

新威胁：闷声发大财的Fodcha僵尸网络

blog.netlab.360.com/men-sheng-fa-da-cai-fodchajiang-shi-wang-luo/

Hui Wang

April 13, 2022

13 April 2022 / Botnet

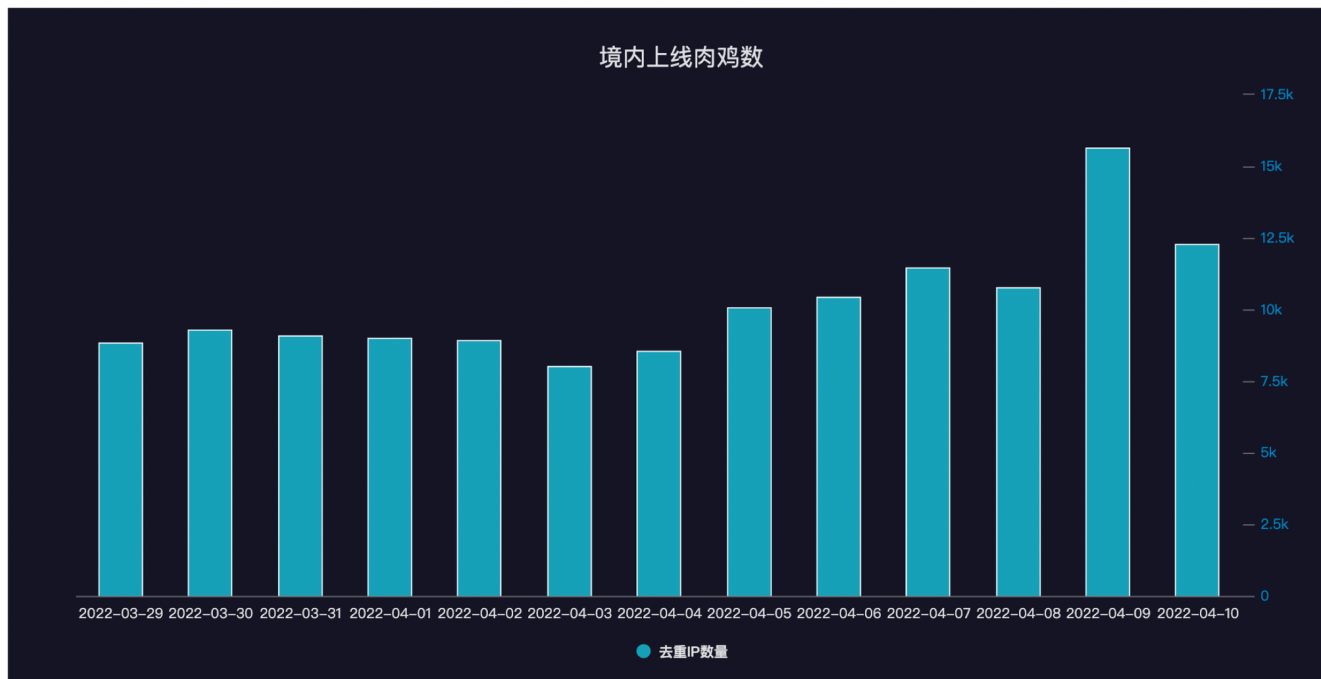
本报告由国家互联网应急中心（CNCERT）与三六零数字安全科技集团有限公司共同发布。

概述

近期，CNCERT和三六零数字安全科技集团有限公司共同监测发现一个新的且在互联网上快速传播的DDoS僵尸网络，通过跟踪监测发现其每日上线境内肉鸡数（以IP数计算）已超过1万、且每日会针对超过100个攻击目标发起攻击，给网络空间带来较大威胁。由于该僵尸网络最初使用的C2域名folded.in，以及使用chacha算法来加密网络流量，我们将其命名为Fodcha。

