Kali Linux: The essential Toolkit for Ethical Hackers in Network Exploitation

Federico Cignoli (544952), Vito Giacalone (546645), Paolo Fabrizio (536594),
Seyed Amin Fattahzadeh (531682), Maral Seyed Ghasemi (531685), Oguzhan Turan (517417)

June 23, 2024

Contents

1 General Introduction

In the ”Enterprise Digital Infrastructure” course, we undertook a project aimed at testing various cybersecurity tools to demonstrate how easily an average user can unknowingly encounter security issues. In an era where cybersecurity is crucial for protecting companies and users from potential attacks, this project aims to raise awareness about the importance of adopting preventive and conscious measures. We utilized advanced tools such as Burp Suite, zPhisher, Nmap, BeEF, Bettercap, and Nikto to explore and test various security scenarios.

Burp Suite allowed us to identify and analyze vulnerabilities in web applications, demonstrating how user data can be exposed to risks. zPhisher was used to simulate phishing attacks, highlighting how easily a user can be deceived into providing sensitive information. Nmap helped us scan networks to identify weak points that attackers could exploit, while BeEF (Browser Exploitation Framework) showed how browsers can be compromised to gain unauthorized access to user data. Bettercap was employed to conduct network reconnaissance and perform man-in-the-middle attacks, showcasing how attackers can intercept and manipulate network traffic. Nikto enabled us to scan web servers for known vulnerabilities, emphasizing the need for regular security updates and patches.

The goal of this project is to provide a practical and in-depth understanding of cybersecurity threats, showing how crucial it is to implement effective security measures and maintain a high level of awareness to protect digital infrastructures and end users.

2 Burp Suite Introduction

Burp Suite is a set of integrated tools designed to test the security of web applications. It is developed by PortSwigger and widely used by cybersecurity professionals, developers, and penetration testing experts to identify and analyze vulnerabilities in web applications.

2.1 What is Burp Suite

Burp Suite is a modular platform that allows you to perform a wide range of security tests on web applications. It offers a set of tools that work together to support the entire testing process, from mapping and analyzing the attack surface of the application to identifying and exploiting vulnerabilities. This can be used to assess if a website is safe, because a website owner can try to use these simple tools, provided by Burp Suite, and if his website didn’t pass some test, he could work to remove that security issue.

2.2 What Can You Do with Burp Suite

- **Proxy:** Burp Suite can intercepts HTTP/HTTPS packets between the browser and the web application, allowing users to inspect and manipulate requests and responses.
- **Intruder:** This tool allows you to perform automated attacks, such as brute force and fuzzing, on various points of a web application to test its resistance to certain types of attacks.
- **Repeater:** It allows you to repeatedly send modified HTTP requests to manually test various parameters and check for vulnerabilities.
- **Sequencer:** It analyzes the randomness of session tokens to assess the security of session management mechanisms.
- **Decoder:** It helps to decode and encode data in various formats, useful for analyzing and manipulating payloads during security tests.
- **Comparer:** It allows you to compare two sets of data, useful for identifying differences between two HTTP requests or responses.

3 Using Burp Suite as a Proxy

3.1 Web caching

As we know, when we search online something of our interest we are connecting to various servers, and in order to visualize a web page we have to send HTTP request and get the corresponding responses. Even a simple web page is made up of a big number of HTTP requests:

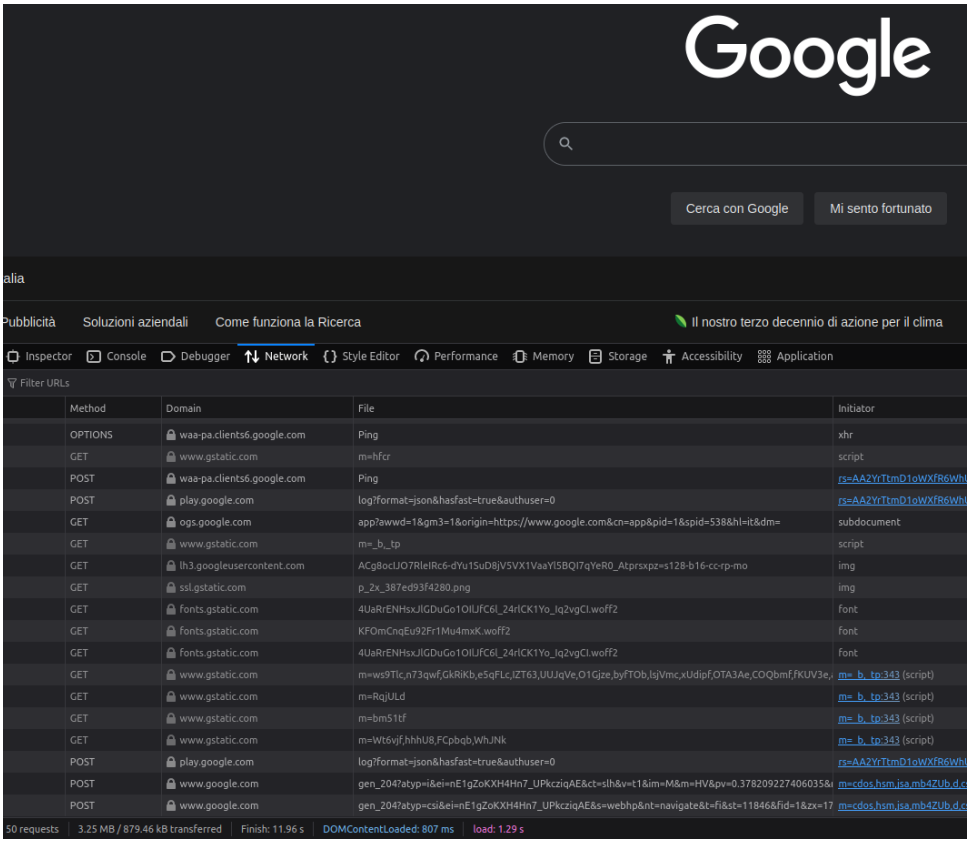


Figure 1: The most searched web page (google.com) is made up of plenty of requests (50 in this case), despite its simple style

since millions of users search on the Web their favourite content at the same time, reducing the number of this requests could be a very good way to light the work of the servers, and this will improve our (user) experience since the pages are loaded faster. This is the purpose of web caches. We know that we can have 2 levels of these caches:

- **Browser cache**: this is a private cache of a single users, so that the next requests could benefit of the content cached with the previous searches.
- **Dedicated caching infrastructure**: this could use for example a proxy server, so this means that every request done from a device in that area (for example, the proxy is shared for all the users in a company) before doing the actual HTTP request will check before if the content is in the browser cache, and then it will check if it is in the proxy cache, this could speed up a lot the time of the requests, because if for example a team is working on some kind of project, we can suppose that they are using the devices to search the same topics, so that the next time we do a request for a page searched from another colleague we could benefit of the proxy cache.

The proxy technology isn't bounded to companies, it's easy to find online some sites that shows some free proxies on which you can connect and exploit their caches in order to speed up your queries.

Proxy IP	Proxy Port	Last Check	Proxy Speed	Uptime	Proxy Country	Anonymity
72.10.160.90	2629	✔ 7 mins ago	1325 ms	4% (93)	Canada	Elite
72.10.164.178	28427	✔ 7 mins ago	1384 ms	4% (92)	Canada	Elite
147.75.122.245	999	✔ 7 mins ago	4464 ms	7% (45)	Colombia - Bogotá	Transparent
67.43.227.226	1165	✔ 7 mins ago	1912 ms	7% (46)	Canada	Elite
182.191.84.39	80	✔ 7 mins ago	4327 ms	7% (82)	Pakistan - Gujranwala	Elite
38.183.152.34	8090	✔ 7 mins ago	4886 ms	14% (23)	Dominican Republic - Bella Vista	Transparent
62.103.66.18	3128	⚠ 7 mins ago	4428 ms	7% (66)	Greece - Nea Smyrni	Transparent

Figure 2: Some free proxies currently available for any purpose

3.2 Issues with open proxies

Unfortunately, these free proxies suffer from many security issues: remember that these are devices on which our HTTP requests go through, so if someone could see the traffic that flows on that proxy, he or she will be able of acquire our information, even the ones that shouldn't be given away.

3.3 Burp's proxy

In Burp Suite we can have a look at the function "Proxy" that simulates the behaviour of a proxy between us and the web server:

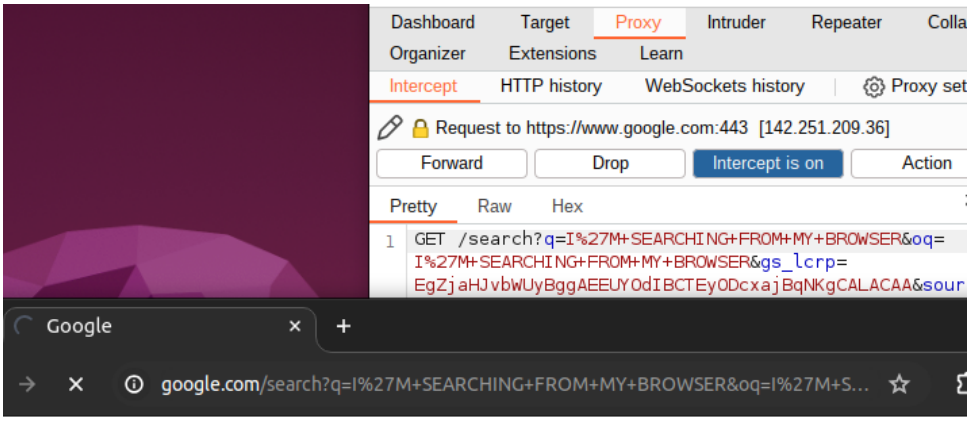


Figure 3: We can see that the proxy shows exactly our request

One could think that if someone can look at the messages that flow on the proxy, they will have access to what we search, since they can look at the pages that we are requesting from the web server, but things are much worse than that. Let's look at some other packets that we can send at the web server:

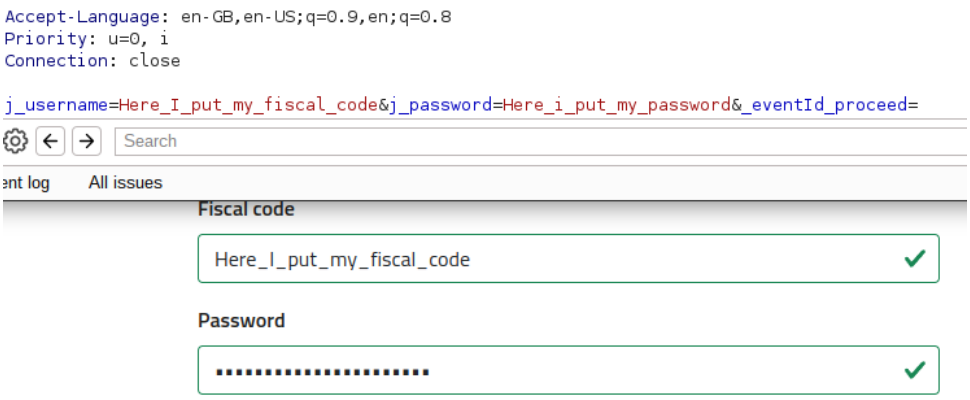


Figure 4: The username and the password are shown as clear text in the proxy, so if the attacker has access to the proxy, he or she can easily steal our credentials

4 Using Burp Suite for brute force attacks

We can use also Burp Suite to exploit brute force attacks. The idea is that we can send an http request with some placeholder for username and password, in order to let the Burp Intruder try many username and password, we can then notice when the bot has found the correct credential (just 1 of the 2), and then we can concentrate to get the other one.

4.1 Obtain the packet

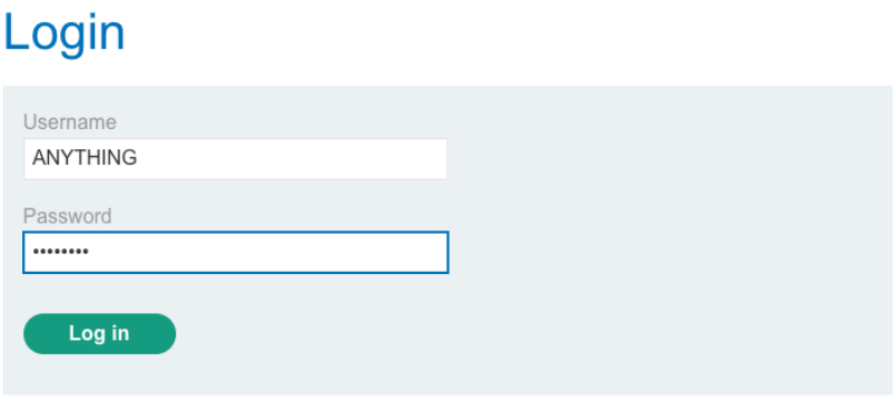


Figure 5: First, we send a fake request with anything that we want as username and password, these are set in order to have a text to replace when we want to do the attack

Then on Burp we have to select the HTTP History tab: here we want to identify the ”POST /login” request. This request is the one discussed above (the one that will contain our placeholder).

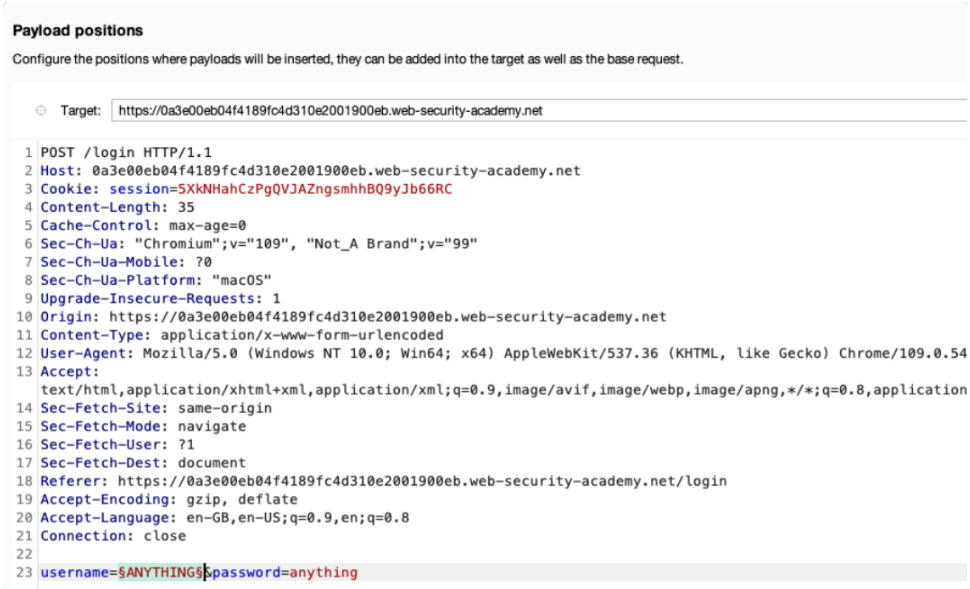


Figure 6: On this request we should put the special character before and after the word that we want to replace (in this case the ”ANYTHING” in the username field)

4.2 Prepare the attack

Then we choose our type of attack. In this case we are using the ”Sniper” attack, that inserts a single set of payloads, one by one, into one or more positions within the request.

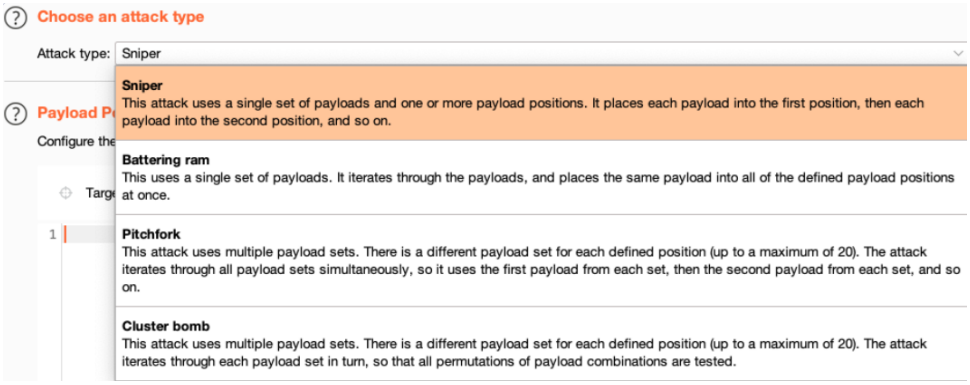


Figure 7: We can see that we can select many types of attacks, for now let’s choose only this one

The next step is to choose the words that we want to use for our brute force attacks, this step is very powerful since we do not have to try all the combination of alphanumeric characters, but if we know a bit our victim we can restrict the search only to the usernames and passwords that we consider as plausible.

How we can know the credentials of our victim?

- Username: many times the username in some services is just the mail of the person, so maybe if I work in the same company with him or her, one of the two credential is already in our possession. Of course if we know that the victim has 2 or more emails, we can just put all of them in the payload and let the intruder try all of the combinations.

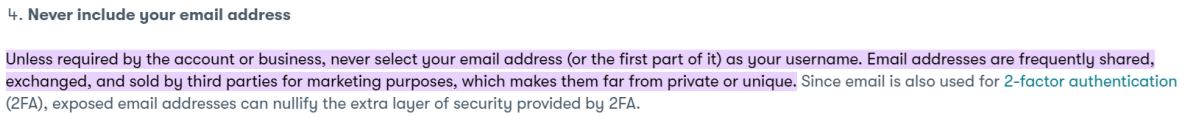


Figure 8: This good practice is almost never used

Another type of usernames that usually are used are the ones that contains personal information, in Italy for example some websites suggest to you (or force you) to insert your Tax ID Number (Codice Fiscale) as your username. This is very bad since they contains personal informations.

Figure 9: Some good practices when we choose an online username

Another problem of this type of usernames are that they could be easily computed with some information that are not too difficult to get!

Figure 10: Excluding the place when you are born (that isn't so difficult to get) the other information are known as long as you know a bit the other person from which you want to steal the credentials

- Password: this field is a bit trickier to know, but we know that many people aren't aware that much on the risks that are on internet and so we can exploit probability in our favor: many people use common passwords!

The Worst Passwords List is an annual list of the 25 most common passwords from each year as produced by internet security firm SplashData.[4] Since 2011, the firm has published the list based on data examined from millions of passwords leaked in data breaches, mostly in North America and Western Europe, over each year. In the 2016 edition, the 25 most common passwords made up more than 10% of the surveyed passwords, with the most common password of 2016, "123456", making up 4%.[5]

Top 25 most common passwords by year according to SplashData

Rank	2011[6]	2012[7]	2013[8]	2014[9]	2015[10]	2016[5]	2017[11]	2018[12]	2019[13]
1	password	password	123456	123456	123456	123456	123456	123456	123456
2	123456	123456	password	password	password	password	password	password	123456789
3	12345678	12345678	12345678	12345	12345678	12345	12345678	123456789	qwerty
4	qwerty	abc123	qwerty	12345678	qwerty	12345678	qwerty	12345678	password
5	abc123	qwerty	abc123	qwerty	12345	football	12345	12345	1234567

Figure 11: These are the list of the top 5 password divided by years, as we can see, the most common passwords are a big percentage on the total

If the password isn't the most common one, we can exploit the fact that people use other types of weak passwords, for example in theory with n ASCII characters we have n*10^{13} different combinations, but the problem is that users typically use common names, birth dates, small sequences and so on.

