Mar 08 2024

NETMANAGEIT

Intelligence Report

Spinning YARN - A New Linux Malware Campaign Targets Docker, Apache Hadoop, Redis and Confluence

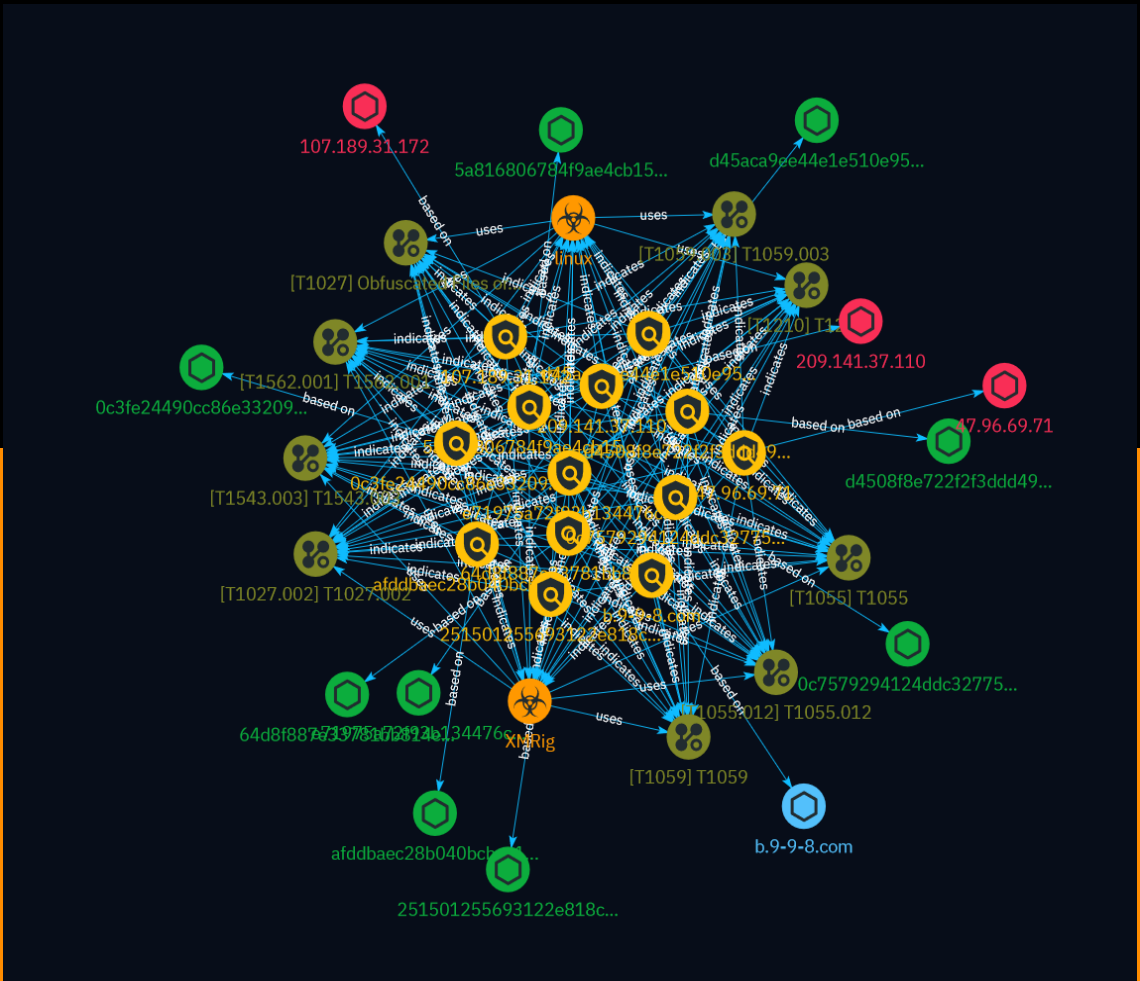

Table of contents

Overview

Entities

Observables

External References

Overview

Description

A new Linux malware campaign has been discovered that targets misconfigured servers running Docker Engine API, Apache Hadoop YARN, Confluence, and Redis. The attackers use four novel Golang binaries to identify vulnerable hosts and conduct RCE attacks, initially gaining access via Docker before deploying XMRig miners, reverse shells, and user mode rootkits on compromised systems.

