ground-truth labeling rather than being truly benign, a hypothesis we discuss in more detail in the next section. In direct comparison, impostor domain-trained classifiers always seem to better represent that class in their high-confidence prediction, with the exception of the ResNet-i02$_{\text{all}}$ classifier, which seems to not have converged during training, resulting in random results.

Finally, the baseline classifiers no longer offer the highest accuracies in this evaluation. While both classifiers still return true positives in the lowest FPR cases, they still fall behind the RF-i-01 classifier. Here, creating templates according to the method by Sakurai et al. on impostor domains does not seem to result in a higher accuracy compared to using the naive filtering method. For the template classifier, looking at the FPs can reveal regular expressions with high entropy reduction scores which are nonetheless too generic, and which can be removed to fine-tune the classifier in practice without re-training. Similarly, the phishing catcher classifier does not require any training at all, and false positives are also easily recognizable and tunable due to the list of suspicious keywords that are used to create the domain ratings.

In all, we found that dataset cleaning and class imbalance during training resulted in classifiers capable of retaining some usefulness even when requiring extremely low FRPs. While the performance of RF classifiers was generally superior, this might be due to a shortcoming of the ground-truth labeling, as the DL classifiers also seem to have learned accurate representations of impostor domains.

10.6. Discussion

In this chapter, we presented the evaluation results of several RF and DL classifiers on a CT classification pipeline, comparing different data filtering techniques and classifier choices. We found, that RF classifiers trained on imbalanced datasets filtered for impostor domains performed best in the low FPR settings that are required to apply this classification technique to real-world data. While the DL classifiers seem to have learned the concept of impostor domains well, it seems as though the ground truth is incomplete, or that there is a large amount of benign websites that is very similar to impostor domains, e.g., in that they include popular targets in long subdomains, as their overall performance was worse than that of the RF classifiers in the low FRP cases. In the following, we discuss potential limitations of the results, as well as their implications on the broader context of early phishing website detection.

10.6.1. Limitations

In general, the results of the our evaluations depend on specific classifier choices, and might change drastically depending on even small decisions in architecture or dataset creation. As such, the trends that were revealed in the second evaluation might have to be confirmed using more classifiers with different architectures to be confirmed in general. We note, however, that several of the presented approaches seem to have had a clear positive effect on classification performance in our experiment, in particular the inclusion of more benign certificates during training. Furthermore, the standard deviation across the three runs that were performed for each RF classifier was low, thus indicating internal validity of the evaluation. Still, it is likely that

further optimizing the classifiers used for the CT log classification task could lead to generally better results in the future.

A second major issue with the evaluation is the lack of ground truth for the evaluation data. While we attempted to alleviate this issue to the extend of our capabilities, even the delayed labeling with the goal of increasing the number of verified phishing domain names is unlikely to have resulted in a complete ground truth. The analysis of high-confidence FPs confirms this finding, as these certificates often include clear similarities to impostor domains for both RF and DL classifiers, in particular those trained on impostor domains. Here, it might be necessary to rely on the insights provided by larger organizations such as browser vendors, or threat intelligence feed or blocklist providers, to be better able to decide whether the found certificates do indeed belong to malicious operations, or how great the impact of blocking them would be on benign browsing. Additionally, a live classification where screenshots are captured for positively classified domain names and threat intelligence, such as GSB, is continually updated and referenced could improve ground-truth labeling. This shortcoming of ground-truth labeling was also experienced by Lin et al., who only confirmed less than half of their positives in threat intelligence feeds, and instead relied on manual labeling of website screenshots to confirm most of the positives [Lin+21].

Furthermore, we did not compare all possible combinations of classifiers and datasets due to time and resource constraints. The classifiers were selected to best fulfill the research objectives of providing estimates on the effect of training data cleaning, class imbalance, and the inclusion of certificate features for DL classifiers, all of which were addressed by the second evaluation. It is, however, possible that some combinations that we did not test would result in deviations from the generalized findings. While we argue that the FP and TP analyses performed to gain additional insights into the classification decisions of the classifiers seem to support our conclusions, they might be further studied for generalizations over different training datasets, classifiers, and evaluation time periods in the future.

Finally, our experiments did not address a number of issues that might be necessary to deploy CT log classifiers in practice. For one, we ignored the robustness of classifiers against adversarial effects. While the proposed setting of central classification and verification might make adversarial attacks less powerful, in addition to the complexity of attacks on discreet values such as domain names in general, future work might look at this concern in more detail. The best performing classifiers in our experiments also all have a bias towards few popular services, which may have to be extended to offer better coverage in general and in practice. When classifying over a longer timespan, classifiers may also have to be retrained regularly, which we did not look at in our evaluations. Lastly, the handling of positives, in particular for domain names with blank, domain parking, or seemingly benign homepages is currently unclear, as these domain names might be phishing websites that require a path or employ website cloaking, or might indeed be legitimate. While the positive classification results could be used internally by larger organizations or email providers to provide early warnings about potential phishing websites, it is therefore somewhat unclear how these results could be employed to provide a more comprehensive protection for the general population. To which extend the detection results can be translated into real-world detection capabilities is therefore an open question for future work.

10.6.2. Implications

First, the two evaluations confirm that the CT log classification pipeline is well suited for the analyses we performed. We found, that it can be used to train and evaluate different classifiers in retrospective analyses, even though the problem of providing ground truth labels in this setting limits the generalizability of the results. We also argue, that it should be relatively easy to perform different types of evaluation as well, for example an analysis on the necessity of retraining, differences between different CT logs, or live classifications.

The classification outcomes of the first evaluation did not indicate usable classifiers for the proposed use case, as the RF and LSTM classifiers did not yield higher numbers of TPs for the given FPR thresholds than previous classifiers, which also did not achieve numbers of TPs in the same order of magnitude as the FPs. It is possible, that these outcomes can be explained with noisy training data. According to estimates in previous work, $62\% - 73\%$ of phishing websites are actually hosted on compromised infrastructure [LJ19; Le +19]. This circumstance might affect the approaches by Sakurai et al. and the Phishing Catcher less than the machine learning classifiers, since the Phishing Catcher makes use of rules that were created by domain experts and therefore is not affected by compromised server certificates, while the template classifier uses regular expressions that represent patterns found in the domain names of malicious certificates. Although the template classifier is trained on the same noisy data, it might be influenced less by benign certificates as we observe fewer patterns within the domain names of compromised servers.

In contrast, the overall results of the second evaluation in Table 10.3 suggest, that phishing classification on CT logs with low FPRs seems to be feasible. Compared to the previous evaluation, we saw general improvements, that likely resulted due to changes in the dataset cleaning, class imbalance, or the larger size of the training datasets overall. In particular, it is possible that the *naive* filtering employed in the second evaluation already removes more noise from the training dataset than the method based on domain knowledge, which was used in the first evaluation. This difference, and other possible filtering methods might be compared in more detail in the future. Note, that the number of certificates overall is lower in the second evaluation, and there might be differences in the quality of ground-truth validation. Consequently, the number of TPs are not directly comparable between the two evaluations.

In the second evaluation, we found that impostor-domain filtering seems to generally be useful, in that it resulted in the best performance for low FPRs, and additionally results in more explainable classification results. We further observed, that RF classifiers seem to be better suited to detect those certificates of phishing websites that are included in blocklists in the low FPR ranges, while the DL methods outperform on higher FPRs. Regardless of which filtering method was used, classification results seem to be particularly well when using class imbalance during classifier training, even if the imbalance is created by removing malicious certificates instead of adding additional benign ones.

Comparing the two proposed filtering methods for a fixed ratio of benign and phishing certificates during training, we find clear differences when comparing the number of TP_i, but not TP. A closer look at the non-impostor TPs generated by the two filtering methods for low FPR thresholds indicates, that this is not only due

to impostor domains that were not detected by the simple rule-based classifier, but also includes certificates with domain names without a reference to a target. This indicates, that there is a class of certificates of phishing websites that do not include impostor domains, making them more complicated to label manually without context, but that is still detectable by RF classifiers. Still, we argue that impostor domain filtering was superior in our use case, as it provided more explainable results that also focus on the specific URL categories that were not covered by other approaches proposed in this thesis. On the other hand, it does have the disadvantage of focusing on only a subset of phishing domains, which might be exploited by attackers to avoid detection. As previously noted, impostor domain filtering furthermore only focuses on a subset of benign targets, which were selected from popular websites and found to be common targets in the analysis in Chapter 4. These problems might be less pronounced in time-based or even naive filtering, however in our experiments these approaches also resulted in worse and less explainable classification results, likely due to the noise of compromised and benign-hosting domain names in the training datasets.

As an advantage of impostor-filtering, the restriction to impostor domains also seems to be successful in removing compromised domains and benign hosting services from the positives of the classifiers, which were likely the main reason for the low performances of RF and DL classifiers in the first evaluation. We argue, that detecting the certificates of compromised (or soon-to-be compromised) domains is not a sensible task, as the certificate would typically be requested before the compromise and therefore should not exhibit any malicious or otherwise remarkable features compared to other benign websites that were not compromised. As such, the early detection of phishing domain names in CT logs is likely to be restricted to a subset of all phishing websites, and in particular to impostor domains. Detecting websites hosted on compromised or benign hosting infrastructure would therefore require different approaches, such as the user education or design intervetions presented in this thesis, or automated approaches that make use of more context than is available in certificates.

The analysis of FP results might indicate, that neither impostor domains nor the time-based approach are a good criterion to filter for during training. While classifiers trained using both methods seem to have learned a representation of domain names resembling impostor domains, the low evaluation results might indicate, that these domains are common even among benign websites. It might therefore make sense to change the setting of the study, and for example provide feedback directly in a browser extension that warns its users when visiting a potential impostor domains. Here, the fact that the models were able to learn the representations well might indicate, that it is possible to train classifiers that are able to extract the target from an impostor domain, which might be used to better reflect a user's comprehension of the URL and therefore improve the signaling power of an extension or similar design intervention.

That it is possible to learn a representation of impostor domains also results in additional interesting venues for future work, including multi-class classification where the output also indicates which target was impersonated, or even classifiers that are able to extract or highlight the exact substring that includes a reference to the target. As a consequence, training a similar classifier to the visual approaches in previous work that do not require retraining when adding new targets might also

be possible. These classifiers might improve the explainability of the results, or might be employed in different contexts, such as in browsers, to warn users about websites that seem to be impersonating a different website. Generally, it may be possible to train classifiers to exhibit better performances for specific categories of URLs, e.g., subdomain posing or typosquatting. This fact seems to be supported by the FP analysis, where we confirm that even when classifiers are trained on the same dataset of impostor-filtered certificates, the domain names detected with high confidence can clearly differ. For example, the $\texttt{RF-i-01}_{\texttt{all}}$ learned a typosquatting campaign of Amazon, while `CL` classified a number of subdomain posing domains with high confidence, indicating that the different classifiers focused on different URL categories.

Compared to the current state of the art in visual detection of phishing attacks (see [Lin+21]), the classifiers evaluated in this chapter had a lower accuracy. This is likely due to several reasons, including the effect of providing additional context in website screenshots. Furthermore, the ground truth is still a problem, as the authors of Phishpedia required manual labeling for more than 50% of their positives. We argue, that the general approach of certificate and domain name classification presented in this chapter still has some merit, as it has several advantages compared to visual approaches. First, the classification of information that is directly included in certificates scales better, as we do not have to download the website, nor do we have to render and make screenshots, which is a complex process that potentially reveals the classifier or the machine it runs on to malware or DoS attacks, and is additionally susceptible to website cloaking. This also makes the classification faster overall, as the website does not have to be downloaded, and the models required for inference are potentially smaller. Domain name analysis is also still possible when a website is not available, for example when a path component is required, as we found to be the case for the majority of 87.90% of the URLs analyzed in Chapter 4. A disadvantage both visual and certificate-based approaches currently share is the restriction of only detecting the targets that were included in training or explicitly provided in a match list, which has to be maintained in practice. In all, the two approaches can complement each other to provide better detection capabilities overall, and might be compared and possibly combined in the future.

Due to the focus on several selected research questions in the second evaluation, a number of questions are still open for future work. Here, different meta-classifiers might be a venue to explore, though it should in theory be possible to use the maximum meta classifier in all cases once the FPR for domain names is sufficiently low. Exploring and evaluating other classifiers might also result in further improvements in classification outcomes. Next, it is possible to train domain-only classifiers on larger sets of domain names of phishing websites in general, even when they do not have a corresponding certificate. This might for example be used to increase the number of impostor domains in the training set, though it is not possible to use time-based filtering without certificates. Additionally, it may be possible to apply a similar detection approach as the domain-only classifiers to directly classify newly registered domains at TLD root zones. While this requires access to the root zones, which is not always available, this would potentially give a more complete view on registered domain names, and provide even earlier detection compared to CT logs, as certificates that include a domain name can only be requested after the domain was registered. Finally, it might be interesting to see how the early detection of phishing

websites influences the phishing ecosystem, e.g., whether attackers switch practices, and how to extend the approaches presented in this chapter to cover spear phishing attacks.

To summarize, this chapter presented a pipeline to perform evaluations on CT logs. In two evaluations, we found that classifiers are able to learn a representation of impostor domains, which are a promising indicator of whether a domain name is malicious. We found, that classifiers can be trained to detect the certificates of phishing websites even when restricted to extremely low FPRs. While our classifiers are still far from detecting all TPs or even only the subset of TP_is, our results indicate that classifiers might be improved enough to deploy them in practice, thus providing an additional layer in the phishing prevention process.

Part IV.

Conclusion

Chapter 11

Conclusion

In this thesis, we presented a multi-layered defense aiming to decrease the risk of phishing attacks using education, design interventions, and automated detection. We utilized a kill chain process model and URL categories to categorize both attacks and defenses, thus emphasizing how the defenses come together to create different layers that complement each other.

In general, this thesis demonstrated how different detection strategies are necessary to prevent different types of phishing attacks, and how prevention strategies can be designed to complement each other by focusing on different categories of URLs and steps in the attack process. The following section summarizes the main findings and implications, followed by an outlook about possible venues for future work.

11.1. Main Findings

The phishing prevention methods presented in this thesis focus on different steps in the phishing process and different URL categories, thus forming a multi-layered defense against phishing attacks.

On the phishing attack kill chain, the automated detection of phishing websites based on the certificates in CT logs makes up the earliest defense presented in this thesis, as it aims to detect phishing attacks in the preparation phase before the attack delivery. Here, we analyzed the **certificates of phishing websites** in an effort to find discerning features that might be used to detect phishing websites in user education or automated detection. We compared two scenarios that roughly translate into automated detection without any additional context (e.g., using CT logs as source), or analyzing the certificate from the view of a user who has an idea about which website they are trying to visit. The direct comparison of phishing and benign certificates unsurprisingly did not reveal any clearly discernible features that separate the certificates of benign and phishing websites, leading us to the conclusion that the information collected from the certificate apart from the included domain names is unlikely to be helpful in automated detection. In the second scenario, however, we did not find any evidence that phishing websites present certificate that actively spoof the information of their target website, in particular information about the organization seemed to not be impersonated in our dataset. While these findings do not solve the threat of website-based phishing in general, they might motivate

the usage of certificate information in user education, even though it can only be applied to benign websites that invest in a more complex validation option for their certificates.

For automated detection, we demonstrated that low FPRs are a vital issue when **classifying certificates on CT logs**, and found that even though the ground-truth labeling was incomplete, several classifiers were able to detect true phishing websites while retaining low FPRs. A comparative evaluation of several classifiers, training datasets and feature sets furthermore revealed, that class imbalance during training improves the classification performance in the low FPR cases, and that an impostor domain-filtered dataset used during training results in more explainable classification outcomes. While RF classifiers were superior to any of the evaluated DL classifiers for low FPR thresholds, we found that the DL models still learned a good representation of impostor domains, which might indicate that resolving problems with ground-truth labeling might yield improved results in the future. As to the question on whether including the information in certificates apart from domain names significantly improves the performance, we found that DL classifiers trained on only domain names performed at least similarly, in some cases even better than those that included additional information from the certificates, which supports the hypothesis that the additional information is not generally useful for classification. In all, we demonstrated that classification of certificates from CT logs is generally possible, as the proposed classifiers approached an acceptable level of FPs that makes it possible to deploy them in practice. However, the information in certificates is restricted to domain names, with the possible inclusion of wildcards which can be used to obfuscate subdomains, and thus only offers protection for impostor URLs where the target reference appears in the RD or a subdomain that is included in the certificate.

As complementary approaches, we therefore also presented several educational games and design interventions, which all aim to disrupt the kill chain by preventing the user action, and focus on different URL categories or delivery methods. A user study where we compared different **URL categories** indicated, that users seem to generally struggle with phishing URLs where the target appears first, as was demonstrated by the low detection scores for URLs of the *subdomain-first/only*, *http credentials*, and *RD* categories. Furthermore, the complexity of benign URLs seems to play a significant role as well, since we found that URLs with path or query components, or those containing subdomains, were more likely to be perceived as malicious. The results imply, that the inclusion of many different categories of URLs increases the predictive power of user studies, since they represent the large diversity of phishing attacks more closely. The results also lay the foundation for user education and other interventions that require knowledge of URLs by revealing which categories of URLs are not detected well even in a lab setting.

The approach to user education covered in this thesis was based on **anti-phishing learning games**. The focus of the newly proposed learning-game prototypes was to test different game mechanics and the effect of personalization, neither of which had a direct effect on the classification performance in the post-tests of the corresponding user studies. We did, however, confirm that differences between unknown and known services appear during gameplay and in the classification tests, thus motivating further research into the effect of fully personalized games and on how well the obtained knowledge, skills and awareness of players transfer to the real world. Regardless of which game was played, players significantly improved their results in the post-test,

but still fell short of "perfect" detection scores across all URL categories. While URLs with a target reference in the path were detected extremely well after playing the games, neither subdomain nor RD URLs improved to the same level. A retention test confirmed, that knowledge was retained over three months, and indicates that there might in fact be advantages to the more complex game mechanics for subdomain URLs. In all, the results of this study can help researchers and developers improve learning games in the future, and motivate further research into the personalization of anti-phishing education. The high accuracies when classifying URLs where the target appears in the path also indicate, that user education might alleviate one of the shortcomings of certificate-based detection, which might in return reduce the risk of URLs with the target in the RD.

Since subdomain URLs were still problematic even after playing a learning game, resulted in the lowest detection accuracy in the user study on URL categories, and are not necessarily detected using automated detection on CT logs, we analyzed **reverse domain name (RDN) notation** as an alternative to the normal URL notation that specifically aims to improve the detection accuracy for subdomain URLs. In the user study conducted to test this hypothesis, we found that there were indeed differences for subdomain URLs, which were detected with higher accuracy in RDN notation, however the accuracy dropped slightly for most other categories, in particular for http credentials. We argue that this is due to the fact that participants were not familiar with the new notation and that the general accuracy would improve if RDN was to be used more commonly. Further analysis revealed, that participants were significantly faster when making decisions about URLs in RDN compared to the normal notation, which might make it more likely that users consult the URL in practice if it is perceived as less cumbersome. While these results demonstrate that RDN might be an alternative notation for URLs that does not remove any information but still makes phishing URLs easier to detect, it remains an open question whether users would actually look at URLs in practice for their classification decisions.

