Botnet

PHAM VAN HAU (PVHAU@HCMIU.EDU.VN)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING-
INTERNATIONAL UNIVERSITY

Botnets

Bots: Autonomous programs performing tasks

Plenty of “benign” bots

e.g. Google bot

Botnets: A **botnet** is a collection of compromised computers controlled by their attacker

Cost of worm attacks

- Morris worm, 1988
  - Infected approximately 6,000 machines
    - 10% of computers connected to the Internet
  - cost ~ $10 million in downtime and cleanup
- Code Red worm, July 16 2001
  - Direct descendant of Morris' worm
  - Infected more than 500,000 servers
    - Programmed to go into infinite sleep mode July 28
  - Caused ~ $2.6 Billion in damages,
- Love Bug worm: $8.75 billion

  - Statistics: Computer Economics Inc., Carlsbad, California

Rise of Botnets

Motivation for malicious activity is shifting

Primary motivation has changed from vandalism and demonstration of programming skills to for-profit activities

- Identity theft, extortion
- Backed by organized crime

Botnets Today

Botnets can be extremely large, with reports of botnets of over 100,000 systems

Average size appears to be dropping

Total estimated number of systems used in botnets is in the millions

Botnet History: How we got here

Early 1990s: IRC bots

eggdrop: automated management of IRC channels

1999-2000: DDoS tools

Trinoo, TFN2k, Stacheldraht

1998-2000: Trojans

BackOrifice, BackOrifice2k, SubSeven

2001- : Worms

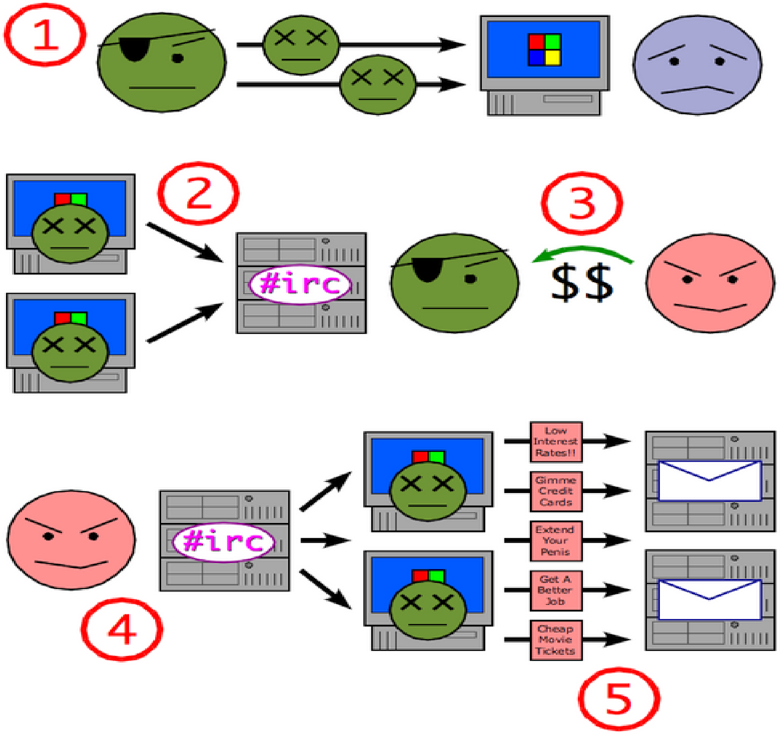
Code Red, Blaster, Sasser

← **Fast spreading capabilities pose big threat**

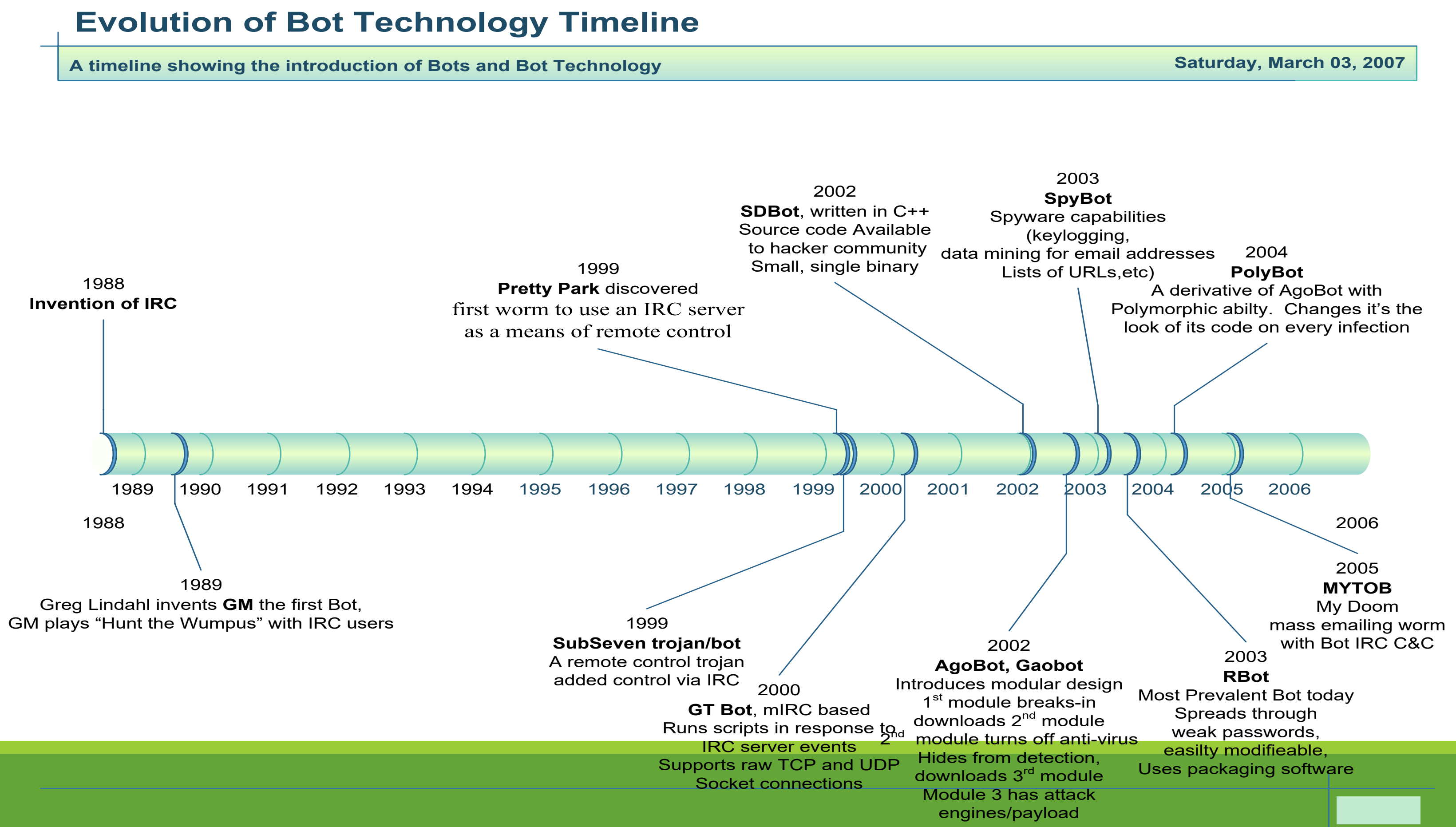
Put these pieces together and add a controller…

Putting it together

- Miscreant (botherd) launches worm, virus, or other mechanism to infect Windows machine.
- et controller via IRC.
- ant for use of botnet.
- pam emails.



Evolution of Bot Technology



Evolution of Bot Technology Timeline

A timeline showing the introduction of Bots and Bot Technology

Saturday, March 03, 2007

1988
Invention of IRC

1989
Greg Lindahl invents **GM** the first Bot,
GM plays “Hunt the Wumpus” with IRC users

1999
Pretty Park discovered
first worm to use an IRC server
as a means of remote control

1999
SubSeven trojan/bot
A remote control trojan
added control via IRC

2000
GT Bot, mIRC based
Runs scripts in response to
IRC server events
Supports raw TCP and UDP
Socket connections

2002
SDBot, written in C++
Source code Available
to hacker community
Small, single binary

2002
AgoBot, Gaobot
Introduces modular design
1st module breaks-in
downloads 2nd module
2nd module turns off anti-virus
Hides from detection,
downloads 3rd module
Module 3 has attack
engines/payload

2003
SpyBot
Spyware capabilities
(keylogging,
data mining for email addresses
Lists of URLs,etc)

2003
RBot
Most Prevalent Bot today
Spreads through
weak passwords,
easilty modifieable,
Uses packaging software

2004
PolyBot
A derivative of AgoBot with
Polymorphic abilty. Changes it’s the
look of its code on every infection

2005
MYTOB
My Doom
mass emailing worm
with Bot IRC C&C

Important Aspects of Botnet

Command and Control Channel

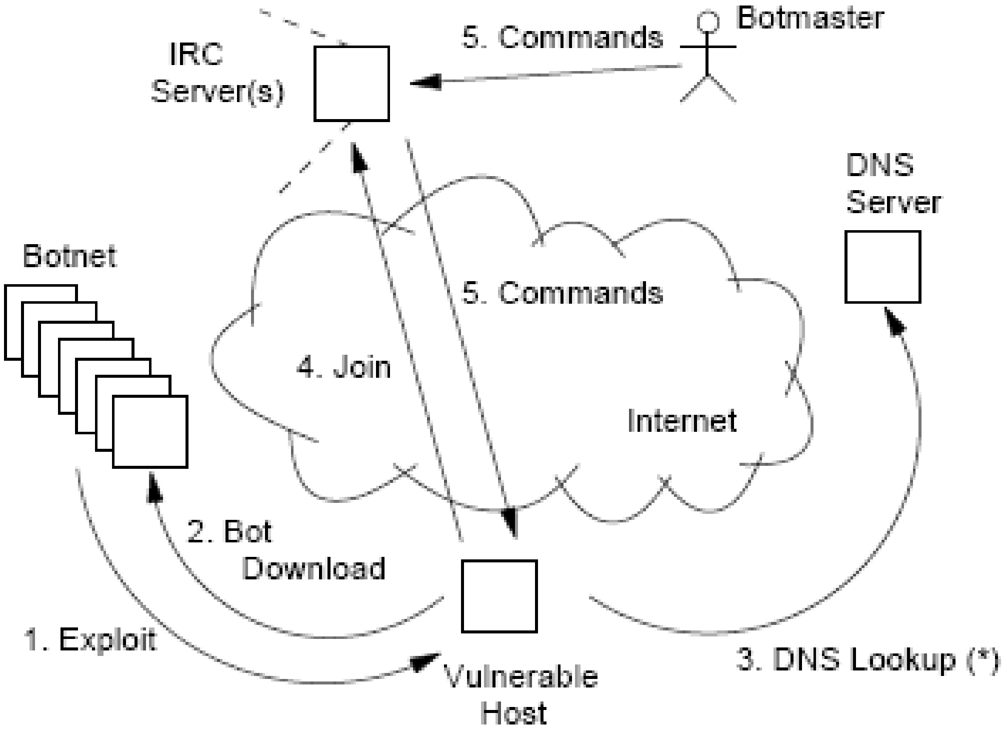
Propagation

Underground activities

Command and Control Technologies

- IRC
- HTTP
- P2P

IRC Command and Control Channel



IRC Command Example

.advscan lsass_445 150 3 9999 –r –s

.advscan – botnet command to scan for vulnerable systems

lsass_445 – attempt to exploit vulnerable hosts using VU#753212

150 – the number of concurrent threads

3 – the number of seconds to delay between scans

9999 – specified amount of time to perform the scanning activity

-r – the IP addresses it attempts to scan should be generated randomly

-s – the scan should be silent and not report its findings back in the channel

HTTP Command and Control Channel

Compromised machines call a PHP script on a specific web server

Web server gets information about compromised machines

Compromised machines connect to the website from time to time to get the commands

