SPLUNK INCIDENT TRIAGE – WEB

COMMAND INJECTION LAB

INTRODUCTION

This lab simulates a Blue Team investigation of a suspected web-based attack detected through Splunk alerts. Unusual HTTP requests containing Base64-encoded data were observed in Apache logs, along with indications that the web server spawned abnormal processes. The goal of this lab is to triage the incident, identify signs of command injection, confirm host-level execution, and reconstruct the attacker's activity using Apache and Sysmon logs.

The investigation focuses on correlating web-layer activity with operating system telemetry to determine the scope and impact of the attack.

SCENARIO OVERVIEW

A web server exposes a CGI script that improperly handles user input. An attacker sends crafted HTTP requests attempting to execute system commands through this script. Some of the commands are obfuscated using Base64 encoding to evade detection.

Splunk raises alerts indicating unusual behavior related to Apache process execution. As a Blue Team analyst, the task is to investigate whether the attack succeeded, what was executed, and what actions the attacker performed after gaining access.

By:Fábio Vieira

Provided: TryHackMe

INVESTIGATION STEPS

DETECT SUSPICIOUS WEB COMMANDS

The first step is to identify HTTP requests that suggest command execution attempts via the web application.

Splunk Query:

```
index=windows_apache_access (cmd.exe OR powershell OR
"powershell.exe" OR "Invoke-Expression")
| table _time host clientip uri_path uri_query status
```

Analysis:

This query searches Apache access logs for indicators of command injection, such as references to cmd.exe or PowerShell. These keywords strongly indicate an attempt to execute system-level commands through the web server.

Several requests contain long Base64-encoded strings within query parameters, suggesting obfuscated payloads.

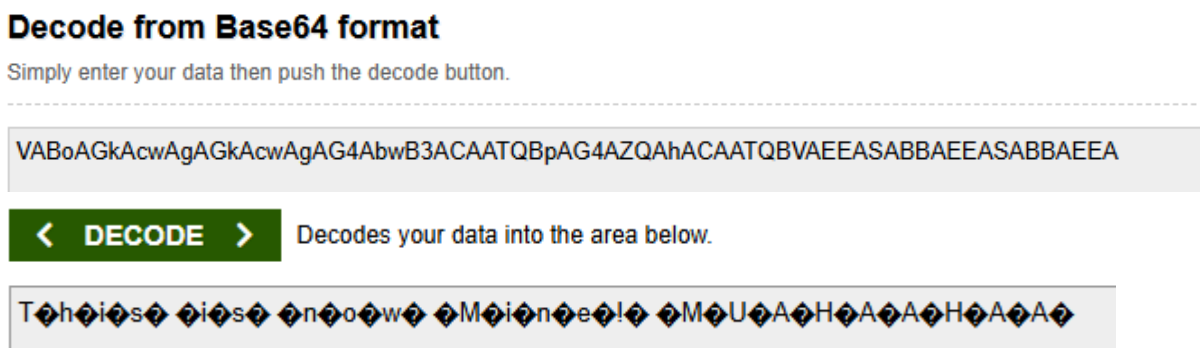
_time	host	clientip	uri_path	uri_query	status
2025-10-27 04:37:36	WebAppServer	10.9.0.217	/cgi-bin/hello.bat	cmd=powershell.exe+-enc+VABoAGkAcwAgAGkAcwAgAG4AbwB3ACAATQBpAG4AZQAhACAATQBVAEEASABBAEEASABBAEEA	200
2025-10-26 21:47:59	WebAppServer	10.9.0.217	/cgi-bin/hello.bat	cmd=cmd.exe	200
2025-10-26 21:48:33	WebAppServer	10.9.0.217	/cgi-bin/hello.bat	cmd=powershell.exe+-enc+VABoAGkAcwAgAGkAcwAgAG4AbwB3ACAATQBpAG4AZQAhACAATQBVAEEASABBAEEASABBAEEA	200
2025-10-27 04:39:10	WebAppServer	10.9.0.217	/cgi-bin/hello.bat	cmd=powershell.exe+-enc+VABoAGkAcwAgAGkAcwAgAG4AbwB3ACAATQBpAG4AZQAhACAATQBVAEEASABBAEEASABBAEEA	200

DECODE SUSPICIOUS BASE64 PAYLOADS

One of the encoded payloads identified in the access logs is:

```
VABoAGkAcwAgAGkAcwAgAG4AbwB3ACAATQBpAG4AZQAhACAATQBVAEEASABBAEEASABBAEEA
```

This string was decoded using a Base64 decoder. The decoded output reveals a simple message, confirming that the attacker was testing encoded PowerShell execution.



This demonstrates an attempt to run obfuscated PowerShell commands via the vulnerable CGI endpoint.

INSPECT APACHE ERROR LOGS FOR EXECUTION FAILURES

Next, Apache error logs are analyzed to determine whether the malicious requests were processed by the backend.

Splunk Query:

```
index=windows_apache_error ("cmd.exe" OR "powershell" OR "Internal Server Error")
```

Analysis:

Requests that result in HTTP 500 “Internal Server Error” responses, such as:

```
/cgi-bin/hello.bat?cmd=powershell
```

indicate that the server attempted to execute the supplied input but failed during processing. This strongly suggests that the attacker’s payload reached the server-side execution logic.

TRACE SUSPICIOUS PROCESS CREATION FROM APACHE

To confirm host-level execution, Sysmon logs are analyzed to identify child processes spawned by Apache.

