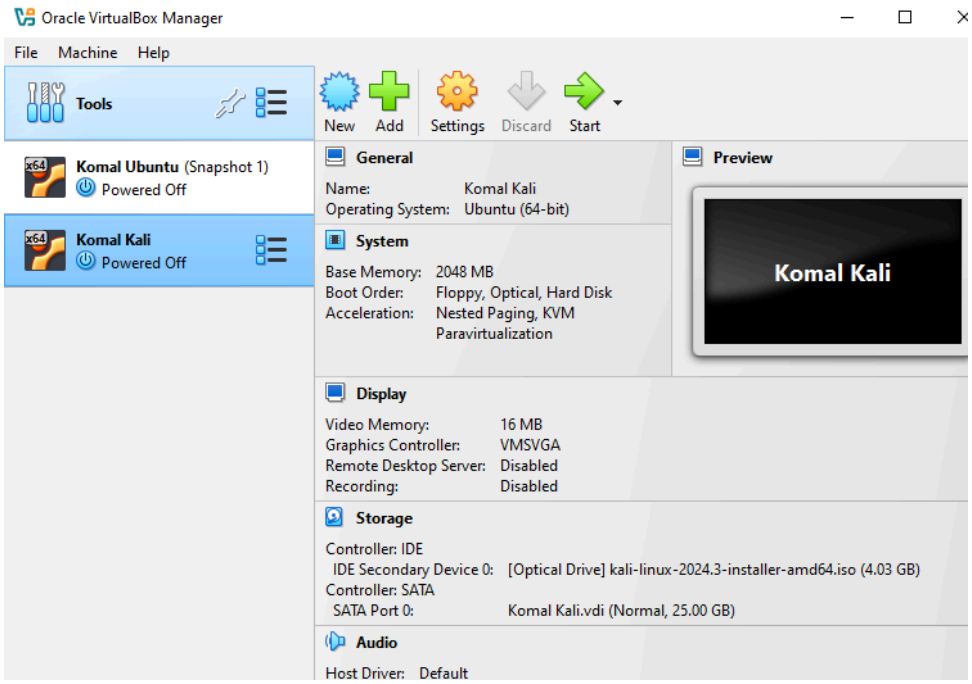
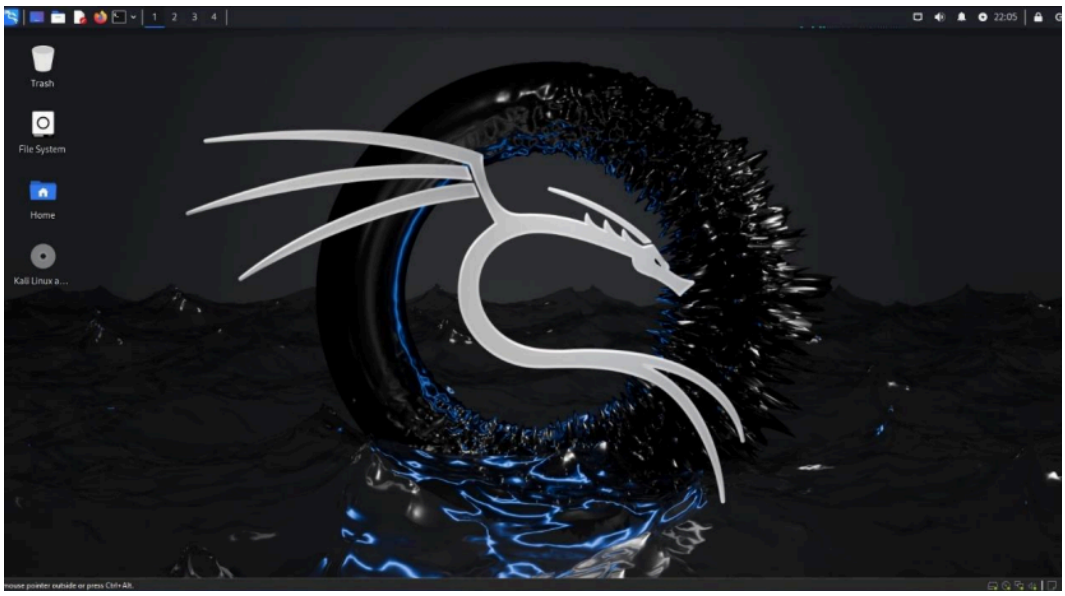
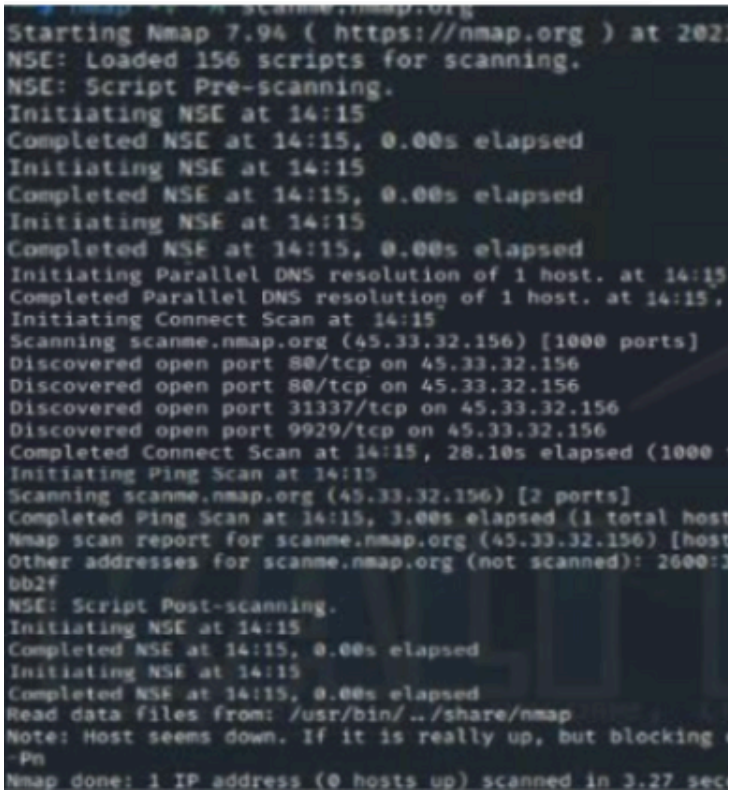
Komal Khan
IT 369 DL1
Lab 4
10/31/2024