Confidence

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

100 / 100

Content

N/A

Indicator

Name

e71975a72f93b134476c8183051fee827ea509b4e888e19d551a8ced6087e15c

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'e71975a72f93b134476c8183051fee827ea509b4e888e19d551a8ced6087e15c']

Name

afddbaec28b040bcbaa13decdc03c1b994d57de244befbdf2de9fe975cae50c4

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'afddbaec28b040bcbaa13decdc03c1b994d57de244befbdf2de9fe975cae50c4']

Name

d45aca9ee44e1e510e951033f7ac72c137fc90129a7d5cd383296b6bd1e3ddb5

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'd45aca9ee44e1e510e951033f7ac72c137fc90129a7d5cd383296b6bd1e3ddb5']

Name

64d8f887e33781bb814eaefa98dd64368da9a8d38bd9da4a76f04a23b6eb9de5

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '64d8f887e33781bb814eaefa98dd64368da9a8d38bd9da4a76f04a23b6eb9de5']

Name

5a816806784f9ae4cb1564a3e07e5b5ef0aa3d568bd3d2af9bc1a0937841d174

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '5a816806784f9ae4cb1564a3e07e5b5ef0aa3d568bd3d2af9bc1a0937841d174']

Name

0c7579294124ddc32775d7cf6b28af21b908123e9ea6ec2d6af01a948caf8b87

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '0c7579294124ddc32775d7cf6b28af21b908123e9ea6ec2d6af01a948caf8b87']

Name

251501255693122e818cadc28ced1ddb0e6bf4a720fd36dbb39bc7dedface8e5

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '251501255693122e818cadc28ced1ddb0e6bf4a720fd36dbb39bc7dedface8e5']

Name

0c3fe24490cc86e332095ef66fe455d17f859e070cb41cbe67d2a9efe93d7ce5

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '0c3fe24490cc86e332095ef66fe455d17f859e070cb41cbe67d2a9efe93d7ce5']

Name

47.96.69.71

Description

- **Zip Code:** N/A - **ISP:** Hangzhou Alibaba Advertising Co. - **ASN:** 37963 - **Organization:** Hangzhou Alibaba Advertising Co. - **Is Crawler:** False - **Timezone:** Asia/Shanghai - **Mobile:** False - **Host:** 47.96.69.71 - **Proxy:** True - **VPN:** True - **TOR:** False - **Active VPN:** False - **Active TOR:** False - **Recent Abuse:** True - **Bot Status:** False - **Connection Type:** Premium required. - **Abuse Velocity:** Premium required. - **Country Code:** CN - **Region:** Zhejiang - **City:** Hangzhou - **Latitude:** 30.29940033 - **Longitude:** 120.16120148

Pattern Type

stix

Pattern

[ipv4-addr:value = '47.96.69.71']

Name

209.141.37.110

Description

- **Zip Code:** N/A - **ISP:** FranTech Solutions - **ASN:** 53667 - **Organization:** FranTech Solutions - **Is Crawler:** False - **Timezone:** America/Los_Angeles -

Mobile: False - **Host:** 209.141.37.110 - **Proxy:** True - **VPN:** True - **TOR:** False - **Active VPN:** False - **Active TOR:** False - **Recent Abuse:** True - **Bot Status:** False - **Connection Type:** Premium required. - **Abuse Velocity:** Premium required. - **Country Code:** US - **Region:** Nevada - **City:** Las Vegas - **Latitude:** 36.10200119 - **Longitude:** -115.1446991

Pattern Type

stix

Pattern

[ipv4-addr:value = '209.141.37.110']

Name

d4508f8e722f2f3ddd49023e7689d8c65389f65c871ef12e3a6635bbaeb7eb6e

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'd4508f8e722f2f3ddd49023e7689d8c65389f65c871ef12e3a6635bbaeb7eb6e']

Name

b.9-9-8.com

Pattern Type

stix

Pattern

[hostname:value = 'b.9-9-8.com']

Name

107.189.31.172

Description

- **Zip Code:** N/A - **ISP:** FranTech Solutions - **ASN:** 53667 - **Organization:** FranTech Solutions - **Is Crawler:** False - **Timezone:** Europe/Luxembourg - **Mobile:** False - **Host:** 107.189.31.172 - **Proxy:** True - **VPN:** True - **TOR:** False - **Active VPN:** False - **Active TOR:** False - **Recent Abuse:** True - **Bot Status:** True - **Connection Type:** Premium required. - **Abuse Velocity:** Premium required. - **Country Code:** LU - **Region:** Luxembourg - **City:** Luxembourg - **Latitude:** 49.61130142 - **Longitude:** 6.12939978

Pattern Type

stix

Pattern

[ipv4-addr:value = '107.189.31.172']

Malware

Name
XMRig

Name
linux

Attack-Pattern

Name
T1027.002
ID
T1027.002
Description
Adversaries may perform software packing or virtual machine software protection to conceal their code. Software packing is a method of compressing or encrypting an executable. Packing an executable changes the file signature in an attempt to avoid signature-based detection. Most decompression techniques decompress the executable code in memory. Virtual machine software protection translates an executable's original code into a special format that only a special virtual machine can run. A virtual machine is then called to run this code.(Citation: ESET FinFisher Jan 2018) Utilities used to perform software packing are called packers. Example packers are MPRESS and UPX. A more comprehensive list of known packers is available, but adversaries may create their own packing techniques that do not leave the same artifacts as well-known packers to evade defenses.(Citation: Awesome Executable Packing)
Name
T1059.003
ID
T1059.003