Splunk Query:

```
index=windows_sysmon ParentImage="*httpd.exe"
```

Analysis:

Under normal circumstances, Apache should not spawn system executables. However, Sysmon logs reveal cases where Apache spawned command-line processes.

Example:

- ParentImage: C:\Apache24\bin\httpd.exe
- Image: C:\Windows\System32\cmd.exe

This is a strong indicator of successful command injection, confirming that the web attack resulted in operating system command execution.

Format | Show: 20 Per Page | View: Table

i		_time	EventCode	_time	Computer	Image	CommandLine	User
>	1	10/26/25 9:48:33.891 PM	1	2025-10-26T21:48:33.891+00:00	WebAppServer	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C ""C:\Apache24\cgi-bin\hello.bat""	WEBAPPSERVER\apache_svc
>	2	10/26/25 9:47:59.274 PM	1	2025-10-26T21:47:59.274+00:00	WebAppServer	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C ""C:\Apache24\cgi-bin\hello.bat""	WEBAPPSERVER\apache_svc
>	3	10/26/25 9:47:35.849 PM	1	2025-10-26T21:47:35.849+00:00	WebAppServer	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C ""C:\Apache24\cgi-bin\hello.bat""	WEBAPPSERVER\apache_svc
>	4	10/26/25 9:45:09.600 PM	1	2025-10-26T21:45:09.600+00:00	WebAppServer	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /C ""C:\Apache24\cgi-bin\hello.bat""	WEBAPPSERVER\apache_svc

CONFIRM ATTACKER RECONNAISSANCE ACTIVITY

After achieving command execution, attackers typically perform basic reconnaissance to understand their execution context.

Splunk Query:

```
index=windows_sysmon *cmd.exe* *whoami*
```

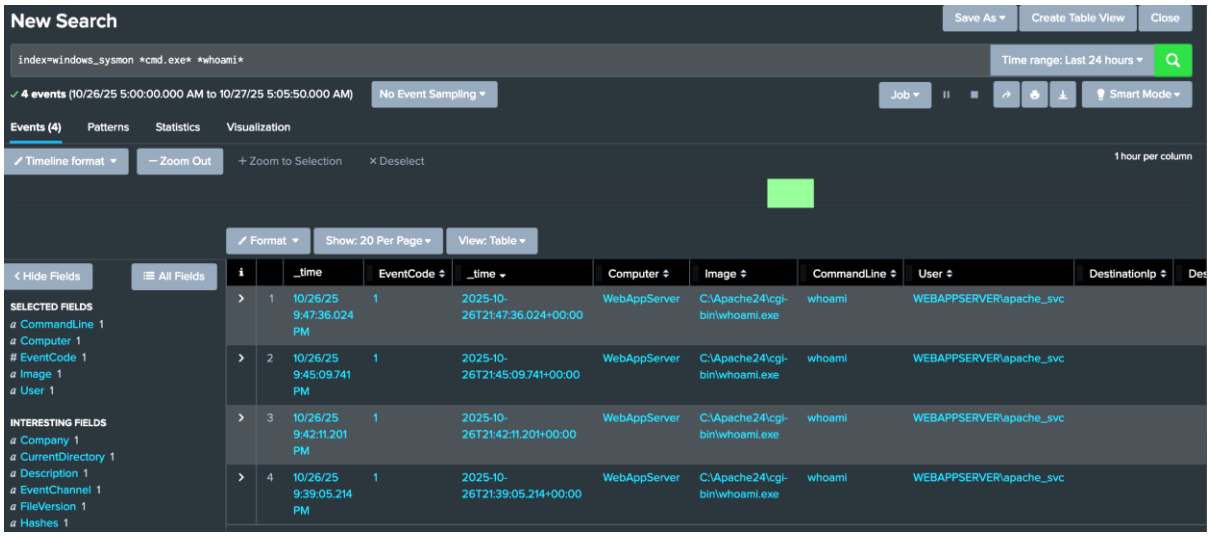
Analysis:

The execution of the `whoami` command confirms post-exploitation activity. This command is commonly used to identify the current user and privilege level.

The presence of this command confirms that the attacker successfully executed commands on the host.

Reconnaissance executable identified:

- Whoami.exe



IDENTIFY BASE64-ENCODED POWERSHELL PAYLOAD EXECUTION

Finally, Sysmon logs are searched for evidence of encoded PowerShell command execution.

Splunk Query:

```
index=windows_sysmon Image="*powershell.exe"
(CommandLine="*enc*" OR CommandLine="*-EncodedCommand*" OR
CommandLine="*Base64*")
```

Analysis:

No successful execution of encoded PowerShell commands is observed. This indicates that while the attacker attempted to execute obfuscated PowerShell payloads, they did not successfully run.

This suggests that defensive controls or execution errors prevented the final payload from executing.

FINDINGS SUMMARY

- A command injection attack was attempted through a vulnerable CGI script.
- The attacker attempted to execute PowerShell via the web server.
- Apache successfully spawned cmd.exe, confirming OS-level execution.
- The attacker executed whoami.exe for reconnaissance.
- Encoded PowerShell payloads were identified but did not execute successfully.

CONCLUSION

This lab demonstrates a complete Blue Team triage workflow using Splunk to investigate a web-based command injection attack. By correlating Apache access logs, Apache error logs, and Sysmon telemetry, it is possible to confirm exploitation, identify attacker actions, and assess the overall impact.

The investigation confirms that the attacker achieved limited command execution on the host but failed to successfully run obfuscated PowerShell payloads. This highlights the importance of strong input validation, monitoring parent-child process relationships, and detecting encoded command execution to defend against web-to-host attack chains.