P2P C&C

Compromised machines get the commands by using P2P protocols such as Gnutella as in Phatbot

Botnet Architecture

Centralized

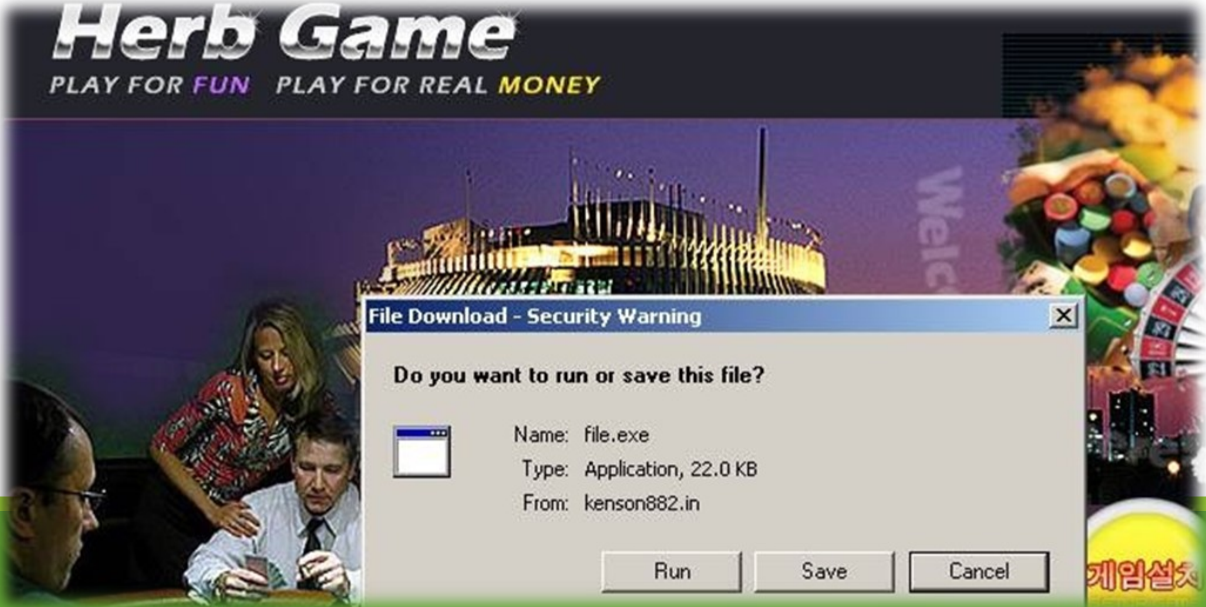
Distributed

Social engineering: download from Internet

Black Energy backdoor does not exploit any vulnerability in the OS system.

The victim needs to execute the malware in order to be infected.

The infection is typically triggered by the victim downloading and executing the backdoor from fake online games web sites.



File Download - Security Warning

Do you want to run or save this file?

Name: file.exe
Type: Application, 22.0 KB
From: kenson882.in

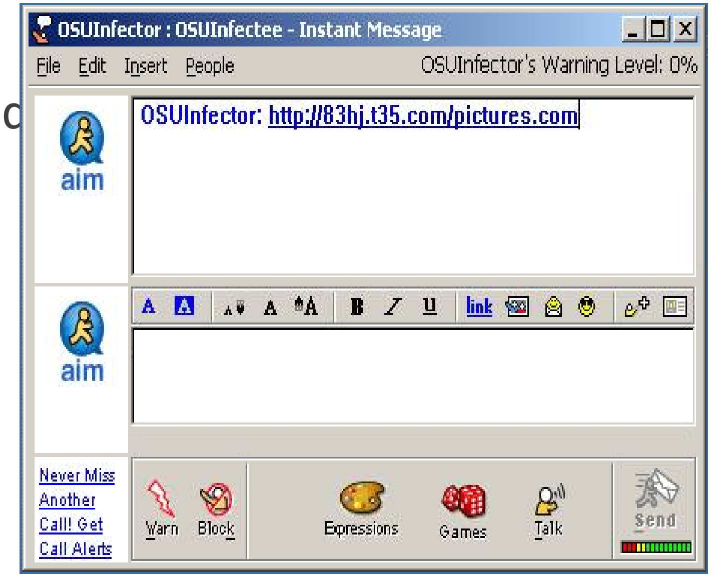
Social Engineering: IM

Bots frequently spread through AOL IM

A bot-infected computer is told to spread through AOL IM

It contacts all of the logged in buddies and sends them a link to a malicious web site

People get a link from a friend, click on it, and asked


OSUInfector : OSUInfectee - Instant Message
File
Edit
Insert
People
OSUInfector's Warning Level: 0%
aim
aim
link
Never Miss Another Call! Get Call Alerts
Warn
Block
Expressions
Games
Talk
Send

OSUInfector: http://83hj.t35.com/pictures.com

P2P Download

Viruses copy itself into shared folders under camouflaged names to lure download/execution

E.g. Swen, Fizzer, Lirva, Benjamin, KwBot, Bodiru, etc.

Kazaaand eDonkeyare popular targets

Client Side Attack

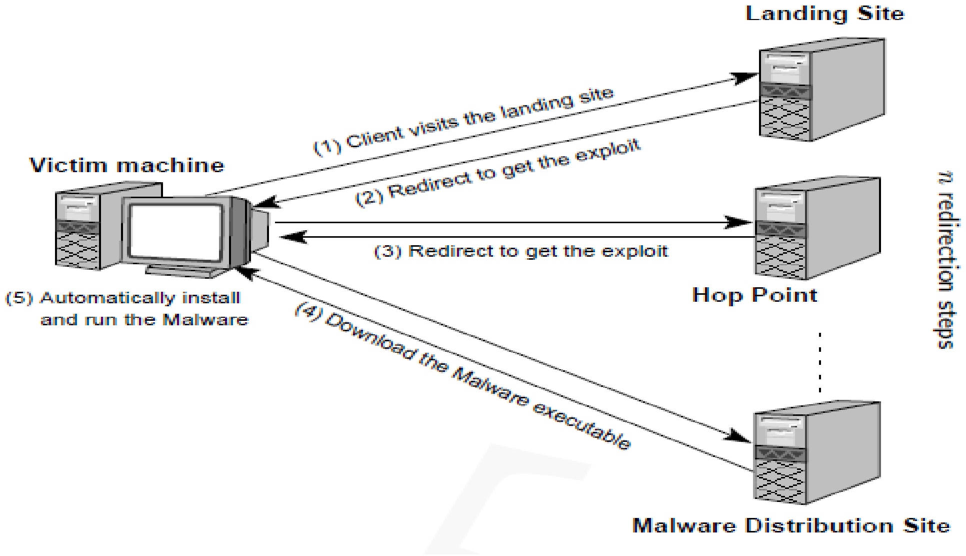
Cross Site Scripting-XSS

drive-by downloads

Mass-mailing virus

drive-by downloads

Approximately 1.3% of the incoming search queries to Google's search engine returned at least one URL labeled as malicious in the results page


Landing Site
(1) Client visits the landing site
(2) Redirect to get the exploit
Victim machine
(3) Redirect to get the exploit
Hop Point
n redirection steps
(5) Automatically install and run the Malware
(4) Download the Malware executable
Malware Distribution Site

Server Side Attack

Remote attack on services located in server machines

Witty worm, Slammer worm, Codered,…

Underground Activities

- D-DOS
- Extortion
- Identity theft
- Spam
- Phising
- Click fraud
- malware distribution

Extortion

We’ve encrypted your files.

Pay me for the key to decrypt them.

We’re DDoSing your website.

Pay me to stop.

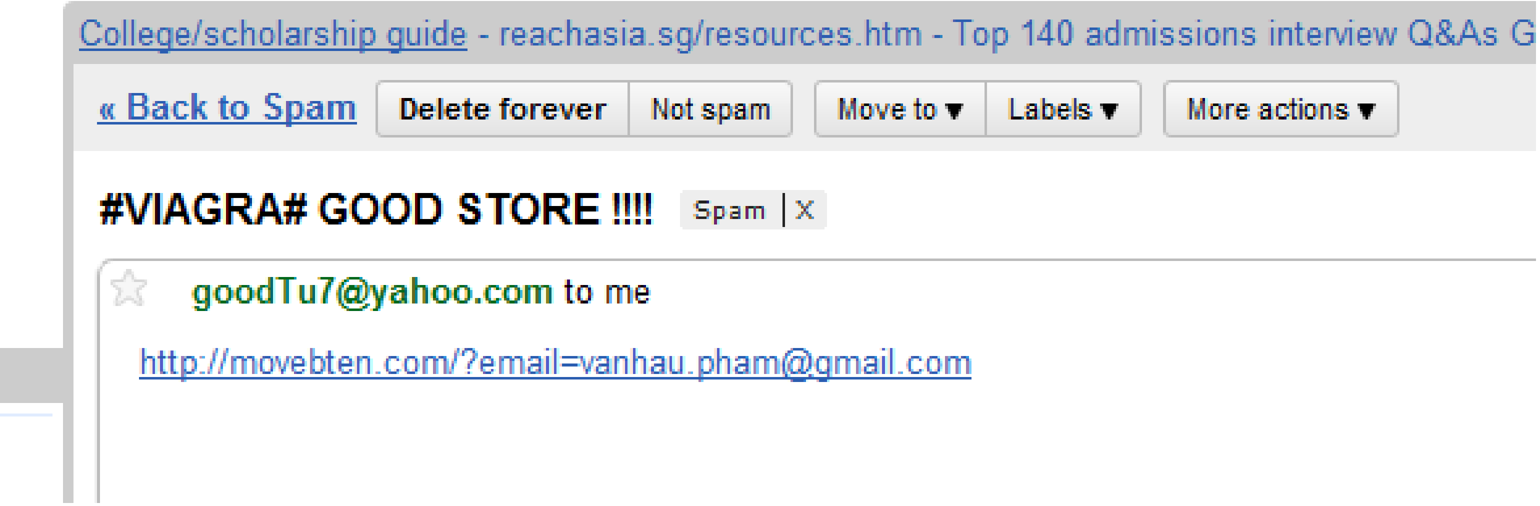
Pay me not to start.

In 2004, botnets attacked dozens of online gambling sites. The bookmakers were told to pay between $10,000 and $50,000 to get their sites back online. (Wired, Nov 2006)

DDos Attacks

In 2000 several famous websites (CCN.com, eBay, Yahoo!) were under DDoS attacks

Spam



College/scholarship guide - reachasia.sg/resources.htm - Top 140 admissions interview Q&As G

#VIAGRA# GOOD STORE !!!!

goodTu7@yahoo.com to me

http://movebten.com/?email=vanhau.pham@gmail.com

Question: **What does spam have to do with botnets?**

Phishing

Social-engineering schemes

Spoofed emails direct users to counterfeit web sites

Trick rec

> “Phishing attacks use both **social engineering** and **technical subterfuge** to steal consumers' personal identity data and financial account credentials.” -- Anti-spam working group

Anti-Ph

15,820 phishing e-mail messages 4367 unique phishing sites identified.

