

Botnet

CSC 348-648



WAKE FOREST
UNIVERSITY

Department of Computer Science

Spring 2013

Botnets

- Network of autonomous programs that can act on instructions
 - Typically a large group of remotely controlled *zombie* systems
 - Machine owners are not aware they have been compromised
 - Controlled and upgraded via IRC or P2P

How is this similar/different than worms?

- Used as a platform for various attacks
 - Distributed denial of service
 - Spam and click fraud
 - Launching pad for new exploits/worms
- “ $\frac{1}{4}$ of hosts ... are members of a botnet” - Vint Cerf

History

- Eggdrop (1993), early IRC bot
What was it used for?
- DDoS bots in the late 1990s, Trin00, TFN, and Stacheldrucht
- Remote Administration Trojans (RATs) in late 1990s
 - Variants of Back Orifice and NetBus, SubSeven, Bionet
 - Related to rootkits
- Modern bots, Agobot (PhatBot, SDBot) and GTBot
 - Active spreading, multiple propagation vectors (combines characteristics of a worm and a Trojan)
 - Remotely controlled (typically via IRC)
 - Many variations of the same code

Tob Sepyt

- Benign bots
 - Help collect information, monitor systems, or environments where a continual, interactive presence is required
 - For example...*
- Gray-area bots
 - Blogbots (for example wikipedia)
 - xdcc and fserve for IRC
 - Trainer bots (MMORPGs)
What's a trainer bot?
- Malicious bots
 - Key characteristics: process forking, with network and file access, and propagation potential

Malicious Botnet Families

- **Agobot**, most sophisticated, 20,000 lines of C/C++ code
 - IRC based command and control
 - DoS, sniffers, key loggers, polymorphic obfuscation
- **SDbot**, simple design with 2,000 lines of C code
 - IRC based command and control
 - Non-malicious base, can be easily expanded
- **SpyBot**, approximately 3,000 lines of code
 - *Possibly evolved from SDbot?*
 - Includes scanning and DDoS abilities
- **GTbot**, functions based on mIRC scripting
 - Collection of mIRC scripts from cracked version of mIRC

Usage

Capability	Ago	DSNX	evil	G-SyS	SD	Spy
Create port redirect	X	X		X	X	X
Other proxy	X					
Download from web	X	X		X	X	X
DNS resolution	X			X	X	
smoke yo stuff	X					
UDP/ping floods	X		X	X	X	
Other DDoS	X			X		X
Scan/spread	X	X		X	X	X
Messin' wit yo gf/bf						X
Spam	X					
Visit URL	X			X	X	

Example of Bot Recruitment

- “The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets,” Cooke, Jahanian, McPherson, 2005
 - Interested in botnet creation and management, but needed to join
- Windows 2000/XP honeypot created
 - Rate limit traffic 12 KBps
 - Local traffic not allowed
 - All traffic logged
- 12 experiments over 1 month, each 12-72 hours
 - Recruited into 15 botnets
 - Bots used DCOM/RPC and LSASS
 - *Only 2 worm infections during this time...*

Administration Tools

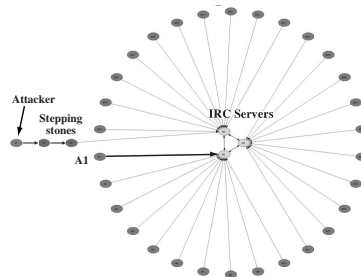
- Legitimate tools are often used by botnets to recruit and communicate
 - Citrix MetaFrame, WinVNC, and PC Anywhere, allows remote control over the machine (*found by port scan, for example port 1494 for Citrix*)
 - Bad installations, crackable password authentication (*for example, breaking into a bank's IBM AS/400 transfer server*)
- Semi-legitimate tools
 - Back Orifice, NetBus
 - Can hide installation/operation, log keystrokes, etc.
 - Considered malicious by anti-virus software