僵尸网络规模

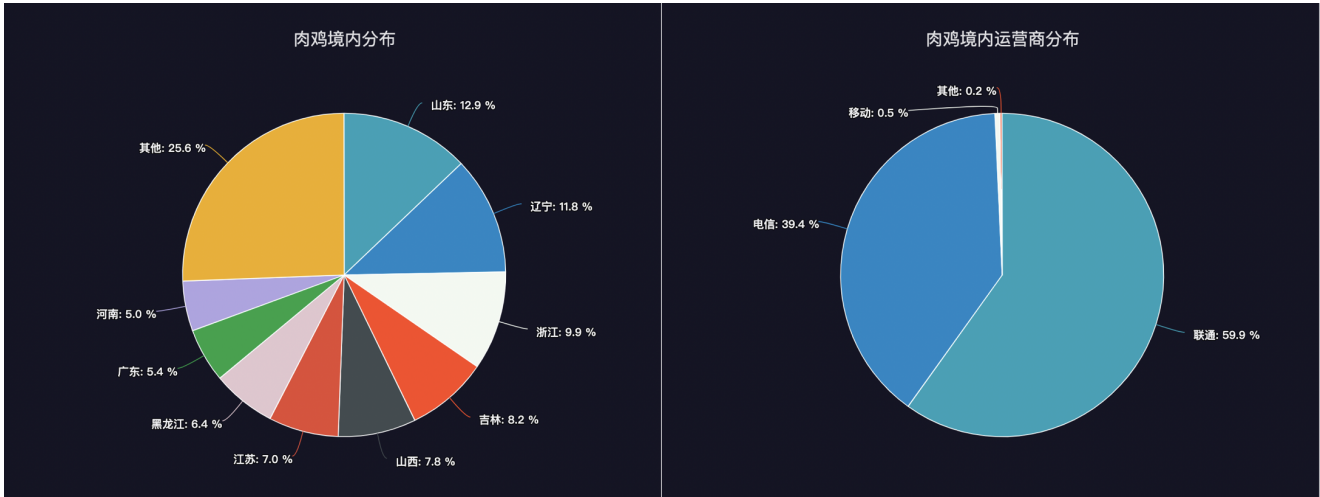
通过监测分析发现，2022年3月29日至4月10日Fodcha僵尸网络日上线境内肉鸡数最高达到1.5万台，累计感染肉鸡数达到6.2万。每日境内上线肉鸡数情况如下。



Netlab按：

根据国外合作伙伴的数据，我们估算该家族全球日活肉鸡数量应该在5.6w+

Fodcha僵尸网络位于境内肉鸡按省份统计，排名前三位的分别为山东省（12.9%）、辽宁省（11.8%）和浙江省（9.9%）；按运营商统计，联通占59.9%，电信占39.4%，移动占0.5%。

