MASTER THESIS

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program IT-Security

Automated Deployment of Phishing Infrastructure

By: Petar Kosic, BA

Student Number: 2010303028

Supervisors: Dipl. Ing. Andreas Happe
Ing. Sebastian Bicchi

Vienna, April 30, 2022

Declaration

“As author and creator of this work to hand, I confirm with my signature knowledge of the relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz /Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I hereby declare that I completed the present work independently and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director if there should be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I further declare that up to this date I have not published the work to hand nor have I presented it to another examination board in the same or similar form. I affirm that the version submitted matches the version in the upload tool.“

Vienna, April 30, 2022

Signature

CYBER	**Unterzeichner**	Petar Kosic
	Datum/Zeit-UTC	2022-04-30T20:29:16+02:00
	Prüfinformation	Informationen zur Prüfung der elektronischen Signatur finden Sie unter: https://www.signaturpruefung.gv.at
Hinweis	Dieses mit einer qualifizierten elektronischen Signatur versehene Dokument hat gemäß Art. 25 Abs. 2 der Verordnung (EU) Nr. 910/2014 vom 23. Juli 2014 ("eIDAS-VO") die gleiche Rechtswirkung wie ein handschriftlich unterschriebenes Dokument.	

Kurzfassung

Phishing ist ein Angriffsvektor für Kriminelle, um Zugangsdaten oder vertrauliche Unterlagen zu stehlen. Sicherheitsexpert:innen nutzen diese Techniken mit der legalen Zustimmung von Unternehmen, um deren Resilienz gegen solche Attacken zu stärken und in der Belegschaft mehr Bewusstsein für derartige Bedrohungen zu schaffen.

Für das Ausführen von Phishing-Assessments ist eine eigene Infrastruktur notwendig, die es ermöglicht, Emails auszusenden und Phishing-Webseiten zur Verfügung zu stellen. Es wurde ein Anforderungskatalog erstellt, um existierende und neue Lösungen vergleichen zu können. Wenngleich verschiedene Open Source Phishing Software Lösungen Phishing Infrastruktur ausrollen können, bietet keine mehrere Phishing Tools in Kombination mit einem sicheren, selbst-gehosteten Mailserver mit automatisierter DNS-Eintrag-Konfiguration. Diese Arbeit präsentiert einen Prototypen mit Fokus auf Datensouveränität. Dieser konfiguriert Phishing-Infrastruktur mit mehreren Phishing-Tools und einem sicher konfigurierten Mailserver auf selbst kontrollierten Servern.

Aufgrund der Überlegung den Prototypen zu veröffentlichen, kam die Frage der ethischen Vereinbarkeit bezüglich der Veröffentlichung auf. Deswegen wurde eine Umfrage erstellt und veröffentlicht, um Meinungen von IT-Security-Expert:innen für eine Diskussion zu sammeln. Weiters wurden die Expert:innen im Rahmen der Umfrage gefragt, ob sie den Prototypen evaluieren würden und ob sie ihn selbst Open Source veröffentlichen würden.

Dabei stellte sich heraus, dass über 80% der Befragten es akzeptabel finden, Code für das automatische Ausrollen von Phishing-Infrastruktur Open Source zu veröffentlichen. Bezüglich der technischen Komponente dieser Arbeit wurde ein Prototyp auf Basis von Ansible implementiert und die Konfiguration der ausgerollten Infrastruktur erfolgreich mittels öffentlicher Test Suites verifiziert. Möglichkeiten für zukünftige Erweiterungen inkludieren weitere Phishing-Software und die Unterstützung automatisierter Hardware-Ressourcen-Konfiguration mittels Lösungen wie z.B. Terraform.

Schlagworte: Phishing, Automatisierung, Ansible, Umfrage, Ethik

Abstract

An infrastructure is needed to send out emails and provide Phishing websites to conduct Phishing engagements. We designed a requirement catalog for Phishing infrastructure to enable a comparison between existing and new solutions. Various open-source tools already exist that can deploy Phishing infrastructure; however, none of the solutions offers multiple Phishing tools that include a self-hosted secure mail server with automated DNS record configuration. This work introduces a novel prototype with a data sovereignty-driven focus to deploy a Phishing infrastructure with multiple Phishing tools on a self-controlled server, including a securely configured mail server.

The fact that this prototype may be published has raised concerns with regard to whether it is ethically justifiable or not to do so. Therefore, we designed and conducted a survey to collect opinions from IT Security professionals on this problem for discussion. Additionally, we asked the professionals in the questionnaire if they would evaluate our prototype and open-source it themselves.

As a result, we discovered that over 80% of the survey participants find it acceptable to open-source deployment solutions for Phishing infrastructure. Regarding the technical component of this work, we implemented the prototype using Ansible and successfully verified the configuration of the infrastructure deployed with public test suites. Possibilities for future work may include adding additional Phishing software and supporting hardware resource allocation through solutions like Terraform.

Keywords: Phishing, Automation, Ansible, Survey, Ethics

Acknowledgements

Der größte Dank geht an meine Lebensgefährtin, welche mich immer unterstützt, aufgemuntert und motiviert hat, Ziel abzuschließen. Mein Dank gebührt insbesondere meinen Eltern, ohne deren Unterstützung ich dieses Ziel nicht erreicht hätte. Danke Mama und Papa!

Ein großer Dank geht an meinen Betreuer Andreas Happe. Sein Wissen und Rat waren eine Bereicherung für mich und haben mir beim Verfassen dieser Arbeit sehr geholfen. Die Diskussionen und Inputs waren äußerst wertvoll und haben zum Erfolg dieser Arbeit beigetragen. Ebenso geht mein Dank an meinen Zweitbetreuer Sebastian Bicchi, mit welchem ich gute Diskussionen geführt habe, die mir bei der Entstehung dieser Arbeit sehr geholfen haben.

Mein Dank geht auch an meine Kolleg:innen, die mich in diesen zwei Jahren begleitet haben. Meist im digitalen Raum, jedoch war auch dort ein guter Austausch, anregende Diskussionen und gegenseitige Motivation stets möglich.

Ebenso möchte ich mich herzlich bei André, Michael und Tonimir für ihr wertvolles Feedback bedanken.

Danke Lena für das produktive Arbeitswochenende.

Danke Klaudia und Clemens für den produktiven Tag bis tief in die Nacht.

Weiters geht ein Dank an alle unerwähnten Freund:innen, die mich bei dieser Arbeit unterstützt und motiviert haben.

Contents

1 Introduction

Phishing is one of the most common threats in the IT world. According to the *Proofpoint 2022 State of the Phish* report, 86% of organizations faced bulk Phishing attacks in 2021 [24, 50]. Such a high number of Phishing attacks are connected to security incidents in companies and the loss of customer data and confidential material, ultimately leading to reputation damage, lawsuits, and General Data Protection Regulation (GDPR) fines for the businesses affected. One way to improve companies' resilience against Phishing is to conduct assessments using dedicated Phishing infrastructure.

We present a feature catalog that was created to compare existing and future Phishing infrastructure solutions. Based on the catalog, we introduced our design of the prototype for the automated deployment solutions for Phishing infrastructure. Then, we present the implementation of our solution. Furthermore, the resulting prototype is verified, and a comparison against other similar solutions is conducted.

The thesis discusses whether open-sourcing deployment solutions for Phishing infrastructure is ethically justifiable. Therefore, a survey was designed and conducted to gather data from IT Security professionals as input for this research question. The answers are evaluated and discussed. The discussion examines considerations and concerns with regard to open-sourcing the prototype that is the result of this research.

Lastly, a conclusion is drawn, and possibilities for future work are discussed.

1.1 Motivation

Customers of IT Security consulting companies want to improve the resilience of their employees against social engineering because they are the first line of defence in the company. Phishing is one possible technique to make employees aware of social engineering attacks. Therefore, conducting Phishing assessments as IT Security professionals is part of the job. Phishing assessments can be divided into defensive and offensive types. The defensive type can be a classical awareness campaign to sensitize the employees about real threats by Phishing. If users click the Phishing link, they are, for example, informed that this was an awareness campaign. Additionally, information is presented on how the user could have identified the Phishing mail.

The other usual scenario is an offensive security engagement with the goal of stealing valid credentials. The credentials obtained are further used to penetrate the network and gain access to internal infrastructure and valuable information. Both scenarios are only legal with companies' written permission that legitimize such assessments.

In all the cases mentioned, emails need to be scheduled, sent out, and corresponding Phishing web pages must be hosted for the assessments. Phishing infrastructure provides the necessary tooling and is the basis of every Phishing campaign. As targets vary among customers, setting up dedicated Phishing infrastructure is a recurring task for IT Security experts.

It is beneficial to automate the setup routine to avoid mistakes during the setup. Besides reducing errors, it also saves time, which can be used for other related assessment tasks like Open Source Intelligence (OSINT) or story creation for the Phishing campaign.

During the initial research for this master thesis, multiple Phishing solutions available on GitHub were researched. However, every solution has some missing features. This motivated us to implement a solution that offers all features desired. These were data sovereignty centric and implied a self-hosted solution, including a mail server with a secure configuration. By handling customer data, the GDPR comes into play, and companies that are specialized in IT Security want and are obliged to have control over their customer data.

1.2 Research Questions

The research questions include two question groups: Automated deployment of self-hosted Phishing tools and the ethical consideration about open-sourcing the corresponding scripts.

1.2.1 Automated Deployment of self-hosted Phishing Tools

The following research questions are focused on the practical part of this master thesis. This part consists of implementing and verifying a prototype for the deployment of Phishing infrastructure.

- RQ 1: What are meaningful requirements for tools that automatically deploy Phishing solutions?
- RQ 2: What are potential deployment solutions based on the requirements?
- RQ 3: Can a secure mail server implementation improve the result of a mail testing suite by 25% versus a non-secure mail server?
- RQ 4: Would IT Security professionals evaluate such a solution, and what would be the reason behind their choice?

1.2.2 Ethical Considerations

It is planned to release the implemented solution as open-source to the public. We consider it important to discuss the ethical aspects and evaluate the opinions of the IT Security community about the ethical acceptance of the intended publication. Therefore, the following research question with corresponding sub-questions are discussed in this thesis:

- RQ 5: Is it ethical to open-source deployment code for automated Phishing infrastructure?

  RQ 5.1: What are the reasons of IT Security professionals who are for or against open-sourcing such code to the public?

  RQ 5.2: Why or why not would IT Security professionals open-source automation scripts for Phishing infrastructure themselves?

Research sub-questions RQ 5.1 and 5.2 may sound similar but are targeting different aspects. The support for the release of deployment code for automated Phishing infrastructure is independent of the motivation in regards to the publication by the participants under their name. Therefore, we wanted to explore if the support for release (RQ 5.1) indicates that participants would publish it by themselves (RQ 5.2).

1.3 Expected Contribution

By answering the research questions, we expected to contribute:

- A qualitative survey to evaluate the opinions of IT Security professionals about ethical considerations, interest in evaluation, and motivation regarding open-sourcing.
- A requirements catalog that can be used for comparison of Phishing infrastructure deployment solutions.
- An extendable prototype that deploys two Phishing software in combination with a secure configured mail server and a comparison with with existing solutions based on the requirements catalog.
- A proof that secure mail server configurations offer a benefit regarding (Phishing) mail acceptance compared to less secure ones.

1.4 Structure of the Work

This work is structured in multiple sections that focus on different parts of this thesis:

- Chapter 2 provides the reader with the necessary background information on Phishing and mail infrastructure.
- Chapter 3 presents state of the art of Phishing techniques and solutions.
- Chapter 4 summarizes three similar solutions that are related to our work.
- Chapter 5 gives an overview of the objectives and presents the corresponding methodology for reaching them.
- Chapter 6 provides insight into the designed requirement catalog.
- Chapter 7 discusses the chosen design for our prototype.
- Chapter 8 evaluates the prototype and the corresponding verification results.
- Chapter 9 describes the ethical considerations and discusses the survey results.
- Chapter 10 concludes the results of this thesis and proposes possibilities for future work.

2 Background

We present necessary background information for the reader in this section. First, Phishing, in general, will be discussed. Then we give an overview of Phishing techniques. We discuss mail server security in the following, including related mechanisms SPF, DKIM, and DMARC. Afterward, the ACME Protocol for TLS certificate issuing is explained. Finally, we provide information about the classification of IT Security teams.

2.1 Phishing

Today Phishing is the most common type of malicious email according to the Microsoft Digital Defense Report 2021 [41]. Targets of criminals are all types of companies, and even big corporations can encounter successful Phishing attacks. A prominent example of successful attacks by Phishing was an attack against Twitter in 2020 [53]. Another major incident that began with a Phishing attack was an attack on the Ukrainian power grid, which led to a power outage [1]. Additionally, besides companies, ordinary users are also targets, .e.g., in February 2022, several OpenSea users who possessed Non-fungible tokens (NFTs) were targeted by a Phishing attack [7]. The attacker tricked them by mail into executing a malicious contract code, which led to 17 users losing an estimated 1.7 million dollars worth of digital goods.

2.2 Phishing Techniques

There is a wide range of techniques to carry out Phishing. The following four are explained in detail because they can be executed with the Phishing tools of our prototype.

Spear Phishing

A Spear Phishing attack is a social engineering attack that targets a specific group of people. The goal is to gather private information like credentials, contact information, or other sensible internal company data. A successful Spear Phishing attack is an initial attack vector for further attacks. For example, obtained credentials can be used to gain access to the company's internal network or access more confidential data. By creating tailored mail for the targeted group, the success rate for successful exploitation is much higher compared to generic bulk Phishing campaigns.

Phishing Domain/URL (Impersonation/Homoglyph)

For a successful Phishing attack with fake Phishing pages, the victims need to be lured into believing the page is legitimate. When visiting a web page, the domain is checked by already trained users. Two techniques are presented below that increase the success rate of Phishing attempts.

The first technique is the impersonation of the company by acquiring domains that look like they belong to the company. This suggests to the user that the email and web page belong to the company. This can be, for example, another Top Level Domain (TLD) that is used instead of the original one. As a fictive example, `acme-corp.com` users should be Phished. Therefore, the domain `acme-corp.support` is registered if the domain is available, and the users are led to believe with a corresponding Phishing mail that the email sender and Phishing page are from the Support team of the company.

The second presented technique is the homoglyph attack. The concept behind this attack is that the attackers use homoglyph domains that look very similar or identical to the legitimate domain of the targeted organization. The simple way of exploitation is replacing similar characters in the URL. For example, `g` can be replaced by `q` or `i` by `l`. When comparing the domains `acme-homepage.com` with `acme-homepaqe.com` the replaced character may not be recognized by the victim. A more sophisticated approach is using Unicode (non-Latin) characters with internationalized domain names (IDNs). An algorithm exists that encodes the IDN domain into an ASCII character representation. It is defined in RFC 3492 [13] and is named *Punycode*. When replacing the ASCII character `a` with the identical looking Cyrillic letter `а` the domain will look identical, but technically they are different domains. The Punycode representation of `example.com` with a Cyrillic `а` equals to `xn-exmple-4nf.com`. Depending on the software, it might show the IDN Unicode representation with the non-Latin characters instead of the Punycode to the user. The Unicode representation might not be distinguishable by sight from the similar non-IDN with only Latin characters. Therefore, victims may believe they are on the correct homepage when visiting a website hosted under an IDN domain.

Page Clone

A simple Phishing method is the page clone. With this attack, the HTML code of a legit login page gets cloned. This technique is often used to clone a simple login form, where most of the time, just minor adaptations need to be done in the HTML code for the credential logging. The cloned page is often served over a similar domain, and the corresponding URL is sent out to the victim via Phishing email. When the victim accesses the page, they see a familiar login field. After submitting the credentials and clicking the login button, the victim is usually forwarded to the actual login page. Most of the time, people will think there might be a mistake and log in again. Still, the credentials are already logged in the background, and the attacker can reuse them for further attacks. This attack technique does not verify the captured credentials, and invalid credentials may be logged. The entire attack flow is depicted in Figure 1.

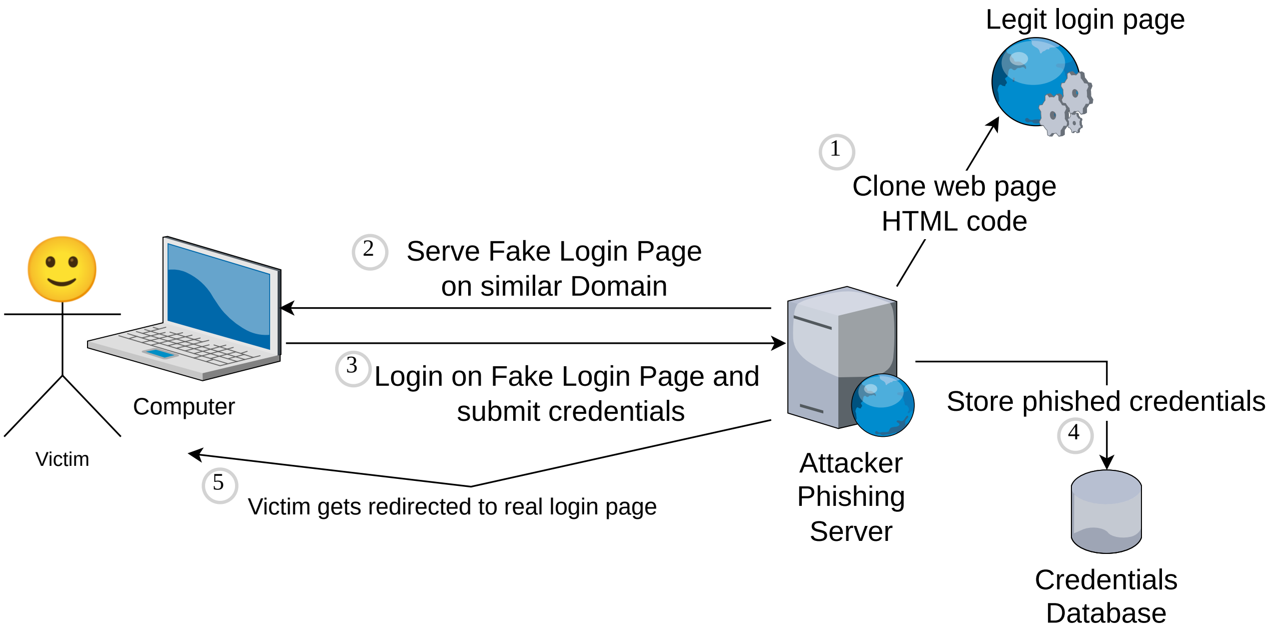

Legit login page
1
Clone web page HTML code
2
Serve Fake Login Page on similar Domain
3
Login on Fake Login Page and submit credentials
Computer
Victim
Attacker Phishing Server
Store phished credentials
4
Credentials Database
5
Victim gets redirected to real login page

Figure 1: Example of a simple Phishing attack using a cloned web page

This technique may not be suitable for complex login pages where some dedicated JavaScript code exists that handles the login procedure or if the login page requires two-factor authentication. For such cases, Reverse Proxying may be a suitable alternative.

Reverse Proxying (Two-Factor Authentication Bypass)

A more sophisticated Phishing technique offers a possibility to bypass most of the two-factor authentication (2FA) protections. This technique is usually used when the target website implements a 2FA authentication, or the login procedure is implemented with some complex JavaScript. The method uses a custom reverse proxy, which sits between the user and the target web service that needs a 2FA authentication. The victim submits the credentials to the malicious Phishing server, which forwards the credentials to the target web service. The proxy relays to the victim the 2FA prompt, and after submission of the user-provided token, the session is successfully established on behalf of the victim. Thus, the Phishing server obtains valid session data for the user account. The user does not realize that they just launched a session to their account via the attacker-controlled infrastructure. Now the attacker can interact with the service with the victim's session. This attack is a machine-in-the-middle (MITM) attack, where the third party stands between victim and server and interacts maliciously. Because the intermediate server just forwards all necessary credentials and tokens, most 2FA tokens like SMS, Push, and OTP are bypassed successfully. The execution of this attack is shown in Figure 2.

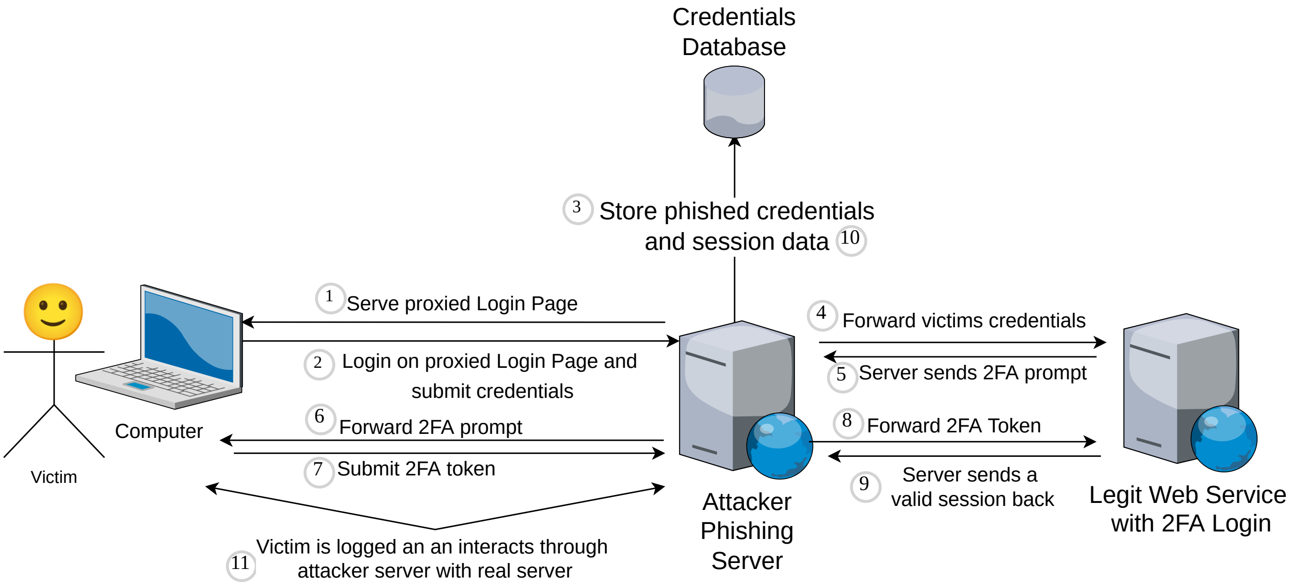

Credentials
Database
3 Store phished credentials
and session data 10
1 Serve proxied Login Page
2 Login on proxied Login Page and
submit credentials
6 Forward 2FA prompt
7 Submit 2FA token
Computer
Victim
11 Victim is logged an an interacts through
attacker server with real server
Attacker
Phishing
Server
4 Forward victims credentials
5 Server sends 2FA prompt
8 Forward 2FA Token
9 Server sends a
valid session back
Legit Web Service
with 2FA Login

Figure 2: Example of a sophisticated 2FA Phishing attack using a special reverse proxy

The 2FA mechanisms which are safe against such attacks are the CTAP1 (known as U2F) and CTAP2 of the FIDO2 framework [23]. They depend on a single-device hardware[1], like a security USB key, and are generally bound to the specific URL of the web service. Therefore, using a different URL and proxying the target web service to the victim does not work with this 2nd-factor option.

In the opposite to the simple page cloning attack, as described in subsubsection 2.2, credentials are not only logged but also verified with this technique because only with correctly provided credential and 2FA token the proxy establishes a valid session. However, the custom Reverse Proxy configuration is a more complex process than a simple HTML clone of a page. It needs to be noted that popular providers like Google detect such proxying attempts and are not easy proxy able without additional tooling and configuration.

2.3 Mail (Server) Security

Getting emails delivered today into users' mailboxes is not a straightforward task. Mail providers implement protection barriers like spam filters, anti-malware filters, and blocklists that need to be overcome. One of the main reason for this is the mass of unsolicited spam emails that accounts for over 45 % of the mail traffic in 2021, according to Kaspersky [52]. Therefore, an email must achieve a specific level of trustworthiness to be successfully delivered to a user's mailbox. If some minimum threshold is not reached, the email will bounce or land in the spam folder, where they are often overlooked. Sometimes emails are just silently discarded by the

[1] The FIDO Alliance proposal in 2022 does not enforce single-device binding anymore and presents a multi-device idea [28]. No conclusive decision was made at the time this thesis was written.

receiving mail server. Implementing solutions like SPF, DKIM, and DMARC provide the mail servers and their sent-out email a solid level of legitimacy. These improve the trustworthiness of emails by validating the sender and offering integrity for sent emails. The details for these three technical solutions are presented below.

SPF

The Sender Policy Framework (SPF) is a proposed standard for the detection of forged email sender addresses. It is defined in Request for Comments (RFC) 7208 [32]. By setting a DNS TXT record on the sending domain, the receiving mail server is able to verify if the sending mail server is allowed to send out the received email. The mail server can accept, refuse, or discard the email based on the result.
An example SPF record:

```
"v=spf1 ip4:192.0.2.0/24 ip4:198.51.100.123 a -all"
```

The `v` parameter defines the SPF version. With the parameter `ip4` the allowed IP addresses for sending are published and `a` allows the resolving A or AAAA record. `-all` means that mails from all other sources that not match the previous mechanisms must be rejected by the receiving mail server.

DKIM

The DomainKeys Identified Mail (DKIM) is an Internet Standard for authentication of emails and the detection of forged email sender addresses. The latest RFC for this standard is RFC 6376 [33]. DKIM provides integrity for sent-out emails by creating a digital signature of defined email fields. Similar to SPF, necessary records for verification are saved in DNS TXT records. The DKIM header in the mail provides a selector and a Signing Domain Identifier (SDID) besides the hashes of the signatures and other fields. The receiving mail server can obtain the corresponding public keys for verification with the selector and the signing domain from the DNS TXT record. An example of a DKIM signed email, and their matching records is provided in Appendix D.

DMARC

The Domain-based Message Authentication, Reporting & Conformance (DMARC) proposes an email authentication protocol. It is based on SPF and DKIM. DMARC is not a standard. Instead, it has the status "informational" and is presented RFC 7489 [34]. The DMARC policy defines how the receiving mail servers should treat SPF and DKIM. Additionally, it is possible to enable a reporting mechanism to receive email failure and statistic reports for the email domain.

2.4 Automatic Certificate Management Environment

The Automatic Certificate Management Environment (ACME) is a protocol for the automated issuing and management of X.509 certificates. It is a proposed standard and published as RFC 8555 [5]. One of the most important purposes of X.509 certificates, also known as Transport Layer Security (TLS) certificates, is to provide secure communication between clients and web servers. The main goal of this communication protocol is to provide a standardized way for the interaction between certificate authorities and users' web servers to issue TLS certificates. The data is transferred in JSON format over HTTPS to a defined Application Programming Interface (API). Since 2018 API version 2 has existed, which supersedes version 1. ACME was initially created for the free Certificate Authority (CA) Let's Encrypt [35] and is meanwhile also supported by various other providers like ZeroSSL [62] or DigiCert [15]. This work uses ACME for the automated issuing of TLS certificates from ZeroSSL and Let's Encrypt for the mail server, web server, and the Phishing tools.

2.5 IT Security Teams

IT Security is a diverse field with many specializations and jobs. For example, people build secure infrastructure or attack companies to find flaws in their security. A color scheme was introduced to allow a classification of the teams in the field. In the paper *Orange Is The New Purple* by Wright [59] the corresponding colors for the IT Security teams are presented and discussed. The three teams that were the leading target group for the survey are the following:

Red Team - Offensive Security

The Red Team represents the IT Security field with an offensive security focus. The work is based on active attacks and goals to break things or hack into companies. Example jobs in this area are Penetration Testing, Black-Box Testing, and Social Engineering. It is essential to know that the Red Team only acts on a legal basis with authorization by a corresponding entity.

Blue Team - Defensive Security

The Blue Team represents the IT Security field with a defensive security focus. Their goal is to defend and monitor IT (Security) infrastructure, work on damage control, and investigate incidents. Example jobs in this area are Incident Responder, Digital Forensics, and Security Analyst.

Purple Team - Offensive and Defensive Security

The Purple Team represents the IT Security field with a combined offensive and defensive focus. Their goal is to provide the best results from the Red and Blue Team worlds.

3 State of the Art

First, an overview of Phishing research is provided. Then the underlying software for our solution is presented. In particular, Gophish and Modlishka are the Phishing software used to create the prototype.

3.1 Phishing Research

Phishing research offers various areas for research. Possible topics for exploration are social engineering techniques, Phishing detection, and specialized Phishing techniques. Four relevant papers for this work are presented below.

