

Foundational Network Defense & Traffic Analysis

Nicholas Massei

Cybersecurity Project/Lab

July 10, 2025

Introduction

In this task, I focused on understanding core network security principles and implementing essential defensive measures in a virtual lab environment. I began by researching the common network threats, including viruses, worms, trojans, and phishing attacks. Then, I wanted to transition into setting up and securing a basic network using VirtualBox – with 2 VMs – and my home router. I planned for these exercises to give me hands-on experience with both network defense in theory, and real-world application.

Network Threats

- **Virus:** A virus is malicious code that can attach itself to a legitimate program or file that spreads when the infected item is run or shared. Simply put, viruses can damage files, corrupt systems, and slow down performance.
- **Worm:** A worm is a standalone program that can replicate and spread across networks without user action. Unlike a virus, it doesn't need to attach to a file. Also, worms can often consume bandwidth or overload systems.
- **Trojan Horse:** A trojan is another type of malicious program that relies on pretending to be harmless or useful software. Once a trojan is installed, it can open a back door for attackers, steal sensitive data, or act as a Remote Access Trojan (RAT) to give full remote access of the system to the attacker.
- **Phishing:** Phishing is a common technique that tricks users through emails, websites, or messages into giving away sensitive information such as user credentials or credit card numbers. Attackers often impersonate trusted entities to seem convincing as well. Phishing attacks can range from broad, generic campaigns, to highly targeted attacks that are carefully researched and personalized, such as spear phishing.

Security Concepts

- **Firewalls:** Block/allow traffic based on rules (Windows Defender Firewall).
- **Encryption:** Secures data in transit (such as HTTPS or WPA2/WPA3).
- **Secure Configs:** Changing default router/admin passwords, disabling unused ports, and others.

Reference

National Institute of Standards and Technology. *Computer Security Resource Center Glossary*. U.S. Department of Commerce. <https://csrc.nist.gov/glossary>

Set up a simple network environment

Set up using VirtualBox – 2 virtual machines (Windows 10 & Ubuntu)

Enabled and configured basic firewall on Windows machine using Windows Defender Firewall

Set up other basic security configurations and network encryption by accessing my router's configuration page by entering the default gateway IP address into my web browser (gateway located by utilizing "ipconfig" in powershell). From there, I navigated to the wireless security settings and confirmed that WPA2-PSK (AES) was enabled to ensure strong encryption for my network. Also replaced the default credentials in use with a strong password via password manager (Bitwarden).

Monitor Network Traffic

Internal network created successfully – Windows and Ubuntu machines are able to communicate.

Launched Wireshark on the Windows VM to sniff packets from the Ubuntu VM across the internal network.

Filtered captured traffic using:

- http – viewed unencrypted web traffic (few generated)

- dns – observed domain name lookups

Suspicious Packets Example (not observed)

- Repeated ping traffic from Ubuntu (this example was observed)
- Traffic destined for unfamiliar IPs
- Example of potential DNS misdirections
- Any other unusual traffic depending on context

Document Findings

Configured basic **Windows Defender Firewall** controls on the Windows VM I set up to enforce traffic control and reduce exposure to unauthorized access to the system.

Accessed the router's administrative interface via the default gateway IP (identified beforehand using ipconfig) to review and update wireless security settings. Confirmed **WPA2-PSK (AES)** was enabled, ensuring a strong encryption standard was in place.

Default Wi-Fi credentials were replaced with a strong, randomly generated password stored securely using **Bitwarden**, reinforcing both password security and future access control.

Created a **VirtualBox** Internal Network environment to connect an Ubuntu and Windows machine together to enable direct communication. Verified connectivity between the systems and initiated live traffic capture (simulating practical packet capture scenarios).

Used **Wireshark** for packet capture on the Windows VM to monitor traffic originating from the Ubuntu VM. Also applied filters to capture **HTTP** and **DNS** traffic specifically, observing key protocol behavior in real time. The visibility into just basic traffic patterns highlighted how much can be observed from packet-level inspection from both offensive (red-team) and defensive (blue-team) objectives.

