Jan 22 2024

NETMANAGEIT

Intelligence Report

Apache ActiveMQ Vulnerability Leads to Stealthy Godzilla Webshell

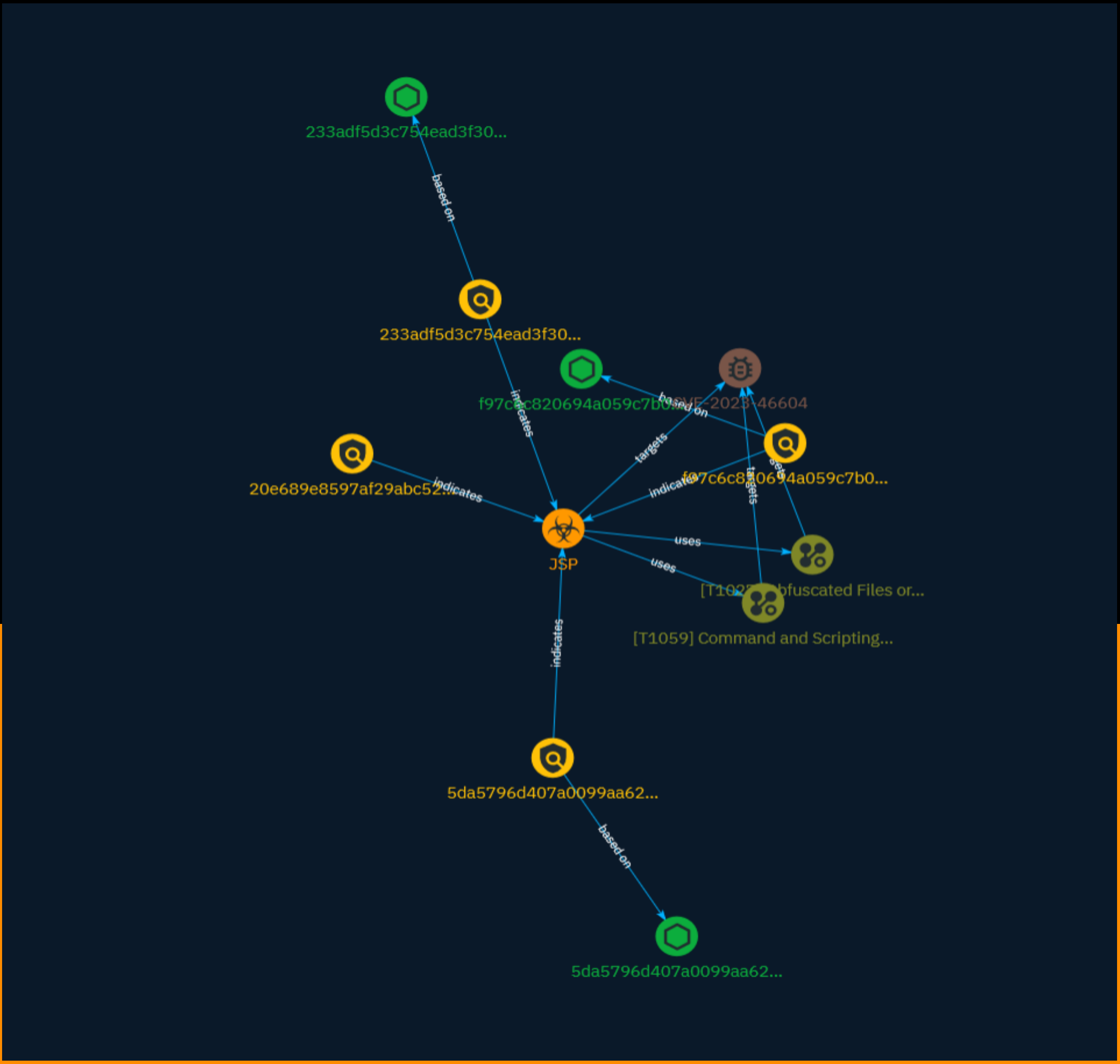

Table of contents

Overview

Entities

Observables

External References

Overview

Description

Researchers have observed a surge in attacks exploiting vulnerabilities in Apache ActiveMQ hosts. In certain cases, these host malicious Java Server Pages (JSP) web shells.