Creating compelling and successful Phishing emails needs social engineering knowledge. In the research paper *An Analysis of Social Engineering Principles in Effective Phishing* Ferreira et al. [22] reviewed literature and presented elements for effective Phishing. The authors introduced five *Principles of Persuasion in Social Engineering* (PPSE) based on the analysis of previous research. After analyzing 52 Phishing emails, they evaluated the most effective PPSE elements and combinations of those. As a result, they presented a comprehensive analysis of the effectiveness of the PPSEs in Phishing emails. These results can not only be incorporated for improved offensive Phishing emails but also the improvement of awareness training material.

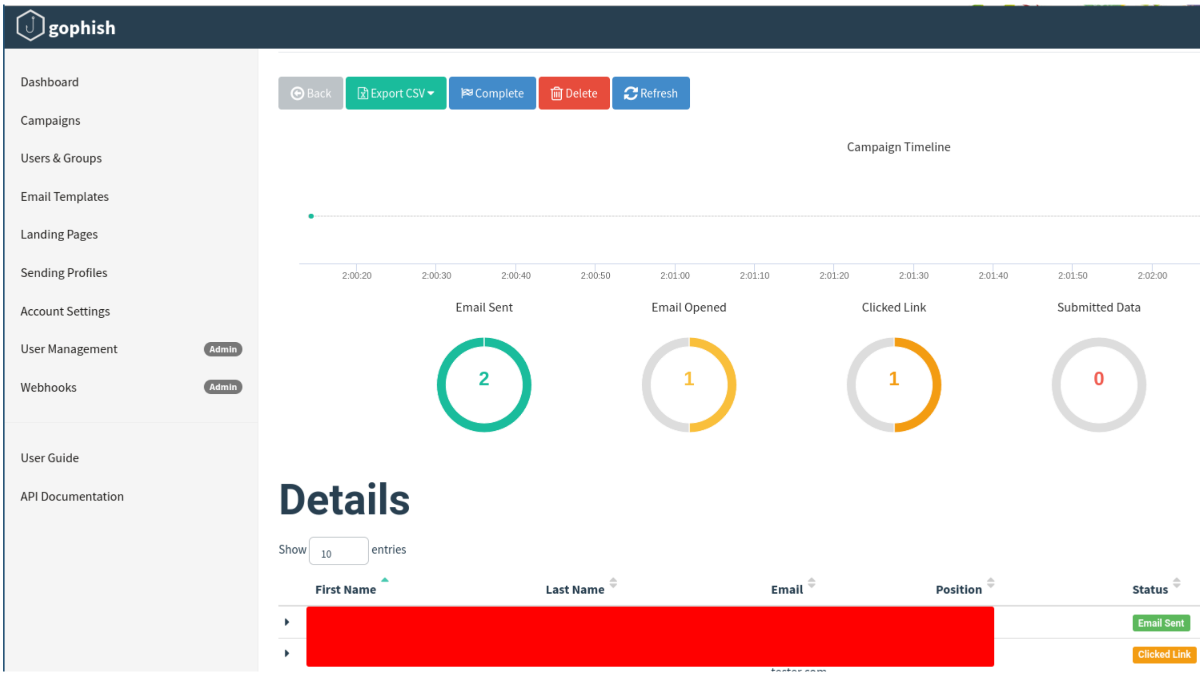
Wash [56] conducted in the paper *How Experts Detect Phishing Scam Emails* a user study with IT Security experts. The study's goal was to find out how IT Security experts identify Phishing emails. For the study, 21 IT Security experts from a single company were chosen. An interview was conducted with the experts where the information was gathered about Phishing Mail detection and the corresponding techniques used. Wash discovered a three-stage process that IT experts use to identify Phishing emails. The process consists of the stages *Sensemaking*, *decision if it is a Phishing email* and *how to deal with emails*. The results of this study and the discussed concepts, like the three-stage process, can be incorporated into future Phishing awareness training to improve employees' skills. Additionally, using the knowledge of offensive Phishing could improve the success rate of Phishing emails to achieve the objectives of the assessment.

Arshad et al. [4] conducted a systematic literature review on Phishing and Anti-Phishing techniques and discussed the most common techniques. The authors used a Systematic Literature Research (LSR) methodology for the review. According to the paper, one of the most common Phishing techniques that are discussed in the reviewed literature is Spear-Phishing, as presented in subsubsection 2.2 on page 5. Spear-Phishing can be conducted with the Phishing tool Gophish which is part of this work. Based on the author's result, Multi-Factor Authentication (MFA) is with 50% one of the most used Anti-Phishing techniques. This work provides Modlishka as a solution that can bypass some MFA methods. The bypass process is presented in subsubsection 2.2 on page 7.

The paper *Why Johnny can't rely on anti-phishing educational interventions to protect himself against contemporary phishing attacks?* by Fernando et al. [21] presents the discovery that obfuscation techniques have changed over time. The authors compared existing Anti-Phishing training material about URL obfuscation with data from real-world Phishing attacks. In the research, two new URL obfuscation techniques were identified that were not covered in the analyzed training material. Those two techniques were obfuscation via HTTPS schema and Internationalized Domain names. Incorporating the results of this research into future awareness training improves the coverage of current Phishing techniques. For offensive Phishing assessments, both new obfuscation techniques can be used with the provided prototype of this thesis.

3.2 Gophish

Gophish is an Open-Source Phishing Framework [60] which is part of the implemented prototype of this master thesis. The code is publicly available on GitHub [61], and there are over 40 contributors. As the name already suggests, the project is implemented in the programming language Go, which offers the possibility to run it on most platforms. The software provides a fully-fledged campaign planning and execution suite as depicted in Figure 3. It is possible to create a Phishing page and enables an uncomplicated way of cloning the HTML Source of target websites. Additionally, it offers sending profiles, various reporting capabilities, Webhook support, and an API. Extensive documentation is available with examples and guides to support new users. Attackers would use Gophish to manage Phishing campaigns and send out emails to victims. Additionally, they would use the page cloning feature, as presented in subsubsection 2.2 on page 6, to clone target websites and steal the valid login credentials of victims with the cloned pages.



	First Name	Last Name	Email	Position	Status
▸	[redacted]	[redacted]	[redacted]	[redacted]	Email Sent
▸	[redacted]	[redacted]	[redacted]	[redacted]	Clicked Link

Figure 3: Gophish Campaign Management Interface

3.3 Modlishka

Modlishka is a HTTP reverse proxy [18]. In contrast to the well-known reverse proxies nginx and Apache, Modlishka serves a particular use case. The unique feature of Modlishka is the transparent reverse proxy that enables bypassing 2FA security. This is done by acting as a third party between a victim and target server and executing a MITM attack. The victim will see the correct page while submitting the credentials. In the background, Modlishka will perform the authentication on behalf of the user. Therefore, all session keys and credentials are logged by Modlishka and enable a valid session through session cookies to be taken over. Attackers would use Modlishka to create proxied fake login pages to bypass 2FA security, as described in subsubsection 2.2 on page 7, and gain access to user accounts of victims.

4 Related Work

In this section the solutions are presented that were found as part of the initial thesis research. Additionally, the features of the solutions described are compared and the result is depicted in Table 1 on page 15.

4.1 Build_A_Phish

Build_A_Phish [57] implements a set of Phishing tools with Ansible and Terraform. With Teraform the infrastructure resources are configured and afterwards the Phishing software is deployed. For improved security the applications are isolated in containers and are not running directly on the host. Two cloud providers are supported and no self-hosted mail server is used. The creators of this project focused on the Operations Security (OPSEC) aspect for such engagements. Therefore, they implemented the Microsoft Azure cloud to hide the real IP address and benefit from the good internet reputation of those services.

4.2 terraform-phishing

The project *terraform-phishing* [6] uses Terraform and Ansible to roll out Gophish [60] in combination with a send-only SMTP server based on Postfix. According to the readme, the mail server is properly configured and achieves a 10/10 result on the mail-tester.com platform. The use of the cloud provider Digital Ocean is implemented and DNS records are automatically added for this provider. Additionally, some extra features like a firewall configuration and slack notifications are included.

4.3 ansible-playbook-gophish

The Ansible Playbook collection from *ansible-playbook-gophish* [47, 48] uses Ansible and Teraform for the deployment of Gophish [60]. The automated provisioning of infrastructure and DNS Records is realized with the cloud provider Digital Ocean. For email sending a Postfix email server is configured. For OPSEC purposes Cloudflare is used as the CDN provider, which hides the original IP of the Phishing server.

Comparison of Related Work solutions			
	Build_A_Phish	terraform-phishing	ansible-playbook-gophish
implemented Phishing tools	Gophish; Evilginx2	Gophish	Gophish
Simple Phishing Scenarios (Fake login page)	✓	✓	✓
Complex Phishing Scenarios (2FA Bypass)	✓	✗	✗
self-hosted mail server	✗	✗	✓
secure mail server configuration: DKIM/SPF/DMARC	✗	✓	✗
setting DNS records automated	✗	✓ - via Terraform (limited)	✓ - via Terraform (limited)
advanced OPSEC	✓ - Cloudflare DNS/Azure	✗	✓ - Cloudflare DNS
(Open-Source) License	MIT	GNU GPLv3	None defined
Resource configuration (e.g. Terraform)	✓	✓	✓
Specific cloud provider support (Terraform)	DigitalOcean	DigitalOcean; Azure	DigitalOcean; Hetzner

Table 1: Comparison of Related Work solutions

As can be seen, none of the presented solutions achieved full coverage regarding selected features.

5 Methodology

This chapter presents the objectives of this work and the corresponding methodology. In the first part, the goals of this thesis are described in detail. Subsequently, we discuss the implementation and verification of the prototype. Afterward, the approach for the survey is presented.

5.1 The Objectives

The first objective of this thesis is the creation of a requirement catalog for solutions for the automated deployment of Phishing infrastructure. We implemented a prototype to roll out the Phishing infrastructure based on that catalog. It is important to note that the solution will only deploy a ready-to-start setup. This means that no Phishing campaign texts will be pre-configured, and no suggestions are made for this procedure. The responsibility lies with the IT Security experts who use this solution. Additionally, a comparison with existing similar solutions is conducted.

The second objective is to discuss the ethical acceptability of open-sourcing such an implementation to the public. Therefore, personal opinions from the information security community for this research question are collected through a survey. Analyzing the input from IT Security professionals may offer insight if opinions differ based on the professional work experience of the survey participants. Additionally, it provides us with valuable information for the decision on releasing the project to the public. During the survey, data will also be gathered for the first objective regarding the evaluation of such solutions.

5.2 Prototype Implementation

This part of the thesis is based on the Technocratic paradigm by Eden [19]. It uses a practical methodological approach for the development of a prototype to satisfy research question RQ 2. We conducted a feasibility study by creating a requirements catalog with mandatory features and additional metrics. Based on this catalog, a prototype was implemented. We verified the reliability of the resulting prototype by a set of testing suites and conducted a comparison in reference to question RQ 3.

5.2.1 Verification of the Implementation

We define a successful implementation of the mail server setup as successfully passing configuration tests on multiple online mail server test suites.

We chose the following free services as suitable test suites:

1. mail-tester.com[1]
   A comprehensive test suite for sent emails, including SpamAssassin checks
2. MX Toolbox[2]
   A test service with various tools to analyze the mail server configuration and reputation (MX records, SPF, DMARC, Blocklists)
3. DMARCLY - DKIM record checker[3]
   A platform for DKIM record verification
4. ✓TLS[4]
   Testing the TLS configuration is offered by CheckTLS

Additionally, we compared two less secure mail server setups against our implementation to showcase the benefit of a secure mail server configuration. Unfortunately, it was not possible for us to test our prototype in a real Phishing engagements due to ethical and legal reasons.

5.2.2 Comparison of the Implementation

Based on the related work in chapter 4, three similar projects were compared against our solution. We created a comparison table with the designed requirements for automated deployment solutions.

5.3 Survey Methodology

We created a survey to discuss research question RQ5, with the sub-questions RQ 5.1 and RQ 5.2. Additionally, the evaluation question RQ 4 was included in the questionnaire. The goal was to design a short, primarily multiple-choice based, survey to gather input from security professionals. We asked three questions regarding ethical opinions, evaluation interest, and consideration about open-sourcing of automation scripts for Phishing infrastructure deployment. For the survey, we chose the constructivism paradigm approach after Guba & Lincoln [26]. We believe that ethical considerations must be explored and decided to use a survey as

[1] https://mail-tester.com
[2] https://mxtoolbox.com
[3] https://dmarcly.com/tools/dkim-record-checker
[4] https://www.checktls.com

a helpful method. The decision for the survey was based on the limited time frame for the collection and the reachability of IT Security experts via the internet. The answers were used as a basis for our critical discussion on ethical considerations.

IT Security experts were the target audience for our survey without a limitation on working experience in the area of IT Security.
We chose multiple platforms to distribute the questionnaire to reach as many IT Security as possible in approximately one month. Besides sending the survey to personal contacts who work in the IT-Security industry, one mailing list[5] was also chosen as a place for distribution. As another distribution platform, LinkedIn was chosen due to the wide range of IT Security professionals using the platform for job opportunities and knowledge exchange. Additionally, Twitter was chosen as a medium to ask for survey participation. Twitter offers the possibility to reach a broad range of IT-Security professionals, provided that an initial peer network exists.

Calculating a population size that represents an accurate number of IT Security professionals is not trivial because there are many specializations and work areas. Instead, we decided to reach as many IT Security professionals as possible. The answers were evaluated by a qualitative approach, as discussed below. We were interested to conduct a qualitative survey to collect and analyze the personal opinions of IT Security professionals. Therefore, we created qualitative questions to obtain the sentiment of the participants. A quantitative survey approach was out of scope for this work because population size is hard to determine for IT Security professionals in general.

5.3.1 Evaluation of the survey

We evaluated the answers to the survey using a mixed-method. All answers to the mandatory questions were implemented as predefined multiple-choice options. Those answers were subject to quantitative analysis. The optional free-text responses for the opinions on the ethical justifiability of releasing deployment scripts for Phishing infrastructure to the public were analyzed using qualitative content analysis. For this analysis, we used the open-access web application QCAmap [20, 40]. We grouped the free-text answers into categories by an Inductive category formation approach [29] for a thematic analysis of pro and contra arguments. We modified the process by skipping the Intra-/inter-coder agreement check because an independent review was not possible. The annotation was based on the full free-text answer. We defined a category as the core statement of an opinion. The category was created by extracting the core statement from an answer and creating a category based on the descriptive meaning of this statement. Using this process has the consequence that if an opinion contains multiple categories, these will be omitted. We aim to reach a high abstraction level with a low interpretation degree for the following category analysis with this approach.

[5] Discuss list by CERT.at

6 Requirement Catalog

Creating a suitable requirement catalog was the prerequisite for the technical implementation of the prototype. This catalog also enables a comparison with similar solutions and the resulting prototype.

6.1 Main features

We define the following four main features as requirements for an automated Phishing deployment solution:

1. Phishing tooling to cover simple (fake login page) and complex (2FA Bypass) scenarios
2. Self-hosted mail server for data sovereignty
3. Secure mail server configuration for trustworthy emails
4. Possibility of automatically configuring DNS records

6.2 Catalog metrics

Designing the requirement catalog for Phishing infrastructure deployment solutions and the comparison with existing solutions must include additional measurable metrics besides those mentioned above. Therefore, the following metrics were chosen as suitable ones for the requirement catalog:

implemented Phishing tools
Which Phishing software is implemented with the solution? Comparing the implemented Phishing tools may help the readers choose the best solution for them.

Simple Scenarios possible (simple fake login Page)
Does the solution offer the possibility to create a simple Phishing scenario? A simple scenario corresponds to a simple page cloning feature for creating fake login pages.

Complex Scenarios possible (2FA Bypass)
Does this solution offer the possibility to create a complex Phishing scenario? A complex scenario corresponds to a more advanced feature that uses a modified reverse proxy to defeat 2FA Security.

Self-hosted mail server setup
Does the solution offer a self-hosted mail server setup? A self-hosted service corresponds to running infrastructure under one's own control and not depending on external mail services for sending mails.

secure mailserver setup (DKIM, SPF, DMARC)
Is the mail server configured with additional security measures like DKIM, SPF, and DMARC? Setting up a secure mail server with security measures like DKIM, SPF and DMARC improves the trustworthiness of sent mails.

automated deployment of DNS records
Does the solution offer the feature to deploy DNS records automatically? Deployment may be through custom scripts in combination with the usage of provider APIs.

advanced OPSEC
Does the solution offer advanced OPSEC capabilities, for example, using CDN providers to hide IP addresses or web services with good reputation.

(Open-Source) License
May it be used for commercial use, or is it restricted to specific use cases? Depending on which kind of (Open-Source) license the solution offers, it may or may not be used by third parties.

resource configuration (Terraform)
Is it supported to create server resources programmatically? Today, various cloud providers offer virtual resources for rent. Using IAC software like Terraform, for example, offers the possibility to combine software deployment via Ansible on pre-configured computing resources.

Specific Cloud Provider support (Terraform)
If the solution supports resource configuration, which cloud providers are supported? When supporting resource configuration, support for cloud providers still needs to be pre-configured before it can be used.

7 Prototype - Design

The prototype was implemented based on the defined set of requirements from the catalog that was created. It was essential to design and build the prototype upon open-source software because there are considerations with regard to releasing our work to the public in the future.

In this section, the chosen stack is presented. Ansible is the main component for the deployment. Therefore, the selection criteria for this configuration software choice are discussed below.

7.1 Ansible

Ansible was selected as the suitable solution for the automated deployment. The decision for this automation tool is based on the following reasons:

Prerequisites

Ansible requires only two programs on the target server as prerequisites: SSH and Python. Those two programs are, by default, available in most of the modern Linux distributions or are at least installable via the packet manager. Besides SSH and Python, only Ansible itself is needed on the executing client.

Agentless

Software like Chef [9], or Puppet [51] depend on a running software on the target host. Ansible, instead, does not rely on a daemon or remotely running agent software. That reduces thereby the attack surface of the implementation. The user chooses when the deployment should happen, and no additional components are needed except those mentioned in the section Prerequisites above.

Community and Stability

Ansible is backed by the well-known and established Linux business Red Hat thus IBM. Based on this, the project has a stable future. Additionally, an active community is working on it, and a broad range of public GitHub repositories with public Playbooks and Roles exist. With Ansible Galaxy [2] a community hub exists, where the community can share and find roles and collections to use in their projects.

Broad Availability on Linux Operating Systems

Due to the fact that Red Hat maintains Ansible, it is broadly available on all major Linux distributions. Since Microsoft introduced the WSL Windows Subsystem for Linux [58] on Windows 10 [30], it is now also possible to use Ansible with the Windows Operating System (OS).

Alternatives

Apart from Ansible, other alternatives exist. They offer the same functionality but differ in features or the programming language. Possible alternatives are Saltstack, Puppet, and self-made bash scripts.

7.2 Mail server

As Mail Transfer Agent (MTA), Postfix was chosen for self-hosted mail sending due to the maturity of the project and broad usage of around 30% in the IT world [38]. The secure mail configurations of DKIM, SPF, and DMARC are discussed in the following sections, including the configured DNS TXT records.

DKIM

For the DKIM implementation we chose OpenDKIM due to the interoperability with Postfix. The used hash size is SHA256 and for signing a 2048 bit RSA key is created.

DNS TXT record value:
`v=DKIM1; h=sha256; k=rsa; s=email; p=<PUBLIC KEY>`

SPF

The SPF configuration defines only our configured server is allowed to send mails. All other servers are not allowed and are excluded with the `-all` keyword.

DNS TXT record value:
`v=spf1 mx ip4:<IPv4 Address> ip6:<IP Address> a:<A record of mail server> -all`

DMARC

For DMARC, we chose only to implement a record with the value `None` which effectively does not affect the delivery method. This was decided as we want to deliver the Phishing emails - even if some of the checks fail. It may be useful in the future to define stricter policies.

We are not interested in receiving email reporting by other mail servers, so no reporting addresses are provided. There is no use for reporting emails, as Phishing is executed usually in a send and forget style. Still, the SPF and DKIM identifier alignment is already defined strictly in the record and can be activated by defining a stricter policy.

DNS TXT record value:
`v=DMARC1; p=none; adkim=s; aspf=s;`

7.3 Web server

For web request handling, nginx was chosen as a suitable open-source web server. It offers a modular configuration, mature documentation, and various available guides on the internet.

7.4 Phishing Software

The objective was to deploy ready-to-start Phishing software. Therefore, we chose the open-source Phishing programs Gophish and Modlishka as suitable solutions because together they cover the simple and complex use-cases for Phishing assessments. Gophish provides campaign management and offers a simple site cloning feature. Modlishka's reverse server capabilities cover the complex use-case when logins with mandatory 2FA need to be bypassed. Both tools are written in Go and do not require, by default, additional software like databases. Additionally, both binary releases are available for download.

7.5 Domain/DNS

A Phishing (test) infrastructure and a corresponding mail server need to be available on the internet. They need a domain tied to the server IP address for proper implementation. Therefore, we used our domain `br0ken.cloud` for the implementation and configured the subdomains `leakybuffer.on.br0ken.cloud` for the mail server and `phishingparty.on.br0ken.cloud` for the Phishing front end. Due to the fact that the domains are registered at the domain provider Porkbun, we chose to implement their API for the automated configuration of DNS records. Porkbun was selected as a domain provider because they offer low prices and a comprehensively documented API for domain configuration.
A template role was created for easy extensibility, which offers a documented way to add new DNS providers that offer a REST API without changing much code in the implemented roles.

The prototype automatically deploys A, MX, SPF, DKIM, and DMARC DNS records if an available DNS API is used. Optionally, AAAA records will be set if the server has a public IPv6 address. The user only needs to set the reverse DNS (PTR record) manually. This is the case because PTR record configuration differs between cloud providers, and we did not implement specific support for a specific provider. Otherwise, all records need to be set manually by the user. A helpful information file is generated by our prototype, where all records that are needed are documented.

7.6 Deployment Infrastructure

To deploy the tools an infrastructure and an operating system is needed. This section presents the chosen solutions for the implementation and deployment of our prototype.

Operating System

As operating system Debian 11, codename "bullseye" was chosen due to the maturity of the operating system and its broad use as a server operating system. Debian is by default available at most cloud service providers for virtual machines.

Local Virtual Machine

For the initial implementation, a local Virtual Machine (VM) based on QEMU was used. All components which did not need internet and DNS records were implemented in combination with this local VM. This enabled the initial implementation of most of the Ansible roles locally before deploying it onto an internet facing system.

Hetzner Cloud

After implementing and testing the roles on the local VM, a Hetzner Cloud VM with a public IPv4 address was rented. The fact that Hetzner offers a very cheap renting model for shared servers with hourly billing was one reason for the choice as a cloud provider. The location in the European Union was also a must-have requirement due to GDPR and the support for European companies. Additionally, Hetzner offers a snapshot feature. Therefore, fast and efficient snapshots could be created, and restoring old status was very uncomplicated.
The Hetzner Cloud Resources are rented for a temporary time and the server and snapshots will be destroyed after the conclusion of the master thesis.

8 Prototype - Results

The prototype implementation based on Ansible deploys a fully functional Phishing infrastructure setup on a server. The Big Picture with the implemented software and their connections is presented in Figure 4. Documentation about the implemented Playbooks and Roles can be found in section 8.1. How the infrastructure can be deployed using Ansible is demonstrated in section 8.2.

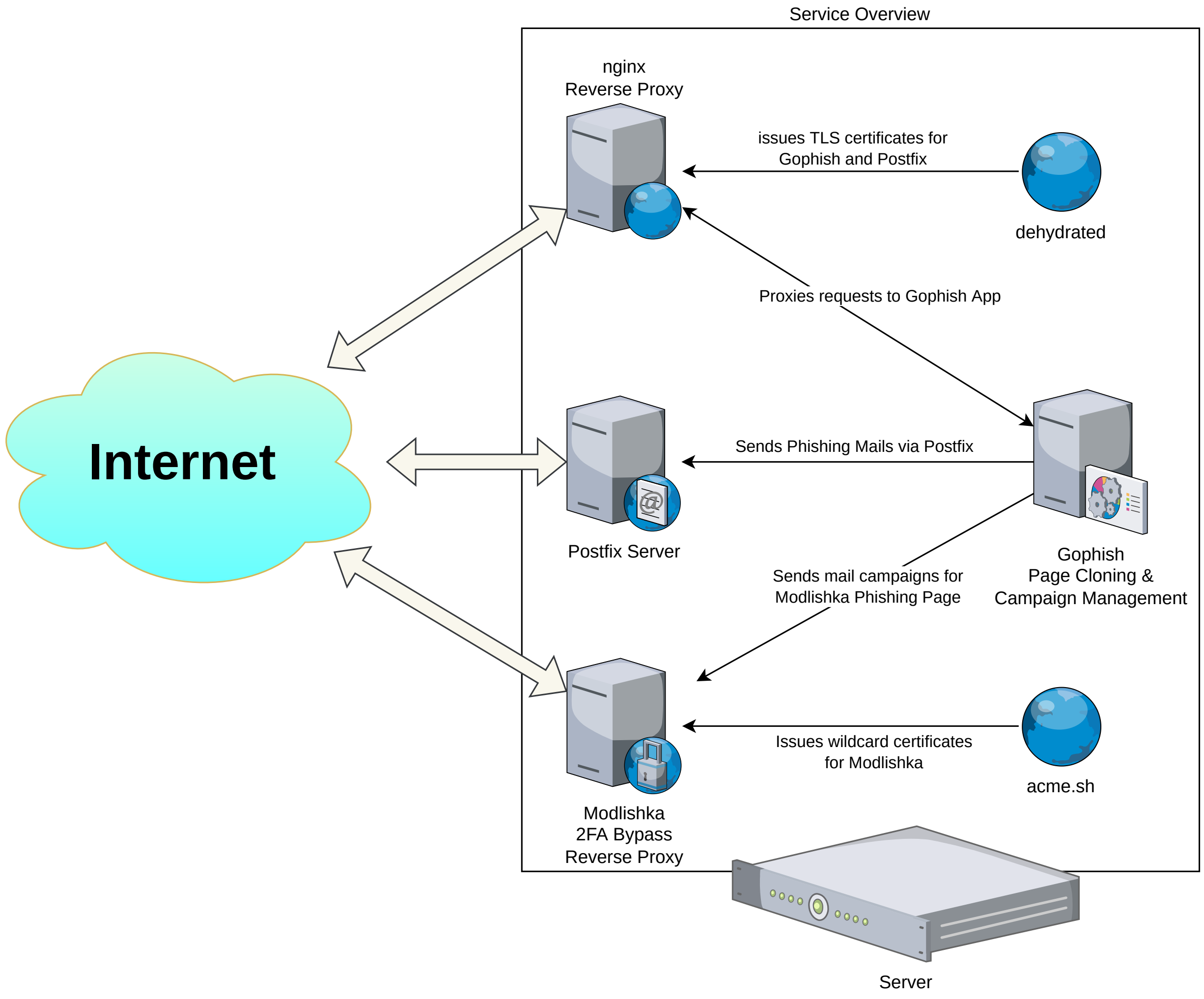


Figure 4: Big Picture of the implemented setup

8.1 Ansible Implementation

Ansible consists of Playbooks that utilize Roles. Roles are implemented code that execute defined steps of commands and the set of commands are usually grouped to deploy a specific part of a modular setup. For example, the complete mail server setup is implemented in one role and the web server setup in another. To ensure good code quality, we linted our code with ansible-lint and used Fully Qualified Collection Name (FQCN) for all functions used in the roles. Additionally, we followed most suggestions from the Ansible 101 Standards Blog post [55], which provides an unofficial list of best practices regarding writing Ansible code.
In the current state, the implementation is designed for a single server that hosts the mail server and the Phishing tools. The Ansible implementation consists of the following Playbooks and Roles:

8.1.1 Playbooks

Two Playbooks were implemented: One Playbook is used for the web server and mail infrastructure deployment and the other one deploys the Phishing infrastructure.

mailsetup.yml

The mail configuration is defined in this Playbook. It deploys a configured postfix mail-server, creates TLS certificates with Let's Encrypt, and sets the DNS Records automatically for the available providers.

This Playbook depends on the following roles:

- DNS/API
- DNS/server_records
- global_handlers
- mailserver/postfix
- nginx/install
- nginx/dehydrated_domain
- system-base

The configuration flow of the mail setup Playbook is presented in Figure 5.

