Author: Jack Fitzgerald

Official Incident Report

Date: Feb, 26, 2022, 06:56 PM

Event ID: 116

Rule Name: SOC166 - Javascript Code Detected in Requested URL

Table of Contents

Alert Details

Severity: Medium

Type: Web Attack

Hostname: WebServer1002

Destination IP Address: 172.16.17.17

Source IP Address: 112.85.42.13

HTTP Request Method: GET

Requested URL: https://172.16.17.17/search/?q=<$script>javascript:$alert(1)<$/script>

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1

Alert Trigger Reason: Javascript code detected in URL

Device Action: Allowed

Based on the information provided in the alert, it appears that an attacker has attempted to **insert Javascript code into a letsdefend domain URL**, running on **WebServer1002** to host **172.16.17.17**. The alert is triggered by rule SOC166 - Javascript code detected in URL.

It is important to check the HTTP Response status code to see if the attack was successful.

The device action is marked **allowed**, indicating that the code injection was executed on the host machine.

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 analyse the logs collected from the host by our security products.

At Feb, 26, 2022, 06:56 PM, **Webserver1002** with the IP address **172.16.17.17** received an HTTP GET request from **112.85.42.13** containing **https://172.16.17.17/search/?q=<$script>javascript:$alert(1)<$/script>** request URL.

It is important to understand why the alert was triggered:

- **Rulename:** Javascript Code Detected in Requested URL
- **Alert Reason:** Javascript code detected in URL
- **Source Address:** 112.85.42.13
- **Destination Address:** 172.16.17.17
- **Protocol:** TCP

It is also important to understand where the traffic is coming from and what the target of the web attack is:

- **Primary User** of WebServer1002 (172.16.16.17): webadmin15
- **Last user logon**: Feb, 02, 2022, 03:40 PM
- Traffic is coming from **outside** (Internet)
- **Location** of 112.85.42.13: Jiangsu, China
- **Reputation** of 112.85.42.13: Poor

To gather this information, we analyse the source address **112.85.42.13** using online analysis tools such as **VirusTotal** and **AbuseIPDB**.

https://www.virustotal.com/gui/ip-address/112.85.42.13

1 detected file communicating with this IP address

https://www.abuseipdb.com/check/112.85.42.13

We should examine the HTTP traffic to understand what sort of web attack is occurring.

This sort of web attack shows patterns of cross-site scripting (XSS). According to OWASP, XSS is a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.

Analysis

The next step of our investigation into this XSS web attack is to analyse the logs of the incident. We found that 8 events occurred originating from this alert, coming from a source address of **112.85.42.13** to the **WebServer1002** destination address of **172.16.17.17**.

The first log we find seems to be regular web server traffic with a successful response.

source_port	49183
destination_address	172.16.17.17
destination_port	443
time	Feb, 26, 2022, 06:34 PM
Raw Log	
Request URL	https://172.16.17.17/
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Request Method	GET
Device Action	Permitted
HTTP Response Size:	1024
HTTP Response Status	200

The second log we find seems to access the about use page, perhaps performing some reconnaissance, with a successful response.

source_port	49182
destination_address	172.16.17.17
destination_port	443
time	Feb, 26, 2022, 06:35 PM
Raw Log	
Request URL	https://172.16.17.17/about-us/
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Request Method	GET
Device Action	Permitted
HTTP Response Size:	3531
HTTP Response Status	200

The third log we find seems to test the search function of the browser to see what response they get, with a successful response.

source_port	48189
destination_address	172.16.17.17
destination_port	443
time	Feb, 26, 2022, 06:45 PM
Raw Log	
Request URL	https://172.16.17.17/search/?q=test
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Request Method	GET
Device Action	Permitted
HTTP Response Size:	885
HTTP Response Status	200

The fourth log we find seems to be a user's attempt to use a function or command with a parameter inside the search bar, with a temporary redirect response.

source_port	47283
destination_address	172.16.17.17
destination_port	443
time	Feb, 26, 2022, 06:46 PM
Raw Log	
Request URL	https://172.16.17.17/search/?q=prompt(8)
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Request Method	GET
Device Action	Permitted
HTTP Response Size:	0
HTTP Response Status	302

The fifth log we find seems to be a XSS payload being inserted into the URL query parameter, with a temporary redirect response.

source_port	49183
destination_address	172.16.17.17
destination_port	443
time	Feb, 26, 2022, 06:46 PM
Raw Log	
Request URL	https://172.16.17.17/search/?q=<$img%20src%20=q%20onerror=prompt(8)$>
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Request Method	GET
Device Action	Permitted
HTTP Response Size:	0
HTTP Response Status	302

