Master thesis
Computing Science

Radboud University   Secura B.V.

Detection and Detection Evasion with Man-in-the-Middle Phishing

Author:
Floris Valentijn
s1031160

First supervisor/assessor:
Bart Mennink
`b.mennink@cs.ru.nl`

External supervisor:
Max van der Linden
`max.vanderlinden@secura.com`

Second assessor:
Gunes Acar
`g.acar@cs.ru.nl`

February 22, 2024

Abstract

Phishing is one of the most prevalent security threats with significant consequences. Man-in-the-middle phishing toolkits are one way to easily set up attacks that also steal multi-factor authentication codes and cookies. Additionally, these types of attacks are a real cybersecurity threat for organizations and people. New techniques are being implemented constantly, and this makes it hard to keep up on the state of the art. Therefore, it is crucial that advancements on detection and detection evasion techniques for these phishing attacks are evaluated. This research aims to understand the detection techniques used by active scanners on the web and the detection evasion techniques that are possible and adapted by the man-in-the-middle phishing toolkit called Evilginx3 in order to evaluate this toolkits effectiveness for red teaming. By studying the existing literature and testing a working setup of Evilginx3 open to the internet, this thesis evaluates the use of Evilginx3, if it can be detected, and why it might not be detected. Evilginx3 shows to be an effective tool that can currently be used for red teaming with minor changes to the initial setup. The experiments distinguishes three types of different scanners, namely scanners focused at newly distributed certificates, scanners focused at Outlook and Gmail email links, and scanners emerging from three different online virus scanners. This research showed that after a month of scanning, the Microsoft 365 phishlet was only detected by seven vendors as shown in a VirusTotal report, and the Cisco Systems phishlet shows no signs of detection at all from these reports. However, from this research it can be argued that these detections can likely be prevented by activating a redirection page. If the target webpage protects itself properly with URL validation, it can be hard to use a working reverse proxy with the webpage to phish a user. In the most recent state of the internet, these findings clarify how man-in-the-middle phishing toolkits can be detected and that Evilginx3 is a valid phishing tool when considering its usefulness in red teaming.

Contents

Chapter 1

Introduction

One of the most common cybersecurity threats are phishing attacks. According to DataProt, 36% of all data breaches in 2022 involved phishing [7]. The word “phishing” originates from “fishing”, and similarly to catching fish, attackers try to lure victims in by using personal information as bait in order to obtain access to a user’s account, computer, or network [25].

These phishing attacks are focused on obtaining credentials for authentication systems, which are generally a combination of a username and a password. Recently, organizations have developed the tendency to think that their authentication systems are improved by using multi-factor authentication (MFA) methods. This means that multiple authentication methods are simultaneously used for authentication, for example adding an SMS code to the username and password combination. However, even these methods have security vulnerabilities because they may still have similar flaws to the password system. Organizations want to protect their assets as well as possible, and this includes protection from a possible phishing attack that allows phishing systems with MFA enabled.

Secura is a security company that focuses on raising the cyber resilience of its clients in various ways. One of these ways is using red teaming assessments to test the security capabilities of these clients. These assessments aim to breach the targeted organization under agreed conditions. Phishing is one of the possible means of achieving this goal. Furthermore, the targeted organization can only be successfully breached if the red teaming assessment is not detected by this organization. Therefore, it is best to execute the least amount of phishing attacks as possible. If it is possible to phish the organization with only one try, it would reduce detection possibilities significantly as opposed to using multiple tries.

Phishing is an important part of red teaming. To phish an organization as a red team, tooling exists to make this process easier. The tools Secura currently uses lack new options that are recently released on newer versions. The newer tools that are available offer more detection evasion capabilities and automatic altering of important phishing aspects such as generation of URLs. This thesis focuses on one of these newly updated tool called "Evilginx3" [16]. This is a recent open-source tool aimed at red teamers that is still being developed, and that allows efficient MFA phishing. The goal of this research is to investigate what Evilginx3 does, what it cannot do, and how it can be used in red teaming assessments in order to strengthen an organization's cybersecurity. Therefore, this research investigates detection and detection prevention techniques to both demonstrate and evaluate the use of Evilginx3 in red teaming.

The main goal of this thesis is to help the Secura red team perform state of the art MFA phishing attacks to breach organizations while staying undetected. The hypothesis is that the efficient usage of Evilginx3 enables the red team to utilize different advantages over other phishing toolkits such as GoPhish [3] and Modlishka [9]. Additionally, this research contains a guide for using the toolkit in realistic red teaming assessment conditions.

To do this, this thesis gives an overview of the various detection possibilities for man-in-the-middle (MITM) phishing toolkits. Additionally, this thesis explains how to setup Evilginx3 and how to use it to perform a successful phishing attack targeting two different websites. As an experiment, this thesis tests the detection of Evilginx3 in practice. This collection of information is especially valuable for the Secura red team in deciding if they deem it valuable enough to start using the tool in their assessments on top of gaining an overview of the current state of MITM phishing detection in general.

A lot of research exists on phishing attacks in general and how they can be prevented. In preparation of this research, we identified no research on version 3.2 of Evilginx and the use of this software in red teaming. However, research exists on detection techniques and MITM phishing looking at various toolkits. As a result, this thesis includes an investigation on existing research to understand the current detection techniques so as to use Evilginx3 as effectively and stealthily as possible.

After the introduction, this thesis continues with Chapter 2 which contains a background on authentication, phishing, phishing tools and red teaming. Chapter 3 contains information about previous work, investigating different phishing attacks and social engineering methods. Chapter 4 explains how a target can be phished by the use of Evilginx3, starting from how to set it up to what options it includes and how to execute it. Chapter 5 investigates how MITM phishing toolkits work and explains the three ways in which they can be detected and what are the possible evasion techniques. Chapter 6 describes how the experiment is set up and what methods are used to perform the experiment. Chapter 7 shows the results of the experiment. Chapter 8 identifies future work on phishing toolkit detection and detection evasion. Finally, Chapter 9 concludes the work.

Chapter 2

Background

The background contains the necessary information to understand the subject. It is separated into four different dimensions ranging from general to more focused. Section 2.1 delves into the workings of authentication and why it is often troublesome. Section 2.2 explains a specific kind of attack called phishing. Section 2.3 looks into the type of phishing tools that are used to perform phishing attacks against MFA systems. Section 2.4 gives an overview of what red teaming is and how it performs phishing attacks to effectively help organizations increase their resilience against them.

2.1 Authentication

Authentication is a complex process because it is necessary to obtain a connection between a physical person and a computer system in order to authenticate. This means that the strength of the authentication depends entirely on the strength of this connection.

There are multiple ways of authenticating someone. We can authenticate by something we know, something we are, something we own, and our location [26]. The most common way of authenticating is by using a password. Passwords count as something we know and are the most abused way of authentication because they are often used in present systems. A physical key or a keycard counts as something we own and a fingerprint as something we are. This section further describes different types of authentication systems and multi-factor authentication (MFA) specifically, which is the main focus of the authentication research investigated in this thesis.

2.1.1 Authentication Systems

Building a secure authentication system is not easy to achieve. It is necessary to take a lot of design decisions in order to achieve this. These are things like deciding what level of security and simplicity you ultimately need. This depends on the use case of the authentication system and the budget of the organization. Bedra et al. (2016) make multiple interesting points regarding design decisions [12].

Design decision examples:

- Should all authentication attempts go through a single server or should they be divided?
- Should we have a single endpoint or let services authenticate at their own endpoint?
- Do we want to use whitelists to provide an always-on authentication approach?
- Do we want to perform third-party integration?

Authorization is another crucial aspect of designing an authentication system. In order to have a proper authorization system integrated with the authentication system, we need to think about design decisions such as which protocol and programming language we are going to make use of. Doing this correctly can protect important file systems and network locations within the system. Failing to do this in a clean way can result in an attacker abusing crucial parts of the system.

Bedra et al. (2016) also distinguish four different authorization schemes and their use cases [12].

- Role-Based Access Control:
  Using a fixed number of roles and assigning them to user accounts. Users are statically placed in environments based on their roles and have access to a fixed set of attributes. The problem is that the implementation cannot be to complex or it becomes hard to control.
- Attribute-Based Authorization:
  Both users and attributes have roles, allowing for a smoother and more complex authorization system. This does become significantly more complicated as opposed to static roles because it is more prone to mistakes.
- Centralized Authorization:
  Users can also request authorization via a centralised point system. This makes sure that every time an attribute is accessed, the user has to be authorized by the system. This does require a centralized authentication point which is an extra system running alongside the existing systems. This system also has to be managed and administrated.

- Adaptive Authorization:
  When it is crucial that a system does not get abused, we can also improve authorization security further. The authorization system can check a lot of metadata to authorize us. This provides more protection against the phishing attacks which we discuss in Section 2.2. However, using such a system takes significantly more work than just taking one of the simpler options discussed before.

Note that is possible to find a middle-ground or even use multiple authorization systems in combination with each other. The best authorization system is just like authentication dependent on the use case of the system.

2.1.2 Multi-Factor Authentication

Multi-factor authentication (MFA) works by giving the user multiple login methods alongside the first one, which is almost always a password. By combining two or more of the authentication methods, we give the user multiple challenges to complete before being able to log in. This ensures that it becomes harder for an attacker to break the authentication because of having to perform multiple authentication methods in succession.

In order to prevent MFA phishing, it is important that the first form of authentication has a strong security as well. This ensures that the attacker cannot go straight to phishing MFA data and first has to phish the password. Passwords should comply with multiple security standards so not to be able to get hacked by other means such as library attacks. The Open Worldwide Application Security Project (OWASP) has multiple authentication recommendations for the use of passwords. For example, passwords of 8 characters or less are considered to be too weak, all characters should be allowed to be used in a password, and there should be a password strength meter included for users when creating a password on their system [26].

OWASP recommends to improve usability when implementing MFA. This means that we should make it easy for users to stay logged in. This can be done either by remembering a browser or allowing IP addresses that the user already authenticated on in the past [26]. However, when implementing this system we also introduce another factor that an attack is able to abuse. The attacker could make it so that he is remembered as a valid login instead of the targeted user.

A problem with 2FA/MFA is that users barely tend to adopt the method if it is optional. Petsas et al. (2015) discovered that, out of 100,000 Google accounts, only 6.4% of their users enabled 2FA. This shows again that users do not like to be bothered by authentication, which makes them more likely to be less secure [28]. The same research also found that password reminders are a significant aid people rely upon. This also adds to the security problems because these reminder systems can also have various vulnerabilities.

The problem with two factor authentication and phishing (Section 2.2) is that we can argue that if we can obtain a password from someone through social engineering, it is also possible to obtain for example a code displayed on their phone or anything that only the user had access to. This begs the question if 2FA is really as strong as it suggests to be, and the answer according to the previously mentioned research is that it really depends on how it is set up and controlled.

There are many more ways to mitigate MFA bypass attacks and this research tries to discuss a wide range of them and how to avoid detection from them wherever possible. These mitigations are for example educating users, email filtering, web filtering, using a physical token, risk based authentication, behavioral analytics, and monitoring for MFA bypass attacks [4].

Figure 2.1 shows how MFA works and how it bypasses the MITM connection between the user, the reverse proxy, and the target website.

