

AMAZON APPLICATION TESTING of

7 Vulnerabilities

Project report about cyber security

in

Pace institute of technology and sciences

BY

DUVVUR HAFIJA(CSE-C)3rd year

22kq1a05e2

Under the esteemed guidance of

Sk. Prem Nazeer

Certified Ethical Hacker Licensed Pentester

ABSTRACT

The project is about complete testing of application using Burp suite community edition web applications is paramount. The project, "Complete Application Testing with seven vulnerabilities," represents by me This project including SQL injection, Cross-Site Scripting (XSS), server side request forgery(SSRF),Brute force attack ,operating system,directory path traversal,authentication.

Employing a blend of manual inspection and automated tools, I identify vulnerabilities, assess their severity, and offer detailed recommendations for mitigation.My primary goal is to enhance the security of the selected application while also fostering a heightened awareness of web application security principles.

The project's outcomes, distilled within a comprehensive report, aim to serve as a beacon of knowledge for the broader community. By unveiling the intricacies of web application security testing and sharing best practices, we aspire to empower others to navigate the evolving landscape of cyber threats with confidence and expertise. In a digital world fraught with vulnerabilities, our dedication to securing web applications stands as a testament to our unwavering commitment to a safer online environment.

CONTENTS

Title

Abstract

Contents

Brute force attack-multi force at a time Vulnerability

Introduction

Methodology

Types of brute force attacks:

Application of brute force attack:

Mitigation:

Conclusion:

SQL(structured query language) injection vulnerability

Introduction

Methodology

Impact of SQL injection:

Detecting SQL injection vulnerabilities:

Union attack:

Examples of SQL injection:

Mitigation:

Conclusion:

CROSS-SITE SCRIPTING(XSS) vulnerability

Introduction

Methodology

Reflected XSS:

XSS contexts:

Preventing XSS attacks:

Exploting XSS vulnerabilities:

Mitigation:

Conclusion:

SSSRF(server-side request forgery) vulnerability

Introduction

Methodology

Impact of SSRF:

Common SSRF attacks:

Finding hidden attack surface for SSRF:

Mitigation:

Conclusion:

OS(operating system)Command Injection Vulnerability

Introduction

Methodology

Injecting OS commands:

Blind command injection vulnerabilities:

Preventing:

Mitigation:

Conclusion:

Directory of Path Traversal Vulnerability

Introduction

Methodology

Reading arbitrary files via path traversal:

Common obstacles:

Preventing:

Mitigation:

Conclusion:

Authentication Vulnerability

Introduction

Methodology

How Vulnerability arise?

Impact of vulnerable authentication

Vulnerability in multi-factor authentication:

Vulnerability in other authentication mechanisms:

Securing your authentication mechanisms:

Mitigation:

Conclusion:

01-Brute Force attacks

Identification of Brute force attacks -multi force at a time

Introduction

This is about brute force attacks at multi force at a time ,it is a hacking method use trail and error to crack passwords login credentials and encryption keys. It for gaining unauthorized access to individual account and networks.

Methodology

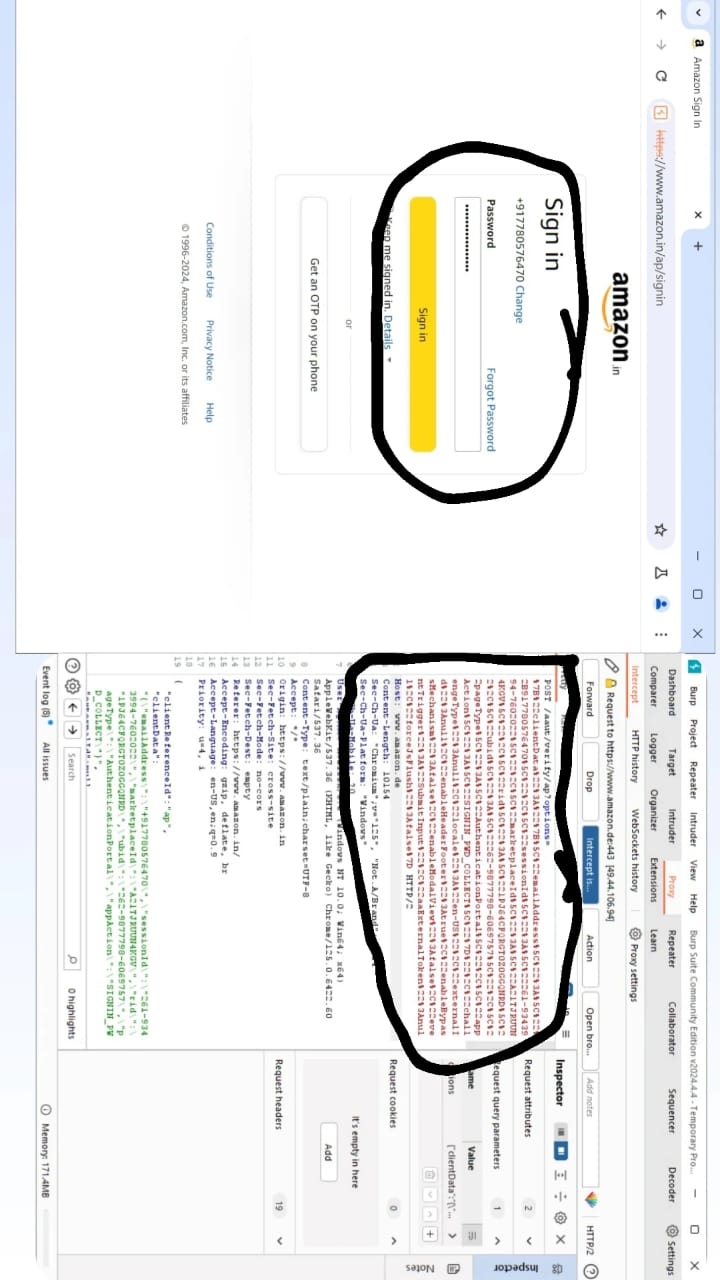
To crack personal detials of login credentials,in brute force attack we will find otp&password.

USING Password:

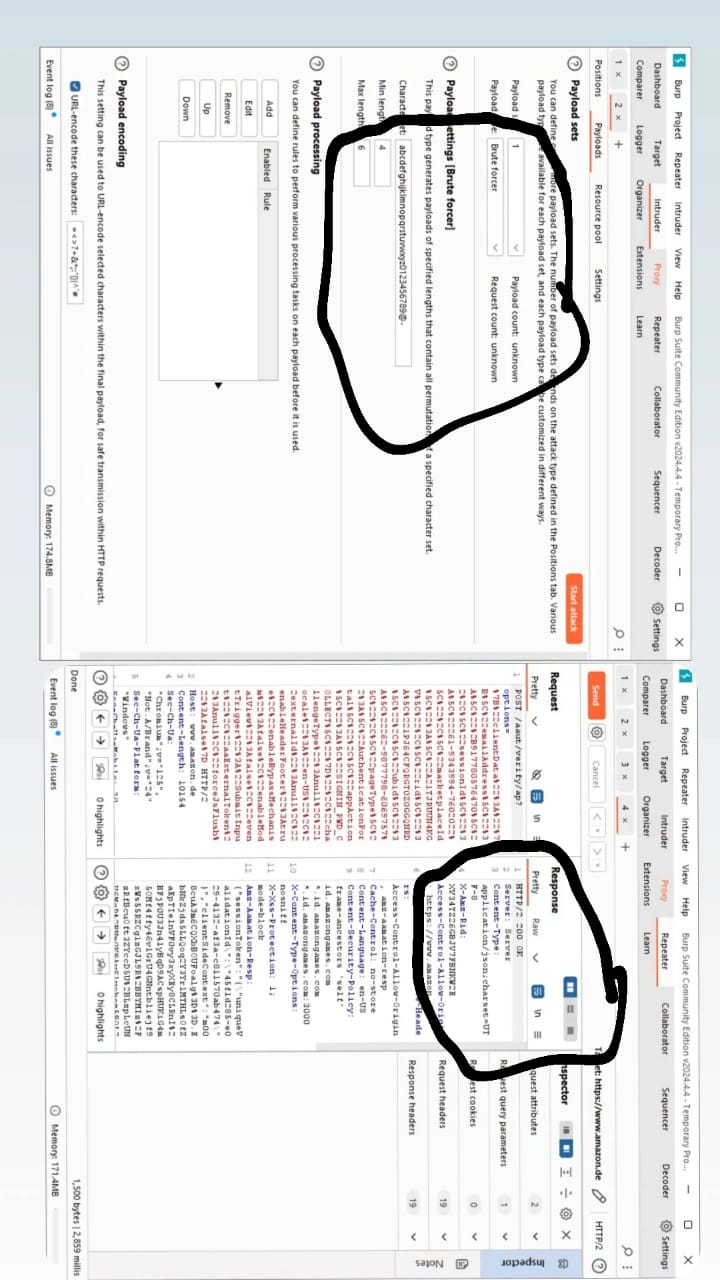
Here we take one application to check brute force attack. The application is AMAZON-shopping application .