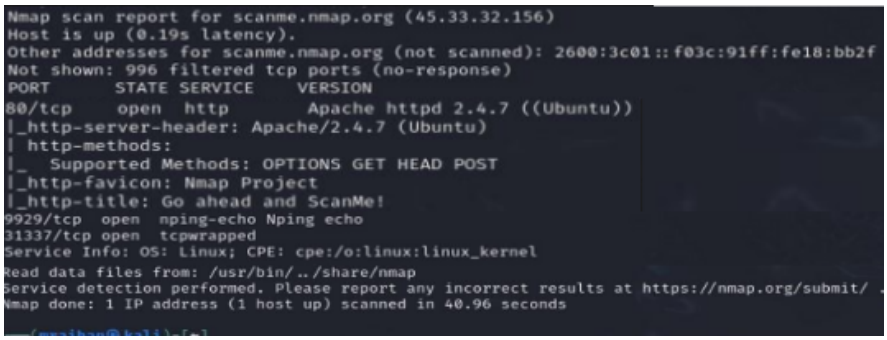
- **(10 pts) Kali**. Demonstrate that you have Kali Linux running as a VM



- **(10 pts) NMap.** Demonstrate the ability to configure and run a network scan across your VMs. At a minimum, you should scan your Ubuntu VM and create a one page document that shows the result of the scan, and then annotate and describe any notable elements shown.