Botnet Steps

1. Exploit a vulnerability to execute a short program
 - Buffer overflows, email viruses, etc...
2. Shellcode downloads and installs actual bot
3. Bot disables firewall and antivirus software
4. Bot *typically* locates IRC server, connects, joins channel
 - Typically need DNS to find out server's IP address
So why rely on DNS?
- Authentication password often stored in bot binary
5. Botmaster issues authenticated commands

IRC

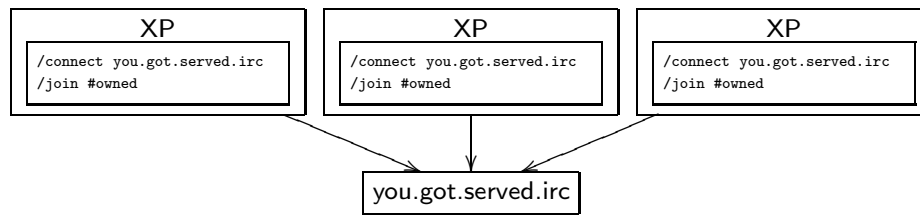
- Internet Relay Chat (IRC) is an open protocol [RFC 1495]
 - A form of Internet chat or conferencing
 - Designed for group communication using *channels*



- Uses a series of TCP clients and servers to communicate
 - Users join a channel (a group of on-line users)
 - Message sent on a channel is received by all on the channel

Why use IRC for botnet control?

Joining the IRC Channel



```
Terminal

(12:59:27pm) -- A9-pcgbdv (A9-pcgbdv@140.134.36.124) has joined (#owned) Users : 1646

(12:59:27pm) (@Attacker) .ddos.synflood 216.209.82.62

(12:59:27pm) -- A6-bpxufrd (A6-bpxufrd@wp95-81.introweb.nl) has joined (#owned) Users : 1647

(12:59:27pm) -- A9-nzmpah (A9-nzmpah@140.122.200.221) has left IRC (Connection reset by peer)

(12:59:28pm) (@Attacker) .scan.enable DCOM

(12:59:28pm) -- A9-tzrkeasv (A9-tzrkeasv@220.89.66.93) has joined (#owned) Users : 1650
```

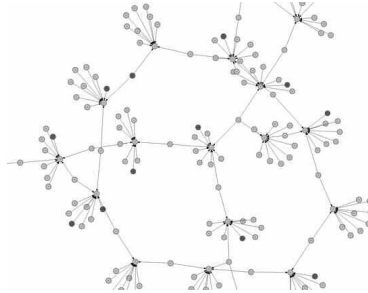
IRC and Bots

- Bots must act like a *normal* IRC client
 - Respond to several IRC commands (NICK, JOIN, PONG, ...)
- Some additional *bonus* commands typically implemented

```
void help(int sock, char *sender, int argc, char **argv) {
    if (mfork(sender) != 0) return;
    Send(sock, "NOTICE_%s: TSUNAMI<target><secs>=Packeteer\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: PAN<target><port><secs>=Synflooder\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: UDP<target><port><secs>=Audpflooder\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: UNKNOWN<target><secs>=Udpflooder\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: NICK<nick>=Changes_nick\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: SERVER<server>=Changes_servers\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: GETSPOOFS=Gets_spoofing\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: SPOOFS<subnet>=Changes_spoofing\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: DISABLE=Disables_packeting\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: ENABLE=Enables_packeting\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: KILL=Kills_client\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: GET<http_address><save_as>=Downloads_file\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: VERSION=Requests_version\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: KILLALL=Kills_packeting\n", sender); sleep(2);
    Send(sock, "NOTICE_%s: HELP=Displays_this\n", sender);
    exit(0);
}
```

Other Communication Methods

- There has been a progression to *different* communication channels
 - IRC traffic can be blocked (by a company or university)
 - Communication IRC patterns can be used to detect botnets



- Peer-to-Peer (P2P) provides a robust communication channel
 - No need for central hosts/network for communication
 - Difficult to trace activity, since no network of servers
 - *AgoBot provides a basic P2P system, but not widely used...*

