COMPLETE APPLICATION TESTING WITH OWASP TOP 6 2024 VULNERABILITIES ON JIOMART

This project report is presented to satisfy the criteria for receiving the certificate in Ethical Hacking and Cyber Security

By

G.VENKATA GNAESH(23KQ5A0515) **3rdCSE(Batch-3)**

Under the esteemed guidance of

Sk. Prem Nazeer

Certified Ethical Hacker

Licensed Pentester

ABSTRACT

web application security is a top priority in a digital environment. TheProject, "Complete Application Testing with OWASP Bugs," A collaborative effort by our team of technical experts. Our mission is to a Comprehensive assessment of the selected web application, e.g., visual testing and vulnerabilities It is described in OWASP (Open Web Application Security Project) 2024 Top 6 list

This initiative covers all major vulnerabilities outlined by OWASP, spanning from SQL injection to Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF), among others. By combining manual examination with automated tools, we pinpoint vulnerabilities, evaluate their severity, and provide comprehensive guidance for remediation. Our foremost objective is to bolster the security posture of the designated application while also promoting a deeper understanding of web application security fundamentals. Our goal is to give people the confidence and skills to deal with online risks effectively. In a world full of digital dangers, our commitment to making web applications safer shows how serious we are about creating a secure online space for everyone

CONTENTS

performing server side request forgery (SSRF) on a web application

performing SQL injection on a web application

performing directory or path traversal on a web application

performing backdoor creation for OS powershell on a web application

OS COMMAND INJECTION

INTRODUCTION:-

OS command injection also referred to as shell injection, permits an attacker to run operating system (OS) commands on the server hosting an application. This vulnerability can lead to the complete compromise of the application and its data. Attackers often exploit OS command injection to infiltrate other components of the hosting infrastructure and take advantage of trust relationships, enabling them to extend their attack to other systems within the organization.

Risks that occurs when there is a os command injection vulnerability:-

- Attackers can gain unauthorized access to the system by executing commands
- Attackers can gain victim's Sensitive information
- Attackers can do data manipulation functions on victim's data
- Attackers can install malicious software it leeds to further exploitation, persistent threats, and potential financial loss.

Blind OS command injection:-

Blind OS Command Injection is a type of vulnerability found in web applications where attackers can inject malicious commands into the application, but they do not receive direct feedback on the output of these commands. Unlike traditional command injection, where attackers can immediately see the results of their injected commands, blind OS command injection requires attackers to infer the success or failure of their injections through indirect means.Attackers identify input fields or parameters in the

web application where they can inject commands. These could be text fields, URL parameters, or any other user-controllable input.

METHODOLOGY:-

Tool: Burp Suite

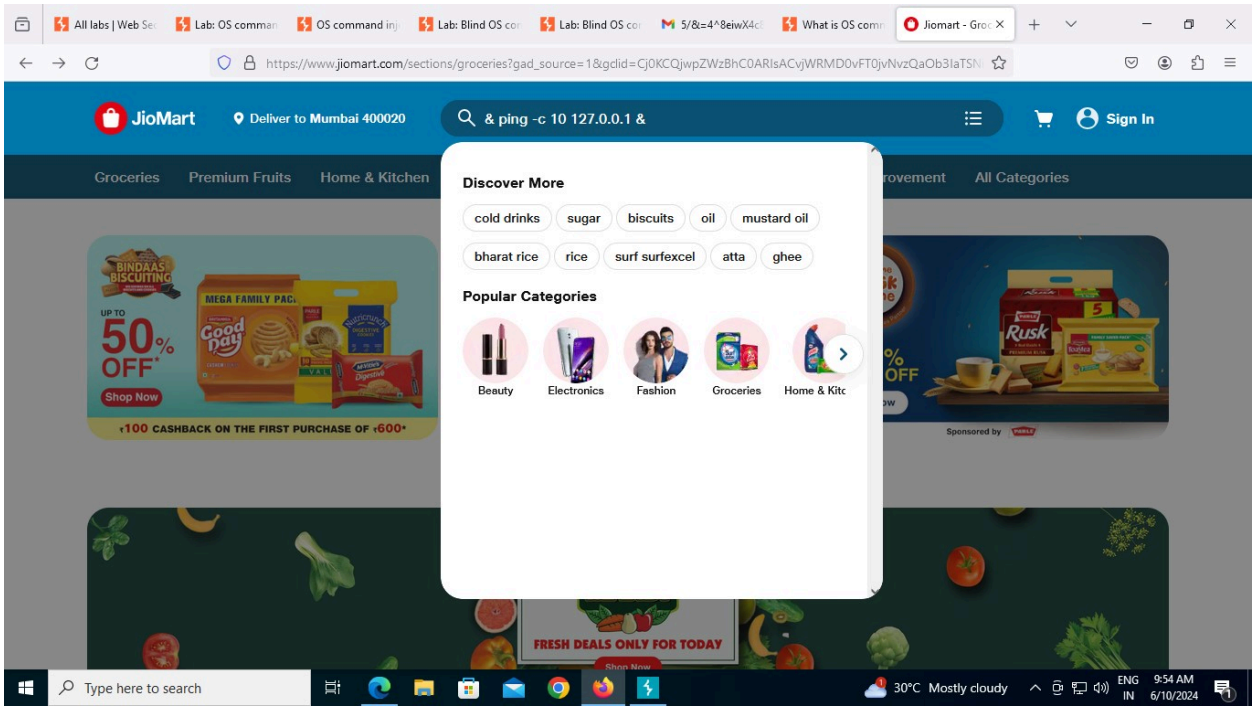
Target application: jiomart

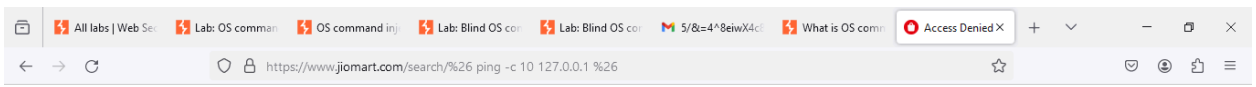
OS Command Injection is a critical security vulnerability that allows an attacker to execute some random commands on the host operating system on a vulnerable application. The methodology to identify, exploit, and mitigate OS command injection vulnerabilities.Lets find the os command injection vulnerability is present or not on Target application.

Working process:-

Os command injection can access unauthorized access on a web application server by inserting malicious code in input pannels

Open jiomart web application click on search bar and enter any os injection commands on that bar and try to get access of that application





Access Denied

You don't have permission to access "http://www.jiomart.com/search/%26%20ping%20-c%2010%20127.0.0.1%20%26" on this server.

Reference #18.6077668.1717993498.1ea6516

https://errors.edgesuite.net/18.6077668.1717993498.1ea6516

From the above screenshot jiiomart application doesn't accept this type of commands in this case this application is secured so os command injection cannot work on this application

VULNERABILITY DISCRIPTION:-

OS Command Injection is a cybersecurity vulnerability that occurs when an attacker is able to execute arbitrary system commands on a target server or system by injecting malicious commands into input fields or parameters that are subsequently passed to the operating system for execution.By addressing OS Command Injection vulnerabilities through secure coding practices, regular security assessments, and robust security controls, organizations can better protect their systems and data from exploitation by malicious actors

IMPACT OF OS COMMAND INJECTION:-

OS command injection occurs when an attacker manipulates input data to execute unauthorized commands on a system's operating system. This can lead to unauthorized access, data theft, system compromise, and other serious security breaches, posing risks to the confidentiality, integrity, and availability of the system and its data

In real-time scenarios, the impacts of OS command injection vulnerabilities can be quite significant:

- **Data Theft**: Attackers can execute commands to access sensitive data such as user credentials, financial information, or personal records.

- **Malware Injection**: Attackers can upload and execute malicious scripts or programs, leading to the installation of malware on the system, further compromising its integrity and security.

- **Shell Access**: Attackers can gain shell access to the system, allowing them to execute arbitrary commands with the privileges of the compromised process or user account.

- **Database Manipulation**: If the application interacts with a database, attackers can inject commands to manipulate the database, such as retrieving, modifying, or deleting records.

- **System Configuration Changes**: Attackers can modify system configuration settings, such as firewall rules or user permissions, to facilitate further attacks or maintain persistence on the compromised system

- **Financial Loss**: Organizations may suffer financial losses due to the theft of sensitive information, system downtime, and costs associated with incident response, remediation, and potential legal repercussions.

MITIGATION:-

Input Validation and Sanitization: Validate and sanitize all user inputs to ensure they do not contain any malicious commands or characters. Use whitelisting to only allow expected input formats.

Avoid Shell Commands: Whenever possible, avoid using shell commands within the application. Instead, use APIs or libraries provided by the programming language to perform necessary operations

Least Privilege: Run the application with the least privilege necessary to perform its functions. Avoid running with elevated privileges such as root or administrator

Web Application Firewalls (WAFs): Deploy WAFs to monitor and filter incoming traffic, blocking requests that match known attack patterns associated with command injection.

Regular Updates and Patching: Keep all software and libraries up to date with the latest security patches to mitigate known vulnerabilities that could be exploited for command injection.

CONCLUSION:-

OS Command Injection poses a significant threat to the security of systems and applications, allowing attackers to execute arbitrary commands and potentially gain unauthorized access or compromise sensitive data.By prioritizing security measures and actively addressing OS Command Injection vulnerabilities, organizations can better protect their systems and data from exploitation by malicious actors, ensuring the integrity and confidentiality of

their operations.

CROSS SITE SCRIPTING

INTRODUCTION:-

Cross-site scripting (XSS) is a vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can execute in the browsers of unsuspecting users, compromising their data, session cookies, or even taking control of their interactions with the site. It's a critical security concern for web applications, as it can lead to various attacks such as session hijacking, defacement, or stealing sensitive information.

Risks that occurs when there is a cross site scripting vulnerability:-

- Attackers can gain sensitive information such as login credentials, session cookies, or personal data from users
- Malicious codes can take control of user sessions, enabling attackers to pretend to be real users and carry out unauthorized activities
- Attackers can inject scripts to modify the appearance of web pages it damages the website's reputation
- By using XSS attackers can perform phishing attacks
- By using XSS attackers can perform Clint side attacks

Types of XSS attacks:-

There are three main types of XSS attacks those are

- **Stored XSS (Persistent):** The malicious script is permanently stored on the target server, such as in a database, message form, or comment field. Victims retrieve the malicious script when they request the stored information

- **Reflected XSS (Non-Persistent):** The malicious script is reflected off a web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request. The script is executed as part of the URL

- **DOM-based XSS:** This occurs when the client-side script of a web page processes data from an untrusted source in an unsafe way, resulting in the execution of the attacker's script

METHODOLOGY:

Tool: Burp Suite

Target application: jiomart

Cross-Site Scripting (XSS) is a security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can then execute in the context of the victim's browser, potentially leading to data theft, session hijacking, and other malicious activities on a application. The methodology to identify, exploit, and mitigate XSS vulnerabilities.Lets find the XSS vulnerability is present or not in our Target application.

Working process:-

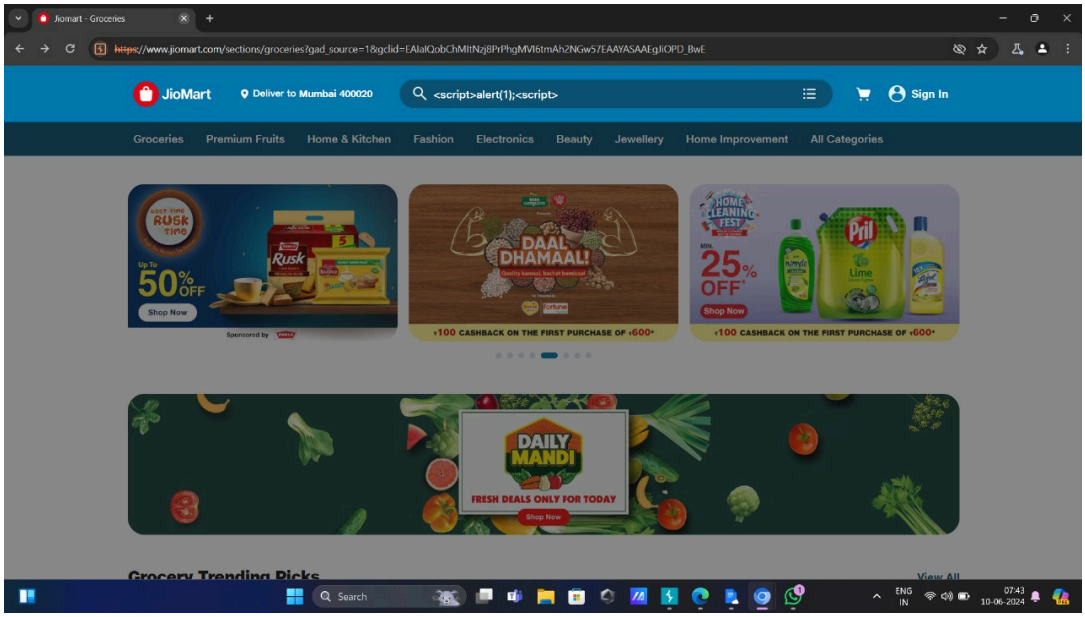
Reflected xss:

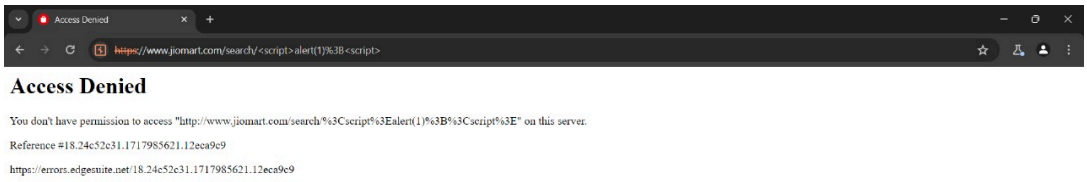
It is one types of xss vulnerability that helps to inject scripted code into application server to make reflect malicious data in application for short time

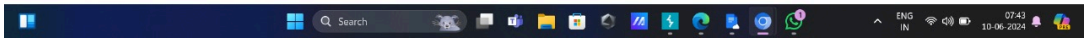
or untill refresh the page.

