Botnets
Secret Puppetry With Computers

Balaji Prasad T.K (bpt@email.arizona.edu)
Nupur Maheshwari (nupurm@email.arizona.edu)
Department of Computer Science
University of Arizona

April 22, 2012
1. What are Botnets?

2. Technical Overview

3. Compare and Contrast

4. Detection and Prevention

5. Evolution and Conclusion
A botnet is a network of zombie computers which are remotely controlled by a botmaster. Components:

- Botmaster
- Zombies
- Communication Channel
- Servers
Botnet Overview
Facts and Stats

- 83% of global spam
- 3 million botnets, 100 spams per minute
- Only 3 survived from 2010
- Why no Linux Botnets?
Bot Stories

- Wiki Leaks - Used Botnet for campaign
  
  anonymous-uses-30000-pc-strong-botnet-in-wikileaks-campaign/

- App Stores - Marketing
Botnets Threat Landscape

- Have managed to bring down websites of biggies like cia.gov (US central investigation agency), SOCA.gov (British serious organised crime agency) etc
- Here is a list of what you can do:
<table>
<thead>
<tr>
<th>Bot</th>
<th>Year</th>
<th>Attack Method</th>
<th>Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>StormBot</td>
<td>Jan 2007</td>
<td>fighting-back capabilities</td>
<td>Spam with Subject - 230 dead as storm batters Europe</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Affected: private computers in Europe and US</td>
</tr>
<tr>
<td>Conflicker</td>
<td>Nov 2009</td>
<td>RPC Request</td>
<td>Affected: French Navy, United Kingdom Ministry of Defence, Manchester City council’s system and police network, German army systems</td>
</tr>
</tbody>
</table>
1. What are Botnets?

2. Technical Overview

3. Compare and Contrast

4. Detection and Prevention

5. Evolution and Conclusion
How They Recruit

**STEP 1:** Probe a range IPS. Wait for a response from an exploitable n/w service or trojan back door.

**STEP 2a:** Response from a targetted port.

**STEP 3:** Send full version of bot binaries to the victim.

**STEP 4:** Victim has established C&C channel.
- Finds a permanent port for communication
- Secures itself from antimalware software
- Attached itself to some windows service to survive reboot

**STEP 5a:** Victim is part of the Zombie network.

**STEP 5b:** Also scan for other victims.
How They Differ

Virus Vs. Worm Vs. Botnet

http://www.youtube.com/watch?v=X1Sc8W5VaR8
How They Propogate

- Scan the network
- Send spam mails
- Drive-by download
- Install malware
How They Obfuscate

- Encryption
- Mutation
- Encoded Peer List
Botnet obfuscation mechanisms

Security Measures

Public Network ✗

Private Network ✓

Use Secret Pass Phase

Patch Up The Point of Entry

Maintain Database of the Enemy
Use a Passcode

Botmaster

Attack
72.223.66.190

Delete Files

72.223.66.190
Use a Passcode

Botmaster
66.171.255.150

Delete Files

Attack
66.171.255.150

CIA
Use a Passcode
Patch Up

1. Botmaster
   - Attack Vulnerability
   - Control
   - Control Lost

2. Botmaster
   - Becomes Zombie and Reports

3. Botmaster 1
   - Becomes Zombie and Reports

4. Botmaster 2
   - Attack Vulnerability
   - Control
Command and Control

- IRC - Internet Relay Chat
- P2P - Peer-to-Peer
- Web Based
Command and Control - IRC

IRC C&C

Normal user

Potential victim

Zombie

IRC

IRC

IRC

Zombie

Bot master

Issue attack command

Attack channel

control channel
Command and Control - P2P

**P2P-based:**
- IRC-based botnets have centralized master which is single point of failure
- In P2P based C&C Botmaster can use any of the nodes to pass commands or collect information from other nodes in the Botnet

**Web-based:**
- Botnets evolved to use HTTP and HTTPS protocols for C&C
  - The bots talk to a web server acting as their master
- Distinct advantage to the adversary as HTTP ports are always enabled
- This C&C merges well with the normal traffic to provide obscurity
Anatomy of 2 High Profile Bots

**AgoBot**
- Also known as Phatbot - oldest known bots
- IRC based bot with a huge arsenal of exploits
- Ability to launch DDoS attacks and harvest passwords through key logging and traffic sniffing