Moving Bots

- IRC operators are aware of bots and the disruption

"... they are constantly on the lookout for thousands of users showing up in a short period of time in a given channel or moving from channel to channel en masse..."
- Bot owners use several techniques to *move* bots
 - Use of dynamic DNS entries, or short TTLs in DNS (fast flux)
 - Have bots switch IRC channels (channel hopping)
 - Having all bots switch IRC servers (server hopping)

Nice spread... (343/643)
 - Use of proxies for standard IRC server ports
 - Avoid standard IRC networks, set up customized botnet-tuned IRC server programs on compromised hosts (*rogue IRC servers*)

Fast Flux

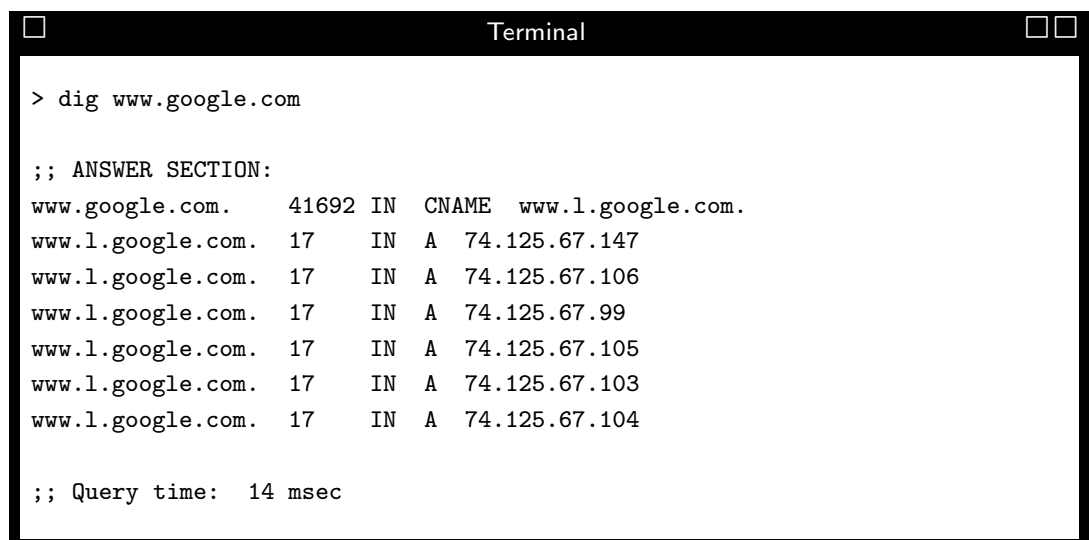
"The goal of fast-flux is for a fully qualified domain name (such as www.example.com) to have multiple (hundreds or even thousands) IP addresses assigned to it."

- Multiple nodes registering and deregistering their address in DNS
 - Combine round robin DNS with a short TTL
- For example consider a browser connecting to `www.pluf.com`
 - If fast flux then connecting to `www.pluf.com` at different times will resolve to different computers
- For malware DNS record will point to proxy
 - Proxy used to issue commands, updates, etc...

So what is the advantage to the attacker?

Are there legitimate uses for flux?

- Consider server load balancing (although there are better ways)



```
Terminal
> dig www.google.com

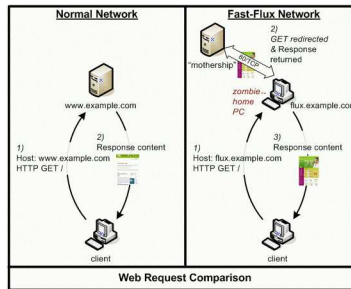
;; ANSWER SECTION:
www.google.com.      41692 IN  CNAME  www.l.google.com.
www.l.google.com.    17     IN  A      74.125.67.147
www.l.google.com.    17     IN  A      74.125.67.106
www.l.google.com.    17     IN  A      74.125.67.99
www.l.google.com.    17     IN  A      74.125.67.105
www.l.google.com.    17     IN  A      74.125.67.103
www.l.google.com.    17     IN  A      74.125.67.104

;; Query time: 14 msec
```