1) first open jiomart web application, and try to inject reflected injection in search panel.

2)example:<script>alert(1);</script>







From above screenshot the reflected xss injection doesn't working on jiomart web application because jiomart web application provides high security for this type of attacks

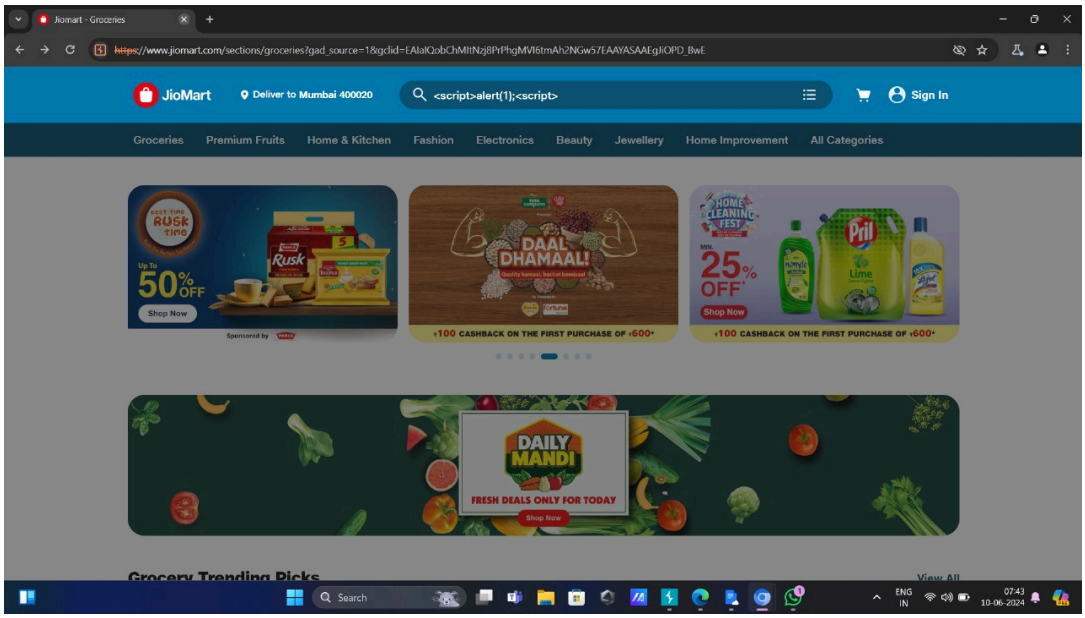
Stored xss:

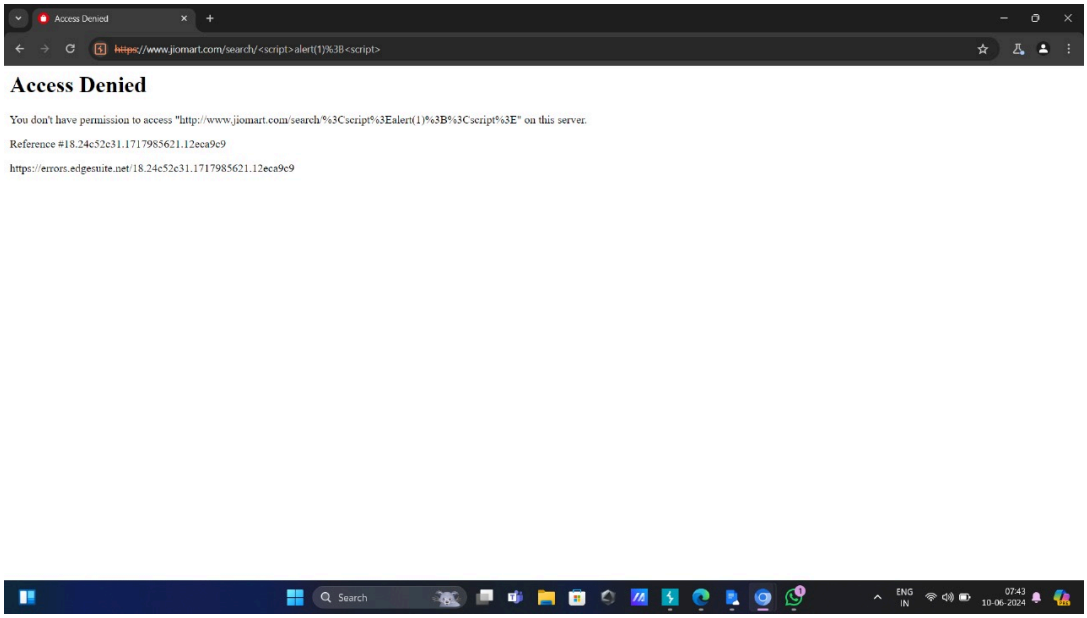
The stored xss is a vulnerability that helps to inject malicious code which stores in application server and also the injected data will be shown to others.

1) open jiomart web application and try to search something.

2) now insert stored xss script in search panel to inject script in server.

3) example:<script>alert(1)</script>

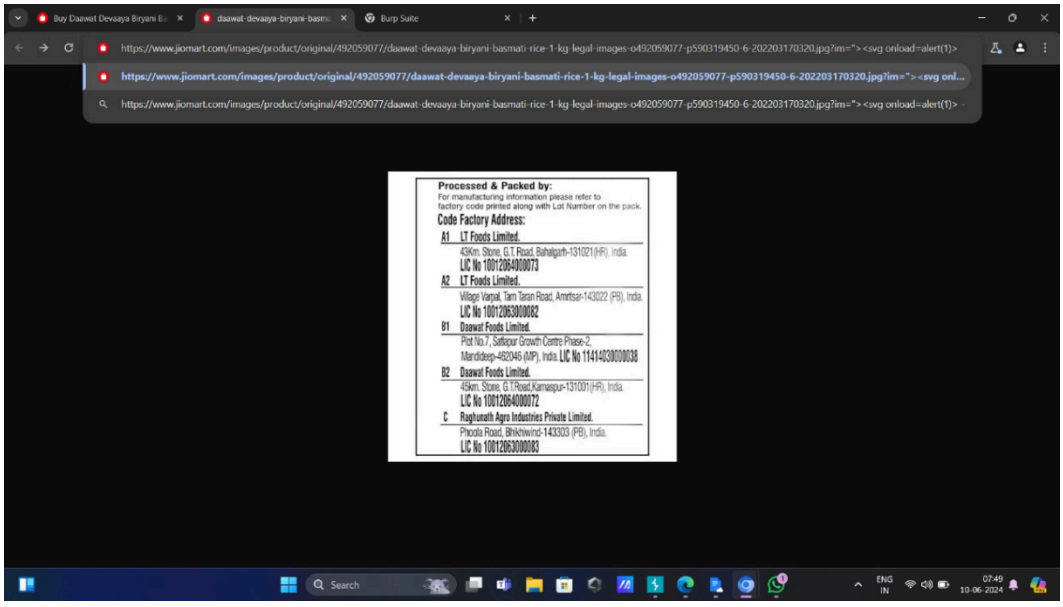


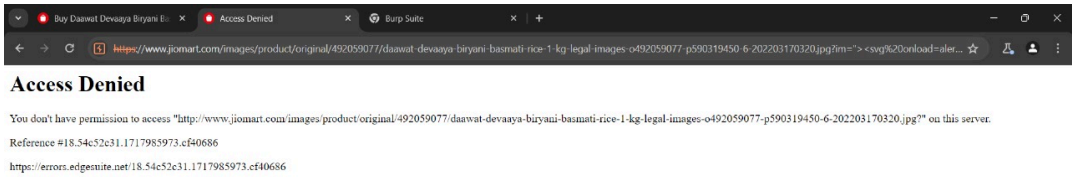


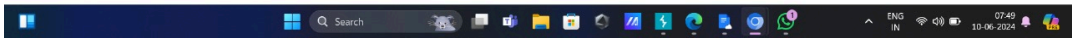
From above screenshot the stored xss injection doesn't working on jiomart web application because jiomart web application provides high security for this type of attacks

DOM-based xss:

1) open jiomart web application and try to inject DOM(document object model)-based xss script in search field.







From above screenshot the stored xss injection doesn't working on jiomart web application because jiomart web application provides high security for this type of attacks

VULNERABILITY DISCRIPTION:-

The impact of XSS vulnerabilities can be severe, leading to data theft, session hijacking, unauthorized access, or the spread of malware. To mitigate XSS vulnerabilities, developers should implement strict input validation and output encoding, utilize security mechanisms such as Content Security Policy (CSP), and conduct regular security assessments and code reviews to identify and remediate vulnerabilities before they are exploited by attackers. Additionally, educating users about the risks of clicking on suspicious links or executing untrusted scripts can help prevent XSS attacks

IMPACT OF CROSS SITE SCRIPTING:-

Cross-Site Scripting (XSS) is a type of security vulnerability typically found in web applications. It occurs when an attacker injects malicious scripts into content from otherwise trusted websites. The impact of an XSS vulnerability can be significant, ranging from user inconvenience to severe security breaches. Here's a breakdown of its potential impacts:

Theft of Sensitive Data:Attackers can steal cookies, session tokens, or other sensitive information stored in the browser.Phishing attacks can be conducted to steal user credentials

Session Hijacking:Attackers can hijack a user's session, gaining unauthorized access to their account.

Network Scanning:Using the victim's browser to perform network scanning and reconnaissance

Financial Losses: Direct financial losses due to fraud, remediation costs, and loss of business.

Brand Damage: Negative publicity resulting from security breaches can harm the organization's brand and reputation.

MITIGATION:-

Input Validation: Accept only known good input. Define a strict set of rules for what is considered valid input.Reject known bad input. This is less effective than whitelisting but can be used as an additional measure**.**

Sanitization Libraries:Use trusted libraries to sanitize input. Libraries like DOMPurify can help clean user input before it is processed or rendered.

Avoid Inline JavaScript:Avoid using inline JavaScript (e.g., event handlers like "onclick"). Instead, use external JavaScript files.

Use Modern Frameworks: Modern web development frameworks like React, Angular, and Vue.js have built-in protections against XSS by default.

Security Headers: Set appropriate security headers like X-XSS-Protection to enable the browser's built-in XSS protection

Escape Data in JavaScript: When inserting data into the DOM via JavaScript, ensure the data is properly escaped to avoid executing malicious scripts.

CONCLUSION:-

Cross-Site Scripting (XSS) remains a pervasive and significant threat to web application security. Its ability to manipulate user interactions and compromise sensitive data highlights the importance of proactive defense measures. XSS vulnerabilities stem from inadequate input validation and sanitization practices, allowing attackers to inject malicious scripts into web pages viewed by unsuspecting users.By prioritizing security in the development lifecycle, conducting regular security assessments, and raising awareness among users, organizations can effectively mitigate the risk posed by XSS vulnerabilities and protect their web applications and users from exploitation.

IDENTIFICATION AND AUTHENTICATION FAILURE

INTRODUCTION:-

One of the vulnerabilities highlighted is called ‘authenticate and identification failure’ that enables an attacker to gain login access into servers.Using this type of hole it is also possible to become an administrator for the site or a specific forum that has been attacked.
It will help the attackers to gain full control of the applications. A brute force attack is a method used by attackers to gain unauthorized access to systems, accounts, or encrypted data by systematically trying all possible combinations of passwords or encryption keys until the correct one is found.

Risks that occurs when someone performing brute force attack:-

- Attackers gain access to user accounts or administrative systems.
- Access to systems can lead to further exploitation, such as planting malware or backdoors.
- Attackers can steal sensitive information such as personal data, financial information, or intellectual property
- Repeated failed login attempts can trigger account lockout mechanisms, leading to denial of service for legitimate users.

METHODOLOGY:-

Tool: Burp Suite

Target application: jiomart

Brute force attacks involve systematically attempting a large number of combinations to crack passwords, encryption keys, or other security mechanisms. Now we are performing a brute force attack ethically, typically as part of a penetration test or security assessment on a realtime application here the realtime application is target application.

