LONDON METROPOLITAN UNIVERSITY

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: Riza Shrestha

London Met ID: 22067117

College ID: np01nt4a220087

Assignment Due Date: Tuesday, 7 May 2024

Assignment Submission Date: Tuesday, 7 May 2024

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.

Abstract

This coursework covers how important authentication is to protecting sensitive data and digital platforms. It explores three primary types of authentications: knowledge-based, possession-based, and inherent traits authentication, and their application in protecting user identities and access privileges. The study highlights vulnerabilities related to broken authentication and describes how such weaknesses can lead to significant security breaches, as demonstrated by actual events like the Equifax data breach. The coursework also covers effective mitigation strategies, including multi-factor authentication (MFA), secure session management, and robust credential handling, to counteract potential security flaws. Through practical demonstrations and theoretical analysis, the report shows the continuous need for advancements in authentication technologies to counter new security threats and protect digital environments.

Table of Contents

Table of Figures

1. Introduction

Broken Authentication Vulnerabilities in web applications.

Authentication is the process of confirming that only the right people and services can access the company's important data and systems. It can be done often through usernames, passwords, or hardware tokens, not only for humans but also for computer systems like servers and software (Cloudflare, 2024).

There are three main types of authentications:

i. Knowledge-based authentication, also known as "something you know," requires the use of secret information, like passwords or security questions, that is known only to the individual.
ii. Possession-based authentication, also known as "Something you have," verifies ownership of physical items like tokens or keys. Examples include hard tokens that are physically attached to computers and soft tokens that are transmitted to devices like smartphones.
iii. Inherent traits authentication, also known as "something you are or do," analyzes inherent features such as fingerprints or facial features. (Cloudflare, 2024)

Figure 1:Types of Authentications (Sorbello, 2021).

Authentication for individuals involves creating a username, password, and additional security measures such as facial recognition or fingerprint scanning. To protect this information, it is not directly stored in the company's database. Instead, passwords turn into hashing, where they are transformed into unique strings of characters known as hashes, which are then stored in the database. When a user attempts to log in, the entered password undergoes the same hashing process, and the resulting hash is compared to the stored hash in the database. If they match, access is granted. The collected data for biometric authentication techniques, such as fingerprint or facial scans, is encrypted, coded, and saved locally on the user's device (Microsoft Security, 2024) .

In my daily life, authentication is also important, especially when logging in to websites or using my devices like my phone and laptop. It protects my identity and personal data. Authentication systems confirm that I am the authorized user every time I log into an account, whether it's online banking, social networking, or email. This safeguards my privacy and prevents unauthorized access. For example, authentication, which asks for a username and password, ensures that I may see course materials and submit assignments securely when I access my college's online portal (MST). Likewise, by requiring valid credentials, authentication protects security during financial transactions and information sharing. However, the risk of security vulnerabilities has also increased, with the "Broken Authentication" vulnerability being a major threat.

The "Broken Authentication" vulnerability arises when an attacker successfully takes the identity of a legitimate user inside a web application. This happens as a result of flaws or weaknesses in the application's authentication system. The authentication process fails to properly verify the identity of users, allowing attackers to exploit vulnerabilities and gain unauthorized access to user accounts or sensitive information. This could result major consequences, including unauthorized data access, identity theft, and privacy breaches (HackerWhite, 2024).

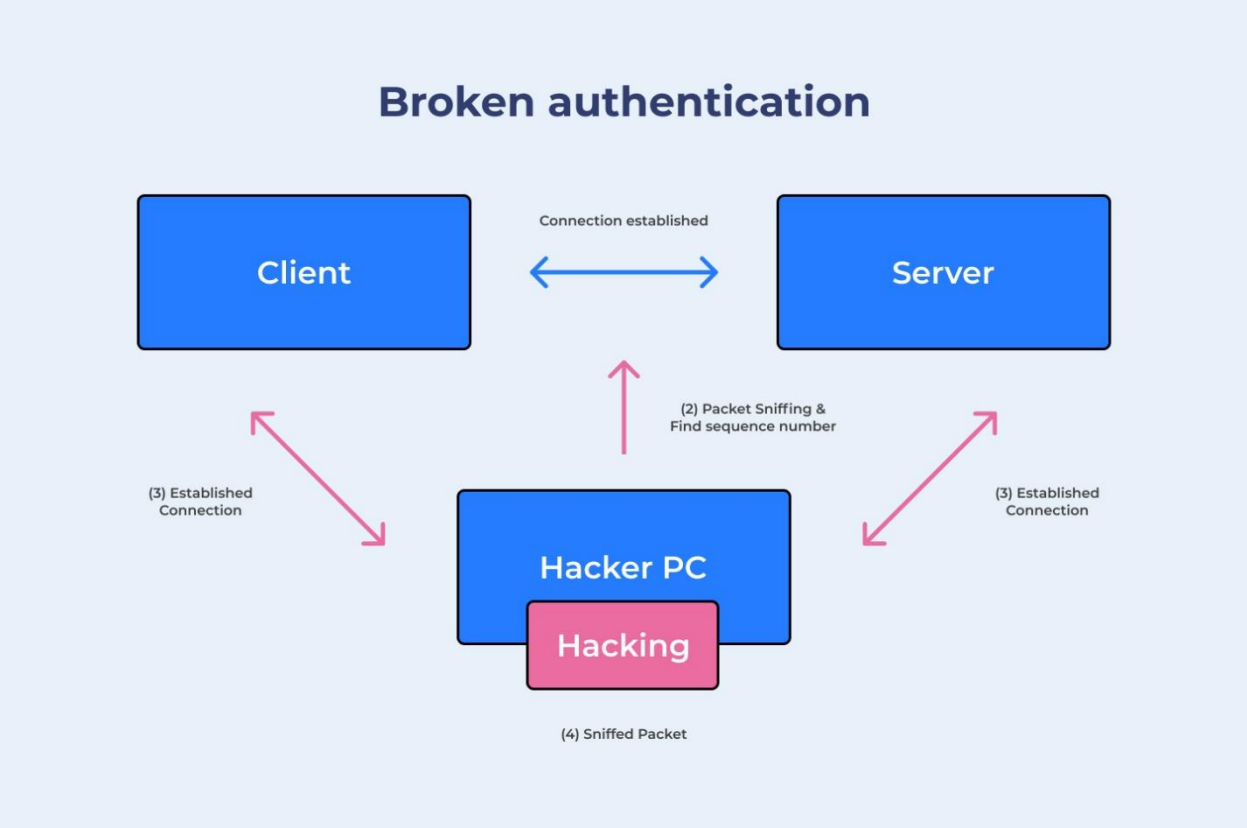

Figure 2: Broken Authentication (VARAKSINA, 2023)

A broken authentication attack puts the risk of private data theft by allowing unauthorized individuals to access a user's account and take credit card and social security numbers. This could lead to identity theft and unauthorized purchases. If hackers manage to gain access in business, they could manipulate data, gain control of the entire system, or access important accounts. Huge financial losses, damage to the business's reputation, a loss in trust from customers, and possibly even legal issues could result from this (Bright, 2024).

1.1 Aim

The aim of this coursework is to deepen the understanding of broken authentication and their role in securing digital systems and data against unauthorized access, thereby enhancing the cybersecurity framework of organizations.

1.2 Objectives

- To understand Broken Authentication vulnerability in web application.
- To practically demonstrate various authentication attacks.
- To create and evaluate effective strategies for mitigating risks associated with broken authentication.
- To evaluate the security of existing authentication methods and technologies against new and emerging cyberthreats.

2. Background

Poor session management and compromised credential management are two major vulnerabilities that often lead to broken authentication. Attackers can gain access when we fail to securely handle login credentials and session data. By taking advantage of these openings, they can enter secretly by acting as legitimate users. This puts personal privacy and data security in danger of unauthorized access and major breaches (Authgear, 2023).

Session Management

A web session is a set of exchanges that take place over a predetermined amount of time between a user and a website. For Example, when using an online shopping website, A user browses products without logging in and later decides to create an account to make a purchase. All the actions, from browsing products to completing the checkout process, form a single web session. The website monitors this session to ensure a smooth shopping experience, whether the user is logged in or not. The session should close when there's no activity to prevent unauthorized access if the user is gone. An attacker could take control and get access to the user's account if the session timeout isn't configured correctly. A session ID is given to a user each time they use a web application. The program can stay connected with the user when they browse the website due to the ID. Session IDs usually appear in URL parameters or cookies (Authgear, 2023).

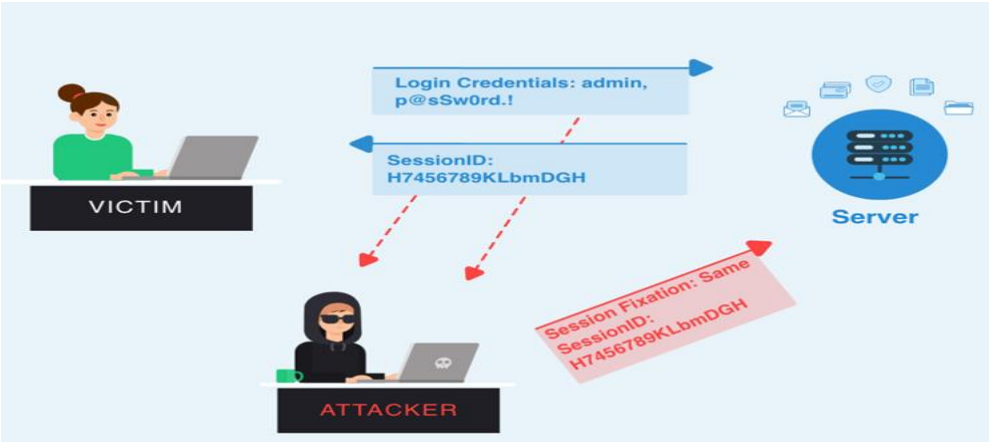

Figure 3: Session Management (DEMIR, 2021).

Creating rules for these sessions is part of session management. This includes determining how long a session can remain active before automatically logging the user out, how session IDs are generated and invalidated, and how securely they are associated with a user's IP address (Poza, 2020). Attacks related to session management fall into various types of broken authentication. These consist of:

i. Session hijacking

Session hijacking is a security threat where attackers exploit vulnerabilities in web applications to steal users' session IDs, which gives them a way to pretend to be legitimate users. For example, if a user forgets to log out of their account on a public computer, an attacker can take control of the session and access private information (Authgear, 2023).

ii. Session ID URL

Session ID URL rewriting occurs when a website's URL reveals a user's session ID, enabling anyone with access to the URL, especially through unprotected Wi-Fi, to continue the session. When session IDs are entered in the URL directly rather than being kept in cookies, a vulnerability occurs (Authgear, 2023).

iii. Session Fixation

In a session fixation attack, the hacker grabs a valid session ID from the website and tricks the user into using it to log in. The hacker can access the user's account if the website does not change the session ID after login (QAwerk, 2023).

Compromised Credentials

Credentials can be compromised due to phishing attacks, data breaches, malware, or major security vulnerabilities. Hackers often use stolen login credentials to gain access to a system. Stolen credentials were the main method in 19% of breaches, according to a 2022 IBM study. Weak passwords, such as "123" or "admin," are frequently used by users, and they are rarely changed. This increases stealing credentials by hackers using brute-force or password-spraying attacks. Some users reuse their passwords for all of their accounts, which leaves them open to credential stuffing attacks. Hackers can take advantage of weaknesses made by developers, such as keeping passwords in plain text or employing poor encryption (QAwerk, 2023). Some attacks using credentials that have been either stolen or compromised are:

i. Credential Stuffing

Credential stuffing happens when hackers take stolen usernames and passwords and try using them on different websites. It's a big problem because lots of people reuse their passwords, making it easier for hackers to break into different accounts. They might also share or sell the stolen login details to other hackers (Authgear, 2023).

To carry out a credential stuffing attack, hackers use a botnet, a network of infected computers, to try stolen username and password combinations on many websites at once. These attacks can overload a company's IT systems, causing websites to handle up to 180 times their usual traffic load (Poza, 2023). Attacker compromised credentials by sending fake emails that act as a legitimate source to trick user into revelling their credentials, using weak password, not properly securing the ssh keys. Once the attacker has access, they can install malware to integrate into a botnet.

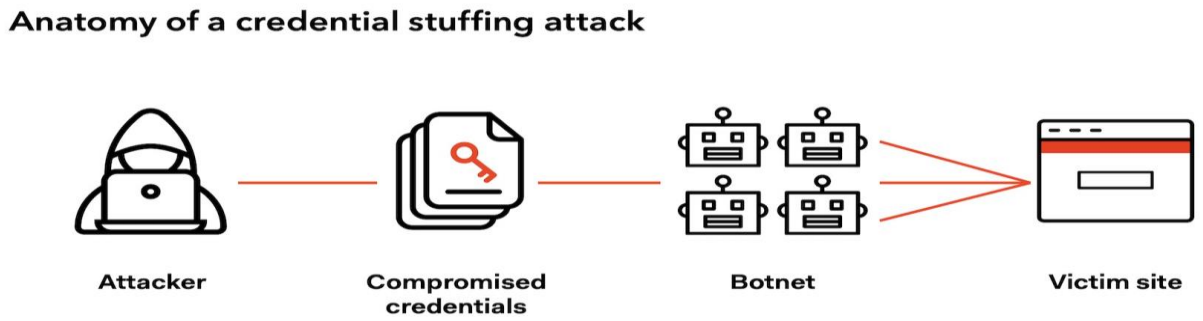

Figure 4: Anatomy of a credential stuffing attack (Poza, 2023).

ii. Password Spraying

Password spraying is a type of brute force attack where hackers try to access accounts using a list of usernames and common passwords like "123456" to break into accounts. According to reports, over 80% of hacking-related breaches involve brute-force methods like password spraying. These attacks often go unnoticed because organizations don't track failed login attempts. (Poza, 2021)

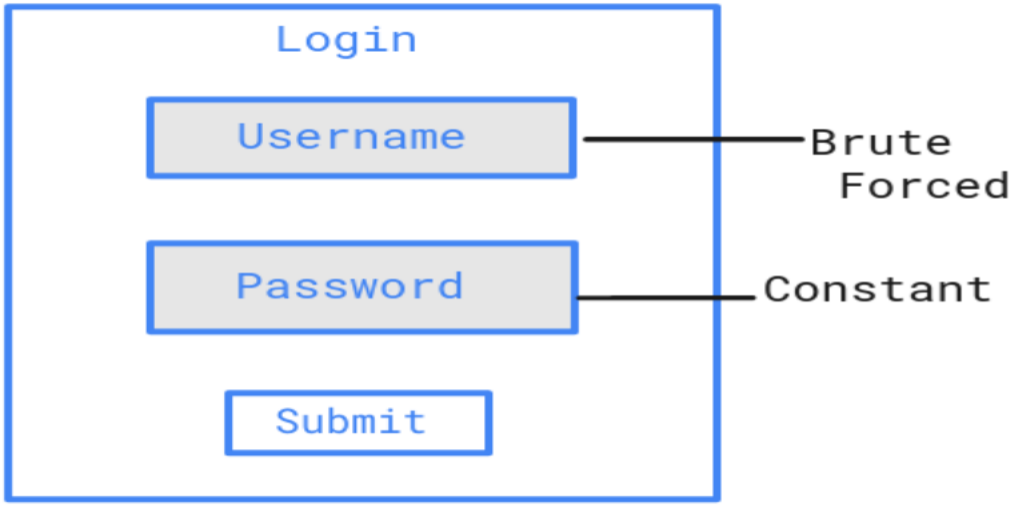

Figure 5:Password Spraying (Ranjan, 2024).

Attackers often exploit password spraying to gain access to systems, taking advantage of user likelihood to use easily guessable passwords. The attacker collects a list of usernames, which they might get from the dark web or create themselves based on patterns in company email addresses and employee information. Then they collect used passwords that can be obtained easily from a number of sources, such as reports, research papers, and online sites. They systematically try various login and password combinations using automated tools until they find one that works. This method allows them to bypass security measures like account lockout or IP address blocking by spacing out their attempts (Poza, 2021).

iii. Phishing attacks

Phishing attacks include sending fake emails by scammers to trick user into giving out their login details or downloading malware, could finally lead to account compromise, or victims could be redirected to fake sites. It pretends to be from a reliable source and asks the recipient to share personal information on a fake website (Cisco, 2024). Phishing is the most common form of social engineering, where deceivers manipulate people into giving away information or assets to the wrong hands. The FBI reports that phishing emails are a preferred method for hackers to spread ransomware to both individuals and organizations. Phishing is currently the second most common reason for data breaches, according to IBM's 2022 Cost of a Data Breach report. With average damages of $4.91 million per event, it is also the most expensive.

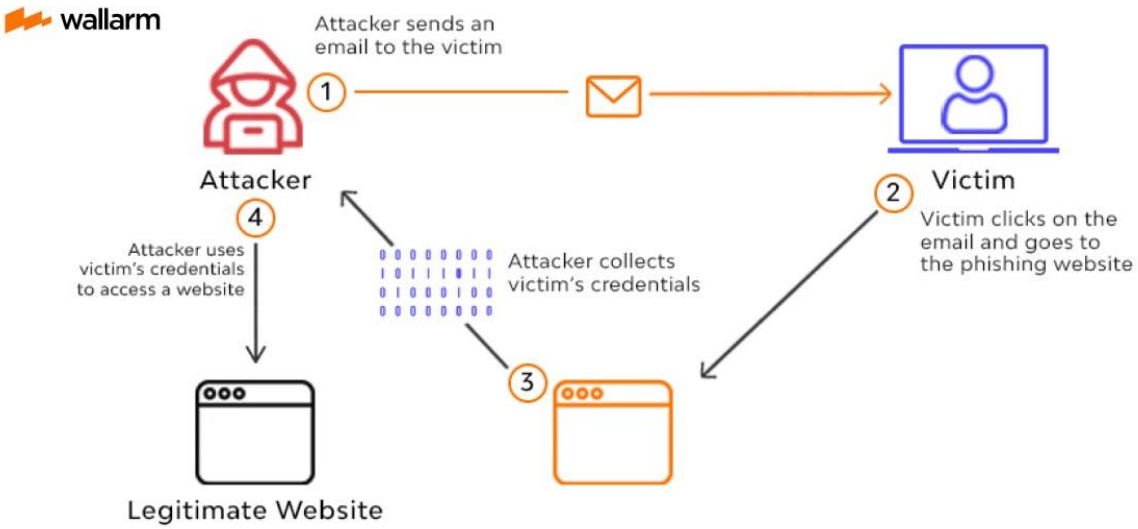

Figure 6: Phishing Attack (Beschokov, 2024).

Impact of Broken Authentication in Web Applications

Web applications are now commonly affected by Broken Authentication, since research shows that a significant number of these systems have this problem. According to research, over 45% of web apps have vulnerabilities related to broken authentication. A failure in limiting the number of authentication attempts is responsible for over one-third of these vulnerabilities. This error enables attackers to begin brute-force attacks, which include repeatedly trying different login and password combinations until the attacker gains unauthorized access to the system. Such vulnerabilities put high risks to the security and integrity of web applications, as they may result to unauthorized access, data breaches, and compromise of sensitive information (Positive Technologies, 2020).