Description

Adversaries may abuse the Windows command shell for execution. The Windows command shell ([cmd](https://attack.mitre.org/software/S0106)) is the primary command prompt on Windows systems. The Windows command prompt can be used to control almost any aspect of a system, with various permission levels required for different subsets of commands. The command prompt can be invoked remotely via [Remote Services](https://attack.mitre.org/techniques/T1021) such as [SSH](https://attack.mitre.org/techniques/T1021/004).(Citation: SSH in Windows) Batch files (ex: .bat or .cmd) also provide the shell with a list of sequential commands to run, as well as normal scripting operations such as conditionals and loops. Common uses of batch files include long or repetitive tasks, or the need to run the same set of commands on multiple systems. Adversaries may leverage [cmd](https://attack.mitre.org/software/S0106) to execute various commands and payloads. Common uses include [cmd](https://attack.mitre.org/software/S0106) to execute a single command, or abusing [cmd](https://attack.mitre.org/software/S0106) interactively with input and output forwarded over a command and control channel.

Name

T1055.012

ID

T1055.012

Description

Adversaries may inject malicious code into suspended and hollowed processes in order to evade process-based defenses. Process hollowing is a method of executing arbitrary code in the address space of a separate live process. Process hollowing is commonly performed by creating a process in a suspended state then unmapping/hollowing its memory, which can then be replaced with malicious code. A victim process can be created with native Windows API calls such as `CreateProcess`, which includes a flag to suspend the processes primary thread. At this point the process can be unmapped using APIs calls such as `ZwUnmapViewOfSection` or `NtUnmapViewOfSection` before being written to, realigned to the injected code, and resumed via `VirtualAllocEx`, `WriteProcessMemory`, `SetThreadContext`, then `ResumeThread` respectively.(Citation: Leitch Hollowing)(Citation: Elastic Process Injection July 2017) This is very similar to [Thread Local Storage](https://attack.mitre.org/techniques/T1055/005) but creates a new process rather than targeting an existing process. This behavior will likely not result in elevated privileges since the

injected process was spawned from (and thus inherits the security context) of the injecting process. However, execution via process hollowing may also evade detection from security products since the execution is masked under a legitimate process.

Name

T1562.001

ID

T1562.001

Description

Adversaries may modify and/or disable security tools to avoid possible detection of their malware/tools and activities. This may take many forms, such as killing security software processes or services, modifying / deleting Registry keys or configuration files so that tools do not operate properly, or other methods to interfere with security tools scanning or reporting information. Adversaries may also disable updates to prevent the latest security patches from reaching tools on victim systems.(Citation: SCADAfence_ransomware) Adversaries may also tamper with artifacts deployed and utilized by security tools. Security tools may make dynamic changes to system components in order to maintain visibility into specific events. For example, security products may load their own modules and/or modify those loaded by processes to facilitate data collection. Similar to [Indicator Blocking](https://attack.mitre.org/techniques/T1562/006), adversaries may unhook or otherwise modify these features added by tools (especially those that exist in userland or are otherwise potentially accessible to adversaries) to avoid detection.(Citation: OutFlank System Calls)(Citation: MDSec System Calls) Adversaries may also focus on specific applications such as Sysmon. For example, the “Start” and “Enable” values in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Autologger\EventLog-Microsoft-Windows-Sysmon-Operational` may be modified to tamper with and potentially disable Sysmon logging.(Citation: disable_win_evt_logging) On network devices, adversaries may attempt to skip digital signature verification checks by altering startup configuration files and effectively disabling firmware verification that typically occurs at boot.(Citation: Fortinet Zero-Day and Custom Malware Used by Suspected Chinese Actor in Espionage Operation)(Citation: Analysis of FG-IR-22-369) In cloud environments, tools disabled by adversaries may include cloud monitoring agents that report back to services such as AWS CloudWatch or Google Cloud Monitor. Furthermore, although defensive tools may have anti-tampering mechanisms, adversaries may abuse tools such as legitimate rootkit removal kits to impair and/or disable these tools.(Citation: chasing_avaddon_ransomware)(Citation: dharma_ransomware)(Citation:

demystifying_ryuk)(Citation: doppelpaymer_crowdstrike) For example, adversaries have used tools such as GMER to find and shut down hidden processes and antivirus software on infected systems.(Citation: demystifying_ryuk) Additionally, adversaries may exploit legitimate drivers from anti-virus software to gain access to kernel space (i.e. [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068)), which may lead to bypassing anti-tampering features.(Citation: avoslocker_ransomware)