A general shortcoming of all URL-based detection approaches is, that not all attacks rely on phishing websites to be successful. We therefore also conducted a user study to compare four different **UIs for email clients** and their effect on phishing email detection accuracy, with the aim of providing a complementary detection method for users that is also effective against website-less attacks and phishing websites hosted on benign infrastructure. The first UI tested in the study served as a baseline without any particular highlighting, and the other UIs added information about the history, highlighted the RD in the sender's email address, or summarized security information included in the email. We found, that two of the three UIs resulted in significant improvements over the baseline, however the study also indicated that users might put too much trust in two of the highlighting options by preferring UIs that could be interpreted as resulting in a binary outcome on whether the email is safe. Still, all three UIs resulted in significantly faster classification times, which might make it more likely that users perform a check for phishing emails in practice by removing some of the complexity involved in the task. We furthermore defined different categories of phishing and benign emails for the study, and found that lateral and spear phishing posed a substantial threat in our study, as neither category was detected well. Overall, this indicates that classification tasks including phishing emails should include a diverse set of email categories, and motivates further studies on the potentially positive effects of email client UI changes.

Taken together, the results of this thesis highlight the importance of reflecting the diversity of phishing attacks in user studies by including URLs and emails of different categories to ensure meaningful results. Furthermore, information about the familiarity with services of URLs and emails in classification tests played an important role in the studies presented in this thesis, and should be considered in future studies. Explicitly differentiating unknown and known services in interventions might also make them reflect the actual situations that users encounter in their real-life activities more closely, which might translate into benefits when knowledge, skills or awareness are required in the real world. Differences between evaluation setup and real world also played an important role in the development of CT log classifiers, which work on real-world data and therefore have additional requirements, for example extremely low FPRs. Finally, the context provided by different categories of impostor domains in this thesis demonstrates the advantages of a multi-layered approach that combines different prevention methods, which might also be of interest in the future. However, non-impostor phishing URLs still made up the majority of phishing URLs in our datasets, which also include attacks that are unlikely to be detectable using URL features alone, as they make use of benign infrastructure in the intended way, which might require different prevention approaches altogether.

11.2. Future Work

While the anti-phishing approaches presented in this thesis have been shown to be effective in specific use-cases, they are not able to provide a complete protection against phishing attacks in general. As such, there are several important venues and open questions for future work, which we discuss in the following.

First, an important next step for all of the proposed methods are **real-world evaluations**. For the design interventions and learning games, real-world evaluations could be performed by using simulated phishing attacks, where users also have to pass the awareness hurdle that was not present in the lab studies. This would give insights into how well education translates into practice, and whether the design interventions actually raise awareness or are ignored or not helpful. For the classification of CT logs, a real-world evaluation would entail live-classification of certificates as soon as they are added to the logs, preferably with screenshot-taking and continuous monitoring of potential positives and blocklists to check for inclusion. This would give an estimate on the effect of blocking impostor domains, and show whether the classifiers are already usable beyond our evaluations. It would also bring the verification issue of potential phishing websites found via domain names in certificates into focus, which is not currently addressed by existing approaches.

The results of our user studies indicate, that education might benefit from more **personalization**. Phishing attacks and susceptibility can depend on the specific user, and most users are likely to act differently in real life when, for example, confronted with unknown services. In particular, we have shown that familiarity had a significant effect on all classification tasks, and that players of anti-phishing learning games also interacted differently with unknown services during gameplay. Furthermore, the rise in personalization in attacks due to spear or lateral phishing attacks, where OSINT information can be abused by attackers to create more believable messages, motivates the usage of more robust educational material that is tailored to the specific context

of the user. As such, further personalized education might be an important venue for future work to ensure, that educational material remains relevant and effective against an evolving threat. Additionally, while we did include service familiarity in our studies, incorporating hosting service abuse as well might also be interesting for future work, as we found that they are common (based on an analysis of the RDs of phishing URLs in our dataset), and hard or even impossible to detect by the approaches presented in this thesis.

An additional potential venue for future work is the **combination of design interventions and education**. In our studies, we found that problems with URL reading remained even after playing anti-phishing learning games. It might be possible to combine URL highlighting, RDN notation, email highlighting, or other design interventions which take care of the complex parsing process of URLs, emails, or certificates and highlight relevant information, with user education that teaches users how to process this highlighted or summarized information to improve their decision process regarding phishing attacks. This might on the one hand reduce the complexity of education, as the complexity of parsing and collecting relevant information is reduced, and also have the benefit of raising awareness of relevant information which is readily available but otherwise ignored or misunderstood. On the other hand, implementing highlighting or summary options in popular browsers or email clients likely requires the cooperation of larger corporations (e.g., browser vendors), who might have to be convinced that information highlighting leads to improvements in combination with education, which they do not have any control about.

More broadly speaking, it has been pointed out by previous research in usable security (e.g., [Cra08; Sas15]) that depending on **user education** is unlikely to provide a robust defensive strategy in the long run and in the face of evolving attackers, and might be replaced by other defensive measures in the future. This result is particularly complicated to achieve for phishing attacks, which explicitly aim at the human factor, and are currently neither prevented by technical nor human-centered defenses. As such, it might be a more promising strategy or even ultimately necessary to provide educational materials and design interventions that can be updated and revised to reflect current attack trends. Whether it is possible to "solve" phishing attacks without education in the future remains an open question.

On the other hand, the **automated detection** of phishing websites is still an ongoing research problem as well. Previous studies have found problems regarding the generalizability of results in this research area, in particular regarding the selection of training and evaluation datasets [Das+19]. These problems might make the transition of automated methods to the real world problematic, when classification performances decrease in the real world or are unable to cope with large numbers of FPs. Here, our results in the context of automated detection of certificates of phishing websites on CT logs indicate, that extremely low numbers of FPs are required, but also that ground-truth validation of realistic test data is challenging, both of which might be improved in the future. We found, that CT logs are a great source for automated detection, since they make large-scale detection on real data possible, thus providing realistic scenarios that directly translate to the real world. Our approach furthermore does not require user data such as browsing behavior, and enables centralized operation and verification. Using only certificates as inputs for the classification does, however, also come at the price of less context compared to

other automated approaches (e.g., using URLs or emails as input). Our evaluation focused on impostor domains, which highlights the importance of context for phishing detection, which might still not be available to an automated classifier in the same extend as a human operator. Future work might look at how to generate or extend the context for automated approaches (i.e., without relying on users), that still remain unobtrusive and do not require invasive analysis of, for example, a user's email history. Here, the research of AI is currently fast evolving, and rapid advances in natural language processing might indicate, that highly successful automated detection engines might be feasible in the foreseeable future.

Overall, creating and evaluating more effective systems that prevent different attacks is still an ongoing challenge in phishing prevention. This thesis did not cover all possible classes of attacks, nor did we explore all steps in the phishing kill chain, which could be covered in the future to provide a more holistic phishing prevention system.

Part V.

Appendices

Appendix A

Additional Information

The following sections contain additional materials and details on the studies and evaluations presented in the main part of the thesis. Further information on impostor URLs (Chapter 4) can be found in Section A.1. Screenshots and detailed statistics for the URLs shown in user study on phishing URL categories are presented in Section A.2. Similarly, additional information on the user study on anti-phishing learning games (Chapter 6) is included in Section A.3. Section A.4 contains information on the user study on RDN notation from Chapter 7, followed by additional details on the email client UI study from Chapter 8 in Section A.5. Finally, Sections A.6 and A.7 include further information on the analysis and automated detection of the certificates of phishing websites presented in Chapters 9 and 10, respectively.

Contents

A.1. Phishing and Impostor URLs

This section includes information on the rule-based classification of impostor domains presented in Chapter 4. Examples for each rule can be found in Table A.1.

Table A.1.: Example domain names matching impostor rules

Rule	Domain	Target RD
RD embedding	amazon.co.jp.s1s33.xyz	amazon.co.jp
e2LD embedding	facebook-875912582.dailyshift.app	facebook.com
Strict typosquatting	instagr-m.us	instagram.com
Relaxed typosquatting	loginn-amazom.myvnc.com	amazon.com

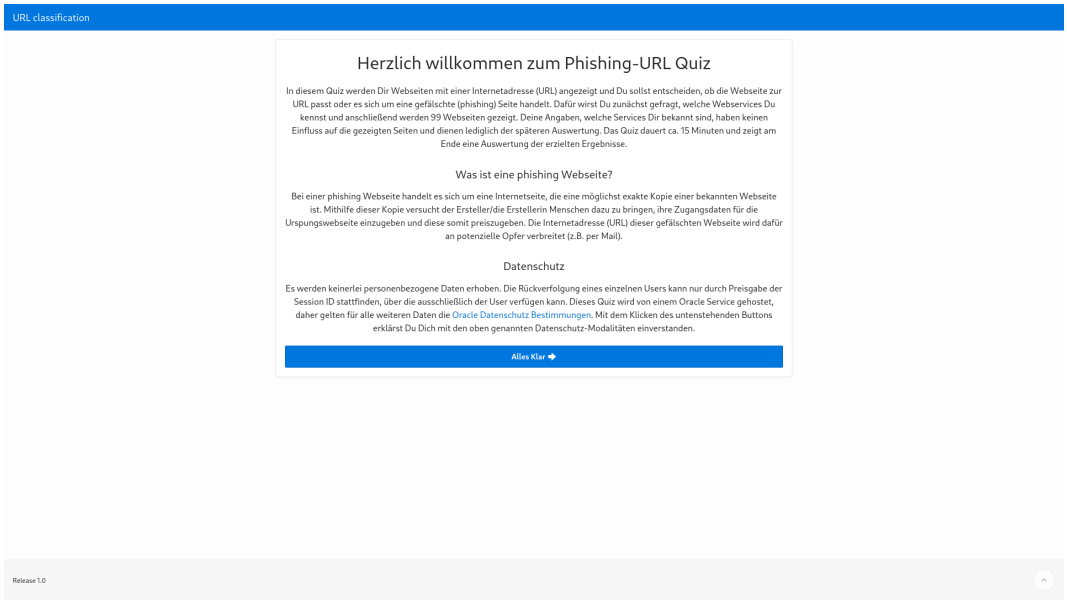


Figure A.1.: Welcome screen of the URL study (taken from [Dre22]).

A.2. Categorization of Impostor URLs

This appendix includes additional information on the URL categorization and user study performed in Chapter 5. First, Table A.2 contains the URL categories defined by Reynolds et al. [Rey+20]. Next, Figures A.1, A.2, A.3, and A.4 depict screenshots of the survey. Furthermore, Tables A.3 and A.4 depict the mean performance scores for the URLs in the user study.

Table A.2.: URL categorization by Reynolds et al. (adapted from [Rey+20])

Category	Description	Example
Typo-squatting	A domain that looks similar but is spelled differently to one known by the victim	`https://tarqet.com`
Subdomain as Domain	Places an unrelated but familiar name as the subdomain for a URL	`https://target.com.sign-in.info`
IP Address	Includes Only an IP address	`http://127.0.0.1/`
IDN Homographs	Use unicode characters that look similar to the true website's name	`https://tärget.com`
HTTP credentials as origin	Use http credentials to precede the FQDN in a URL	`https://target.com@n593.biz`
No Apparent Identity	URL contains only unrecognizable strings or a description of function	`https://kjgsksdg93528.com`
Self-declared secure	Recognizable hostname is prepended with "secure"	`https://secure-target.com`
Ambiguous Delimiter	Puts delimiters (e.g., `@`) in parts of the URL where they have no effect	`https://target.com@other.com#@example.com`
Unfamiliar TLD	Uses an unfamiliar TLD to terminate the FQDN instead of a more common TLD	`https://target.com-issues.support`
Overrunning Subdomain	Uses a long chain of subdomains to obscure the FQDN	`https://www.target.com.js2awp-1lf8xe89770by5cyxqbwewp.gvicw9vl45lie-csmcmcut7z95qcms.etz5811-eiue348wi0li27dh8jtkku.mx`
URL Encoded Characters	Encodes characters in the URL to hide important delimiter characters	`https://target.com%41%41%41%2e%41%52`
Query parameters or Fragment Posing	Places familiar hostnames in the query or fragment portion of the URL	`https://get-help.page?target.com` `https://192.17.42.13#target.com`
Path Posing	Place familiar hostnames in the path portion of the URL	`https://connection22.co/target.com`

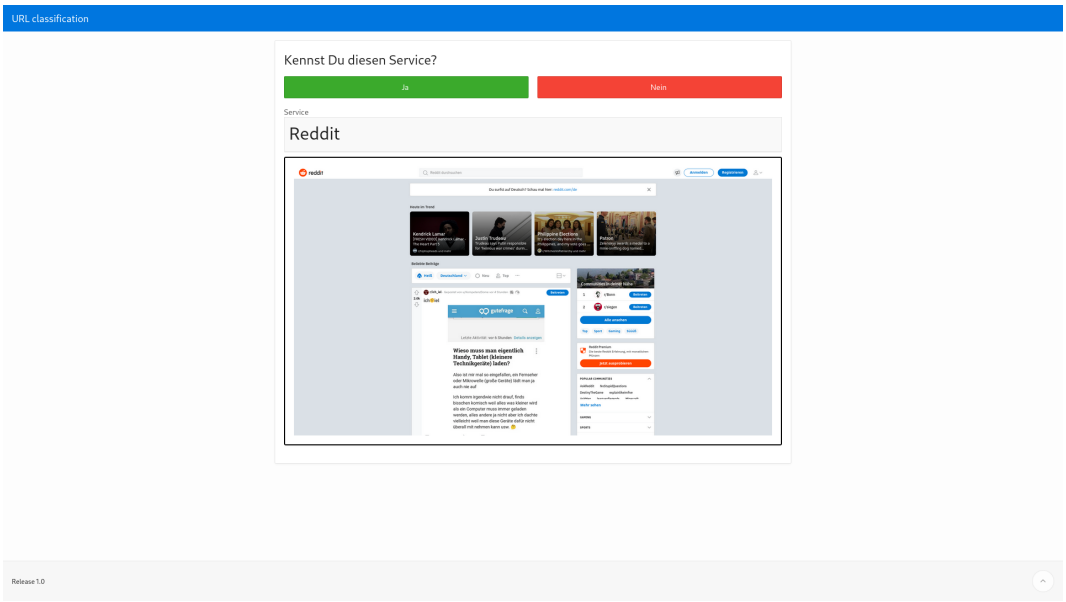


Figure A.2.: Screenshot of the service familiarity questionnaire in the URL study (taken from [Dre22]).

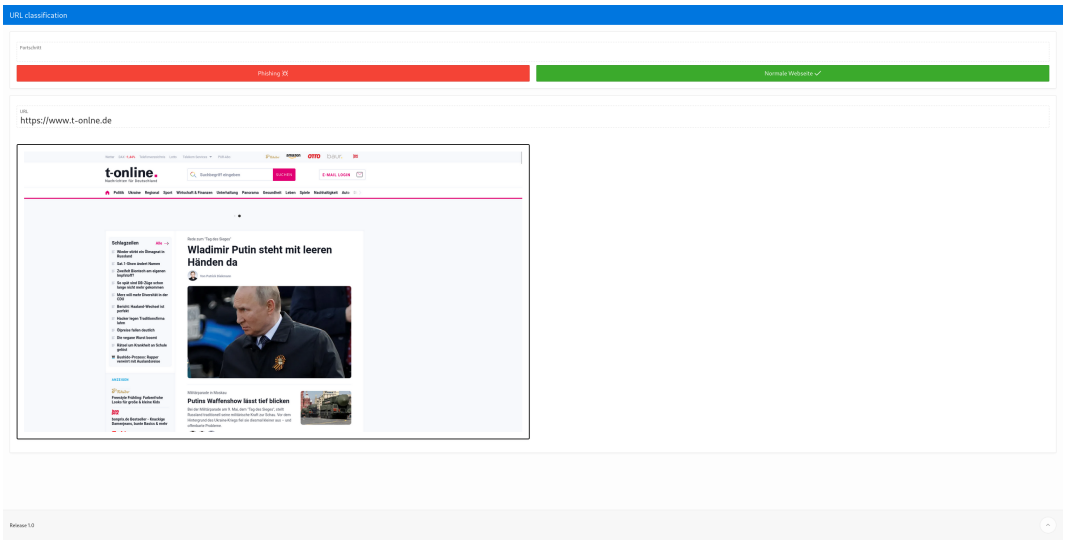


Figure A.3.: Screenshot of the URL study classification task (taken from [Dre22]).

Table A.3.: Performances for benign URLs in URL category study (URLs of unfamiliar services were removed).

URL	N	Category	Mean
https://www.dhl.de	37	Plain	1.0
https://www.facebook.com	38	Plain	1.0
https://www.merkur.de	18	Plain	1.0
https://www.derwesten.de	10	Plain	1.0
https://www.welt.de	39	Plain	0.974
https://www.chefkoch.de	38	Plain	0.974
https://www.samsung.com	32	Plain	0.969
https://www.web.de	30	Plain	0.967
https://www.amazon.com	41	Plain	0.951
https://www.chip.de	37	Plain	0.919
https://www.twitch.tv	24	Plain	0.917
https://www.ebay-kleinanzeigen.de	41	Plain	0.878
https://www.merkur.de?request?a=8801701589660449956	18	Path	0.833
https://www.chip.de?request=7734971300542504185	37	Path	0.811
https://www.instagram.com?client_id=9148891928820696079	40	Path	0.775
https://www.sport1.de?request=b%27%5Cxbe%5Cxca%5Cxd[...]	17	Path	0.765
https://www.chefkoch.de?client_id=3256578733479199268	38	Path	0.763
https://www.paypal.com?client_id=b%27c%5Cxaa%2A%5C[...]	41	Path	0.756
https://www.paypal.com?request=b%27%5Cxf9%5Cx15%5Cn[...]	40	Path	0.75
https://www.t-online.de?request=1510910431402647394	32	Path	0.75
https://www.samsung.com?request=b%27%5Cn%5Cxe4%7B3Z[...]	32	Path	0.75
https://www.tagesschau.de?request=3792775142391940030	41	Path	0.732
https://www.chip.de?client_id?a=1895698834972220894	37	Path	0.73
https://www.paypal.com?request=b%27KJyV%5Cxcf%5Cx17...]	43	Path	0.721
https://www.twitch.tv?client_id=b%27%5Cxaa%5Cxd6%5[...]	25	Path	0.72
https://www.paypal.com?client_id=b%27%5Cx92G%605%5[...]	38	Path	0.711
https://www.amazon.com?request=9171341737121226924	41	Path	0.707
https://www.merkur.de?client_id?a=b%27%5Cxfa %5Cxdd[...]	17	Path	0.706
https://www.chefkoch.de?request=9683977234168182615	36	Path	0.694
https://www.instagram.com?client_id?a=b%27Z%5Cxc8%5[...]	38	Path	0.684
https://www.amazon.com?request=b%27%5Cxea%5Cx13%5Cx[...]	41	Path	0.683
https://www.spiegel.de?request=2948715188917027892	37	Path	0.676
https://www.instagram.com?request=b%27%5Cx9f A%5E %[...]	42	Path	0.667
https://www.idealo.de?request?a=b%27%5Cxe6%5Cxc0%5C[...]	33	Path	0.667
https://www.idealo.de?request=7097990411080245371	30	Path	0.667
https://www.gmx.net?client_id=b%22%5Cx9e%5Cx95%5Cx[...]	27	Path	0.667
https://www.derwesten.de?client_id=b%27%5Cxd9Tc%5Cx[...]	9	Path	0.667
https://www.ebay-kleinanzeigen.de?client_id?a=b%27%5[...]	41	Path	0.659
https://www.web.de?client_id=3415526714293118578	32	Path	0.625
https://www.gmx.net?client_id=b%27%5Cxa2%5Cxca%5Cx[...]	30	Path	0.567

Table A.4.: Performances for phishing URLs in URL category study (URLs of unfamiliar services were removed).

