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: Kipa Shrestha

London Met ID: 22066682

College ID: np01nt4a220120

Assignment Due Date: Tuesday, 7 May 2024

Assignment Submission Date: Monday, 6 May 2024

Word Count (Where Required): 4791

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 discusses the widespread issue of broken authentication vulnerabilities in web applications and explains the serious threats they cause to user privacy and data security. It highlights the vital requirement for strong authentication processes by looking at common authentication issues like weak session management, insufficient password policies, and poor multi-factor authentication implementations. The report demonstrates how attackers use these vulnerabilities to increase privileges within web applications, gain unauthorised access, and steal user accounts through practical demonstrations and real-world situations. In addition, it analyses several strategies and tactics for mitigating broken authentication vulnerabilities, assessing their benefits, drawbacks, and compatibility for various contexts. This coursework contributes to the ongoing efforts to protect the integrity of web applications and strengthen information systems against growing cyber threats by offering practical implementation insights and practical recommendations.

Table of Contents

Table of Figures

1. Introduction

Authentication is a process used by companies to confirm that only authorised users, services, and apps with the appropriate permissions have access to organisational resources. It's an important part of cybersecurity because gaining unauthorised access to systems is an attacker's main aim. They achieve this by taking the passwords and usernames of users who can log in. Authentication is important because it helps organizations protect their systems, data, networks, websites, and applications from attacks. It helps people protect the privacy of their personal information, allowing them to do less risky business online, such as banking or investing. When authentication processes are weak, it's easier for an attacker to get access to accounts, either by figuring out users' passwords or by tricking them into giving up their credentials (Microsoft Security, 2024).

Figure 1: Authentication (Tiwari, 2024).

1.1. Broken authentication vulnerabilities in web applications

No matter what online platforms or applications you use, you are never fully protected against cyberattacks. Broken Authentication is a threat to application security as it can provide an attacker access to session tokens, keys, and passwords. In the worst case, this could result in complete authority over the system and additional exploitation of user identities. Broken authentication attacks aim to take authority over one or more accounts giving the attacker the same privileges as the target. Authentication is “broken” when attackers can expose passwords, keys or session tokens, user account information, and other details to obtain user identities (Secure Flag, 2024).

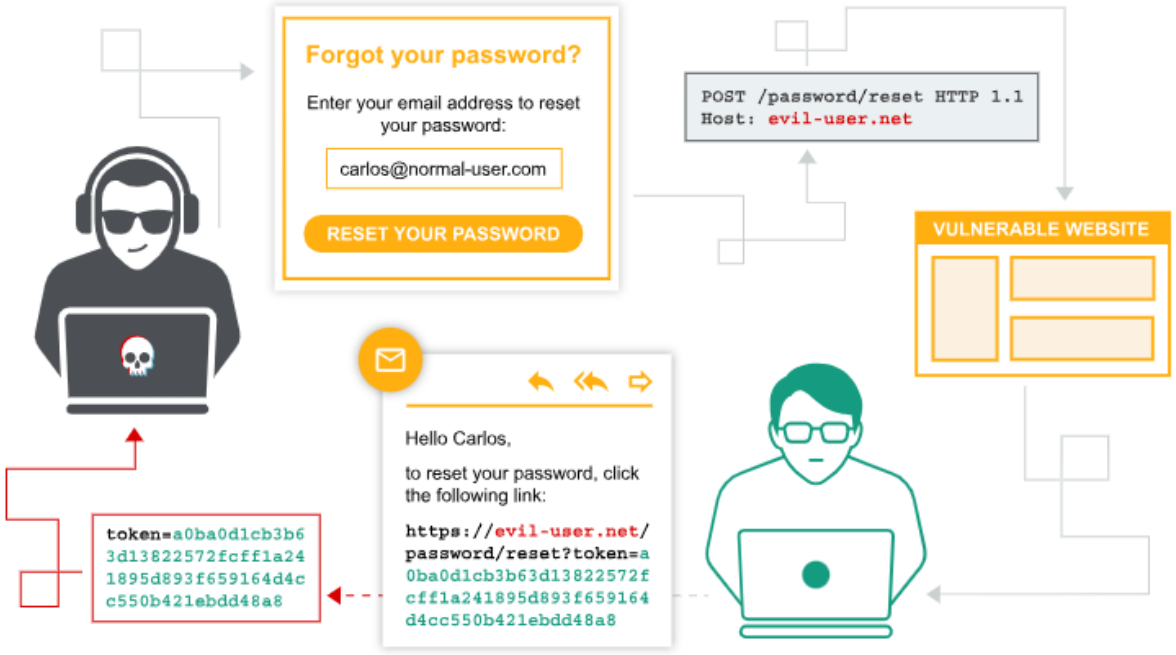


Figure 2: Authentication Vulnerabilities (Port Swigger, 2024).

Session management refers to the techniques used to manage the state of a user's session as they interact with an application. Broken authentication and session management processes involve the use of various tokens, such as cookies or session IDs, to keep track of a user's identity and access rights. When these tokens are not created, saved, or validated properly, vulnerabilities in session management and broken authentication arise. For example, a brute-force attack can be used to guess user credentials if an application uses weak passwords. Similarly, if session IDs are not randomly generated or are not invalidated after a certain period, attackers can hijack sessions and imitate legitimate users (LinkedIn, 2023).

There are several examples of broken authentication vulnerabilities that highlight the potential risks. Weak passwords that can be guessed easily, such as "12345" or "password123" are a popular example that attackers can take advantage of. Another example is when there is a lack of proper session expiration, which allows an attacker to reuse a user's session and obtain unauthorised access even after the user logs out. Attackers can keep trying different usernames and password combinations until they find one that works if a program does not take precautions against brute-force attacks. Broken authentication vulnerabilities can also include weak protections against session hijacking, account lockouts, and session fixation.

1.2. Technical terminologies

- **Authentication**
  Authentication is a process used by companies to confirm that only authorised users, services, and apps with the appropriate permissions have access to organisational resources. It's an important part of cybersecurity because gaining unauthorised access to systems is an attacker's main aim (Microsoft Security, 2024).

- **Authorization**
  Authorization is a process by which a server determines if the client has permission to use a resource or access a file. It is generally used with authentication so that the server has some idea of who the client is seeking access. Different forms of authentication may be needed for authorization in some cases passwords may be necessary, but not in others (Boston University, 2024).

Figure 3: Authentication and Authorization (Bhargav, 2024).

- **Session Management**
  Broken authentication and session management are often mentioned together to point out that authentication is more complex than just usernames and passwords. Sessions and credentials are used by web applications to identify users. Attackers can guess the identity of authorised users and obtain unauthorised access by taking advantage of weaknesses in session management or authentication (Poza, 2020).

- **Brute Force Attack**
  Brute force attacks are a type of hacking technique where passwords, login credentials, and encryption keys are cracked through a process of trial and error. It is a simple but effective strategy to gain unauthorised access to user accounts, company networks, and systems. The hacker tries multiple usernames and passwords, often using a computer to test a wide range of combinations until they find the correct login information (Fortinet, 2024).

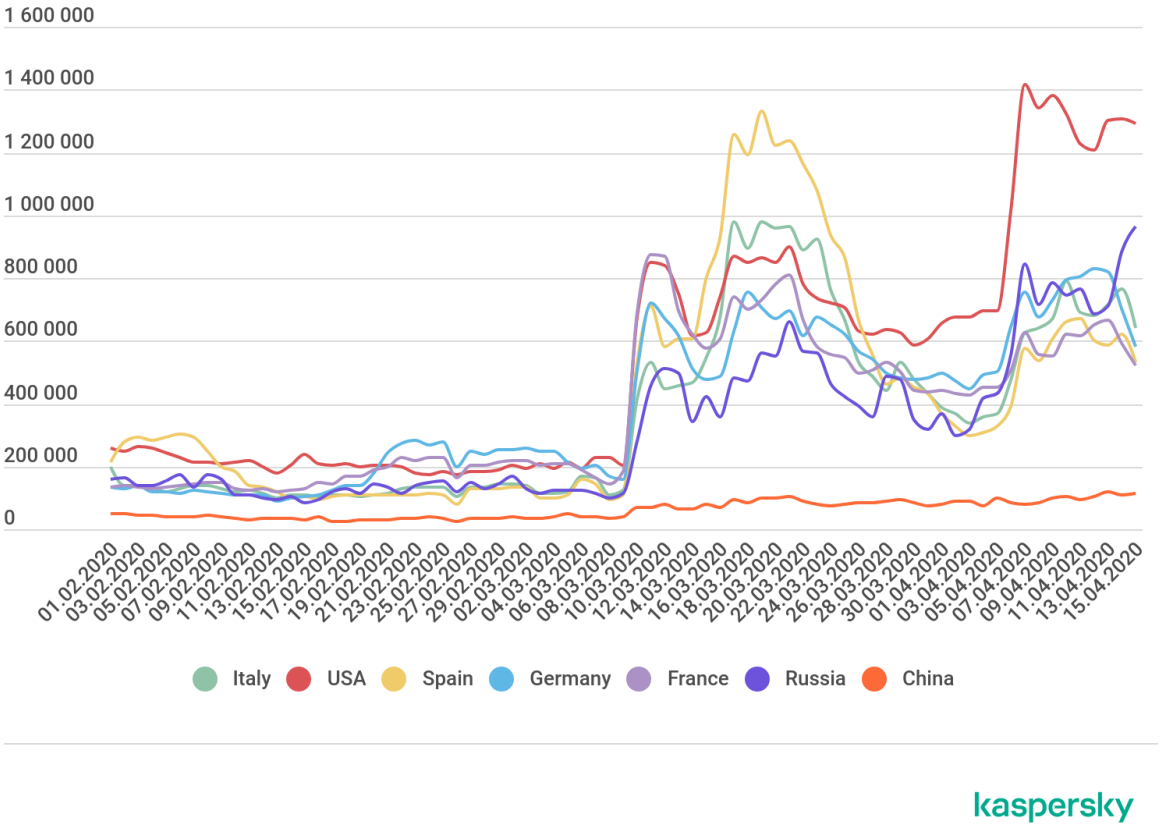


Figure 4: Growth in the number of attacks by the Bruteforce (Securelist, 2020).

1.3. Aim and Objectives

1.3.1. Aim

The main aim of this coursework is to understand and address Broken Authentication vulnerabilities in web applications to improve their security.

1.3.2. Objectives

- To identify what Broken Authentication vulnerabilities are and why they're a problem for web applications.
- To explore how these vulnerabilities occur and their potential impact.
- To demonstrate how attackers exploit Broken Authentication vulnerabilities.
- To provide practical solutions and strategies to mitigate these vulnerabilities and enhance web application security.

2. Background

The impact of broken authentication attacks can be devastating for both an organization and its customers. When attackers exploit these vulnerabilities, they gain unauthorized access to user accounts, personal data, sensitive business information, and more. This not only leads to a breach of privacy and potential financial losses but can also severely tarnish the reputation of the impacted organization. An end-user may have unauthorised access to their account because of a broken authentication attack, which could result in the loss of sensitive data such as social security numbers and credit card details. This might also lead to other types of personal harm, such as identity theft and unauthorized transactions. For businesses, the consequences can be even more severe. If the attack is successful, hackers might be able to get access to privileged accounts and use those accounts to manipulate data, perform malicious actions, or even take over the entire system. This may result in significant financial losses, a negative impact on the organization's reputation, a loss in customer trust, and even legal consequences (Maric, 2024).

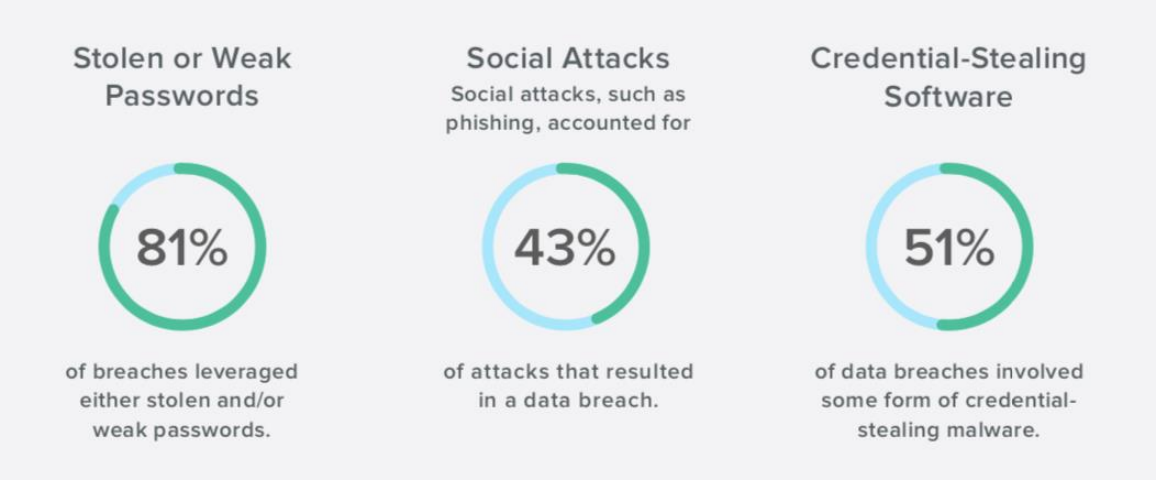


Figure 5: Attacks that exploit broken authentication (Okta, 2024).

The identity attacks that are most likely to impact an organization are:

Attack #1: Broad-based phishing campaigns:

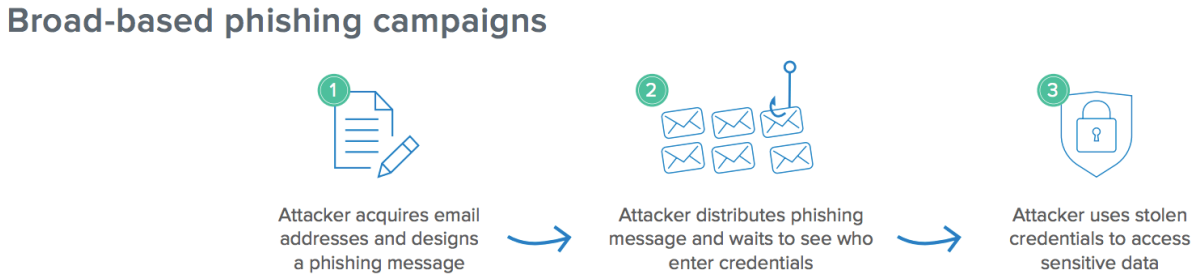


Figure 6: Broad-based phishing campaigns (Okta, 2024).

A broad-based phishing campaign recognizes that threat agents must gain access to only a few accounts or one admin account to compromise the organization. An attacker uses phishing to send a fake message, such as a fake Google login page, to many email addresses or phone numbers. They hope someone falls for it and enters their login credential. Even in well-trained organisations about one in twenty people, may fall for the trap. Once they have those credentials, the attacker can use them to access data or even perform more targeted attacks, like pretending to be that person to get into more sensitive areas. According to a Verizon investigation, stolen or weak passwords are a basic entry point for hackers which means this is normally just the beginning of a larger cyberattack (Okta, 2024).

Attack #2: Spear phishing campaigns

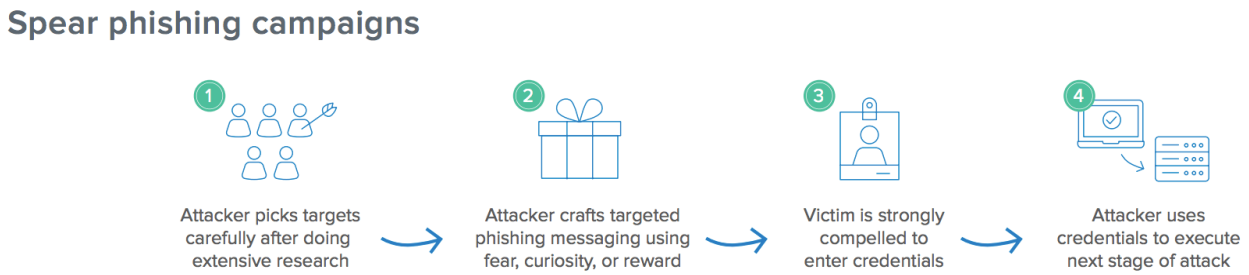


Figure 7: Spear phishing campaigns (Okta, 2024).

Spear phishing is a targeted form of phishing that often involves more research in designing the target list and phishing message. Unlike normal phishing, which takes a shotgun approach, spear phishing is more like sniper fire. Instead of randomly sending fake messages to many people, the attacker intentionally selects a small group and completes their research. To make their message look trustworthy, they search social media and other internet sources, frequently pretending as someone the target knows. To trick the victim into clicking a link or providing login information, they might bring up a recent business event. Once they obtain those credentials, hackers can proceed with their attack or access confidential information. To achieve their goals, it all comes down to manipulation and precision (Okta, 2024).

Attack #3: Credential stuffing

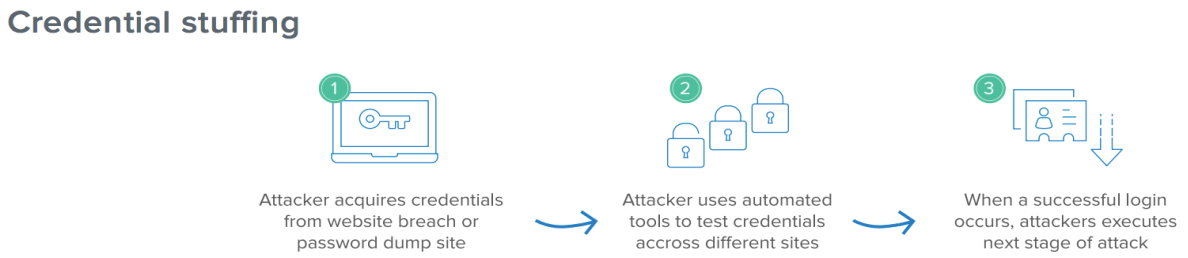


Figure 8: Credential stuffing (Okta, 2024).

Credential stuffing is a form of brute force attack that takes advantage of our struggle to select unique passwords across our various accounts. It is like an attacker's shortcut to accessing the user's accounts. They are aware that most users reuse their passwords for multiple websites, and with the large number of online accounts we manage, it's hard not to. Attackers attack quickly when a data breach occurs, and your login credentials are exposed. In the hopes that you have used the same password elsewhere, they take those credentials that they have stolen and enter them into other websites. Approximately 73% of passwords are duplicates, thus they're frequently correct. This method can be automated by bots, which can quickly test hundreds of combinations. According to Akamai, more than 40% of login attempts worldwide are malicious, thanks to these automated credential stuffing attacks. For the attackers, it's a numbers game: they cast a wide trap and pull in anything they can (Okta, 2024).