These foundational security measures can significantly reduce the attack surface by limiting unauthorized access to systems. Even in simple lab environment, practicing enforcing firewall rules,

securing wireless settings, and monitoring traffic tools (such as Wireshark) help simulate the layered defenses needed to detect and prevent threats – and is a great learning experience for anyone interested in information security.

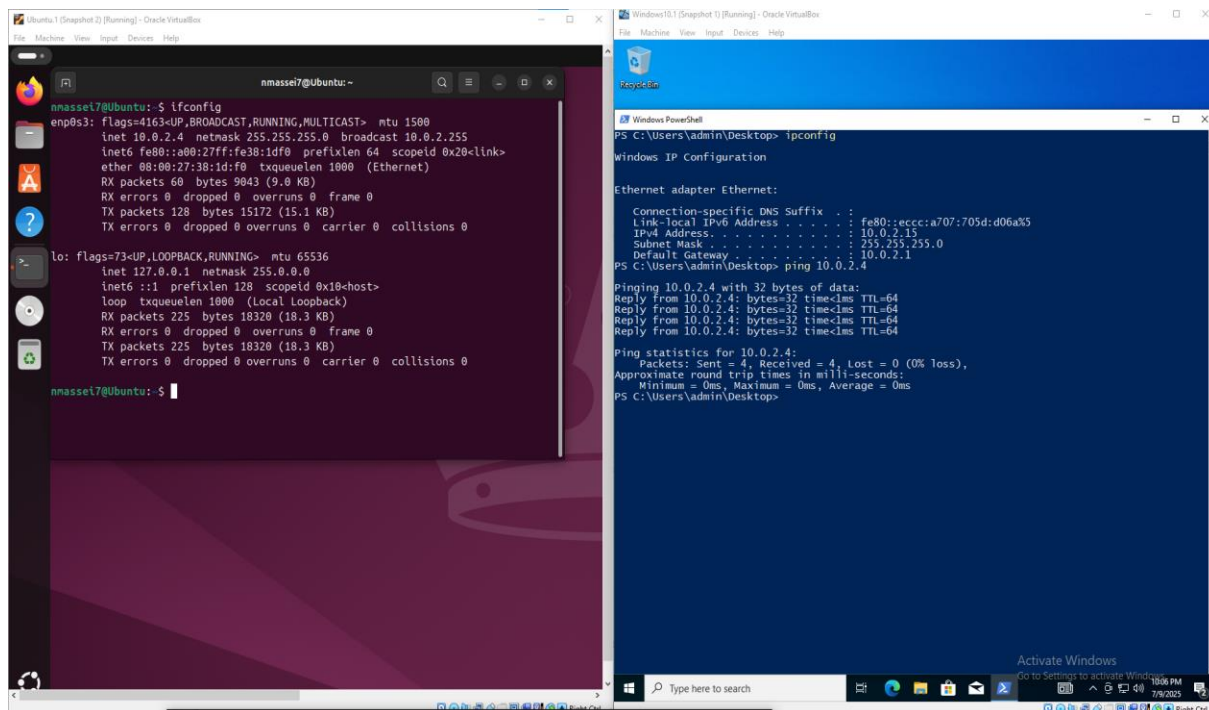
Challenges Faced

While setting up two virtual machines (Windows 10 and Ubuntu), I encountered configuration issues related to internal networking and firewall rule conflicts. Wireshark also initially failed to capture traffic until the correct adapter settings were applied. Troubleshooting these issues deepened my understanding of how VMs are configured.

Reflect on Security Best Practices

Even in larger or more complex environments, the importance of basic security measures really stands out. These small steps I went through like enabling a firewall, using strong passwords, and securing the wireless network create a solid foundation that everything else builds on. While they might seem simple at first, I've come to see how they play a huge role in reducing risk and keeping systems protected. Larger networks obviously need more advanced solutions like intrusion detection systems (IDS), endpoint protection, and network segmentation with VLANs, but the basics still matter just as much.