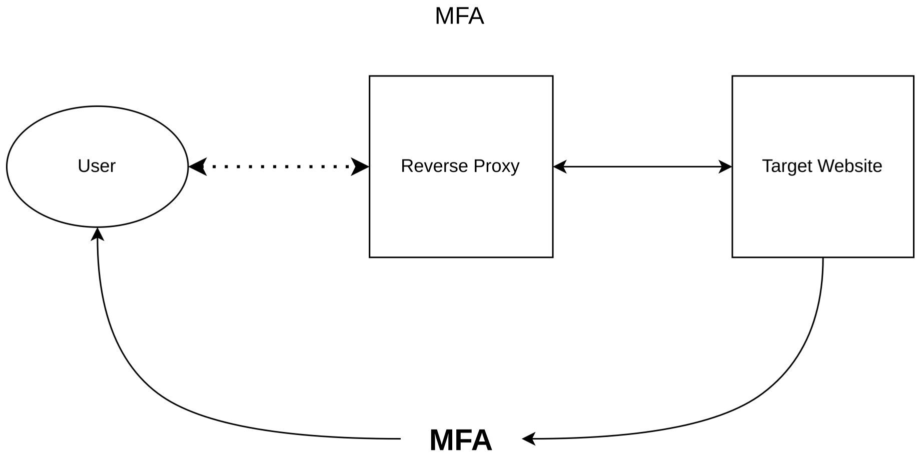


Figure 2.1: Multi Factor Authentication

2.1.3 Non-Conventional Authentication

When thinking of authentication, we usually think about a system requesting a password from a user, the user entering the password, and the system checking it and granting access to the user. Although verification is an important part of the authentication process, the user does not necessarily have to remember and physically enter a password for it to work. For example, it is possible to use metadata for authentication of the user or by using public key cryptography for the user to prove himself.

There are more ways to make authentication different from the usual password process. Somayali et al. (2013) propose to make authentication more interesting to counter the fact that people do not want to be bothered with passwords. A narrative would make the user go through a text-based adventure in order to authenticate [34]. A concern could be that social engineering may become an even larger issue because users may give their passwords away without noticing through answering questions asked by social engineers.

2.2 Phishing

Phishing is a type of social engineering attack with the goal of getting victims to reveal their personal information or install malware [43]. By using this personal information, the attacker can gain access to the victims personal accounts as well. Subsequently, this personal account can be used to perform a wide range of different criminal activities like impersonation attacks, stealing money, or identity fraud.

2.2.1 Phishing Attacks

There are a number of different types of phishing attacks. The 5 most common of these phishing attacks are the following [18]:

- Email Phishing:
  This is the most common attack performed by attackers. The attacker creates an email that mimics a genuine organization. Next, he sends this email to thousands of potential victims hoping that one of them takes the bait.

- Spear Phishing:
  This works through investigating the victim to find out personal information beforehand. This information includes their name, job title, location, and more specific information concerning their job. This email will look much more like a genuine email than a normal phishing email as a result.

- Whaling:
  These are phishing attacks aimed at important targets within an organization. Their goal is to make this target execute certain tasks. By pretending to be a chief executive officer (CEO), attackers can make

the target perform these tasks without questioning the credibility of the email.

- Smishing and Vishing:
  In this case, the attacker uses SMS messages (smishing) or phonecalls (vishing) to social engineer their victims. These messages can be similar to emails in that they trick a target into visiting a malicious website or performing malicious actions.

- Angler Phishing:
  Using social media, attackers can use information that people publicly disclose to construct an attack. An attacker pretends to be a customer support employee willing to help a customer in need. While in this role, he requests personal information or tries to have the customer click a malicious link.

In order to protect against phishing, it is important to understand that these different types of phishing attacks exist and that any new technology may introduce new types of attacks.

For research on spear phishing, Meyers et al. (2018) have investigated specifically the training of security experts to perform spear phishing [21]. The participants used the target's name in a google search in order to find information that can be abused. They call this learning technique SiEVE, and they found that the security experts found more information on the targets and created more effective phishing attacks by using this technique. This proves that following certain guidelines increases the effectiveness of social engineering attacks.

Furthermore, when spear phishing gets combined with AI systems, these attacks become even stronger because of how fast different kinds of personal information can be turned into a convincing phish [22]. On top of this, AI also becomes better at cloning voices, so vishing also becomes a realistic problem. How much AI will affect the field of focus is not within the scope of this research.

2.2.2 Man-in-the-Middle Phishing

A man-in-the-middle (MITM) attack is an attack were an adversary is sitting in the middle of a conversation between two or more agents. A MITM phishing attack is an attack where the adversary keeps copying a real website and showing it to the victim while using the information at the real website himself [20]. The power of this MITM phishing attack is that it can phish multi-factor authentication codes that are sent over the same channel because they are located in the HTTP Post Request. The user gets redirected to the multi-factor web page where he must enter his authentication code that he receive for instance on their mobile phone. The victim copies this page as well and tricks the victim into believing it is also the real page. Just like the phishing page would steal the password, the phishing page now also steals the code. The power of this attack is the fact that we immediately enter the code into the real page and receive a session token that we can use to stay logged in for a longer period of time for as long as the victim does not notice the attack having occurred.

Figure 2.2 Shows how the MITM toolkit is setup and how the reverse proxy sits in the middle of the connection. The dashed line around the reverse proxy and the target webpage highlights that the reverse proxy and the target webpage are directly connected on the internet.

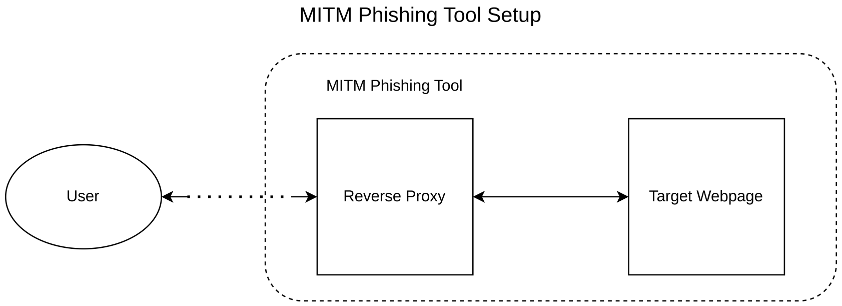


Figure 2.2: MITM phishing setup

2.2.3 Indicators of Compromise

Indicators of compromise are pieces of forensic information that are left after an attack [42]. There are various ways which make it possible to detect phishing attacks. By looking at all sorts of visual and background information it may be possible for tools to detect website vulnerabilities before or after they are being exploited.

Examples of visual cues: Bad use of language, unexpected link contents, different format than regular, unknown domain name, unknown sender, unknown CC.

Examples of technical cues [14]: Unknown IP addresses, unexpected HTML response size, trying too much non-existent logins, account activity irregularities, location irregularities, database irregularities, DNS request irregularities.

Indicators of compromise are an important part of understanding phishing detection. Since this is a large list of cues, it is important for phishing detection that we can clearly check these indicators for our specific phishing attack.

It is also important to learn from the indicators of compromise in recent attacks since attacks are constantly evolving and getting more complex [8]. It is important to have a system set up in a way that new indicators of compromise can be found when doing forensic investigations.

2.2.4 Phishing Prevention

The ultimate goal of red teaming assessments is to highlight security issues in the organization that requested the assessment. However, the organization who receives the red teaming assessment typically already has security measures in place that were identified through penetration tests. The red teaming assessment is thus more focused on complex attacks and should try to avoid common detection systems. Thus, it is important to understand what are the ways that phishing can both be prevented and detected. The World Wide Web Consortium highlights five key principles that an anti-phishing solution must take into account [39]:

1. The trusted user interface for authentication must be based on a secret because all user interface is spoofable.

2. A trusted channel cannot be trusted because an attacker can use a trusted channel.

3. The client must also authenticate the server because an unauthenticated server can easily ask for more confidential information.

4. A cleartext password must not be revealed during any phase of authorization because an attacker will fool the user into completing any standard process.

5. The anti-phishing solution must integrate with existing password-based authentication because users are trained to use passwords.

We must also understand that these principles are not followed by all organizations, and that we can take advantage of this fact when finding out that this is the case. Ultimately, the red teaming assessment highlights this flaw and possibly others.

2.2.5 Social Engineering

In order to get a phishing attack to work, it is necessary to trick the target into thinking that the phishing attempt is an interaction with a legitimate server. This is true because it is impossible to completely replicate a real counterpart since the destination address will be different unless spoofed. Social engineering is defined as a physiological manipulation technique aimed at making people take certain actions or disclosing confidential information [44]. All social engineering techniques make use of human decision-making attributes known as cognitive biases. A cognitive bias is defined as the construction of a person's reality [41]. This cognitive bias makes sure that humans can make decisions faster than usual and that this might save them from possibly dangerous situations. Contradictory, in case of social engineering attacks, this puts them into a dangerous situation instead. As such, an important aspect of social engineering is to force the person to take fast decisions in order to make their cognitive bias take over instead of giving them time to think critically. When any person gets allowed time to think critically the social engineering attack will fail in most cases.

Because of all this, social engineering can be especially devastating to humans since they often target sensitive parts in order to trigger a rushed response from the victim. Thus, it is important that awareness is spread and that these attacks are prevented as much as possible to prevent unnecessary cases of successful social engineering attacks.

2.3 Phishing Tools

Phishing tools are essential equipment for red teamers and penetration testers. They allow a range of phishing attacks to be more easily generated. Since there are such a number of important factors to be considered, attackers cannot be fully designing their phishing attacks from scratch for all different cases. This research focuses on a phishing toolkit called 'Evilginx3'. [16] This is a recent toolkit that is still improving at the time of this research. We look at a possibility to use this toolkit effectively for red teaming and more specifically at what the detection implications may be when using it to perform a phishing attack.

2.3.1 Types of Phishing Attack Tools

In order to understand Evilginx3, it is necessary to also understand what other phishing tools are doing and if they are doing things differently than each other. It is also important to understand that these phishing tools are designed for security testing. This means that we need to setup the phishing tool in a way that prevents any indicator that could inform a victim or security program about the usage of the tool [35]. The goal is to use the tool in a similar way that an attacker would use it. Thus, this renders this tool a viable option for attacks when utilized correctly.

2.3.2 Evilginx3

Evilginx3 is a MITM phishing toolkit that uses a configuration to setup a reverse proxy talking to a target website. It is written in the programming language GO as it is a commonly used language for security tools. This toolkit is particularly effective against MFA phishing because of its MITM setup. This research focuses on the newest version of Evilginx which is version 3.2. Evilginx3 enables this type of phishing from a command-line interface. In order to execute these advanced phishes, Evilginx3 requires its user to create phishing website templates called phishlets. These phishlets have to created specifically for a targeted webpage to create a realistic imitation and thus make it usable for a phishing attack. Evilginx3 is built up out of previously existing server creation software called "Nginx“, this software creates a server that can act either as a web server or a proxy and reverse proxy server for email [24]. Evilginx3 implements this server software and in combination with phishlets recreates common websites in order to phish users. The tool can be used to setup specific and realistic phishing attacks to test an organizations security in a red teaming assessment. This thesis investigates the MITM phishing toolkit Evilginx3 as its main focus.

2.3.3 Phishing Detection Tools

Phishing detection tools are systems that try to find phishing cues when users access the internet. These tools can be used as an additional counter measure to ensure the users security.

According to Regenscheid and Galluzzo (2023), There are four important attack vectors that phishing resistant authenticators must address [1].

1. Impersonated Websites:
   Authenticator services must not be allowed access on domains that are not registered. There should be a whitelist or a channel present that checks this.

2. Man-in-the-Middle:
   Authenticator services must make sure that there can not be an attacker present between the user and the website. This can be achieved by using cryptography in order to create a channel.

3. User Entry:
   All information entered by the user must be protected from entering the internet. This can be achieved by using encryption methods on the user-entered data from the users machine.

4. Replay:
   Authenticator services must make sure that the same authentication

message cannot be used repeatedly in order to prevent a listening attacker from using the intercepted information.

The article also mentions that in reality MFA might not be desired for authentication because there exist phishing resistant authenticators that are easier, faster, and more convenient. The previously mentioned attack vectors are the ones that must be effectively managed in order to build a good authenticator. This research explores which authenticators these are and leaves their phishing vulnerabilities as future work.