URL	N	Category	Mean
https://knqj9wrm8etoowtls1z387qqdv459qq6f635tht094lbp.w[...]	37	Subdomain-end	1.0
https://0i17460ibjn9bwwjoqk5e2q9h283p8gef3sdd0yr.qq3dlm[...]	38	Subdomain-end	1.0
https://www.ott0.de?client_id=4594320839498719144	38	Typo	1.0
https://www.wlet.de?client_id=5902123609469213206	36	Typo	1.0
https://www.derwesten-secure.de?request=b%27%5Cxfe0%[...]	9	RD	1.0
https://7g40m8niwctnsyevnw8e3o6zg1i1p8bwcaxrrf4zwp.8ttd[...]	43	Subdomain-end	0.977
https://www.y0utube.com?client_id=4691860970233155429	41	Typo	0.976
https://www.agesschau.de	41	Typo	0.976
https://i4r915ew8gb8aouj.de?www.ebay.de	40	Path	0.975
https://www.kbay.de	40	Typo	0.975
https://www.eba.de	39	Typo	0.974
https://kvuohyy4lgurbdyf3jepmqa21d3x1stsw7gov5idkzn.www[...]	39	Subdomain-end	0.974
https://9mbsfotkhtljpfu4.com/www.google.com	38	Path	0.974
https://www.vveb.de	33	Typo	0.97
https://www.gosogle.com?request=b%27%5Cx9f%5Cxde%5C[...]	41	Typo	0.951
https://k3qglq6stzqw8ggh.com/www.ebay-kleinanzeigen.de	41	Path	0.951
https://www.wikipedio.org?request=b%27G%5Cx93%7D+%5[...]	41	Typo	0.951
https://www.ebacy.de	39	Typo	0.949
https://www.instaagram.com?client_id=7958564013988669330	39	Typo	0.949
https://www.lebay-kleinanzeigen.de?request=b%27%5Cxb0[...]	39	Typo	0.949
https://www.wikipediaa.org?request=2927723380819768289	38	Typo	0.947
https://www.hcip.de?request=b%27%28%5Cx0ei%5Cx90%5[...]	37	Typo	0.946
https://vsotqst5rn2e7e84.com?www.spiegel.de	37	Path	0.946
https://www.sporz1.de?request=8254018817974639890	17	Typo	0.941
https://www.iidealo.de?request=b%27%5Cxa1%5Cxb1%5Cx[...]	32	Typo	0.938
https://www.eb.de	31	Typo	0.935
https://www.gax.net?request=4137666067858483598	29	Typo	0.931
https://www.pypal.com?client_id=b%27o%7C%5Cx1d+%5C[...]	42	Typo	0.929
https://www.ebay-kleinanze1gen.de?request=b%27%5Cxe9...]	39	Typo	0.923
https://www.foocus.de	38	Typo	0.921
https://www.link-tagesschau.de?request=b%27%2A5%5Cx1[...]	39	RD	0.897
https://www.reddti.com	26	Typo	0.885
https://www.googe.com	40	Typo	0.875
https://www.t-onlne.de	32	Typo	0.875
https://www.safe-samsung.com	31	RD	0.806
https://www.wikipeia.org?request=3842376134860118984	39	Typo	0.795
https://www.wikipedia.link?request=b%27%5Cxc8%5Cx87...]	38	RD	0.789
https://www.chefkoch.secure	36	RD	0.778
https://www.help-reddit.com?request=b%27%5Cxb2%5Cx87%27	26	RD	0.731
https://www.amazon.com@%61%72%2e%76%69%77%38%34[...]	40	Http-credentials	0.725
https://www.otto.site	35	RD	0.714
https://www.dhl.de.5dwe3rq531adkzkftonuh16ylsblia14xtd7[...]	38	Subdomain-first	0.711
https://www.ebay-kleinanzeigens.de?request=644325675462[...]	42	Typo	0.69
https://www.t-onilne.de	32	Typo	0.688
https://www.idealo.de@0z3tu4tsiuet7wdt.de?request=b%27[...]	32	Http-credentials	0.688
https://www.ebay.de@%37%76%6b%32%65%2e%61%72?re[...]	40	Http-credentials	0.675
https://www.google.com.support-page.com?request=b%27A2[...]	40	Subdomain-first	0.675
https://www.google-help.com	39	RD	0.667
https://www.derwesten.de.ves9wi3uhkhrz5v2.de	9	Subdomain-first	0.667
https://www.focus-link.de?client_id=849977719853577737	37	RD	0.649
https://www.twitter.com@vgjdjepskv4l0f8l.com?client_id[...]	36	Http-credentials	0.639
https://www.ideao.de?client_id=b%27%5Cxa4%5Cxc5%60[...]	31	Typo	0.581
https://www.dhl.de.link-support.com?client_id=b%27%5[...]	37	Subdomain-first	0.568
https://www.idealo.de.%6b%32%79%72%62%2e%61%72	32	Subdomain-first	0.562
https://www.sport1.de.2bstrc6zl31d61aq91z6ff06zg84kiw2o[...]	18	Subdomain-first	0.556
https://www.otto.de.i3qblme04koxujzd.de?client_id=2369[...]	39	Subdomain-first	0.538
https://www.sport1.de.%78%67%37%75%35%2e%61%72?[...]	17	Subdomain-first	0.529
https://www.derwesten.de%2e%71%75%63%66%73%2e%6[...]	9	Subdomain-first	0.444
https://www.youtube.com%2e%77%32%37%37%73%2e%61%72	41	Subdomain-first	0.39

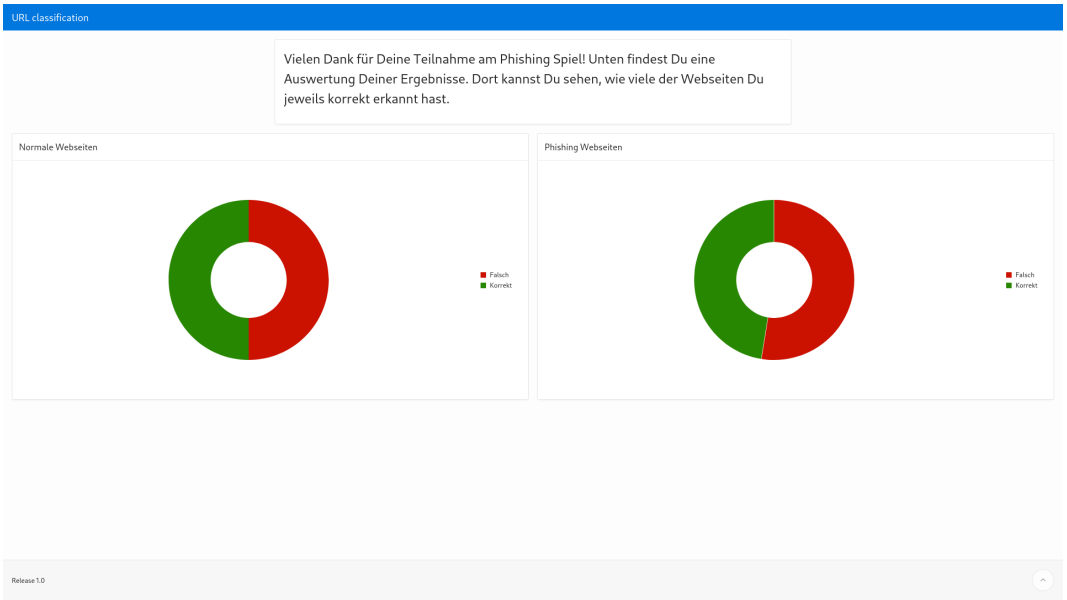


Figure A.4.: Screenshot of the feedback in the URL study (taken from [Dre22]).

Appendix Table A.5.: Summarized analysis results of POG data set (x = identified category; g = guessed category; y = yes; n = no; – = n/a) (taken from [11])

	Target Gr.					Edu. Context										factual						conceptual						procedural						meta-cognitive					
Reference	End-users	Non-CS Students	CS Students	Employees	IT employees	Primary School	Middle School	High School	College/University	Corporate	Informal	available?	focus: phishing?	digital?	remember	understand	apply	analyze	evaluate	create	remember	understand	apply	analyze	evaluate	create	remember	understand	apply	analyze	evaluate	create	remember	understand	apply	analyze	evaluate	create
[She+07]	x										x	y	y	y	x						x	x					x		x									
[HGG15]	x										x	n	y	y	x	x	x				x	x	x				x	x	x									
[Huy+17]		x							x			y	y	y	x	x					x	x																
[WJZ18]			x						x			y	y	y	x	x					x																	
[GKG15]		x				x	x					y	n	y	x	x	x	x	x		x	x	x	x	x		x	x										
[Lu18]			x						x			y	n	y	x	x	x	x	x		x	x	x	x	x		x	x	x	x	x							
[KW18]				x						x		y	n	y	x	x					x	x	x						x									
[BC14]	x										x	y	y	y	x	x					x	x					x											
[Wen+19]	x										x	y	y	y	x	x					x	x																
[Gey19]	x										x	y	n	y	x	x					x	x					x	x										
[ABP18]	x										x	n	n	y	x	x					x	x																
[ALM15]	g										g	n	y	y	x						x																	
[BA19]	g										g	n	y	y	x						x																	
[BC16]	x										x	–	y	n	x	x	x				x	x	x				x	x	x									
[Bau+17]		x				x	x					n	n	y	x	x					x	x																
[BP16]					x					x		–	n	n		x						x	x	x				x	x	x	x	x	x	x	x	x		
[BPF16]				x						x		–	n	n	x	x	x				x	x	x				x	x										
[Bha19]	x										x	n	n	y	x	x					x	x																
[CMB11]		x							x			n	n	y	x												x		x									
[CJ+18]				x						x		n	y	y	x	x					x	x					x		x									
[Con+07]			x						x			n	y	y	x	x		x			x							x	x									
[FMS19]				x						x		n	n	y	x						x																	
[Fre+19]				x						x		–	n	n			x	x	x						x	x				x	x	x	x		x	x	x	
[GP13]	x										x	–	n	n		x	x	x	x		x		x					x		x	x			x				
[Heb+17]					x					x		n	y	y			x				x						x											
[Kat+17]	g										g	n	n	y							x								x									
[Kul19]	g										g	n	n	y	x	x					x	x																
[Lop+18]		x					x					n	y	y	x	x	x				x																	
[MPJ18]		x				x	x					n	n	y	x	x	x				x	x	x				x	x	x				x	x				
[MAB17]	x										x	n	y	y	x	x					x	x																
[MNS10]				x						x		n	n	y	x		x	x			x						x		x									
[Ola+14]		x						x				n	n	y	x	x	x	x			x						x		x									
[OZ15]	g										g	n	n	y	x	x	x				x	x																
[RL16]					x					x		–	n	n				x	x	x						x				x	x	x		x	x			
[SRV15]	x										x	–	y	n	x			x			x	x		x														
[Tse+11]	g										g	n	n	y	x	x					x	x																
[Tse+15]	x	x					x				x	–	n	n	x						x																	
[Vuk12]	x										x	n	n	y	x	x					x	x																
[Yan+12]		x							x		x	n	y	y	x						x												x					
[Yas+18]			x						x			–	n	n	x	x	x	x		x			x				x	x			x		x					

A.3. Game-based Anti-Phishing Education

This appendix includes additional information on the anti-phishing learning games presented in Chapter 6. Table A.5 shows the POG dataset and the analysis results for anti-phishing learning games. Table A.6 depicts the learning goals of the proposed learning games. The mean performance scores for all URLs, results of the recognition of services questionnaire, and questions in the demographics questionnaire of the corresponding user study are shown in Tables A.7, A.8, A.9, and A.10. Table A.11 demonstrates, how the detailed URL categories from Chapter 5 map to the categories taught in the games.

Appendix Table A.6.: Learning goals including the mapping to the four games. Learning goals are marked with x if they apply to the analysis, decision, or personalized game (A), or creation game (C)

After playing the learning game, players should be able to ...			A	C
Remember	...	know the structure of URLs by recalling its components.	x	x
	...	name the manipulation techniques for URLs by listing the manipulation techniques for individual components.	x	x
	...	know the manipulation techniques for URLs by describing the manipulation of the components.	x	x
Understand	...	understand the structure of URLs by explaining the purpose of the components.	x	x
	...	understand the manipulation of the structure of URLs by explaining manipulation techniques for the components.	x	x
Apply	...	determine the individual components of a URL by performing URL parsing.	x	x
	...	compose valid URLs by combining the (necessary) components in the correct order.		x
	...	compose valid URLs by creating the (necessary) components in the correct order.		x
	...	change the structure of a URL by modifying components.		x
	...	manipulate the structure of a URL by modifying (necessary) components based on specific rules.		x
Analyze	...	analyze the structure of a URL by identifying the components.	x	x
	...	detect manipulations in the structure of a URL by identifying manipulated components.	x	
	...	recognize the manipulation technique applied to a URL by identifying/recognizing the manipulated component.	x	
Evaluate	...	assess the correctness of the structure of a URL by checking the components.	x	
	...	assess the manipulation of the structure of a URL by checking the components and identifying manipulated components.	x	
	...	distinguish benign URLs from manipulated URLs by comparing both URLs in terms of applied manipulation(s).	x	
Create	...	create correct URLs by creating and combining the (necessary) components.		x
	...	create manipulated URLs by manipulating and combining (necessary) components based on rules and the URL structure.		x

Appendix Table A.7.: URLs of the URL classification test comparing learning games in pre- and post-test and their mean performance scores

URL	N	Category	Pre	$Post_C$	$Post_A$	$Post_D$	$Post_P$
https://www.amazon.de/ap/signin?openid.pape.m[...]	182	Benign	0.5	0.667	0.85	0.911	0.755
https://meine.deutsche-bank.de/trxm/db/	174	Benign	0.506	0.681	0.711	0.705	0.689
https://accounts.google.com/signin/v2/identif[...]	182	Benign	0.604	0.604	0.675	0.711	0.612
https://vk.com/	33	Benign	0.905	1.0	0.889	1.0	0.875
https://www.gmx.net/	165	Benign	0.921	0.86	0.947	0.974	0.933
https://www.otto.de/user/login?entryPoint=log[...]	129	Benign	0.943	0.875	0.944	1.0	0.933
https://www.reddit.com/login/	137	Benign	0.971	0.971	1.0	1.0	0.974
https://www.focus.de/ajax/login/community_lo[...]	166	Benign	-	0.651	0.75	0.738	0.556
https://www.netflix.com/de-en/login	182	Benign	-	0.875	0.95	1.0	0.959
https://web.de/	153	Benign	-	0.975	0.921	0.944	0.897
https://214.156.43.197/login.live.com/	182	IP	0.83	0.875	0.95	1.0	0.98
https://b1ovam5.org/otto.de/	129	Path	0.948	0.938	1.0	1.0	0.978
https://uyvgo8i.net/RsHZdqidvhidpFbRVa/accoun[...]	182	Random	0.874	0.583	1.0	0.978	1.0
https://store.steamposered.com/login/	79	RD	0.33	0.448	0.45	0.5	0.467
https://mircosoft.com/login.srf?wa=wsignin1.0[...]	182	RD	0.495	0.438	0.725	0.6	0.653
https://v-k.com/	33	RD	0.5	0.444	0.667	0.75	0.938
https://sso.immoblienscout24.de/sso/login	176	RD	0.511	0.522	0.615	0.523	0.702
https://meine.deutsche-bank.online/trxm/db/	170	RD	0.612	0.791	0.684	0.568	0.711
https://amazon-secureserver.de/ap/signin?open[...]	182	RD	0.72	0.896	0.8	0.911	0.816
https://wwwcommerzbank.de/lp/login	175	RD	0.846	0.894	0.974	0.909	0.957
https://www.netflix.com-co.support/login	182	Subdomain	0.632	0.833	0.85	0.8	0.857
https://ebay.de.login.9ontzcjkgj2k.ru/ws/eBay[...]	133	Subdomain	0.796	0.917	0.975	0.978	0.917
https://gmx.net%6B%73%35%66%6C%6A%33%[...]	165	URL encoding	0.909	0.953	0.947	0.923	0.933
https://www.yi19p83.info/iWOaXLrmMRaymXsqdl/l[...]	182	Random	-	0.75	1.0	0.978	1.0
https://www.dropbox-account.com/login?hl=de&[...]	179	RD	-	0.766	0.55	0.5	0.479
https://www.paypall.de/signin?SignIn&UsingSS[...]	182	RD	-	0.854	0.95	0.956	0.959
https://netglix.com/de-en/login	182	RD	-	0.938	1.0	0.933	0.959
https://icloud.com-de.support/	171	Subdomain	-	0.867	0.718	0.75	0.86
https://www.twitch.tv.support.i1oc8c8.3pyozv3[...]	133	Subdomain	-	0.906	0.964	0.972	0.973
https://www.dropbox.com%70%6C%79%74%67%[...]	179	URL encoding	-	0.894	0.875	0.864	0.958

Appendix Table A.8.: URLs of the URL classification test comparing learning games in pre- and retention-test and their mean performance scores

