01

NETWORKING / PACKET ANALYSIS (WIRESHARK LAB 4.5)

Wireshark TCP Three-Way Handshake Capture and PyShark Verification

Live packet capture of a SYN / SYN-ACK / ACK handshake to gaia.cs.umass.edu, cross-verified with a custom Python PyShark script

COURSE
Networking / Packet Analysis (Wireshark Lab 4.5)

DATE
17th October 2024

ENVIRONMENT
Local host at 192.168.0.119 (private home LAN, 192.168.0.0/24)

PAGES (REPORT)
3

STUDENT
Cody Richard

ORGANIZATION
Full Sail University

TECHNICAL BRIEF · EXECUTIVE SUMMARY

Wireshark TCP Three-Way Handshake Capture and PyShark Verification

For this lab I captured and dissected a live TCP three-way handshake in Wireshark, triggered by uploading the file alice.txt to gaia.cs.umass.edu via the UMass Wireshark-labs file-upload page (http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html). From the capture I identified the connection's source endpoint (my machine, 192.168.0.119 on ephemeral source port 38904) and destination endpoint (gaia.cs.umass.edu, 128.119.245.12 on port 80/HTTP), and isolated the SYN, SYN-ACK, and ACK exchange using the display filter tcp.flags.syn == 1 || tcp.flags.ack == 1. To independently confirm the findings I wrote a Python PyShark script that re-parsed an exported PCAP (Test.pcapng) and programmatically extracted the same IPs and ports from the first SYN packet. The script output matched the live capture exactly, double-verifying every answer.

▸ Objectives

- Capture a live TCP connection in Wireshark generated by an HTTP file upload to gaia.cs.umass.edu
- Identify the source IP address and TCP source port used by the local computer
- Identify the destination IP address and TCP destination port of the gaia.cs.umass.edu server
- Isolate and explain the TCP three-way handshake (SYN, SYN-ACK, ACK) using display filters
- Independently verify the captured connection details with a custom Python PyShark script

▸ Environment

```
Local host at 192.168.0.119 (private home LAN, 192.168.0.0/24)
```

```
Remote target gaia.cs.umass.edu at 128.119.245.12, port 80 (HTTP)
```

```
Wireshark live capture with TCP display filtering
```

```
Python 3 with the PyShark library parsing an exported PCAP
```

```
Exported capture file Test.pcapng located on the user's Desktop
```

▸ Tools

`Wireshark` `Python 3`

`PyShark (pyshark.FileCapture)`

`PCAP / Test.pcapng`

WALKTHROUGH & EVIDENCE

This lab captures and dissects a live TCP three-way handshake in Wireshark, triggered by uploading alice.txt to the UMass Wireshark-labs file-upload endpoint on gaia.cs.umass.edu. I isolate the SYN / SYN-ACK / ACK exchange with a tcp.flags display filter, then independently re-verify every endpoint by parsing an exported PCAP with a custom Python PyShark script. Both methods agree exactly, double-confirming the connection's source and destination IPs and ports.

CAPTURE

Live Handshake in Wireshark

Traffic was generated by uploading alice.txt to http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html, forcing a fresh TCP connection. The capture surfaces the complete SYN / SYN-ACK / ACK exchange between my host and the gaia server on port 80.

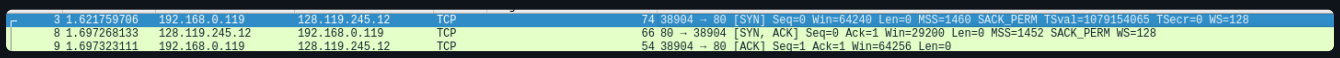


FIG 01 — The filtered Wireshark capture showing the full three-way handshake: SYN from 192.168.0.119:38904 to 128.119.245.12:80 (Seq=0, Win=64240), the server's SYN, ACK reply from 128.119.245.12 (Seq=0 Ack=1, MSS=1452), and the final ACK from 192.168.0.119 (Seq=1 Ack=1) completing the connection over HTTP/port 80.