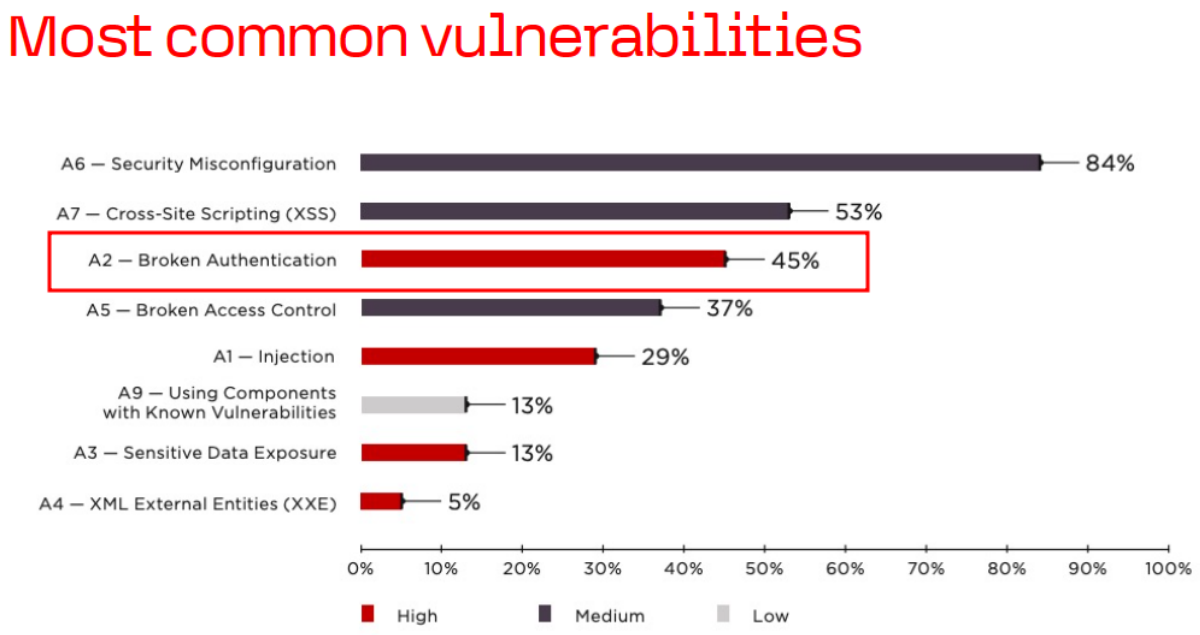

Figure 7: Common Vulnerabilities (Positive Technologies, 2020)

According to OWASP (Open Web Application Security Project), broken authentication is a major threat to websites and APIs. It is ranked as the second worst vulnerability for APIs and the top one for regular websites (Bright, 2024).

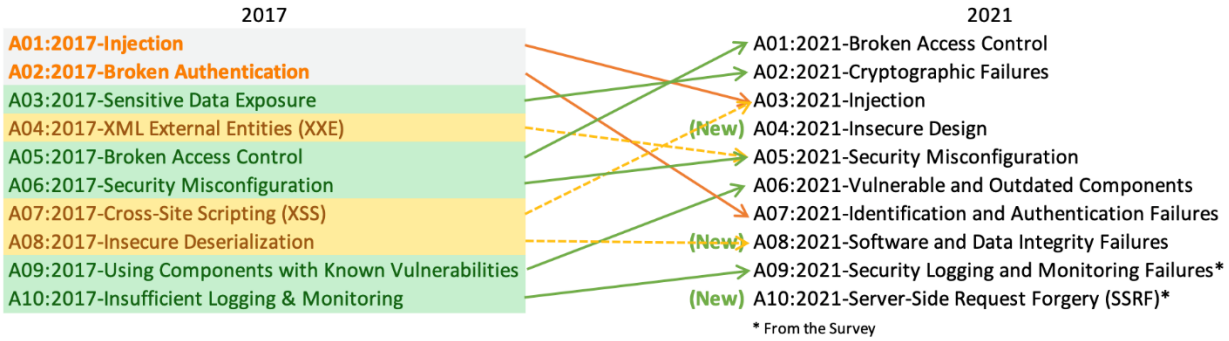

Figure 8: OWASP Top 10 2021 (N, 2021).

The term "Broken Authentication" used to be at the number two spot on the OWASP Top 10 list in 2017, but it's now been moved down to number seven in 2021. This shift is good news and shows that our web apps are becoming more secure. OWASP, the Open Web Application Security Project, noted that the change is partly due to the wider adoption of standardized frameworks that are designed to handle authentication more securely (N, 2021). Previously, broken authentication used to be a huge concern because it covered all the way attackers could enter into systems by pretending to be someone else. This could happen because of Weaknesses in the user verification or session management processes. These kinds of security vulnerabilities are becoming less common due to improvement in secure coding practices and the adoption of strong authentication frameworks and protocols such as OAuth and OpenID Connect. These technologies offer more secure management of identities and sessions, significantly lower the risk of authentication-related attacks. The growing use of Multi-factor Authentication (MFA) and stronger policy enforcement across different platforms has led to an increase in security regarding user authentication procedures.

Secure methods of authentication should stay a top priority for developers and security experts, even though they have moved down the list. Preventing authentication attacks requires regular updates and patches that must be used to authentication frameworks and systems, following the latest security standards, and being careful while putting best practices into action.

A broken authentication vulnerability led to the Equifax data breach in March 2017, which may have exposed the personal information of 143 million customers. Attackers took use of a known vulnerability in Apache Struts that Equifax was aware of but neglected to fix. Sensitive data, including names, addresses, Social Security numbers, and driver's license numbers, were made available to unauthorized individuals through this incident. Afterwards failures in system segmentation, plain text password storage, and expired encryption certificates were made easier by data exfiltration. The harm was made worse by Equifax's improper management of the hack and delayed disclosure, which resulted in criticism and legal consequences (Fruhlinger, 2020).

3. Demonstration

I. Username enumeration via different responses

This attack is done using PortSwigger's Web Security Academy lab. In this attack, I am going to explore the security problem of username enumeration. This occurs when a web application's response to login attempts provides information about the existence of a username. An attacker can determine whether usernames are legitimate, for example, if the answer differs between incorrect username inputs and incorrect password inputs for correct usernames. One can identify legitimate usernames by identifying the differences in responses. This greatly speeds up password cracking attempts because the attacker already knows which usernames to target.

Title	Username enumeration via different responses
Payload Used	Username and password
CVSS	5.3
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Criticality	Medium
OWASP Category	Broken Authentication
Tools Used	Burp Suite

Step 1: Navigating to the website and clicking on “My account”.

Figure 9: Access the Website.

Step 2: Entering “summer” as the username and a password, then attempting to login the website.

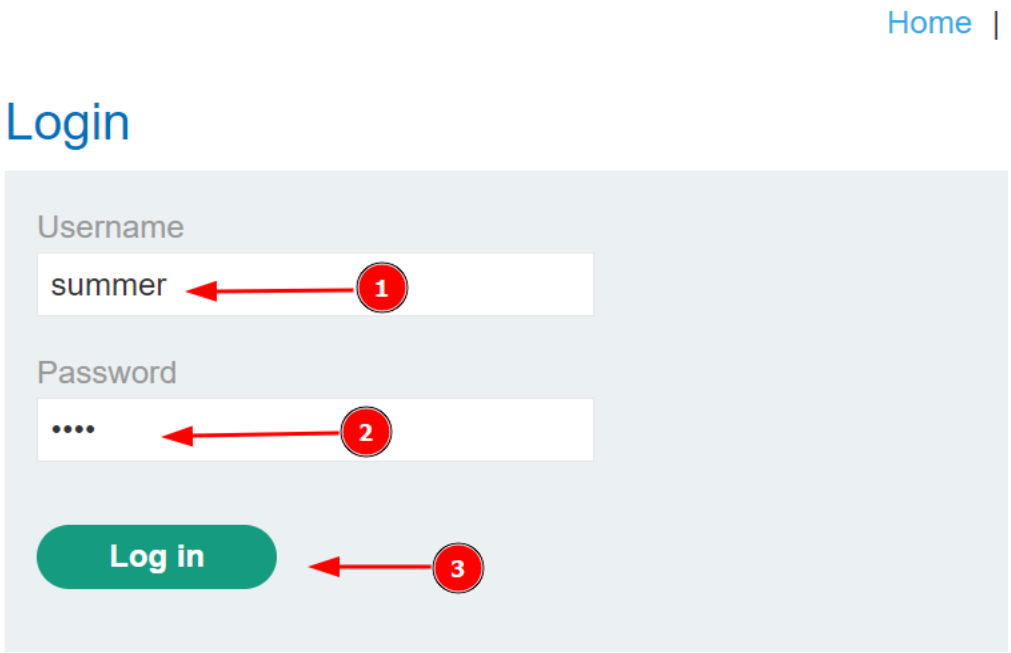

Figure 10: Attempt to log in.

Step 3: In the Burp Suite, locating the POST method for the login attempt. Right-clicking on the request and selecting “Send to Intruder.”

The HTTP POST method sends data to a server to create or update resources. It is commonly used in login forms, to submit data securely from the browser to the server, as the data transmitted via POST is not visible in the URL, unlike the GET method, where data can be seen in the URL.

The "Intruder" is a powerful tool designed for automated and customized testing of web applications. It is useful for tasks like performing attacks on web pages to test the security of their parameters by manipulating and repeating HTTP requests.

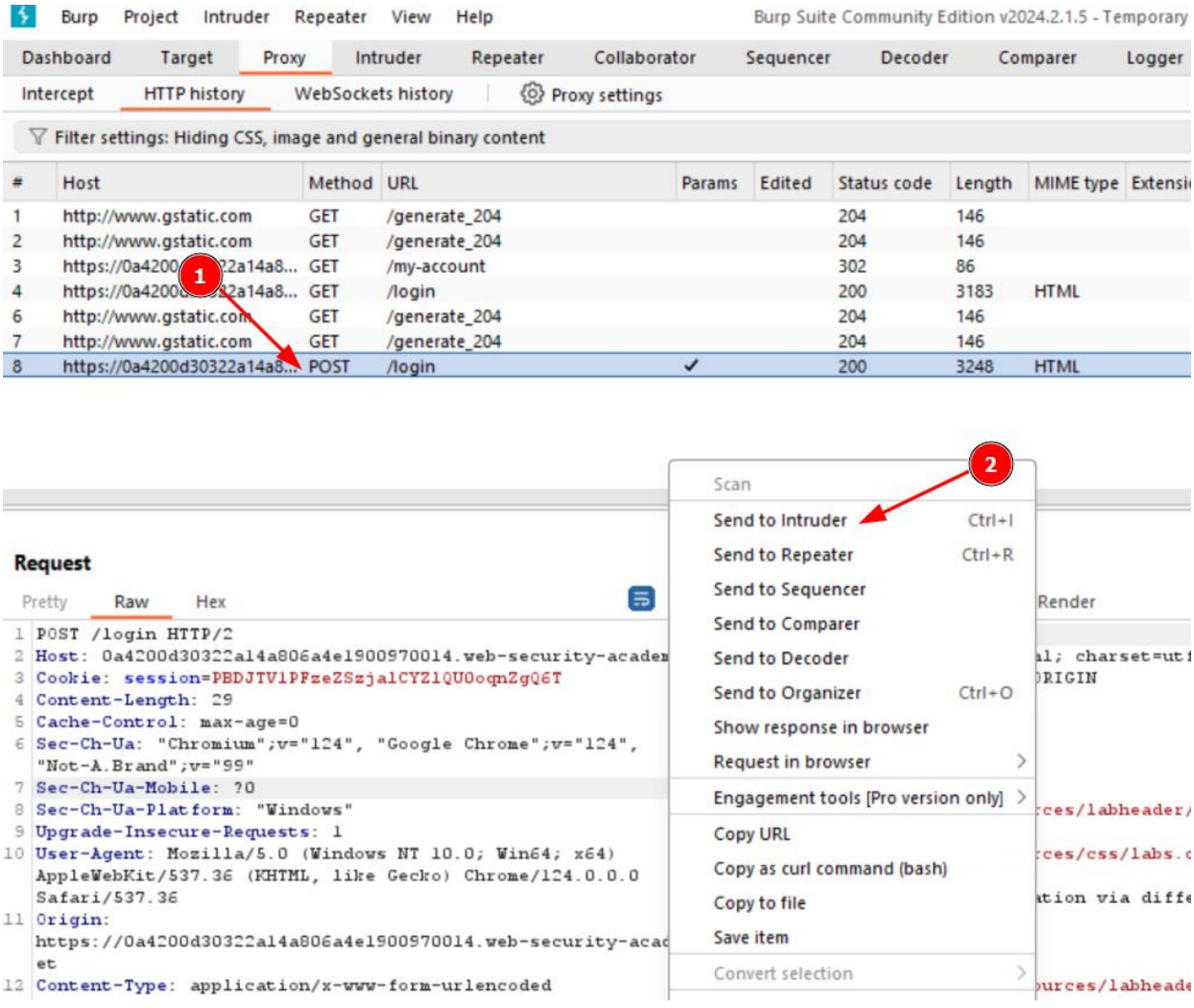

Figure 11: Send Request to Intruder.

Step 4: In Burp Intruder, going to the “Positions” tab, highlighting the “username” field, and clicking on “Add” to add/insert the payload positions.

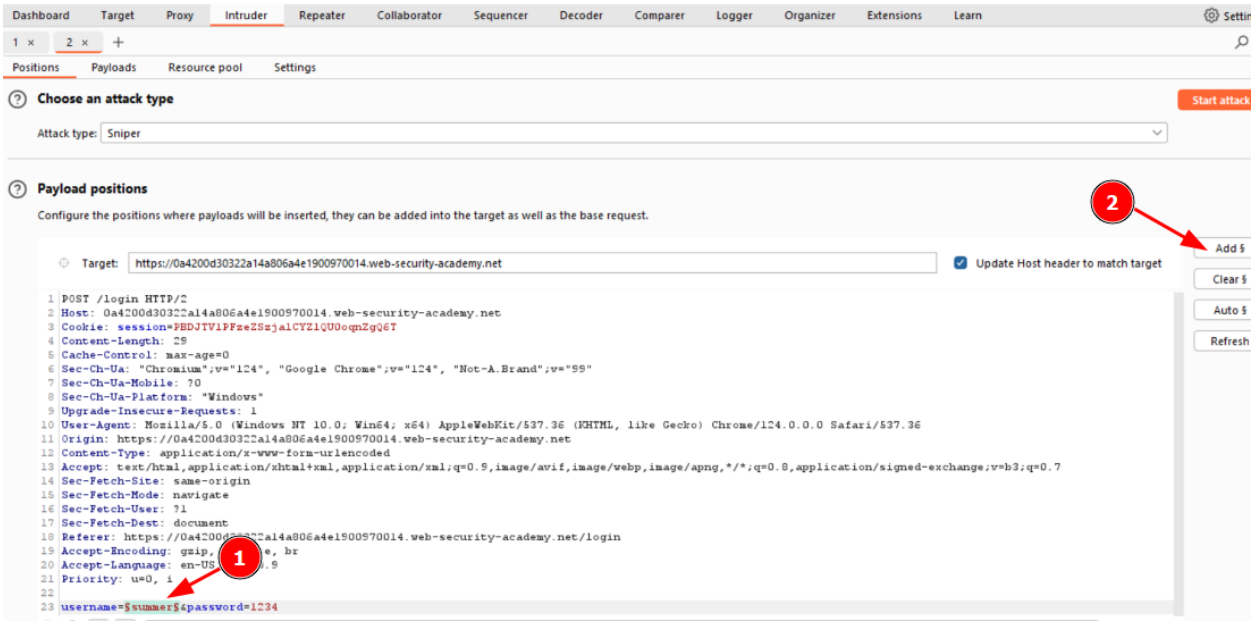

Figure 12: Set Payload Position.

Step 5: A list of potential usernames is given. Copying the username.

```
albuquerque
alerts
alpha
alterwind
am
amarillo
americas
an
anaheim
analyzer
announce
announcements
antivirus
ao
ap
apache
```

Figure 13: Copy username.

Step 6: In Burp Intruder, navigating to the “Payloads” tab, setting the payload type to “Simple list”, and pasting the list of usernames.

Starting the attack.

Payloads tab: This is the configuration section for the data (also known as "payloads") that will be added to the HTTP request.

Simple list: This enter a list of objects (usernames in this case) that will be used one at a time in the attack. For every attack attempt, a single item from the list will take the place of a specified part of the HTTP request (such the username field).

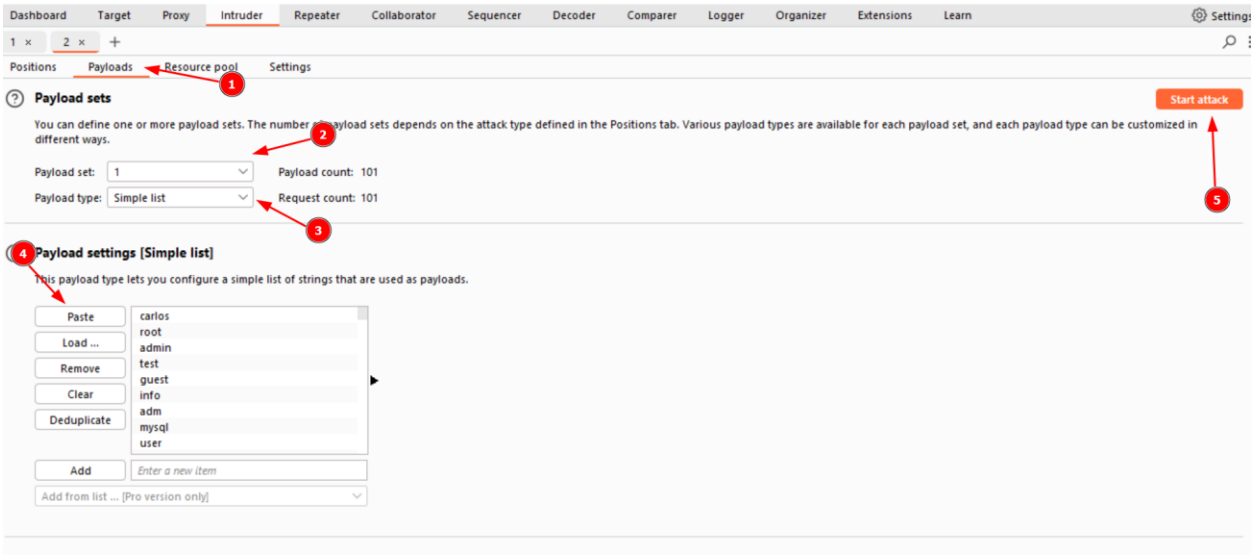

Figure 14: Start Username Enumeration Attack.

Step 7: To distinguish the username from the list of payloads shown, their length values are compared. The length 3250 of “Anaheim” is different than other payload, indicating that “Anaheim” is the valid username.

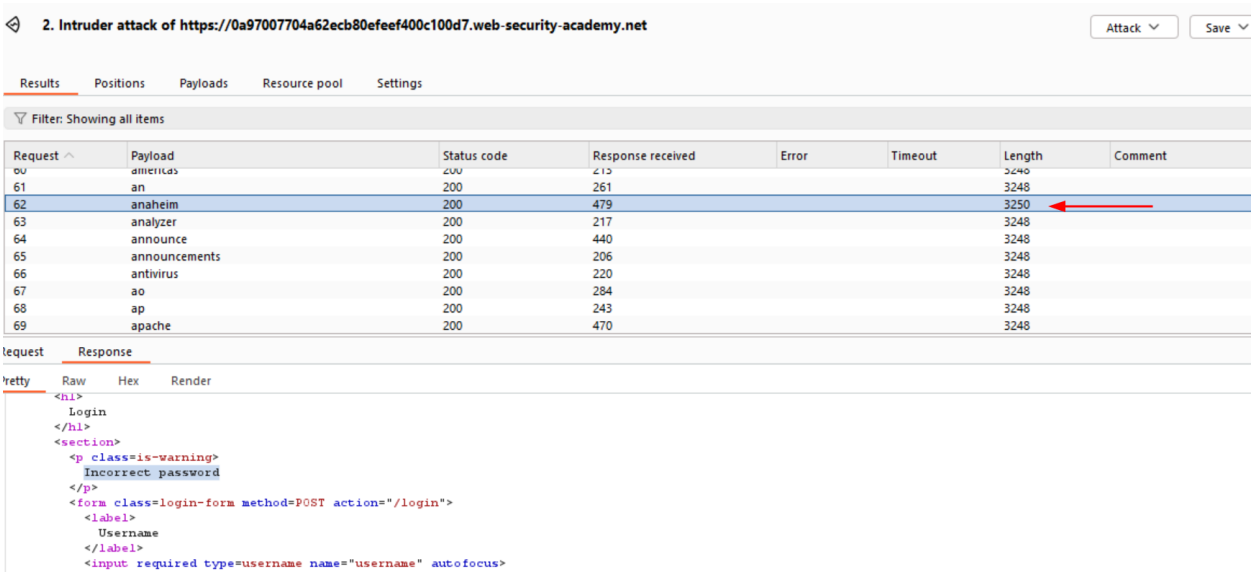

Figure 15: Identify Valid Username.