Preexisting research on phishing detection tools is readily available. These papers investigate how well current phishing detection tools can detect phishing and build new phishing tools followed by an assessment on its effectiveness. Sharma et al. (2017) did a comparative analysis and awareness survey on various phishing detection tools [33]. They found that some phishing toolbar detection tools are very effective at detecting mass phishing attacks. Thus, it is important to deviate from the mass phishing attacks in order to remain stealthy. Zhang et al. (2021) invented a new phishing detection tool called CrawlPhish [46]. This tool aims to detect cloaking of phishing attacks. They also did a user study that found that the tool does not discourage victims from visiting the site. Geng et al. (2018) also invented a new phishing detection tool called RRPhish [15]. This thesis investigates these tools to find out if they contain interesting detection techniques. These detection techniques are then used to find new detection prevention techniques that are then also implemented into Evilginx3.

2.3.4 Dangers of Phishing Tools

Phishing tools may also have unknown side effects. For example, these tools may also be abused by attackers instead of being used by red teamers and penetration testers as intended. Criminals can get phishing tools fairly easily through the web. Other criminals are creating these phishing tools to eventually make money out of it. These specific phishing tools often contain malware that distributes information such as phished usernames and passwords, IP addresses visited, and personal information of the attacker towards the creator of the tool, making them unsuitable for penetration testing purposes. The presence of these many phishing tools also increases the amount of phishing attacks being launched towards organizations.

2.4 Red Teaming

This research focuses on phishing from a red teaming perspective. In order investigate this, we need to understand what every teams role is, how we look at phishing from a red teaming perspective, what it is that the red team does, and what strategies are used in this context.

2.4.1 Blue, Red, White, and Purple Teaming

The job of the blue team is to protect their systems and detect if there is anyone trying to infiltrate their systems. The job of the red team is to infiltrate the blue team's system without being detected. The white team is independent of the red team and the blue team but are aware of both team's actions. This is to prevent actual incidents from happening such as the occurrence of a real attack or the red team breaking a part of a system that is crucial to the organization. The purple team is a combination of red teamers and blue teamers that together try to solve security problems within the organization. The purple team can be a powerful way to quickly improve an organizations cyber resilience. However, some security issues may only be detected by the red team because of the fact that they work independently from the blue team.

2.4.2 Phishing as a Service

The red team tries to imitate advanced persistent threats (APTs) that have a wide range of resources available. The goal is to make organizations more resilient against these highly motivated attackers as well as the more common attacks. This will help the organization in the long run by preventing both the financial and juridical devastation of certain attacks when successful. Thus, organizations have requested increasingly more red teaming assessments in the last years. Red teams typically conform to standards which include various certifications obtained by their members. Additionally, they often invent new techniques that help them attack a specific system.

Rodriguez-Corzo et al. (2018) designed a methodological model based on Gophish to face phishing vulnerabilities within companies [31]. This research essentially gives a way to help companies face cybersecurity issues through using phishing attacks as a learning factor. This is an interesting model to review in order to improve the usage of Evilginx3 in red teaming.

2.4.3 Penetration Testing

The crucial difference between penetration testing and red teaming is that penetration testing aims at trying to identify as much vulnerabilities as possible in a system or organisation. However, red teaming aims to break security trough the path of least resistance while hiding from the blue team to discover vulnerabilities which cannot be identified trough other means. The end goal of this research is to investigate how these phishing attacks can be used in red teaming assessment, so this includes preventing detection by the blue team. It is however still important to know what the possibilities are. Thus, taking a penetration testing approach towards different systems can be crucial before starting red teaming assessments. This research also aims to perform this and to investigate what detection techniques are possible by the blue team.

2.4.4 Strategies

Red teaming assessments should follow a strategy in order to conform with the standard that the hiring organization desires. These strategies should be well documented and designed such that the organization can optimally do its job. This strategy includes the communication with the target organization, the way that different attacks are performed, the technology that is used, the way in which the technology is used, and how specific details about the assessment are reported.

It is also important that red teaming is correctly learned to beginners. Meyers et al. (2018) has investigated a strategy that effectively learns red teaming techniques to students [21]. These students improved their red teaming skills by following this strategy over time. However, it is not yet clear if professionals can benefit from this.

2.4.5 OPSEC

Both the red team and the blue team should try to conform with operational security (OPSEC) in order to prevent security mistakes when trying to carry out their function. The blue team focuses on detecting traces of intrusion through a set of rules that prevent them from making common security mistakes. These are rules such as always logging important data and using more advanced authentication. The red team focuses on detection evasion by also following a set of rules that prevent unnecessary detection by the blue team. This approach optimizes what both teams can learn from each other in case both teams use it correctly.

Chapter 3

Previous Work

Phishing attacks have been widely investigated both on a social engineering and a technical level. We can use this previous research to understand how phishing attacks work and how the dynamics between humans and machine are manifested from a phishing point of view. We can separate the existing work best into two different categories. The first category is social engineering attacks aimed at fooling humans. It is possible to prevent these attacks in two important ways, which are human training and automated detection through algorithms or AI. The second category are technical aspects. This category focuses on the ability to phish more complex systems and the possible phishing tools and tricks that are most recently released.

3.1 Social Engineering Aspects

Social engineering is a complex topic with many human factors. It is impossible to detect this with tools alone, this makes it necessary to look at human thought patterns and human behavior.

3.1.1 Human Detection Capabilities

Wash (2020) shows in his research that human methods of detecting phishing attacks use different information than technical methods of phishing. He found that humans extensively use contextual knowledge and information about typical behaviours [40].

Research by Alsharnouby et al. (2015) has shown that users are only able to successfully detect 53% of phishing attacks from looking at the browser and trying to see a difference between a phishing attempt or the real website [2]. This shows that simply relying on preventing social engineering of users has no possibility of entirely solving the issue.

Katelin A Moul (2019) argues that education alone will never be able to protect all users [23]. Despite things like significantly obvious email warning banners there are still users who had their accounts compromised. These users simply follow any direction given to them while ignoring any suspicious indicators that they might be phished. However, MFA still plays a significant role in improving phishing prevention. My take on this is that this might have to do with common phishing attacks not targeting MFA.

3.1.2 Mitigating Social Engineering

Sumner et al. (2019) discusses the mitigation of social engineering attackers from three aspects: detection, education and training, and susceptibility [36]. They found it is best to train and educate people in combination with automated phishing detection. They argue it is most important to emphasize training on groups of people that have recently left college (age 18-25). Furthermore, they explain: "There have been few research papers regarding social engineering attacks on atypical organizations such as college or university, and practically none focusing on their faculty and staff". This means that many organizations have different needs and all of them have to be separately evaluated for social engineering vulnerabilities.

3.1.3 Lateral Phishing Attacks

Lateral phishing attacks are phishing attacks were compromised accounts from within the organization are used to perform phishing attacks on other members of the organization. These attacks are especially tricky to detect since their origin is legitimate. Chitare et al. (2023) interviewed organization employees in the UK and India, and it seems that most of these employees' approach to email identification was not optimal for detecting lateral phishing attacks [5].

3.2 Technical Aspects

This research focuses on the appliance of phishing tools to detect security vulnerabilities within organizations. Thus, this category investigates existing research that investigates different possible phishing tools in existence, the possibility to phish different types of authentication systems, and the appliance if phishing attacks in the real world in order to improve and organizations cyber resilience.

3.2.1 Phishing Tools

Kondracki et al. (2021) presents an analysis of MITM phishing toolkits used in the wild [19]. They found that it is possible to detect phishing toolkits with 99.9% accuracy by using their proposed classifier. The classifier consists of multiple attributes that create a fingerprint of the websites created by the MITM phishing toolkit. They also found that the TLS fingerprint alone is enough to detect the presence of a MITM phishing toolkit.

Ulqinaku et al. (2019) propose 2FA-PP to prevent run time phishing attacks on 2FA authentication systems. According to their research the problem with 2FA credentials is that phishers can create a "counterfeit proxy site", which an adversary can then forward in real-time to the legitimate website [38]. 2FA-PP prevents this by making use of Bluetooth low Energy (BLE). What happens is that the app and the browser work together to make sure that the user does not get sent to a malicious web server.

Patil et al. (2019) give a methodical overview on phishing detection along with an organized way to construct an anti-phishing framework [27]. Their research gives multiple features that can be possible indicators of compromise, they can be used to construct a machine learning algorithm that detects phishing websites.

Rajab (2018) gives another feature-based phishing detection tool [29]. The features discussed in this research can also be used as indicators for creating an effective phishing tool.

3.2.2 Phishing MFA Systems

Henricks and Kettani (2019) explore the possibilities that MFA has over a single password [17]. They also consider whether MFA has security flaws. In their research they argue that biometrics are a bad way of authentication because the information cannot be altered to create a new password. Biometrics could be abused in ransom attacks because it is privacy sensitive information as well as a complex password. Furthermore, secondary authentication SMS text messaging could also be abused. It has become common for attackers to social engineer cellphone providers. They could for example talk providers into moving this information onto a different SIM card. Additionally, they talk about phishing attacks on MFA that have been more common, which is further investigated in this research.

3.2.3 Comparing Phishing Tools

El Aassal and Verma (2019) investigated if defending against phishing attacks is on par with the creation and usage of phishing attacks [10]. They also did a comparison on the tools: SecurityIQ-PhishSim, Gophish, LUCY Phishing Software, Simple Phishing Toolkit, Social-Engineer Toolkit, King-Phisher, SpeedPhish Framework, Phishing Frenzy, Wombat Security - Thread-

Sim, SpearPhisher Beta, and PyPhisher. After the studies, they identified and focused on 4 different end-user-based detection and awareness aspects:

1. Ensuring an employee knows how to view the header of an email
2. Detecting if an employee actually analyzes the email
3. Detecting if an employee checks link locations before clicking them
4. Detecting if an employee searches for similar email within their mailbox

These four detection and awareness aspects are important to understand when defending against social engineering attacks.

3.2.4 Phishing Detection

Ellahi et al. (2022) investigated detection techniques against two-factor authentication (2FA) phishing [11]. They carried out a specific man-in-the-middle attack scheme in order to calculate the success rate of the attacks performed. They found that among Instagram, COINBASE, Gmail, and Facebook, Gmail is the most effective at detecting 2FA phishing attacks. COINBASE checks IP addresses and sends an email when the IP address is not recognized. Gmail uses multiple AI techniques and one of their techniques is to verify if the current URL matches the string of the real page. Facebook encrypts user credentials on the client side to prevent attackers sniffing the traffic from learning about them.

Zhang et al. (2022) have developed a browsers-app anti-phishing tool for users to prevent phishing called Spartacus [47]. There are many anti-phishing crawlers that are constantly trying to find phishing websites and black list them for other users of the internet. Phishing servers have found a solution to this, they try to detect anti-phishing crawlers before presenting them with a phishing website. They achieve this is by fingerprinting browsers and looking at request headers. Spartacus uses this fact by inserting specific keywords and pieces of information into the HTTP requests, this makes it so that phishing websites misjudge these users as phishing crawlers and thus prevent them from accessing the phishing website. Their research showed that Spartacus protected against 82.3% of the websites they tested.

3.2.5 Browser-in-the-Middle

Tzschoppe et al. (2023) present a different type of effective phishing attack against MFA systems called browser-in-the-middle [37]. Their research showed that from the 13 websites, 12 were vulnerable to this type of phishing attack. They argue that this browser-in-the-middle phishing attack can potentially be used for highly effective targeted phishing. However, for large-scale phishing aimed at different users it is unlikely to be very effective. How much this is like Evilginx3 will be further explored in this research.

3.2.6 Evilgophish

Evans has created a git repository containing an integration of Evilginx3 and Gophish [13]. His reasoning is that Evilginx3 lacks tracking possibilities, social engineering options, and a graphical user interface. By combining these possibilities for email phishing of Gophish with the MFA website phishing of Evilginx3 we can have the best of both of these tools integrated into one.

Chapter 4

Evilginx3

Evilginx3 is a man-in-the-middle (MITM) phishing toolkit that can be used to setup a reverse proxy between a user and a webpage. This toolkit can be used to intercept a victim's credentials while still giving them access to the target webpage. This chapter explains how the toolkit works and how to use it. Section 4.1 talks about the installation of the toolkit. Section 4.2 analyses the setup options of Evilginx3. Section 4.3 explores how phishlets are developed for use with the toolkit. Section 4.4 shows how to perform a phishing attack with the use of Evilginx3.