FILTERING

Isolating the Handshake

A single TCP-flags display filter reduces the live capture to exactly the handshake packets, hiding all other traffic so the SYN and ACK rows are immediately readable.

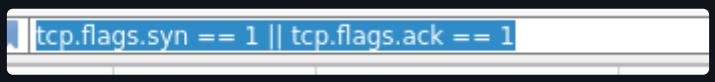
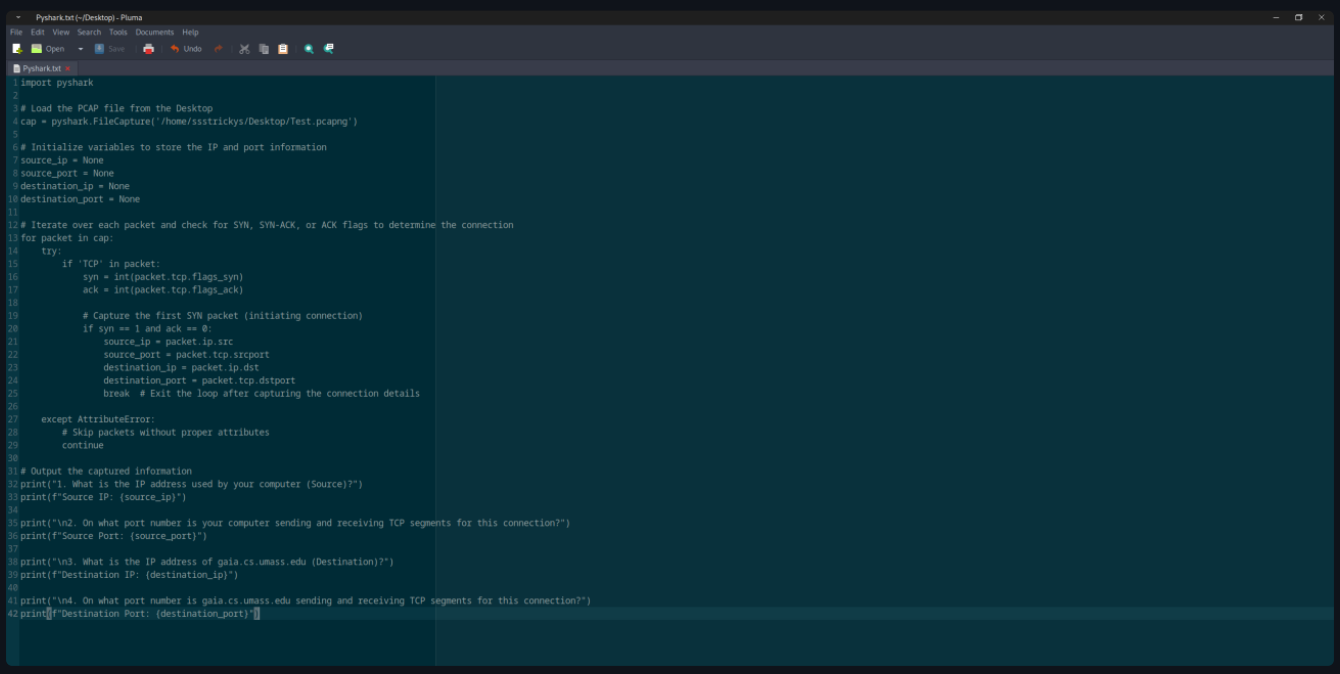


FIG 02 — The Wireshark display filter tcp.flags.syn == 1 || tcp.flags.ack == 1, which matches any segment with the SYN or ACK flag set in the TCP header and filters out the rest of the traffic.

VERIFICATION

PyShark Cross-Check Script

To independently validate the live findings, I wrote a Python PyShark script that re-parses the exported PCAP offline, isolates the first SYN packet, and extracts the same four endpoint values the lab asks for.



