Official Incident Report

Event ID: 189

Rule Name: SOC227 - Microsoft SharePoint Server Elevation of Privilege - Possible CVE-2023-29357 Exploitation

Table of contents

Alert

Based on the information that the alert provided, it appears that there are some suspicious network traffic detected on a Windows server named "**MS-SharePointServer**" with an IP address of **172.16.17.233**. The Alert is triggered by the **SOC227** rule for **Microsoft SharePoint Server Elevation of Privilege - Possible CVE-2023-29357 Exploitation**.

> *CVE-2023-29357, the security flaw can let unauthenticated attackers gain administrator privileges following successful exploitation in low-complexity attacks that don't require user interaction.*

https://www.bleepingcomputer.com/news/security/exploit-released-for-microsoft-sharepoint-server-auth-bypass-flaw/

The device action is marked as "allowed", indicating that no action was taken by the device to prevent or block the execution of the file.

SEVERITY	DATE	RULE NAME	EVENTID	TYPE	ACTION
High	2023-10-06 8:05	SOC227 - Microsoft SharePoint Server Elevation of Privilege - Possible CVE-2023-29357 Exploitation	189	Web Attack	Allowed

The CVE-2023-29357 vulnerability is a critical privilege escalation vulnerability that, when combined with other vulnerabilities, could lead to remote code execution. A CVSS score of 9.8 (Critical) an

EventID :	189
Event Time :	2023-10-06 20:05
Rule :	SOC227 - Microsoft SharePoint Server Elevation of Privilege - Possible CVE-2023-29357 Exploitation
Level :	Security Analyst
Hostname :	MS-SharePointServer
Destination IP Address :	172.16.17.233
Source IP Address :	39.91.166.222
HTTP Request Method :	GET
Requested URL :	/_api/web/siteusers
User-Agent :	python-requests/2.28.1
Alert Trigger Reason :	This activity may be indicative of an attempt to exploit the CVE-2023-29357 vulnerability, which could potentially lead to unauthorized access and privilege escalation within the SharePoint server.
Device Action :	Allowed

Based on the provided trigger reason, the potential exploitation activity for CVE-2023-29357 is detected on the MS-SharePointServer.

There is also an attachment named SP-IIS.zip which we can download and analyse.

Overall, it appears that there may be malicious network activity occurring on the system, and further investigation is needed to identify the extent of the activity and determine any necessary actions to remediate the situation.

Detection

Verify

As the playbook suggests we can start investigating the alert by understanding why the alert was triggered

Examine the rule name. Rule names are usually created specifically for the attack to be detected. By examining the rule name, you can understand which attack you are facing.

> The rule name mentioned in the alert is "SOC227 rule for Microsoft SharePoint Server Elevation of Privilege - Possible CVE-2023-29357 Exploitation." It suggests that the alert is related to the detection of a potential attempt to exploit the CVE-2023-29357 vulnerability within a Microsoft SharePoint Server, with a focus on the elevation of privilege. This rule name is specific and indicates that the alert is related to a security threat associated with the SharePoint Server.

Detect between which two devices the traffic is occurring. It's a good starting point to understand the situation by learning about the direction of traffic, what protocol is used between devices, etc.

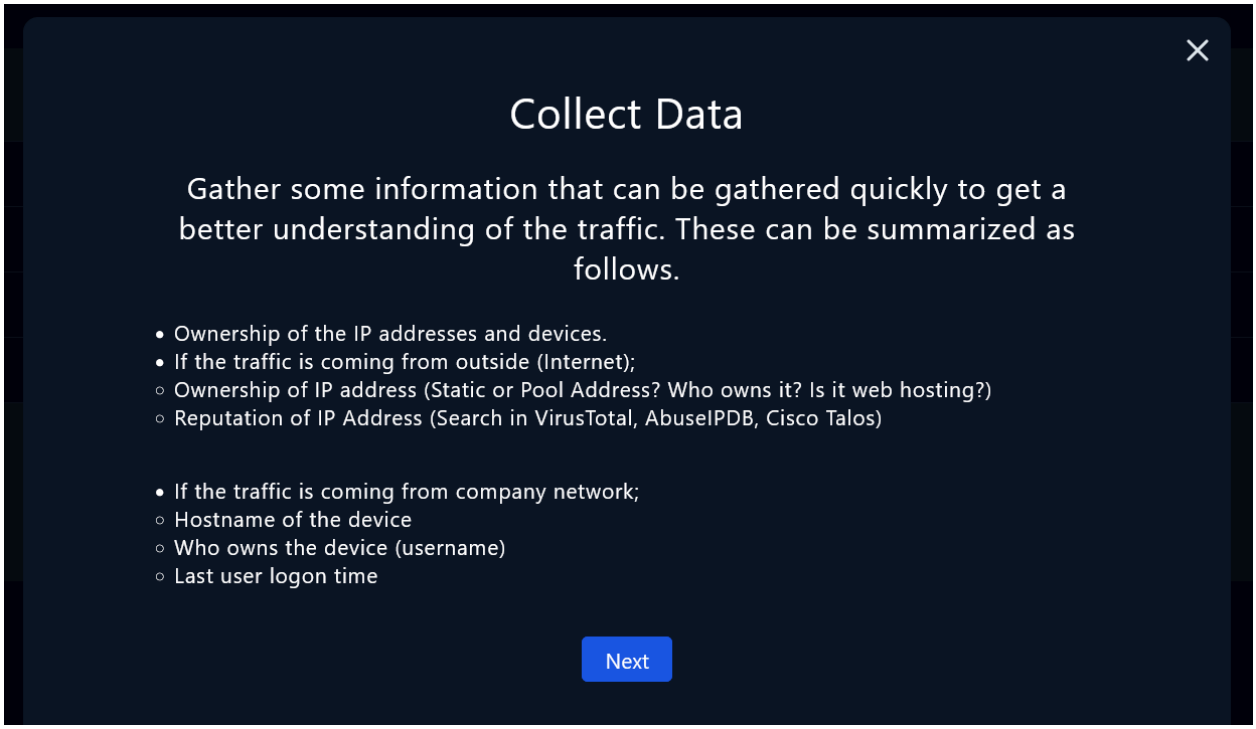
The alert provides information about the source and destination IP addresses involved in the suspicious network traffic:

- Source IP Address: 39.91.166[.]222
- Destination IP Address (Hostname): 172.16.17.233 (MS-SharePointServer)

In this case, traffic is occurring between the source IP (39.91.166[.]222) and the destination IP (172.16.17.233), which corresponds to the Windows server named "MS-SharePointServer." This establishes the direction of the network traffic and the devices involved, with the source likely being the potential attacker, and the destination being the server that is being targeted for the suspicious activity.

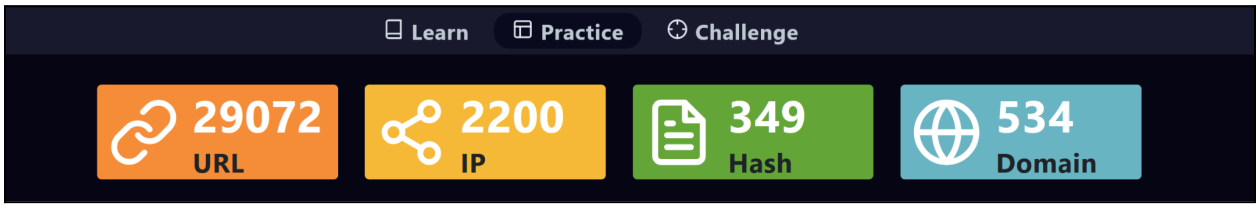
Collect Data

The next step in the playbook leads us to collect data and gather information about the relevant IP address.



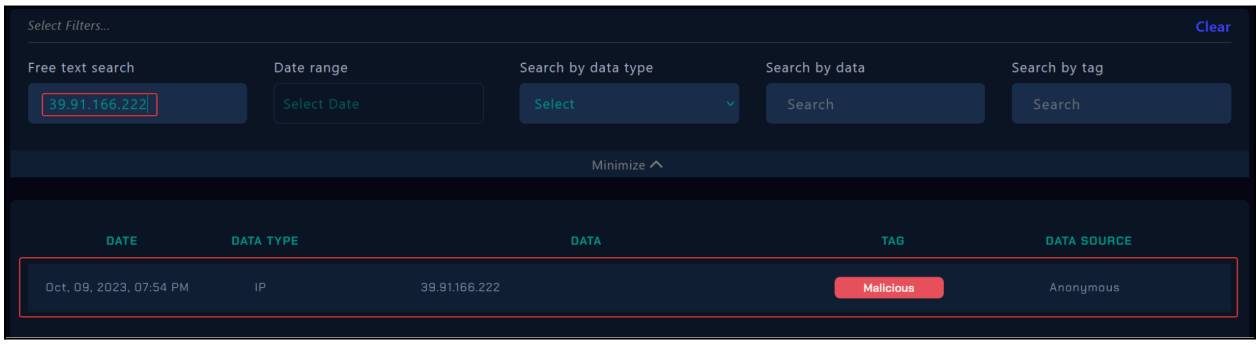
Examining whether the IP address or domain has been linked to prior malicious activities and ownership of the IP address can provide insights into the current activity.

On the LetsDefend threat intel tab, you'll find a comprehensive database dedicated to cataloging maliciously used information, such as IP addresses, domains, and other indicators of compromise.

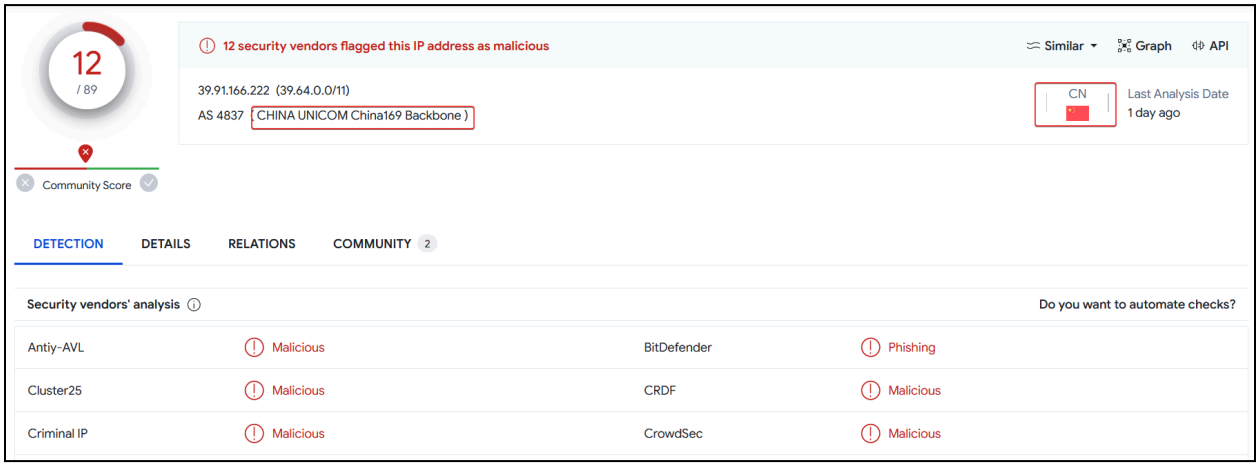


https://app.letsdefend.io/threath-intelligence-feed

Upon cross-referencing the destination IP address discovered in the log management system with the Threat Intel tab, it was determined that the address has been categorized as both Command and Control (C2) and malicious in nature



