A Purple Team Security Assessment of a Maturity Level 1 Healthcare Provider: Detection Engineering and Validation on a Segmented Enterprise Network

Final Project Exercise Logs

Group 2

, Abdul-Ramon Adeyemi Folarin, &

, , , &

School of Information Technology Administration & Security, Seneca Polytechnic

SPR708NBB: Attack and Defense

Professor Ahmad Ghobeiti

August 12, 2026

Student Integrity Statement: I/we declare that the attached assignment is my/our own work in accordance with the Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) or distributed to other students

Table of Contents

How to Read This Log 3
Pass 1: Baseline Execution (Gaps Open) 4
Preparation Verification 4
Remediation Log 9
Gap 1: File Integrity Monitoring on websrv-01 9
Gap 2: Web Application Firewall for Mutillidae 9
Gap 3: SIEM SSH Restriction and Self-Protection Rule 9
Gap 4: Baseline Backup 9
Pass 2: Validation (Gaps Closed) 11
Detection Matrix Summary 13

How to Read This Log

This document records every action taken during both passes of the exercise: the baseline pass (Pass 1, gaps open) and the validation pass (Pass 2, gaps closed). Each entry follows the SCYTHE PTEF format: the technique is stated, the Red Team action and Blue Team observation are recorded separately, and the detection outcome is assessed against three questions: Was it prevented? Was it alerted? Was it logged?

Outcome codes: Prevented means the action was technically stopped before it could succeed. Alerted means the SIEM or IDS generated an actionable alert. Logged only means the event appears in logs but generated no alert. Not detected means no trace was found in any monitoring source. Partial means a control provided some barrier but did not fully prevent or detect the activity.

Colour code	Meaning
Green (Prevented / Alerted)	Control is working correctly
Yellow (Logged only / Partial)	Some visibility but gap remains
Red (Not detected)	No coverage; finding raised

Pass 1: Baseline Execution (Gaps Open)

The attack chain was run once with the four identified control gaps left open. The goal was to produce a realistic before picture: what does the SIEM actually see, what does it miss, and where does the chain advance without resistance? Honest documentation of misses is the primary output of this pass.

Preparation Verification

Before executing the chain, the monitoring stack was confirmed operational.

```
admin-01: sudo docker ps | grep -i wazuh
core-01: sudo systemctl status suricata --no-pager
admin-01: sudo docker exec single-node_wazuh.manager_1 /var/ossec/bin/agent_control -l
```

Result: Wazuh manager, indexer, and dashboard running. Suricata active on core-01. Agents active: dc-dhcp-01 (ID 001), websrv-01 (ID 003). Agents disconnected: websrv-02 (ID 002), user-01 (ID 004). FIM on websrv-01 confirmed not configured for attack-surface paths. Evidence: Figs. 2 through 7.

Step 1 | Reconnaissance	**T1595 Active Scanning**
Timestamp	2026-07-31 15:01 EDT
Machines	extattacker to websrv-01
Tools / commands	nmap -sV -sC -p- -T4 172.16.22.194; nikto -h http://172.16.22.194:8082/
Red Team action	Ran a full-port SYN scan and service enumeration against websrv-01 from the external attacker host. Followed with a Nikto web crawl of the Mutillidae application on port 8082. Scan revealed ports 22, 80, 139, 445, 8082, and a .git repository and /passwords/ directory listing exposed on port 80.
Blue Team observation	Queried eve.json on core-01 for alert events from the attacker source IP during the scan window. No alert events were returned for either the Nmap or Nikto activity.
Detection outcome	Not detected
Evidence	Figs. 8, 9. No Suricata scan alert generated. Gap: no scan-detection rule loaded on the IDS sensor.

Step 2 | Initial Access	**T1190 Exploit Public-Facing Application; T1505.003 Web Shell**
Timestamp	2026-08-01 22:42 UTC
Machines	extattacker to websrv-01
Tools / commands	Browser and curl; DNS Lookup command injection on Mutillidae; base64-encoded PHP payload

Red Team action	Exploited the Mutillidae DNS Lookup page (Security Level 0) via command injection. Because the web root (/app) was not writable by www-data, used a base64-encoded payload to write a minimal PHP shell to /tmp/shell.php. Executed the shell via the Arbitrary File Inclusion vulnerability at index.php?page=/tmp/shell.php. Confirmed RCE with the id command returning uid=33(www-data).
Blue Team observation	Checked Wazuh Security Events for websrv-01 during the exploitation window. No FIM alert for /tmp/shell.php creation. Confirmed the shared agent.conf had no syscheck block covering /tmp or the web root. The exploit request was not blocked because Mutillidae was not behind a WAF.
Detection outcome	Not detected
Evidence	Figs. 10, 11, 12, 13. Web shell confirmed active. No WAF block. No FIM alert.