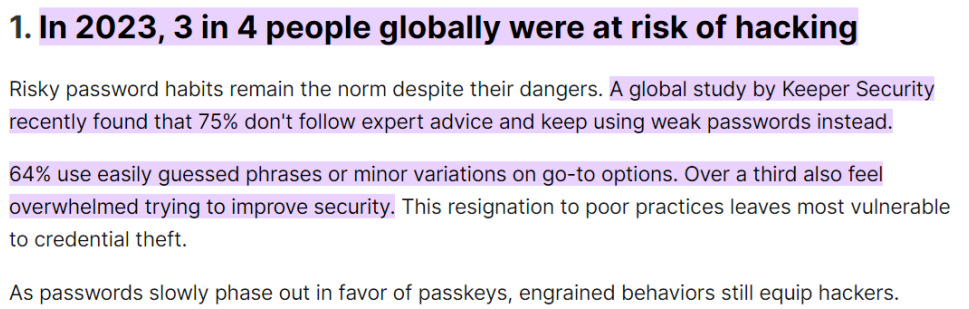


Figure 12: A very huge number of users use these weak password to defend their accounts

4.3 Perform the attack

Now that we have our candidate usernames and passwords we can start the attack, basically we write in a list all of the plausible credentials that we want to use:

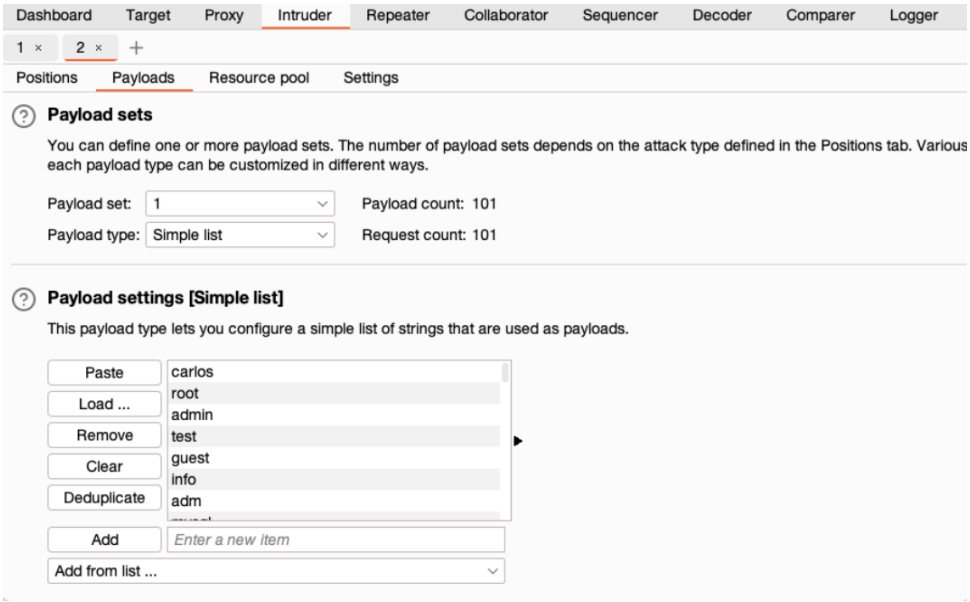


Figure 13: Here some common words are used, but here is where we have to put the possible credentials that we want to use

And then we let the intruder do its work, here is a picture to understand what it do:

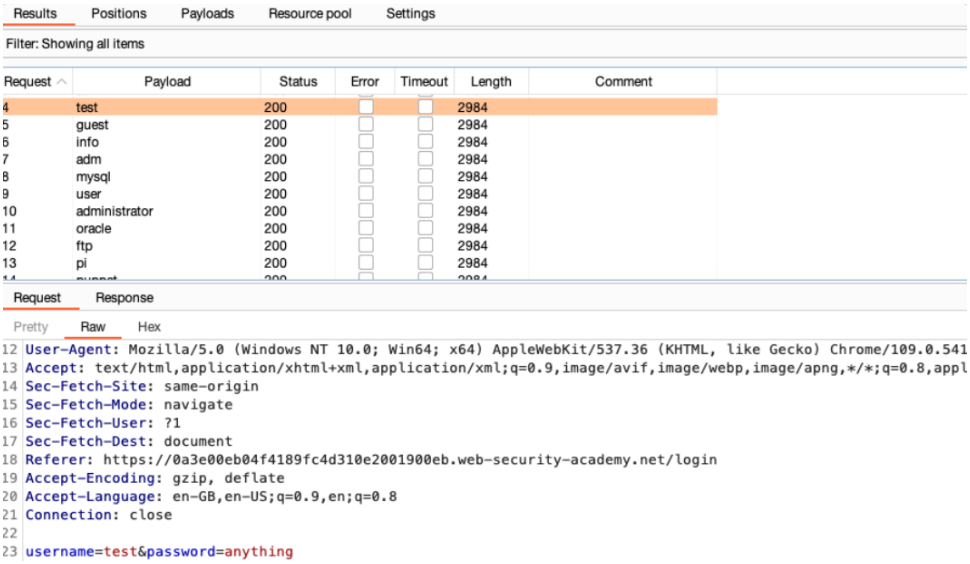


Figure 14: We can see that it's replacing our previous word with the one that we gave in the list, one at a time

Of course we could put 2 placeholders (one for the username and one for the password) and let the intruder try all the combinations, but this means that, if I have 1000 possible usernames and 1000 possible passwords, I have to try with 1000*1000 combinations. But there is a workaround, we can try to guess before the username (while the password could even be wrong) and then try to guess the other credential once the first is known:

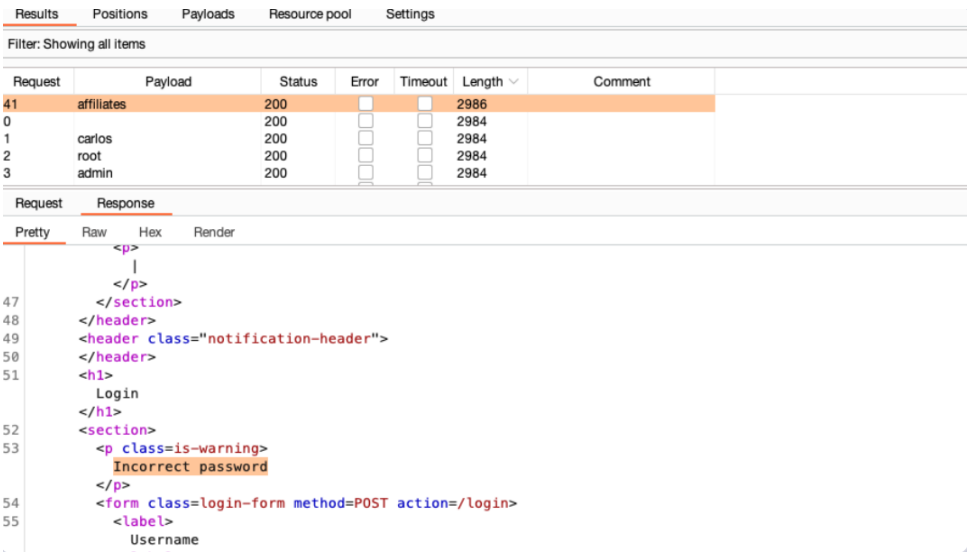


Figure 15: If we order the payloads by length, we can see that the others have pretty much the same length: if we inspect these packets we can see the text "invalid username", but then we can notice that one packet has a different length, in fact if we inspect this one the message is different: invalid password, meaning that the first chek is completed!

Then we can just repeat the experiment with the correct username, with the objective to obtain the password:

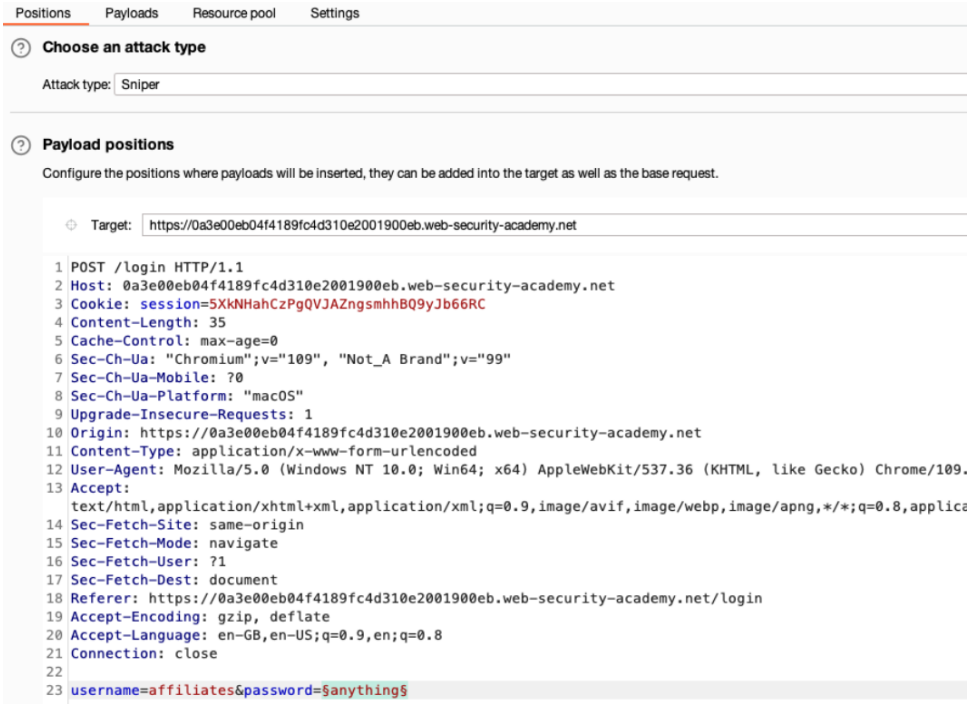


Figure 16: Now the placeholder are on the password, meaning that we can replace the value that we want with it

5 Web Phishing

Website phishing is a malicious tactic where an attacker creates a deceptive copy of a legitimate website to trick users into believing they are interacting with a trusted website. This fraudulent site aims to steal sensitive information such as usernames, passwords, and financial details.

By mimicking the appearance and functionality of a genuine website, attackers can lure victims into providing their personal data, which are collected and used for malicious purposes.

Email phishing, on the other hand, involves crafting an email that appears to originate from a trusted source. Attackers forge the email header to make it look like the message is coming from a legitimate entity, such as a bank, a reputable company, or even a known contact. The goal is to convince the recipient to open the email, trust its content, and respond or click on a link inside it. These links often lead to phishing websites designed to capture the user's sensitive information.

A particularly insidious form of phishing combines these two methods. Attackers send a convincing email containing a link to a fake website. When the user clicks the link, they are redirected to a site that looks nearly identical to the legitimate one. Believing they are on a trusted site, users may enter their login credentials, personal details, or financial information, that are captured by the attackers and used for fraudulent activities.

The effectiveness of these phishing tactics hinges on the abilities of the attackers to mimic legitimate communications and websites convincingly. To counteract these threats, individuals and organizations need to be vigilant and implement several defensive measures. These include:

- **Education and Awareness:** Regular training on how to recognize phishing attempts is crucial. Users should be educated about the signs of phishing emails and fraudulent websites, such as unusual URL structures, spelling errors, and unexpected requests for personal information.
- **Email Filters and Security Tools:** Implementing robust email filtering solutions can help detect and block phishing attempts before they reach the user's inbox. Additionally, security software can provide warnings about potentially malicious links.
- **Two-Factor Authentication (2FA):** Enabling 2FA adds an extra layer of security, requiring not just a password but also a second form of verification, such as a code sent to a mobile device.
- **Regular Software Updates:** Keeping browsers, operating systems, and other software updated ensures that the latest security patches are applied, reducing the risk of exploitation through known vulnerabilities.
- **Phishing Simulations:** Conducting regular phishing simulations can help users practice recognizing and responding to phishing attempts in a controlled environment, enhancing their ability to detect real threats.
- **Verification of Suspicious Emails:** Encouraging users to verify the authenticity of suspicious emails by contacting the presumed sender through a different communication channel can prevent successful phishing attacks.

By understanding the mechanics of website and email phishing, and implementing comprehensive security measures, both individuals and organizations can significantly reduce the risk of falling victim to these increasingly sophisticated attacks.

5.1 Zphisher

Zphisher is an open-source tool widely used for performing phishing attacks, which are attempts to obtain sensitive information such as usernames, passwords, and credit card details by disguising as a trustworthy entity on the internet.

Zphisher exploits social engineering techniques to trick victims and gain access to their confidential information and steal credentials to access various services. It's designed to be user-friendly even for those without deep technical knowledge. One of the distinguishing features of Zphisher is its simple and intuitive user interface, which allows users to configure and launch phishing attacks with just a few commands.

It offers a wide range of pre-configured phishing templates for various popular services like PayPal, Facebook, Google, Instagram, Twitter, Netflix, and many others. These templates realistically mimic the login pages of these services, increasing the likelihood of a successful attack.

In addition to the variety of available templates, Zphisher also has the ability to make phishing pages accessible over the internet, bypassing local network restrictions. This tool also allows customization of phishing templates in order to adapt them to specific scenarios or targets.

Despite its capabilities, it is important to emphasize that the use of Zphisher, or any other phishing tool, for illegal purposes is strictly prohibited and punishable by law. In this project, it has been used just for educational purposes and the various attempts have been done only among the components of the group.

5.1.1 The Test

Zphisher has been executed on a Kali Linux virtual machine on the Google's cloud platform using Google's cloud shell. Starting the application, the interface is the following:

Figure 17: Zphisher console interface

With this menu, it's possible to choose the type of service that we want to replicate. Suppose to choose Twitch (option 11), the program will return a set of URLs that will redirect the user to a fake login page extremely similar to the original one:

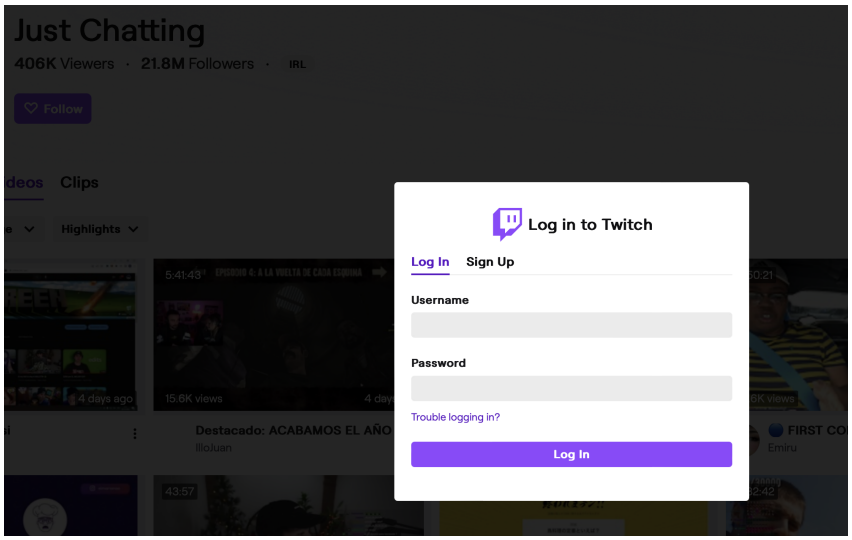


Figure 18: Twitch fake login page

Once the URL is generated, on the attacker's side, it's possible to:

- Spoof the victim's IP address
- Steal credentials

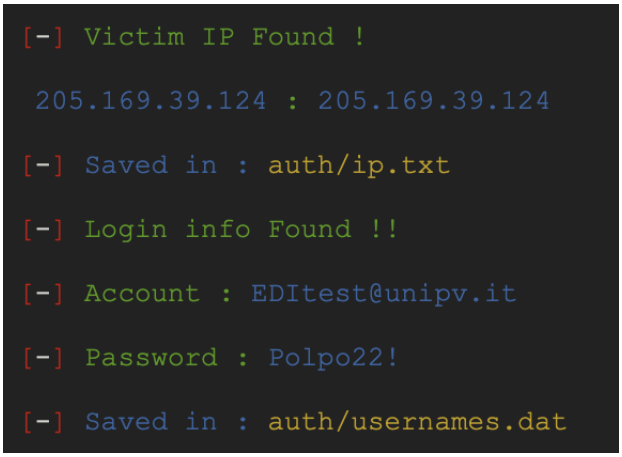


Figure 19: Attacker's interface showing captured data

Listing 1: Zphisher Output

```
vito_giacalone01@cloudshell:~/cloudshell_open/zphisher-4/auth$ more usernames.dat
    ip.txt
::::::::::::::
usernames.dat
::::::::::::::
Twitch Username: EDITest@unipv.it Pass: Polpo22!
::::::::::::::
Netflix Username: EDITest@unipv.it Pass: boom123!
```

```
:::::::::::::::
ip.txt
:::::::::::::::
IP: 151.82.112.44
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:126.0) Gecko/20100101
    Firefox/126.0

IP: 205.169.39.124
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15
(KHTML, like Gecko) Version/17.4.1 Safari/605.0.15
```

As it's possible to see in the third line of listing 1, the file **usernames.dat** contains the stolen username and password of the service. Another file, **ip.txt**, contains the spoofed IP of the victim and other details like the user agent:

- Operating system: MacOS Catalina
- Browser engine: AppleWebKit/605.1.15
- Browser: Safari 17.4.1

Using Safari, it's possible to access the fake website directly. But by changing the browser and using Mozilla Firefox, we immediately get a warning message. This also highlights the importance that Mozilla places on fighting phishing.

Analyzing the fake web page using Mozilla's developer tools, the only suspicious thing is the fact that a big platform like Twitch receives only 1 HTML and 1 embedded object:

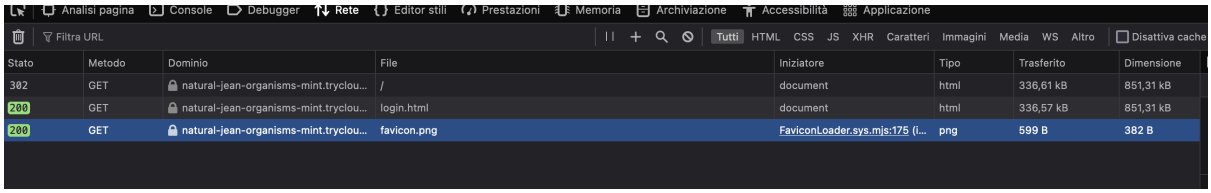


Figure 20: Developer tools showing minimal embedded objects

In fact, inspecting the official web page there are about 100 embedded objects; only an expert user, through analysis, can understand if the web page might be fake or not.

5.2 Mail Phishing

Typically, software like Zphisher is used to perform authorized penetration tests inside a company by sending phishing emails to its employees. These emails are used to make the employees conscious of the risks that arise from paying little attention to the received emails. These malicious emails are often ”socially engineered” (for example, the message contains words like ”urgent communication”) to trick the reader into believing that the email is an official communication.