URL	N	Category	Pre	Retn_C	Ret_A	Ret_D	Ret_P
https://meine.deutsche-bank.de/trxm/db/	62	Benign	0.456	0.52	0.647	0.55	0.706
https://www.amazon.de/ap/signin?openid.pape.m[...]	82	Benign	0.5	0.68	0.706	0.762	0.737
https://accounts.google.com/signin/v2/identif[...]	82	Benign	0.598	0.52	0.765	0.619	0.421
https://vk.com/	17	Benign	0.824	1.0	1.0	1.0	0.5
https://www.gmx.net/	76	Benign	0.921	0.958	1.0	1.0	0.824
https://www.otto.de/user/login?entryPoint=log[...]	63	Benign	0.925	0.8	0.882	0.905	0.941
https://www.reddit.com/login/	61	Benign	0.984	0.889	1.0	1.0	0.812
https://www.twitch.tv/	61	Benign	-	0.632	1.0	1.0	0.8
https://www.facebook.com/login/device-based/r[...]	82	Benign	-	0.8	0.941	0.857	0.789
https://www.dropbox.com/login	81	Benign	-	0.92	0.882	1.0	0.944
https://214.156.43.197/login.live.com/	82	IP	0.817	0.92	0.882	0.905	1.0
https://b1ovam5.org/otto.de/	63	Path	0.938	0.96	1.0	0.952	0.941
https://uyvgo8i.net/RsHZdqidvhidpFbRVa/accoun[...]	82	Random	0.866	0.76	0.882	0.952	0.895
https://store.steamposered.com/login/	45	RD	0.289	0.412	0.571	0.667	0.556
https://mircosoft.com/login.srf?wa=wsignin1.0[...]	82	RD	0.354	0.56	0.647	0.714	0.684
https://v-k.com/	17	RD	0.471	0.5	1.0	0.6	0.75
https://sso.immoblienscout24.de/sso/login	79	RD	0.494	0.667	0.647	0.8	0.667
https://meine.deutsche-bank.online/trxm/db/	62	RD	0.62	0.76	0.588	0.5	0.882
https://amazon-secureserver.de/ap/signin?open[...]	82	RD	0.671	0.84	0.706	0.81	0.842
https://wwwcommerzbank.de/lp/login	80	RD	0.812	0.96	1.0	0.9	0.778
https://www.netflix.com-co.support/login	82	Subdomain	0.598	0.8	0.882	0.476	0.737
https://ebay.de.login.9ontzcjkgj2k.ru/ws/eBay[...]	81	Subdomain	0.815	0.76	0.941	0.905	0.944
https://gmx.net%6B%73%35%66%6C%6A%33%[...]	76	URL encoding	0.855	0.875	0.875	0.895	0.941
https://www.45m64or.ru/NZYJolaEiBSOSOCoWm/sso[...]	82	Random	-	0.88	1.0	1.0	0.947
https://www.fodus.de/ajax/login/	75	RD	-	0.333	0.647	0.5	0.625
https://meine.deutssche-bank.de/?client_id=H[...]	62	RD	-	1.0	0.882	0.95	0.882
https://www.commerzbank.de-account.support/de-en/	80	Subdomain	-	0.84	0.706	0.55	0.778
https://login.live.com.id.online/de/login.exe[...]	82	Subdomain	-	0.92	0.882	0.905	0.895
https://idealo.de%76%73%6C%38%6A%6D%31[...]	68	URL encoding	-	0.87	0.8	0.938	1.0

Appendix Table A.9.: Absolute (and relative) results of the Recognition of Services questionnaire

Service	Used	Known	Unknown
Amazon	124 (93.23%)	9 (6.77%)	0
Commerzbank	20 (15.04%)	109 (81.96%)	4 (3.01%)
Deutsche Bank	16 (12.03%)	113 (84.96%)	4 (3.01%)
Dropbox	96 (72.18%)	35 (26.32%)	2 (1.51%)
eBay	90 (67.67%)	43 (32.33%)	0
eBay Kleinanzeigen	106 (79.70%)	27 (20.30%)	0
Facebook	106 (79.70%)	27 (20.30%)	0
FOCUS	16 (12.03%)	105 (78.95%)	12 (9.02%)
GMX	39 (29.32%)	81 (60.90%)	13 (9.77%)
iCloud	53 (39.85%)	75 (56.39%)	5 (3.76%)
ImmobilienScout24	49 (36.84%)	80 (60.15%)	4 (3.01%)
Microsoft	104 (78.20%)	29 (21.81%)	0
Netflix	108 (81.20%)	25 (18.80%)	0
OTTO	42 (31.58%)	87 (65.41%)	4 (3.01%)
PayPal	113 (84.96%)	20 (15.04%)	0
Reddit	27 (20.30%)	71 (53.38%)	35 (26.32%)
Steam	27 (20.30%)	52 (39.10%)	54 (40.60%)
Twitch	19 (14.27%)	77 (57.90%)	37 (27.82%)
VK	3 (2.26%)	23 (17.29%)	107 (80.45%)
WEB.DE	43 (32.33%)	71 (53.38%)	19 (14.29%)
YouTube	121 (90.98%)	12 (9.02%)	0

Appendix Table A.10.: Demographics questionnaire including answer types and options

Question	Answer type	Answer options
What is your gender?	single-choice	Female; Male; Diverse; No answer
How old are you?	single-choice	14 or younger; 15-19; 20-24; 25-29; 30-34; 35-39; 40 and older; No answer
What is your highest degree?	single-choice	No school degree; Middle school; High school graduate, diploma or the equivalent; Vocational Training; Bachelor's degree; Master's degree; Diploma; Doctorate degree; Other; No answer
Did you participate in Computer Science classes (e.g. in school or university)?	single-choice	No Computer Science classes; Less than 6 months; 6 to 12 months; 1 to 2 years; More than 2 years; No answer
How would you rate your prior knowledge in the following topics? (Computer Science | IT-Security | Phishing)	6-point Likert scale	None; Very little; Little; Some; Much; Very much

Appendix Table A.11.: Phishing URL categories presented in the analysis, decision or personalized (A) or creation (C) games. Indicates whether a category is included (y) for categories which are not differentiated in the games, whether it does not appear (n), or how the category is referred to in the games in all other cases.

	Category	Sub-category	A	C
RD Base	Generic		n	y
	Random		y	y
	URL Encoding	(any)	n	n
	IP Address		IP	n
	Modified TLD		n	RD
Placement	Full path posing	Path posing	Path	Path
		Query posing	n	n
	Subdomain posing	first/only	Subdomain	Subdomain
		middle/last	n	n
	Http credentials		n	n
	RD		y	y
e2LD Modification	Combosquatting	(any)	RD	RD
	Typosquatting	(any)	RD	RD
	IDN		n	n
Random			Random	n
Benign			No-Phish	n

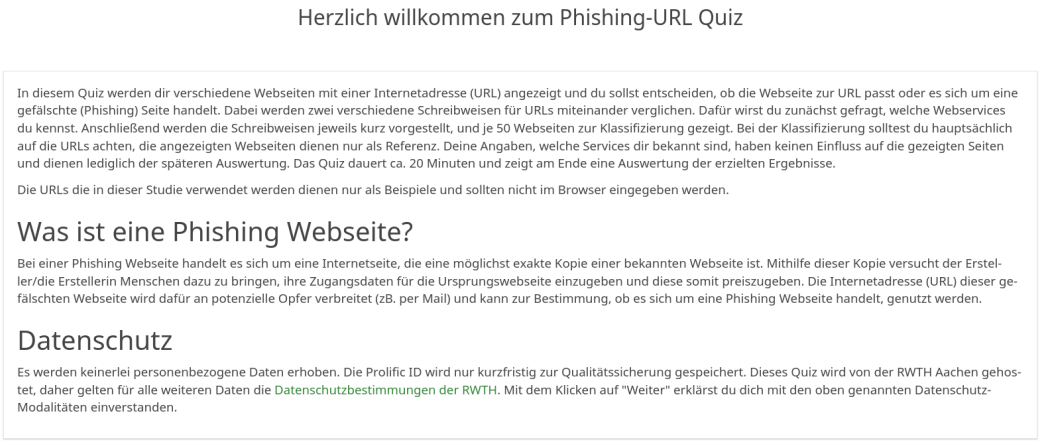


Figure A.5.: Welcome screen of the URL notation study.

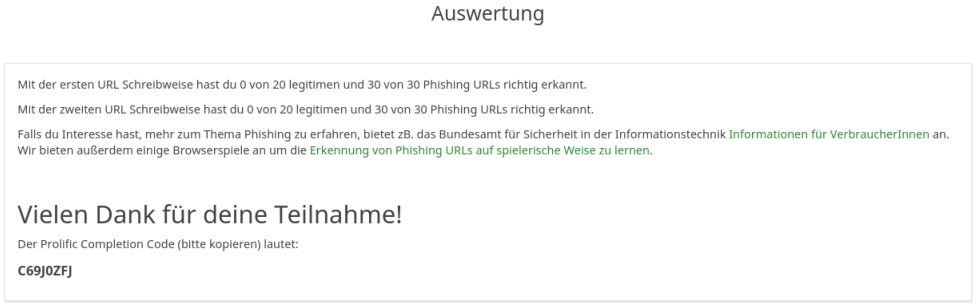


Figure A.6.: Feedback screen of the URL notation study.

A.4. Reverse Domain Name Notation

This appendix includes additional information on the user study comparing different URL notations in Chapter 7. Figures A.5, A.6, A.7, A.8, and A.9 depict screenshots from the survey. Tables A.12 and A.13 show the classification performances for the URLs in the user study.

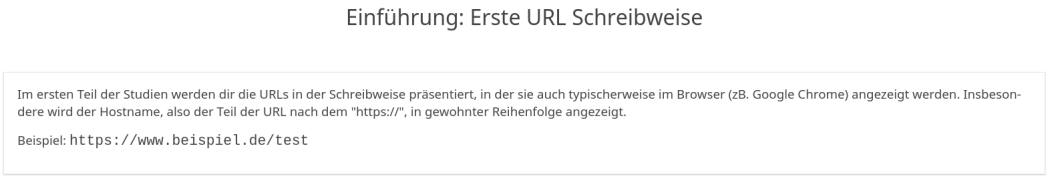


Figure A.7.: Introduction to normal URL notation.

Figure A.8.: Introduction to RDN URL notation.

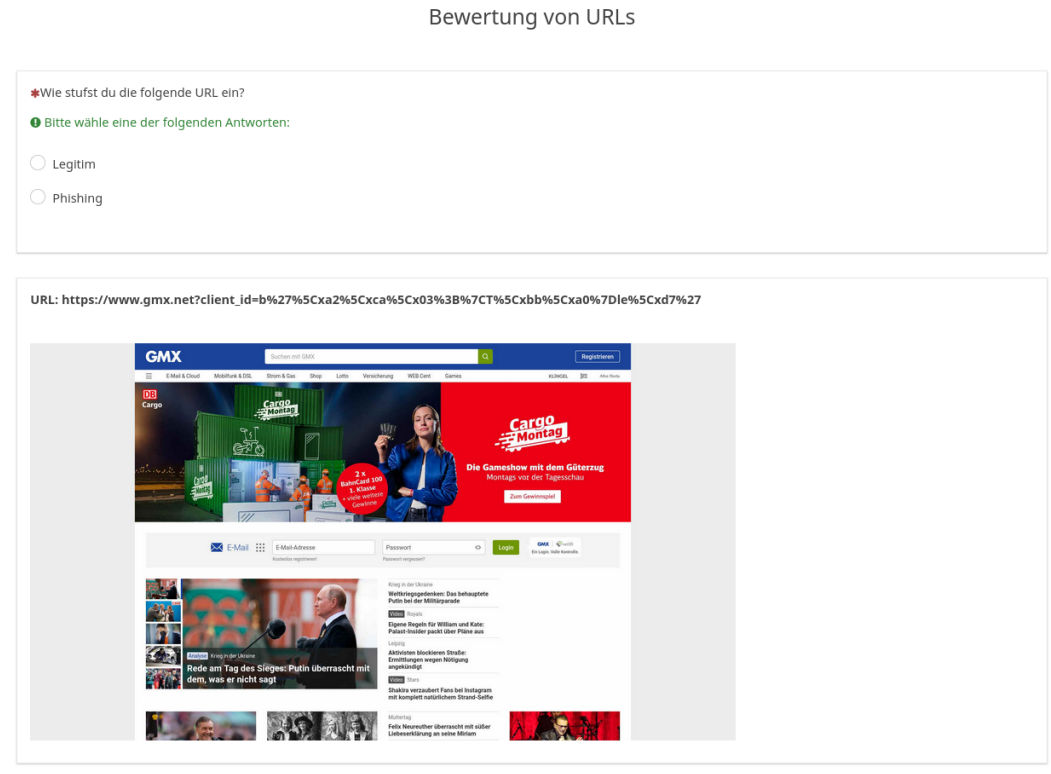


Figure A.9.: Example URL and website of classification task in URL notation study.

Appendix Table A.12.: Performances for benign and phishing URLs in normal URL notation (URLs of unfamiliar services were removed).

URL	N	Category	Mean
https://www.samsung.com	45	Plain	1.0
https://www.amazon.com	47	Plain	1.0
https://www.facebook.com	47	Plain	1.0
https://www.chip.de	26	Plain	1.0
https://www.merkur.de	17	Plain	1.0
https://www.chefkoch.de?request=9683977234168182615	18	Path	0.944
https://www.chefkoch.de?client_id=3256578733479199268	18	Path	0.889
https://www.instagram.com?client_id=9148891928820696079	47	Path	0.872
https://www.ebay-kleinanzeigen.de	38	Plain	0.842
https://www.twitch.tv?client_id=b%27%5Cxaa%5Cxd6%5[...]	41	Path	0.829
https://www.gmx.net?client_id=b%22%5Cx9e%5Cx95%5Cx[...]	22	Path	0.818
https://www.gmx.net?client_id=b%27%5Cxa2%5Cxca%5Cx[...]	22	Path	0.818
https://www.chip.de/client_id?a=1895698834972220894	26	Path	0.808
https://www.instagram.com/client_id?a=b%27Z%5Cxc8%5[...]	47	Path	0.787
https://www.amazon.com?request=9171341737121226924	47	Path	0.766
https://www.paypal.com?request=b%27%5Cxf9%5Cx15%5Cn[...]	47	Path	0.766
https://www.paypal.com?request=b%27KJyV%5Cxcf%5Cx17...]	47	Path	0.745
https://www.tagesschau.de?request=3792775142391940030	31	Path	0.742
https://www.merkur.de/client_id?a=b%27%5Cxfa %5Cxdd[...]	17	Path	0.706
https://www.sport1.de?request=b%27%5Cxbe%5Cxca%5Cxd[...]	22	Path	0.636
https://www.lebay-kleinanzeigen.de?request=b%27%5Cxb0[...]	38	Typo	1.0
https://9mbsfotkhtljpfu4.com/www.google.com	47	Path	1.0
https://www.gosogle.com?request=b%27%5Cx9f%5Cxde%5C[...]	47	Typo	1.0
https://vsotqst5rn2e7e84.com?www.spiegel.de	36	Path	1.0
https://7g40m8niwctnsyevnw8e3o6zg1i1p8bwcaxrrf4zwp.8ttd[...]	31	Subdomain-end	1.0
https://www.wlet.de?client_id=5902123609469213206	32	Typo	1.0
https://www.y0utube.com?client_id=4691860970233155429	47	Typo	0.979
https://www.reddti.com	47	Typo	0.957
https://www.wikipedio.org?request=b%27G%5Cx93%7D+%5[...]	47	Typo	0.957
https://www.wikipediaa.org?request=2927723380819768289	47	Typo	0.957
https://0i17460ibjn9bwwjoqk5e2q9h283p8gef3sdd0yr.qq3dlm[...]	47	Subdomain-end	0.957
https://www.sporz1.de?request=8254018817974639890	22	Typo	0.909
https://www.agesschau.de	31	Typo	0.903
https://www.foocus.de	29	Typo	0.897
https://www.pypal.com?client_id=b%27o%7C%5Cx1d+%5C[...]	47	Typo	0.851
https://www.wikipeia.org?request=3842376134860118984	47	Typo	0.83
https://www.vveb.de	26	Typo	0.808
https://www.otto.site	31	RD	0.774
https://www.wikipedia.link?request=b%27%5Cxc8%5Cx87...]	47	RD	0.766
https://www.eb.de	26	Typo	0.731
https://www.sport1.de.2bstrc6zl31d61aq91z6ff06zg84kiw2o[...]	22	Subdomain-first	0.727
https://www.twitter.com@vgjdjepskv4l0f8l.com?client_id[...]	46	Http credentials	0.717
https://www.link-tagesschau.de?request=b%27%2A5%5Cx1[...]	31	RD	0.71
https://www.google-help.com	47	RD	0.702
https://www.google.com.support-page.com?request=b%27A2[...]	47	Subdomain-first	0.681
https://www.otto.de.i3qblme04koxujzd.de?client_id=2369[...]	31	Subdomain-first	0.645
https://www.derwesten-secure.de?request=b%27%5Cxfe0%[...]	13	RD	0.615
https://www.idealo.de.%6b%32%79%72%62%2e%61%72	23	Subdomain-first	0.609
https://www.ebay.de@%37%76%6b%32%65%2e%61%72?re[...]	47	Http credentials	0.596
https://www.youtube.com%2e%77%32%37%37%73%2e%61%72	47	Subdomain-first	0.404

Appendix Table A.13.: Performances for benign and phishing URLs in RDN notation (URLs of unfamiliar services were removed).