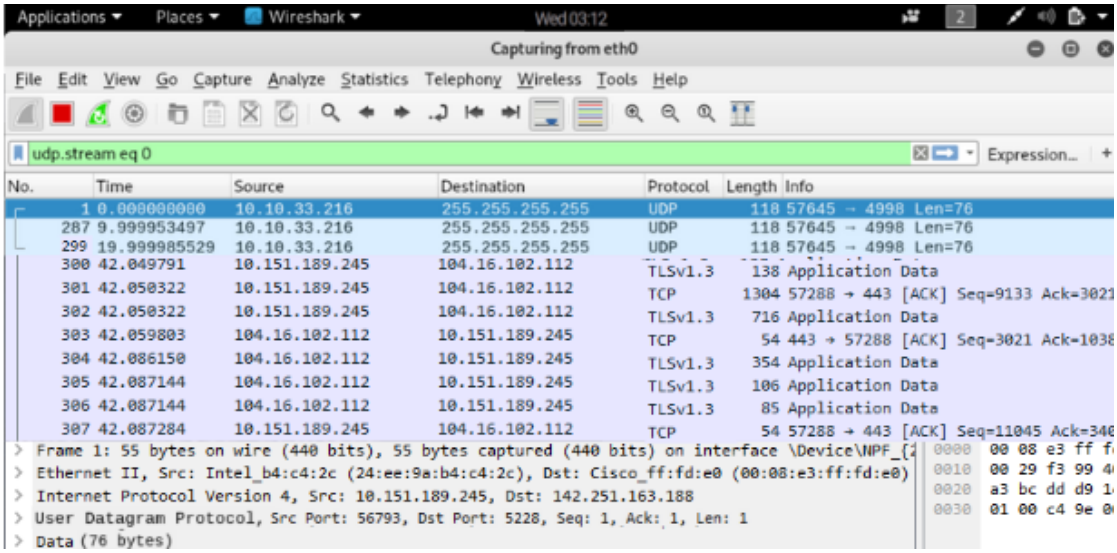



I had to open Ubuntu's terminal so I could get its IP, which is how I could target it on the nmap scan ('nmap *IP address*' to scan my LAMP VM network). When I used the command, it reported back with the current open ports, which was 4 TCP ports, 2 at port 80, 1 at port 31337, and 1 at port 9929. The port at 80 (HTTP service) is for outgoing communications, while TCP port at 31337 allows remote control of the OS. The TCP port 9929 hosts the Nping Echo service, which lets users observe how things change during transit by comparing the sent server packets with those received by the user on the other end. There were 4 open ports discovered, meaning there are 4 TCP port

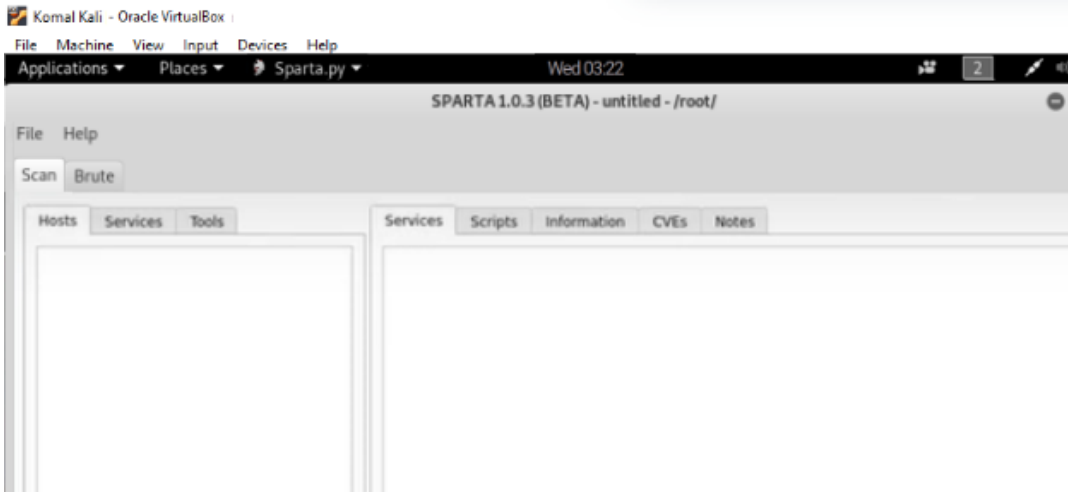
numbers that are configured to accept packets. The scan report also showed the services, the OS and version of my Ubuntu VM.

