# COMPLETE GRAMMERLY 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**

**D.UMA VENKATA REDDY(23KQ5A0514).** **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

# SQL INJECTION

## INTRODUCTION:

SQL Injection is a type of vulnerability that occurs when an attacker injects malicious SQL commands into input fields of an application to manipulate sql query. These commands are then executed by the backend SQL server, leading to unintended results that the application developer did not anticipate.

Sql injection is one of the powerfull technique in cyber - attacks due to its risk factor and its contains A1 priority in OWASP vulnarabilities. sql injection vulnarability occures when application does not sanitize the user input data.

## Risks that occures when there is a sql injection vulnarability:

- View private data or restricted data in database
- Add , delete or modury the data which is database
- Gain administrative access to the database.
- Compromise the server by using database as an entry point.
- Launch daniel-of-service attack or disrupt the database infrastructure.

## Types of SQL injections:

- **Blind sql injection**

  Blind SQL Injection is a type of SQL Injection attack where the attacker indirectly discovers information by analyzing 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.

- **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.

- **Union sql injection**

  Union sql injection is also one of the sql injection which exploit vulnabilities and manipulate sql query to retrive data/ sensitive data from other tables in database.

## METHODOLOGY :

**Tool:** Burp Suite

**Target Website:** Grammerly

## 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/*

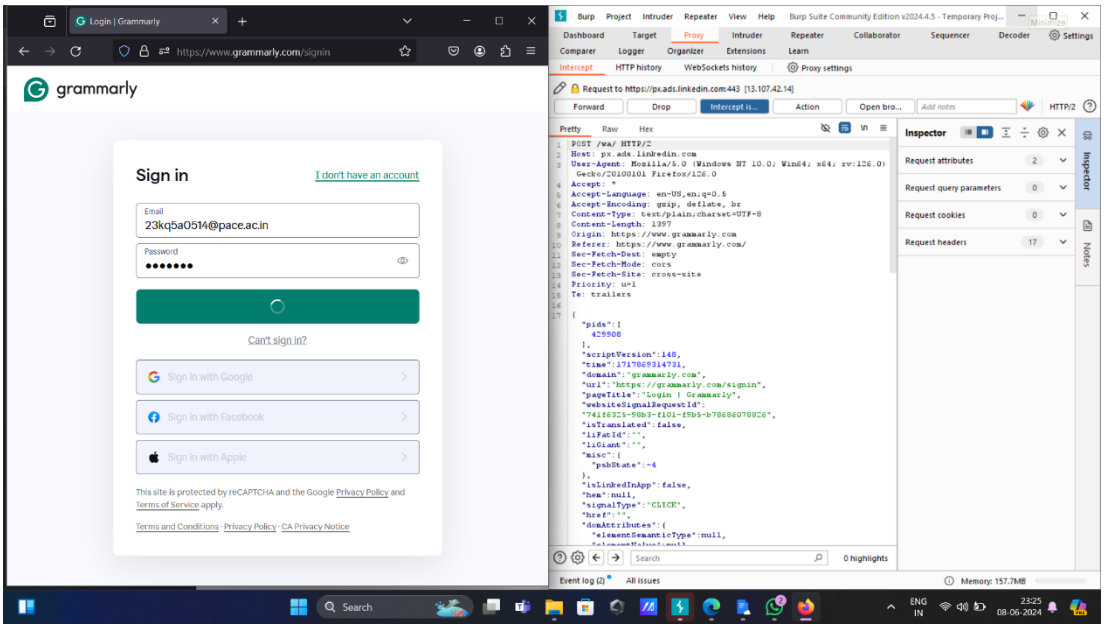
## Tool based testing:

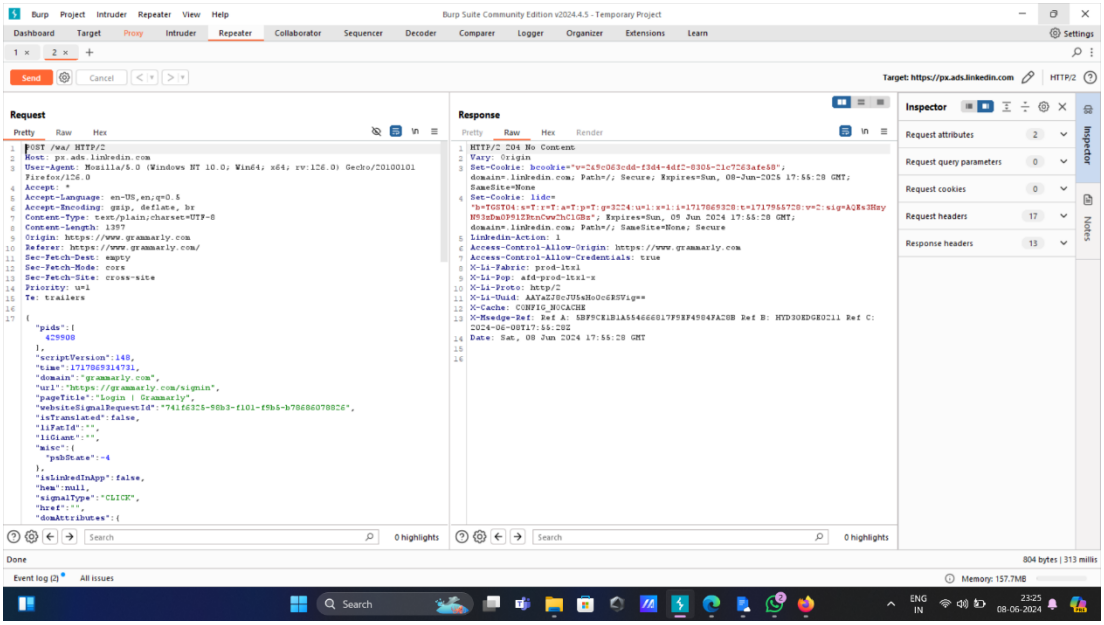
1) First open grammerly login page and give some random credentials and intercept it with burpsuite.

2) search for username in proxy request capture and modify it with blind sql payloads.

3) if there is a sql injection vulnarability then you will login into

admin panel by using this payload.



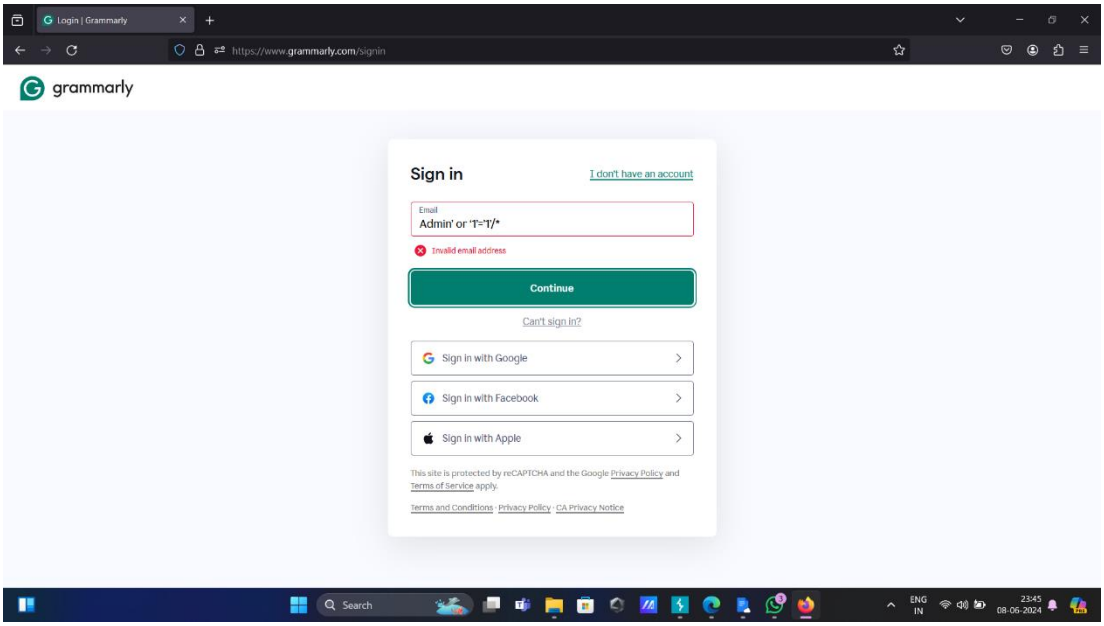

In the above example the captured request doesn't show username due to encryption and abstraction.

That means the login credentails are encrypted in this grammerly web application.

## Manual testing:

1) first find the login page in grammerly web application

2) insert payload in username field and try to access the admin panel.




In the above example the username input field have input validation and also dynamic username verification. that's why performing the manual sql injection attack is impossible.

## Error-based sql injection:

Payload used:

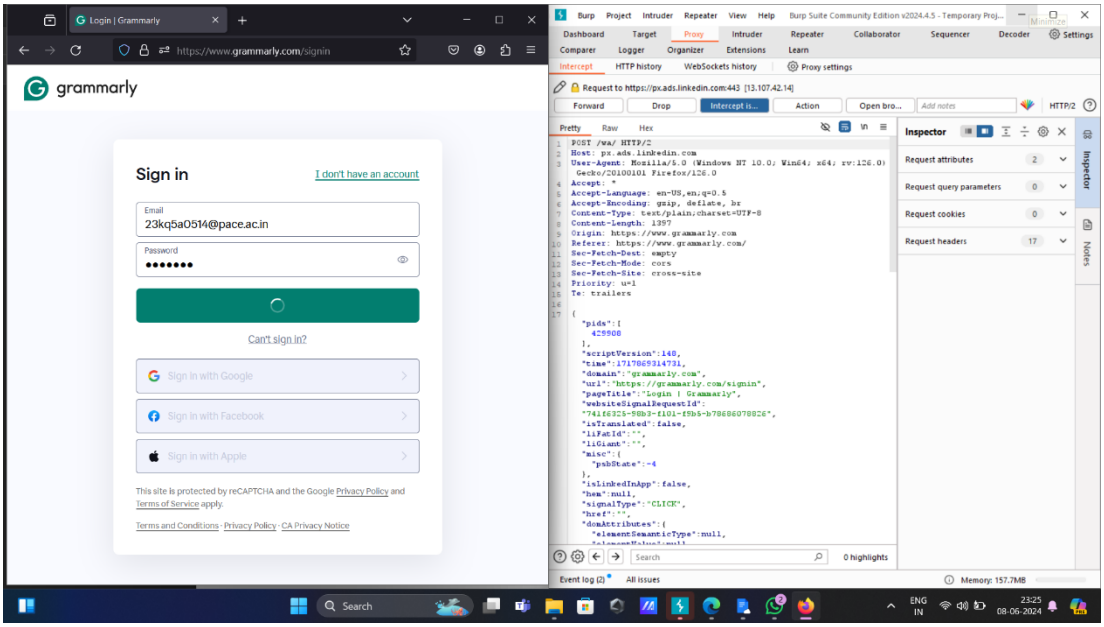
' OR '1'='1

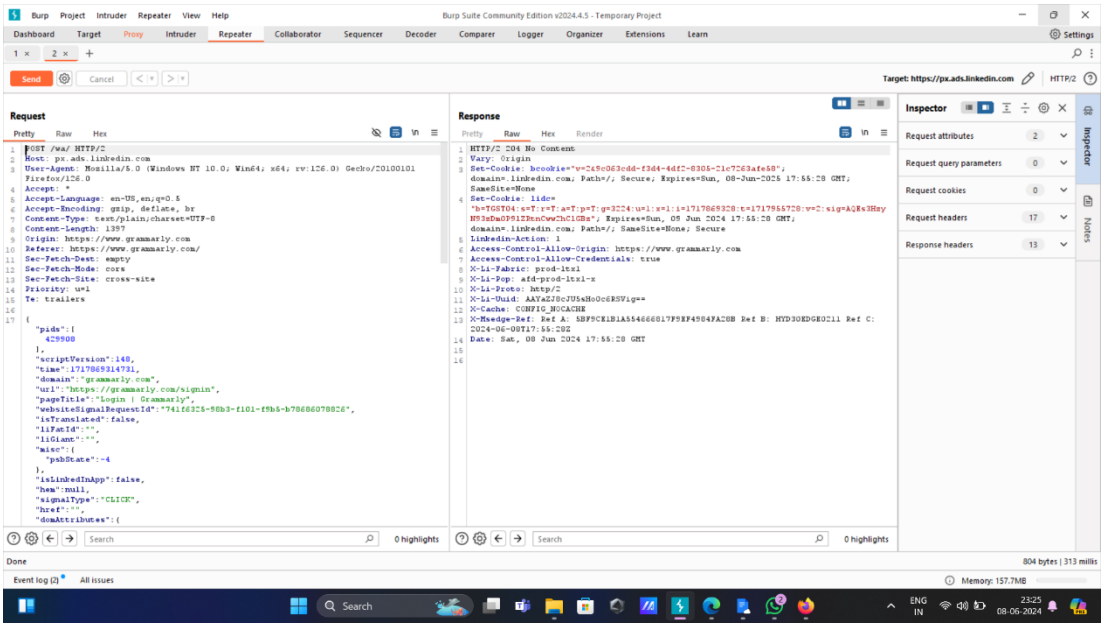
## Process:

Tool based:

1) First enter random credentials in input panel

2) Intercept the login request and modify the username field with error- based payload.



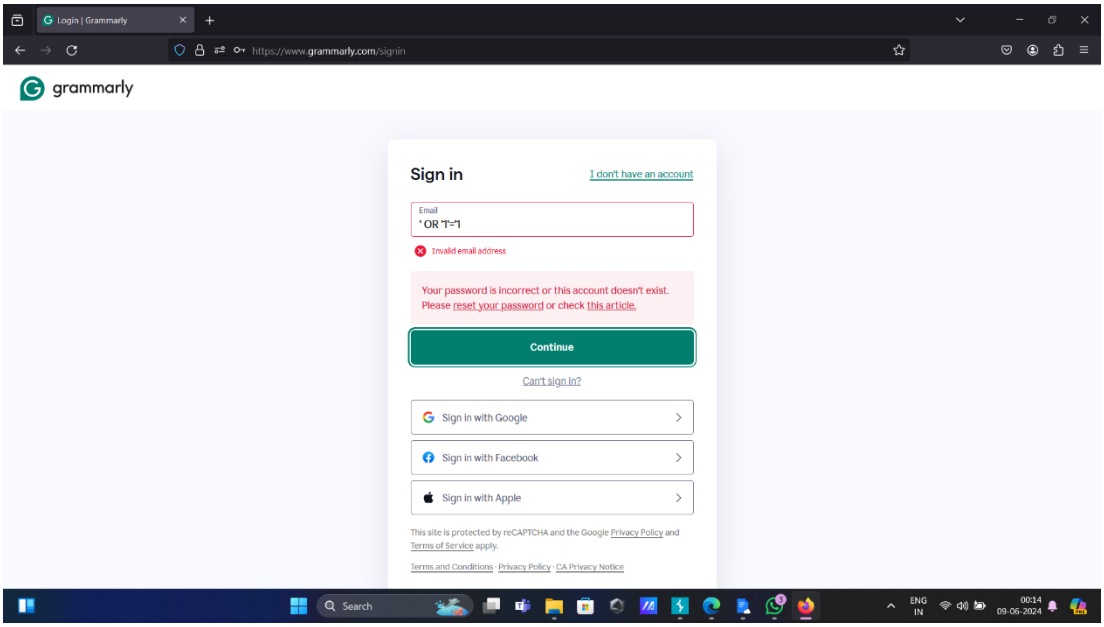


In the above example the username and password fields are encrypted so we cant modify the username to inject payload.