I also think user awareness is key to network security and this really showed in changing default credentials. Things like regular software updates, multi-factor authentication, and avoiding phishing attempts can make a big difference and would be the first things to cover in educating people on the importance of network security. I would also show real-world examples of attacks that happened simply because the basics were ignored as people tend to learn best when they see the actual consequences of oversights like these. Even simple habits, when practiced consistently, can go a long way in strengthening the overall security posture and lowering the attack surface for threat actors.



Successful Ping to connected machines

The image shows a Wireshark network traffic capture. The top pane displays a list of captured packets:

No.	Time	Source	Destination	Protocol	Length	Info
480	114.078331	72.247.234.254	10.0.2.4	OCSP	927	Response
607	115.540797	10.0.2.4	142.251.186.94	OCSP	489	Request
608	115.549910	10.0.2.4	142.251.186.94	OCSP	489	Request
611	115.555894	10.0.2.4	142.251.186.94	OCSP	489	Request
612	115.608000	142.251.186.94	10.0.2.4	OCSP	963	Response
613	115.608000	142.251.186.94	10.0.2.4	OCSP	963	Response
614	115.608000	142.251.186.94	10.0.2.4	OCSP	963	Response
698	115.911952	10.0.2.4	142.251.186.94	OCSP	489	Request
706	115.953755	142.251.186.94	10.0.2.4	OCSP	965	Response
775	116.133516	10.0.2.4	172.64.149.23	OCSP	493	Request
776	116.133624	10.0.2.4	172.64.149.23	OCSP	493	Request
779	116.133295	10.0.2.4	172.64.149.23	OCSP	493	Request
787	116.205339	172.64.149.23	10.0.2.4	OCSP	1342	Response
789	116.213460	172.64.149.23	10.0.2.4	OCSP	1342	Response
792	116.228677	172.64.149.23	10.0.2.4	OCSP	1342	Response
923	116.683206	10.0.2.4	34.107.221.82	HTTP	372	GET /success.txt?ip=10.0.2.4 HTTP/1.1
926	116.689548	10.0.2.4	34.107.221.82	HTTP	355	GET /canonical.html HTTP/1.1
927	116.789133	34.107.221.82	10.0.2.4	HTTP	270	HTTP/1.1 200 OK (text/plain)
928	116.748711	34.107.221.82	10.0.2.4	HTTP	352	HTTP/1.1 200 OK (text/html)
933	116.723328	10.0.2.4	34.107.221.82	HTTP	372	GET /success.txt?ip=10.0.2.4 HTTP/1.1
936	116.749543	34.107.221.82	10.0.2.4	HTTP	270	HTTP/1.1 200 OK (text/plain)
957	117.161096	10.0.2.4	142.251.186.94	OCSP	496	Request

The bottom pane shows the details of the selected packet (No. 11549):

```

Frame 11549: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface \Device\NPF_{7029E6DF-8353-4709-8309-ACD1E98A31F}
  Section number: 1
  Interface id: 0 (\Device\NPF_{7029E6DF-8353-4709-8309-ACD1E98A31F})
  Encapsulation type: Ethernet (1)
  Arrival Time: Jul 9, 2025 23:00:11.732104000 Central Daylight Time
  UTC Arrival Time: Jul 10, 2025 04:00:11.732104000 UTC
  Epoch Arrival Time: 1752120011.732104000
  [Time shift for this packet: 0.000000000 seconds]
  [Time delta from previous captured frame: 0.000126000 seconds]
  [Time delta from previous displayed frame: 6.164403000 seconds]
  [Time since reference or first frame: 156.912599000 seconds]
  Frame Number: 11549
  Frame Length: 142 bytes (1136 bits)
  Capture Length: 142 bytes (1136 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: ethertype:ip:tcp:http]
  [Coloring Rule Names: HTTP]
  [Coloring Rule String: http || tcp.port == 80 || http2]
  Ethernet II, Src: PCSysntec_38:1d:f0 (08:00:27:38:1d:f0), Dst: 52:54:00:12:35:00 (52:54:00:12:35:00)
  Internet Protocol Version 4, Src: 10.0.2.4, Dst: 91.189.91.49
  Transmission Control Protocol, Src Port: 44056, Dst Port: 80, Seq: 1, Ack: 1, Len: 88
  Hypertext Transfer Protocol