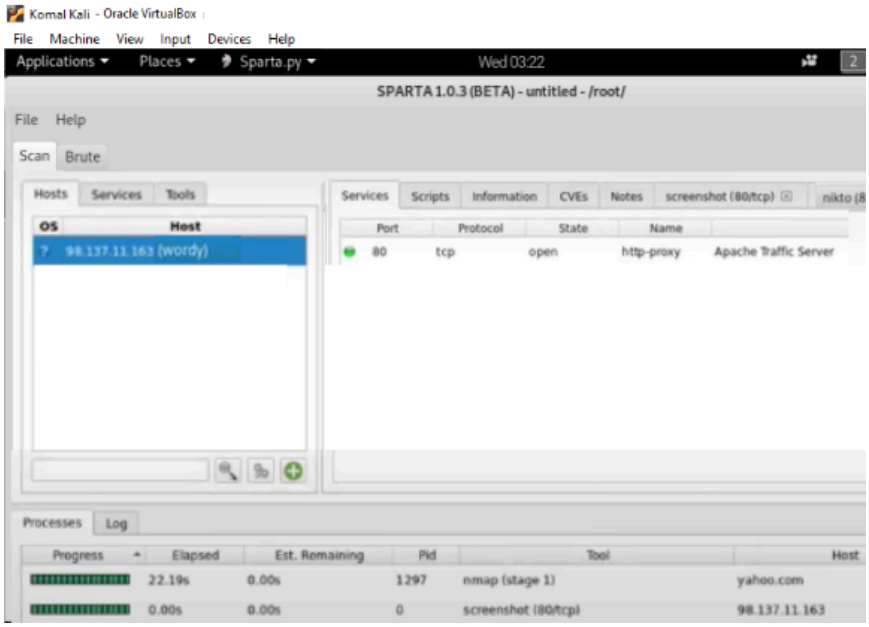
- **(10 pts) Wireshark.** Demonstrate the ability to run Wireshark to provide network traffic surveillance. Identify the protocols, packets, and relevant information from your scan to identify the types of traffic. Provide screenshot.