Name

T1210

ID

T1210

Description

Adversaries may exploit remote services to gain unauthorized access to internal systems once inside of a network. Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. A common goal for post-compromise exploitation of remote services is for lateral movement to enable access to a remote system. An adversary may need to determine if the remote system is in a vulnerable state, which may be done through [Network Service Discovery](https://attack.mitre.org/techniques/T1046) or other Discovery methods looking for common, vulnerable software that may be deployed in the network, the lack of certain patches that may indicate vulnerabilities, or security software that may be used to detect or contain remote exploitation. Servers are likely a high value target for lateral movement exploitation, but endpoint systems may also be at risk if they provide an advantage or access to additional resources. There are several well-known vulnerabilities that exist in common services such as SMB (Citation: CIS Multiple SMB Vulnerabilities) and RDP (Citation: NVD CVE-2017-0176) as well as applications that may be used within internal networks such as MySQL (Citation: NVD CVE-2016-6662) and web server services.(Citation: NVD CVE-2014-7169) Depending on the permissions level of the vulnerable remote service an adversary may achieve [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068) as a result of lateral movement exploitation as well.

Name

T1059

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)

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

T1055

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

T1543.003

ID

T1543.003

Description

Adversaries may create or modify Windows services to repeatedly execute malicious payloads as part of persistence. When Windows boots up, it starts programs or applications called services that perform background system functions.(Citation: TechNet Services) Windows service configuration information, including the file path to the service's executable or recovery programs/commands, is stored in the Windows Registry. Adversaries may install a new service or modify an existing service to execute at startup in order to persist on a system. Service configurations can be set or modified using system utilities (such as sc.exe), by directly modifying the Registry, or by interacting directly with the Windows API. Adversaries may also use services to install and execute malicious drivers. For example, after dropping a driver file (ex: `.sys`) to disk, the payload can be loaded and registered via [Native API](https://attack.mitre.org/techniques/T1106) functions such as `CreateServiceW()` (or manually via functions such as `ZwLoadDriver()` and `ZwSetValueKey()`), by creating the required service Registry values (i.e. [Modify Registry](https://attack.mitre.org/techniques/T1112)), or by using command-line utilities such as `PnPUtil.exe`.(Citation: Symantec W.32 Stuxnet Dossier)(Citation: Crowdstrike DriveSlayer February 2022)(Citation: Unit42 AcidBox June 2020) Adversaries may leverage these drivers as [Rootkit](https://attack.mitre.org/techniques/T1014)s to hide the presence of malicious activity on a system. Adversaries may also load a signed yet vulnerable driver onto a compromised machine (known as "Bring Your Own Vulnerable Driver" (BYOVD)) as part of [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068).(Citation: ESET InvisiMole June 2020)(Citation: Unit42 AcidBox June 2020) Services may be created with administrator privileges but are executed under SYSTEM privileges, so an adversary may also use a service to escalate privileges. Adversaries may also directly start services through [Service Execution](https://attack.mitre.org/techniques/T1569/002). To make detection analysis more challenging, malicious services may also incorporate [Masquerade

Task or Service](https://attack.mitre.org/techniques/T1036/004) (ex: using a service and/or payload name related to a legitimate OS or benign software component).

StixFile

Value
e71975a72f93b134476c8183051fee827ea509b4e888e19d551a8ced6087e15c
d45aca9ee44e1e510e951033f7ac72c137fc90129a7d5cd383296b6bd1e3ddb5
afddbaec28b040bcbaa13decdc03c1b994d57de244befbdf2de9fe975cae50c4
64d8f887e33781bb814eaefa98dd64368da9a8d38bd9da4a76f04a23b6eb9de5
5a816806784f9ae4cb1564a3e07e5b5ef0aa3d568bd3d2af9bc1a0937841d174
251501255693122e818cadc28ced1ddb0e6bf4a720fd36dbb39bc7dedface8e5
0c7579294124ddc32775d7cf6b28af21b908123e9ea6ec2d6af01a948caf8b87
0c3fe24490cc86e332095ef66fe455d17f859e070cb41cbe67d2a9efe93d7ce5
d4508f8e722f2f3ddd49023e7689d8c65389f65c871ef12e3a6635bbaeb7eb6e

IPv4-Addr

Value
47.96.69.71
209.141.37.110
107.189.31.172

Hostname

Value
b.9-9-8.com

External References

- https://www.cadosecurity.com/spinning-yarn-a-new-linux-malware-campaign-targets-docker-apache-hadoop-redis-and-confluence/
- https://otx.alienvault.com/pulse/65eb4446548a548c2092d41c