An example of a phishing email is the following one:

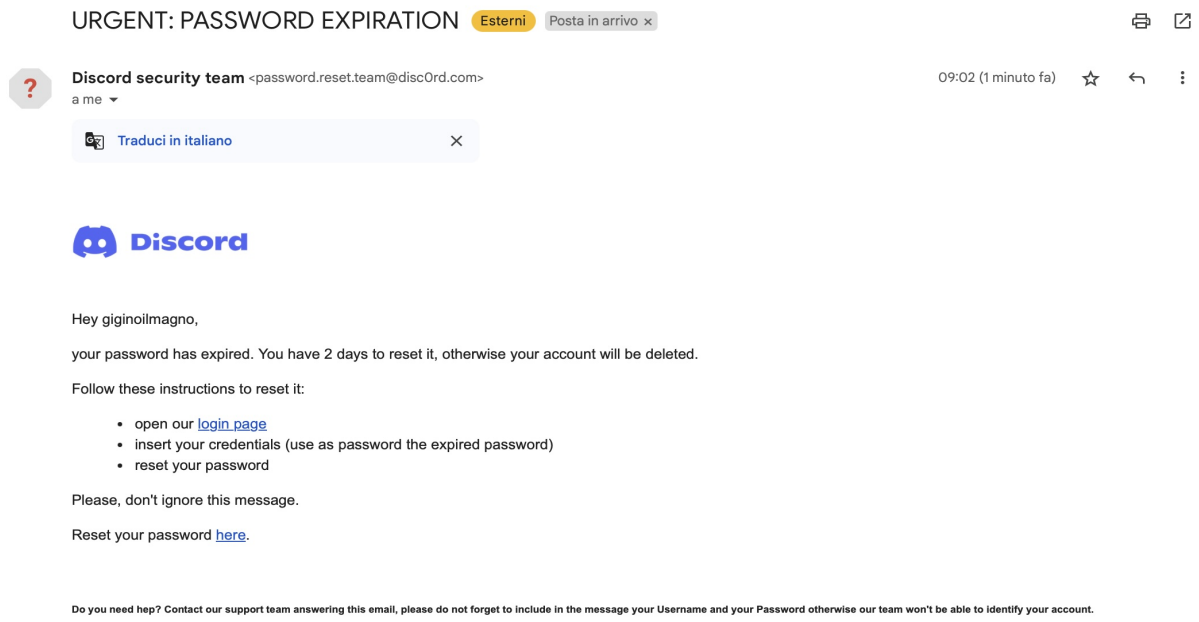


Figure 21: Example of a phishing email

At first sight, it seems like an official communication, but there are a few things to which we must pay attention:

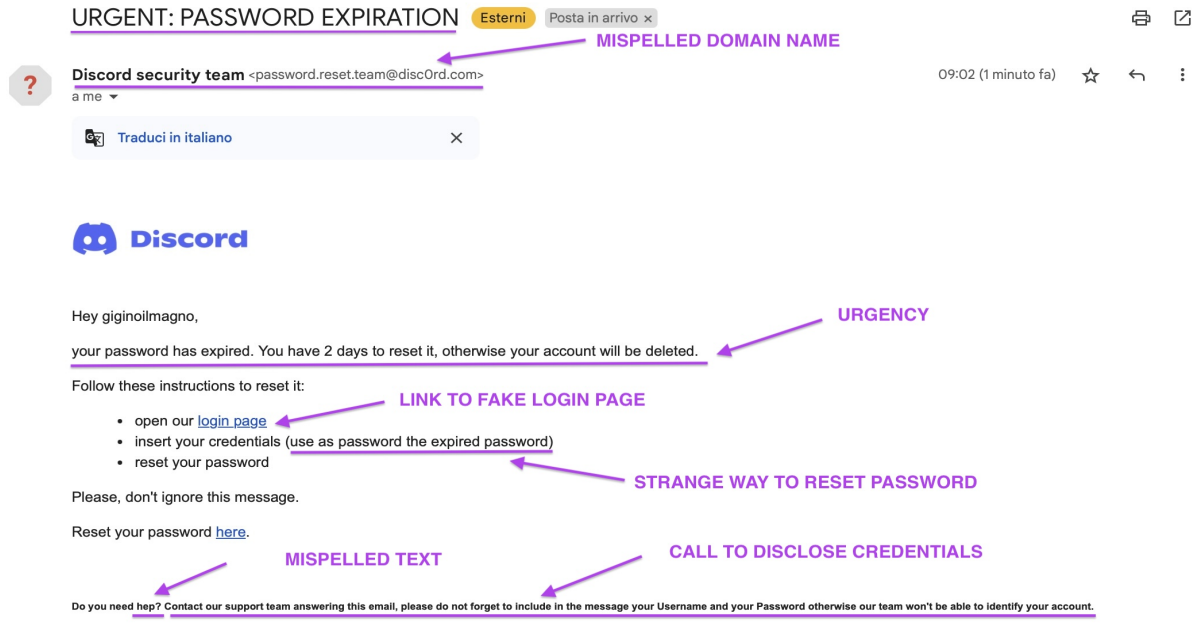


Figure 22: Key points of a phishing email

- **Sender's email address:** The sender's email address is password.reset.team.@disc0rd.com. It's important to notice that **discord** is spelled with a 0 instead of an **o**, **disc0rd**.
- **Urgent message:** The subject line says **URGENT: PASSWORD EXPIRATION**. This provokes a sense of urgency for the recipient to act quickly without paying too much attention to the content of the message.
- **Threat of negative consequences:** The message says that the account will be deleted if the password is not updated within 2 days. This type of threat is typical of phishing emails to instill fear and provoke immediate action.
- **Request for personal information:** The instructions include entering credentials, including the expired password. Companies never ask their users to provide passwords or sensitive data via email.
- **Suspicious link:** The link associated with the words **login page** and **here** can be malicious and can redirect to a fake website designed to steal the credentials.
- **Spelling and grammar errors:** Errors like **do you need hep?** instead of **do you need help?** are common in phishing attacks.

Trying to click on the link to reset the password, there are two possible scenarios:

- If the browser we're using adopts strong security measures, we will get an alert message:

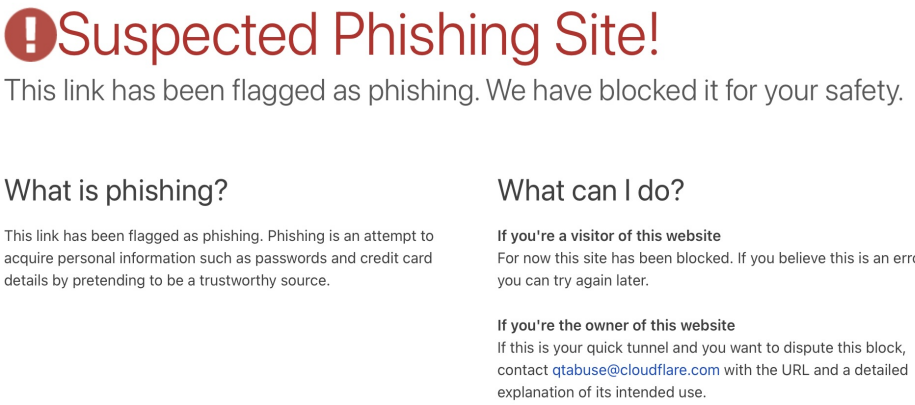


Figure 23: Security alert blocking phishing attempt

- If the browser we're using doesn't adopt strong security measures, we will not have any alert message and will be redirected to the fake login page:

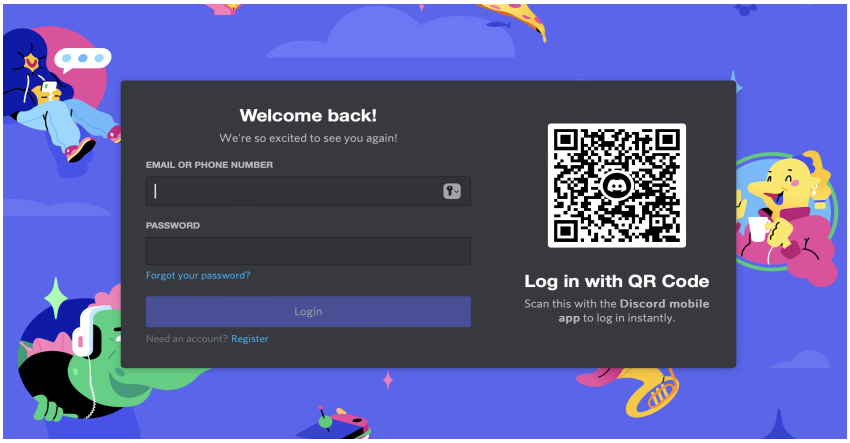


Figure 24: Fake Discord login page

Here there are the details of the email message:

Listing 2: Email Details

```
Delivered-To: vito.giacalone01@universitadipavia.it
Received: by 2002:a2e:9991:0:b0:2ea:84ce:214f with SMTP id w17csp626327lji;
        Fri, 7 Jun 2024 00:02:01 -0700 (PDT)
X-Google-Smtp-Source: AGHT+IGavnveIZXz5OUlL09wPjT9Svwu4Mz4NamAb7d0Bec9R
2IluuV+M2EjAiBM/6ov4UF1KVPEX-
Received: by 2002:a05:6a00:2d06:b0:702:78a2:8d26 with SMTP
id d2e1a72fcca58-7040c75b10bmr1599235b3a.33.1717743721169;
        Fri, 07 Jun 2024 00:02:01 -0700 (PDT)
ARC-Seal: i=1; a=rsa-sha256; t=1717743721; cv=none;
        d=google.com; s=arc-20160816;
        b=NR+ggwtEgdpe6AHVCVkKfC9sUZqE4uSH5vLJtrOy5udrQ/oOa6n45EPJcUEKf/9Ghp
         667Csjz9h0cSjW9mhP3FpU86a9eqgaiiVQj1fv6AmooSlwP1rEVJZSQg9XzcJjX+7nLv
         v9NIt1KyNlOwdkomZsnFdByWukZF3igUD5S7V+dJOenRi0Ud1zZoBHc9KDNjgCcmJ50g
         jKLnaZMzOv70MQ+JWIqm7fUBgrGKKgOD/JXtY+7Rp/YYGXmm9zGHSx9AX7Np7K/ARcc2
         8YqlaY3r0lFzpfFliwxLpqQiAQYmOGMu19IiFPq7tcaEwLhP3rlfDjnmazpg8zZuXQYG
         i9fQ==
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com; s=arc
    -20160816;
        h=date:message-id:reply-to:errors-to:importance:from:subject:to;
        bh=JLlULTieSxl4GWNttgaiHjqKuBjkCOuYO60qT7vXdlE=;
        fh=gwmYFKFHeLkK6CwQaT3IPA+RO0xsg7o0mRPGgbVNHAg=;
        b=WVsBdIn+Nov6CCvEABNdCzO7aCwUA5/s4IvqEopJ25tM4Cyf+2s52kE4HTBgt3lZWk
         kq9EgM63LvZrqo6yE3Q3yK9ks0ZgManVaGJKg+Kz00feoI4g3qOsIpEk0WgpXZ1pu0HE
         +ISDRWpk9wkCcdiMoPwAOxp5T/BRmJy4NYlzVqKXeVCZg6u6e7lkMtp8+sIw7tNu2ZUM
         jdmKvMDte+NQXCuMC8JnnNeSKYHqAkGOyVTa5k832RPCG6ceR+ZcnmKY+aN3KjtE6/Gr
         USDTwLE2FgEhRoMB+R/vbDHe0qbU2gK5uBzwDMJKDQv67Sgl6QVMldYfklQfS26r6M0A
         2QWw==;
        dara=google.com
ARC-Authentication-Results: i=1; mx.google.com;
       spf=neutral (google.com: 114.29.236.247 is neither permitted nor denied by
           best guess record for domain of password.reset.team@disc0rd.com) smtp.
           mailfrom=password.reset.team@disc0rd.com
Return-Path: <password.reset.team@disc0rd.com>
Received: from emkei.cz (emkei.cz. [114.29.236.247])
        by mx.google.com with ESMTPS id 41be03b00d2f7-6de20831de8si2605153a12
            .150.2024.06.07.00.02.00
        for <vito.giacalone01@universitadipavia.it>
        (version=TLS1_3 cipher=TLS_AES_256_GCM_SHA384 bits=256/256);
        Fri, 07 Jun 2024 00:02:01 -0700 (PDT)
Received-SPF: neutral (google.com: 114.29.236.247 is neither permitted nor denied
    by best guess record for domain of password.reset.team@disc0rd.com) client-ip
    =114.29.236.247;
Authentication-Results: mx.google.com;
       spf=neutral (google.com: 114.29.236.247 is neither permitted nor denied by
           best guess record for domain of password.reset.team@disc0rd.com) smtp.
           mailfrom=password.reset.team@disc0rd.com
Received: by emkei.cz (Postfix, from userid 33) id 3BE6E1345; Fri,
  7 Jun 2024 09:01:59 +0200 (CEST)
To: vito.giacalone01@universitadipavia.it
Subject: URGENT: PASSWORD EXPIRATION
From: Discord security team <password.reset.team@disc0rd.com>
X-Priority: 3 (Normal)
Importance: Normal
Errors-To: password.reset.team@disc0rd.com
Reply-To: password.reset.team@disc0rd.com
Content-Type: text/html; charset=utf-8
Message-Id: <20240607070159.3BE6E1345@emkei.cz>
Date: Fri,
```

```
  7 Jun 2024 09:01:59 +0200 (CEST)

<p><img src="https://1000logos.net/wp-content/uploads/2021/06/Discord-logo.png" alt
    ="dl" width="154" height="87" /></p>
<p>Hey giginoilmagno,</p>
<p>your password has expired. You have 2 days to reset it, otherwise your account
    will be deleted.</p>
<p>Follow these instructions to reset it:</p>
<ul>
<li>open our <a href="https://communication-benjamin-gt-mats.trycloudflare.com">
    login page</a></li>
<li>insert your credentials (use as password the expired password)</li>
<li>reset your password</li>
</ul>
<p>Please, don't ignore this message.</p>
<p>Reset your password <a href="https://communication-benjamin-gt-mats.
    trycloudflare.com">here</a>.</p>
<p> </p>
<h6>Do you need help? Contact our support team

 answering this email, please do not forget to include in the message your Username
     and your Password otherwise our team won't be able to identify your account.</
    h6>
```

In the message, the SPF check is neutral as it's possible to see in the field **Received-SPF**. This means that Google wasn't able to establish if the sender's IP was authorized to send emails. Therefore, we don't have any definitive information about email spoofing. Typically, this result comes from the fact that there's no specific SPF record for the sender's domain or the SPF record is not properly configured. To have complete protection against spoofing, it's necessary to implement SPF correctly, and it should be combined with DKIM and DMARC.

In this case, there is neither a DKIM signature nor a DMARC check.

6 Nmap

Nmap (Network Mapper) is an open-source tool used for network discovery and security auditing.

After studying and exploring the various functions of Nmap, I decided to test the IDLE scan and the Operating System Detection using Nmap. The first function helps to identify hosts and services on a computer network by sending packets and analyzing the responses. Commonly used for network inventory, managing service upgrade schedules, and monitoring host or service uptime, Nmap is valuable for network administrators and security professionals. The second feature is a remote OS detection using TCP/IP stack fingerprinting. This feature is used, for example, to detect breaches and to implement the best solution to ensuring the best security.

In the next sub-section, I will show the steps to do an IDLE scan with Nmap, the insights obtained from the scan results, and its implications in ethical hacking.

6.1 IDLE Scan

Idle scan is an advanced scan method that allows for a truly blind TCP port scan of the target. A truly blind TCP port scan means no packets are sent to the target from your real IP address. Instead, a unique side-channel attack exploits predictable IP fragmentation ID sequence generation on the zombie host to gather information about the open ports on the target. IDS systems will display the scan as coming from the zombie machine you specify. The idle scan is based on the following three facts. As we know, a possible way to determine whether a TCP port is open is to send a SYN packet to the port. The target machine will respond with a SYN/ACK packet if the port is open and RST if the port is closed. A machine that receives an unexpected SYN/ACK packet will respond with an RST. An unexpected RST will be ignored. Every IP packet on the internet has a fragment identification number (IPID). Since many operating systems simply increment this number for each packet they send, probing for the IPID can tell an attacker how many packets have been sent since the last probe.

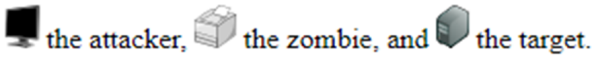


Figure 25: legenda

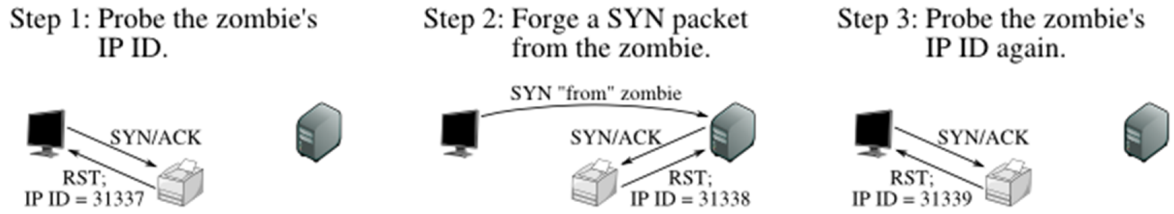


Figure 26:

The first step is to probe the IPID of the zombie system. The attacker sends a SYN/ACK to the zombie. Since the zombie does not expect the packet, it sends back a RST with an IPID (Figure 26).

The second step is to forge a SYN packet from the zombie to the target system. The target sends a SYN/ACK in response to the SYN that appears to come from the zombie. Since the zombie does not expect the packet, it sends back a RST and so it increments its IPID in the process.

The third step is to probe the zombie's IPID again. The attacker sends a SYN/ACK to the zombie again. The RST packet of the zombie has an IPID which is increased by two since the first step, so the port is open.

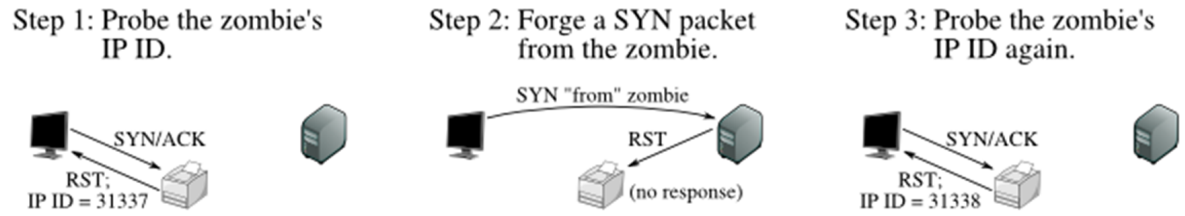


Figure 27:

If the target port is closed (Figure 27):

The first step is to probe the IPID of the zombie system. The attacker sends a SYN/ACK to the zombie. Since the zombie does not expect the packet, it sends back a RST with an IPID.

The second step is to forge a SYN packet from the zombie to the target system. The target sends a RST because the port is closed in response to the SYN that appears to come from the zombie. The zombie ignores the unexpected RST, so its IPID does not change.

The third step is to probe the zombie's IPID again. The attacker sends a SYN/ACK to the zombie again. The RST packet of the zombie has an IPID which is increased by only one since the first step, so the port is not open.

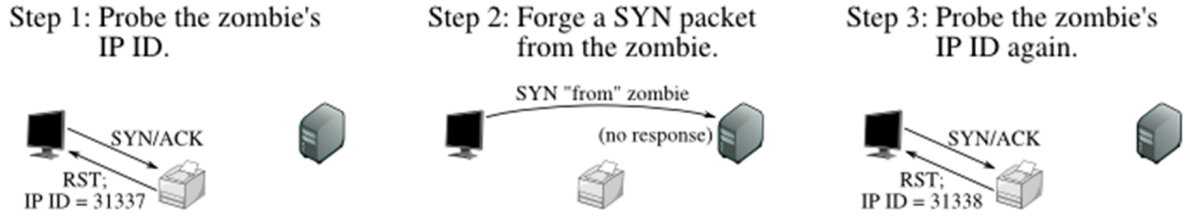


Figure 28:

If the target port is filtered(Figure 28):

The first step is to probe the IPID of the zombie system. The attacker sends a SYN/ACK to the zombie. Since the zombie does not expect the packet, it sends back a RST with an IPID. The second step is to forge a SYN packet from the zombie to the target system. The target, filtering its port, ignores the SYN that appears to come from the zombie. The zombie is unaware that anything happens, so its IPID remains the same. The third step is to probe the zombie's IPID again. The attacker sends a SYN/ACK to the zombie again. The RST packet of the zombie has an IPID which is increased by only one since the first step, so the port is not open. From the point of view of the attacker, the filtered port is identical from a closed port.