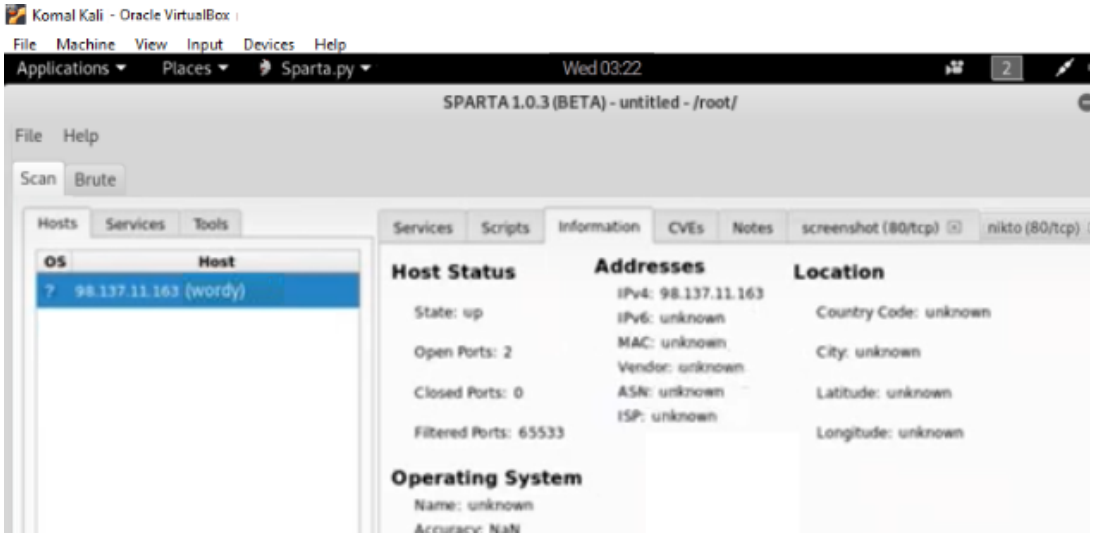


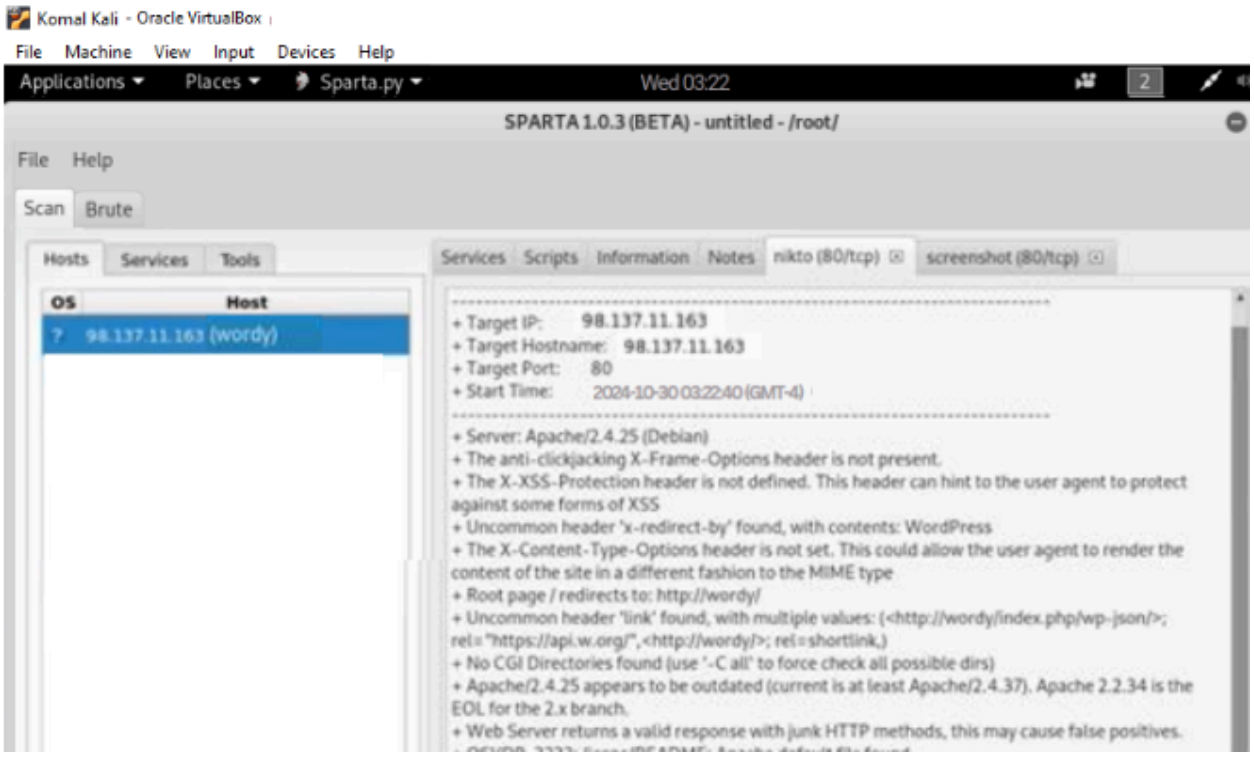

The Protocols in the Protocol column include UDP, TCP, and TLS, with each packet's source and destination address in its corresponding column. UDP connections are faster but less reliable, and an Ubuntu DNS traffic request uses port 53, while Ubuntu's VM powering up (or requesting an IP address) sends it on port 67 (DNS traffic I think). The TCP protocols are HTTP/HTTPS traffic, which involves the internet tabs I had open, and shows the handshakes ('ACK' in the info column refers to the servers acknowledgement).

- **(10 pts) Legion.** Demonstrate the ability to run Legion and show the output from the various tabs, especially Nikto.