Step 3 | Execution / Persistence	**T1059 Command and Scripting Interpreter; T1053.003 Scheduled Task/Cron**
Timestamp	2026-08-01 22:47 UTC
Machines	websrv-01
Tools / commands	curl via the web shell; crontab injection via the shell
Red Team action	Issued whoami, uname -a, and id through the web shell to confirm interactive execution. Added a cron job via the shell using the (crontab -l; echo ...) | crontab - pattern, setting a five-minute beacon to the attacker host. Confirmed crontab -l showed the entry.
Blue Team observation	Monitored Wazuh for any rootcheck, process-creation, or cron-related alert on websrv-01. No alerts generated for either the command execution or the persistence mechanism.
Detection outcome	Not detected
Evidence	Figs. 14, 15. Cron entry confirmed written. No SIEM alert for command execution or persistence.

Step 4 | Credential Access	**T1552.001 Credentials in Files**
Timestamp	2026-08-01 22:48 UTC
Machines	websrv-01
Tools / commands	curl via the web shell; cat on /app/includes/database-config.php

Red Team action	Recovered the database configuration file through the web shell. The file exposed DB_HOST (127.0.0.1), DB_USERNAME (root), and DB_PASSWORD (blank) in plaintext. An adjacent HTML comment disclosed the fallback credential "samurai". Also read /etc/passwd to enumerate system accounts and identify msingh827.
Blue Team observation	No endpoint command-logging was in place on websrv-01. The file read generated no alert. The credentials were noted and used in subsequent steps.
Detection outcome	Not detected
Evidence	Fig. 16. Credentials in /app/includes/database-config.php confirmed. HTML comment exposed fallback credential. No alert.

Step 5 | Lateral Movement	**T1021.004 Remote Services: SSH; T1078 Valid Accounts**
Timestamp	2026-08-01 21:41 EDT
Machines	extattacker to admin-01
Tools / commands	ssh msingh827@172.16.44.66
Red Team action	Attempted SSH from extattacker to admin-01 using the msingh827 account. The connection reached the host and returned a Verification code prompt (TOTP MFA), followed by a password prompt. Multiple attempts failed due to MFA. Lateral movement to the SIEM was partially blocked by multi-factor authentication.
Blue Team observation	Checked auth.log on the admin-01 host directly (not via the SIEM container, which lacks a bind-mount to the host log). The host log showed a PAM authentication failure from 172.16.22.3. The SIEM generated no alert because the manager container does not ingest the host auth.log. No level-12 rule existed at this point.
Detection outcome	Partial
Evidence	Figs. 17, 18. SSH reached the host and was challenged by MFA. SIEM blind to its own authentication log. Gap: dual-role exposure and missing self-protection rule.

Step 6 | Defense Impairment	**T1562.001 Impair Defenses: Disable or Modify Tools**
Timestamp	2026-08-01 21:33 EDT
Machines	admin-01
Tools / commands	sudo docker stop single-node_wazuh.manager_1; sudo docker start single-node_wazuh.manager_1

Red Team action	From the SIEM host (logged in as msingh827 with sudo), stopped the Wazuh manager container to simulate blinding the monitoring stack. Immediately restored it. Suricata runs on core-01, not admin-01, so it was unaffected by this step. NOTE: This action disconnected all four enrolled agents simultaneously. The Windows DC agent did not automatically reconnect.
Blue Team observation	The stop event was not captured because the SIEM itself was the target. The unintended consequence of all agents disconnecting was observed after restart. The DC agent (ID 001) remained disconnected for the remainder of Pass 1.
Detection outcome	Not detected
Evidence	Fig. 19. Wazuh stop and restart captured. Cascading agent disconnect documented as operational consequence and additional finding (F-11 monitoring resilience).

Step 7 | Lateral Movement	**T1110.003 Password Spraying; T1021.002 SMB/Windows Admin Shares**
Timestamp	2026-08-01 19:42 EDT
Machines	extattacker to dc-dhcp-01
Tools / commands	netexec smb 172.16.44.130 -u /tmp/users.txt -p <password>
Red Team action	Ran a controlled password spray against the Domain Controller using a username list derived from credentials recovered in Step 4 and system accounts. One spray attempt with the password recovered from the lab environment succeeded for the administrator account, returning Pwn3d! Enumerated shares (ADMIN$, C$, NETLOGON, SYSVOL all with READ/WRITE) and all 11 domain user accounts. Null authentication was also confirmed enabled on the DC (Null Auth: True).
Blue Team observation	DC agent was disconnected at this point due to the Step 6 restart. No Windows logon events (4624/4625) or Sysmon network-connection events reached the SIEM. The domain administrator compromise was completely invisible to the monitoring stack.
Detection outcome	Not detected
Evidence	Figs. 20, 21, 22. Full domain admin access confirmed. Zero SIEM telemetry from the DC. Additional finding: Null Auth enabled (F-08).

