Botnets

Usman Jafarey

Including slides from The Zombie Roundup by Cooke, Jahanian, McPherson of the University of Michigan
The Old Days...

Worms

These attacks disrupt infrastructure

DoS

February 8, 2000

Yahoo Attributes a Lengthy Service Failure to an Attack

By MATT RICHTEL

SAN FRANCISCO, Feb. 7 -- Yahoo Inc. blamed a "planned attack" by computer hackers for a service failure that lasted nearly three hours today, in a rare interruption of one of the most popular and best performing sites on the World Wide Web.
A Dramatic Escalation/Transformation

ID Theft

Phishing

These attacks directly target people

SPAM

Spyware
Rise of the Zombies

- New *personal* attacks often rely on another resource (e.g. phishing site, SPAM relay)
- Anonymous use of resource highly desirable
  => attackers use another compromised system as a proxy!

  *Attackers have learned a compromised system is more useful alive than dead!*

This talk is about detecting and disrupting access to the anonymous infrastructure used in these attacks
The Botnet

UK Broadband → JP Corp

Internet Backbone

US Corp → Attack Command

US Broadband
Bot History and Structure

- Not New: An original use, help Internet Relay Chat (IRC) Operators (*Eggdrop/1993*)
- Nefarious attack bots soon emerged (*DDoS*)
- Developed Sophisticated Hiding and Attack Capabilities (*SubSeven, Bot/Bionet Bot*)
- Modern Bots: (*AgoBot*[PhatBot], *GTBot*[rBot]*)

**WORM**

- **Communication**: IRC (can be encrypted)
- **Attack**: DoS, SPAM Relay, Phishing Site…
- **Propagation**: Vulnerabilities, File Shares, P2P…
Big Bad Bots

- Total infected bot hosts 800,000 - 900,000
  [CERT CA-2003-08]
  > 100,000 nodes/botnet
- 1000’s of new bots each day [Symantec 2005]
- Many articles/press citing thousands of infected hosts [IEEE S&P, Register]
- Difficult to measure:
  => Population likely much much larger!
• Very little hard data on botnets!
• We asked operators (five Tier-1 & Tier-2 ops):
  • They are actively fighting the problem
  • # of Botnets - *increasing*
  • Bots per Botnet - *decreasing*
    *Used to be 80k-140k, now 1000s (evasion/economics?)*
  • More firepower:
    *Broadband (1Mbps Up) x 100s == OC3!!!*
• Custom botnets (all .edu, .gov/.mil) - economics?
Bot/Botnet Measurements - Honeypot

- Windows 2000/XP Honeypot
- Placed behind proxy:
  1. Rate limit traffic 12KB/s
  2. Disallow local network
  3. Log all traffic
- 12 experimental runs over a month:
  - 12-72 hour traces > 100MBs
  - Recruited into least 15 unique botnets
  - Bots used DCOM/RPC, LSASS

=> Bots are extremely prevalent
Detecting and Stopping Bots

1. Prevent systems from getting infected
2. Directly detect *bot* communications between *bots* and between *bots* and *bot controllers*
3. Detect the secondary features of a *bot* infection like propagation or attacks
Prevent Infection

- Well developed methods:
  - Anti-virus
  - Firewalls
  - Patching

- But:
  - Might not directly control of systems (ISPs)
  - Can’t upgrade certain systems (Win98 DAQ)
  - Complex infection vectors: App-level (javascript, AIM)
  - Custom threat (Israeli trojan)

- Naïve to assume 100% protected
Many bots use IRC for Command and Control

Detect IRC Bot Commands
- Offramp TCP port 6667
- Inspect Payloads (advscan...) [honeynet05]
- IRC Behavior [Racine04]
### Detecting Bot Communication...

<table>
<thead>
<tr>
<th>Topology</th>
<th>Design Complexity</th>
<th>Message Latency</th>
<th>Survivability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Peer-to-Peer</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Random</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

**Taxonomy of Bot Communication Topologies**

- Reliance on detecting *Bot Communication* degenerates into arms race between bot authors and defenders
- Communication is very flexible
  - Easy to Encrypt/Obfuscate
Relying on detecting bot communication is *not* viable in the long term.

- Leverage *all* available bot characteristics.
- Build detectors for each bot behavior.
Behavioral Bot Detection

• Preliminary evidence very promising:
• Strong correlation between bot communication and bot propagation

Correlating data sources from a large live network (payloads & IMS dark IP sensors):

<table>
<thead>
<tr>
<th>Bot Command Detected</th>
<th>Δ IMS Detection Time</th>
<th>Scan Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipscan r.r.r.r dcom2</td>
<td>11 secs</td>
<td>Global Random</td>
</tr>
<tr>
<td>ipscan s.s.s.s dcom2</td>
<td>0 secs</td>
<td>Global Seq.</td>
</tr>
<tr>
<td>ipscan 24.s.s.s dcom2</td>
<td>-</td>
<td>Local 24/8 Seq.</td>
</tr>
<tr>
<td>ipscan 69.27.s.s dcom2</td>
<td>-</td>
<td>Local 69.27/16 Seq.</td>
</tr>
<tr>
<td>ipscan s.s.s lsass</td>
<td>0 secs</td>
<td>Local /8 Seq.</td>
</tr>
<tr>
<td>ipscan s.s webdav3</td>
<td>0 secs</td>
<td>Local /16 Seq.</td>
</tr>
</tbody>
</table>
Botnet Disruption

