Author: Jack Fitzgerald

Official Incident Report

Date: Nov, 21, 2023, 12:24 PM

Event ID: 201

Rule Name: SOC239 - Remote Code Execution Detected in Splunk Enterprise

Table of Contents

Alert Details 3
Detection 4
Verify 4
Analysis 5
Containment 9
Summary 10
Lessons Learned 11
Remediation Actions 11
Appendix 12
MITRE ATT&CK 12
Artifacts 12
LetsDefend Playbook 12

Alert Details

Severity: High

Type: Unauthorized Access

Source IP Address: 180.101.88.240

Destination IP Address: 172.16.20.13

Hostname: Splunk Enterprise

HTTP Request Method: POST

Requested URL: http://18.219.80.54:8000/en-US/splunkd/__upload/indexing/preview?output_mode=json&props.NO_BINARY_CHECK=1&input.path=shell.xsl

Trigger File Path: /opt/splunk/var/run/splunk/dispatch/1700556926.3/shell.xsl

Alert Trigger Reason: Detected a malicious XSLT upload in Splunk Enterprise with the potential to trigger remote code execution.

Device Action: Allowed

Based on the information provided in the alert, it appears that an XSLT file has been uploaded to the Splunk Enterprise with the potential to trigger RCE. The alert is triggered by rule SOC239 - Remote Code Execution Detected in Splunk Enterprise.

Overall, it appears that the alert may be **suspicious**, and further investigation is needed to identify the extent of the alert and determine if any necessary actions are required to remediate the situation.

Detection

Verify

As a security analyst, one of the first steps we take to verify the alert and determine whether it is a **false positive** or a **true positive** incident is to investigate the Splunk Enterprise and XSLT file formats.

Splunk Enterprise is a tool that allows you to search, analyze, and visualize your data with powerful, easy-to-understand dashboards.

XSLT (eXtensible Stylesheet Language Transformations) is a powerful **XML-based** language used to transform one XML document into another XML document or a different format like HTML, text, or PDF.

According to iNE, injection issues happen when the user input is blindly trusted without thinking of the consequences. If the right conditions are set, then this can result in data exfiltration, RCE, XSS, and much more.

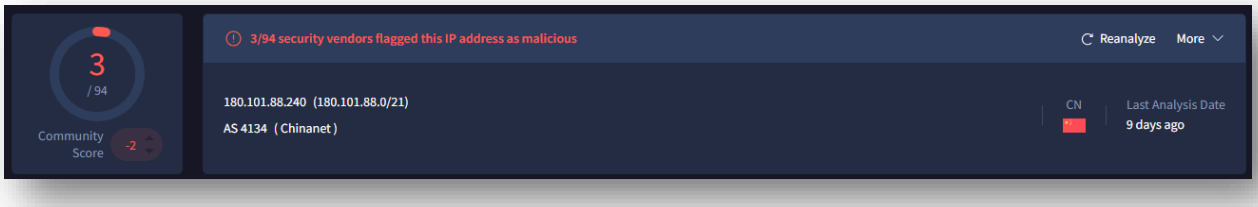
The file to investigate for this alert is 'shell.xsl', via POST to 'http://18.219.80.54:8000/en-US/splunkd/__upload/indexing/preview?output_mode=json&props.NO_BINARY_CHECK=1&input.path=shell.xsl', with the trigger file path '/opt/splunk/var/run/splunk/dispatch/1700556926.3/shell.xsl'.

The IP Source Address that this file was uploaded from is 180.101.88.240.

Analysis

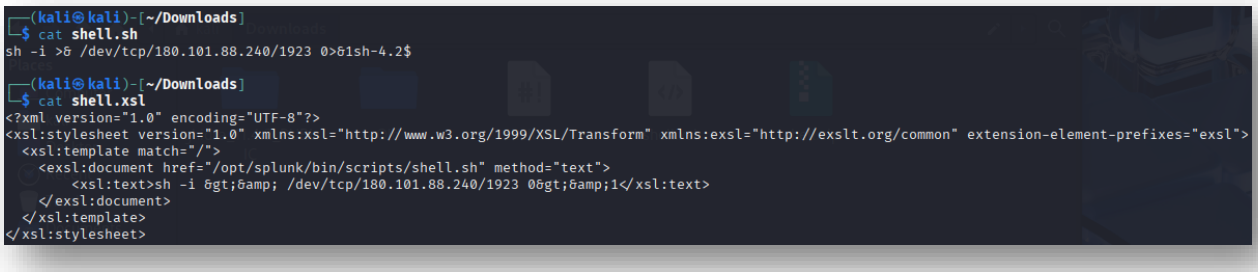
Now that we have detected the alert and its details, we can start by analyzing the IP found in the alert.

https://www.virustotal.com/gui/ip-address/180.101.88.240



https://www.abuseipdb.com/check/180.101.88.240

During investigation of the SOC alert for **potential XSLT-triggered remote code execution**, two artefacts were identified: shell.xsl and shell.sh. Together, these files confirm successful exploitation rather than a benign transformation or failed attempt.