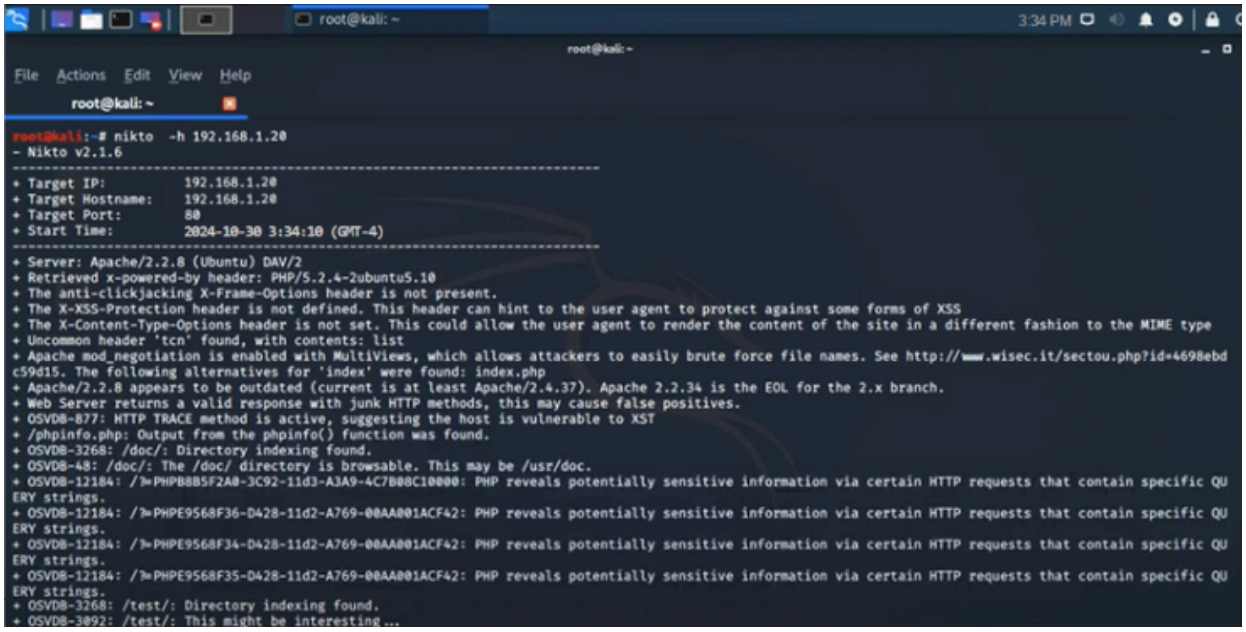

Komal Kali - Oracle VirtualBox
File Machine View Input Devices Help
Applications Places Sparta.py Wed 03:22
SPARTA 1.0.3 (BETA) - untitled - /root/
File Help
Scan Brute
Hosts Services Tools
Services Scripts Information CVEs Notes


Komal Kali - Oracle VirtualBox
File Machine View Input Devices Help
Applications Places Sparta.py Wed 03:22
SPARTA 1.0.3 (BETA) - untitled - /root/
File Help
Scan Brute
Hosts Services Tools
OS Host
98.137.11.163 (wordy)
Services Scripts Information CVEs Notes screenshot (80/tcp) nikto (8
Port Protocol State Name
80 tcp open http-proxy Apache Traffic Server
Processes Log
Progress Elapsed Est. Remaining Pid Tool Host
22.19s 0.00s 1297 nmap (stage 1) yahoo.com
0.00s 0.00s 0 screenshot (80/tcp) 98.137.11.163


Komal Kali - Oracle VirtualBox
File Machine View Input Devices Help
Applications Places Sparta.py Wed 03:22
SPARTA 1.0.3 (BETA) - untitled - /root/
File Help
Scan Brute
Hosts Services Tools
OS Host
98.137.11.163 (wordy)
Services Scripts Information CVEs Notes screenshot (80/tcp) nikto (80/tcp)
Host Status
State: up
Open Ports: 2
Closed Ports: 0
Filtered Ports: 65533
Operating System
Name: unknown
Addresses
IPv4: 98.137.11.163
IPv6: unknown
MAC: unknown
Vendor: unknown
ASN: unknown
ISP: unknown
Location
Country Code: unknown
City: unknown
Latitude: unknown
Longitude: unknown




- **(10 pts) Nikto**. Describe what Nikto provides with respect to your LAMP/Yoga App. You may need to start Nikto from one of the tabs or via the Kali command line. Nikto can be run from command line as: **nikto -h <LAMP IP>** (your LAMP stack IP address). Nikto can also be run from Legion by going to Services Tab >> HTTP >> nikto

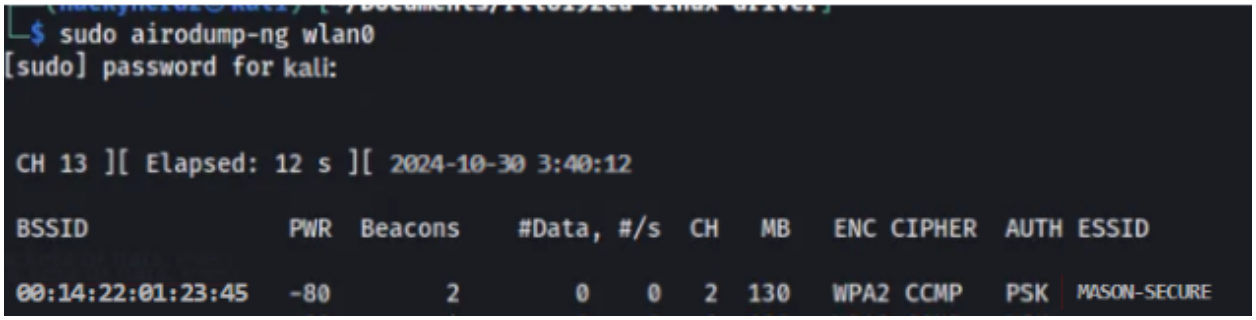


LAMP (Linux, Apache, MySQL, PHP) can be scanned for vulnerabilities using Nikto, and running it on Legion on virtualbox tests/analyzes the security of the web app hosted on the Ubuntu server. Apache and PHP can have common vulnerabilities that attackers exploit, which can be found from a Nikto scan.