Step 1:

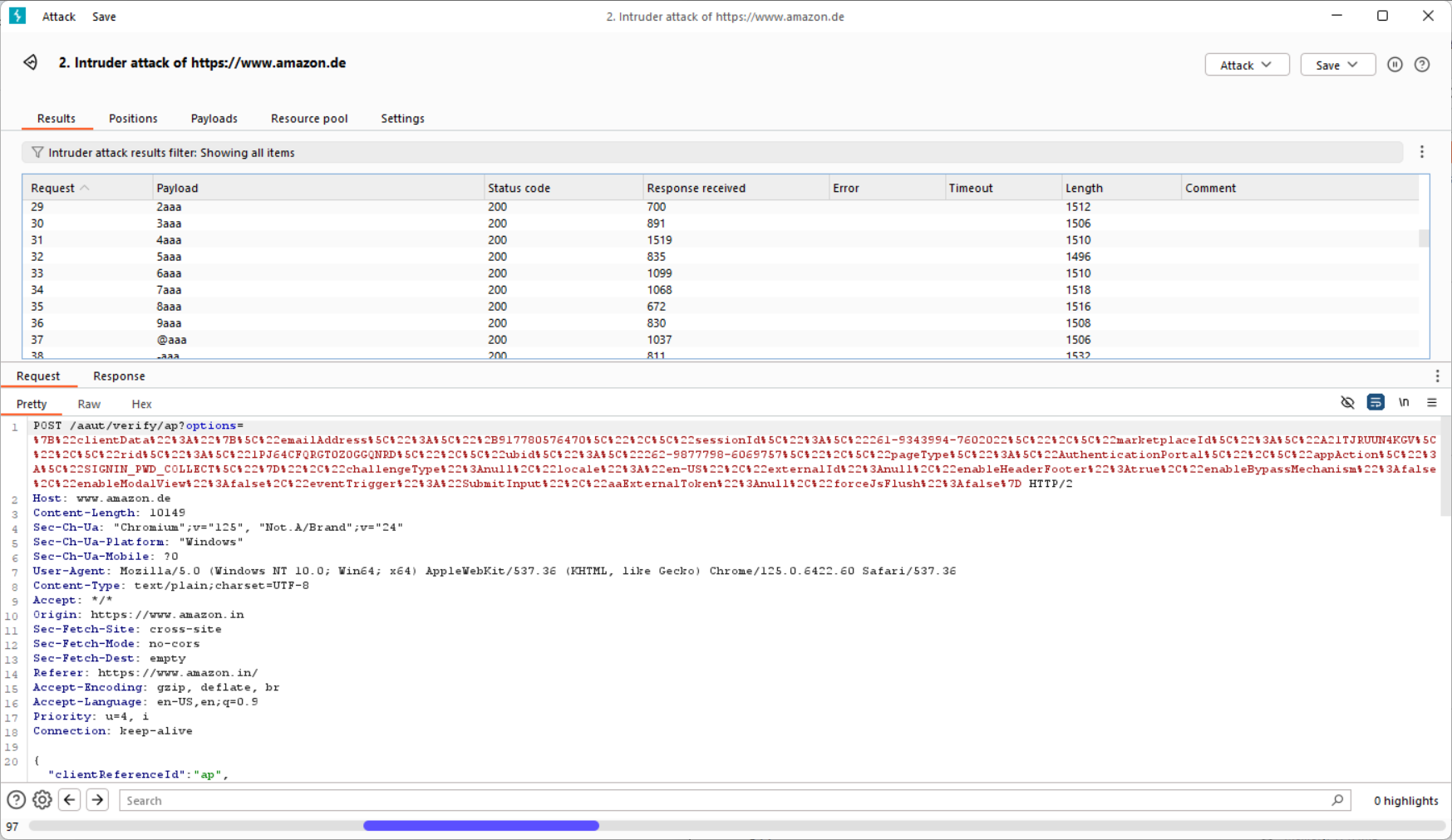
Firstly we have to login in amazon application next we have to login with our gmail(or)phone number&give a unreal password ,before sign in we have to on intercept in burp and next sign.



Step 2:

When we on the intercept the code will appear as upper fig ,next we have to send the code to the repeater to see response.the response will seen as 200 Ok. Next we have to send the code to the intruder as shown in fig,in intruder we have to add positions.next set thepayload type:brute forcer,give min length:4,max length:6 at last start attack.

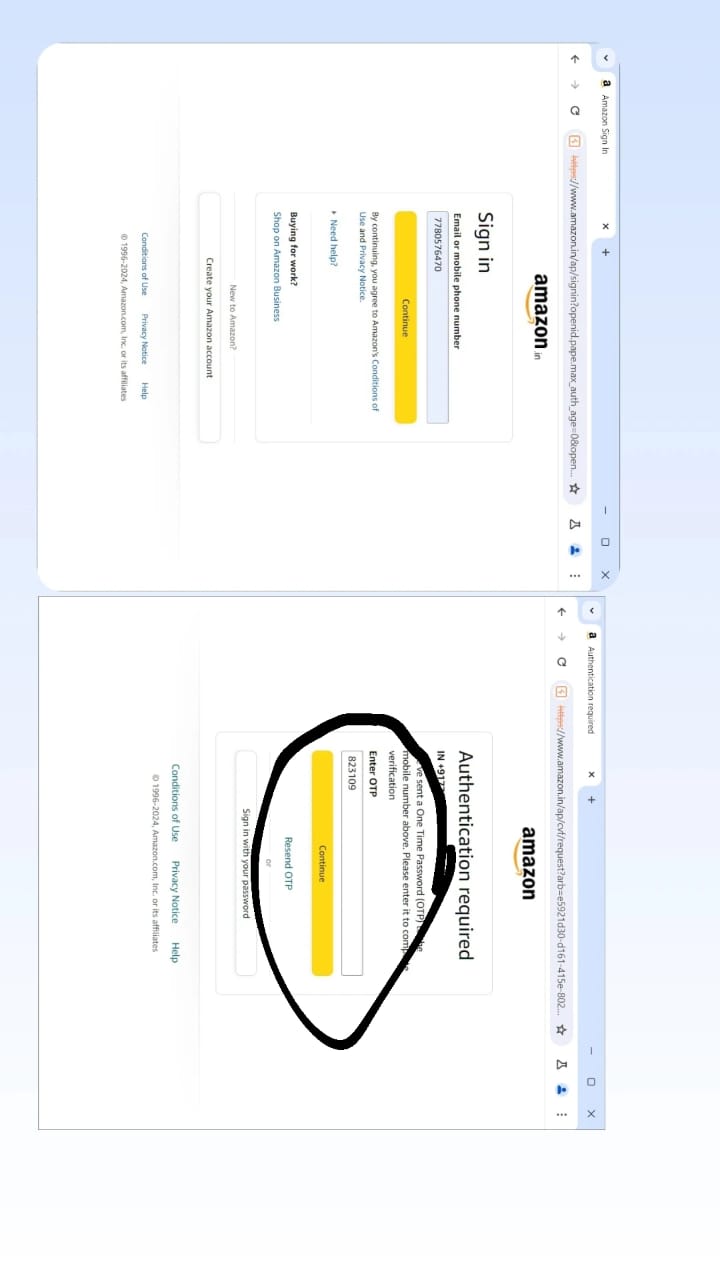
Step 3:

Here the result when we start attack,it takes lot of time to complete this.

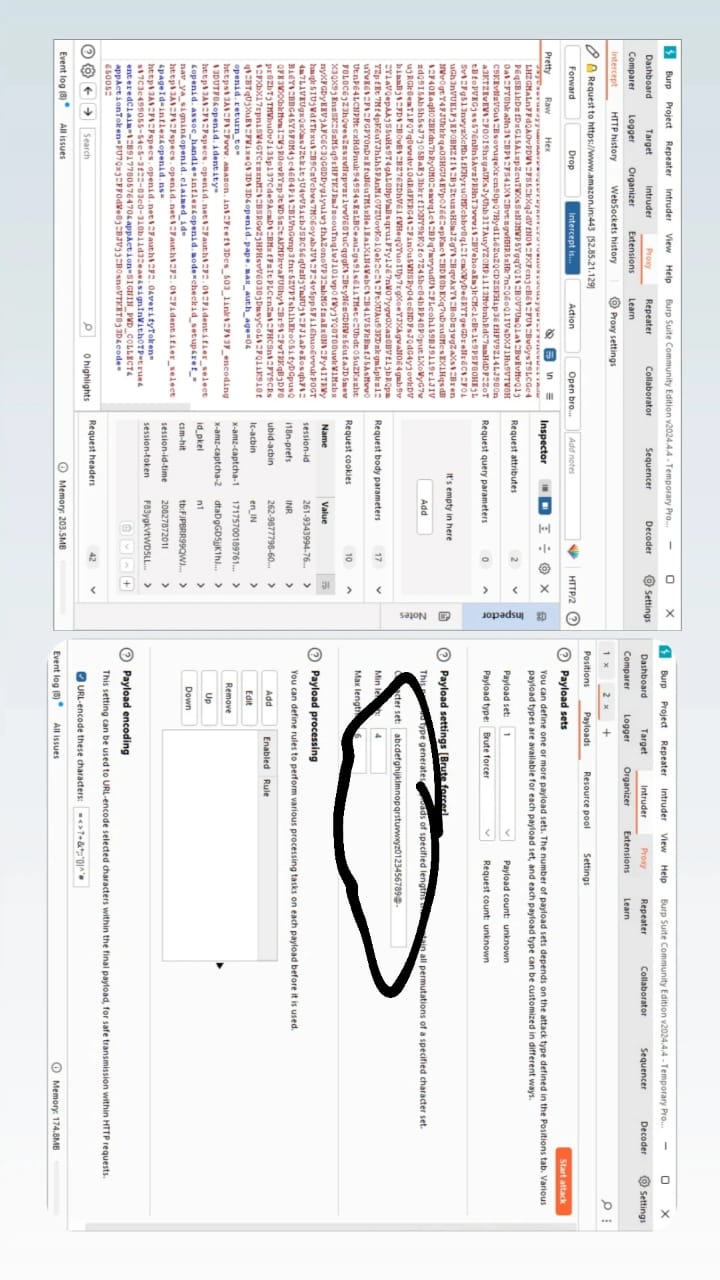
Using OTP:

Step 1:

First we have to login in amazon application .next we have to login with our gmail (or)phone number&it generates the otp to yourmobile and enter unreal otp to it before sign in on the intercept.