Step 8: Return to Login page and entering the username Anaheim which shows incorrect password.

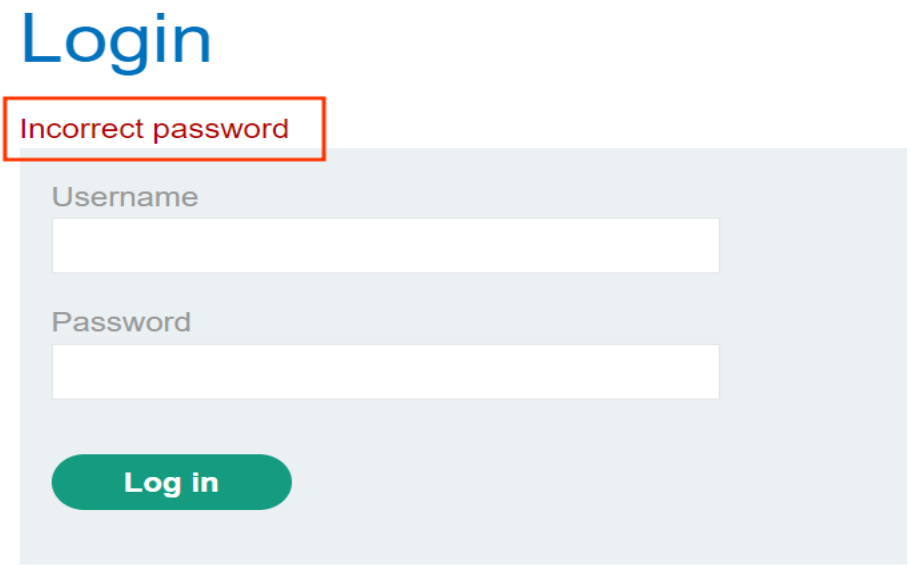

Figure 16: Incorrect Password.

Step 9: In the Burp Intruder, resetting the positions and changing the username as “anaheim” and setting the “password” field as the new payload position.

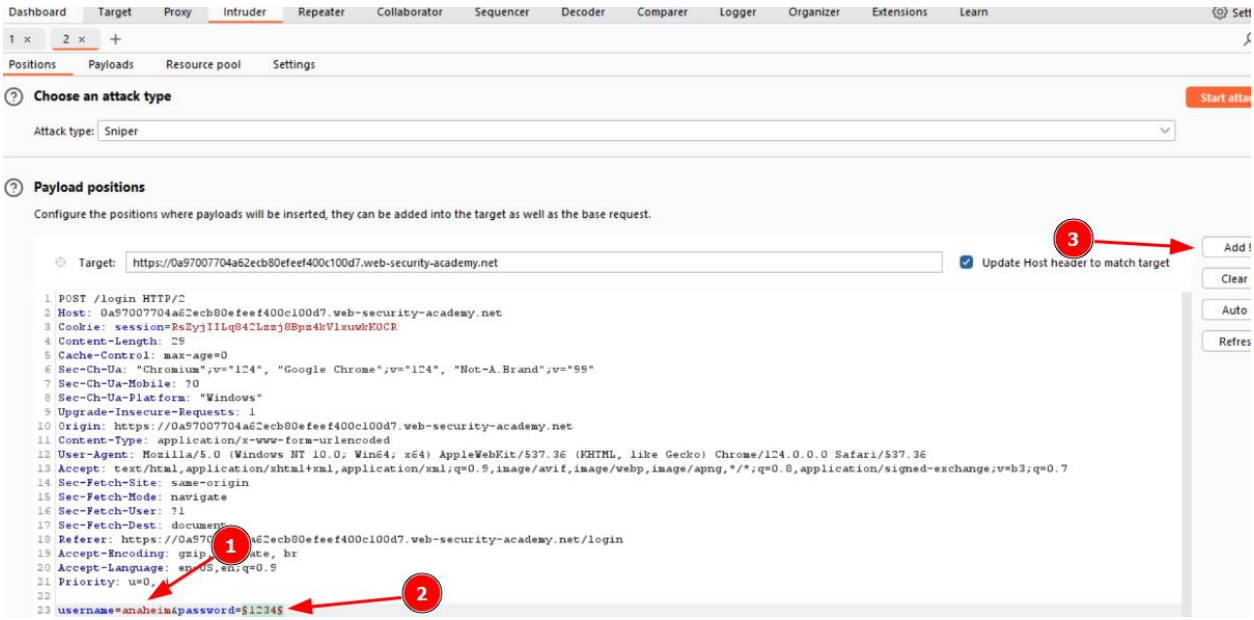

Figure 17: Setup Password Brute Force.

Step 10: A list of potential passwords is given. Copying the passwords.

```
123456
password
12345678
qwerty
123456789
12345
1234
111111
1234567
dragon
123123
baseball
abc123
football
monkey
letmein
```

Figure 18: Copy the passwords.

Step 11: Navigating to “Payloads”, clearing the previous payloads then pasting the list of passwords, and starting the attack.

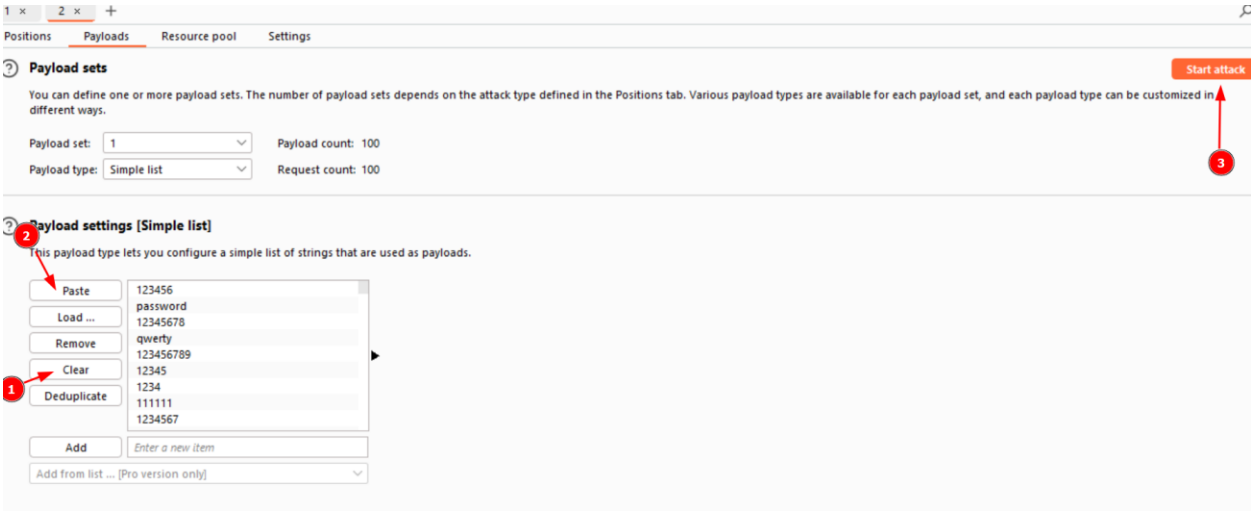

Figure 19: Start password Brute Force Attack.

Step 12: After the attack the length 189 of Payload 123123 is shorter and different from other length, indicating that 123123 is the password for anaheim.

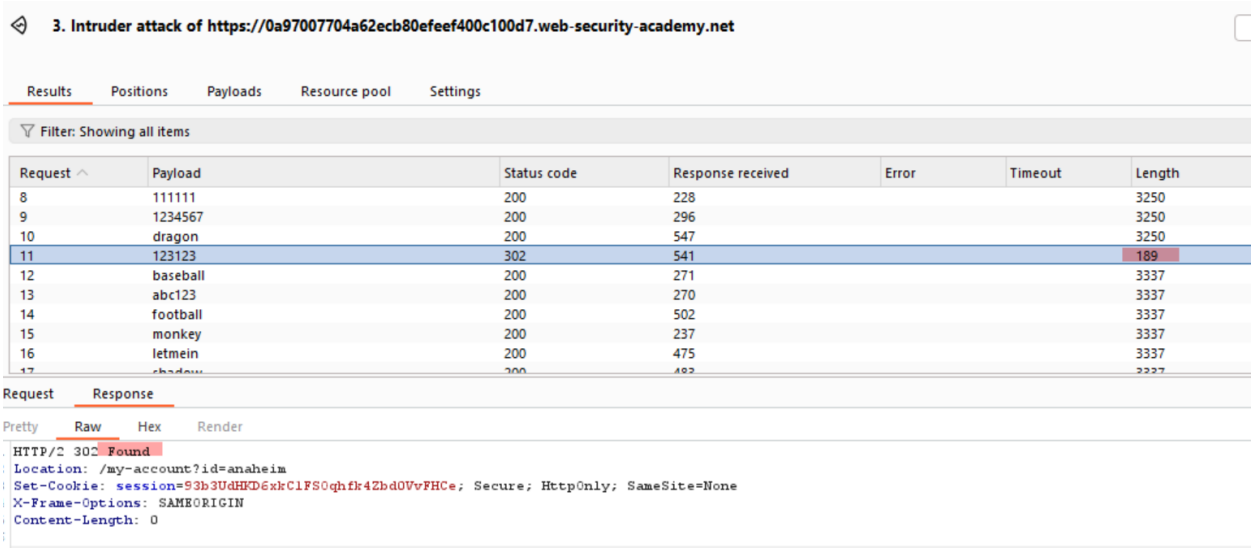

Figure 20: Identify Valid Password.

Step 13: To check if the username and password is correct or not. Return to the Login page, entering “anaheim” as username and “123123” as the password and clicking on Log in button.

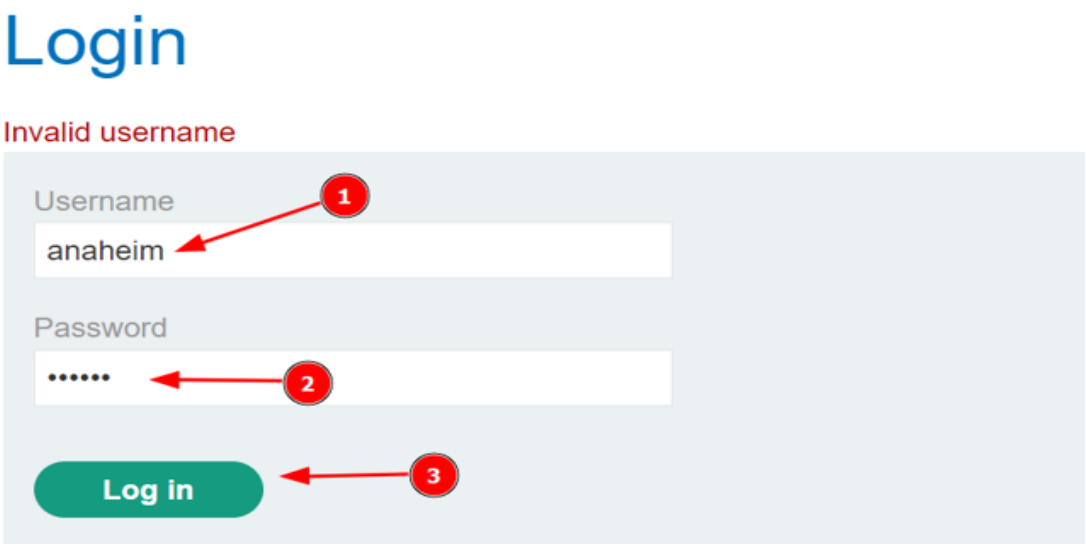

Figure 21: Check Username and Password.

Step 14: The username and password are correct.

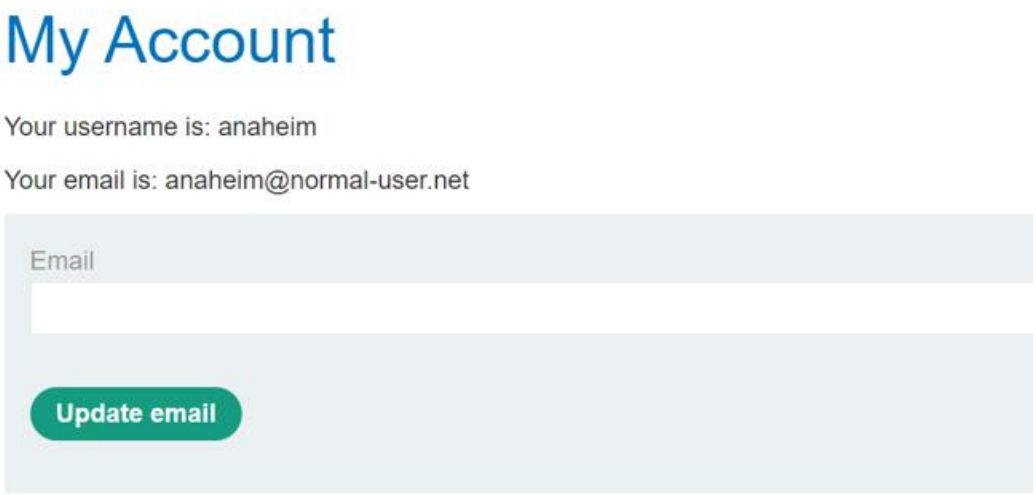

Figure 22: Successful Login confirmed.

II. Broken brute-force protection, IP block

In this attack, I will be able to exploit a weakness in how a web application responds to login attempts. To prevent brute-force attacks, the system will block an IP address if failed to log in too many times. This is meant to prevent attackers from trying repeated password and username combinations in an attempt to obtain unauthorized access. We'll see whether there is a method over this security precaution. We can reset the count of repeated failed login attempts by alternating between the credentials of the intended victim and those of a known, legitimate user. By using this method, the system is kept from reaching the limit which would normally block our IP address, allowing us to keep trying different password combinations until we discover the one that works.

Title	Broken brute-force protection, IP block
Payload Used	Username and password
CVSS	6.5
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N
Criticality	Medium
OWASP Category	Broken Authentication
Tools Used	Burp Suite

Step 1: Navigating to the website and clicking on “My account”.

Figure 23: Access My Account Page.

Step 2: Entering “summer” as the username and a password, then attempting to login the website.

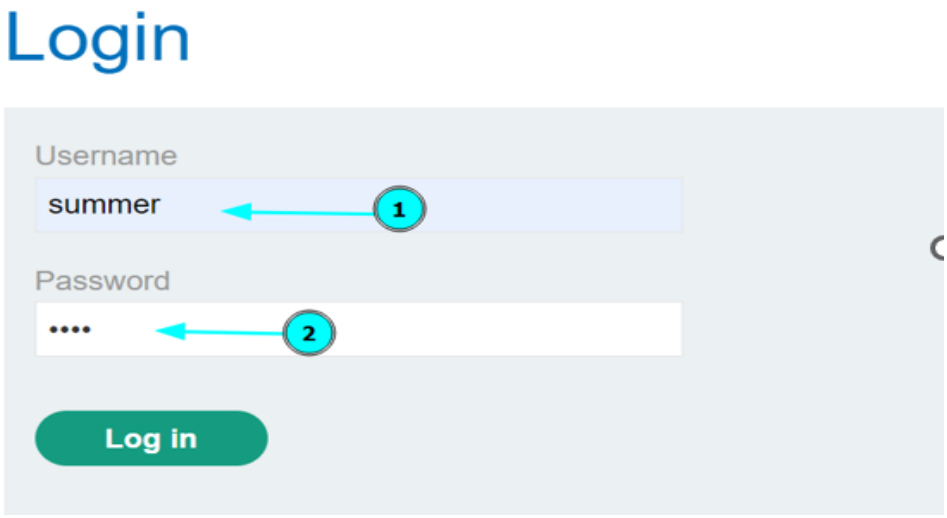

Figure 24: Initial Login Attempt.

Step 3: The response indicates an invalid username.

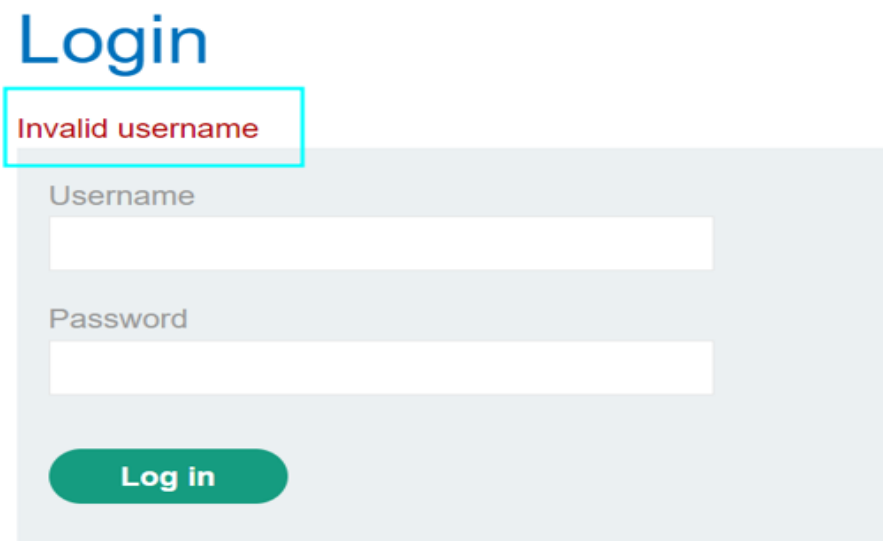

Figure 25: Invalid Username.

Step 4: In the Burp Suite, locating the POST method for the login attempt. Right-clicking on the request and selecting “Send to Intruder.”

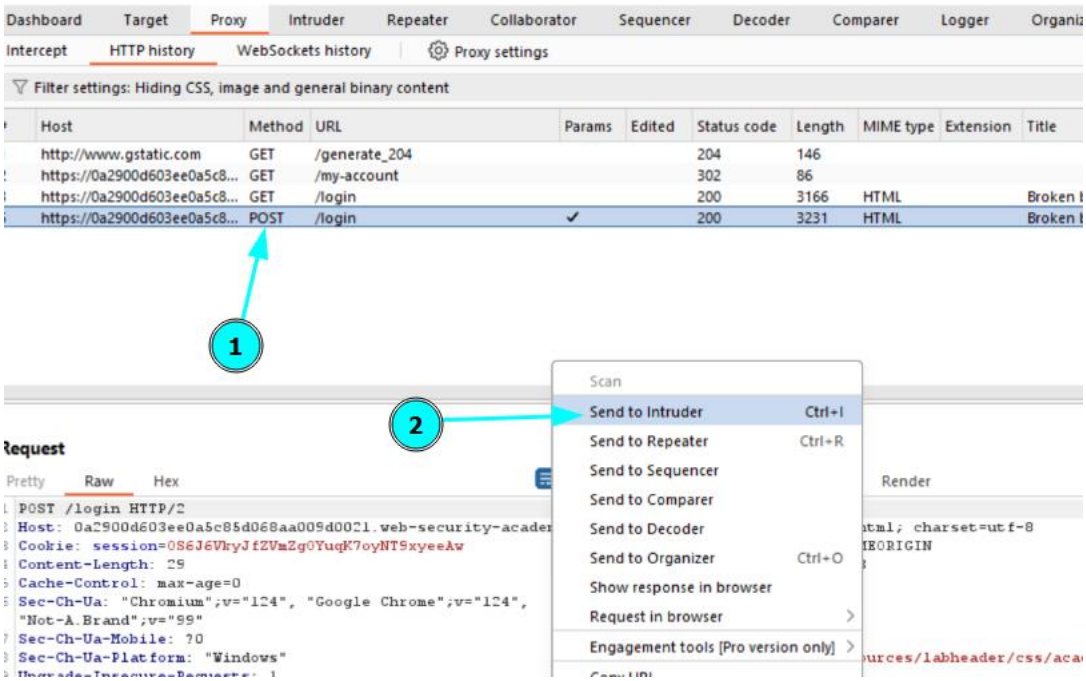

Figure 26: Send Request to Intruder.

Step 5: Returning to the Login page and entering a random username and password.