Manual testing:

1) First open the login page in grammerly application

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




In this example the input field have input validation and sanitization so we cant perform error-based injection in 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--

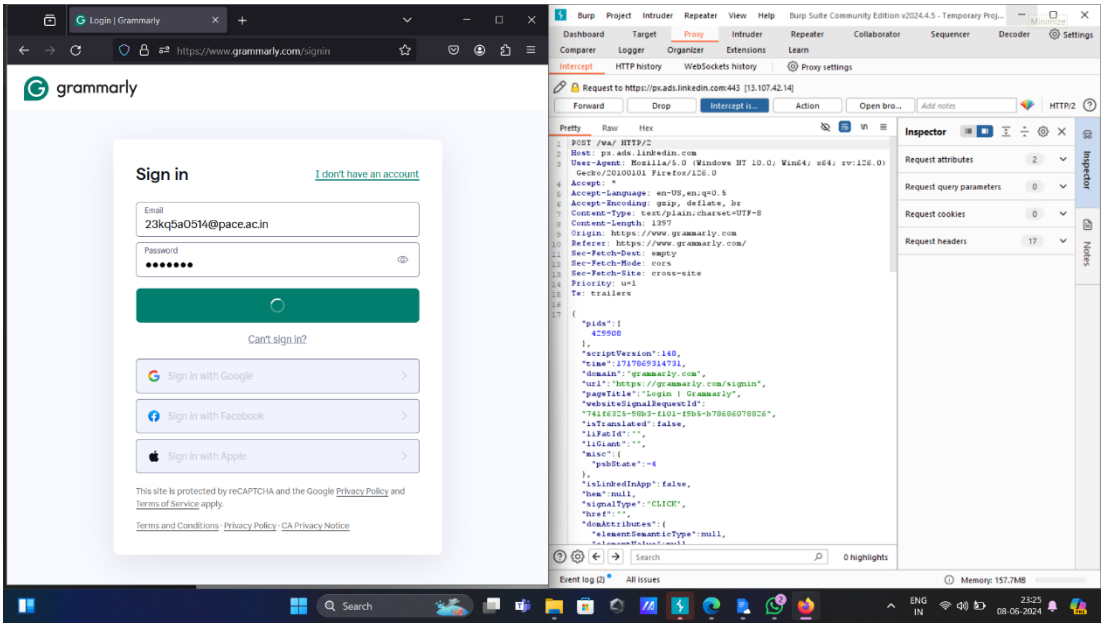
' UNION SELECT NULL,NULL,NULL,'a'--

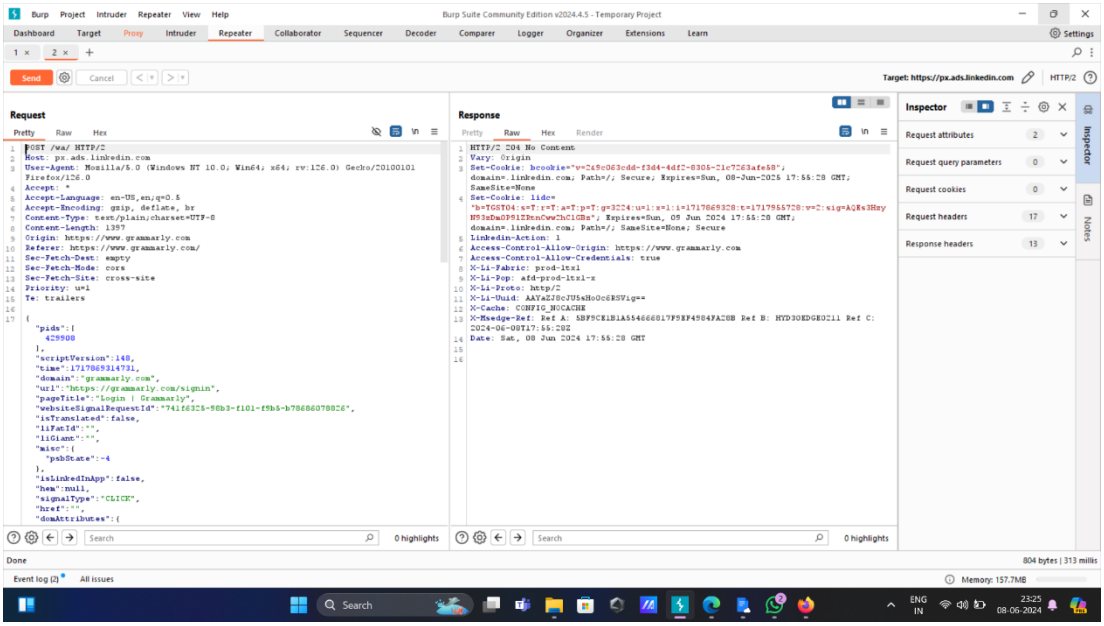
## PROCESS:

Tool based testing:

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

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



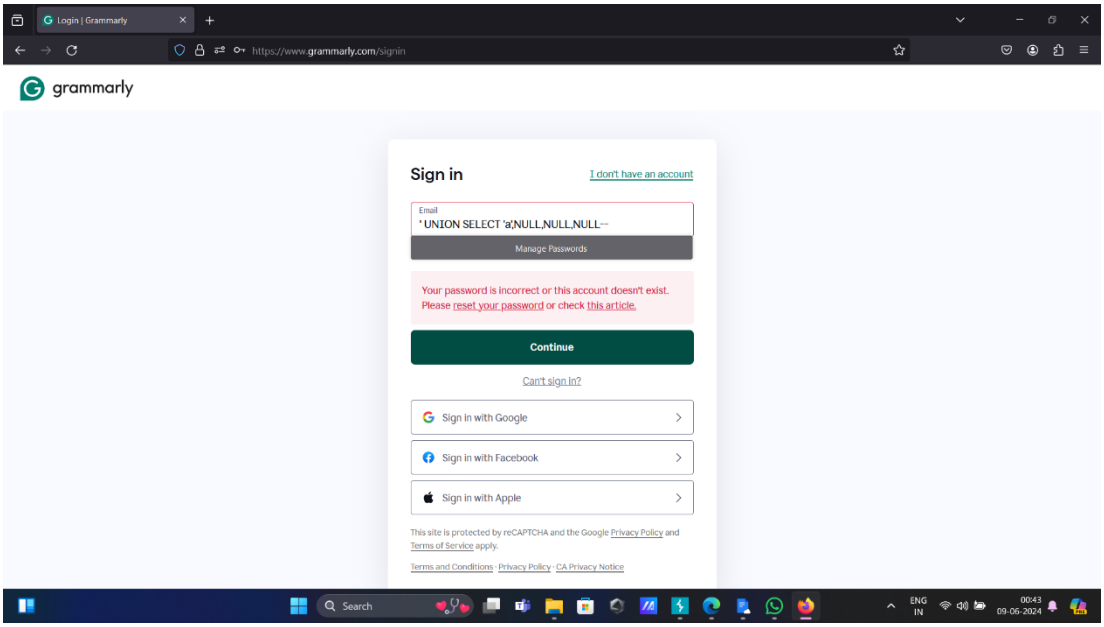


In the above example the captured request have data encryption and abstraction that's why the input field data is invisible. so we cant modify credentials data with payloads to perform Union sql injection attack.

Manual testing:

1) First we need to open login page on grammerly

2) we should enter payload in username field instead of email.




In the above example the username input field have input validation and input sanitization.so we cant perform union sql injection attack on this wep page.

## IMPACT OF SQL INJECTION:

When we have a web application with this sql injecion that means that web application doesn't safe. if we stored lots of login credentials in database this vulnarability allowa attackers to steal them and misuse them.

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

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

**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....

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

**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.

**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.

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

## MITIGATION:

**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.

**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 charecters, no special charecters etc..... These rules helps to validate the data and helps to detect injection queries.

**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.

**Penetration Testing:** Conduct regular penetration testing to stimulate real-world attacks and identify SQL injection vulnerabilities Fix any identified issues promptly.

**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 vulnarability 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 vulnarabilty and also demand for ransome that means through this vulnarability organization may face financial loss and also business loss as well.

# DIRECTORY OR PATH TRAVERSAL

## INTRODUCTION:

The directory OR path traversal is siviour vulnarabillity that cause to make web applicaation vulnarable. As its name the directory means a folder or collection of files.

Some web application have lots of url like login page url , home page url etc..... These all urls are gathered at one place called as directory url. if we find that directory url then we can clearly analyse entire application architecture. Also we can get sensitive data like logins, sensitive data etc.... this vulnarability crucial for web applications.

## METHODOLOGY:

This assesment or the directory traversal vulnarability in the grammerly web application. This vulnarability assesment provides complete path traversal analysis in grammerly like following process

### Reconnaissance:

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

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

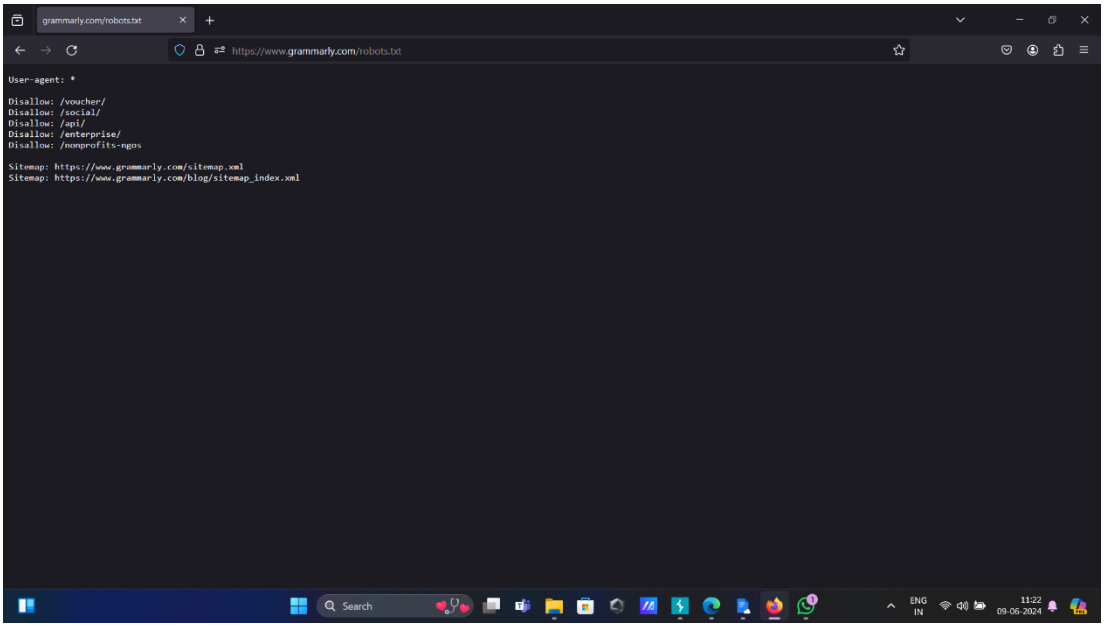
## PROCESS:

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 grammerly home page url and try directory paths after grammelry like www.grammerly.com/path.

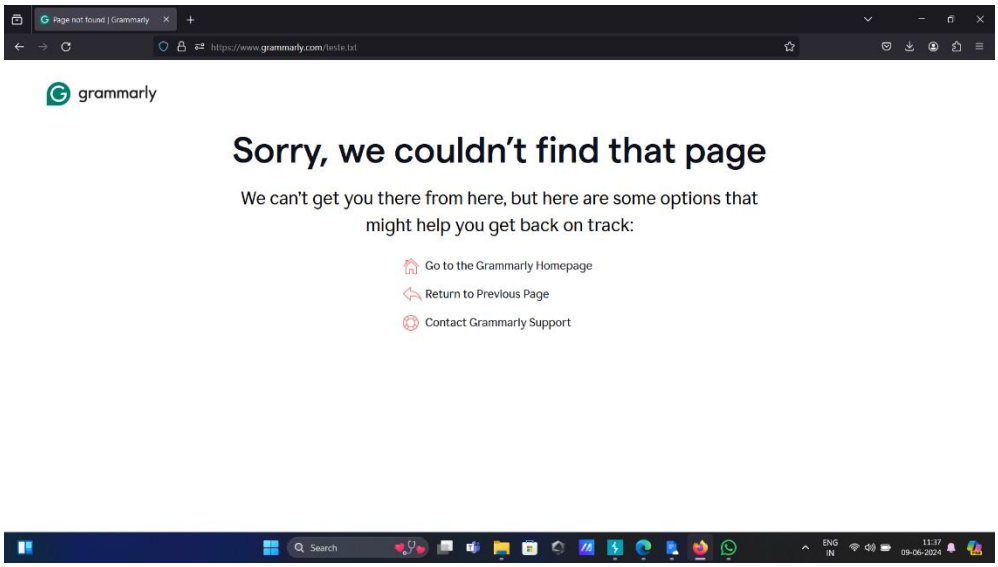



As shown in above screenshot the robots.txt in grammerly redirect us to site map urls directory. this url have some site map urls that may contain some data or we can redirect to some other pages using these sub urls .