4.1 Installation

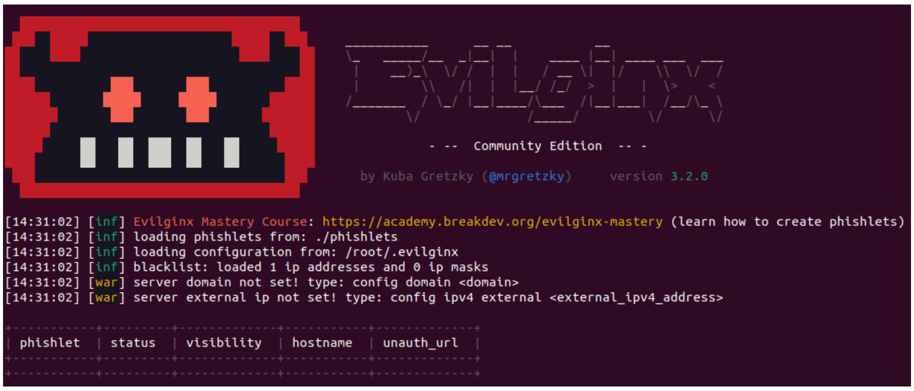
The setup of Evilginx3 can be achieved on both Windows and Linux operating systems through building the source code. This phishing tool is command line only, which means that everything from setup to execution has to be performed from a command line interface. Consequently, the tool is organized in such a way that it is possible for users with only basic computer knowledge to use the tool effectively, although understanding the software and languages like GO and JavaScript gives can give more options for detection evasion. After downloading the software from the GitHub page, the command `make` can be used within the directory. This then builds the instance.

Then, it is possible to execute the built instance of Evilginx3 by using the following command.

```
sudo ./build/evilginx -p ./phishlets -t ./redirectors -developer
```

Adding `-p ./phishlets` or `-t ./redirectors` sets the directories for both these options respectively. Its purpose is explained in Section 4.2.2 for phishlets and Section 4.2.4 for redirectors.

Figure 4.1 shows the startup screen of Evilginx3 after executing the command. In this figure we can see that we are running version 3.2.0 of the tool and some other information for example the loading of the phishlets, configuration, blacklist files, and that we still need to setup a domain and external IPv4.



```
                    - --  Community Edition  -- -

          by Kuba Gretzky (@mrgretzky)     version 3.2.0

[14:31:02] [inf] Evilginx Mastery Course: https://academy.breakdev.org/evilginx-mastery (learn how to create phishlets)
[14:31:02] [inf] loading phishlets from: ./phishlets
[14:31:02] [inf] loading configuration from: /root/.evilginx
[14:31:02] [inf] blacklist: loaded 1 ip addresses and 0 ip masks
[14:31:02] [war] server domain not set! type: config domain <domain>
[14:31:02] [war] server external ip not set! type: config ipv4 external <external_ipv4_address>

+-----------+---------+-------------+-----------+-------------+
| phishlet  | status  | visibility  | hostname  | unauth_url  |
+-----------+---------+-------------+-----------+-------------+
+-----------+---------+-------------+-----------+-------------+
```

Figure 4.1: Evilginx3 startup screen

4.2 Setup

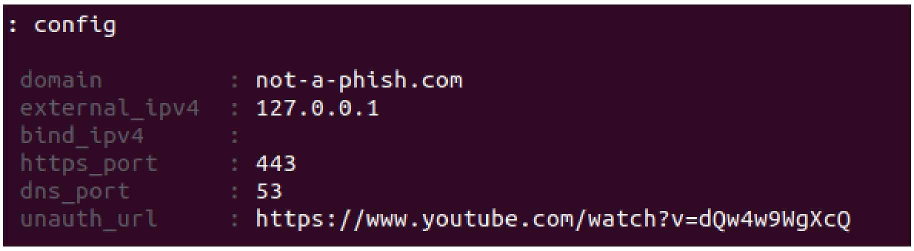
In order to successfully create a phishing link connected to a landing page, Evilginx3 requires some setup [16]. This setup is extensive to allow phishes to be as customized as possible. This setup consists of a configuration, phishlets, lures, redirectors, sessions, proxy, and blacklist.

4.2.1 Configuration

Evilginx3 has to be configured by giving phishlets, lures, IP, DNS, and domain. By using this network related information, Evilginx3 obtains the source code of the website located in general-domain. It then applies all the extra settings to make the website suitable for phishing. Figure 4.2 shows what this option looks like from the command line interface. These options can be changed by using their respective command.

4.2.2 Phishlets

Phishlets are a way to easily configure a new phishing page. Evilginx3 is set up in a way that allows phishlets to be configured so that it is easy to imitate new websites. These phishlets contain a range of options both network and application layer based. Figure 4.3 shows a table with the installed phishlets currently on the system. These phishlets can be enabled

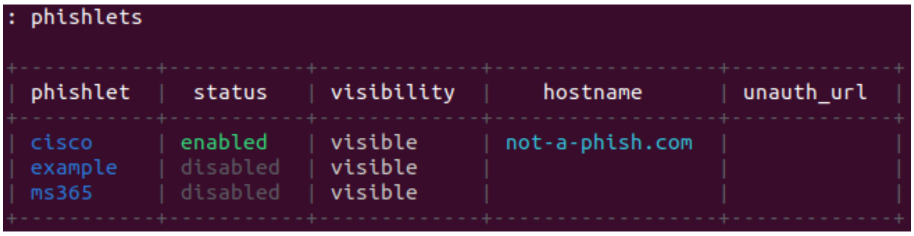


```
: config

 domain           : not-a-phish.com
 external_ipv4    : 127.0.0.1
 bind_ipv4        :
 https_port       : 443
 dns_port         : 53
 unauth_url       : https://www.youtube.com/watch?v=dQw4w9WgXcQ
```

Figure 4.2: Configuration

or disabled for quick use. Additionally they can be made invisible, and they can be given a separate hostname and unauth_url.



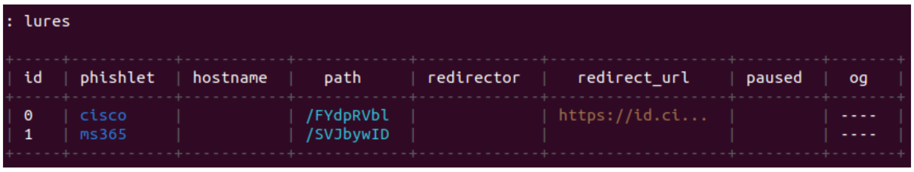
```
: phishlets

+-----------+-----------+------------+-----------------+------------+
| phishlet  |  status   | visibility |    hostname     | unauth_url |
+-----------+-----------+------------+-----------------+------------+
| cisco     | enabled   | visible    | not-a-phish.com |            |
| example   | disabled  | visible    |                 |            |
| ms365     | disabled  | visible    |                 |            |
+-----------+-----------+------------+-----------------+------------+
```

Figure 4.3: Phishlets

4.2.3 Lures

A lure is a link that directly points towards the phishing login screen. Evilginx3 gives options for customizing these links so that they are most effective for phishing. These lures also give the option for redirecting the visitor to a different URL by pausing the lure. It is also possible to change the hostname as long as the top-level domain stays the same. Additionally, the path can be changed to look like anything the attacker wants. Figure 4.4 shows what this table looks like when called from the command-line interface.



```
: lures

+----+-----------+-----------+-----------+------------+-----------------+---------+------+
| id | phishlet  | hostname  |   path    | redirector |  redirect_url   | paused  |  og  |
+----+-----------+-----------+-----------+------------+-----------------+---------+------+
| 0  | cisco     |           | /FYdpRVbl |            | https://id.ci...|         | ---- |
| 1  | ms365     |           | /SVJbywID |            |                 |         | ---- |
+----+-----------+-----------+-----------+------------+-----------------+---------+------+
```

Figure 4.4: Lures

4.2.4 Redirectors

Redirectors are a way to avoid direct detection. This redirector is a website that forwards the user to the phishing link. For example, we could first send the user to a simple blog which then forwards the user to the phishing page. Since this blog looks like a real website, a simple scanner will fail to detect this attack. It is possible to make this website as complicated as necessary in order to avoid scanners or fool victims in any way. Common options to avoid scanners are making it user interactable by placing a button that needs to be clicked, some information to be entered or using a more complex system to distinguish a human and a scanner. Figure 4.5 shows the default redirector page provided by Evilginx3 where the name and download file can be changed by the user. When the user clicks the “Download invoice.pdf“ button, he/she will be redirected to the reverse proxy. This redirector page can crafted into any page possible.

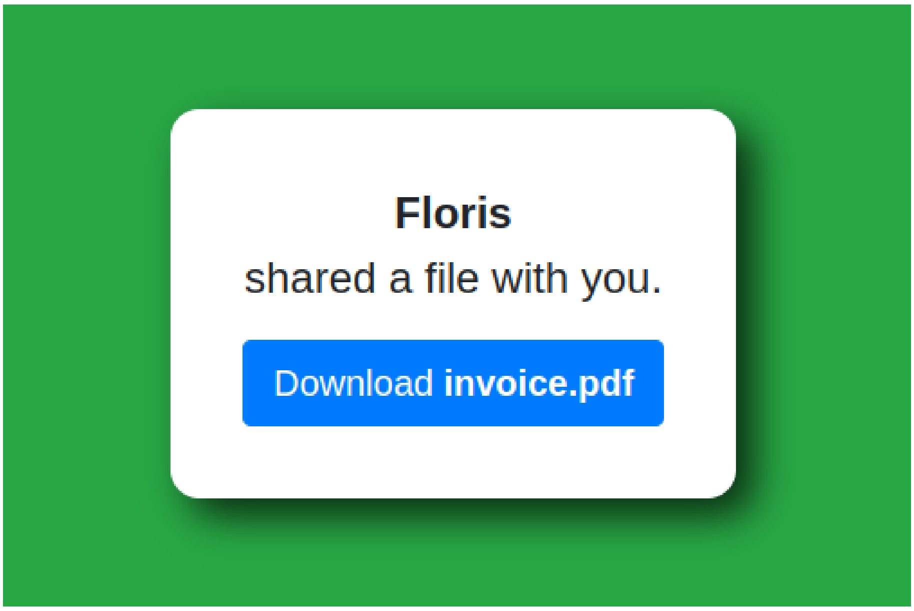
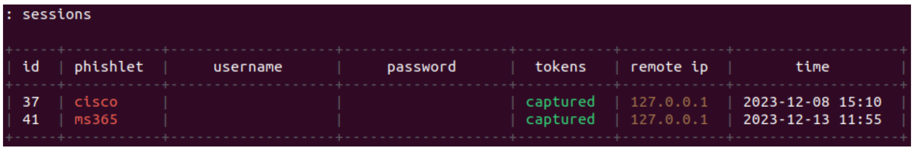

Floris
shared a file with you.

Figure 4.5: Redirector Page

4.2.5 Sessions

Sessions are the history of the users that opened the phishing links. These sessions are saved along with their credentials. The cookies obtained by successfully phishing a user can be used to log in straight to the users account. These cookies are stored in JSON format for ease of use. Browser apps exist to turn this JSON into session cookies for the browser to store them correctly. The browser is logged in impersonated as the user with this session cookie stored into its storage. Figure 4.6 shows two sessions that were successfully captured by the phishlets, the username and password are hidden for privacy reasons and because testing had to performed with real credentials.