Attack #4: Password spraying

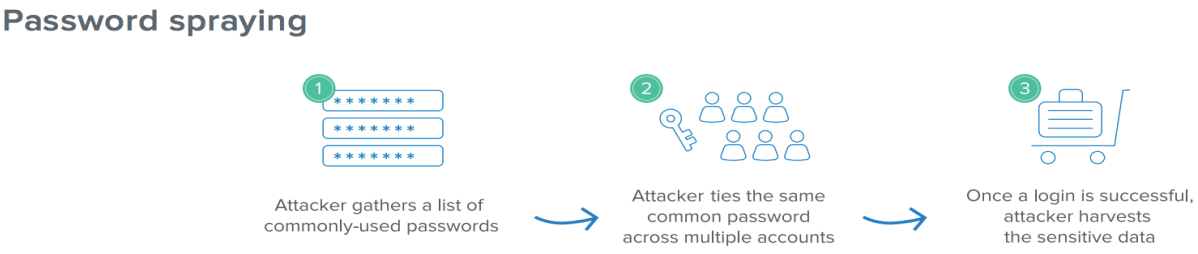


Figure 9: Password spraying (Okta, 2024).

Password spraying is like trying to use one key to unlock several doors. Attackers know that many people use simple passwords like "password123" or "12345," which have been hacked millions of times. They use a short list of these popular passwords for many accounts, instead of trying many passwords for a single account. This lowers the chance of getting caught because they're not trying to log into one account with tons of failed login attempts. Once they find a match, they can proceed to the next breach or take advantage of the sensitive data they are seeking. It's all about exploiting our bad password habits to their advantage (Okta, 2024).

Attack #5: Man-in-the-Middle (MitM) attacks

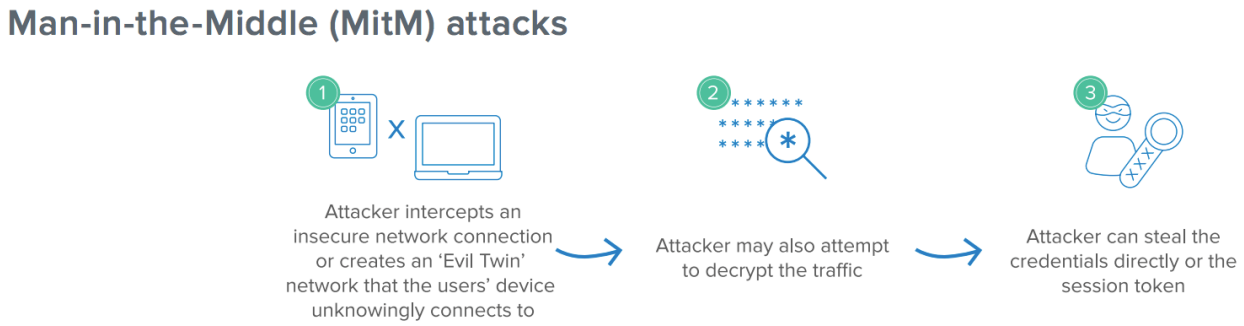


Figure 10: MitM attacks (Okta, 2024).

A MitM attack on an organization is a highly targeted attack that can result in a full take of credentials and data-in-transit if executed correctly. After intercepting a network connection, an attacker can also take advantage of session hijacking that compromises the web session by stealing the session token. In a network interception attack, the attacker sets up a fake Wi-Fi hotspot, often imitating a legitimate one like Starbucks Wi-Fi. The attacker can see all the data going through when people connect to it. They may attempt to fool users into installing an unauthorised certificate to decode the data if it is encrypted. Before users even authenticate, the attacker could steal their credentials by monitoring their inputs. Also, they can gain access to the account and perform more attacks if they manage to intercept a session token after authentication. It is like eavesdropping in on a private discussion, except the attacker is listening in on your digital interactions (Okta, 2024).

In recent years, broken authentication attacks have accounted for many of the worst data breaches, and security experts sound the alarm about this underrecognized threat. Broken authentication was #2 on the 2017 OWASP Top 10 list including various issues such as weak passwords, session fixation and inadequate session expiration policies. In 2021 the Broken Authentication category was renamed Identification and Authentication Failures. This category includes vulnerabilities like weak or default passwords, insecure password recovery mechanisms, and inadequate session expiration, among others (Contrast Security, 2024).

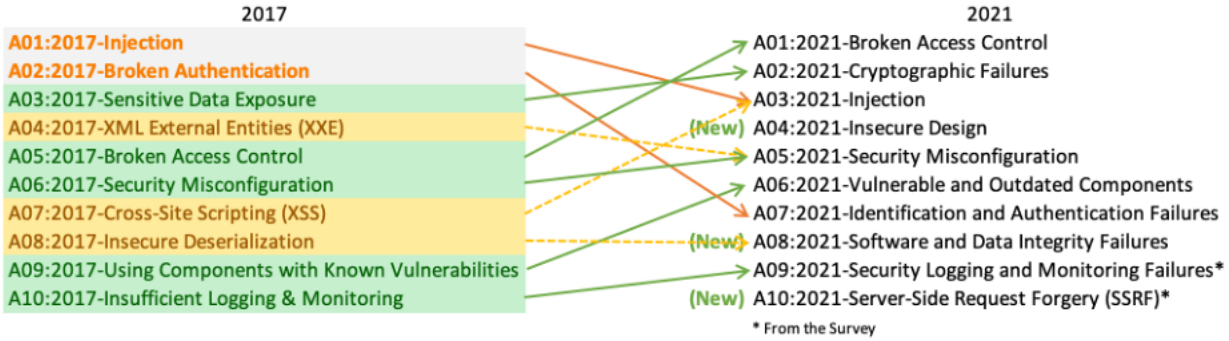


Figure 11: Broken Authentication ranked in 2017 and 2021.

Marriott Data Breach

The highlighted issue in the Marriott data breaches is broken authentication. It means that Marriott's process for determining if users were authorised to access its computer systems was not secure. By obtaining the usernames and passwords of two Marriott employees, the hackers were able to gain access to Marriott's network. About 5.2 million visitors' names, birthdates, phone numbers, and credit account numbers were stolen in this hack. According to Marriott, sensitive information including credit card numbers and passwords was secure. Even though Marriott told affected guests about the breach and offered a way to track their personal information, it shows that Marriott needs to make their security stronger to stop this from happening again (Gupta, 2024).

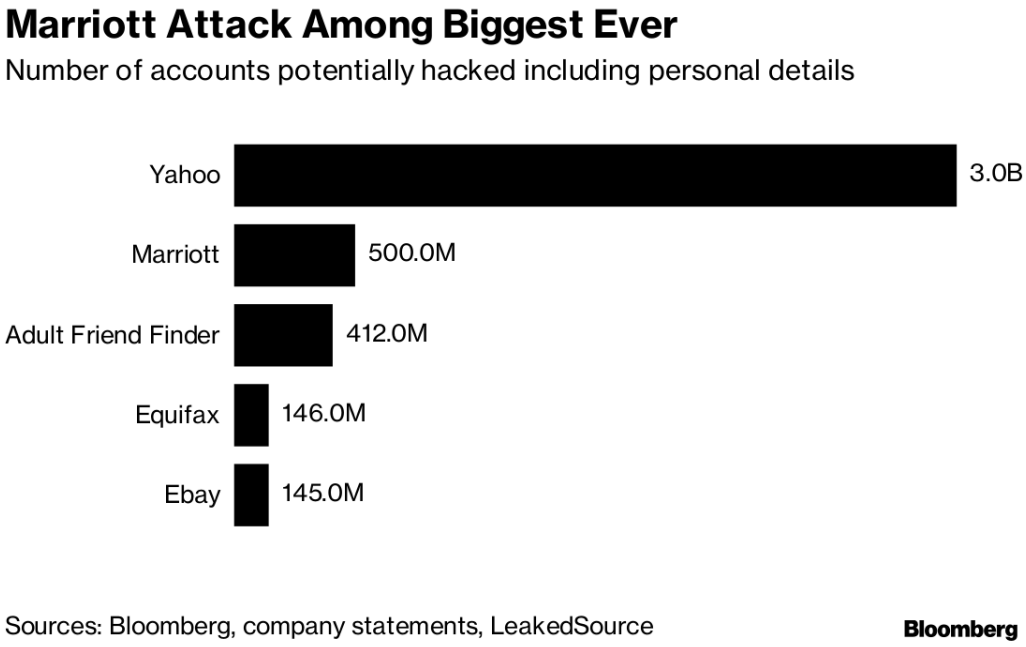


Figure 12: Number of hacked accounts during Marriott breach (Bhaktavatsalam, 2018).

South Carolina Department of Revenue

The importance of having strong authentication processes is highlighted by the South Carolina Department of Revenue data breach, in which a foreign hacker gathered 3.6 million Social Security numbers and 387,000 credit card details. This breach linked with a similar incident at the IRS in 2015, reveals a systemic failure in security measures. Default passwords in the authentication layer and unencrypted sensitive data fields (especially Social Security numbers) expose serious vulnerabilities in the organization's safety measures. The $16 billion that was stolen from 15.4 million American consumers in 2016 alone is proof that these mistakes not only put people at risk of identity fraud but also caused them to suffer significant financial losses. Strong security measures must be prioritised by organisations immediately to safeguard sensitive data and reduce the danger of data breaches. The breach serves as an eye-opening example of the tragic consequences of insufficient identification and authentication processes (Chua, 2022).

3. Demonstration

Username enumeration via different responses

Demonstrating "Username Enumeration via Different Responses" in a lab setting like PortSwigger's educates about the risk of unpredictable authentication error handling. It reveals how attackers exploit response variations to identify valid usernames, highlighting the need for constant error messages. This practical experience improves understanding of common attack techniques and encourages developers to implement secure coding practices, strengthening web application security in the process.

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:N/I:L/A:N
Criticality	Medium
OWASP Category	Broken Authentication
Tools used	Burpsuite

Step 01. Launch the Burp Suite, login to the web application with an invalid username and password.

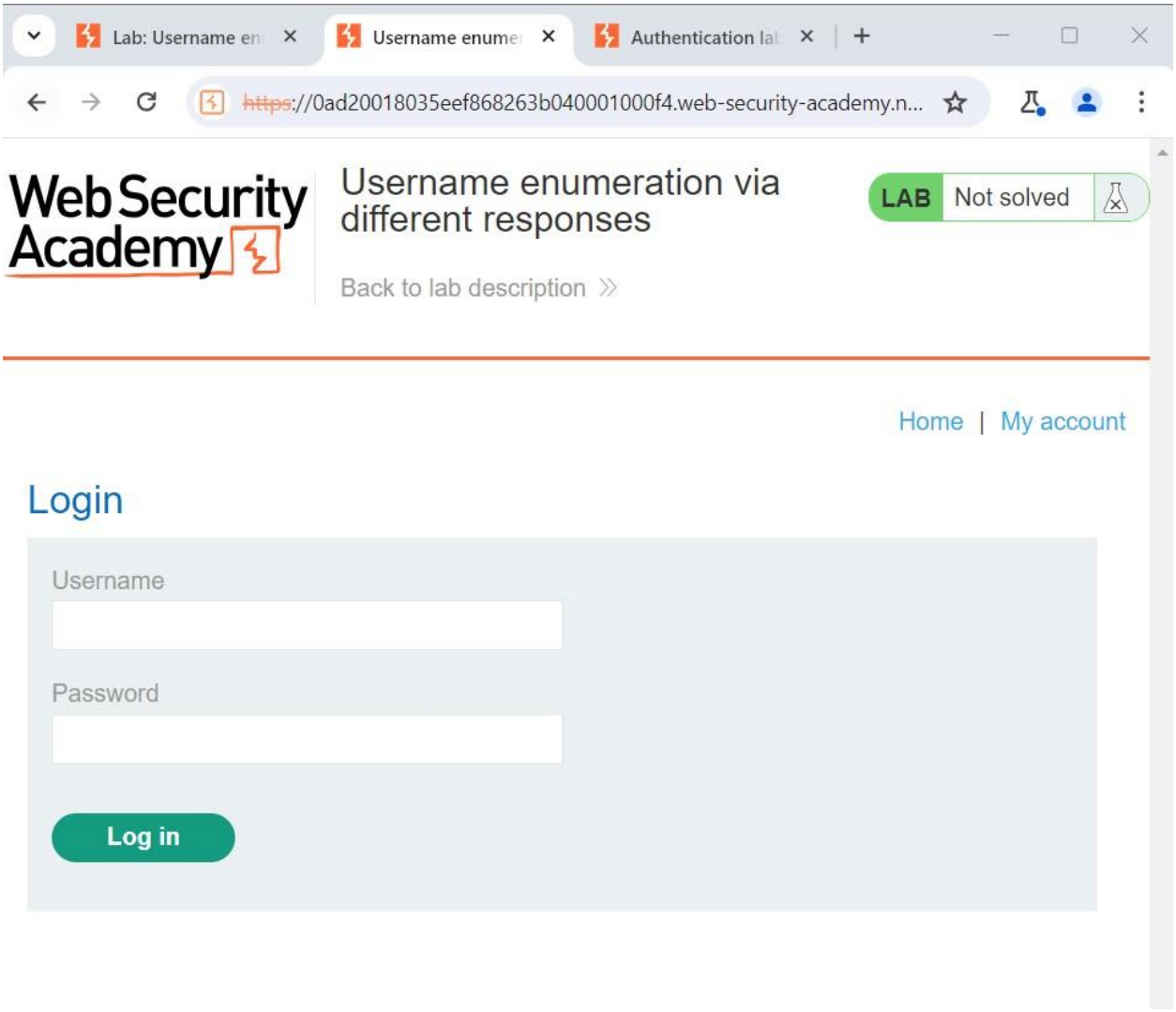


Figure 13: Login attempt.

Step 02. After submitting an invalid username and password, “Invalid username” is popped up.

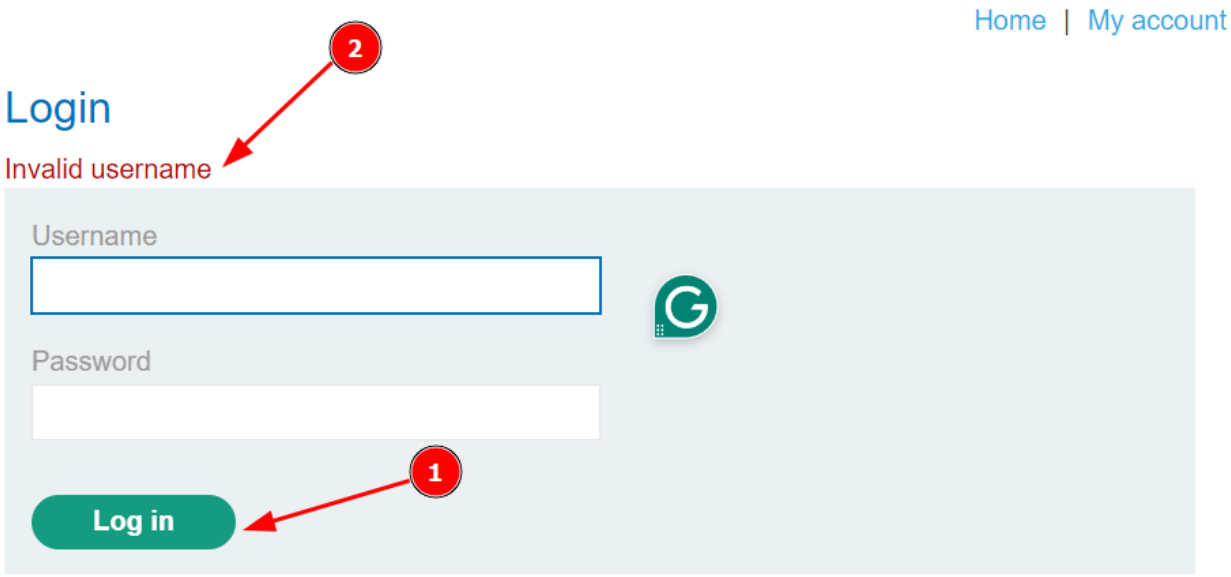


Figure 14: Invalid username and password submission.

Step 03. In Burp Suite's Proxy tab, go to the HTTP history and find the **POST/login** request.

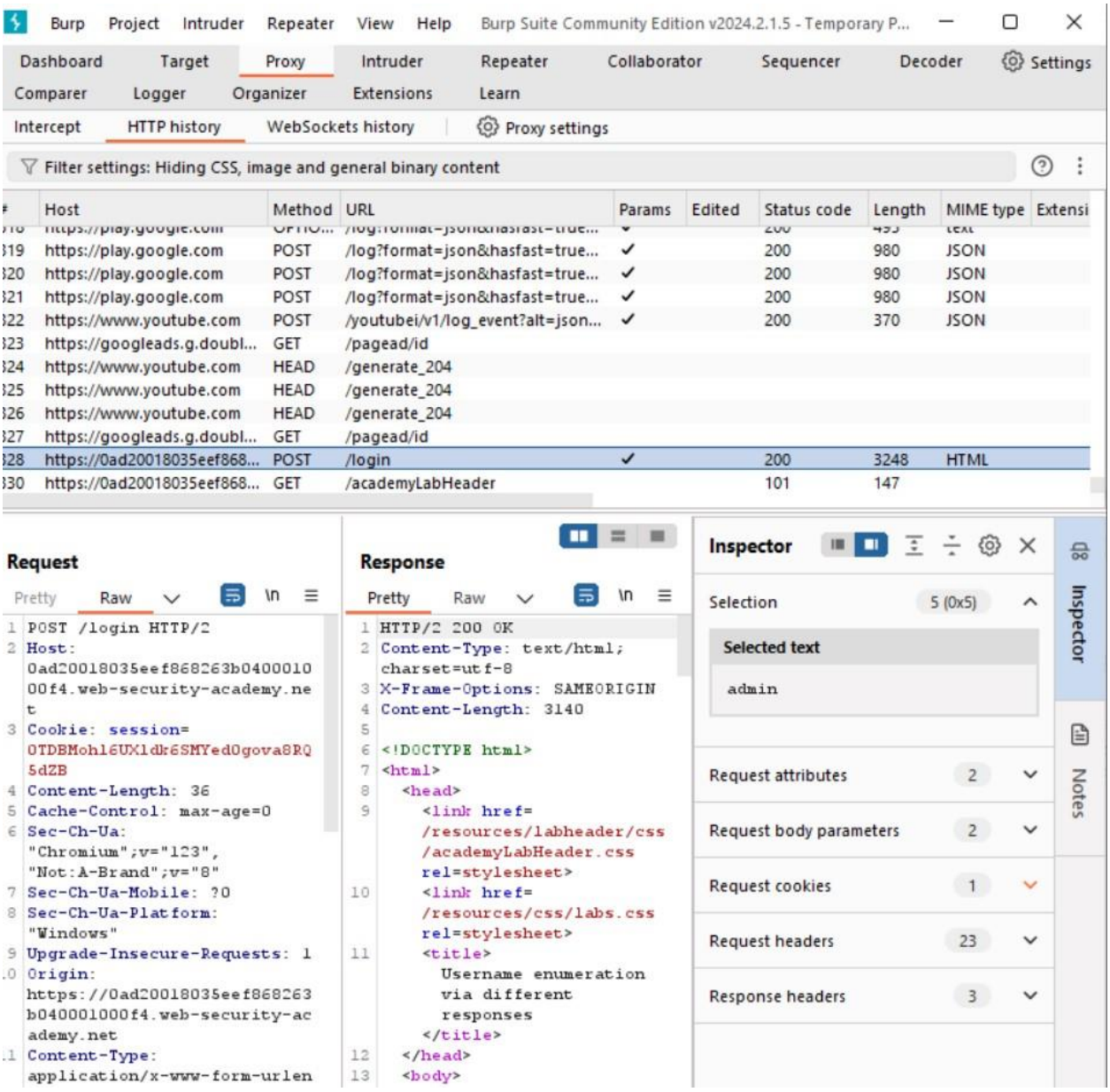


Figure 15: Find POST/login.