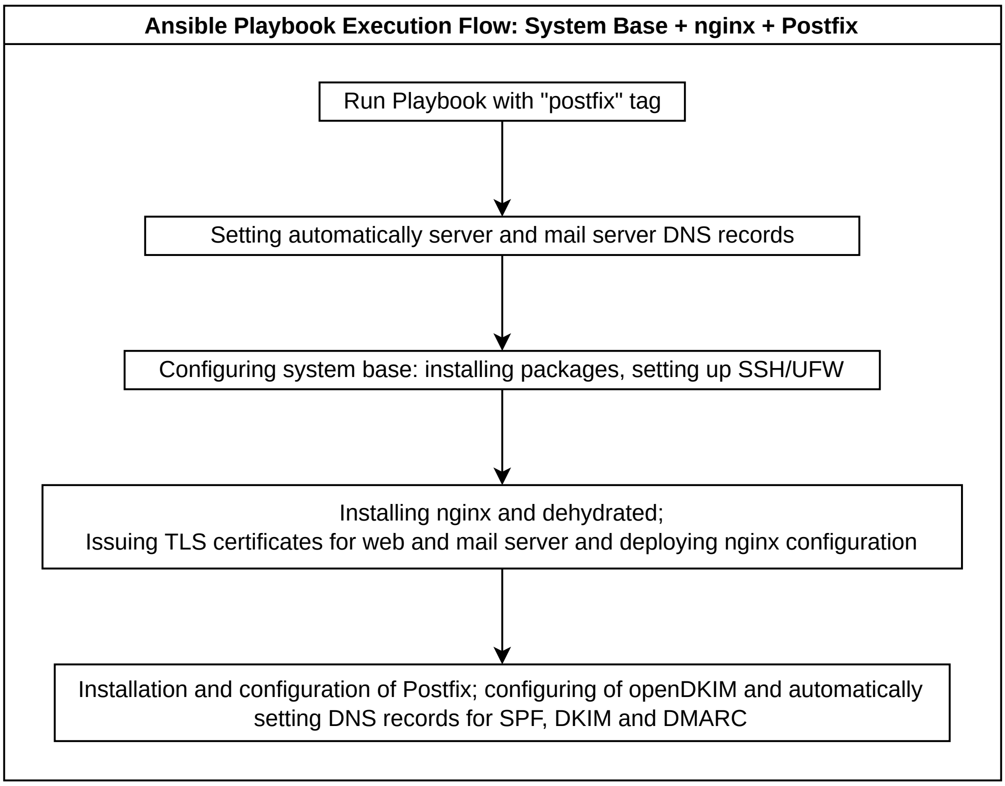

Ansible Playbook Execution Flow: System Base + nginx + Postfix
Run Playbook with "postfix" tag
Setting automatically server and mail server DNS records
Configuring system base: installing packages, setting up SSH/UFW
Installing nginx and dehydrated;
Issuing TLS certificates for web and mail server and deploying nginx configuration
Installation and configuration of Postfix; configuring of openDKIM and automatically setting DNS records for SPF, DKIM and DMARC

Figure 5: Mail Server Setup - Ansible Playbook Flow

phishsetup.yml

The deployment for Gophish and Modlishka is implemented in the phishsetup.yml Playbook. The Gophish Play deploys a Gophish instance and preconfigures the local mail server. Additionally, a random or predefined admin password is automatically set. The Modlishka Play downloads and installs the Modlishka reverse proxy. Due to the need for a wildcard certificate, the acme.sh script is installed and responsible for creating a wildcard certificate for the configured Phishing domain. The two available Phishing templates for Office 365 and Google are installed with correct certificates. The templates do not work and are available as configuration examples for learning purposes.

This Playbook depends on the following roles:

- DNS/API
- DNS/acme.sh
- global_handlers
- phishing-tools/gophish
- phishing-tools/modlishka

The configuration flow of the Phishing tool Playbook is depicted in Figure 6.

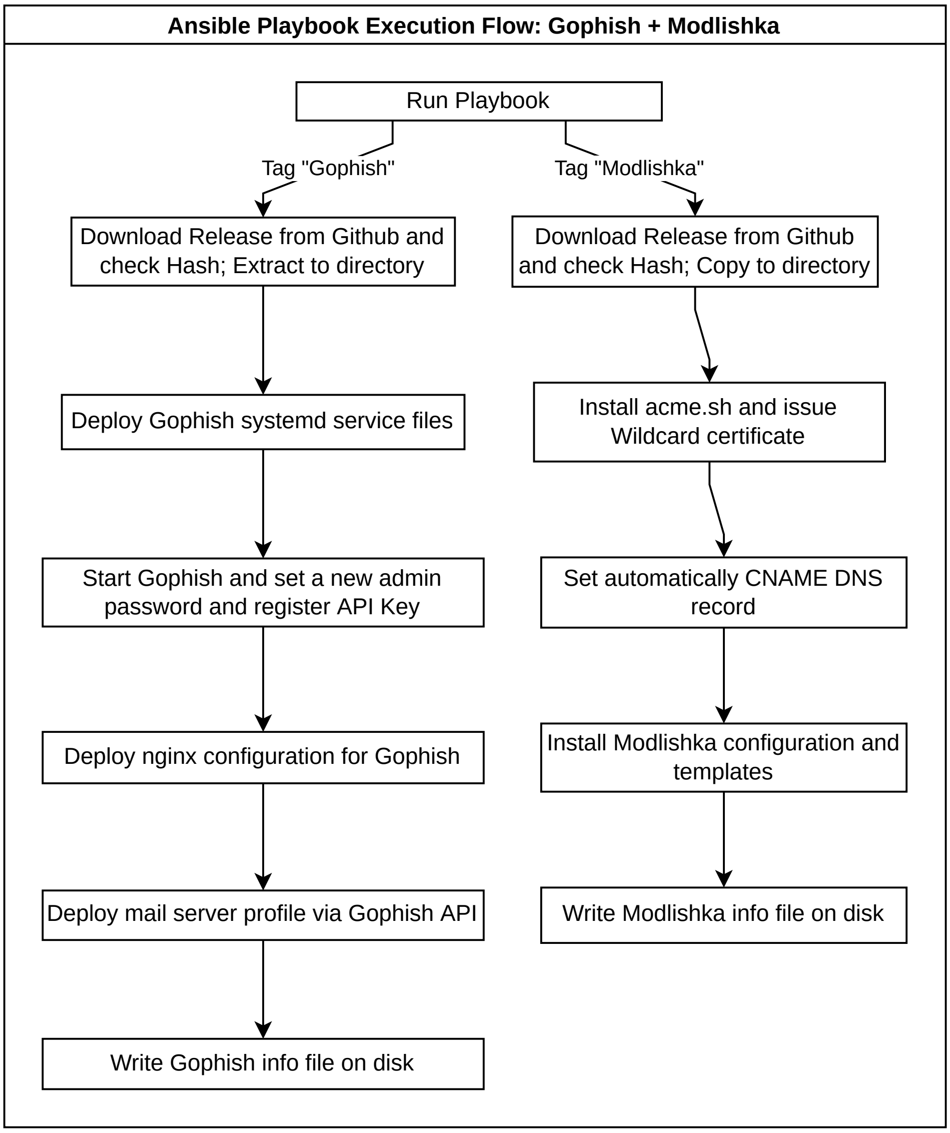

Ansible Playbook Execution Flow: Gophish + Modlishka
Run Playbook
Tag "Gophish"
Tag "Modlishka"
Download Release from Github and check Hash; Extract to directory
Deploy Gophish systemd service files
Start Gophish and set a new admin password and register API Key
Deploy nginx configuration for Gophish
Deploy mail server profile via Gophish API
Write Gophish info file on disk
Download Release from Github and check Hash; Copy to directory
Install acme.sh and issue Wildcard certificate
Set automatically CNAME DNS record
Install Modlishka configuration and templates
Write Modlishka info file on disk

Figure 6: Gophish and Modlishka Setup - Ansible Playbook Flow

8.1.2 Roles

Various roles were implemented that perform specific tasks for the prototype deployment. In the following every role with their corresponding task is presented.

DNS/acme.sh

The acme.sh script provides a full ACME protocol implementation written in pure Shell script. The role installs acme.sh on the system. It is used for the issuing of the wildcard certificates for Modlishka.

DNS/API/Porkbun

For the automated configuration of the necessary DNS records the API [3] of the domain registrar Porkbun was implemented in this role. The Example role is derived from this role.

DNS/API/EXAMPLE

This role is a commented version of the Porkbun DNS API role. It serves as a template role for the implementation of new API providers.

DNS/server_records

The server_records role configures all server DNS records that are needed. These records are A and AAAA for the mail server and phishing server domain. Additionally, a reminder is provided to set the reverse DNS entry for the server IP manually.

global_handlers

Instead of defining separate handlers in their corresponding roles, it is a suitable option to make them globally available in the script. Unfortunately, there is no standardized way to implement global handlers in Ansible. We chose to define a task including all handlers and import it as a handler in all Playbooks. This offers a central place to manage all handlers that are used instead of maintaining them across multiple files.

mailserver/postfix

This role deploys the postfix mail server and configures SPF, DKIM, and DMARC based on the SimpoLab Mailserver Guide [39]. For the DKIM implementation, the software OpenDKIM is used.
After successful deployment, the file `Mailserver_and_DNS_info.txt` is created on the local host that contains an overview of all DNS records that are needed for the mail server installation. If a DNS API is provided, those records should already be automatically set during the deployment. In this case, only the PTR record needs manual configuration.

nginx

This role installs the nginx web server and dehydrated [14]. nginx is a crucial component for Gophish and the mail and web TLS certificates. Dehydrated is a letsencrypt/acme-client implementation entirely written in bash. It generated in combination with nginx the corresponding server certificates for Postfix and Gophish. Nginx proxies all requests to the Gophish Phishing front end.

phishing-tools/gophish

The Gophish roles download the latest Gophish release binary from Github. Currently, Version *v0.11.0* is implemented, and integrity checks are executed by verification of the SHA256 checksum. A systemd service file is installed to start Gophish on boot. The script also extracts the initial password from the log and registers a new random password for the admin user. Additionally, the API key is extracted, and the mail server configuration is configured via API.
After successful deployment, the file `Gophish_info.txt` is created on the local host that contains credentials for the Gophish installation and further information and links for the usage of the tool.

phishing-tools/modlishka

The latest Modlishka release is downloaded from Github. An integrity check with a SHA256 sum for version *v 1.1.0* is conducted and, after successful checking, saved to disk.¸ During the setup, the acme.sh script is installed, and wildcard certificates are issued. The available Google and Office 365 templates are preconfigured with the wildcard TLS certificates that have been issued. Because the templates are non-functional in the provided state, the prepared configuration files serve as learning examples for new configuration files.
After successful deployment, the file `Modlishka_info.txt` is created on the local host that contains further information and helpful links for the usage of the tool.

system-base

This role reconfigures the server and installs basic software packages. During the reconfiguration, a SSH port is set, SSH password authentication is disabled, and the Uncomplicated Firewall (`ufw`) is enabled. The programs that are additionally installed are provided for system administration (e.g. `tmux, sudo, htop`) and debugging (e.g. `net-tools, tcpdump, strace`).

Additional Script - Gophish Mail Config

During the implementation of Gophish, the API needed to be used for the configuration of the mail server. To obtain the API key for the API usage in a fresh installation, it is required to log in with initial credentials and set a new password. Therefore, the Python script `gophish_login_config.py` was created that automatically assigns a new password and returns the API key. The returned key was used for the next Ansible task to deploy the mail configuration via the REST interface.

8.2 Deploying the Setup

To deploy the setup, specific prerequisites must be satisfied, and the configuration files must be configured accordingly.

Prerequisites
The following prerequisites need to be satisfied before deploying:

Server with Debian 11
The implementation is tailored for Debian 11 but should also work for Debian derivatives like Ubuntu.

Port 25 & 465 open
Cloud Providers often block Port 25 and 465 to prevent email scammers and spammers from abusing their service. Those ports are needed for email sending and must be enabled. This is typically configured on an external Firewall and is out of scope for the prototype.

(Public) IPv4 and/or IPv6 Address
The Phishing server should be publicly available over IPv4 or IPv6. The Phishing mail receivers should be able to reach the Phishing server.

Domain(s)
As minimum one Domain should be available to be configured for the Phishing and mail server setup.

Email Address
A email address is needed for the account registration at TLS certificate issuing organizations (Let's Encrypt, ZeroSSL)

Signed Rules of Engagement
When conducting Phishing assessments against company employees, written permission (Rules of Engagement) needs to be signed by the hiring company or the internal principal.

How to deploy

First the target server in the `inventory.ini` should be defined. It is recommended to configure the SSH server in the local SSH config under ~/`.ssh/config` and add it to the *Deploy_Server* group in the Ansible inventory.

Copy or rename the example variable configuration `all.yml.example` under `./group_vars` to `all.yml` and fill out the config according to the variables. Comments are included in the configuration file and should be a guide through the process.

After the variable configuration, the Playbooks can be executed. The implementation consists of two separate Playbooks. Executing the Playbooks without specifying a tag will print out a help text with available options.

The following Ansible Playbook commands will deploy the setups:

- Base system with nginx and postfix:
  `ansible-playbook mailsetup.yml -t postfix`
- Gophish:
  `ansible-playbook phishsetup.yml -t gophish`
- Modlishka:
  `ansible-playbook phishsetup.yml -t modlishka`

8.3 Prototype Advantages

Deploying infrastructure with dedicated automation software offers advantages over manual deployment. In this section, two advantages of our prototype are presented and underlined with measurements.

8.3.1 Error Prevention and Setup Reproducibility

Deploying the infrastructure with software built for automated deployment leads to the prevention of configuration errors under the assumption that there are no mistakes in the deployment code. The following mistakes can be prevented by using our prototype:

- missing DNS Records
- misconfiguration of DNS Records
- misconfiguration of the postfix mail server
- errors during the OpenDKIM configuration
- missing or wrong SPF, DKIM, or DMARC records
- incorrectly issued TLS certificates
- errors in the web server configuration
- mistakes in the configuration of Gophish
- mistakes in the configuration of Modlishka

Additionally, no reproducibility and consistency are guaranteed when the deployment is done manually because human errors can introduce bugs while configuring. Ansible's automated deployment creates a consistent and reproducible Phishing infrastructure environment.

8.3.2 Deployment Speed - Measurement

We measured the time for the deployment of the infrastructure of the two Ansible Playbooks. For the test, a Hetzner CX 11 cloud instance was rented, and the execution time of the Ansible Playbooks was measured with the Linux `time` command.

Execution time of the Prototype Playbooks

The three implemented deployments for email infrastructure, Gophish, and Modlishka, were executed independently, and the needed time was measured.

Execution time for the mail setup Playbook

```
ansible-playbook mailsetup.yml -t postfix -v 14,90s user 2,74s system
11% cpu 2:31,43 total
```

The system and mail server setup deployment took 2 minutes and 32 seconds.

Execution time for the Gophish Playbook

```
ansible-playbook phishsetup.yml -t gophish -v 6,10s user 1,25s system
12% cpu 1:00,29 total
```

The deployment of Gophish took 1 minute and 1 second. Because of the unreliable CSRF implementation of Gophish, the time may vary in reality. In another test run, it only took around 26 seconds.

Execution time for the Modlishka Playbook

```
ansible-playbook phishsetup.yml -t modliskha -v 29,39s user 3,67s
system 6% cpu 8:16,69 total
```

The deployment of Modlishka took 8 minutes and 17 seconds. According to the Ansible output, the longest time took the certificate issuing with acme.sh with around 7 minutes and 37 seconds.

Summing up the runtime of all executed Playbooks, we get 11 minutes and 50 seconds. Adding one more minute for manually setting the reverse DNS pointer in the settings of the cloud provider results in a total time of 12 minutes and 50 seconds.

Estimation about manual deployment

Based on our professional experience, we estimate the following times needed for the following tasks if the infrastructure deployment is done manually:

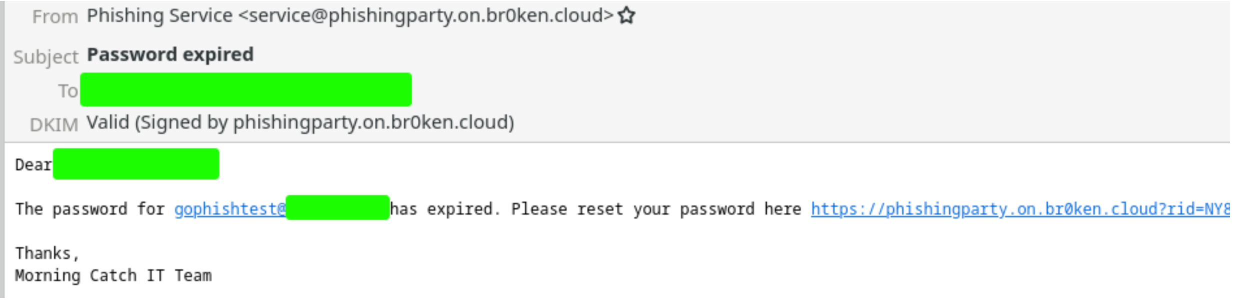
- Configuring the initially needed DNS records (A, AAAA, PTR): 5 minutes
- Installing system packages and configuring the reverse proxy: 30 minutes
- Installing and configuring mail server (postfix) including OpenDKIM: 180 minutes
- Creating the correct SPF, DKIM, and DMARC records and setting them: 35 minutes
- Installing and configuring Gophish: 60 minutes
- Installing and configuring Modlishka and acme.sh: 90 minutes

Summing up the estimated manual deployment times results in 6 hours and 40 minutes.

The significant time saving of over 6 hours combined with the discussed error prevention proves that an automated deployment script is superior to manual deployment.

8.4 Implementation Verification

To verify our implementation, we deployed the setup using our Ansible Playbooks. An obviously fake Phishing Campaign was sent via Gophish to mail-tester.com to verify the mail server configuration to simulate a realistic mail sending scenario. The mail was sent by the sender `Phishing Service <service@phishingparty.on.br0ken.cloud>` and included a fictive "Password expired" story as shown in Figure 7.



From Phishing Service <service@phishingparty.on.br0ken.cloud> ✩
Subject **Password expired**
To [redacted]
DKIM Valid (Signed by phishingparty.on.br0ken.cloud)

```
Dear [redacted]

The password for gophishtest@[redacted] has expired. Please reset your password here https://phishingparty.on.br0ken.cloud?rid=NY8

Thanks,
Morning Catch IT Team
```

Figure 7: Sent out fake Phishing email

Configuration

The following configuration was used for our test setup:

- Hetzner Cloud Server Instance CX11 (2GB RAM, 20GB SSD, 20TB Traffc - 4,19€/Month)
- Server IPv6: `2a01:4f8:c0c:589e::1`
- Server IPv4: `49.12.207.135`
- Mail server domain: `leakybuffer.br0ken.cloud`
- Phishing server domain: `phishingparty.on.br0ken.cloud`
- DKIM Selector: `dkimparty`

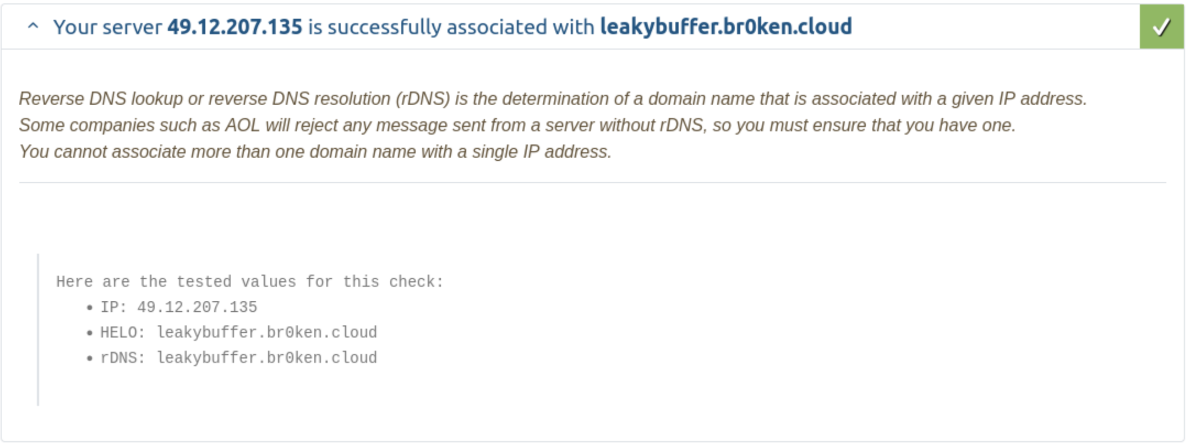
The correct (security) configuration of the deployment was regularly verified during implementation with the test suites from subsection 5.2.1. The final verification was carried out by the mail-tester.com platform.

8.4.1 DNS Records

In this section, the verification of the deployed DNS records is documented. Correctly configured DNS records are vital for the proper delivery of mails.

PTR/A/AAAA/MX

The mail-tester.com suite was chosen to proof the valid reverse DNS (PTR) record in Figure 8 and A record in Figure 9.

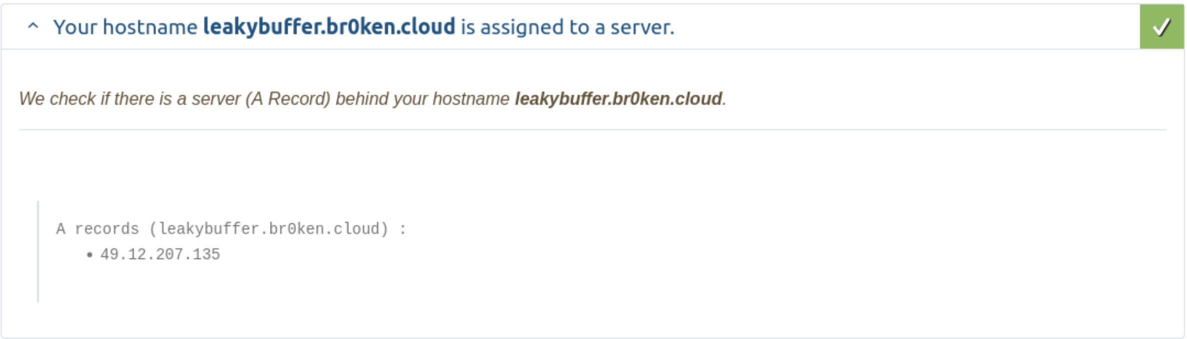


^ Your server **49.12.207.135** is successfully associated with **leakybuffer.br0ken.cloud** ✓

Reverse DNS lookup or reverse DNS resolution (rDNS) is the determination of a domain name that is associated with a given IP address.
Some companies such as AOL will reject any message sent from a server without rDNS, so you must ensure that you have one.
You cannot associate more than one domain name with a single IP address.

```
Here are the tested values for this check:
  • IP: 49.12.207.135
  • HELO: leakybuffer.br0ken.cloud
  • rDNS: leakybuffer.br0ken.cloud
```

Figure 8: Reverse DNS verification by mail-tester.com [43]



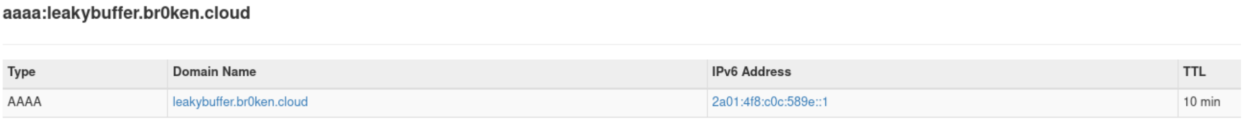
^ Your hostname **leakybuffer.br0ken.cloud** is assigned to a server. ✓

We check if there is a server (A Record) behind your hostname ***leakybuffer.br0ken.cloud****.*

```
A records (leakybuffer.br0ken.cloud) :
   • 49.12.207.135
```

Figure 9: A record verification by mail-tester.com [43]

Because mail-tester.com only uses IPv4 for receiving emails and does not check IPv6, we used MX Toolbox to look up the IPv6 record as seen in Figure 10.

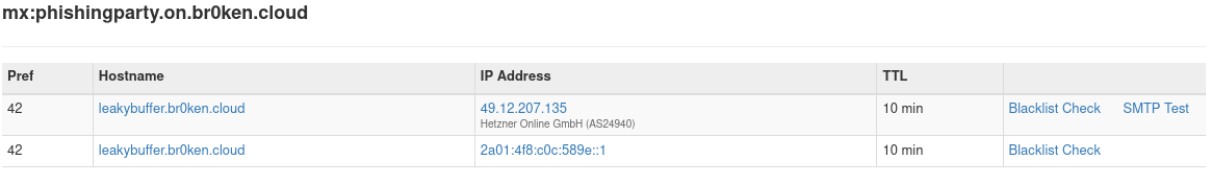


aaaa:leakybuffer.br0ken.cloud

Type	Domain Name	IPv6 Address	TTL
AAAA	leakybuffer.br0ken.cloud	2a01:4f8:c0c:589e::1	10 min

Figure 10: AAAA record verification by MX Toolbox [42]

The MX Suite was also used to test the correctness of the MX record, as shown in Figure 11.



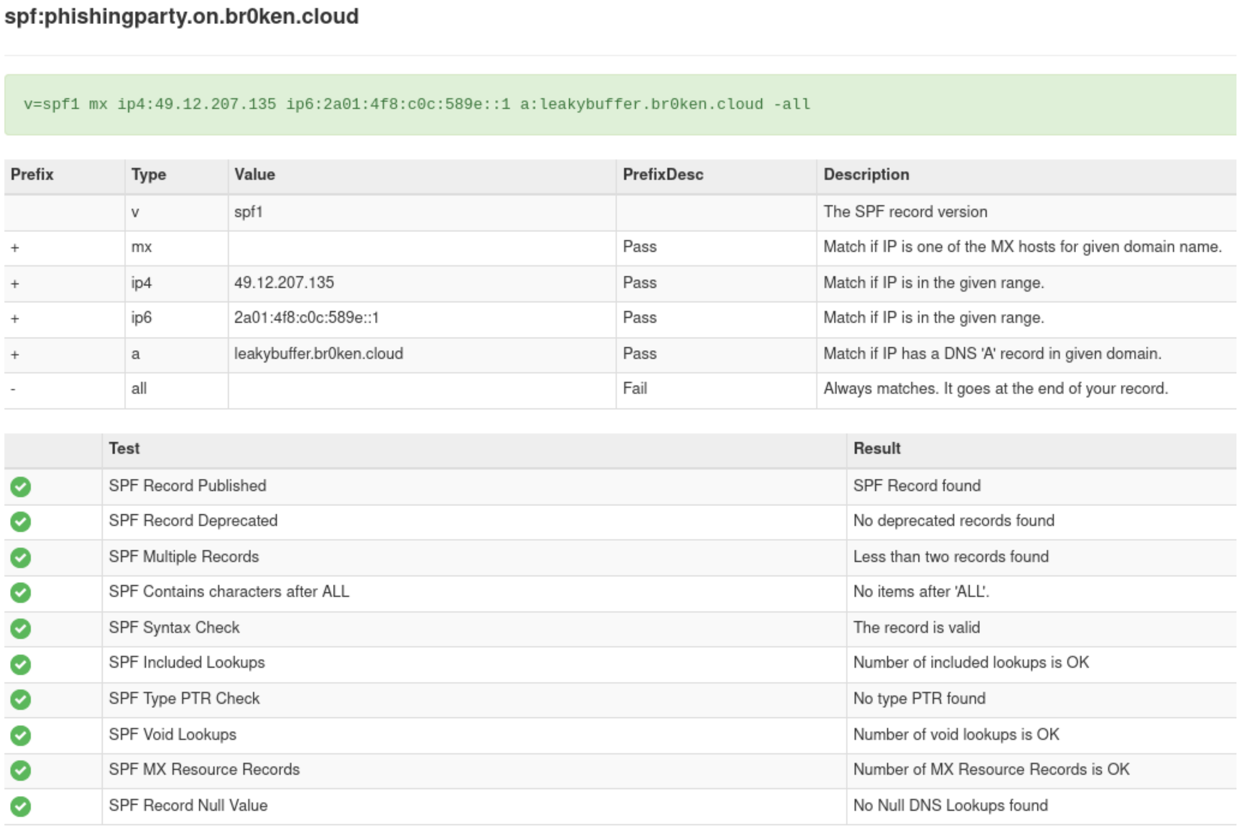
mx:phishingparty.on.br0ken.cloud

Pref	Hostname	IP Address	TTL	
42	leakybuffer.br0ken.cloud	49.12.207.135 Hetzner Online GmbH (AS24940)	10 min	Blacklist Check SMTP Test
42	leakybuffer.br0ken.cloud	2a01:4f8:c0c:589e::1	10 min	Blacklist Check

Figure 11: MX record verification by MX Toolbox [42]

SPF

The configured SPF record is also valid as depicted in Figure 12.



spf:phishingparty.on.br0ken.cloud

```
v=spf1 mx ip4:49.12.207.135 ip6:2a01:4f8:c0c:589e::1 a:leakybuffer.br0ken.cloud -all
```

Prefix	Type	Value	PrefixDesc	Description
	v	spf1		The SPF record version
+	mx		Pass	Match if IP is one of the MX hosts for given domain name.
+	ip4	49.12.207.135	Pass	Match if IP is in the given range.
+	ip6	2a01:4f8:c0c:589e::1	Pass	Match if IP is in the given range.
+	a	leakybuffer.br0ken.cloud	Pass	Match if IP has a DNS 'A' record in given domain.
-	all		Fail	Always matches. It goes at the end of your record.