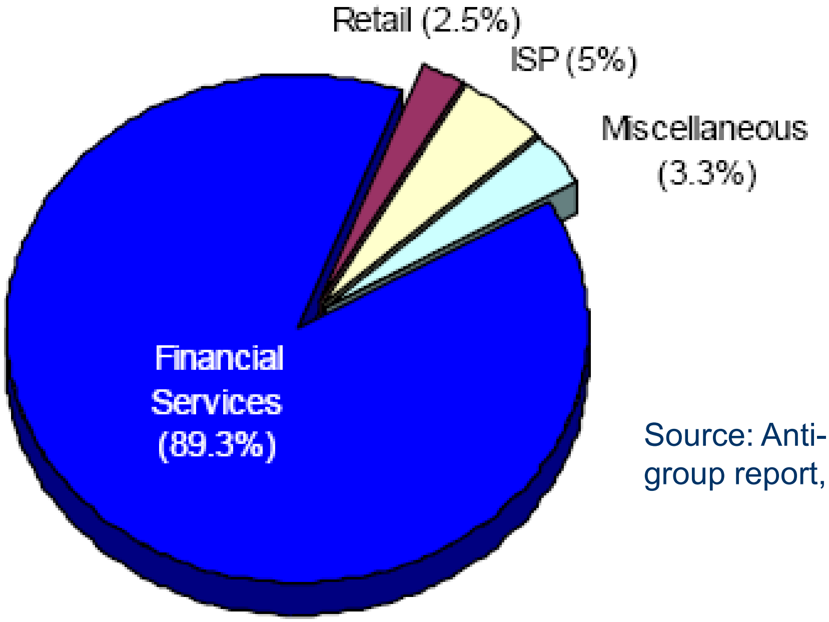
96 brand names were hijacked.

Average time a site stayed on-line was 5.5 days.

Question: What does phishing have to do with botnets?

Which web sites are being phished?

Financial services by far the most targeted sites



Source: Anti-phishing working group report, Dec. 2005

Click Fraud

Pay-per-click advertising

Publishers display links from **advertisers**

Advertising networks act as middlemen

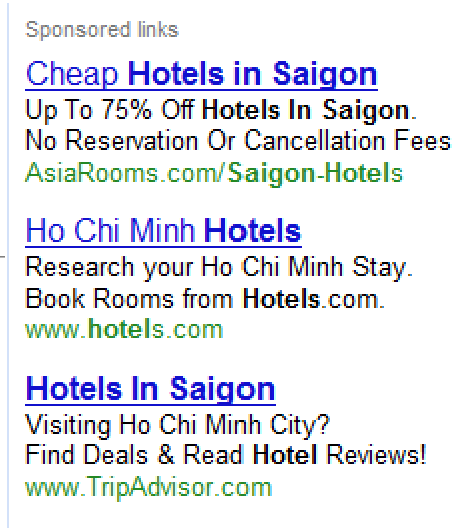
- Sometimes the same as publishers (*e.g.,* Google)

Click fraud: botnets used to click on pay-per-click ads

Motivation

Competition between advertisers

Revenue generation by bogus content provider



Sponsored links

Cheap **Hotels in Saigon**
Up To 75% Off **Hotels In Saigon**.
No Reservation Or Cancellation Fees
AsiaRooms.com/**Saigon-Hotels**

Ho Chi Minh **Hotels**
Research your Ho Chi Minh Stay.
Book Rooms from **Hotels**.com.
www.**hotels**.com

Hotels In Saigon
Visiting Ho Chi Minh City?
Find Deals & Read **Hotel** Reviews!
www.TripAdvisor.com

Case Study

Objectives

Highlight the richness and diversity of bot codebases

Identify commonalities between codebases

Consider how knowledge of these botnet mechanisms can lead to development of more effective defense mechanisms

Case Study

Attributes of bots to analyze

Architecture

Botnet Control Mechanisms

Host Control Mechanisms

Propagation Mechanisms

Target Exploits and Attack Mechanisms

Malware Delivery Mechanisms

Case Study

Four bot codebases

Agobot 4.0 pre-release

SDBot 05b

SpyBot 1.4

GT Bot with DCOM

Agobot

AKA Gaobot, Phatbot

First referenced in October, 2002

Most sophisticated of the four codebases

Typically around 20,000 lines of C/C++

Monolithic architecture

Adheres to structured design and software engineering principles

Modular, standard data structures, code documentation

Exhibits creativity in design

Agobot

Components

IRC-based command and control mechanism

Large collection of target exploits

Ability to launch different kinds of DoS attacks

Modules for shell encodings and limited polymorphism

Mechanisms to frustrate disassembly by well known tools

Agobot

Components

Ability to harvest local host for sensitive information, such as Paypal passwords and AOL keys through traffic sniffing, key logging or searching registry entries

Mechanisms to defend and fortify compromised systems

Over 580 variants

SDBot

First referenced in October, 2002

Hundreds of variants

Fairly simple compared to Agobot

Slightly over 2,000 lines of C

Main source tree does not contain any overtly malicious code modules

Published under GPL

Primarily provides a utilitarian IRC-based command and control system

SDBot

- Easy to extend
- Large number of patches that provide more sophisticated malicious capabilities and diffuse responsibility
  - Scanning
  - DoS attacks
  - Sniffers
  - Information harvesting
  - Encryption routines
- Over 80 patches

SpyBot

First referenced in April, 2003

Hundreds of variants

Fairly compact, around 3,000 lines of C

Shares much of SDBot’s command and control engine

No explicit attempt to diffuse accountability

SpyBot

Capabilities

- NetBIOS, Kuang, Netdevil and KaZaa exploits
- Scanning capabilities
- Modules for launching flooding attacks

Efficient

Does not exhibit modularity or breadth of capabilities of Agobot

GT Bot

AKA Global Threat Bot, Aristotles

First referenced in April, 1998

Over 100 variants

Simple design

Limited set of functions based on the scripting capabilities of mIRC

Includes HideWindow program to keep the bot hidden

GT Bot

Includes BNC, a proxy system for anonymity

Includes psexec.exe to facilitate remote process execution

Nothing to suggest it was designed to be extensible

Different versions for specific malicious intents

- With DCOM includes DCOM exploits

Points of Analysis

Botnet Control Mechanisms

Host Control Mechanisms

Propagation Mechanisms

Target Exploits and Attack Mechanisms

Malware Delivery Mechanisms

Botnet Control Mechanisms

Command language and **control protocols** are used to operate botnets remotely after target systems have been compromised

All analyzed bots base C&C on IRC

Disruption of communication can render a botnet useless

Network operators can sniff for specific commands in IRC traffic and identify compromised systems

Botnet Control Mechanisms

Agobot

C&C system derived from IRC

Standard IRC is used to establish connections

IRC and commands developed for Agobot are used for command language

SDBot

Command language is lightweight version of IRC

Has IRC cloning and spying

Botnet Control Mechanisms

SpyBot

Command language is a subset of SDBot’s command language

GT Bot

Simplest command language of the bot families

Large variations across different versions

Points of Analysis

Botnet Control Mechanisms

Host Control Mechanisms

Propagation Mechanisms

Target Exploits and Attack Mechanisms

Malware Delivery Mechanisms

Host Control Mechanisms

The mechanisms used by the bot to manipulate a victim host once it has been compromised

Fortify the local system against malicious attacks

Disable anti-virus software

Harvest sensitive information

Host Control Mechanisms

Agobot

Commands to secure system

Broad set of commands to harvest sensitive information

pctrl commands to list or kill processes running on host

inst commands to add or delete autostart entries

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

Host Control Mechanisms

SDBot

Limited capabilities

Basic remote execution commands

Some ability to gather local information

Auxiliary patches add more capabilities

SDBot Commands

Command	Description
download <url> <dest> <action>	Downloaded specified file and execute if action is 1
killthread <thread#>	Kill specified thread
update <url> <id>	If bot ID is different than current, download “sdbot executable” and update
sysinfo	List host system information (CPU/RAM/OS and uptime)
execute <visibility> <file> parameters	Run a specified program (visibility is 0/1)
cdkey/getcdkey	Return keys of popular games e.g., Halflife, Soldier of Fortune etc.

Host Control Mechanisms

SpyBot

Similar capabilities to Agobot

Local file manipulation

Key logging

Process/system manipulation, remote command execution

SpyBot Commands

Command	Description
delete <filename>	Delete a specified file
execute <filename>	Execute a specified file
rename <origfile> <newfile>	Rename a specified file
makedir <dirname>	Create a specified directory
startkeylogger	Starts the on-line keylogger
stopkeylogger	Stops the keylogger
sendkeys <keys>	Simulates key presses
keyboardlights	Flashes remote keyboard lights 50x
passwords	Lists the RAS passwords in Windows 9x systems
listprocesses	Return a list of all running

Command	Description
listprocesses	Return a list of all running processes
killprocess <processname>	Kills the specified process
threads	Returns a list of all running threads
killthread < number >	Kills a specified thread
disconnect <number>	Disconnect the bot for number seconds
reboot	Reboot the system
cd-rom <0/1>	Open/close cd-rom
opencmd	Starts cmd.exe (hidden)
cmd <command>	Sends a command to cmd.exe
get <filename>	Triggers DCC send on bot

Host Control Mechanisms

GT Bot

Most limited capabilities

Base capabilities are only gathering local system information and running or deleting local files

Many versions with more capabilities

Points of Analysis

Botnet Control Mechanisms

Host Control Mechanisms

Propagation Mechanisms

Target Exploits and Attack Mechanisms

Malware Delivery Mechanisms

Propagation Mechanisms

The mechanisms bots use to search for new host systems

Traditionally horizontal or vertical scans

- Horizontal is one port across an address range
- Vertical is across a port range on an address

Propagation Mechanisms

Agobot

Relatively simple, essentially vertical and horizontal scanning

SDBot

No scanning or propagation in base distribution

Variants with horizontal, vertical scanning and more complex methods

Propagation Mechanisms

SpyBot

Simple horizontal and vertical scanning

GT Bot

Simple horizontal and vertical scanning

Due to simplicity and uniformity of methods, it may be possible to develop statistical finger printing methods to identify scans from botnets

Points of Analysis

Botnet Control Mechanisms

Host Control Mechanisms

Propagation Mechanisms

Target Exploits and Attack Mechanisms

Malware Delivery Mechanisms

Exploits and Attack Mechanisms

Specific methods for attacking known vulnerabilities on target systems

Agobot

Includes an ever broadening set of exploits

Agobot exploits

Bagle scanner

DCOM scanners

MyDoom scanner

Dameware scanner

NetBIOS scanner

Radmin scanner

MS-SQL scanner

Generic DDoS module

Exploits and Attack Mechanisms

SDBot

No exploits in standard distribution

Modules for sending UDP and ICMP packets

- DoS

Numerous variants with exploits

Numerous variants with DDoS attack modules

Exploits and Attack Mechanisms

SpyBot

Exploits depend on version of SpyBot

- Wide range of exploits

Evaluated version has attacks on open NetBIOS shares

DDoS interface closely related to SDBot

- UDP, ICMP, and TCP SYN

Exploits and Attack Mechanisms

GT Bot

- This variant has RPC-DCOM exploits and Simple ICMP floods
- Many variants with many exploits and DoS capabilities

