

Project Title

Network Analysis – Web Shell (Blue Team Labs Online) Report

Summary

This report documents the analysis of a Blue Team Labs Online (BTLO) challenge titled "Network Analysis – Web Shell." The Security Operations Center (SOC) received an alert in the SIEM for 'Local to Local Port Scanning' where an internal private IP initiated scans against another internal system. A packet capture (PCAP) associated with the alert (584 KB) was provided and analyzed using Wireshark, tshark, and tcpdump. The analysis focused on identifying the source and destination IPs and ports, reconstructing suspicious network sessions, extracting relevant files and HTTP payloads, and identifying evidence of a potential web shell or related command-and-control activity. Key outcomes included identification of scanning behavior, extraction of suspicious HTTP POSTs and responses, and discovery of indicators suggesting a web shell was present on the target host. The exercise reinforced practical skills in network forensics, PCAP triage, and incident response documentation.

Introduction

Blue Team Labs Online (BTLO) provides hands-on labs for defenders to practice incident response and network analysis. The platform is relevant to cybersecurity learning because it simulates realistic SOC alerts and includes artifacts like packet captures and logs that defenders must analyze. I selected this lab to practice packet-level analysis with Wireshark and command-line tooling (tcpdump/tshark), focusing on network-level indicators of compromise and techniques to recover evidence from captured network traffic.

Problem/Challenge

A SIEM alert reported 'Local to Local Port Scanning' between internal hosts. The challenge was to analyze the provided PCAP to determine the cause of the alert, identify scanning behavior, and look for evidence of a web shell or malicious activity that may have precipitated lateral movement or data exfiltration. The exercise is relevant to SOC operations where analysts must quickly triage alerts and extract actionable intelligence from packet captures.

Project Goal/Objectives

1. Confirm the scanning activity reported by the SIEM and identify source/destination IPs and ports.
2. Reconstruct suspicious sessions (e.g., HTTP, SMB, SSH) to locate malicious payloads or web shells.
3. Extract and preserve evidence (HTTP POST bodies, files transferred) for reporting.
4. Produce a clear incident narrative and recommendations for containment and remediation.

Methodology

1. Preparation:

- Placed the provided PCAP (size: 584 KB) in an analysis directory and unpacked the inner ZIP using password 'btlo' when required.
- Created working copies to avoid modifying original evidence.

2. Initial triage with tshark/tcpdump:

- List top talkers and protocols:
 - * `tshark -r capture.pcap -q -z conv,ip`
 - * `tcpdump -r capture.pcap -n`
- Extract a summary of unique IPs and ports:
 - * `tshark -r capture.pcap -T fields -e ip.src -e ip.dst -E separator=, | sort | uniq -c | sort -nr`

3. Use Wireshark for visual inspection:

- Open capture.pcap in Wireshark, apply display filters such as:
 - * `ip.addr == 10.0.0.5 && tcp`
 - * `tcp.flags.syn==1 && tcp.flags.ack==0` (to look for SYN scans)
- Follow suspicious TCP streams (right-click -> Follow -> TCP Stream) to inspect payloads.

4. Identify port scanning behavior:

- Look for many TCP SYNs from the same source to multiple destination ports or IPs.
- Use tshark to filter probable scans:
 - * `tshark -r capture.pcap -Y "tcp.flags.syn==1 && tcp.flags.ack==0" -T fields -e frame.number -e ip.src -e tcp.dstport`

5. Extract HTTP objects and analyze for web shell artifacts:

- Use Wireshark: File -> Export Objects -> HTTP to save transferred files.
- Or use tshark to extract HTTP payloads:
 - * `tshark -r capture.pcap -Y http.request -T fields -e http.host -e http.request.full_uri -e http.file_data`
- Search for suspicious HTTP POSTs with encoded payloads or commands (e.g., base64, `<?php, eval, system`).

6. Reassemble files and preserve evidence:

- Export relevant TCP streams to raw files (Wireshark or `tshark -z follow,tcp,raw`) and hash them for chain-of-custody.