Confidence

This value represents the confidence in the correctness of the data contained within this report.

15 / 100

Content

N/A

Attack-Pattern

Name
Obfuscated Files or Information
ID
T1027
Description

Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses. Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and [Deobfuscate/Decode Files or Information](https://attack.mitre.org/techniques/T1140) for [User Execution](https://attack.mitre.org/techniques/T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016) Adversaries may also use compressed or archived scripts, such as JavaScript. Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. (Citation: Linux/Cdorked.A We Live Security Analysis) Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. (Citation: Carbon Black Obfuscation Sept 2016) Adversaries may also abuse [Command Obfuscation](https://attack.mitre.org/techniques/T1027/010) to obscure commands executed from payloads or directly via [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059). Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and application control mechanisms. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Revoke-Obfuscation July 2017)(Citation: PaloAlto EncodedCommand March 2017)

Name

Command and Scripting Interpreter

ID

T1059

Description

Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common feature across many different platforms. Most systems come with some built-in command-line interface and scripting capabilities, for example, macOS and Linux distributions include some flavor of [Unix Shell](https://attack.mitre.org/techniques/T1059/004) while Windows installations include the [Windows Command Shell](https://attack.mitre.org/techniques/T1059/003) and [PowerShell](https://attack.mitre.org/techniques/T1059/001). There are also cross-platform interpreters such as [Python] (https://attack.mitre.org/techniques/T1059/006), as well as those commonly associated with client applications such as [JavaScript](https://attack.mitre.org/techniques/T1059/007) and [Visual Basic](https://attack.mitre.org/techniques/T1059/005). Adversaries may abuse these technologies in various ways as a means of executing arbitrary commands. Commands and scripts can be embedded in [Initial Access](https://attack.mitre.org/tactics/TA0001) payloads delivered to victims as lure documents or as secondary payloads downloaded from an existing C2. Adversaries may also execute commands through interactive terminals/shells, as well as utilize various [Remote Services](https://attack.mitre.org/techniques/T1021) in order to achieve remote Execution. (Citation: Powershell Remote Commands)(Citation: Cisco IOS Software Integrity Assurance - Command History)(Citation: Remote Shell Execution in Python)

Indicator

Name

233adf5d3c754ead3f304a4891d367884dd615d74d9983119546bebb346b7bf7

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'233adf5d3c754ead3f304a4891d367884dd615d74d9983119546bebb346b7bf7']

Name

5da5796d407a0099aa624b1ea73a877a5197b3b31529d94f2467dce19fe3a74a

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'5da5796d407a0099aa624b1ea73a877a5197b3b31529d94f2467dce19fe3a74a']

Name

20e689e8597af29abc5221d8b5b60db3d78e2053

Description

Detects Godzilla Webshell JSP Code

Pattern Type

yara

Pattern

rule Godzilla_webshell { meta: description = "Detects Godzilla Webshell JSP Code" strings: $s1 = "String xc=" ascii wide $s2 = "String pass=" ascii wide $s3 = "String md5=md5(pass+xc)" ascii wide $s4 = "payload" ascii wide $s5 = "X(ClassLoader z)" ascii wide condition: all of them }

Name

f97c6c820694a059c7b0b2f3abe1f614b925dd4ab233d11472b062325ffb67be

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'f97c6c820694a059c7b0b2f3abe1f614b925dd4ab233d11472b062325ffb67be']

Malware

Name
JSP

Vulnerability

Name
CVE-2023-46604
Description
Apache ActiveMQ contains a deserialization of untrusted data vulnerability that may allow a remote attacker with network access to a broker to run shell commands by manipulating serialized class types in the OpenWire protocol to cause the broker to instantiate any class on the classpath.

StixFile

Value
233adf5d3c754ead3f304a4891d367884dd615d74d9983119546bebb346b7bf7
5da5796d407a0099aa624b1ea73a877a5197b3b31529d94f2467dce19fe3a74a
f97c6c820694a059c7b0b2f3abe1f614b925dd4ab233d11472b062325ffb67be

External References

- https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/apache-activemq-vulnerability-leads-to-stealthy-godzilla-webshell/
- https://otx.alienvault.com/pulse/65a98ab3cc5a953a2fa6dcff