Step 04. Now send it to Burp Intruder.

Burp Intruder enables you to configure attacks that send the same request over and over again, inserting different payloads into predefined positions each time (Portswigger, 2024).

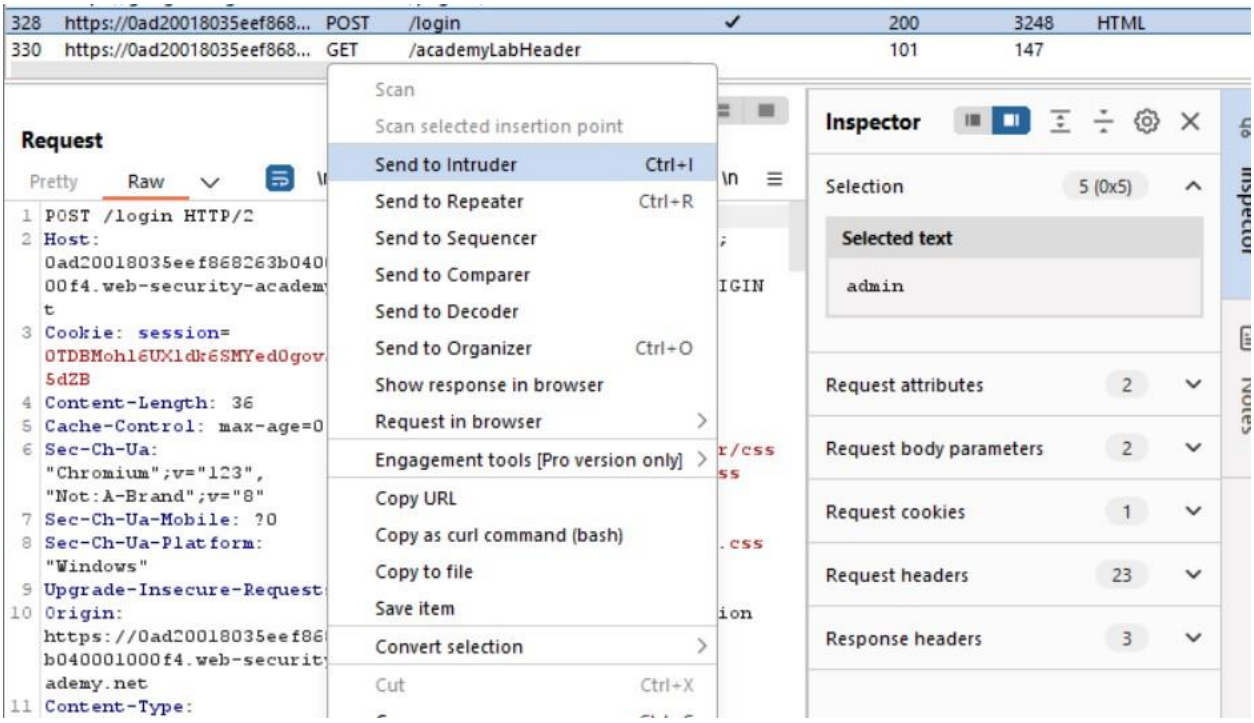


Figure 16: Send to Intruder.

Step 05. In Burp Intruder, go to the positions tab. Notice that the username parameter is automatically set as a payload position. Select the Sniper attack type.

Sniper attack places each payload into each payload position in turn. It uses a single payload set. The total number of requests generated in the attack is the product of the number of positions and the number of payloads in the payload set. The Sniper attack is useful for fuzzing a number of request parameters individually for common vulnerabilities (Portswigger, 2024).

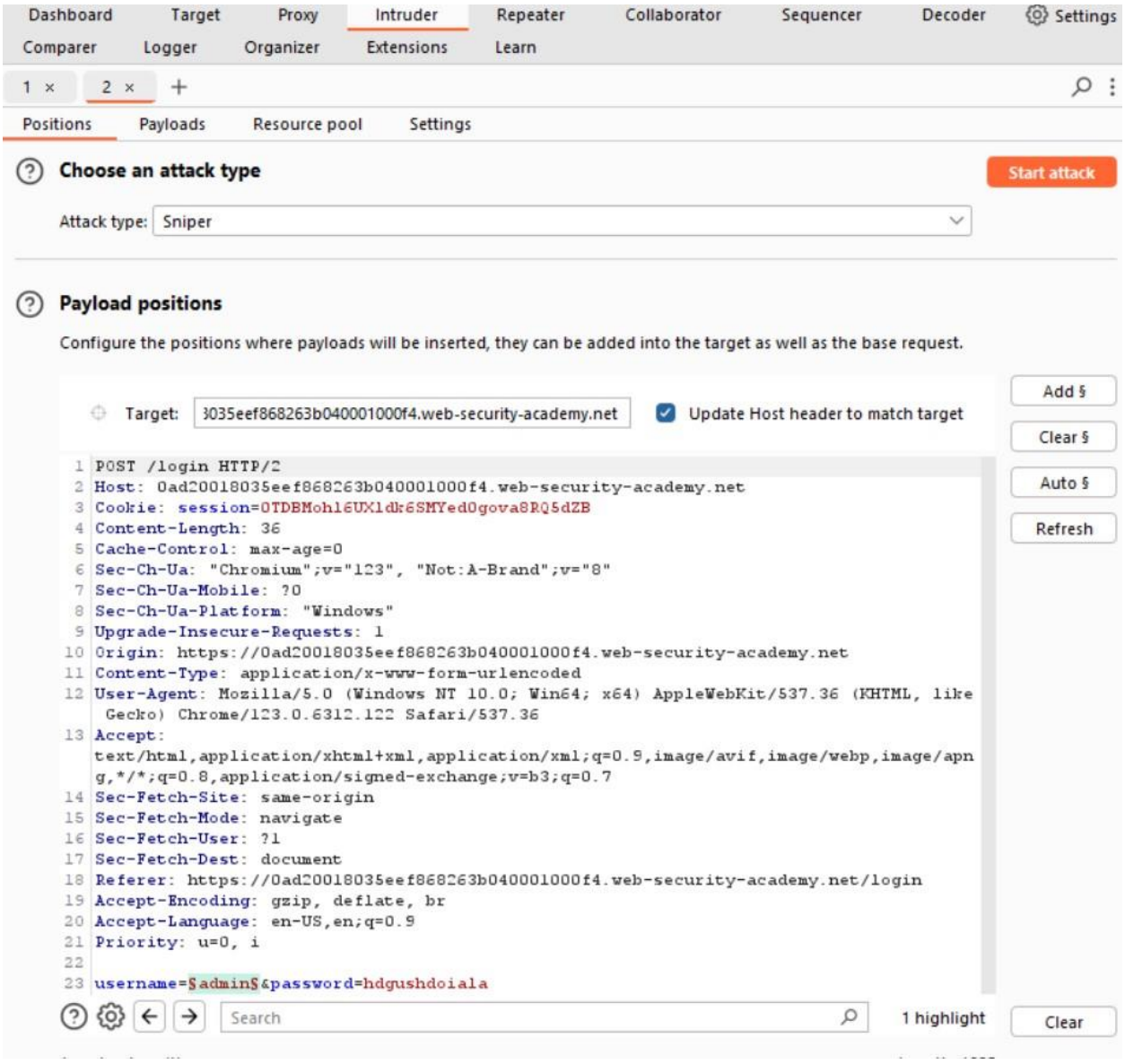


Figure 17: Set Payload Position.

Step 06. On the payloads tab, select Simple list payload type. Paste the list of candidate usernames and click start attack.

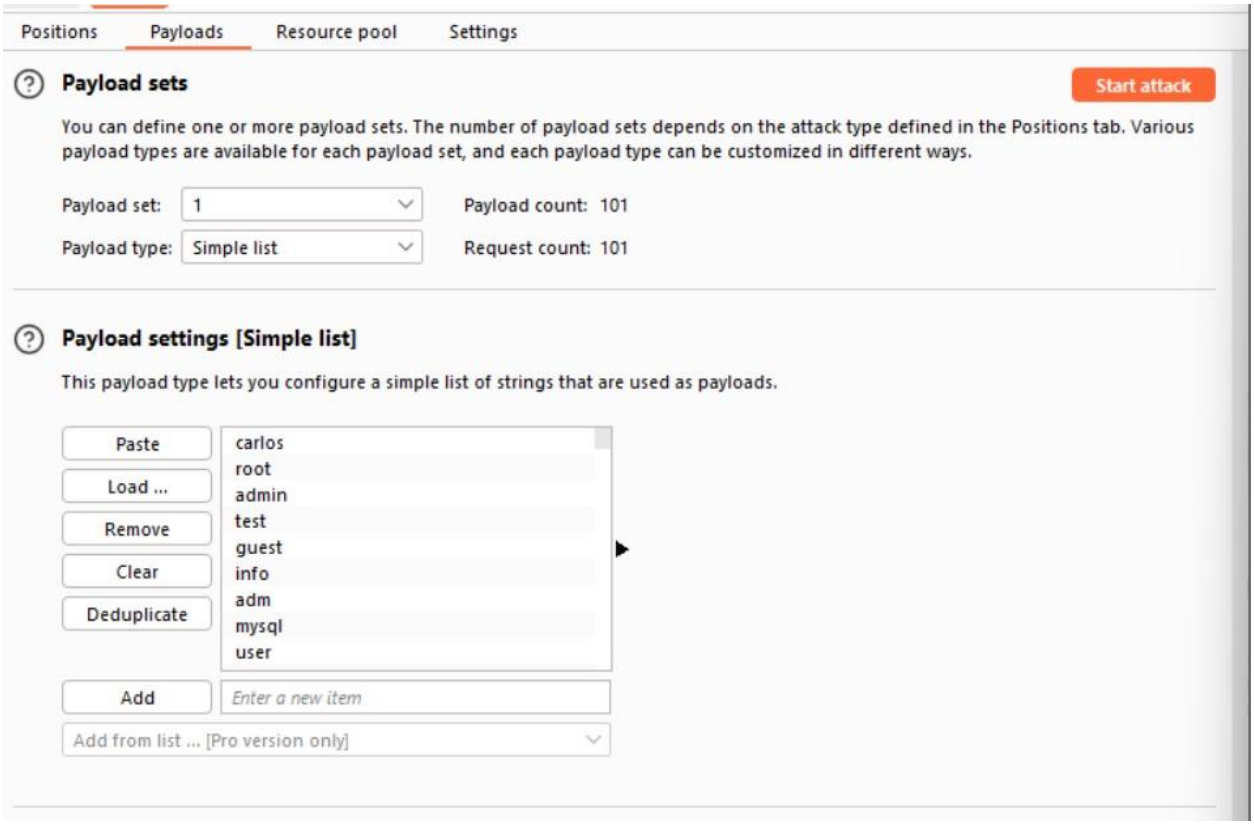
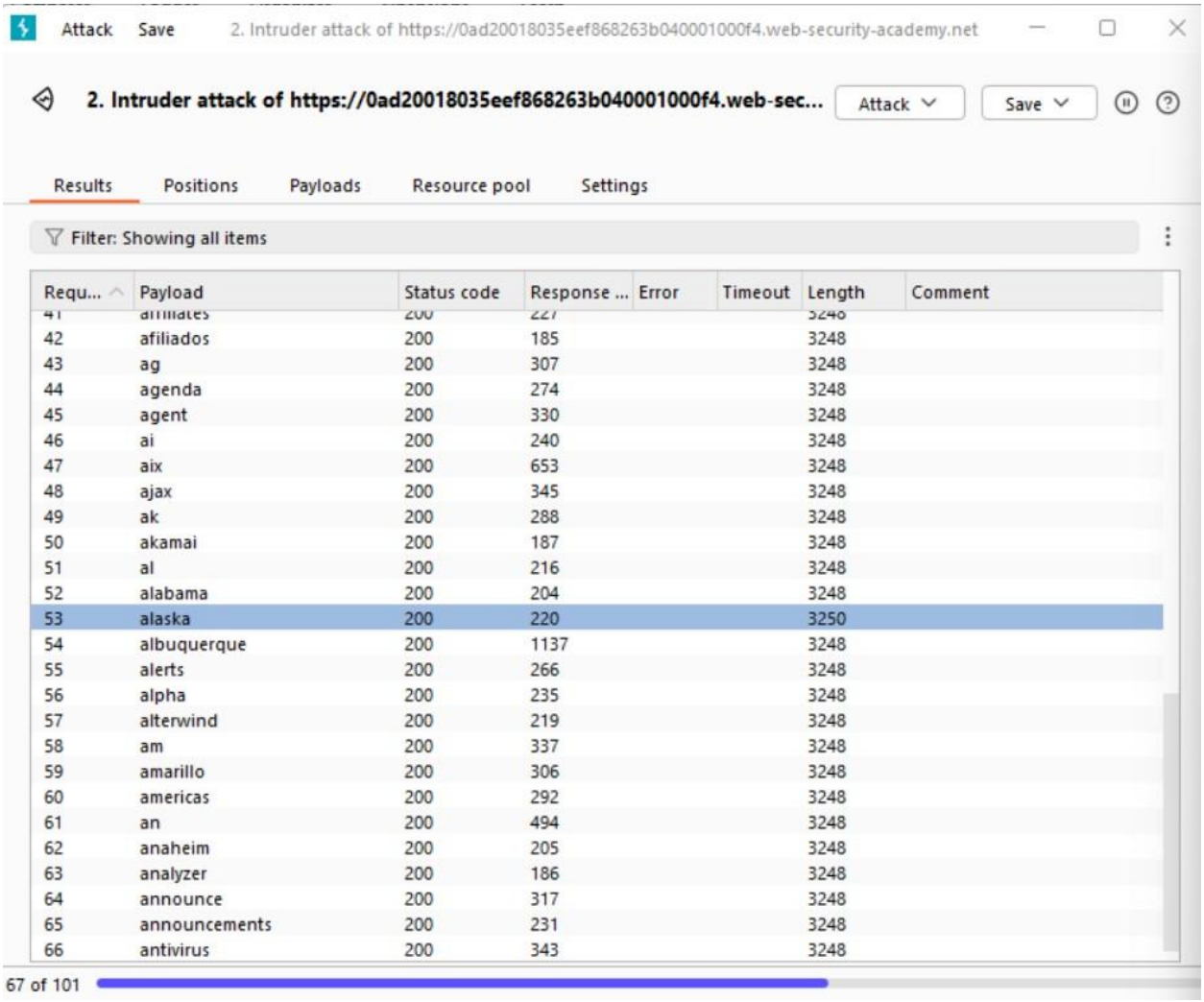


Figure 18: Paste candidate list and start attack.

Step 07. Notice that the length column is different from the others.



Requ...	Payload	Status code	Response ...	Error	Timeout	Length	Comment
42	afiliados	200	185			3248	
43	ag	200	307			3248	
44	agenda	200	274			3248	
45	agent	200	330			3248	
46	ai	200	240			3248	
47	aix	200	653			3248	
48	ajax	200	345			3248	
49	ak	200	288			3248	
50	akamai	200	187			3248	
51	al	200	216			3248	
52	alabama	200	204			3248	
53	alaska	200	220			3250	
54	albuquerque	200	1137			3248	
55	alerts	200	266			3248	
56	alpha	200	235			3248	
57	alterwind	200	219			3248	
58	am	200	337			3248	
59	amarillo	200	306			3248	
60	americas	200	292			3248	
61	an	200	494			3248	
62	anaheim	200	205			3248	
63	analyzer	200	186			3248	
64	announce	200	317			3248	
65	announcements	200	231			3248	
66	antivirus	200	343			3248	

Figure 19: Find different length column than other.

Step 08. The response contains the message “Incorrect password”. Compare the response with other responses.

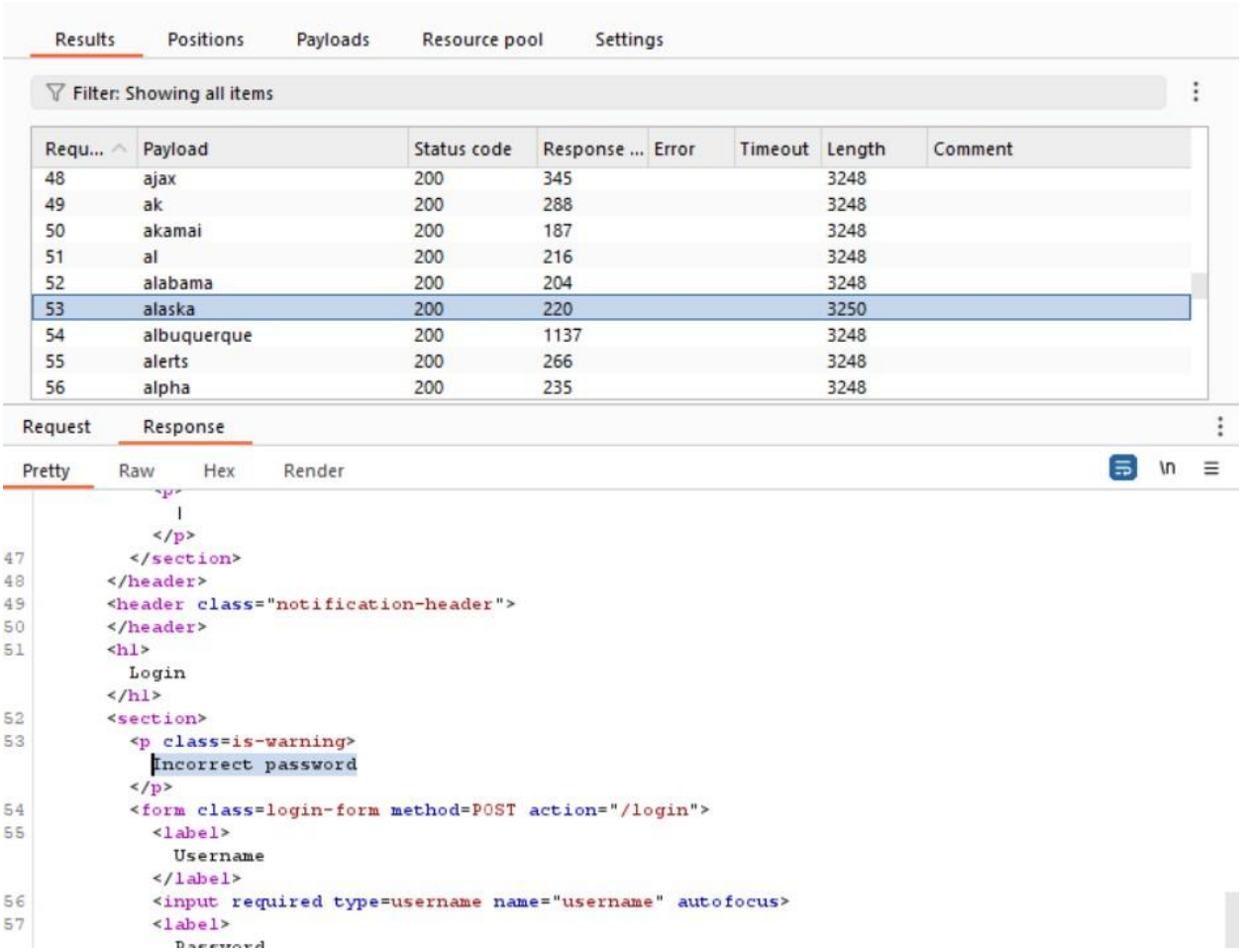


Figure 20: Different response than other.