```

http packets captured including multiple "GET" requests and OCSP packets sent over http

The screenshot displays a Wireshark interface with a list of captured network packets. The top pane shows a list of packets with columns for No., Time, Source, Destination, and Protocol Length. The bottom pane shows a detailed view of a selected packet (No. 193), which is a DNS query and response. The details pane shows the following information:

- Section number: 1
- Interface id: 0 (Device\NPF_{7D29ECD9-8353-4789-8309-ACD1E98A31F})
- Encapsulation type: Ethernet (I)
- Arrival Time: Jul 9, 2025 22:59:30.932715000 Central Daylight Time
- UTC Arrival Time: Jul 9, 2025 03:59:30.932715000 UTC
- Epoch Arrival Time: 1752119970.932715000
- [Time shift for this packet: 0.000000000 seconds]
- [Time delta from previous captured frame: 0.000059000 seconds]
- [Time delta from previous displayed frame: 0.000059000 seconds]
- [Time since reference or first frame: 116.111120000 seconds]
- Frame Number: 764
- Frame Length: 193 bytes (1544 bits)
- Capture Length: 193 bytes (1544 bits)
- [Frame is marked: False]
- [Frame is ignored: False]
- [Protocols in frame: eth:ethertype:ip:udp:dns]
- [Coloring Rule Name: UDP]
- [Coloring Rule String: udp]

The packet list shows the following details for the selected packet (No. 193):

No.	Time	Source	Destination	Protocol Length	Info
193	116.113961	68.105.29.11	10.0.2.4	DNS	169 Standard query response 0x420f AAAA ocsip.sectigo.com CNAME ocsip.comodoca.com.cdn.cloudflare.net A 172.64.149.23 A 184.18.38.233 OPT

The packet details pane shows the following details for the selected packet (No. 193):

```

0000  00 00 27 38 1d f0 52 54  00 12 35 00 00 00 43 00  --B RT S E
0010  00 33 14 62 00 00 ff 11  39 00 44 69 1d 00 0e 00  --B ... 9'G...
0020  02 04 00 35 ad c8 00 9f  50 17 cd 39 81 80 00 01  --5 .... P . 9 ...
0030  00 03 00 00 01 04 6f 63  73 70 07 73 65 63 74    --o csp sect
0040  69 67 6f 03 63 6f 64 00  00 1c 00 01 c8 0c 00 05  --o csp
0050  00 01 00 00 00 00 26 04  6f 63 73 70 08 63 6f    --... & ocsip co
0060  6d 6f 64 6f 63 61 03 63  6f 64 03 63 64 6e 0e 63  --o dca c m cdn c
0070  6c 6f 75 64 66 6c 61 72  63 03 6e 65 74 00 c8 7e  --ou f l a r e r e t
0080  00 1c 00 01 00 00 75 00  10 26 06 47 00 44 00    --... u . & G D
0090  00 00 00 00 00 00 12 26  09 c8 2e 00 1c 00 01    --... h & G ...
00a0  00 00 00 75 00 10 26 06  47 00 44 00 00 00 00    --... & G D ...
00b0  00 00 ac 40 95 17 00 00  29 02 00 00 00 00 00    --... ) ...
00c0  00
  
```

Captured and observed DNS traffic displaying the records requested during browsing activity.