Splunk SIEM Lab. Building an Investigation Dashboard and Hunting Web Attack Activity

Overview

In this Blue Team Level 1 lab, I worked from an operational Splunk investigation dashboard and used a SOC style workflow to validate alerts, pivot into raw events, correlate across Suricata and FortiGate UTM telemetry, and improve the dashboard so day to day triage becomes faster and more reliable. The emphasis was evidence first. Every conclusion was tied back to a log artifact.

SOC handoff summary

Classification: Multiple security events observed, scanning and exploitation attempts, plus a reverse shell alert requiring immediate validation
Highest priority lead: FortiGate reverse shell alert tied to internal server host we5201srv.waynecorpinc.local
Supporting leads: Web recon targeting phpinfo endpoints, vulnerability exploit attempts with CVE reference, automated scanner fingerprinting, severity distribution reporting
Confidence: High on observed activity, Medium on impact without host level confirmation in this lab dataset

Environment and telemetry sources

Platforms used
Splunk for investigation and dashboarding
Suricata IDS alerts and HTTP telemetry
FortiGate UTM threat logs and reference links

Investigation approach

1. Start in the dashboard to identify the highest signal alerts and categories.
2. Pivot into the underlying events to confirm the artifact that proves the alert.
3. Use focused searches to reduce noise and answer one question at a time.
4. Cross check key details between Suricata and FortiGate logs to avoid single source conclusions.
5. Add or modify dashboard panels only after searches are correct and stable.

Key investigation highlights

1. Web recon and attempted information leak via phpinfo access
   I pivoted from a Suricata “Information Leak” alert into the raw events and validated the exact request details.

Evidence confirmed
Source IP: 40.80.148.42
Destination IP: 192.168.250.70
Signature: ET WEB SERVER WEB PHP phpinfo access
Targeted paths: imreallynotbatman.com/phpinfo.php and imreallynotbatman.com/phpinfo.php5
Outcome: HTTP 404 on both requests, which indicates the attempt failed
Action: allowed, which still matters because it shows the traffic was permitted and observed in telemetry

Why this matters
Even a failed recon attempt is useful. It supports attribution, it informs block or rate limit decisions, and it is a strong signal for tuning detections and building suppression rules that still preserve true positives.

2. **Trojan category review and signature profiling**
   From the dashboard, I pivoted into a Suricata trojan related alert category and used category and signature context to narrow the event set to the most investigation relevant patterns. This is a practical SOC habit. Reduce noise first, then validate what remains.

3. **CVE validation and severity confirmation from successful HTTP activity**
   I filtered Suricata HTTP activity down to successful responses, using HTTP status 200 as the gate, then reviewed the linked reference material to validate vulnerability context. The advisory cited a CVSS v3 score of 9.8, Critical, which supports prioritization when the traffic indicates success.

4. **Internal host identification from a FortiGate reverse shell alert**
   I pivoted from a FortiGate alert titled MS.Windows.CMD.ReverseShell and used Splunk pivots to resolve the associated internal server name.

Confirmed host
we5201srv.waynecorpinc.local

Why this matters
This is the step that turns an abstract alert into an actionable asset lead. Without the hostname, response actions stall. With the hostname, containment and scoping can begin immediately.

5. **Affected product and CVE extraction from FortiGate reference links**
   For Apache.Roller.OGNL.Injection.Remote.Code.Execution, I used the FortiGate reference link workflow to extract the affected product and the CVE.

Extracted from the linked reference
Affected product: Apache Software Foundation Apache Roller prior to 5.0.2
CVE: CVE-2013-4212

6. **Scanner fingerprinting by correlating FortiGate category and Suricata signature**
   The FortiGate category with the highest volume was Acunetix.Web. I then pivoted into Suricata and confirmed the scanner version in the signature field.

Evidence confirmed
ET SCAN Acunetix Version 6 (Free Edition) Scan Detected

Why this matters
This is clean correlation across tools. One dataset suggests the scanner family, the other provides a precise fingerprint. That supports tuning, blocking, and reporting.

Dashboard engineering and operational reporting
After validating searches, I extended the dashboard with two severity based panels and converted them into pie charts for fast prioritization.

