GoTitan Botnet - Ongoing Exploitation on Apache ActiveMQ

fortinet.com/blog/threat-research/gotitan-botnet-exploitation-on-apache-activemq

November 28, 2023

Article Contents

By Cara Lin | November 28, 2023
Affected Platforms: Any OS running Apache Active MQ versions prior to 5.15.16, 5.16.7, 5.17.6, and 5.18.3
Impacted Parties: Any organization
Impact: Remote attackers gain control of the vulnerable systems
Severity Level: Critical

This past October, Apache issued a critical advisory addressing CVE-2023-46604, a vulnerability involving the deserialization of untrusted data in Apache. On November 2, the Cybersecurity and Infrastructure Security Agency (CISA) added CVE-2023-46604 to its known exploited list, KEV Catalog, indicating this vulnerability's high risk and impact. Fortiguard Labs also released an outbreak alert and a threat signal report about the active exploitation of CVE-2023-46604, providing more details and recommendations for mitigation.

Technical details and proof-of-concept (PoC) code for CVE-2023-46604 are publicly available, making it easier for attackers to exploit this vulnerability. In recent weeks, Fortiguard Labs has detected numerous threat actors exploiting CVE-2023-46604 to disseminate diverse strains of malware. Our analysis has unveiled the emergence of a newly discovered Golang-based botnet named GoTitan and a .NET program called "PrCtrl Rat," equipped with remote control capabilities. Additionally, we have identified other well-known malware and tools in play. Initially developed as an advanced penetration testing tool and red teaming framework, Sliver supports various callback protocols, including DNS, TCP, and HTTP(S), streamlining egress processes. Kinsing has solidified its position in cryptojacking operations, showcasing its ability to quickly capitalize on newly discovered vulnerabilities. Meanwhile, Ddostf, with a history dating back to 2016, continues to exhibit its proficiency in executing targeted Distributed Denial of Service (DDoS) attacks.

This article will detail the exploitation and provide insights into the malware associated with these recent attacks.

Exploitation

The attacker initiates a connection to ActiveMQ through the OpenWire protocol, typically on port 61616. By transmitting a crafted packet, the attacker triggers the system to unmarshal a class under their control. This action, in turn, prompts the vulnerable server to retrieve and load a class configuration XML file from a specified remote URL, requiring the presence of a predefined XML file hosted externally.

The known exploitation of this vulnerability involves leveraging the “ClassPathXmlApplicationContext” to load a malicious XML application configuration file from a network location via HTTP. Figure 1 shows the captured attacking traffic. The malicious XML file defines the arbitrary code intended to execute on the compromised machine. Attackers can set parameters like “cmd” or “bash” to achieve code execution on the remote vulnerable server (Figure 2).

In the following sections, we will explain how the malware works and what it does on infected systems.

Figure 1: Attacking traffic for CVE-2023-46604

Figure 2: Malicious XML files

GoTitan

Figure 3: GoTitan's XML file

GoTitan is a new botnet discovered earlier this month. It is written in the Go programming language and is downloaded from a malicious URL, “hxxp://91.92.242.14/main-linux-amd64s”. The attacker only provides binaries for x64 architectures, and the malware performs some checks before running. It also creates a file named "c.log" that records the execution time and program status. This file seems to be a debug log for the developer, which suggests that GoTitan is still in an early stage of development.

Figure 4: Save the log file

It replicates itself as “/.mod” within the system and establishes a recurring execution by registering in the cron. It then retrieves the C2 IP address and gathers essential information about the compromised endpoint, including architecture, memory, and CPU details. Compiling all the collected data using “<==>” as separators, it transmits its collected information to the C2 server. The C2 message initiates with the hard coded string “Titan<==>”.

Figure 5: Construct C2 message

Figure 6: C2 traffic session for GoTitan

GoTitan communicates with its C2 server by sending “\xFE\xFE” as a heartbeat signal and waiting for further instructions. When it receives a command, it passes it to a function named “handle_socket_func2” that determines an attack method. GoTitan supports ten different methods of launching distributed denial-of-service (DDoS) attacks: UDP, UDP HEX, TCP, TLS, RAW, HTTP GET, HTTP POST, HTTP HEAD, and HTTP PUT.