.

Step 09. Notice that other response messages as “Invalid username”.

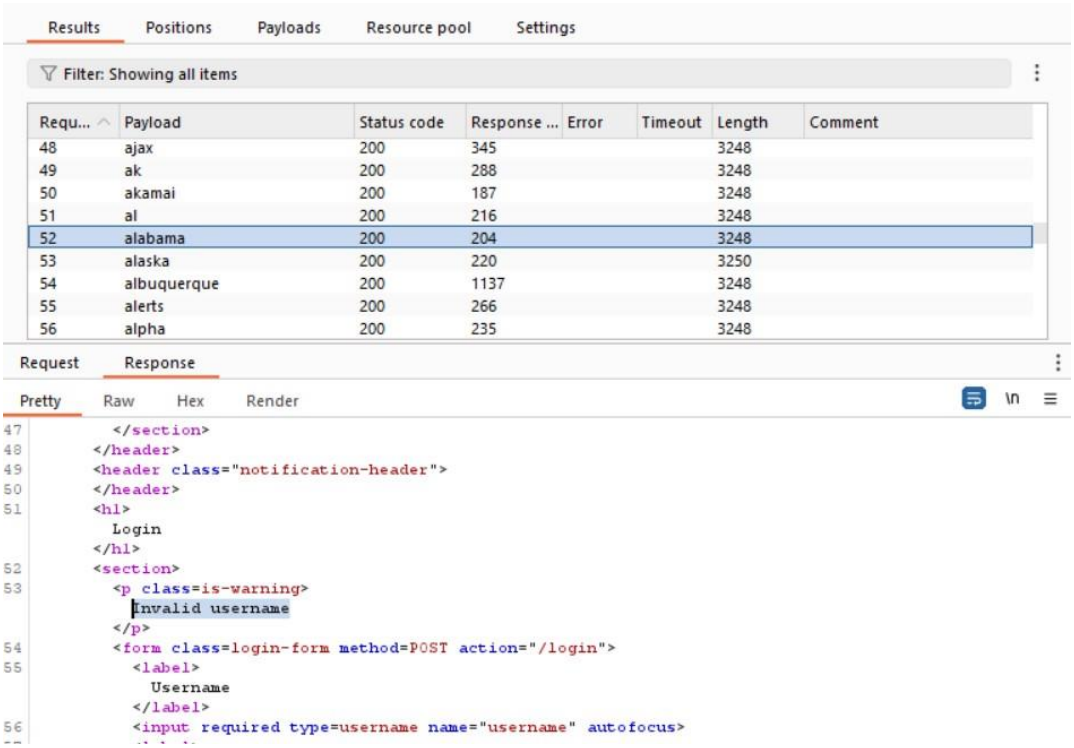


Figure 21: Other response.

Step 10. Clear and change the username parameter to the identified username and add payload position to the password parameter.

Figure 22: Add payload position to password.

Step 11. On the payloads tab, replace the list with candidate passwords and start the attack.

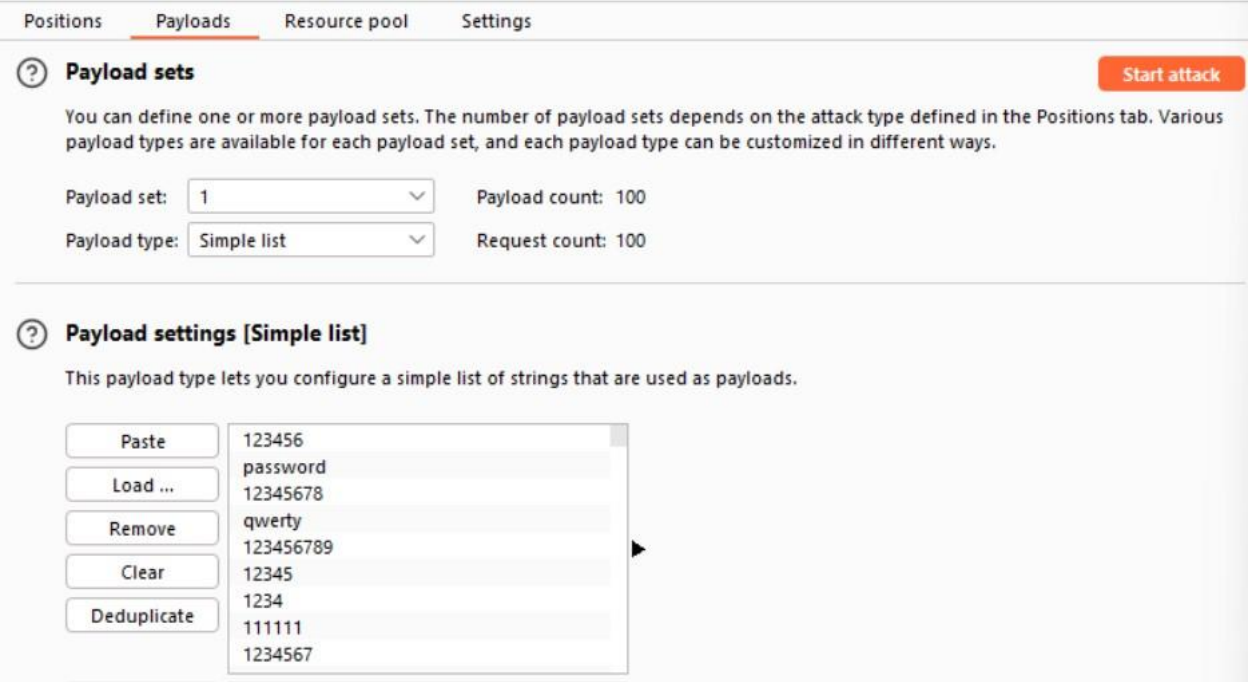


Figure 23: Paste candidate passwords and start the attack.

Step 12. Notice the status column with “302” response, indicating a successful login.

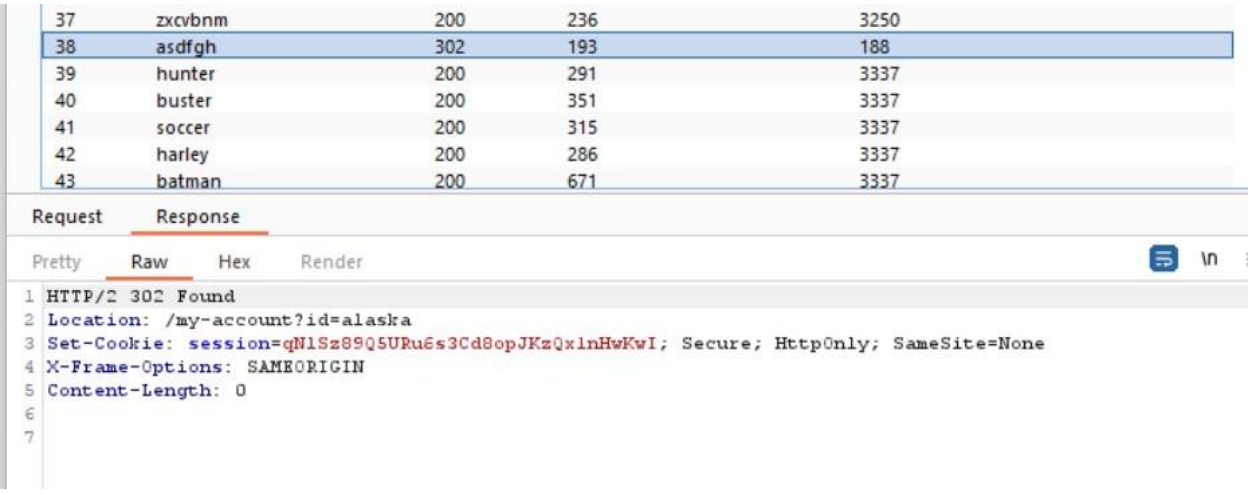


Figure 24: Find "302: response.

Step 13. Now login using the identified username and password.

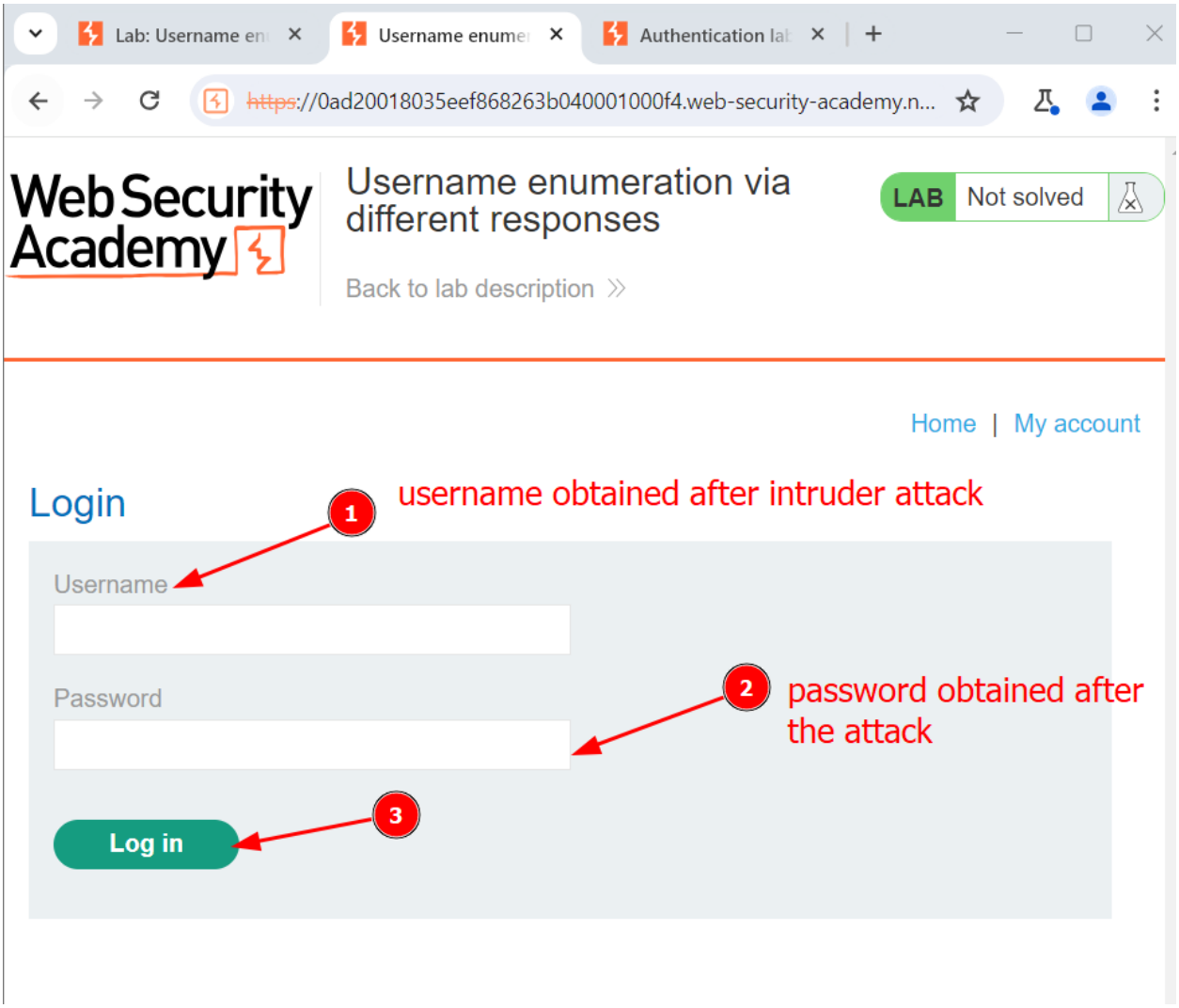


Figure 25: Login with identified username and password.

By exploiting differences in error messages or response codes, attackers can effectively enumerate valid usernames on a web application. Developers must ensure consistent error handling to prevent such enumeration attacks.

2FA Simple Bypass

Demonstrating a "2FA Simple Bypass" vulnerability in a lab environment provides important information on the weaknesses of two-factor authentication (2FA) systems. It highlights how attackers can bypass 2FA easily by looking for setup errors or tricking users. This makes people realise how important it is to have robust security and motivates them to make sure that their real-world authentication systems are reliable.

Title	2FA Simple Bypass
Payload used	
CVSS	10.0
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
Criticality	Critical
OWASP Category	Broken Authentication
Tools used	Burpsuite

Step 01. Follow the login process and enter your credentials.

Figure 26: Login process.

Step 02. Click on the “Email client” button to access to your emails.

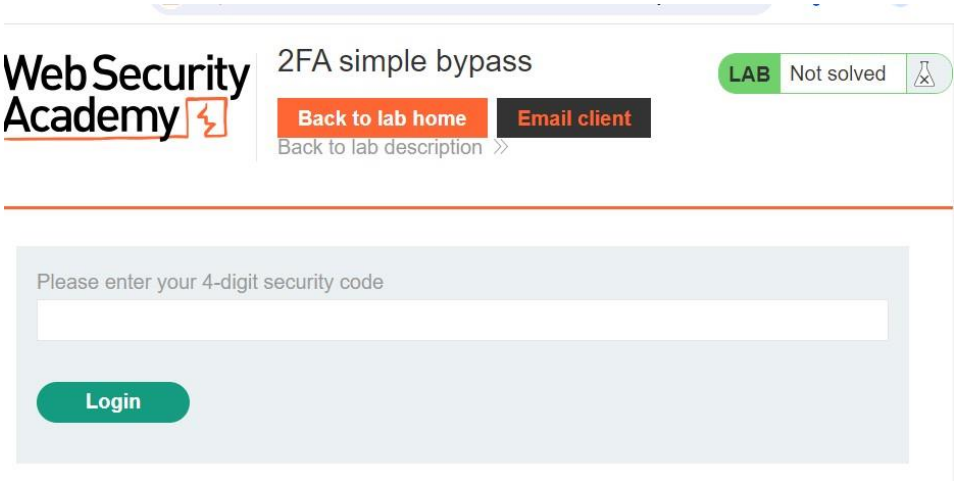


Figure 27: Access to the email.

Step 03. Copy and paste the verification code.

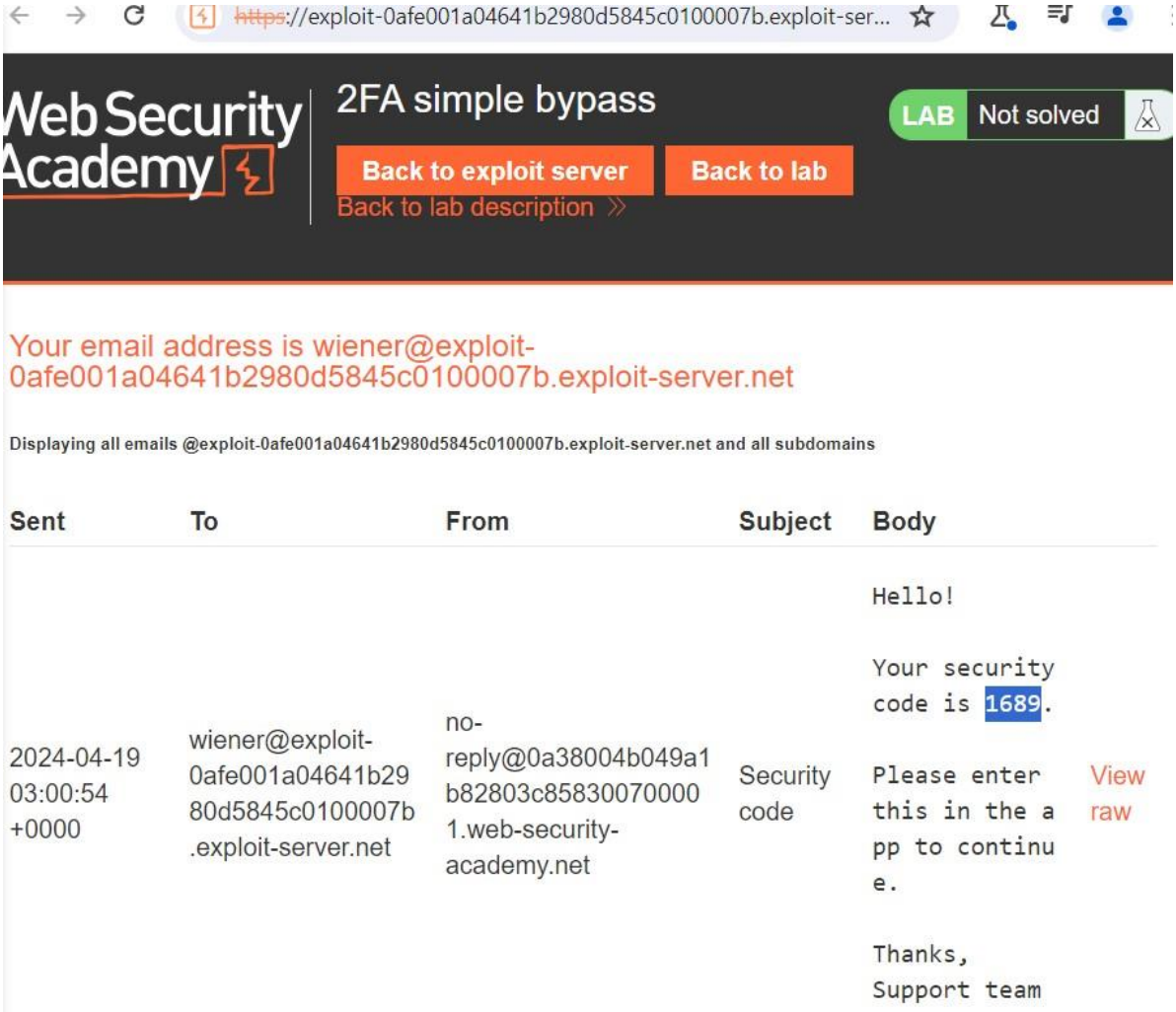


Figure 28: Verification code.

Step 04. Once you have logged in, go to your account page, and make a note of the URL.

Figure 29: Note the URL.

Step 05. Log out your account and use the victim's credentials to initiate the login process.