Bots will likely become more like Agobot, each version having many exploits

Points of Analysis

Botnet Control Mechanisms

Host Control Mechanisms

Propagation Mechanisms

Target Exploits and Attack Mechanisms

Malware Delivery Mechanisms

Malware Delivery Mechanism

The mechanisms bots use to deliver exploits

Packers and shell encoders used to compress and obfuscate code

SDBot, SpyBot, and GT Bot deliver exploit and encoded malware in one script

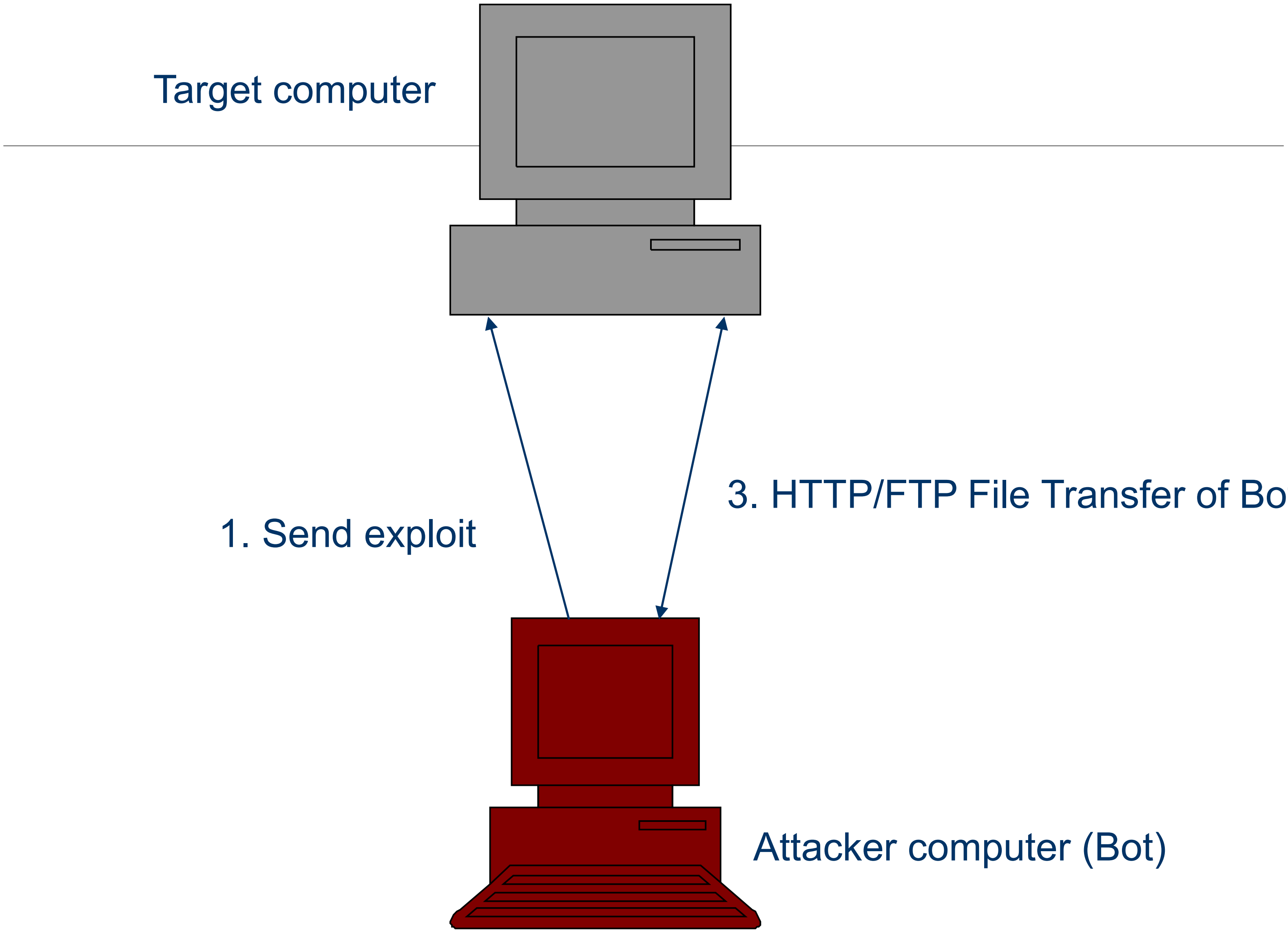
Agobot separates exploits and delivery

Exploit vulnerability and open shell on remote host

Encoded malware binary delivered by HTTP or FTP

Enables encoder to be used across exploits, streamlining codebase and potentially diversifying the resulting bit streams

Agobot Delivery



Botnet Operation

General

Assign a new random nickname to the bot

Cause the bot to display its status

Cause the bot to display system information

Cause the bot to quit IRC and terminate itself

Change the nickname of the bot

Completely remove the bot from the system

Display the bot version or ID

Display the information about the bot

Make the bot execute a .EXE file

IRC Commands

Cause the bot to display network information

Disconnect the bot from IRC

Make the bot change IRC modes

Make the bot change the server Cvars

Make the bot join an IRC channel

Make the bot part an IRC channel

Make the bot quit from IRC

Make the bot reconnect to IRC

Redirection

Redirect a TCP port to another host

Redirect GRE traffic that results to proxy PPTP VPN connections

DDoS Attacks

Redirect a TCP port to another host

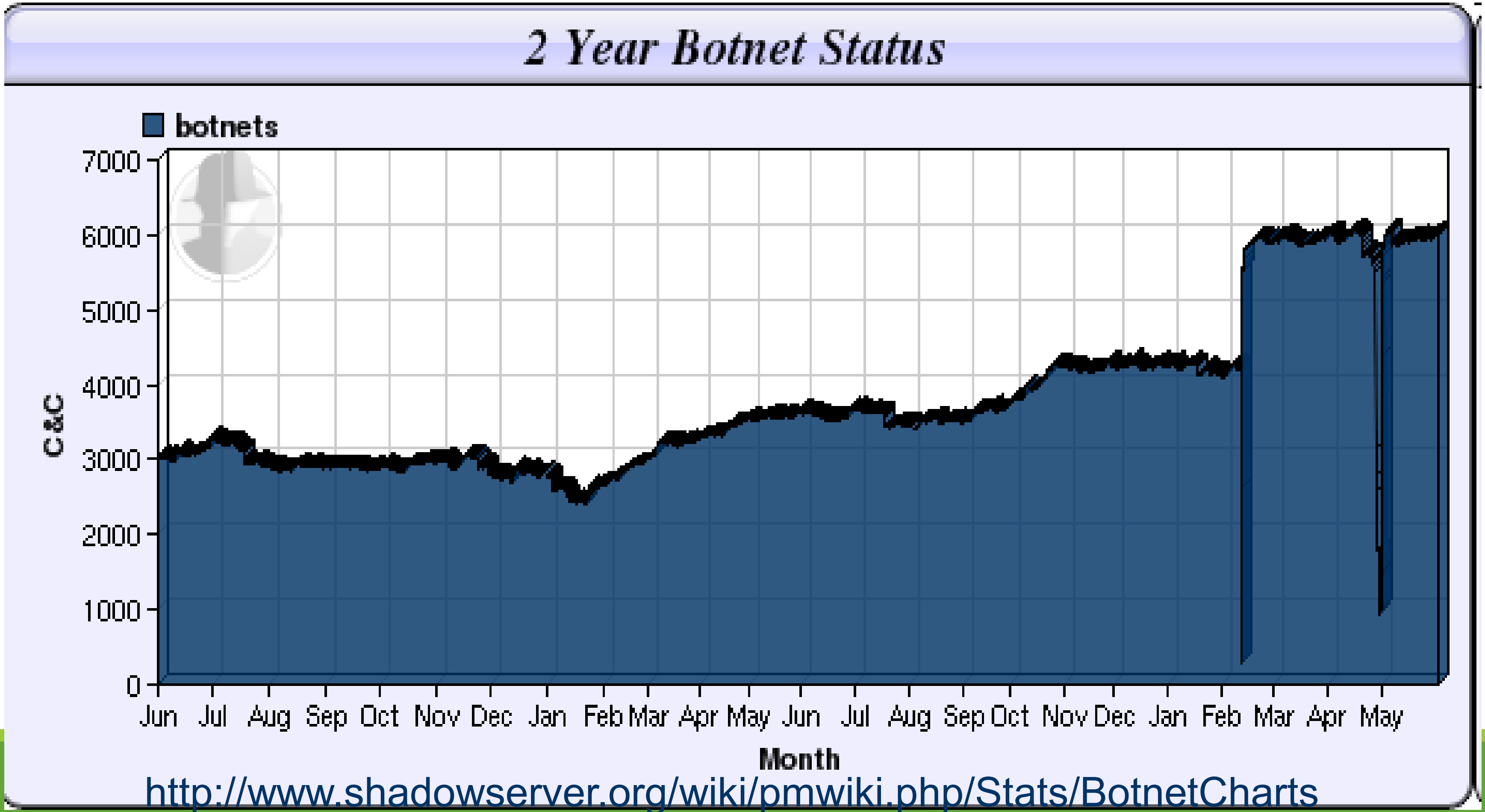
Redirect GRE traffic that results to proxy PPTP VPN connections

Information theft

Steal CD keys of popular games

Program termination

Number of Botnets



http://www.shadowserver.org/wiki/pmwiki.php/Stats/BotnetCharts

Botnet Detection and Tracking

Network Intrusion Detection Systems (*e.g.*, Snort)

Signature: alert tcp any any -> any any (msg:"Agobot/Phatbot Infection Successful"; flow:established; content:"221”)

traffic analysis

Honeynets: gather information

Run unpatched version of Windows

Usually infected within 10 minutes

Snooping on IRC Servers

Article: “Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation” by Thorsten Holz

Revealing Botnet Membership Using DNSBL Counter-Intelligence

Anirudh Ramachandran, Nick Feamster and David Dagon

College of Computing, Georgia Institute of Technology

From the presses…

“Botnets send masses of spam until they are blacklisted by anti-spam firms. Once blacklisted, the owner sells the botnet to people who launch denial-of-service (DDOS) attacks.”

“Spam clubs also advertise lists of botnets on hire and fresh proxies -- computers that have recently been taken over.”

-- Steve Linford, CEO, Spamhaus

ZDNet UK News, September 2004

Motivation for this work

Fact: Bot-herds advertise and sell their “clean” bots at a premium

Insight: If the claims are true, they must be looking up their bots’ status in some blacklist!

Opportunistic Application: Might it be possible to mine DNS Blacklist *queries* to reveal such *reconnaissance* activity?