Sliver

Figure 7: Sliver's XML file

Sliver, an open-source penetration testing tool developed in the Go language and available on GitHub, possesses the potential for misuse when wielded by threat actors due to its diverse features catering to each stage of penetration testing. Threat actors can leverage Sliver to compromise and control multiple targets across various platforms and architectures. The tool enables the generation of customized implants designed to elude detection, allowing for the execution of commands, file uploads and downloads, screenshot capture, and more on infected systems.

When communicating with the C2 server at “91[.]92[.]240[.]41” via HTTP requests, Sliver dynamically selects decoders for C2 messages based on parameters in the URI. Additionally, Sliver supports various encoders, including Base32, Base58, Base64, English encoder, Gzip, Hex, and PNG. The encoded C2 communication in HTTP protocol is shown in Figure 8.

Figure 8: C2 session for Sliver

PrCtrl Rat

Figure 9: PrCtrl Rat’s XML file

The attacker retrieves the execution file from “hxxp://199[.]231[.]186[.]249:8000/unifo.dat” and stores it as “svc_veeam.exe”. The file 'unifo.dat' is a .Net framework program initially labeled as “prcli.exe” that was created in August and still spread via CVE-2023-46604. Figure 10 shows the PDB path and detailed information.

Figure 10: Information for uninfo.dat

For persistence, it adds “Security Service” with the current process into the registry “HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run.”

It then starts the connection to C2 server “173[.]214[.]167[.]155.” Once the command is received from a remote server, it checks for a length of four. If not, it exits the program. It supports five commands:

cmdc: Running cmd.exe with a specific command and returning the result to the server.

file: Get file system information on a target system, such as drives or the directory, and files.

- upld: Upload file.
- dnld: Download file.
- ping: Heartbeat.

As of this writing, we have yet to receive any messages from the server, and the motive behind disseminating this tool remains unclear. However, once it infiltrates a user's environment, the remote server gains control over the system.

Kinsing

Figure 11: Kinsing's XML file

Kinsing fetches the bash script from “194[.]38[.]22[.]53/acb.sh.” It serves the following purposes:

- System Configuration: Modifies system parameters, such as disabling the firewall, flushing iptables rules, and turning off the NMI watchdog.
- Dependency Check: Verifies the existence of curl or wget and installs them if they are absent.
- Process Cleanup: Terminates processes associated with specific executable names and competing miners.

Binary Download and Verification: Downloads a main binary and a shared object file and then verifies the integrity of the downloaded binary using MD5 checksum.

- System Configuration: Creates a system service configuration file for the downloaded binary.
- Cronjob Setting: Removes specific entries from the crontab related to known malicious activities. Adds a new cronjob to periodically execute a command fetched from a remote server hxxp://185[.]122[.]204[.]197/acb.sh
- Cleanup: Clears command history and removes bash history files.

Ddostf

Figure 12: Ddostf's XML file

The batch script used by Ddostf is retrieved from “hxxp://42[.]121[.]111[.]112:81/xml.sh.” It configures the history log with “+o” to prevent the recording of the current session. It then installs curl to download additional execution files and eliminate any traces.

Figure 13: Batch script to deploying Ddostf

The executable file “tomcat” includes the recognizable string “ddos.tf” and the Base64-encoded string for “v8.ter.tf.” Its characteristics align with those of a threat actor who had targeted China in 2018.

Figure 14: Ddostf's binary data

It first verifies that it has root privilege and that the process is running on the device. It then ensures that it will persist on the device by executing the command shown below.

Figure 15: Ddostf's setting

Ddostf includes a hard-coded string, “TF-Linux kernel…,” which appends either “SYN-“ or “UDP-“ in its C2 message, depending on whether the process runs with root privileges.