**Teste.text:**

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

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



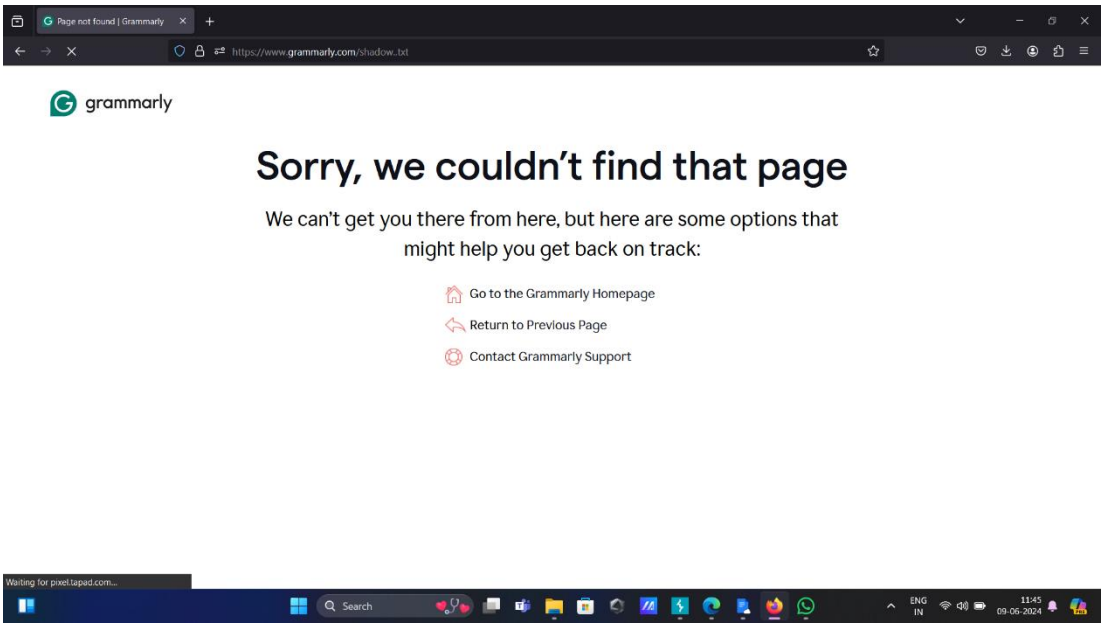
As shown in above screenshot the grammerly application servers doesn't have any teste.case design directory to visit.that means they always keep it confidential that's why they didn't create this directory.

**Shadow.txt:**

The shadow.txt path show 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 vulnarabilty to give access to get oether  login credentials . its like siviour vulnarability in web applications

1) first open grammerly home page and try to insert shadow.txt path after home page url.




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

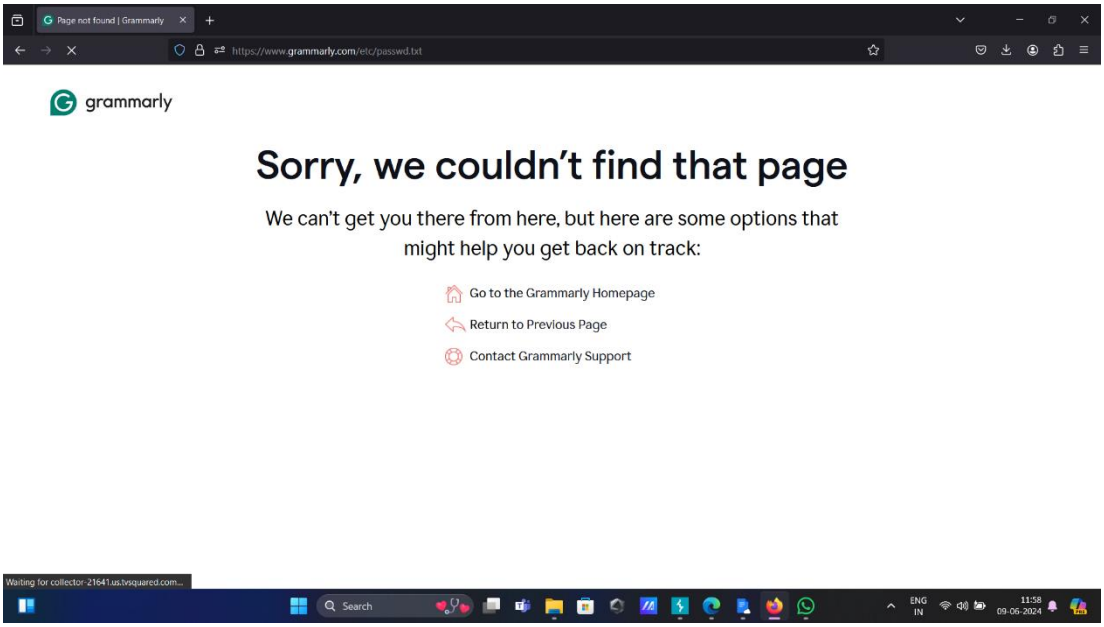
**Etc/passwd.txt:**

The etc/passwd.txt directory provides passwords to

usernames.that means if web application have this directory then it is a vulnarability which leads to login data breach.

This vulnability cause to siviour security fault which allow attackers to steal confidentail data like others usernames and passwords.

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




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

If incase there as directory like this that means the cofidentail login credentails will be theft.

## VULNARABILITY DESCRIPTION:

The directory or path traversal vulnarability is one of the major vulnarability that most of attackers use it to acces the confidentail and sensitive data of application.

This vulnarabilty have A04 level siviority in OWASP. so this vulnarability give opportunity to steal data and iot leads to financial loss for organizations.

The path or directory traversal vulnarability also show the way to

gain full access control of application without permission.

## IMPACT OF THIS VULNARABILITY:

The path or directory traversal vulnarability have high impact on application organization like it allows attacker to steal confidentail data and senstive data from application servers.

The impact of this vulnarability will be like this:

- Data breach or data steal
- Sensitive data exposure
- Others private data exposure
- Loss Both low level and high level login privilizes
- Organizations will loss control on their application
- Financial loss
- Demand for ransome
- 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 vulnarabilities and correcting them leads to make application vulnarable 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 vulnarability that controls access trafic and application and sanitize the accessing flow to keep application safe and secure.

## Conclusion:

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

# CROSS SITE SCRIPTING(XSS)

## INTRODUNCTION:

The cross site scripting is web application vulnarability that occures when attackers inject malicious code which is developed by HTML and Java script into input panels in web applications.

This cross site scripting also called as xss, it allows attackers to inject data into web applications and it will display in victims browsers.that means the injected data will also display in others browsers when they visited attacked application.

**Types of xss:**

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

- Reflected xss:

  The reflected cross site scipting is a xss vulnarability type that used to inject xss script into aplictions which is present for short time only.that means after some time the reflected scrpit will be remove automatically or it will remove after refreshing the page.

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

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

- Stored xss:

  The stored xss is also a cross site scripting vulnarability type that used to inject xss script into applicatiions which will be stored in application databases.

  The injected data will be store in application permanently and it will romove 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 vulnarability types which

allows attackers to inject scripted injection into applications.unlike the other types of xss the DOM- based xss occures due to modification in the client-side code rather than the server-side response doesn't change it self. But ythe client-side code executes differently due to malicious DOM manupulation.

## METHODOLOGY:

The goal of this assesment is to find xss vulnarabilities in grammerly web application.

For this I am using grammerly web interface and xss scripting to make attack operations.

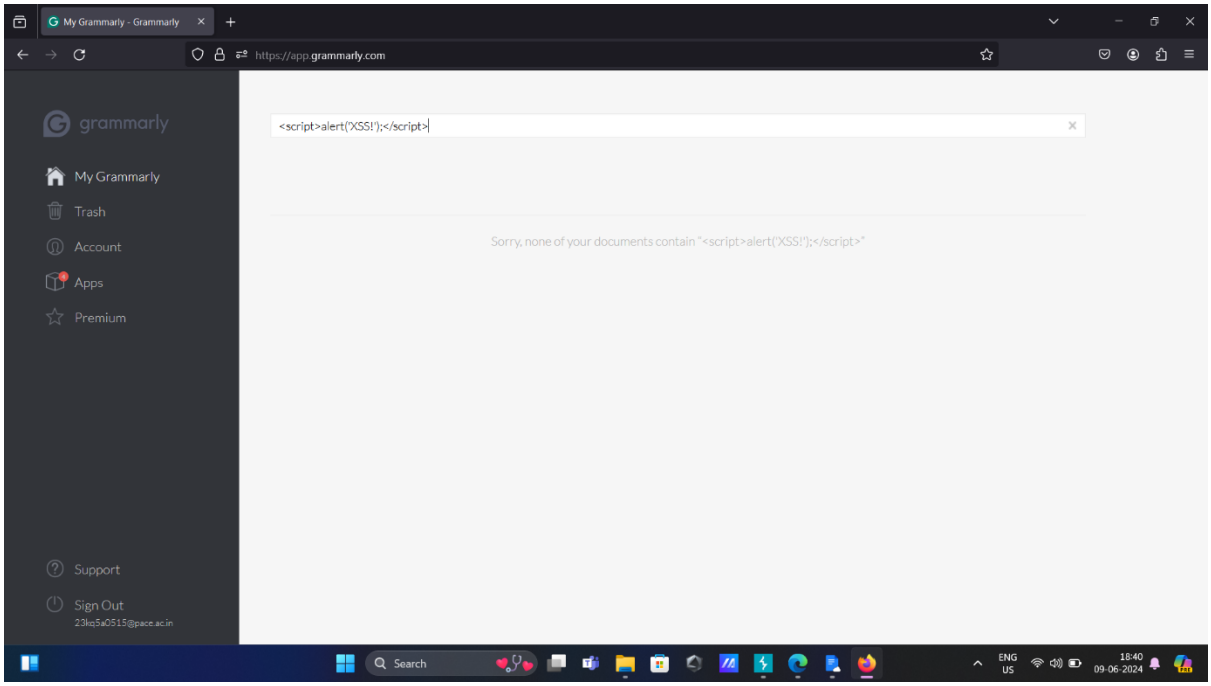
## PROCESS:

Reflected xss:

The reflected xss is one types of xss vulnarability 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 grammerly 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 deos not working on grammerly web application due to high security and dynamic output.

The grammerly doesn't have any xss vulnarability .

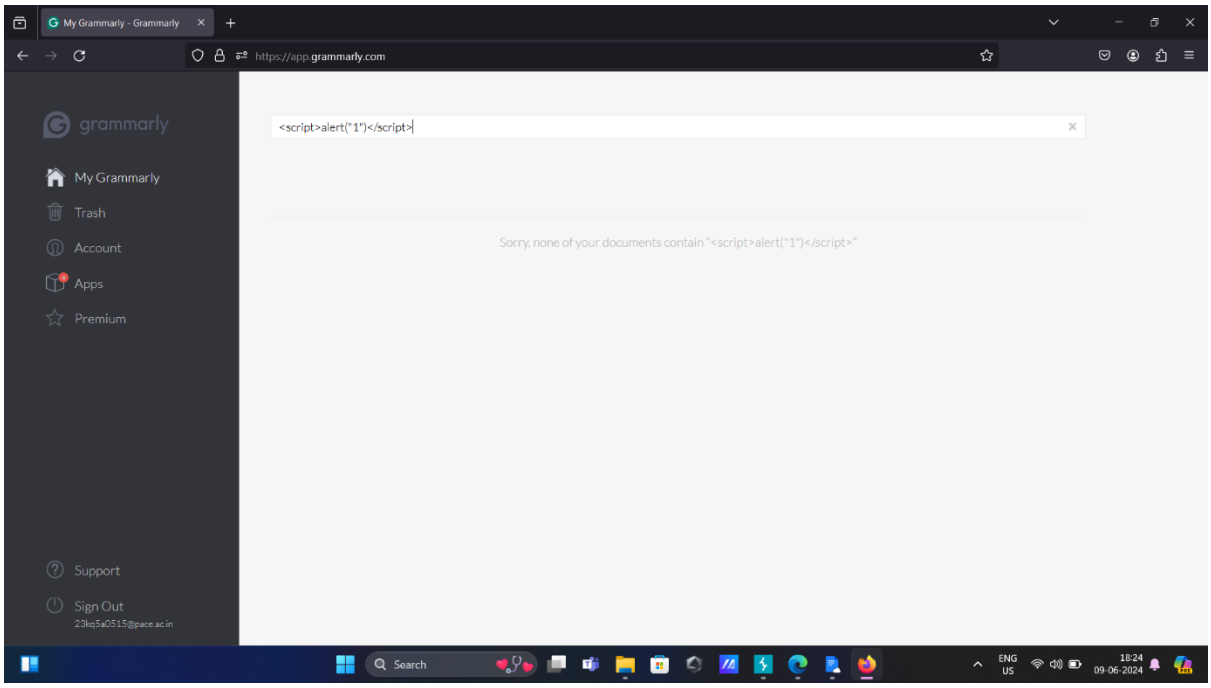
Stored xss:

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

1) first open grammerly 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>

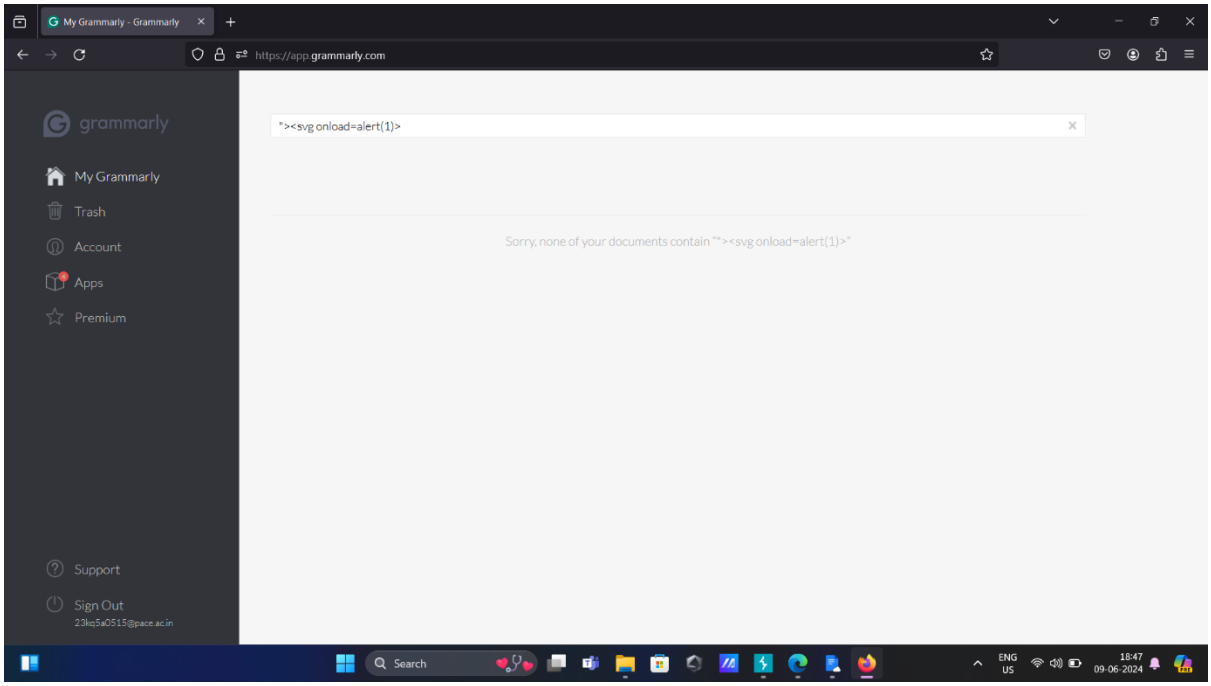



As shown in above screenshot the grammerly application does not have any stores xss vulnarability. That means the grammerly have high

security and dynamic query output that's why stored xss injection is difficult in this application.