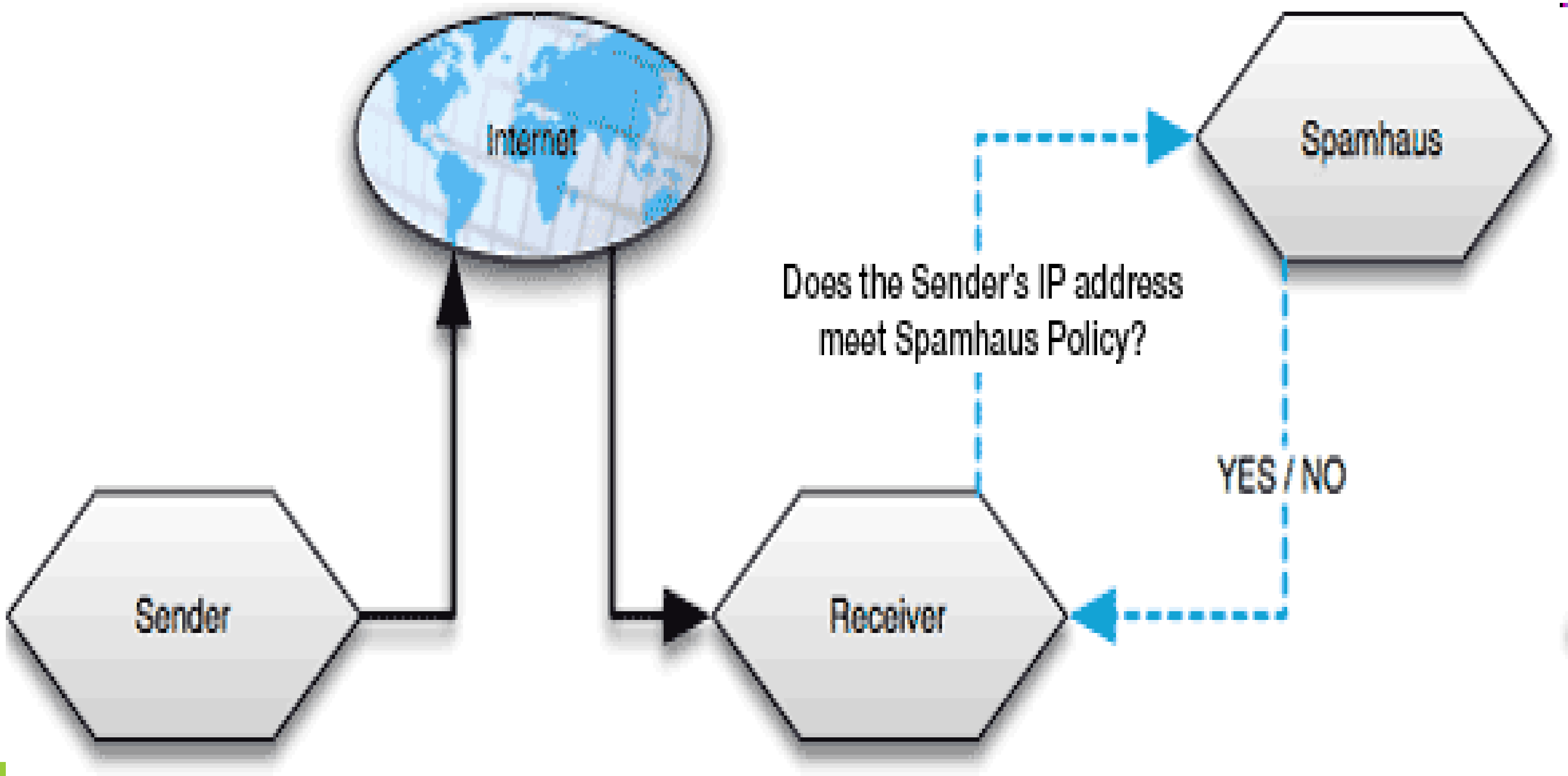
DNS Blacklists – How they work

First: Mail Abuse Prevention System (MAPS)

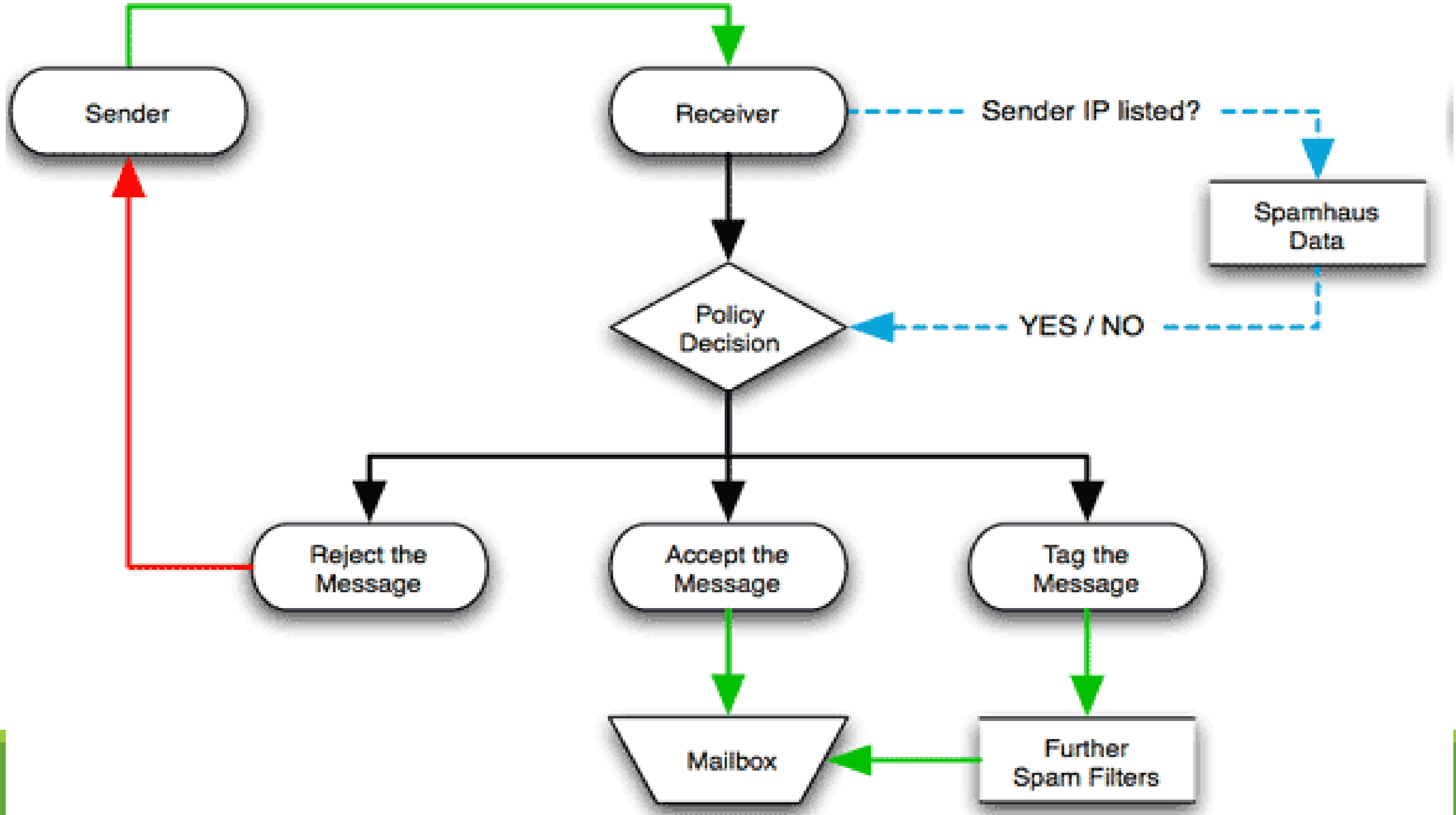
- Paul Vixie, Dave Rand -- 1996

Today: Spamhaus, spamcop, dnsrbl.org etc.