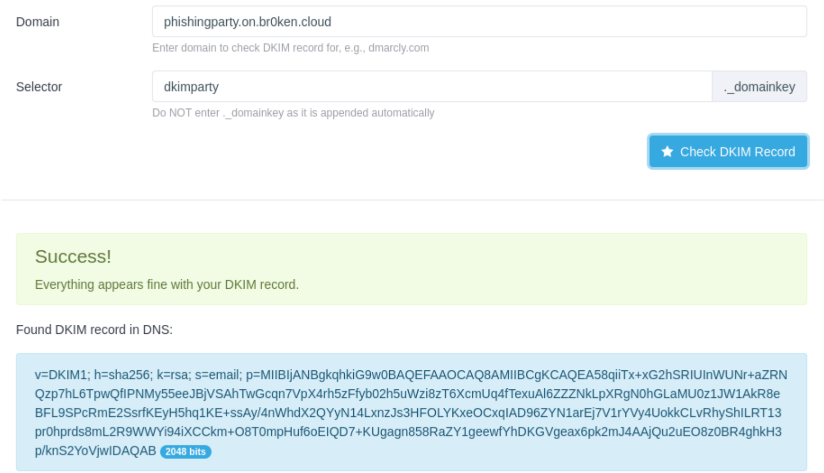
	Test	Result
✓	SPF Record Published	SPF Record found
✓	SPF Record Deprecated	No deprecated records found
✓	SPF Multiple Records	Less than two records found
✓	SPF Contains characters after ALL	No items after 'ALL'.
✓	SPF Syntax Check	The record is valid
✓	SPF Included Lookups	Number of included lookups is OK
✓	SPF Type PTR Check	No type PTR found
✓	SPF Void Lookups	Number of void lookups is OK
✓	SPF MX Resource Records	Number of MX Resource Records is OK
✓	SPF Record Null Value	No Null DNS Lookups found

Figure 12: SPF record verification by MX Toolbox [42]

DKIM

For DKIM, a TXT record needs to be deployed for the host following host: `<DKIM Selector>._domainkey.<Mail Sender Domain>` (The values in angle brackets are placeholder.)

The result in Figure 13 of the DMCARLY record checker certifies that we deployed a correct DKIM TXT record:



Domain: phishingparty.on.br0ken.cloud

Enter domain to check DKIM record for, e.g., dmarcly.com

Selector: dkimparty ._domainkey

Do NOT enter ._domainkey as it is appended automatically

Success!

Everything appears fine with your DKIM record.

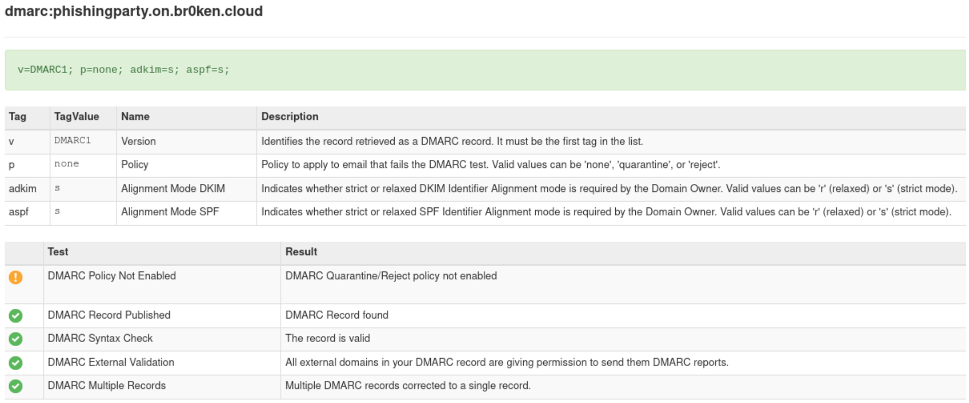
Found DKIM record in DNS:

```
v=DKIM1; h=sha256; k=rsa; s=email; p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA58qiiTx+xG2hSRIUInWUNr+aZRN
Qzp7hL6TpwQfIPNMy55eeJBjVSAhTwGcqn7VpX4rh5zFfyb02h5uWzi8zT6XcmUq4fTexuAl6ZZZNkLpXRgN0hGLaMU0z1JW1AkR8e
BFL9SPcRmE2SsrfKEyH5hq1KE+ssAy/4nWhdX2QYyN14LxnzJs3HFOLYKxeOCxqIAD96ZYN1arEj7V1rYVy4UokkCLvRhyShILRT13
pr0hprds8mL2R9WWYi94iXCCkm+O8T0mpHuf6oEIQD7+KUgagn858RaZY1geewfYhDKGVgeax6pk2mJ4AAjQu2uEO8z0BR4ghkH3
p/knS2YoVjwIDAQAB 2048 bits
```

Figure 13: DKIM record verification by DMARCLY [16]

DMARC

The DMARC record is correctly set in Figure 14 but no policy is enabled as discussed in section 7.2 on page 22.



dmarc:phishingparty.on.br0ken.cloud

```
v=DMARC1; p=none; adkim=s; aspf=s;
```

Tag	TagValue	Name	Description
v	DMARC1	Version	Identifies the record retrieved as a DMARC record. It must be the first tag in the list.
p	none	Policy	Policy to apply to email that fails the DMARC test. Valid values can be 'none', 'quarantine', or 'reject'.
adkim	s	Alignment Mode DKIM	Indicates whether strict or relaxed DKIM Identifier Alignment mode is required by the Domain Owner. Valid values can be 'r' (relaxed) or 's' (strict mode).
aspf	s	Alignment Mode SPF	Indicates whether strict or relaxed SPF Identifier Alignment mode is required by the Domain Owner. Valid values can be 'r' (relaxed) or 's' (strict mode).

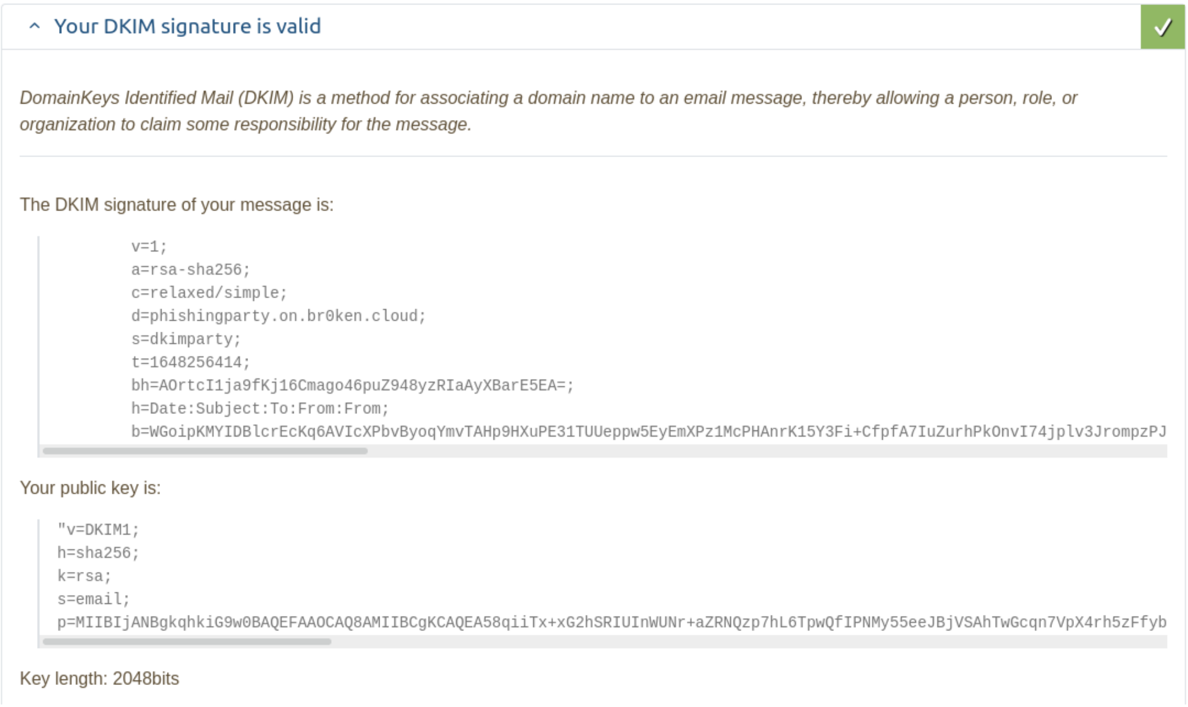
	Test	Result
!	DMARC Policy Not Enabled	DMARC Quarantine/Reject policy not enabled
✓	DMARC Record Published	DMARC Record found
✓	DMARC Syntax Check	The record is valid
✓	DMARC External Validation	All external domains in your DMARC record are giving permission to send them DMARC reports.
✓	DMARC Multiple Records	Multiple DMARC records corrected to a single record.

Figure 14: DMARC record verification by MX Toolbox [42]

The email with the complete DKIM signature is provided in Appendix D.

8.4.2 DKIM

While DNS records are static, DKIM generates on-the-fly the corresponding signatures for every sent-out mail. Thus, we need to verify if the signatures are correctly created. The mail-tester.com result in Figure 15 verifies that our mail server correctly signs the sent mails.



^ Your DKIM signature is valid ✓

DomainKeys Identified Mail (DKIM) is a method for associating a domain name to an email message, thereby allowing a person, role, or organization to claim some responsibility for the message.

The DKIM signature of your message is:

```
        v=1;
        a=rsa-sha256;
        c=relaxed/simple;
        d=phishingparty.on.br0ken.cloud;
        s=dkimparty;
        t=1648256414;
        bh=AOrtcI1ja9fKj16Cmago46puZ948yzRIaAyXBarE5EA=;
        h=Date:Subject:To:From:From;
        b=WGoipKMYIDBlcrEcKq6AVIcXPbvByoqYmvTAHp9HXuPE31TUUeppw5EyEmXPz1McPHAnrK15Y3Fi+CfpfA7IuZurhPkOnvI74jplv3JrompzPJ
```

Your public key is:

```
"v=DKIM1;
h=sha256;
k=rsa;
s=email;
p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA58qiiTx+xG2hSRIUInWUNr+aZRNQzp7hL6TpwQfIPNMy55eeJBjVSAhTwGcqn7VpX4rh5zFfyb
```

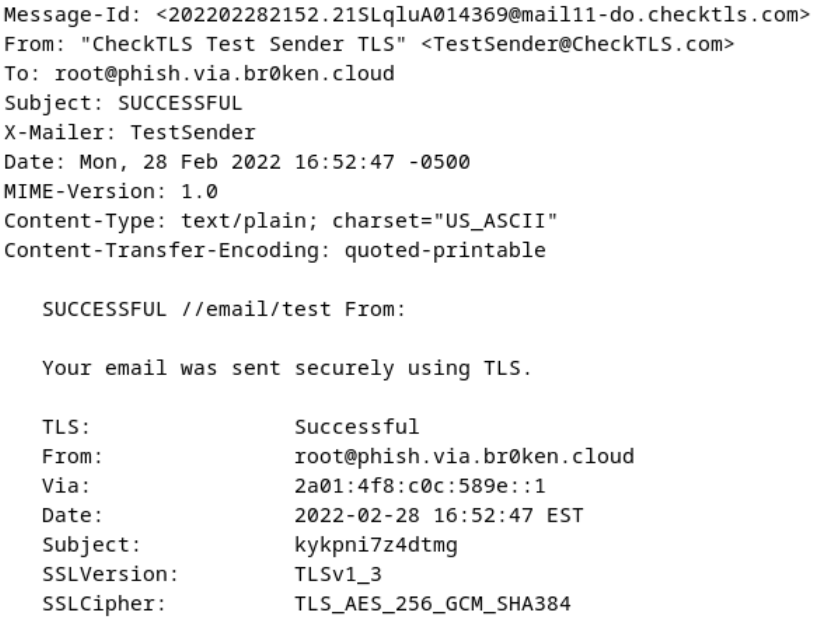
Key length: 2048bits

Figure 15: Valid DKIM signature verified by mail-tester.com [43]

It can also be seen in Figure 7 that the DKIM check by the Thunderbird extension DKIM Verifier [36] is valid, and the mail is correctly signed by the mail server.

8.4.3 TLS

As mail-tester.com does not check the TLS configuration of the mail server for sending and receiving mails, CheckTLS was used to verify the correctness of our implementation. For the verification of secure transmission of emails to other mail servers, we used the TestSender functionality of CheckTLS. The response mail in Figure 16 demonstrates that our implementation uses TLS for mail sending.

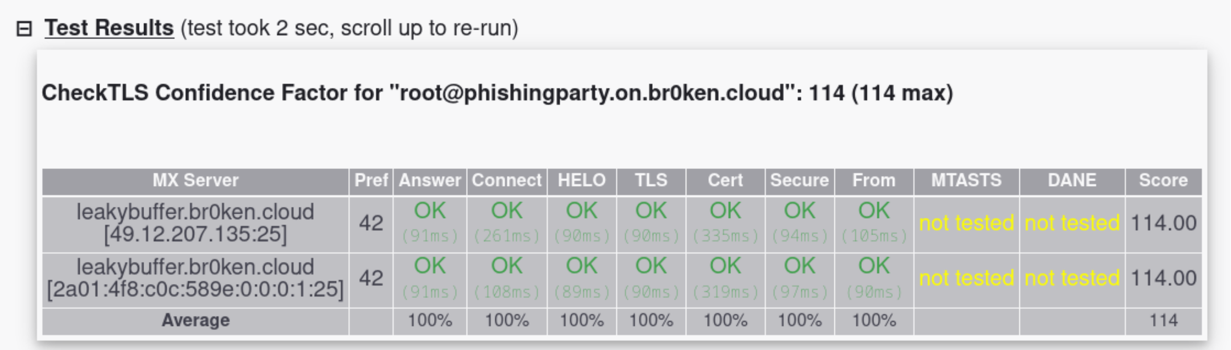


```
Message-Id: <202202282152.21SLqluA014369@mail11-do.checktls.com>
From: "CheckTLS Test Sender TLS" <TestSender@CheckTLS.com>
To: root@phish.via.br0ken.cloud
Subject: SUCCESSFUL
X-Mailer: TestSender
Date: Mon, 28 Feb 2022 16:52:47 -0500
MIME-Version: 1.0
Content-Type: text/plain; charset="US_ASCII"
Content-Transfer-Encoding: quoted-printable

   SUCCESSFUL //email/test From:

   Your email was sent securely using TLS.

   TLS:                Successful
   From:               root@phish.via.br0ken.cloud
   Via:                2a01:4f8:c0c:589e::1
   Date:               2022-02-28 16:52:47 EST
   Subject:            kykpni7z4dtmg
   SSLVersion:         TLSv1_3
   SSLCipher:          TLS_AES_256_GCM_SHA384
```

Figure 16: Sending emails using TLS verified by checktls.com [8] TestSender

The detailed results are available in Appendix D.

Validating TLS support for receiving mails is useful because mail server security solutions may check if the mail server supports it and could discard the mail, classify it as spam or warn the user. According to the TestReceiver result in Figure 17, the server supports secure mail submission over IPv4 and IPv6 from other mail servers.



⊟ **Test Results** (test took 2 sec, scroll up to re-run)

CheckTLS Confidence Factor for "root@phishingparty.on.br0ken.cloud": 114 (114 max)

MX Server	Pref	Answer	Connect	HELO	TLS	Cert	Secure	From	MTASTS	DANE	Score
leakybuffer.br0ken.cloud [49.12.207.135:25]	42	OK (91ms)	OK (261ms)	OK (90ms)	OK (90ms)	OK (335ms)	OK (94ms)	OK (105ms)	not tested	not tested	114.00
leakybuffer.br0ken.cloud [2a01:4f8:c0c:589e:0:0:0:1:25]	42	OK (91ms)	OK (108ms)	OK (89ms)	OK (90ms)	OK (319ms)	OK (97ms)	OK (90ms)	not tested	not tested	114.00
Average		100%	100%	100%	100%	100%	100%	100%			114

Figure 17: Receiving emails securely verified by checktls.com [8] TestReceiver

8.5 Comparison with less secure mail server implementations

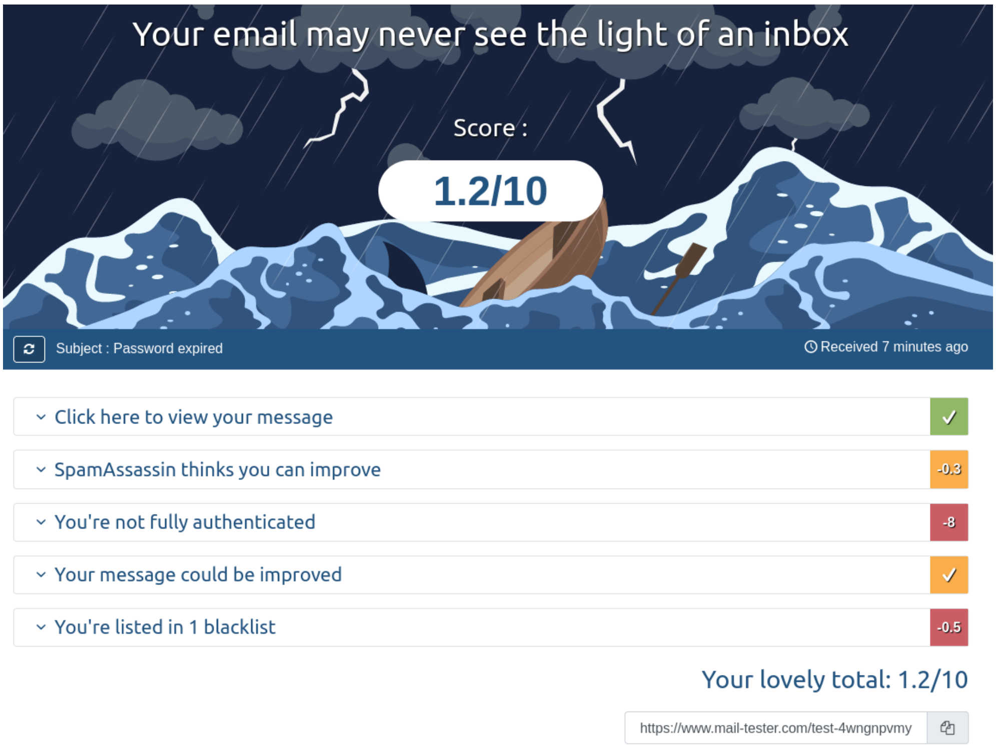
A comparison was conducted to prove that implementing measures like SPF, DKIM, and DMARC offers a benefit over a simple mail server setup. The identical tests were carried out using the free online mail server test suite mail-tester.com.

The following mail configurations were tested:

- Configuration 0: mail server with no DNS records set
- Configuration 1: mail server with correct reverse DNS, A record, and MX record
- Configuration 2: Like Configuration 1 but with additional configured SPF, DKIM, and DMARC

Tested mail server configurations			
	Configuration 0	Configuration 1	Configuration 2
A record for hostname	✗	✓	✓
MX record for hostname	✗	✓	✓
rDNS configured	✗	✓	✓
DKIM configured	✗	✗	✓
SPF Record	✗	✗	✓
DMARC Record	✗	✗	✓

Table 2: Tested mail server configurations


Your email may never see the light of an inbox

Score :

1.2/10

Subject : Password expired

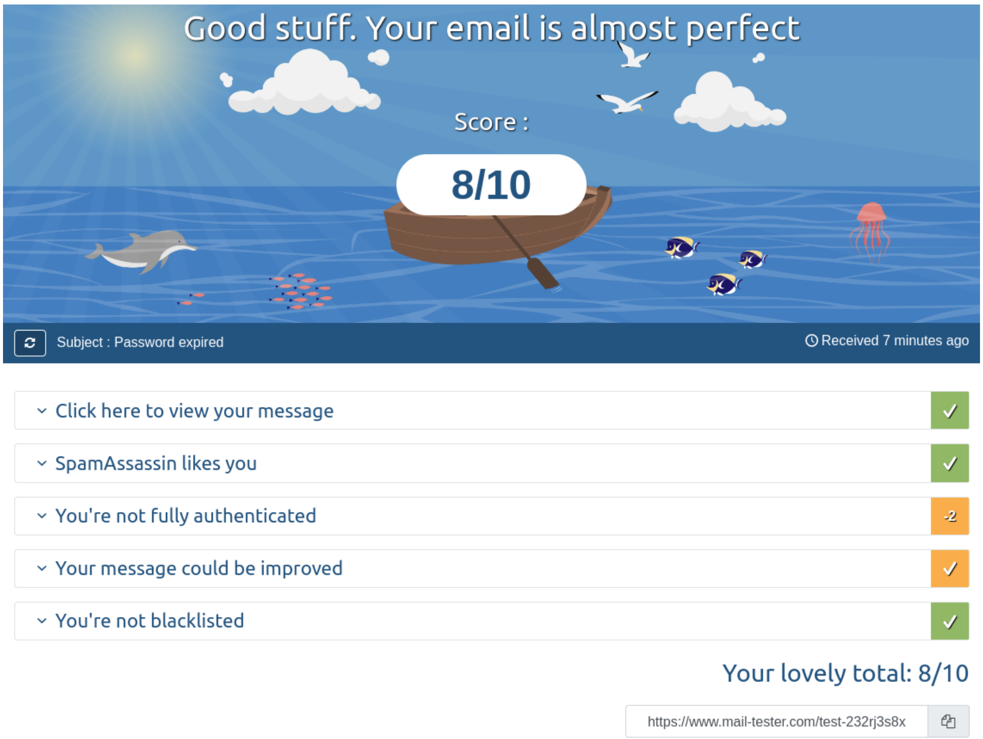
Received 7 minutes ago

- Click here to view your message ✓
- SpamAssassin thinks you can improve -0.3
- You're not fully authenticated -8
- Your message could be improved ✓
- You're listed in 1 blacklist -0.5

Your lovely total: 1.2/10

https://www.mail-tester.com/test-4wngnpvmy

Figure 18: Result Configuration 0: mail server with no records [10]


Good stuff. Your email is almost perfect

Score :

8/10

Subject : Password expired

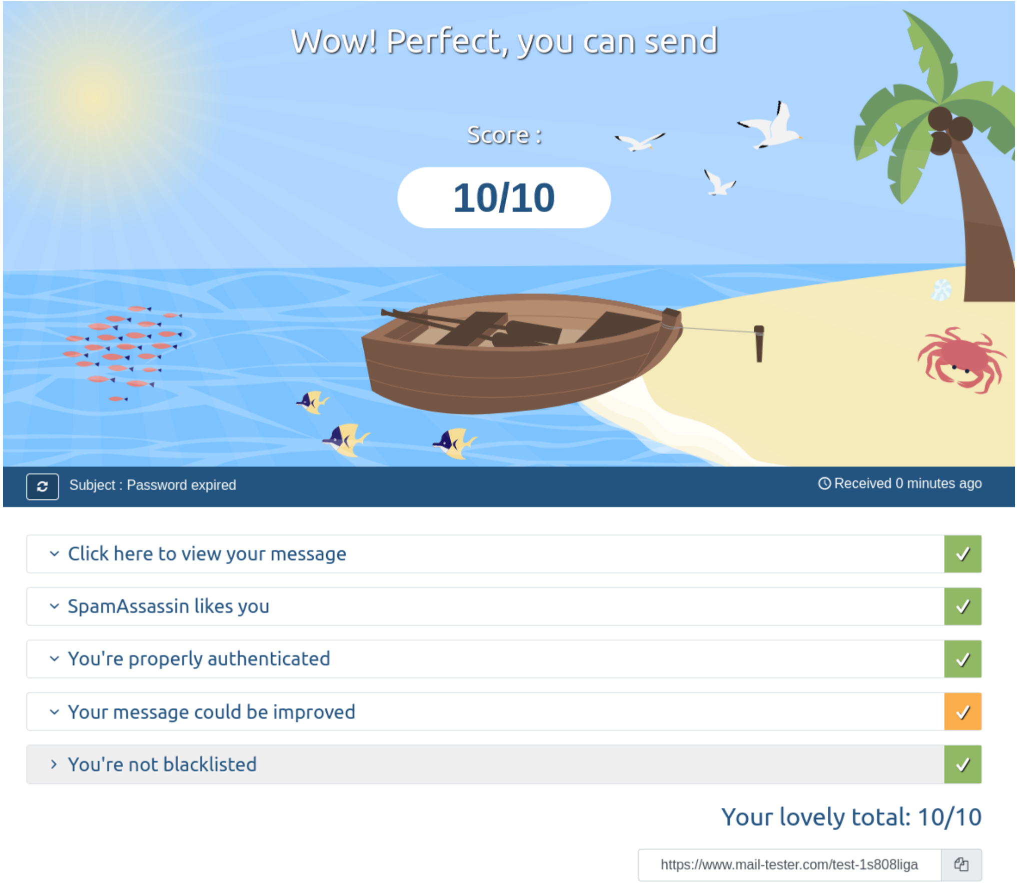
Received 7 minutes ago

- Click here to view your message ✓
- SpamAssassin likes you ✓
- You're not fully authenticated -2
- Your message could be improved ✓
- You're not blacklisted ✓

Your lovely total: 8/10

https://www.mail-tester.com/test-232rj3s8x

Figure 19: Result Configuration 1: mail server with basic records [11]


Wow! Perfect, you can send

Score :

10/10

Subject : Password expired

Received 0 minutes ago

- Click here to view your message ✓
- SpamAssassin likes you ✓
- You're properly authenticated ✓
- Your message could be improved ✓
- You're not blacklisted ✓

Your lovely total: 10/10

https://www.mail-tester.com/test-1s808liga

Figure 20: Result Configuration 2: mail server with additional security measures [12]

Test results of the setups compared			
	Configuration 0	Configuration 1	Configuration 2
SpamAssasin Score	-0.3	✓	✓
Authentication	-8	-2	✓
Message Improvement	✓	✓	✓
Blocklist	✓ (-0.5)[1]	✓	✓
Final Score	1.7/10	8/10	10/10

Table 3: mail-tester.com result comparison for the configurations

The result presented in Table 3 shows clearly that our secure mail server configuration reaches the maximum score in the testing suite. A mail server without SPF, DKIM, and DMARC still yields a good result, and the email acceptance will be high.

[1] We assume that the IP is not on a Blocklist for this test - Therefore, the resulting 0.5 points were omitted

Running a mail server with no set records will likely result in the rejection of emails. Therefore, the chance of getting a mail delivered will be close to 0.

Based on the results in Table 3, the following point deduction scheme for mail-tester.com could be derived:

Reason	Points deducted
Relay HELO differs from its IP's reverse DNS (SpamAssassin)	-0.3
Missing or wrong SPF (Authentication)	-1
Not DKIM signed (Authentication)	-1
Missing MX record (Authentication)	-3
Missing A record (Authentication)	-3
Listed on a Blocklist - per list entry	-0.5

Table 4: mail-tester.com point deductions schema

8.6 Comparison with solutions from related work

Based on the designed requirements catalog, our solution was compared to three similar projects, which are presented in chapter 4. Table 5 shows an overview of the differences between the solutions. All implementations offer support for the simple Phishing scenario with Gophish, but only two of them, including ours, provide the possibility for complex Phishing scenarios. Two implementations implement a self-hosted mail server setup, but only our implementation configures additional security features like SPF, DKIM, or DMARC. For the automated setting of DNS records, terraform-phishing and ansible-playbook-gophish deploy them via Terraform. On the other hand, our solution offers a native implementation via Ansible for this task. Additionally, our implementation is extendable with other DNS providers, including those that are not covered by Terraform. Advanced OPSEC configuration is not implemented in the prototype due to ethical concerns, as discussed in chapter 9. The licensing is diverse, and the planned license for our prototype is still under consideration. Unlike all other solutions, our prototype does not offer resource configuration via dedicated software like Terraform because it was beyond the scope of this thesis. Nevertheless, our solution is cloud provider agnostic compared to the other solutions.