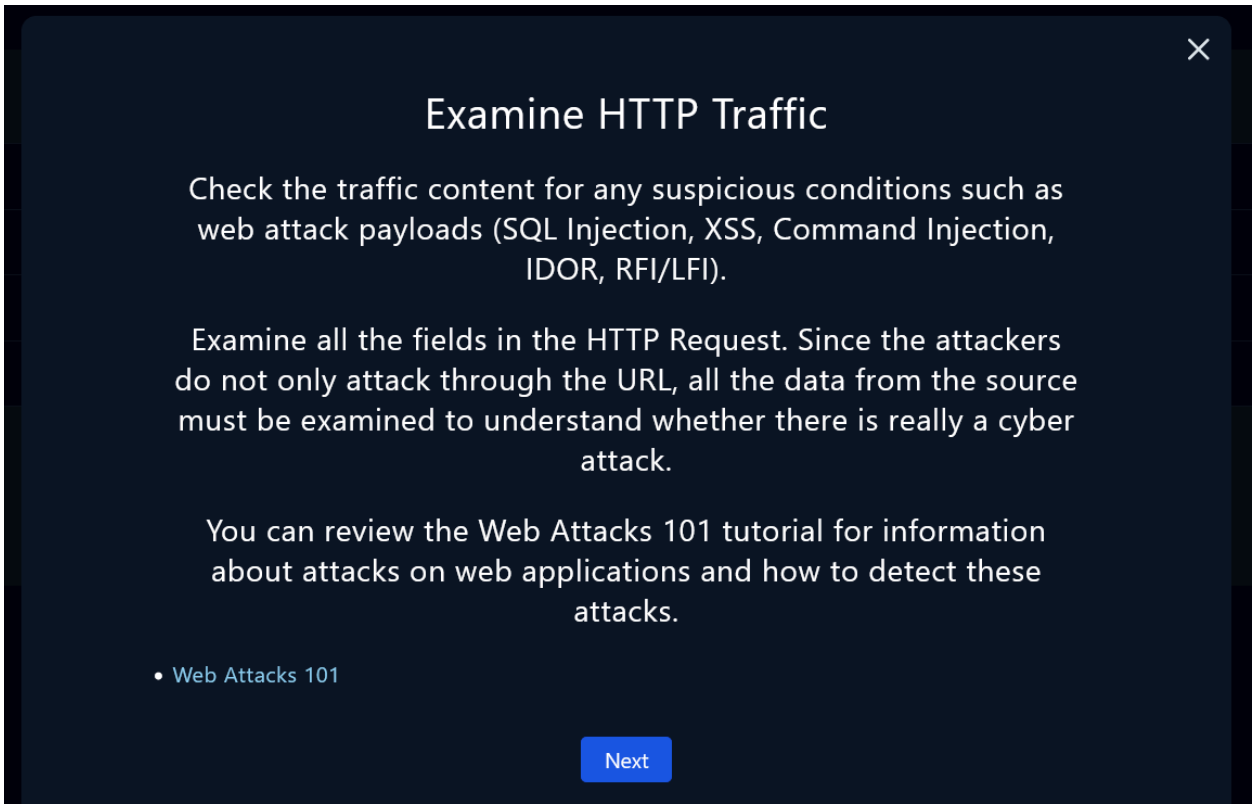
By cross-referencing the IP address with threat intelligence platforms such as Abuseip or Virustotal, we discovered that the IP address is malicious and reported many times.



Based on the information provided by VirusTotal, it appears that the IP address has been flagged as malicious by **12** antivirus engines. Additionally, the geolocation of the IP address is China.

Examine The Traffic

The third step of the playbook involves examining the HTTP traffic.