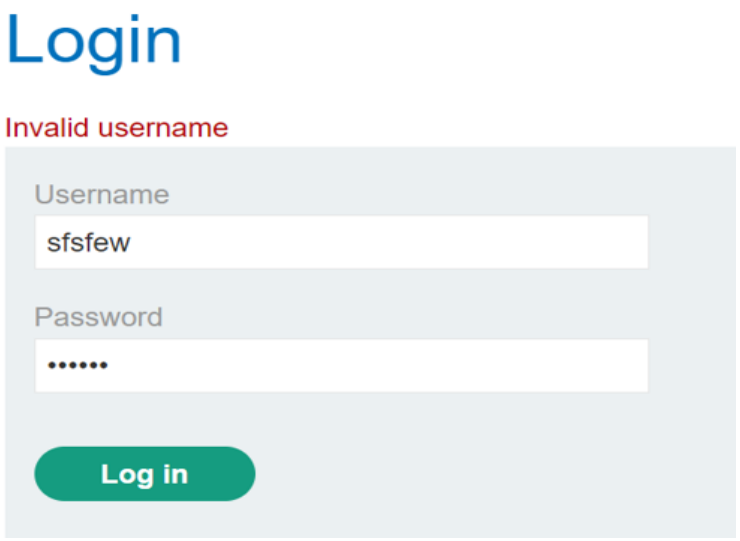

Figure 27: Login attempt.

Step 6: Continue entering random usernames and passwords until a notification of too many failed attempts is received.

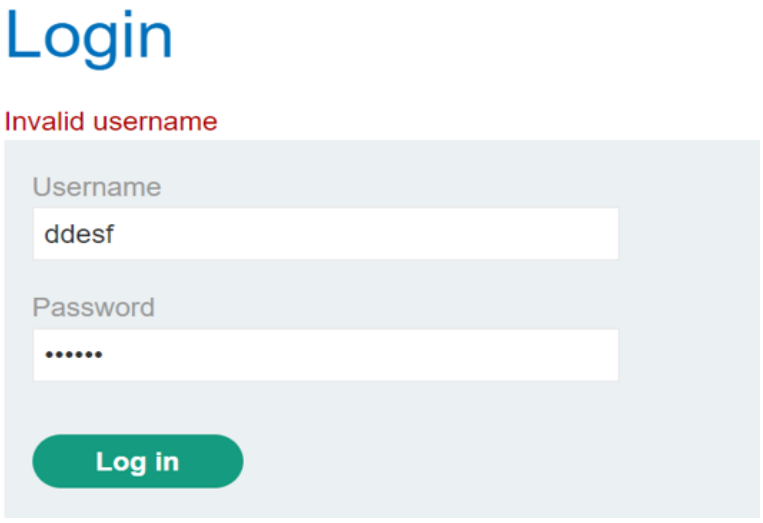

Figure 28: Repeated Login Attempts.

Step 7: After entering again username and password, the message that too many incorrect login attempts have been made, indicating an IP block.

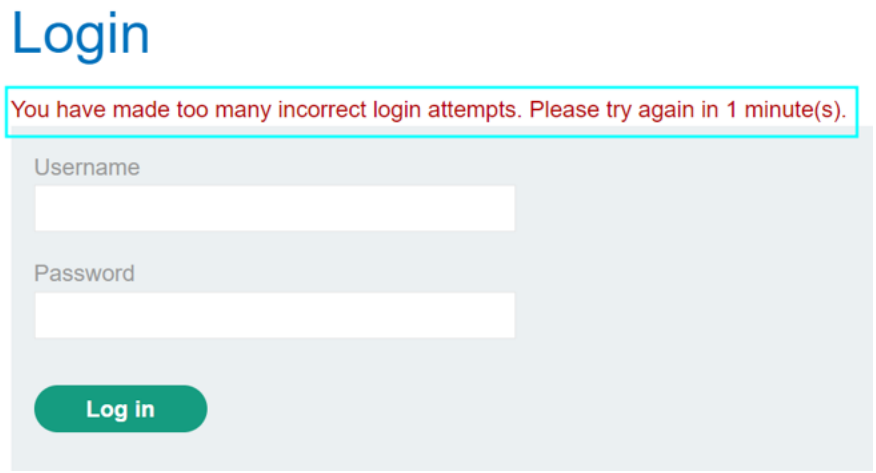

Figure 29: IP Block Notification.

Step 8: Returning to Burp Intruder, in positions selecting the username field, and setting the attack type to pitchfork.

- "Pitchfork" in Burp Intruder allows user to systematically test multiple inputs across different fields in a web form by using parallel payloads from multiple lists and stops when the shortest list runs out of entries.

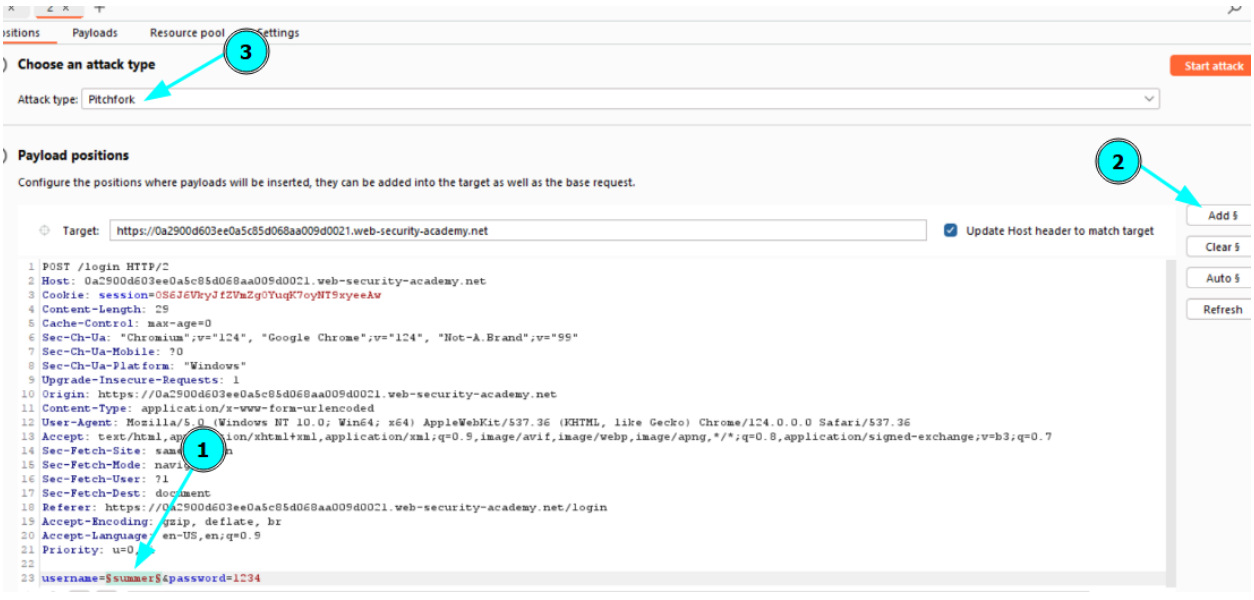

Figure 30: Configure Burp Intruder.

Step 9: Creating a list alternating between known valid username “wiener” and the victim’s username “carlos”. Copying the username.

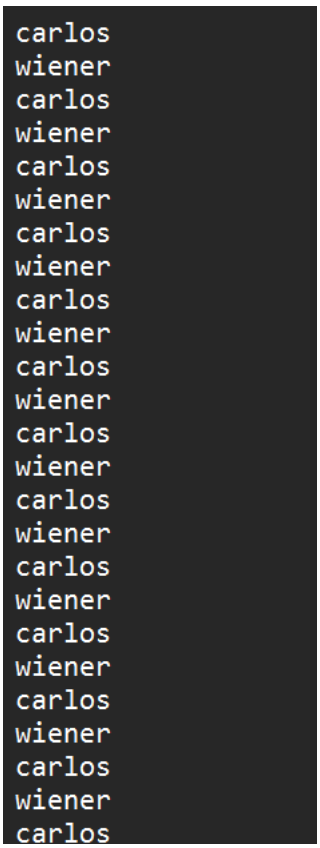

Figure 31: Copy Username.

Step 10: In Burp Intruder, under the “Payloads” tab, pasting the alternating list of usernames.

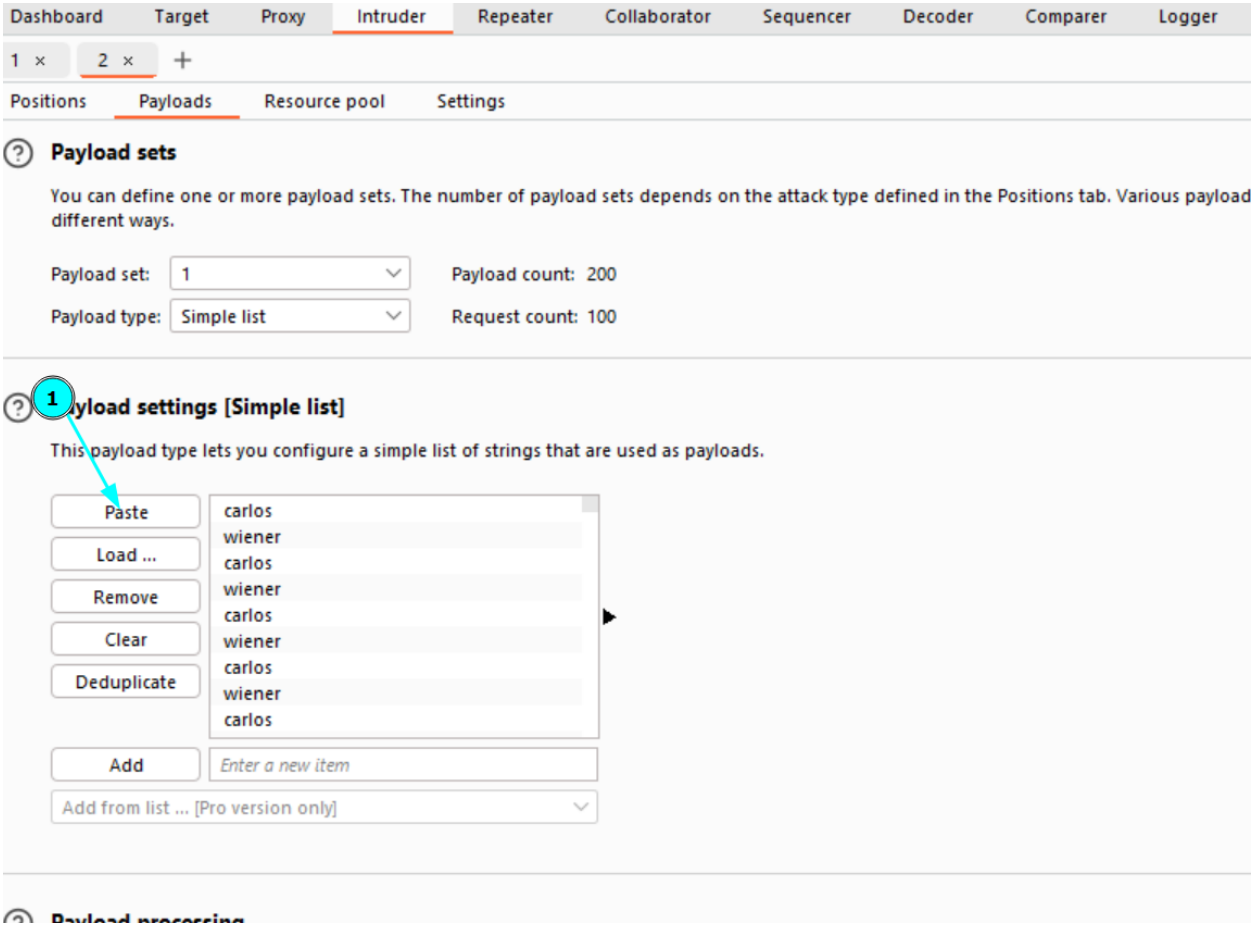

Figure 32: Pasting Username.

Step 11: Return to the “Position” tab, selecting the password field, and adding it to the payloads.

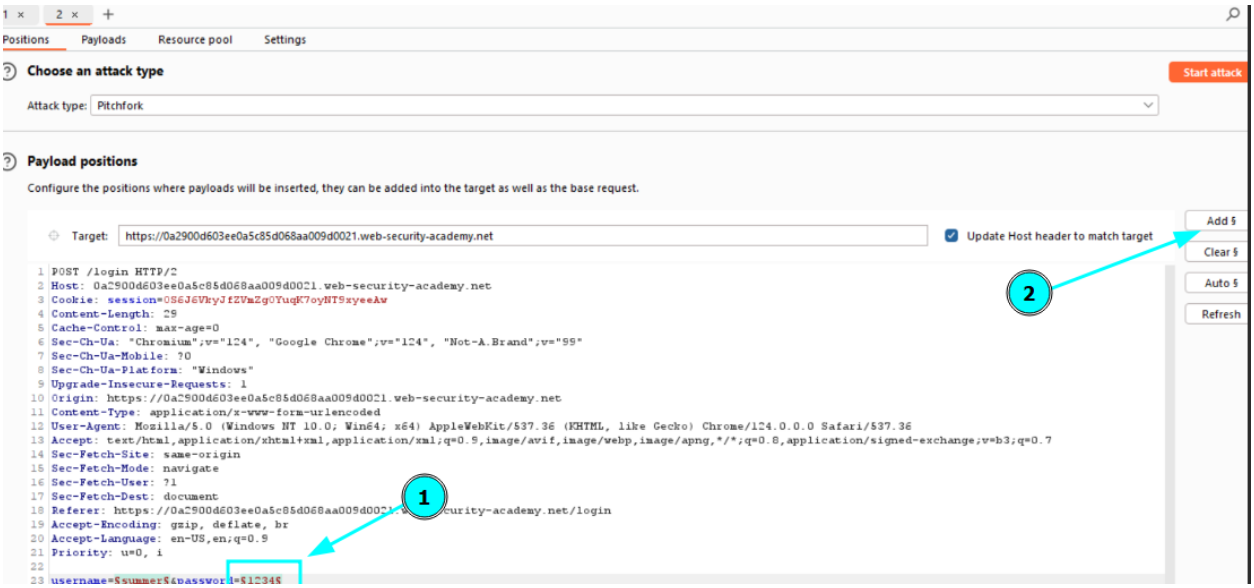

Figure 33: Set Password Field.

Step 12: Arranging a list of candidate passwords where known valid password “peter” is placed in alternation. Copying the password.

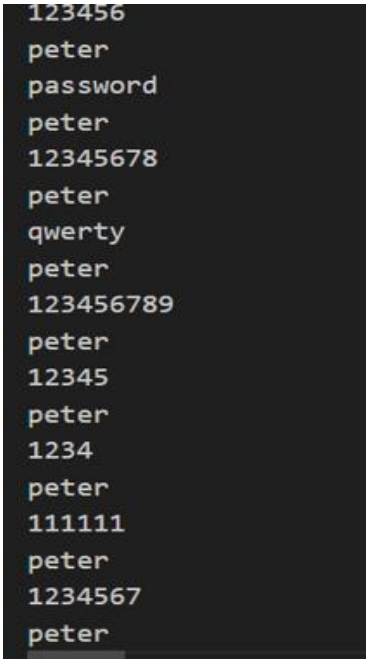

Figure 34: Copying the Passwords.

Step 13: In the payloads tab, setting payload to 2, pasting the alternating password list, and starting the attack.

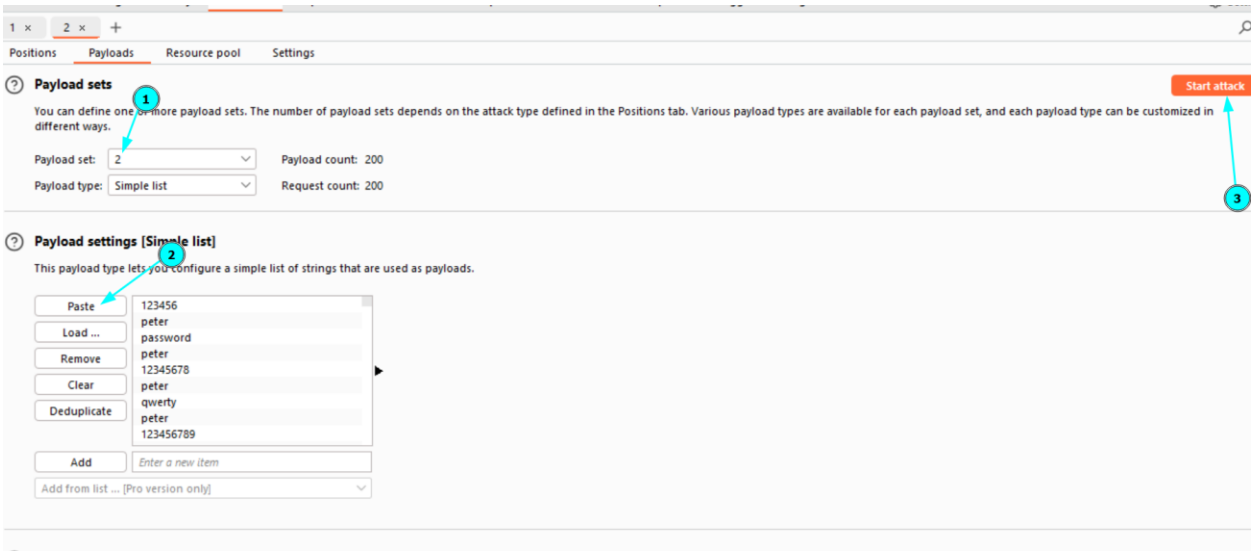

Figure 35: Start the Attack.

Step 14: Ater the attack, wiener of payload 1 and peter of payload2 length is given 188. Searching for carlos length also188.

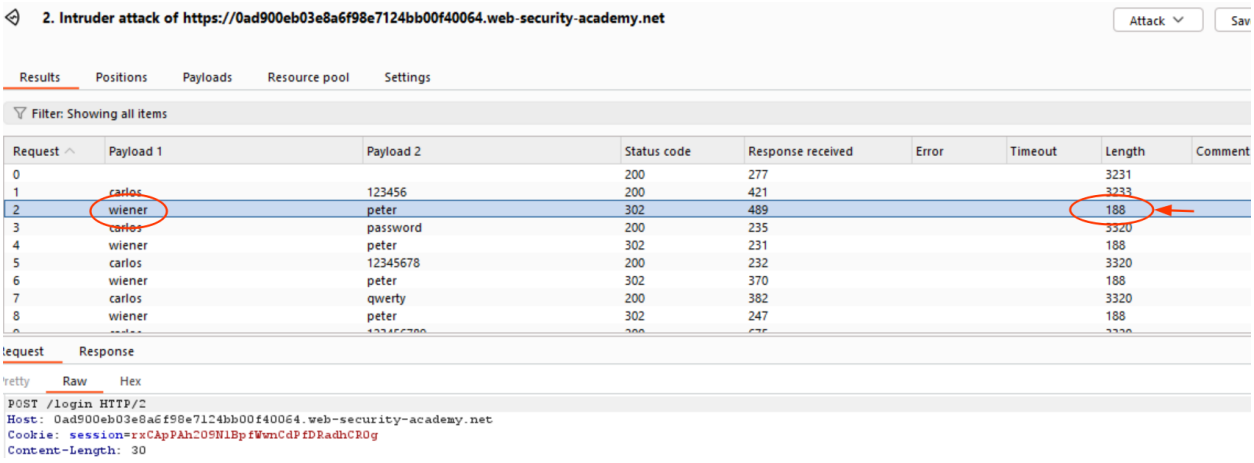

Figure 36: Analyse Attack Result.

Step 15: From the attack results, the password corresponding to “carlos” having length 188 has payload 2 as 12345 which means that the password for carlos is 12345.