6.2 IDLE Test

To be able to perform an idle scan, we first need to have a zombie computer on the network, which has incremental IPID sequencing. Hopefully, we have an Nmap script to help us find a computer appropriate to become a zombie. The script is: `ipidseq.nse`. To use the script, I used the command `nmap --script ipidseq <IP block> --top-ports 2` where our IP block is '192.168.178.0/24'.

```
         (federicokali)-[~]
   $    ls /usr/share/nmap/scripts/ipid*
/usr/share/nmap/scripts/ipidseq.nse

         (federicokali)-[~]
   $    sudo nmap --script ipidseq 192.168.178.0/24 --top-ports 2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-06 23:12 CEST
Nmap scan report for 192.168.178.1
Host is up (0.0012s latency).

PORT   STATE  SERVICE
23/tcp closed telnet
```

```
80/tcp open   http

Host script results:
|_ipidseq: All zeros

Nmap scan report for 192.168.178.24
Host is up (0.78s latency).

PORT   STATE  SERVICE
23/tcp closed telnet
80/tcp closed http

Host script results:
|_ipidseq: All zeros

Nmap scan report for 192.168.178.39
Host is up (0.0069s latency).

PORT   STATE    SERVICE
23/tcp filtered telnet
80/tcp open     http

Host script results:
|_ipidseq: Incremental!

Nmap scan report for 192.168.178.40
Host is up (0.0013s latency).

PORT   STATE  SERVICE
23/tcp closed telnet
80/tcp open   http

Host script results:
|_ipidseq: All zeros

Nmap scan report for 192.168.178.46
Host is up (0.40s latency).

PORT   STATE  SERVICE
23/tcp closed telnet
80/tcp closed http

Host script results:
|_ipidseq: All zeros

Nmap scan report for 192.168.178.60
Host is up (0.00050s latency).

PORT   STATE  SERVICE
23/tcp closed telnet
80/tcp open   http

Host script results:
|_ipidseq: All zeros

Nmap scan report for 192.168.178.86
Host is up (0.0094s latency).

PORT   STATE  SERVICE
23/tcp closed telnet
80/tcp closed http
```

```
74
75  Host script results:
76  |_ipidseq: Incremental!
77
78  Nmap scan report for host.docker.internal (192.168.178.88)
79  Host is up (0.0016s latency).
80
81  PORT   STATE    SERVICE
82  23/tcp filtered telnet
83  80/tcp filtered http
84
85  Nmap scan report for 192.168.178.108
86  Host is up (0.36s latency).
87
88  PORT   STATE  SERVICE
89  23/tcp closed telnet
90  80/tcp closed http
91
92  Host script results:
93  |_ipidseq: All zeros
94
95  Nmap scan report for 192.168.178.109
96  Host is up (0.015s latency).
97
98  PORT   STATE  SERVICE
99  23/tcp closed telnet
100 80/tcp closed http
101
102 Host script results:
103 |_ipidseq: Randomized
104
105 Nmap done: 256 IP addresses (10 hosts up) scanned in 41.63 seconds
```

The terminal output shows the results and explain to us that: Each block starts with ”Nmap scan report for,”followed by the IP address of the host and optionally the hostname if available. The status of the host (”up” if reachable, ”filtered” if some ports are blocked, and ”closed” if ports are closed) is displayed, along with the host’s latency (the time taken to respond to the scan). For each host, there is a list of ports and their own status (open, closed or filtered), along with the associated service (if known, is shown). In addition, is displayed the IPID (IP fragmentation ID) of the hosts and provides information about their behavior. The results can be categorized as follows:

1. **All zeros**: it means the IPID is not incremented and remains constantly at zero. This could indicate an inactive host or an operating system that does not use IPID.
2. **Incremental**: An ”Incremental” IPID pattern indicates that the IPID is sequentially incremented with each of the packet sent. Hosts, with this pattern, are a very good candidates to become zombies in an idle scan, as they allow for reliable prediction of the next IPID.
3. **Randomized**: If the IPID is ”Randomized,” it means that the IPID is generated randomly. This may indicate a scanning mitigation strategy or a particular operating system configuration that generates random IPIDs. Hosts with randomized IPIDs are not suitable for an idle scan, as they make it difficult to predict the next IPID.

As a first step, in Figure 29, I executed the command: `sudo nmap -sI 192.168.178.1 -Pn -n 192.168.178.60` with the first IP address of the zombie relative to my machine, which has an all-zeros IPID sequence, and the second IP is my target. So, as you see, Nmap reports that the zombie’s IPID sequence class is all zeros, so we should find another system.

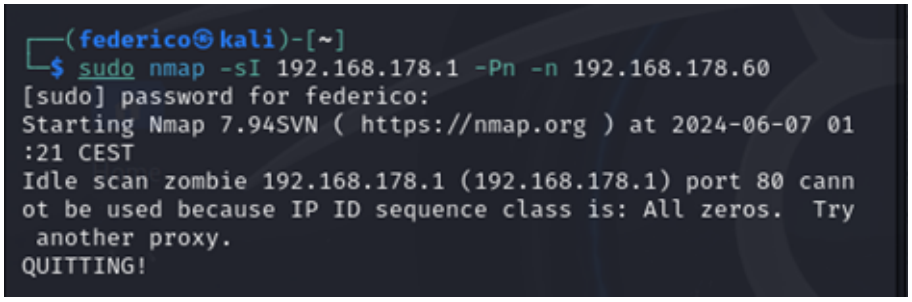


Figure 29:

For the second attempt, in Figure 30, I used a system with the IP address `192.168.178.109`, which has a randomized IPID sequence class. As you can see again, it's just not suitable to be a zombie.

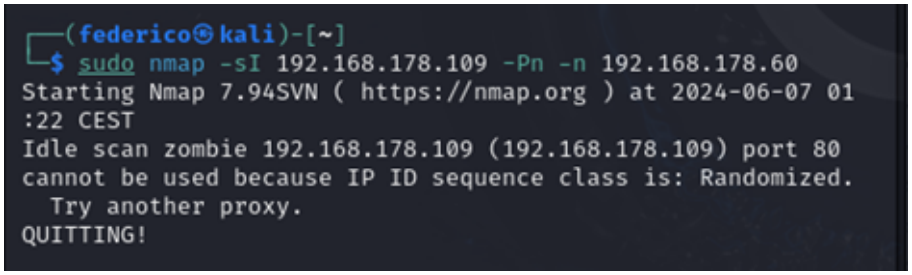


Figure 30:

Finally, in Figure 31, I used a system with the IP address 192.168.178.39, which has an incremental IPID sequence class. I scanned the top three ports, and the scan was completed successfully.

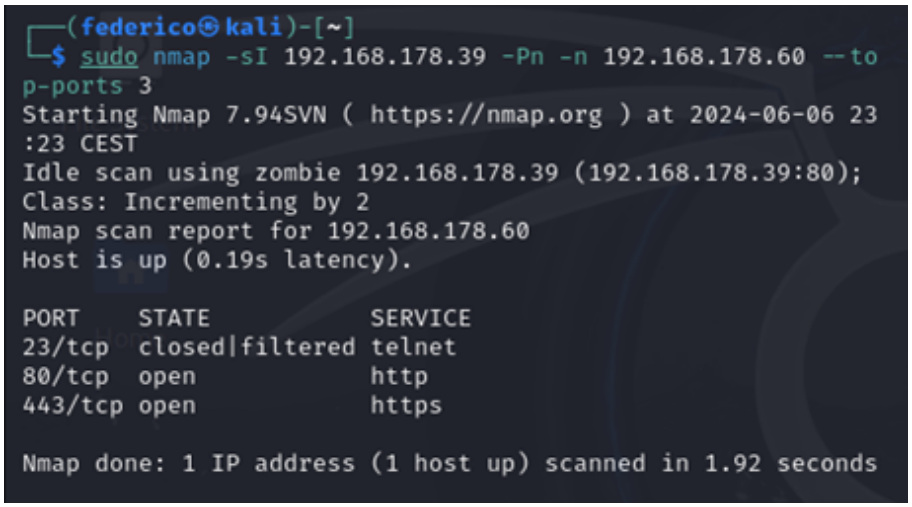


Figure 31:

To compare the results, I wanted to perform a SYN scan in another terminal screen with the same conditions (Figure 32).

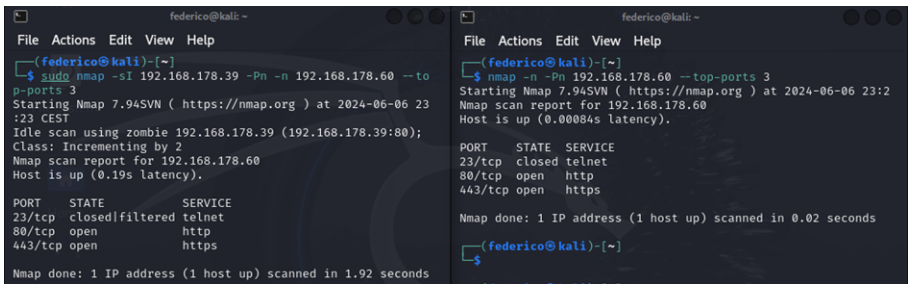


Figure 32:

In both scans, ports 443 and 80 are open, while according to the SYN scan, port 23 is closed. This indicates that the idle scan cannot distinguish the closed port from the filtered port.

Running the last query with the ”reason” option again, ports 443 and 80 are flagged as open because the IPID has changed each time. Since the IPID has not changed for port 23, it is flagged as closed or filtered. This is shows in Figure 33.

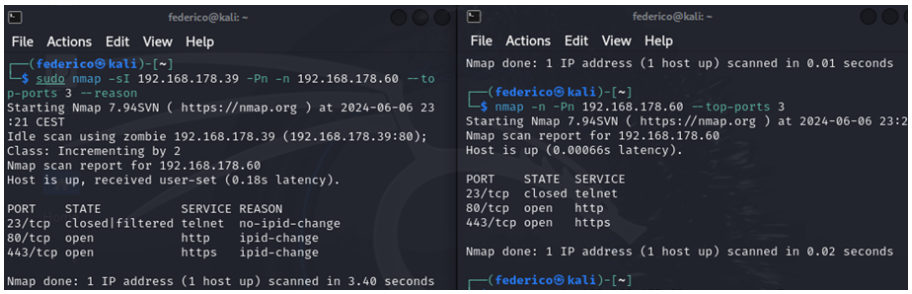


Figure 33:

Tag	**Descrizione**
-n	Potrebbe indicare un file di output personalizzato. Comunemente usato per salvare l’output della scansione.
-nP	Disabilita la scansione ping. Nmap non invia pacchetti di ping per determinare la raggiungibilità degli host prima della scansione delle porte.
-sI	Specifica l’IP dell’host ”zombie” per eseguire una scansione idle. Nasconde l’origine della scansione utilizzando un host intermediario.
–top-ports 3	Limita la scansione alle prime 3 porte più comuni su ogni host, riducendo il tempo di scansione.
–top-ports 3 –reason	Aggiunge informazioni sulla ragione per cui una porta è aperta, chiusa o filtrata nell’output della scansione. Include dettagli sulla risposta del servizio, come un reset TCP.

Table 1: Nmap tag references

6.3 IDLE Conclusion

In conclusion, using IDLE scan through Nmap can discover many cybersecurity issues within a network:

1. **False positives / False negatives:** IDLE scan can generate false positives or false negatives during the detection of the state of ports on the network. For example, a filtered port might be wrongly identified as closed or open, or vice versa.
2. **Difficulty in detecting attacks:** since IDLE scan can be more prudent compared to traditional scanning techniques, it may be harder for defence systems to identify and respond to attacks.
3. **Abuse for malicious activities:** attackers can discover IDLE scan to carry out malicious activities such as identifying vulnerable devices on the network, detecting open ports for future attacks or mapping network topology for other purposes.
4. **Difficulty in responding to attacks:** due to the stealthy nature of IDLE scan, defenders might struggle to detect and promptly respond to ongoing attacks, compromising the overall security of the network.

6.4 Operating System Detection

Another of Nmap's best-known features is remote OS detection using TCP/IP stack fingerprinting. Essentially, Nmap sends a series of TCP and UDP packets to the target host and scans each bit of the responses after performing many tests, such as TCP ISN sampling, TCP option support and ordering, IPID sampling, and the initial window size check. Nmap compares the results to its Nmap OS database that contains more than 2000 known OS fingerprints and prints out the OS details if there's a match. Each fingerprint includes a textual description of the OS and a ranking, which provides the vendor name, the underlying OS, the OS generation, and other important information.

I started using the command: `nmap -n -sS 192.168.178.88 --top-ports 1000 -O`, I used SYN scan (-sS) for this test. The target system is my workstation (where is running my VM). I choose the 1000 ports to make the query faster. In addition, I put an uppercase -O for OS detection. Moreover, -Pn to disabling ping.

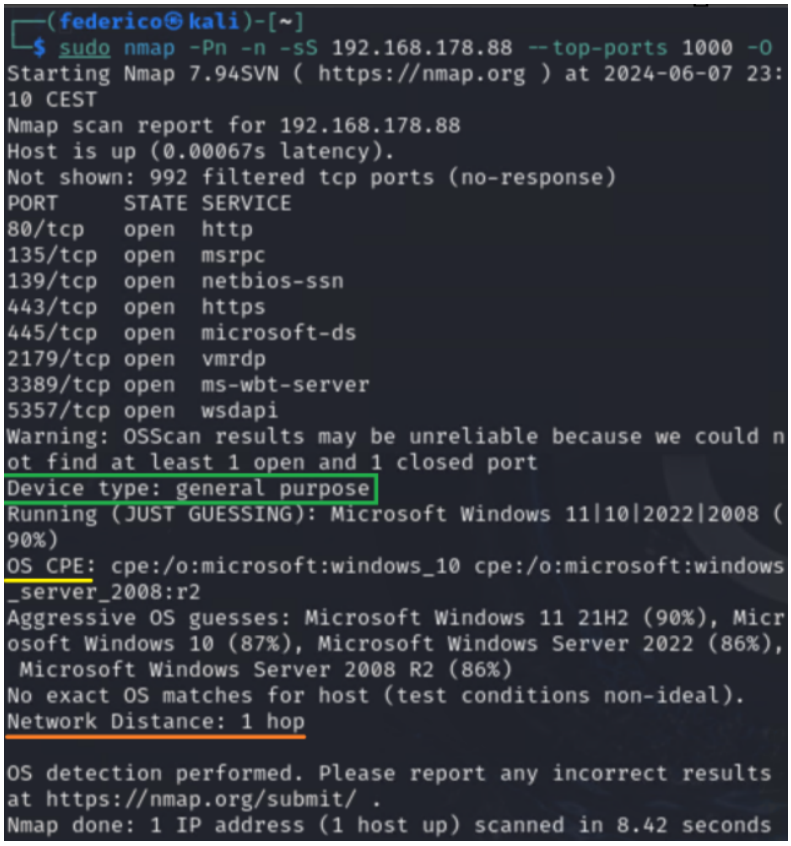


Figure 34:

In Figure 34, we can see that the device type is "general purpose", which means that the device is not specialized for only one purpose but it could be used to perform different tasks. Moreover, **OS CPE** is a standardized identifier used to describe the operating system of a device. It follows a structured format and provides specific details about the operating system, such as the vendor, product name, and version. This helps Nmap users to accurately identify and classify the operating systems of scanned devices. Finally, The **1 hop** indicates the number of intermediary devices, such as routers or switches, between the scanning host and the target host.

If the user want a more aggressive Namp scan and consequently a more accurate result, you can use `nmap -n -sS <target> -top-ports 1000 -O -osscan-guess`.

Trying to change the target for running this command, I chose my personal computer, and as soon as I executed the command, I immediately received a notification from my antivirus warning me that a network attack had been detected. In the figure below, you will find a couple of screenshots related to the notification.

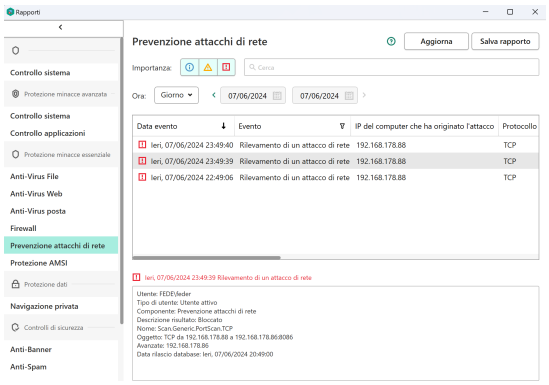


Figure 35:

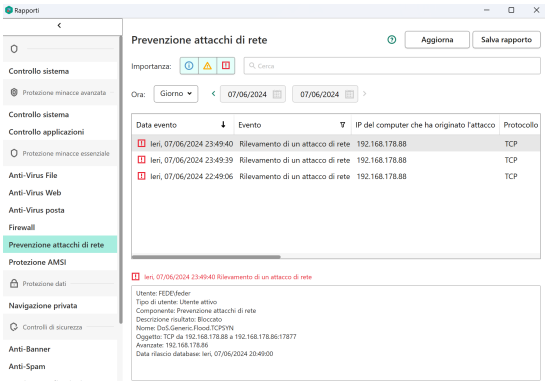


Figure 36:

Date and Time	Description	IP Address	Protocol	Port	User	Status	Attack Type	Action	Details
Yesterday, 07/06/2024 23:49:40	Network attack detected	192.168.178.88	TCP	17877	FEDE\feder	Active User	Blocked: DoS.Generic.Flood.TCPSYN	DoS.Generic.Flood.TCPSYN	Blocked
Yesterday, 07/06/2024 23:49:39	Network attack detected	192.168.178.88	TCP	8086	FEDE\feder	Active User	Blocked: Scan.Generic.PortScan.TCP	Scan.Generic.PortScan.TCP	Blocked

Figure 37: saving report

From these screenshots (Fiures 35, 36, 37), we can observe crucial information. First of all, the **user** who send the command via Nmap, identified as ”FEDE/feder”.

The **description result** clearly states that it has been blocked. The **object** provides details on the suspicious activity, including the protocol used (TCP) and the IP addresses involved, and the destination port.

The **name** specifies the type of attack detected, which in this case is ”Scan.Generic.PortScan.TCP” or ”DoS. Generic.Flood.TCPSYN”, indicating respectively an attempt to scan TCP ports or a SYN Flood attack.

Finally, I tryed the command:
`nmap -n -sS <target> --top-ports 100 -O --osscan-guess`, where this last tag improve the accuracy of OS identification during a scan. It is important to explain that is only an hypotesis, may not always be 100% accurate. Moreover, another important information is related to the TCP/IP fingerprint: This refers to collecting information specific to the target's TCP/IP network behavior. By analyzing these details, nmap attempts to infer the operating system running on the target machine.

```
┌──(federico㉿kali)-[~]
└─$ sudo nmap -n -sS 192.168.178.40 --top-ports 100 -O --osscan-guess
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-08 00:54 CEST
Nmap scan report for 192.168.178.40
Host is up (0.0012s latency).
Not shown: 96 closed tcp ports (reset)
PORT    STATE    SERVICE
22/tcp  open     ssh
53/tcp  filtered domain
80/tcp  open     http
443/tcp open     https
Aggressive OS guesses: Linux 2.6.23 - 2.6.38 (96%), D-Link DIR-600 or DIR-645 WAP (Linux 2.6.33) (94%), Linux 2.6.22 - 2.6.36 (93%), Linux 2.6.8 - 2.6.30 (93%), DD-WRT v24-sp2 (Linux 2.6.34) (93%), DD-WRT v24-sp1 (Linux 2.4.36) (93%), Linux 2.6.32 (92%), Ubiquiti AirMax NanoStation WAP (Linux 2.6.32) (92%), Ubiquiti Pico Station WAP (AirOS 5.2.6) (92%), Linux 2.6.17 - 2.6.36 (92%)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=6/8%OT=22%CT=7%CU=31886%PV=Y%DS=2%DC=I%G=Y%TM=66638
OS:FBCB%P=x86_64-pc-linux-gnu)SEQ(SP=C9%GCD=1%ISR=D0%TI=Z%CI=Z%II=I%TS=8)SEQ
OS:(SP=C9%GCD=3%ISR=D0%TI=Z%CI=Z%II=I%TS=8)OPS(O1=M5B4ST11NW1%O2=M5B4ST11NW
OS:1%O3=M5B4NNT11NW1%O4=M5B4ST11NW1%O5=M5B4ST11NW1%O6=M5B4ST11)WIN(W1=16A0%
OS:W2=16A0%W3=16A0%W4=16A0%W5=16A0%W6=16A0)ECN(R=Y%DF=Y%T=40%W=16D0%O=M5B4N
OS:NSNW1%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=
OS:Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=A
OS:R%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=N)U1(R=Y%D
OS:F=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=4B1F%RUD=G)IE(R=Y%DFI=N%
OS:T=40%CD=S)

Network Distance: 2 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.83 seconds
```

Figure 38:

In Figure 38, I underlined in green the correct OS of the target.