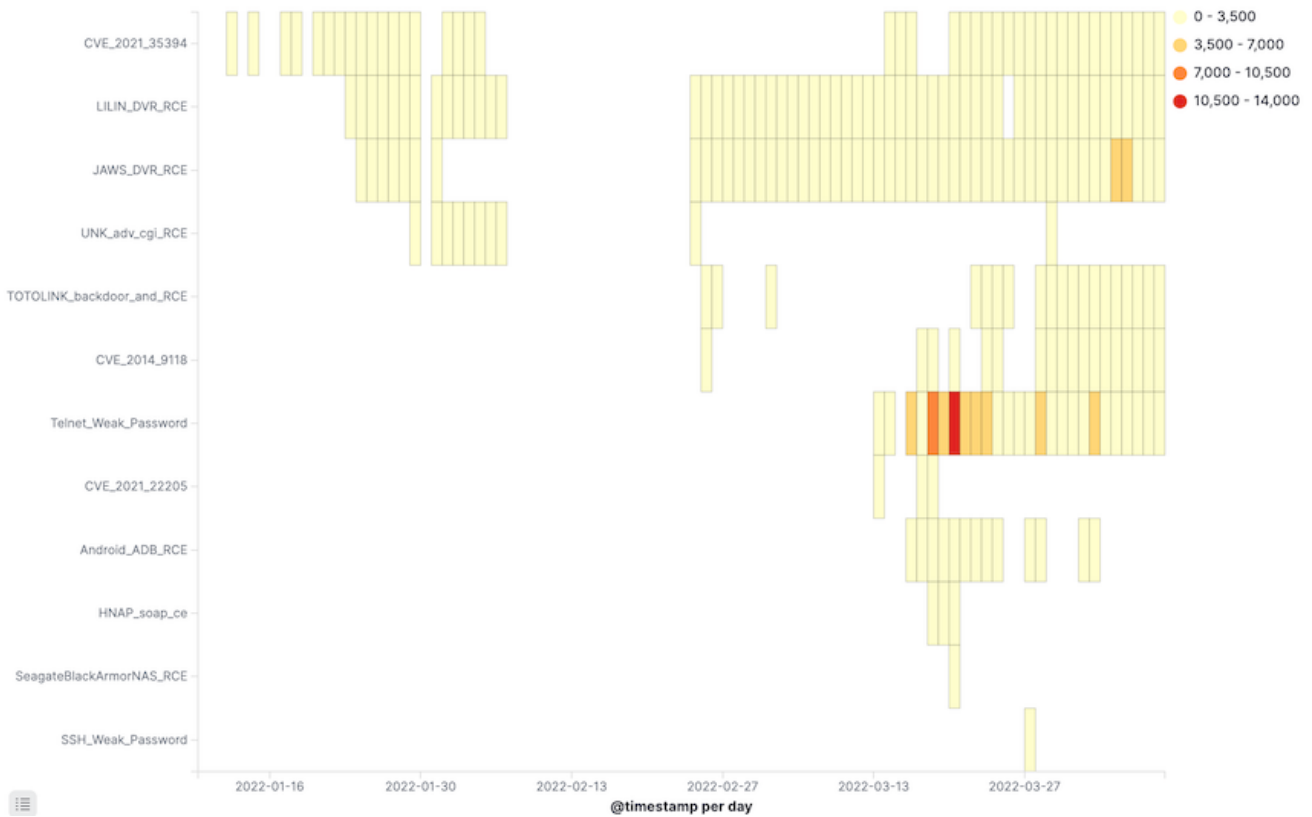


传播方式

通过跟踪监测，我们发现Fodcha主要通过以下NDay漏洞和Telnet/SSH弱口令传播，另外根据我们的数据分析，Fodcha的运营者还会利用Telnet暴破工具进行Telnet暴力破解。

Netlab按：

Telnet暴力破解扫描使用的是我们内部命名的 **Crazyfia** 的Telnet暴破工具，Fodcha的运营者会根据 **Crazyfia** 的扫描结果植入 **Fodcha** 样本。



漏洞列表：

Vulnerability	Affected Device/Service
<u>Android ADB Debug Server RCE</u>	Android
<u>CVE-2021-22205</u>	GitLab
<u>CVE-2021-35394</u>	Realtek Jungle SDK
<u>JAWS Webserver unauthenticated shell command execution</u>	MVPower DVR
<u>LILIN DVR RCE</u>	LILIN DVR
<u>TOTOLINK Routers Backdoor</u>	TOTOLINK Routers
<u>ZHONE Router Web RCE</u>	ZHONE Router

样本分析

Fodcha僵尸网络包括针对mips、mips1、arm、x86等CPU架构的样本。在近3个月的时间中，我们捕获的Fodcha样本可以分成v1、v2二个版本，它们的主要功能几乎是一样的，通过交叉对比不同版本，我们总结了Fodcha的以下4个主要特性，可以看出Fodcha运营者试图隐藏C2并在C2之间进行负载均衡。

Version	Chacha20	C2 Format	C2	MAPPING(Domain<->IP)	MAPPING(IP<->PORT)
v1	yes	plaintext	folded.in	1:N	N:1
v2	yes	ciphertext	fridgexperts.cc	1:N	N:10