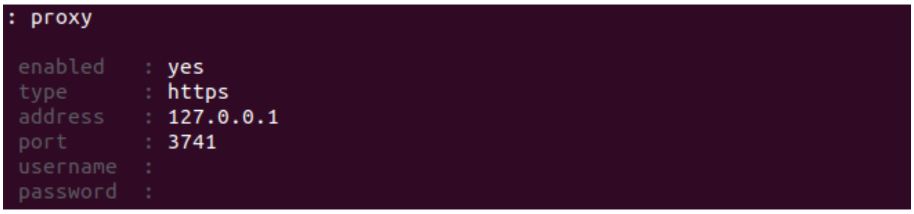
```
: sessions

+-----+-------------+--------------------+--------------------+------------+-------------+--------------------+
| id  | phishlet    |      username      |      password      |   tokens   | remote ip   |        time        |
+-----+-------------+--------------------+--------------------+------------+-------------+--------------------+
| 37  | cisco       |                    |                    | captured   | 127.0.0.1   | 2023-12-08 15:10   |
| 41  | ms365       |                    |                    | captured   | 127.0.0.1   | 2023-12-13 11:55   |
+-----+-------------+--------------------+--------------------+------------+-------------+--------------------+
```

Figure 4.6: Sessions

4.2.6 Proxy

Proxy is an option to make the requests originate from a different IP address and possibly other useful purposes such as development. This proxy resides in front of the reverse proxy and fools the target webpage to thinking that it is interacting with a different IP address. Figure 4.7 shows a configuration of such a proxy within the program from the command-line interface.



```
: proxy

 enabled   : yes
 type      : https
 address   : 127.0.0.1
 port      : 3741
 username  :
 password  :
```

Figure 4.7: Proxy

4.2.7 Blacklist

The blacklist gives an option for not allowing certain IPs from accessing your server. Using this correctly can thus make it possible to avoid detection from the most common scanners by blacklisting them. The options for the blacklist are:

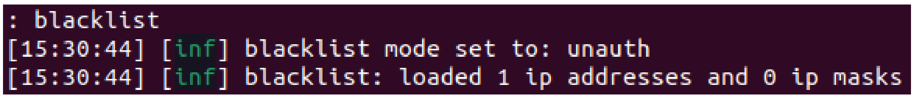
- all:
  This option blacklists all IP addresses that try to access the registered phishing domain. This can be useful to enable when the campaign has not started and the landing page visits by victims are not yet expected.

- unauth:
  This is the default option which blacklists all IP addresses of requests that target anything other than the exact phishing URL that is set up. When the victim clicks the lure URL that we provided for him he will be correctly redirected to the landing page. This option is crucial since it denies all scanners that do not know the lure address to directly access the page and extensively fingerprint it.

- noadd:
  This is similar to unauth, but this option only blocks all unauthorized requests instead of also adding them to the blacklist.

- off:
  This turns the blacklist off.

Figure 4.8 shows the mode the blacklist is set to, and it shows how many IP addresses and masks are currently loaded in the blacklist from previous interactions.



```
: blacklist
[15:30:44] [inf] blacklist mode set to: unauth
[15:30:44] [inf] blacklist: loaded 1 ip addresses and 0 ip masks
```

Figure 4.8: Blacklist

4.3 Phishlet Development

The phishlets of Evilginx3 are customizable files that make it possible to imitate different websites through the creation of them. These phishlets are built of [16]:

1. *header*:
   Used for specifying the version number that the phishlet is compatible with and giving a default redirect URL.

2. *params*:
   Makes it possible to define parameters for use in the phishlets. These parameters can then be used to create the final phishlet or more subsets of phishlets called child phishlets.

3. *proxy_hosts*:
   This option specifies the information that is used as phishing domain and URLs.

4. *sub_filters*:
   Filters are used to modify the original website content. What happens is that we host a modified version of the original website with filtered content. With this, we can make sure all the links point in the right direction and that we can also remove a websites anti-phishing security measures.

5. *auth_tokens*:
   This defines the tokens that need to be captured in order to count the phish as a success. It is possible to look at the cookies, body, and header to find the right tokens.

6. *credentials*:
   This specifies where the username and password are located within the HTTP POST. You can search through the post request through regular expression to any information necessary.

7. *auth_urls*:
   Since some cookie names are generated randomly there has to be another way to know if the phish was successful. We can specify an URL that only becomes accessible after logging in to check this.

8. *login*:
   Specifies the domain path of the login pages for the website that we are trying to imitate.

9. *force_post*:
   If we want to force users to authenticate with the remember option enabled, we can specify a way to enable this option within the phished website.

10. *js_inject*:
    This specifies which JavaScript scripts we want to inject into the page. This can also be useful for obfuscation.

11. *intercept*:
    This option specifies what HTTP requests we want to intercept and give a self generated response to.

4.4 The Attack

Attacking a certain victim starts with having them open a link and ends with them having entered their credentials and logged in. Thus, we need to first distribute the link to the target in order to execute the phishing attack. Evilginx3 only provides the phishing URL and the phishing attack, but it gives no possibility for creating emails or tracking the links. When the target clicks the link, they are redirected towards the landing page. They fill in their credentials and through a MITM attack we can log-in as the victim.

Since the user has connected to a MITM setup, they have no idea they visit the wrong website. They fill in the password field exactly the same as usual and the MITM server mirrors this action on the target website. The user continues interacting with the MITM webpage, if they do not do this, the session will be lost because a different session is setup for the target and the MITM client.

When the Evilginx3 toolkit is successfully setup with a phishlet, it is possible to effectively phish a user including the MFA step. The victim first clicks the link and arrives at the landing page. Then, the victim enters their password into the correct field and presses the corresponding login button. Typically on a new page, the website asks the victim for an MFA code. The user enters this in the same way, and the code will be saved along with the password. This is possible because Evilginx3 sets up a MITM interaction in between the victim and the target website. Through dynamic altering of the webpage and setting up the correct domains, the MITM interaction is maintained across interactions thus allowing the MFA code to be phished in the same way that the password and personal information is phished.

Chapter 5

Detection Methods

This thesis distinguishes four different types of detection of MITM phishing toolkits, which are obtained from analyzing a MITM phishing attack. Section 5.1 explains the detection surface that arises from the interaction of the toolkit to the internet and how the four different types of detection methods arise. Sections 5.2, 5.3, 5.4, and 5.5 explain the detection and evasion methods separately.

5.1 Detection Surface

Phishing attacks set up by Evilginx3 can be detected in four ways. This was found by analyzing the Evilginx3 setup. These four ways come from the different interactions of Evilginx3 in order to successfully perform MITM phishing.

The first way is the interaction of the Evilginx3 client with the real page in order to act like a proxy within the attack. The webserver that the real page is on has the ability to obtain information from the client and can possibly identify the Evilginx3 toolkit by fingerprinting it.

The second way is through the copying of the page by Evilginx3. Evilginx3 copies the webpage to its own webpage by using HTTP requests. It is possible for the target webpage developers to build measures into the code that detect if the page has been copied. For example, a frequently used piece of code is one that checks if the webpage URL matches the URL listed within the code. As shown in Chapter 4.1, Evilginx3 provides measures to changing this code. However, developers have the possibility to make this piece of code as obscure as possible, making it a significant amount of work to remove it by a possible attacker.

The third way is the interaction between the user and the Evilginx3 webserver. A user can check the webserver for authenticity because the webserver has to allow users to access the page in order for it to be functional. This can be done by fingerprinting public information on the webserver.

It is worth noting that detection can also be based on solely the domain name used to host the phishing website. This specific detection method is out of scope of this research and thus a non-related name is chosen for any tests and experiments.

Figure 5.1 Shows the different types of detection and their respective location of detection in the network. Website Fingerprinting can be performed from a scanner acting as a user. Timing Analysis and TLS Fingerprinting can both be done by the server and the user. URL Validation must be performed by the webpage itself.

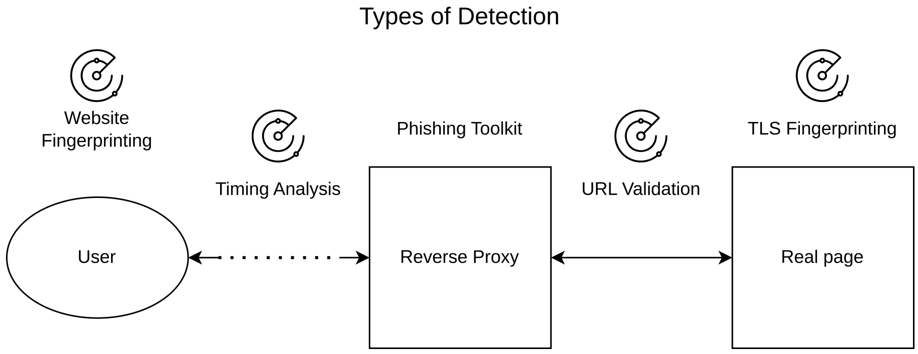


Figure 5.1: Setup

5.2 Website Fingerprinting

Website fingerprinting is a technique used for fingerprinting the application layer of the webpage in combination with other user accessible data. Section 5.2.1 explains how this method can be used for detection of phishing websites. Section 5.2.2 explains how this detection can theoretically be evaded.

5.2.1 Detection

To detect a phishing website we can check a range of different fingerprinting information. Combining this information effectively compares the website to known websites on the web in order to detect phishing attacks.

There are some indicators for knowing if a website is compromised. Sanglerdsinlapachai et al. (2010) gives a list of features according to CANTINA

[32]. CANTINA is a phishing detector that creates a fingerprint of suspicious webpages and compares them with common existing pages to detect if they are used for phishing. The following list gives a fair director on what the most well-known indicators of compromise entail according to CANTINA.

- Age of Domain:
  Most phishing pages do not exist for longer than a year. A relatively fresh page thus has much higher chances of being a phishing page. This fresh page thus is a good indicator of compromise considering the legitimate page should always be older.

- Known Images:
  A lot of phishing pages simply copy pieces from the website they are trying to imitate. This includes using the exact same image. If we remember a list of images of the most visited websites, we can compare them to the images on the domain of the phishing website. In case we find such a domain with the same images we can argue that there is a high probability that we are dealing with a phishing website.

- Suspicious URL:
  When characters like '@' or '-' are used within URLs, we might have a probability that we are dealing with a phishing website. Other things like holding a list of URLs of commonly visited website might also tell that the phishing website URL is suspicious.

- Suspicious Links:
  The same can be done for URLs that the links of the website point to. If any of these URLs is suspicious, then we can also increase the probability that a website is used for phishing.

- IP Address:
  We can check if the URL is tied to a fixed IP address. If we cannot find an IP or the IP is suspicious, we can increase the probability of the website being a phishing website.

- Dots in URL:
  If there are too many dots in the URL then this might be an indicator because attackers can use subdomains within their URL to resemble the target website URL. For example, microsoft.com can be imitated as mic.rosoft.com where attackers can get as creative as possible.

- Form:
  If an already suspicious page has input fields that collect a user's crucial information, it increases the probability that we are dealing with a phishing website.

- TF-IDF:
  Term frequency-inverse document frequency is proposed by CANTINA [48]. It focuses on trying to compare the contents of a phishing page with the real target website. It uses five keywords to send a query to a search engine in order to find if a website with similar content already exists. The most suspicious of these websites is then likely to be a phishing website.

Furthermore, generic scanners are constantly checking if any new certificates get issued for any domains. When this happens, the scanners try to access the webpage on this domain and then to crawl it. After performing a quick new setup of the Microsoft 365 phishlet we already notice a list of scanners going over this domain. As an example, we obtained the list shown in Figure 5.1 within 1 minute after obtaining a certificate for the webpage. This list shows what some of the scanners are that look at freshly distributed certificates.

IP	**Organization**
51.81.245.138	OVH US LLC
104.164.173.196	Armstrong Networks
134.122.89.242	DigitalOcean LLC
154.28.229.138	Aventice LLC
207.154.240.169	DigitalOcean LLC
159.203.63.67	DigitalOcean LLC
146.70.165.254	M247 Europe SRL
193.143.1.139	Proton66 OOO
96.44.189.108	QuadraNet Enterprises LLC
195.211.77.140	Lantek LLC
195.211.77.142	Lantek LLC

Table 5.1: Example Scanners

5.2.2 Evasion

Website fingerprinting detection can be evaded by either obfuscating and altering the code such that it does not look similar to the target website, while still being able to reverse proxy the website, or by preventing the scanners from accessing the website altogether. Figure 5.2 shows how a splash page can be put in front of the reverse proxy. This splash page is a small webpage that requires a user to press a button to be redirected to the reverse proxy. Scanners are typically not smart enough to interact with the page and will not notice anything suspicious about the page. If they are able to do this anyway, we can make the splash page more advanced by adding an input field or using a Capcha system instead which is a system to distinguish between a human and a bot.