Step 8 | Impact	**T1486 Data Encrypted for Impact (simulation)**
Timestamp	2026-08-02 00:39 UTC
Machines	websrv-01 (Samba share)

Tools / commands	bash loop; cp with .locked extension; echo ransom note
Red Team action	Ran the ransomware simulation against /srv/files/home/msingh827/upload. Copied IMPORTANT.txt and proof.txt to .locked variants and wrote README_RESTORE.txt. Confirmed the .locked files and ransom note were present in the share directory. No real encryption was performed.
Blue Team observation	Checked Wazuh for FIM alerts on websrv-01 during the file-change window. No FIM alert was generated because syscheck was not configured to watch the share path. The share was confirmed accessible and modified without any detection.
Detection outcome	Not detected
Evidence	Fig. 23. Ransomware simulation files confirmed. No FIM alert from the SIEM. Gap: syscheck not watching the share path.

Remediation Log

Following the baseline pass, four control gaps were closed. Each change is documented here with the commands applied and the verification step that confirmed the fix before the validation pass.

Gap 1: File Integrity Monitoring on websrv-01

Added syscheck directories targeting the Samba share and web root, fixed FIM database ownership, and set the scan frequency to 60 seconds.

```
# websrv-01
sudo sed -i 's|<directories>/bin,/sbin,/boot</directories>|<directories>/bin,/sbin,/boot</directories>\n    <directories
check_all="yes" realtime="no">/srv/files/home/msingh827/upload</directories>\n    <directories check_all="yes"
realtime="no">/var/www/html</directories>|' /var/ossec/etc/ossec.conf
sudo sed -i 's|<frequency>43200</frequency>|<frequency>60</frequency>|' /var/ossec/etc/ossec.conf
sudo chown wazuh:wazuh /var/ossec/queue/fim/db/fim.db
sudo systemctl restart wazuh-agent
```

Outcome: FIM confirmed monitoring /srv/files/home/msingh827/upload and /var/www/html at 60-second intervals. Realtime mode was switched to scheduled because the Samba filesystem does not support inotify. Evidence: Figs. 24, 25.

Gap 2: Web Application Firewall for Mutillidae

Placed nginx+ModSecurity reverse proxy in front of Mutillidae on port 80 using the existing ModSecurity installation and the OWASP CRS rules folder.

```
# websrv-01 - added custom rules to /etc/nginx/modsec/rules-mutillidae.conf
SecRuleEngine On
SecRule ARGS "@rx (?i)(;|\\|)`\\s*(id|whoami|uname|cat|ls|echo|wget|curl|chmod|python|perl|sh|bash)"
"id:700001,phase:2,deny,status:403"
SecRule ARGS "@rx (?i)base64\\s*-d" "id:700002,phase:2,deny,status:403"
sudo nginx -t && sudo systemctl reload nginx
```

Outcome: HTTP 403 returned for the command-injection payload on re-run. ModSecurity loaded 51 rules. Evidence: Fig. 26.

Gap 3: SIEM SSH Restriction and Self-Protection Rule

Added a custom Wazuh rule (ID 100200) for SSH logins to the SIEM host and restricted SSH at the host firewall.

```
# admin-01 - inside the manager container
cat > /var/ossec/etc/rules/local_rules.xml << EOF
<group name="siem_protection,">
  <rule id="100200" level="12">
    <if_sid>5715</if_sid>
    <description>SSH login accepted on SIEM host admin-01 - possible lateral movement</description>
  </rule>
</group>
EOF
# validated with wazuh-logtest - rule 100200 fires at level 12
```

Outcome: wazuh-logtest confirmed rule 100200 fires at level 12 for any accepted SSH auth on admin-01. SSH from extattacker returns Permission denied (keyboard-interactive). Evidence: Figs. 27.

Gap 4: Baseline Backup

```
# websrv-01
sudo mkdir -p /opt/backups
sudo tar czf /opt/backups/webroot_baseline.tar.gz /var/www/html
```

```
sudo tar czf /opt/backups/share_baseline.tar.gz /srv/files/home/msingh827/upload
ls -lh /opt/backups/
```

Outcome: Baseline backups created at /opt/backups. Recovery confirmed by restoring the share after Pass 2 simulation. Evidence: Fig. 28.