DOM-based xss:

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




As shown in above screenshot the grammerly doesn't have any vulnarability because that application have high secure and dynamic wuery output. that's why the grammerly application doesn't have any vulnarabilities.

## IMPACT ASSESSMENT:

The xss vulnarabilities shows a way to attackers to inject malicious script and dengerous malwares. Through this vulnarability 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 vulnarabilty can leads to siviority to application and also users as well.

## MITIGATION:

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

## CONCLUSION:

The xss vulnarability is a sivior vulnarability that causes to inject malicious code in servers and dangerous files/docs into servers. through this the attackers can gain acccess to device conrrol and also application control.

# IDENTIFICATION AND AUTHENTICATION FAILURE

## INTRODUCTION:

Authentication and identification failure is a vulnarability that allows attacker to gain login access into servers .

The attackers can also gain admin access through this vulnarability

And it will helps to attackers to gain full control application

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

- Brute force vulnarability
- 2FA broken vulnarability

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

## METHODOLOGY:

This assesment admis to identify the identification and authentification failure vulnarability and exploit it with brute force attack and 2FA broken. For this we are using burp suite tool and grammerly.

## BRUTE FORCE :

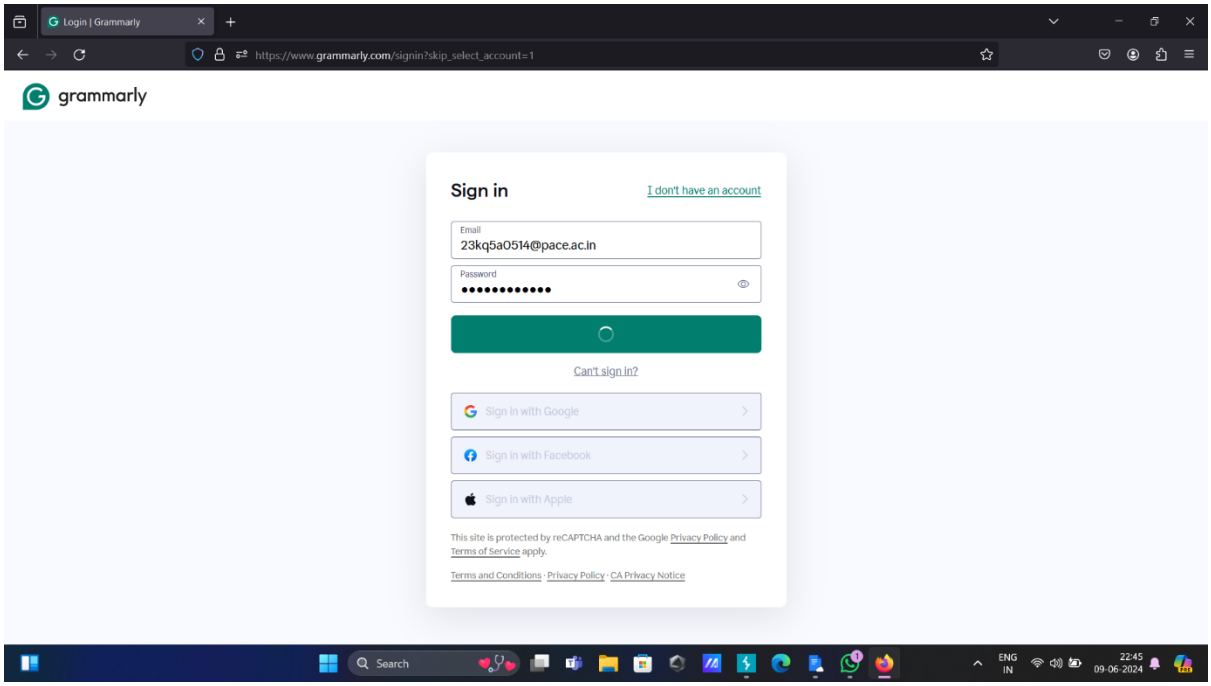
### PROCESS:

Brute force attack is an attack which makes using a authentication failure vulnarability. 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 them one input panel by using burp suite tool.

**Steps:**

1) Open grammerly website login page and give some random credentials in input feild



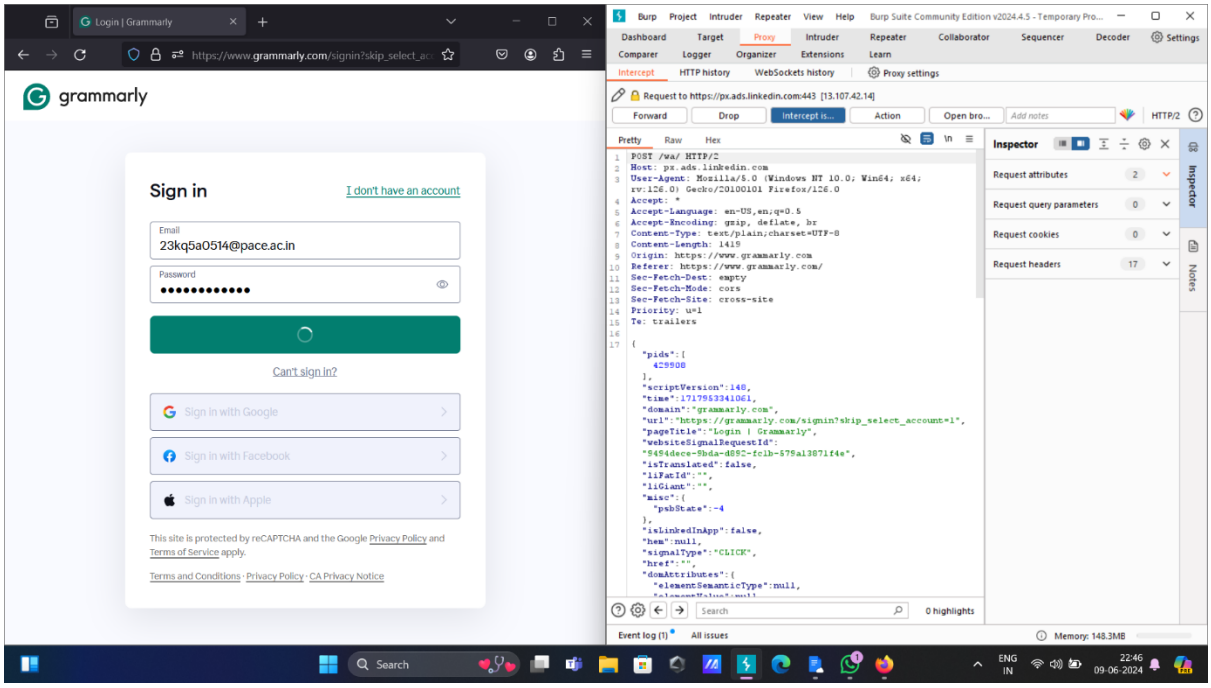

2) Intercept the login functionality with burp suite.

3) Find the credentials which gaves 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 fount 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 grammerly 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 remidy

technique for brute force attack, the attempt limitation limitate 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 remidy 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 expire that means the login page doesn't keep as longer, it will expire after particular time.

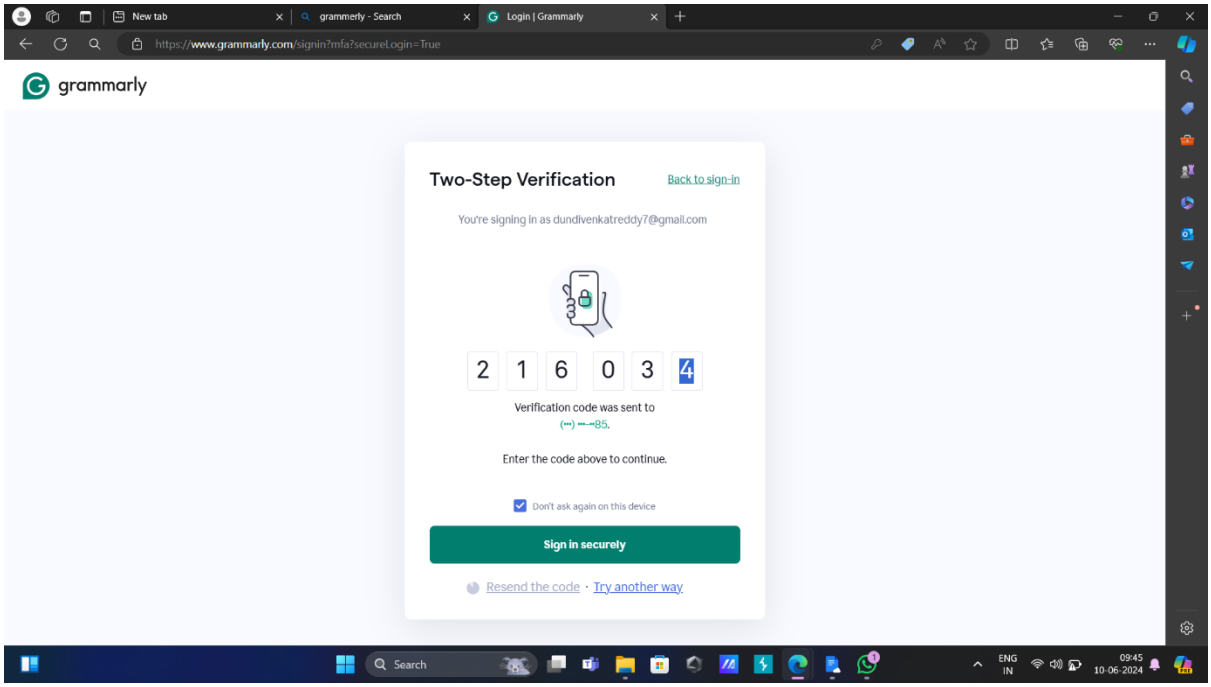
## 2FA BREAKING:

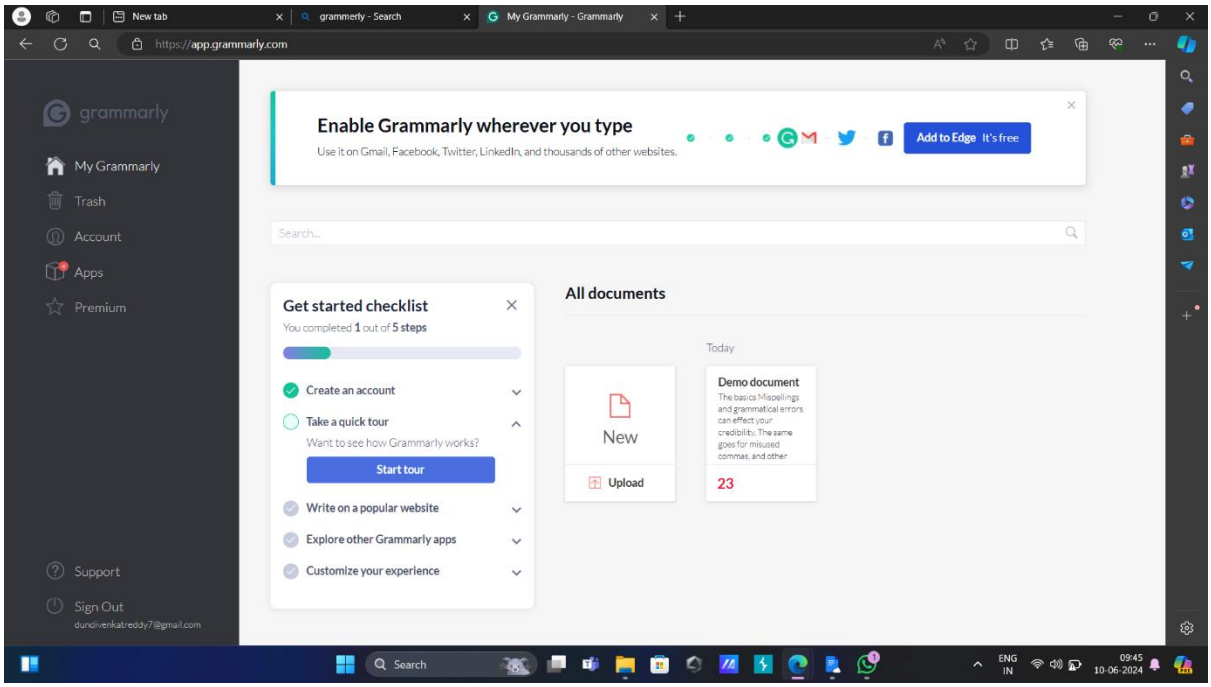
### PROCESS:

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 permmision from account holder, but in this 2FA breaking that security acceptence is no need, why because the attacker an bypass that page using this identification of authentication vulnarability.