– Second column is the refresh rate in seconds

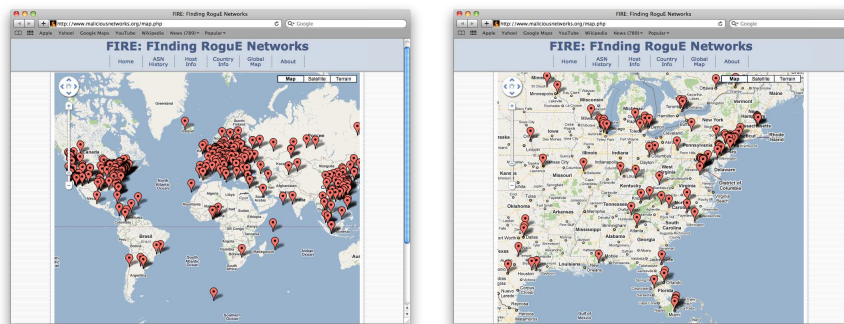
- Digging Canadian Pharmacy is more funner

Example Flux



- Normal DNS, browser connects to `www.example.com`
 - DNS returns the IP address to `www.example.com`
- Single flux, browser connects to `flux.example.com`
 - Current DNS points to a zombie PC
 - zombie PC is a proxy for mothership

Rogue Networks



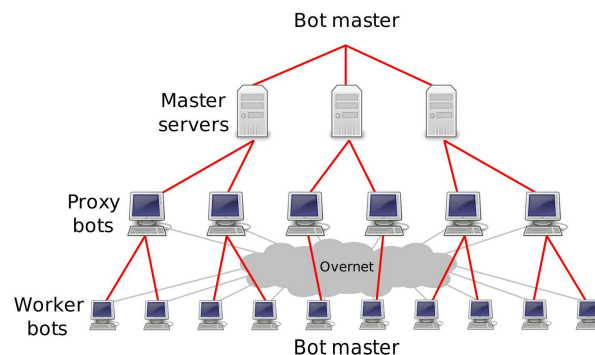
- Botnets (and other malware) often rely on *safe* networks
 - ISP that do not respond quickly to attack complaints
 - Obtain address space and use fast flux to rotate servers

"flexibility consists of being able to change the IP addresses of the nameservers and allowing longevity since the registrars are lax in taking down the sites despite complaints ... The domains are registered frequently, and automatically, which is key to their resistance. "

Botnet + Spam = Money

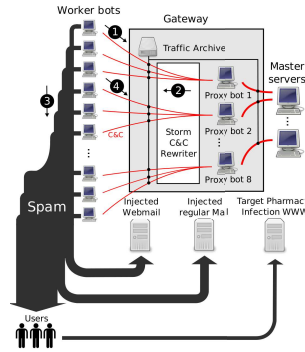
- *What is the purpose of a botnet? Attacks? Money? Attack Money?*
 - Often botnets exist for profit, for example generating spam
 - *But what is the conversion rate of spam?*
- “Spamalytics: An Empirical Analysis of Spam Marketing Conversion”
 - 2008 paper, interested in probability of a sale resulting from spam
 - *Infiltrated* existing Storm botnet and analyzed three campaigns
- Storm is a hierarchical, P2P botnet that propagates via spam
 - Victims receive email with a URL to a botnet trojan
 - Bots communicate with UDP-based Overnet, a Distributed Hash Table (DHT) P2P, and a custom TCP protocol for C&C

Storm Hierarchy



- **Worker nodes** request work, upon receiving orders send spam
- **Proxy bots** are conduits between workers and servers
 - So why have proxies?*
- **Master Servers** give commands to workers and receive status reports
 - Normally located in *safe* networks

Spam Campaign



1. Worker finds a Proxy and sends an update request to a Master Server
 - At boot, worker will attempt to maintain 20 peers on Overnet
 - DHT keys change over time, but hosts remain in sync
2. Server forwards a spam workload
 - Contains spam template, email addresses, and dictionaries
 - Templates are written in a custom macro language

- Macros insert words from dictionaries and generate a *unique* email that appears to be from a valid Mail Transfer Agent (MTA)

Why so much work for generating an email?