本文选取最新的V2 X86 CPU架构的样本为主要的分析对象，它的基本信息如下：

```
8ea56a9fa9b11b15443b369f49fa9719
ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, stripped
Packer:None
```

Fodcha的功能非常简单，当它在被侵入设备运行时，首先会检测运行时的参数，如果不带参数，则直接退出，这是一种对通过沙箱抽取IOC的简单对抗；如果带有参数，则首先解密出敏感资源，在Console上输出**here we are**，然后使用随机字符串伪装进程名，最后和C2建立通信，等待执行C2下发的指令，下文将着重介绍Fodcha的解密方法和网络通信。

解密敏感资源

Fodcha使用一种多重Xor的加密方式来保护其敏感资源。

```

v0 = &C2;
v1 = calloc(0x10u, 1u);
byte_8052184 = 15;
dword_8052180[0] = (int)v1;
v2 = 0;
do
{
    v3 = *v0++ ^ aFjifnaefsedifs[v2 % 20];
    *(_BYTE *)(v2 + dword_8052180[0]) = v3 % 255;
    v4 = v2++;
    *(_BYTE *)(dword_8052180[0] + v4) ^= dword_8052028;
}
while ( v2 != 15 );
v5 = 0;
do
{
    *(_BYTE *)dword_8052180[0] ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 1) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 2) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 3) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 4) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 5) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 6) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 7) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 8) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 9) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 10) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 11) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 12) ^= aFjifnaefsedifs[v5];
    *(_BYTE *)(dword_8052180[0] + 13) ^= aFjifnaefsedifs[v5];
    v6 = aFjifnaefsedifs[v5++];
    *(_BYTE *)(dword_8052180[0] + 14) ^= v6;
}
while ( v5 != 20 );

```

其对应的python实现如下所示，以样本中的密文 EB D3 EB C9 C2 EF F6 FD FD FC FB F1 A3 FB E9 为例，解密后正是Fodcha的C2：fridgexperts.cc。

```

cipher = [0xEB, 0xD3, 0xEB, 0xC9, 0xC2, 0xEF, 0xF6, 0xFD, 0xFD, 0xFC, 0xFB, 0xF1,
0xA3, 0xFB, 0xE9]
key = [0x66, 0x4A, 0x69, 0x46, 0x4E, 0x61, 0x65, 0x66, 0x73, 0x65, 0x64, 0x69, 0x66,
0x73, 0x61, 0x69, 0x66, 0x73, 0x69,00]

tmp=[]

for i in range(len(cipher)):
    tmp.append((cipher[i] ^ key[i])%0xff^0xbe)

for i in range(len(tmp)):
    for j in key:
        tmp[i]^=j

out=''.join([chr(i) for i in tmp])
print(out)

```

网络通信

Fodcha通过以下代码片段和C2建立连接，

```

v4.sin_family = 2;
*(__DWORD *)&v4.sin_port = (unsigned __int16)__ROR2__(port_list[rand_next() % 0xAu], 8);
v0 = (char *)val_get(0, 0);
v1 = (void **)sub_804E4E0(v0);
v2 = v1;
if ( v1 )
{
    v3 = v1[1];
    v4.sin_addr.s_addr = v3[rand_next() % (unsigned int)*(unsigned __int8 *)v1];
    wrap_free(v2);
    fd = __GI_socket(2, 1, 0);
    __GI___libc_fcntl(fd, 4, (struct flock *)0x800, v4.sin_port);
    __libc_connect(fd, &v4, 16);
}

```

其中C2域名的DNS A记录IP与PORT的对应关系为N:10。

```

root@debian:~# dig fridgexperts.cc +short
54.248.67.216
54.250.46.94
13.125.38.158
170.187.187.99
45.61.139.116
172.105.241.100
13.232.96.171
159.65.158.148
138.68.10.149
194.53.108.159

```

C2 IP

N:10

```

port_list      dw 4359
               dw 8345
               dw 8234
               dw 8693
               dw 8221
               dw 43745
               dw 7654
               dw 7324
               dw 43231
               dw 1111