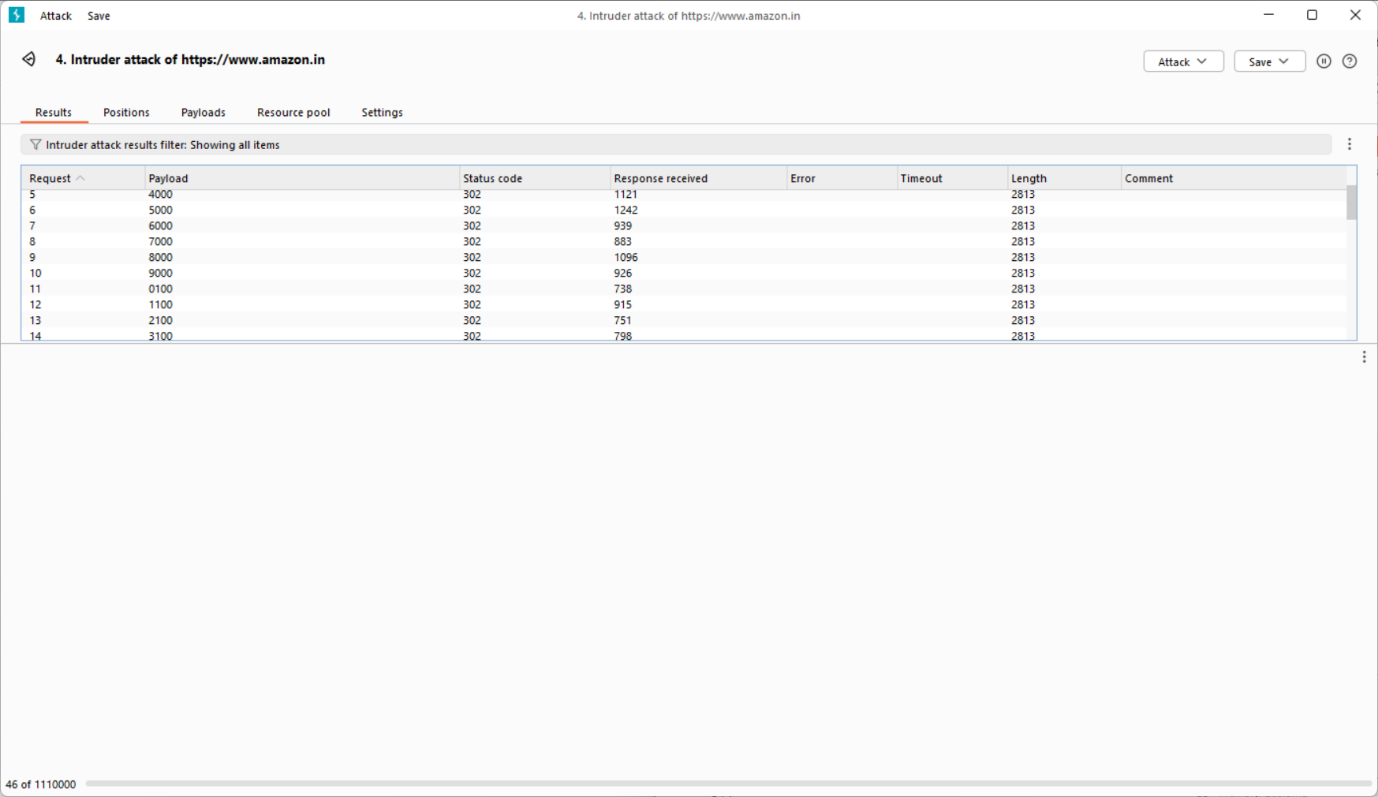


Step 2:

When we on the intercept the code will appear ,then the code send to the repeater to see response,the response is 200 ok its correct &next we have to send the code to the intruder ,set the position & next set the payload type as brute forcer,character set :it not at all as the password we have to keep only numbers and delete alphabets ,at last start attack.

Step 3:

When we start attack ,the result is shown below.it take lot of time so we have to discard it.



Types of brute force attacks:

1.Simple brute force

2.Dictionary

3.Hybrid brute force

4.Reverse brute force

5.credential stuffing

1.Simple brute force attack:

A simple brute force attack occurs when a hacker attempts login credentials manually without using any software

.

2.Dictionary attack:

A Dictionary attack is a basic form of brute force hacking in which the attacker selects atarget ,then tests possible password that individuals username.

3.Hybrid brute force attack:

A hybrid brute force attack is when a hacker combines a dictionary attack method with a simple brute force attack method with a simple brute force attack.

4.Reverse brute force attack:

A reverse brute force attack sees an attacker begin the process with a known password,which is typically discovered through a network breach.

5.Credential stuffing attack:

Credential stuffing preys on users weak password efferuettes. Attackers collects username&password combination they have stolen,which they then test on other websites.

Applications:

Sorting, Searching, String matching, cryptography, puzzles such as sudoku or the travelling sales person problem.

1.Get:

In this method is used to retrive data from a server.

2.Post:

This method is typically used to submit data to be processed by the server.

3.Put:

This method is used to createor replace a resource on the server with the request payload

4.Patch:

This method is used to partially update an existing resource on a server.

5.Delete:

This method is used to delete a specific resource from the server.

Mitigation:

1.We want to use strong passwords ,brute force relies on weak passwords.

2.Using CAPTCHAs verification some security will provides to our account.

3.Limit the login attempts,it may cause blocking of your account.

4.Use two factor authentication,with this the another layer security.

5.Restricting IP access will protect sensitive login and admin pages from brute force attacks.

Conclusion:

The conclusion to a brute force attack largely depends on the context. If successful, it means the attacker has gained unauthorized access. In our amazon application we did brute force attack,in that it will not shows any output because we use normal burp suite community edition.

02-SQL INJECTION

(SQL injection vulnerability in where clause allowing retrieval of hidden data)

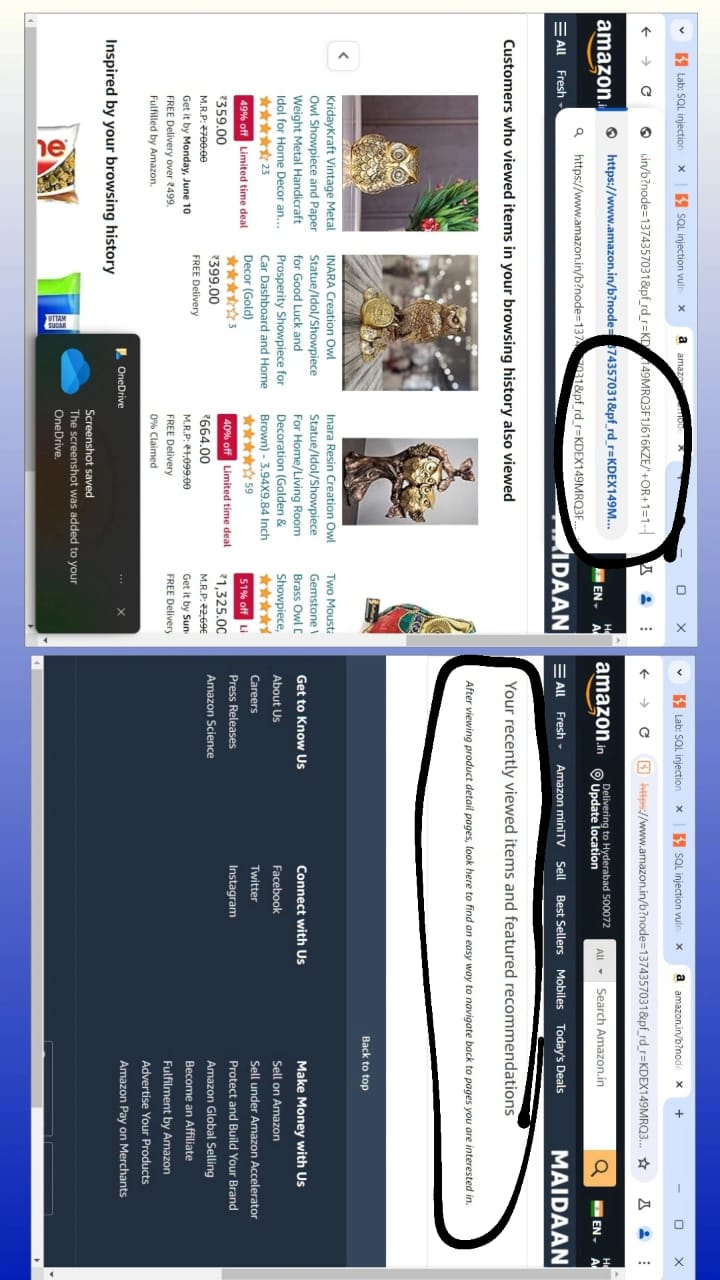
Introduction:

The sql(structured query language*)* is a common attack vector that uses malicious SQL code for backend database manipulation to access information that was not intended to be displayed. This information may include any number of items, including sensitive company data, user lists or private customer details.we performing sql injection on amazon application.

Methodology:

The attacker runs a malicious SQL statement on the database.Since SQL injection can be used to alter database values, wipe out the entire database,and steal database content,it poses a serious security risk.

First we have to login in amazon page,next select a category in amazon,give “(‘+or+1=1--)” this on url as shown in below fig.



Impact of SQL injection:

A successful sql injection attack can result in unauthorized access to sensitive data,such as :passwords,credit card details,personal user information.The deletion of entire tables and in certain cases.

Detecting SQL injection vulnerabilities:

You can detect SQL injection manually using a systematic set of tests against every entry point in the application.To do this,you would typically submit:The single quote character’and look for errors or other anomalies.

Union attacks:

When an application is vulnerable to SQL injection,and the results of the query are returned within the applications responses,you can use the UNION Keyword to retrive data from other tables within the database.This is commonly known as a SQL injection UNION attack.

Examples of SQL injection :

There are lots of SQL injection vulnerabilities,attacks,techniques,that occur in different situations.Some common SQL injection examples include:

RETRIEVING HIDDEN DATA

SUBVERTING APPLICATION LOGIC

UNION ATTACKS

BLIND SQL INJECTION

Mitigation:

Another effective coding practice to mitigate the risk of SQL injection is parameterized statements.Parameterized statements separate user inputs from the SQL query,eliminating the need for manual escaping.

Conclusion :

When we test sql injection vulnerability in where clause allowing retrival of hidden data with using amazon application ,the output will be not seem because we did manual testing.