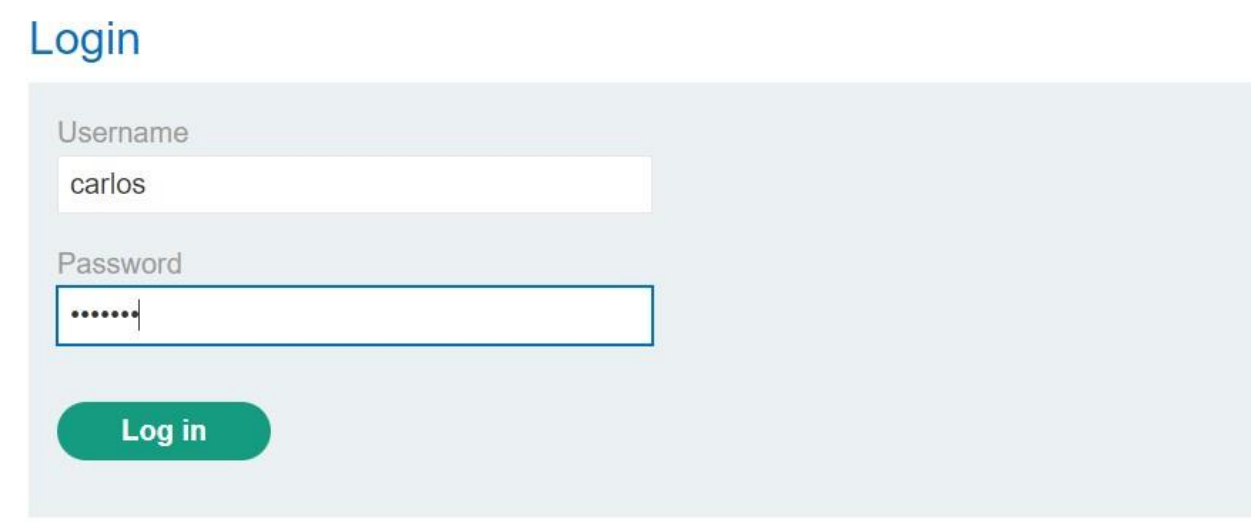


Figure 30: Login using victim's credentials.

Step 06. When you're asked for the 2FA verification code, change the URL in the browser's address bar to navigate to /my-account.

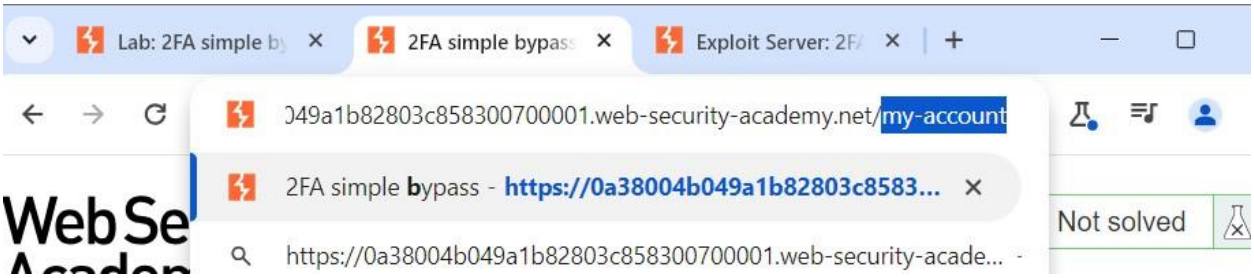


Figure 31: Change URL.

The lab is solved when the /my-account page loads successfully without requiring the 2FA verification code.

By manipulating the URL during the 2FA process, attackers can bypass two-factor authentication and gain unauthorized access to user accounts. Developers must implement robust security measures to prevent such bypass techniques.

Password reset broken logic

To demonstrate how attackers can exploit broken password reset functionality to reset passwords for the users; accounts on a web application.

Title	Password reset broken logic
Payload used	Username and Password
CVSS	10.0
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
Criticality	Critical
OWASP Category	Broken Authentication
Tools used	Burpsuite

Step 01. With Burp Suite running, click the "Forgot your password?" link on the login page.

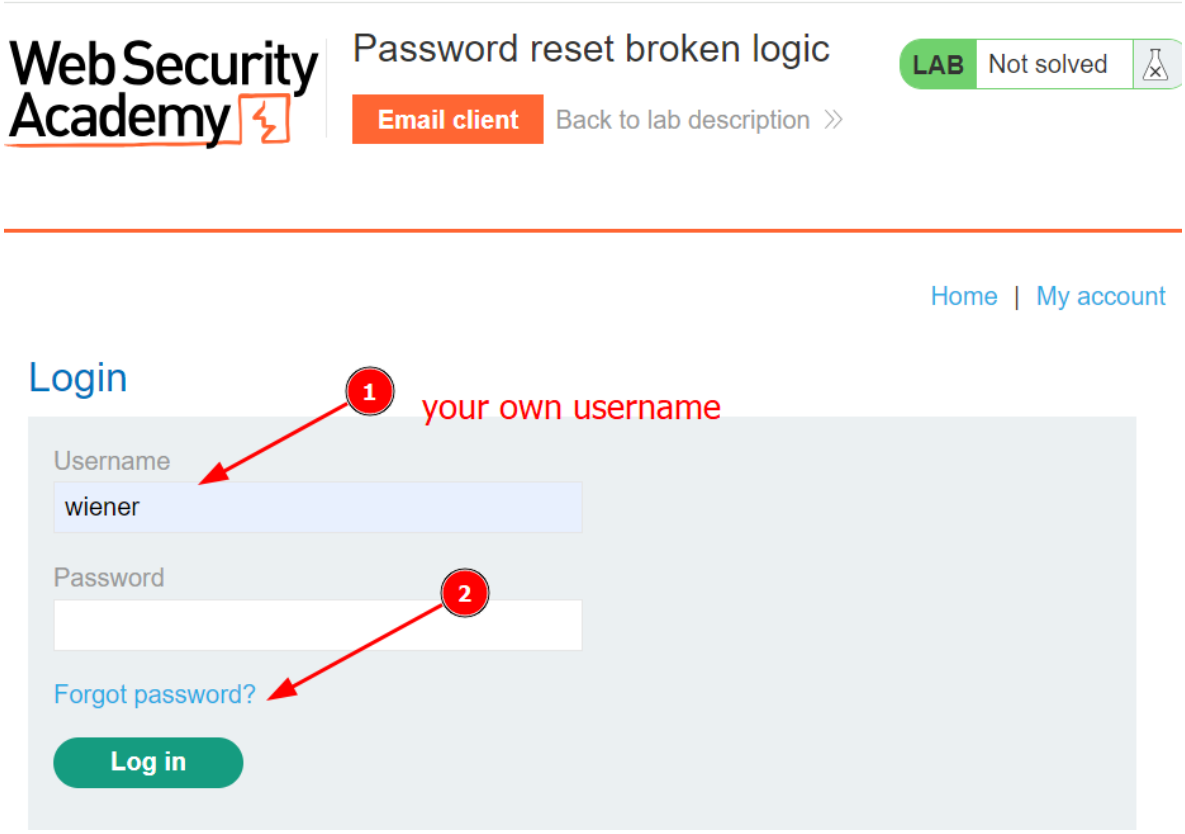


Figure 32: Login process.

Step 02. Enter and submit your username. Click the ‘email client” button to view the password reset email.

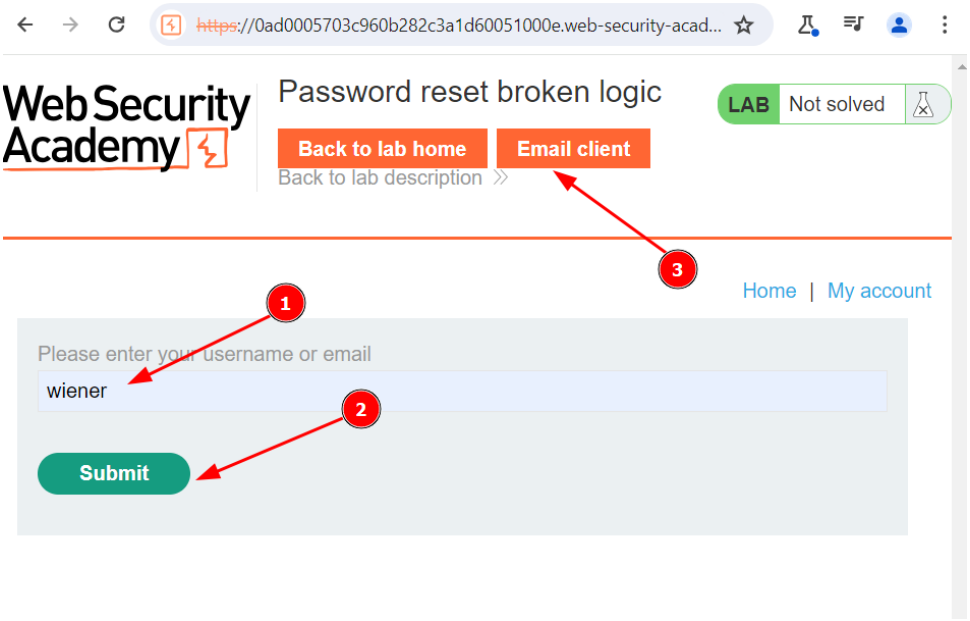


Figure 33 Access to the email.

Step 03. Access the password reset email and click the provided link.

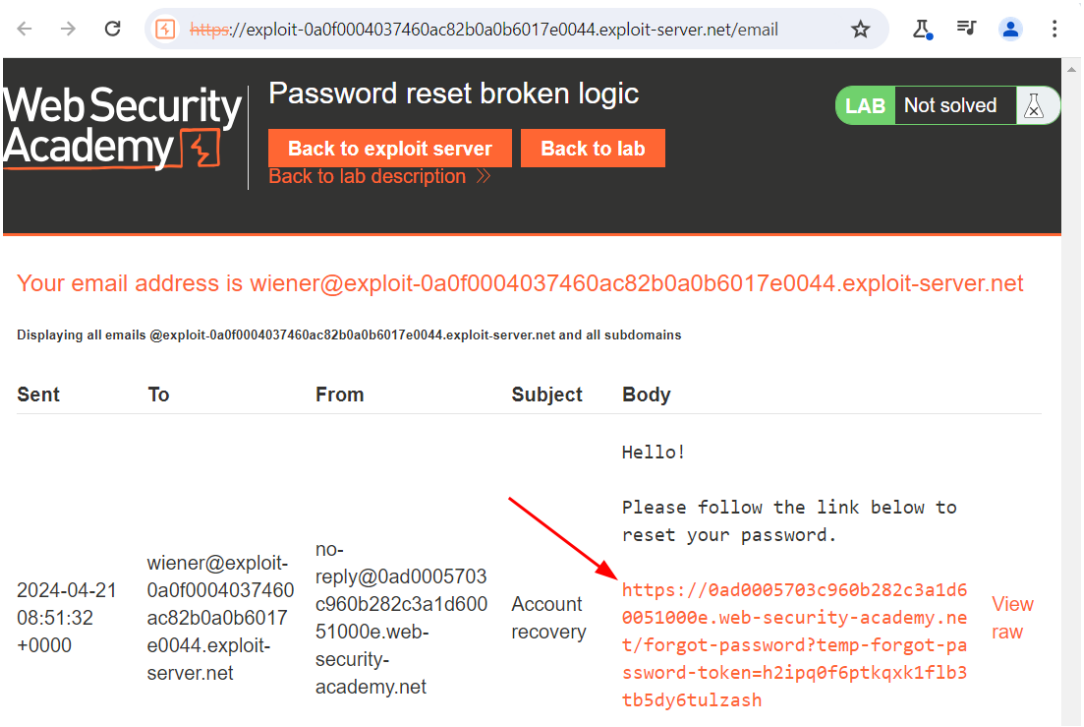


Figure 34: Click the provided link.

Step 04. Reset your password to a new one of your choice.

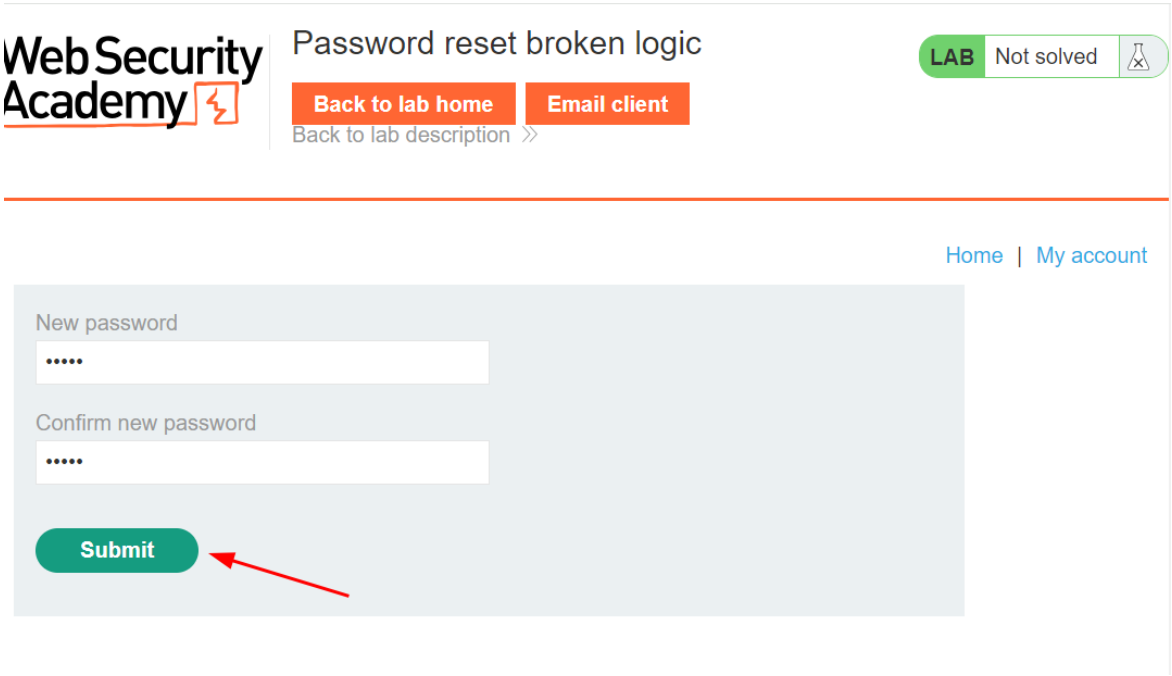


Figure 35: Password reset process.

Step 05. In Burp Suite's Proxy tab, go to the HTTP history. Notice that when submitting the new password, the **POST/forgot-password?temp-forgot-password-token** request contains the username as a hidden input. Now send the **POST** request to the Burp Repeater.

Burp Repeater enables you to work on multiple messages simultaneously, each in its own tab. Any modifications you make to a message are saved in the tab's history (Portswigger, 2024).

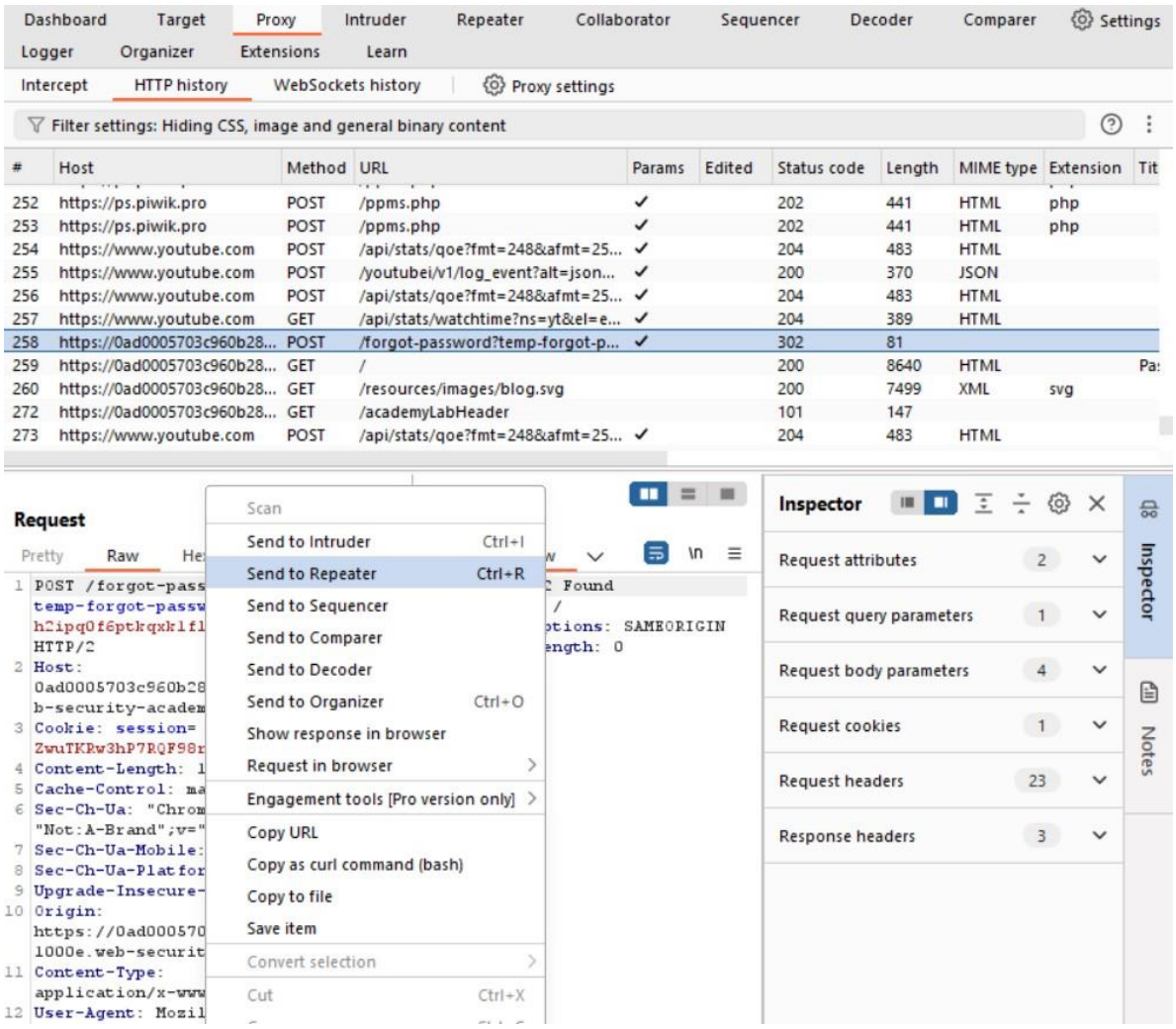


Figure 36: Send POST request to Repeater.

Step 06. Delete the value of the temp-forgot-password-token parameter in both the URL and request body. Observe that the password reset functionality still works, confirming that the token is not being checked.

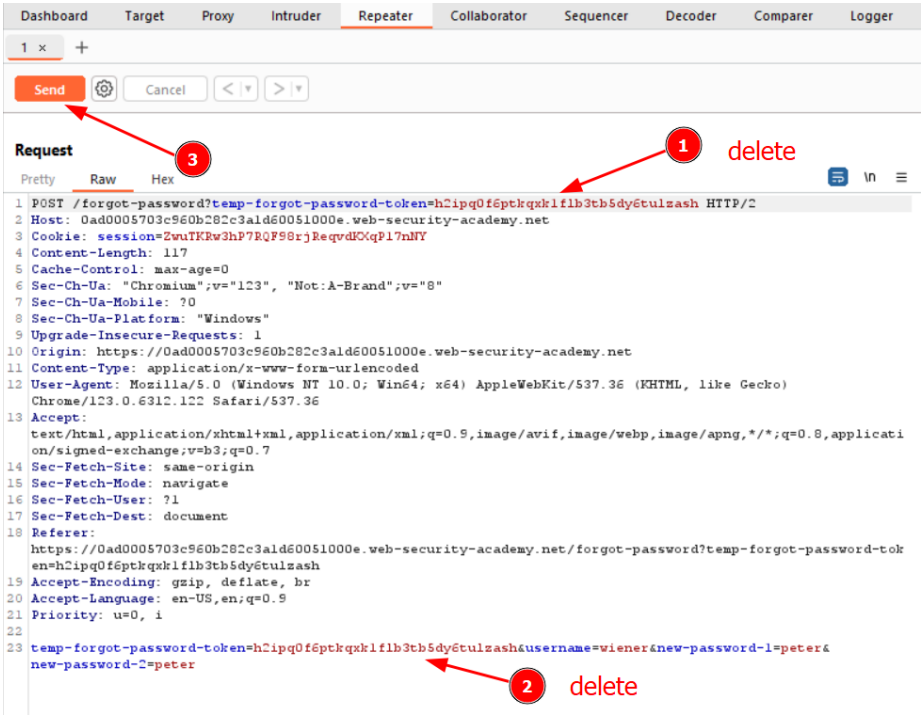


Figure 37: Delete the temp-forgot-password-token value.