- Once you detect a bot how to shut it down?
- Two goals
  1. Take down the bot
  2. Take down the botnet
- Problem is similar to infiltrating a gang: monitoring the bot => provide info on botnet (i.e. a “narc”)
- Problem is complicated because many botnets span many countries
Conclusions

- Bots provide support infrastructure for a large range of devastating Internet attacks
- IRC-based botnet detection may be effective tool today
- Tomorrow must focus on holistic view of bot behavior
- Interesting questions:
  - How do we measure bots?
  - Who is responsible for cleanup? (Organizations/ISPs/Law Enforcement)
  - Global enforcement => bots in US attack China?
Botnet

- Collection of infected systems
- Controlled by one party
Most commonly used Bot families

- Agobot
- SDBot
- SpyBot
- GT Bot
Agobot

- Most sophisticated
- 20,000 lines C/C++ code
- IRC based command/control
- Large collection of target exploits
- Capable of many DoS attack types
- Shell encoding/polymorphic obfuscation
- Traffic sniffers/key logging
- Defend/fortify compromised system
- Ability to frustrate disassembly
SDBot

- Simpler than Agobot, 2,000 lines C code
- Non-malicious at base
- Utilitarian IRC-based command/control
- Easily extended for malicious purposes
  - Scanning
  - DoS Attacks
  - Sniffers
  - Information harvesting
  - Encryption
SpyBot

- <3,000 lines C code
- Possibly evolved from SDBot
  - Similar command/control engine
  - No attempts to hide malicious purposes
GT Bot

- Functions based on mIRC scripting capabilities
- HideWindow program hides bot on local system
- Port scanning, DoS attacks, exploits for RPC and NetBIOS
- Variance in codebase size, structure, complexity, implementation
- Convergence in set of functions
  - Possibility for defense systems effective across bot families
- Bot families extensible
- Agobot likely to become dominant
Control

- All of the above use IRC for command/control
  - Disrupt IRC, disable bots
  - Sniff IRC traffic for commands
  - Shutdown channels used for Botnets
- IRC operators play central role in stopping botnet traffic
- Automated traffic identification required
- Future botnets may move away from IRC
  - Move to P2P communication
  - Traffic fingerprinting still useful for identification
Host control

- Fortify system against other malicious attacks
- Disable anti-virus software
- Harvest sensitive information
  - PayPal, software keys, etc.
  - Economic incentives for botnets
- Stresses need to patch/protect systems prior to attack
- Stronger protection boundaries required across applications in OSes
Propagation

- Horizontal scans
  - Single port across address range
- Vertical scans
  - Single IP across range of ports
- Current scanning techniques simple
  - Fingerprinting to identify scans
- Future methods
  - Flash, more stealthy
- Source code examination
  - Propagation models
Exploits/Attacks

- Agobot
  - Has the most elaborate set
  - Several scanners, various flooding mechanisms for DDoS
- SDBot
  - None in standard
  - UDP/ICMP packet modules usable for flooding
  - Variants include DDoS
- SpyBot
  - NetBIOS attacks
  - UDP/TCP/ICMP SYN Floods, similar to SDBot
  - Variants include more
- GTBot
  - RPC-DCOM exploits
  - ICMP Floods, variants include UDP/TCP SYN floods
- Required for protection
  - Host-based anti-virus
  - Network intrusion detection
  - Prevention signatures sets
- Future
  - More bots capable of launching multiple exploits
- DDoS highlight danger of large botnets
Delivery

- Packers, shell encoders for distribution
- Malware packaged in single script
- Agobot separates exploits from delivery
  - Exploit vulnerability
    - Buffer overflow
  - Open shell on host
  - Upload binary via HTTP or FTP
  - Encoder can be used across multiple exploits
  - Streamlines codebase
- NIDS/NIPS need knowledge of shell codes/perform simple decoding
- NIDS incorporate follow-up connection detection for exploit/delivery separation prevention
Obfuscation

- Hide details of network transmissions
- Only slightly provided by encoding
- Same key used in encoding => signature matching
- Polymorphism – generate random encodings, evades signature matching
  - Agobot
    - POLY_TYPE_XOR
    - POLY_TYPE_SWAP (swap consecutive bytes)
    - POLY_TYPE_ROR (rotate right)
    - POLY_TYPE_ROL (rotate left)
- NIDS/Anti-virus eventually need to develop protection against polymorphism
Deception

- Detection evasion once installed
- a.k.a. *rootkits*
- Agobot
  - Debugger tests
  - VMWare tests
  - Anti-virus process termination
  - Pointing DNS for anti-virus to localhost
- Shows merging between botnets/trojans/etc.
  - Honeynet monitors must be aware of VM attacks
  - Better tools for dynamic malware analysis
  - Improved rootkit detection/anti-virus as deception improves