03-Cross site scripting(XSS)

vulnerability:

(Reflected XSS into HTML context with nothing encoded)

Introduction:

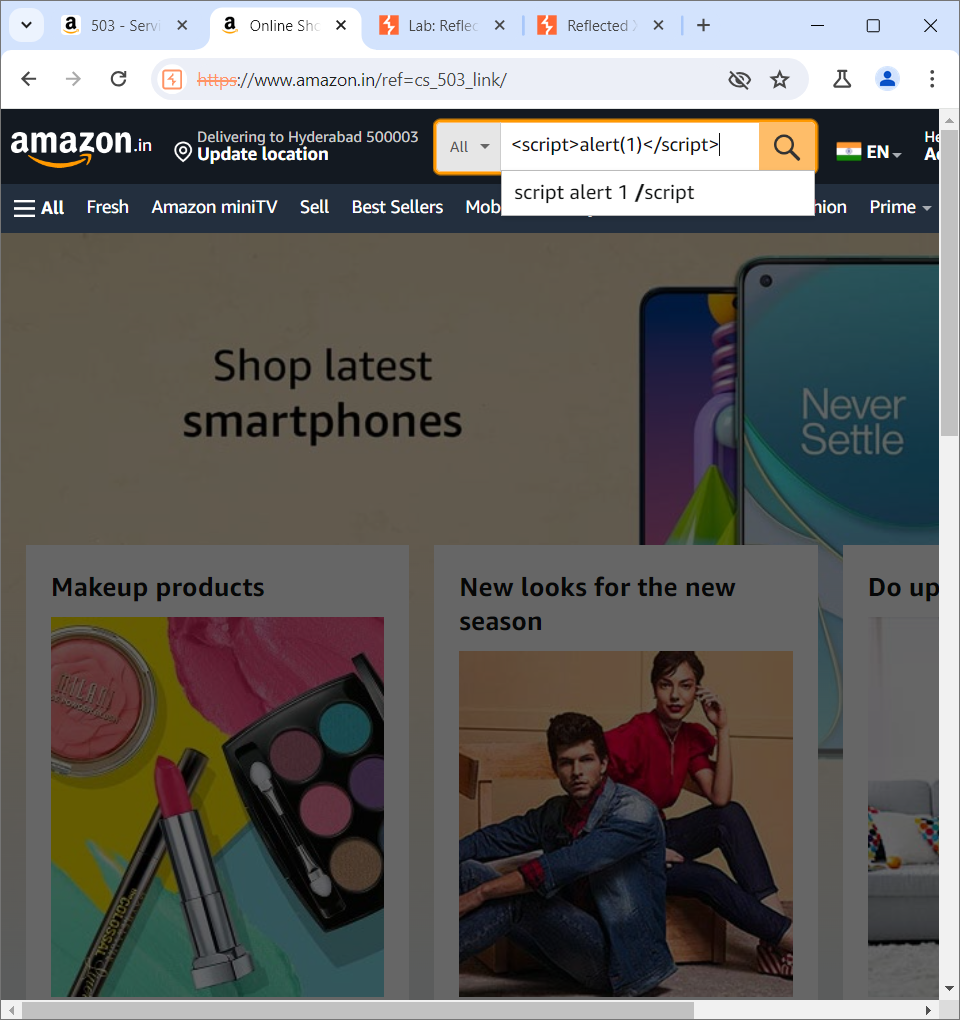
Cross-site scripting also known as XSS ,it is aweb security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application.It allows an attacker to circumvent the same origin policy,which is designed to segregate different websites from each other.cross-site scripting works by manipulating a vunerable web site so that it returns malicious javascript to users.

Methodology:

To carry out a cross-site scripting attack.an attacker injects a malicious script into user-provided input.Attackers can also carry out an attack by modifying a request.If the web app is vulnerable to XSS attacks,the user-supplied input executes as code.

Here we use alert function in cross site scripting &next we want to login amazon account after in search bar we have to give this function

(<script>alert(1)</script>).



Reflected XSS:

The reflected cross site scripting (or XSS) arises when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way.

Cross site scripting contexts:

When testing for reflected and stored XSS,a key task to identify the XSS context.The location within the response where attacker-controllable data appears.Any input validation or other processing that is being performed on that is being performed on that data by the application.

When the XSS context is text between HTMLtags.you need to introduce some new HTML tags designed to trigger exection of javascript.

Preventing XSS:

The general principle for preventing cross site scripting vulnerabilities and ways of using various common technologies for protectig against XSS attacks.Cross-site scripting can generally be achieved via two layers of defense:

\*Encode data on output

\*validate input on arrival

Explloting Cross-site vulnerabilities:

The traditional way to prove that you have found a cross scripting vulnerability is to create a popup using the alert() function.This isn’t because XSS has anything to do with popups,it’s simply a way to prove that you can execute arbitrary Javascript on a given domain.you might notice some people using alert(document.domain).This is a way of making it explict which domain the javascript is excuting on.

Mitigation:

We should sanitize the user input before sorting or displaying it.Sanitize input:use a library like XSS to sanitize the input, ensuring that any HTML tags are rendered harmless.

Conclusion:

When we test Cross site scripting (reflectedXSS into HTML context with nothing encoded) with amazon application,the output will not seem because we did manual testing.

04-SSRF(server-side request

forgery) Vulnerability

(Basic SSRF against the local server)

Introduction:

Server side request forgery is a web security vulnerability that allows an attacker to cause the server side application to make requests to an unintended location.

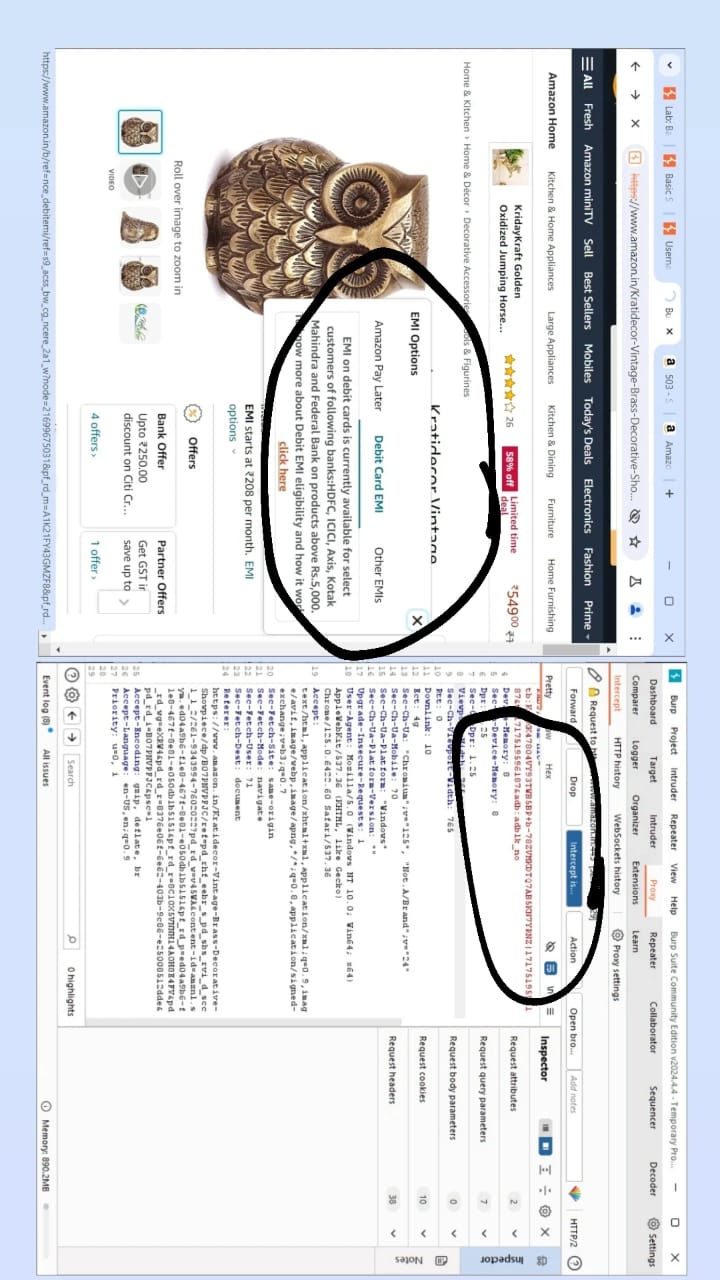
In a typical SSRF attack might cause the server to make a connection to internal-only services within the organization infrastructure.In other cases,they may be able to force the server to connect to arbitary external systems.This could leak sensitive data,such as authorization credentials.

Methodology:

In a server SSRF attack, attackers exploit a process in which a browser or other client system directly accesses a URL on the server. The attacker will replace the original URL with another, typically using the IP 127.0. 0.1 or the hostname “localhost”, which point to the local file system on the server.

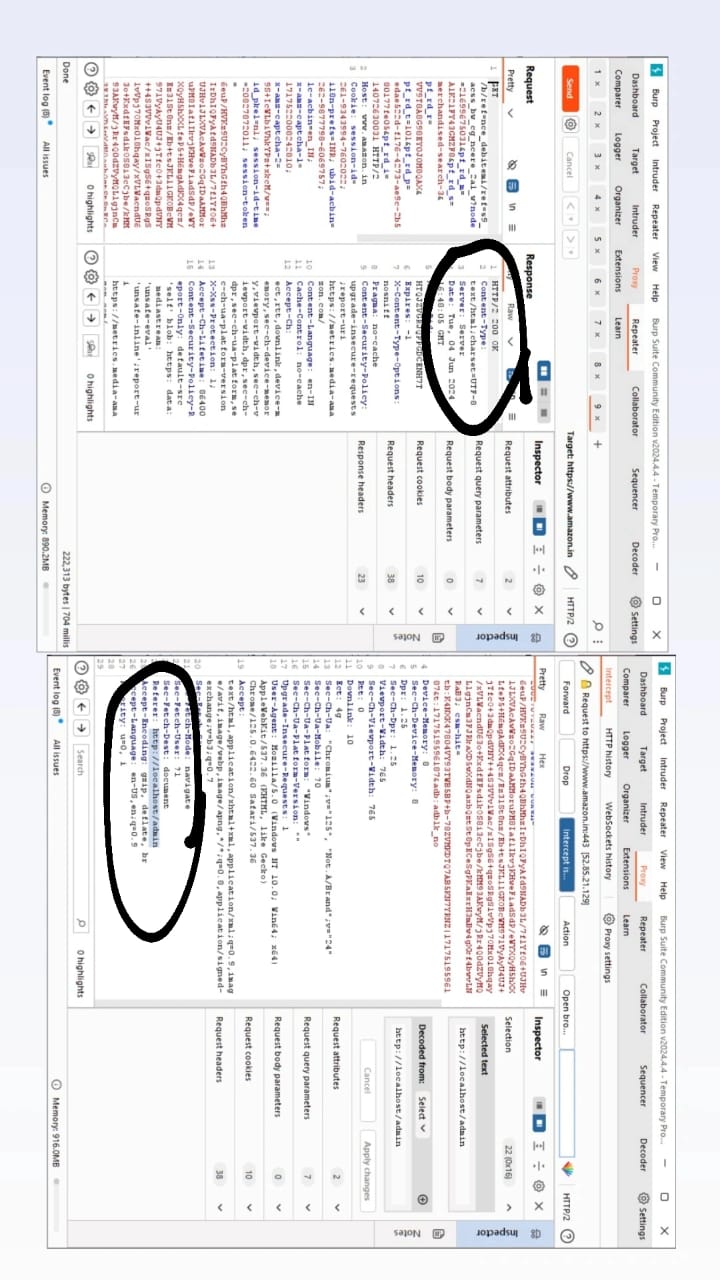
Step 1:

First we have to login amazon ,next select a product and click on product stock details before clicking it in the burp suite we have to on the intercept.