```

C2 PORT

当成功和C2建立连接后，Bot与C2必须经过5轮交互，才能真正和C2建立通信。我们使用 `arm` 做为分组字串，产生了下图所示的网络流量：

```

00000000 ee 00 00 11 ff .....
00000000 26 14 2d 4d 58 d2 9e 26 67 98 bc e4 ef 69 b9 04 &.-MX..& g...i..
00000010 e6 d0 73 17 5c 4f 71 33 9f 97 18 f7 31 8d d4 d6 ..s.\Oq3 ....1...
00000020 2f 8a 5c da 57 50 a6 64 d7 98 f5 5d [.\.WP.d ...]
00000005 99 9e 95 f6 32 .....2
0000002C 55 00 00 aa ff U....
0000000A fe 00 03 fe fe .....
0000000F ad ec f8 ...

```

我们来详细介绍下此流量是如何生成的：

Step 1 : Bot--->C2 (定长5字节)

硬编码的 `ee 00 00` 通过tcp/ip checksum方法，计算得到2字节的校验值`0xff11`，将它填到末尾2字节处。

```

def checksum(data):
    s = 0
    n = len(data) % 2
    for i in range(0, len(data)-n, 2):
        s+= ord(data[i]) + (ord(data[i+1]) << 8)
    if n:
        s+= ord(data[i+1])
    while (s >> 16):
        s = (s & 0xFFFF) + (s >> 16)
    s = ~s & 0xffff
    return s

```

Step 2 : C2--->BOT(2次，第一次32字节；第二次12字节)

注意key与nonce由C2端生成，不是固定的。

前32字节为chacha20算法的key

```

26 14 2d 4d 58 d2 9e 26 67 98 bc e4 ef 69 b9 04
e6 d0 73 17 5c 4f 71 33 9f 97 18 f7 31 8d d4 d6

```

后12字节为chacha20算法的nonce

```

2f 8a 5c da 57 50 a6 64 d7 98 f5 5d

```

Step 3: BOT--->C2 (定长5字节)

硬编码的 `55 00 00` 通过checksum，计算得到校验值`0xffaa`，填到末尾2字节，变成 `55 00 00 aa ff`，然后使用chacha20算法加密，轮数为1，得到 `99 9e 95 f6 32`。

Step 4: C2--->BOT(定长5字节)

此时如果收到的5字节的格式为 `0x55`开头，最后2字节为校验值 则说明前面的交互是对的，进入Step 5要求BOT开始发送分组信息。

Step 5 : Bot--->C2(2次，第一次5字节，第二次分组)

- 第一次
硬编码的 `fe 00 00`，第三个字节真为分组长度，变成 `fe 00 03`，计算得到校验值 `0xfefe`，填到尾部得到 `fe 00 03 fe fe`

- 第二次
分组字符串 `arm`，使用chacha20加密，轮数为1，得到 `ad ec f8`

至此BOT成功上线，开始等待执行C2下发的指令，指令码及其含义如下所示：

- 0x69, Heartbeat


```
00000031 69 00 00 96 ff i....
00000012 70 00 00 8f ff p....
00000036 69 00 00 96 ff i....
00000017 70 00 00 8f ff p....
```
- 0xEB, DDoS Attack
- 0xFB, Exit

C2跟踪

我们的僵尸网络跟踪系统数据显示 `Fodcha` 从诞生起就一直有对外发起DDoS攻击，其攻击目标趋势如下：



可以看到，该家族的DDoS行为是非常活跃：

- 攻击最猛烈的时候是 2022-03-01，跟踪到超过130k条指令
- 最近一周，日均指令超过7k，针对100+目标

同时，我们从DNS的角度，也可以清晰的看到该家族的C2域名在 2022-03-19前后做了一次更替，对应前述样本分析部分中 v1 到 v2 的转变。



Netlab按：

v1 到 v2 的转变，是因为v1版本的C2对应的IP被国外某云厂商处置过，因此Fodcha的运营者迫不得已重新上线了v2版本，更新了C2。新C2映射到十几个IP，而且分布在美国、韩国、日本、印度多个国家，同时分散在 Amazon、DediPath、DigitalOcean、Linode等多个平台，充满了“危机意识”。