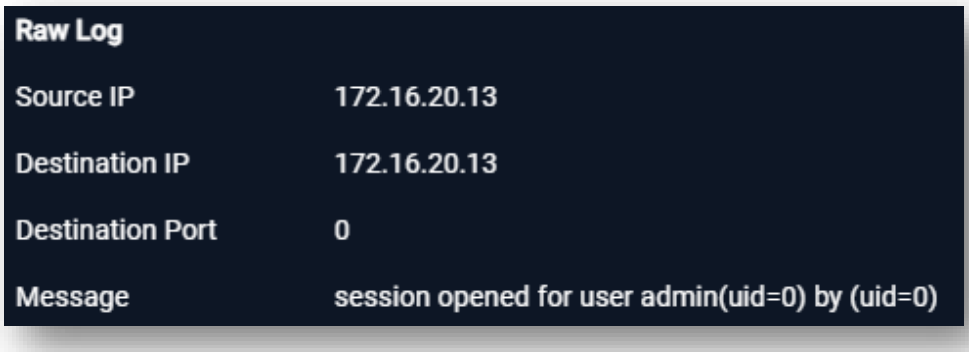
The file shell.xsl contains a malicious XSLT stylesheet that abuses the extension to **write arbitrary files to disk**. Specifically, the stylesheet writes a shell script to /opt/splunk/bin/scripts/shell.sh. This behaviour exceeds normal XSLT processing and indicates that the XSLT engine was running with extensions enabled and sufficient filesystem permissions, enabling arbitrary file writing as part of the transformation process.

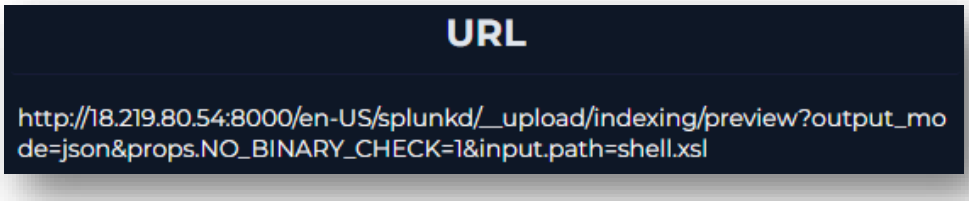
The generated file, shell.sh, contains a **Bash reverse shell** command that initiates an outbound TCP connection to 180.101.88.240 on port 1923 and provides the remote host with an interactive shell on the affected system. The contents of this script align with common post-exploitation techniques used to establish command-and-control access following successful remote code execution.

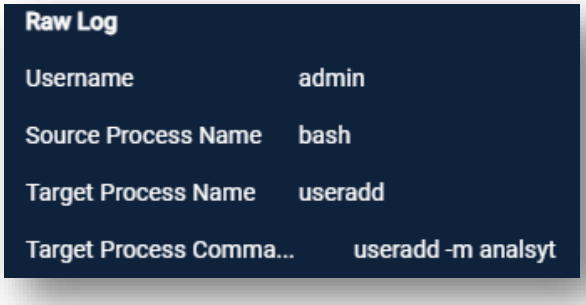
The presence and contents of both files demonstrate a clear exploitation chain: malicious XSLT input was processed by a vulnerable component, resulting in the creation of a reverse shell payload, which was subsequently executed.

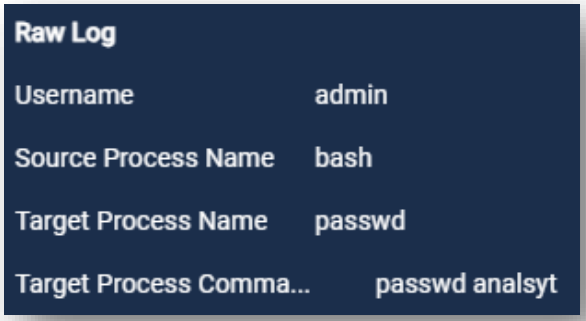
Next, we will investigate Log Management to find any further details of the alert. There were 4 events (before Nov, 21, 2023, 12:24 PM UTC) from Destination Address 172.16.20.13.