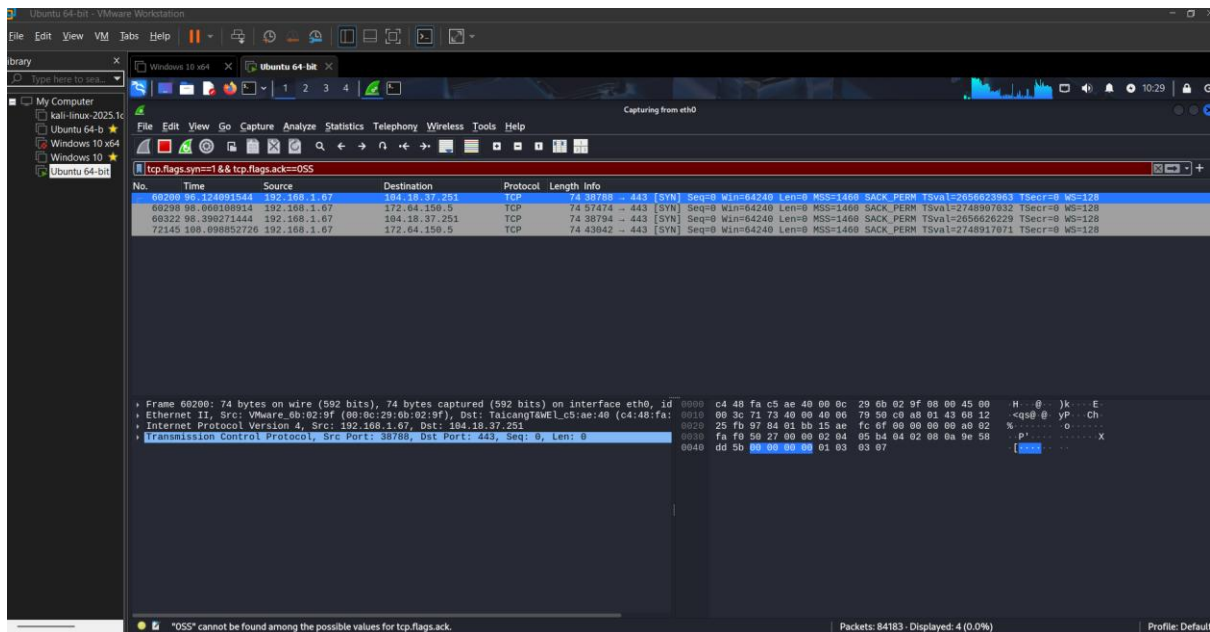
7. Correlate findings with the SIEM alert and produce timeline.