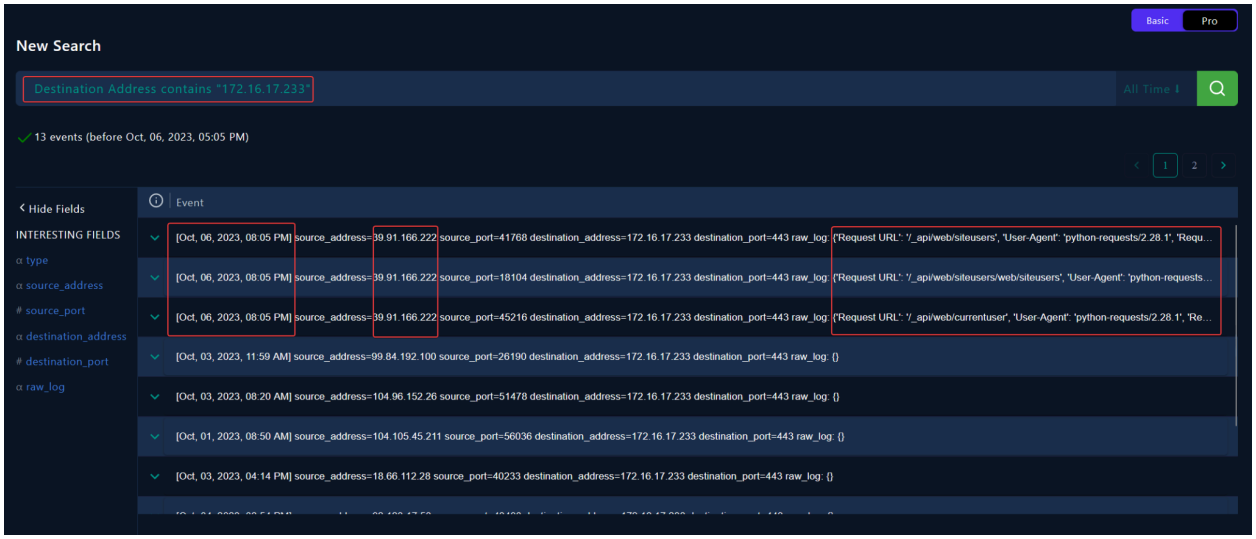
The SP-IIS.log file provided in the alert details contains the IIS logs. We can start the traffic analysis from here initially.

```
Version/6.0 Mobile/10A5376e Safari/8536.25 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
80.237.234.150 - - [06/Oct/2023:20:05:04 +0000] "GET /favicon.ico HTTP/1.1" 200 3638 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20131215 Firefox/24.0 Iceweasel/24.2.0"
208.91.156.11 - - [06/Oct/2023:20:05:46 +0000] "GET /files/logstash/logstash-1.3.2-monolithic.jar HTTP/1.1" 404 324 "-" "Chef Client/10.18.2 (ruby-1.9.3-p327; ohai-6.16.0; x86_64-linux;
+http://opscode.com)"
95.214.53.99 - - [06/Oct/2023:20:05:06 +0000] "GET /_api/web/siteusers HTTP/1.1" 200 1453 "-" "python-requests/2.28.1"
95.214.53.99 - - [06/Oct/2023:20:05:06 +0000] "GET /_api/web/siteusers/web/siteusers HTTP/1.1" 404 1453 "-" "python-requests/2.28.1"
95.214.53.99 - - [06/Oct/2023:20:05:06 +0000] "GET /_api/web/currentuser HTTP/1.1" 200 1071 "-" "python-requests/2.28.1"
202.101.244.118 - - [06/Oct/2023:20:05:22 +0000] "GET / HTTP/1.0" 200 37932 "http://www.letsdefend.io/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:21.0) Gecko/20100101 Firefox/21.0"
202.101.244.118 - - [06/Oct/2023:20:05:27 +0000] "GET /blog/geekery/installing-windows-8-consumer-preview.html HTTP/1.0" 200 8948 "http://www.letsdefend.io/" "Mozilla/5.0 (Macintosh;
Intel Mac OS X 10.7; rv:21.0) Gecko/20100101 Firefox/21.0"
50.103.242.61 - - [06/Oct/2023:20:05:07 +0000] "GET /articles/dynamic-dns-with-dhcp/ HTTP/1.1" 200 18848 "https://www.google.com/" "Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36"
```

Since this attack contains a 0-day that targets Sharepoint we can get the URL that triggered the alarm as a reference to analyze.

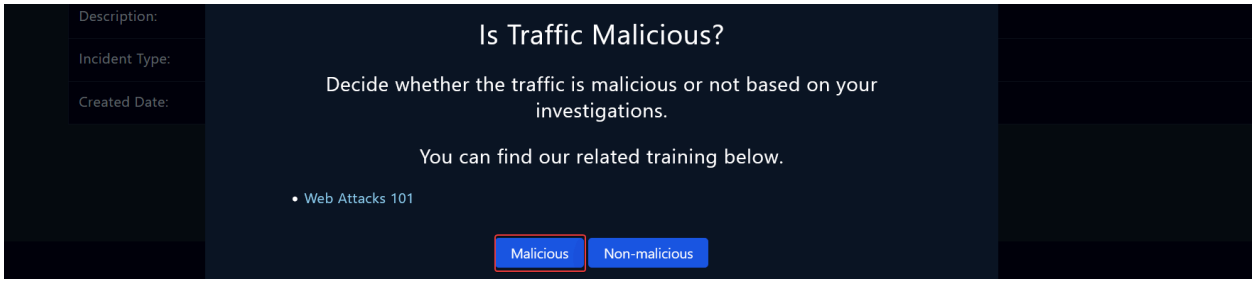
This log entry indicates that three “GET” requests were made to different URIs under the path “/_api/web/siteusers”. As seen in the user-agent part the requests were made by a Python script, using the requests library. Two of the three requests appear to have resulted in a “200” OK success status response code **indicating that the requests have succeeded**.

For the extended analysis we can analyze network traffic on the log management page. By filtering the IP address of the MS-SharePointServer as the destination address we can access the related logs.



There are permitted and HTTP 200 response-coded malicious network traffic events on the log management system.

Based on our analysis, we have confirmed that the traffic is **malicious and permitted.**

