

Lab 7 Report

Jason Tolbert

The Pennsylvania State University

IST 894-001: Capstone Experience

Dr. Michael Bartolacci, Instructor

Sunday, March 9th

Table of Contents

General Overview..... 3

Technical Overview 5

References 7

Screenshots 8

General Overview

This lab guides demonstrates to participants several basic network and web application vulnerabilities.

The lab begins by introducing participants to packet analysis. Packet analysis, broadly speaking, is the interception and examination of data that travels across a computer network. Packet analysis allows security professionals to monitor traffic patterns and detect anomalies that might be indicative of a cyberattack (Joseph et al., 2024). Users use Wireshark (a popular graphical packet analysis program) and tcpdump (a popular command-line program for the same) to capture, analyze, and understand the contents of network packets traveling to and from their machine.

The lab continues by teaching participants of the dangers of insecure network protocols, namely Telnet (a protocol for remotely connecting to one computer's command line from another) and FTP (a protocol for wirelessly transferring files between two computers).

Telnet and FTP are insecure because they are unencrypted. Unencrypted traffic is far easier for users of packet analysis tools — like Wireshark — to analyze than encrypted traffic (Sikos, 2020). This, as participants learn, is especially dangerous when sensitive information is part of that traffic. Both Telnet and FTP typically require the user to provide credentials before connecting to remote machine. Participants use preset credentials to establish authenticated connections over both protocols — only to check Wireshark and

see that their credentials have been captured in plain text. (It's worth noting that the insecurity of Telnet and FTP isn't much of a concern these days; both protocols have been obsoleted by more secure successors (Nixon & Devaraj, 2016) and are rarely used in modern computing environments.)

The lab then pivots from network security to web app security. Participants are taught about cross-site scripting (XSS). XSS is a vulnerability which allows client-side scripts to be injected into web pages and run on the machines of their visitors. Specifically, participants learn of three types of XSS: reflected XSS, where a website accepts input from a user and renders it back to them in an unsafe way; stored XSS, where a malicious script is persistently stored somewhere on the target server and is run on the user's machine whenever their browser requests information from that location on the server; and DOM-based XSS, where a malicious script is injected through vulnerabilities in the website's JavaScript code (Drissi, 2024).

Technical Overview

This lab provides participants hands-on experience with network traffic analysis and teaches them of the risks of insecure network protocols and common web vulnerabilities.

The lab begins by introducing participants to packet sniffing and traffic analysis. Most of this segment is focused on Wireshark, but participants also get exposure to tcpdump. Participants capture traffic as it goes between their own machine and a remote one and analyze their contents.

After being introduced to Wireshark, participants use it to explore firsthand the dangers of using outdated, unencrypted, network protocols — namely, Telnet and FTP. After establishing and closing a Telnet connection with a remote machine, participants return to Wireshark to see the packets captured during the connection. Examination reveals to them that their Telnet credentials were exposed in plain text. Participants then do a similar exercise with FTP. Both Telnet and FTP have long been obsoleted by the more secure SSH and SFTP. This is also touched on in the lab — participants open a premade Wireshark capture of an SSH connection and see for themselves that unlike the packets from the Telnet connection, the SSH packets are encrypted, and no credentials are plainly exposed.

The lab then pivots to web app vulnerabilities. Participants explore vulnerabilities on both the client side and server side. All three kinds of cross-site scripting — stored, reflected,

and DOM-based — are covered. Though participants never exploit XSS to run any actual malicious code, they do use it to perform relatively harmless actions — like triggering JavaScript alerts — that allow them to get an idea of what XSS could potentially be used for in the hands of a real attacker. In addition to XSS, the lab has participants explore SQL injection and directory traversal in the hopes of stressing to them the importance of sanitizing user inputs.

References

Drissi, S. Z. (2024). *Cross-site scripting (XSS) attacks and penetration testing*.

Joseph, G., Osamor, J., & Olajide, F. (2024). A systematic review of network packet sniffing tools for enhancing cybersecurity in business applications. *International Journal of Intelligent Computing Research*, 15(1).

Nixon, J. S., & Devaraj, D. (2016). *Vulnerability study of remote protocols: Telnet & SSH and a proposed method for secure communication using telnet over IPSec in windows platform*.