6.5 Operating System Detection Conclusion

The use of Nmap for OS detection, in a Cybersecurity context, is essential to understand the network and identify potential flaws. Knowing the hosts' operating system allows administrators to understand which security patches need to be implemented, ensuring greater security. Furthermore, OS detection helps to more easily identify ongoing attacks or compromised devices, with better analysis and through a more targeted and effective response to threats.

7 Bettercap

Using BetterCap on Kali Linux

7.1 Introduction

Bettercap is a powerful, flexible, and easy-to-use network attack and monitoring tool for security professionals. It is commonly used on Kali Linux for network security assessments.

7.2 Overview

Bettercap is designed for network reconnaissance, man-in-the-middle attacks, and various other network-related attacks.

developed by Simone Margaritelli (@evilsocket).

7.2.1 Key features

7.2.2 Network Sniffing

Captures network traffic and supports various protocols such as HTTP, HTTPS, TCP, and UDP.

Can sniff credentials and cookies from unencrypted and poorly encrypted connections.

7.2.3 Man-In-The-Midle(MITM) Attacks

Supports ARP spoofing to intercept network traffic between two devices.

DNS spoofing to redirect traffic to malicious sites.

7.2.4 Proxy

HTTP/HTTPS proxy for monitoring and altering HTTP and HTTPS traffic in real-time.

7.2.5 Reconnaissance

Can scan and identify devices on the network.

Collects information like IP addresses, MAC addresses, and device manufacturers.

7.2.6 Wifi Attacks

Can perform de-authentication attacks to disconnect devices from Wi-Fi networks. Supports WPA/WPA2 handshaking and cracking.

7.2.7 Modular Design

Extensible with a modular design, allowing additional functionalities through scripts and plugins.

7.3 Installation

First, you have to open your terminal on Kali Linux. and right these commands (get sure first you have updated your application)

1-update your packages.

2-install bettercap

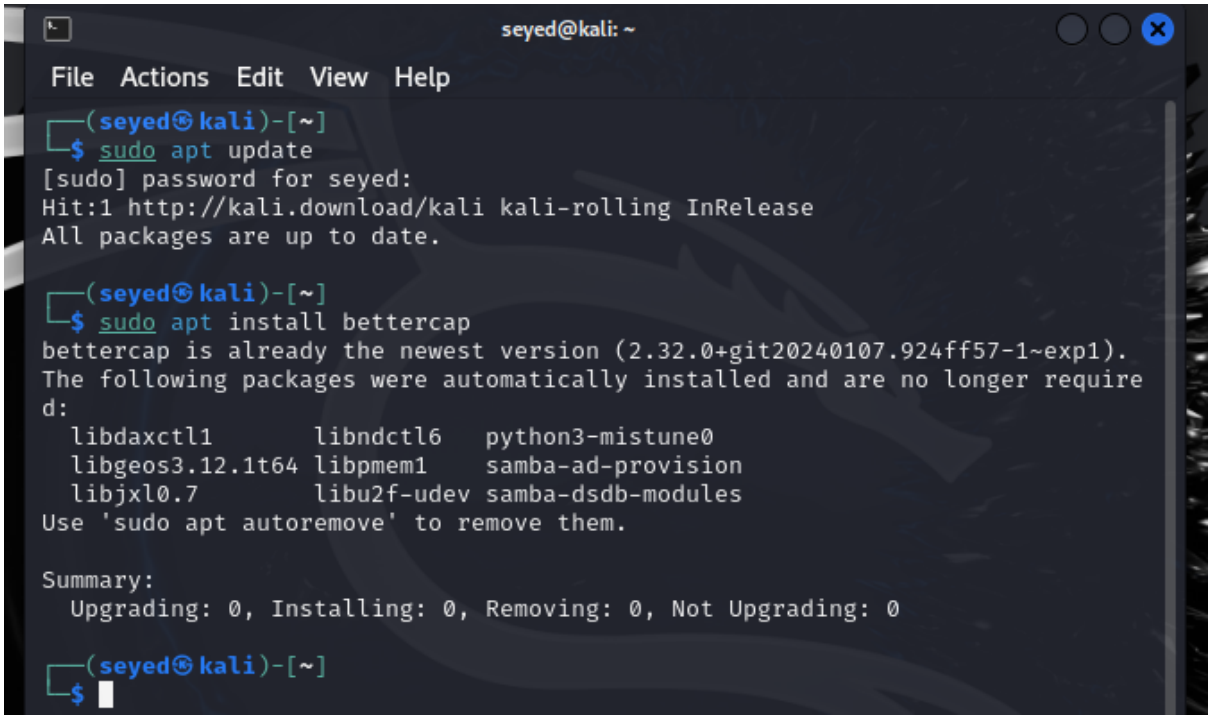
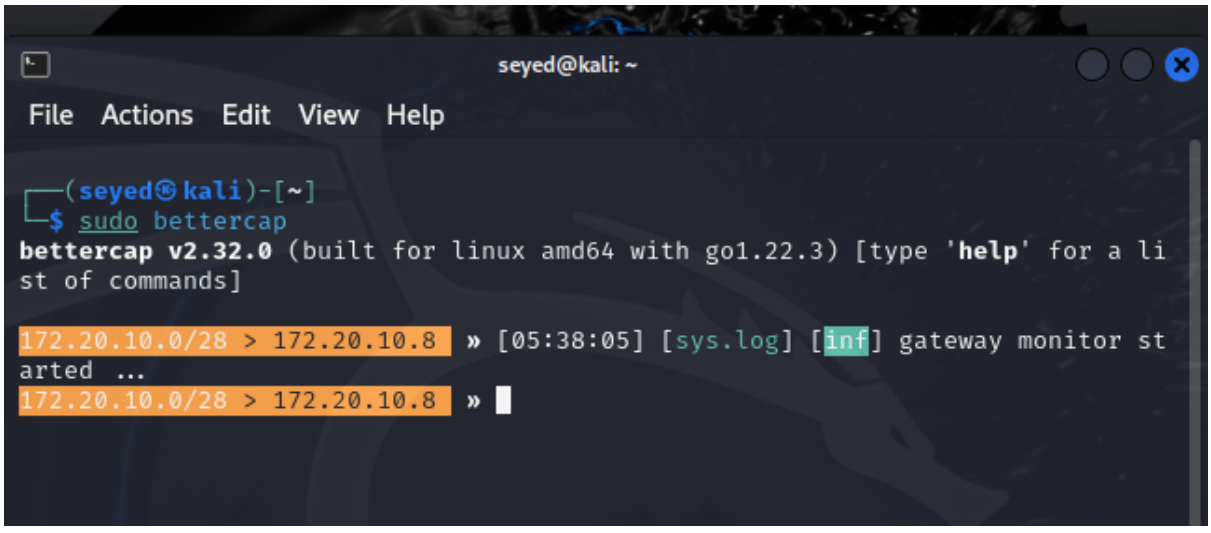


Figure 39: since I have already installed it

7.4 Run the Bettercap

Now, it is time to run it and elevate permissions.



7.5 HELP command

We use this command to see what we are doing and what tools are running. and also can show us available modules and options that we can do.

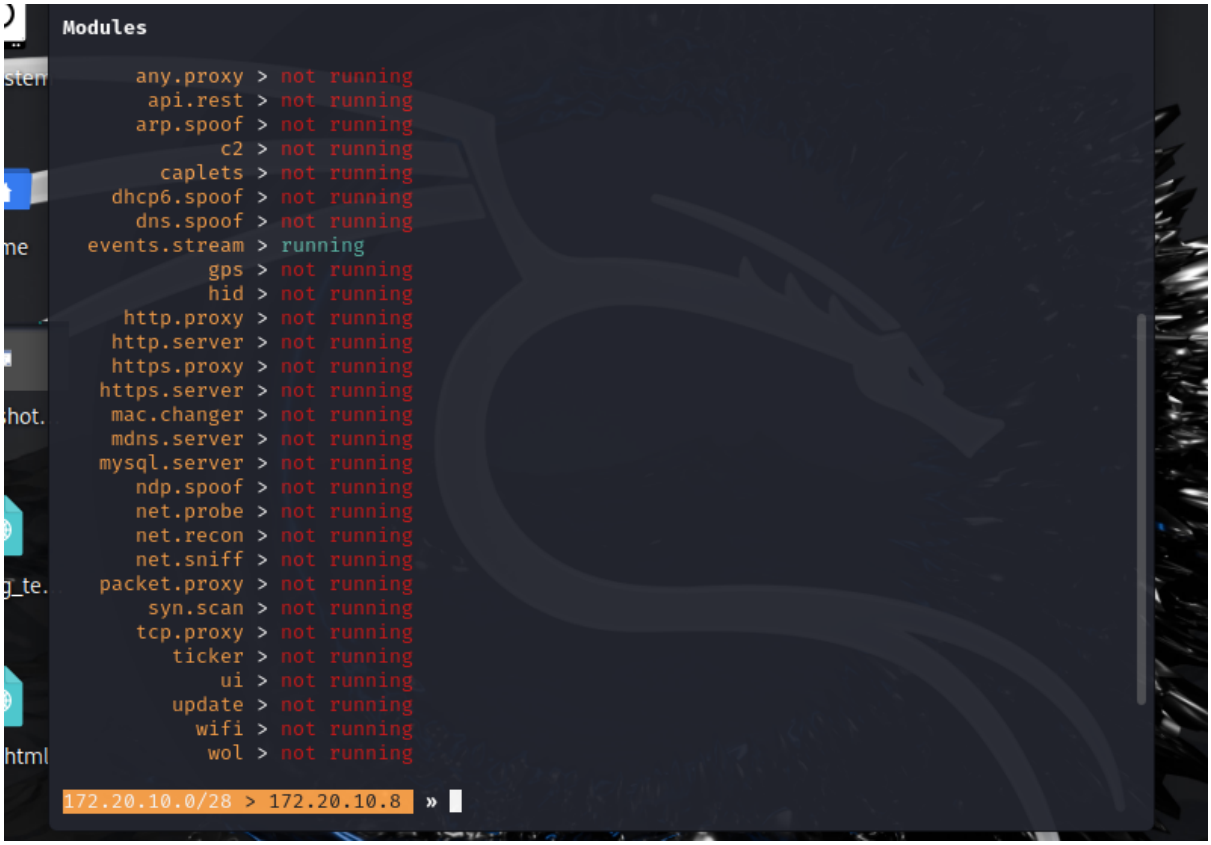
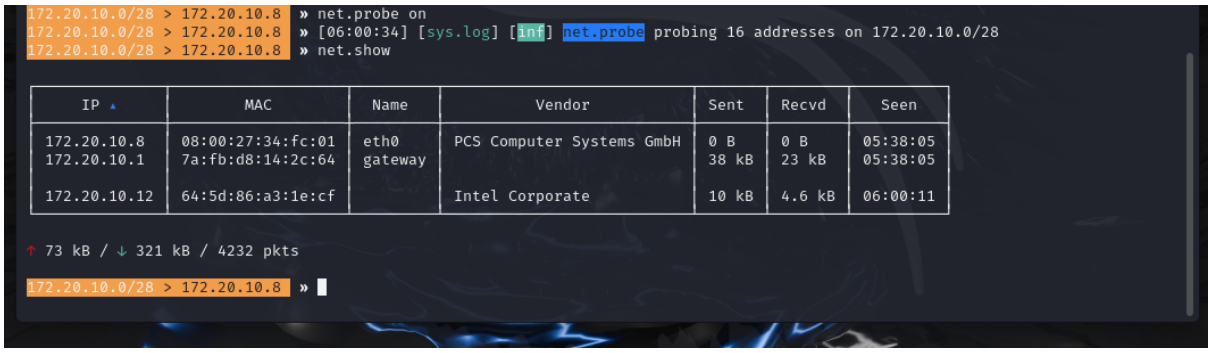


Figure 40: You can see currently which modules are running and which are not

7.6 Using modules to see IP and devices

First, you have to turn on the module **net.probe on**

Then, we want to see the table of devices which are connected to the wifi, their IP addresses, and available devices and also how many packets are received and sent and also last seen using the command: **net.show**



7.7 Tracking (man in the middle)

I want to target my host machine by configuring Bettercap to use the victim's browser for spoofing.

To achieve this, specify the IP address you want to spoof and the IP address of your gateway in Bettercap.

Then, configure our Kali Linux machine to act as a router. This setup will allow us to intercept and analyze the data traffic from any device attempting to connect, creating a man-in-the-middle scenario where we can monitor communications between both parties.(who listens on both sides)

-1 : **set arp.spoof.targets 172.20.10.3, 172.20.10.1**

-2 : **arp.spoof on**

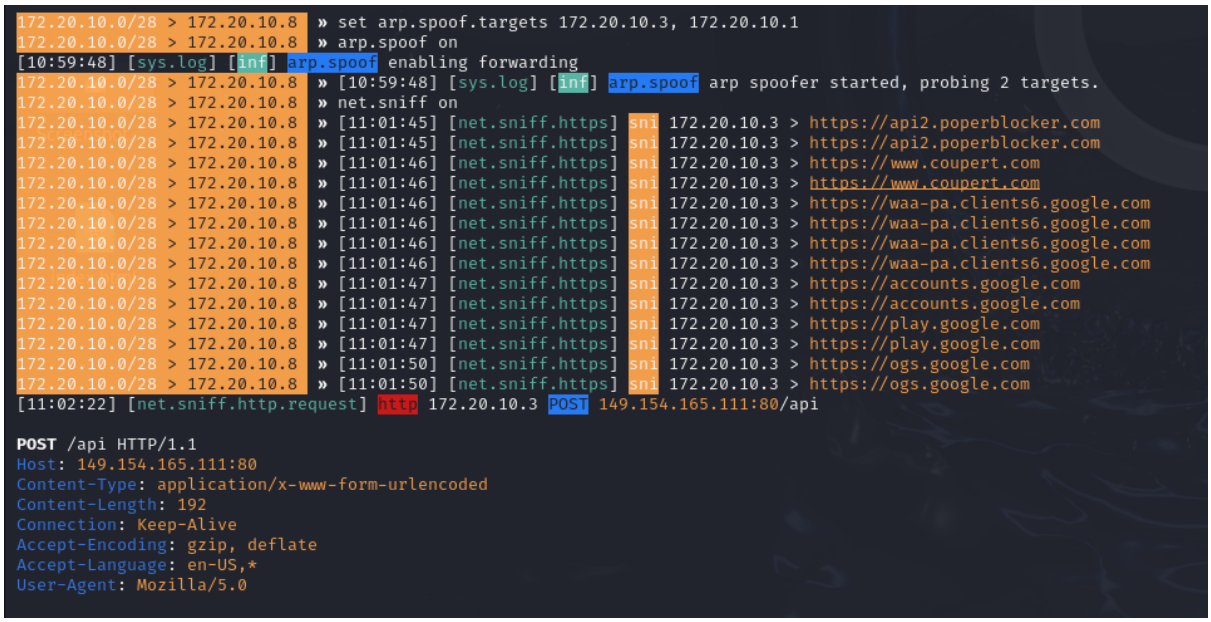


Figure 41: Post information

7.8 Checking the Browser(Victim)

When I search for something in my browser, it shows all domains of that website for example I searched Google shows requesting. This works on any website that our victim requested. and It doesn't matter it's running on HTTP and HTTPS we are still able to sniff it. Now it is very scary because attackers are able to see every single website you are visiting and possibly. customize a social engineering attack based on the websites that you are visiting

We are also able to see every traffic that is coming from this device or this target as well regardless If they visit a website themselves or they have an application running in the background that is communicating with the internet we will be able to see that traffic and sniff it as well.

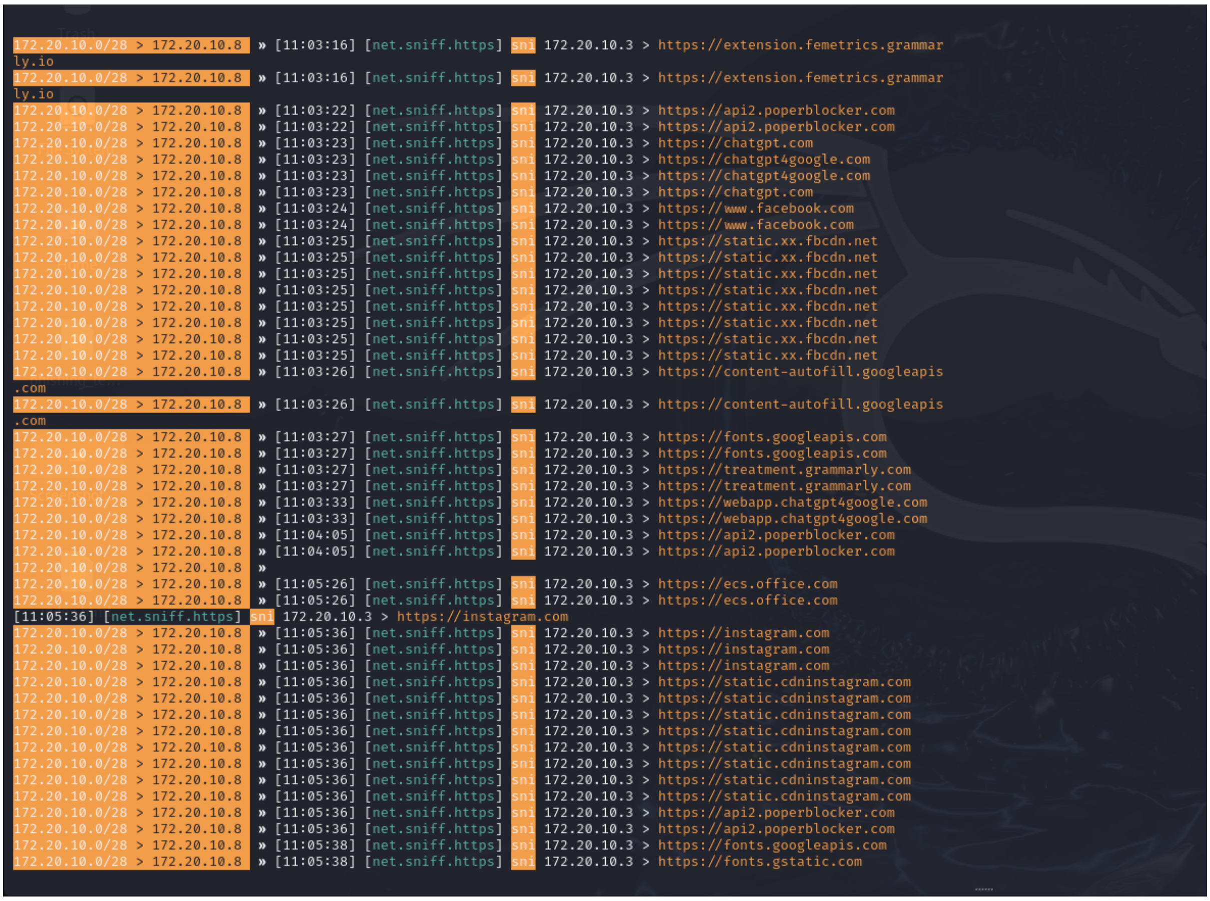


Figure 42: I visited google.com, Instagram

7.9 Attackers

How attackers are able to redirect any website a target is requesting to a malicious one? without the target even noticing? how is this possible?

7.9.1 DNS

DNS spoofing (or DNS cache poisoning) is a technique where the attacker sends falsified DNS responses to a target's DNS queries. This can cause the target to be redirected to a malicious site controlled by the attacker. Bettercap can facilitate DNS spoofing by:

1-Intercepting DNS Requests: Bettercap captures DNS requests made by the target.

2-Sending Fake DNS Responses: It responds to these requests with incorrect IP addresses, directing the target to malicious sites instead of the intended ones.

7.9.2 Redirect the victim browser

This module (**Dns.spoof**) will allow the attacker to redirect any website the victim is requesting however first I want to create my website.

In this example, I will redirect the victim to my own websites running on Kali Linux.

7.10 Creating the Website

In this section, I have created my own website which is based on HTML,CSS and JS. it is a home appliance website. Every user can log in, create an account and buy items. It is similar to the Bosch home appliance website. So in this situation, I want to deceive the user. because of the resemblance of my website to the Bosch website.

we can easily spin up on a website hosted on our Kali Linux machine locally using Apache.