Phishing infrastructure deployment solutions				
	Phishing Oida! (our solution)	Build_A_Phish	terraform-phishing	ansible-playbook-gophish
implemented Phishing tools	Gophish; Modlishka	Gophish; Evilginx2	Gophish	Gophish
Simple Phishing Scenarios (Fake login page)	✓	✓	✓	✓
Complex Phishing Scenarios (2FA Bypass)	✓	✓	✗	✗
self-hosted mail server	✓	✗	✗	✓
secure mail server configuration: DKIM/SPF/DMARC	✓	✗	✓	✗
setting DNS records automated	✓ - via Ansible (limited)	✗	✓ - via Terraform (limited)	✓ - via Terraform (limited)
advanced OPSEC	✗	✓ - Cloudflare DNS/Azure	✗	✓ - Cloudflare DNS
(Open-Source) License	To be discussed	MIT	GNU GPLv3	None defined
Resource configuration (e.g. Terraform)	✗	✓	✓	✓
Specific cloud provider support (Terraform)	No	DigitalOcean	DigitalOcean; Azure	DigitalOcean; Hetzner

Table 5: Comparison of Phishing deployment code

8.7 Discussion

The designed requirements catalog offers a measurement opportunity for current and future solutions. By comparing our solution to other solutions, we prove that our prototype covers all defined main features in chapter 6 and show that the other solutions do not satisfy them.

Our prototype presents a possible deployment solution. It offers security professionals a suitable way to deploy Phishing infrastructure on their infrastructure and use it for legal assessments. For example, it could be used for running Phishing campaigns internally by the internal security engineer to improve awareness about Phishing or for an offensive security assessment in a red team exercise.

The test suites used verify that the mail server is properly configured security-wise when it comes to mail delivery. The comparison with less secure mail servers shows that our secure mail server implementation is significantly better by over 80% than a mail server setup lacking basic configuration. In comparison with a mail server with basic settings but no additional security settings, it is still better by 20%. Regarding research question RQ3, our solution did not manage to improve the result by 25% compared to a basic non-secure mail server setup.

Due to the usage of Ansible and the modular approach, our solution allows easy extensibility for additional tools and features.

8.7.1 Limitations

The presented solution only offers deployment for the Phishing infrastructure itself. It does not aim to set up a fully-fledged and ready-to-send Phishing campaign. We believe that this is the responsibility of the users. As the target groups are IT Security experts, they should have the knowledge to design and execute legal Phishing assessments. Therefore, our solution does not cover steps to configure Phishing campaigns themselves.

8.7.2 Implementation Pitfalls

Two pitfalls occurred during the prototype implementation regarding the Gophish CSRF implementation and issuing wildcard certificates via dehydrated.

Gophish - non reliable CSRF implementation

To obtain the API key, one must be logged in. When logging in with the initial password for the first time, a new password needs to be set. After the successful login with the new credentials, the API key is delivered in the HTML body of the landing page. To improve security the Gophish software implements a Cross-Site Request Forgery (CSRF) protection that requires a unique token that has been provided to be sent along with the request. A Python script, as referenced in subsubsection 8.1.2, was implemented that automated the login procedure, set the new password, and extracted the API key. While testing the script, we encountered the CSRF error `Forbidden - CSRF token invalid` despite the fact that we sent the correct one. In the project's issues tracker on GitHub, other people also reported some problems with the CSRF functionality. It proved our assumption about the non-reliability of this feature. Therefore, we implemented a while loop in the Pythons script that executed the procedure multiple times until it was successful.

Issuing Wildcard Certificates with dehydrated

When implementing the Postfix and Gophish deployment, we decided to issue certificates for those two tools with nginx in combination with dehydrated. Dehydrated was chosen due to the availability of the package in the Debian repositories and good documentation. During the implementation of Modlishka, we realized that we need to issue wildcard certificates that require an ACME DNS01 challenge. During the DNS challenge, specific DNS TXT records must be deployed on the domain for which a wildcard certificate is requested. Due to the nature of dehydrated that requires a script hook for this step, we realized that it is not implementable in Ansible with already available code for deploying DNS records. Instead of implementing an additional shell script for the script hook, we decided to rely on acme.sh as a different certificate issuing tool to create wildcard certificates for Modlishka. acme.sh was chosen because it offers support for multiple DNS providers for record deployment and could be implemented in Ansible.

9 Ethical Considerations and Evaluation - Survey

We are planning to release the prototype to the public in the future. We had ethical considerations because this technology could not only be used for good but could also to do harm when used by criminals. The FH Technikum Vienna did not have at the time this thesis was written a dedicated ethics commission that could be asked for advice. Therefore, we decided to ask IT Security professionals for their opinion on this topic. The idea was to get feedback and use it for the discussion and later on for the decision about the publication. Additionally, we thought about how we could create a helpful prototype that could be published that would do good and would be less harmful in general. With these considerations in mind, we decided not to implement advanced methods for obfuscation like hiding infrastructure IPs through CDNs and pre-configured Phishing Campaigns themselves. The prototype is designed for an expert that knows how to exercise Phishing engagements. If additional obfuscation is needed for assessments, like Red Teaming, the experts should implement those extra techniques themselves.

9.1 Survey Structure

The first page of the survey shows an informative introduction text about the questionnaire and requires the use of data and the privacy policy to be accepted. The survey consists of eleven questions, split into five mandatory and six optional ones.

The mandatory questions consist of:

- 3 x yes/no
- 2 x multiple-choice

The three yes/no questions were followed by an optional free-text entry, which asked for the personal opinion on why releasing the source code to the public was or was not ethical. The other led to multiple-choice questions, which differed depending on the person's choice. For the possible answers to the multiple-choice questions, typical options were considered.

The optional questions were divided into:

- 4 x single choice
- 1 x free-text
- 1 x multiple-choice

Therefore, persons only need to answer five questions. The small number of mandatory questions combined with a multiple-choice possibility improves the acceptance for participation and the completion of the survey. Another important aspect was the privacy by default approach while designing the questionnaire. It was essential to collect as many answers as possible without having the possibility to trace the answers back to a participating individual. Therefore, no personal data was gathered except an optional email address to notify the respondent about the result. The email address was collected in a separate survey and was not linkable to the answers given by the person. Additionally, the design of the survey enables an efficient evaluation of the responses Figure 21 depicts the flow of the survey.

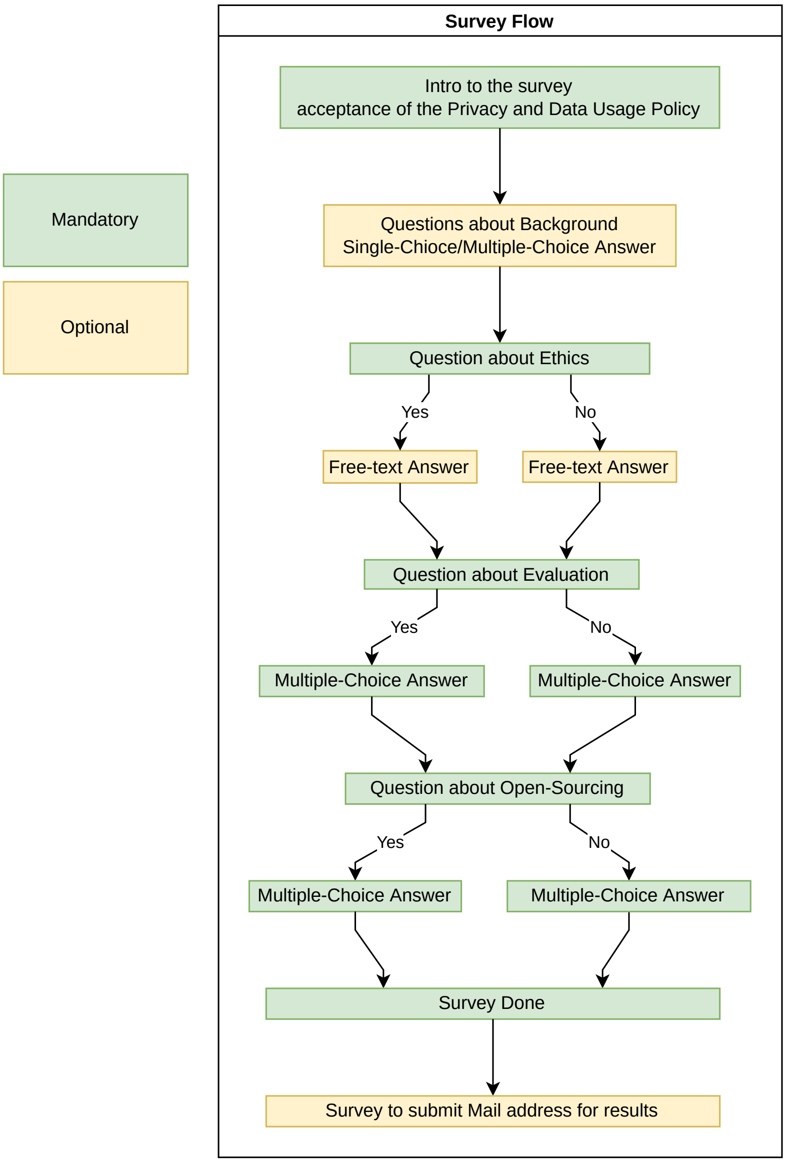


Figure 21: Survey Flow

The survey questions are provided in Appendix B.

9.2 Survey - Technical Implementation

Due to the aspiration for data sovereignty, the survey was set up on a personal server infrastructure. For the survey, the well-known Free and Open Source Software (FOSS) LimeSurvey [37] was chosen. For security reasons and separation, it was deployed in a container-managed environment with Podman [49]. The well-maintained and recent LimeSurvey Container Image [45, 44] from Markus Opolka [46] was picked as the basis for the container. nginx was used as a reverse proxy, and Let's Encrypt provided the TLS certificates to assure a secure connection to the server. For the survey, the domain `phish.on.br0ken.cloud` was chosen, and a single link was provided that was used by all attendees conducting the survey. This ensured that anonymous participation for all participants in the survey was possible. Additionally, all tracking settings were turned off in the LimeSurvey settings for the survey.

To prevent collecting personally identifiable information, the IP addresses and the timestamps of the accesses were not logged on the nginx server. A custom logging format was used to accomplish this. The code in listing 1 replaces all IP addresses by the dummy value `:badc:ab1e::`. To remove additional linkable data, the custom nginx log format, which is defined in code listing 2, does not log the timestamps and referrers for the web server accesses.

```
map $remote_addr $remote_addr_anon {
       ~(?P<ip>[^:]+:[^:]+):      :badc:ab1e::;
       default                    :badc:ab1e::;
}
```

Listing 1: nginx - IP address replacement script

```
    log_format loganon_no_time '$remote_addr_anon
↪   $http_x_forwarded_for - $remote_user '
    '"$request_method $scheme://$host$request_uri
↪   $server_protocol" '
    '$status $body_bytes_sent $request_time';
```

Listing 2: nginx - Custom log format for the LimeSurvey instance

9.3 Survey Results

The survey was publicly available for four weeks from 2022-03-12 until 2022-04-11. A total of 474 persons conducted the survey, and 350 completed it. This equals a completion rate of nearly 74%. Only completed questionnaires are counted for the evaluation because they provide a common basis for analysis.

9.3.1 Background Information about the participants

The first optional question asked the participants about their background. All except one participant answered the optional background questions.

Source of the Survey

The results in Figure 22 and Table 6 represent the distribution of sources from where the participants got the links for the survey. Twitter is leading with nearly 37%. The Other Source results combine various platforms, which are not mentioned in the answer options. With over 26%, it was the 2nd best source for answers. Surveys directly forwarded to people contributed nearly 23% to the complete ones. Participants that got the questionnaire via email accounted for over 11% of the answers. LinkedIn was the least successful platform and accounted for only under 3% of the completed questionnaires.

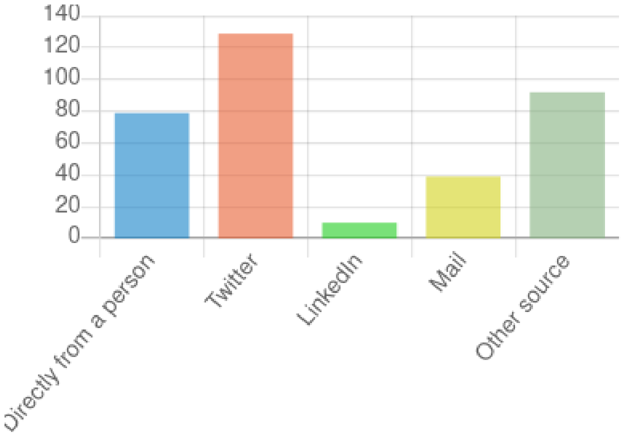


Figure 22: Source of the survey

Answer	Count	Percentage
Twitter	129	36.96%
Other source	92	26.36%
Directly from a person	79	22.64%
Mail	39	11.17%
LinkedIn	10	2.87%
Total	**349**	**100%**

Table 6: Survey Source - total numbers

Years of Experience in the IT Field

The results in Table 7 regarding years of experience in the IT Field shows that over 80% of the participants have more than five years of experience. Instead, nearly 20% of the participants have up to 4 years of experience.

Answer	Count	Percentage
< 1	17	4.87%
1-2	14	4.01%
2+	38	10.89%
5+	62	17.77%
10+	218	62.46%
Total	**349**	**100%**

Table 7: Years of experience in the IT field - total numbers

IT Security Team current belonging

Table 8 represents the self-evaluation of the participants based on the four choices offered and their current situation. This question only asked for the current team status of the participants. The most chosen answer with 40% was *None of the available choices*. The other 60% were distributed over the remaining three options.

It is noticeable that over half of those are part of the *Blue Team - Defensive Security*. *Red Team - Offensive Security* is the smallest group in this survey with only around 11%, followed by the *Purple Team* with 16%.

Answer	Count	Percentage
Red Team - Offensive Security	39	11.17%
Blue Team - Defensive Security	115	32.95%
Purple Team - Defensive and Offensive Security	56	16.05%
None of the available choices	139	39.83%
Total	**349**	**100%**

Table 8: IT Security Team belonging - total numbers

Years of Experience in the IT Security Field

With the questionnaire, we primarily targeted IT Security professionals. Therefore, it was interesting to know the years of experience in the field. According to the data, nearly 59% of the participants have more than five years of experience working in the IT Security industry. Of the remaining group, over 18% have two or more years, and 17% have less than one year of work experience. The smallest group of nearly 5.5% works between 1 and 2 years in this field.

Answer	Count	Percentage
< 1	60	17.19%
1-2	19	5.44%
2+	64	18.34%
5+	99	28.37%
10+	107	30.66%
Total	**349**	**100%**

Table 9: Years of experience in the IT Security field - total numbers

Areas of expertise

The area of expertise was a multiple-choice question in the survey. People could choose in which areas they have knowledge and work experience. Most of the choices are well distributed. However, System Administration is the job where most people have gained expertise. In contrast to System Administration, only a little more than one quarter conducted academic research.
We allowed multiple choice in these questions without restricting the number of selected items. Based on the numbers, persons have chosen an average of four positions in which they have gained experience. (Average = 4.14).

Answer	**Count**	**Percentage**[1]
System Administration	229	65.62%
Software Development	187	53.58%
Building IT Architecture	180	51.58%
IT Security Consulting	171	49.00%
Incident-Response	159	45.56%
Blue-Team	142	40.69%
Red-Team/Pentesting	141	40.40%
IT Management	133	38.11%
Academic Research	97	27.79%
None of the available choices	6	1.72%
Total	**1445**	**100%**

Table 10: Areas of Expertise IT (Security) Field - total numbers

9.3.2 Opinions about Ethical Justifiability for Public Release

The participants' opinions show a precise result regarding the ethical justifiability to release deployment code for Phishing infrastructure. Over 83% find it justifiable to release it to the public. Instead, only around 16% are against the publication of such code.

Answer	**Count**	**Percentage**
Yes	293	83.71%
No	57	16.29%
Total	**350**	**100%**

Table 11: Opinion about ethical justifiability to open-source deployment code - total numbers

To get additional insight and answer research sub-question RQ 5.1, we analyzed the free-text answers of the participants.
Therefore, a qualitative analysis was conducted by grouping the pro and contra arguments into categories.

[1] The percentage is derived from 349 participants that equals all those who answered the optional background questions.

Pro Arguments

Of the 292 participants who find it ethical to release the code to the public, 175 submitted optional textual reasoning. The answer rate equals nearly 60%.

Answer provided	Count	Percentage
Yes	175	59.93%
No	117	40.07%
Total	**292**[2]	**100%**

Table 12: Pro opinion answers - total numbers

For every free-text response, the core statement was analyzed and categorized. As a result, 17 categories and one category for none meaningful answers were created. Table 13 shows the statistics for each category, including their corresponding percentage. We could extract the following two main statements that appeared in most of the questions:

- Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist to carry out such activities
- People/companies can learn from the published software and therefore protect themselves

The two most provided categorized reasons with the same amount of opinions state that releasing helps defenders and it will not impact criminality because criminals already have their tooling. The second summary states that people can learn about the methods and techniques and defend themselves based on this with the published software. Another category supports the publication because it provides an easy way to set up Phshing infrastructure to provide training on awareness to the users. Four answers could not be categorized into a meaningful category in the context of the question.

The free-text answers, including our categorizations, are provided in Appendix C.

[2] There exists one answer for the pro arguments in the dataset that is defined as *Not displayed*, and we do not know what happened with this answer inside LimeSurvey. Therefore, we used 292 as a basis instead of 293.

Category	**Count**	**Percentage**
Helps the defenders, open-sourcing will not have an impact on criminal activity because the tools already exist	38	21.71 %
People/companies can learn from the published software and therefore protect themselves	38	21.71 %
Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users	23	13.14 %
Software is Dual Use - Responsible is the user	16	9.14 %
Community can check code, fix bugs, extend features and it can be a basis for future work or research	11	6.29 %
Public Tools create awareness and motivation to tackle the threats/problems	7	4.00 %
Information should be free/public for everyone	7	4.00 %
Security by obscurity is not a sufficient basis for security	7	4.00 %
Will benefit for defenders more than criminals	6	3.43 %
The concept is nothing new or special and the complexity for implementing the solution it not that high	5	2.86 %
cannot be meaningfully categorized	4	2.29 %
Phishing Tools are already available, Open Sourcing does not change anything and it is also possible that someone else will make it	4	2.29 %
Commercial tools already exists, therefore, open-source would be an alternative	2	1.14 %
It should provide a limited infrastructure with a defined legitimate scope	2	1.14 %
Automation can not replicate good Phishing tactics and procedures	2	1.14 %
Phishing should be defended with existing security solutions	1	0.57 %
We need to educate our employees against such threats	1	0.57 %
In favor of all activities that stop hackers and improves resilience of business	1	0.57 %
Total	**175**	**100%**

Table 13: Pro opinion free-text answers - total numbers

Contra Arguments

Of the 57 participants who did not find it ethical to release the code to the public, 31 submitted optional textual reasoning.

Answer provided	Count	Percentage
Yes	31	54.39%
No	26	45.61%
Total	**57**	**100%**

Table 14: Contra opinion answers - total numbers

For every free-text response, the core statement was analyzed and categorized. As a result, eight categories and one category for none meaningful answers were created. Table 15 shows the statistics for each category, including their corresponding percentage. The following three main reasons against publication could be extracted from the answers:

- Exploitation by criminals or "Script Kiddies"
- Will do more harm than good
- Should not make things easier for attackers or "Script Kiddies"

The reason most given against was that releasing it to the public would do more harm than good. E.g., the damage by malicious actors would outnumber the benefit from the legitimate usage. There were also significant concerns that it would be exploited by criminals or so-called "Script Kiddies" and make exploitation and abuse easier for malicious attackers. Two answers could not be categorized into a meaningful category in the context of the question.

The free-text answers, including our categorizations, are provided in Appendix C.

Category	Count	Percentage
Will do more harm than good	9	29.03%
Exploitation by criminals or "Script Kiddies"	7	22.58%
Should not make things easier for attackers or "Script Kiddies"	7	22.58%
Because it is unethical in their opinion	2	6.45%
cannot be meaningfully categorized	2	6.45%
No benefit for the general public	1	3.23%
Should not be publicly available	1	3.23%
Should be only available for legit usecases	1	3.23%
Additional malicious code could be introduced	1	3.23%
Total	**31**	**100%**

Table 15: Contra opinion free-text answers - total numbers

9.3.3 Position on open-sourcing Deployment Code themselves

Even though nearly 84% say it is ethically justifiable to release code to the public, just around 79% would also open-source it themselves.
We provided multiple reasons as a choice in the survey with regard to motivation to publish code by themselves to answer research sub-questions RQ 5.2.

Answer	Count	Percentage
Yes	277	79.14%
No	73	20.86%
Total	**350**	**100%**

Table 16: Position on open-sourcing Deployment Code themselves - total numbers

Motivation for open-sourcing code themselves

Several pre-provided statements for this question supported open-sourcing code by themselves. The results in Table 17 show three key arguments for the motivation behind releasing code under their name. In general, contribution to the improvement of IT Security has the most votes, followed by trusting in the concept of open-source software, and people that could contribute new features. The majority of the participants would also give something back to the community with such a project. Using the project for their project portfolio to improve their CV was the least chosen reason with around 26%.

We allowed multiple choice in these questions without restricting the number of selected items. Based on the numbers, persons have chosen an average of three to four positive arguments (Average = 3.60).

Answer	**Count**	**Percentage**
I would contribute to improving IT Security in general	206	74.37%
In my opinion, open-sourcing such software is the way to go because I believe in this concept	203	73.29%
Other people could contribute and implement new features	201	72.56%
I would give something back to the community	187	67.51%
The community could take over the maintenance of the project	118	42.60%
It would be beneficial for my project portfolio, and I would mention it in my CV	73	26.35%
None of the available choices	9	3.25%
Total	**997**	**100%**

Table 17: Position for open-sourcing Deployment Code themselves - total numbers

Motivation against open-sourcing code themselves

According to the result, 20% of participants would not open-source such solutions under their name. Unlike the answers in favor of open-sourcing, none of the responses have a majority in relation to the number of voters. The biggest concern against open-sourcing was the missing time for maintenance and management of such a project. According to 30% of the persons, the publication would violate their company policy or contract of employment. Just around 11% would fear negative publicity from such a publication. Only two people were in favor of proprietary software.

We allowed multiple choice in these questions without restricting the number of selected items. Based on the numbers, persons have chosen an average of one to two negative arguments, with a small tendency towards two choices. (Average = 1.64).

Answer	Count	Percentage
I have no time to invest in the maintenance or management of the project	26	35.62%
It would not be ethical to open source it because criminals or "Script Kiddies" could use it for illegal activities	25	34.25%
The tool should be only available to a limited circle of people (e.g., security professionals)	23	31.51%
It may violate my company policy or my working contract	22	30.14%
None of the available choices	9	12.33%
I fear negative publicity because of the project	8	10.96%
I would not like to share my work with everyone	5	6.85%
In my opinion, proprietary software is the better solution	2	2.74%
Total	**120**	**100%**

Table 18: Position against open-sourcing Deployment Code themselves - total numbers

9.3.4 Interest about Prototype Evaluation

Having conducted the survey, we wanted to explore the interest in evaluating our created prototype. Therefore, we asked the participants if they would evaluate our solution and provided arguments for or against it. Table 19 shows that two-thirds of the users would be interested in evaluating our solution. This result is an additional motivation for us to open-source the prototype to the public.

Answer	Count	Percentage
Yes	233	66.57%
No	117	33.43%
Total	**350**	**100%**

Table 19: Interest in prototype evaluation - total numbers

The arguments are evaluated below regarding the evaluation.

Arguments for Evaluation

Forwarding the prototype to colleagues that would use it for awareness training was the most selected argument with over 58%. Almost half of the participants offer defensive security training, and using our prototype would help them save time setting up the infrastructure for the awareness training. Providing the prototype code would help over 38% of the participants improve their Ansible skills. For over one-fourth of the people, it would enable them to add Phishing Assessments to their offered security services. Based on the answers, it is evident that we provided a suitable set of positive arguments supporting the interest for evaluation because only 4% of the participants did not identify with any available choices.

We allowed multiple choice in these questions without restricting the number of selected items. Based on the numbers, persons have chosen an average of two to three arguments (Average = 2.48).

Answer	Count	Percentage
I could send it to colleagues, which would probably use it for awareness training	136	58.37%
I do defensive security, and it could save me time for setting up a Phishing training	115	49.36%
It would help me improve my Ansible skills	89	38.20%
I do offensive security, and it could save me time for setting up a Phishing campaign	71	30.47%
I could send it to colleagues, which would probably use it for offensive assessments	69	29.61%
It would enable me to add Phishing Assessments to my offer of security services	63	27.04%
None of the available choices	10	4.29%
Total	**553**	**100%**

Table 20: Arguments in favor of evaluation - total numbers

Arguments against Evaluation

Only one-third of the participants would not evaluate our prototype. The most chosen argument for over 40% was the lack of time to conduct internal Phishing assessments. The second most selected argument was a fair point: People are not familiar with Ansible or do not have time/interest to learn it. Less chosen aspects stated that the persons already use some software for Phishing or (want to) pay external companies for this use case. For 30% of the participants, none of the provided arguments were fitting.

We allowed multiple choice in these questions without restricting the number of selected items. Based on the numbers, persons have chosen an average of one to two arguments, with a tendency towards a single choice (Average = 1.36).

Answer	Count	Percentage
I don’t have time to conduct internal Phishing Assessments	48	41.03%
None of the available choices	36	30.77%
I am not familiar with Ansible and do not have the time or interest to learn it	28	23.93%
I (want to) pay external companies or consultants to do the Phishing Assessments	13	11.11%
I am already using my tooling (self-scripted) and would not like to change this	12	10.26%
I am already using commercial software for this use-case	12	10.26%
I am already using open-source software for this use-case	10	8.55%
Total	**159**	**100%**

Table 21: Arguments against evaluation - total numbers

9.4 Discussion

The number of participants and their detailed answers to the free-text questions provide us with sufficient material for a discussion regarding research question 5 and the sub-questions.

9.4.1 Discussing Participants Background

The nearly complete number of information provided on source and background offers some additional conclusions on the given answers.

Most of the survey advertisements were conducted via social media platforms like Twitter or chat platforms, which was a key to the success of this survey.

With the help of Twitter users with a reach of over 100.000 followers who shared our tweet, we could reach a high number of participants over this platform. Therefore, having support from people with a high follower count means it is possible to reach a significant number of people.

The survey was also shared over the platforms Matrix, Mattermost, Mastodon, and Discord. According to the results, forwarding surveys directly to people is effective and counted for nearly one-quarter of the answers.

We only sent the questionnaire to a single mailing list. The "discuss" mailing list of the Austrian Computer Emergency Response Team (CERT) was selected because the subscribers are persons in the Austrian IT Security field working in a broad range of positions. The 39 answers are a solid return rate for a single mail.

Due to our limited reach on LinkedIn, the result of under 3% was not surprising for us.

Overall, the promotion campaign for the survey was successful, and we reached our goal of collecting enough participants to get statistically significant data.

We asked for a self-assessment of the current position for the team question. The choice for the team question that stood out with nearly 40% is *None of the available choices*. Therefore, the persons could have belonged in the past to one of the other teams that were not chosen. One explanation for this result may be the long work experience of the participants. Workers with multiple years of experience tend to move up the job ladder and switch from technical jobs to management. In this position, they may not execute practical, technical work anymore and, consequently, probably do not have an affiliation with the available team choices in the management role. Another possible explanation is that people working in positions like software engineers would rather choose the *None of the available choices* as the best suitable option. Nevertheless, over 60% of the participants chose one of the other three available teams.

The results about years of experience in the IT and IT Security industry show that most answers are based on extensive experience in those fields.

Evaluation of results regarding areas of expertise leads to the conclusion that most participants are from the IT Security industry. As only around one-quarter of the people have a background in academic research, this is not surprising because academia is not a requirement to start working in the technological sector.