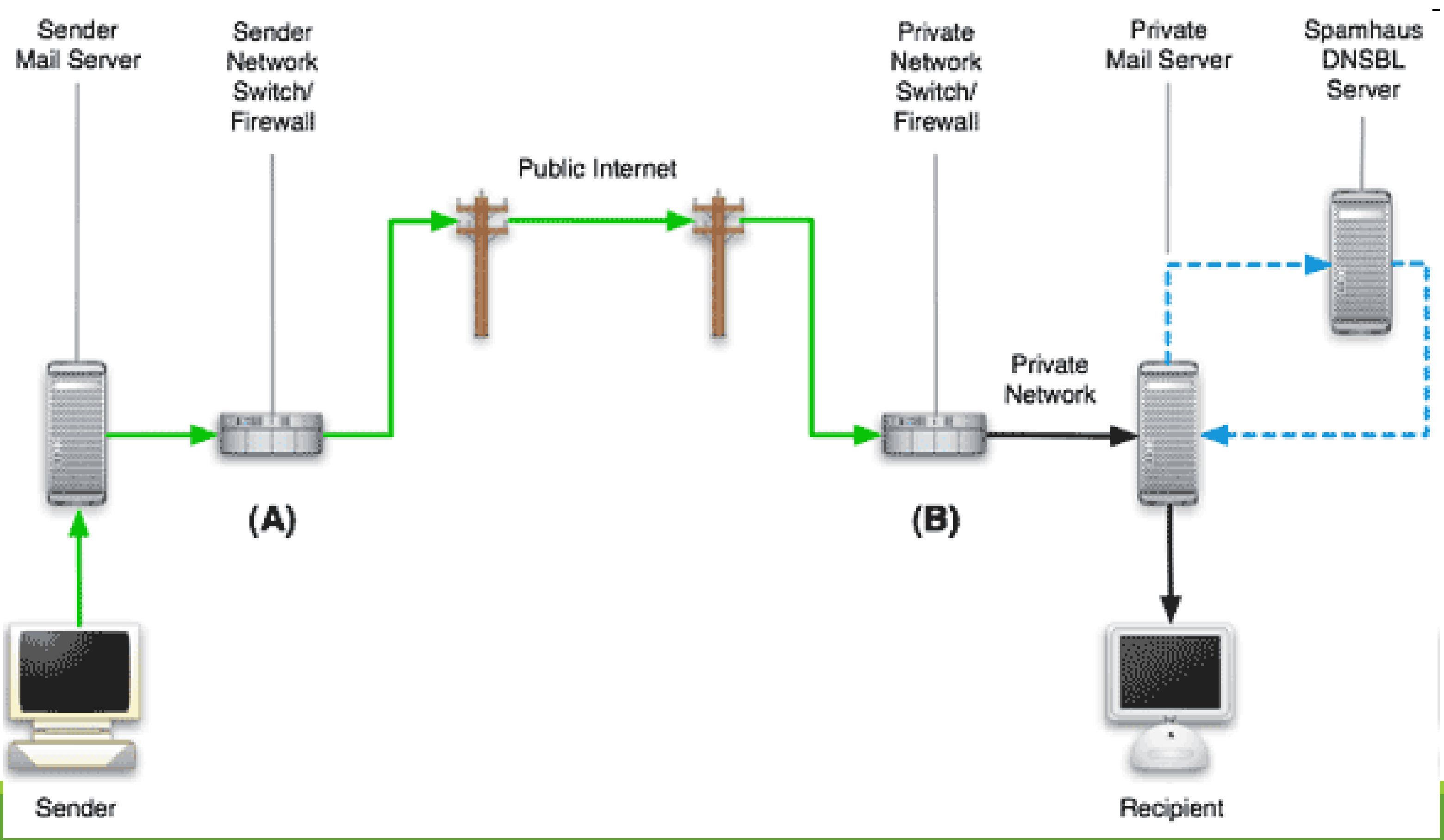
Spamhaus: How it works



Spamhaus: How it works



Spamhaus: How it works



Detecting Reconnaissance

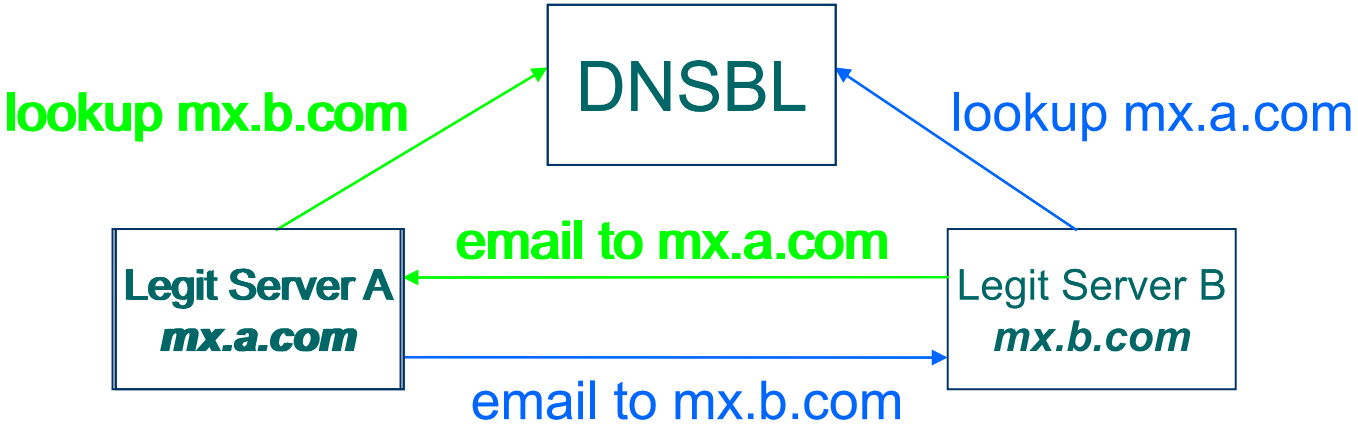
Key Requirement: Distinguish reconnaissance queries from queries performed by legitimate mail servers

Our Solution: Develop heuristics based on the spatial and temporal properties of a *DNSBL Query Graph*

We focus (mostly) on spatial heuristics

Heuristics

- *Spatial Heuristic:* Legitimate mail servers will perform queries *and be the object of queries.*

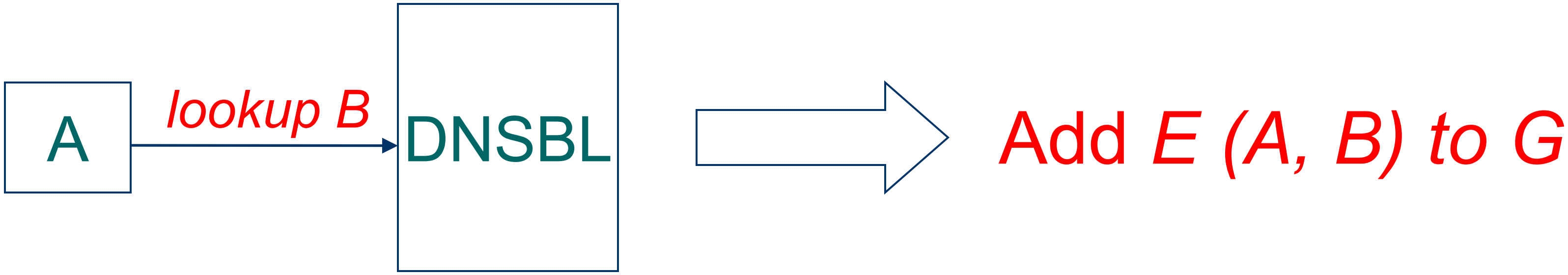


  - *Hosts issuing reconnaissance queries usually will not be queried*

- *Temporal Heuristic:* Legitimate lookups reflect arrival patterns of legitimate email

Applying the Spatial Heuristic

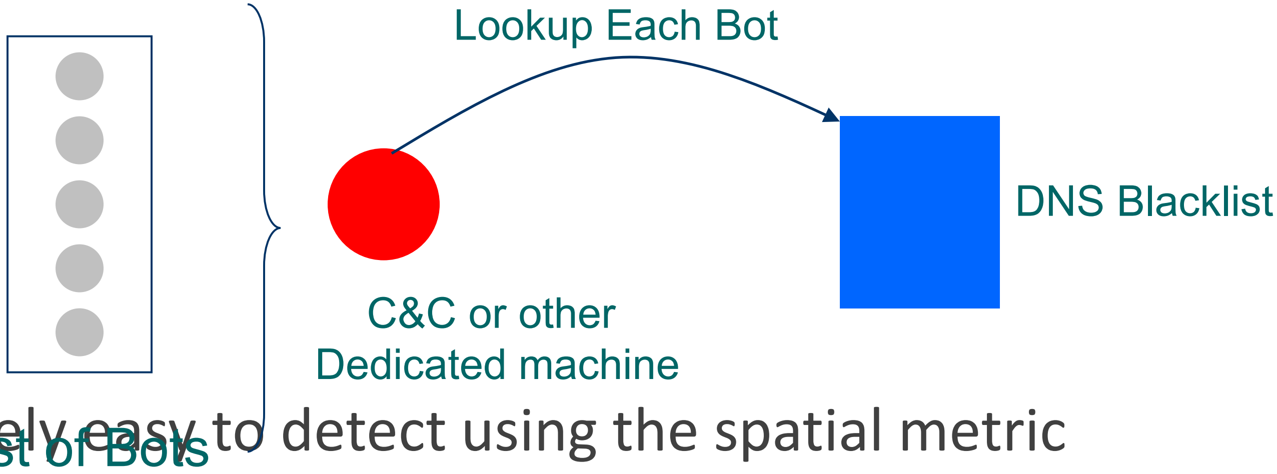
Construct the directed *DNSBL Query Graph G*



Extract nodes (and their connected components) with the highest values of the spatial metric λ, where λ = (Out-degree/In-degree)

Third-Party Reconnaissance

Third-party performs reconnaissance query



Relatively easy to detect using the spatial metric

Other Techniques

Self-Reconnaissance

Each bot looks itself up

This should not happen normally (at least, not *en-masse*) – thus, easy to detect

Distributed Reconnaissance

Bots perform lookups for other bots

Complex to deploy and operate

A Multifaceted Approach to Understanding the Botnet Phenomenon

Authors :

Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, Andreas Terzis

Computer Science Department

Johns Hopkins University

Measuring Botnets

Three Distinct Phases

Malware Collection

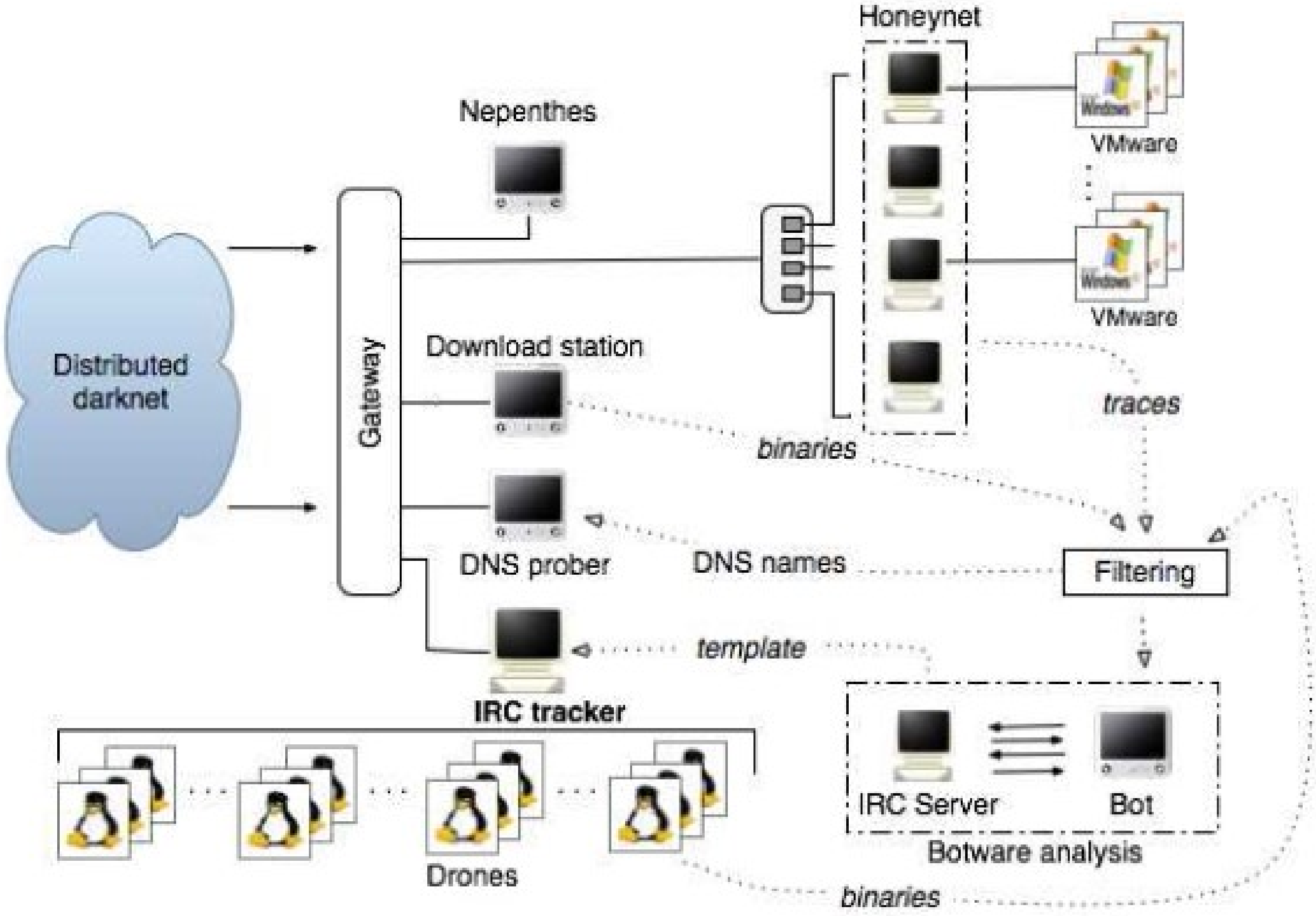
Collect as many bot binaries as possible

Binary analysis via gray-box testing

Extract the features of suspicious binaries

Longitudinal tracking

Track how bots spread and its reach



Darknet : Denotes an allocated but unused portion of the IP address space.