The logs show a session was opened for the admin user, followed by the shell.xsl file upload, followed by a new user being added to the system via bash with username: analsyt | password: analsyt.



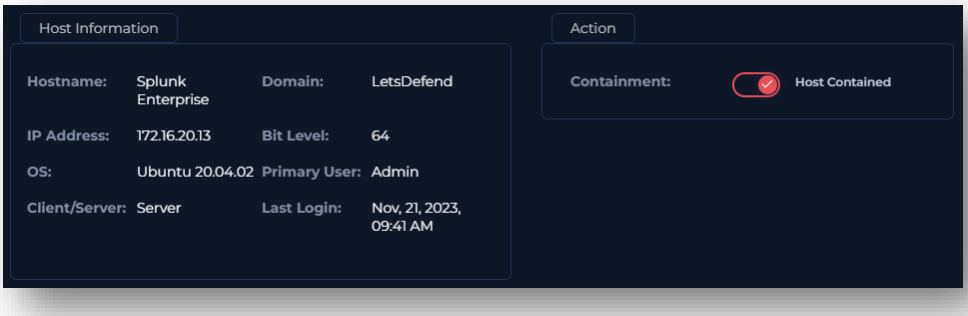





Raw Log
Username
admin
Source Process Name
bash
Target Process Name
passwd
Target Process Comma...
passwd analsyt

Containment

Based on the information gathered during the investigation, it is highly likely that the system has been compromised. Immediate isolation of the system from the network is required.



Summary

The incident involves a compromise system named **Splunk Enterprise** with an IP address of **172.16.20.13**. The alert was triggered by an XSLT file that had been uploaded to the Splunk Enterprise with the potential to trigger RCE, based on the rule SOC239 - Remote Code Execution Detected in Splunk Enterprise.

An investigation into the alert for potential XSLT-triggered remote code execution confirmed a successful compromise of the affected system. A malicious XSLT file (shell.xsl) was used to write a reverse shell script (shell.sh) to disk, which enabled outbound command-and-control access to an external IP address.

Log analysis shows the attacker authenticated as the admin user, uploaded the malicious XSLT payload, and subsequently executed post-exploitation activity, including the creation of a new local user account. These findings indicate unauthorized access, successful code execution, and active attacker control of the system.

Based on the findings of the incident, no immediate action needs to be taken to isolate the compromised system, and the event was identified as a **True Positive**.

Lessons Learned

- The core vulnerability stemmed from Splunk Enterprise not safely sanitizing user-provided Extensible Stylesheet Language Transformations (XSLT)
- The exploit often required valid credentials, with default credentials like admin usernames and passwords, making systems particularly vulnerable
- This incident is a reminder that even widely used, powerful software can have critical flaws, emphasizing the need for continuous vulnerability scanning and management programs to identify and address security flaws before they are exploited

Remediation Actions

- Treat Splunk administrator accounts with the highest security, like root accounts
- Change all default usernames and passwords immediately after installation and set a minimum password length
- Use non-administrator accounts for routine tasks like searching and reporting
- Implement detection analytics to identify potential RCE attempts via user-supplied XSLT by monitoring splunkd_ui logs for specific URI patterns and status codes indicative of injection attempts
- Use rigorous input validation and sanitization in all code, especially when handling data transformation capabilities

Appendix

MITRE ATT&CK

MITRE Tactics	MITRE Techniques
Initial Access	T1190 - Exploit Public-Facing Application
Initial Access	T1078.001 - Valid Accounts: Default Accounts
Execution	T1059.004 - Command and Scripting Interpreter: Unix Shell
Persistence	T1136.001 - Create Account: Local Account
Defense Evasion	T1220 - XSL Script Processing
Discovery	T1087.001 - Account Discovery: Local Account
Command and Control	T1071.001 - Application Layer Protocol: Web Protocols

Artifacts

Value	Comment	Type
http://18.219.80.54:8000/en-US/splunkd/__upload/indexing/preview?output_mode=json&props.NO_BINARY_CHECK=1&input.path=shell.xsl	Requested URL	URL Address
180.101.88.240	Attacker/C2 Server	IP Address

LetsDefend Playbook

LetsDefend Event ID: 201