Step 07. Request a new password reset and change it again.

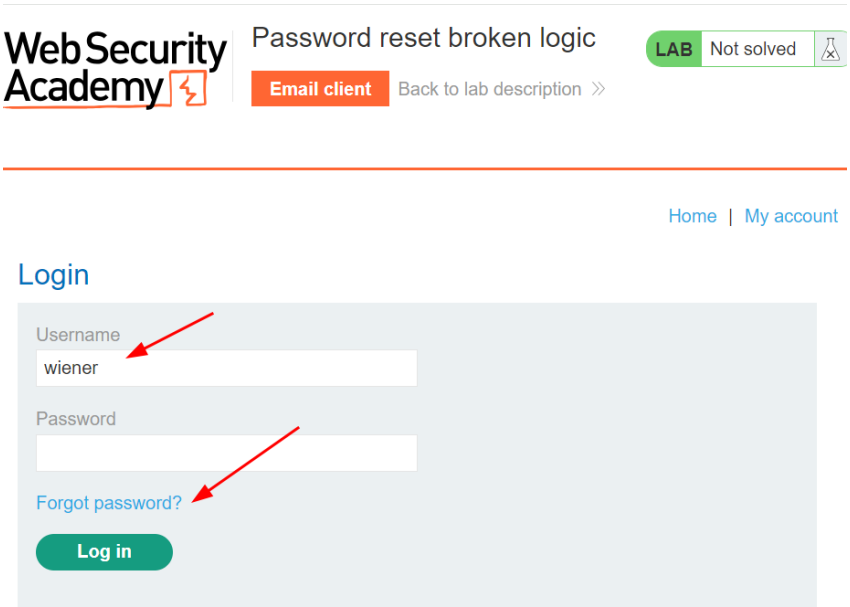


Figure 38: New password reset.

Step 08. Send the **POST/forgot-password?temp-forgot-password-token** request to Burp Repeater.

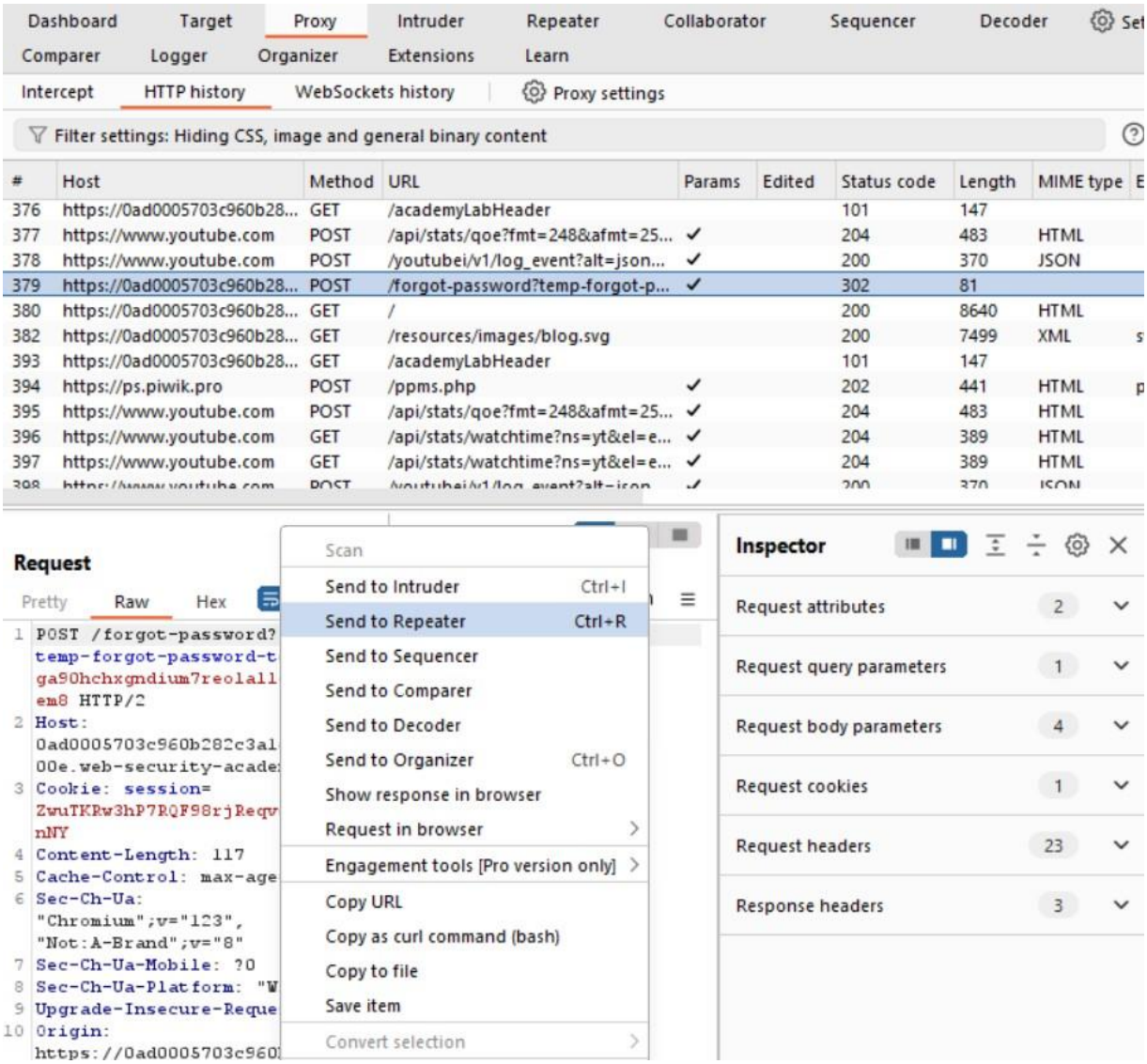


Figure 39: Send it to Repeater.

Step 09. Delete the value of the temp-forgot-password-token parameter in both the URL and request body. Change the username parameter to the target user's username (carlos).

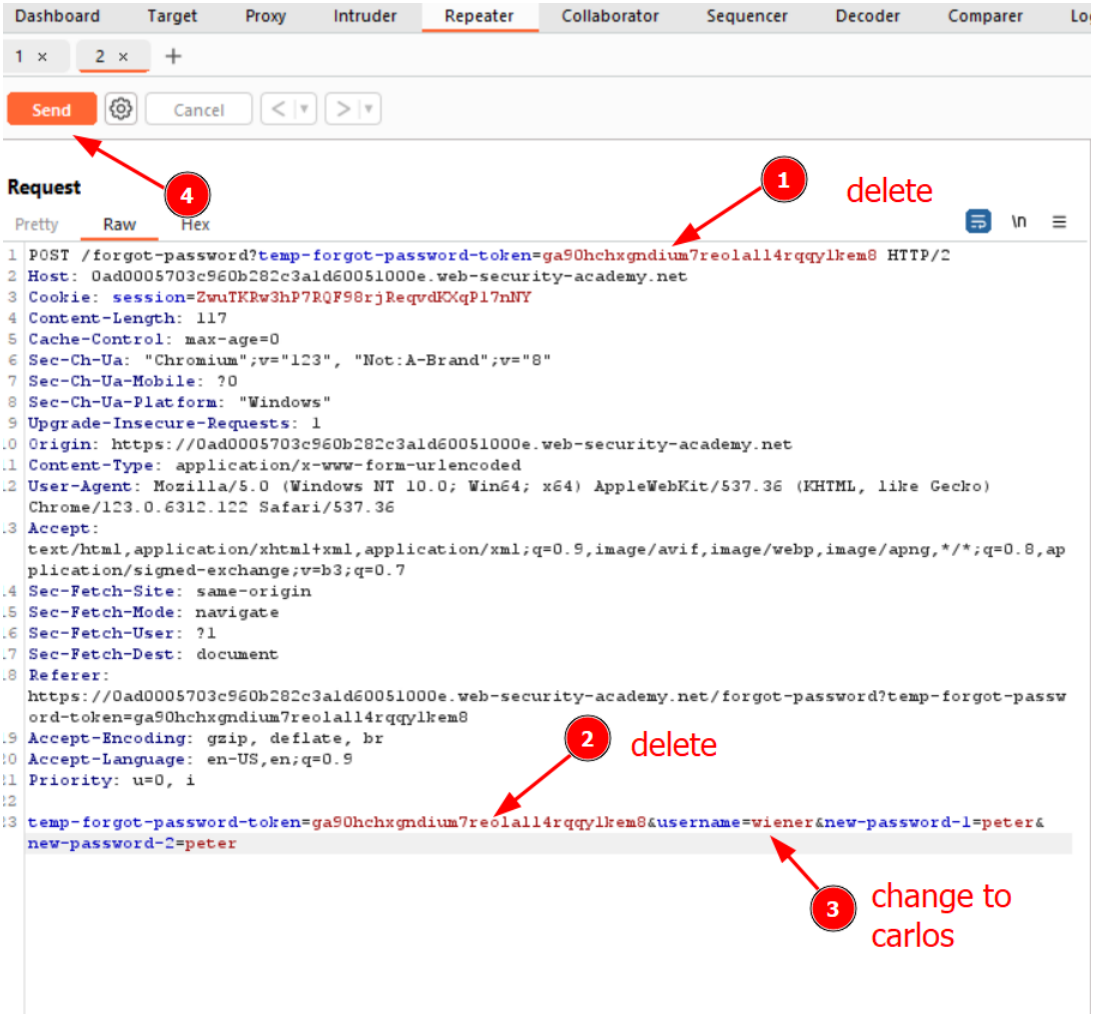


Figure 40 Change the username.

Step 13. Now log in to the victim's account.

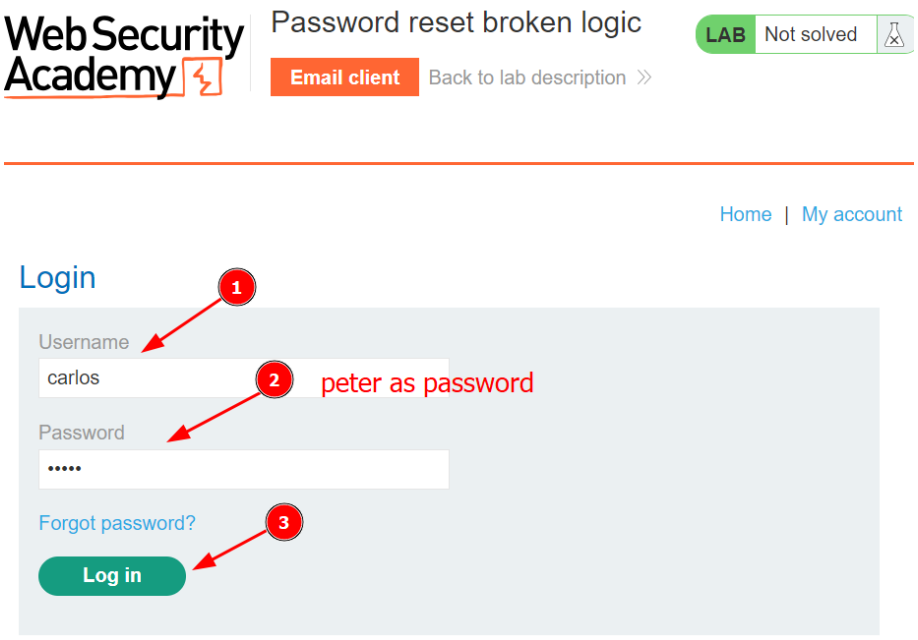


Figure 41: Access victim's account.

Password reset broken logic can allow attackers to reset passwords for other users' accounts and gain unauthorized access. Developers must implement robust password reset mechanisms to prevent such exploits and protect user accounts.

Username enumeration via account lock

Demonstrating "Username Enumeration via Account Lock" in a lab setting explains how hackers might take advantage of account lock mechanisms to find valid usernames. The demonstration highlights the significance of consistent error handling by demonstrating how attackers might deduce usernames from variations in system responses to locked and non-existent accounts. After understanding the risks of implementing account locks properly, participants are inspired to improve security measures to prevent username enumeration assaults.

Title	Username Enumeration via Account lock
Payload used	Username and password
CVSS	9.3
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:L/A:H
Criticality	Critical
OWASP Category	Broken Authentication
Tools used	Burpsuite

Step 01. While the Burp Suite is running, investigate the login page and submit an invalid username and password. In the Proxy tab, go to the HTTP history and find the **POST/login** request. Send it to Burp Intruder.

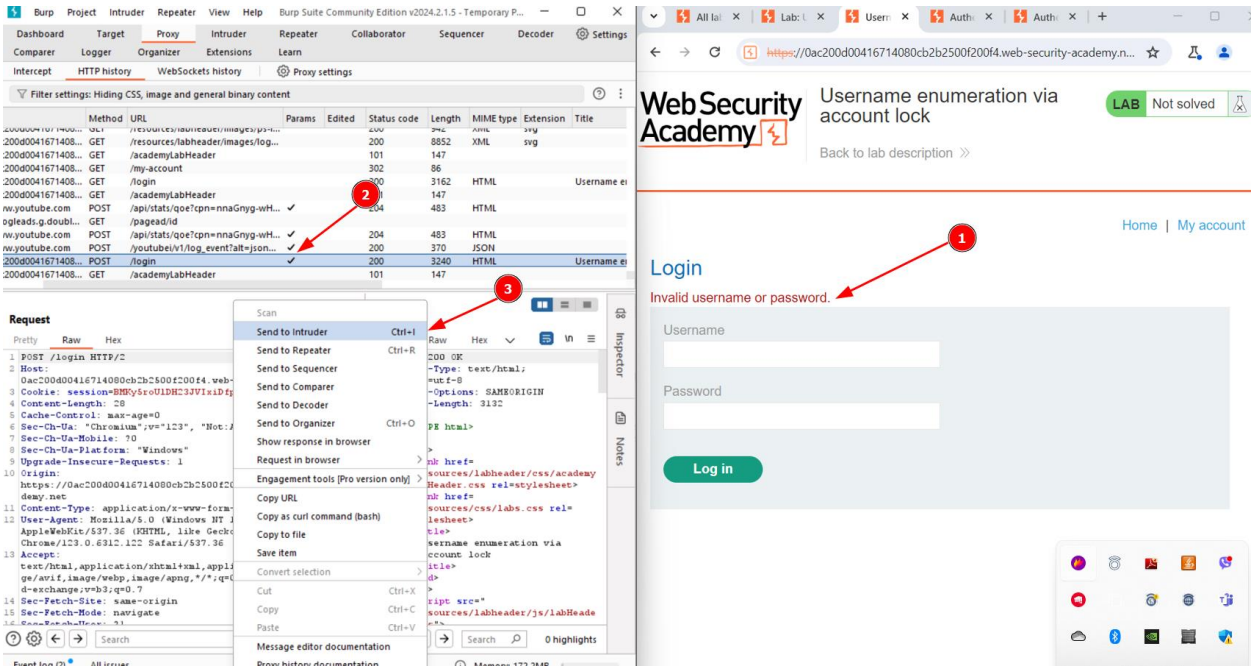


Figure 42: Invalid login process and send POST/login request to Intruder.

Step 02. In Burp Intruder, select the Cluster bomb attack type. Add a payload position to the username parameter and a blank payload position to the end of the request body.

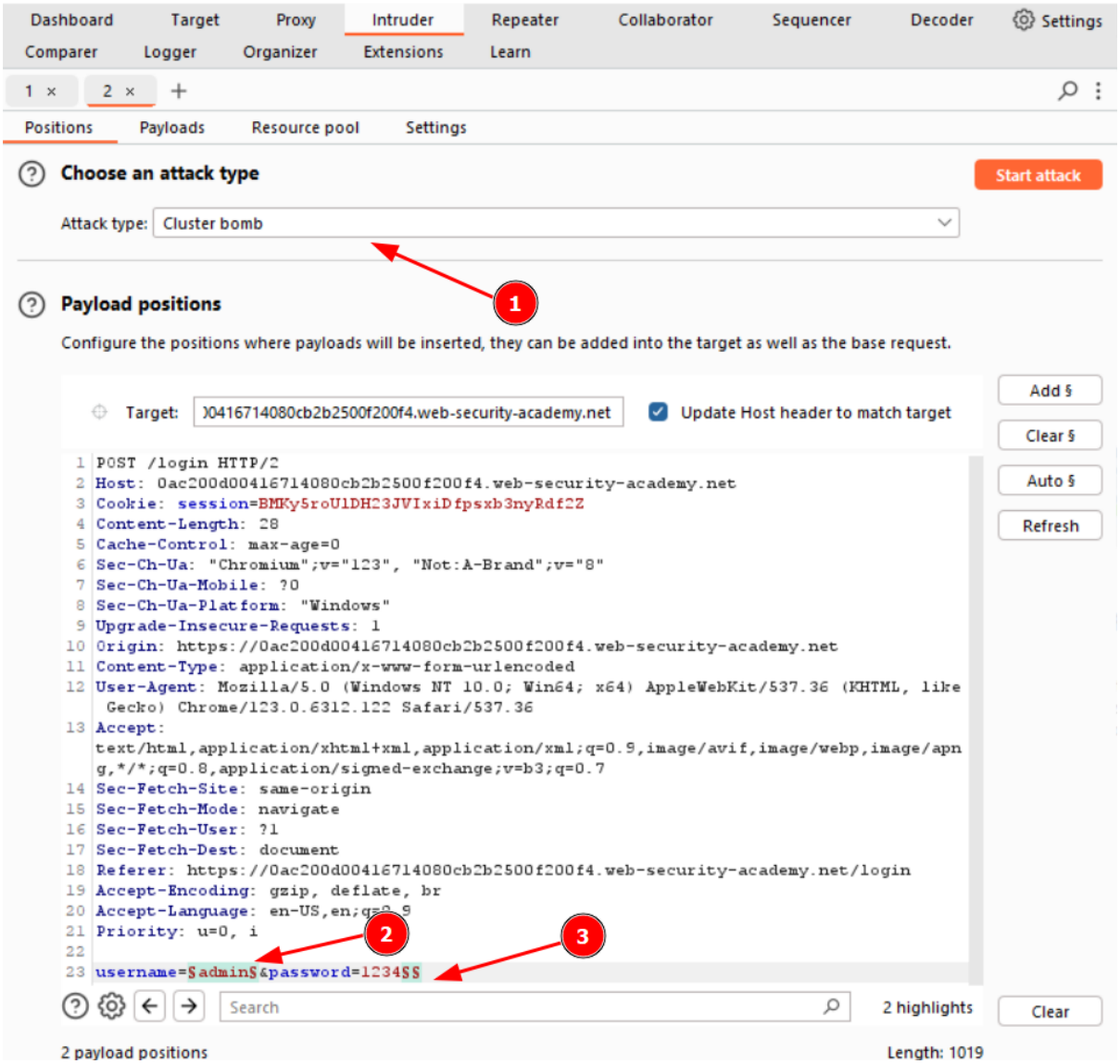


Figure 43: Add payload positions.