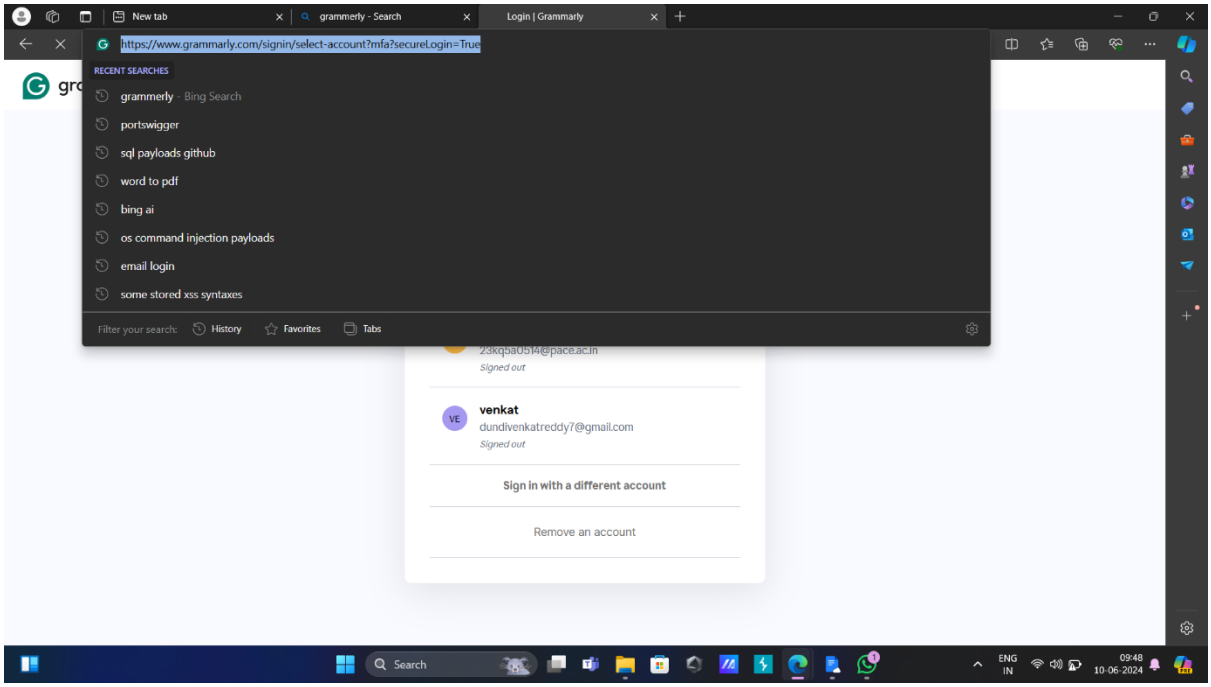
1) open grammerly web application two factor authentication and login to an account with 2FA code.

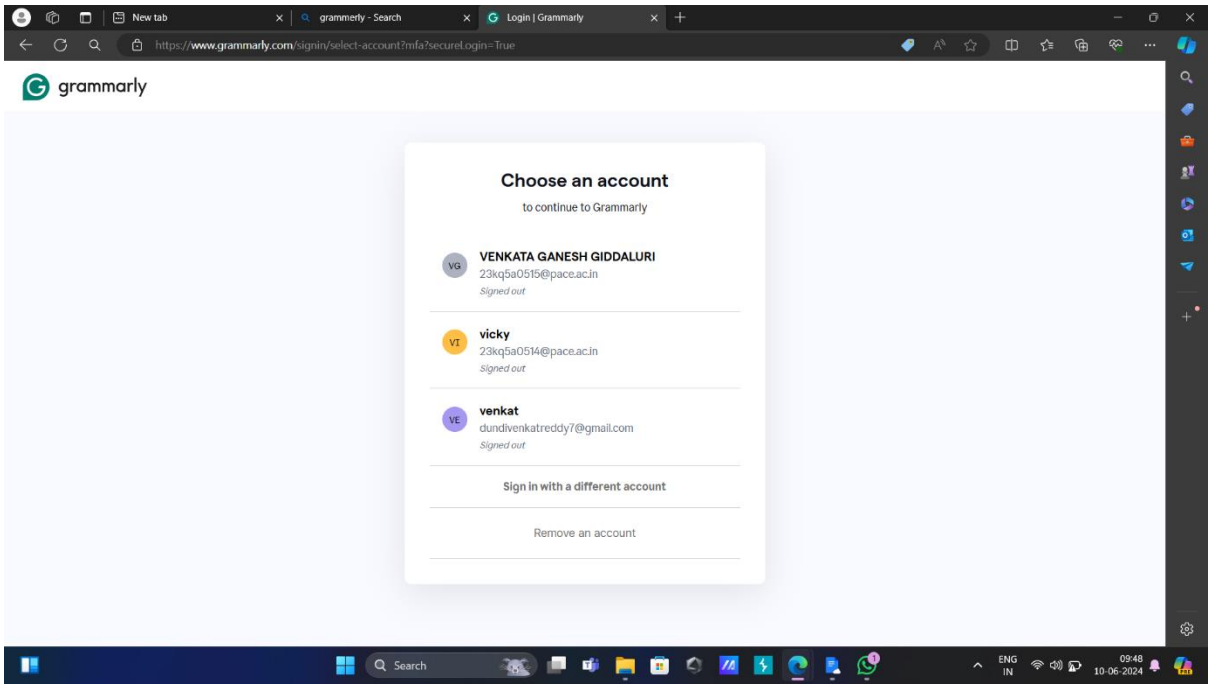



2) Find the login access parameter in url and copy it.

3) Now try to login into another account and in two factor authentication page charge parameters of url. Replace it with access granted parameter.


New tab
grammerly - Search
Login | Grammarly
https://www.grammarly.com/signin/select-account?mfa?secureLogin=True
RECENT SEARCHES
grammerly - Bing Search
portswigger
sql payloads github
word to pdf
bing ai
os command injection payloads
email login
some stored xss syntaxes
Filter your search:
History
Favorites
Tabs
venkat
dundivenkatreddy7@gmail.com
Signed out
Sign in with a different account
Remove an account
Search


New tab
grammerly - Search
Login | Grammarly
https://www.grammarly.com/signin/select-account?mfa?secureLogin=True
grammarly
Choose an account
to continue to Grammarly
VENKATA GANESH GIDDALURI
23kq5a0515@pace.ac.in
Signed out
vicky
23kq5a0514@pace.ac.in
Signed out
venkat
dundivenkatreddy7@gmail.com
Signed out
Sign in with a different account
Remove an account
Search

## IMPACT ASSESSMENT:

This vulnarability is usefull to attacker to by pass two-factor authentication. thourgh 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 vulnarability by encrypting url in request process attackers doesn't have any way to change url to bypass 2FA.

- **Url paramenter sanitization:** The url parameter sanitization can also be as a mitigation to thi 2FA breaking like parameters are very crusial 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 vulnarability aims to allow attackers to steal senstive data like login credentials from server database and also it causes to steal admin level access to this application.

# SERVER-SIDE REQUEST FORGERY(SSRF)

## INTRODUNCTION:

SSRF vulnerabilities enable attackers to manipulate the application into making unauthorised requests to internal or external resources, mainly it leades to data exposure, unauthorised access, or other security threats.

The aim of this assesment is to identify and expose the SSRF vulnarabillity in grammerly web application.

## METHODOLOGY:

The goal of this assesment is to identify and exploit the SSRF vulnarability in grammerly web application using "file:///etc/passwd".

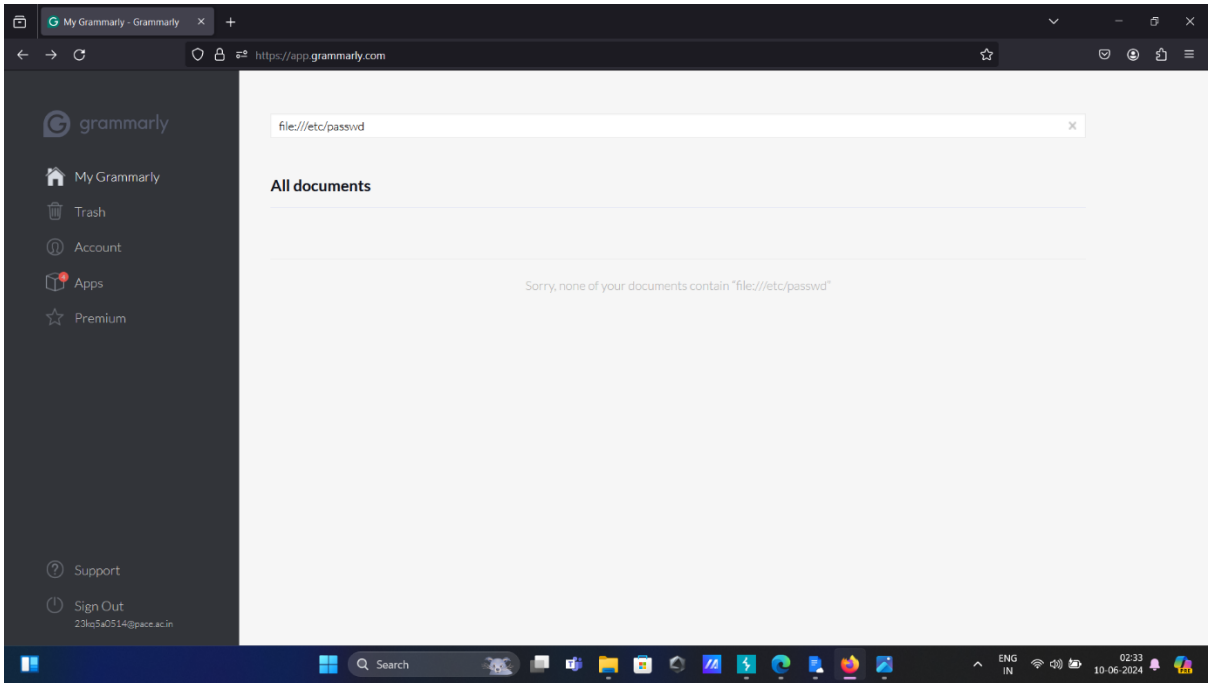
## PROCESS:

The ssrf if a vulnarability that can manipulate the request at server side that can be cause to security flaw .

1) open grammerly web application and loggin with your redentials.

2) open serach 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 grammerly application doesn't have any ssrf vulnarability to make ssrf attack. due to input sanitization. so this application is ssrf vulnarable free.

## IMPACT ASSESSMENT:

The SSRF vulnarability manipulate server 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 siviour 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 vulnarability 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:

The os command injection is a security vulnarability that occures when an attacker can inject arbitary query in applications.

The os command injection create a path to get sensitive content access or high level privilizes by injection malicious arbitary code into application via input panels.

## METHODOLOGY:

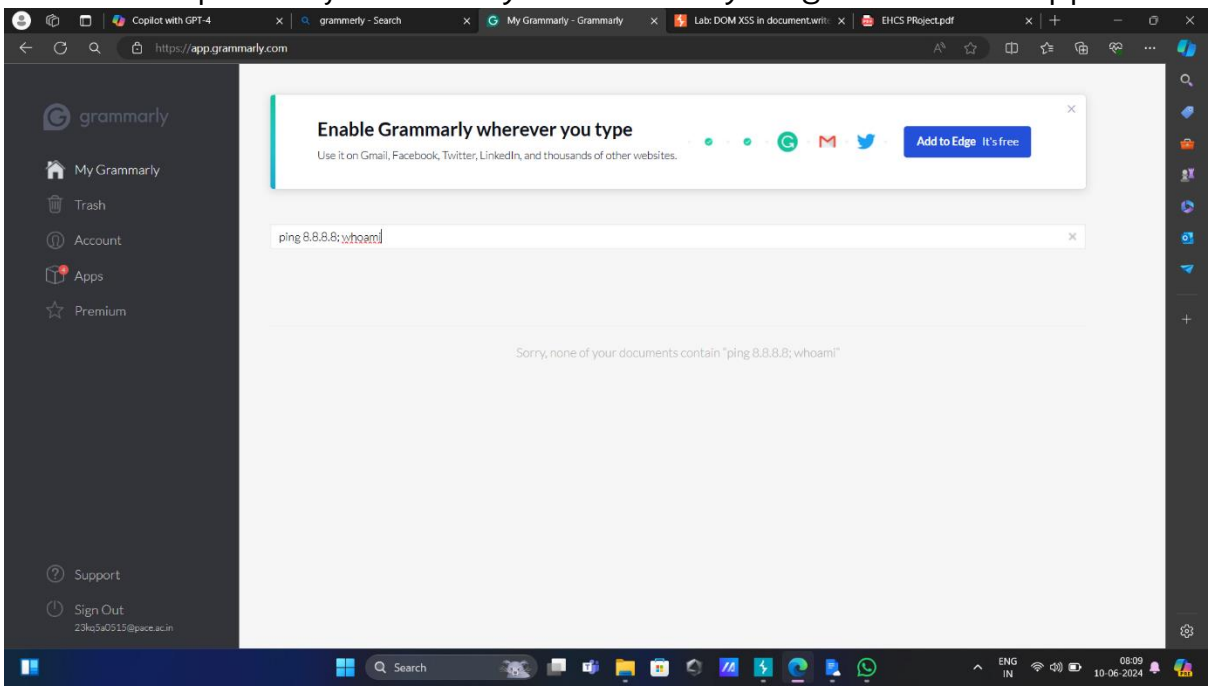
The motive of this assessment it to find os command injection vulnarability in grammerly.com by injection some malicious arbitary code into input panels.

## PROCESS:

The os command injection can give access to gain unortharized access of an application server by injecting malicious arbitary code in input panels.

1) open grammerly web application.

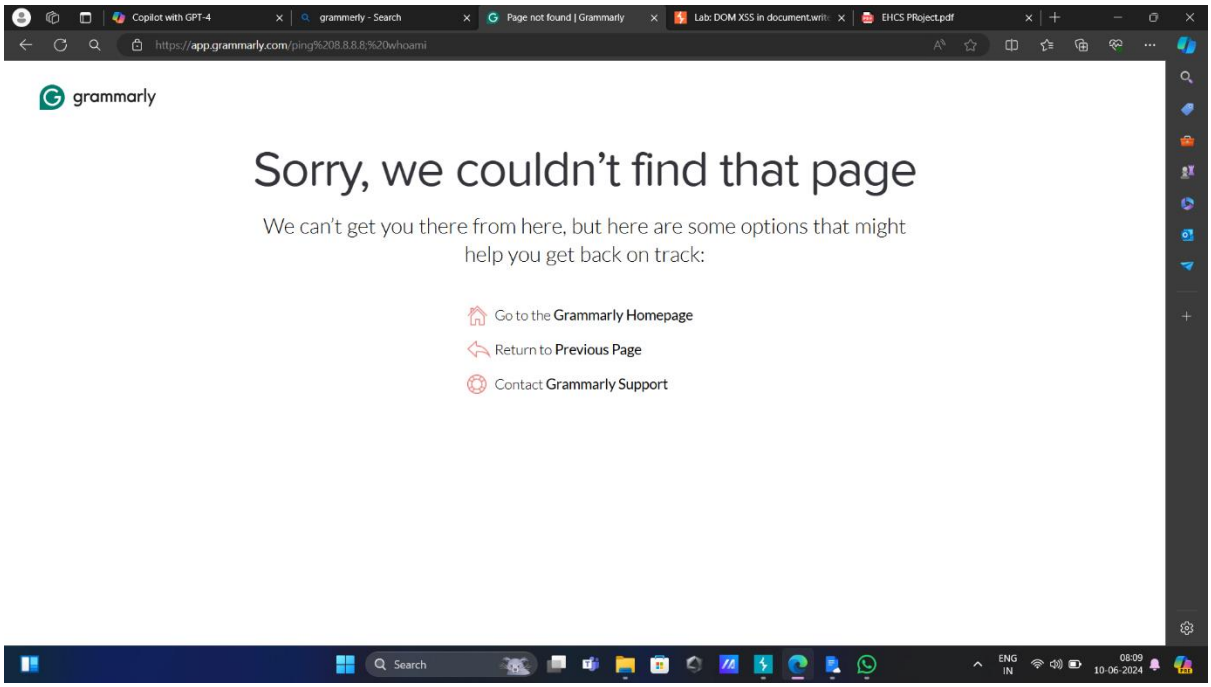
2)In search panel inject arbitary code and try to get access of application



As shown in above screenshot the grammerly web appliation does not accept arbitary malicious injections in input panels.