3. Bot generates unique email per address in list and sends via SMTP
4. When a bot has exhausted the list, sends a report back to the Proxy

Storm Infiltration

"Some have estimated that by September 2007 the Storm botnet was running on anywhere from 1 million to 50 million computer systems. Other sources have placed the size of the botnet to be around 250,000 to 1 million compromised systems." (wiki)

"a vast collection of compromised computers once responsible for sending an estimated 20 percent of all spam." (Krebs)

- Created 8 unmodified Storm proxy bots using VMs
 - Proxy bots communicated via a controlled gateway
 - Passively observed spam related commands and data
 - Actively changed some individual elements of the communications
- Parsed and rewrote C&C messages
 - Rewrote the spam template, dictionaries and referenced URLs

Any legal or ethical issues here?

Measurements

- Spam delivery (*effectiveness of spam filters*)
 - Created email accounts with typical webmail providers
 - Interested in percentage of emails that pass the spam filter
- Click-through and conversion (*percentage that visit a spam URL*)
 - Created e-pharmacy site (spam) and e-postcard site (propagation)
 - Sites mimic those associated in the campaigns, but checkout resulted in a 404 error (no harm done to the potential victim)
 - Assumed a purchase attempt was a conversion

What about credit card information?

- Also tried to separate crawler traffic from victim traffic
 - Blacklisted hosts that accessed robots.txt, hosts that did not load images, and malformed requests (possibly an exploit attempt)

Experiment Ethics

"We have been careful to design experiments that we believe are both consistent with current U.S. legal doctrine and are fundamentally ethical as well. ...we believe the ethical basis for our work is far easier to explain: we strictly reduce harm."

- Instrumented proxy bots do not create new harm
- Instrumented proxies are passive
- C&C are modified, but modifications reduce harm

Experimental Findings

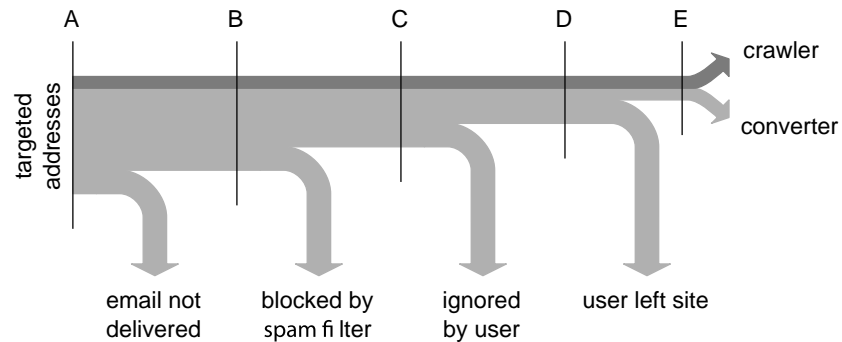
- Number of workers and emails per campaign

Campaign	Dates	Workers	E-mails
Pharmacy	March 21 - April 15	31,348	347,590,389
Postcard	March 9 - April 15	17,639	83,665,479
April Fool	March 31 - April 2	3,678	38,651,124
		Total	469,906,992

- 10 most targeted email address domains (over all campaigns)

hotmail.com	8.47%	sbcglobal.net	0.93%
yahoo.com	5.05%	mail.ru	0.86%
gmail.com	3.17%	shaw.ca	0.61%
aol.com	2.15%	wanadoo.fr	0.61%
yahoo.co.in	1.13%	msn.com	0.58%
Total		23.8%	