9.4.2 Discussing Opinions about Ethical Justifiability for Public Release

Over 80% of the participants support the publication of our prototype. The submitted free-text answers provided a broad range of explanations for their decision. Grouping them into 18 categories to summarize their core message was challenging but could be successfully managed. This result encourages us to pursue our intention to release our prototype to the public.

However, also contra arguments were provided that were considered by us. Although the participants provided only a limited number of 31 free-text answers for contra arguments questions, it was still possible to gather valuable opinions that go against the publication of our prototype. The top three arguments are understandable and will be considered for our open-source plans.

Two opinions were very elaborative, and we want to discuss them thoroughly below:

Free-text answer number 8 was categorized into **Will do more harm than good**. The author of this answer criticizes that our question is flawed because we did not define a framework for ethics. We acknowledge the criticism, and in retrospect, we should have formulated the question in a more precise way. The person recommends conducting a sensible disclosure to provide information about possible targets of such tooling. We will take this feedback on board and include an appropriate disclosure and license.

Opinion number 17 was categorized into **Should not make things easier for attackers or "Script Kiddies"**. The participant argues against publication because it would lower the barrier for bad guys by providing a fully weaponized version. As a measure, we considered these concerns and did not provide any material regarding content aspects of Phishing or methods for execution. Additionally, we abstained from implementing advanced obfuscation methods for advanced OPSEC. The author argued that professionals should think more about harm reduction in current times than in the past. We believe that reducing harm is the main goal of IT Security in general, although it is hard to balance harm reduction and publishing information to support it. We believe that by not publishing practical solutions, there will not be a high motivation or pressure to improve security and make the status quo better.

Based on the evaluation of the pro and contra arguments, we still believe that open sourcing our prototype offers a helpful contribution to the improvement of IT Security in general and, in particular, awareness against Phishing.

9.4.3 Discussing Motivation regarding Open-Sourcing

After evaluating the numbers and comparing them with the answers about the ethical aspect of open-sourcing, not everyone would also favor releasing the code to the public. The difference is only around 4.5 % but still noticeable.

One aspect of the pro arguments that was surprising for us was that the least chosen option was the one mentioning the project in the CV. We thought more people would include it in their resumes to improve their job interview chances.

When analyzing the option choice, it is evident that the participants have chosen twice as many positive options as those that had to select negative ones.

9.4.4 Discussing Interest about Evaluation

The survey results regarding the interest in evaluation show that two-thirds of the Interviewees would be interested in evaluating it. One of the primary motivations for the evaluation is forwarding it to colleagues for awareness training. On the other hand, most people would not evaluate it due to lack of time. The answers from the contra arguments also show that we did not provide enough reasonable arguments against evaluation because 30% of the participants did not identify with any available option.

9.4.5 Limitations and Dangers to Integrity

Our survey depicted a small subset of the whole IT Security professionals population. All the answers provided are based on our subjective interpretation and can lead to another conclusion when analyzed by a third party.

The survey results with regard to the interest in evaluation show that pre-provided answers only offer a limited view. For example, 30% of the contra arguments are unknown due to the lack of suitable options.

The result shows significant support for the concept of open-source software. There was only a small acceptance of proprietary solutions. Therefore, the result may have a sampling bias, resulting in the preference for open-source software.

The technical implementation of the survey did not prevent multiple survey entries by a single person. Implementing such restrictions would have been a highly technical effort, often leading to lesser privacy and not guaranteeing 100 percent effectiveness. Therefore, it may be theoretically possible that one or more persons could have tampered with the survey results and harmed the integrity of the results.

10 Conclusion

A requirements catalog was created that provides the possibility of comparing the current and future solutions. The defined metrics inspect different aspects of deployment code for Phishing infrastructure and enable informed decision-making for interested IT Security professionals.

This work introduces a novel solution for a low-cost way to execute Phishing exercises. Having implemented the prototype, we offer fast deployment of Phishing infrastructure for legal assessments on self-controlled infrastructure. The solution is cloud-provider agnostic and does not enforce a specific cloud provider for the deployment of the Phishing infrastructure and lets the users decide. The comparison with other solutions shows that the prototype implements unique features not completely covered by other work. Our solution provides companies and individuals with a suitable alternative against costly third-party services while supporting data sovereignty. Additionally, Ansible offers easy extensibility, and the code is documented by design.

A qualitative survey was conducted, and the evaluation of the answers provided valuable insights into the participants' views. The opinions provided by the IT Security professionals are a helpful contribution when considering to open source the prototype. The final decision about releasing the prototype to the public will be made in the future. With the release, we intend to contribute to a better IT Security posture and help the IT Security community reach its goals of making the digital world a secure place.

10.1 Future Work

Future work could extend our solution by adding new deployment code for other open-source Phishing software. Possible prospective candidates could be Fiercephish [31] with Dovecot [17], and Evilginx2 [25]. With the addition of those tools, additional features would be available, and a broader tool choice for the users would be established.
Extending the Playbooks with Infrastructure as Code (IAC) solutions like Terraform could improve the workflow due to the possibility of automatically provisioning computing resources on cloud providers like Hetzner or Digital Ocean. Another improvement would be implementing a suitable solution to access the Gophish administration interface through the reverse proxy instead of relying on SSH port forwarding. Additionally, providing software like dnstwist [54] or URLCrazy [27] for homoglyph domain discovery could support the Phishing assessment.

Bibliography

[1] Robert M. Lee - (SANS);Michael J. Assante - (SANS);Tim Conway - (SANS). *Analysis of the CyberAttack on theUkrainian Power Grid*. Tech. rep. E-ISAC; SANS, 2016. URL: https://web.archive.org/web/20200726145246/https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf (visited on 03/06/2022).

[2] *Ansible Galaxy*. Red Hat, Inc. URL: https://galaxy.ansible.com (visited on 03/09/2022).

[3] *API Documentation - Version 3 - Beta*. Porkbun. URL: https://porkbun.com/api/json/v3/documentation (visited on 02/20/2022).

[4] Ayesha Arshad et al. "A Systematic Literature Review on Phishing and Anti-Phishing Techniques". In: *Pakistan J Engg & Tech 2021, 4, 163-168* (Apr. 2021). arXiv: 2104.01255 [cs.CR].

[5] Richard Barnes et al. *Automatic Certificate Management Environment (ACME)*. RFC 8555. Mar. 2019. DOI: 10.17487/RFC8555. URL: https://www.rfc-editor.org/info/rfc8555.

[6] boh. *Github - terraform-phishing*. URL: https://github.com/boh/terraform-phishing (visited on 03/07/2022).

[7] Russell Brandom. *$1.7 million in NFTs stolen in apparent phishing attack on OpenSea users*. The Verge. URL: https://www.theverge.com/2022/2/20/22943228/opensea-phishing-hack-smart-contract-bug-stolen-nft (visited on 03/06/2022).

[8] *CheckTLS website*. SecurEmail, LLC. URL: https://www.checktls.com (visited on 02/27/2022).

[9] *Chef software DevOps Automation Solutions: Chef*. Chef. URL: https://www.chef.io/ (visited on 04/02/2022).

[10] *Configuration 0: no secure mail server - mail-tester.com result*. mail-tester.com. URL: https://web.archive.org/web/20220329001814/https://www.mail-tester.com/test-4wngnpvmy (visited on 03/29/2022).

[11] *Configuration 1: mail server with TLS - mail-tester.com result*. mail-tester.com. URL: https://web.archive.org/web/20220328223312/https://www.mail-tester.com/test-232rj3s8x (visited on 03/29/2022).

[12] *Configuration 2: Secure mail server configuration test with mail-tester.com*. mail-tester.com. URL: https://web.archive.org/web/20220326010116/https://www.mail-tester.com/test-1s808liga (visited on 03/26/2022).

[13] Adam M. Costello. *Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)*. RFC 3492. Mar. 2003. DOI: 10.17487/RFC3492. URL: https://www.rfc-editor.org/info/rfc3492.

[14] *dehydrated*. Lukas Schauer, dehydrated. URL: https://dehydrated.io/ (visited on 04/19/2022).

[15] *digicert*. digicert. URL: https://www.digicert.com/ (visited on 04/22/2022).

[16] *DKIM record checker*. DMARCLY. URL: https://dmarcly.com/tools/dkim-record-checker (visited on 03/28/2022).

[17] *Dovecot*. Dovecot. URL: https://www.dovecot.org/ (visited on 04/24/2022).

[18] Piotr Duszyński. *Modlishka*. URL: https://github.com/drk1wi/Modlishka (visited on 04/09/2022).

[19] Amnon H. Eden. "Three Paradigms of Computer Science". In: *Minds Mach.* 17.2 (2007), pp. 135–167. DOI: 10.1007/s11023-007-9060-8.

[20] Thomas Fenzl and Philipp Mayring. "QCAmap: eine interaktive Webapplikation für Qualitative Inhaltsanalyse". In: *Zeitschrift für Soziologie der Erziehung und Sozialisation* 37 (Aug. 2017), pp. 333–339. URL: https://www.qcamap.org/ui/en/home.

[21] Matheesha Fernando and Nalin Asanka Gamagedara Arachchilage. "Why Johnny can't rely on anti-phishing educational interventions to protect himself against contemporary phishing attacks?" In: *Australasian Conference on Information Systems, Perth, 2019* (Apr. 2020). arXiv: 2004.13262 [cs.CR].

[22] Ana Ferreira and Gabriele Lenzini. "An analysis of social engineering principles in effective phishing". In: Verona, Italy. Verona, Italy: IEEE, 2015, pp. 9–16. ISBN: 978-1-5090-0178-1. DOI: 10.1109/STAST.2015.10.

[23] *FIDO Alliance Specifications Overview*. FIDO Alliance. URL: https://fidoalliance.org/specifications/ (visited on 03/28/2022).

[24] *Github Mirror - 2022 State of the Phish*. proofpoint. URL: https://github.com/jacobdjwilson/awesome-annual-security-reports/blob/main/Annual%20Security%20Reports/2022/Proofpoint-State-of-the-Phish-2022.pdf (visited on 03/26/2022).

[25] Kuba Gretzky. *evilginx2*. URL: https://github.com/kgretzky/evilginx2.

[26] E.G. Guba and Y.S. Lincoln. "Competing Paradigms in Qualitative Research". In: *Handbook of Qualitative Research, SAGE, Thousand Oaks* (1994), pp. 105–117.

[27] Andrew Horton. *URLCrazy*. URL: https://github.com/urbanadventurer/urlcrazy.

[28] *How FIDO Addresses a Full Range of Use Cases*. Tech. rep. FIDO Alliance, 2022. URL: https://media.fidoalliance.org/wp-content/uploads/2022/03/How-FIDO-Addresses-a-Full-Range-of-Use-Cases.pdf.

[29] *Inductive category formation*. qcamap.org. URL: https://www.qcamap.org/ui/assets/tutorials/en/Steps_Rules_Inductive.pdf (visited on 04/14/2022).

[30] *Install WSL*. Microsoft. URL: https://docs.microsoft.com/en-us/windows/wsl/install (visited on 03/21/2022).

[31] Chris King. *FiercePhish*. URL: https://github.com/Raikia/FiercePhish (visited on 04/21/2022).

[32] Scott Kitterman. *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1*. RFC 7208. Apr. 2014. DOI: 10.17487/RFC7208. URL: https://www.rfc-editor.org/info/rfc7208.

[33] Murray Kucherawy, Dave Crocker, and Tony Hansen. *DomainKeys Identified Mail (DKIM) Signatures*. RFC 6376. Sept. 2011. DOI: 10.17487/RFC6376. URL: https://www.rfc-editor.org/info/rfc6376.

[34] Murray Kucherawy and Elizabeth Zwicky. *Domain-based Message Authentication, Reporting, and Conformance (DMARC)*. RFC 7489. Mar. 2015. DOI: 10.17487/RFC7489. URL: https://www.rfc-editor.org/info/rfc7489.

[35] *Let's Encrypt*. Let's Encrypt. URL: https://letsencrypt.org/ (visited on 04/22/2022).

[36] Philippe Lieser. *DKIM Verifier*. URL: https://github.com/lieser/dkim_verifier (visited on 04/03/2022).

[37] *LimeSurvey: An Open Source survey tool*. LimeSurvey GmbH, Hamburg, Germany. URL: http://www.limesurvey.org (visited on 03/26/2022).

[38] *Mail (MX) Server Survey*. E-Soft Inc. URL: http://www.securityspace.com/s_survey/data/man.202201/mxsurvey.html (visited on 03/31/2022).

[39] *Mailserver - A simple, yet complete mailserver setup guide.* SimpoLab. URL: https://github.com/SimpoLab/mailserver (visited on 02/23/2022).

[40] Philipp Mayring. *Qualitative content analysis - theoretical foundation, basic procedures and software solution*. Jan. 2014.

[41] Microsoft. *The 2021 Microsoft Digital Defense Report*. Tech. rep. Microsoft, 2021. URL: https://info.microsoft.com/ww-landing-Microsoft-Digital-Defense-Report-Gate.html?LCID=EN-USdigital-defense-report&rtc=1.

[42] *MX Toolbox*. MXToolBox Inc. URL: https://mxtoolbox.com (visited on 04/28/2022).

[43] *Newsletters spam test by mail-tester.com*. WOOBEO. URL: https://www.mail-tester.com/ (visited on 04/28/2022).

[44] Markus Opolka. *Dockerhub - LimeSurvey Docker*. URL: https://hub.docker.com/r/martialblog/limesurvey/ (visited on 03/26/2022).

[45] Markus Opolka. *Github - LimeSurvey Docker*. URL: https://github.com/martialblog/docker-limesurvey (visited on 03/26/2022).

[46] Markus Opolka. *Website of Markus Opolka*. URL: https://www.martialblog.de/ (visited on 03/26/2022).

[47] Justin Perdok. *Github - ansible-playbook-gophish*. URL: https://github.com/justin-p/ansible-playbook-gophish (visited on 03/07/2022).

[48] Justin Perdok. *Github - ansible-role-gophish*. URL: https://github.com/justin-p/ansible-role-gophish (visited on 03/07/2022).

[49] *Podman*. containers. URL: https://podman.io/ (visited on 03/26/2022).

[50] proofpoint. *2022 State of the Phish*. Tech. rep. proofpoint, 2022. URL: https://www.proofpoint.com/us/resources/threat-reports/state-of-phish.

[51] Puppet. *Powerful Infrastructure Automation and delivery: Puppet*. Puppet. URL: https://puppet.com/ (visited on 04/02/2022).

[52] *Spam and phishing in 2021*. Kaspersky. URL: https://securelist.com/spam-and-phishing-in-2021/105713/ (visited on 04/22/2022).

[53] Twitter. *An update on our security incident*. URL: https://blog.twitter.com/en_us/topics/company/2020/an-update-on-our-security-incident (visited on 03/06/2022).

[54] Marcin Ulikowski. *dnstwist*. URL: https://github.com/elceef/dnstwist (visited on 04/29/2022).

[55] John Wadleigh. *Ansible 101 - Standards*. URL: https://www.ansiblejunky.com/blog/ansible-101-standards/ (visited on 02/26/2022).

[56] Rick Wash. "How Experts Detect Phishing Scam Emails". In: *Proc. ACM Hum. Comput. Interact.* 4.CSCW2 (2020), 160:1–160:28. DOI: 10.1145/3415231.

[57] ralphte; wikijm. *Github - Build_A_Phish*. URL: https://github.com/ralphte/build%5C_a%5C_phish (visited on 03/07/2022).

[58] *Windows Subsystem for Linux Documentation*. Microsoft. URL: https://docs.microsoft.com/en-us/windows/wsl/ (visited on 03/21/2022).

[59] April C. Wright. "Orange Is The New Purple". In: *BlackHat USA* (2017). URL: https://www.blackhat.com/docs/us-17/wednesday/us-17-Wright-Orange-Is-The-New-Purple-wp.pdf.

[60] Jordan Wright. *Gophish*. Gophish. URL: https://getgophish.com (visited on 03/06/2022).

[61] Jordan Wright. *Gophish - Github*. Gophish. URL: https://github.com/gophish/gophish (visited on 03/06/2022).

[62] *ZeroSSL*. ZeroSSL. URL: https://zerossl.com/ (visited on 04/22/2022).

List of Figures

List of Tables

List of source codes

List of Abbreviations

OSINT	Open Source Intelligence
FOSS	Free and Open Source Software
TLS	Transport Layer Security
VM	Virtual Machine
MTA	Mail Transfer Agent
NFTs	Non-fungible tokens
API	Application Programming Interface
2FA	two-factor authentication
MITM	machine-in-the-middle
WSL	Windows Subsystem for Linux
OS	Operating System
OPSEC	Operations Security
IAC	Infrastructure as Code
SPF	Sender Policy Framework
DKIM	DomainKeys Identified Mail
DMARC	Domain-based Message Authentication, Reporting & Conformance
GDPR	General Data Protection Regulation
FQCN	Fully Qualified Collection Name
CSRF	Cross-Site Request Forgery
CERT	Computer Emergency Response Team
RFC	Request for Comments
ACME	Automatic Certificate Management Environment
CA	Certificate Authority

A Git commit of the technical Implementation

The code of the prototype was versioned with Git. The latest Git commit at the time of submitting this thesis for grading was:
`c785f841f50f42415aa0567f0f1a7abeee3cc528`

The code is hosted on the private repository https://github.com/Hetti/Phishing-Oida on Github. Access to the repository was granted to the supervisors. The repository may be made public in the future as discussed in chapter 10 on page 66.

B Questionnaire

Info:
○ → This field is part of a single choice question
□ → This field is part of a multiple choice question (minimum 1 choice must be checked)
FREETEXT → Free text field

It would be awesome, if you could give me some information about your background:

Optional questions You can also partly answer them or skip it completely! If you want to skip - just click the Next button at the bottom.

From which source did you get this survey?

□ Twitter

□ LinkedIn

□ Directly from a person

□ Mail

□ Other Source

How many years of professional experience do you have in the IT field?

□ < 1

□ 1-2

□ 2+

□ 5+

□ 10+

In which IT Security team do you currently belong?
Please choose the best fitting one with that you identify.
For further details check out: https://www.blackhat.com/docs/us-17/wednesday/us-17-Wright-Orange-Is-The-New-Purple-wp.pdf (link opens a new tab!)

☐ Red Team - Offensive Security

☐ Blue Team - Defensive Security

☐ Purple Team - Defensive and Offensive Security

☐ None of the available choices

How many years of professional experience to you have in the IT Security field?

☐ < 1

☐ 1-2

☐ 2+

☐ 5+

☐ 10+

In which areas do you have experience?

☐ IT Management

☐ System Administration

☐ Incident-Response

☐ IT Security Consulting

☐ Building IT Architecture

☐ Software Development

☐ Blue-Team

☐ Academic Research

☐ Red-Team/Pentesting

☐ None of the available choices

Deployment code for Phishing infrastructure & Ethics

Is it in your opinion ethical to open-source deployment code for Phishing infrastructure?

○ Yes

○ No

Attention: The **answers** for this free text question **will be published** in my master theses.
Personal Information (Name,Surname,Email) will **not** be published.
Please do not add any personals information to the answer. Thank you.

If the choice is ✓Yes:
Optional free text question
Why is it in your opinion ethical to publish deployment code for automated Phishing infrastructure?
Please elaborate your opinion:

FREETEXT

If the choice is ⊘No:
Optional free text question
Why is it in your opinion not ethical to publish deployment code for Phishing infrastructure?
Please elaborate your opinion:

FREETEXT

Evaluation of the Tool

Would you be interested in evaluating such a tool yourself to improve awareness about phishing?

○ Yes

○ No

If the choice is ✓Yes:

In what way would the deployment script be helpful for you?

☐ I could send it to colleagues, which would probably use it for awareness training

☐ I could send it to colleagues, which would probably use it for offensive assessments

☐ It would enable me to add Phishing Assessments to my offer of security services

☐ It would help me improve my Ansible skills

☐ I do offensive security, and it could save me time for setting up a Phishing campaign

☐ I do defensive security, and it could save me time for setting up a Phishing training

☐ None of the available choices

If the choice is ⊘No:

Would you be interested in evaluating such a tool yourself to improve awareness about phishing?

☐ I am already using my tooling (self-scripted) and would not like to change this

☐ I (want to) pay external companies or consultants to do the Phishing Assessments

☐ I don't have time to conduct internal Phishing Assessments

☐ I am not familiar with Ansible and do not have the time or interest to learn it

☐ I am already using open-source software for this use-case

☐ I am already using commercial software for this use-case

☐ None of the available choices

Open Source?!

If you would implement this Ansible Playbook - would you publish it under an open-source license (e.g., MIT)?

○ Yes

○ No

If the choice is ✓Yes:
What would be your motivation to open-source your solution to the public?

□ I would contribute to improving IT Security in general

□ Other people could contribute and implement new features

□ It would be beneficial for my project portfolio, and I would mention it in my CV

□ In my opinion, open-sourcing such software is the way to go because I believe in this concept

□ I would give something back to the community

□ The community could take over the maintenance of the project

□ None of the available choices

If the choice is ⊘No:
What would be your motivation not to open-source your solution to the public?

□ It would not be ethical to open source it because criminals or "Script Kiddies" could use it for illegal activities

□ I fear negative publicity because of the project

□ In my opinion, proprietary software is the better solution

□ I would not like to share my work with everyone

□ I have no time to invest in the maintenance or management of the project

□ The tool should be only available to a limited circle of people (e.g., security professionals)

□ It may violate my company policy or my working contract

□ None of the available choices

C Free-text Answers

C.1 Opinions which support that it is ethical to open-source deployment code

1. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: A grey/black market already exists, so open source will not have an impact on criminal activity.

2. **[cannot be meaningfully categorized]**: It lowers the barrier for white harp his hinge awareness campaigns

3. **[Public Tools create awareness and motivation to tackle the threats/problems]**: IMHO, the focus should lay on making the whole mail stack secure albeit the existence of such automated phishing stacks. Their existence should not be a problem, but more a hint that the current mail situation is not mature enough to survive such a simple “attack”.

4. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Pisses off the blue teams ;) Also something about helping out people who want to do good without worrying about helping those who want to do bad as they’ve already got way better tooling anyways.

5. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: This allows more teams to set up example infrastructure for trainings.

6. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: I expect the impact of such tools on real phishing operations (which probably have the funds to buy software anyway) to be minimal; they however seem useful for academic and red-teaming purposes. My answer might change however if the tools are specifically tailored to the needs of large phishing operations instead of being focused on proof-of-concept or small-scale use.

7. **[Information should be free/public for everyone]**: It’s more ethical to allow everyone to have access to it, than to charge for it to only certain parties. The tool could be built by anyone, and the practice of phishing isn’t an unthinkable capability, therefore it will come into existence if it isn’t already. By constructing and providing the tool it may be used for negative reasons, but it can also benefit society positively. The use which the tool is put to would be the primary ethical or unethical act.

8. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: I assume that the purpose of this infrastructure is to help blue teams and awareness teams to conduct phishing resistance training and education more easily. Otherwise, i.e. if the purpose is to help the phishers, then my answer would be 'unethical' If the purpose if the code is to help in anti-phishing training, then I can justify this because the bad guys know already how to do this and don't need help from this.

9. **[People/companies can learn from the published software and therefore protect themselves]**: Everything, that van be thought or invented, will be thought or invented. It is no prevention, to not publish it. On the other hand, people can learn from published sources, how to protect themselves and their infrastructure.

10. **[People/companies can learn from the published software and therefore protect themselves]**: Understanding an deconstructing any type of sourcecode offers a learning opportunity possibly vital for detection and defense.

11. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: There already exists quite a bit of infrastructure and publishing basetooling is nothing that reduce barrier of entry. More complicated obfuscation tooling or similiar is unethical since it lowers the barrier quite a bit with little blue team benefit.

12. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Because it helps organizations prepare for the event of a phishing attack without building infrastructure themselves. Attackers already have (some) infrastructure (or "prize" building it into their scam), but it may be out of budget for legitimate operations.

13. **[Software is Dual Use - Responsible is the user]**: It's not about the tools. It's about who is using them, how, and why.

14. **[People/companies can learn from the published software and therefore protect themselves]**: the benefits and knowledge provided by open-sourcing code outweighs the risk

15. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: So we can easier train people to overcome phishing attacks.

16. **[Software is Dual Use - Responsible is the user]**: As with any ethically ambiguous tools, it's a matter of how the tool is used that matters. In this case, there are certain reasons someone could ethically use such a tool - for, say, demonstrations of viability of phishing emails. Or to study methods of countering such a tool.

17. **[People/companies can learn from the published software and therefore protect themselves]**: Open source attacker infrastructure helps protect against attacks by making it easier to understand how attacks work. From the red team perspective, it saves significantly on setup time as well, leading to more effective exercises.

18. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: Phishing simulations are in my opinion a big part of the awareness training for employees. And there are reasons to refrain from using (professional) phising simulation services, as they aren't flexible enough or don't provide enough anonymization. An automated deployed phishing infrastructure can help to increase awareness of phishing and make it easier for a company to roll their own phishing simulation.

19. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: Training for users

20. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: A security team should be able to perform phishing campaigns for educational purposes. Also called "awareness training". There is alternatives to OSS like Lucy for such awareness trainings.

21. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Building thise tooles is not so hart, that it would stop that many non ethical hacers from doing evel shit. But ethical hackers have to code thise same tooles over and over again in order to show copanys this atac vector. I think that making it open source would mace the world a saver place in the end.

22. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: Making it easier to test, will lead to more testing. More testing, will lead to improvements in awareness over time.

23. **[Security by obscurity is not a sufficient basis for security]**: the methods are available anyway, the tools make it easier for security researchers to test for vulnerabilities. security by obscurity is not a good strategy

24. **[Software is Dual Use - Responsible is the user]**: Man kann auch Pistolen legal kaufen. Entscheidend ist wie sie eingesetzt werden.

25. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: It can help people red-team their own org/train their staff. red teams often do not have many resources (time and money), while criminal attackers have heir marketplaces, crimeware-as-a-service etc and wouldn't profit much.

26. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: The best method for training people for phishing attacks is showing examples. Open-Sourcing not only allows IT-Sec teams of companies to easily provide a live example for

training but also makes many other people aware of a specific form of phishing. The more "ordinary" people know about the different phishing methods, the better - thus more publicity is imo desirable.

27. **[Community can check code, fix bugs, extend features and it can be a basis for future work or research]**: The tools exist anyways, but having them easily available can improve research. Another similar case would be metasploit.
28. **[Information should be free/public for everyone]**: Knowledge is power
29. **[Community can check code, fix bugs, extend features and it can be a basis for future work or research]**: Vulnerabilities in the code and quality of the code can be identified.
30. **[People/companies can learn from the published software and therefore protect themselves]**: Only the availability of such tools in open source form ensures that it can be studied, implying that proper countermeasures can be developed in an effective way as well.
31. **[People/companies can learn from the published software and therefore protect themselves]**: By realeasing the source code, it will make defending against these attacks easier.
32. **[Community can check code, fix bugs, extend features and it can be a basis for future work or research]**: To expand ideas for further technologies that build on this idea To create awareness, how and what happens behind the curtain
33. **[People/companies can learn from the published software and therefore protect themselves]**: Like other but similar tools/systems it is ok to publish such a system as example/experimentation platform. It could be used to demostrate real world attacks to people not familiar with pishing attacks much more easily. This educational purpose weights, in my opinion, much more than possible abuse.
34. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: There are tons of actors running attacks for many years, so imho publishing as open source doesn't add much to the threat but makes it easy for it people to set it up as part of an internal awareness campaign etc.
35. **[People/companies can learn from the published software and therefore protect themselves]**: To make it easier to learn and to test
36. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: Such tools can help IT security teams running fake phishing campaigns against their own employees as part of anti-phishing awareness trainings.

37. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: There are plenty of tools and even services already available for criminal use. One additional tool wouldn't make a difference. A well-documented tool would be useful in training for various roles. Pentesters may use it to confront end users, administrators and developers with harmless attacks. The technical roles may also gain insight into the mechanisms of phishing tool, learn about their shortcomings and how to counter them.

38. **[Community can check code, fix bugs, extend features and it can be a basis for future work or research]**: Security researches needs access to this tools too.

39. **[People/companies can learn from the published software and therefore protect themselves]**: if it‘s secret nobody knows how to defend against such campaigns

40. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: To able every single Small-Medium Size company leveraging it which finally contributes to have a more secure society.

41. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: To help companies set up their own scenarios

42. **[Community can check code, fix bugs, extend features and it can be a basis for future work or research]**: The more code is publicly available, the better systems like EDR or behavioural analysing software (may them be with AI or without) can detect such code.

43. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Against the publication one could argue that it is made easier for malicious attackers or phishers if such software is open source and freely accessible. However, phishing attacks are not provoked by such software. Attackers who are interested in carrying out malicious phishing attacks are already doing so. The release of the software can help to conduct professional phishing campaigns in companies that either do not have the money or do not have the time to build such a campaign "from scratch". Thus, I think that a release can benefit the security of companies while having no negative effect.

44. **[Community can check code, fix bugs, extend features and it can be a basis for future work or research]**: This will allow more research information on this subject and help community to learn from others.

45. **[The concept is nothing new or special and the complexity for implementing the solution it not that high]**: Enables realistic testing but doesn't provide resources or techniques that aren't currently available.

46. **[Software is Dual Use - Responsible is the user]**: Vulnerabilities for which no poc code exist are often not taken seriously. There must be easy proof that attack scenarios work and are feasible. Deployment code is only the tooling. It can be used for the good and the bad. Like a hammer, a knife or nmap.

47. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: The attackers already have the means. These kind of tools generally tend to help out the blue team more with tools they do not have the skill or time to create themselves.

48. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: There are already resources available and this won't make it easier for the bad guys, it will help the good guys as well.

49. **[People/companies can learn from the published software and therefore protect themselves]**: Making it available might help people to protect themselves and learn about it. Therefore it might be useful. It's not ethical to use those for attacks.

50. **[Security by obscurity is not a sufficient basis for security]**: Keeping code secret is not a sufficient basis for security.

51. **[Software is Dual Use - Responsible is the user]**: There is nothing inherently wrong or unethical in a tool.

52. **[Automation can not replicate good Phishing tactics and procedures]**: While I do t like making actual bad ppl's lives easier, there are lots of things about good phishes that automated infrax can't replicate, like domain reputation, sliding into an existing email chain, etc.

53. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: A. It already exists. b. Malicious actors are capable of building their own. If we give them a way to do it with they can be caught when they get lazy. C. If tools are released publicly, iocs can be built. Ex gophish, evilgnix2.. Etc D. By releasing tools publicly, Blue teams/researchers can leverage these automation tools to map out and identify threats before an attack happens. E. Red teams/Pentesters only have so much time. When working with a client, often times there is less required concern about getting caught but about what can be learned from the excersise. Automation allows for these style excersises to be more consistent and available for clients who don't want a longer drawn out excersise.

54. **[Information should be free/public for everyone]**: I just go with the hacker ethics: 1. "Access to computers—and anything which might teach you something about the way the world works—should be unlimited and total. Always yield to the Hands-On Imperative!" 2. "All information should be free" 3. "Mistrust authority—promote decentralization" 4.

"Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, sex, or position" 5. "You can create art and beauty on a computer" 6. "Computers can change your life for the better" and the additions from the german CCC 7. Don't search in other peoples data 8. Use Public data and protect private data. Under these ethics it is ethical to publish this

55. **[People/companies can learn from the published software and therefore protect themselves]**: Even if not published, such things WILL Be (or likely IS) available for people and groups operating in this field in some professional way. Something like this can also be used for learning how this can be done (using for other purposes to) and maybe even for finding ways to protect against. So publishing something like this is not going to make the current situation worse.

56. **[Public Tools create awareness and motivation to tackle the threats/problems]**: Phishing doesn't disappear without open source tools. The open source tool can be used for awareness. Maybe the world moves on from ancient and vulnerable email-based comms sooner if everyone and their grandma are able to send phishing mails.

57. **[Information should be free/public for everyone]**: Not only information wants to be free, also code! As people are probably more confronted with this topic when there is a (open) source available people have to evolve more regarding this topic. Also this would increase the possibility that not only criminals (who are using such infra anyway) are using such infrastructure but also good willing people. That also increases the possibility that people learn how to prevent problems that come with phishing before they are seriously attacked.

58. **[People/companies can learn from the published software and therefore protect themselves]**: Other people can learn from it, and especially learn from what it would leverage and possibly try to abuse, so you know what to defend against. A good part of the work is likely not specific to phishing itself from an infra perspective.

59. **[Software is Dual Use - Responsible is the user]**: All offensive itsec tools can be considered dual use. Most offensive organisations either developed automation tooling already or are procuring it.

60. **[People/companies can learn from the published software and therefore protect themselves]**: Making the steps of conducting such activity visible for the defenders and system administrators, as especially when it comes to email infrastructure, these things seem to be a total black box for many.

61. **[Commercial tools already exists, therefore, open-source would be an alternative]**: There are already commercial phishing tools out there to test the employees (or other targets which have consented, more or less) responses to such a threat, an open source tool for this would be pretty amazing. In my opinion it is just as ethical as publishing code

for exploiting a security vulnerability which has already been disclosed a long time ago, except you are exploiting the biggest vulnerability in the whole world: humans.

62. **[In favor of all activities that stop hackers and improves resilience of business]**: I am for all activities to stop hacker and any attacks against professionell business.
63. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Criminals already have automated infrastructure available and you might as well give options to security teams or managed security service providers that cost less (as in license).
64. **[Phishing Tools are already available, Open Sourcing does not change anything and it is also possible that someone else will make it]**: Much like Evilginx, eventually, someone else will make it. That said, I think there is an obligation to add anti-script kiddie measures to the source code (as Evilginx does)
65. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: helps building awareness (like lock picking instructions)
66. **[Public Tools create awareness and motivation to tackle the threats/problems]**: Public availability of tools generates awareness and willingness to counter threats which are cheaper to be downplayed otherwise.
67. **[People/companies can learn from the published software and therefore protect themselves]**: Each one teach one. Adversaries may find other options for their campaign anyway.
68. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: I consider open-source development of Phishing infrastructure ethical mostly because test-phishing is one of the best and most important ways to raise awareness for phishing and help educate and protect against phishing. Making the tools necessary for this available to the general public and easing access to phishing tool helps to make this easier and better. Attackers will phish either way, but your average blue team might not, if the tools are not there and there is not enough time to develop such tools.
69. **[People/companies can learn from the published software and therefore protect themselves]**: to show which methods are used in this deployment, and to act against it, and to secure your system, which might be invulnerable to this
70. **[The concept is nothing new or special and the complexity for implementing the solution it not that high]**: easy enoughto redofor anyone medium skilled
71. **[We need to educate our employees against such threats]**: We should play with open cards. Yes, phishing is a big problem. Yes, we have to train our employees well and be prepared as best as possible for such attacks

72. **[Software is Dual Use - Responsible is the user]**: Because as long as you don't directly harm someone, it's good to bring knowledge to a broader field. Also, just by pushing a Phishing infrastructure doesn't mean that a reader can use it to produce harm, since he has to have already knowledge in this field to do so and if he has the necessary knowledge, he could also produce that kind of harm without the Phishing infrastructure (even toe it might take him a little bit longer)
73. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: It always helps blue teams with not much experience in this field, to have tools that are ready to go. To train the people in the own organization. We have to assume, that black hats always have the capability to build something like this themselves. But it does not help, that not all IT organizations know how to train people against phishing. So, the more information we have about it, the more people we can train. This helps to broaden the awareness overall and makes phishing harder.
74. **[People/companies can learn from the published software and therefore protect themselves]**: Share and grow should be the motto, especially in the area of IT security. In my opinion, the publication can contribute to the fact that this can be used for tests of the own infrastructure and thus vulnerabilities can be detected and fixed. It is ethically justifiable as long as no harm can come to an individual person and it is only exploited for internal vulnerability detection.
75. **[Phishing Tools are already available, Open Sourcing does not change anything and it is also possible that someone else will make it]**: Phishing tools are available either way. Open Sourcing it doesn't change that. There is however an argument to be made for providing easier access to such tools, but I think all in all it is still ethical
76. **[Will benefit for defenders more than criminals]**: Any suggestion, whether implied or overt, that the open availability of tools will benefit criminals more than other parties is incorrect. This is not only based on common sense, but on countless examples going back decades.
77. **[People/companies can learn from the published software and therefore protect themselves]**: It can be used for internal awareness trainings. We have to understand the tools of the other side, so we can just as well develop it together. It lowers the barrier to entry, but we need working strategies againt those low-effort actors either way.
78. **[Software is Dual Use - Responsible is the user]**: Because source code doesn't harm anything and to prevent such attack it is nessesary to analyze the code base.
79. **[The concept is nothing new or special and the complexity for implementing the solution it not that high]**: It's not exactly rocket science.
80. **[People/companies can learn from the published software and therefore protect themselves]**: Red team source code should be open source to elaborate a standard and

enable a deep understanding of what the tool tries to do. Open Source also eases to adapt the code to own special needs.

81. **[People/companies can learn from the published software and therefore protect themselves]**: Although in a grey area, it is ethically okay to publish such software for purposes of education. An open source tool which can be understood can perhaps help to defend against phishing attacks.

82. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: By publishing the tool for the world to see, you give security professionals doing security assessments the opportunity to use this method to test their systems.

83. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Not having open source code for phishing hasn't stopped anyone from sending phishing emails, it just stops defenders and pentesters from improving their methods.

84. **[The concept is nothing new or special and the complexity for implementing the solution it not that high]**: I do not (yet?) see what is special about phising infrastructure in particular.

85. **[Public Tools create awareness and motivation to tackle the threats/problems]**: The only way to force software-vendors to better protect against phishing attacks is to make these attacks abundant by making them even lower cost. Unlike other attacker-defender games, phishing is a game we can actually win through e.g. WebAuthn.

86. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Any marginally sophisticated attacking group already has this internally. By releasing it, you enable companies to assess the current risk level to phishing, security awareness training programs and the ability to do valuable demos to executive leadership. Could it be used for evil? Sure. As can a screwdriver or hammer. If this was automating some novel and devastating attack path, I may feel different. However, because it is now old and devastating, anything to level the playing field is of value.

87. **[Phishing Tools are already available, Open Sourcing does not change anything and it is also possible that someone else will make it]**: If you don't someone else will do it in the dark. It's better if a community, small or big, can monitor and learn from the code how to recognize and stop phishing. From my perspective the benefits are greather than the risks. "There si no better disinfectant than sun light"

88. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Helping reasearch and security training for phising its good to have code publicly shared probably outweighs negative impact on a bad actor using it - as bad actors with resources could do this and more anyway

89. **[People/companies can learn from the published software and therefore protect themselves]**: There is still legitmate use for tooling like this, and it lso helps understand the functional taks performed by phishers, which can lead to better security practices.

90. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: professional scams would already have a process for this and are done without the publishing of this tool small scams are also already commonplace and will likely not bother with it I would reccon the benefit is far greater for not all to well versed IT admins trying to size up the knowledge of ther personel to identify phishing mails. I would sugges adding a line in the licence for the published code regulating the usage for cases where it is allowed for tests - while legally not binding or not easy to track, it could function as a moral anchor. (Dynamite can be used as a tool and for weapons to reference Nobel)

91. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Automated deployment of individual components such as mail servers including secure configuration and creation of phishing campaigns already exists. E.g. https://mailinabox.email/. An Ansible playbook would only be a combination of these. Phishing campaigns differ only in content and intent from newsletter/mail campaigns. The fact that existing stuff can be used for abusive purposes cannot be prevented. In the specific case of phishing campaigns, the components already exist and only need to be combined. The mass of phishing mails sent out shows that attackers have no problems setting up a corresponding infrastructure themselves.

92. **[It should provide a limited infrastructure with a defined legitimate scope]**: The purpose of the provided tooling/software components has to be clearly limited for the setup of phishing infrastructure to perform phishing simulations with a defined scope whereby consent of the targeted entities (competent representatives) has to be obtained upfront. It must not be used to distribute any malicious content. It may collect information to serve the purpose of a phishing simulation but not any confidential data (e.g. passwords).

93. **[Information should be free/public for everyone]**: Open-Source is always good.

94. **[Security by obscurity is not a sufficient basis for security]**: Information needs to be freely available. If it's closed source it is only "security by obscurity".

95. **[Software is Dual Use - Responsible is the user]**: The code itself is not unethical - the malicious use of it would be. to publish it, it could also be used to study and train skills needed to tackle such infrastructure.

96. **[People/companies can learn from the published software and therefore protect themselves]**: Security can't rely on the hope that blackhats just don't know how to do stuff. We all have to understand how attacks work, simulate them and harden our infras-

tructure, and sometimes also our users against these attacks. Simulated phishing attacks could help to achieve this goal.

97. **[People/companies can learn from the published software and therefore protect themselves]**: As a defender you have to know, which tools are available and how the work. This tool will help cooperations to strengthen their defense.

98. **[Security by obscurity is not a sufficient basis for security]**: Aus dem gleichen Grund warum man Allgemein Red-Teaming Tools open sourcen sollte. Security by obscurity funktioniert nie. Und nur weils es ned öffentlich gibt heißt ned, dass es ned schon wer hat. Grad Phishing is ein Thema das es schon sehr ausgereift gibt. Open-Source (und somit quasi gratis - ausser die lizenz sagt es is lizensierpflichtig im businesscontext) kann hier nur der awareness/den verteidigern helfen, da das budget für komerzielle lösungen meist sowieso fehlt

99. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: For similar reasons you open-source any kind of "attacking" software, e.g.: * you won't prevent adversaries from attacking by not open-sourcing it * it allows defenders to conduct ethical tests (in that case something along the lines of awareness campaigns)

100. **[People/companies can learn from the published software and therefore protect themselves]**: to learn from it and improve your systems based on that... yes it can of course be misused, but if you really want to get such code you will get it anyway, also if it is not open source

101. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: In my opinion, this introduces the exact same dilemma as with any other dual use software (or tools in general). In order to determine whether this is ethical, I think we need to ask ourselves "what' the potential benefit?" and "what's the potential damage?" and then assess which of these has a greater weight. As far as I understand the situation, a huge potential benefit is that it becomes a lot easier (esp. for small/middle-sized enterprises with the proverbial single-person IT department) to provide training and mock-phishing for the employees, making them more resilient against "real" phishing attacks. On the other hand, I don't believe there is too much damage to be done; there already is a lot of phishing happening, which appears to already scale really well, so there most likely already is a lot of automation tooling available, reducing the relative negative impact of such a new solution. In addition, most mailservers abused for "real" phishing get blocklisted sooner or later anyway, no matter their configuration, so some additional defences are already in place. Another question we can ask ourselves is "where do we need to stop?": By the same arguments, we could argue that it's already unethical to publish the mailserver part of this solution, because it can be abused to send spam email.

102. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: In my opinion it is ethical because it is the same

discussion with automated exploit tools that are available such as metasploit. We need to have easy access to security tools so that we can prepare for attacks and reinforce the security of companies and people. The bad actors will always have access to such tools.

103. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: Yes it is, as it allows for building more realistic and scalable threat scenarios as a defender (e.g., for employee training). It also allows the defender to test their own system against a (probably up-to-date and peer-reviewed) realistic threat infrastructure.

104. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: In my opinion, the deployment helps to make phishing more accessible for users (companies, etc.) who would otherwise be deterred by conventional providers and possible pricing models.

105. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: Deployment isn't the hard part of starting a phishing campaign, but can on the other hand help a lot when used for educational purposes.

106. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: The information and the tools already exists. Currently they are available only to entities who have the time and money and motivation to put in a lot of work for a later payoff. These are mostly malicious entities as security defence never has the budget or time and clear-cut goal of an malicious attacker. In order to enable a corporate blue-team to be more effective, information about and examples of existing threats needs to be more accessible.

107. **[Software is Dual Use - Responsible is the user]**: The open-sourcing of dual-use code gives the defenders at least the opportunity to craft and test defences. Keeping it closed-source gives attackers and criminals an advantage that is a greater threat than the nuisance of skript kiddies getting access to phishing-made-easy.

108. **[Will benefit for defenders more than criminals]**: Phishing is already one of the most commonly used attack methods. Therefore, I think that the added value of an open source tool that can be used to run, simulate and rehearse phishing campaigns is greater than the damage that can be done if such a tool is misused by attackers.

109. **[Will benefit for defenders more than criminals]**: Unlike n-day exploits, it is just a "faster" way. If the (your) tool is really good, it might produce high-quality phishing mails, that might be abused by an attacker, nevertheless, the value of training a lot of people in an fast and inexpensive way can help far more people than it might theoretically be able to hurt.

110. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: Access to security tooling allows people (at least in principle) to test their own security

with nearly professional grade tools. Besides, the infrastructure is not the most important factor in a phishing campaign and it is a factor that "the bad guys" will manage either way.

111. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Such thing is not that hard for any bad actors, so not publishing wouldn't make much difference.

112. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: If there would have been a "I'm not sure about it" option, I'd have chosen that one. I chose yes because I think making it open source enables more blue teams to do a better job.

113. **[Software is Dual Use - Responsible is the user]**: In my opinion it is some kind of "dual use good/Technology" - a little bit like a knife, weapon etc. You can use it for benign and malicious purposes. Additionally i think that it is unrealistically to "ban" hacking tool - there will always be someone to publish such technologies. Other tools like Metasploit, nmap etc. are also available and enable hacking attacks

114. **[Software is Dual Use - Responsible is the user]**: Well, gophish is officially a tool to create awareness and eventually educate employees that CAN be used for malicious phishing. Is it ethical to build knives?

115. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Bad guys already have their tooling in place. They can afford to spend time on building a phishing infrastructure, because they make money off of their attacks. Good guys (e.g. internal security teams) on the other hand, could profit from an easy to deploy solution to improve their company's awareness.

116. **[Community can check code, fix bugs, extend features and it can be a basis for future work or research]**: It provides access to a big community, it means that the software quality improves. The more people are involved, more testing and bugs reporting will be carried out. Additionally other development can be added to extend functionalities and keep the software up to day. Phishing is a continuous activity, which will evolve, for this reason the software has to be regularly updated. A big disadvantage is that phishing generators will learn about the methods the software uses and they can use it for avoiding detection. At the end of the day, security should not be based on obscurity.

117. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Closed Source in IT-Security has so far never led to long lasting IT-Security. Exploring this field while publishing the results can in my opinion also lead to better defense mechanisms. Of course this approach leads to a residual risk, because public access to such code/scripts will result in harmful usage. To be realistic: in particular closed groups, there is such code probably already available (also there are already some open source tools) - and we yet cannot really defend against it.. Also, as far

as I know, such tools/toolboxes are widely available by commercial providers as SaaS. In this context, I do not see a large ethical risk to this approach.

118. **[Will benefit for defenders more than criminals]**: This seems to be a decision between enabling the study of and proliferating harmful technology. I hope that the harm will be greater than the damage.

119. **[People/companies can learn from the published software and therefore protect themselves]**: Because it helps defenders understand how phishing infra can be deployed and used by attackers. Security should never work by gatekeeping information (such as exploits or the like), because attackers will always find ways of gaining this kind of knowledge. By releasing it to the public the burden of understanding and defending oneself is lowered for everyone.

120. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: Publishing such code makes it easier to use it for security training lessons.

121. **[People/companies can learn from the published software and therefore protect themselves]**: It is useful to get insights into the behavior and procedures how a phishing attack works. Furthermore, it can be used to test the own software/infrastructure.

122. **[People/companies can learn from the published software and therefore protect themselves]**: Open sourcing the code makes sure it is available for anybody to use, inspect, improve and derive. This will be particularly important to students, small organizations and others that otherwise could not afford commercial solutions for doing their own security awareness training or internal phish-testing. (While it makes no difference whatsoever to any criminals.)

123. **[People/companies can learn from the published software and therefore protect themselves]**: If it isn't open source, it's even harder to resist or combat it.

124. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: OpenSource simply means that more (a wider group of) people get access, the black team will always have access in the deep-web. So the greater attention would give more opportunity to grow awareness and counter meassurements.

125. **[cannot be meaningfully categorized]**: The real question is, what would make it unethical? There is very little reason to ever not publish source code, particularly here.

126. **[Community can check code, fix bugs, extend features and it can be a basis for future work or research]**: for e.g. research

127. **[People/companies can learn from the published software and therefore protect themselves]**: independent of question but might be interesting for your work:

https://www.youtube.com/watch?v=VglCgoljztE Why is it in your opinion ethical to publish deployment code for automated Phishing infrastructure? –> bring out the knowledge –> helps defending

128. **[Information should be free/public for everyone]**: Open Source is always the ethical thing to do :-)

129. **[Community can check code, fix bugs, extend features and it can be a basis for future work or research]**: Because research can be done in that way

130. **[Software is Dual Use - Responsible is the user]**: It doesn't harm. A knife can be a weapon or used for buttering your bread. Open Source infrastructure can be used to practice and learn how phishing works or attack an organisation

131. **[Commercial tools already exists, therefore, open-source would be an alternative]**: Companies Like knowbe4 or phishlabs sell phishing tests as a service. Providing such infrastructure as open source allows orgs to roll their own phishing tests. I don't see a problem in doing an open source version of a commercial product.

132. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Most technologies are at least dual-use. Deployment code for phishing infrastructure is among them. It may decrease the effort for malicious actors, but they can resort to Crime as a Service infrastructure anyway. It decreases effort for red teams, researchers and other benevolent actors, which is good.

133. **[Community can check code, fix bugs, extend features and it can be a basis for future work or research]**: Enables open community efforts. Facilitates research, learning/education, awarness, insight and enables everyone to participate whatever their intereset/motivation

134. **[Security by obscurity is not a sufficient basis for security]**: Attackers will attack anyways - with or without this open-sourced code. While it may become easier for attackers, it also becomes easier for "defenders" to analyze the tactics and how to avoid them. Security through obscurity never works!

135. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Good open source tooling is key to understand attack surfaces better. In this case you are providing a platform that would help teams with lesser knowledge to simulate phishing against their users, which is better than being attacked by the bad guys. In the end the bad guys will phish anyway if you pubish or not. Unethical would be to publish an tool or attack vector befor it could be closed and you would knowingly harm/risk others. But phishing is there for ages and will continue to be a problem.

136. **[It should provide a limited infrastructure with a defined legitimate scope]**: If the infrastructure is simply for deploying phishing pages for testing purposes, that's fine. However, I think there should be some way to differentiate from an actual login page to discourage its use in actual phishing (perhaps a password, token, or watermark). You might also want to make sure that information from the pages cannot be logged and only records success rate.

137. **[Public Tools create awareness and motivation to tackle the threats/problems]**: I believe it increases public awareness of phishing methodology.

138. **[Community can check code, fix bugs, extend features and it can be a basis for future work or research]**: Its important that people know what the code is running.

139. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: It furthers the capability of those that can use the tools for good - for simulation and growth. There are plenty of existing toolkits attackers can use already if need be, the risk presented by adding to the pile is minimal.

140. **[Public Tools create awareness and motivation to tackle the threats/problems]**: Phishing is such a large-scale Problem already that we need better defenses anyway, increasing the pressure through easily-available tools may be beneficial in the long run

141. **[Public Tools create awareness and motivation to tackle the threats/problems]**: Publicly accessible tools are the foundation of improving the cyber security level.

142. **[cannot be meaningfully categorized]**: why not?

143. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Because the bad guys already have their own tooling, and this helps organizations conduct internal exercieses.

144. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Bad actors are surely using that same sort of automation. Having the ability to spin up comparable environments for internal training will surely be beneficial.

145. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: to ensure a positive experience and develop a model of employee trust and supportive engagement

146. **[Will benefit for defenders more than criminals]**: Defenders using these tools is a larger benefit than e.g. attacker also using it. Commercial tools exist as do undergroud tools

147. **[People/companies can learn from the published software and therefore protect themselves]**: For training and educational purposes

148. **[People/companies can learn from the published software and therefore protect themselves]**: Publishing code helps understanding tactics & procedures from a phisher's point of view and therefore helps building defenses against phishing.

149. **[Information should be free/public for everyone]**: 1) the more information exits, the better teams can prepare e.g. tools that got more public attention also resulted in a lot of defensive strategies 2) the tools are already out there so it's „just a collection" that can pretty well be used as a benchmark of sorts 3) as always all information should be free as longbow it doesn't include private data :)

150. **[People/companies can learn from the published software and therefore protect themselves]**: Publishing such tools help people to understand how it works. It helps to train people how to protect against phishing attacks.

151. **[cannot be meaningfully categorized]**: Verteidigung ist der beste Angriff

152. **[People/companies can learn from the published software and therefore protect themselves]**: Defense should always be one step ahead of attackers, thus defenders need to check their security against realistic offensive tools (exploits, etc) as well as gain knowledge about offensive techniques. Also not-knowing about offensive techniques (if they were not published), would make it a lot harder to develop defensive tools.

153. **[People/companies can learn from the published software and therefore protect themselves]**: Open sourcing gives all stakeholders the option to counteract.

154. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Helping people to create self assessment and awareness programs with as little cost as possible far outways the risk of misuse. „Professional" Phishers with malicious intent have their own toolsets already and will have little benefit.

155. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: Its out there anyway. This helps the defenders

156. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: By providing an automated open source phishing infrastructure it would become easier to enable administrators and IT security staff to raise awareness for this kind of threat. You can only fight an enemy you know, so even administrators with less knowledge about the process of phishing and social engineering in general could get valuable insights.

157. **[Software is Dual Use - Responsible is the user]**: Open Source (deending on license) does not define a specific use case, it is a definiton of availability. MIT License f.i. allows to use the code for any application. Furthermore, code is not bad or good, it may have other purposes and may also be used as a basis for learning.

158. **[People/companies can learn from the published software and therefore protect themselves]**: So that others can learn and utilize this for their teams.

159. **[People/companies can learn from the published software and therefore protect themselves]**: To show what's possible and demonstrate techniques.

160. **[Security by obscurity is not a sufficient basis for security]**: Closed source never helped anyone. Source Code is not a security perimeter.

161. **[People/companies can learn from the published software and therefore protect themselves]**: It can be used to learn.

162. **[Software is Dual Use - Responsible is the user]**: For the same reason it I don't see any ethical issues with other security tools, such as BeEF or Gophish, being open sourced - they are tools, code. They aren't malicious, the way they are used could potentially be. Off-the-Record: I have yet to see a single threat actor running their own mailserver with even a basic level of effort put into it. Why go the extra mile when you can simply abuse a compromised mail account you already have. So with all due respect to your research, another reason why I don't see any ethical question is the fact that I highly doubt the availability of the toolset / automated deployment you describe would have an impact on the operations of bad guys.

163. **[Will benefit for defenders more than criminals]**: While "giving the code" to threat actors is a conser, the value of open source collaboration and I formation sharing outweighs it. Hiding phishing code is like hoarding zero day exploits. Financially lucrative and leads to a worse technological landscape due to ignorance and uncertainty.

164. **[People/companies can learn from the published software and therefore protect themselves]**: One cannot defend against attacks one does not understand.

165. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: devops is hard enough without ppl hiding tricks

166. **[Automation can not replicate good Phishing tactics and procedures]**: A real, motivated attacker will find a way to deploy a phishing-infrastructure as well. As with all security tools, many people will be supported if they find an easy way in the future to test their security. In this case, it's the most critical part of modern it-security that is going to be tested: "Layer 8", also known as the user. The phishing infrastructure is the one thing. What's more important and even harder to set up is the exploit itself, some payload to trick the user into handing over his or her credentials or to open a lure document to deploy malware at the endpoint.

167. **[The concept is nothing new or special and the complexity for implementing the solution it not that high]**: Nothing new. I published my Ansible, terraform scripts as well. It helps people to learn, build own tools and become more resilient.

168. **[Software is Dual Use - Responsible is the user]**: Phishing wird aktiv betrieben, und Phishing Infrastruktur wird genutzt, egal ob die dafür geeignete Technologie der breiten

Masse zur Verfügung gestellt wird, oder nicht. Die Technologie (und deren Offenlegung) selbst kann niemals schuld daran sein, dass Technologie missbräuchlich verwendet wird. Im Gegenteil - Wenn Deployment Code für automatisiertes Phishing öffentlich zur Verfügung steht, besteht die Chance, dass sich Unternehmen/Organisationen selbst mit dem Thema mehr auseinandersetzen, Phishing Simulationen vermehrt durchführen und geeignete Gegenmaßnahmen setzen und die Awareness steigern.

169. **[Phishing Tools are already available, Open Sourcing does not change anything and it is also possible that someone else will make it]**: If you don't release it, someone else will release something similar, or has already done so. Open source will help other devs on the defense-side to optimize their filters.