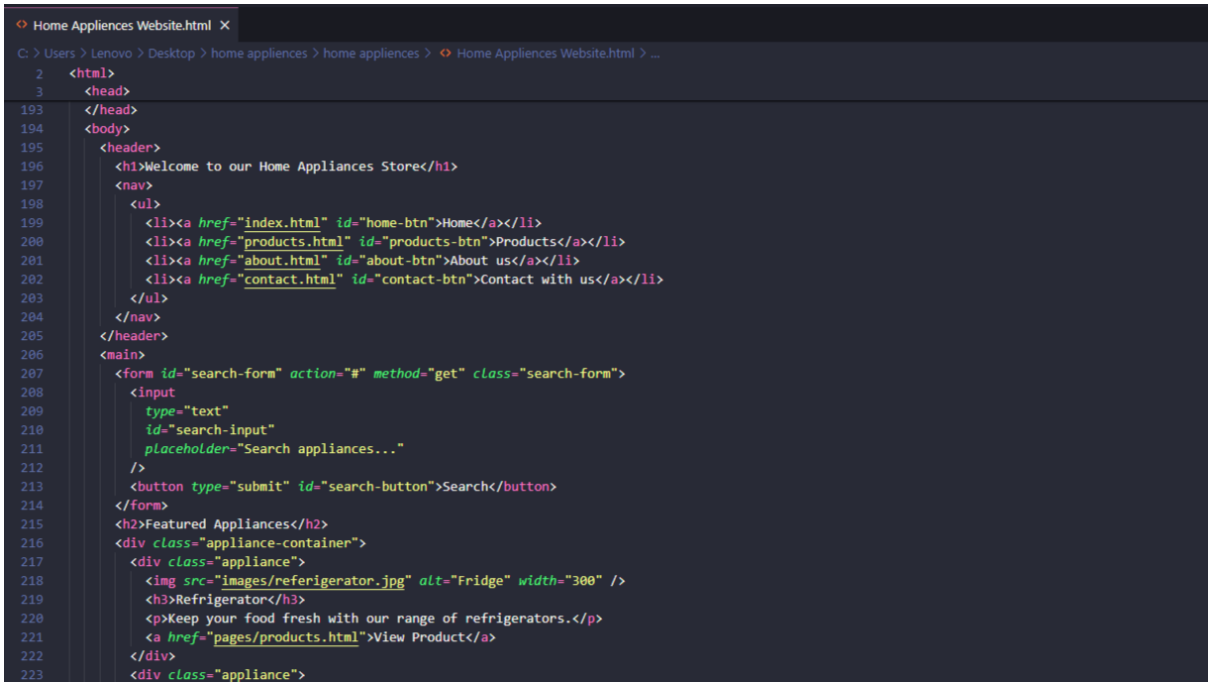


Figure 43: Brief Html code

7.10.1 Apache Server

let's open a new terminal window and all we have to do here is to turn on the Apache server service so let's do **sudo service apache2 start** and **ifconfig** to get the IP address.

In this section, I have modified the Apache 2 default page by going to**/var/www/html/index.html** location and modifying the index.html file.

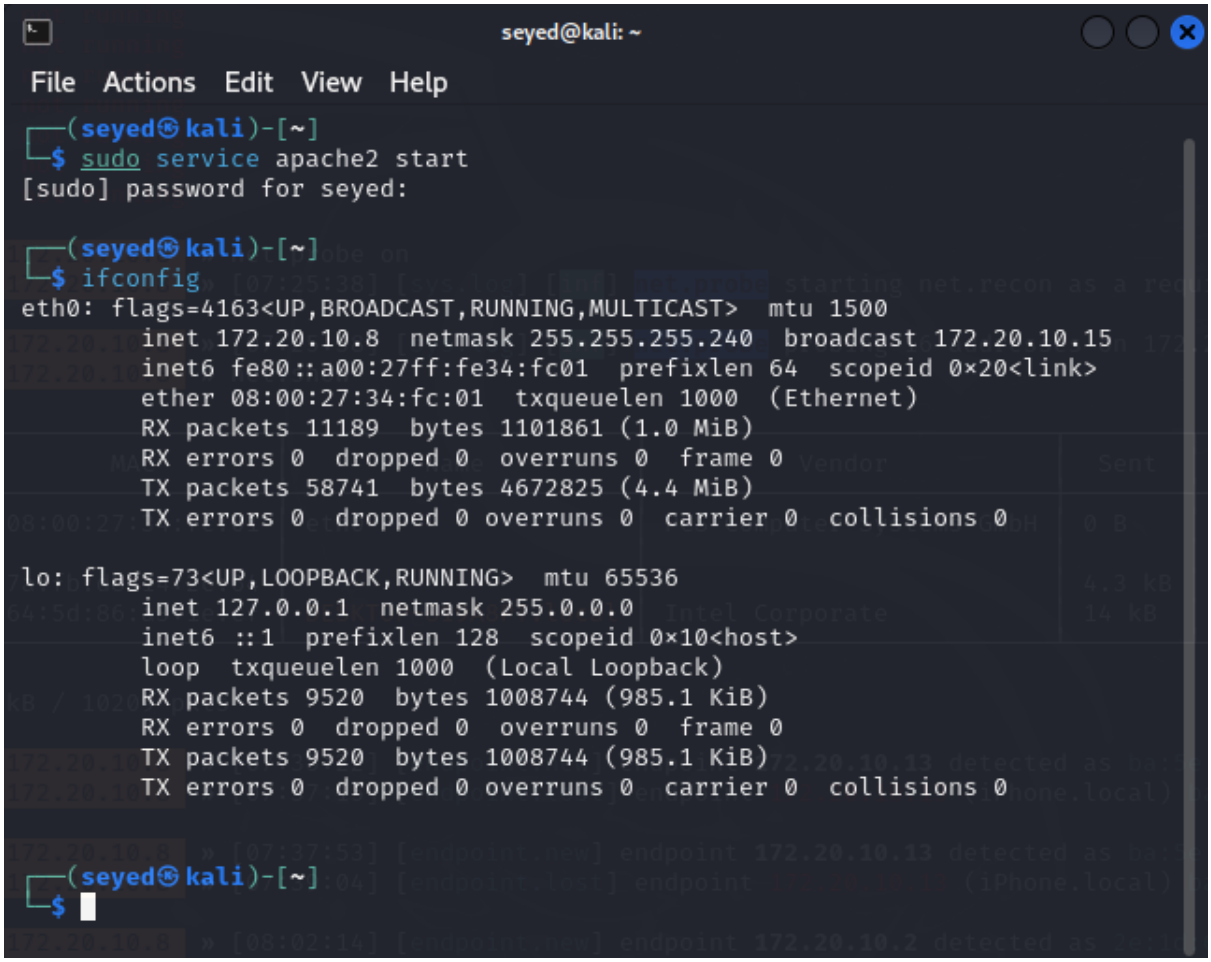


Figure 44: Apache Server is running

7.10.2 My Website

The result of writing my IP address on my browser.

It shows my website and its features and login button.

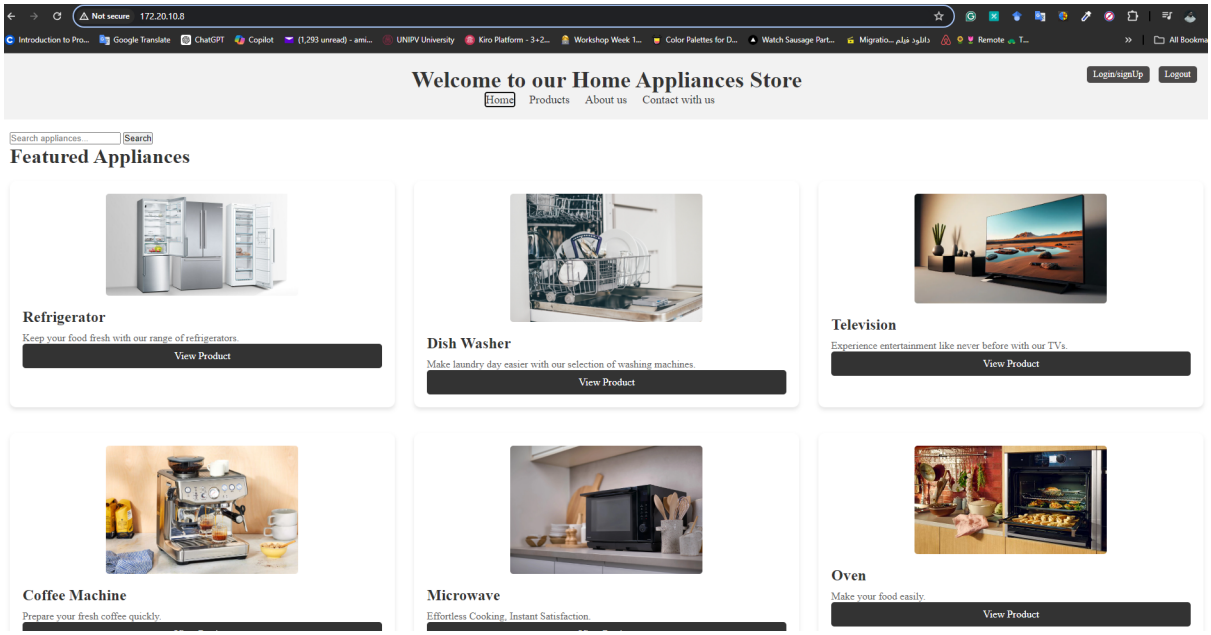


Figure 45: This is my website

7.10.3 Output of spoofing using DNS

In this section, I went back to my Bettercap terminal. I wrote:**set dns.spoof.domains bosch.com** (The front of the domain it is the name of the domain that I want to spoof) Turn the module on: **dns.spoof on**

This command will tell better cap to initiate a DNS spoofing attack and if the victim visits for example bosch.com it will redirect them to our local website that we have running on Kali Linux.

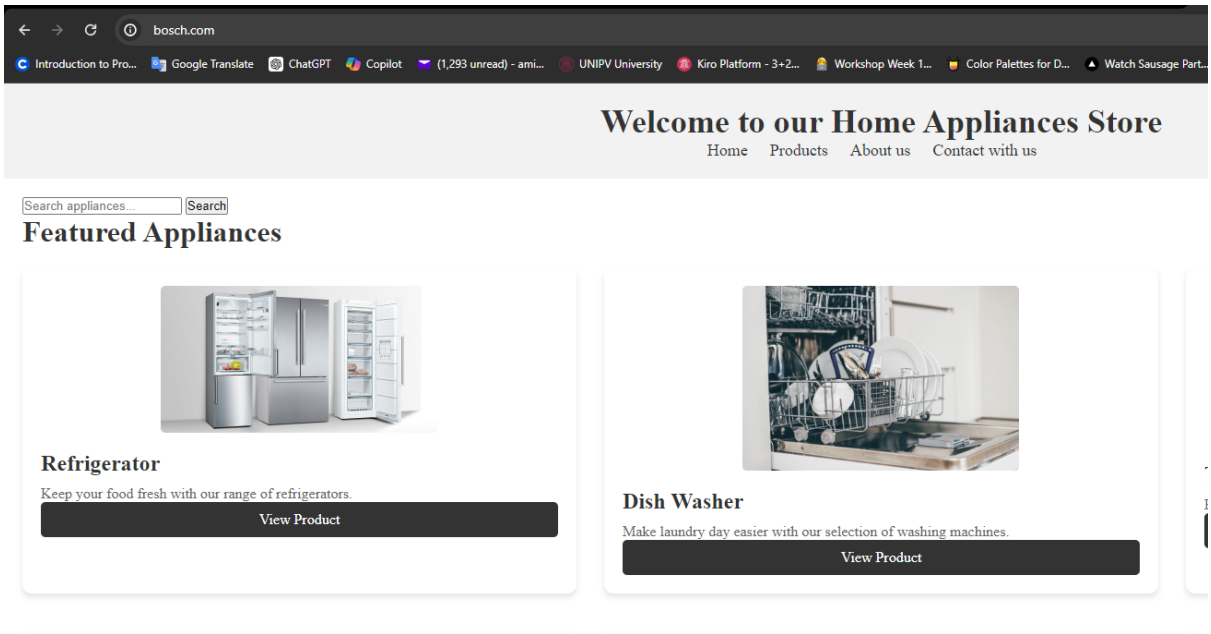


Figure 46: look at the address bar.

7.11 Conclusion

I got automatically redirected to the fake website that I have running on my Kali Linux machine.

As a victim, this would appear completely normal to me because the URL displayed is correct—it's bosch.com—and everything looks legitimate. In this scenario, I'm merely demonstrating a fake website I created, but malicious actors could easily replicate the actual bosch.com login page and use it instead.

This technique is extremely dangerous. For instance, hackers can create fake social media login pages that closely mimic the authentic ones. Many unsuspecting victims would be deceived by the realistic appearance and provide their login credentials, thinking they are on a legitimate site. As a result, attackers gain access to valuable personal information without the victim ever realizing what has happened.

The seamless nature of this attack, combined with the convincing appearance of the fake sites, makes it highly effective and poses a significant threat to online security. Users need to be vigilant and aware of such risks to protect themselves from falling into these traps.

8 BeEF

BeEF, short for Browser Exploitation Framework, is a powerful penetration testing tool that focuses on web browsers. It is designed to provide effective client-side attack vectors and is an essential tool for security professionals. BeEF allows testers to assess the actual security posture of a target environment by using client-side attack vectors.

8.1 Key Features of BeEF

- **Browser Hooking:** BeEF hooks one or more web browsers for the purpose of launching command modules against them. This is done by injecting JavaScript code into the target browser, which then communicates back to the BeEF server.
- **Command Modules:** BeEF includes a wide range of modules for exploiting and testing the security of the hooked browsers. These modules can perform various tasks, such as information gathering, network attacks, and social engineering.
- **Extensive Exploit Library:** BeEF comes with an extensive library of exploits that can target different browsers and browser versions. This library is constantly updated by the community to include the latest vulnerabilities.
- **Integration with Other Tools:** BeEF can integrate with other penetration testing tools like Metasploit. This allows for more complex attack scenarios and exploitation workflows.
- **User-Friendly Interface:** BeEF features a web-based interface that provides an intuitive environment for managing hooked browsers and launching exploits.

8.2 What Can BeEF Do?

1. Information Gathering
    - **Fingerprinting:** Gather detailed information about the target browser, such as its type, version, plug-ins installed, and more.
    - **Geolocation:** Determine the geographical location of the hooked browser.
2. Exploitation
    - **Launching Browser Exploits:** Execute exploits against the hooked browser to gain further control or to exploit specific vulnerabilities.
    - **Network Attacks:** Conduct network attacks from the hooked browser, such as port scanning and service identification.
3. Social Engineering
    - **Phishing:** Create convincing phishing pages to capture user credentials.
    - **Clickjacking:** Implement clickjacking attacks to trick users into clicking on concealed elements.
4. Post-Exploitation
    - **Persistent Hooking:** Maintain control over the browser by ensuring the hook is re-established even after the browser is closed and reopened.
    - **Data Exfiltration:** Extract sensitive data from the target environment through the hooked browser.
5. Real-Time Interactions
    - **Command Execution:** Execute JavaScript commands in real-time on the hooked browser.
    - **Dynamic Payloads:** Deliver and execute dynamic payloads for more sophisticated attack scenarios.

8.3 Setting Up BeEF on Kali Linux

- **Install beef:**

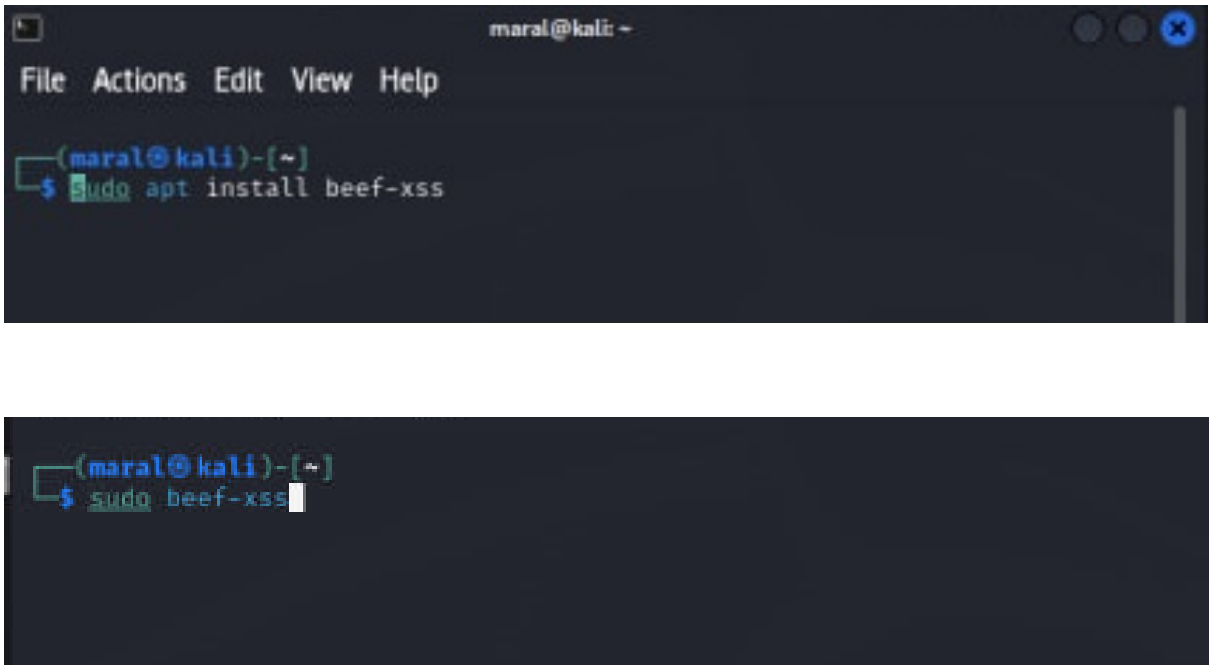


Figure 47: install beef-xss

and then Access the BeEF control panel by navigating to http://localhost:3000/ui/panel in a web browser. and Login with your credentials.