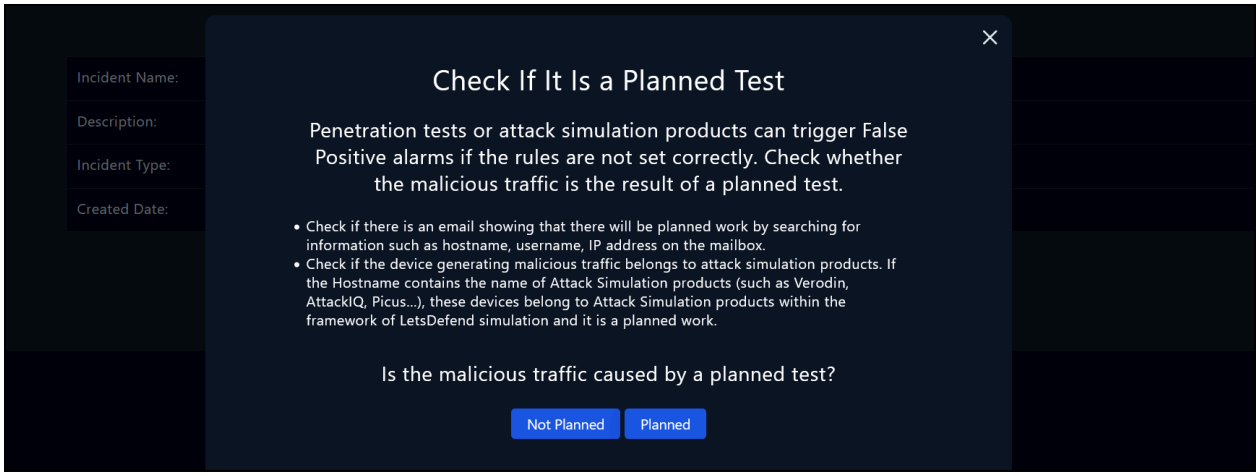


Analysis

The analysis confirms that the relevant attack type is a web attack for Microsoft Sharepoint. The answer for the attack type is Other.

```
Version/6.0 Mobile/10A5376e Safari/8536.25 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
80.237.234.150 - - [06/Oct/2023:20:05:04 +0000] "GET /favicon.ico HTTP/1.1" 200 3638 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20131215 Firefox/24.0 Iceweasel/24.2.0"
208.91.156.11 - - [06/Oct/2023:20:05:46 +0000] "GET /files/logstash/logstash-1.3.2-monolithic.jar HTTP/1.1" 404 324 "-" "Chef Client/10.18.2 (ruby-1.9.3-p327; ohai-6.16.0; x86_64-linux;
+http://opscode.com)"
95.214.53.99 - - [06/Oct/2023:20:05:06 +0000] "GET /_api/web/siteusers HTTP/1.1" 200 1453 "-" "python-requests/2.28.1"
95.214.53.99 - - [06/Oct/2023:20:05:06 +0000] "GET /_api/web/siteusers/web/siteusers HTTP/1.1" 404 1453 "-" "python-requests/2.28.1"
95.214.53.99 - - [06/Oct/2023:20:05:06 +0000] "GET /_api/web/currentuser HTTP/1.1" 200 1071 "-" "python-requests/2.28.1"
202.101.244.118 - - [06/Oct/2023:20:05:22 +0000] "GET / HTTP/1.0" 200 37932 "http://www.letsdefend.io/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:21.0) Gecko/20100101 Firefox/21.0"
202.101.244.118 - - [06/Oct/2023:20:05:27 +0000] "GET /blog/geekery/installing-windows-8-consumer-preview.html HTTP/1.0" 200 8948 "http://www.letsdefend.io/" "Mozilla/5.0 (Macintosh;
Intel Mac OS X 10.7; rv:21.0) Gecko/20100101 Firefox/21.0"
50.103.242.61 - - [06/Oct/2023:20:05:07 +0000] "GET /articles/dynamic-dns-with-dhcp/ HTTP/1.1" 200 18848 "https://www.google.com/" "Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36"
```

The IP and hostname information of the relevant hostname were searched within the emails received during the specified dates. However, no evidence related to a planned activity has been observed through this investigation.



Threat Hunting

Next step of the playbook involves examining the direction of the traffic.

In this section, we will engage in a sort of **threat-hunting** exercise. We already have some IOCs about the attack.

```
95.214.53.99 - - [06/Oct/2023:20:05:06 +0000] "GET /_api/web/siteusers HTTP/1.1" 200 1453 "-" "python-requests/2.28.1"
95.214.53.99 - - [06/Oct/2023:20:05:06 +0000] "GET /_api/web/siteusers/web/siteusers HTTP/1.1" 404 1453 "-" "python-requests/2.28.1"
95.214.53.99 - - [06/Oct/2023:20:05:06 +0000] "GET /_api/web/currentuser HTTP/1.1" 200 1071 "-" "python-requests/2.28.1"
```

From the provided log entries, here are some potential Indicators of Compromise (IOCs):