Sikos, L. F. (2020). Packet analysis for network forensics: A comprehensive survey. *Forensic Science International: Digital Investigation*, 32, 200892.
<https://doi.org/10.1016/j.fsidi.2019.200892>

Screenshots

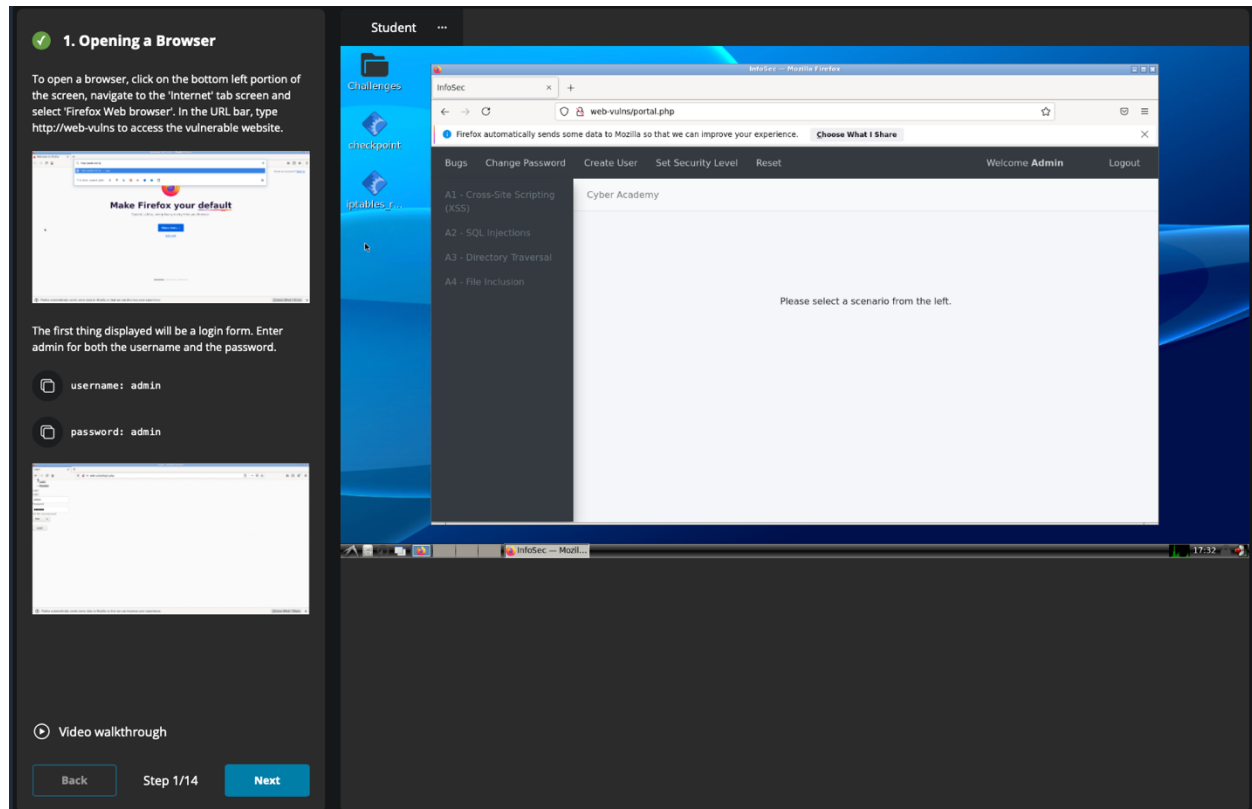


Figure 1. The web-vulns main page.

5. Stored XSS attacks - 2

An attack vector in this case would be to include a link in the comments that when clicked, hijacks the user account or gathers user data. Adversaries usually make these links look unsuspicious to trick the users into clicking them. The HTML code for such an attack would look like the following:

To view something interesting, click here

Write the line above in the input box and click submit. The output will look like in the image below.

XSS - Stored (Blog)

To view something interesting, click here

Try opening the link by opening it in a new tab

IMPORTANT: Make sure to open the link in **NEW TAB** so you do not navigate away from [http://web-vulns](\"http://web-vulns\")

⚡ Need a hint?

🎥 Video walkthrough

Back

Step 5/14

Next

Student

InfoSec

404 Not Found

web-vulns/xss_stored_1.php

Bugs

Change Password

Create User

Set Security Level

Reset

Welcome Admin

Logout

Cyber Academy

XSS - Stored (Blog)

To view something interesting, click here

Submit

Add: ☒

Show all: ☐

Delete: ☐

Your entry was added to our blog!

#	Owner	Date	Entry
1	admin	2025-03-06 23:36:52	XSS attack - stored
2	admin	2025-03-06 23:38:01	To view something interesting, click here

InfoSec - Mozilla

InfoSec - Mozilla

17:38

Figure 5. A second example of a stored XSS attack.

Kali Linux - Student

The Wireshark Network Analyzer

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

Welcome to Wireshark

Capture

...using this filter: Enter a capture filter ... All interfaces shown

eth0

any

Loopback: lo

bluetooth-monitor

nftlog

nftqueue

dbus-system

dbus-session

Learn

User's Guide

Wiki

Questions and Answers

Mailing Lists

SharkFest

Wireshark Discord

Donate

You are running Wireshark 4.4.2.