Step 03. On the payloads tab, for the first set, select Simple list payload type. Paste the list of candidate usernames.

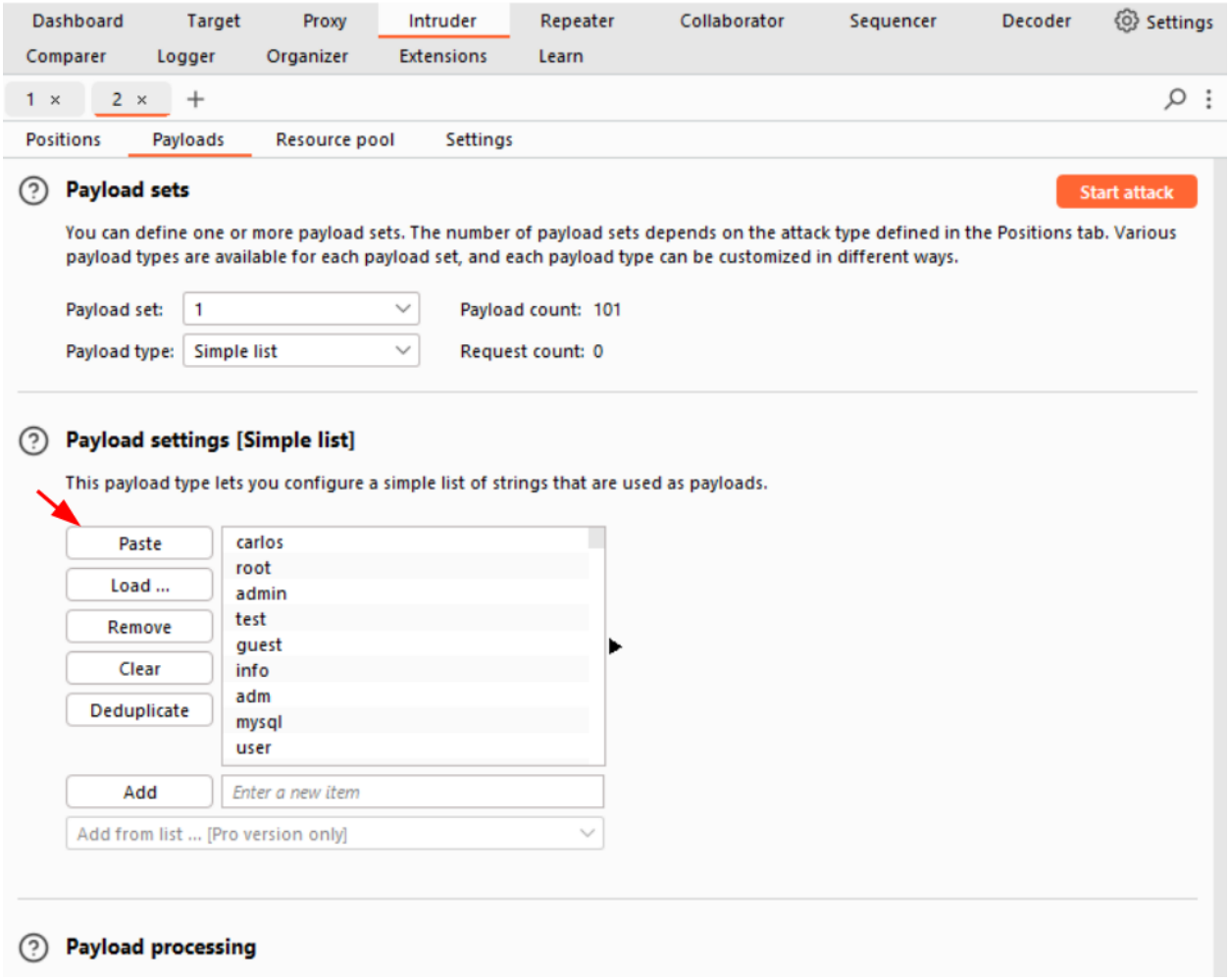


Figure 44: First payload set.

Step 04. For the second set, select the Null payloads type and generate 5 payloads. Start the attack. In the attack results, note the username with longer responses and a different error message indicating too many incorrect login attempts.

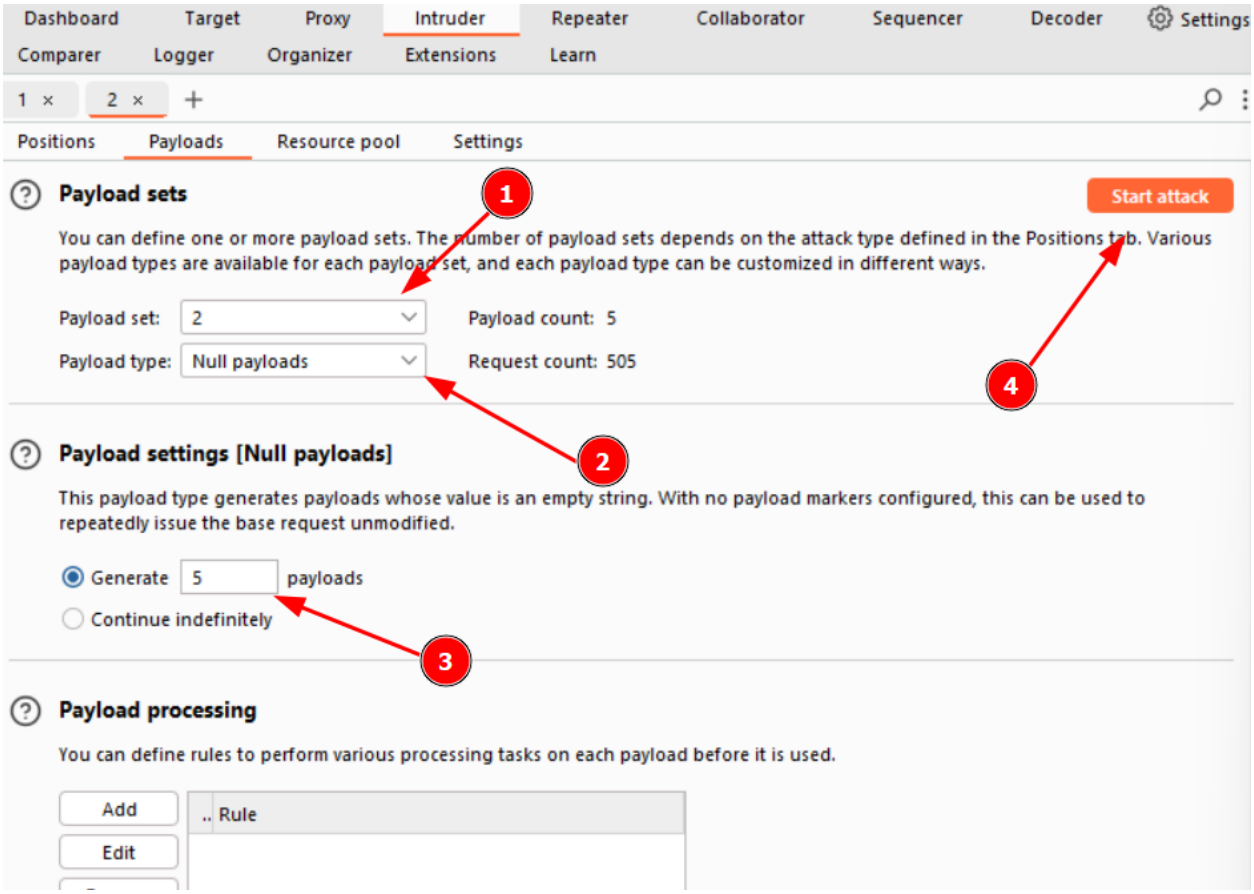


Figure 45: Second payload set and start attack.

Step 05. Create a new Burp Intruder attack on the POST/login request and select the sniper attack type. Set the username parameter to the identified username and add a payload position to the password parameter.

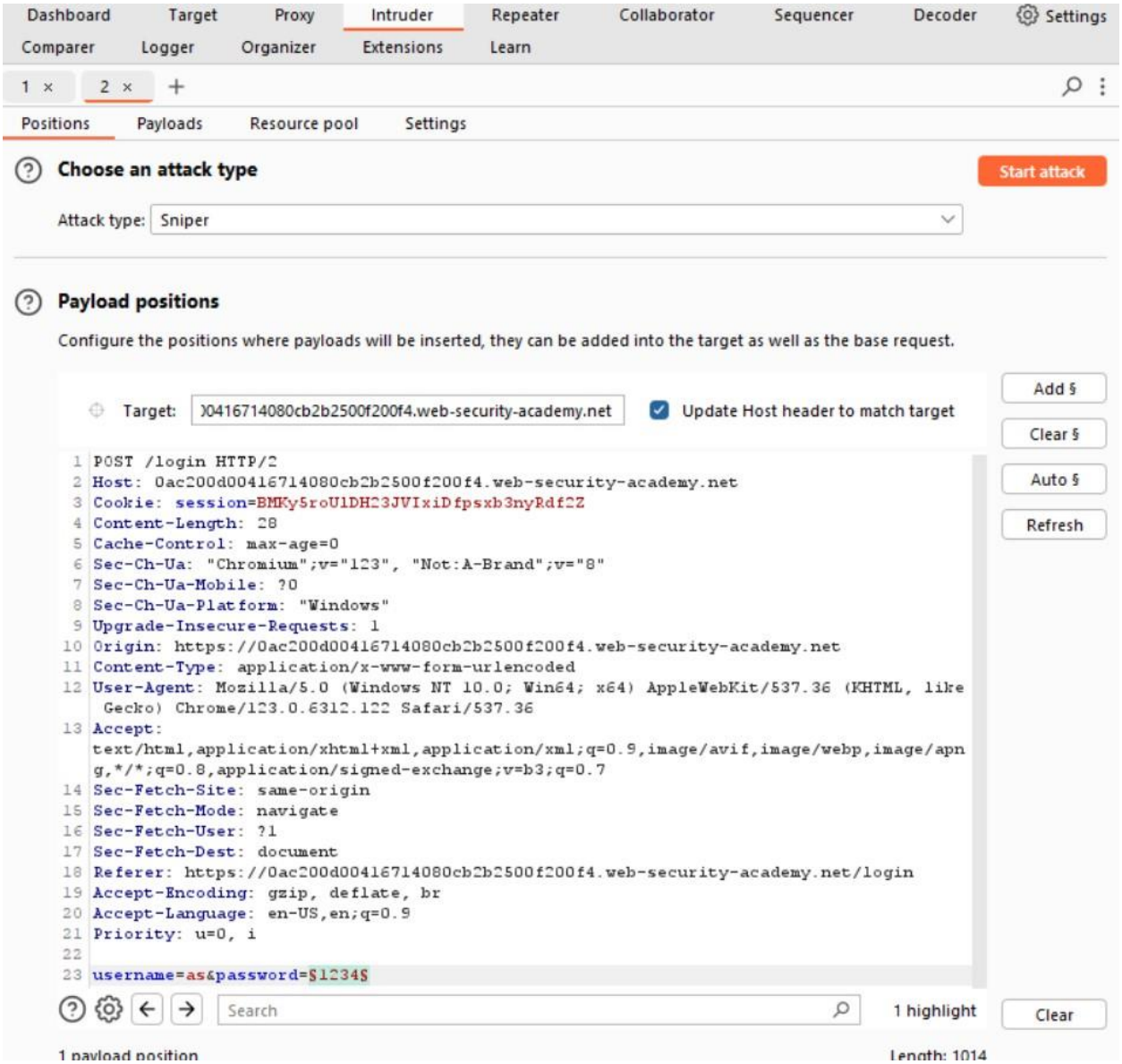


Figure 46: New Intruder attack and set payload positions.

Step 06. Add the list of passwords to the payload set.

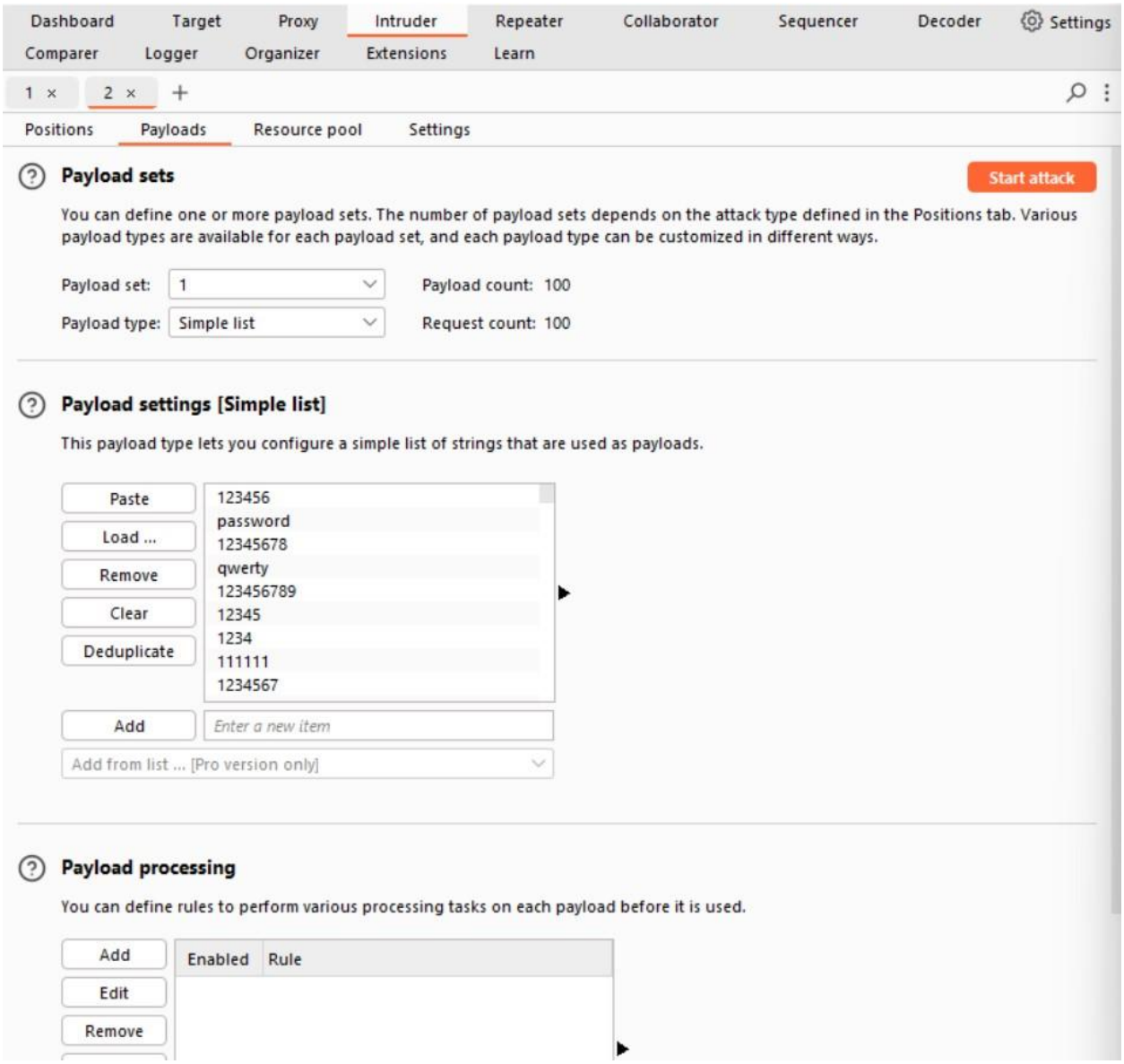


Figure 47: Paste candidate passwords and start the attack.

Step 07. Create a grep extract rule for the error message and start the attack.

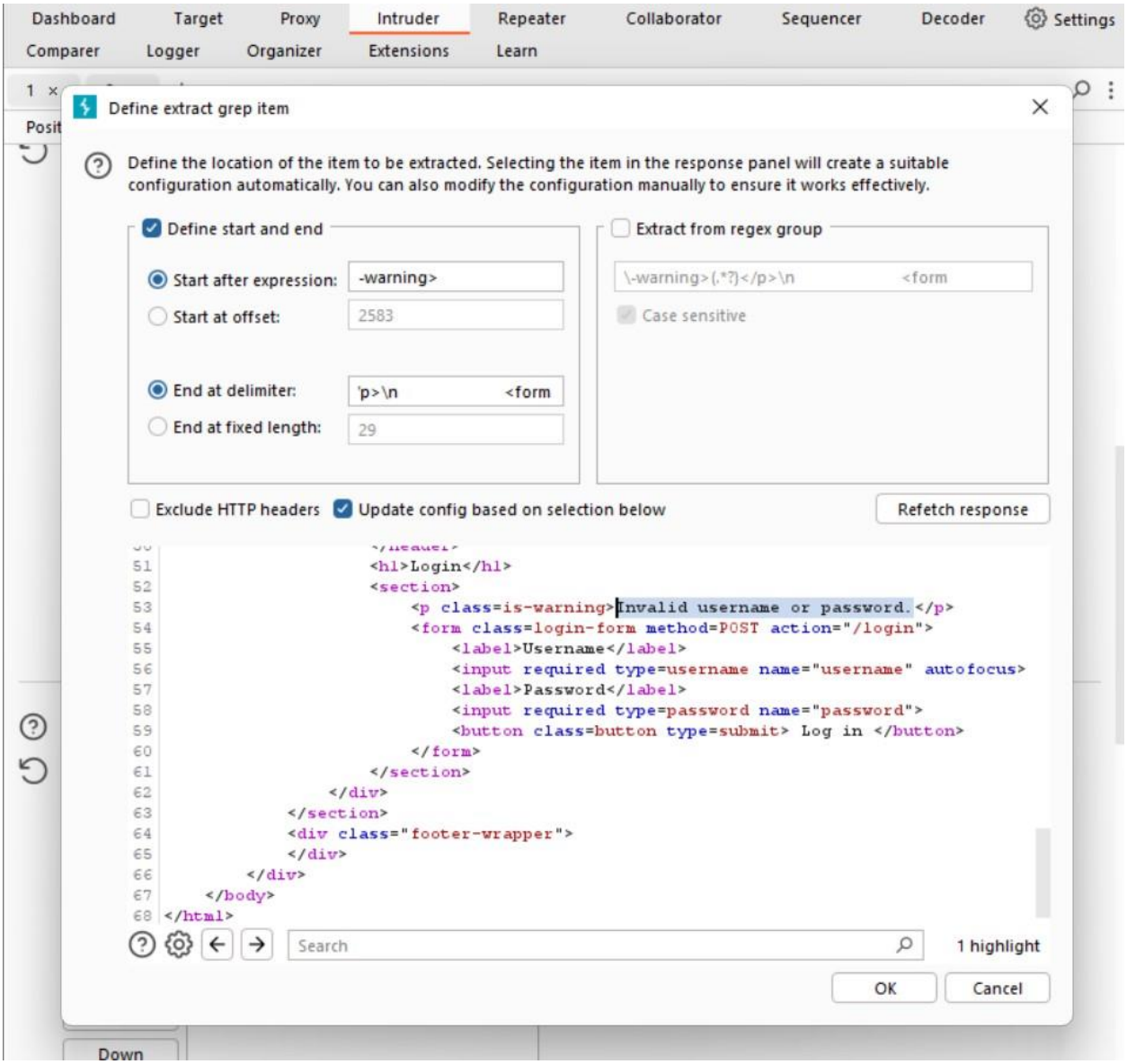


Figure 48: Grep extract rule.