Suricata alert severity distribution
High severity slice showed 28.824 percent when hovered

FortiGate alert severity distribution
Critical severity slice showed 0.401 percent when hovered

Operational takeaway
The environment showed heavy scanning noise with a smaller critical slice. A severity distribution view is useful because it helps analysts quickly focus where response time matters most.

Recommended response actions, production safe guidance
Immediate triage actions
Confirm the reverse shell alert on we5201srv.waynecorpinc.local with host telemetry, then isolate if validated
Identify the internal destination service and confirm whether any successful exploitation occurred

Containment and tuning actions
Block or rate limit repeated phpinfo style recon paths where appropriate
Add detections that alert on scanner fingerprints such as Acunetix Version 6, with thresholds to reduce noise
Track and prioritize successful HTTP responses tied to critical CVSS vulnerabilities

Hunting actions

Hunt for additional activity from 40.80.148.42 and similar sources across the same time window

Hunt for other assets receiving OGNL injection style attempts, then validate patch posture for affected products

Skills demonstrated

SIEM triage and focused hunting in Splunk

Alert validation through raw event evidence extraction

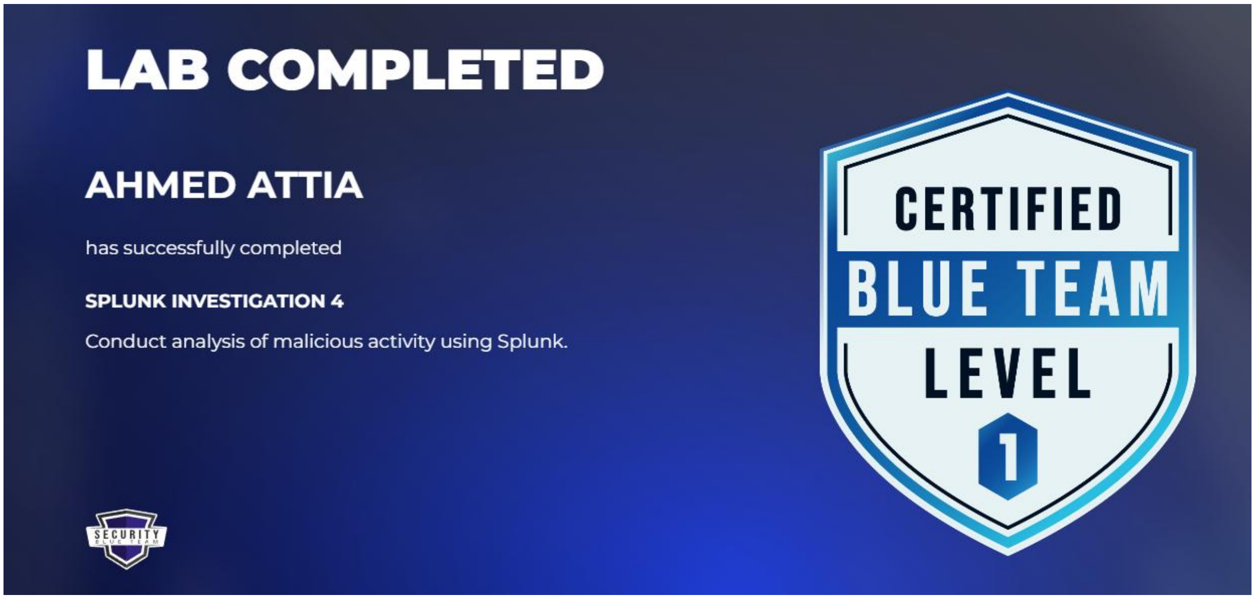
Cross source correlation between Suricata IDS and FortiGate UTM telemetry

Vulnerability context validation using reference links and severity scoring

Dashboard panel creation and visualization for operational reporting

Takeaway

This lab reinforced a practical SOC workflow. Start from a dashboard, validate the alert with raw artifacts, correlate across telemetry sources, then improve the dashboard so the next investigation is faster, more consistent, and easier to operationalize.



LAB COMPLETED

AHMED ATTIA

has successfully completed

SPLUNK INVESTIGATION 4

Conduct analysis of malicious activity using Splunk.

CERTIFIED BLUE TEAM LEVEL 1

SECURITY BLUE TEAM