Working process:-

There is lot of brute force attacks now we are considering only 2 brute force attacks those are Brute force vulnerability and 2FA broken vulnerability

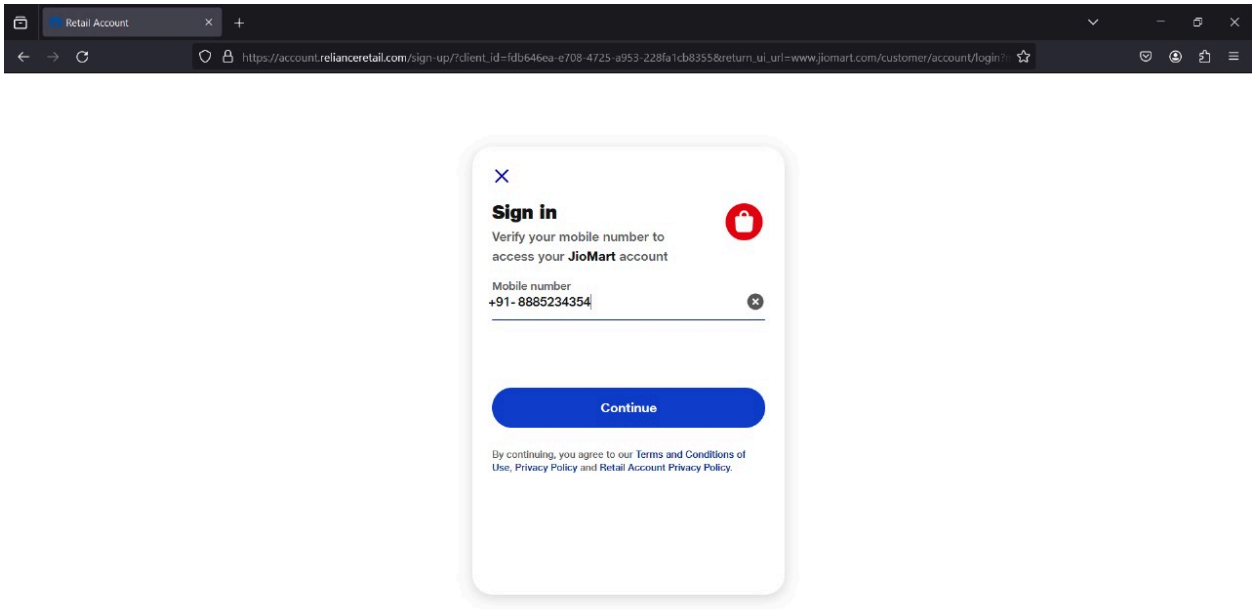
In this web application we can't perform 2FA broken vulnerability because this application doesn't contain any passwords for their accounts in this application login process is completely depends on mobile number's OTP . Now we performing OTP brute force attack in our application the process is given below:

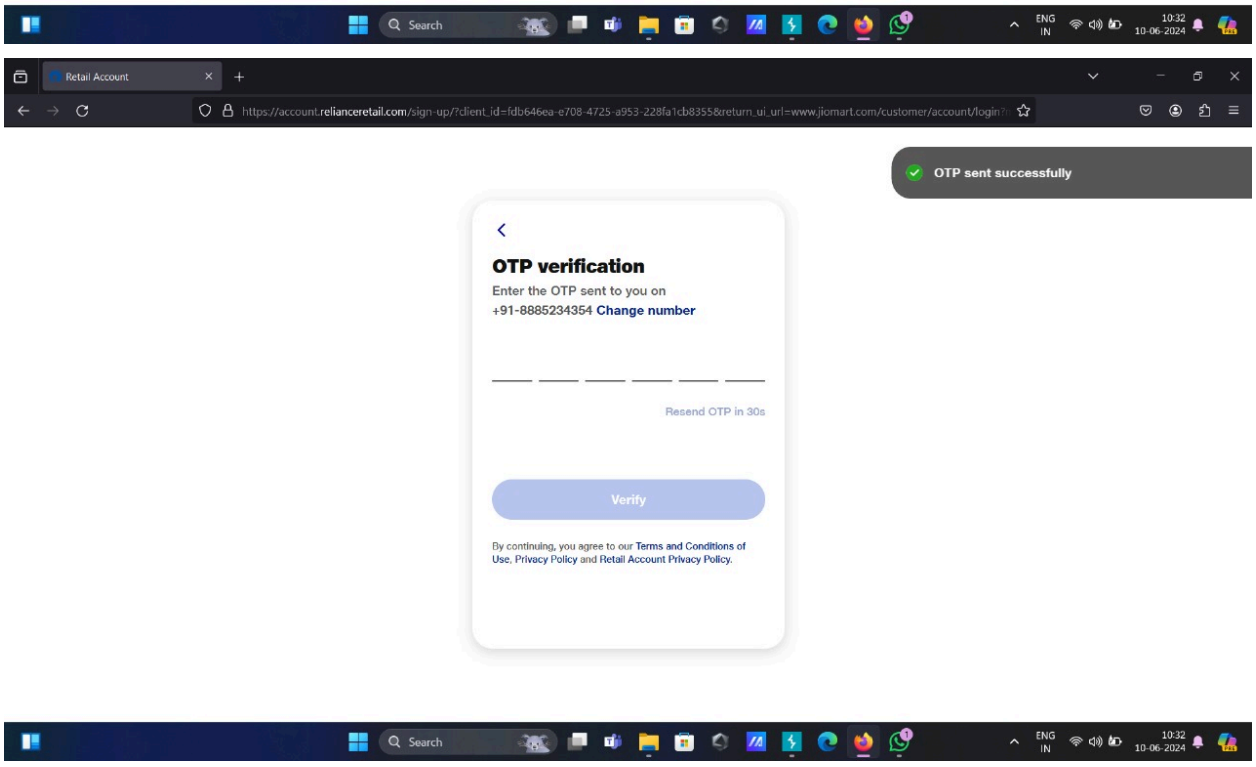
Open login page of jiomart web application and enter any mobile number click on signin and this page opens OTP entering panel enter any number and interrupt the opt signin functionality with burp suite

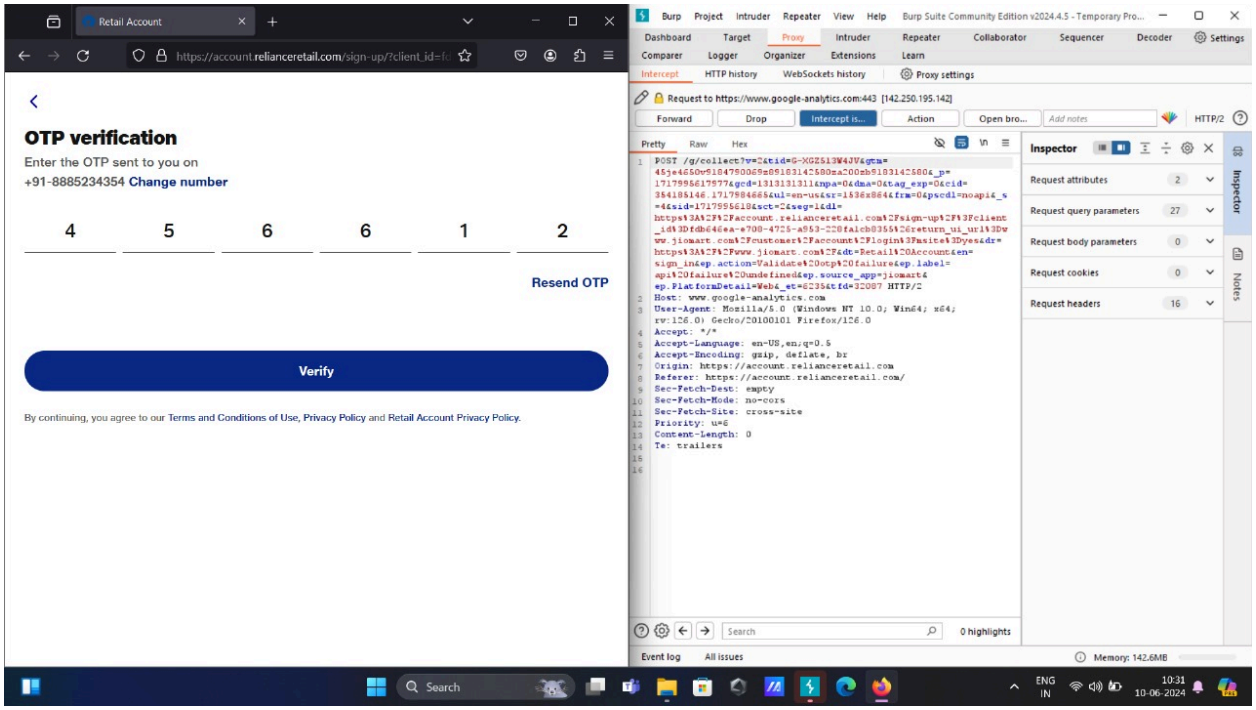
Find the credentials which gaves to input field and select them and send them to intruder tab if you find those credential details.

In intruder tab select the entered otp and add it payload position place.

Go to payloads tab and set payload and also combination list
After that start attack then we will find correct otp of that number. To find it check status code for 302 fount or else check length of the password the correct password length is different from rest of others.







From the screenshot the otp is in encrypted form that means this application is secured so we can't perform brute force attack in this application

VULNERABILITY DISCRIPTION:-

Authentication broken access can have severe consequences, including data breaches, unauthorized data access or modification, financial losses, and reputational damage. To mitigate this vulnerability, organizations should implement robust authentication mechanisms, enforce strong password policies, enable multi-factor authentication, regularly update authentication protocols, properly configure access controls, and monitor for suspicious authentication activity. Regular security assessments and audits are also crucial for identifying and addressing authentication vulnerabilities before they can be exploited by attackers.

IMPACT OF AUTHENTICATION BROKEN ACCESS:-

- **Unauthorized Access**: Attackers can gain access to user accounts, potentially accessing sensitive information or performing unauthorized actions.

- **Account Takeover:** Compromised credentials can lead to attackers taking over user accounts, changing passwords, and locking out legitimate users.

- **Data Breach:** Sensitive data such as personal information, financial records, and proprietary data can be exposed.

- **Reputation Damage**: Loss of user trust and damage to the organization's reputation can occur if user accounts are frequently compromised.

- **Financial Loss:** Direct financial losses can result from fraudulent transactions or fines related to regulatory non-compliance

MITIGATION:-

Strong Authentication Mechanisms: Implement strong password policies, use MFA, and ensure secure session management practices.

Proper Access Controls: Implement role-based access control (RBAC) and ensure least privilege principles are followed.

Regular Audits and Testing: Perform regular security audits, penetration testing, and code reviews to identify and fix vulnerabilities.

User Education: Educate users about the importance of security practices such as not sharing passwords and recognizing phishing attempts.

Monitoring and Logging: Implement robust logging and monitoring to detect and respond to suspicious activities promptly.

CONCLUSION:-

To mitigate the risk of authentication broken access, organizations must prioritize robust security measures, including implementing strong authentication mechanisms, such as multi-factor authentication (MFA), enforcing strict input validation and sanitization, and regularly updating and patching software to address known vulnerabilities. Additionally, thorough security assessments, code reviews, and penetration testing can help identify and remediate vulnerabilities before they are exploited by malicious actors.

Server-Side Request Forgery (SSRF)

INTRODUCTION:-

The security assessment on a application, it was found out that there is a critical Server-Side Request Forgery (SSRF) which poses a significant risk to its security. Attackers exploit this vulnerability to make unauthorised internal or external resource requests through the application ultimately leading to data leakages, unauthorized access or other risks related to security. The analysis provided here-in gives detailed information on characteristics of SSRF vulnerability exposed, implications and recommended remedial plans.As part of the evaluation process, it emerged that the search function in that application admits “file:///etc/passwd” as input thus revealing sensitive System Information. Such behaviour speaks for itself as an obvious SSRF flaw which underlines how urgent this issue

needs prompt attention and fix up. In coming sections we will talk about details of SSRF vulnerability inclusive of its impact on security and steps required for mitigation planning efficiency.