170. **[Allows easier setup of Phishing Infrastructure for Awarenesstrainings for users]**: Open source software and finished setups of such gives everybody the chance to act like the attackers to simulate attacks. I can reach a broader audience to do such simulations and trainings.

171. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: I believe that while people with deep malicious intent will take the effort to build up infrastructure on their own anyway. Easily available tools like an open-source playbook could however be used in organization of different sizes to raise awareness about the topic of Phishing and IT security in general without too much effort.

172. **[People/companies can learn from the published software and therefore protect themselves]**: Obviously We can know more about it.

173. **[Helps the defenders; open-sourcing will not have an impact on criminal activity because the tools already exist]**: In my opinion, persons with malicious interests will manage to get their goal anyway. E.g. for phishing they can do most of it with existing frameworks and manual labor. However, for pentesters and other security professionals, making tools open source is a very positive approach, for many reasons. To name a few of them: * We like to analyse the tools that we use to make sure they are safe. * We may need to adapt tools to fit for specific usecases and restrictions that we are facing.

174. **[Security by obscurity is not a sufficient basis for security]**: Because security by obscurity is helpful to nobody. Yes, it is a controversial thing to opensource attacking tools but I think it is even more controversial to offer closed source attacking tools for financial benefits like cobalt strike. Better open source it so that defenders can embrace it just alike

175. **[Phishing should be defended with existing security solutions]**: There is not reason it should be unethical. Phishing should anyhow be solved with U2F or FIDO.

C.2 Opinions which support that it is not ethical to open-source deployment code

1. **[Should not be publicly available]**: I don't think tools like this should be publicly available!
2. **[Exploitation by criminals or "Script Kiddies"]**: I don't think it's ethical to fully open-source deployment code for Phishing infrastructure since it could be abused by criminals. However, I believe it would be valuable to share some parts of it in order to educate people on what is possible. In my opinion some critical parts of the deployment should be removed to limit easy abuse.
3. **[Will do more harm than good]**: Will make more harm than good, especially against small to medium companies which don't have strong security.
4. **[Additional malicious code could be introduced]**: You cannot vet the ethics of those building it and therefore malicious code could be introduced.
5. **[Exploitation by criminals or "Script Kiddies"]**: The question is basically, do OS mean it will freely be distributed? If this ist the case, we don't need further script kiddies, who cost money to all of us. So from this standpoint it's non-ethical. If it means, that the code is provided OS to anybody who identifies himself to the provider and agrees to only use it for studying purposes, than this would be acceptable. But there is still a lot of risk.
6. **[Will do more harm than good]**: I do _not_ think that the project would offer more advantages than disadvantages to the public. Obviously with everything that can be used maliciously as well as by a legit pentester, it's always a hard consideration. However, I think in the case of phishing, there is a small difference. With other tools (e.g. Metasploit) one can always argue, that the infrastructure you want to test looks the same as any infrastructure you would use it against as a malicious actor. With phishing on the other hand, you are attacking people so, in my opinion, it's perfectly fine to reduce technical obstacles by simply whitelisting the sending email servers for the duration of the campaign within the target company's infrastructure. That way, you don't need a perfect email server setup. I've done phishing awareness campaigns for over 5 years and never had a problem with this approach. At the same time, spam filter need every bit of technical information they can get to analyse an email and things like (not-) correctly set up SPF, DMARC, etc. are often times strong indicators for those filters. Since those things are, as stated above, not that important for a legit pentester, I think that bad actors would gain disproportionately more from a technical solution to overcome these already low obstacles to send out a strong campaign. In the very few cases that spam filters (and not people) should be tested against such a strong set up, I would argue that the people testing and validating those filters already know their way around email infrastructre and the need for such an automation is not that high.

7. **[Exploitation by criminals or "Script Kiddies"]**: >90% of users would be malicious actors tryint to automate their phishing toolchain

8. **[Will do more harm than good]**: first of all you'd need to define a framework for ethical as an attribute to answer this question on a common basis, missing this I can only derive from my personal ethical definition, which is a bit of a problem as I'd have to lay out the full framework here, which is probably a problem for the length of the text-answer - in short - the question whether publishing code is ethical or not ethical is flawed. so basically the question could be reframed to: does publishing this deployment benefit society? one could argue for transparenency and having something in the public strengthens countermeasures, well - it does - at least for those with enough ressources to follow the issues one might see in their own infrastructure, but basically ethical hacking has already answered the question (after lots of discussions and still counter arguments arise) do at least some sort of sensible disclosure, especially regarding the possible targets of this phising-infrastructure that might be deployed using your os-tool-set. that said the kind of phishing that might be attempted using this toolset would probably be needed as an information to establish a possible yes or no on this question in the end - for protection of the greater good it is a no. and the premise of the question should be re-worked.

9. **[Should not make things easier for attackers or "Script Kiddies"]**: There is no point in making things easier for attackers.

10. **[Exploitation by criminals or "Script Kiddies"]**: Phishing per-se has, awareness campaigns excluded, mainly malicious intents. A full blown infrastructure may encourage even less-specialized persons (e.g. script kiddies) to bring their intentions into the real world rather of dreaming of them whilst being to incompetent to cause major damage. While the tool and the creation of the tool may be ethically neutral, providing it to a person with malicious intent rather tilts to the non-ethical side. Given the many facets IT security and/or related intentions may have, this is IMO not a black or white matter, there is a lot of grey here.

11. **[Should not make things easier for attackers or "Script Kiddies"]**: it makes it easier for everyone to deploy phishing sites and infrastructure. the "cost" at the attackers side should be as high as possible

12. **[Will do more harm than good]**: Your survey did not exactly specify what you mean by "phishing infrastructure", I'm assuming you mean infrastructure that can be used to deploy phishing attacks (for whatever reason). I approach most ethical questions by comparing benefits with potential harms: * Benefits: - Your work makes phishing trainings that use attacks easier and more common. However, if these kind of attacks are a good idea is still an open question. The learning effects to avoid users clicking phishing links seem to be short term, but we can't be sure how the 'gotcha' affects users' future perceptions and decisions. * Potential Harms: - Your work makes it easier for malicious attackers to deploy

phishing attacks. Currently, we do not have a good countermeasure to phishing. So while I don't see the benefits of this kind of phishing training, publishing the code could lead to more phishing attacks that we can't protect against. While I think having an ansible cookbook is useful, I would be careful about distributing it.

13. **[Should not make things easier for attackers or "Script Kiddies"]**: Unlike technical exploits, phishing exploits trust and we should not make it easy for phishers to set up good infrastructure for such attacks.

14. **[Will do more harm than good]**: To easy to abuse and would cause more harm than good on a global scale

15. **[Exploitation by criminals or "Script Kiddies"]**: You are building tools. Tools are handy. Tools are useful. Some of them are weapons. This one is a weapon used by capitalism and criminals motivated by capitalism and quick gains. It's unuseful for society but for quick benefits on the loss of victims, and generates troubles and misinformations. It is generating nuisances and solves no problem. It is not a gift to humanity.

16. **[Should not make things easier for attackers or "Script Kiddies"]**: Phishing infrastructure (as I understand it): Creating Web pages, setting up domains, deploying web services, preparing mail cannons. Professionals who provide Phishing attacks as a service to test customers awareness have their own infrastructure to perform simulated attacks. The number of targets (organisations, companies) is typically low, the benefit of automation is, in my opinion, low. Deploying Phishing infrastructure with automation tools like Ansible allows to deploy attacks against a large number of victims with little effort. This is not the typical use case for White Hat Phishing campaigns. My understanding of Phishing reality (from projects I was involved and discussions with security professionals): The majority of possible Phishing victims have no or very little protection (private people and small businesses). In my opinion the overall risk will increase if more people with low ethics have access to such tools. Small businesses have no surplus resources to set up their own Phishing campaign and thus have no benefit from automated tools. Professional security and social engineering consultants already have their infrastructure like stated above. Tools that can be abused spread fast in circles with low ethics. Conclusion: Open source automation for Phishing might not beneficial for the overall security.

17. **[Should not make things easier for attackers or "Script Kiddies"]**: In the past I may have said that it was ethical to publish this open source. But over time, my views have shifted somewhat as I see more and more offensive tools simply being used by bad actors directly. Of course, these things can be re-created by the bad guys, and not publishing them will not make them go away, but I am beginning to think that in certain cases we might just want to avoid making things easy when we don't have to. I do think it is important that red and blue teams can properly perform attacks like the bad guys, and

sometimes that means publishing things so they can be understood and tested or simulated. Some attacks are very complex and having specialists publish tools or exploit code that allow things to be tested is very helpful. In the case of phishing however, discussion of the important steps and things to consider should be enough for a professional to put together their own. When this is possible, I think it makes sense to avoid providing a fully weaponised version that could be used immediately by the bad guys, lowering the barriers of entry for them. This is a nuanced issue, and I would not strongly object to a public phishing tool, but I do believe that as professionals we need to think more about harm reduction than we have in the past.

18. **[Will do more harm than good]**: I understand that publishing the code as open source would provide opportunities for learning and understanding - but I think we are not yet ready for it to become relatively easy for large amount of people to access this type technology that can be used for malicious purposes against the masses. The masses can't learn as fast as the "others" can. I think it could cause a lot of harm for people who are not so tech-savy. I don't think it's right. Information and learnings can be shared by other means.

19. **[Will do more harm than good]**: Do not deliver tools, ideas a.s.o. to harm someone to the public

20. **[Because it is unethical in their opinion]**: Making and distributing tools for scam is unethical. Phishing is scam.

21. **[Will do more harm than good]**: When publishing anything that can be used maliciously, one should always consider the benefit vs the damage. In the case of phishing, the whole act of simulated-phishing as a useful training method for improving security is under debate. The consensus seems to be that it needs to be combined with extensive training and awareness campaigns, which means the benefit for defenders to have this part as a easy to use open source tools is small. For attackers though, phishing works and they already use it extensively. Giving them tools to do it more efficiently and cheaply will just reduce their cost and give them the ability to do more of it, including giving people who were previously unable to do it the ability to do it. So in summary, the benefit of open sourcing offensive tooling is questionable, but the downside is clear.

22. **[cannot be meaningfully categorized]**: There is this very short time between delays of hacks and their announcement to get a better hack

23. **[Because it is unethical in their opinion]**: No kind of phishing is ethical and this anything that makes phishing easier is unethical in my opinion.

24. **[Should not make things easier for attackers or "Script Kiddies"]**: While I would agree a motivated attacker or pen tester would have little trouble solving the challenges this code alleviates, the "script kiddie" would benefit most from turn key. Withholding code

will not mitigate the risk they impose, yet does maintain the slightly higher bar to entry. I apply this to the open source utilities this code only automates deployment around as well.

25. **[cannot be meaningfully categorized]**: (I am no native speaker, so please ignore bad english) I voted with "No", because "yes" fitted less. I dislike both answers. The mentioned action has no ethical onotation

26. **[Exploitation by criminals or "Script Kiddies"]**: Enables abuse by malicious parties. And I don't see a legitimate use.

27. **[Should not make things easier for attackers or "Script Kiddies"]**: Because it makes it easier for script kiddies to harm inexperienced or just strait up "stupid"/naiv users. There would be need some sort of tool to let only ethical hackers (persons who say they are ethical) to get these resources.

28. **[Should be only available for legit usecases]**: it's too easy to do as it is. if it were somehow able to make it only live in some given system's infrastructure (ie a given red/blue domain) so that it could be safely contained, that'd be one thing - but that pretty much ain't possible

29. **[Will do more harm than good]**: I believe that phishing is generally an issue that can be solved through technical means. As such, I see no benefits to phishing simulations or their use as an awareness tool in a company. As such, from an idealist/ethical point of view, I see little to no value in the availability of such a toolkit to anyone interested compared to the likelyhood of damage in the same way that while I am not against gun ownership, I do not believe that these sophisticated tools should be freely available to anyone who wishes.

30. **[Exploitation by criminals or "Script Kiddies"]**: Not only is it too easily repurposed, but the actual blue team value of phishing simulation is dwindling. Resources are better spent on other efforts. People have to click on things as part of their jobs, and securing that is more important.

31. **[No benefit for the general public]**: There is no benefit to the general public of doing so. Exploits and rootkits can be used for teaching purposes, but phishing infrastrucure is usually not complex, not sphisticated, and not interesting.

D Configuration Validation Data

mail-tester.com result - Source of the mail that was sent to mail-tester.com

```
Received: by mail-tester.com (Postfix, from userid 500)
        id B0D1BA9756; Sat, 26 Mar 2022 02:00:17 +0100 (CET)
X-Spam-Checker-Version: SpamAssassin 3.4.2 (2018-09-13) on
 ↪  mail-tester.com
X-Spam-Level:
X-Spam-Status: No/-0.1/5.0
X-Spam-Test-Scores:
 ↪  DKIM_SIGNED=0.1,DKIM_VALID=-0.1,DKIM_VALID_AU=-0.1,
        SPF_HELO_NONE=0.001,SPF_PASS=-0.001,URIBL_BLOCKED=0.001
X-Spam-Last-External-IP: 49.12.207.135
X-Spam-Last-External-HELO: leakybuffer.br0ken.cloud
X-Spam-Last-External-rDNS: leakybuffer.br0ken.cloud
X-Spam-Date-of-Scan: Sat, 26 Mar 2022 02:00:17 +0100
X-Spam-Report:
        *  0.0 URIBL_BLOCKED ADMINISTRATOR NOTICE: The query to URIBL
         ↪  was
        *      blocked.  See
        *
         ↪  http://wiki.apache.org/spamassassin/DnsBlocklists#dnsbl-block
        *      for more information.
        *      [URIs: mail-tester.com]
        * -0.0 SPF_PASS SPF: sender matches SPF record
        *  0.0 SPF_HELO_NONE SPF: HELO does not publish an SPF Record
        *  0.1 DKIM_SIGNED Message has a DKIM or DK signature, not
         ↪  necessarily
        *       valid
        * -0.1 DKIM_VALID Message has at least one valid DKIM or DK
         ↪  signature
        * -0.1 DKIM_VALID_AU Message has a valid DKIM or DK signature
         ↪  from
        *      author's domain
```

```
Received-SPF: Pass (sender SPF authorized) identity=mailfrom;
 ↪  client-ip=49.12.207.135; helo=leakybuffer.br0ken.cloud;
 ↪  envelope-from=service@phishingparty.on.br0ken.cloud;
 ↪  receiver=test-1s808liga@srv1.mail-tester.com
DMARC-Filter: OpenDMARC Filter v1.3.1 mail-tester.com 499F4A95A6
Authentication-Results: mail-tester.com; dmarc=pass
 ↪  header.from=phishingparty.on.br0ken.cloud
Authentication-Results: mail-tester.com;
        dkim=pass (2048-bit key; unprotected)
         ↪  header.d=phishingparty.on.br0ken.cloud
         ↪  header.i=@phishingparty.on.br0ken.cloud
         ↪  header.b=WGoipKMY;
        dkim-atps=neutral
Received: from leakybuffer.br0ken.cloud (leakybuffer.br0ken.cloud
 ↪  [49.12.207.135])
        (using TLSv1.2 with cipher ADH-AES256-GCM-SHA384 (256/256
         ↪  bits))
        (No client certificate requested)
        by mail-tester.com (Postfix) with ESMTPS id 499F4A95A6
        for <test-1s808liga@srv1.mail-tester.com>; Sat, 26 Mar 2022
         ↪  02:00:14 +0100 (CET)
Received: from leakybuffer.br0ken.cloud (localhost [127.0.0.1])
        by leakybuffer.br0ken.cloud (Postfix) with ESMTPS id
         ↪  4D9533F7C3
        for <test-1s808liga@srv1.mail-tester.com>; Sat, 26 Mar 2022
         ↪  01:00:14 +0000 (UTC)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/simple;
        d=phishingparty.on.br0ken.cloud; s=dkimparty; t=1648256414;
        bh=AOrtcI1ja9fKj16Cmago46puZ948yzRIaAyXBarE5EA=;
        h=Date:Subject:To:From:From;
        b=WGoipKMYIDBlcrEcKq6AVIcXPbvByoqYmvTAHp9HXuPE31TUUeppw5EyEmXPz1McP

          ↪  HAnrK15Y3Fi+CfpfA7IuZurhPkOnvI74jplv3JrompzPJrsOFZf4kabi9GlcdjlIuu

          ↪  ptKWhOKsizeErHf+oyWGP+2511Jx5o6VSUu7rPnt/oNfU1ZPFU1epmrs43HFdxs+9q

          ↪  KDHt2V8/ea+fuLKtyLGC56dQrlrQMyoaP7KVpnv2xi/mo8q+mmCJZnV35qaSRoqTK0

          ↪  8vkLgzTtATVozJg/bFmsqoDG4metfX3p2ahKMww4jv+FCfUx8JHO5D/iYOZe7yIGPq
         OH1ClaLutTOIg==
Mime-Version: 1.0
Date: Sat, 26 Mar 2022 01:00:14 +0000
Subject: Password expired
To: "Test-mail User" <test-1s808liga@srv1.mail-tester.com>
From: "Phishing Service" <service@phishingparty.on.br0ken.cloud>
X-Mailer: gophish
Message-Id:
 ↪  <1648256414316858213.131319.5855275837414557197@leakybuffer.br0ken.cloud>
X-Phishing: In Progress
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable

Dear Test-mail,

The password for test-1s808liga@srv1.mail-tester.com has expired.
 ↪  Please re=
set your password here
 ↪  https://phishingparty.on.br0ken.cloud?rid=3DDy4LsgX

Thanks,
Morning Catch IT Team
```

mail-tester.com result - DKIM Signature

```
v=1;
a=rsa-sha256;
c=relaxed/simple;
d=phishingparty.on.br0ken.cloud;
s=dkimparty;
t=1648256414;
bh=AOrtcI1ja9fKj16Cmago46puZ948yzRIaAyXBarE5EA=;
h=Date:Subject:To:From:From;
b=WGoipKMYIDBlcrEcKq6AVIcXPbvByoqYmvTAHp9HXuPE31TUUeppw5EyEmXPz1McPHAnrK
15Y3Fi+CfpfA7IuZurhPkOnvI74jplv3JrompzPJrsOFZf4kabi9GlcdjlIuuptKWhOKsi
zeErHf+oyWGP+2511Jx5o6VSUu7rPnt/oNfU1ZPFU1epmrs43HFdxs+9qKDHt2V8/ea+fu
LKtyLGC56dQrlrQMyoaP7KVpnv2xi/mo8q+mmCJZnV35qaSRoqTK08vkLgzTtATVozJg/b
FmsqoDG4metfX3p2ahKMww4jv+FCfUx8JHO5D/iYOZe7yIGPqOH1ClaLutTOIg==
```

mail-tester.com result - DKIM Public Key

```
v=DKIM1;
h=sha256;
k=rsa;
s=email;
p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA58qiiTx+xG2hSRIUInWUNr+aZR
NQzp7hL6TpwQfIPNMy55eeJBjVSAhTwGcqn7VpX4rh5zFfyb02h5uWzi8zT6XcmUq4fTex
uAl6ZZZNkLpXRgN0hGLaMU0z1JW1AkR8eBFL9SPcRmE2SsrfKEyH5hq1KE+ssAy/4nWhdX
2QYyN14LxnzJs3HFOLYKxeOCxqIAD96ZYN1arEj7V1rYVy4UokkCLvRhyShILRT13pr0hp
rds8mL2R9WWYi94iXCCkm+O8T0mpHuf6oEIQD7+KUgagn858RaZY1geewfYhDKGVgeax6p
k2mJ4AAjQu2uEO8z0BR4ghkH3p/knS2YoVjwIDAQAB
```

Test result mail from CheckTLS TestReceiver

```
From TestSender@CheckTLS.com  Sat Mar 26 01:16:21 2022
Return-Path: <TestSender@CheckTLS.com>
X-Original-To: root@phishingparty.on.br0ken.cloud
Delivered-To: root@phishingparty.on.br0ken.cloud
Received: from mail11-do.checktls.com (mail11-do.checktls.com
 ↪  [134.209.47.28])
        by leakybuffer.br0ken.cloud (Postfix) with ESMTPS id
         ↪  ECCBB3F7B3
        for <root@phishingparty.on.br0ken.cloud>; Sat, 26 Mar 2022
         ↪  01:16:20 +0000 (UTC)
Authentication-Results: leakybuffer.br0ken.cloud;
        dkim=pass (1024-bit key; secure) header.d=checktls.com
         ↪  header.i=@checktls.com header.a=rsa-sha256
         ↪  header.s=default header.b=AOs6lFmr;
        dkim-atps=neutral
Received: from localhost.localdomain (ts11-do.private.checktls.com
 ↪  [10.132.127.32])
        by mail11-do.checktls.com (8.15.2/8.15.2) with ESMTP id
         ↪  22Q1FJF8032562
        for <root@phishingparty.on.br0ken.cloud>; Fri, 25 Mar 2022
         ↪  21:15:19 -0400
DMARC-Filter: OpenDMARC Filter v1.3.2 mail11-do.checktls.com
 ↪  22Q1FJF8032562
Authentication-Results: mail11-do.checktls.com; dmarc=pass
 ↪  (p=quarantine dis=none) header.from=CheckTLS.com
```

```
Authentication-Results: mail11-do.checktls.com; spf=pass
 ↪  smtp.mailfrom=TestSender@CheckTLS.com
DKIM-Filter: OpenDKIM Filter v2.11.0 mail11-do.checktls.com
 ↪  22Q1FJF8032562
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=checktls.com;
        s=default; t=1648257319;
        bh=E9DD+nXaF1SU1O9+5nsN5M7xL9Dyofd+9VlYqLhm8io=;
        h=From:To:Subject:Date:From;
        b=AOs6lFmr9/kCohqO729CoOye/TdW1rZzWzXfj0jJSzWW/oYyxDCchDeJYzi7hASC6

          ↪  q7zTl46zdD5aaDjf/G8RkP6NSFYMEbo1fOX1RyRXBRwSjYbl+3bPp7/Wr1eABje+ni
         IhYSsmlc+wqgrkZZTvUSiWMHWYO/l+5fIaipOScw=
Message-Id: <202203260115.22Q1FJF8032562@mail11-do.checktls.com>
From: "CheckTLS Test Sender TLS" <TestSender@CheckTLS.com>
To: root@phishingparty.on.br0ken.cloud
Subject: SUCCESSFUL
X-Mailer: TestSender
Date: Fri, 25 Mar 2022 21:15:19 -0400
MIME-Version: 1.0
Content-Type: text/plain; charset="US_ASCII"
Content-Transfer-Encoding: quoted-printable

   SUCCESSFUL //email/test From:

   Your email was sent securely using TLS.

   TLS:                   Successful
   From:                  root@phishingparty.on.br0ken.cloud
   Via:                   2a01:4f8:c0c:589e::1
   Date:                  2022-03-25 21:15:17 EDT
   Subject:               gmbrrs58rhz89
   SSLVersion:            TLSv1_3
   SSLCipher:             TLS_AES_256_GCM_SHA384
   ClientCert:            n/a
   SNI:                   [undefined]
   SPF_mfrom.Record:      v=3Dspf1 mx ip4:49.12.207.135
    ↪  ip6:2a01:4f8:c0c:58=
9e::1 a:leakybuffer.br0ken.cloud -all
   SPF_mfrom:             pass:
    ↪  local=3D"phishingparty.on.br0ken.cloud: 2a0=
1:4f8:c0c:589e::1 is authorized to use
 ↪  'phishingparty.on.br0ken.cloud' in '=
mfrom' identity (mechanism 'mx' matched)"
   SPF_helo:              none: local=3D"leakybuffer.br0ken.cloud: No
    ↪  appli=
cable sender policy available"
   DKIM:                  pass:
    ↪  signature=3D"@phishingparty.on.br0ken.cloud=
" result=3D"pass"
   DKIM_policy.sender:    "o=3D~"(default), result=3D"accept"
   DKIM_policy.author:    "o=3D~"(default), result=3D"accept"
   DKIM_policy.ADSP:      ""(default), result=3D"accept"
   DMARC_result:          pass
   DMARC_disposition:     none
   DMARC_dkim:            pass
   DMARC_dkim_align:      strict
   DMARC_spf:             pass
   DMARC_spf_align:       strict
   DMARC_published.v:     DMARC1
   DMARC_published.p:     none
   DMARC_published.sp:    n/a
   DMARC_published.adkim: s
   DMARC_published.aspf:  s
   DMARC_published.rua:   n/a
   DMARC_published.ruf:   n/a
   DMARC_published.rf:    n/a
   DMARC_published.ri:    n/a
   DMARC_published.pct:   n/a

   (this email intentionally has limited formatting)
The transcript of the eMail SMTP session is below, with:
--> this is a line from your email system to us (~~> when encrypted)
<-- this is a line to your email system from us (<~~ when encrypted)
=3D=3D=3D this is a line about the tls negotiation (cypher, cert,
 ↪  etc)
*** this is an error, warning, or info line that the test found

<-- 220 ts11-do.checktls.com ESMTP TestSender Fri, 25 Mar 2022
 ↪  21:15:15 -04=
00
```

```
--> EHLO leakybuffer.br0ken.cloud
<-- 250-ts11-do.checktls.com Hello mail-ej1-x62c.google.com
 ↪  [2a01:4f8:c0c:5=
89e::1], pleased to meet you
<-- 250-ENHANCEDSTATUSCODES
<-- 250-8BITMIME
<-- 250-STARTTLS
<-- 250 HELP
--> STARTTLS
<-- 220 Ready to start TLS
=3D=3D=3D=3D tls negotiation successful =3D=3D=3D=3D
~~> EHLO leakybuffer.br0ken.cloud
<~~ 250-ts11-do.checktls.com Hello mail-ej1-x62c.google.com
 ↪  [2a01:4f8:c0c:5=
89e::1], pleased to meet you
<~~ 250-ENHANCEDSTATUSCODES
<~~ 250-8BITMIME
<~~ 250 HELP
~~> MAIL FROM:<root@phishingparty.on.br0ken.cloud>
<~~ 250 Ok - mail from root@phishingparty.on.br0ken.cloud
~~> RCPT TO:<test@TestSender.CheckTLS.com>
<~~ 250 Ok - recipient test@TestSender.CheckTLS.com
~~> DATA
<~~ 354 Send data.  End with CRLF.CRLF
~~> Received: by leakybuffer.br0ken.cloud (Postfix, from userid 0)
~~>     id E5B7D3F7C3; Sat, 26 Mar 2022 01:15:14 +0000 (UTC)
~~> DKIM-Signature: v=3D1; a=3Drsa-sha256; c=3Drelaxed/simple;
~~>     d=3Dphishingparty.on.br0ken.cloud; s=3Ddkimparty;
 ↪  t=3D1648257314;
~~>     bh=3D4Xwx57LXGMwcNh58I4VeAbBlnzu0wZSiVeQCP+eBGNo=3D;
~~>     h=3Dto:from:subject:Date:From;
~~>
 ↪  b=3Dd+lmUfMAwOhtapPEa/9tUgyZ1yi5f7PRPHMgG7W4cHAShJayzyqbnD9nliTcOFS=
8r
~~>
 ↪  /5b9oaVYBPXs1lXatGM7pcDyBxVaG5NU8FV0A3sCFxbZuOtJDpgmq67XQECukgWRJW
~~>
 ↪  lV5FhnjCo9xbSNJXAiX+xOjGmCZbhzLesgV/3R0ji7OwjQarQyVpBk+PwLeeBcOWJv
~~>
 ↪  CrRSxaNo9sFRuYQItN4V7XF5XSx+7KovjU/lHLeIZl5oNxJd2QqEeiBjvEpWY6EAWh
```

```
~~>
 ↪  bPsdcO8BFkQHtOnBj8hJcUagfyPOeoywlFhjTHdJH/vZrVFrA16W1kY9CHagVIkYt4
~~>      QFZd1uvdPoqXw=3D=3D
~~> to: test@TestSender.CheckTLS.com
~~> from: root@phishingparty.on.br0ken.cloud
~~> subject: gmbrrs58rhz89
~~> Message-Id: <20220326011514.E5B7D3F7C3@leakybuffer.br0ken.cloud>
~~> Date: Sat, 26 Mar 2022 01:15:14 +0000 (UTC)
~~>
~~> This is a test message.
~~> TLS
~~> TEXTRESULT
~~> Headers
~~> SSLVERSION
~~> SSLCIPHER
~~> ClientCert
~~> SNI
~~> SPF
~~> DKIM
~~> DMARC
~~> SMTP
~~> .
<~~ 250 Ok
~~> QUIT
<~~ 221 ts11-do.checktls.com closing connection
```