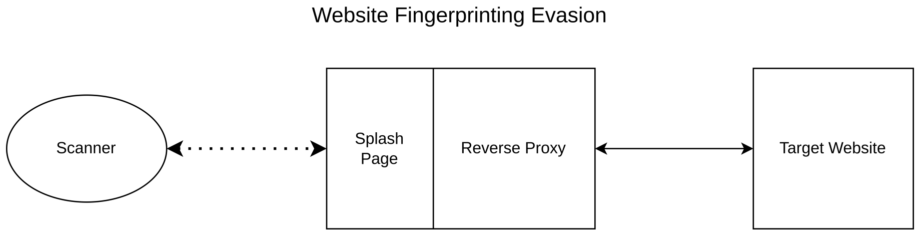


Figure 5.2: Website Fingerprinting Evasion

5.3 TLS Fingerprinting

TLS fingerprinting is a technique that looks at the TLS data that is exchanged in first contact with the server. This data contains version data and what protocols are supported. It is possible to use it as fingerprinting data, and this fingerprint quickly becomes unique in combination with other data.

5.3.1 Detection

An alternative way of fingerprinting looks at metadata of the connection. one of such methods is looking at TLS version information. In this research we also looked briefly at a tool called Phoca for this. From this tool we obtained a JSON file that lists features of the website that is fingerprinted.

- TLS Version Supported
- TLS Library

5.3.2 Evasion

This kind of fingerprinting is especially hard to evade because it means changing the web server. This makes it very suitable for detecting a phishing toolkit.

5.4 Timing Analysis

Timing analysis focuses on the timing between sent packages from the viewpoint of the client interacting with the server. This makes it possible for the client to detect that they are talking through a reverse proxy and not directly to the server.

5.4.1 Detection

A reverse proxy adds an extra layer of complexity to the web interaction. Figure 5.3 shows that the timing analysis method tries to detect a specific type of overhead between the interactions. Thus overhead exists out of the time that the target pages takes to interact with the reverse proxy plus the time that the reverse proxy takes to interact with the user. This time in comparison with a user trying to access the real webpage can possibly be an indicator that we are connected to a reverse proxy. This does however assume that we have a fingerprint of an already existing connection of the toolkit with the target webpage.

- TCP SYN/ACK RTT
- TLS Client Hello RTT
- Malformed TLS Client Hello RTT
- TLS Handshake Timing
- HTTP GET Request Timing
- HTTP GET Request w/o Host Header Timing
- Malformed HTTP GET Request Timing
- HTTPS GET Request Timing
- HTTPS GET Request w/o Host Header Timing
- Malformed HTTPS GET Request Timing
- TCP SYN/ACK to HTTP GET Request Timing Ratio
- TCP SYN/ACK to Malformed HTTP GET Request Timing Ratio
- TCP SYN/ACK to Malformed HTTPS GET Request Timing Ratio
- Malformed to Valid HTTPS GET Request Timing Ratio

5.4.2 Evasion

Reverse proxies are commonly used by web servers for different purposes. It can thus be hard to detect a webpage purely from timing detection. Many other factors can also have an effect on the timing detection. Additionally, it is possible to change the code to a different configuration to give the toolkit a different fingerprint. Since defenders do not have this new toolkit, it can become hard to use this method reliably.

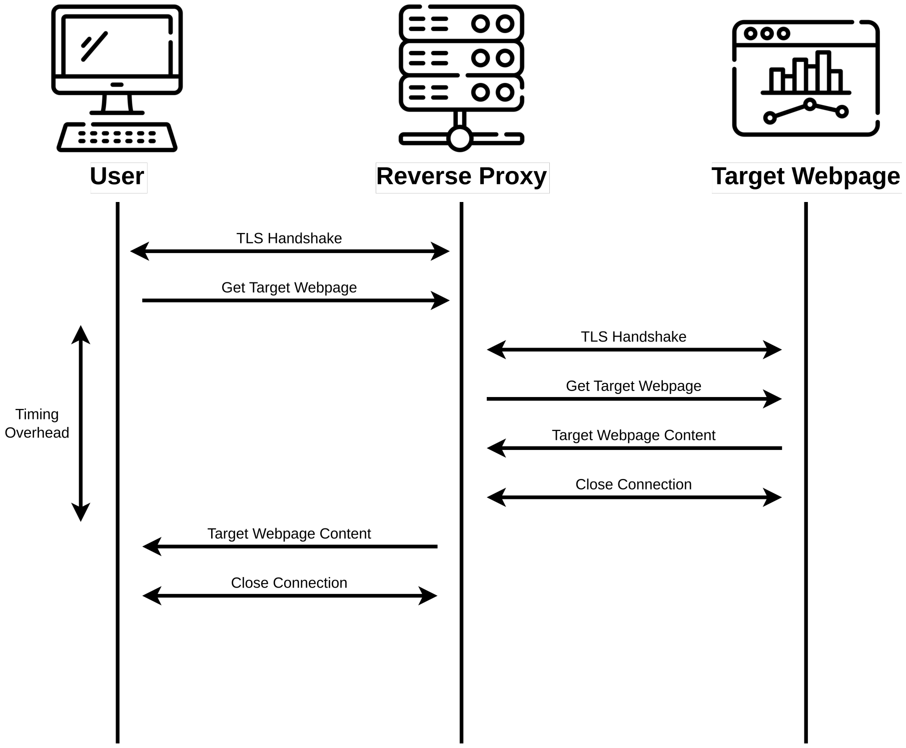


Figure 5.3: Timing Detection

5.5 URL Validation

URL Validation is a detection technique that can be implemented into the website itself to detect if it is running on a different domain and thus on a reverse proxy. It is often a piece of JavaScript code that is executed the moment the code is run on the clients browser. The client would thus not be able to access the website when the piece of code is present.

5.5.1 Detection

A piece of code can be used to check if the URL matches the known URLs that the webpage is allowed to be originating from.

Obfuscation is possible by encrypting the JavaScript code and by making the code hard to read for attackers. By obfuscation it would thus not be possible to cut out the piece of URL validation code by not effecting other parts of the code thus making the attack ineffective. This is a very strong protection mechanism because of its simplicity and effectiveness. However, this detection method may also affect legitimate reverse proxy and make the code harder to create and understand for developers.

5.5.2 Evasion

The only possible way that this detection method can be evaded is through alteration of the code. Theoretically, this is always possible for attackers. However, by using obfuscation it can become too complex to be worth the effort for attackers. An example of a website using such a method is that of LinkedIn. It implements a piece of JavaScript code into the web page that executes on the users machine. It checks the URL that it is currently running on and compares in to a URL stored inside the website's infrastructure. Upon seeing that the URL does not match the URL that has been stored, it sends a warning to both the user and the LinkedIn server that this is happening.

Chapter 6

Methodology

The focus of this experiment is to find out if Evilginx3 is detectable by current scanners on the web. From this result we then argue if Evilginx3 is a valid phishing tool for use in red teaming assessments. Section 6.1 explains how the system is set up with a working version of Evilginx3. Section 6.2 lists what webpages are targeted in the experiment and why. Section 6.3 lists what scanners are targeted in the experiment and what methods are used to target them. Section 6.4 describes the strategy used to test phishing detection with the explained setup.

6.1 System Setup

Setting up a working Evilginx3 instance for phishing users requires several design choices. These design choices include the operating system, the infrastructure, the IP address, the domain name, and the internal structure of Evilginx3.

- **Operating System:**
  Evilginx3 can be setup either on Linux or Windows systems. For this experiment we used a server running a Linux operating system for which we used Ubuntu.
  - Operating system: Linux Ubuntu 20.04.6 LTS
- **Infrastructure:**
  For both security and convenience we used a virtualization program called Docker.
  - Vitualization system: Docker v24.0.5
- **IP address:**
  The IP address is used to tie the server to a network location.

  - IP address: 136.144.188.49

- **Domain Name:**
  For the experiment we used only one domain name although it would have been preferable to use multiple domain names. It is not clear how much the effect of the domain name is on detection, but the used domain name deems effective.

  - Domain name: `myfirstblog.nl`

- **Evilginx3 Structure:**
  Evilginx3 is setup with the most basic option to allow for alterations as we are detected. Since we did not get detected during the tests, there was no need to alter them.

  - Lures: Standard options
  - Redirectors: Off
  - Redirection: Youtube - Rick Astley
  - Proxy: Off
  - Blacklist: Unauth

6.2 Target Webpages

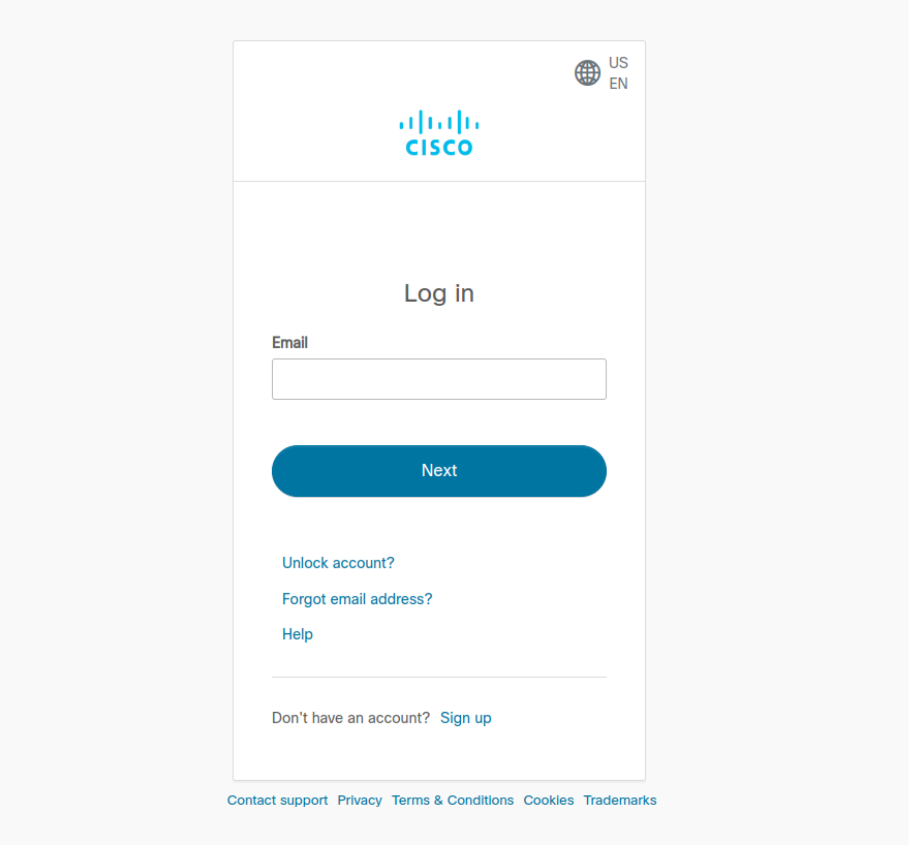
For the purposes of the Secura red team and because we had to limit the scope as of time and resource constraints, we decided that the most interesting websites to imitate and check detection of are `https://www.cisco.com/` and `https://www.microsoft.com/`. This section further explains the websites, the companies and their possibilities, and how the phishlet is created.

6.2.1 Cisco

Cisco provides many services including cloud based services allowing access to important processes and data. A signification amount of companies use Cisco to host their servers and give them easy access to online management services. This makes it a valuable target for hackers. Testing an organizations phishing security against phishing Cisco credentials can thus give valuable information. Figure 6.1 shows what the Cisco login page looks like. The user first needs to enter an email address and only afterwards can enter the password followed by the MFA code.