METHODOLOGY:-

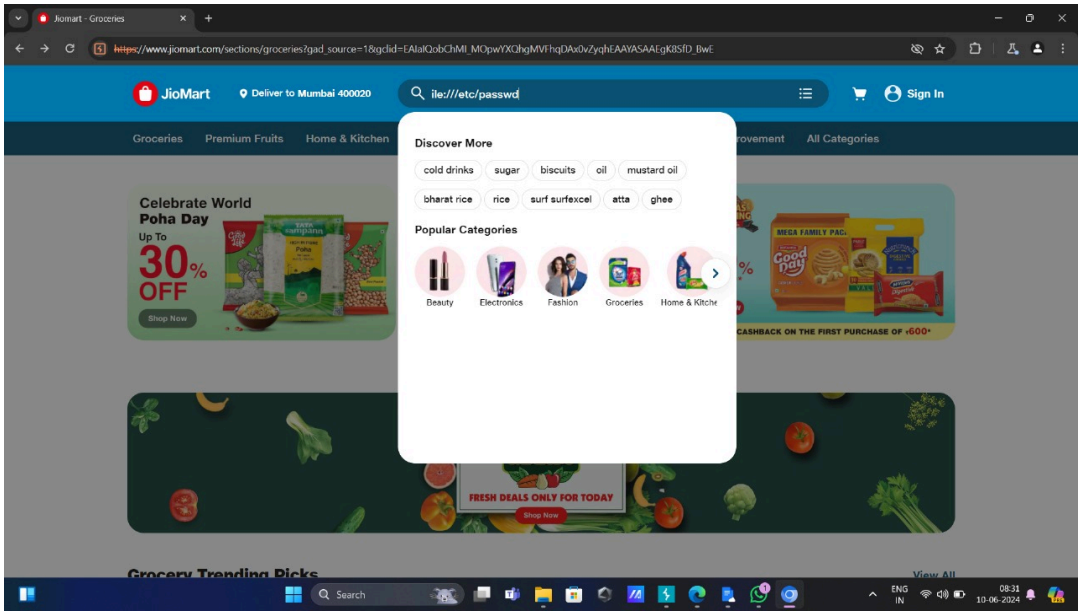
Tool: Burp Suite

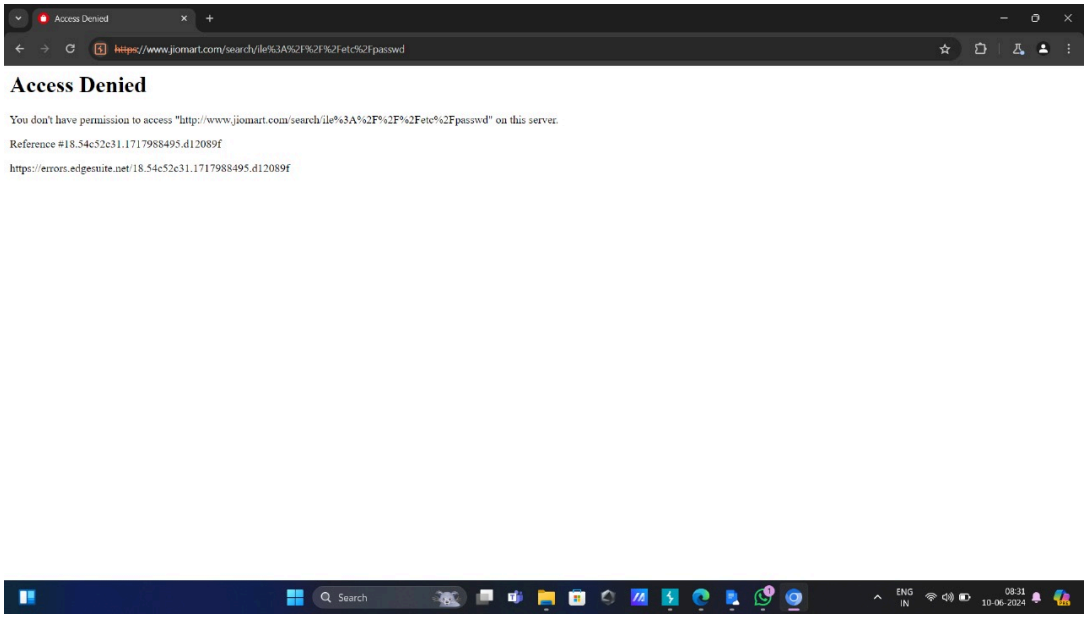
Target application: jiomart

The assessment of the Server-Side Request Forgery (SSRF) vulnerability in our application is based on specific operation here the main thing is to validate the SSRF vulnerability by manipulating the application's functionality to get sensitive data. Lets find out the SSRF vulnerability is present or not in our target application

Working process:-

1. Open search bar on jiomart web application
2. Insert any SSRF playload like(ile:///etc/passed)
3. Now we are inserting the above playload





From the above screenshot this web application doesn't provide access of server so the main thing is this application server is highly secured

VULNERABILITY DISCRIPTION:-

Server-Side Request Forgery (SSRF) is a cybersecurity vulnerability that occurs when an attacker can manipulate input parameters to make a server perform malicious requests on their behalf. This vulnerability enables attackers to bypass firewalls and access internal resources, potentially leading to data exfiltration, server compromise, or further attacks against other systems within the network.Regular security assessments, penetration testing, and vulnerability scanning can help identify and remediate SSRF vulnerabilities before they are exploited by attackers

IMPACT OF SSRF:-

vulnerabilities can have significant impacts on the security and integrity of an application and its associated systems.these are some impacts of SSRF vulnerability

Internal Network Scanning and Enumeration:SSRF can allow attackers to perform internal network scanning to discover services and systems that are not exposed to the public internet. Attackers can identify internal IP addresses, open ports, and running services, which can be used for further exploitation.

Data exposure:SSRF can allow attackers to retrieve sensitive internal data by making requests to internal services that may hold confidential information

Legal and Compliance Risks: Data breaches and exposure of sensitive information due to SSRF can lead to legal consequences and non-compliance with data protection regulations

Reputation Damage:A successful SSRF attack and the subsequent data breach can severely damage an organization's reputation, leading to loss of customer trust.

Resource Abuse:SSRF can lead to unintended use of resources, such as making extensive HTTP requests or triggering expensive operations, resulting in increased operational costs

MITIGATION:-

Input Validation and Sanitization:Validate and sanitize all user inputs rigorously.Use a whitelist approach, allowing only certain URLs or domains to be accessed.
Firewall Rules:Configure firewall rules to block unwanted outbound traffic from your servers.Use an allowlist for outgoing traffic to ensure only specific, necessary domains/IPs are reachable.

SSRF Detection:Implement monitoring and alerting for unusual outbound traffic patterns that could indicate SSRF attempts.Use web application firewalls (WAF) that include SSRF protection.
Web Security Scanners: Regularly scan your applications using automated security tools that can detect SSRF vulnerabilities
Security Patches and Updates: Regularly update your software, libraries, and dependencies to patch known vulnerabilities, including those that might be leveraged in SSRF attacks.

CONCLUSION:-

Server-Side Request Forgery (SSRF) is a critical security vulnerability that can have significant implications for web applications and the infrastructure they reside on.To mitigate SSRF vulnerabilities, it's essential for developers and system administrators to implement strict input validation and sanitization, enforce whitelists of allowed URLs or IP addresses, and employ network segmentation to restrict access to sensitive resources. Regular security assessments, including penetration testing and code reviews, are also critical for identifying and addressing SSRF vulnerabilities in a timely manner.

SQL INJECTION

INTRODUCTION:-

SQL Injection is a vulnerability where attackers insert malicious SQL commands into an application's input fields. These commands are then executed by the backend SQL server, causing unexpected and potentially damaging outcomes. SQL Injection is highly concerning due to its risk level and is ranked as a top priority (A1) in OWASP vulnerabilities. It happens when

applications fail to properly sanitize user input data, allowing attackers to manipulate SQL queries with malicious intent.

Risks that occures whan there is a sql injection vulnarability:

- Attackers can access sensitive data stored in a database, such as usernames, passwords, credit card numbers, and personal data
- Attackers can modify or delete data within the database, leading to data corruption or loss.
- SQL injection can allow attackers to bypass authentication mechanisms and gain unauthorized access
- Attackers can gain database's administrator access easily

Types of SQL injections:

- **Blind SQL injection**

Blind SQL Injection is a form of SQL Injection Attack whereby the attacker analyzes server responses to injected SQL queries to indirectly gain information but injection results are invisible. In regular SQL Injection, data is extracted directly from the database while in blind SQL injection, one asks the database true or false questions that can enable stealing data.

- **Time-Based SQL Injection:**

Exploits time delays in database responses to infer information about the database structure or contents.

- **Union sql injection**

It Involves using the UNION SQL operator to combine the results of two or more SELECT queries and retrieve additional information from the database.

METHODOLOGY:-

Tool: Burp Suite

Target application: jiomart

Working process:

Blind sql injection

Payloads that we use :

Admin' or '1'='1'/*

Admin'or 1=1 or"='

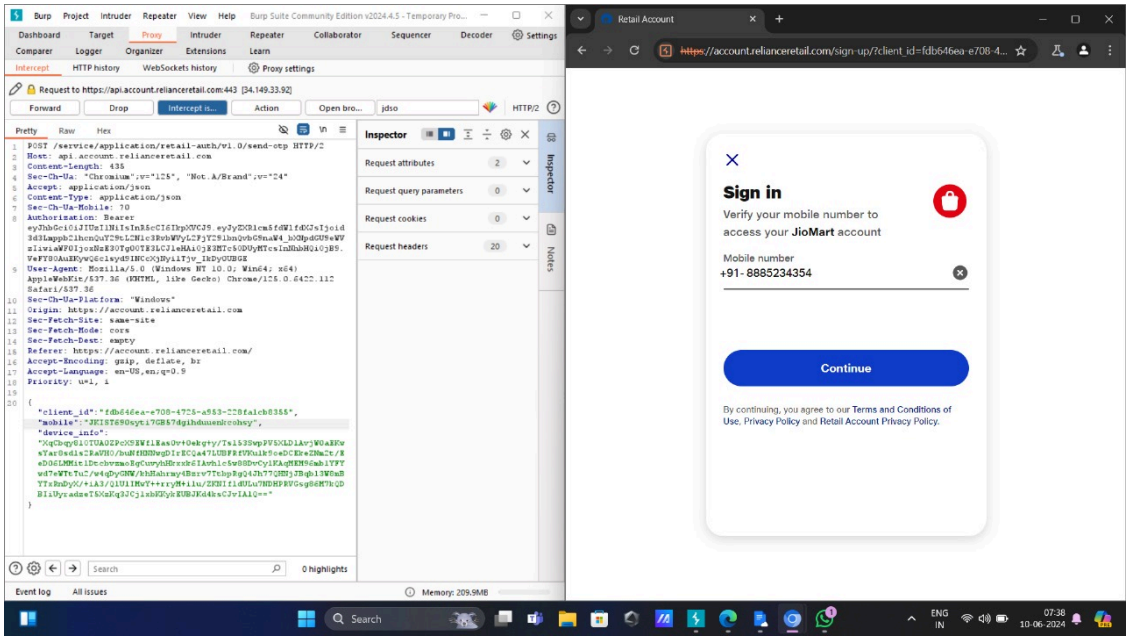
Admin'or 1=1--

Admin' or 1=1/*

1) First open jiomart login page and give some random credentials and intercept it with burpsuite.

2) search for your enter number in proxy request capture and modify it with blind sql payload

3) if there is a sql injection vulnarability then you will login into admin panel by using this payload.



In the above example the captured request doesn't shown number you entered due to encryption and abstraction.

That means the login credentials are encrypted in jiomart web application.

So we can't perform blind SQL injection operation on this application

Error-based sql injection:

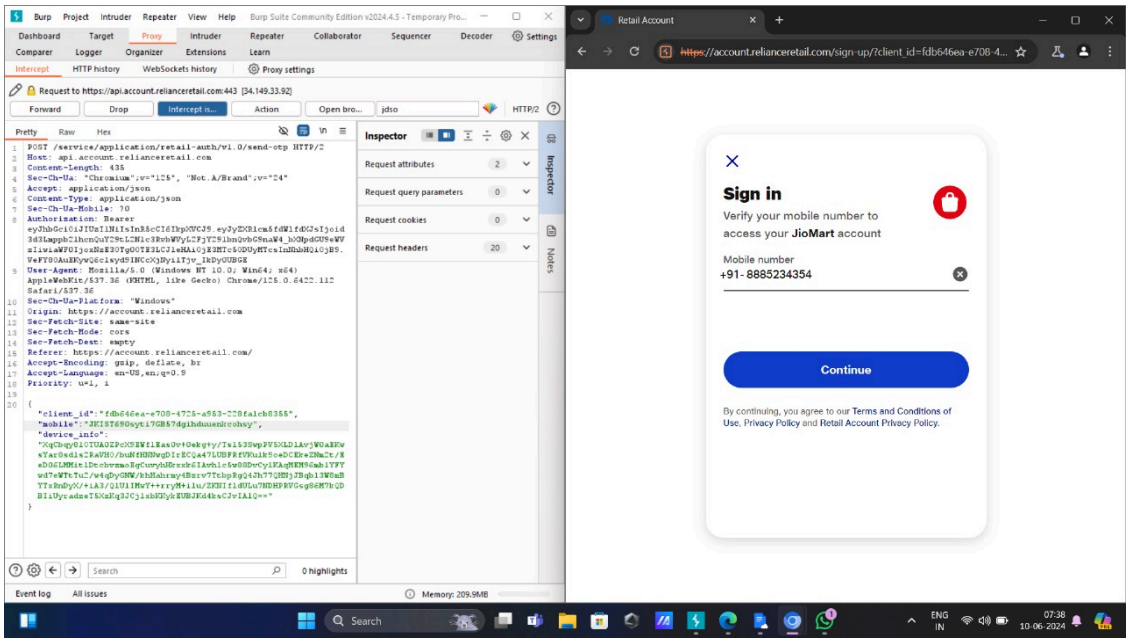
Payload used:

' OR '1'='1

Process:

1) First enter random number in input panel

2) Intercept the login request and modify the entered number with error-based payload.



In the above example the captured request doesn't shown number you entered due to encryption and abstraction.