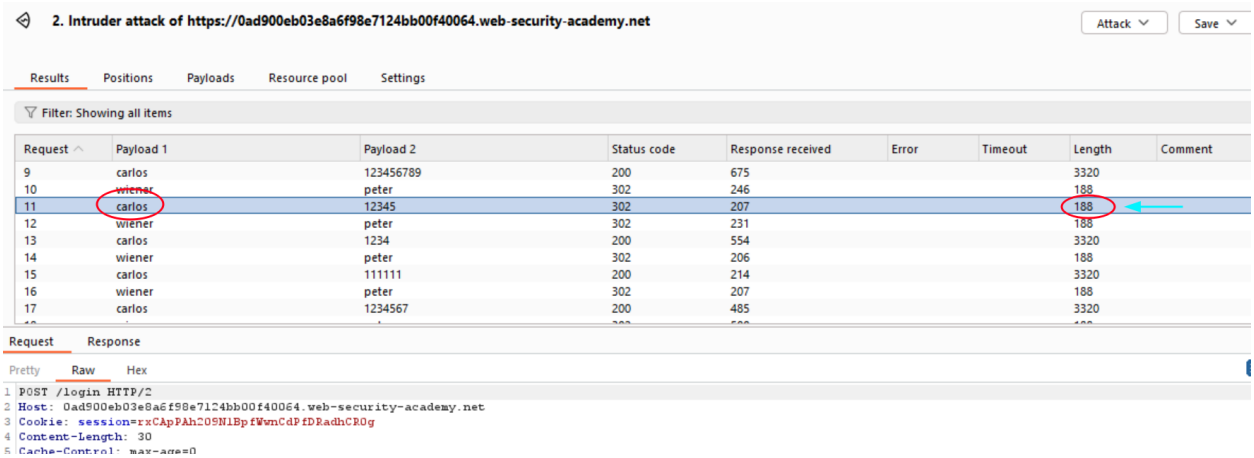

Figure 37: Identify Correct Password.

Step 18: Return to the login page, entering “carlos” as the username and “12345” as password which successfully login as carlos.

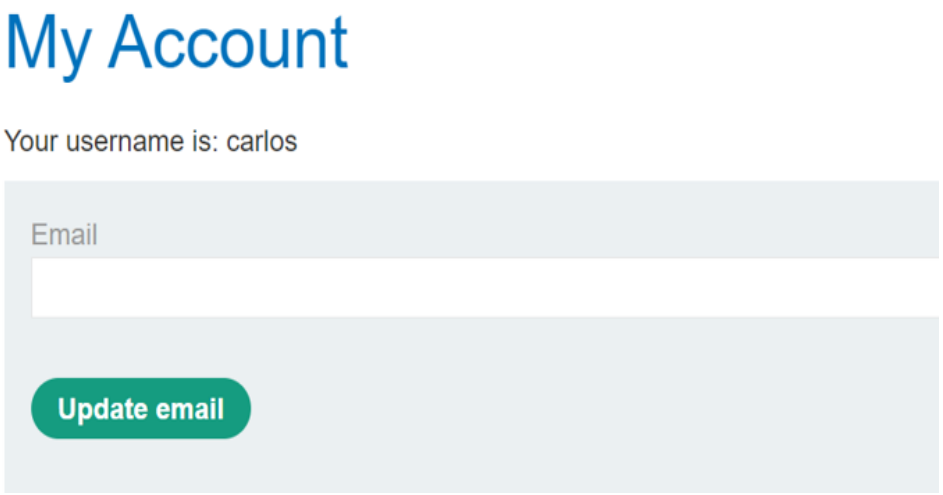

Figure 38: Successful Login

III. Password reset poisoning via middleware.

In this attack, I have to use middleware techniques to attack a vulnerability called "password reset poisoning". This type of attack includes redirecting or intercepting data as it moves between the client and the server by modifying web traffic. The configuration replicates a scenario in which one can access an email client on an exploit server using the login credentials of a known user. By taking control of the password reset process, the goal is to reset and modify the password of another user.

Title	Password reset poisoning via middleware.
Payload Used	Usernames Modified URL for password reset. Forged X-Forwarded-Host header values
CVSS	10.0
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N
Criticality	critical
OWASP Category	Broken Authentication
Tools Used	Burp Suite, Exploit Server, Web Browser

Step 1: Navigating to the website and clicking on “My account”.

Figure 39: Access My Account Page.

Step 2: Clicking on “Forgot password?”

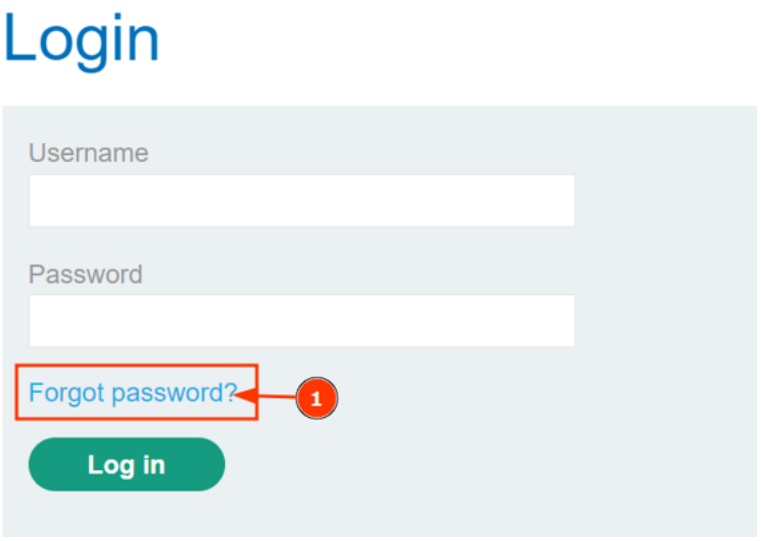

Figure 40: Forgot Password Option.

Step 3: Entering username as “wiener” into the password reset form.

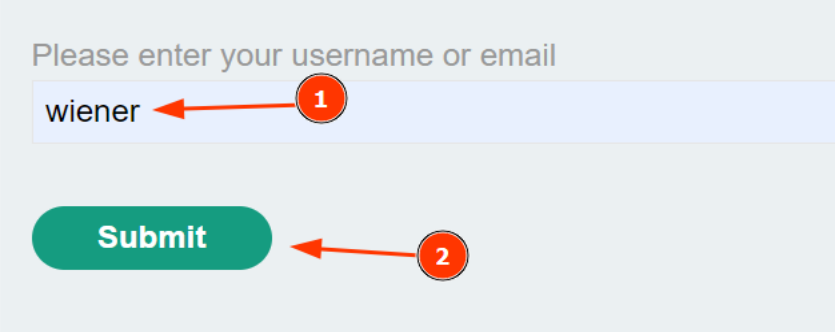

Figure 41: Submit Username for Reset.

Step 4: Navigating to the exploit server.

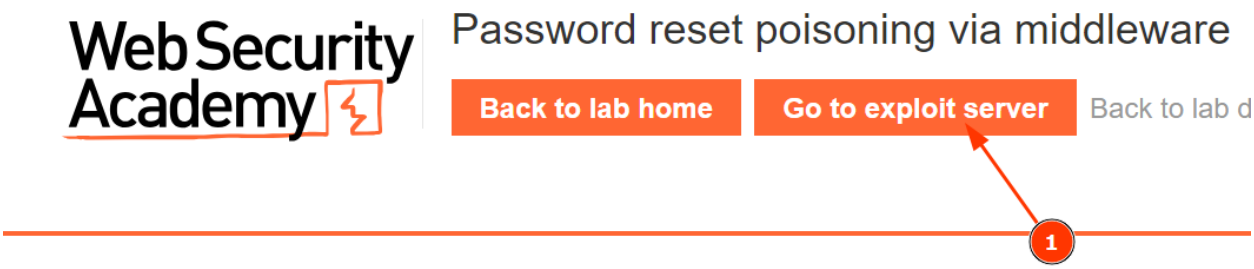

Figure 42: Access the Exploit Server.

Step 5: On the exploit server, clicking on the Email client to check for received emails.

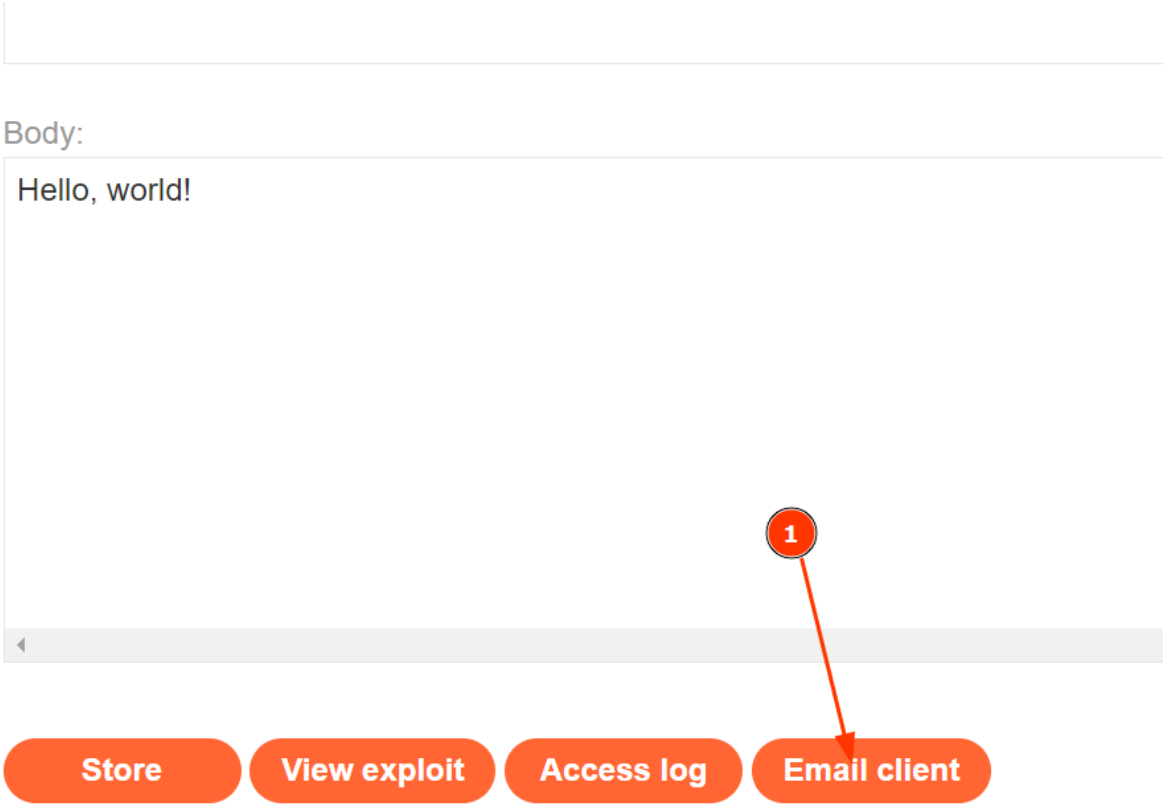

Figure 43: Exploit Server Email Client.

Step 6: A link is provided in the email to reset the password.

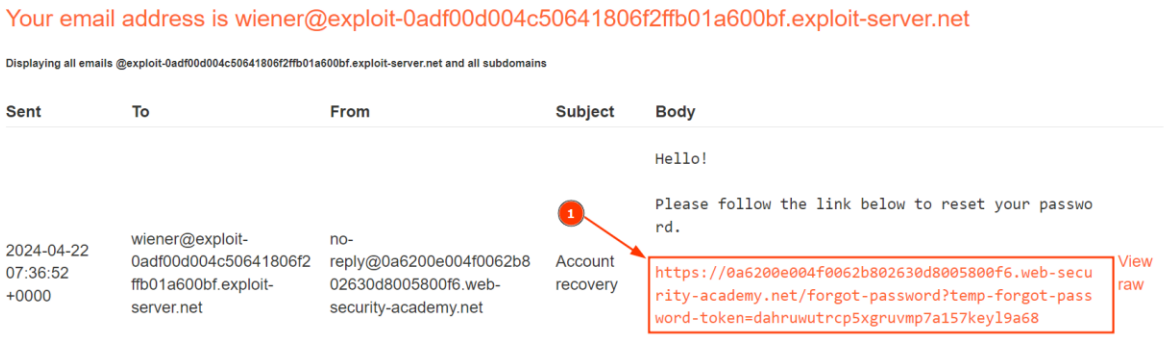

Figure 44: Password Reset Link in the Email.

Step 7: In the Burp Suite, locating the POST method used for the forget-password attempt. Right-clicking on the request and selecting “Send to Repeater.”

- Repeater: The Burp Repeater is used to manually test the security and functionality of web applications by manipulating and resending HTTP requests to the server.

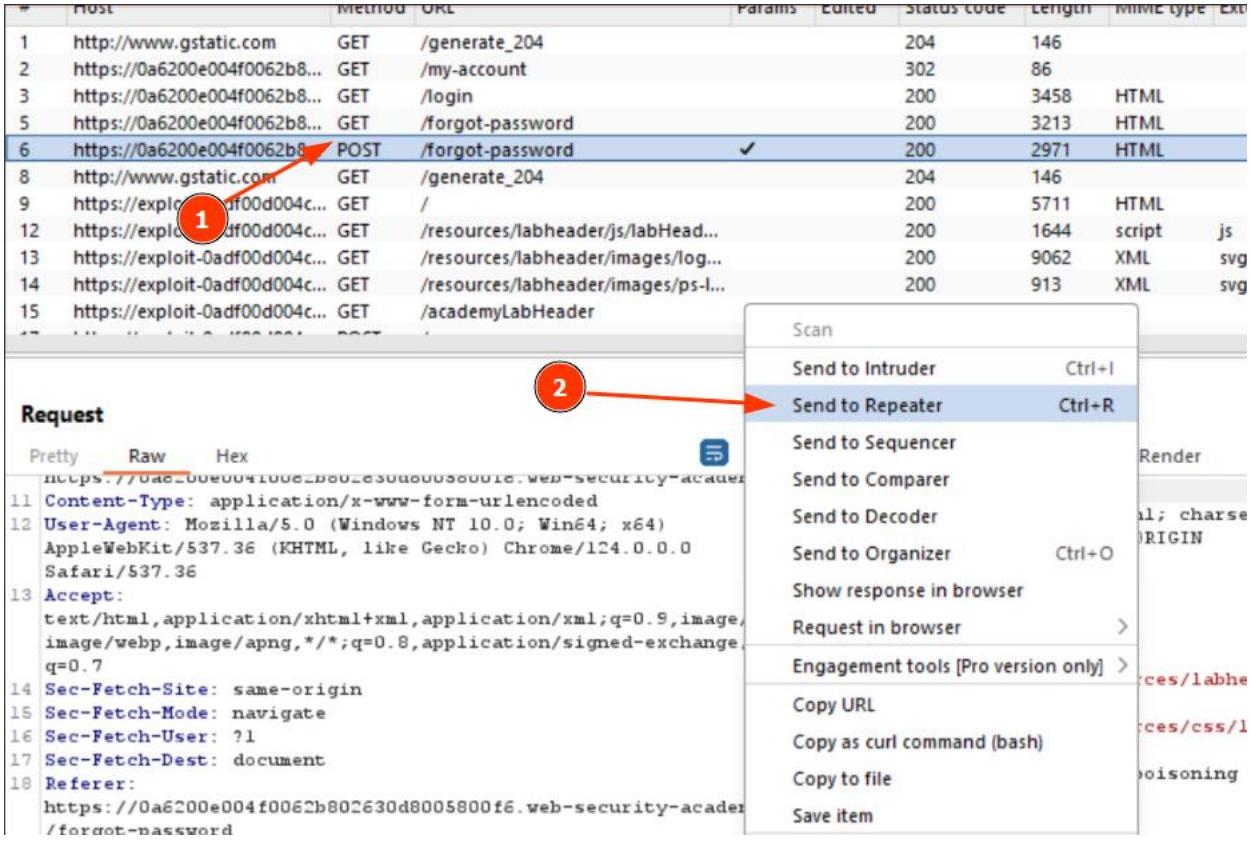

Figure 45: Send Request to Repeater.

Step 8: In the Burp Repeater, adding the “X-Forwarded-Host” header.

- The X-Forwarded-Host header shows the original host requested by the client in a proxy server, keeping the hostname for proper server responses and security handling.

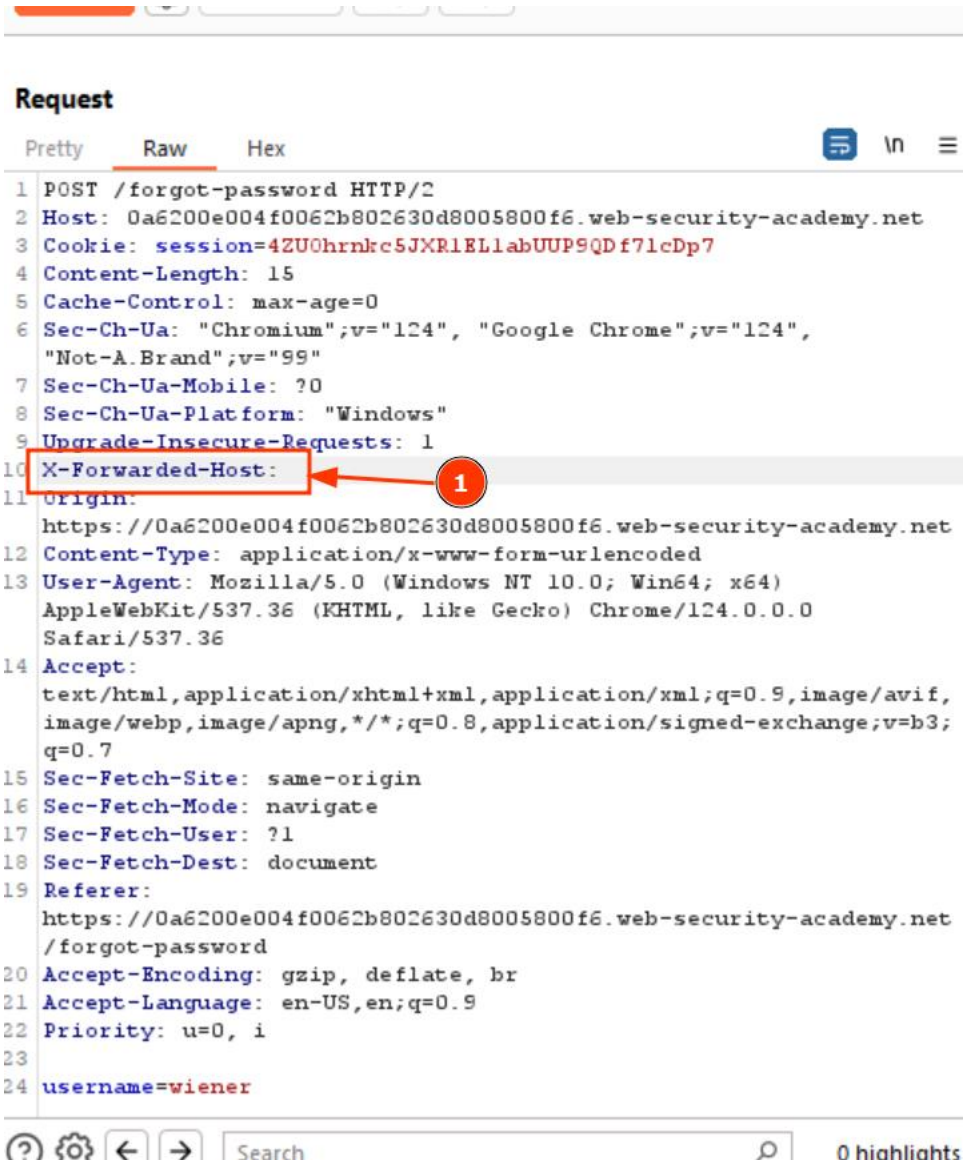

Figure 46: Modify Headers in Burp Repeater.

Step 9: Return to the web and copy the URL from the exploit server.