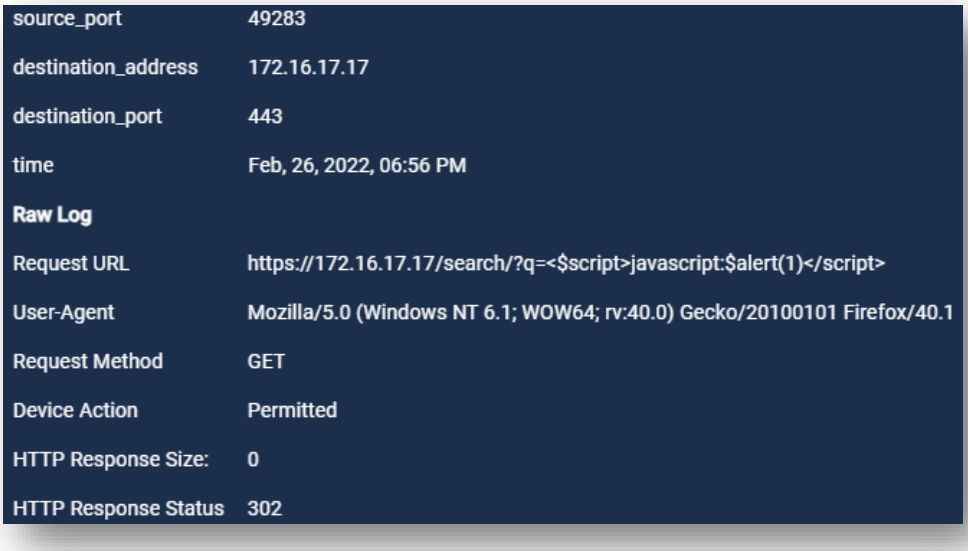
The sixth log we find seems to be another XSS attempt, trying to get the browser to run alert(1) by injecting HTML/JS into the q parameter. If the web app reflects that parameter into a page without proper escaping, this kind of input can cause script execution in a victim's browser. The page responded with a temporary redirect response.

source_port	49263
destination_address	172.16.17.17
destination_port	443
time	Feb, 26, 2022, 06:53 PM
Raw Log	
Request URL	https://172.16.17.17/search/?q=<$svg><$script%20?>$alert(1)
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Request Method	GET
Device Action	Permitted
HTTP Response Size:	0
HTTP Response Status	302

The seventh log we find seems to attempt to inject a script into a page and then try to call whatever global names exist until one of them behaves like a function, with a temporary redirect response.

source_port	49243
destination_address	172.16.17.17
destination_port	443
time	Feb, 26, 2022, 06:50 PM
Raw Log	
Request URL	https://172.16.17.17/search/?q=<$script>$for((i)in(self))eval(i)(1)<$/script>
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Request Method	GET
Device Action	Permitted
HTTP Response Size:	0
HTTP Response Status	302

The eighth and final log we find is the log that was alerted by the SIEM. It appears to be a javascript injection attempt, trying to evade detection, with a temporary redirect response.



source_port	49283
destination_address	172.16.17.17
destination_port	443
time	Feb, 26, 2022, 06:56 PM
Raw Log	
Request URL	https://172.16.17.17/search/?q=<$script>javascript:$alert(1)</script>
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Request Method	GET
Device Action	Permitted
HTTP Response Size:	0
HTTP Response Status	302

From these analyses, we can conclude that the attacker's **XSS** attack on **WebServer1002** was **unsuccessful**, due to HTTP Response Size: 0, and HTTP Response Status: 302 (redirect).

Containment

Based on the information gathered during the investigation, it is highly unlikely that the system has been compromised. There is no need to isolate the system from the network or escalate this matter to Tier 2.

Summary

The incident involves a non-compromised system named **WebServer1002** with an IP address of **172.16.17.17**. The alert was triggered by the detection of a javascript injection into a URL, based on the rule SOC166 - Javascript Code Detected in Requested URL.

Upon further analysis, it was discovered that the source address **112.85.42.13** was used, communicated with the server 8 different times, performing multiple **XSS** code injection attempts. This is evidenced by the log analysis seen on Feb, 26, 2022.

Based on the findings of the incident, a legitimate XSS attempt was detected and blocked by existing web application protections, with repeated HTTP 302 redirects preventing execution. Logging and detection operated correctly, and SOC triage validated the threat.

Improvements include ensuring intentional input validation and reviewing redirect behaviour. While no compromise occurred, the incident reinforces the need for continuous monitoring and defense enhancement.

No immediate action needs to be taken to isolate the compromised system, as the SIEM handled and redirected the attack, and the event was identified as a **True Positive**.

Lessons Learned

- Current WAF rules, sanitization, or application-side input validation is effective at blocking basic XSS probes
- Behavioral measures (redirects, safe error pages) helped contain injection attempts
- This event is a reminder that regular probing of publicly accessible endpoints is constant and expected

Remediation Actions

- Validate that all user input is sanitized and encoded server-side
- Review WAF logs and confirm block rules are intentional and up to date
- Add rate limiting/throttling if repeated XSS attempts came from the same IP

Appendix

MITRE ATT&CK

MITRE Tactics	MITRE Techniques
Initial Access	T1190 - Exploit Public-Facing Application
Execution	T1059.007 - Command and Scripting Interpreter: JavaScript
Reconnaissance	T1595 - Active Scanning
Reconnaissance	T1594 - Search Victim-Owned Websites

Artifacts

Value	Comment	Type
https://172.16.17.17/search/?q=<$script>javascript:$alert(1)<$/script>	Alerted XSS URL request	URL Address
112.85.42.13	Attacker source address	IP Address

LetsDefend Playbook

LetsDefend Event ID: 116