The os command injection deos 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 grammerly.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 siviour vulnarability that can give attacker to gain high level privilizes and access to sensitive data. That means the os command injection shows a way to attack application servers by injecting malicious arbitary 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.

- **Parametarized queries:** use parametarized queries policy when users interacting with database to prevent it from os command injection
- **Leat privilizes:** Run applications with low privilizes that can helps to make hard for attackers to gain admin level access.
- **Whitelisting:** Explicity allow only know commands like safe listed commands only.

# ADDITIONAL PROJECT

## BACKDOOR CREATION FOR OS POWERSHELL

### INTRODUCATION:

The back door creation of os powershell is a process that create a payload in kali linux by using social engineering toolkit.

In this process we will pass ip address of victims and port number which allows us.

The backdoor for os powershell can control entire device os by giving simple commands that means we can control entire operating system of victims by using this backdoor.

### METHODOLOGY:

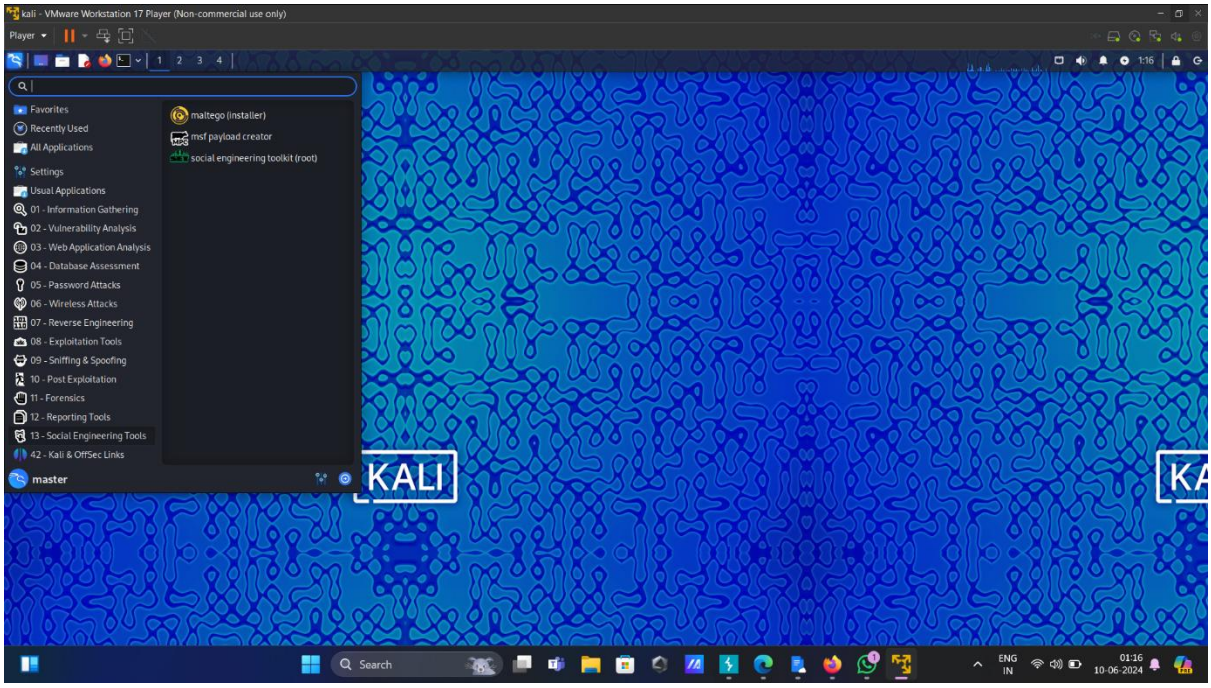
This backdoor creation can be created by attacker and through injecting it in victims powershell the attacker can control entire os of the victimes device.

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

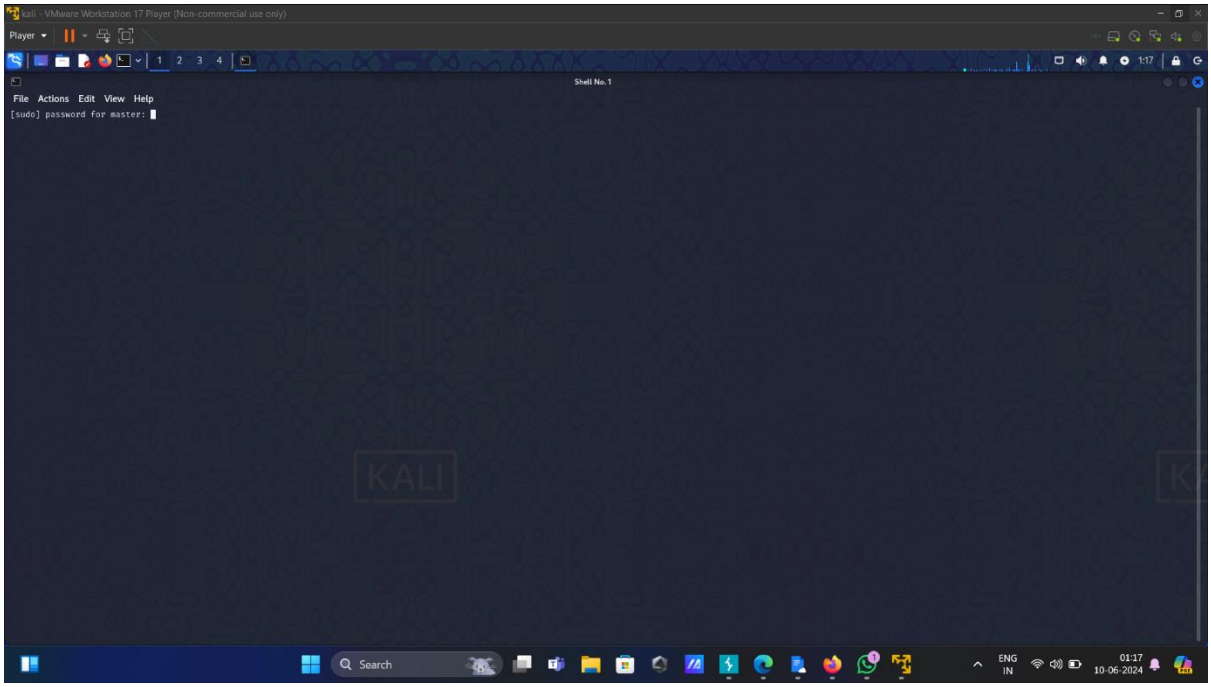
And also we use “kali linux” operating system.

### PROCESS:

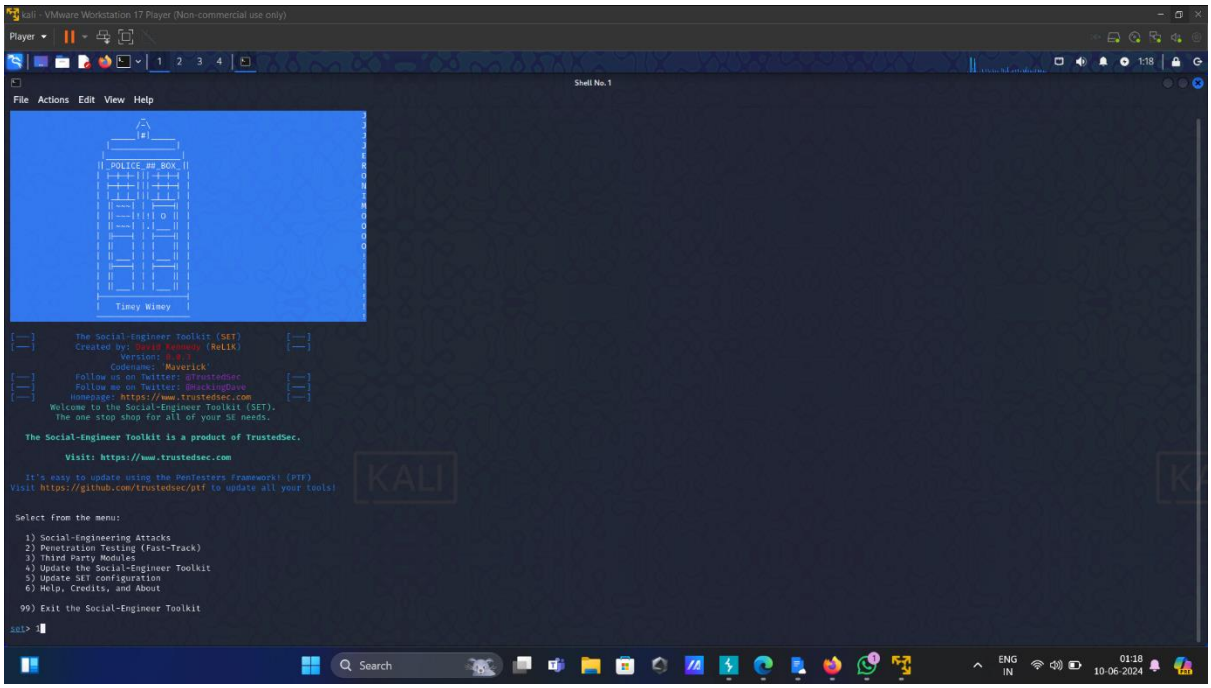
1) First we have to open social engineering toolkit.



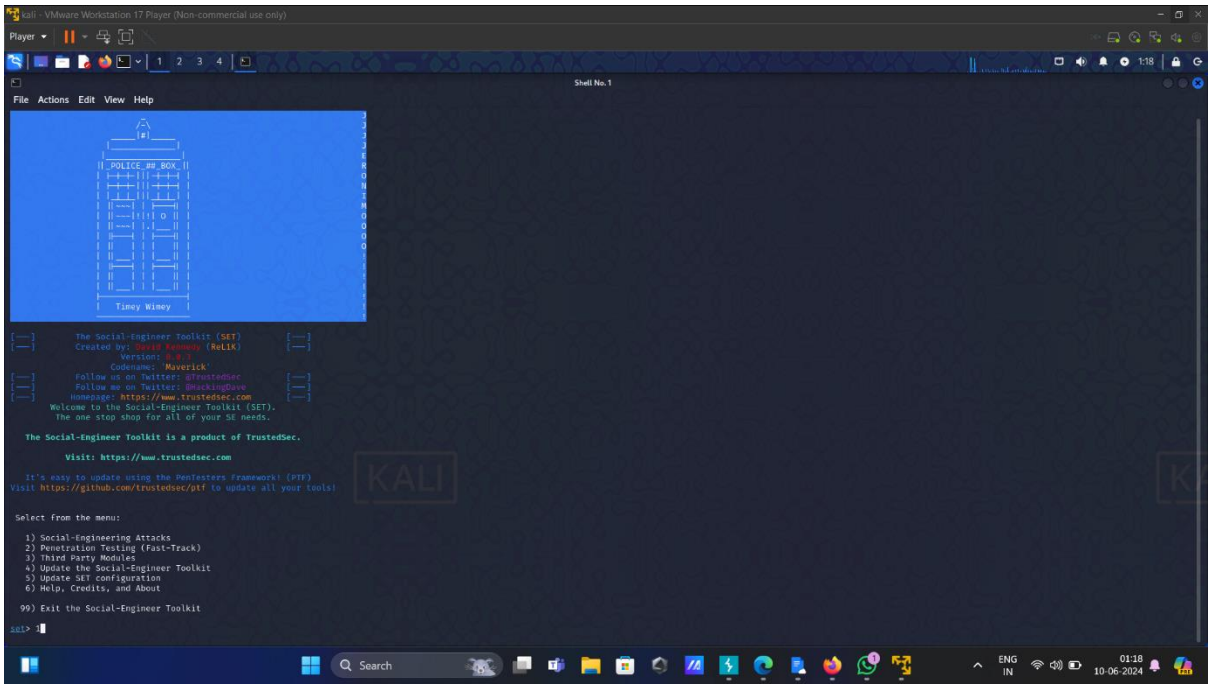

2) give password to root access.