URL	N	Category	Mean
https://de.dhl.www	47	Plain	0.979
https://de.welt.www	32	Plain	0.969
https://de.chefkoch.www	18	Plain	0.944
https://tv.twitch.www	41	Plain	0.927
https://de.web.www	26	Plain	0.923
https://de.derwesten.www	13	Plain	0.846
https://de.t-online.www?request=1510910431402647394	30	Path	0.833
https://de.merkur.www/request?a=8801701589660449956	17	Path	0.824
https://de.web.www?client_id=3415526714293118578	26	Path	0.808
https://de.spiegel.www?request=2948715188917027892	36	Path	0.778
https://de.chip.www?request=7734971300542504185	26	Path	0.769
https://com.amazon.www?request=b%27%5Cxea%5Cx13%5Cx[...]	47	Path	0.766
https://com.paypal.www?client_id=b%27c%5Cxaa%2A%5C[...]	47	Path	0.745
https://de.idealo.www/request?a=b%27%5Cxe6%5Cxc0%5C[...]	23	Path	0.739
https://com.instagram.www?request=b%27%5Cx9f A%5E %[...]	47	Path	0.723
https://com.paypal.www?client_id=b%27%5Cx92G%605%5[...]	47	Path	0.723
https://com.samsung.www?request=b%27%5Cn%5Cxe4%7B3Z[...]	45	Path	0.711
https://de.idealo.www?request=7097990411080245371	23	Path	0.696
https://de.derwesten.www?client_id=b%27%5Cxd9Tc%5Cx[...]	13	Path	0.692
https://de.ebay-kleinanzeigen.www/client_id?a=b%27%5[...]	38	Path	0.658
https://%78%67%37%75%35%2e%61%72.de.sport1.www?[...]	22	Subdomain-first	1.0
https://%39%6e%66%6e%77%2e%61%72.enw00zx2ewzmwc[...]	36	Subdomain-end	1.0
https://de.7xl3qpajwpdvayfi.5dfjrqwskixsq2mvc88jq9nphhn[...]	47	Subdomain-end	1.0
https://de.iidealo.www?request=b%27%5Cxa1%5Cxb1%5Cx[...]	23	Typo	1.0
https://net.gax.www?request=4137666067858483598	22	Typo	1.0
https://de.hcip.www?request=b%27%28%5Cx0ei%5Cx90%5[...]	26	Typo	1.0
https://de.eba.www	47	Typo	0.979
https://de.ebacy.www	47	Typo	0.979
https://de.ott0.www?client_id=4594320839498719144	31	Typo	0.968
https://de.kbay.www	47	Typo	0.957
https://de.ebay-kleinanze1gen.www?request=b%27%5Cxe9...]	38	Typo	0.947
https://%79%78%77%72%66%2e%61%72.28gxxbedqgryvm[...]	47	Subdomain-first	0.936
https://de.ves9wi3uhkhrz5v2.de.derwesten.www	13	Subdomain-first	0.923
https://com.k3qglq6stzqw8ggh/www.ebay-kleinanzeigen.de	38	Path	0.895
https://de.i4r915ew8gb8aouj?www.ebay.de	47	Path	0.894
https://de.ideao.www?client_id=b%27%5Cxa4%5Cxc5%60[...]	23	Typo	0.87
https://de%2e%71%75%63%66%73%2e%61%72.derweste[...]	13	Subdomain-first	0.846
https://com.googe.www	47	Typo	0.809
https://com.safe-samsung.www	45	RD	0.8
https://de.t-onlne.www	30	Typo	0.8
https://com.instaagram.www?client_id=7958564013988669330	47	Typo	0.745
https://com.help-reddit.www?request=b%27%5Cxb2%5Cx87%27	47	RD	0.723
https://page.tagesschau.www?request=b%27%5Cxc8%5Cx87[...]	31	RD	0.71
https://com.link-support.de.dhl.www?client_id=b%27%5[...]	47	RD	0.681
https://de.focus-link.www?client_id=849977719853577737	29	RD	0.655
https://de.idealo.www@de.0z3tu4tsiuet7wdt?request=b%27[...]	23	Http credentials	0.652
https://de.t-onilne.www	30	Typo	0.633
https://de.ebay-kleinanzeigens.www?request=644325675462[...]	38	Typo	0.632
https://secure.chefkoch.www	18	RD	0.611
https://com.amazon.www@%61%72%2e%76%69%77%38%34[...]	47	Http credentials	0.489

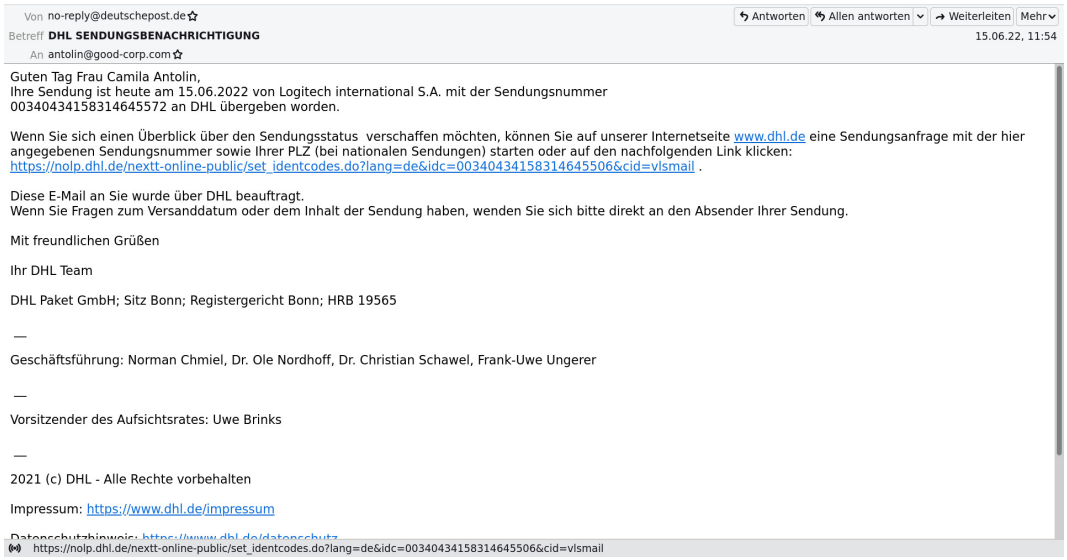


Figure A.10.: Example email screenshot for the benign subscription service (b-1) category.

A.5. Anti-Phishing Design Interventions for Email Clients

This appendix includes additional information on the email UIs analyzed in Chapter 8. Figures A.10, A.11, A.12, A.13, A.14, A.15, and A.16 depict example email screenshots that were shown in the user study for each of the three benign and four phishing email categories. Figure A.17 shows the introduction to the survey, with Figures A.18 and A.19 depicting examples for the UI introductions and email classification task. Table A.14 presents the mean performance scores for all emails that were shown in the survey. Table A.15 includes the statements for the feedback part of the questionnaire.

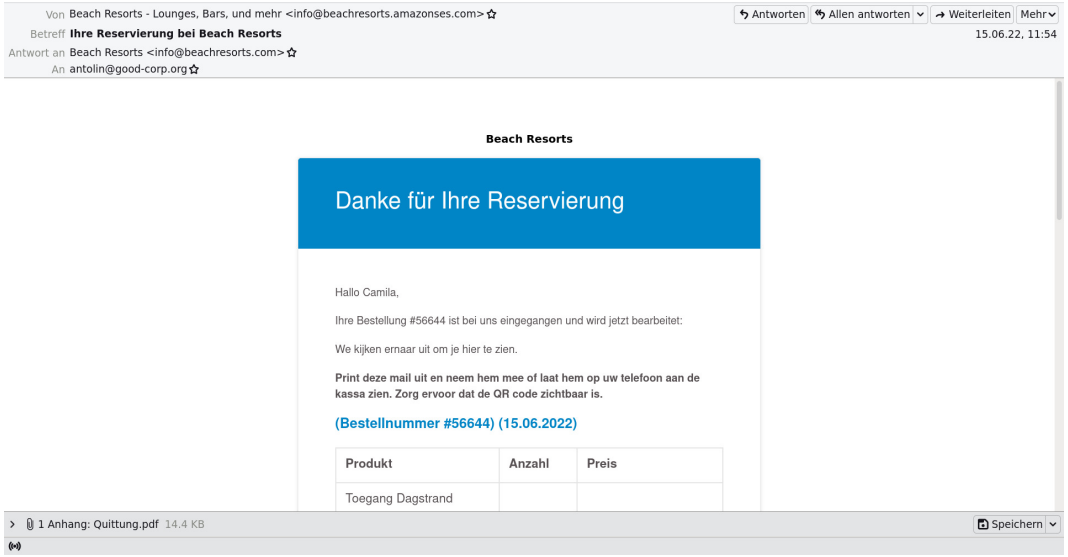


Figure A.11.: Example email screenshot for the benign fictional service (b-2) category.

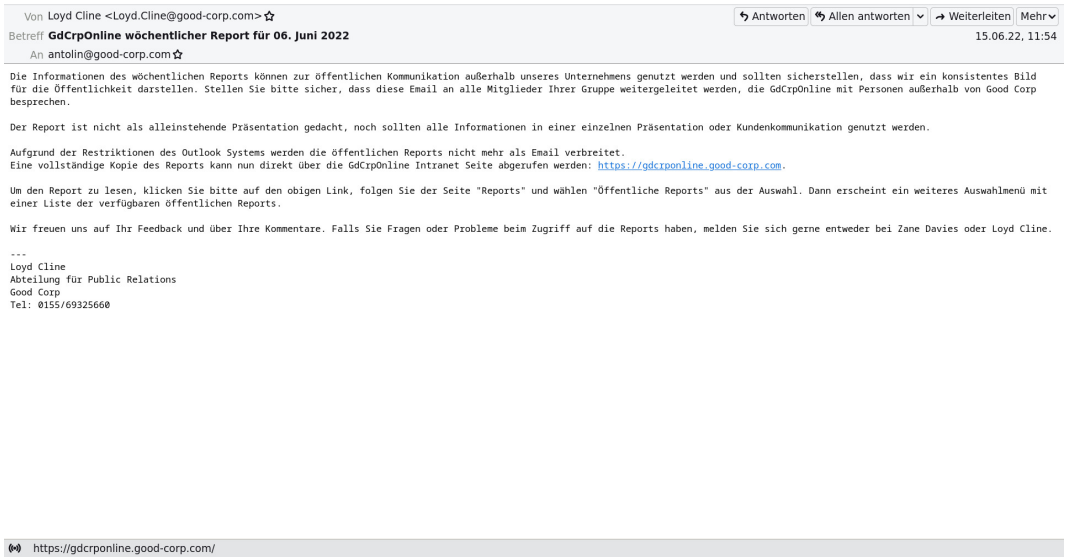


Figure A.12.: Example email screenshot for the benign company (b-3) category.

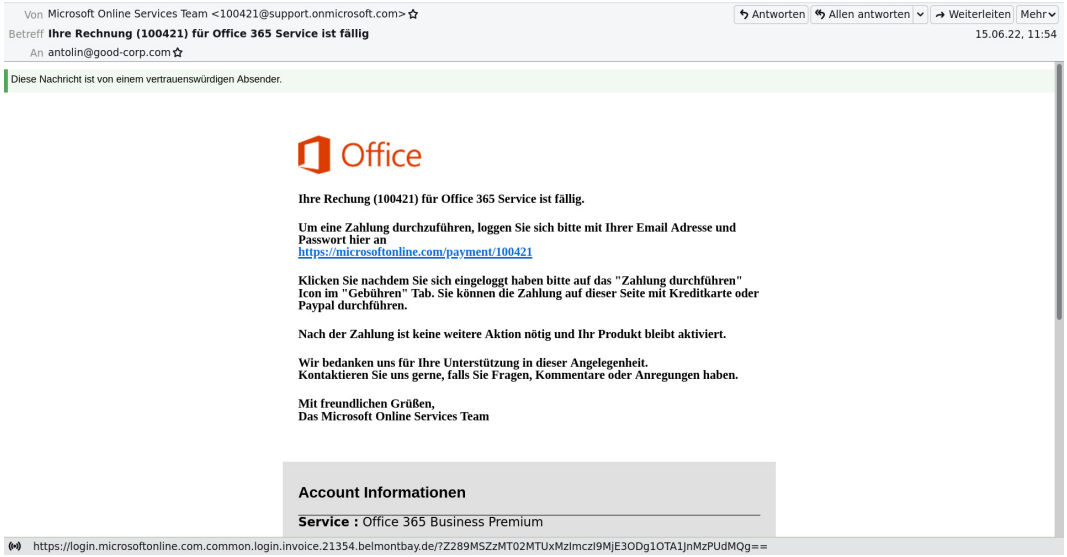


Figure A.13.: Example email screenshot for the mass phishing (p-1) category.

Figure A.14.: Example email screenshot for the spear phishing (p-2) category.

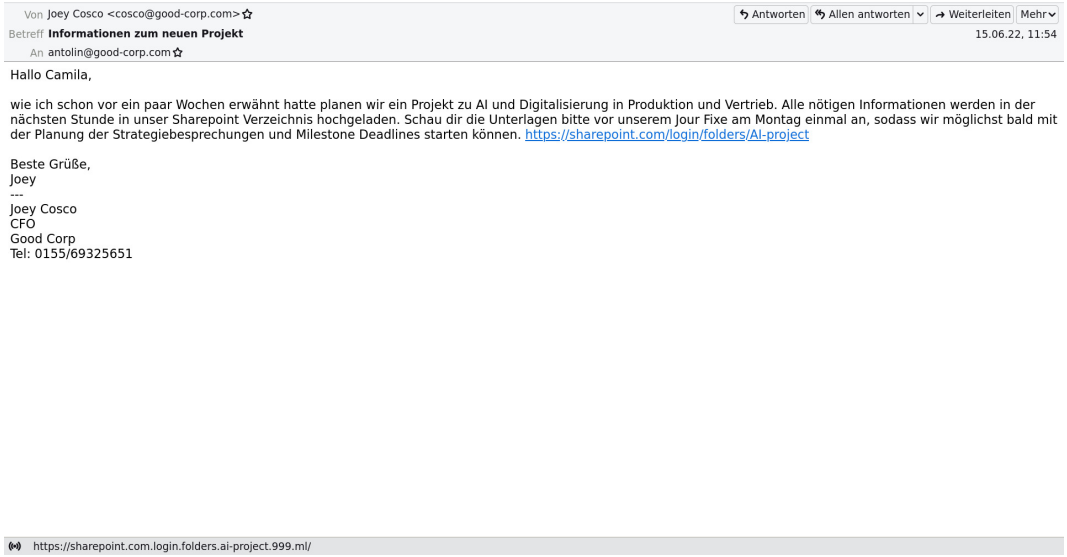


Figure A.15.: Example email screenshot for the lateral phishing (p-3) category.

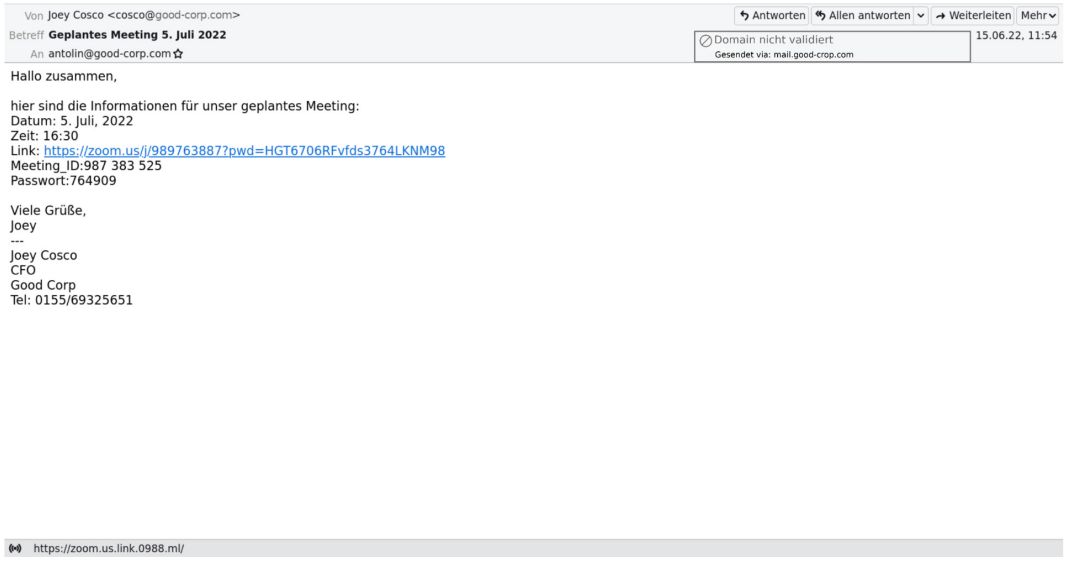


Figure A.16.: Example email screenshot for the phishing (p-4) category that only appears in the spoofing UI.

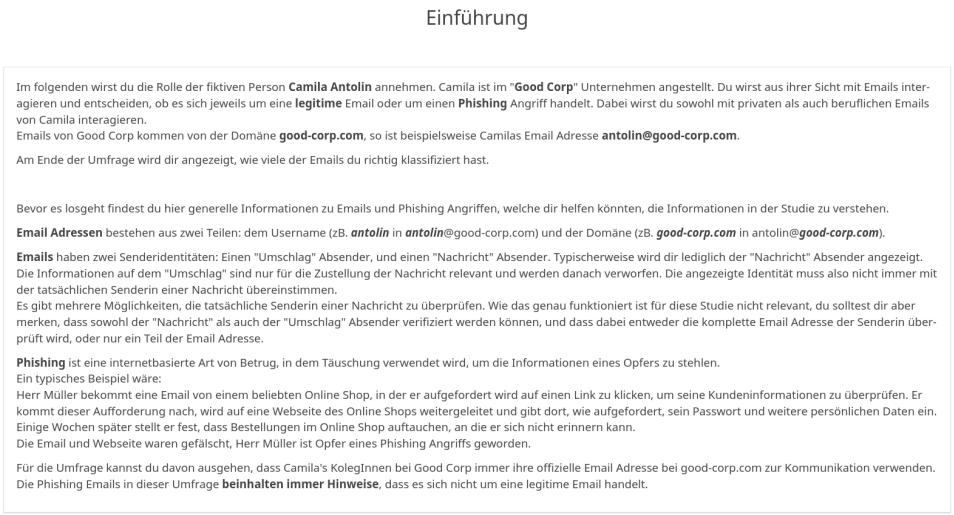


Figure A.17.: Introduction text in email survey.

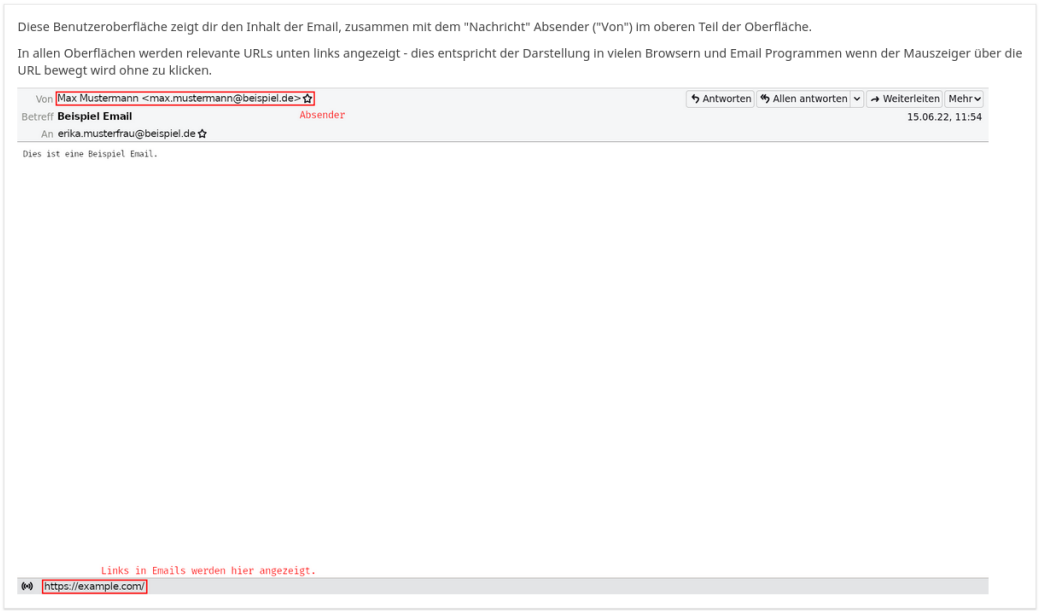


Figure A.18.: Introduction text and image for the plain UI.