Figure 16: Send C2 message

Ddostf incorporates 13 attack methods: SYN_Flood, WZSYN_Flood, ICMP_Flood, GET_Flood, GETFT_Flood, HEAD_Flood, POST_Flood, xzcc_Flood, TCP_Flood, WZTCP_Flood, ack_Flood, WZUDP_Flood, and UDP_Flood. Additionally, it defines a function called “DNS_Flood,” which is not included in the current switch cases and is possibly intended for future enhancements.

Figure 17: DNS flood function

Conclusion

Despite the release of a patch for CVE-2023-46604 over a month ago, threat actors persist in exploiting this vulnerability to distribute malware on susceptible servers. This blog introduces newly discovered threats, including the Golang-based botnet GoTitan and the .NET program “PrCtrl Rat,” which have emerged as a consequence of this exploitation.

Additionally, users should remain vigilant against ongoing exploits by Sliver, Kinsing, and Ddostf. It is crucial to prioritize system updates and patching and regularly monitor security advisories to effectively mitigate the risk of exploitation.

Fortinet Protections

The malware described in this report are detected and blocked by FortiGuard Antivirus as:

XML/Agent.E2ED!tr
BASH/Miner.BPH!tr
BASH/Agent.5C93!tr
ELF/GoTitan.AR!tr
Linux/Sliver.AE!tr
ELF/Ddostf.D!tr
MSIL/Agent.F3D5!tr

FortiGate, FortiMail, FortiClient, and FortiEDR support the FortiGuard AntiVirus service. The FortiGuard AntiVirus engine is a part of each of those solutions. As a result, customers who have these products with up-to-date protections are protected.

Fortinet has also released an IPS signature to proactively protect our customers from the threats contained in the report:

CVE-2023-46604: Apache.ActiveMQ.CVE-2023-46604.Code.Execution

The URLs are rated as “Malicious Websites” by the FortiGuard Web Filtering service.

We also suggest that organizations use Fortinet’s free NSE training module: NSE 1 – Information Security Awareness. This module is designed to help end users learn how to identify and protect themselves from phishing attacks.

FortiGuard IP Reputation and Anti-Botnet Security Service proactively block these attacks by aggregating malicious source IP data from the Fortinet distributed network of threat sensors, CERTs, MITRE, cooperative competitors, and other global sources that collaborate to provide up-to-date threat intelligence about hostile sources.

If you believe this or any other cybersecurity threat has impacted your organization, please contact our Global FortiGuard Incident Response Team.

IOCs

IP List

185[.]122[.]204[.]197
194[.]38[.]22[.]53
42[.]121[.]111[.]112
91[.]92[.]242[.]14
199[.]231[.]186[.]249
173[.]214[.]167[.]155
91[.]92[.]240[.]41

Files

f75cb3e540b96cd54a966c512c854c832807e354772ae1a326b758394b01b607
dbf8ba47a5973c86fef32c2d696b09e1930a8384087c62ace1aa5c4084ee1a3f
1a3d9960a1685707f8cc2bc447c88f5c3278454fbf0a35a7959717ad835348cd
d8f55bbbcc20e81e46b9bf78f93b73f002c76a8fcdb4dc2ae21b8609445c14f9
0cc60a0c480e4d898fa77ab501bbd2afaf3f5fb89a2917a31e7f5fdaa6c3879c
ed09f95f4b4b482207bb300ff6ec15ed8ca5fdde97af02fa9fbe01adaaf7673b
bfce7938591dd9fa3e1368d7eb86fc7f11e935349437fc11de4f124bbbc16dee
f5a36570506bfaff60b684cd26dde3a64a3db4eaa9da78a1434cfd4b390ef3d5
5acf5ce55678519cd65e001d3f600fa1de288f1cd3e203b4c9439979f4b67175
923f2be3d55fcdab7da5cb2be3c16dfcc1582b83d1e4a831236445a52ca81878
b90abde8f449bbe6bec9495386fab1833c0654f83c7b2f5ebcf5b14743c30600