6.2.2 Microsoft

Microsoft is the number one largest company in the world by market cap [6]. Obtaining credentials for this service gives the attack access to `office.com`



US
EN

Log in

Email

Unlock account?

Forgot email address?

Help

Don't have an account? Sign up

Contact support Privacy Terms & Conditions Cookies Trademarks

Figure 6.1: Cisco Login Page

containing Outlook, Teams, Word, and more of office 365 apps. This means that an attacker would be able to access files and read emails by using the toolkit on this single authentication system. Figure 6.1 shows what the Microsoft 365 login page looks like. Again, the user has to enter an email and only then the password followed by the MFA code.

6.3 Attracting Scanners

In order to test the detection of Evilginx3, we tested the setup within a real environment. We opened the website to the internet and made sure that no firewall is preventing any access. We set up certificates, and we then sent the phishing URL through email to different email managers. We also sent the phishing URL to different scanners focused on finding malicious websites.

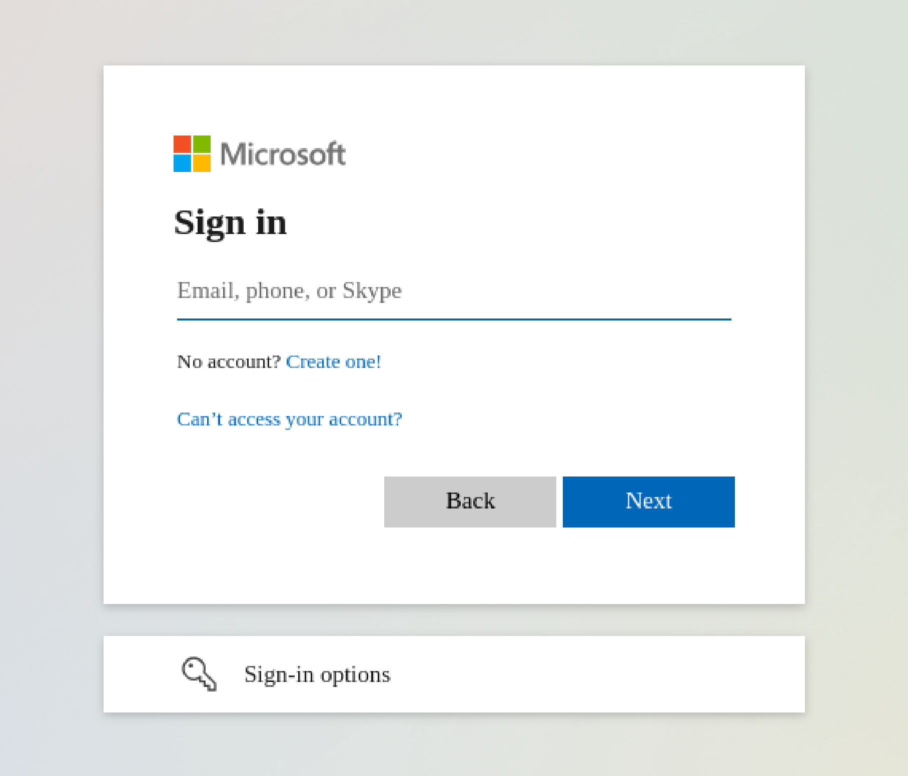


Figure 6.2: Microsoft Login Page

6.3.1 Mailing

We tested the scanners arising from sending an email to a Gmail account opened via `google.com` and an Hotmail account opened via `office.com`. We crafted this mail by using a generic email address and subject name and placing the phishing lure in the content section. We also tried making the phishing mail more suspicious by altering the content of the email. However, no results were found in difference of scanners targeting the phishing page.

6.3.2 Websites

By providing the phishing URL to three divergent services we tried to cover most of the important scanners that are easily activated for the URL. These are:

- `https://www.virustotal.com/gui/home/upload`
- `https://polyswarm.network/scan`
- `https://metadefender.opswat.com/`

We entered the lure URL for both the phishlets into the websites input field and clicked the start scanning button. We kept doing this everyday for the phishlet that was activated.

6.3.3 Open Source

This research also considered an open source scanner. We looked specifically at Phoca [19] because of its focus on network aspects for detecting MITM phishing toolkits. It lists various indicators of compromise that it uses to label a website as phishing or not. It does this by comparing the webpage to previously saved fingerprints of phishing toolkits.

6.4 Strategy

For the research we set up the two phishlets on a single Evilginx3 instance. We left all configuration options at the default and set the hostname to our domain `myfirstblog.nl`. We set the server open to the internet by disabling our firewall and activated the Cisco Systems phishlet. After the toolkit obtained the certificates from `letsencrypt.org` the first scanners already sent requests to the toolkit. After a few hours we tested the same URL on the three detection websites specified in Section 6.3.2. We then continued testing our URL on these websites everyday for two weeks. After finding no signs of detection, we used the same strategy for the Microsoft 365 phishlets while using the same domain but a different lure URL.

Chapter 7

Results

The experiment proves that Evilginx3 in its default configuration was not detected by scanners in the first four weeks of being activated considering the Cisco Systems phishlet, and is hard to detect considering the Microsoft 365 phishlet. Section 7.1 presents the main findings from the experiment. Section 7.2 discusses the importance of the results and its limitations.

7.1 Main Findings

Results show we found only seven scanners that could specifically detect the Microsoft 365 phishing page. It is unlikely that there are scanners present in the current state of the internet that can specifically detect the Evilginx3 phishing toolkits presence without access to the webpage itself. The experiment showed that it is hard to detect a phishing page or to tell it apart from a real reverse proxy. From the results we found that even after four weeks of thorough scanning of the toolkit's URL, none of the websites providing the scanners show to have found any malicious intentions on the Cisco Systems website, while we have shown in Section 4.5 that we are able to phish users credentials effectively through its use. The website also does not appear on any active block list and thus it is still possible to phish a user after two weeks of up-time. The seven scanners that detected the Microsoft 365 webpage did not report the site to any block lists. By using a different lure the detection was no longer possible.

The first two detections happened on the 10th day after testing the setup with the Microsoft 365 phishlet. VirusTotal showed four scanners that successfully labeled the website as a phishing website. These scanners are: BitDefender, Fortinet, G-Data, and VIPRE. As stated before, our phishing toolkit was still accessible by the browser and showed no signs of being on any block list. Two weeks later three more detections showed up from providing the URL into VirusTotal. These scanners are: alphaMountain.ai, CyRadar, and Sophos.

7.2 Discussion

The results obtained from the experiment are significantly useful for red teaming assignments because of their focus on detection evasion. From these results we can conclude that the hypothesis that a scanner implementing a similar technique to Phoca is presently scanning the web is unlikely in the current state of the web. We also noticed that the Cisco Systems Phishlet is much harder to detect than the Microsoft 365 Phishlet. This is likely due to Microsoft 365 being a company with a larger footprint, and that their login page is thus more often fingerprinted.

Limitations are present because in practice, detection takes time and many detection mechanisms are not public. According to previous research, detection may happen after more than two weeks of exposure to online scanners [19]. Thus, setting up an extensive experiment would require a different research strategy with access to additional resources. It is also interesting to create more phishlets for more different target website to compare which websites are mostly fingerprinted. Another limitation is that we did not focus on specific security program virus scanners that are scanning all emails within organizations. This leaves us with limited knowledge of detection in real scenarios when doing red team assessments.

The obtained results are especially useful to give research on the use and detection of MITM phishing toolkits a head start. Future toolkit will implement more advanced options and future detection techniques will use different ways of searching the web. Additionally, red teaming assessments will be able to use the Evilginx3 phishing toolkit from consulting this thesis.

Chapter 8

Future Work

This research identifies five different future work opportunities considering the detection of MITM phishing toolkits. Firstly, scanners are not optimized for the toolkits while research suggests that they have the possibility to detect them effectively. Secondly, research also suggests that proper obfuscation of the URL validation code can make it significantly hard for attackers to use this phishing method. Additionally, Evilginx3 can be combined with Gophish in order to utilize its tracking functionality for more effective phishing campaigns. Furthermore, a future version of Evilginx will implement a new function called evilpuppet which will add new phishing capabilities. Finally, recent rise in machine learning possibilities provide new opportunities for both phishing detection and evasion that can be researched.

8.1 Scanners

While this research found that the current version of Evilginx3 cannot be detected by the scanners on the web, it is no guarantee that it will not be detected in the future or at the present moment. By creating or analyzing scanners specialized at phishing toolkits, we can acquire a more clear view on detection and evasion possibilities in order to prevent detection altogether.

Additionally, more scanners may exist than the ones that are covered in this thesis. For example, virus vendors exist that can be implemented into an organization's infrastructure. These vendors may provide additional scanners than the ones publicly exposed on the web right now. These scanners may be more specialized and thus be able to detect the phishing toolkit.

Furthermore, it is interesting to redo the experiment within one to two years to see the advancements of MITM phishing detection on the web. When better detection methods are in place and more properly utilized it is possible to do a more wide experiment that would test different detection evasion techniques for their effectiveness.

8.2 Obfuscation

There is a wide range of obfuscation options for services to hide their code from attackers. As shown, these options effectively prevent malicious reverse proxies from sitting before their websites. By looking into these options and the new ones that are used in the future we can improve detection and find more evasion techniques to utilize in our phishing campaigns.

8.3 Evilgophish

Explore opportunities to improve on Evilgophish and to create a fully working and optimal phishing tool for red teaming [13]. It is also possible to find some other way of using an email phishing tool in combination with Evilginx3. Exploring these options gives red teamers a more useful tool for imitating advanced attackers targeting the organization.

8.4 Evilpuppet

The new upcoming version of Evilginx will include puppeteer software to simulate a user using the target website instead of pretending to be a mere reverse proxy. This advanced way of phishing can allow for even more interesting attacks. Research on the possibilities of this advanced setup may show even more uses for the red team.

8.5 Machine Learning

Machine learning methods are constantly improving and beginning to be well-developed enough that they can effectively be used in phishing and phishing detection mechanisms. In order to protect society from the dangers of more advanced phishing attacks, it is important that these machine learning advancements get investigated in order to understand what they can and cannot be used for. The previous research in combination with this research provides a basis for this [29, 45, 30].

Chapter 9

Conclusions

This research explained how Evilginx3 works and how it can be used for red teaming assessments. Additionally, it identified four different methods in which Evilginx3 can be detected. These are website fingerprinting, TLS fingerprinting, timing analysis, and URL validation. Website fingerprinting is only possible when the cloaking from Evilginx3 is overcome. When it is, this research shows that it is possible to detect Evilginx3 with online scanners. Thus, organizations can protect themselves against MITM phishing attacks by using email scanners and security software that checks links Evilginx3 does not provide extra options to counter this. TLS fingerprinting should best be used in combination with network timing analysis in order to detect cloaked MITM phishing toolkits present on the web. URL validation seems a valid protection method when properly obfuscated. Its drawbacks are that the attack may always find a way to remove the piece of code responsible for the detection even though it is obfuscated. Detection solutions implementing these methods have been shown effective at detecting these toolkits by other research. However, as this research discovered, these solutions are not actively used for the scanning of possible phishing websites. All these ways can and should be combined in order to provide a good phishing protection for organizations. As such, Evilginx3 is a useful phishing toolkit that can effectively be used for red teaming assignments. Furthermore, it is important to also think about the existence of detection methods and how they might be evaded in the future. Considering OPSEC, it is important that website fingerprinting is minimized as much as possible in the first place by only activating phishing links when it is sure that a user is going to interact with it. Actively altering the website is not recommended because it is a complex and time-consuming process.

Bibliography

[1] Regenscheid. A and Galluzzo. R. Nist-phishing-resistance. https://www.nist.gov/blogs/cybersecurity-insights/phishing-resistance-protecting-keys-your-kingdom.