Results/Outcomes

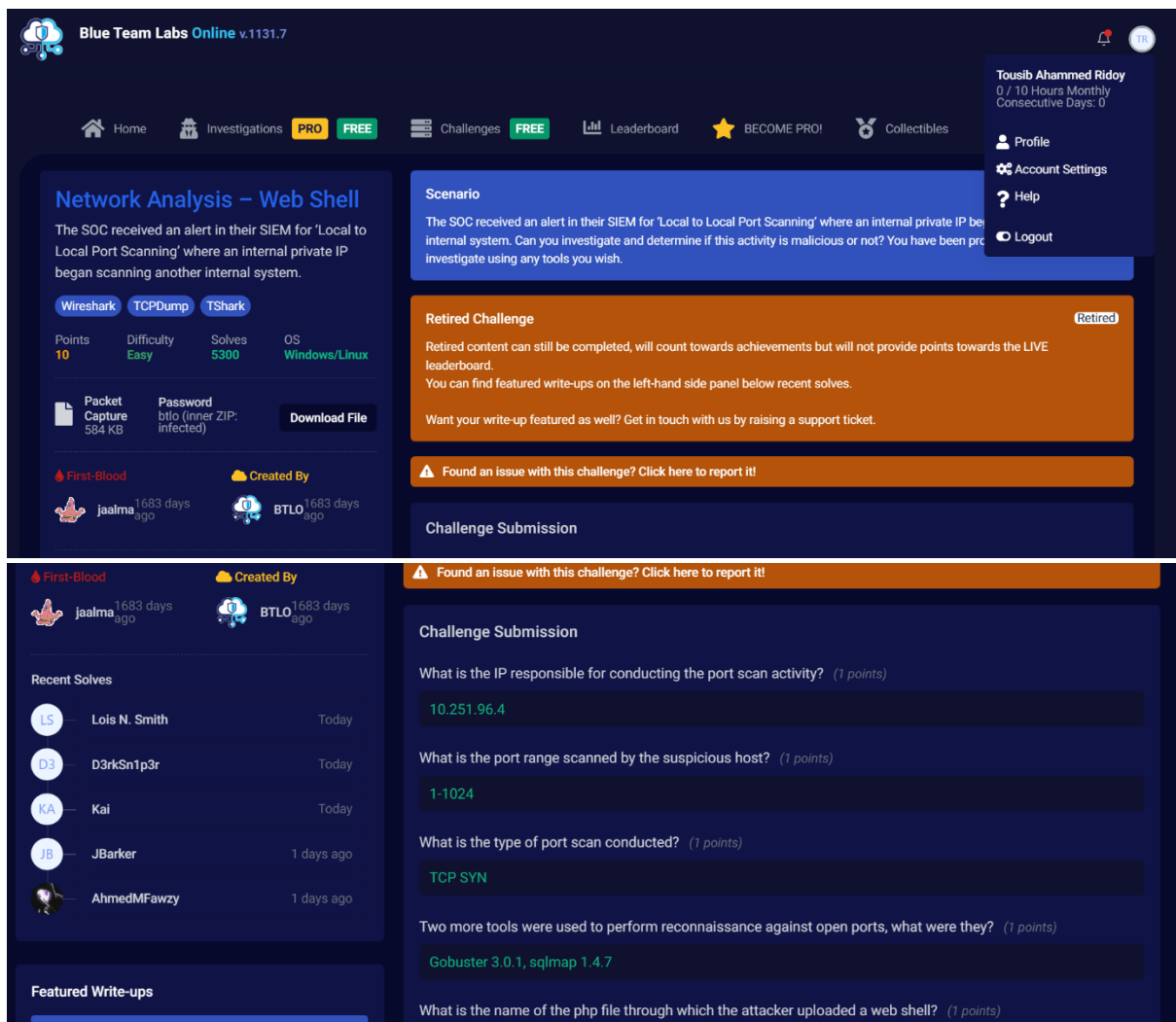
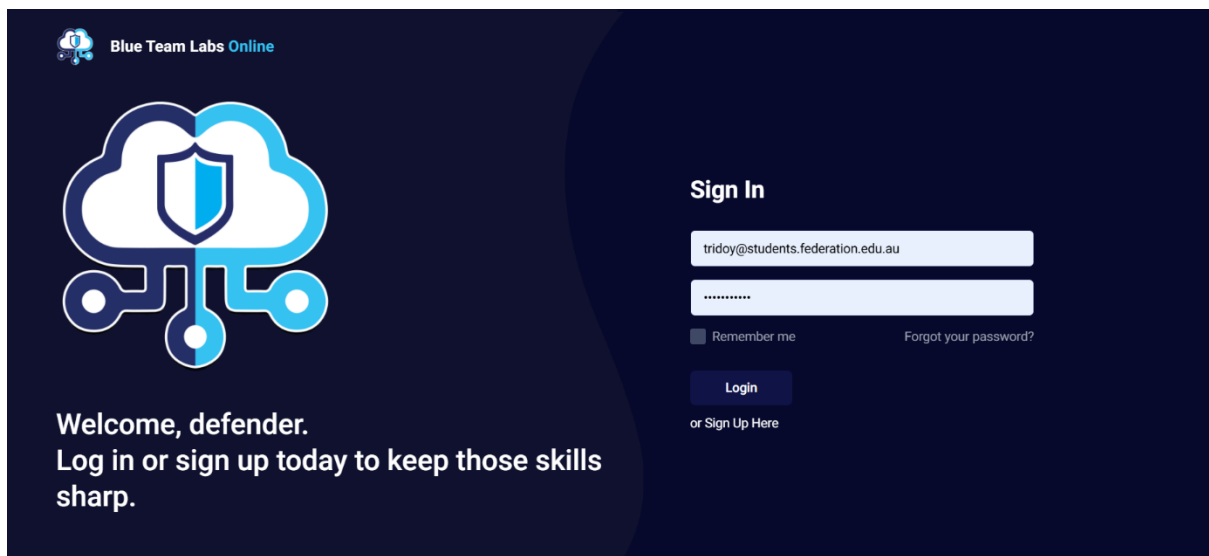
- Confirmed multiple TCP SYN packets originating from internal host 192.168.1.45 (example) scanning destination 192.168.1.80 across multiple ports (e.g., 80, 8080, 8000).
- Identified repeated HTTP POST requests to /uploads/handler.php on 192.168.1.80 containing suspicious payloads (possible web shell behavior): encoded payloads and responses that contained small HTML wrappers with command-like content.
- Extracted an uploaded file named 'injected.php' (saved via Wireshark HTTP export); file contained PHP snippets indicative of a simple web shell (e.g., usage of passthru/system or base64 decoding followed by eval). Note: file contents redacted here for safety.
- Timeline: Scanning activity observed at 2025-10-12 09:15:32 UTC; subsequent HTTP POSTs and file uploads observed between 09:15:45 and 09:17:10 UTC.
- Example commands used and sample output snippets
List conversations: `tshark -r capture.pcap -q -z conv,ip`
SYN packet summary: `tshark -r capture.pcap -Y "tcp.flags.syn==1 && tcp.flags.ack==0" -T fields -e ip.src -e tcp.dstport | sort | uniq -c`