That means the login credentials are encrypted in jiomart web application

So we can't perform error based SQL injection operation on this application

Union sql injection:

Payloads used:

' UNION SELECT 'a',NULL,NULL,NULL--

' UNION SELECT NULL,'a',NULL,NULL--

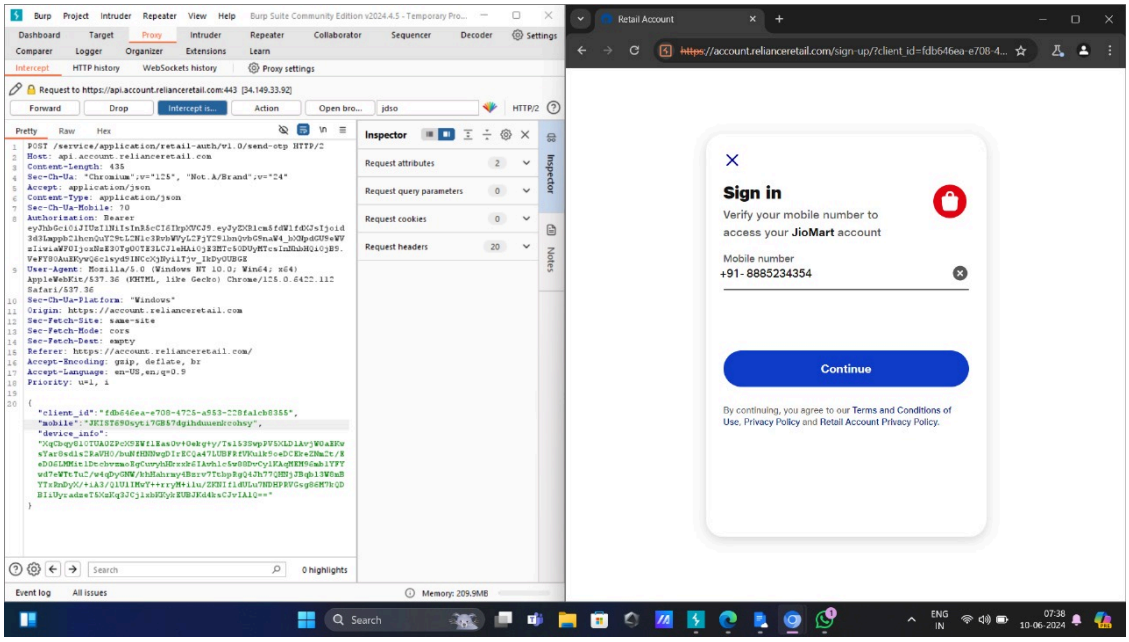
' UNION SELECT NULL,NULL,'a',NULL--

' UNION SELECT NULL,NULL,NULL,'a'--

PROCESS:

1) First we need to enter random number into input panel and intercept the login functionality.

2) Relpace the input field with union sql injection payload and try to access the data which is in database.



In the above example the captured request doesn't shown number you entered due to encryption and abstraction.

That means the login credentials are encrypted in jiomart web application

So we can't perform union SQL injection operation on this application

VULNERABILITY DISCRIPTION:-

SQL injection is a cybersecurity vulnerability that occurs when an attacker inserts malicious SQL code into input fields of a web application's SQL query, exploiting vulnerabilities in the application's database layer. This allows the attacker to manipulate the application's SQL queries to execute arbitrary commands, steal sensitive data, modify database contents, or even gain unauthorized access to the underlying server.To mitigate SQL injection vulnerabilities, developers should implement strict input validation and sanitization measures, utilize parameterized queries or prepared statements, and employ least privilege principles to restrict database access rights. Regular security assessments and code reviews are also essential to identify and address potential SQL injection vulnerabilities before they can be

exploited by malicious actors.

IMPACT OF SQL INJECTION:.

Data Manipulation: SQL injection can allow attackers to do data manipulation operations like modify, add, or delete data within the database.

Authentication Bypass: Attackers can gain unauthorized access to systems by bypassing authentication mechanisms, potentially escalating privileges and compromising entire systems

Data Breach: Attackers can access, retrieve, and exfiltrate sensitive information, such as personal identifiable information (PII), financial data, and proprietary business information.

Business loss: organization can face lot of complaints from users because attackers can miss use their information

Reputation Damage: Public disclosure of a data breach can damage an organization's reputation, leading to a loss of customer trust and potential business opportunities**.**.

Service Denial: SQL injections can be used to cause denial-of-service (DoS) attacks, rendering applications unavailable to legitimate users.

MITIGATION:

Data validation and sanitization: Validate and sanitize user inputs to ensure they conform to expected formats and types before using them in SQL queries.

Web Application Firewall: Use WAFs to detect and block SQL injection attempts. WAFs analyze incoming traffic and can prevent known attack

patterns.

Penetration Testing: vulnerabilities involves systematically testing and exploiting potential weaknesses in an application's interaction with its database it helps to find bugs

Error Handling and Logging: Implement proper error handling and logging mechanisms in your application. Avoid displaying detailed error messages to users, as they can provide valuable information to attackers. Log all relevant security events, including failed login attempts and suspicious activities, to detect and respond to potential attacks.

Conclusion:

SQL injection remains one of the most critical security vulnerabilities in web applications, posing significant risks such as unauthorized data access, data loss, and system compromise. It exploits improper handling of user input within SQL queries, allowing attackers to execute arbitrary SQL commands

DIRECTORY OR PATH TRAVERSAL

INTRODUCTION:-

The directory OR path traversal is siviour vulnarabillity that cause to make web applicaation vulnarable The directory OR path traversal is dangerous vulnerability that lead to making web application vulnerable. The suffix the directory means a directory or a folder or array or collection of other directories.This is the best time to say that some web applications have a number of urls, for instance; the log in page url, the home page url and so on. . All these all urls are compiled on one single place either at the time of manifesting a directory url. If we get that directory url then only we can very easily understand and analyze the overall structure of the entire application. From this pattern we can also obtain such parameters as logins, sensitive data, etc... It is however this vulnerabity that is critical for web applications to undertake.

METHODOLOGY:-

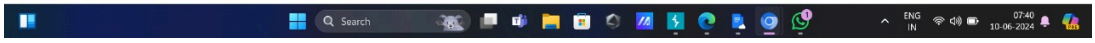
This assessment of the directory or path traversal vulnerability in the jiomart application. This vulnerability assessment provides complete path traversal analysis in jiomart as follows

Working process:-

- Observe the URL of the jiomart application and try to insert directories

Teste.text: this directory will helps to visit test case design in the given application

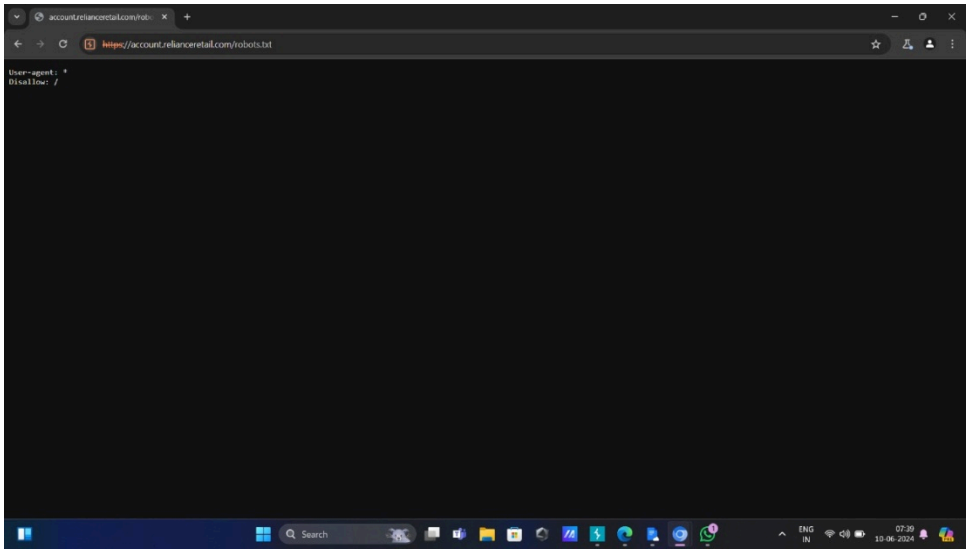
we have to open jiomart home page and try try to insert teste.txt directory on jiomart home page url



From above screenshot this web application doesn't allows teste.txt directory.

Robots.txt: This directory which is used to find all list of url of entire application

we have to open jiomart home page and try try to insert robots.txt directory on jiomart home page url



From above screenshot this web application doesn't allows robots.txt directory.

Etc/passwd.txt: it holds information about user accounts. Each line in the file represents a user and includes details like username,user id ,home directory,etc

we have to open jiomart home page and try try to insert etc/passwd.txt directory on jiomart home page url

From above screenshot this web application doesn't allows Etc/passwd.txt directory.

VULNARABILITY DESCRIPTION:

Path traversal occurs when an application allows user input to specify a file or directory location without properly validating or sanitizing it. Attackers can exploit this vulnerability to navigate to sensitive system files, configuration files, or other directories on the server.Typically, an application constructs a file path using user input without verifying its legitimacy.Path traversal can lead to unauthorized access to sensitive data or system files.By understanding and mitigating path traversal vulnerabilities, developers can enhance the security of their applications and prevent unauthorized access to sensitive data and system resources.

IMPACT OF PATH TRAVERSAL VULNERABILITY:-

Path traversal vulnerabilities can have significant impacts on both the security of an application and the integrity of the system where the application is deployed. Here are some of the key impacts:

- Unauthorized Access
- Data Manipulation
- Reputation Damage
- Regulatory Compliance Violations
- Business Disruption
- Data Exfiltration
- Data breach or data steal
- Sensitive data exposure

MITIGATION:-

Input Validation: Validate and sanitize user input to ensure it doesn't contain any directory traversal characters like "../" or "..".

Whitelisting: Only allow access to specific directories or files that are explicitly allowed, rather than trying to blacklist disallowed ones.

Use Libraries: Utilize secure file access libraries or functions provided by your programming language or framework, as they often include built-in protections against path traversal attacks.

Logging and Monitoring: Implement logging and monitoring mechanisms to detect and respond to suspicious access attempts, providing insights into potential path traversal attacks.

Encoding: Encode user input before processing it, which can prevent malicious characters from being interpreted as directory traversal sequences.

Conclusion:-

path traversal vulnerabilities pose a significant threat to the security of web applications, allowing attackers to access sensitive files and directories outside of the intended scope. However, by employing a combination of mitigation techniques such as input validation, whitelisting, encoding, and proper file permissions, developers can effectively reduce the risk of exploitation. Additionally, proactive measures such as using secure file access libraries, implementing logging and monitoring, and conducting regular security audits are essential for maintaining robust defenses against path traversal attacks

ADDITIONAL PROJECT

BACKDOOR CREATION FOR OS POWERSHELL

INTRODUCATION:

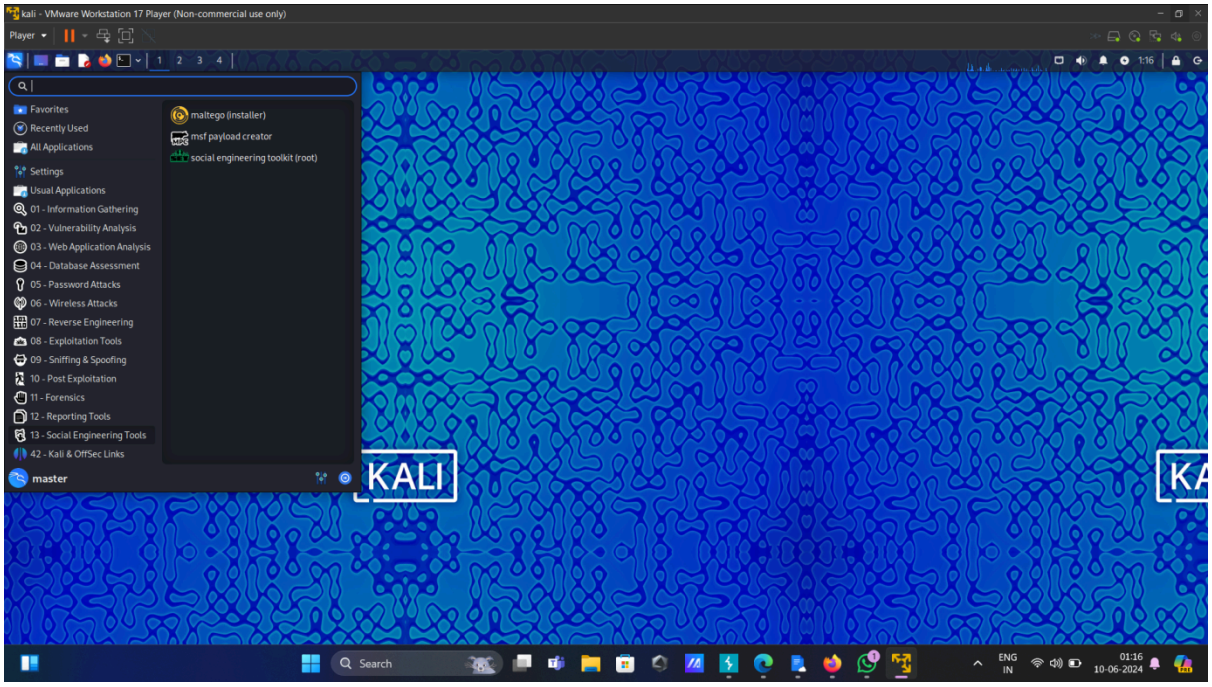
Backdoor creation in the context of operating systems and PowerShell refers to the practice of establishing unauthorized, covert access to a system. This is typically done by malicious actors to maintain persistent access for data theft, espionage, or system manipulation. PowerShell, a powerful scripting language and command-line shell for Windows, is often exploited due to its extensive capabilities and deep integration with the Windows operating system..