- Spam conversion pipeline



Stage	Pharmacy	Postcard	April Fool
<i>A</i> Spam Targets	347,590,389 (100%)	83,655,479 (100%)	40,135,487 (100 %)
<i>B</i> MTA Delivery (est.)	82,700,000 (23.8%)	21,100,000 (25.2%)	10,100,000 (25.2%)
<i>C</i> Inbox Delivery			
<i>D</i> User Site Visits	10,522 (0.00303%)	3,827 (0.00457%)	2,721 (0.00680%)
<i>E</i> User Conversions	28 (0.0000081%)	316 (0.000378%)	225 (0.000561%)

- Percentage of messages delivered to a user's inbox for web accounts

Spam Filter	Pharmacy	Postcard	April Fool
Gmail	0.00683%	0.00176%	0.00226%
Yahoo	0.00173%	0.000542%	none
Hotmail	none	none	none
Barracuda	0.131%	N/A	0.00826%

- In other words...

- 1 in 12,500,000 pharmacy spams lead to a purchase
- 1 in 265,000 greeting card spams lead to an infected machine
- 1 in 178,000 April Fool's Day spams lead to an infected machine
- 1 in 10 people visiting an infection website downloaded the executable and ran it

Spam Profit

- 26 campaign days (over 35M emails) resulted in 28 sales
 - This is a conversion rate of 0.00001%
 - The product price was \$100, so the revenue was \$2,731.88
- Assuming the paper only infiltrated 1.5% of the botnet
 - Total botnet revenue would be \$7,000
- If you extrapolate botnet growth, the revenue is \$3.5M/year
 - Including estimated operating costs, profit is \$1.75M/year

Rustock Botnet Takedown?

"The global volume of junk e-mail sent worldwide took a massive nosedive today [3/11/2011] following what appears to be a coordinated takedown of the Rustock botnet, one of the worlds most active spam-generating machines." (Krebs)

- Rustock botnet is another spam generator
 - Rootkits host computer
 - C&C is sent as HTTP based forum posts with encrypted content
 - Uses a *DNS* filter to disguise IP addresses of controllers

- The takedown

"This looks like a widespread campaign to have either these [Internet addresses] null-routed or the abuse contacts at various ISPs have shut them down uniformly, Stewart said. It looks to me like someone has gone and methodically tracked these [addresses] and had them taken out one way or another." (SecureWorks)

Will it return?

Another Profitable Botnet

- Bot herder called 0x80 (Washington Post: Invasion of the Computer Snatchers)
 - Owns and manages over 13,000 bots in more than 20 countries
 - Earns approximately \$6,800/month, works 2 minutes/day
- *What exactly does he do?*
 - Infected PCs download adware then search for new victims
 - Adware displays ads and records victim's online habits
 - Bot collects password, e-mail, social, credit information

A New Market

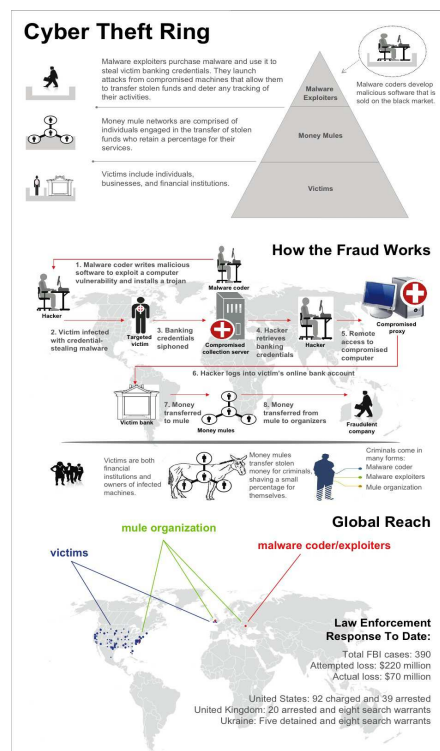


current
prices

Rank	Previous	Service	Current %	Previous %	Price
1	2	bank accounts	22	21	\$10 - \$1000
2	1	credit cards	13	22	\$0.40 - \$20
3	7	full identity	9	6	\$1 - \$15
4	–	auction account	7	–	\$2.50/week - \$50/week
5	8	scam	7	6	\$1 - \$10
6	4	email server	6	8	\$4 - \$30
7	5	email addresses	5	6	\$0.83/MB - \$10/MB
8	3	email passwords	5	8	\$4 - \$30
9	–	drop (request or offer)	5	–	10 - 50%
10	6	proxies	5	6	\$1.50 - \$30

Botnet Software for Sale?