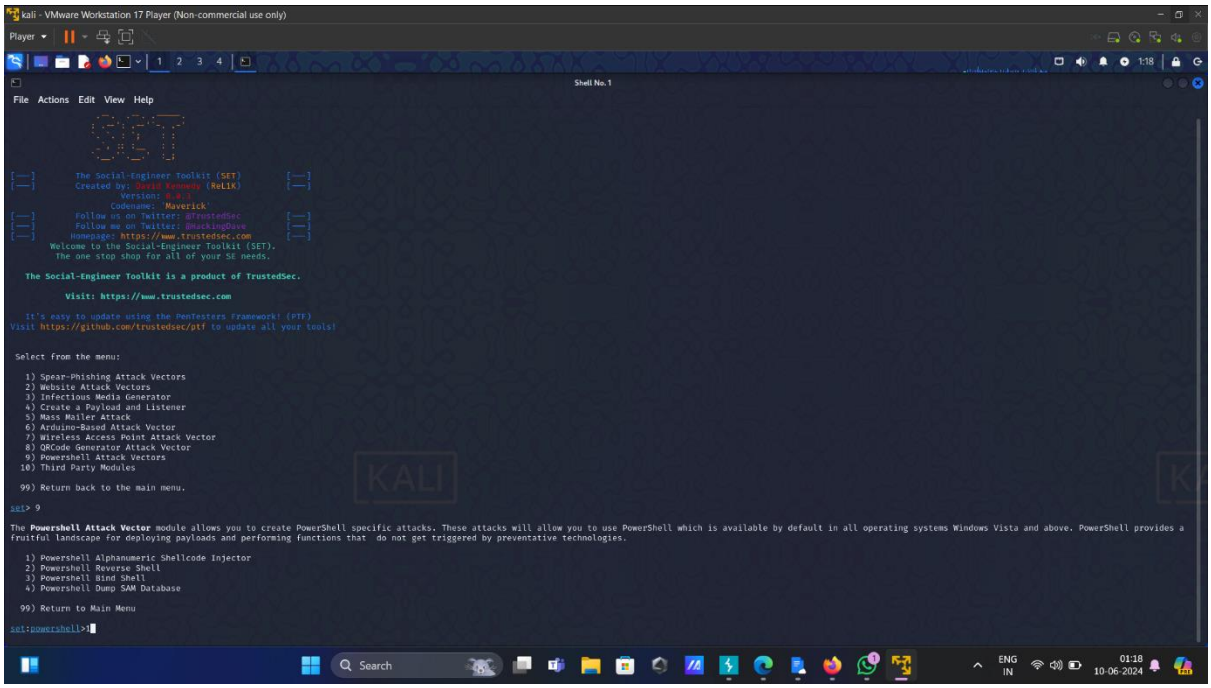
3) when the tool is opened select social- engineering attack.



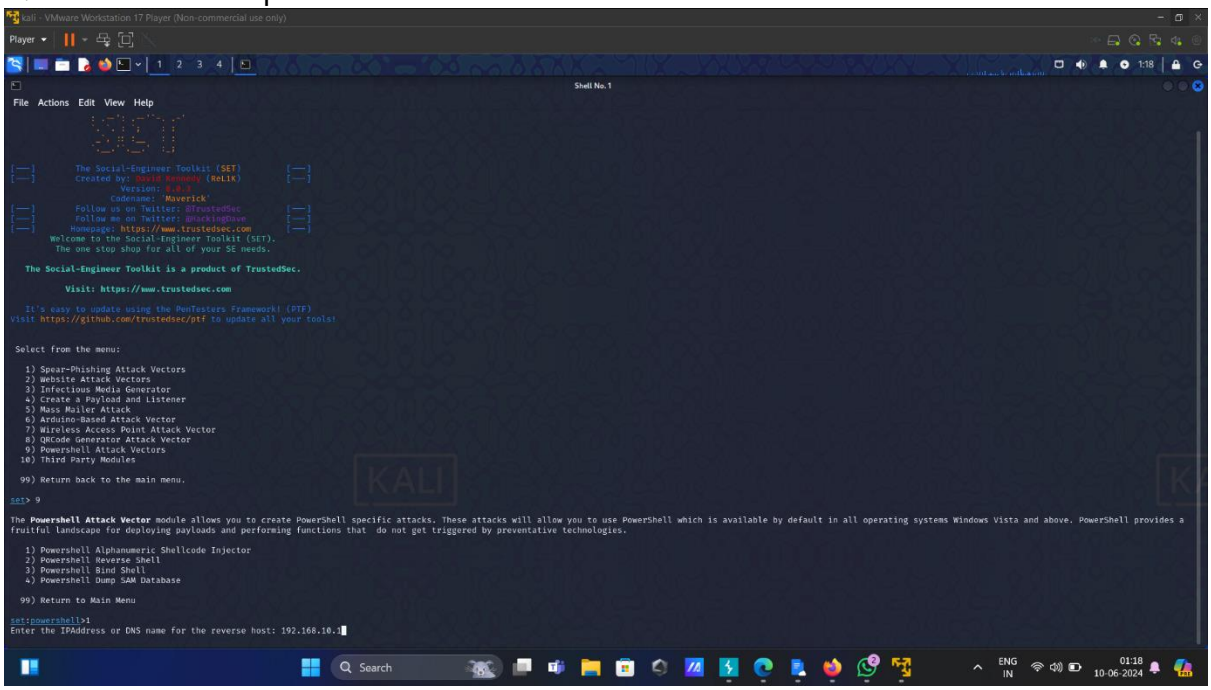

4) After that select powershell vector attack



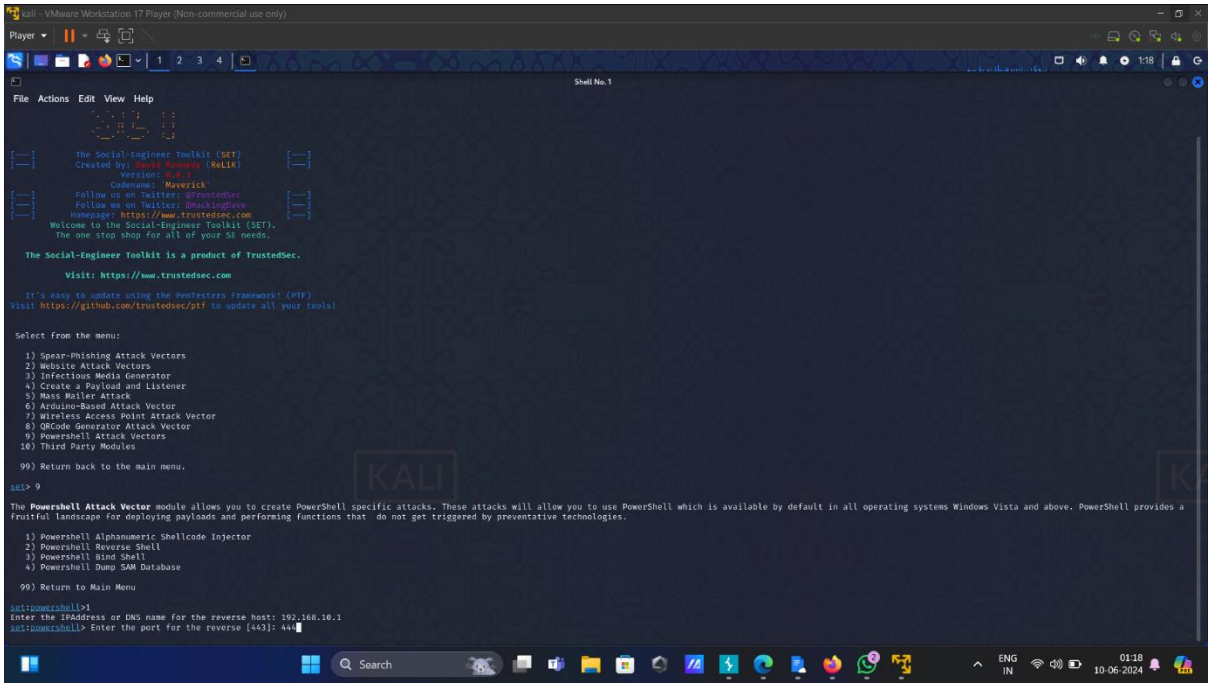
5)select powershell alphanumeric vector attack



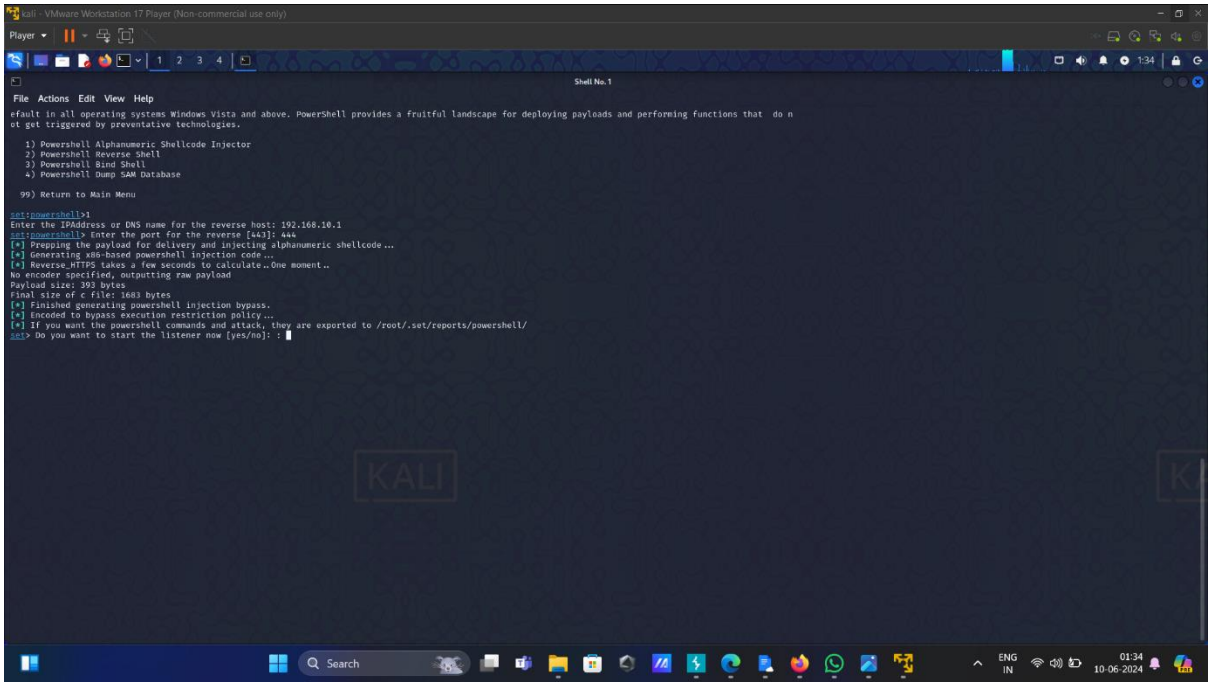

6)Enter victims ip address.



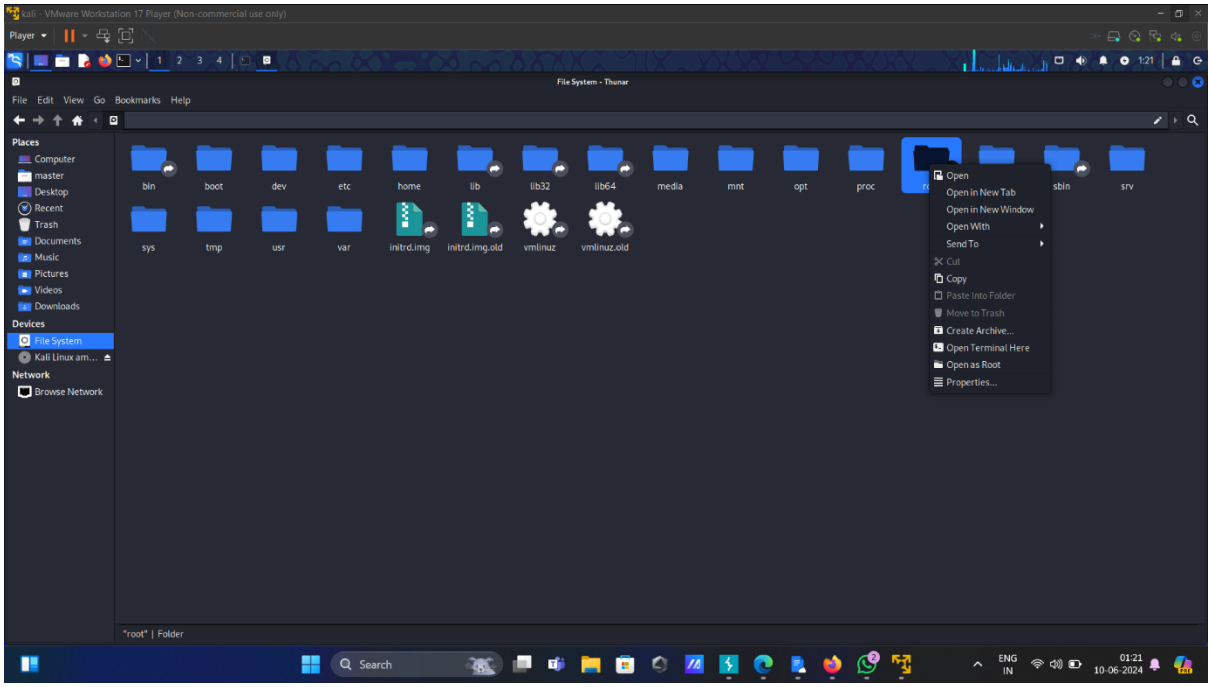

7) Enter port number.



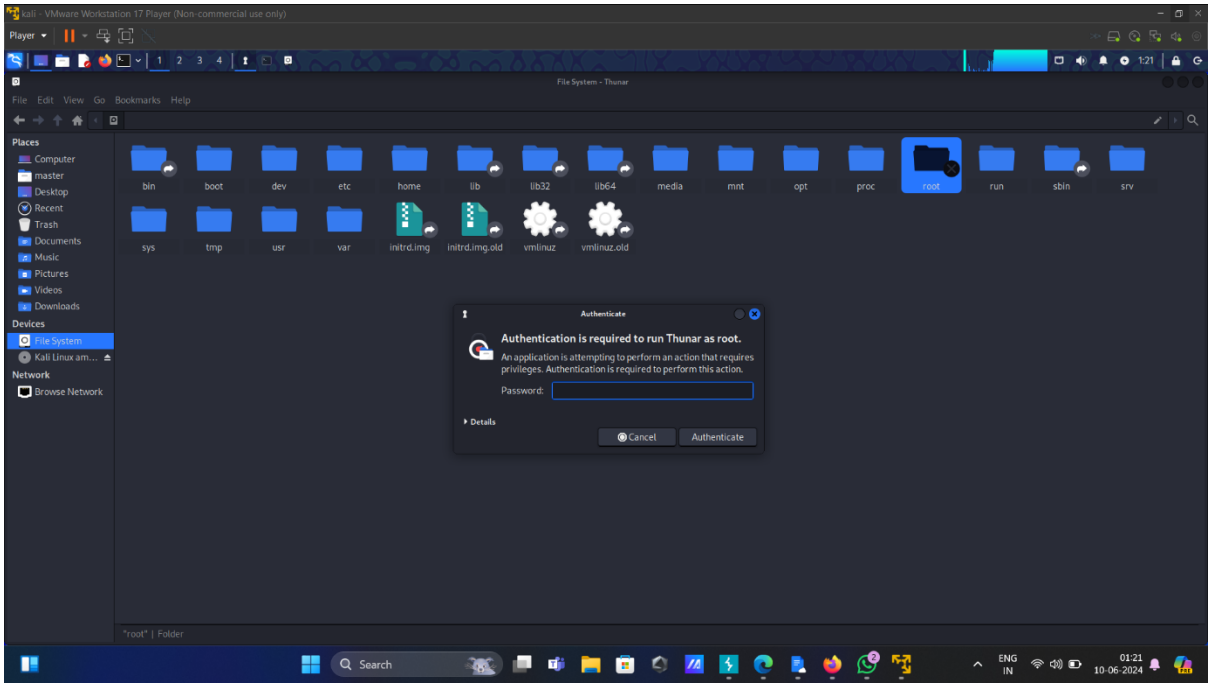

8) After that the payload is ready that means we can check that payload in file system.