- **(50 pts) Cyber Challenge!**

I chose Aircrack-ng, Whatweb, and ClamAV as my three additional Kali tools for this mission, as cyber attacks are to be expected in this region. Because each member of the team has their own personal laptop and cellphone that will have documents, data and communications from day-to-day operations on it, all while using public wifi, there are many vulnerabilities to be aware of. The public wifi allows for man-in-the-middle attacks, as public wifis are rarely encrypted and can easily be intercepted to steal credentials or data, especially due to the frequent communication back to the home office. In addition to this, each member of the team is vulnerable to attacks like phishing or keylogging on both their phone and laptop, both online and in person if anything is lost or left unattended in public. As mentioned, as the team's cyber analyst, my goal is to protect the other three members from having their systems broken into while still allowing them to do their job and communicate home daily. Using Kali Linux on my laptop, I have identified three additional tools and prepared a guide on their capabilities, usage, and potential value for the team.

**Aircrack-ng**




The first tool I chose was Aircrack-ng, which is a suite for wireless network security. This suite can be useful since we will be using public wifi, as it provides monitoring capabilities, as well as packet capturing and wifi network analysis. An example of its usage can be to secure endpoints, to check and assess wifi security before and after team members connect to wifi. This can prevent data/communications from being intercepted via MITM attacks. Airodump-ng can be used in particular to scan the nearby networks and look at security configurations and see which wifi networks have the minimum levels of encryption, for example WPA2 vs WEP. First, aircrack-ng has to be installed and monitor mode has to be enabled (**sudo apt install aircrack-ng** and **sudo**

**airmon-ng**). Once it has been set up, airodump-ng is used for capturing raw wifi packets within range on the network interface specified (in this case, wlan0). It gathers information (seen in the table) about the nearby wifi networks like their MAC address of each access point (BSSID), signal strength (PWR), encryption type (ENC), # of data packets sent, etc. Being able to see the encryption protocol and type is especially helpful, as well as the BSSID (the unique MAC address of the access point) and ESSID (network name). Knowing access points can be helpful to detect unauthorized access points and rogue access points/evil twin networks that mimic legitimate Wi-Fi networks that could trick users to connect.

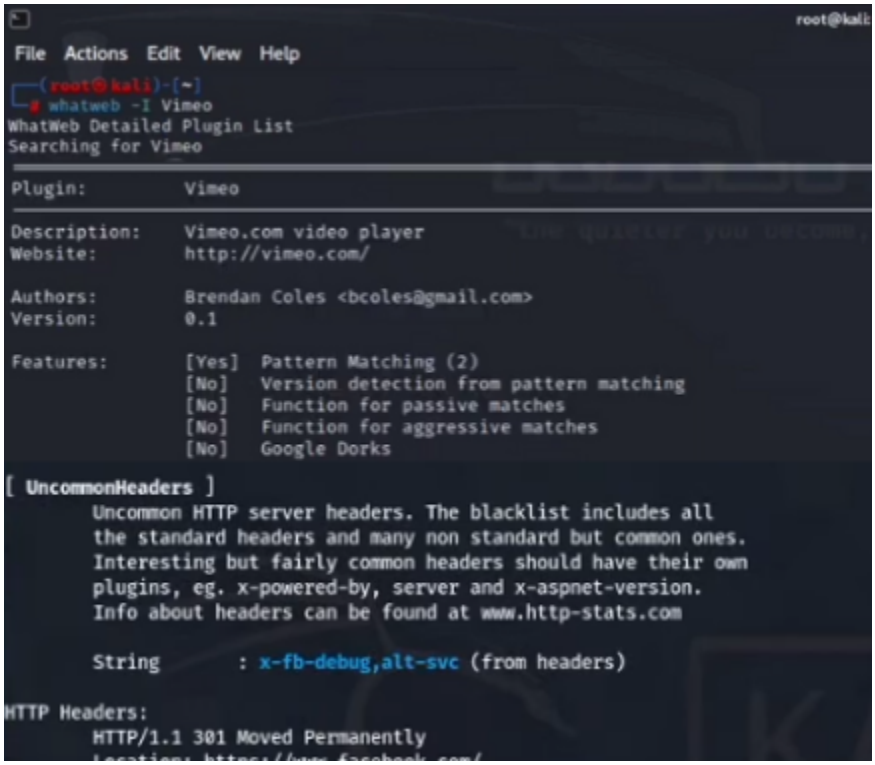
### Whatweb

```
https://outlook.office.com/ [301 Moved Permanently] Country[UNITED STATES][US],
 HTTPServer[Microsoft-IIS/10.0],
 IP[52.96.43.162],
 Microsoft-IIS[10.0],
 RedirectLocation[https://outlook.office.com/owa/],
 Strict-Transport-Security[max-age=31536000; includeSubDomains; preload],
 UncommonHeaders[request-id,x-feserver,x-requestid,alt-svc,x-feproxyinfo,x-feefzinfo,ms-cv],
 X-Powered-By[ASP.NET]
https://outlook.office.com/owa/ [302 Found] Cookies[ClientId,HostSwitchPrg,OIDC,OpenIdConnec
 Country[UNITED STATES][US],
 HTTPServer[Microsoft-IIS/10.0],
 HttpOnly[HostSwitchPrg,OIDC,OpenIdConnect.code.v1,OpenIdConnect.id_token.v1,OpenIdConnect.
 IP[52.96.181.34],
 Microsoft-IIS[10.0],
 RedirectLocation[https://login.microsoftonline.com/common/oauth2/authorize?client_id=000000
 Strict-Transport-Security[max-age=31536000; includeSubDomains; preload],
 Title[Object moved],
 UncommonHeaders[request-id,x-calculatedbetarget,x-backendhttpstatus,x-rum-validated,x-rum-
 X-UA-Compatible[IE=EmulateIE7]
```