Step 08. In the attack results, at the grep extract column, note the password that didn't return any error message.

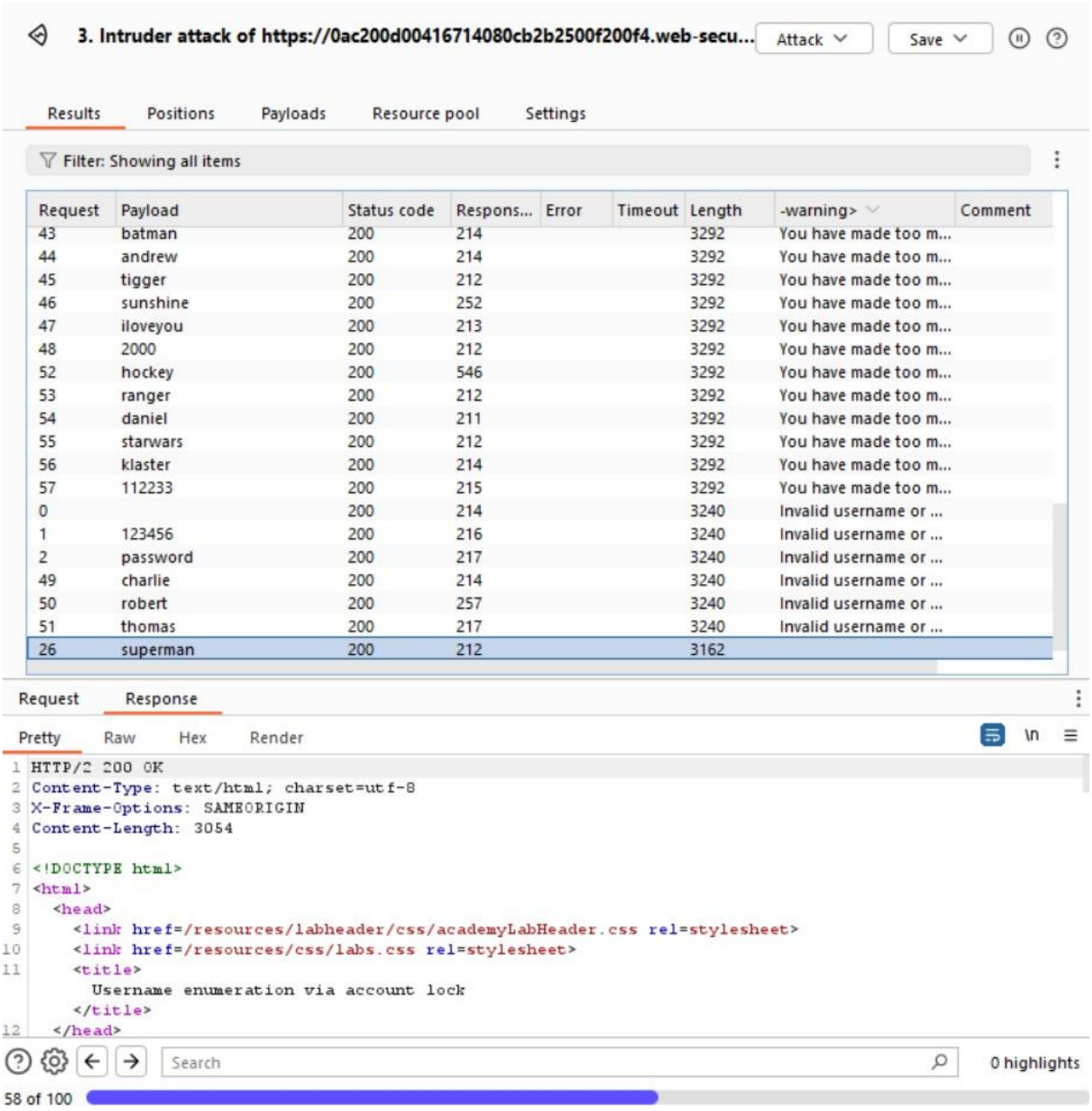


Figure 49: Examine the attack.

Step 09. Log in using the identified username and password.

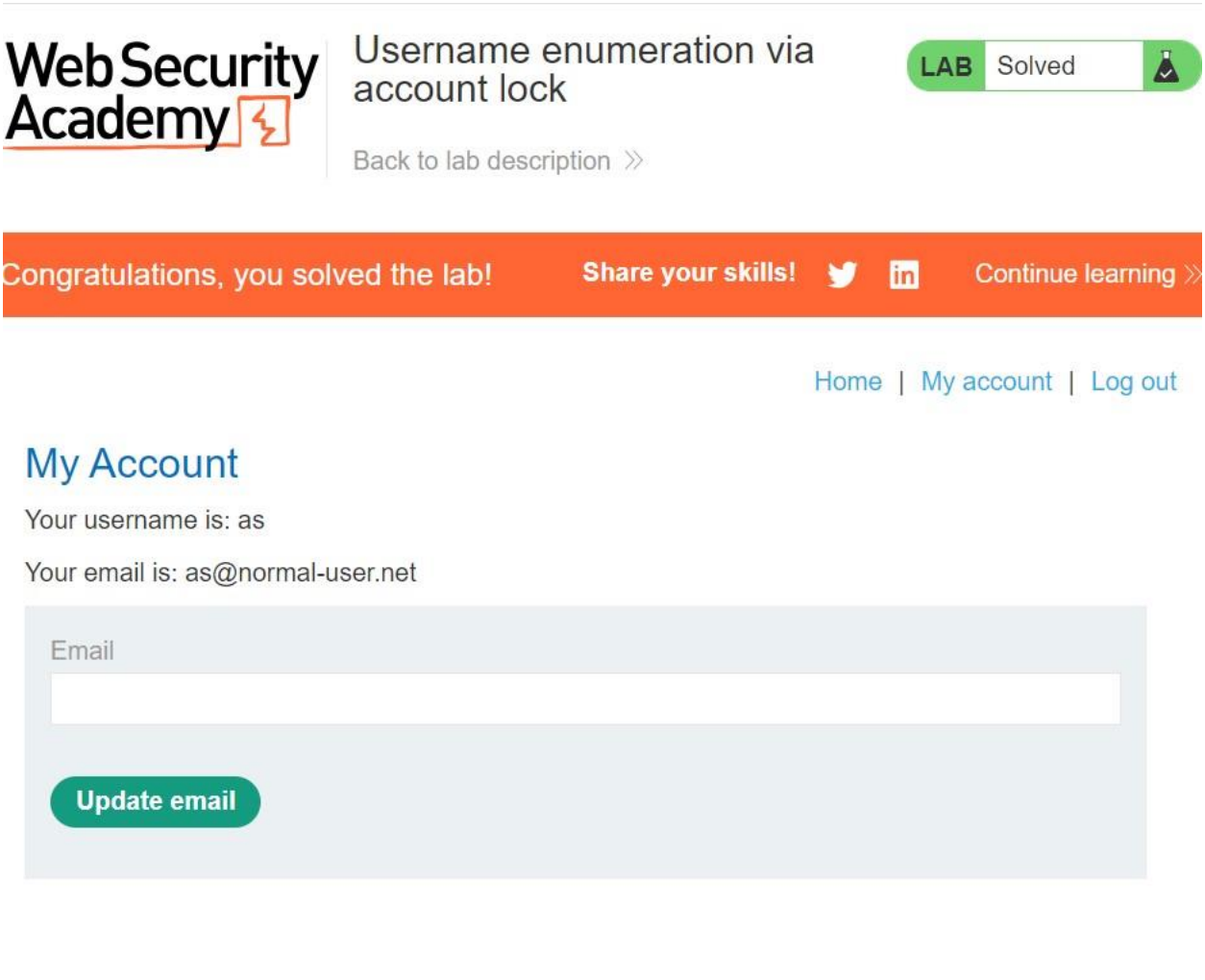


Figure 50: Login to the victim's account.

4. Mitigation

Preventing broken authentication requires a different strategy to deal with vulnerabilities at various stages of the authentication and session management processes.

a. Implement Multi-Factor Authentication (MFA)

   MFA is one of the best ways to prevent broken authentication attacks. It improves security by requiring two or more verification methods, such as biometrics, or OTPs. It helps in preventing brute-force attacks, credential stuffing, and the reuse of stolen credentials.

b. Create Strong Password Policies

   Implementing strong password policies is important for preventing broken authentication attacks. Ensure that passwords are complex, are changed regularly, and are not shared between different accounts. Use a combination of capital and lowercase letters, numbers, and special characters to create strong passwords. Educate users on the importance of having unique passwords (Maric, 2024).

c. Control Session Length

   To stay safe from hackers messing with your logins, it's good idea to keep the online sessions short. It means that your session on a website or app is limited in time once you log in. As a result, it is more difficult for attackers to log into your account and pretend as you. Also, it's a good idea to have a website log you out automatically if you aren't using it for a long. In this way, in case that you forget to log out, nobody else will be able to access your data while you're not around (Maric, 2024).

d. Rotate and Invalidate Session IDs

To stay safe online, it's important to rotate and invalidate session IDs. Consider these IDs like secret codes that identify your online session. It is more difficult for hackers to figure out if they are changed regularly or after significant acts. Furthermore, it's a good idea to remove the session ID as soon as you're done using a website. In this way, even if someone gets their hands on it, they can't use it to sneak into your account later (Maric, 2024).

e. Implement Brute-Force Protection

To protect your system from brute-force attacks, you can set limits on how many times someone can try to log in. If they make too many mistakes, their account will be temporarily locked. After a certain number of unsuccessful attempts, you can even force them to wait longer between tries. It can also be beneficial to include CAPTCHAs to prevent automated attempts by bots. Furthermore, you can restrict access to users from trustworthy sources or ban questionable IP addresses that attempt multiple login attempts. By taking these precautions, attackers will find it much more difficult to guess passwords and access your system (Maric, 2024).

These mitigation strategies help improve the overall security posture of web applications by solving different aspects of broken authentication vulnerabilities. Organisations must employ some of these strategies that are specific to their individual requirements and risk assessment.

5. Evaluation

Multi-Factor Authentication (MFA)

Multi-factor authentication (MFA) enhances security by requiring multiple forms of verification, reducing the risk of unauthorized access, even if one factor is exposed. It is used in many different industries, including corporate networks, email services, and online banking. However, MFA can make things less user-friendly and require additional implementation and maintenance efforts. Furthermore, some techniques, such as biometrics, may cause privacy issues and problems with regulatory compliance. Despite these challenges, MFA remains a valuable tool for organizations aiming to improve authentication security. It is particularly well-suited for applications handling sensitive data, such as online banking, healthcare portals, and corporate networks.

Strong Password Policies

Strong Password Policies provide significant advantages in enhancing the security of user accounts by enforcing complex password requirements and mitigating the risk of password-based attacks such as brute force and dictionary attacks. They provide a minimum level of security and apply to almost all systems and applications that demand user authentication. These rules, however, mostly rely on user awareness and compliance, which can be difficult to enforce and may result in resistance or non-compliance. Furthermore, overly tight rules might frustrate users and promote password reuse, which could weaken security measures. Regular maintenance and resource allocation are required to keep up with creating threats and best practices, requiring regular upgrades and adjustments. Almost all systems and applications that need user authentication, such as social media platforms, e-commerce websites, and business systems, can benefit from Strong Password Policies.

Control Session Length

Control Session Length offers significant advantages in mitigating the risk of session hijacking attacks by limiting the duration of user sessions and reducing the range of opportunities for attackers to exploit stolen credentials. This mitigation technique is applicable to a wide range of systems, including web applications, online services, and remote access systems. Still, there are drawbacks to consider. Short session lengths can disrupt user experience and productivity, leading to frustration and potential abandonment of the system. The compromises between security and usability must be carefully considered before implementing this strategy, as excessive controls could make it inconvenient for users. Furthermore, implementing shorter session durations can occasionally require users to complete extra verification processes to extend their sessions, complicating the user experience. Control Session Length is important for web applications, online services, and remote access systems where session security is necessary.

Rotate and Invalidate Session IDs

Rotate and Invalidate Session IDs offer significant advantages in preventing session fixation and replay attacks by regularly changing session identifiers, improving security without significantly impacting user experience. This mitigation technique is applicable to a wide range of systems, including web applications, online services, and session-based authentication systems. However, there are disadvantages to consider. Frequent rotation of session IDs may lead to increased server load and impact application performance, particularly in high-traffic environments. Furthermore, to provide flawless session management, applying this technique involves cooperation between client and server components, which could be difficult in systems that are remote or complicated. In addition, the regular switching of session IDs could cause issues with state management and session persistence, which could have unexpected consequences or vulnerabilities. Rotate and Invalidate Session IDs is commonly used in web applications, online services, and session-based authentication systems to prevent session-related attacks.

6. Conclusion

In conclusion, the exploration of broken authentication vulnerabilities in web applications highlights the serious security implications of these vulnerabilities and the urgent need for effective mitigation techniques. We explored the reasons behind these vulnerabilities as well as their effects and possible solutions, and we have concluded that robust authentication processes are essential to protecting sensitive data. The accessibility with which attackers might exploit these vulnerabilities has been highlighted through practical demonstrations, highlighting the significance of taking preventive action. Although there are many interesting possibilities for development with different mitigation strategies, it is important to recognise their limitations and support a comprehensive security approach. Prioritising the implementation of secure authentication techniques and maintaining an attentive focus on new risks is essential for organisations to properly protect their web applications and minimise the risks associated with hacked authentication. Organisations can minimise the potential consequences of security breaches and maintain the integrity and confidentiality of their digital assets in an increasingly connected world by implementing proactive security measures and building a strong security infrastructure.

References

Bhaktavatsalam, S. V., 2018. *Marriott Hit by Starwood Hack That Ranks Among Biggest Ever - Bloomberg.* [Online] Available at: https://www.bloomberg.com/news/articles/2018-11-30/marriott-found-unauthorized-starwood-database-access-since-2014-jp3xbq64
[Accessed 20 April 2024].

Bhargav, A., 2024. *Authentication vs Authorization – What's the difference?.* [Online] Available at: https://www.ssl2buy.com/wiki/authentication-vs-authorization-whats-the-difference
[Accessed 16 April 2024].

Boston University, 2024. *Understanding Authentication, Authorization, and Encryption : TechWeb : Boston University.* [Online] Available at: https://www.bu.edu/tech/about/security-resources/bestpractice/auth/#:~:text=Authorization%20is%20a%20process%20by,is%20that%20is%20requesting%20access.
[Accessed 15 April 2024].

Chua, I., 2022. *Real Life Examples of Web Vulnerabilities (OWASP Top 10).* [Online] Available at: https://www.horangi.com/blog/real-life-examples-of-web-vulnerabilities
[Accessed 20 April 2024].

Contrast Security, 2024. *Broken Authentication.* [Online] Available at: https://www.contrastsecurity.com/glossary/broken-authentication#:~:text=Broken%20authentication%20attacks%20aim%20to,details%20to%20assume%20user%20identities.
[Accessed 15 April 2024].

Fortinet, 2024. *What is a Brute Force Attack? | Definition, Types & How It Works.* [Online] Available at: https://www.fortinet.com/resources/cyberglossary/brute-force-attack
[Accessed 15 April 2024].

Geeksforgeeks, 2024. *Broken Authentication Vulnerability - GeeksforGeeks.* [Online] Available at: https://www.geeksforgeeks.org/broken-authentication-vulnerability/
[Accessed 15 April 2024].

Gupta, D., 2024. *Marriott Data Breach 2020: 5.2 Mn Guest Records Were Stolen | LoginRadius.* [Online] Available at: https://www.loginradius.com/blog/identity/marriott-data-breach-2020/#:~:text=In%20mid%2DJanuary%202020%2C%20Marriott,following%20a%20breach%20in%202018.
[Accessed 15 April 2024].

Gupta, D., 2024. *Understanding Broken Authentication: Risks & Prevention.* [Online] Available at: https://loginradius.com/blog/identity/what-is-broken-authentication/
[Accessed 15 April 2024].

LinkedIn, 2023. *Understanding Broken Authentication and Its Implications for Cybersecurity.* [Online]
Available at: https://www.linkedin.com/pulse/understanding-broken-authentication-its-implications-cybersecurity#:~:text=Broken%20authentication%20and%20session%20management%20vulnerabilities%20arise%20when%20these%20tokens,used%20to%20guess%20user%20credentials.
[Accessed 15 April 2024].

Maric, N., 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 15 April 2024].

Microsoft Security, 2024. *What Is Authentication? Definition and Methods | Microsoft Security.* [Online]
Available at: https://www.microsoft.com/en-us/security/business/security-101/what-is-authentication
[Accessed 15 April 2024].

Okta, 2024. *5 Identity Attacks that Exploit Your Broken Authentication | Okta.* [Online]
Available at: https://www.okta.com/resources/whitepaper/5-identity-attacks-that-exploit-your-broken-authentication/
[Accessed 16 April 2024].

Port Swigger, 2024. *Authentication vulnerabilities | Web Security Academy.* [Online]
Available at: https://portswigger.net/web-security/authentication
[Accessed 13 April 2024].

Portswigger, 2024. *Burp Intruder attack types - PortSwigger.* [Online]
Available at: https://portswigger.net/burp/documentation/desktop/tools/intruder/configure-attack/attack-types#:~:text=Sniper,payloads%20in%20the%20payload%20set.
[Accessed 20 April 2024].

Portswigger, 2024. *Burp Repeater - PortSwigger.* [Online]
Available at: https://portswigger.net/burp/documentation/desktop/tools/repeater
[Accessed 25 April 2024].

Portswigger, 2024. *Getting started with Burp Intruder - PortSwigger.* [Online]
Available at: https://portswigger.net/burp/documentation/desktop/tools/intruder/getting-started#:~:text=Burp%20Intruder%20is%20a%20powerful,into%20predefined%20positions%20each%20time.
[Accessed 20 April 2024].

Poza, D., 2020. *What Is Broken Authentication?.* [Online] Available at: https://auth0.com/blog/what-is-broken-authentication/ [Accessed 15 April 2024].

Secure Flag, 2024. *Broken Authentication Vulnerability | SecureFlag Security Knowledge Base.* [Online] Available at: https://knowledge-base.secureflag.com/vulnerabilities/broken_authentication/broken_authentication_vulnerability.html [Accessed 15 April 2024].

Securelist, 2020. *Remote spring: the rise of RDP bruteforce attacks | Securelist.* [Online] Available at: https://securelist.com/remote-spring-the-rise-of-rdp-bruteforce-attacks/96820/ [Accessed 16 April 2024].

Tiwari, M., 2024. *What is Authentication?.* [Online] Available at: https://www.loginradius.com/blog/identity/what-is-authentication/ [Accessed 15 April 2024].