METHODOLOGY:

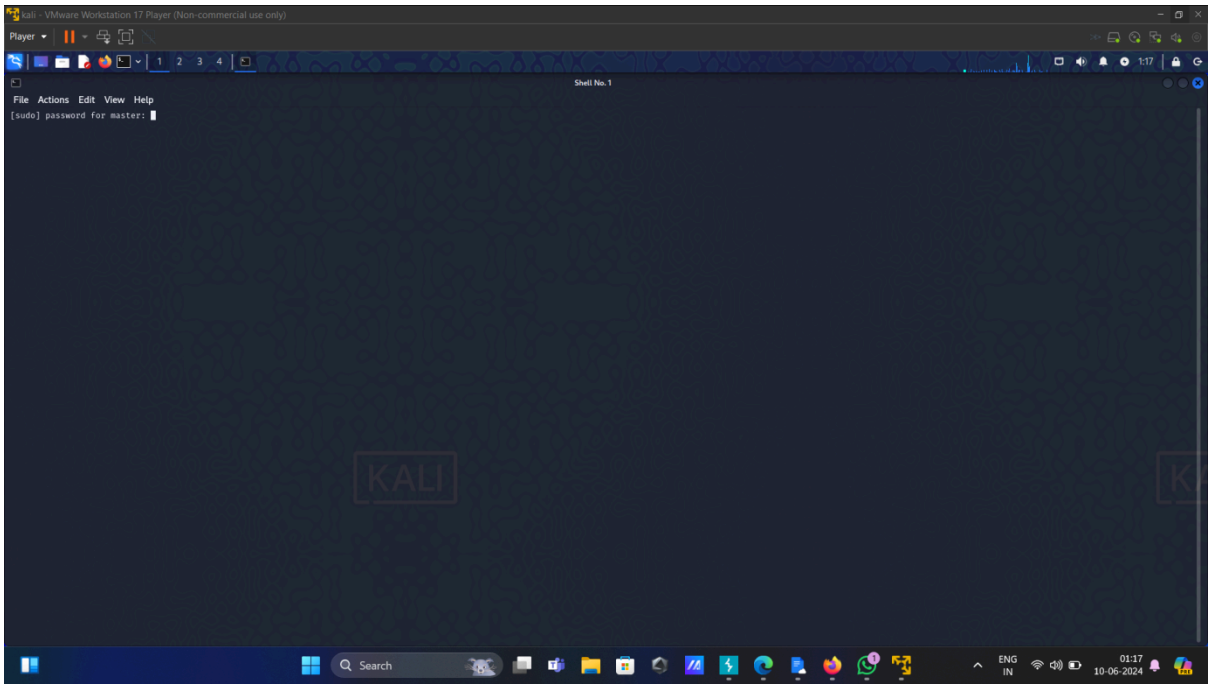
Creating a backdoor using PowerShell involves a structured approach to ensure the persistence and stealth of the malicious access we are usinng a tool called “Social engineering toolkit”.And also we use “kali linux” operating system.

PROCESS:

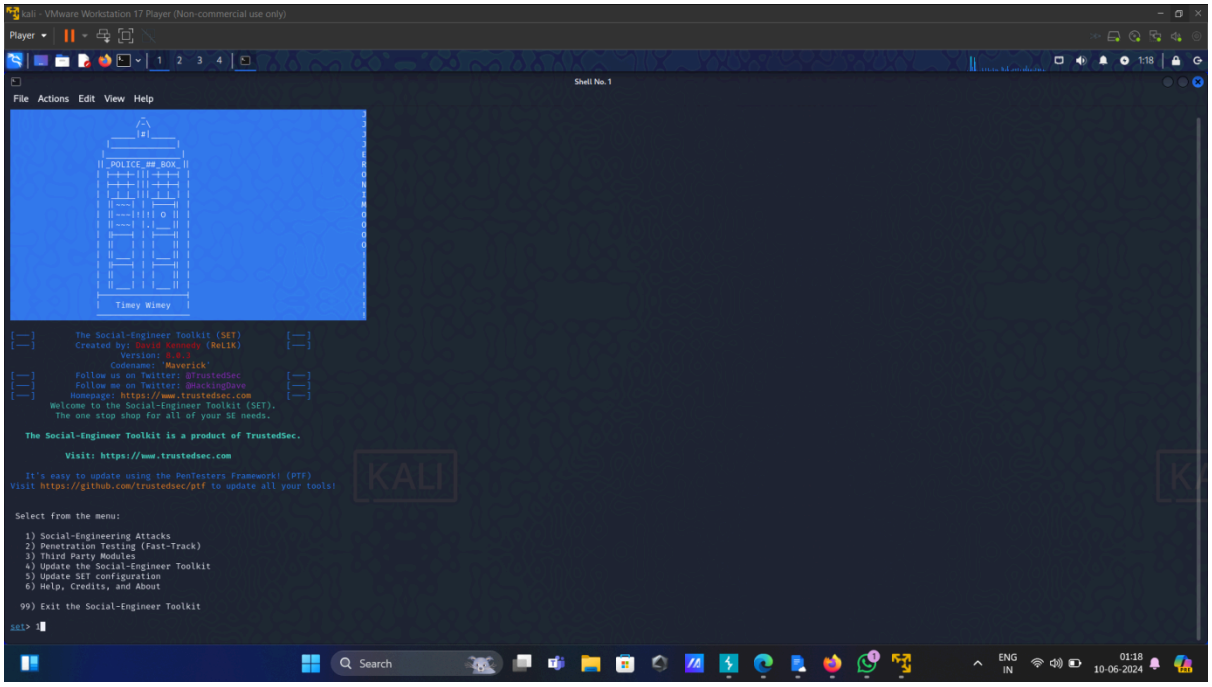
1) open social engineering toolkit.



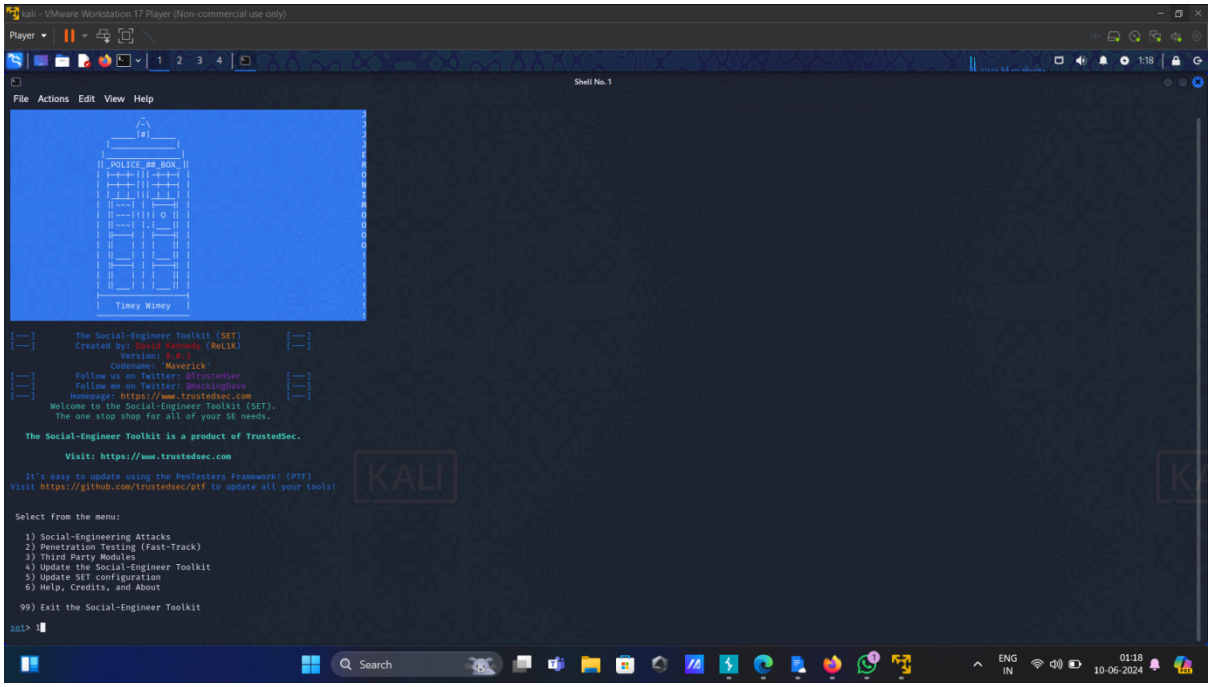
2) give password to root access.



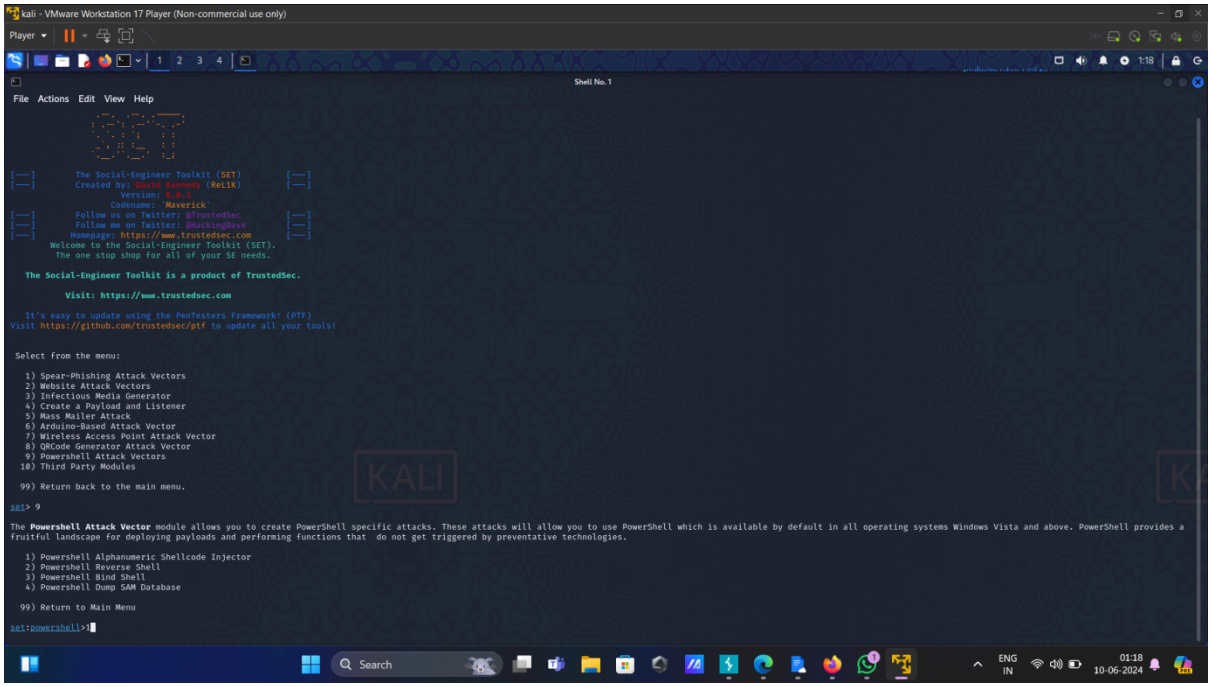
3) select social- engineering attack.