Whatweb is the second tool I chose, which is an extension of Legion (like Nikto). This tool identifies websites and their information, by fingerprinting the web technologies they use. The technologies identified include web servers, inserted devices, plugins, programming languages, underlying web servers, frameworks, etc. This tool's value lies in helping analyze the architecture and software of a target website, to identify and assess their vulnerabilities, conduct penetration testing, and gather general intelligence to prevent malicious activity. For example, learning of the web servers or CMS's being used and identifying their versions can help us find outdated or vulnerable softwares, and this information can be shared as XML, JSON, or even CSV for data analysis. If any unexpected technologies or configurations are found, it can also be helpful to detect phishing and/or malicious sites. Trusted sites that the team uses can also be monitored with Whatweb to detect compromise and gain insight into what vulnerabilities we have that attackers could exploit. This can be used with the **whatweb -I *site name*** command; to see for yourself how it works this website can be used as an equivalent to the Whatweb tool on KaliLinux (assuming you don't have Kali): https://whatweb.net/. In

the ‘enter a domain to analyze’ search box, I’ve analyzed https://outlook.office.com/, which will be our main means of communication between the team, to assess vulnerabilities. In it, you can see the site utilizes an HTTPS server, which is the secure version of HTTP. You can also see the IP address, country, and the server it uses, which can help detect DNS hijacking/domain theft or phishing attacks.

Below, I’ve used the **whatweb -l** command on Kali to identify detailed results for Vimeo, including HTTP headers, cookies, plugins/version numbers. The plugin that it uses is ‘Videosmate-Organizer,’ and further down you can see the version and HTTP headers.




### ClamAV

The third tool I chose was ClamAV, which detects and removes various types of malwares like viruses, trojans, worms, etc. For obvious reasons, this seems like a reasonable choice, as detecting malware doesn’t mean much if it can’t be removed. A **clamscan** scans downloaded and incoming files, directories and/or emails, as well as real-time scanning to monitor file transfers and network activity to detect malware in the teams’ laptops. The **clamd** feature allows for continuous, on-demand scanning, which would be particularly useful due to the frequent communication to the home office. A download/attachment can be scanned before it is opened using **clamscan /path/to/downloaded/file.extension** to ensure they are safe to open. Regular scanning of shared files/directories can also be done to keep the shared resources and the online work environment clean and secure, minimizing the spread of malware across devices.

It can also scan external media like USBs that can be used to transfer files securely without having to rely on the Internet, particularly when on public wifi. Each time a USB drive is connected, ClamAV can scan the entire drive using **clamscan -r /media/usb-drive/**. Not only does ClamAV offer proactive and live malware defense, but it is also lightweight and efficient, and therefore allows for quick response time to minimize disruptions to the teams work.