Malware Collection

Nepenthes
Download station

Nepenthes is a low interaction honeypot

Nepenthes mimics the replies generated by vulnerable services in order to collect the first stage exploit

Modules in nepenthes

Resolve DNS asynchronous

Emulate vulnerabilities

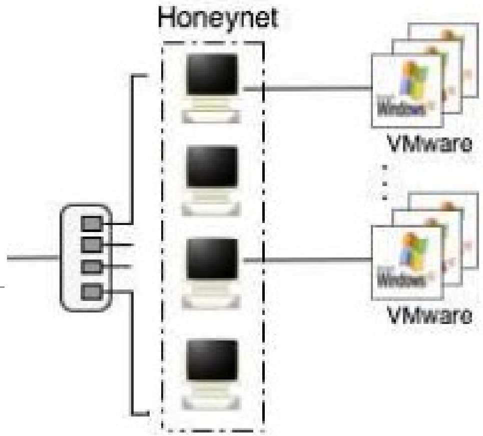
Download files – Done here by the Download Station

Submit the downloaded files

Trigger events

Shellcode handler

Malware Collection



Honeynets also used along

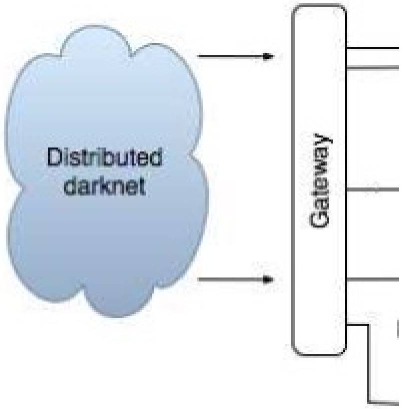
with nepenthes

Catches exploits missed by nepenthes

Unpatched Windows XP are run which is base copy

Infected honeypot compared with base to identify Botnet binary

Gateway



Routing to different components

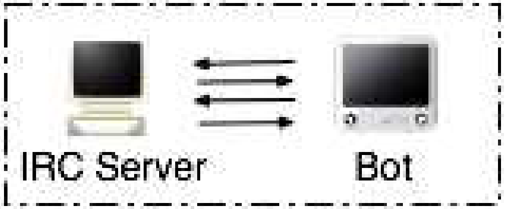
Firewall : Prevent outbound attacks & self infection by honeypots

Detect & Analyze outgoing traffic for infections in honeypot

Only 1 infection in a honeypot

Several other functions

Binary Analysis



Botware analysis

Two logically distinct phases

Derive a network fingerprint of the binary

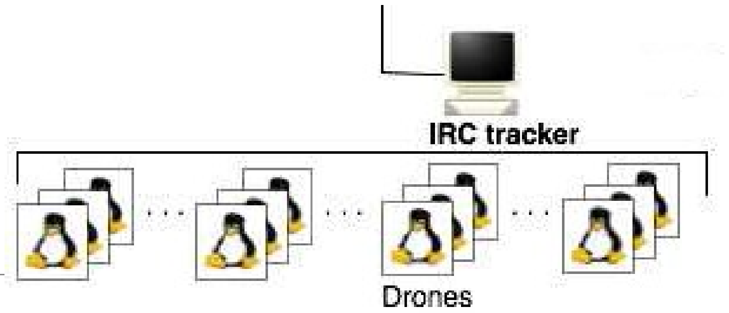
Derive IRC-specific features of the binary

$$f_{net} = \langle \texttt{DNS}, \texttt{IPs}, \texttt{Ports}, \texttt{scan} \rangle$$

$$f_{irc} = \langle \texttt{PASS}, \texttt{NICK}, \texttt{USER}, \texttt{MODE}, \texttt{JOIN} \rangle$$

- IRC Server learns Botnet “dialect” - Template
- Learn how to correctly mimic bot’s behavior - Subject bot to a barrage of commands

IRC Tracker



Use template to mimic bot

Connect to real IRC server

Communicate with botmaster using bot “dialect”

Drones modified and used to act as IRC Client by the tracker to Cover lot of IP address

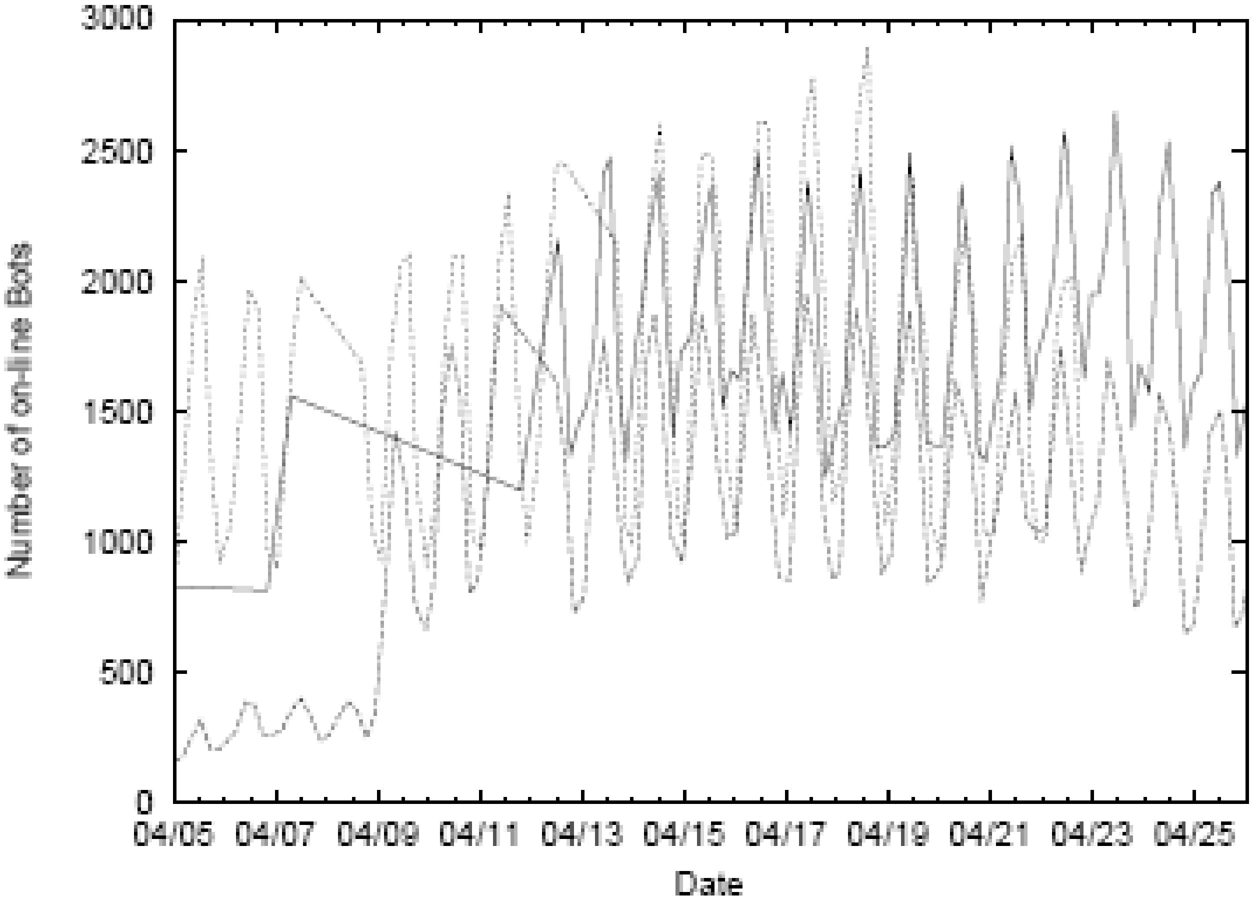
DNS Tracker

Bots issue DNS queries to resolve the IP addresses of their IRC servers

Tracker uses DNS requests

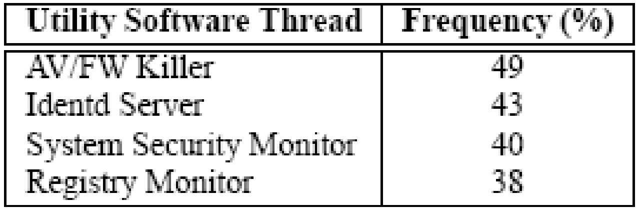
Maintain hits to a server

Botnet Online Population



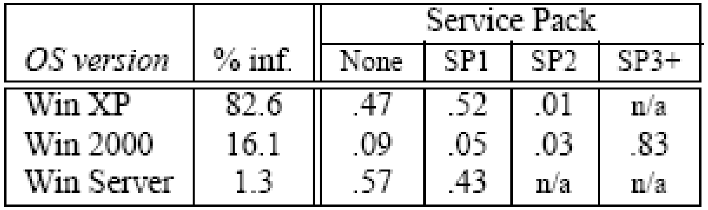
Botnet Software axonomy

Services Launched in Victim Machine

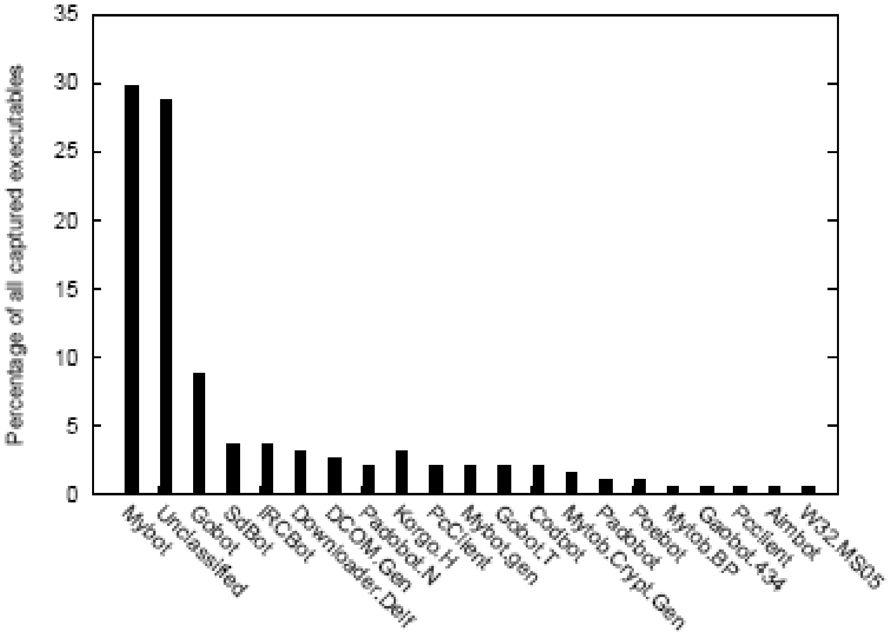


Utility Software Thread	Frequency (%)
AV/FW Killer	49
Identd Server	43
System Security Monitor	40
Registry Monitor	38