Ready to load or capture

No Packets

Profile: Default

INFOSEC

Exit Lab

Instructions Resources

network troubleshooting, analysis, software development, and communications protocol development. It is also employed for educational purposes and for intercepting network traffic. Like tcpdump, Wireshark captures packets, but it offers a graphical user interface along with advanced sorting and filtering features. Wireshark allows users to set network interface controllers to promiscuous mode, meaning all traffic on that interface— including unicast traffic not addressed to the NIC's MAC address— will be displayed. Similar to tcpdump, superuser privileges are required on some systems to run Wireshark.

sudo wireshark

Network Traffic

In the window that appears after running the command, select *eth0* as the network interface for Wireshark to sniff packets. This interface is chosen because both the student (in this case, a Telnet client) and the Telnet server are connected to this network. To start sniffing, click the Wireshark logo in the upper left corner.

Since Wireshark was initiated using the terminal, the terminal screen will appear to have frozen, not allowing other commands to be executed. To use the terminal you will need to open a *new terminal* session.

How does Wireshark differ from tcpdump?

Type one of the choices listed.

It can handle more data
It offers a graphical user interface
It is faster at processing

Check

Previous Next

58 Minutes Remaining

Figure 6. The Wireshark interface.

The screenshot displays an Ubuntu 20.4 desktop environment. A terminal window is open, showing the execution of the `iptables-restore` command to restore rules from a file named `/iptable-rules.txt`. The terminal output indicates that the rules were successfully restored and the chains were flushed. The desktop background is a stylized, low-poly geometric pattern in shades of purple and orange. On the left side, there is a vertical dock with icons for the Dash, Home, Applications, and various system utilities. The top of the screen shows the system menu with the date and time (Mar 9 14:26) and the network status (Ubuntu 20.4 - firewall).

On the right side of the screen, there is an "INFOSEC" lab interface. It contains instructions for the lab, a "Route" section explaining the purpose of the firewall, and a "Check" button to verify the network interface. The interface also shows a "Previous" and "Next" navigation bar and a "47 Minutes Remaining" timer.

```
ubuntu-user@ubuntu19: ~  
ubuntu-user@ubuntu19:~$ sudo sh -c 'iptables-restore -v < /iptable-rules.txt'  
[sudo] password for ubuntu-user:  
# Generated by iptables-save v1.8.4 on Mon Jun 28 11:53:48 2021  
Flushing chain 'INPUT'  
Flushing chain 'FORWARD'  
Flushing chain 'OUTPUT'  
# Completed on Mon Jun 28 11:53:48 2021  
ubuntu-user@ubuntu19:~$
```

INFOSEC

Instructions Resources

To begin this lab we must first load in our firewall rules with the following command:

```
sudo sh -c 'iptables-restore -v < /iptable-rules.txt'
```

When prompted for the password, type:

```
passw0rd!
```

Route

Since the firewall connects two entities (the webserver and the student), specific rules must be set to allow proper communication. To view the network, use the `route` command, which displays the host's routing table. Devices use these tables to make routing decisions for TCP/IP packets. On individual hosts, the routing table typically includes the default gateway, subnet, and active network interface.

Run the following command on the firewall:

```
route
```

The output reveals that the firewall is part of the 192.168.1.0/24 subnet. It also shows the routing for packets whose destination is not specified, which are sent to the default gateway 192.168.1.1

Which network interface is displayed in the output?

Check

Previous Next

47 Minutes Remaining

Figure 9. Restoring iptables rules from a text file.

ActivitiesTerminalUbuntu 20.4 - firewallMar 9 14:27

ubuntu-user

Trash

ubuntu-user@ubuntu19: ~

```
ubuntu-user@ubuntu19:~$ sudo sh -c 'iptables-restore -v < /iptables-rules.txt'
[sudo] password for ubuntu-user:
# Generated by iptables-save v1.8.4 on Mon Jun 28 11:53:48 2021
Flushing chain 'INPUT'
Flushing chain 'FORWARD'
Flushing chain 'OUTPUT'
# Completed on Mon Jun 28 11:53:48 2021
ubuntu-user@ubuntu19:~$ route

Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 192.168.1.1 0.0.0.0 UG 100 0 0 ens32
link-local 0.0.0.0 255.255.0.0 U 1000 0 0 ens32
192.168.1.0 0.0.0.0 255.255.255.0 U 100 0 0 ens32
ubuntu-user@ubuntu19:~$
```

INFOSECExit Lab

InstructionsResources

To begin this lab we must first load in our firewall rules with the following command:

```
sudo sh -c 'iptables-restore -v < /iptables-rules.txt'
```

When prompted for the password, type:

```
password!
```

Route

Since the firewall connects two entities (the webserver and the student), specific rules must be set to allow proper communication. To view the network, use the route command, which displays the host's routing table. Devices use these tables to make routing decisions for TCP/IP packets. On individual hosts, the routing table typically includes the default gateway, subnet, and active network interface.

Run the following command on the firewall:

```
route
```

The output reveals that the firewall is part of the 192.168.1.0/24 subnet. It also shows the routing for packets whose destination is not specified, which are sent to the default gateway 192.168.1.1

Which network interface is displayed in the output?

Check

PreviousNext

46 Minutes Remaining

Figure 10. The output of the "route" command.