**SDBot**
- Known since 2002-Hundreds of variants providing a wide range of capabilities
- Core code is very compact when compared to AgoBot with just 2000 lines of C code
- Extension of code to add a newer capability is very straightforward - also diffuses accountability of the creator.
Botnet Control Mechanism

AgoBot

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bot.execute</td>
<td>Makes the bot execute a specific .exe</td>
</tr>
<tr>
<td>bot.sysinfo</td>
<td>Echo the bots system information</td>
</tr>
<tr>
<td>bot.status</td>
<td>Echo bot status information</td>
</tr>
<tr>
<td>bot.nick</td>
<td>Changes the nickname of the bot</td>
</tr>
<tr>
<td>bot.open</td>
<td>Opens a specified file</td>
</tr>
<tr>
<td>bot.remove</td>
<td>Removes the bot from the host</td>
</tr>
</tbody>
</table>

SDBot uses commands like

- Ping & Pong
- Join request to establish IRC connection
- Commands sent by the master include: KICK, NICK, PART.
- All other commands will be sent as part of the PRIVMSG, NOTICE or TOPIC IRC messages
Host Control Mechanism

*AgoBot*
- Secure the system
- Harvest commands
- Pctrl commands
- Inst commands

*SDBot*
- Download
- Kill thread
- Sysinfo
- Execute
- Update
### Attack Mechanism

**AgoBot**
- Scans for backdoors left by *other* worms
- Exploits RPC Buffer Overflow in windows
- Brute force SQL servers
- DDos

**SDBot**
- Capabilities are relatively benign
- Creator can disown
- Extends to UDP and ICMP

```
udp/ping <host to attack>
< portno.ofpackets > < packetsize >
```
Obfuscation and deception mechanism

*AgoBot*
- Swapping consecutive bytes
- Rotate left / Rotate right
- Polymorphic encoding
- Looked for debuggers
- Installed virtual machines
- Kills antivirus processes
- Alters DNS servers of the AV/SW companies

*SDBot did not have any such capabilities.*
1. What are Botnets?

2. Technical Overview

3. Compare and Contrast

4. Detection and Prevention

5. Evolution and Conclusion
Anomaly based detection

- Scanning involves sending TCP SYN and other control packets to find open ports
- Calculate TCP work weight - fraction of TCP packets that were control packets
  \[ w = \frac{\text{SYN}_n + \text{ACK}_n + \text{FIN}_n}{\text{TCP}_n} \]
- Anomalous values caught. Won't work with "Idle scanning"
What is Idle scanning?

Master-M

STEP1: Find the IP of RST packet of B

STEP2: Spoof the IP address of B and send a SYN to P

STEP3: P responds to B with a SYN+ACK (or RST if the port was closed)

STEP4: Clueless B sends a RST

STEP5: M now finds if the IPID field was incremented

Innocent Bot - B

Potential Victim - P
we can form a *Host Exposure Map* which captures the host-port combinations of the connections in which the host generally involves.

Data should be obtained by initially training the system and capturing the pattern.

Any activity on the host which doesn’t fall in the Exposure Map can be reported.
Detection by dialog co-relation

- The victimized host goes into specific states during interaction with master
- The dialog co-relation engine sits at the perimeter of the network and make use of the services of *Intrusion Detection Systems (IDS)*

![Diagram](image)

**S1:** Inbound scan  
**S2:** Attempt an exploit  
**S3:** Binary download  
**S4:** Establish C&C with master  
**S5:** Outbound scan for victims
First process involves detection of hosts in the network that involve in p2p communication - Statistical Finger printing

Separation of legitimate p2p hosts from the malicious ones - persistence pattern and interaction pattern
1. What are Botnets?
2. Technical Overview
3. Compare and Contrast
4. Detection and Prevention
5. Evolution and Conclusion
Evolution of botnets

(a) Botmaster - a single point of failure

(b) Victims controlled by IRC network

(c) Master and other victims actively scan for recruits

(d) No sophisticated control but easy to maintain Botnet.
Evolution of Botnets

(a) Botmaster sits surreptitiously in a p2p network
(b) No more a Single of failure
(c) Newer devices are part of Botnet
(d) Other protocols other than IRC are also used for C&C
(e) No active scanning of recruits. Trick other users with baits (malicious links, pirated software)
Security begins from personal responsibility.
- Install *security updates* for OS, browser etc promptly
- Don’t visit untrusted links
- Avoid using peer-to-peer software
- *Block JavaScript*
- *Watch your ports* for unexpected inbound and outbound traffic.

http://www.youtube.com/watch?v=SubxMZxhiKo