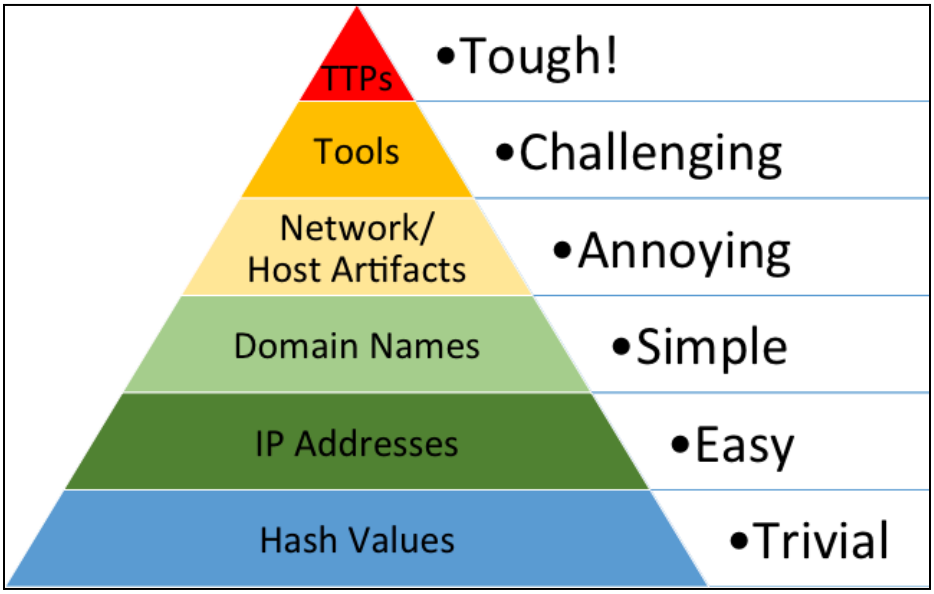
1. IP Address:
   - 39.91.166[.]222

2. URLs:
   - /_api/web/siteusers
   - /_api/web/siteusers/web/siteusers
   - /_api/web/currentuser

3. User-Agent:
   - "python-requests/2.28.1"

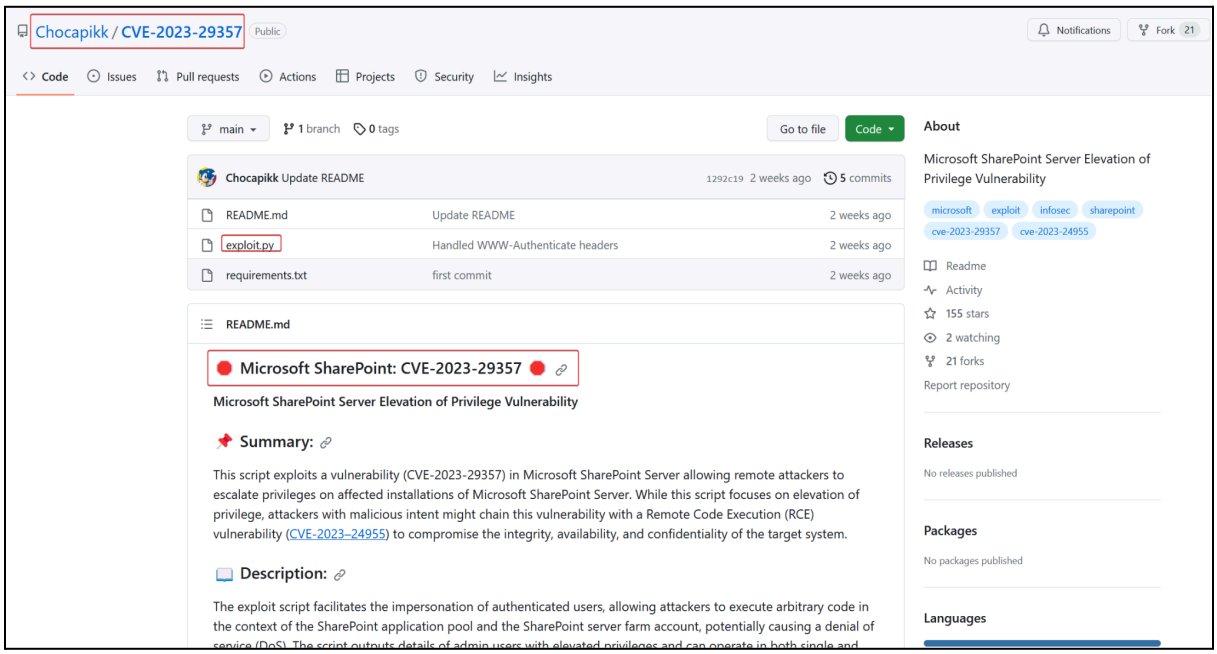
These IOCs can be useful for monitoring and investigating potential security threats and incidents. It's important to analyze the logs further to understand the context and determine if any suspicious activities or security incidents are associated with these IOCs.

As observed in the "Pyramid of Pain," while detecting IOCs at the base of the pyramid may be relatively straightforward, for the attacker, changing these indicators is equally effortless.



http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

Therefore, to enhance our threat-hunting efforts, we should descend from the top of the pyramid to the foundation of the attack, understanding the technical tactics and procedures. This approach may be more challenging but is ultimately more effective in establishing a detection mechanism.



https://github.com/Chocapikk/CVE-2023-29357

The script's specific usage for replicating the vulnerability has been observed in real-world incidents. Scripts with educational purposes can be found on GitHub as well. By analyzing exploit.py we can gather more information about exploit.

CVE-2023-29357 / exploit.py

Code Blame 264 lines (211 loc) · 10.9 KB

```
92                  "Authorization": f"Bearer {jwt_token}",
93                  "X-PROOF_TOKEN": jwt_token,
94              }
95
96              endpoint_url = self.url.strip() + '/_api/web/currentuser'
97              response = requests.get(endpoint_url, headers=headers, verify=False, timeout=5)
98              if response.status_code == 200:
99                  try:
100                     parsed_response = json.loads(response.text)
101                     console.print(f"[+] Spoofing succeeded for {user.get('Title', 'Unknown User')}: {user.get('Email', 'N/A')} at '/_api/web/currentuser'", style="bold green")
102                     console.print(json.dumps(parsed_response, indent=4), style="bold green")
103                 except json.JSONDecodeError:
104                     console.print(f"[+] Spoofing succeeded for {user.get('Title', 'Unknown User')}: {user.get('Email', 'N/A')} at '/_api/web/currentuser'", style="bold green")
105                     console.print(f"Received non-JSON response:\n{response.text}", style="bold yellow")
106             else:
107                 console.print(f"[-] Spoofing failed for {user.get('Title', 'Unknown User')}: {user.get('Email', 'N/A')} at '/_api/web/currentuser'. Status code: {response.status_code}", style="bold red")
108
109     def create_jwt_token(self) -> str:
134         return jwt_token
135
136
137     def authenticate_with_token(self, token: str) -> Union[bool, List[Dict[str, str]]]:
138         headers = {
139             "Accept": "application/json",
140             "Authorization": f"Bearer {token}",
141             "X-PROOF_TOKEN": token,
142         }
143
144         response = requests.get(self.url + '/_api/web/siteusers', headers=headers, verify=False, timeout=5)
145
146         if self.verbose:
147             console.print("[!] Attempting authentication for", self.url, "with token", style="bold yellow")
148
149         if response.status_code == 200:
150             try:
151                 parsed_response = json.loads(response.text)
152                 users = parsed_response.get('value', [])
153                 admin_users = [user for user in users if user.get('IsSiteAdmin', False) is True]
154                 admin_info_list = []
```

The code is a Python script that attempts to spoof admin users by generating JWT tokens, making HTTP requests with those tokens, and then displaying the results in an output. Here is a Yara rule written for the detection of the successful exploitation of CVE-2023-29357 on Microsoft SharePoint servers with the published Python POC.