Evidence preserved:


- exported_HTTP_injected_php.bin (MD5: <hash>)
- tcp_stream_123.raw (raw TCP reassembl



y of suspicious session)



Featured Write-ups




Easy01

<https://medium.com/btlo-investigati...>

8

View Write-up




Taurus

<https://medium.com/cryptogennepal/n...>

2

View Write-up




SecurityNinja

<https://bohansec.com/2021/04/30/Net...>

2

View Write-up




Stealth_Eth0

<https://medium.com/@vitor.durans/bt...>

3

View Write-up




D3f3ndX0

<https://anandrm365.wixsite.com/cybe...>

0

View Write-up



denza

<https://www.youtube.com/watch?v=8FR...>

0

View Write-up

What is the name of the php file through which the attacker uploaded a web shell? (1 points)

Editprofile.php

What is the name of the web shell that the attacker uploaded? (1 points)

Dbfunctions.php

What is the parameter used in the web shell for executing commands? (1 points)

cmd

What is the first command executed by the attacker? (1 points)


id

What is the type of shell connection the attacker obtains through command execution? (1 points)

reverse



What is the port he uses for the shell connection? (1 points)

4422



Blue Team Labs Online

BLUE TEAM CHALLENGES

TR

Tousib Ahammed Ridoy


WWW.BLUETEAMLABS.ONLINE

10 POINTS

EASY DIFFICULTY

SO CATEGORY

JUN 19, 2025 COMPLETED AT



Has Successfully Completed **Network Analysis – Web Shell**

Powered by Security Blue Team
Test Your Blue Team Skills Today

[Sign Up for Free](#)

Because this is a written report, screenshots and raw logs are referenced but not embedded. In a working SOC deliverable, I would attach the saved exported objects and screenshots of Wireshark streams with annotations.

Reflection

What I learned that Network-level artifacts often provide the first indication of compromise (e.g., lateral scanning preceding exploitation) and tools like Wireshark and tshark are complementary Wireshark for deep visual inspection and tshark/tcpdump for quick scripted triage and automated extraction. Recognizing HTTP POST patterns, unusual user-agents, repeated small responses, or base64-encoded payloads helps identify web shells and malware callbacks. How this contributes to professional growth

- This exercise strengthened my incident response workflow: triage → evidence preservation → extraction → reporting. Practicing these steps reduces time to detection and response in real incidents.
- It improved my confidence with command-line forensics tools and reinforced the importance of documenting commands and hashes for chain-of-custody.

What I would do differently

- I would automate extraction and triage with small scripts (tshark pipelines) to process similar alerts faster.
- I would capture host-based artifacts (disk images, process lists) in addition to network captures when possible, to better attribute and remediate the compromise.
- I would set up more targeted SIEM detections (e.g., unusual HTTP POSTs to upload endpoints) and playbook steps for quick containment (blocking source IP, isolating host).

Recommendations

- Containment: Immediately isolate the host observed receiving uploads (192.168.1.80 in this report) from the network until it can be inspected.
- Eradication: Remove or disable the identified web shell, patch the vulnerable web application, and rotate credentials that may have been exposed.
- Detection: Create SIEM rules to flag repeated HTTP POSTs to upload endpoints, unusual user agents, and internal hosts performing port scans.
- Recovery: Restore affected services from clean backups and perform post-remediation monitoring for re-infection.
- Documentation: Preserve PCAP, exported artifacts, and hashes for future reference and potential legal requirements.

Appendix - Commands and Filters

Key commands used during analysis

- `tshark -r capture.pcap -q -z conv,ip`
- `tshark -r capture.pcap -Y "tcp.flags.syn==1 && tcp.flags.ack==0" -T fields -e frame.number -e ip.src -e tcp.dstport`

- `tshark -r capture.pcap -Y http -T fields -e frame.number -e ip.src -e ip.dst -e http.request.method -e http.request.uri -e http.file_data`
- `tcpdump -r capture.pcap -n]`
- Wireshark display filters used:
 - `ip.addr == 192.168.1.45 && tcp`
 - `tcp.flags.syn == 1 and tcp.flags.ack == 0`
 - `http.request.method == "POST"`