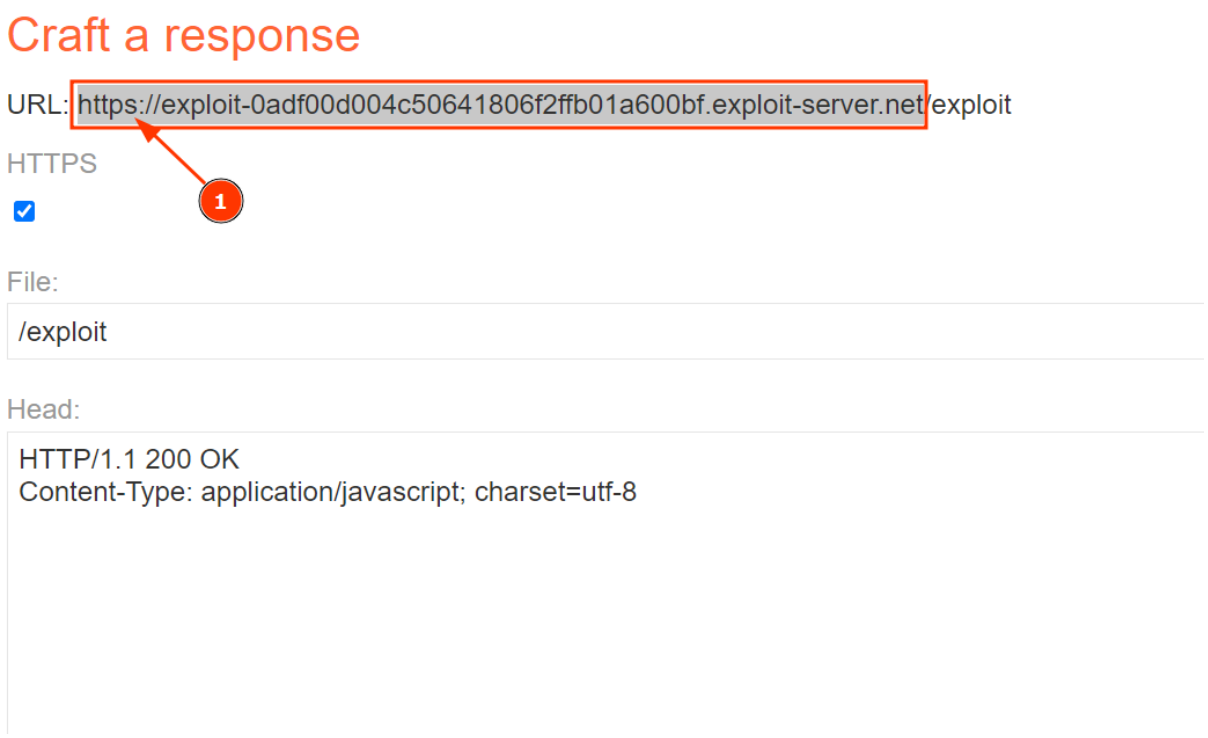

Figure 47: Copy URL.

Step 10: Pasting the copied URL into the “X-Forwarded-Host” header and changing the username parameter to “carlos” in Burp Repeater. Then, sending the modified request.

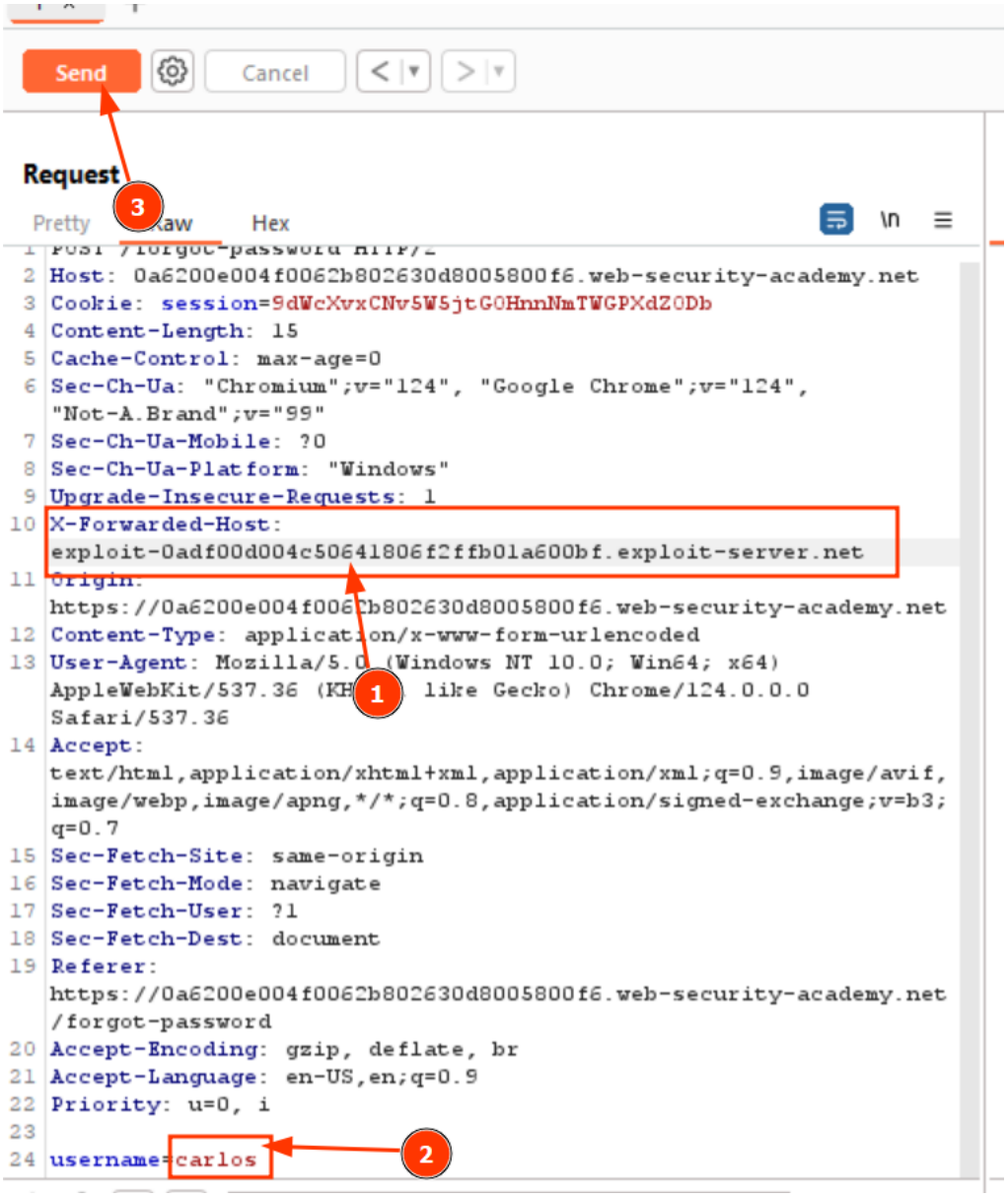

Figure 48: Use Modifies URL in Attack.

Step 11: Return to web application and view the access log.

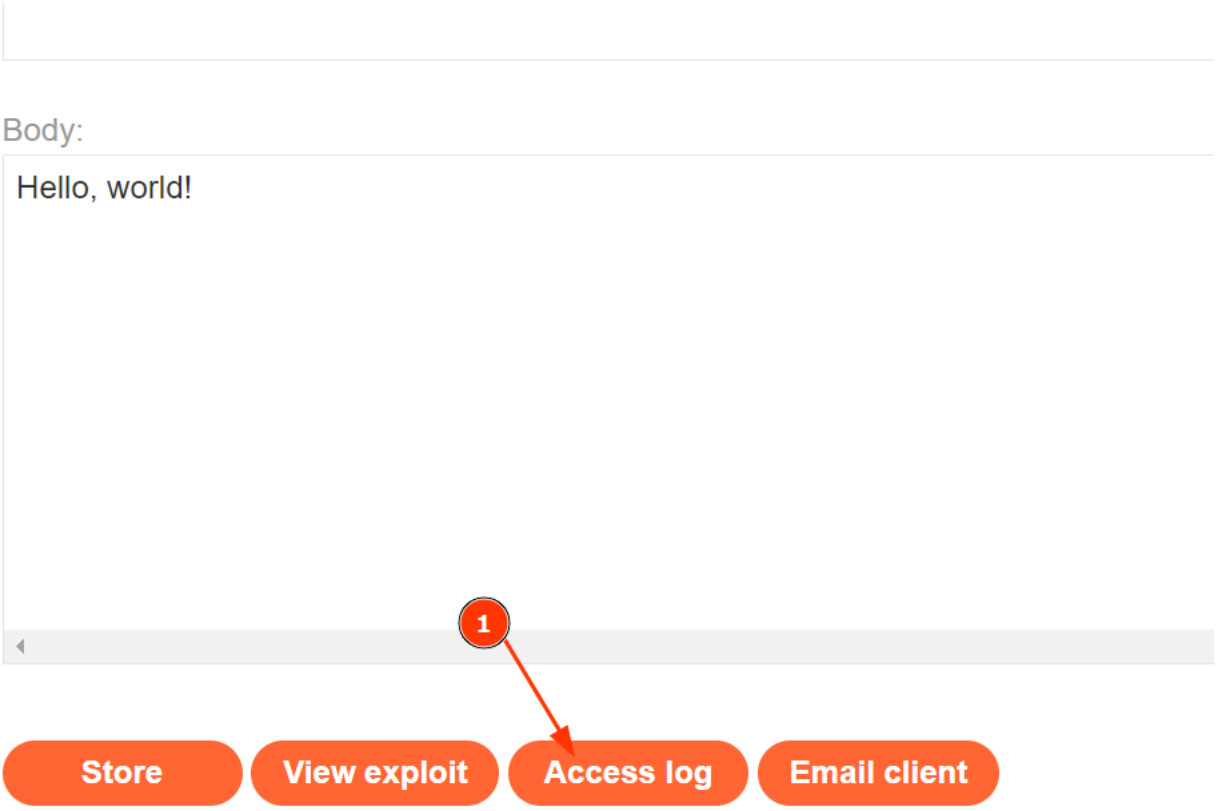

Figure 49: View Access Log.

Step 12: A different IP address can be seen, copying the “forget-password-token” associated with it.

Figure 50: Copy Forgot-Password Token.

Step 13: Going back to exploit server and opening the email client again.

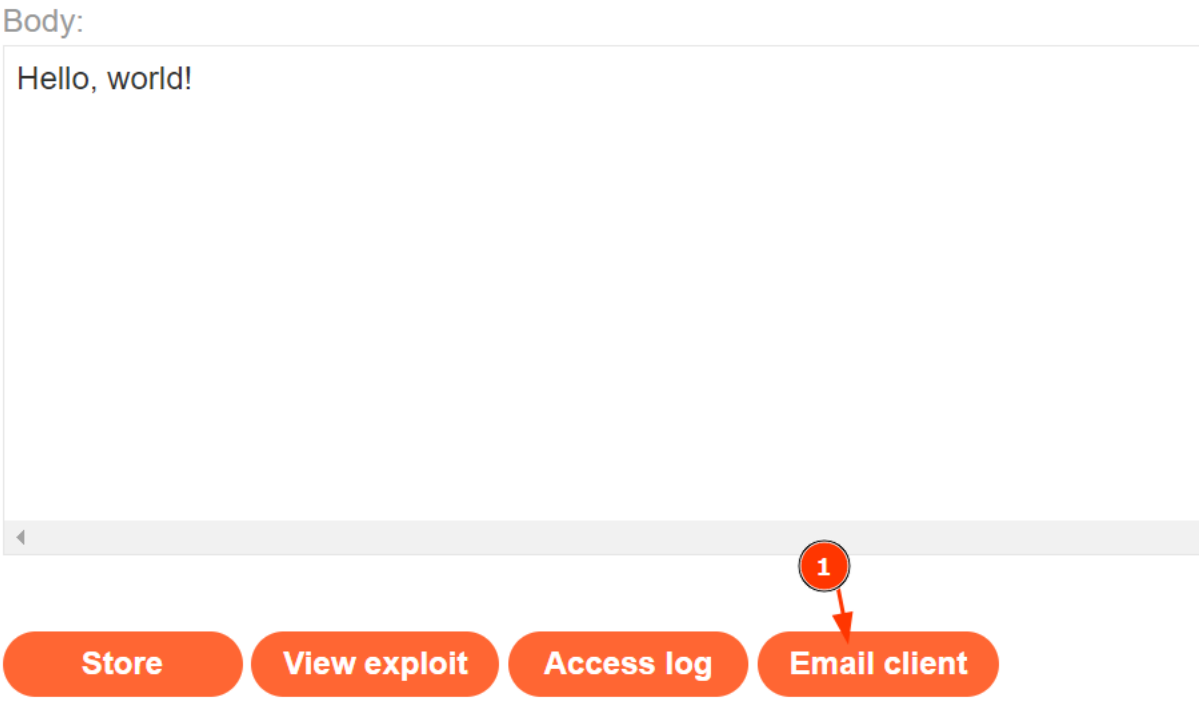

Figure 51: Return to Email Client.

Step 14: Copying the password reset link from the email.

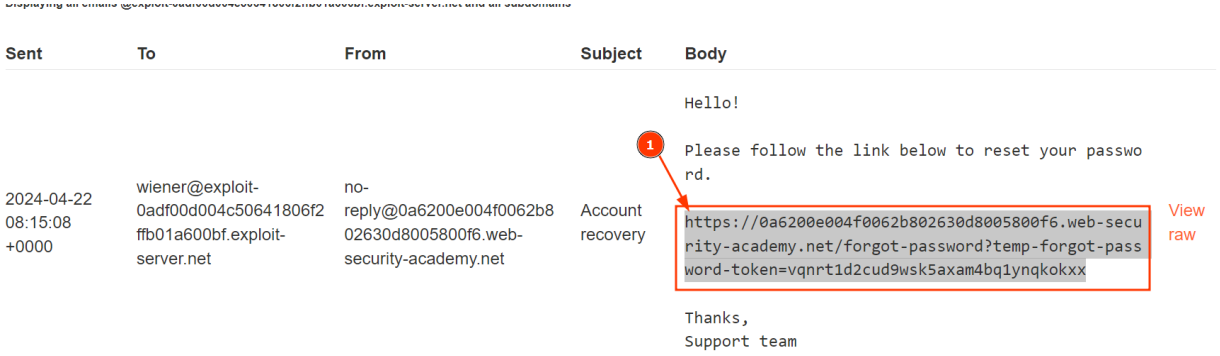

Figure 52: Copy Password Reset Link

Step 15: Pasting the link into a new browser tab and replacing the existing token with the earlier copied token.

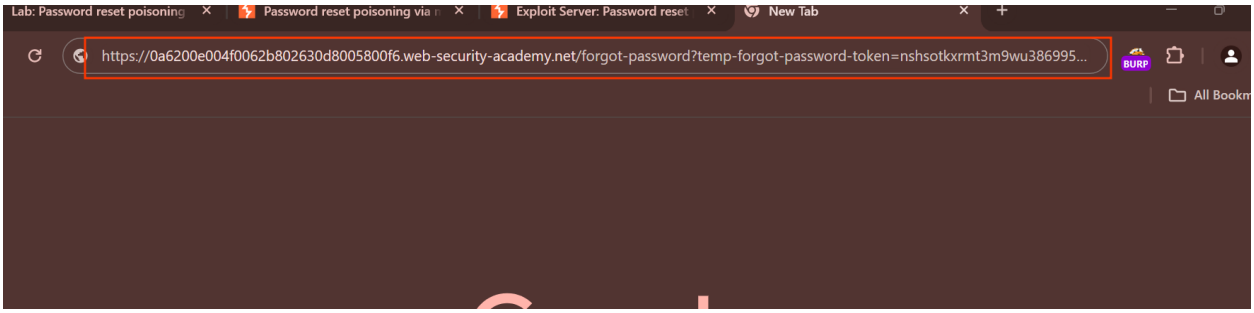

Figure 53: Modify and Use Password Reset Link.

Step 16: Entering a new password for the account and submitting it.

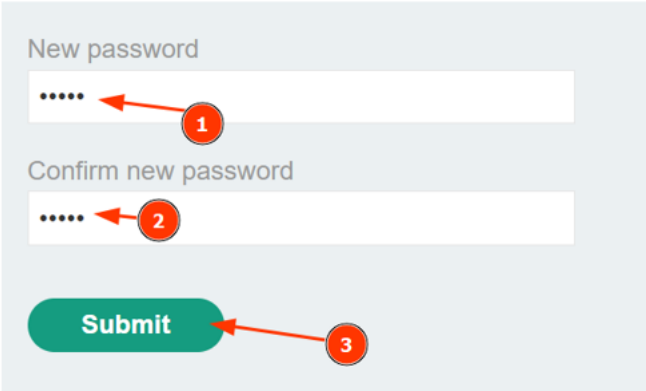

Figure 54: Enter New Password.

Step 17: Navigate back to the “My account” login page.

Figure 55: Navigate to Login Page.

Step 18: Enter username “carlos” and the new password, then log in.

Figure 56: Login with New Credentials.

Step 19: Verifying that the login was successful and logged in as “carlos”.

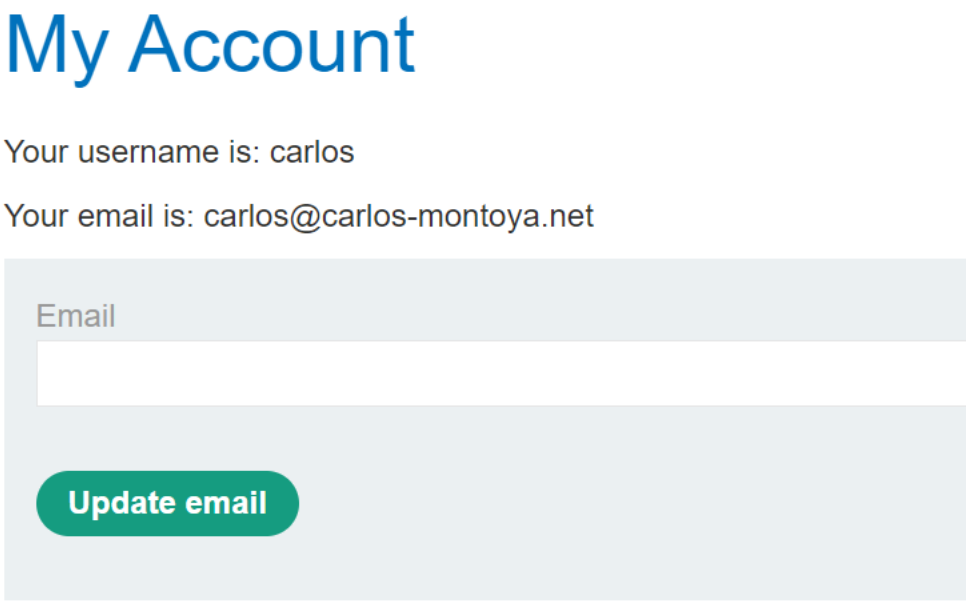

Figure 57: Successful Login as Carlos.

IV. Brute-forcing a stay-logged-in cookie.

In this attack, I will be demonstrating a potential security vulnerability in a web application's stay-logged-in feature This feature uses cookies to keep users logged into the site without needing to re-enter their credentials. The cookie used in this process contains sensitive information. Someone could be able to access user accounts without authorization if they manage to crack and modify this cookie.

Title	Brute-force forcing a stay-logged-in cookie.
Payload Used	Modified cookies containing usernames and MD5 hashed passwords, Base64 encoded credentials
CVSS	9.1
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
Criticality	Critical
OWASP Category	Broken Authentication
Tools Used	Burp Suite, Web Browser

Step 1: Navigating to the website and clicking on “My account”.

Figure 58: Access the Website.

Step 2: Entering “wiener” as the username and a password, check the “stay logged in” option and attempting to login.

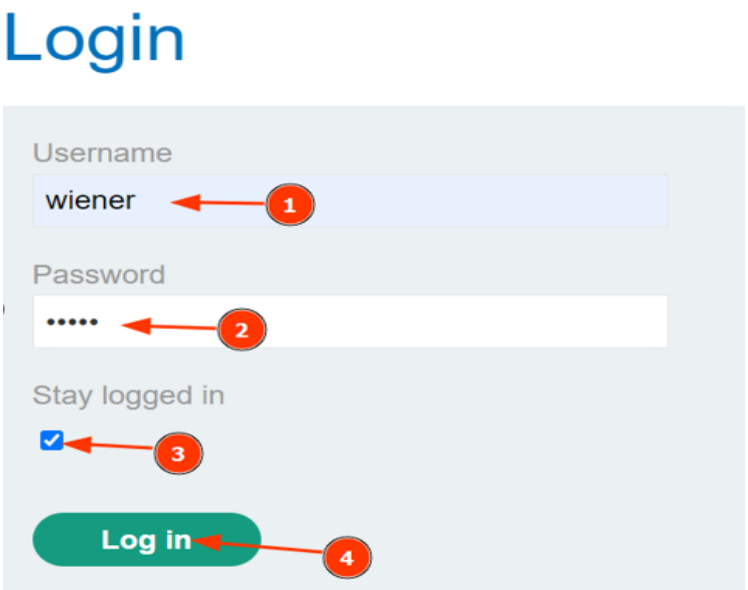

Figure 59: Attempt to login.