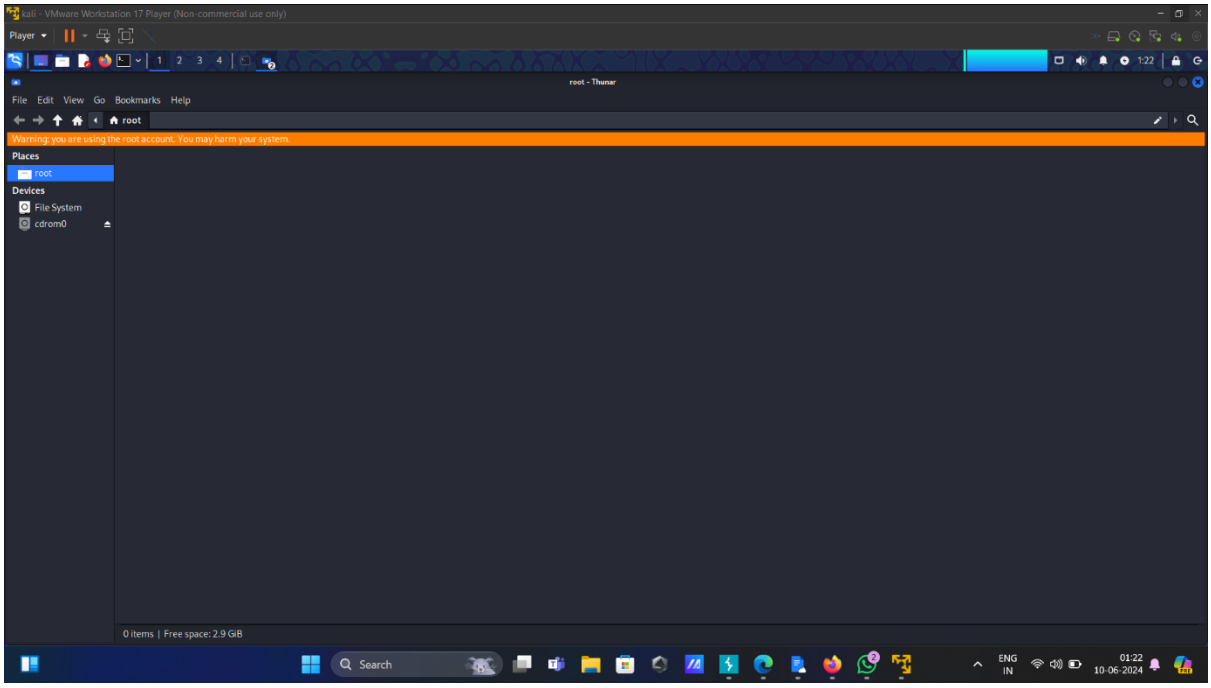

9) select root folder in file system.



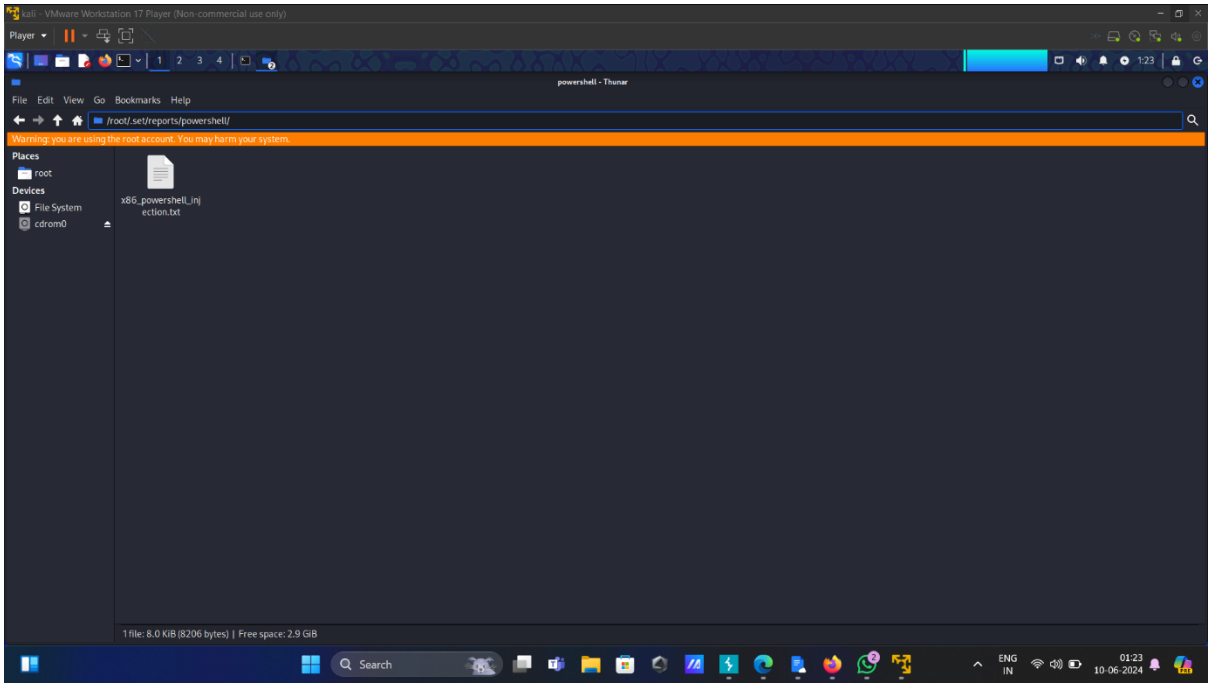

10)Give authentication password to get root access.



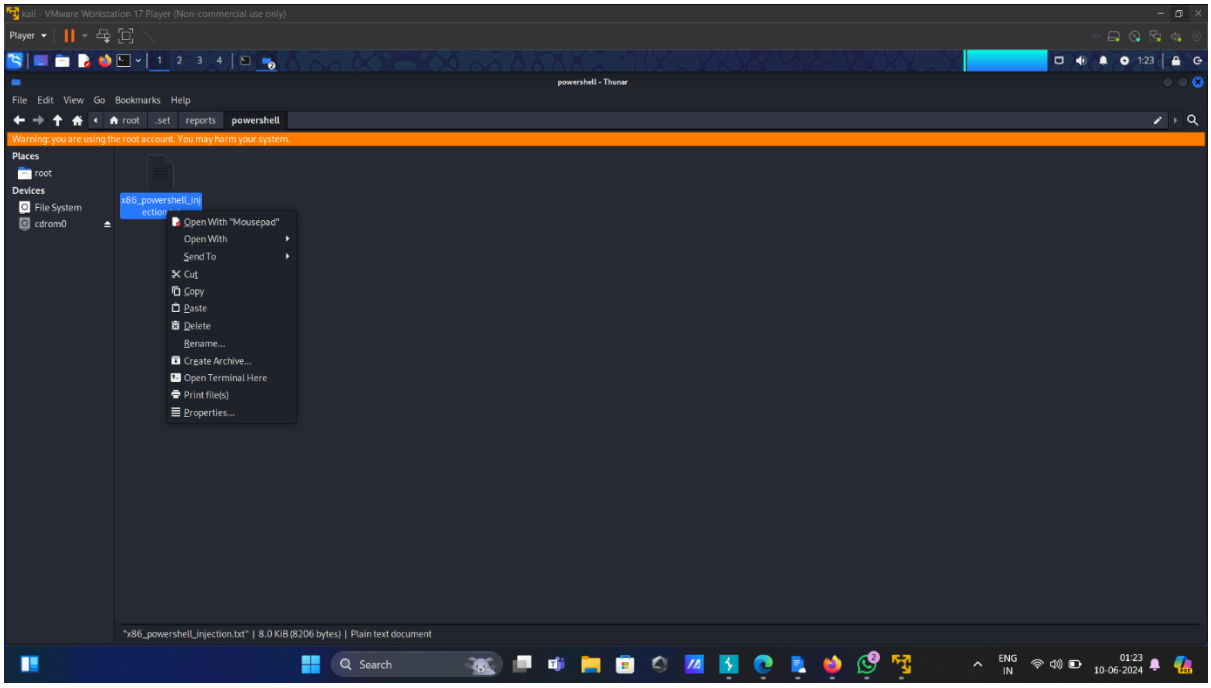

11) After that we will see this interface.



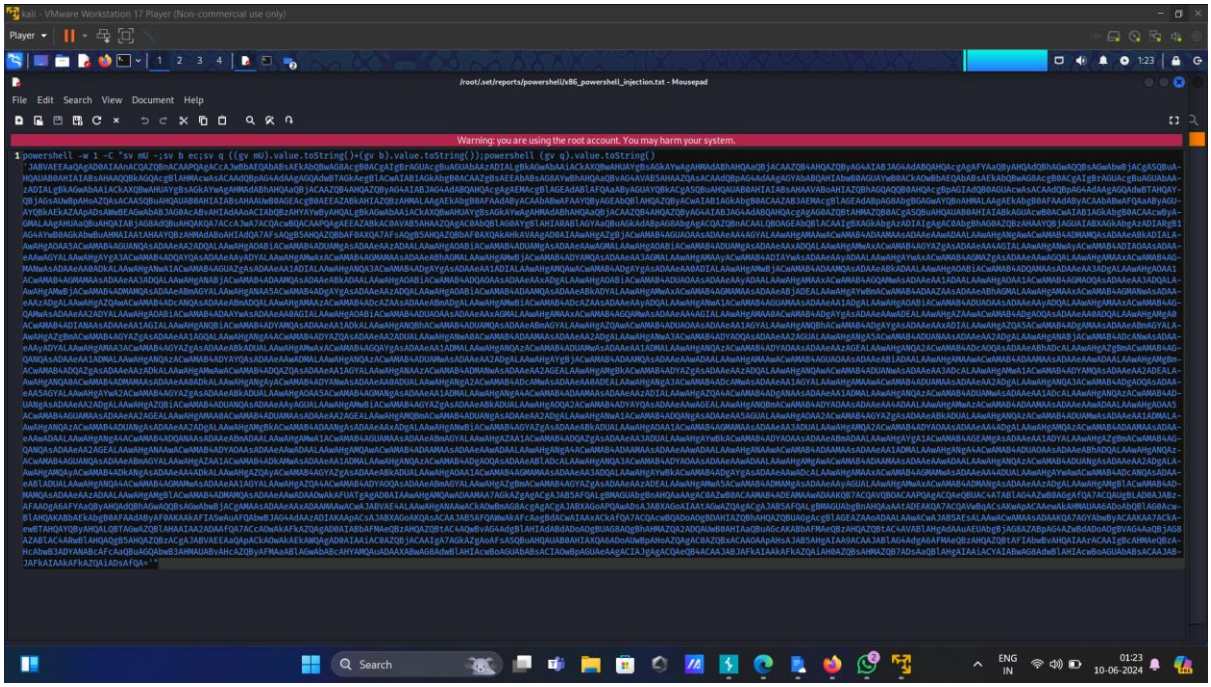

12)Enter path in search which we get in payload creation.




13) Open that file in notepad.

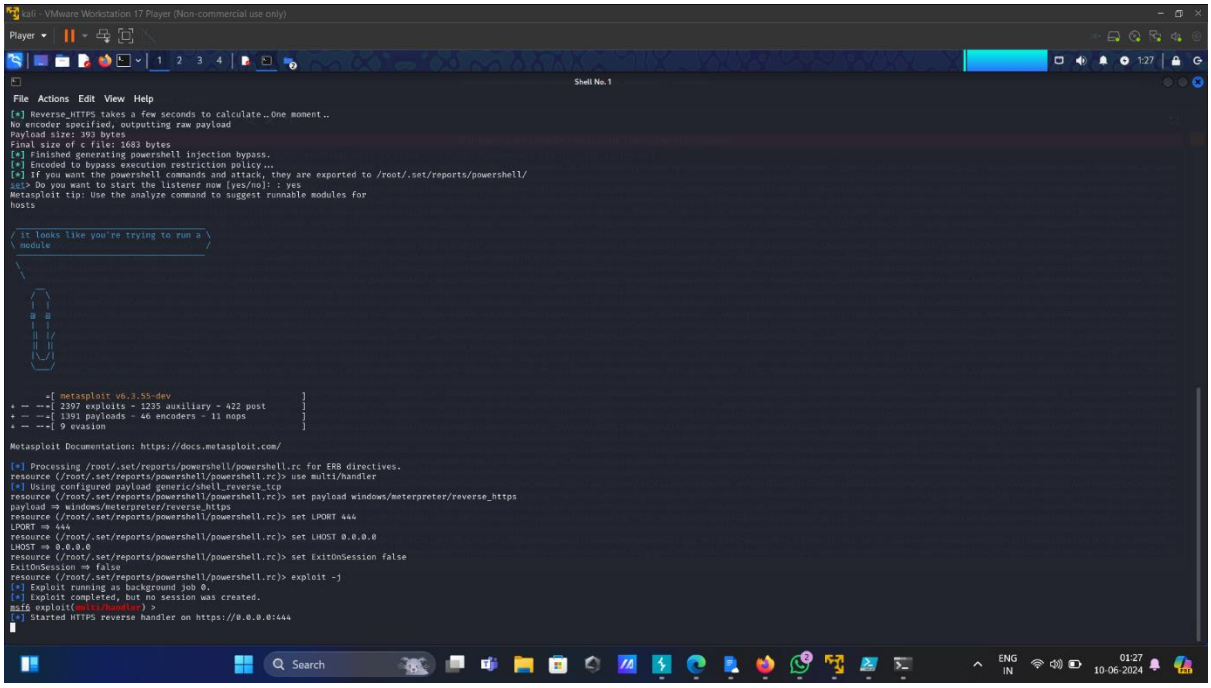



14) Copy that payloadwhich display in notepad.




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:

The backdoor creation provide attackers to control victims device.thorugh 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 cliking links from unknows sources like spam mail or messages without country code etc....
- **Keep systems private:** avoid sharing your system with others.