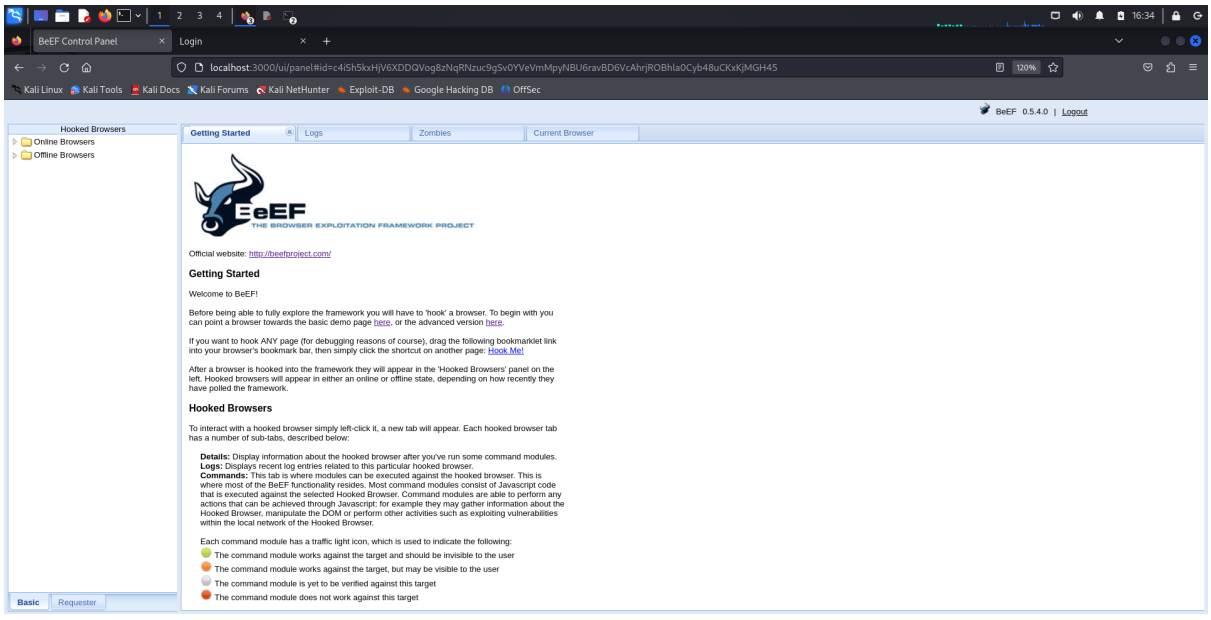


Figure 48: BeEF control panel

- **Add hooked browser:** In this project, we have utilized a local server to host a URL containing a hook script, which can be shared with targets for testing purposes. Below are the steps to set up and run a local server with an HTML file that includes the hook script:

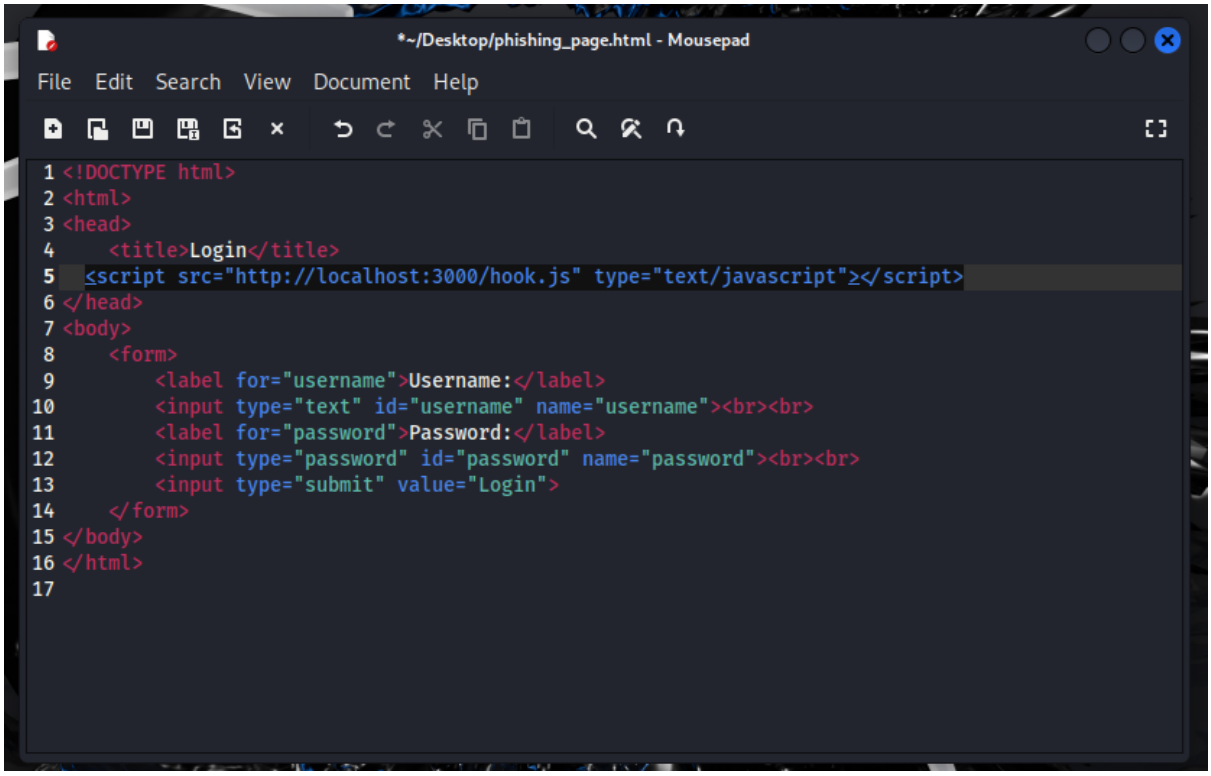


Figure 49: HTML file with Hooked Script

1. Create an HTML File: Prepare an HTML file that contains the necessary hook script. Ensure that the script is correctly embedded within the HTML structure.
2. Set Up a Local Server: Use a local server to host the HTML file. This can be achieved using various tools such as Python's built-in HTTP server, Apache, or any other preferred local server software.
3. Run the Local Server: Start the local server to make the HTML file accessible via a local URL. For example, if using Python, navigate to the directory containing the HTML file and run the following command:

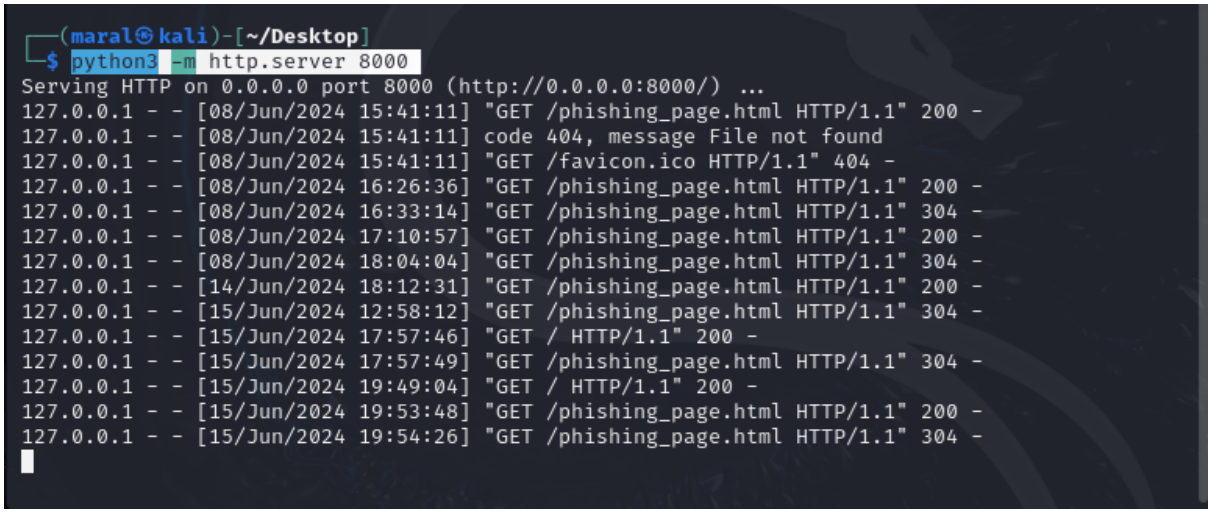


Figure 50: Run python Server

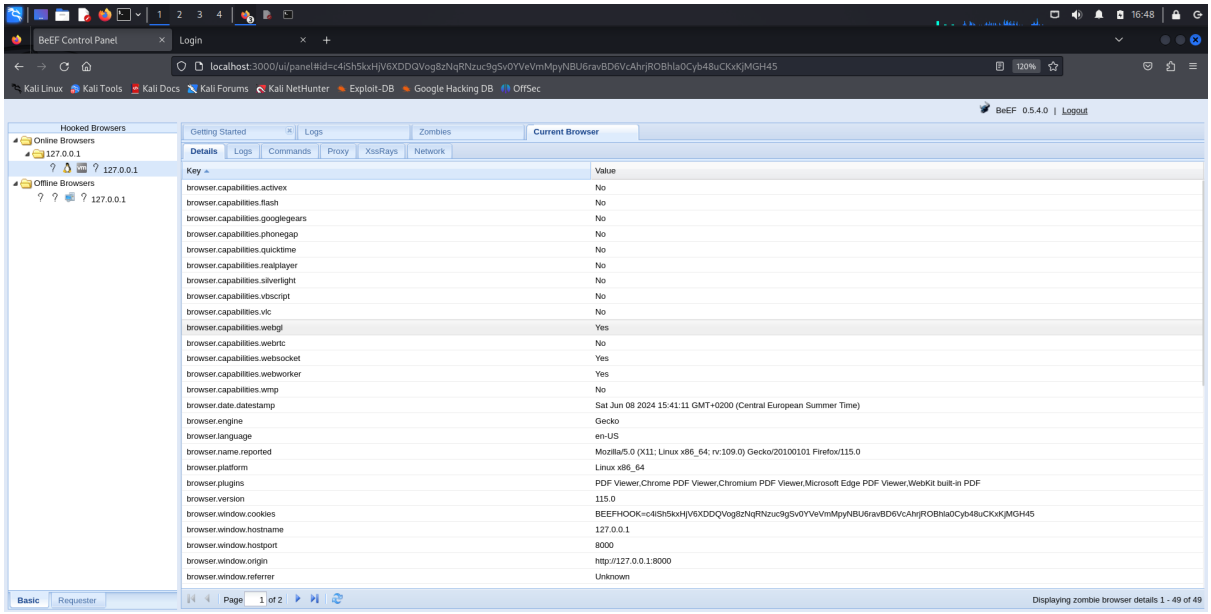
This will start the server on port 8000, and the HTML file will be accessible at http://localhost:8000.

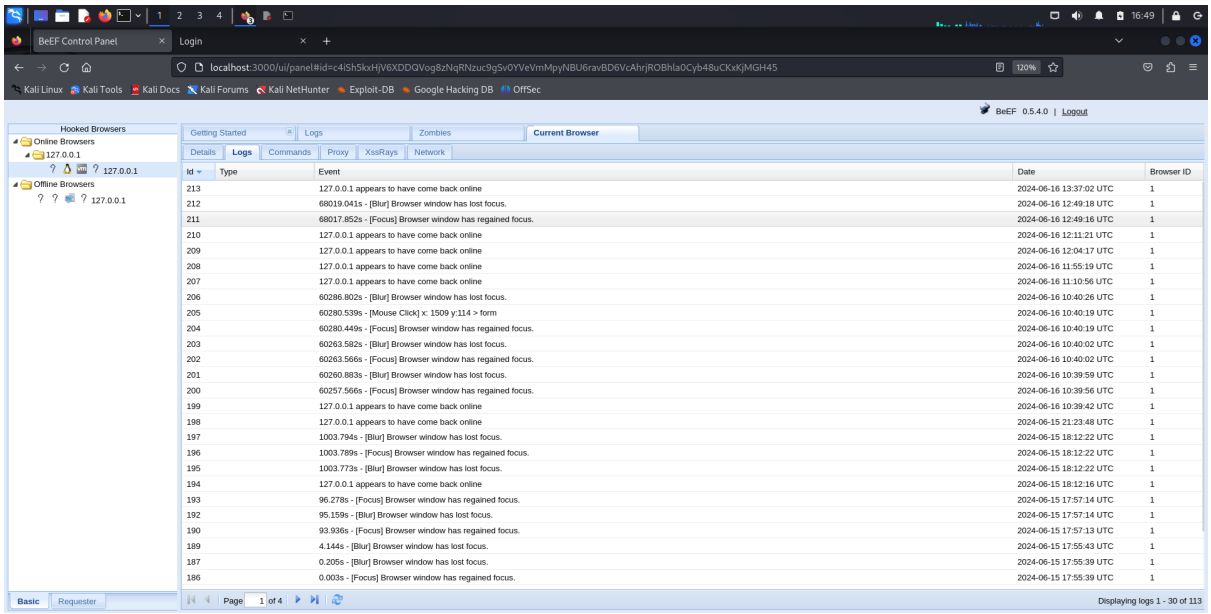
By following these steps, you can effectively host and share the URL using your local server environment.

8.4 Information Gathering Using BeEF

When the hook script is executed on the server, the BeEF (Browser Exploitation Framework) control panel will detect and add the hooked browser to its list. Under the ”Hooked Browsers” section, you will see the newly connected browser listed. By selecting the hooked browser, you can view detailed information about the connected browser in the ”Current Browser” tab.

In detail, the information includes browser specifics such as browser language, IP address, and other relevant data. Additionally, the control panel’s log provides a comprehensive record of activities and interactions, allowing you to track various actions performed by the hooked browser.





8.5 Social Engineering Attacks Using BeEF

Under the ”Command” tabs within the BeEF (Browser Exploitation Framework) control panel, various social engineering options are available. These options were utilized to gain insights into their functionality and operation.

- **Google Phishing:** To simulate a phishing attack that mimics the Google login page to capture user credentials.

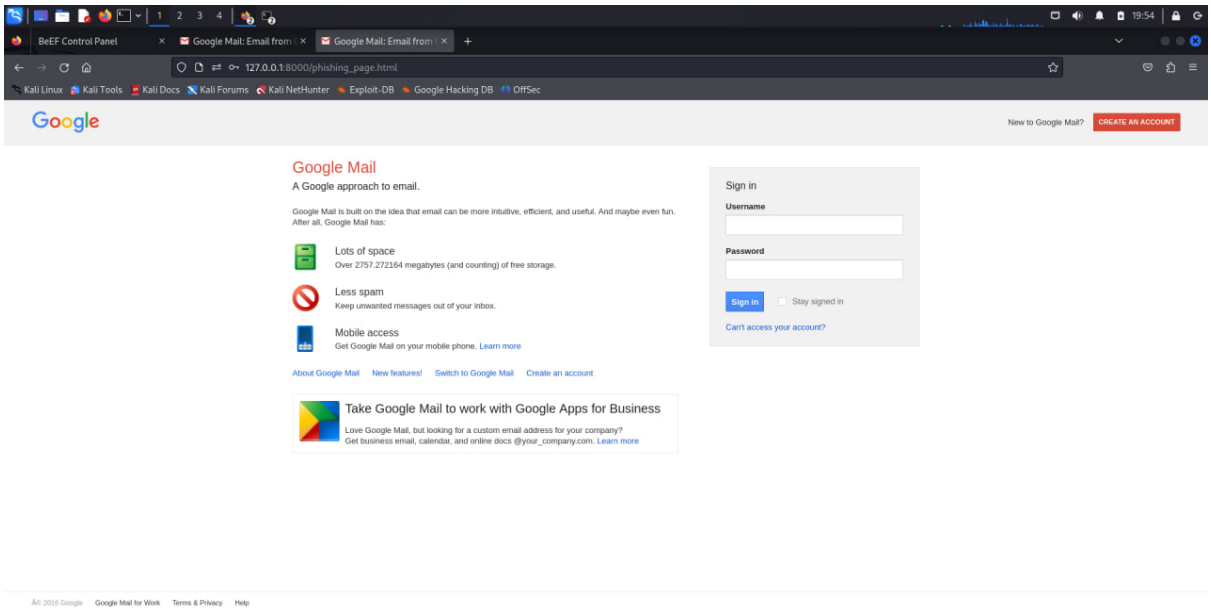


Figure 51: Google phishing

how to Execution:

1. Send the cloned link to the target via email or social media.
2. Once the target enters their credentials, the data is captured and sent to the attacker’s server.
3. The BeEF control panel will show the hooked browser, and further exploits can be launched from there.

- **Facebook Top-Up Scam:** To trick users into entering their Facebook credentials under the guise of a promotional top-up offer.

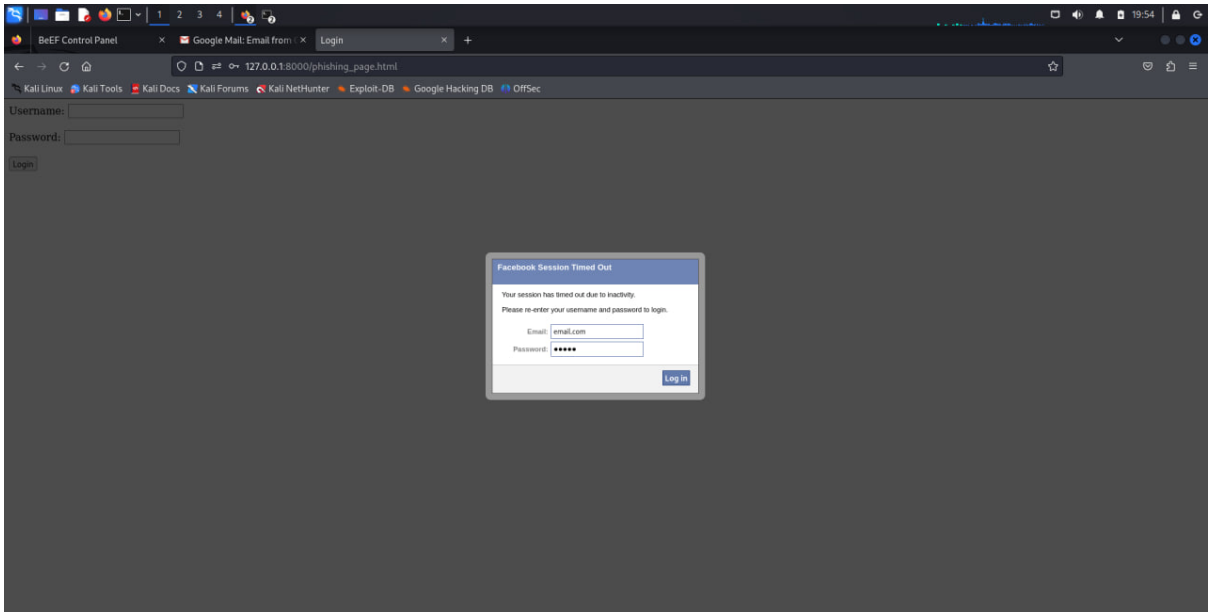


Figure 52: Facebook Top-up

how to Execution:

1. Share the promotional link on social media or through email.
2. When the target clicks on the link and attempts to claim the top-up by logging in, their credentials are captured.
3. BeEF can then be used to monitor and manipulate the target's browser session.

- **Notification Bar (Notifbar) Exploit:** To use a fake notification bar to deceive users into taking actions that could compromise their security.

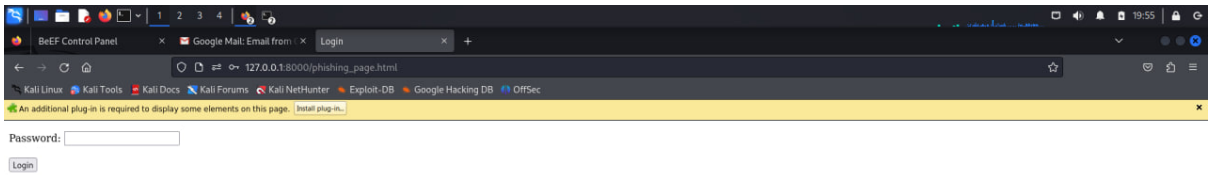


Figure 53: Fake Notification Bar

how to Execution:

1. When the target interacts with the notification bar, BeEF hooks their browser.
2. Use BeEF's extensive library of commands to execute further attacks, such as keylogging, stealing cookies, or redirecting the user to malicious sites.

8.6 Analysis of Results

- **Phishing Effectiveness:** The Google phishing attack successfully captured credentials from unsuspecting users. This underscores the importance of educating users about the risks of phishing and the need for robust email filtering and security measures.
- **Facebook Top-Up Scam:** The Facebook top-up scam was particularly effective on social media, highlighting the ease with which attackers can exploit human curiosity and trust. Users should be wary of too-good-to-be-true offers and promotions.
- **Notifbar Exploit:** The notification bar exploit demonstrated how easily users can be deceived by seemingly legitimate notifications. This reinforces the necessity for users to verify the authenticity of notifications and updates.

8.7 Conclusion

The use of BeEF on Kali Linux for social engineering demonstrates the significant threat posed by browser-based attacks. By understanding the techniques and tools used by attackers, we can better prepare and protect against these threats. This project underscores the importance of vigilance, user education, and the implementation of comprehensive security measures to mitigate the risks associated with social engineering attacks.

9 Nikto

Nikto is an open-source tool that scans web servers for potential vulnerabilities and security issues. It checks for outdated software versions, and server configurations like multiple index files and HTTP server options, and attempts to identify installed web servers, software, insecure files, scripts, and other security risks. Nikto's scan items and plugins are regularly updated and can be automated for convenience. It comes pre-installed on Kali Linux and is designed to test web servers in the quickest time possible.

9.1 Basic Scan and Other Options

There are so many options for scanning the website or IP. Figure 54 shows some commands that can be used. For example, the port number can be specified for a scan. With -SSL, the scan can be forced to use an SSL/TLS connection. Also, if you use the tuning option, you can focus on specific tests like injections or file enumeration, or it is possible to exclude them.