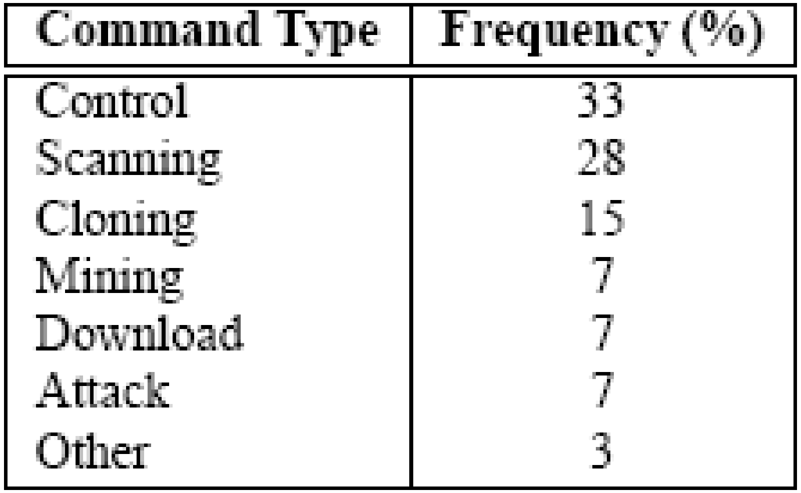
OS of Exploited Host



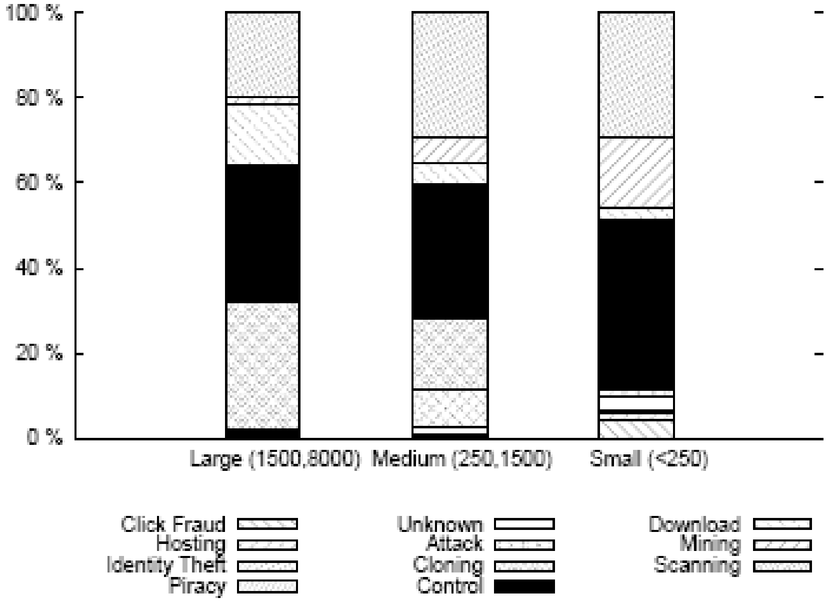
		Service Pack			
OS version	% inf.	None	SP1	SP2	SP3+
Win XP	82.6	.47	.52	.01	n/a
Win 2000	16.1	.09	.05	.03	.83
Win Server	1.3	.57	.43	n/a	n/a



Botmaster Analysis



Command Type	Frequency (%)
Control	33
Scanning	28
Cloning	15
Mining	7
Download	7
Attack	7
Other	3



Strengths

All aspects of a botnet analyzed

No prior analysis of bots

Ability to model various types of bots

Weakness

Only Microsoft Windows systems analyzed

Focus on IRC-based bots as they are predominant

Online Scam Hosting is Dynamic

- The sites pointed to by a URL that is received in an email message may point to different sites
- Maintains agility as sites are shut down, blacklisted, etc.
- One mechanism for hosting sites: **fast flux**

Example

Multi-homed DNS

FQDN maps to 3 or more IP addresses

botnet1.example.com pointing to 127.0.0.1

botnet1.example.com pointing to 127.0.0.2

botnet1.example.com pointing to 127.0.0.3

botnet1.example.com pointing to 127.0.0.4

botnet1.example.com pointing to 127.0.0.5

botnet1.example.com pointing to 127.0.0.6

Dynamic DNS used thru commercial site

Change IP addresses quickly

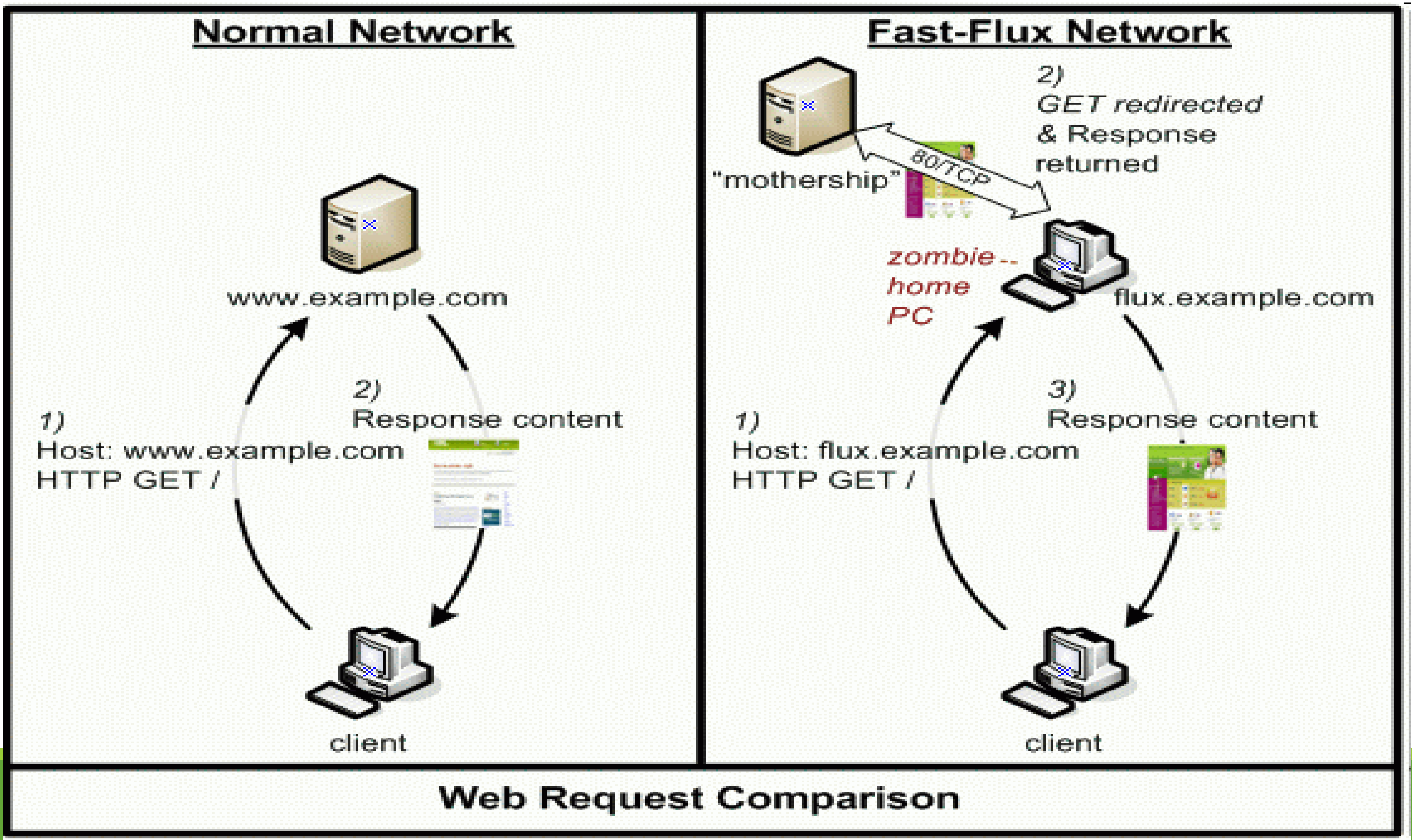
Short DNS TTLs for clients

Remap DNS often, check at boot

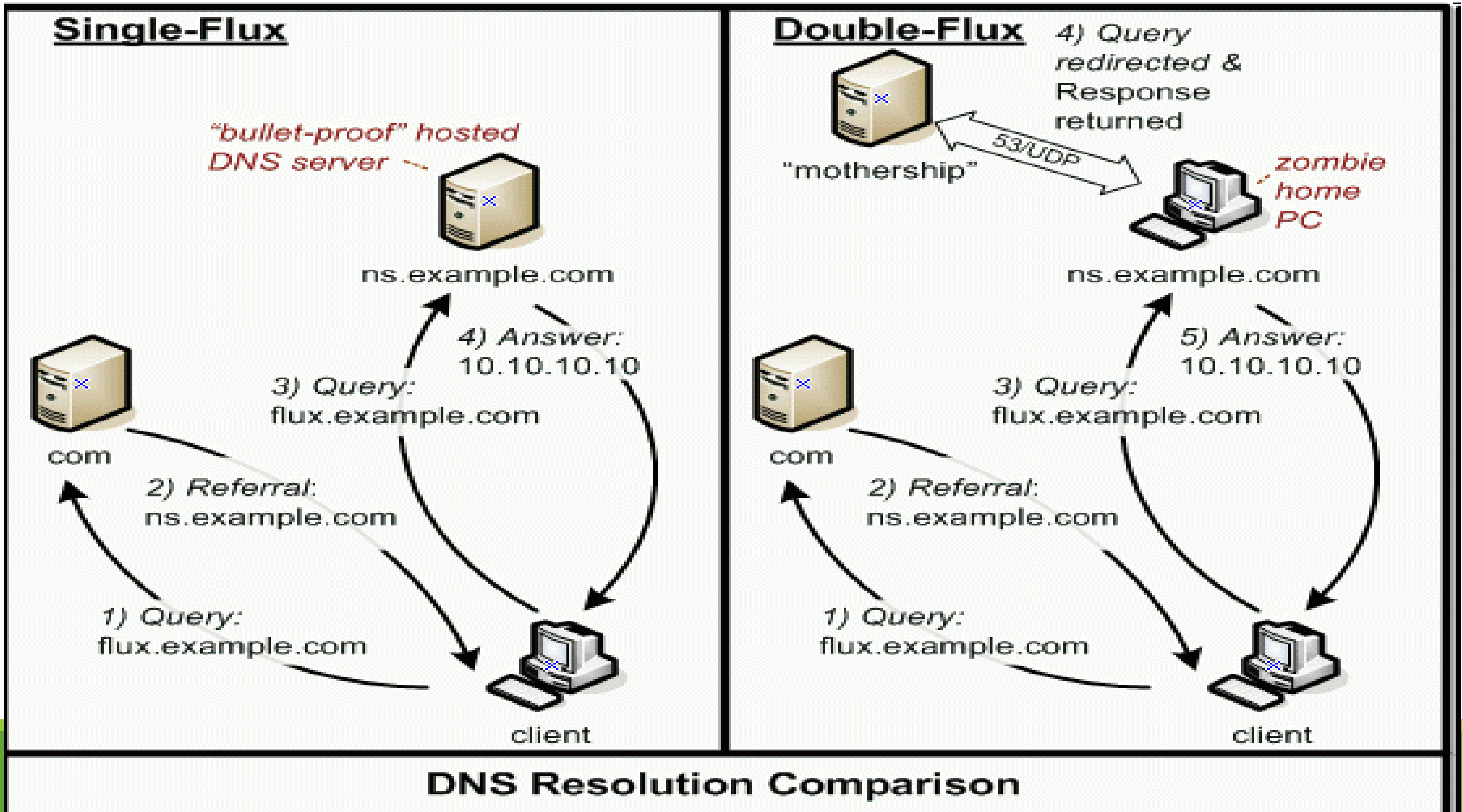
FastFlux DNS

Change IP addresses and/or DNS names quickly (for spam < 5 minutes) and often

Single Fast Flux Network



DOUBLE-FLUX SERVICE NETWORKS


Single-Flux
"bullet-proof" hosted DNS server
ns.example.com
com
4) Answer: 10.10.10.10
3) Query: flux.example.com
2) Referral: ns.example.com
1) Query: flux.example.com
client
Double-Flux
4) Query redirected & Response returned
"mothership"
53/UDP
zombie home PC
ns.example.com
com
5) Answer: 10.10.10.10
3) Query: flux.example.com
2) Referral: ns.example.com
1) Query: flux.example.com
client
DNS Resolution Comparison

Fast-Flux Detection

Number of A records returned per query

Number of NS records returned

The diversity of unrelated networks represented

http://dnsbl.abuse.ch/index.php