防范建议

请广大网民及服务/产品厂商强化风险意识，加强安全防范，避免不必要的经济损失，主要建议包括：1、及时修复相关系统漏洞。2、不使用弱密码或默认密码，定期更换密码。

当发现主机感染僵尸木马程序后，立即核实主机受控情况和入侵途径，并对受害主机进行清理。

IoC

样本MD5

0e3ff1a19fcd087138ec85d5dba59715
1b637faa5e424966393928cd6df31849
208e72261e10672caa60070c770644ba
2251cf2ed00229c8804fc91868b3c1cb
2a02e6502db381fa4d4aeb356633af73
2ed0c36ebbeddb65015d01e6244a2846
2fe2deeb66e1a08ea18dab520988d9e4
37adb95cbe4875a9f072ff7f2ee4d4ae
3fc8ae41752c7715f7550dabda0eb3ba
40f53c47d360c1c773338ef5c42332f8
4635112e2dfe5068a4fe1ebb1c5c8771
525670acfd097fa0762262d9298c3b3b
54e4334baa01289fa4ee966a806ef7f1
5567bebd550f26f0a6df17b95507ca6d
5bdb128072c02f52153eaeaa6899a5b1
6244e9da30a69997cf2e61d8391976d9
65dd4b23518cba77caab3e8170af8001
6788598e9c37d79fd02b7c570141ddcf
760b2c21c40e33599b0a10cf0958cfd4
792fdd3b9f0360b2bbee5864845c324c
7a6ebf1567de7e432f09f53ad14d7bc5
9413d6d7b875f071314e8acae2f7e390
954879959743a7c63784d1204efc7ed3
977b4f1a153e7943c4db6e5a3bf40345
9defda7768d2d806b06775c5768428c4
9dfa80650f974dffe2bda3ff8495b394
a996e86b511037713a1be09ee7af7490
b11d8e45f7888ce85a67f98ed7f2cd89
b1776a09d5490702c12d85ab6c6186cd
b774ad07f0384c61f96a7897e87f96c0
c99db0e8c3ecab4dd7f13f3946374720
c9cbf28561272c705c5a6b44897757ca
cbdb65e4765fbd7bcae93b393698724c
d9c240dbed6dfc584a20246e8a79bdae
e372e5ca89dbb7b5c1f9f58fe68a8fc7
ebf81131188e3454fe066380fa469d22
fe58b08ea78f3e6b1f59e5fe40447b11

下载链接

<http://139.177.195.192/bins/arm>
<http://139.177.195.192/bins/arm5>
<http://139.177.195.192/bins/arm7>
<http://139.177.195.192/bins/mips>
<http://139.177.195.192/bins/realtek.mips>
<http://139.177.195.192/blah>
<http://139.177.195.192/linnn>
<http://139.177.195.192/skidrt>
<http://139.177.195.192/z.sh>
<http://162.33.179.171/bins/arm>
<http://162.33.179.171/bins/arm7>
<http://162.33.179.171/bins/mpsl>
<http://162.33.179.171/bins/realtek.mips>
<http://162.33.179.171/bins/realtek.mpsl>
<http://162.33.179.171/blah>
<http://162.33.179.171/k.sh>
<http://162.33.179.171/linnn>
<http://162.33.179.171/z.sh>
<http://206.188.197.104/bins/arm7>
<http://206.188.197.104/bins/realtek.mips>
<http://206.188.197.104/skidrt>
<http://31.214.245.253/bins/arm>
<http://31.214.245.253/bins/arm7>
<http://31.214.245.253/bins/mips>
<http://31.214.245.253/bins/mpsl>
<http://31.214.245.253/bins/x86>
<http://31.214.245.253/k.sh>
<http://31.214.245.253/kk.sh>

C2域名

folded.in
fridgexperts.cc