signature-base / yara / expl_sharepoint_cve_2023_29357.yar

Neo23x0 Update expl_sharepoint_cve_2023_29357.yar

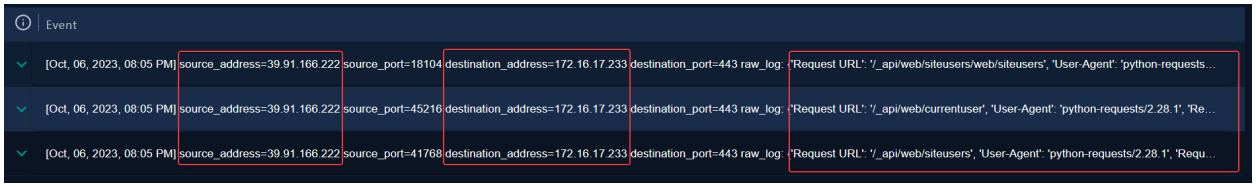
Code Blame 61 lines (55 loc) · 2.02 KB

```
1
2   rule LOG_EXPL_SharePoint_CVE_2023_29357_Sep23_1 {
3      meta:
4         description = "Detects log entries that could indicate a successful exploitation of CVE-2023-29357 on Microsoft SharePoint servers with the published Python POC"
5         author = "Florian Roth (with help from @LuemmelSec)"
6         reference = "https://twitter.com/Gi7w0rm/status/1706764212704591953?s=20"
7         date = "2023-09-28"
8         modified = "2023-10-01"
9         score = 70
10     strings:
11        /*
12           references:
13           https://x.com/TH3C0DEX/status/1707503935596925048?s=20
14           https://x.com/theluemmel/status/1707653715627311360?s=20 (plus private chat)
15        */
16        $xr1 = /GET [a-z\.\/_]{0,40}\/web\/(siteusers|currentuser) - (80|443) .{10,200} (python-requests\/[0-9\.]{3,8}|-) [^ ]{1,160} [^4]0[0-9] /
17     condition:
18        $xr1
19  }
```

https://github.com/Neo23x0/signature-base/blob/master/yara/expl_sharepoint_cve_2023_29357.yar

As we saw from the exploit and yara rule, exploitation web request for the SharePoint servers contains “**siteusers**” or “**currentuser**” which starts with “**/_api/web/**” we can build our custom query from this to **hunt** on **log management** screen.

To determine the direction of traffic, we will review the all logs we gathered from our security products on the log management page.

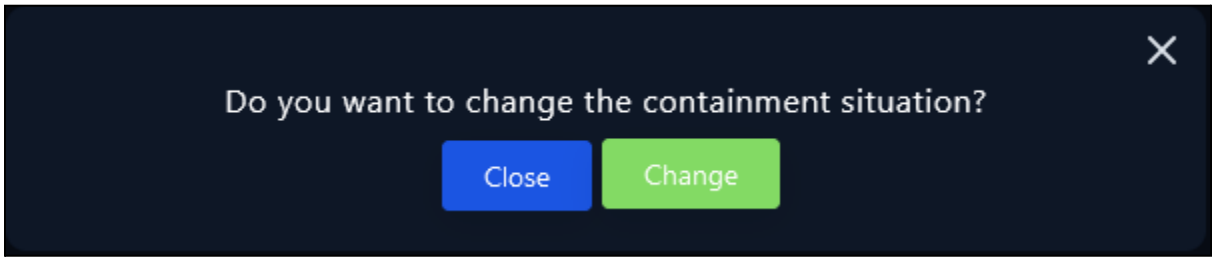


In the log management page, all of the malicious traffic is from Internet -> Company Network.

So the answer for this playbook step is Internet -> Company Network

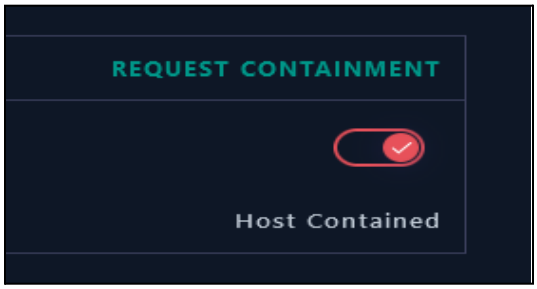
Containment

Based on the information gathered during the investigation, it is highly likely that the system has been compromised. To prevent further data loss or unauthorized access, it is recommended to isolate the system from the network immediately.



Isolation of the host can be made from the endpoint security tab.

Hostname	MS-SharePointServer
IP Address	172.16.17.233



After the containment we can close the alert from the investigation channel.

Summary

The alert report details the detection of suspicious network traffic on a Windows server named "MS-SharePointServer" with the IP address 172.16.17.233. The alert was triggered by the SOC219 rule for Microsoft SharePoint Server Elevation of Privilege, specifically targeting the possible exploitation of CVE-2023-29357. This vulnerability allows unauthenticated attackers to gain administrator privileges without user interaction.

The device action for this alert was marked as "allowed," indicating that no action was taken by the device to prevent or block the execution of the file.

Key Findings from the Investigation:

1. Rule Name: The rule name, "SOC227 rule for Microsoft SharePoint Server Elevation of Privilege - Possible CVE-2023-29357 Exploitation," suggests that the alert is related to the detection of a potential attempt to exploit the CVE-2023-29357 vulnerability within a Microsoft SharePoint Server, focusing on the elevation of privilege.
2. Devices Involved: The source IP address (39.91.166[.]222) is attempting to communicate with the destination IP address (172.16.17.233, MS-SharePointServer). This establishes the direction of the network traffic, with the source likely being the potential attacker, and the destination being the server being targeted.
3. Threat Intelligence: Cross-referencing the destination IP address revealed that it had been categorized as Command and Control (C2) and malicious. Multiple threat intelligence platforms also flagged the IP address as malicious, with a geolocation in China.
4. HTTP Traffic Analysis: Examination of the provided SP-IIS.log file showed multiple "GET" requests to URIs under "/_api/web/siteusers" with a "python-requests/2.28.1" User-Agent. Most requests resulted in "200" OK responses, indicating successful execution.
5. Indicators of Compromise (IOCs): The investigation identified several potential IOCs, including IP addresses and URLs related to the attack, as well as the "python-requests/2.28.1" User-Agent. These IOCs are crucial for ongoing monitoring and threat detection.
6. Threat Hunting: The investigation highlighted the need to explore the technical tactics and procedures of the attack to establish a more effective detection mechanism. GitHub repositories containing scripts for exploiting the CVE-2023-29357 vulnerability were discovered, which may provide valuable insights.
7. Direction of Traffic: All malicious traffic identified in the log management system originated from the Internet and targeted the Company Network.