Step 3: In burp suite, turn on the intercept, forwarding the request to capture it.

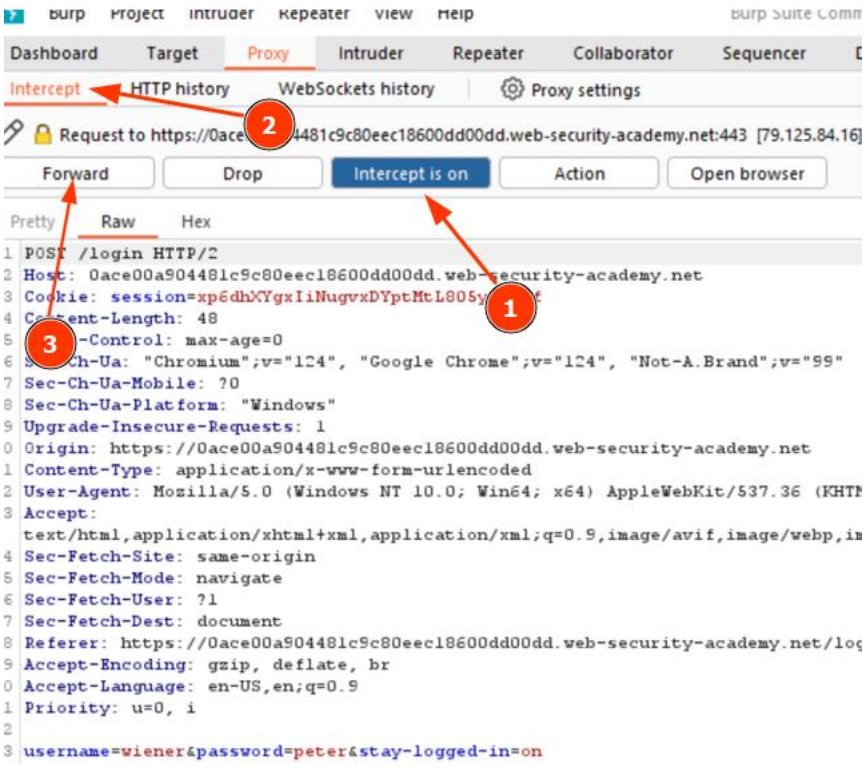

Figure 60: Intercept Login Request.

Step 4: Log out of the website.

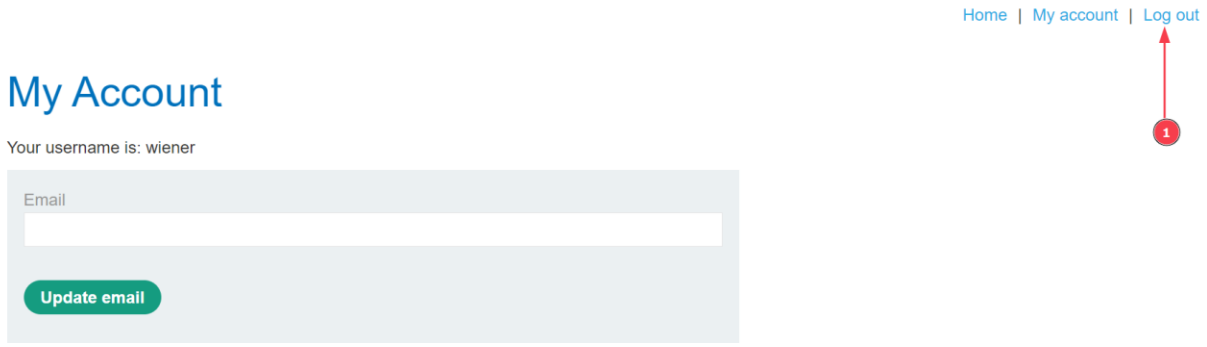

Figure 61: Log Out

Step 5: Selecting the stay-logged-in cookies in the request. Observing that it's base64 encoded and decoded pattern in the inspector.

base64(username+':'+md5HashOfPassword)

- Base64 is a method for encoding data using only readable characters (letters, digits, and a few symbols). It is commonly used to encode binary data into a string of ASCII characters.

Figure 62: Inspect Cookie Format.

Step 6: Sending the request to Burp Intruder, forwarding the request and turning off the intercept.

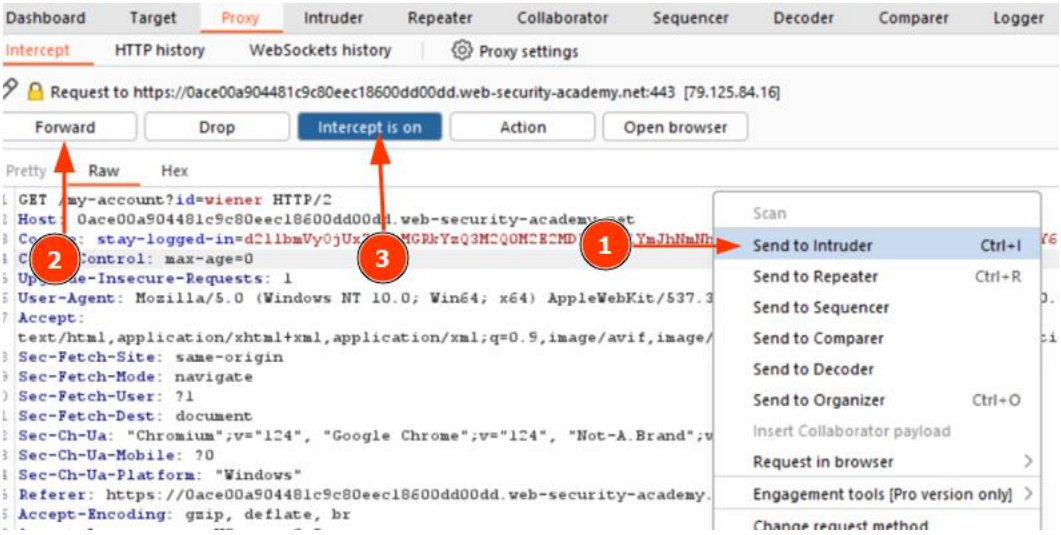

Figure 63: Send Request to Intruder.

Step 7: Changing the username in the cookie from "wiener" to "carlos". Highlighting the stay-logged-in cookie for modification in Intruder.

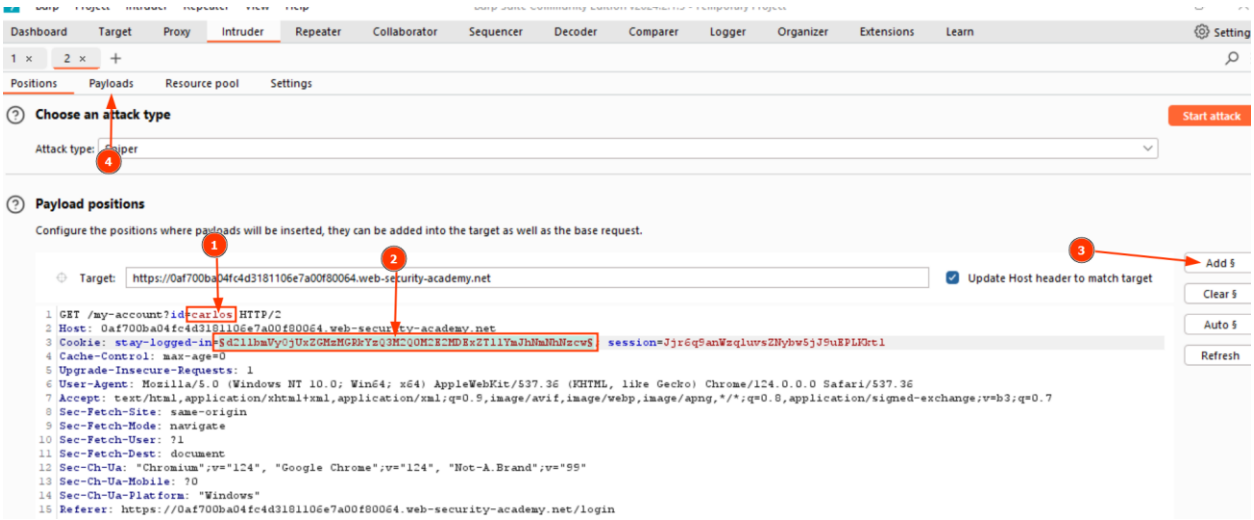

Figure 64: Set Up Intruder for Brute Force.

Step 8: Copying a list of potential passwords for brute-forcing.

```
123456
password
12345678
qwerty
123456789
12345
1234
111111
1234567
dragon
123123
baseball
abc123
football
monkey
```

Figure 65: Copy Password List.

Step 9: Pasting the password list into the Payloads section. Setting payload processing rules to hash each password with MD5.

- MD5 is a cryptographic hashing function that produces a 128-bit hash value from input data.

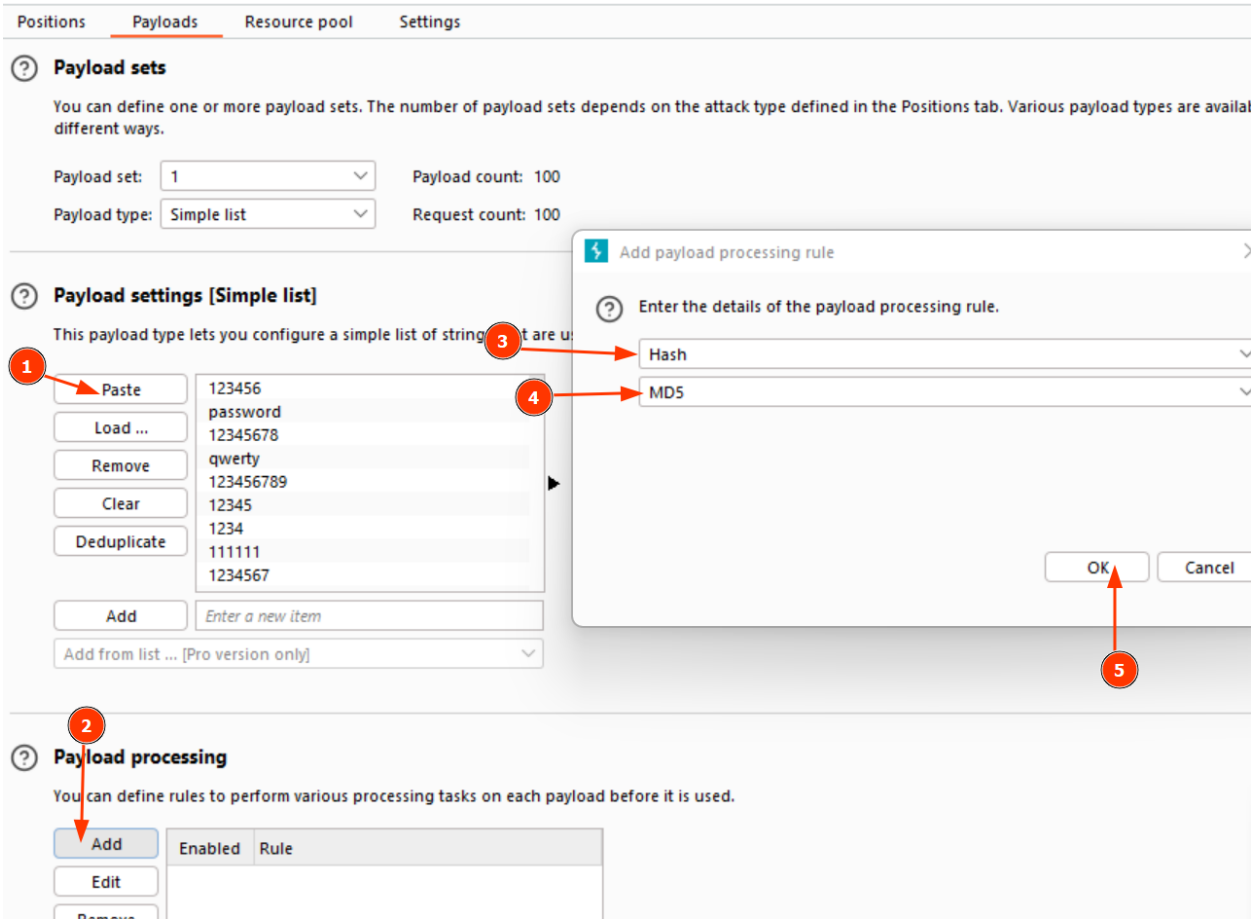

Figure 66: Paste Passwords into Intruder.

Step 10: Setting a payload processing rule to add the prefix "carlos:" to each hashed password.

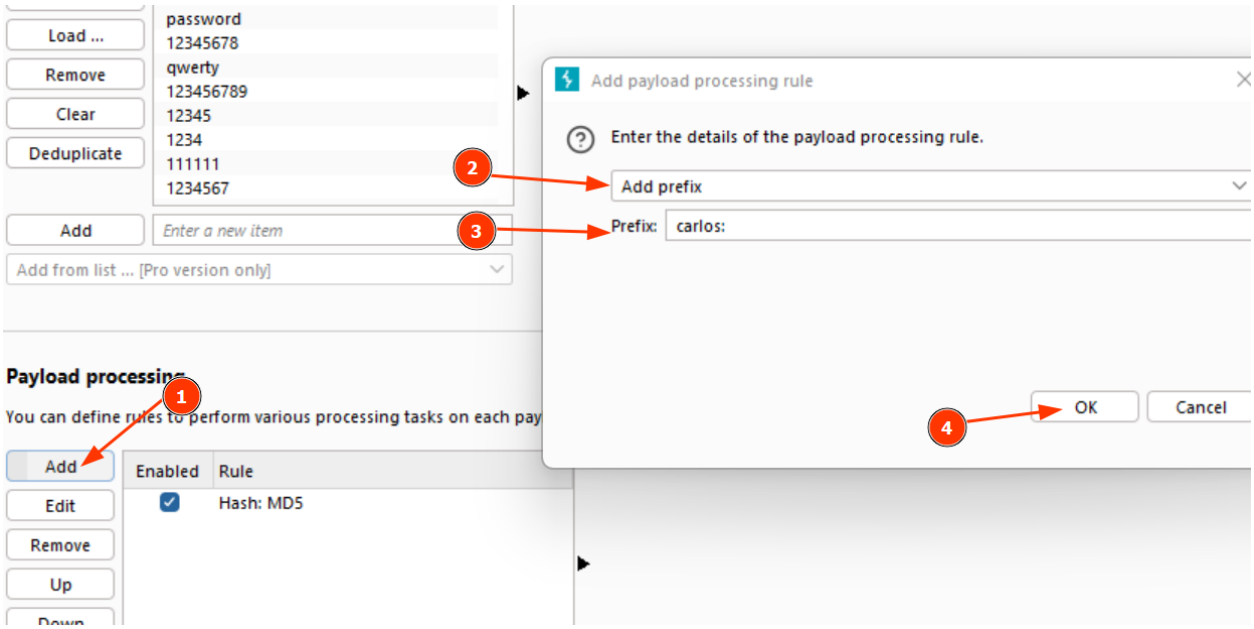

Figure 67: Configure Payload Prefix.

Step 11: Applying another payload processing rule to encode each payload in Base64.

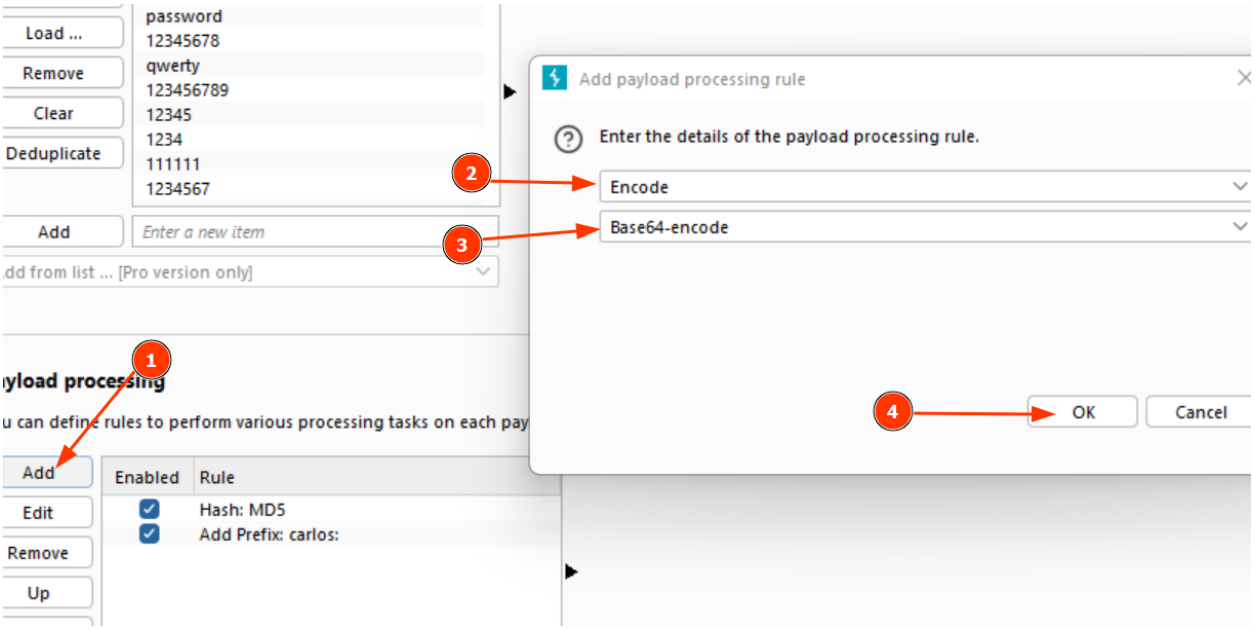

Figure 68: Encode Payloads.

Step 12: Initiating the attack and monitoring the response.

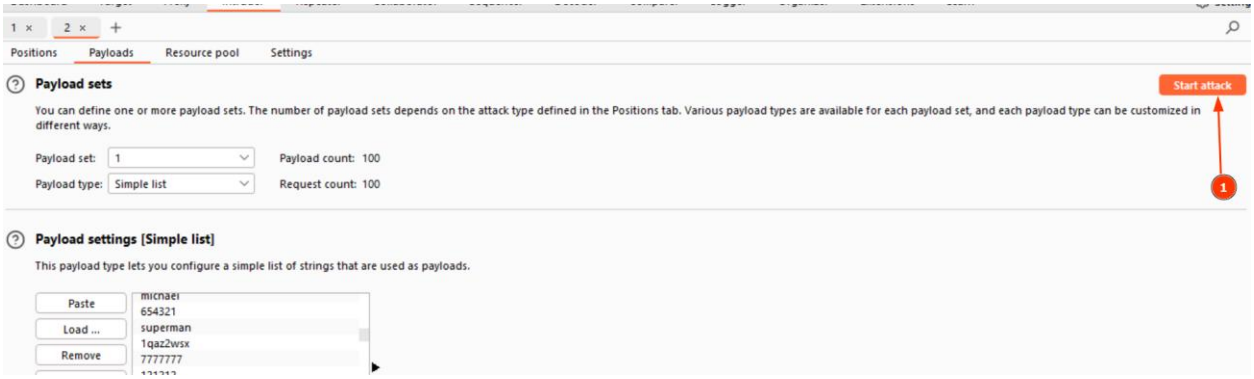

Figure 69: Start Attack.

Step 13: Identifying the successful payload by looking for the request where 66 has 200 as status code which shows the username is: carlos.