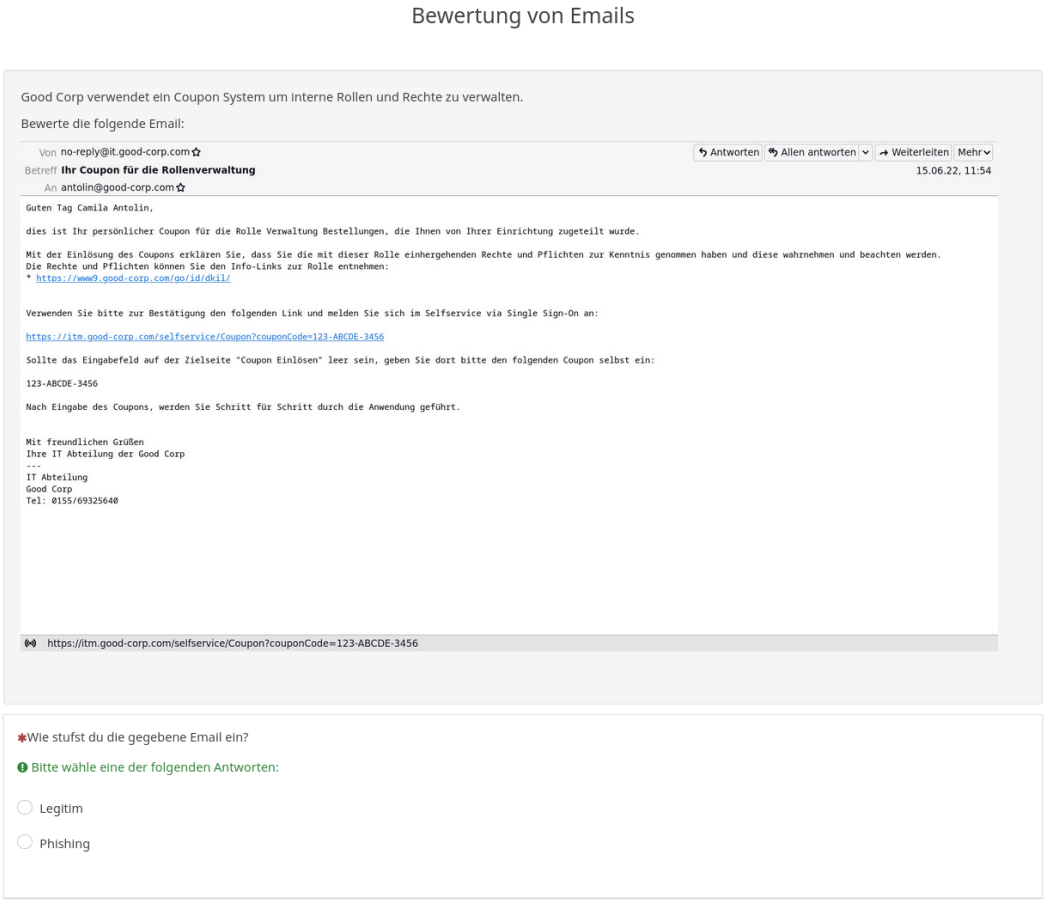


Figure A.19.: Example for the classification task for emails.

Appendix Table A.14.: Mean performances for emails for the different UIs.

Service	Malicious Indicator	UI	N	Category	Mean
DHL	None	Plain	50	b-1	0.64
Fictional	None	Plain	50	b-2	0.64
Company	None	Plain	50	b-3	0.62
Microsoft	From, URL	Plain	50	p-1	0.78
Company	From, URL	Plain	50	p-2	0.52
Sharepoint	URL	Plain	50	p-3	0.1
Spotify	None	History	50	b-1	0.74
Fictional	None	History	50	b-2	0.84
Company	None	History	50	b-3	0.94
Company	None	History	50	b-3	0.82
Dhl	From, URL	History	50	p-1	0.74
Company	From	History	50	p-2	0.54
Paypal	URL	History	50	p-3	0.18
Zoom	URL	History	50	p-3	0.06
Spotify	None	Highlighting	50	b-1	0.92
Deutsche Bahn	None	Highlighting	50	b-1	0.88
Fictional	None	Highlighting	50	b-2	0.44
Company	None	Highlighting	50	b-3	0.7
PayPal	From, URL	Highlighting	50	p-1	0.94
Volksbank	From, URL	Highlighting	50	p-1	0.86
Microsoft	From, URL	Highlighting	50	p-2	0.78
Zoom	URL	Highlighting	50	p-3	0.1
Google	None	Spoofing	50	b-1	0.94
PayPal	None	Spoofing	50	b-1	0.88
Fictional	None	Spoofing	50	b-2	0.68
Company	None	Spoofing	50	b-3	0.92
Postbank	From, URL	Spoofing	50	p-1	0.7
Company	From, URL	Spoofing	50	p-2	0.44
Sharepoint	URL	Spoofing	50	p-3	0.04
Zoom	UI, URL	Spoofing	50	p-4	0.34

Appendix Table A.15.: Feedback for different UIs. Answer options ranged from 1 - “trifft gar nicht zu” to 6 - “trifft voll und ganz zu”.

Statement
Das gegebene UI erleichtert es mir, Phishing Emails zu erkennen.
Die Oberfläche zeigt überflüssige Informationen an.
Die Oberfläche hilft mir, legitime Emails zu erkennen.
Ich würde die Oberfläche für meine Emails nutzen.
Die in der Oberfläche angezeigten Informationen sind verständlich.
In der Oberfläche fehlen essenzielle Informationen.

Appendix Table A.16.: Number of benign certificates for the 15 most popular issuers

Issuer `CN`	Count
COMODO ECC Domain Validation Secure Server CA 2	7189
Let's Encrypt Authority X3	6854
COMODO RSA Domain Validation Secure Server CA	4027
CloudFlare Inc ECC CA-2	2564
Amazon	1908
DigiCert SHA2 Secure Server CA	1744
Go Daddy Secure Certificate Authority - G2	1722
GeoTrust RSA CA 2018	1426
RapidSSL RSA CA 2018	1015
DigiCert SHA2 Extended Validation Server CA	1001
GlobalSign Organization Validation CA - SHA256 - G2	825
GlobalSign CloudSSL CA - SHA256 - G3	624
cPanel, Inc. Certification Authority	612
DigiCert SHA2 High Assurance Server CA	571
COMODO RSA Organization Validation Secure Server CA	523

A.6. Analyzing Certificates of Phishing Websites

This appendix includes additional information on the analysis of certificates of phishing and benign websites performed in Chapter 9. Tables A.16 and A.17 depict the most commonly used issuers of certificates in our benign and phishing datasets.

Appendix Table A.17.: Number of phishing certificates for the 15 most popular issuers

Issuer `CN`	Count
Let's Encrypt Authority X3	3259
cPanel, Inc. Certification Authority	2103
RapidSSL TLS RSA CA G1	862
COMODO RSA Domain Validation Secure Server CA	502
COMODO ECC Domain Validation Secure Server CA 2	489
CloudFlare Inc ECC CA-2	474
DigiCert SHA2 Secure Server CA	321
Go Daddy Secure Certificate Authority - G2	272
Google Internet Authority G3	188
RapidSSL RSA CA 2018	128
Microsoft IT TLS CA 1	88
GlobalSign CloudSSL CA - SHA256 - G3	74
Actalis Domain Validation Server CA G1	70
Amazon	63
DigiCert SHA2 High Assurance Server CA	58

A.7. Automated Phishing Detection Using CT Log Analysis

This appendix includes additional information on the automated detection of certificates of phishing websites performed in Chapter 10. Table A.18 includes all keywords that were used as features. Tables A.19 and A.20 presents all features used in the classification task.

Appendix Table A.18.: Full list of words used as keyword-features

secure	login	mail	account	online
support	sites	services	service	docs
update	signin	info	security	help
verify	recovery	mobile	secureserver	storage
center	verification	auth	promo	free
paypal	runescape	google	apple	jppost
sharepoint	sagawa	appleid	amazon	icloud
windows	office	facebook	1drv	live
onedrive	ebay	allegro	itau	bankofamerica
cartetitolari	viabcp			

Appendix Table A.19.: Part 1 of extracted certificate and domain feature values for a benign (c_0) and a phishing certificate (c_1). CN_{c_0} = *anycast.ftl.netflix.com*, CN_{c_1} = *paypal-secured.ga*. Features selected during feature selection are marked. Categorical features: *issuer*, *key_algorithm*

#	Feature	Selected	Type	Output	$\mathcal{F}(\mathbf{c_0})$	$\mathcal{F}(\mathbf{c_1})$
1	is_ov		certificate	binary	1	0
2	is_ev	✓	certificate	binary	0	0
3	is_dv		certificate	binary	0	1
4	sub_has_c		certificate	binary	1	0
5	sub_has_st		certificate	binary	1	0
6	sub_has_l		certificate	binary	1	0
7	sub_only_cn	✓	certificate	binary	0	1
8	sub_has_cn		certificate	binary	1	1
9	sub_dn_count		certificate	integer	6	1
10	sub_char_count	✓	certificate	integer	64	17
11	sub_ext_count		certificate	integer	10	9
12	valid_period	✓	certificate	integer	36	90
13	policies_count		certificate	integer	2	2
14	is_wildcard		certificate	binary	1	0
15	has_ocsp		certificate	binary	1	1
16	has_cdp	✓	certificate	binary	1	0
17	san_count	✓	certificate	integer	7	2
18	average_sd_count	✓	certificate	rational	4.14286	2.50000
19	san_tld_count	✓	certificate	integer	2	1
20	key_algorithm		certificate	integer	2	1
21	key_size		certificate	integer	256	2048
22	issuer	✓	certificate	integer	0	1
23	sub_cn_entropy	✓	domain	rational	2.54753	2.47625
24	sub_cn_is_com	✓	domain	binary	1	0
25	name_san_entropy	✓	domain	rational	0.24027	0.08737
26	has_uppercase_letters		domain	binary	0	0
27	num_dash	✓	domain	integer	0	1
28	num_dash_rd	✓	domain	integer	0	1
29	num_tokens	✓	domain	integer	4	3
30	tld_in_token	✓	domain	binary	1	0
31	https_in_domain		domain	binary	0	0
32	longest_token	✓	domain	integer	7	7
33	special_char_ratio	✓	domain	rational	0.13043	0.11765
34	is_ip		domain	binary	0	0
35	is_idn_domain	✓	domain	binary	0	0

Appendix Table A.20.: Part 2 of extracted certificate and domain feature values for a benign (c_0) and a phishing certificate (c_1)

#	Feature	Selected	Type	Output	$\mathcal{F}(\mathbf{c_0})$	$\mathcal{F}(\mathbf{c_1})$
36	san_to_alexa_entropy	✓	domain	rational	0.57761	0.74982
37	vowel_ratio	✓	domain	rational	0.23529	0.38462
38	digit_ratio	✓	domain	rational	0.00000	0.00000
39	length	✓	domain	integer	23	17
40	contains_wwwdot	✓	domain	binary	0	0
41	contains_subdomain_of_only_digits		domain	binary	0	0
42	subdomain_lengths_mean	✓	domain	rational	5.66667	14.00000
43	parts	✓	domain	integer	3	1
44	contains_digits	✓	domain	binary	0	0
45	has_valid_tld		domain	binary	1	1
46	contains_one_char_subdomains	✓	domain	binary	0	0
47	prefix_repetition		domain	binary	0	0
48	char_diversity	✓	domain	rational	0.64706	0.78571
49	contains_tld_as_infix	✓	domain	binary	1	0
50	alphabet_size	✓	domain	integer	11	11
51	shannon_entropy	✓	domain	rational	3.33718	3.37878
52	hex_part_ratio	✓	domain	rational	0.00000	0.00000
53	underscore_ratio		domain	rational	0.00000	0.00000
54	ratio_of_repeated_chars	✓	domain	rational	0.45455	0.27273
55	consecutive_consonant_ratio	✓	domain	rational	0.64706	0.14286
56	consecutive_digits_ratio	✓	domain	rational	0.00000	0.00000
57	1_gram_std	✓	domain	rational	0.65555	0.44536
58	1_gram_median	✓	domain	integer	1	1
59	1_gram_mean	✓	domain	rational	1.54545	1.27273
60	1_gram_min		domain	integer	1	1
61	1_gram_max	✓	domain	integer	3	2
62	1_gram_bottom_quartile	✓	domain	rational	1.00000	1.00000
63	1_gram_top_quartile	✓	domain	rational	2.00000	1.50000
64	2_gram_std	✓	domain	rational	0.24944	0.27639
65	2_gram_median		domain	integer	1	1
66	2_gram_mean	✓	domain	rational	1.06667	1.08333
67	2_gram_min		domain	integer	1	1
68	2_gram_max	✓	domain	integer	2	2
69	2_gram_bottom_quartile		domain	rational	1.00000	1.00000
70	2_gram_top_quartile	✓	domain	rational	1.00000	1.00000
71	3_gram_std	✓	domain	rational	0.00000	0.00000
72	3_gram_median		domain	integer	1	1
73	3_gram_mean	✓	domain	rational	1.00000	1.00000
74	3_gram_min		domain	integer	1	1
75	3_gram_max	✓	domain	integer	1	1
76	3_gram_bottom_quartile		domain	rational	1.00000	1.00000
77	3_gram_top_quartile	✓	domain	rational	1.00000	1.00000

Appendix B

Statement of Originality

The research presented in this thesis was conducted at the *Research Group IT-Security*, RWTH Aachen, which is led by Ulrike Meyer. Furthermore, the anti-phishing learning games presented in Chapter 6 are the result of a close collaborative effort with the *Learning Technologies Research Group*, RWTH Aachen, conducted in the context of the research training group "Human Centered Systems Security" sponsored by the state of North Rhine-Westphalia. In her role as doctoral supervisor, Ulrike Meyer provided guidance, feedback, and support for all publications, as well as the overall doctoral process. The publications were also positively affected by the discussions provided by other members of the *Research Group IT-Security*, and the members of the "Human Centered Systems Security" research training group. Finally, Ulrik Schroeder and Martin Wolf provided feedback and ideas in the context of the ERBSE project as part of the research training group. Due to the high degree of collaboration in some of the works presented in this thesis, the following list provides more details on the specific contributions of the authors to the relevant publications.

- **Finding Phish in a Haystack: A Pipeline for Phishing Classification on Certificate Transparency Logs** [1]. Arthur Drichel, Vincent Drury, Justus von Brandt, and Ulrike Meyer.
  This paper presents the CT log detection pipeline and its first evaluation which were adapted in Chapter 10. It was created in close collaboration with Arthur Drichel, with equal contribution from both of us. This includes the design of the pipeline, as well as the evaluation setup and result discussion, while the implementation was heavily supported by Justus von Brandt.

- **Dating Phish: An Analysis of the Life Cycles of Phishing Attacks and Campaigns** [2]. Vincent Drury, Luisa Lux, and Ulrike Meyer.
  This paper presents several analyses to determine the age and duration of phishing attacks and campaigns. It is the result of collaborative work with Luisa Lux, where my main contributions are supervising and supporting the design and implementation of the data collection and analysis functionality. I also took the lead in the analysis and discussion of the resulting dataset, as well as writing the paper.

- **Certified Phishing: Taking a Look at Public Key Certificates of Phishing Websites** [3]. Vincent Drury and Ulrike Meyer.

This paper presents the analysis of certificates which was adapted in Chapter 9. My main contributions to this paper are the design and implementation of the certificate collection process, as well as the analysis and discussion of the resulting dataset.

- **No Phishing With the Wrong Bait: Reducing the Phishing Risk by Address Separation** [4]. Vincent Drury and Ulrike Meyer.
  This paper presents email address separation as an anti-phishing technique which aims to prevent attackers from obtaining relevant email addresses. My main contributions to this paper are the design of the approach and its theoretical analysis and discussion based on pre-defined scenarios and known email-based phishing attacks.

- **Analyzing and Creating Malicious URLs: A Comparative Study on Anti-Phishing Learning Games** [5]. Vincent Drury, René Röpke, Ulrik Schroeder, and Ulrike Meyer.
  This paper presents the comparison of three anti-phishing learning games, which was adapted in Chapter 6. It is the result of close collaborative work with René Röpke, where we both contributed equally. As such, designing and implementing the games, as well as the design, evaluation, and discussion of the user study, are joint work.

- **Exploring Different Game Mechanics for Anti-Phishing Learning Games** [6]. René Röpke, Vincent Drury, Ulrike Meyer, and Ulrik Schroeder.
  This paper introduces two of the anti-phishing learning games presented in Chapter 6. It is based on collaborative work with René Röpke, where we both contributed equally to the design and implementation of the learning games, as well as the preliminary study presented in the paper, while René Röpke took the lead in writing the paper.

- **Exploring and Evaluating Different Game Mechanics for Anti-Phishing Learning Games** [7]. René Röpke, Vincent Drury, Ulrike Meyer, and Ulrik Schroeder.
  This paper extends the previous paper by an analysis of three anti-phishing learning games, including the retention test which was adapted in Chapter 6. Similar to the previous paper, the design and execution of the user study, as well as its evaluation and discussion, where collaborative work with René Röpke, where we contributed equally, while René Röpke took the lead in writing the paper.

- **More Than Meets the Eye – An Anti-Phishing Learning Game with a Focus on Phishing Emails** [8]. René Röpke, Vincent Jakob Drury, Philipp Peess, Tobias Johnen, Ulrike Meyer, and Ulrik Schroeder.
  This paper presents an anti-phishing learning game with a focus on emails, which requires players to analyze and construct phishing emails. My main contributions to this paper are the supervision of design and implementation of the game, as well as contributing to writing the paper.

- **A modular architecture for personalized learning content in anti-phishing learning games** [9]. René Röpke, Vincent Jakob Drury, Ulrik Schroeder, and Ulrike Meyer.

This paper presents the implementation of a personalization architecture for anti-phishing learning games. It is the result of collaborative work with René Röpke, where my main contributions are participation in the design and implementation of the pipeline, as well as contributing to writing the paper.

- **Better the Phish You Know: Evaluating Personalization in Anti-Phishing Learning Games** [10]. René Röpke, Vincent Jakob Drury, Ulrike Meyer, and Ulrik Schroeder.
  This paper introduces the personalized game presented in Chapter 6, together with an evaluation comparing it to the analysis game. The paper is the result of collaborative work with René Röpke, in which we both contributed equally. This includes the design, evaluation, and discussion of the corresponding user study, as well as writing the paper with equal contribution.

- **A Pond Full of Phishing Games - Analysis of Learning Games for Anti-Phishing Education** [11]. René Röpke, Klemens Koehler, Vincent Drury, Ulrik Schroeder, Martin R. Wolf, and Ulrike Meyer.
  This paper presents the systematic literature review and comparison of state-of-the-art anti-phishing learning games which was partly summarized and modified for Section 6.1. It is collaborative work with René Röpke and Klemens Köhler, where my main contribution was the analysis of game content. Writing the paper was joint work, where we each contributed equally.

- **Towards personalized game-based learning in anti-phishing education** [12]. René Röpke, Ulrik Schroeder, Vincent Jakob Drury, and Ulrike Meyer.
  This paper presents a concept of a personalization architecture for anti-phishing learning games. It is collaborative work with René Röpke, where my main contributions are collaborating on the design of the pipeline, as well as contributing to writing the paper.