Lesson Learned

- Timely threat intelligence is crucial for identifying and responding to emerging vulnerabilities and exploits.

- Monitoring for specific indicators of compromise (IOCs) helps detect potential security threats, but they should be supplemented with in-depth analysis.

- Effective threat hunting and detailed investigation are essential to understand the scope of an attack and its potential impact on the organization.

Remediation Actions

- Employ WAFs to filter and block malicious HTTP requests that exploit vulnerabilities in web applications..

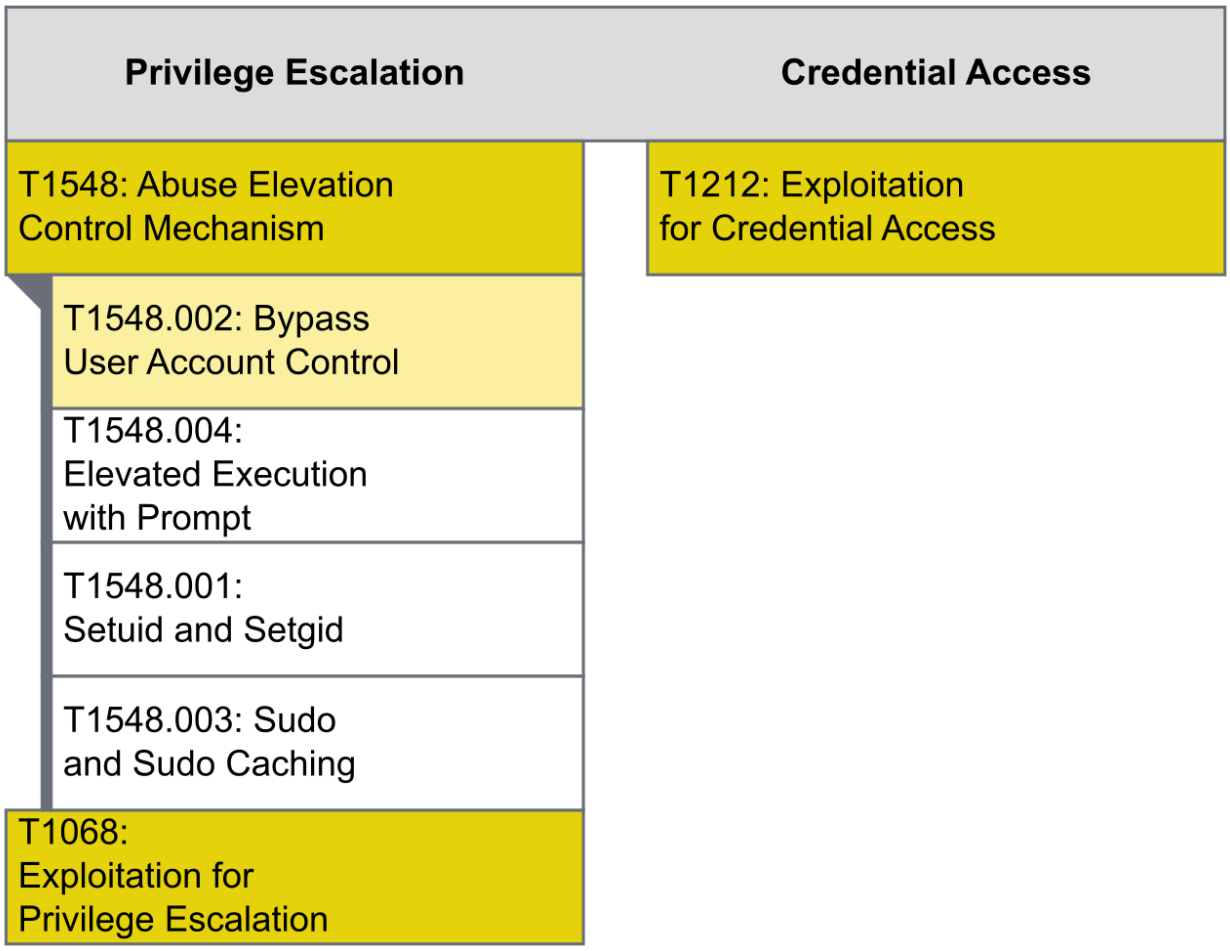
- Apply security patches or updates to address the CVE-2023-29357 vulnerability in the Microsoft SharePoint Server to eliminate the attack vector.

- Continuously monitor and update threat intelligence sources to stay informed about emerging threats and vulnerabilities.

- Isolate the compromised machine from the network to prevent the attacker from accessing other resources and systems within the organization.

Appendix

MITRE ATT&CK



MITRE Tactics	MITRE Techniques
Privilege Escalation	T1548: Abuse Elevation Control Mechanism
Privilege Escalation	T1068: Exploitation for Privelege Escalation
Credential Access	T1212: Exploitation for Credential Access

Artifacts

IOC TYPE	VALUE
URI	/_api/web/siteusers
URI	/_api/web/siteusers/web/siteusers
URI	/_api/web/currentuser
IPv4	39[.]91[.]166[.]222
User-Agent	python-requests/2.28.1
POC	https://github[.]com/Chocapikk/CVE-2023-29357