- ZeusS is a trojan that steals banking info using keyloggers
 - A typical user may do a lot of typing before entering any financial info, so newest version includes *no-sheir* option
 - Typically spread via drive-bys and facebook phishing
 - Software is for sale, \$700 to \$3,000, depending on version
- Hackers used ZeusS to infected computers around the world
 - Secretly captured passwords, account numbers, and other data used to log into online banking accounts
 - Unauthorized money transfers, often routing funds to other accounts controlled by a network of *money mules*
 - Members of the theft ring managed to steal \$70 million before busted by FBI

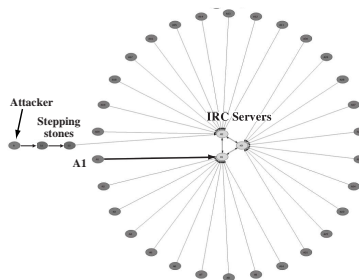


Detecting and Preventing

- Many bots use IRC for command and control
 - Detect IRC commands, packet payloads, IRC behavior
 - *Easy to change communication characteristics...*
- Observe and correlate commands and behavior?
 - Bots typically communication, propagate, and attack
 - Each command generates certain traffic patterns...
- Observe patterns of DNS requests?
 - “... botnets tend to use subdomains; legitimate directories use subdirectories ...”
 - For example DNS request for botnet1.wfu.edu as compared to www.wfu.edu/home/nirre/pluf/

Detecting Botnet Communications

- It may be difficult to detect botnet communications
 - Payloads may be encrypted or disguised as *normal* conversations
- *What about the pattern of communications?*

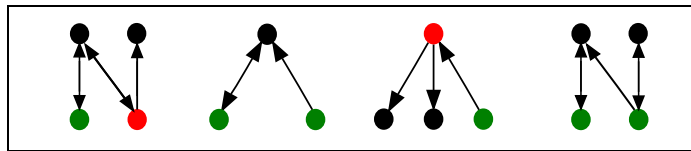


- Certain communication patterns may exist

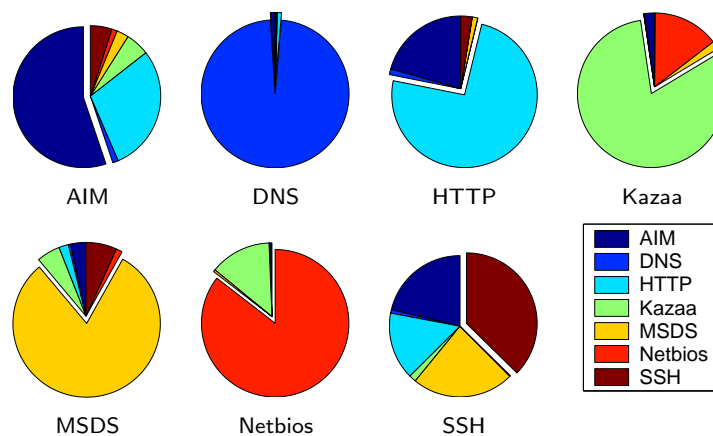
Is it possible to infer application based on communication patterns?

Interaction Patterns

- Can we determine what users are doing based on interactions?
 - Do not care about packet contents, just the pattern of interactions
- Motifs (interaction patterns) have been applied elsewhere
 - Gene regulation, neural networks, ecosystem food webs, electronic circuits (forward logic chips, digital fractional multipliers), and World Wide Web
- Certain motifs can be linked to specific functions



Motif Profile Results



- Results very good, 85% accuracy for most applications
- Next question: *determine the associations...*