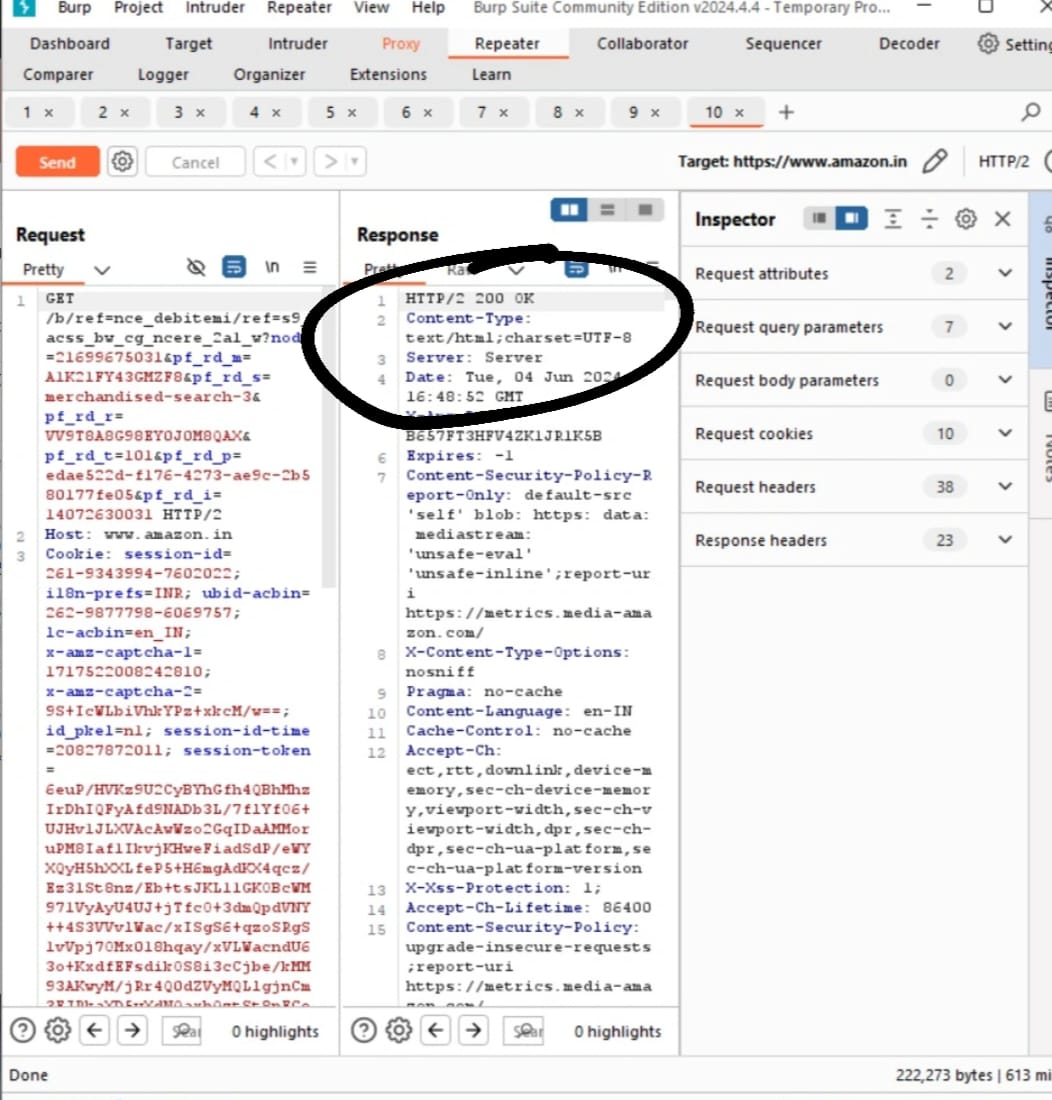
Step 2:

Next we have to send the code to request and the response is 200 ok,next forward it.Again change the stock details next we have to replace url into <http://localhost/admin>



Step 3:

Next we have to send the code to the repeater to see response ok then forward it.

`

Impact of SSRF:

A successful SSRF attack can often result in authorized actions or access to data within the organization.This can be in the vulnerable application,or on other back-end systems that the application can communicate with.In some situations,the SSRF vulnerability might allow an attacker to perform arbitrary command execution.

Common SSRF attack:

A Server-Side Request Forgery (SSRF) attack involves an attacker abusing server functionality to access or modify resources. The attacker targets an application that supports data imports from URLs or allows them to read data from URLs.

Finding hidden attack surface for SSRF:

Many Server-side request forgery vulnerabilities are easy to find,because the application’s normal traffic involves request parameters containing full URLs.Other examples of SSRF are harder to locate.

Mitigation:

SSRF can be mitigated through application layer controls—the application can check a target address is allowed before creating a connection. Note that it is not enough just to verify the address and establish a connection. This approach is still vulnerable to “time-of-check” and “time-of-use” vulnerabilities.

Conclusion:

When we check SSRF vulnerability with amazon application we see some codes and didn’t see any result in amazon application because we use normal burp suite community edition tool.

05-OS command injection

Vulnerability

(os command injection,simple case)

Introduction:

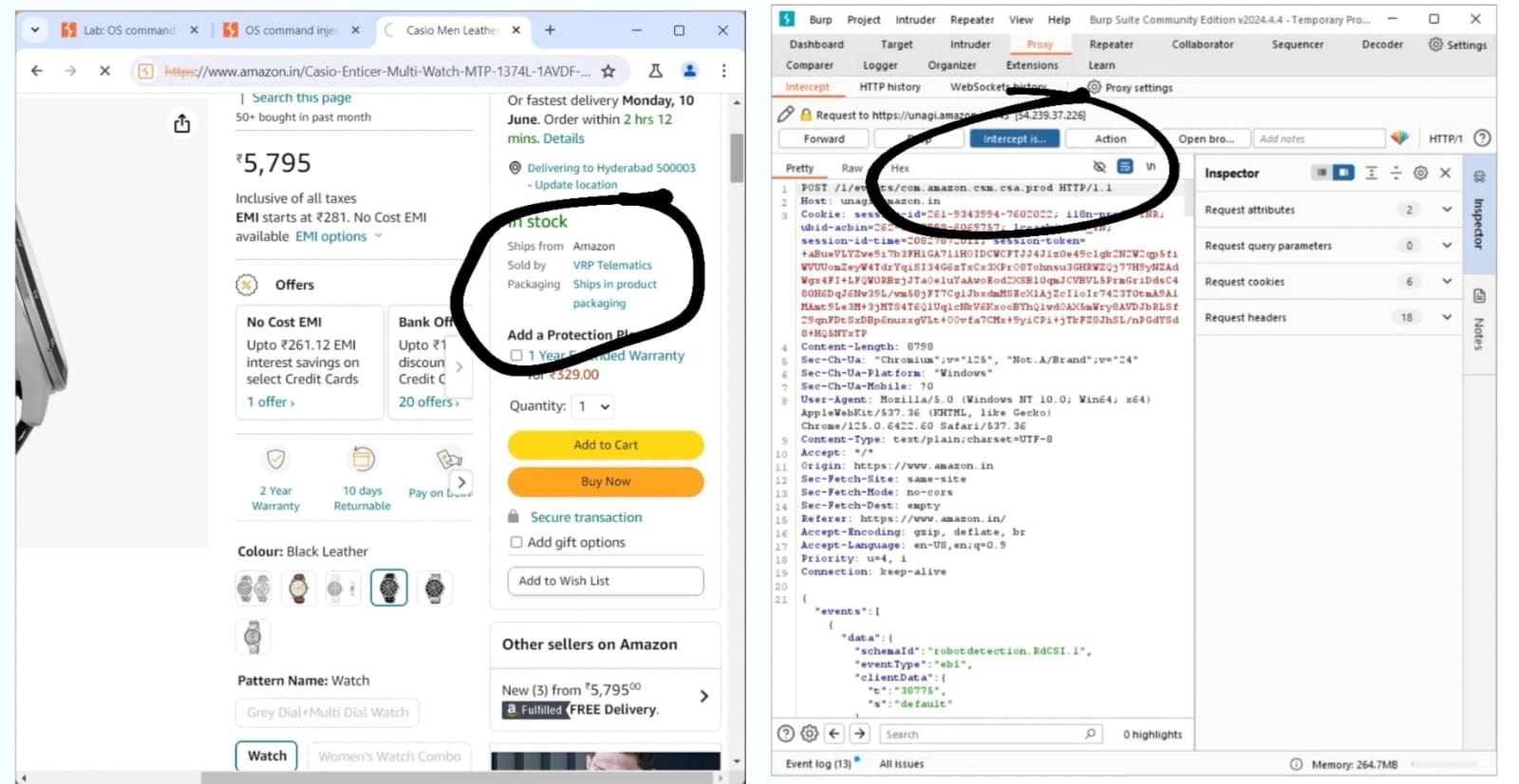
OS command injection is also known as shell injection.It allows an attacker to execute operating system(OS) commands on the server that is runnning an application,and typically fully compromise the application and its data.Often,an attacker can levarage an OS command injection vulnerability to compromise other parts of the hosting infrastructure,and exploit trust relationships to pivot the attack to other systems within the organization.

Methodology:

OS command injection is a technique used via a web interface in order to execute OS commands on a web server. The user supplies operating system commands through a web interface in order to execute OS commands. Any web interface that is not properly sanitized is subject to this exploit.