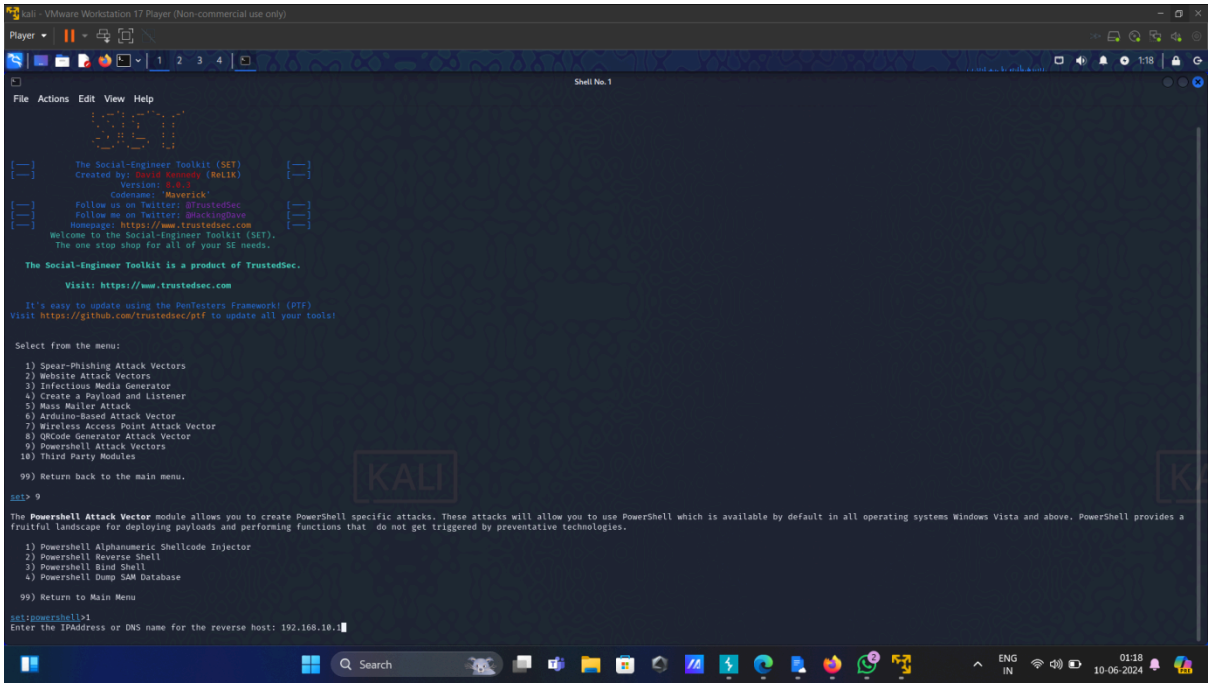
4)select powershell vector attack



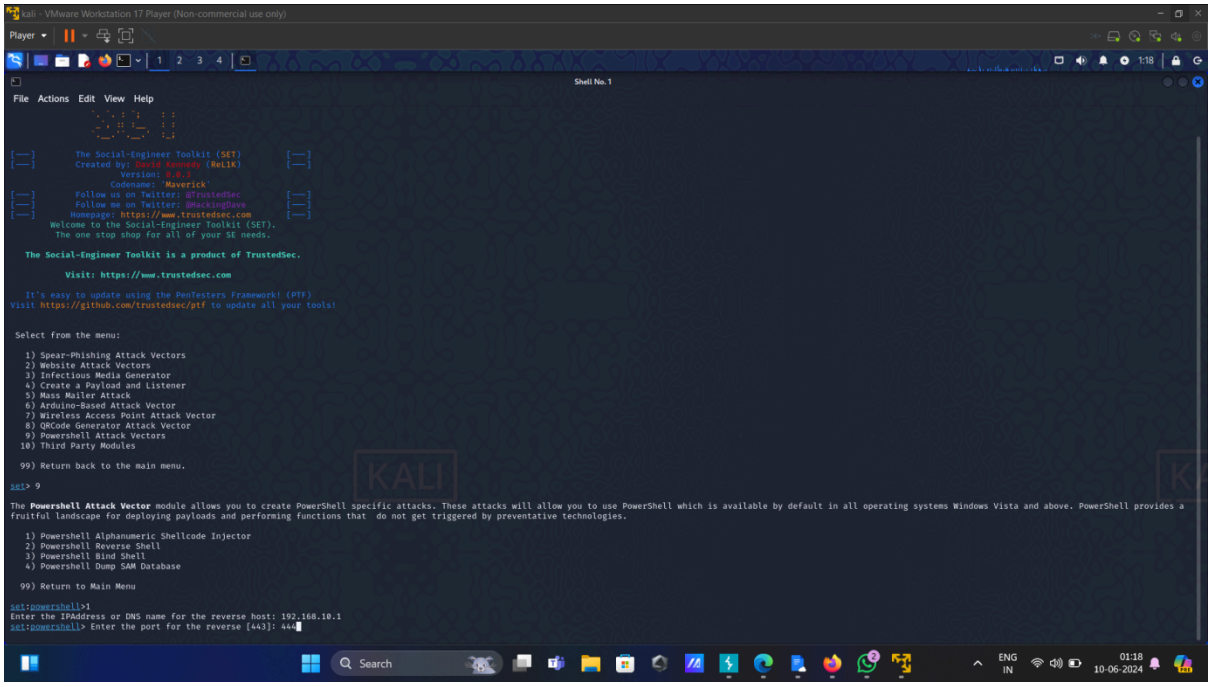
5)select powershell alphanumeric vector attack



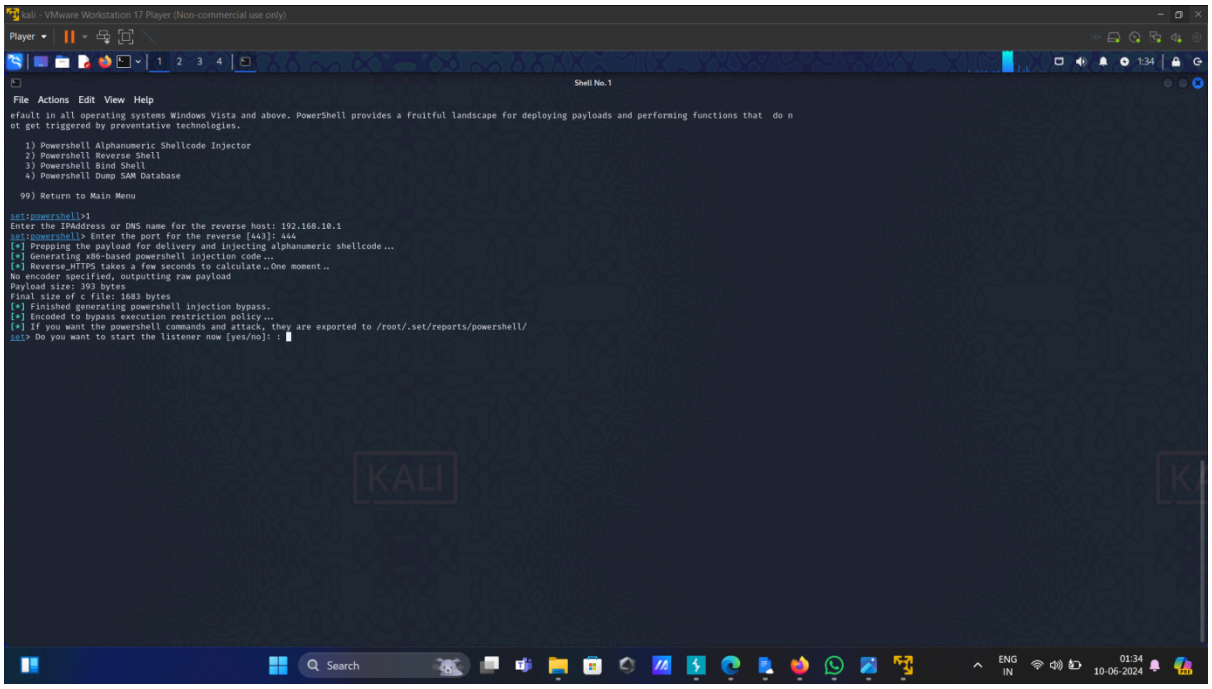
6)Enter victims ip address.



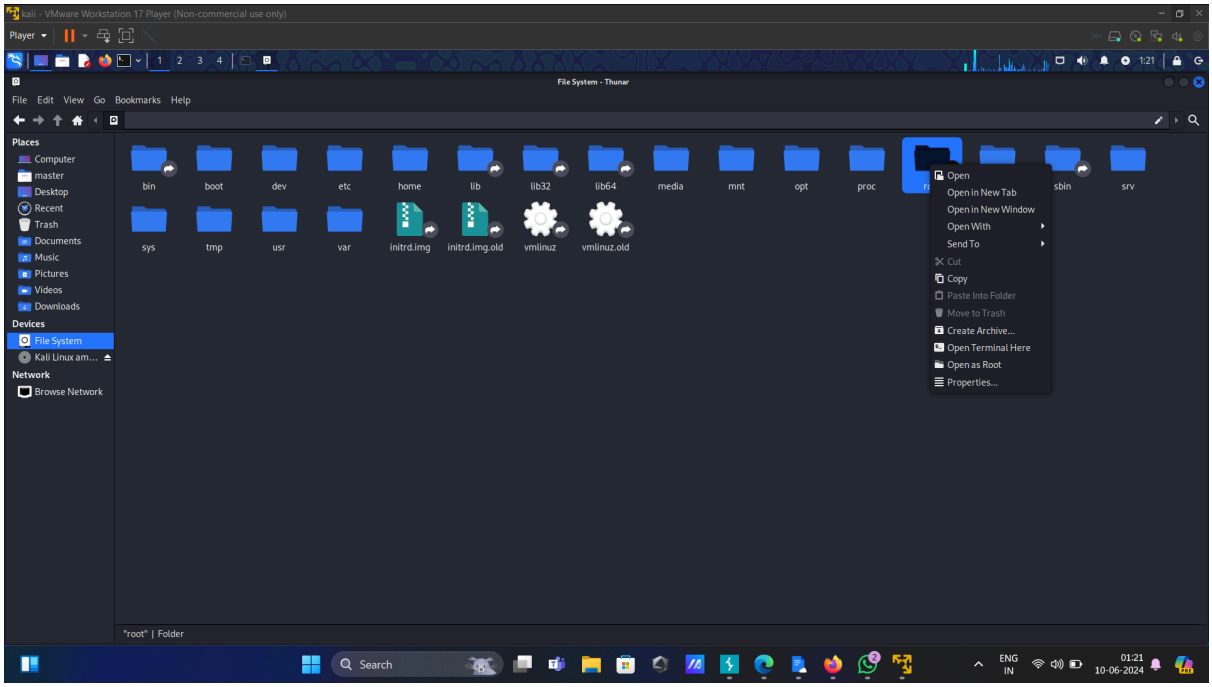
7) Enter port number.



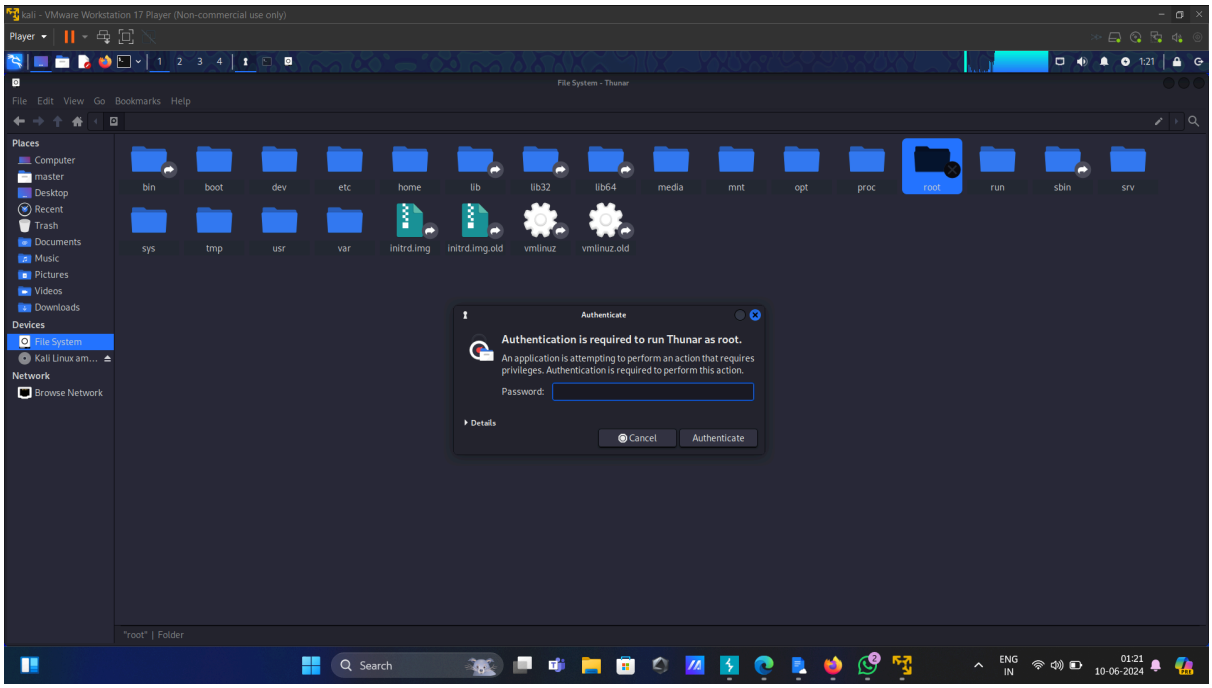
8) After that the payload is ready that means we can check that payload in file system.



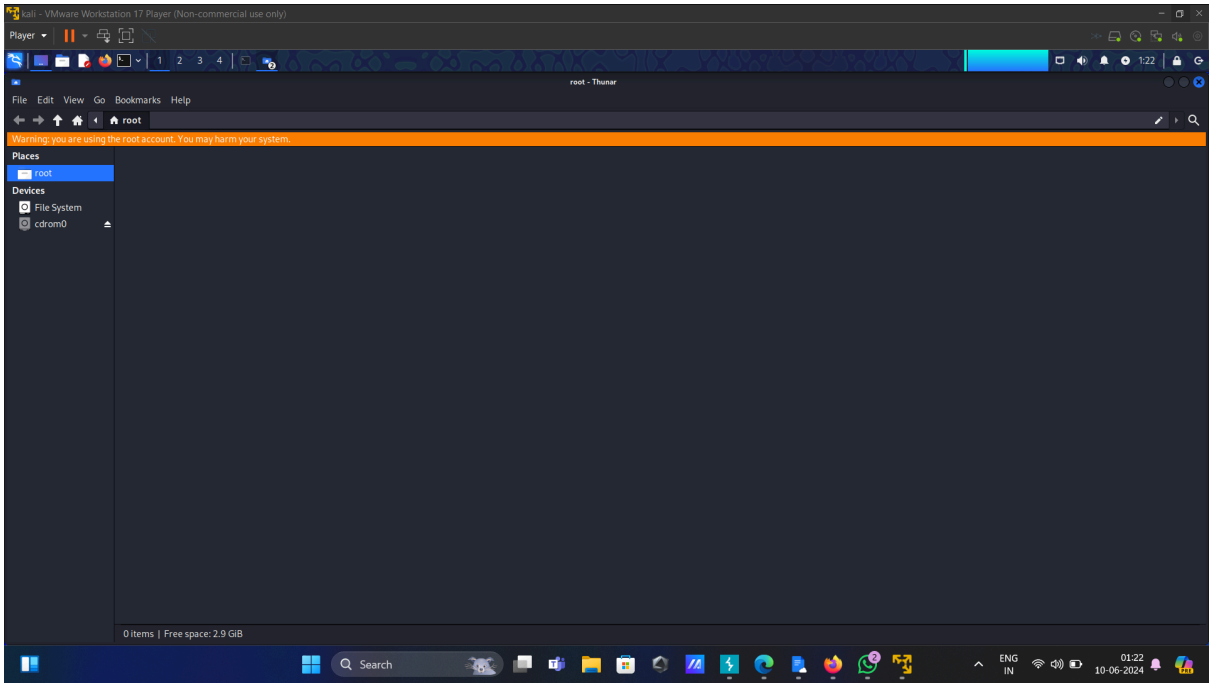
9) select root folder in file system.