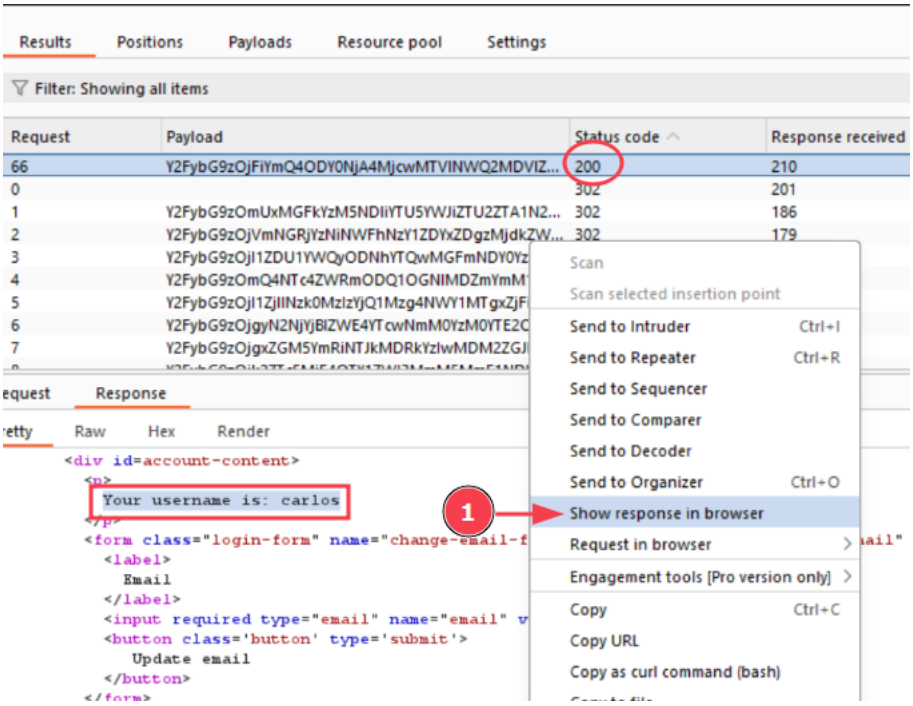

Figure 70: Analyse Response.

Step 14: Copying the responses from the successful request.

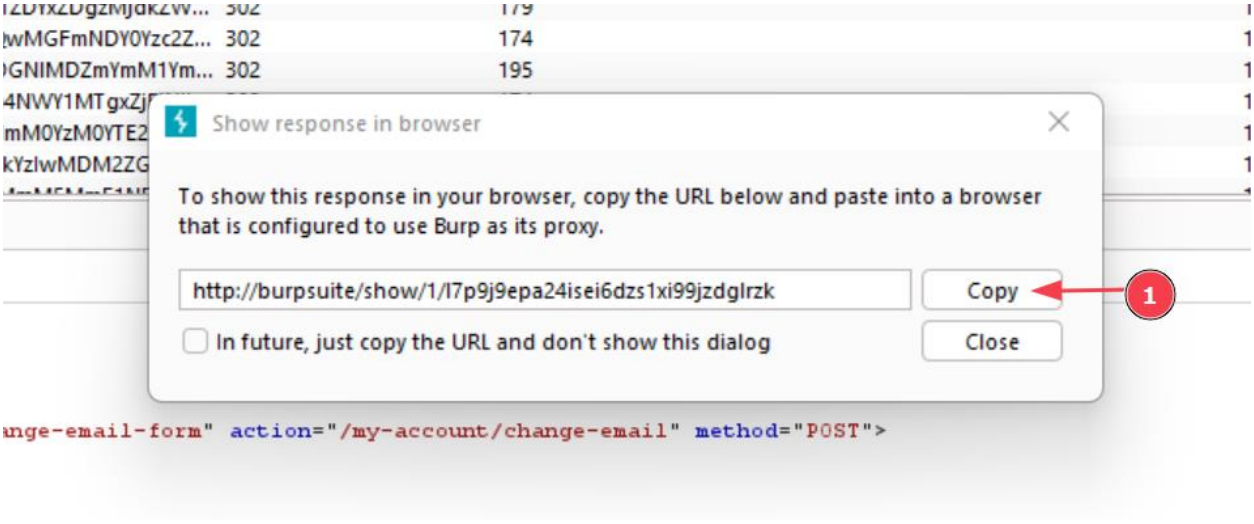

Figure 71: Copy the URL.

Step 15: Pasting the URL in a browser to log in as carlos.

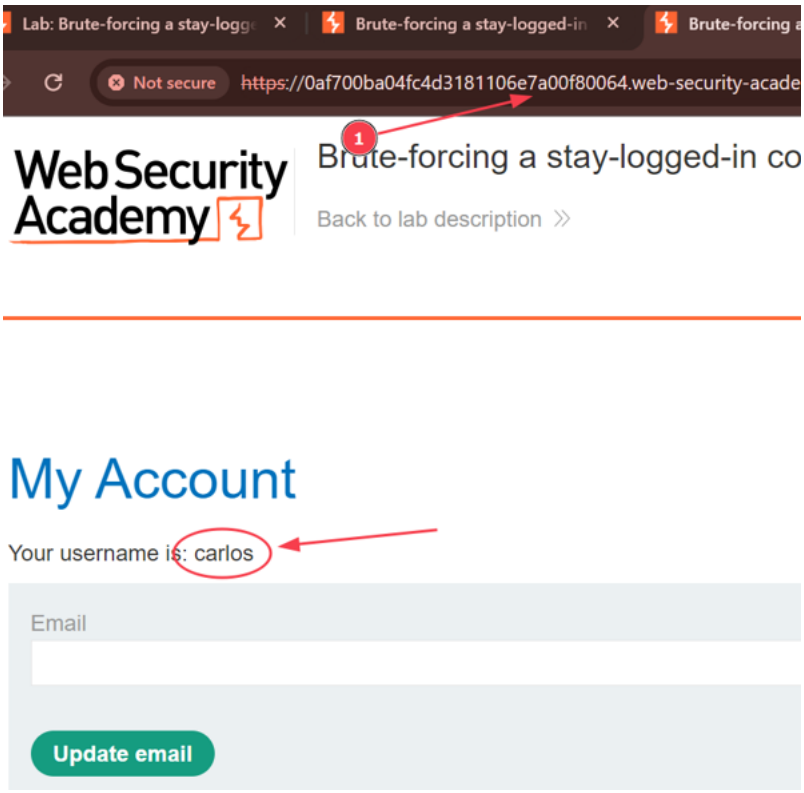

Figure 72: Log in as carlos.

4. Mitigation

- **Implement Brute-Force Protection:** The implementation of security measures like limiting the number of unsuccessful login attempts, delaying login attempts, and applying CAPTCHAs to distinguish human users from bots is necessary to protect against brute-force attacks, in which attackers try different password combinations to gain unauthorized access. Managing access through blacklisting suspicious IP addresses and whitelisting trusted ones can further enhance the system's security against such unwanted activities (Bright, 2024).
- **Multi-Factor Authentication (MFA):** MFA requires users to authenticate themselves using multiple proofs of identification from different categories, reducing the likelihood of unauthorized access. MFA includes factors that are own (keys, smartphones), know (passwords, PINs), or are inherent (biometrics, such as fingerprints, face recognition), with unknown or hidden factors (like browser fingerprinting or GPS location) that are silently verified. An attacker still needs the other factor to get access, even if they manage to steal the user's password. This strong security setup maintains a balance between high security and user convenience by not only making it difficult for attackers to compromise multiple security measures but also adjusting to the risk level of each login attempt, requiring additional verification in unusual or high-risk scenarios (BUTLER, 2022).
- **Set Session Expiration Timeouts**: Setting session expiration timeouts is essential for security to reduce the likelihood of hijacking, important applications may need to reauthenticate every two to five minutes. Depending on how the app is used, there are different recommended session lengths. For example, an office worker may need their session to stay active for approximately eight hours. Idle timeouts are one kind of timeout that reduces the time a hacker has to take advantage of a session ID by ending the session if no activity is found for a certain period of time. While renewal timeouts refresh the session ID during ongoing activity, invalidating the previous ID with a new request, absolute timeouts automatically terminate the session after reaching

its maximum duration. To ensure security, once these limits are reached, the session should be invalidated on both the client and server sides to prevent attackers from manipulating client-side parameters to extend the session (QAwerk, 2023).

- **Secure Credential Management:** Strong hashing algorithms, such as bcrypt or Argon2, are used in secure credential management to convert passwords into complex, scrambled formats that are very hard and time-consuming to decode, making brute-force attacks impractical. It is also important to avoid storing passwords in plain text or using simple encoding methods like base64, which can be easily reversed instead, passwords should be securely hashed and encoded to prevent easy access or decryption. Adding salt to hashing improves security even further. By adding a random piece of data, known as a 'salt,' to each password before it is hashed, it ensures that even if two users have the same password, their stored hashes will be unique (Gupta, 2024).

5. Evaluation

Advantages of Implementing Brute-Force Protection

- Time delays and limits on unsuccessful login attempts make it harder for hackers to guess passwords, reducing the risk of unwanted access.
- CAPTCHAs effectively block automated scripts, protecting against mass automated brute-force attacks.
- More security is added by controlling who has access to the system through the blacklisting of suspicious IP addresses and the whitelisting of trusted ones.
- Prevents sensitive user data from being compromised by reducing the number of methods of attack available to hackers.

Disadvantage of Implementing Brute-Force Protection

- Strict login restrictions may cause a bad user experience by frustrating legitimate users, especially those who have forgotten their passwords.
- When legitimate users share an IP address or use dynamic IPs, IP blacklisting may unintentionally block them, preventing access and causing irritation.
- Determined attackers might find ways around brute-force protections, such as using VPNs to change their IP addresses or developing CAPTCHA-solving bots.
- If new users find the login process overly difficult, strong security measures may discourage them from registering.

Application Area

i. Financial Services: Brute-force security is essential to safeguarding sensitive financial data and client accounts in sectors like banking and finance where security is of the highest priority.
ii. E-commerce: online shops to protect user accounts from unauthorized access while protecting payment and personal information.
iii. Government Services: To prevent data breaches and uphold public confidence, government websites that manage sensitive information such as civic records, personal data, and other data can use brute-force protection.

Advantage of Multi-Factor Authentication (MFA)

- MFA increases security by requiring multiple types of verification, making unauthorized access more difficult. Even if one factor like a password is compromised, the presence of additional factors such as a biometric or a physical device helps protect the account.
- MFA lowers the overall risk of fraud and identity theft by using multiple layers of defense, as an attacker would have to compromise more than one authentication factor to obtain unauthorized access.
- With MFA, security settings can be changed according to the risk involved in an action. More thorough methods of verification are sometimes needed for operations that carry higher levels of risk.
- MFA can increase trust by proving that strong security in place to protect users' financial and personal information as they become more security conscious.

Advantage of Multi-Factor Authentication (MFA)

- MFA can make logging in more difficult, which may frustrate users who prefer quick access. This could lead to negative user experiences and reduced satisfaction or productivity, especially if the users use the service regularly.
- Hardware such as tokens or biometric scanners, software development, and continuous maintenance need additional costs to set up MFA. For small businesses, these costs can be expensive.
- Users must maintain the security of their authentication factors such as not sharing security tokens or making sure their phones are safe for MFA to be effective. The benefits of MFA security can be affected by careless users.
- Physical devices, such as mobile phones or security tokens, might be misplaced, denying users access. Biometric scanners can also fail or be fooled, though advances in technology are reducing these risks.

Application Area

i. Education: MFA is used by educational institutions to protect teacher and student accounts, which hold student and personal data as well as academic records.
ii. Financial Services: MFA protects customer accounts and transactions, which often involve sensitive information, for banks, financial businesses, and insurance companies.
iii. Corporate VPNs: MFA protects remote workers' VPN access by ensuring that only authorized users have access to a company's private networks and critical information.

Advantage of Setting Session Expiration Timeouts

- Unauthorized session hijacking is much less likely when inactive timeouts and session expiration are configured. A compromised user's session ID will only be active for a short period of time, reducing the amount of time an attacker can take advantage of it.
- Idle timeouts and expiration serve as automatic checks to make sure no session is left running unchecked for an extended period of time. This feature requires re-authentication to continue the session, adds an extra layer of security.
- Timeouts reduce the period during which attacks can occur. If an attacker gains access to a session ID, they have only a limited time before the session expires and the ID becomes useless.

Disadvantages of Setting Session Expiration Timeouts:

- Users may become irritated by frequent re-authentication if timeouts are short. This can result in a bad user experience, when users are reading or conducting research tasks which require for long periods of time without user engagement.
- Sudden session expirations may result in the loss of unsaved work in apps where work progress is not automatically saved, upsetting users and reducing productivity.

- Finding the right duration for a session can be difficult. A setting that is too long could raise security problems whereas a setting that is too short could irritate users. It can be difficult to find the right balance between the application's user experience and security requirements.

Application Area

i. Banking and Financial Services: Due to the importance of transactions, online banking systems and financial applications often use strict timeout policies to ensure that sessions end on time following a period of inactivity or the completion of a transaction.
ii. Educational Platforms: To safeguard student information and course materials, online learning management systems employ session timeouts as these systems store test results, grades, and personal information.
iii. Government Services: Session timeouts are used by digital government services to allow users access to personal data and citizen services to guard against abuse and unlawful access to confidential government information.

Advantage of Secure Credential Management:

- Using strong hashing algorithms like bcrypt or Argon2 makes passwords very hard to decode, significantly increasing protection against brute-force attacks.
- Secure hashing and encoding practices ensure that passwords are never stored in plain text, maintain confidentiality of user data.
- Salting hashes adds an extra layer of security by ensuring that even identical passwords have unique hashes which prevents attackers from effectively using pre-computed hash attack techniques.

Disadvantage of Secure Credential Management:

- Strong hashing algorithms can slow down system performance due to their computational intensity.
- Setting up and maintaining a secure system requires technical expertise and can be filled with mistakes if not managed properly.
- The need for stronger passwords and slower login times due to complex hashing can sometimes irritate users while improving security.

Application Area

i. E-commerce: To protect customer accounts and transaction data, online retailers use secure credential management.
ii. Corporate Networks: To protect access to their internal networks, databases, and other digital resources, companies of all sizes use secure credential management.
iii. Telecommunications: In order to prevent data breaches and maintain regulatory compliance, telecom businesses must implement strong security procedures because they handle substantial amounts of user data.

6. Conclusion

This coursework provided a thorough understanding of methods for authentication and showed the importance they are to protecting data on digital networks. To ensure that only authorized users have access to sensitive systems and data, authentication serve as a first line of defense. The details of how we protect our digital identities are revealed by studying knowledge-based, possession-based, and inherent characteristic authentication.

A main focus of the coursework was the risks of broken authentication, a security vulnerability that arises from wrong use of authentication processes. Because to this vulnerability, attackers can take advantage of weaknesses in the authentication procedure, which could result in unauthorized access and data breaches. Attacks including brute-force, session hijacking, and username enumeration were demonstrated in real-world scenarios to provide helpful understanding into how vulnerabilities might result in major security breaches. To mitigate these risks, the coursework cover methods such as implementing Multi-Factor Authentication (MFA), setting session expiration timeouts, and securely managing credentials.

In conclusion, it is impossible to understate the importance of advanced methods of authentication as we continue introducing technology into every aspect of our lives, both personally and professionally. A preventive approach to security involving changing technologies, policy modifications, and education is necessary in the ongoing a conflict against cyber threats. We can better equip ourselves for the challenges of the future and ensure a more secure digital environment for all users by being aware of the methods of authentication and the possibilities for exploitation. The foundation for understanding these ideas has been set up by this coursework, which has also brought attention to the constant need for creativity and attention in the cybersecurity industry.

7. Bibliography

Authgear, 2023. *Broken Authentication: What Is It and How to Prevent It.* [Online]
Available at: https://www.authgear.com/post/broken-authentication-what-is-it-and-how-to-prevent-it
[Accessed 15 April 2024].

Cisco, 2024. *What Is a Phishing Attack? Definition and Types.* [Online]
Available at: https://www.cisco.com/c/en_in/products/security/email-security/what-is-phishing.html
[Accessed 16 April 2024].

HackerWhite, 2024. *Broken Authentication Vulnerability: Understanding & Mitigating the Risks in Web Application.* [Online]
Available at: https://hackerwhite.com/vulnerability101/web-application/broken-authentication-vulnerability
[Accessed 14 April 2024].

Beschokov, M., 2024. *What is Phishing Attack? Types and Examples.* [Online]
Available at: https://www.wallarm.com/what/types-of-phishing-attacks-and-business-impact
[Accessed 16 April 2024].

Bright, 2024. *Broken Authentication: Impact, Examples, and How to Fix It.* [Online]
Available at: https://brightsec.com/blog/broken-authentication-impact-examples-and-how-to-fix-it/
[Accessed 14 April 2024].

BUTLER, S., 2022. *What Is Multi-Factor Authentication (MFA)?.* [Online]
Available at: https://www.howtogeek.com/794722/what-is-multi-factor-authentication-mfa/#:~:text=The%20reasoning%20behind%20MFA%20is%20that%20there%27s%20an,need%2C%20especially%20if%20they%27re%20very%20different%20in%20nature.
[Accessed 29 April 2024].

Cloudflare, 2024. *What is authentication?.* [Online]
Available at: https://www.cloudflare.com/learning/access-management/what-is-authentication/
[Accessed 14 April 2024].

DEMIR, B., 2021. *Anatomy of the Session Management Tests | Cobalt.* [Online]
Available at: https://www.cobalt.io/blog/anatomy-of-the-session-management-tests
[Accessed 15 April 2024].

Fruhlinger, J., 2020. *Equifax data breach FAQ: What happened, who was affected, what was the impact? | CSO Online.* [Online]
Available at: https://www.csoonline.com/article/567833/equifax-data-breach-faq-what-happened-who-was-affected-what-was-the-impact.html
[Accessed 17 April 2024].

Gupta, D., 2024. *Understanding Broken Authentication: Risks & Prevention.* [Online]
Available at: https://www.loginradius.com/blog/identity/what-is-broken-authentication/
[Accessed 29 April 2024].

Microsoft Security, 2024. *What Is Authentication? Definition and Methods.* [Online]
Available at: What Is Authentication? Definition and Methods | Microsoft Security
[Accessed 14 April 2024].

N, B., 2021. *OWASP TOP 10 2021 Released - Cyber Security News.* [Online]
Available at: https://cybersecuritynews.com/owasp-top-10-2021/
[Accessed 23 April 2024].

Positive Technologies, 2020. *Web Applications vulnerabilities and threats: attacks statistics for 2019.* [Online]
Available at: https://www.ptsecurity.com/ww-en/analytics/web-vulnerabilities-2020/#:~:text=Broken%20Authentication%20was%20found%20in,and%20access%20the%20web%20application
[Accessed 17 April 2024].

Poza, D., 2020. *What Is Broken Authentication?.* [Online]
Available at: https://auth0.com/blog/what-is-broken-authentication/
[Accessed 14 April 2024].

Poza, D., 2021. *What Is Password Spraying? How to Stop Password Spraying Attacks.* [Online]
Available at: https://auth0.com/blog/what-is-password-spraying-how-to-stop-password-spraying-attacks/
[Accessed 16 April 2024].

Poza, D., 2023. *What Is Credential Stuffing? How To Prevent Credential Stuffing Attacks.* [Online]
Available at: https://auth0.com/blog/what-is-credential-stuffing/
[Accessed 16 April 2024].

QAwerk, 2023. *What Is Broken Authentication? Examples and How to Prevent It.* [Online]
Available at: https://qawerk.com/blog/what-is-broken-authentication/
[Accessed 15 April 2024].

Ranjan, R., 2024. *Password Spraying Attack | OWASP Foundation.* [Online]
Available at: https://owasp.org/www-community/attacks/Password_Spraying_Attack
[Accessed 16 April 2024].

Sorbello, I., 2021. *2FA, MFA and Step Up Authentication - Identity Hub - | Transmit Security.* [Online]
Available at: https://transmitsecurity.com/blog/what-is-2fa-vs-mfa-sca-and-step-ups
[Accessed 14 April 2024].

VARAKSINA, S., 2023. *How to Secure a Website from Hackers: Vulnerabilities + List of Tips - Mind Studios.* [Online]
Available at: https://themindstudios.com/blog/how-to-secure-a-website-from-hackers/
[Accessed 15 April 2024].