Pass 2: Validation (Gaps Closed)

The same chain was re-run with the four gaps closed to produce direct before-and-after evidence. Only the steps with a remediable gap were fully re-run; steps where no fix was applicable were documented as unchanged.

Step 2 (re-run) | Initial Access	T1190 Exploit Public-Facing Application
Timestamp	2026-08-02 01:32 UTC
Machines	extattacker to websrv-01
Tools / commands	curl with the same DNS Lookup injection payload
Red Team action	Replayed the exact command-injection payload used in Step 2 Pass 1 against port 80, which is now proxied through ModSecurity.
Blue Team observation	The WAF returned HTTP 403 Forbidden. The ModSecurity rules correctly matched the semicolon-separated shell command pattern. The exploit was blocked before reaching the application.
Detection outcome	Prevented
Evidence	Figs. 29, 30. HTTP 403 confirmed. Same payload that succeeded in Pass 1 is now blocked.

Step 2/8 (FIM validation) | Persistence / Impact	T1505.003 Web Shell; T1486 Impact (simulation)
Timestamp	2026-08-02 01:58 UTC
Machines	websrv-01
Tools / commands	sudo touch; syscheck forced via agent_control; curl for FIM alert query
Red Team action	Created a test file at /srv/files/home/msingh827/upload/shell_test.php to trigger FIM. Ran the ransomware simulation again (cp to .locked, ransom note) after a fresh syscheck baseline.
Blue Team observation	The Wazuh alerts log (/var/ossec/logs/alerts/alerts.log) on the manager recorded a syscheck_entry_modified alert for the share path, confirming FIM is now generating events. The ossec.log confirmed syscheck monitored the path on each 60-second cycle.
Detection outcome	Alerted
Evidence	Figs. 31. FIM alert confirmed in the manager alert log. Evidence of detection that was absent in Pass 1.

Step 5 (re-run) | Lateral Movement	T1021.004 SSH; T1078 Valid Accounts
Timestamp	2026-08-02 22:00 EDT
Machines	extattacker to admin-01
Tools / commands	ssh -o ConnectTimeout=5 -o BatchMode=yes msingh827@172.16.44.66; wazuh-logtest
Red Team action	Repeated the SSH attempt from extattacker to admin-01.
Blue Team observation	SSH returned Permission denied (keyboard-interactive), confirming the host still rejects the login. The wazuh-logtest utility was used to validate that rule 100200 fires at level 12 against a representative auth.log line, confirming the detection capability is in place even when the SSH is not authenticated.
Detection outcome	Prevented / Alerted
Evidence	Fig. 32. SSH denied. Rule 100200 validated at level 12 via logtest.

Step 8 (re-run) | Impact	T1486 Data Encrypted for Impact (simulation)
Timestamp	2026-08-02 02:01 UTC
Machines	websrv-01
Tools / commands	bash loop; tar restore; rm of .locked files
Red Team action	Re-ran the ransomware simulation against the Samba share: copied files to .locked extension and wrote README_RESTORE.txt.
Blue Team observation	Restored the share from the baseline backup at /opt/backups/share_baseline.tar.gz. Removed .locked files and README_RESTORE.txt. Confirmed the share returned to its clean pre-incident state with only the original IMPORTANT.txt and proof.txt present.
Detection outcome	Alerted / Recovered
Evidence	Fig. 33. Simulation re-run and recovery from backup confirmed. Before and after listing captured in one terminal screenshot.

Detection Matrix Summary

The table below consolidates the prevent/alert/log outcome for every technique across both passes. P = Prevented, A = Alerted, L = Logged only, N = Not detected.

Technique	Expected detection	Pass 1	Pass 2
T1595 Active Scanning	Suricata scan signature	N	N (rule pending)
T1190 Exploit + T1505.003 Web Shell	WAF block; FIM on shell path	N	P + A
T1059 Execution; T1053.003 Cron	Endpoint command logging; rootcheck	N	N (out of scope)
T1552.001 Credentials in Files	Endpoint file-access monitoring	N	N (out of scope)
T1021.004 SSH to SIEM; T1078	SSH auth rule; MFA	Partial	P + A
T1562.001 Impair Defenses	Service-stop detection	N	N (self-referential)
T1110.003 Spray; T1021.002 SMB DC	Windows logon 4624/4625; Sysmon	N	Not live-validated
T1486 Impact (simulation)	FIM mass-change on share	N	A + Recovered