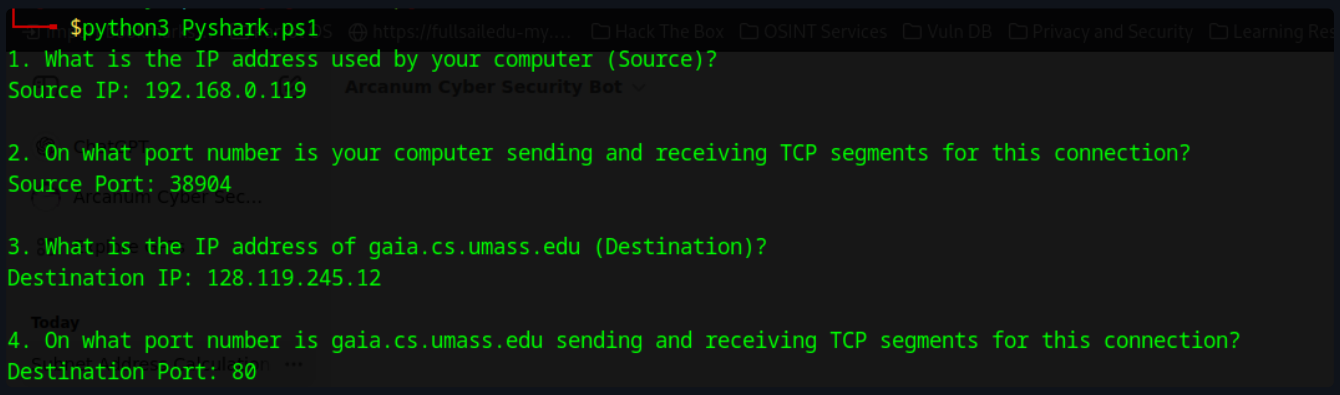
```
import pyshark

# Load the PCAP file from the Desktop
cap = pyshark.FileCapture('/home/ssstrickys/Desktop/Test.pcapng')

# Initialize variables to store the IP and port information
source_ip = None
source_port = None
destination_ip = None
destination_port = None

# Iterate over each packet and check for SYN, SYN-ACK, or ACK flags to determine the connection
for packet in cap:
    try:
        if 'TCP' in packet:
            syn = int(packet.tcp.flags_syn)
            ack = int(packet.tcp.flags_ack)

            # Capture the first SYN packet (initiating connection)
            if syn == 1 and ack == 0:
                source_ip = packet.ip.src
                source_port = packet.tcp.srcport
                destination_ip = packet.ip.dst
                destination_port = packet.tcp.dstport
                break  # Exit the loop after capturing the connection details

    except AttributeError:
        # Skip packets without proper attributes
        continue

# Output the captured information
print("1. What is the IP address used by your computer (Source)?")
print(f"Source IP: {source_ip}")

print("\n2. On what port number is your computer sending and receiving TCP segments for this connection?")
print(f"Source Port: {source_port}")

print("\n3. What is the IP address of gaia.cs.umass.edu (Destination)?")
print(f"Destination IP: {destination_ip}")

print("\n4. On what port number is gaia.cs.umass.edu sending and receiving TCP segments for this connection?")
print(f"Destination Port: {destination_port}")
```

FIG 03 — The PyShark source (Pyshark.txt): pyshark.FileCapture loads Test.pcapng from the Desktop, iterates each packet, casts tcp.flags_syn and tcp.flags_ack to int, and on the first SYN (syn == 1 and ack == 0) records ip.src, tcp.srcport, ip.dst, and tcp.dstport before breaking and printing an answer for each lab question.

RESULTS

Independent Confirmation

Running the script against the exported PCAP reproduced every value read from the live Wireshark capture, double-verifying the connection details.



```
$python3 Pyshark.ps1
1. What is the IP address used by your computer (Source)?
Source IP: 192.168.0.119

2. On what port number is your computer sending and receiving TCP segments for this connection?
Source Port: 38904

3. What is the IP address of gaia.cs.umass.edu (Destination)?
Destination IP: 128.119.245.12

4. On what port number is gaia.cs.umass.edu sending and receiving TCP segments for this connection?
Destination Port: 80
```

FIG 04 — Terminal output from running python3 Pyshark.ps1: Source IP 192.168.0.119, Source Port 38904, Destination IP 128.119.245.12, Destination Port 80 — a one-to-one match with the live Wireshark capture.