List of Figures and Tables

List of Figures

List of Tables

Bibliography

[AAC15] Mohamed Alsharnouby, Furkan Alaca, and Sonia Chiasson. "Why Phishing Still Works: User Strategies for Combating Phishing Attacks". In: *Human-Computer Studies* 82 (2015).

[ABP18] Dina Aladawy, Kristian Beckers, and Sebastian Pape. "PERSUADED: Fighting Social Engineering Attacks with a Serious Game". In: *Trust, Privacy and Security in Digital Business*. Springer, 2018.

[Abr+21] Hossein Abroshan, Jan Devos, Geert Poels, and Eric Laermans. "Phishing happens beyond technology: the effects of human behaviors and demographics on each step of a phishing process". In: *IEEE Access* 9 (2021).

[Alk+21] Zainab Alkhalil, Chaminda Hewage, Liqaa Nawaf, and Imtiaz Khan. "Phishing attacks: A recent comprehensive study and a new anatomy". In: *Frontiers in Computer Science* 3 (2021).

[ALK14] Eric Amankwa, Marianne Loock, and Elmarie Kritzinger. "A conceptual analysis of information security education, information security training and information security awareness definitions". In: *The 9th International Conference for Internet Technology and Secured Transactions (ICITST 2014)*. IEEE, 2014.

[ALM15] Nalin Asanka Gamagedara Arachchilage, Steve Love, and Carsten Maple. "Can a Mobile Game Teach Computer Users to Thwart Phishing Attacks?" In: *arXiv preprint arXiv:1511.01622* (2015).

[Alq+20] Fatima Alqubaisi, Ahmad Samer Wazan, Liza Ahmad, and David W Chadwick. "Should we rush to implement password-less single factor fido2 based authentication?" In: *2020 12th Annual Undergraduate Research Conference on Applied Computing (URC)*. IEEE, 2020.

[And+15] Bonnie Brinton Anderson, C Brock Kirwan, Jeffrey L Jenkins, David Eargle, Seth Howard, and Anthony Vance. "How polymorphic warnings reduce habituation in the brain: Insights from an fMRI study". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2015.

[APW21] APWG. *APWG Phishing Activity Trends Report, 2nd Quarter 2021*. Tech. rep. Anti-Phishing Working Group, 2021. URL: `https://docs.apwg.org/reports/apwg_trends_report_q2_2021.pdf`.

[APW22] APWG. *APWG Phishing Activity Trends Report, 3rd Quarter 2022*. Tech. rep. Anti-Phishing Working Group, 2022. URL: https://docs.apwg.org/reports/apwg_trends_report_q3_2022.pdf.

[AVW20] Sara Albakry, Kami Vaniea, and Maria K Wolters. "What is this URL's destination? Empirical evaluation of users' URL reading". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2020.

[AZ17] Ahmed Aleroud and Lina Zhou. "Phishing environments, techniques, and countermeasures: A survey". In: *Computers & Security* 68 (2017).

[BA19] Gitanjali Baral and Nalin Asanka Gamagedara Arachchilage. "Building Confidence not to be Phished Through a Gamified Approach: Conceptualising User's Self-Efficacy in Phishing Threat Avoidance Behaviour". In: *Cybersecurity and Cyberforensics Conf. (CCC)*. IEEE, 2019.

[Bah+17] Alejandro Correa Bahnsen, Eduardo Contreras Bohorquez, Sergio Villegas, Javier Vargas, and Fabio A González. "Classifying phishing URLs using recurrent neural networks". In: *2017 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2017.

[Bau+17] Gerhild Bauer, Daniel Martinek, Simone Kriglstein, Günter Wallner, and Rebbeca Wölfle. "Digital game-based learning with "Internet Hero": a game about the internet for children aged 9–12 years". In: *Context Matters!* New Academic Press, 2017.

[BC14] Clemens Bergmann and Gamze Canova. "Design, Implementation and Evaluation of an Anti-Phishing Education App". MA thesis. Technische Universität Darmstadt, 2014.

[BC16] M. Baslyman and S. Chiasson. ""Smells Phishy?": An educational game about online phishing scams". In: *2016 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2016.

[Bha19] Jagjot Bhardwaj. "Design of a Game for Cybersecurity Awareness". MA thesis. North Dakota State University, 2019.

[BJC19] AJ Burns, M Eric Johnson, and Deanna D Caputo. "Spear phishing in a barrel: Insights from a targeted phishing campaign". In: *Journal of Organizational Computing and Electronic Commerce* 29.1 (2019).

[BP16] K. Beckers and S. Pape. "A Serious Game for Eliciting Social Engineering Security Requirements". In: *Int. Requirements Engineering Conf. (RE)*. IEEE, 2016.

[BPF16] Kristian Beckers, Sebastian Pape, and Veronika Fries. "HATCH: Hack and Trick Capricious Humans - a Serious Game on Social Engineering". In: *Int. BCS Human Computer Interaction Conf.: Companion Volume*. HCI '16. BCS Learning & Development Ltd., 2016.

[Cal+07] Jon Callas, Lutz Donnerhacke, Hal Finney, David Shaw, and Rodney Thayer. *OpenPGP Message Format*. Tech. rep. RFC4880, 2007. URL: https://tools.ietf.org/html/rfc4880.

[Can+15] Gamze Canova, Melanie Volkamer, Clemens Bergmann, and Benjamin Reinheimer. "NoPhish App Evaluation: Lab and Retention Study". In: *Proceedings of the Workshop on Usable Security and Privacy (USEC 2015)*. Internet Society, 2015.

[CHK11] Dave Crocker, Tony Hansen, and Murray S. Kucherawy. *DomainKeys Identified Mail (DKIM) Signatures*. Tech. rep. RFC6376, 2011. URL: `https://tools.ietf.org/html/rfc6376`.

[Cia06] Robert B Cialdini. *Influence: The Psychology of Persuasion, Revised Edition*. New York: William Morrow, 2006.

[CJ+18] Gokul CJ, Sankalp Pandit, Sukanya Vaddepalli, Harshal Tupsamudre, Vijayanand Banahatti, and Sachin Lodha. "PHISHY - A Serious Game to Train Enterprise Users on Phishing Awareness". In: *Annual Symp. on Computer-Human Interaction in Play Companion Extended Abstracts*. ACM, 2018.

[CMB11] Sonia Chiasson, Manas Modi, and Robert Biddle. "Auction Hero: The Design of a Game to Learn and Teach about Computer Security". In: *E-Learn: World Conf. on E-Learning in Corporate, Government, Healthcare, and Higher Education 2011*. AACE, 2011.

[Con+07] Benjamin D Cone, Cynthia E Irvine, Michael F Thompson, and Thuy D Nguyen. "A video game for cyber security training and awareness". Englisch. In: *Computers & Security* 26.1 (2007).

[Cra08] Lorrie F Cranor. "A framework for reasoning about the human in the loop". In: *Usability, Psychology, and Security 2008 (UPSEC '08)*. USENIX, 2008.

[Cro09] Dave Crocker. *Internet Mail Architecture*. Tech. rep. RFC5598, 2009. URL: `https://tools.ietf.org/html/rfc5598`.

[Cuc+19] Tom Cuchta, Brian Blackwood, Thomas R Devine, Robert J Niichel, Kristina M Daniels, Caleb H Lutjens, Sydney Maibach, and Ryan J Stephenson. "Human risk factors in cybersecurity". In: *Proceedings of the 20th annual SIG conference on information technology education*. ACM, 2019.

[Dai04] Leslie Daigle. *WHOIS Protocol Specification*. Tech. rep. RFC3912, 2004. URL: `https://tools.ietf.org/html/rfc3912`.

[Dam64] Fred J Damerau. "A technique for computer detection and correction of spelling errors". In: *Communications of the ACM* 7.3 (1964).

[Das+19] Avisha Das, Shahryar Baki, Ayman El Aassal, Rakesh Verma, and Arthur Dunbar. "SoK: a comprehensive reexamination of phishing research from the security perspective". In: *Communications Surveys & Tutorials* 22.1 (2019).

[DD10] Patrick Dwyer and Zhenhai Duan. "MDMap: Assisting users in identifying phishing emails". In: *Proceedings of 7th annual collaboration, ELECTRONIC messaging, Anti-ABUSE and spam conference (CEAS)*. Citeseer, 2010.

[DD23] Vincent Drury and Arthur Drichel. *Phishing URL and Certificate Datasets*. 2023. URL: `osf.io/dky2m`.

[De +21] Ravindu De Silva, Mohamed Nabeel, Charith Elvitigala, Issa Khalil, Ting Yu, and Chamath Keppitiyagama. "Compromised or Attacker-Owned: A Large Scale Classification and Study of Hosting Domains of Malicious URLs". In: *Proceedings of the 30th USENIX Security Symposium*. USENIX, 2021.

[DG06] Jesse Davis and Mark Goadrich. "The Relationship between Precision-Recall and ROC Curves". In: *International Conference on Machine Learning*. ACM, 2006.

[DHC07] Julie S Downs, Mandy Holbrook, and Lorrie Faith Cranor. "Behavioral response to phishing risk". In: *2007 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE. 2007.

[Don+15] Zheng Dong, Apu Kapadia, Jim Blythe, and L. Jean Camp. "Beyond the lock icon: real-time detection of phishing websites using public key certificates". In: *2015 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2015.

[Dre22] Jakob Drees. "Towards a More Granular Categorization of Phishing URLs to Improve Anti-Phishing Education". MA thesis. RWTH Aachen University, 2022.

[Dri+20] Arthur Drichel, Ulrike Meyer, Samuel Schüppen, and Dominik Teubert. "Analyzing the Real-World Applicability of DGA Classifiers". In: *Proceedings of the 15th International Conference on Availability, Reliability and Security*. ACM, 2020.

[Dru23] Vincent Drury. *Data of Phishing URL Categories and Reverse Domain Name (RDN) Studies*. 2023. URL: `osf.io/q563m`.

[DTH06] Rachna Dhamija, J. D. Tygar, and Marti Hearst. "Why Phishing Works". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2006.

[ES18] Yahia Elsayed and Ahmed Shosha. "Large scale detection of IDN domain name masquerading". In: *2018 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2018.

[Fel+15] Adrienne Porter Felt, Alex Ainslie, Robert W Reeder, Sunny Consolvo, Somas Thyagaraja, Alan Bettes, Helen Harris, and Jeff Grimes. "Improving SSL warnings: Comprehension and adherence". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2015.

[FEP19] Edona Fasllija, Hasan Ferit Enişer, and Bernd Prünster. "Phish-Hook: Detecting Phishing Certificates Using Certificate Transparency Logs". In: *International Conference on Security and Privacy in Communication Systems*. Springer, 2019.

[FMS19] Dorota Filipczuk, Charles Mason, and Stephen Snow. "Using a Game to Explore Notions of Responsibility for Cyber Security in Organisations". In: *Extended Abstracts of the 2019 CHI Conf. on Human Factors in Computing Systems*. ACM, 2019.

[For18] CA Browser Forum. *EV SSL Certificate Guidelines 1.6.8*. `https://cabforum.org/extended-validation/`. 2018.

[For19] CA Browser Forum. *CA-Browser Forum BR 1.6.3*. `https://cabforum.org/baseline-requirements-documents/`. 2019.

[Fra+21] Anjuli Franz, Verena Zimmermann, Gregor Albrecht, Katrin Hartwig, Christian Reuter, Alexander Benlian, and Joachim Vogt. "SoK: Still Plenty of Phish in the Sea—A Taxonomy of User-Oriented Phishing Interventions and Avenues for Future Research". In: *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. USENIX, 2021.

[Fre+19] S. Frey, A. Rashid, P. Anthonysamy, M. Pinto-Albuquerque, and S. A. Naqvi. "The Good, the Bad and the Ugly: A Study of Security Decisions in a Cyber-Physical Systems Game". In: *Transactions on Software Engineering* 45.5 (2019).

[Gar+07] Sujata Garera, Niels Provos, Monica Chew, and Aviel D Rubin. "A framework for detection and measurement of phishing attacks". In: *Proceedings of the 2007 workshop on Recurring malcode*. ACM, 2007.

[Gey19] Julian Geywitz. ""What the Hack?" - Konzeption und Implementierung eines erweiterbaren und adaptiven Serious Game zur Verbesserung von Information Security Awareness". MA thesis. University of Applied Sciences, 2019.

[GKG15] Filippos Giannakas, Georgios Kambourakis, and Stefanos Gritzalis. "CyberAware: A mobile game-based app for cybersecurity education and awareness". In: *Int. Conf. on Interactive Mobile Communication Technologies and Learning (IMCL)*. IEEE, 2015.

[Goo23] Google. *Google Xenon 2020-2023 certificate transparency logs*. https://ct.googleapis.com/logs/xenon2020/ https://ct.googleapis.com/logs/xenon2021/ https://ct.googleapis.com/logs/xenon2022/ https://ct.googleapis.com/logs/xenon2023/. 2023.

[Gop+21] Shakthidhar Gopavaram, Jayati Dev, Marthie Grobler, DongInn Kim, Sanchari Das, and L Jean Camp. "Cross-National Study on Phishing Resilience". In: *Proceedings of the Workshop on Usable Security and Privacy (USEC 2021)*. Internet Society, 2021.

[GP13] Mark Gondree and Zachary N. J. Peterson. "Valuing Security by Getting [d0x3d!]: Experiences with a Network Security Board Game". In: *Workshop on Cyber Security Experimentation and Test (CSET)*. USENIX, 2013.

[Heb+17] Alexander James Hebert, Chandler O Reynolds, Kyle Joseph Stack, and Rosemary Celeste Lindsay. *Lock_Out: A Cybersecurity MQP and Game*. Englisch. Final Report. Worcester Polytechnic Institute, 2017.

[HGG15] Matthew L. Hale, Rose F. Gamble, and Philip Gamble. "CyberPhishing: A Game-Based Platform for Phishing Awareness Testing". In: *Hawaii Int. Conf. on System Sciences*. Vol. 48. IEEE, 2015.

[HKB16] X. Han, N. Kheir, and D. Balzarotti. "PhishEye: Live Monitoring of Sandboxed Phishing Kits". In: *Proceedings of the 2016 SIGSAC Conference on Computer and Communications Security (CCS '16)*. ACM, 2016, pp. 1402–1413.

[Ho+19] Grant Ho, Asaf Cidon, Lior Gavish, Marco Schweighauser, Vern Paxson, Stefan Savage, Geoffrey M Voelker, and David A Wagner. "Detecting and characterizing lateral phishing at scale". In: *Proceedings of the 28th Usenix Security Symposium*. USENIX, 2019.

[Hu+21] Hang Hu, Steve TK Jan, Yang Wang, and Gang Wang. "Assessing Browser-level Defense against IDN-based Phishing." In: *Proceedings of the 30th USENIX Security Symposium*. USENIX, 2021.

[Huy+17] Duy Huynh, Phuc Luong, Hiroyuki Iida, and Razvan Beuran. "Design and Evaluation of a Cybersecurity Awareness Training Game". In: *Entertainment Computing – ICEC 2017*. Springer, 2017.

[HW18] Hang Hu and Gang Wang. "End-to-end measurements of email spoofing attacks". In: *Proceedings of the 27th USENIX Security Symposium*. USENIX, 2018.

[Inv21] Federal Bureau of Investigation. *Internet Crime Report 2021*. Tech. rep. 2021. URL: `https://www.ic3.gov/Media/PDF/AnnualReport/2021_IC3Report.pdf`.

[Jac+07] Collin Jackson, Daniel R Simon, Desney S Tan, and Adam Barth. "An evaluation of extended validation and picture-in-picture phishing attacks". In: *Financial Cryptography and Data Security: 11th International Conference (FC 2007)*. Springer, 2007.

[Joh22] Tobias Johnen. "Analysis and Detection of Impostor Domains". MA thesis. RWTH Aachen University, 2022.

[JR16] S Carolin Jeeva and Elijah Blessing Rajsingh. "Intelligent phishing url detection using association rule mining". In: *Human-centric Computing and Information Sciences* 6.1 (2016).

[Kas22] Kaspersky. *Spam and phishing in 2021*. Tech. rep. Kaspersky, 2022. URL: `https://securelist.com/spam-and-phishing-in-2021/105713/`.

[Kat+17] E. Katsadouros, D. Kogias, L. Toumanidis, C. Chatzigeorgiou, and C. Z. Patrikakis. "Teaching network security through a scavenger hunt game". In: *Global Engineering Education Conf. (EDUCON)*. IEEE, 2017.

[Kim+22] Taeri Kim, Noseong Park, Jiwon Hong, and Sang-Wook Kim. "Phishing URL Detection: A Network-based Approach Robust to Evasion". In: *Proceedings of the 2022 SIGSAC Conference on Computer and Communications Security*. ACM, 2022.

[Kit14] Scott Kitterman. *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1*. Tech. rep. RFC7208, 2014. URL: `https://tools.ietf.org/html/rfc7208`.

[Kle08] John C. Klensin. *Simple Mail Transfer Protocol*. Tech. rep. RFC5321, 2008. URL: `https://tools.ietf.org/html/rfc5321`.

[Kra02] David R. Krathwohl. "A Revision of Bloom's Taxonomy: An Overview". In: *Theory Into Practice* 41.4 (2002).

[Kuc19] Murray S. Kucherawy. *Message Header Field for Indicating Message Authentication Status*. Tech. rep. RFC8601, 2019. URL: `https://tools.ietf.org/html/rfc8601`.

[Kul19] Vikas Krishnarao Kulkarni. "Basic Cybersecurity Awareness Through Gaming". MA thesis. North Dakota State University, 2019.

[Kum+08] Ponnurangam Kumaraguru, Steve Sheng, Alessandro Acquisti, Lorrie Faith Cranor, and Jason Hong. "Lessons from a real world evaluation of anti-phishing training". In: *2008 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2008.

[Kum+09] Ponnurangam Kumaraguru, Justin Cranshaw, Alessandro Acquisti, Lorrie Cranor, Jason Hong, Mary Ann Blair, and Theodore Pham. "School of phish: a real-world evaluation of anti-phishing training". In: *Fifth Symposium on Usable Privacy and Security (SOUPS 2009)*. USENIX, 2009.

[Kun+16] Alexandra Kunz, Melanie Volkamer, Simon Stockhardt, Sven Palberg, Tessa Lottermann, and Eric Piegert. "Nophish: evaluation of a web application that teaches people being aware of phishing attacks". In: *Informatik 2016* (2016).

[Kun+21] Johannes Kunke, Stephan Wiefling, Markus Ullmann, and Luigi Lo Iacono. "Evaluation of account recovery strategies with FIDO2-based passwordless authentication". In: *arXiv preprint arXiv:2105.12477* (2021).

[KW18] Johannes A König and Martin R Wolf. "GHOST: An Evaluated Competence Developing Game for Cybersecurity Awareness Training". In: *Advances in Security* 11.3 & 4 (2018).