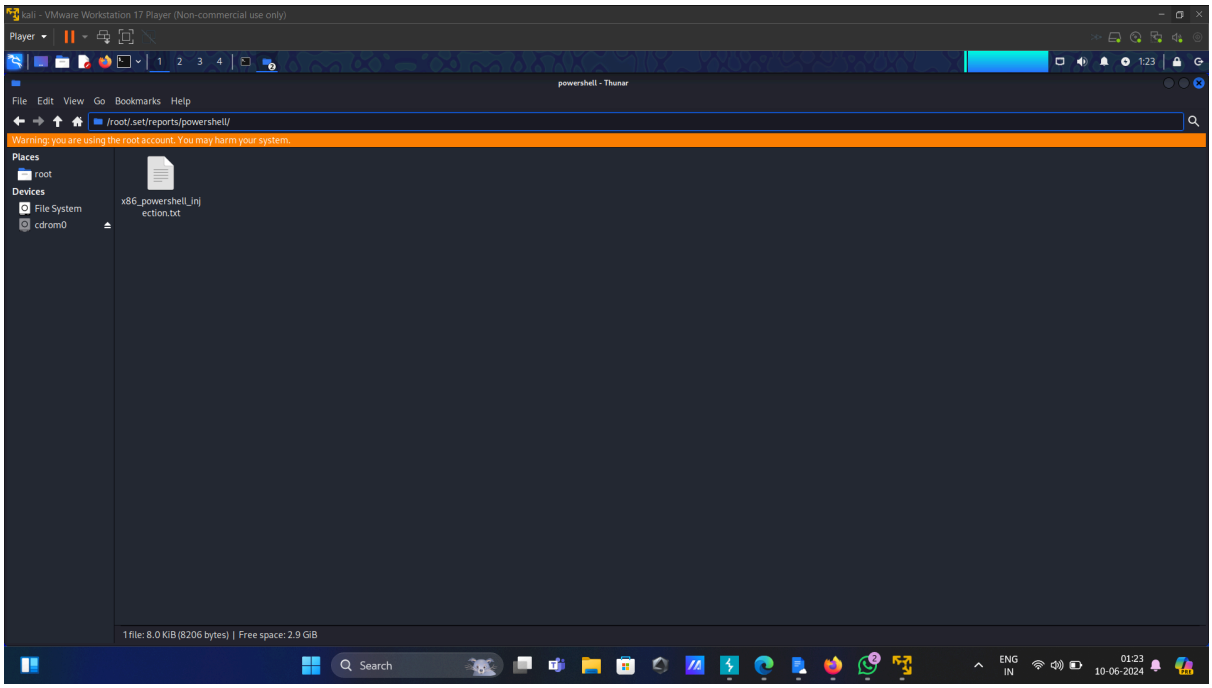
10)Give autentication passowrd to get root access.



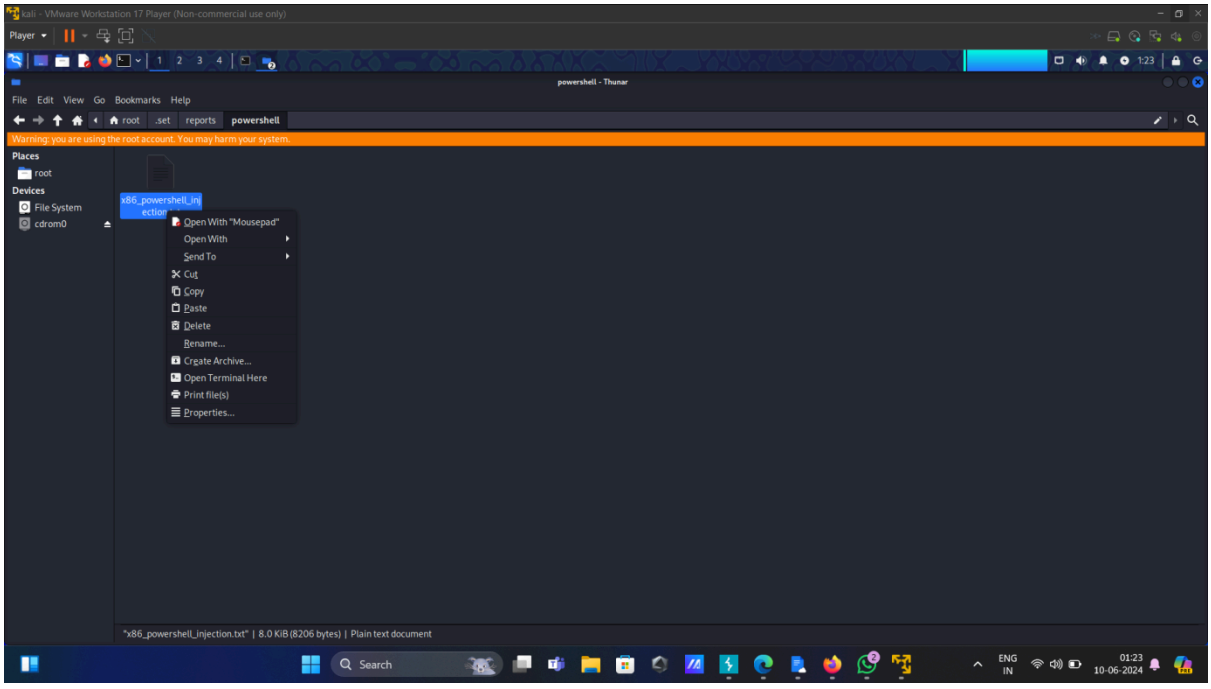
11) After that we will see this interface.



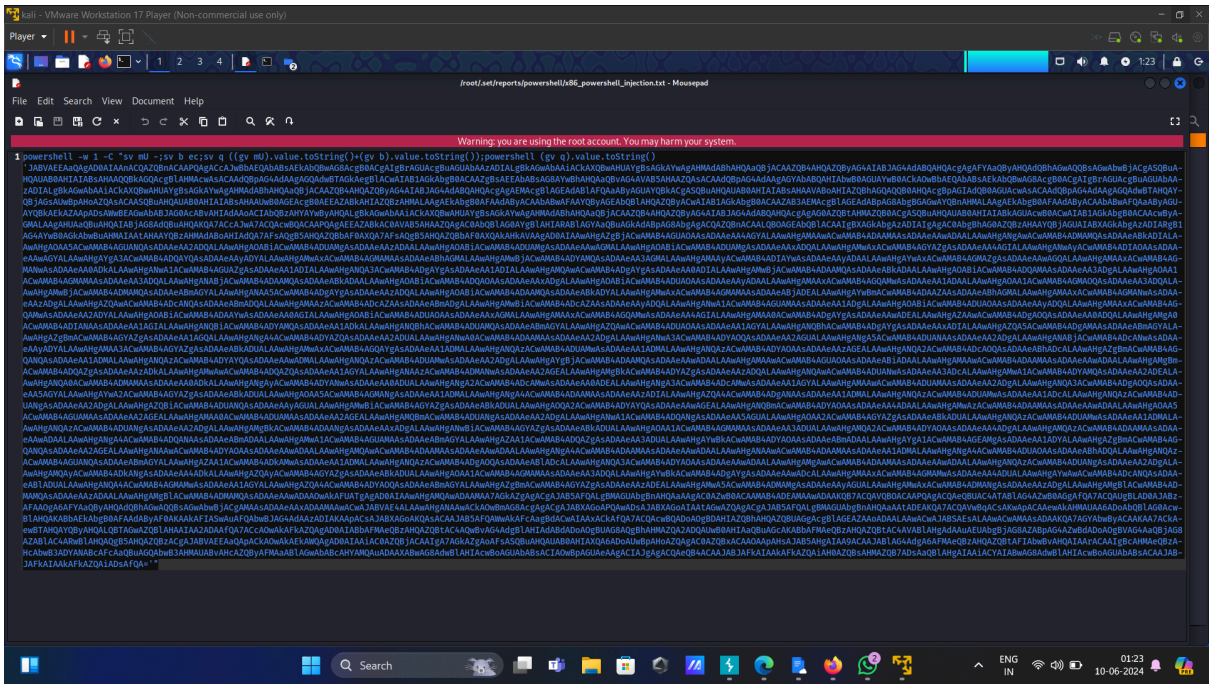
12)Enter path in search which we get in payload creation.



13) Open that file in notepad.

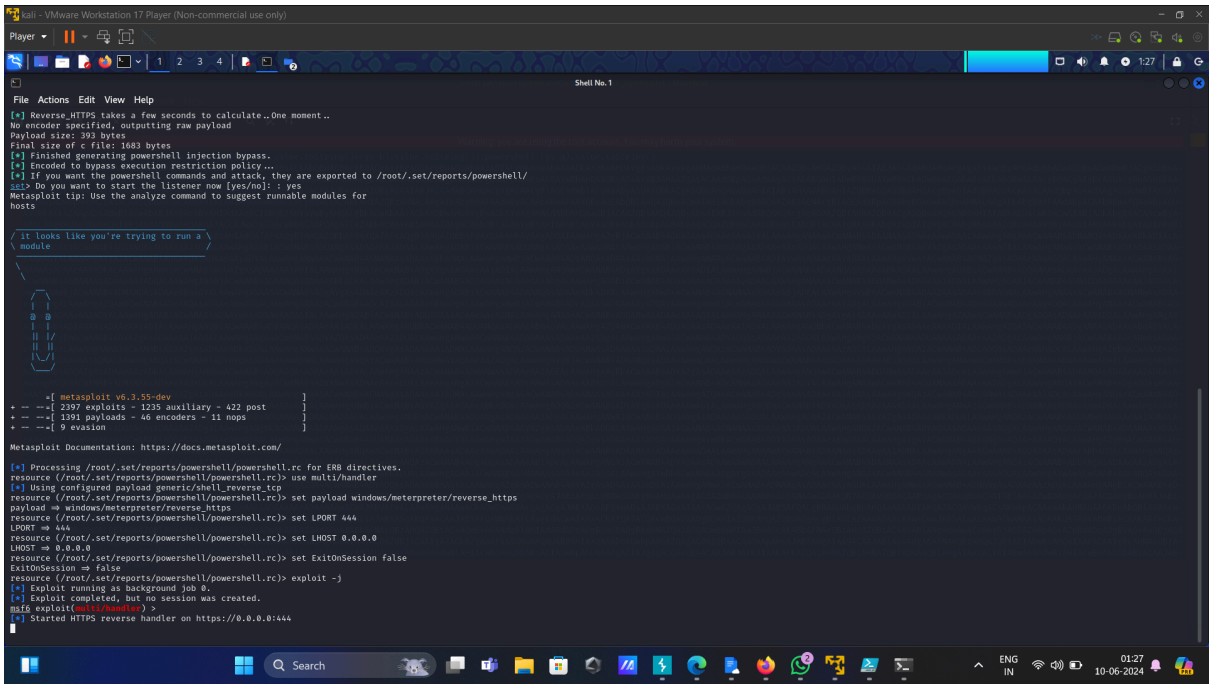


14) Copy that payload which display in notepad.



15)After that we will inject that payload in powershell of victims device.

And after that we need to set listener.



16) Then we can control victims device by passing commmands.

IMPACT ASSESMENT:

Creating a backdoor allows attackers to take control of a victim's device. Through this access, attackers can perform various malicious activities, such as using the device for criminal purposes, stealing sensitive information, causing financial losses, and more. A backdoor in the operating system can grant attackers complete control over the victim's device.

MITIGAION:

- **Powershell Antivirus :** The PowerShell antivirus offers advanced security technology to protect devices from backdoors like this
- **Avoiding unknow links:** Avoid clicking on links from unknown sources,

such as spam emails or messages without a country code

- **Keep systems private:** avoid sharing your system with others

CONCLUSION :-

Creating backdoors using PowerShell in operating systems is a potent method that attackers employ to maintain persistent access to compromised systems. PowerShell's integration with Windows, coupled with its powerful scripting capabilities, makes it an attractive tool for both system administrators and malicious actors. The process typically involves leveraging PowerShell's ability to execute commands and scripts that can manipulate system configurations, exfiltrate data, or communicate with command and control servers