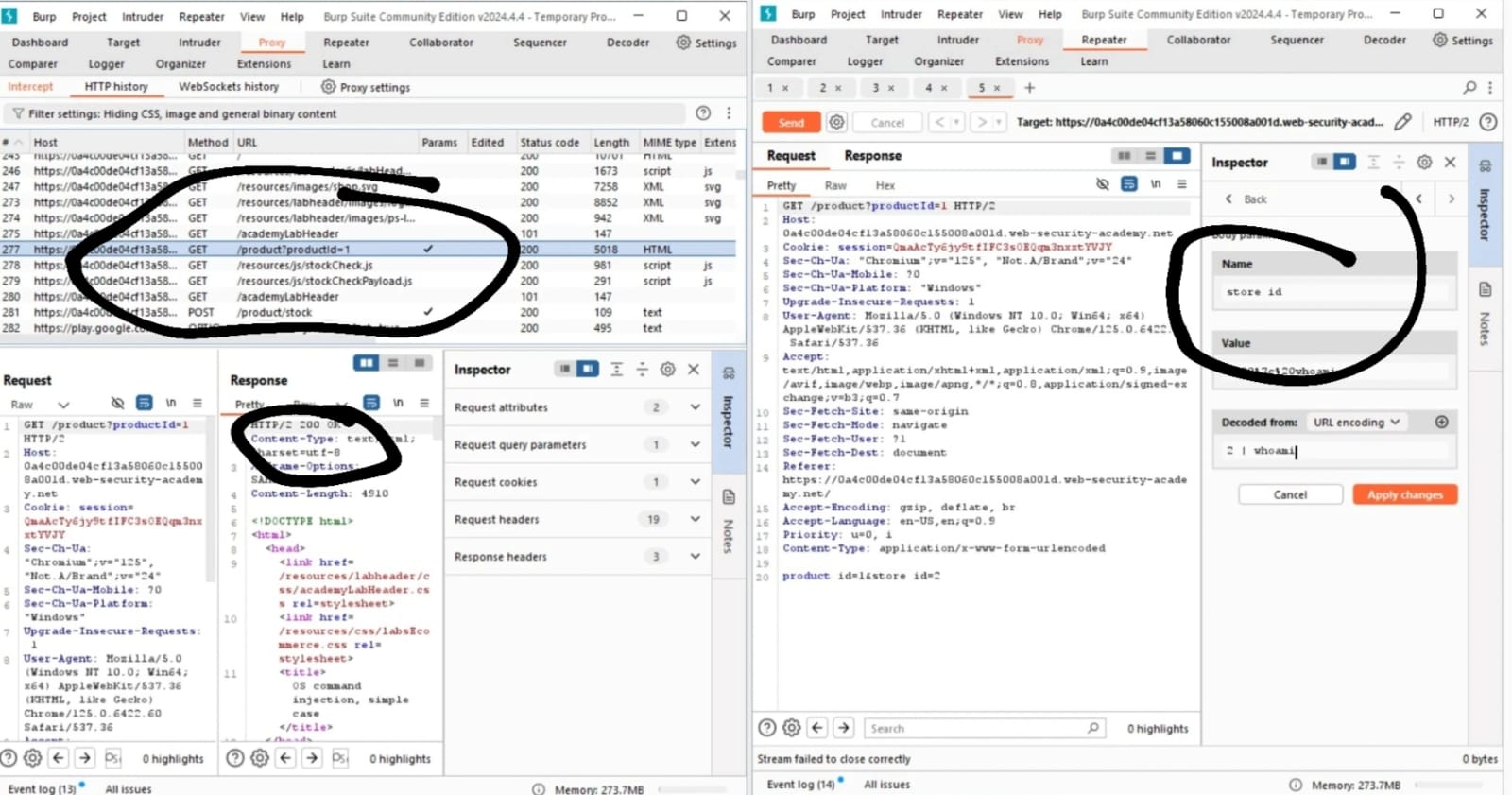
Step 1:

First we have to select an product in amazon application.then on the intercept in seconds check the stock.



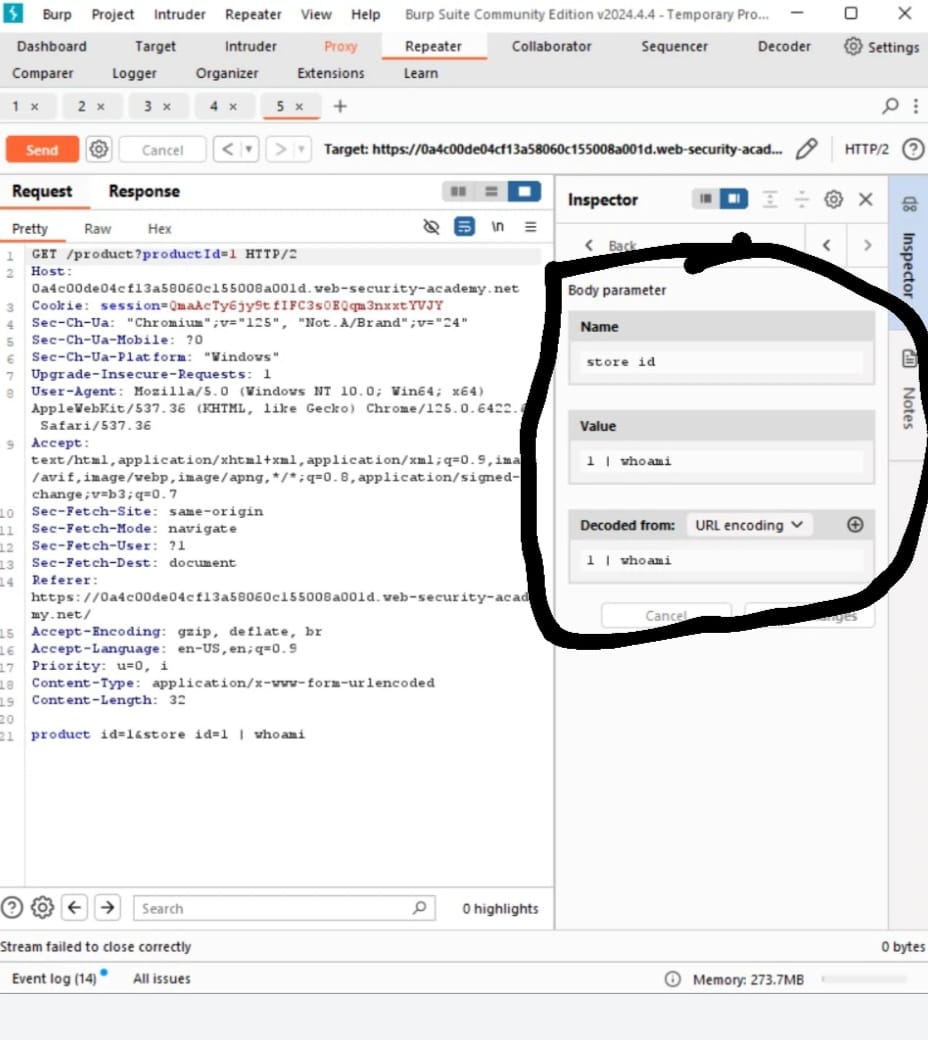
Step 2:

Next we have to go through HTTP history in that we have to check the product id which have tick mark to it then it response will 200 ok.next we have to go to body parameters to modify store id.



Step 3:

Next we have to add in store id decoded and value as”1 | whoami”



Step 4:

Forward it and off the intercept.

Blind command injection vulnerabilities:

Many instances of OS command injection are blind vulnerabilities.This means that the application does not return the output from the command within its HTTP response.Blind vulnerabilities can still be exploited,but different techniques are required.

Preventing:

The most effective way to prevent OS command injection vulnerabilities is to never call out to OS commands from application-layer code.In almost all cases,there are different ways to implement the required functionality using safer platform APIs

Mitigation:

\*Validating against a whitelist of permitted values.

\*Validating that the input is a number.

\*Validating that the input contains only alphanumeric characters, no other syntax or whitespace.

Conclusion:

We take amazon application to test the OS command injection vulnerability,we did the steps,in our app it doesnot seem any result because we use normal burp suite community edition tool.

06-Directory of Path Traversal

Vulnerability

(File path traversal,simple case)

Introduction:

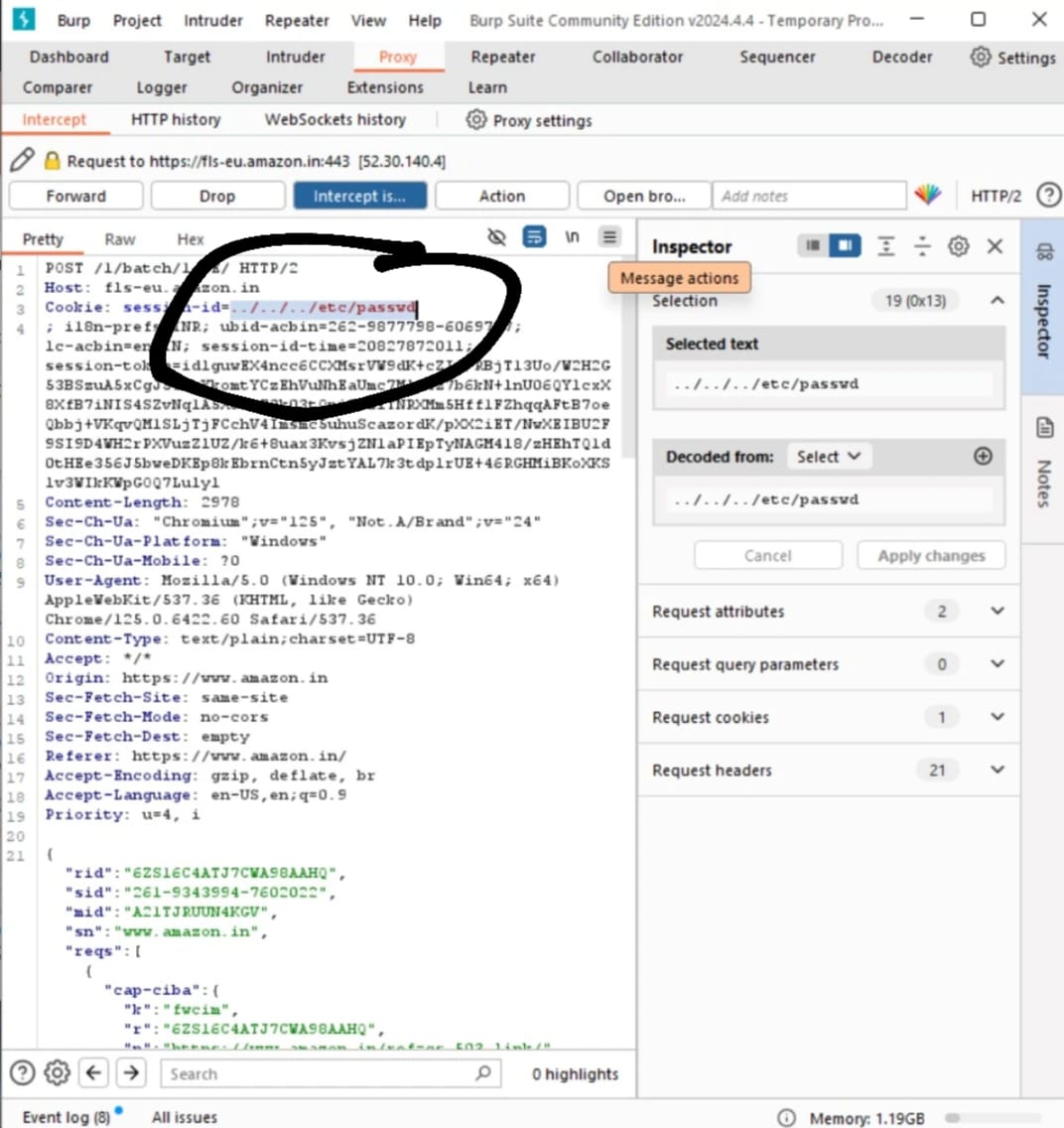
Path traversal is known as directory traversal.These vulnerabilities enable an attacker to read arbitrary files on the server that is running an application.This might include:Application code and data,Credentials for back-end systems,Sensitive operating system files.