[KZ15] Murray S. Kucherawy and Elizabeth Zwicky. *Domain-based Message Authentication, Reporting, and Conformance (DMARC)*. Tech. rep. RFC7489, 2015. URL: `https://tools.ietf.org/html/rfc7489`.

[Las+21] Leona Lassak, Annika Hildebrandt, Maximilian Golla, and Blase Ur. "" It's Stored, Hopefully, on an Encrypted Server": Mitigating Users' Misconceptions About FIDO2 Biometric WebAuthn." In: *Proceedings of the 30th USENIX Security Symposium*. USENIX, 2021.

[Las14] Elmer EH Lastdrager. "Achieving a consensual definition of phishing based on a systematic review of the literature". In: *Crime Science* 3.1 (2014).

[Le +19] Sophie Le Page, Guy-Vincent Jourdan, Gregor V. Bochmann, Iosif-Viorel Onut, and Jason Flood. "Domain Classifier: Compromised Machines Versus Malicious Registrations". In: *International Conference on Web Engineering (ICWE)*. Springer, 2019.

[Li+19] Bingyu Li, Jingqiang Lin, Fengjun Li, Qiongxiao Wang, Qi Li, Jiwu Jing, and Congli Wang. "Certificate transparency in the wild: Exploring the reliability of monitors". In: *Proceedings of the 2019 SIGSAC Conference on Computer and Communications Security (CCS '19)*. ACM, 2019.

[Lin+11] Eric Lin, Saul Greenberg, Eileah Trotter, David Ma, and John Aycock. "Does domain highlighting help people identify phishing sites?" In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011.

[Lin+21] Yun Lin, Ruofan Liu, Dinil Mon Divakaran, Jun Yang Ng, Qing Zhou Chan, Yiwen Lu, Yuxuan Si, Fan Zhang, and Jin Song Dong. "Phishpedia: a hybrid deep learning based approach to visually identify phishing webpages". In: *Proceedings of the 30th USENIX Security Symposium*. USENIX, 2021.

[LJ19] Sophie Le Page and Guy-Vincent Jourdan. "Victim or Attacker? A Multi-dataset Domain Classification of Phishing Attacks". In: *International Conference on Privacy, Security and Trust*. IEEE, 2019.

[LLK13] Ben Laurie, Adam Langley, and Emilia Kasper. *Certificate Transparency*. Tech. rep. RFC6962, 2013. URL: `https://tools.ietf.org/html/rfc6962`.

[Lop+18] Inês Lopes, Yuliya Morenets, Pedro R. M. Inácio, and Frutuoso Silva. "Cyber-Detective: a game for cyber crime prevention". In: *Play2Learn*. 2018.

[Lou+13] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. "Understanding variable importances in forests of randomized trees". In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2013.

[Lu18] Yang Lu. "CyberCraft, a security serious game". MA thesis. Politecnico di Torino, 2018.

[Lya+20] Sanam Ghorbani Lyastani, Michael Schilling, Michaela Neumayr, Michael Backes, and Sven Bugiel. "Is FIDO2 the kingslayer of user authentication? A comparative usability study of FIDO2 passwordless authentication". In: *2020 Symposium on Security and Privacy (SP)*. IEEE, 2020.

[MAB17] G. Misra, N. A. G. Arachchilage, and S. Berkovsky. "Phish phinder: a game design approach to enhance user confidence in mitigating phishing attacks". In: *Int. Symp. on Human Aspects of Information Security & Assurance (HAISA 2017)*. DBLP computer science bibliography, 2017.

[Mal16] Malwarebytes Labs. *Petya- Taking Ransomware To The Low Level*. `https://blog.malwarebytes.com/threat-analysis/2016/04/petya-ransomware/`. Accessed February 27, 2020. 2016.

[Mar+20] Sourena Maroofi, Maciej Korczyński, Cristian Hesselman, Benoît Ampeau, and Andrzej Duda. "COMAR: Classification of Compromised versus Maliciously Registered Domains". In: *2020 European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2020.

[Mat+21] Tenga Matsuura, Ayako A Hasegawa, Mitsuaki Akiyama, and Tatsuya Mori. "Careless Participants Are Essential for Our Phishing Study: Understanding the Impact of Screening Methods". In: *Proceedings of the 2021 European Symposium on Usable Security*. ACM, 2021.

[MCS09] T. Moore, R. Clayton, and H. Stern. "Temporal Correlations between Spam and Phishing Websites". In: *Proceedings of the 2nd Workshop on Large-Scale Exploits and Emergent Threats (LEET'09)*. USENIX, 2009.

[MG08] D. K. McGrath and M. Gupta. "Behind Phishing: An Examination of Phisher Modi Operandi". In: *Proceedings of the 1st Workshop on Large-Scale Exploits and Emergent Threats (LEET'08)*. USENIX, 2008.

[MNS10] Thomas Monk, Johan van Niekerk, and Rossouw von Solms. "Sweetening the Medicine: Educating Users about Information Security by Means of Game Play". In: *Annual Research Conf. of the South African Institute of Computer Scientists and Information Technologists*. ACM, 2010.

[Moc87] Paul Mockapetris. *Domain Names - Implementation and Specification*. Tech. rep. RFC1035, 1987. URL: `https://tools.ietf.org/html/rfc1035`.

[MPJ18] Josephina Mikka-Muntuumo, Anicia Peters, and Hussin Jazri. "Cyber-Bullet - Share Your Story: An Interactive Game for Stimulating Awareness on the Harm and Negative Effects of the Internet". In: *African Conf. for Human Computer Interaction: Thriving Communities*. ACM, 2018.

[MTM14] Rami M Mohammad, Fadi Thabtah, and Lee McCluskey. "Predicting phishing websites based on self-structuring neural network". In: *Neural Computing and Applications* 25.2 (2014).

[MX22] Rosana Montanẽz Rodriguez and Shouhuai Xu. "Cyber Social Engineering Kill Chain". In: *International Conference on Science of Cyber Security*. Springer, 2022.

[NAN08] Boris New, Verónica Araújo, and Thierry Nazzi. "Differential processing of consonants and vowels in lexical access through reading". In: *Psychological Science* 19.12 (2008).

[NCB17] James Nicholson, Lynne Coventry, and Pam Briggs. "Can we fight social engineering attacks by social means? Assessing social salience as a means to improve phish detection". In: *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*. USENIX, 2017.

[Oes+18] Adam Oest, Yeganeh Safei, Adam Doupé, Gail-Joon Ahn, Brad Wardman, and Gary Warner. "Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis". In: *2018 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2018.

[Oes+19] Adam Oest, Yeganeh Safaei, Adam Doupé, Gail-Joon Ahn, Brad Wardman, and Kevin Tyers. "Phishfarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists". In: *2019 Symposium on Security and Privacy (SP)*. IEEE, 2019.

[Oes+20a] A. Oest, P. Zhang, B. Wardman, E. Nunes, J. Burgis, A. Zand, K. Thomas, A. Doupé, and G. Ahn. "Sunrise to Sunset: Analyzing the End-to-end Life Cycle and Effectiveness of Phishing Attacks at Scale". In: *Proceedings of the 29th USENIX Security Symposium*. USENIX, 2020.

[Oes+20b] Adam Oest, Yeganeh Safaei, Penghui Zhang, Brad Wardman, Kevin Tyers, Yan Shoshitaishvili, Adam Doupé, and Gail-Joon Ahn. "Phishtime: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists". In: *Proceedings of the 29th USENIX Security Symposium*. USENIX, 2020.

[Ola+14] Marc Olano et al. "SecurityEmpire: Development and Evaluation of a Digital Game to Promote Cybersecurity Education". In: *Summit on Gaming, Games, and Gamification in Security Education*. USENIX, 2014.

[Oli+17] Daniela Oliveira et al. "Dissecting Spear Phishing Emails for Older vs Young Adults: On the Interplay of Weapons of Influence and Life Domains in Predicting Susceptibility to Phishing". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2017.

[OZ15] Abdus-Samad Temitope Olanrewaju and Nur Haryani Zakaria. "Social engineering awareness game (SEAG): an empirical evaluation of using game towards improving information security awareness". Englisch. In: *Int. Conf. on Computing and Informatics*. 2015.

[PHK15] Jan L. Plass, Bruce D. Homer, and Charles K. Kinzer. "Foundations of Game-Based Learning". In: *Educational Psychologist* 50.4 (2015).

[Ram04] Blake Ramsdell. *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification*. Tech. rep. RFC3851, 2004. URL: `https://tools.ietf.org/html/rfc3851`.

[RD23] René Röpke and Vincent Drury. *Evaluation Data of Anti-Phishing Learning Games*. 2023. URL: `osf.io/3ydw9`.

[Res00] Eric Rescorla. *Http over tls, RFC 2818*. Tech. rep. RFC2818, 2000. URL: `https://tools.ietf.org/html/rfc2818`.

[Res08] Paul Resnick. *Internet message format*. Tech. rep. RFC5322, 2008. URL: `https://tools.ietf.org/html/rfc5322`.

[Rey+20] Joshua Reynolds, Deepak Kumar, Zane Ma, Rohan Subramanian, Meishan Wu, Martin Shelton, Joshua Mason, Emily Stark, and Michael Bailey. "Measuring identity confusion with uniform resource locators". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2020.

[RL16] Andreas Rieb and Ulrike Lechner. "Operation Digital Chameleon: Towards an Open Cybersecurity Method". In: *Int. Symp. on Open Collaboration*. ACM, 2016.

[Rob+] Richard Roberts, Rachel Walter, Daniela Lulli, and Dave Levin. ".how .you .spot .whoswho .online .sucks: Deceiving Users with Generic Top-Level Domains". In: ().

[Rob+19] Richard Roberts, Yaelle Goldschlag, Rachel Walter, Taejoong Chung, Alan Mislove, and Dave Levin. "You are who you appear to be: A longitudinal study of domain impersonation in tls certificates". In: *Proceedings of the 2019 SIGSAC Conference on Computer and Communications Security (CCS '19)*. ACM, 2019.

[Sah+19] Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, and Banu Diri. "Machine learning based phishing detection from URLs". In: *Expert Systems with Applications* 117 (2019).

[Sak+20] Yuji Sakurai, Takuya Watanabe, Tetsuya Okuda, Mitsuaki Akiyama, and Tatsuya Mori. "Discovering HTTPSified Phishing Websites Using the TLS Certificates Footprints". In: *2020 European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2020.

[Sal+22] Said Salloum, Tarek Gaber, Sunil Vadera, and Khaled Sharan. "A systematic literature review on phishing email detection using natural language processing techniques". In: *IEEE Access* (2022).

[Sán+22] Manuel Sánchez-Paniagua, Eduardo Fidalgo Fernández, Enrique Alegre, Wesam Al-Nabki, and Víctor González-Castro. "Phishing URL Detection: A Real-Case Scenario Through Login URLs". In: *IEEE Access* 10 (2022).

[Sas15] Angela Sasse. "Scaring and bullying people into security won't work". In: *Security & Privacy* 13.3 (2015).

[Sch+18a] Quirin Scheitle, Oliver Gasser, Theodor Nolte, Johanna Amann, Lexi Brent, Georg Carle, Ralph Holz, Thomas C. Schmidt, and Matthias Wählisch. "The Rise of Certificate Transparency and Its Implications on the Internet Ecosystem". In: *Internet Measurement Conference*. ACM, 2018.

[Sch+18b] Samuel Schüppen, Dominik Teubert, Patrick Herrmann, and Ulrike Meyer. "FANCI: Feature-Based Automated NXDomain Classification and Intelligence". In: *Proceedings of the 27th USENIX Security Symposium*. USENIX, 2018.

[She+07] Steve Sheng, Bryant Magnien, Ponnurangam Kumaraguru, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, and Elizabeth Nunge. "Anti-Phishing Phil: The Design and Evaluation of a Game That Teaches People Not to Fall for Phish". In: *Third Symposium on Usable Privacy and Security (SOUPS 2007)*. ACM, 2007.

[She+09] Steve Sheng, Brad Wardman, Gary Warner, Lorrie Cranor, Jason Hong, and Chengshan Zhang. "An empirical analysis of phishing blacklists". In: Carnegie Mellon University, 2009.

[SL20] Mario Silic and Paul Benjamin Lowry. "Using design-science based gamification to improve organizational security training and compliance". In: *Journal of management information systems* 37.1 (2020).

[SRV15] Simon Stockhardt, Benjamin Reinheimer, and Melanie Volkamer. "Über die Wirksamkeit von Anti-Phishing-Training". In: *Mensch und Computer 2015 - Workshopband*. Oldenbourg Wissenschaftsverlag, 2015.

[SS16] Alexander Streicher and Jan D Smeddinck. "Personalized and adaptive serious games". In: *Entertainment Computing and Serious Games: International GI-Dagstuhl Seminar 15283*. Springer, 2016.

[Sub+17] Abdulhamit Subasi, Esraa Molah, Fatin Almkallawi, and Touseef J Chaudhery. "Intelligent phishing website detection using random forest classifier". In: *International Conference on Electrical and Computing Technologies and Applications*. IEEE, 2017.

[SUM17] Jeffrey Spaulding, Shambhu Upadhyaya, and Aziz Mohaisen. "You've been tricked! A user study of the effectiveness of typosquatting techniques". In: *Proceedings of the 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2017.

[Szu+14] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. "The long "taile" of typosquatting domain names". In: *Proceedings of the 23rd USENIX Security Symposium*. USENIX, 2014.

[TCB18] Ivan Torroledo, Luis David Camacho, and Alejandro Correa Bahnsen. "Hunting malicious TLS certificates with deep neural networks". In: *Workshop on Artificial Intelligence and Security*. ACM, 2018.

[Tha+19] Tran Phuong Thao, Yukiko Sawaya, Hoang-Quoc Nguyen-Son, Akira Yamada, Ayumu Kubota, Tran Van Sang, and Rie Shigetomi Yamaguchi. "Influences of human demographics, brand familiarity and security backgrounds on homograph recognition". In: *arXiv preprint arXiv:1904.10595* (2019).

[Thi22] Moritz Thiele. "Evaluating and Enhancing Phishing Classification based on Certificate Transparency Logs". MA thesis. RWTH Aachen University, 2022.

[Tho+19] Christopher Thompson, Martin Shelton, Emily Stark, Maximilian Walker, Emily Schechter, and Adrienne Porter Felt. "The web's identity crisis: understanding the effectiveness of website identity indicators". In: *Proceedings of the 28th USENIX Security Symposium*. USENIX, 2019.

[TL20] Jan Tolsdorf and Luigi Lo Iacono. "Vision: Shred If Insecure – Persuasive Message Design as a Lesson and Alternative to Previous Approaches to Usable Secure Email Interfaces". In: *2020 European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2020.

[Tse+11] S. Tseng, K. Chen, T. Lee, and J. Weng. "Automatic content generation for anti-phishing education game". In: *Int. Conf. on Electrical and Control Engineering*. IEEE, 2011.

[Tse+15] Shian-Shyong Tseng, Tsung-Yu Yang, Jui-Feng Weng, and Yuh-Jye Wang. "Building a game-based internet security learning system by ontology crystallization approach". In: *Int. Conf. on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE)*. CSREA Press, 2015.

[Ver23] Verizon. *DBIR - Data Breach Investigations Report 2022*. Tech. rep. 2023. URL: `https://www.verizon.com/business/resources/reports/2022/dbir/2022-data-breach-investigations-report-dbir.pdf`.

[Vol+17] Melanie Volkamer, Karen Renaud, Benjamin Reinheimer, and Alexandra Kunz. "User experiences of torpedo: Tooltip-powered phishing email detection". In: *Computers & Security* 71 (2017).

[VRG16] Melanie Volkamer, Karen Renaud, and Paul Gerber. "Spot the phish by checking the pruned URL". In: *Information & Computer Security* 24.4 (2016).

[Vuk12] Era Vuksani. "Device Dash: Designing, Implementing, and Evaluating an Educational Computer Security Game". PhD. Thesis. Wellesley College & MIT Lincoln Laboratory, 2012.

[Was20] Rick Wash. "How experts detect phishing scam emails". In: *Human-Computer Interaction* 4.CSCW2 (2020).

[WDL21] Stephan Wiefling, Markus Dürmuth, and Luigi Lo Iacono. "What's in score for website users: A data-driven long-term study on risk-based authentication characteristics". In: *Financial Cryptography and Data Security: 25th International Conference (FC 2021)*. Springer, 2021.

[Wen+19] Zikai Alex Wen, Zhiqiu Lin, Rowena Chen, and Erik Andersen. "What. Hack: Engaging Anti-Phishing Training Through a Role-Playing Phishing Simulation Game". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2019.

[WHJ18] Emma J Williams, Joanne Hinds, and Adam N Joinson. "Exploring susceptibility to phishing in the workplace". In: *International Journal of Human-Computer Studies* 120 (2018).

[WJZ18] Patrickson Weanquoi, Jaris Johnson, and Jinghua Zhang. "Using a game to improve phishing awareness". In: *Cybersecurity Education, Research and Practice* 2018.2 (2018).

[WLR16] Jingguo Wang, Yuan Li, and H Raghav Rao. "Overconfidence in phishing email detection". In: *Journal of the Association for Information Systems* 17.11 (2016).

[Woo+16] Jonathan Woodbridge, Hyrum S. Anderson, Anjum Ahuja, and Daniel Grant. "Predicting Domain Generation Algorithms with Long Short-Term Memory Networks". In: *arXiv:1611.00791* (2016).

[WRN10] Colin Whittaker, Brian Ryner, and Marria Nazif. "Large-scale automatic classification of phishing pages". In: *Network and Distributed System Security Symposium (NDSS '10)*. Internet Society, 2010.

[WW13] Martin R Wolf and Ute Wiese. "A comparative transformation model for process changes using serious games". In: *Proceedings of the 2nd International Conference on Serious Games and Applications for Health (SeGAH)*. IEEE, 2013.

[Yan+12] C. Yang, S. Tseng, T. Lee, J. Weng, and K. Chen. "Building an Anti-phishing Game to Enhance Network Security Literacy Learning". In: *Int. Conf. on Advanced Learning Technologies*. Vol. 12. IEEE, 2012.

[Yas+18] Affan Yasin, Lin Liu, Tong Li, Jianmin Wang, and Didar Zowghi. "Design and preliminary evaluation of a cyber Security Requirements Education Game (SREG)". In: *Information and Software Technology* 95 (2018).

[Zha+21] Penghui Zhang et al. "Crawlphish: Large-scale analysis of client-side cloaking techniques in phishing". In: *2021 Symposium on Security and Privacy (SP)*. IEEE, 2021.

[Zha+22] Penghui Zhang et al. “I’m SPARTACUS, No, I’m SPARTACUS: Proactively Protecting Users from Phishing by Intentionally Triggering Cloaking Behavior”. In: *Proceedings of the 2022 SIGSAC Conference on Computer and Communications Security*. ACM, 2022.