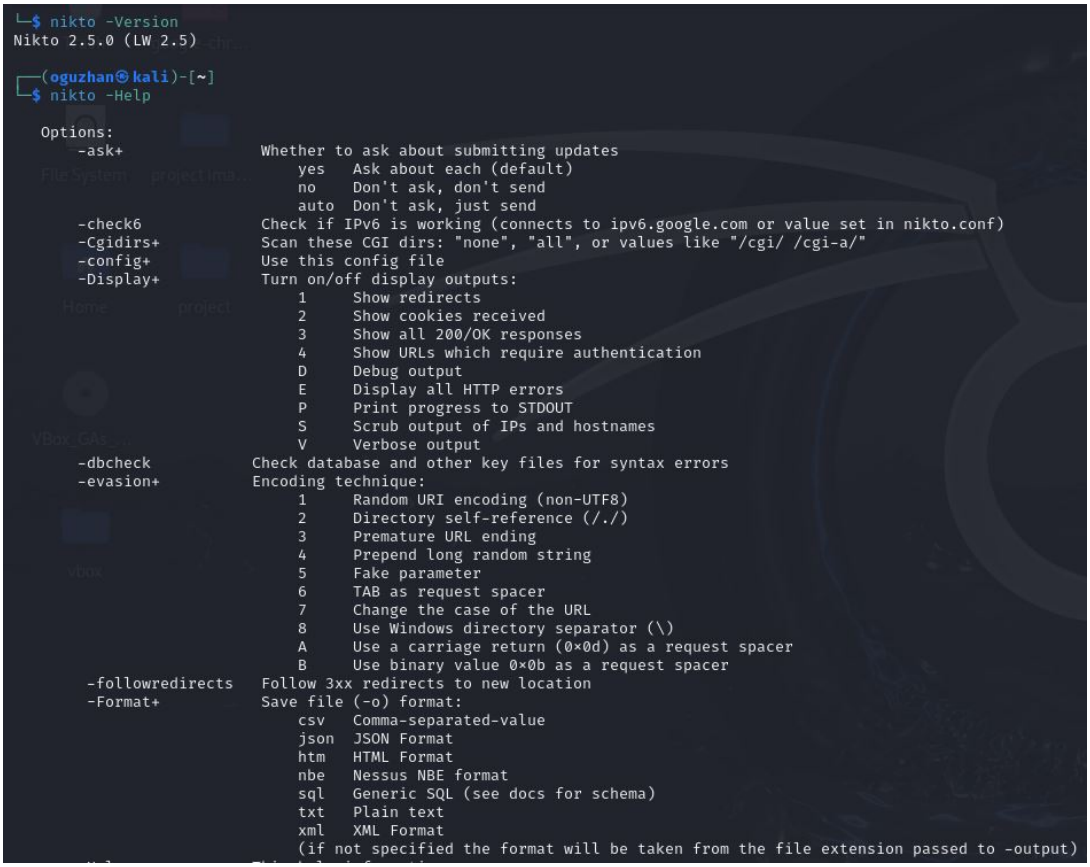


Figure 54: Options of Nikto

I started by scanning the dummy website scanme.nmap.org because scanning regular websites needs permission and could raise ethical and legal issues. The command is: `nikto -h scanme.nmap.org`. I was able to find the target IP, hostname, and port number after scanning, as shown in Figure 55. There are IPv4 and IPv6 addresses that are associated with the website. The IPv4 address 45.33.32.156 on port 80, which is the default port for HTTP, was the target of the scan. After additional research, it was discovered that the web server is running Apache 2.4.7 on Ubuntu; however, it was pointed out that this version is out of date, which would expose the server. The next line shows that the server has failed to return the "X-frame-options" HTTP response header, which describes the browser's frame setup. Not having this header could result in clickjacking attacks, where users are misled into clicking on something other than what they see. It's also noticeable that the X-Content-Type-Options header is not set in the next line. If this header is not sent, the browser may make assumptions about the data, leading to security risks such as MIME sniffing attacks. For example, a malicious actor could upload files

that can be interpreted as HTML and carry out attacks. Secondly, no common gateway interface (CGI) folders were discovered. It could be configured in this way on purpose to restrict the execution of scripts, but it should be checked.

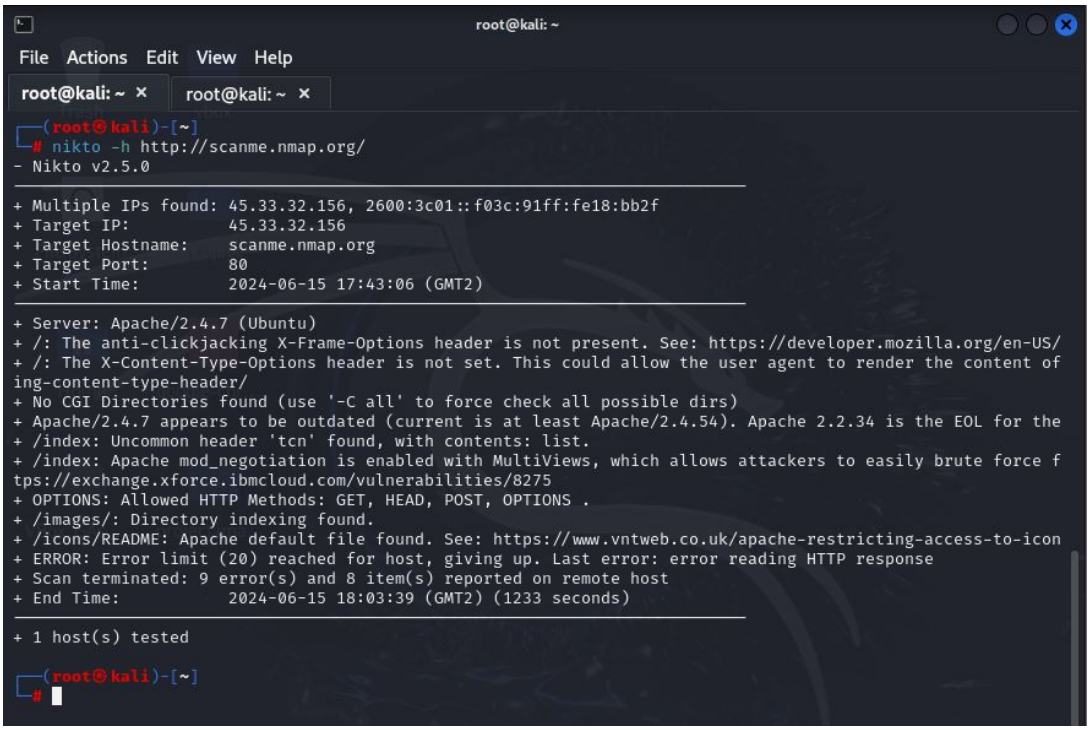


Figure 55: Basic Scan to Dummy Website

As shown in Figure 56, I checked it using "-C all" as well; however, with Nikto, I am missing any additional information for this trial.

Figure 56: nikto -h scanme.nmap.org -C all

When we continue on Figure 55, we can observe that Apache mod_negotiation is enabled on the other line. This module is used for choosing the ideal document for the client. However, if an error occurs, such as an invalid accept header coming from the client, the server will reply with a possible directory listing. Since an attacker may learn about filenames and extensions and can start a brute-force attack. In the next step, allowed HTTP methods are shown. These are safer methods; however, there could be methods like DELETE or TRACE if the configuration is wrong. Revealing some methods could result in a security breach.

For the next two lines, it is seen that one index and file are found, respectively. The index does not belong to the original website. Figure 57 is the homepage of the dummy website.

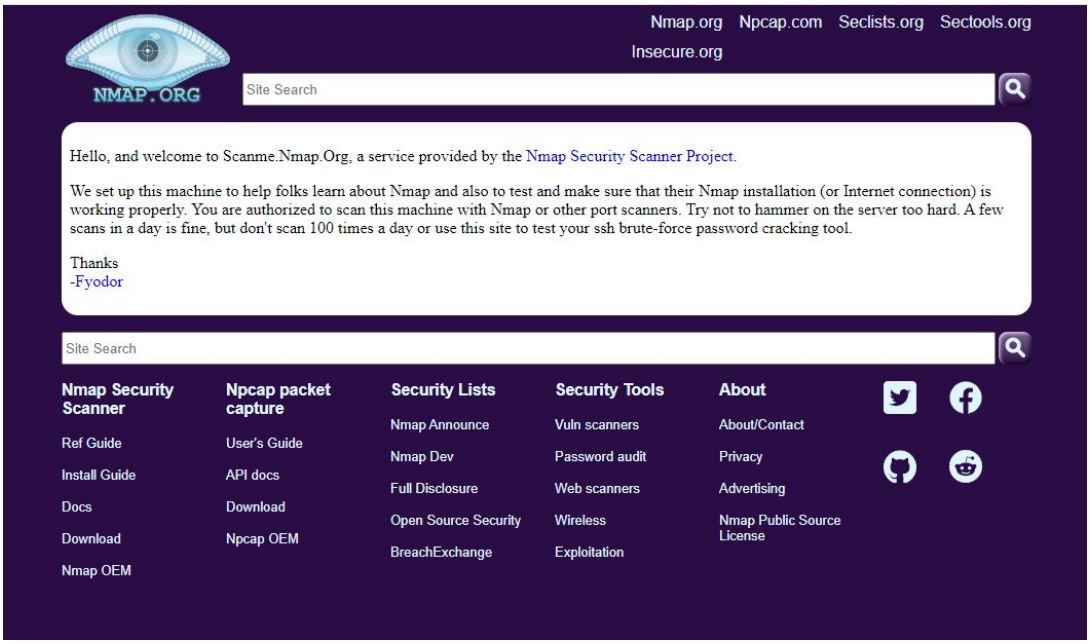


Figure 57: Homepage of Dummy Website

When the user or attacker goes to this directory, it is seen in Figure 58 that there is an image. Therefore, this kind of index may allow anyone to reach the directory and can lead to information leakage.

Figure 58: Directory of Image

With the other document path, it is possible to reach the Apache README file, which is shown in Figure 59. Also, this can provide information to attackers about server configurations.

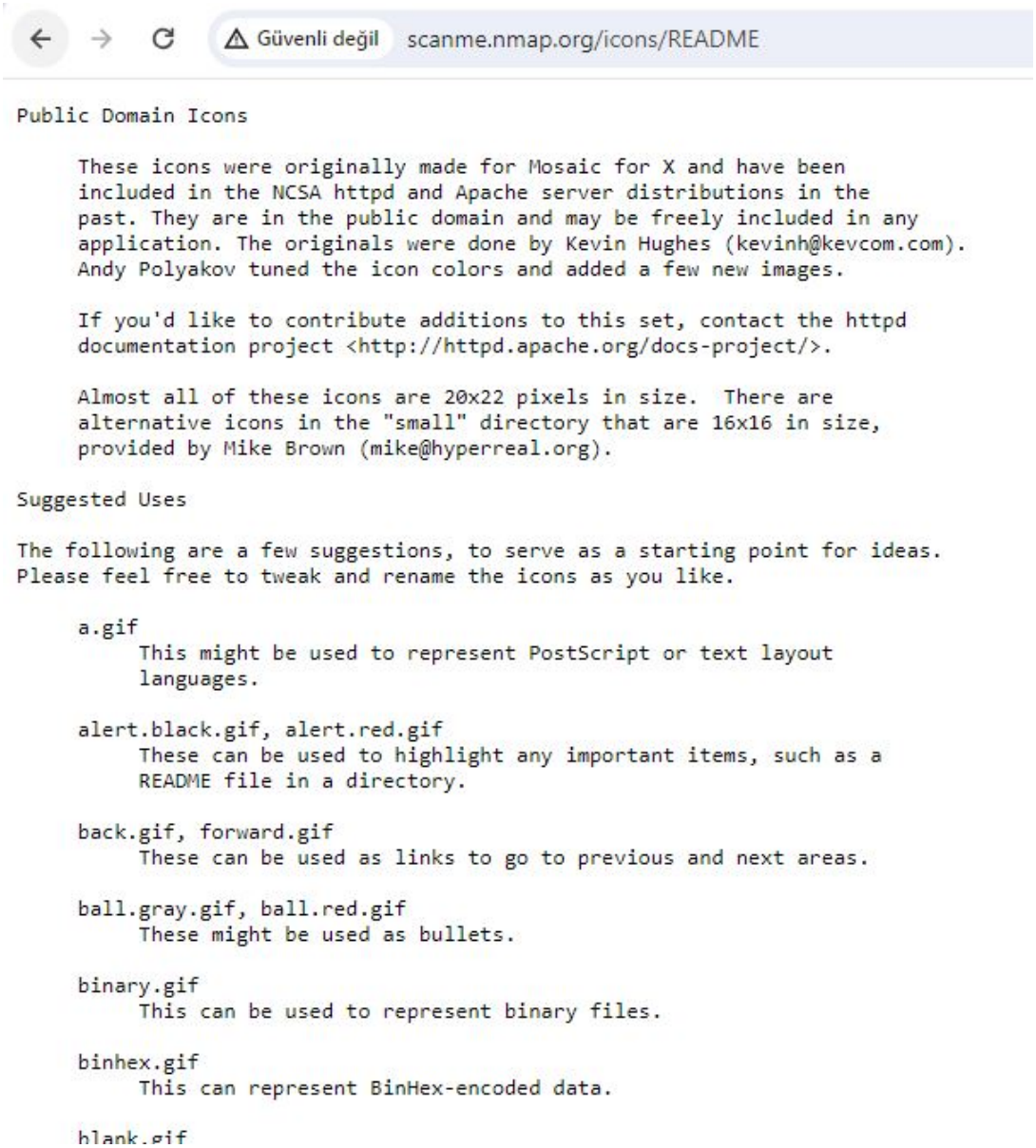


Figure 59: Directory of Apache README file

For this case and resolving the issues, these can be recommended:

- Updating Apache server and configuration(restriction)
- Configuration of the headers
  - X-Frame-Options: SAMEORIGIN
  - X-Content-Type-Options: nosniff
- Disabling directory indexing

By following these steps, the risks of scanme.nmap.org against attacks may be reduced.

9.2 Scan of Personal Website and Additional Attributes

The fake website is utilizing port 80, which is for HTTP. When we try to use SSL force to scan, we end up getting no results as in Figure 60.

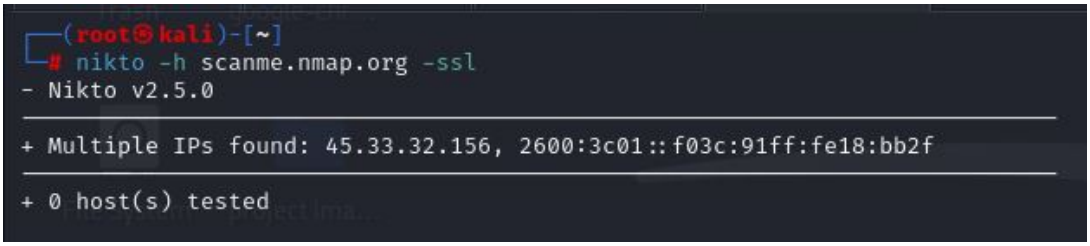


Figure 60: SSL scan force to dummy website

Since I don't have any permission to scan any website, I decided to scan my personal website. Yet, when I analyze my website that is visible to you, the outcome varies(Figure 61). There are both IPv4 and IPv6 addresses, and the scan allows us to identify the target IP. This time, the scan was carried out on port 443, which is the standard port for HTTPS. Following that, we observe the SSL details of my website, employing AES 256-bit encryption issued by Google Trust Services. Following this, the server appears to us as a Cloudflare. There are missing header parts on my website, much like the dummy website, so these headers need to be included. In contrast to the fake website, there are two distinct headers called 'x-redirect-by' and 'x-turbo-charged-by'. My website utilizes WordPress for content management and the LiteSpeed plugin to enhance performance. These two headings may not represent specific weaknesses, but as the owner, it is important for me to continually maintain these systems. Later, we notice the website utilizing TLS and the Strict Transport Security HTTP header, but with max_age=0. Browsers do not store this policy in their cache. Since this setup is equal to zero, HSTS is disabled on the server. Setting it to 1 year is necessary to prevent users from bypassing SSL and it serves as protection for webservers against downgrade attacks.

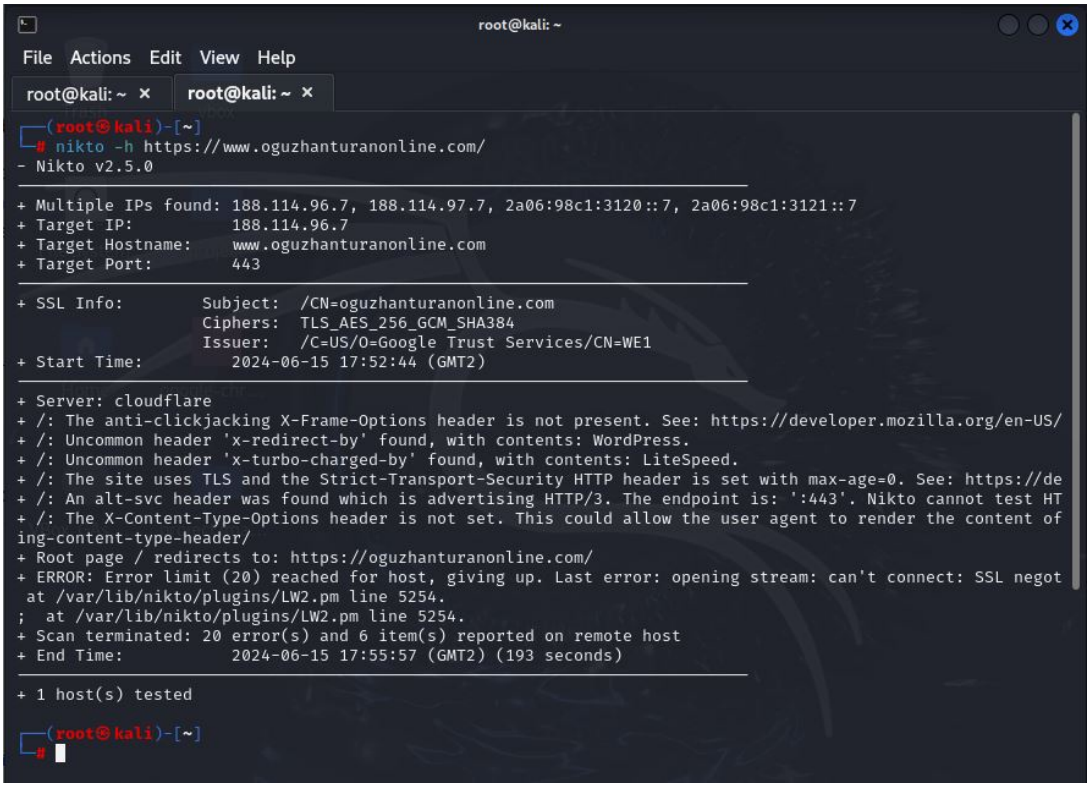


Figure 61: Scan results of personal website

Additional attributes of Nikto are present as well.

- nikto -h example.com -port 80: It has the capability to scan a designated port.
- nikto -h example.com -output /path/to/file.name: It is possible for us to store the outcomes. It might come in handy during extensive scans.
- nikto -Plugin headers -h example.com: Furthermore, we have the option to designate a plugin to broaden its capabilities to collect additional data. An illustration of the header's plugin can be found on the mock website. (Figure 62)

Figure 62: Plugin scan of dummy website

This is what I experienced with this tool. So, I can use this tool to enhance my website's security, but attackers could see the points to attack.

10 Final Conclusion

In conclusion, cybersecurity has become a crucial element in our digital age, where an increasing amount of personal and business data is stored and transmitted online. Hackers are continuously refining their techniques to exploit vulnerabilities and deceive users. Therefore, it is essential for users to be aware of these threats and adopt adequate protection measures, such as using up-to-date security software, practicing safe browsing habits, and continuously educating themselves on recognizing phishing attempts and other online traps. Only through a combination of advanced technology and awareness can we effectively defend against the increasingly sophisticated threats of the modern digital landscape.