Methodology:

Validate the user input before processing it. Ideally, compare the user input with a whitelist of permitted values.After validating the supplied input, append the input to the base directory and use a platform filesystem API to canonicalize the path.

Step 1:

We have to login amazon and give right click on image select open in new tab,before it on the intercept.next change file name as “../../../etc/passwd”and next check the response and forward it



Reading arbitrary files via path traversal:

A path traversal vulnerability allows an attacker to access files on your web server to which they should not have access.They do this by tricking either the web server or the web application running on it into returning files that exist outside of the web root folder.

Common obstacles:

Many applications that place user input into file paths implement defenses against path traversal attacks.These can often be by passed.If an application strips or blocks directory traversal sequences from the user-supplied filename,it might be possible to bypass the defnse using a variety of techniques.

Preventing:

To prevent path traversal your web server,update your web server and operating system to the latest versions available.This vulnerability has been known for a while,and it is likely your web servers latest version is not vulnerable

Mitigation:

To protect against path traversal attacks, it is important to follow best practices for web application security. This includes implementing input validation and sanitation, using file system permissions to restrict access to sensitive files and directories, and monitoring web server logs for suspicious activity.

Conclusion:

We did path traversal with amazon application it doesnot seem any output in amazon application because we use normal burp suite community edition tool.

07-Authentication

(Username enumeration via different responses)

Introduction:

Authentication is the process of verifying the identity of a user or client.Websites are potentially exposed to anyone who is connected to the internet.This makes robust

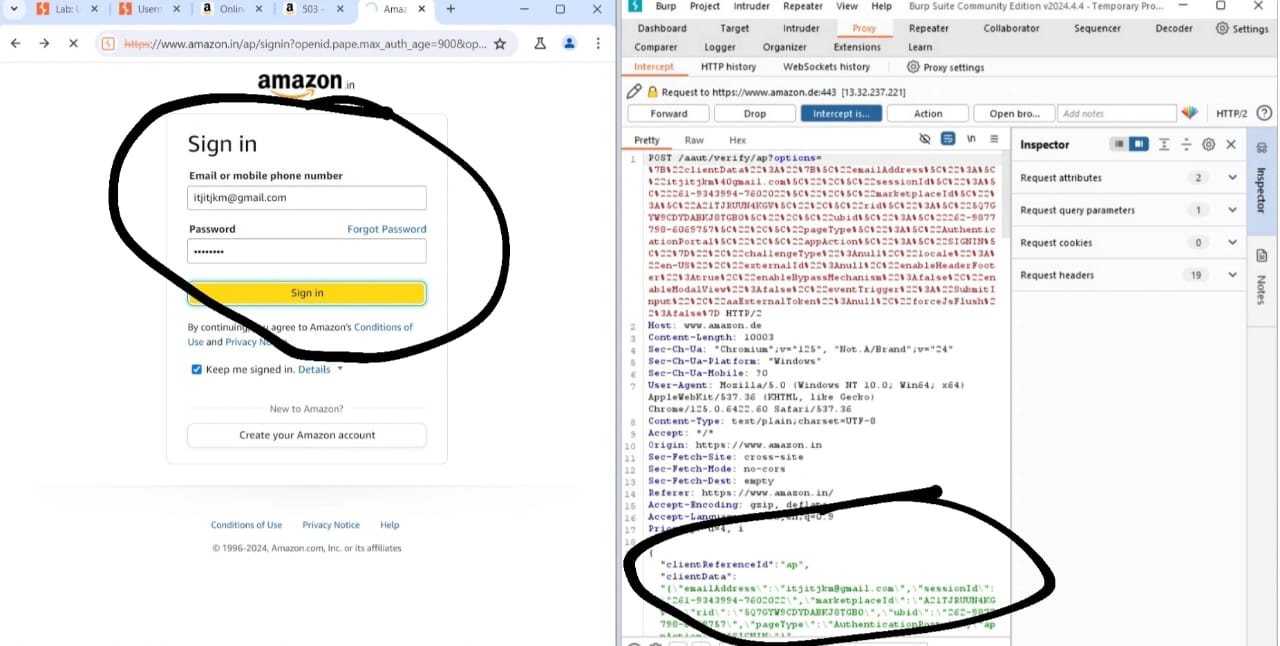
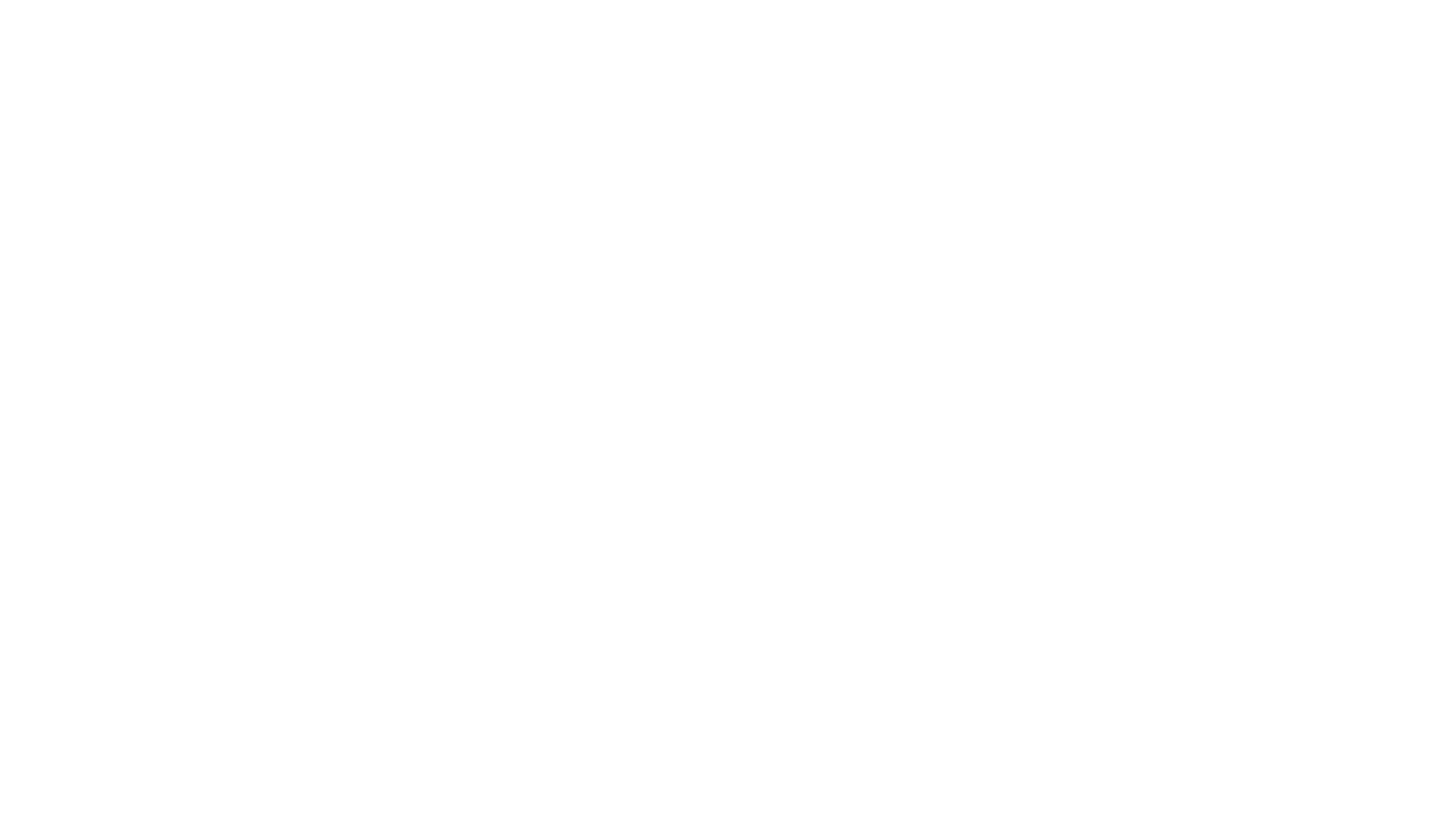
Authentication mechanisms integral to effective web security.

Methodology:

Most vulnerabilities in authentication mechanisms occur in one of two ways: The authentication mechanisms are weak because they fail to adequately protect against brute-force attacks. Logic flaws or poor coding in the implementation allow the authentication mechanisms to be bypassed entirely by an attacker.