[2] Mohamed Alsharnouby, Furkan Alaca, and Sonia Chiasson. Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies*, 82:69–82, 2015.

[3] Aware7. Phishing tools. https://aware7.com/blog/the-12-best-tools-for-phishing-simulations/.

[4] Wylie Bayes. Bypassing mfa with gophish and evilginx2. https://wyliebayes.com/bypassing-mfa-with-gophish-and-evilginx2/.

[5] Neeranjan Chitare, Lynne Coventry, and James Nicholson. "it may take ages": Understanding human-centred lateral phishing attack detection in organisations. In *Proceedings of the 2023 European Symposium on Usable Security*, EuroUSEC '23, page 344–355, New York, NY, USA, 2023. Association for Computing Machinery.

[6] Companiesmarketcap. Market cap. https://companiesmarketcap.com/.

[7] Dataprot. Phishing statistics. https://dataprot.net/statistics/phishing-statistics/.

[8] Justin E. Doak, Joe B. Ingram, Sam A. Mulder, John H. Naegle, Jonathan A. Cox, James B. Aimone, Kevin R. Dixon, Conrad D. James, and David R. Follett. Tracking cyber adversaries with adaptive indicators of compromise. In *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 7–12, 2017.

[9] Piotr Duszyński. Modlishka. https://github.com/drk1wi/Modlishka.

[10] Ayman El Aassal and Rakesh Verma. Spears against shields: Are defenders winning the phishing war? In *Proceedings of the ACM International Workshop on Security and Privacy Analytics*, IWSPA '19, page 15–24, New York, NY, USA, 2019. Association for Computing Machinery.

[11] Osama Ellahi, Muhammad Umer, Ahmed Raza, and Kashif Rehman. Analyzing 2fa phishing attacks and their prevention techniques. In *2022 International Conference on Smart Information Systems and Technologies (SIST)*, pages 1–6, 2022.

[12] Bedra et al. Authentication system design. `https://cybersecurity.ieee.org/blog/2016/06/02/design-best-practices-for-an-authentication-system/`.

[13] Dylan Evans. Evilgophish. `https://github.com/fin3ss3g0d/evilgophish`.

[14] Fortinet. Technical indicators of compromise. `https://www.fortinet.com/resources/cyberglossary/indicators-of-compromise`.

[15] Guang-Gang Geng, Zhi-Wei Yan, Yu Zeng, and Xiao-Bo Jin. Rrphish: Anti-phishing via mining brand resources request. In *2018 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–2, 2018.

[16] Kuba Gretzky. Evilginx guide. `https://help.evilginx.com/docs/guides/sessions`.

[17] Aaron Henricks and Houssain Kettani. On data protection using multifactor authentication. In *Proceedings of the 2019 International Conference on Information System and System Management*, ISSM 2019, page 1–4, New York, NY, USA, 2020. Association for Computing Machinery.

[18] Luke Irwin. The 5 most common types of phishing attack. `https://www.itgovernance.eu/blog/en/the-5-most-common-types-of-phishing-attack`.

[19] Brian Kondracki, Babak Amin Azad, Oleksii Starov, and Nick Nikiforakis. Catching transparent phish: Analyzing and detecting mitm phishing toolkits. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 36–50, New York, NY, USA, 2021. Association for Computing Machinery.

[20] Suiqiang Deng Lucas Hu, Howard Tong and Alex Starov. Man-in-the-middle phishing. `https://unit42.paloaltonetworks.com/meddler-phishing-attacks/`.

[21] Jared James Meyers, Derek L. Hansen, Justin S. Giboney, and Dale C. Rowe. Training future cybersecurity professionals in spear phishing using sieve. In *Proceedings of the 19th Annual SIG Conference on Information Technology Education*, SIGITE '18, page 135–140, New York, NY, USA, 2018. Association for Computing Machinery.

[22] Microsoft. Ai and phishing. `https://www.microsoft.com/en-us/microsoft-365-life-hacks/privacy-and-safety/how-ai-changing-phishing-scams`.

[23] Katelin A Moul. Avoid phishing traps. In *Proceedings of the 2019 ACM SIGUCCS Annual Conference*, SIGUCCS '19, page 199–208, New York, NY, USA, 2019. Association for Computing Machinery.

[24] Nginx. Nginx explained. `https://www.nginx.com/resources/glossary/nginx/`.

[25] Nist. Nist-phishing. `https://www.nist.gov/itl/smallbusinesscyber/guidance-topic/phishing`.

[26] OWASP. Owasp authentication. `https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html`.

[27] Srushti Patil and Sudhir Dhage. A methodical overview on phishing detection along with an organized way to construct an anti-phishing framework. In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pages 588–593, 2019.

[28] Thanasis Petsas, Giorgos Tsirantonakis, Elias Athanasopoulos, and Sotiris Ioannidis. Two-factor authentication: Is the world ready? quantifying 2fa adoption. In *Proceedings of the Eighth European Workshop on System Security*, EuroSec '15, New York, NY, USA, 2015. Association for Computing Machinery.

[29] Majed Rajab. An anti-phishing method based on feature analysis. In *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing*, ICMLSC '18, page 133–139, New York, NY, USA, 2018. Association for Computing Machinery.

[30] Sadia Parvin Ripa, Fahmida Islam, and Mohammad Arifuzzaman. The emergence threat of phishing attack and the detection techniques using machine learning models. In *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, pages 1–6, 2021.

[31] Julio Alexander Rodríguez-Corzo, Alix E. Rojas, and Camilo Mejía-Moncayo. Methodological model based on gophish to face phishing

vulnerabilities in sme. In *2018 ICAI Workshops (ICAIW)*, pages 1–6, 2018.

[32] Nuttapong Sanglerdsinlapachai and Arnon Rungsawang. Web phishing detection using classifier ensemble. In *Proceedings of the 12th International Conference on Information Integration and Web-Based Applications & Services*, iiWAS '10, page 210–215, New York, NY, USA, 2010. Association for Computing Machinery.

[33] Himani Sharma, Er. Meenakshi, and Sandeep Kaur Bhatia. A comparative analysis and awareness survey of phishing detection tools. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 1437–1442, 2017.

[34] Anil Somayaji, David Mould, and Carson Brown. Towards narrative authentication: Or, against boring authentication. In *Proceedings of the 2013 New Security Paradigms Workshop*, NSPW '13, page 57–64, New York, NY, USA, 2013. Association for Computing Machinery.

[35] Sprocketsecurity. Setup gophish for detection evasion. `https://www.sprocketsecurity.com/resources/never-had-a-bad-day-phishing-how-to-set-up-gophish-to-evade-security-controls`.

[36] Alex Sumner and Xiaohong Yuan. Mitigating phishing attacks: An overview. In *Proceedings of the 2019 ACM Southeast Conference*, ACM SE '19, page 72–77, New York, NY, USA, 2019. Association for Computing Machinery.

[37] Jonas Tzschoppe and Hans Löhr. Browser-in-the-middle - evaluation of a modern approach to phishing. In *Proceedings of the 16th European Workshop on System Security*, EUROSEC '23, page 15–20, New York, NY, USA, 2023. Association for Computing Machinery.

[38] Enis Ulqinaku, Daniele Lain, and Srdjan Capkun. 2fa-pp: 2nd factor phishing prevention. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '19, page 60–70, New York, NY, USA, 2019. Association for Computing Machinery.

[39] W3C. Anti-phishing. `https://www.w3.org/2005/Security/usability-ws/papers/37-google/`.

[40] Rick Wash. How experts detect phishing scam emails. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW2), oct 2020.

[41] Wikipedia. Cognitive bias. `https://en.wikipedia.org/wiki/Cognitive_bias`.

[42] Wikipedia. Indicator of compromise. https://en.wikipedia.org/wiki/Indicator_of_compromise.

[43] Wikipedia. Phishing. https://en.wikipedia.org/wiki/Phishing.

[44] Wikipedia. Social engineering. https://en.wikipedia.org/wiki/Social_engineering_(security).

[45] Victor Zeng, Shahryar Baki, Ayman El Aassal, Rakesh Verma, Luis Felipe Teixeira De Moraes, and Avisha Das. Diverse datasets and a customizable benchmarking framework for phishing. In *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*, IWSPA '20, page 35–41, New York, NY, USA, 2020. Association for Computing Machinery.

[46] Penghui Zhang, Adam Oest, Haehyun Cho, Zhibo Sun, RC Johnson, Brad Wardman, Shaown Sarker, Alexandros Kapravelos, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, Adam Doupé, and Gail-Joon Ahn. Crawlphish: Large-scale analysis of client-side cloaking techniques in phishing. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1109–1124, 2021.

[47] Penghui Zhang, Zhibo Sun, Sukwha Kyung, Hans Walter Behrens, Zion Leonahenahe Basque, Haehyun Cho, Adam Oest, Ruoyu Wang, Tiffany Bao, Yan Shoshitaishvili, Gail-Joon Ahn, and Adam Doupé. I'm spartacus, no, i'm spartacus: Proactively protecting users from phishing by intentionally triggering cloaking behavior. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 3165–3179, New York, NY, USA, 2022. Association for Computing Machinery.

[48] Yue Zhang, Jason I. Hong, and Lorrie F. Cranor. Cantina: A content-based approach to detecting phishing web sites. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 639–648, New York, NY, USA, 2007. Association for Computing Machinery.

Appendix A

Phishlets

A.1 Cisco Systems

```
min_ver: '3.2.0'
proxy_hosts:
  - {phish_sub: 'id', orig_sub: 'id', domain: 'cisco.com',
    session: true, is_landing: true, auto_filter: true}
  - {phish_sub: 'apps-id', orig_sub: 'apps-id',
    domain: 'cisco.com', session: false, is_landing: false,
    auto_filter: true}
  - {phish_sub: 'www', orig_sub: 'www', domain: 'cisco.com',
    session: false, is_landing: false, auto_filter: true}
sub_filters: []
auth_tokens:
  - domain: 'id.cisco.com'
    keys: ['sid']
credentials:
  username:
    key: ''
    search: '"username":"([^"]*)'
    type: 'json'
  password:
    key: ''
    search: '"password":"([^"]*)'
    type: 'json'
  custom:
    key: ''
    search: '"passCode":"([^"]*)'
    type: 'json'
login:
  domain: 'id.cisco.com'
  path: '/'
```

A.2 Microsoft 365

```
min_ver: '3.2.0'
proxy_hosts:
  - {phish_sub: 'login', orig_sub: 'login',
    domain: 'microsoftonline.com', session: true,
    is_landing: true}
  - {phish_sub: 'logon', orig_sub: 'login', domain: 'live.com',
    session: true, is_landing: false}
  - {phish_sub: 'www', orig_sub: 'www', domain: 'office.com',
    session: true, is_landing: false}
sub_filters: []
auth_tokens:
  - domain: '.live.com'
    keys: ['.*:regexp']
  - domain: 'live.com'
    keys: ['.*:regexp']
  - domain: '.login.live.com'
    keys: ['.*:regexp']
  - domain: 'login.live.com'
    keys: ['.*:regexp']
  - domain: '.login.microsoftonline.com'
    keys: ['.*:regexp']
  - domain: 'login.microsoftonline.com'
    keys: ['.*:regexp']
  - domain: '.microsoft.com'
    keys: ['.*:regexp']
  - domain: 'microsoft.com'
    keys: ['.*:regexp']
  - domain: '.office.com'
    keys: ['.*:regexp']
  - domain: 'office.com'
    keys: ['.*:regexp']
  - domain: '.www.office.com'
    keys: ['.*:regexp']
  - domain: 'www.office.com'
    keys: ['.*:regexp']
auth_urls:
  - '/landingv2'
credentials:
  username:
    key: ’login’
    search: ’(.*)’
    type: ’post’
  password:
    key: ’passwd’
    search: ’(.*)’
    type: ’post’
  custom:
    key: ’otc’
    search: ’(.*)’
    type: ’post’
login:
  domain: ’login.microsoftonline.com’
  path: ’/’
```