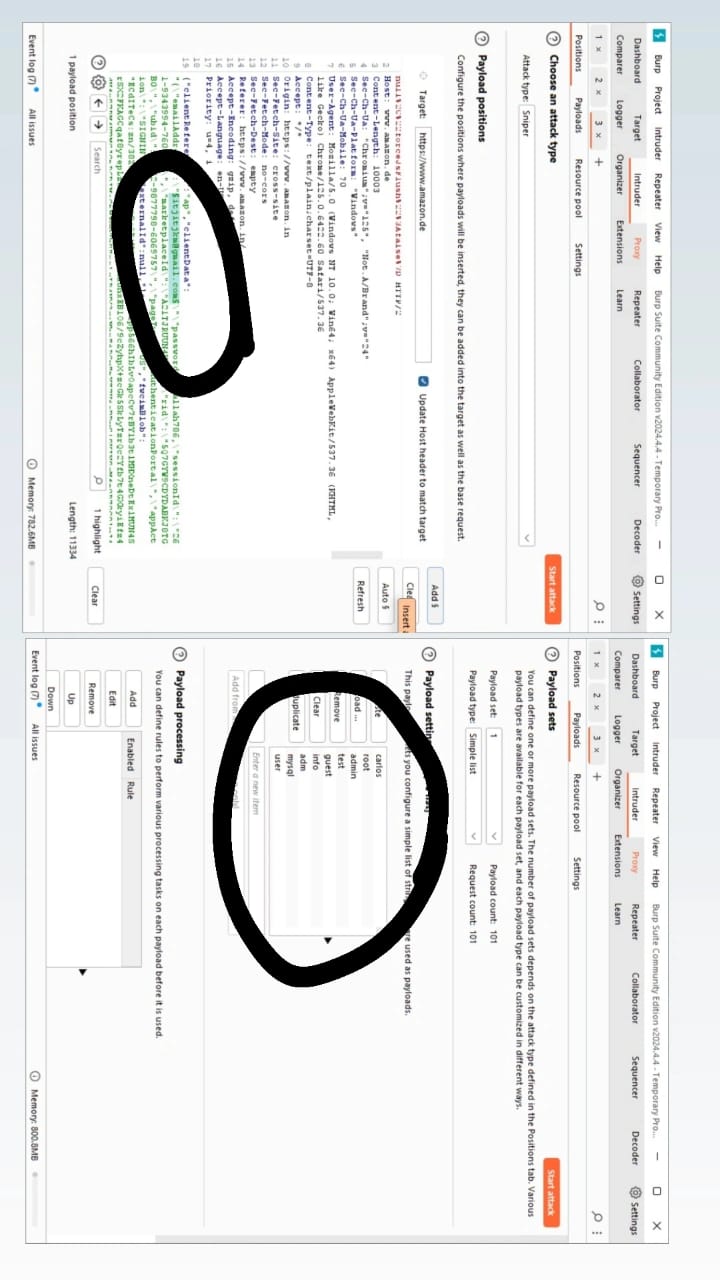
Step 1:

In this username enumeration via different responses.we find usernames and passwords.so we have to give fake username and password.



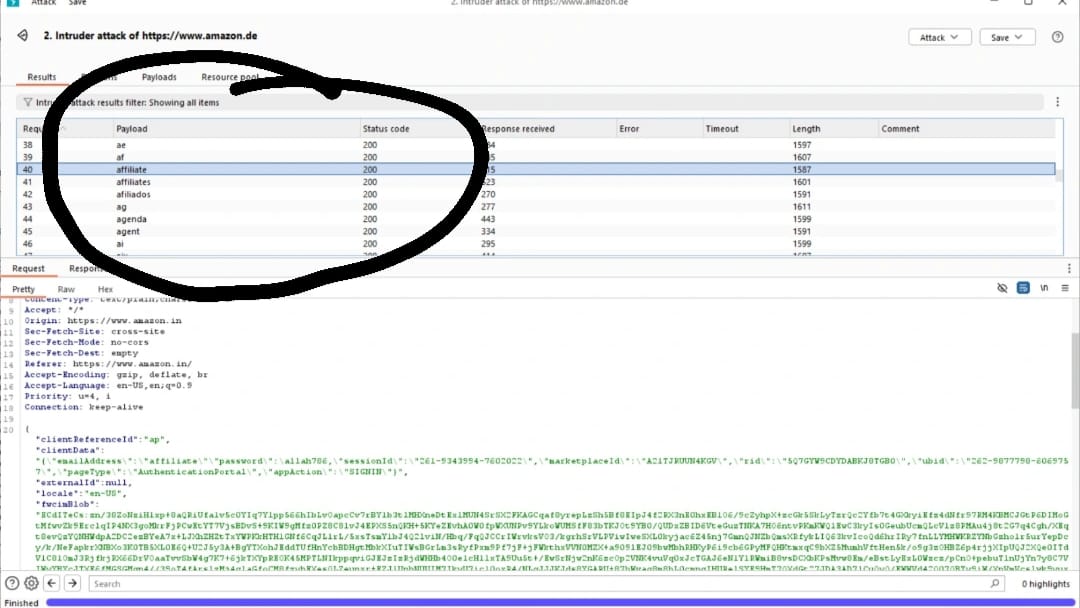
Step 2:

First we have to find username so we have to send this to intruder to set position to the username and set the payload type simple list then next paste all ur username and start attack.



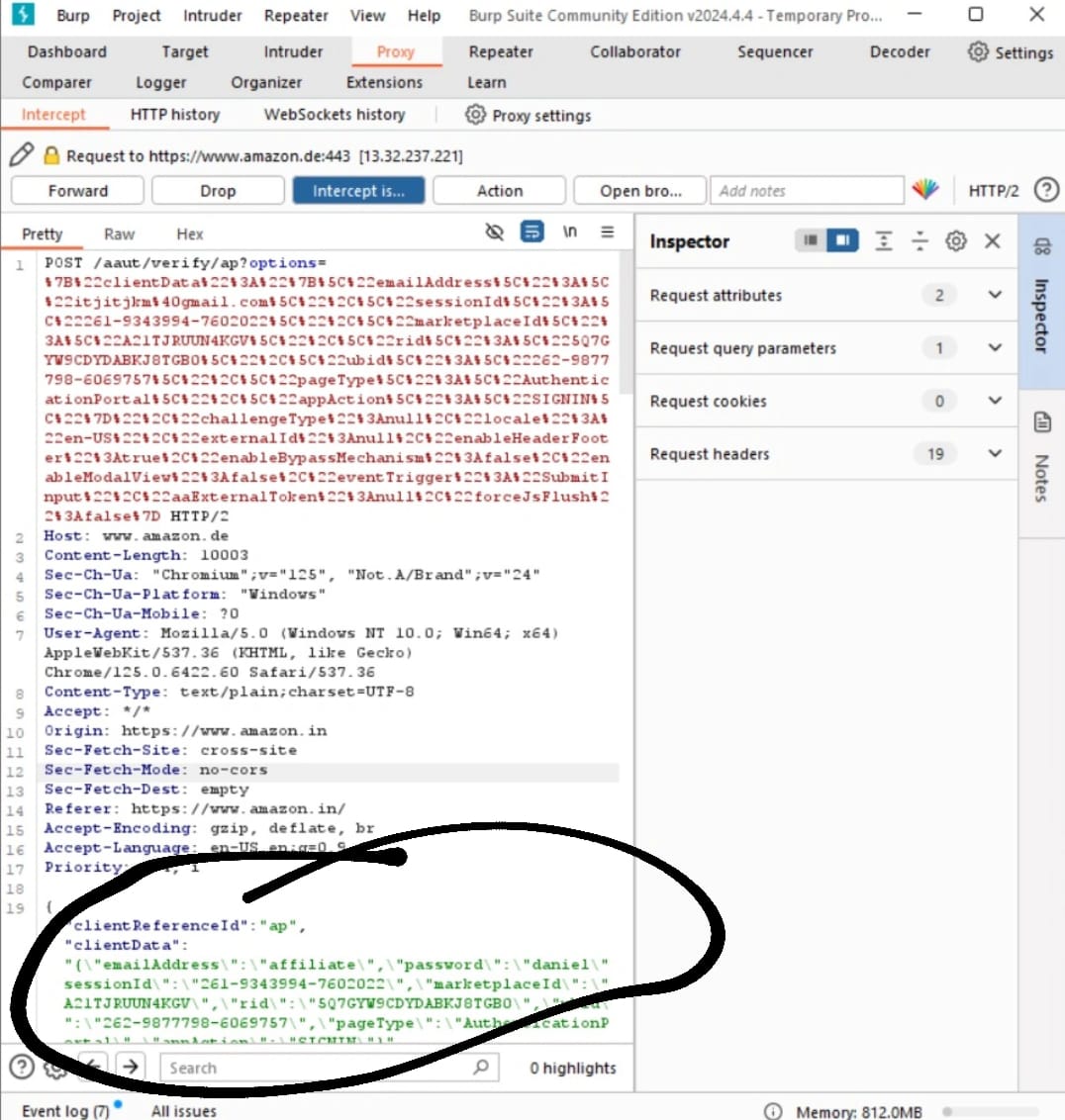
Step 3:

The same as the password enumaration process.then the result is …..



Step 4:

The username is affilate and password ad password is daniel



Step 5:

Forward it and off the intercept.

How vulnerability a rise?

Most Vulnerability in authentication mechanisms occur in one of two ways:

\*The authentication mechanisms are weak because they fail to adequately protect

Aginst brute\_force attacks.

\*Logic flaws or poor coding in the implementation allow the authentication mechanisms to be bypassed entirely by an attacker.This is sometimes called”broken authentication”.

Impact of vulnerable authentication:

The impact of authentication vulnerabilities can be severe.If an attacker bypasses authnetication or brute-forces their way into another users account,they have access to all the data and functionality that the compromised account has.If they are able to compromise a high privileged account,such as system adminstrator,they could take full control over the entire application and potentially gain access to internal infrastructure.

Vulnerability in multi-factor authentication:

In this section,we will look at some of the vulnerabilities that can occur in multi-factor authentication mechanisms.We have also provided several interactive labs to demonstrate how you can exploit these vulnerabilities in multi-factor authentication.

Vulnerability in other authentication mechanisms:

In addition to the basic login functionality,most websites provide supplementary functionality to allow users to manage their account.For example,users can typically change their password or reset their password when they forget it.These mechanisms can also introduce vulnerabilities that can be exploited by an attacker.

Securing your Authentication mechanisms:

In this section,we will talk about how you can prevent some of the vulnerabilities we have discussed from occcuring in your authentication mechanisms.Authentication is a complex topic and, as we have demonstrated,it is unfortunately all too easy for weakness and flaws to creep in.Outlining every possible.However,there are several general principles that you should always follow.

Mitigation:

One of the most common authentication vulnerabilities is weak passwords. Many users still opt for easily guessable passwords, such as "123456" or "password." Creating strong, unique passwords for each account is essential to mitigate this risk.

Conclusion:

In this authentication (User name enumeration via different responses) in this we find out the username and password.when we insert the output of username and password in amazon application it doesnot open we use sample user name and password from port swigger.