Botnet Analysis & Defense

Dawn Song
dawnsong@cs.berkeley.edu

What is a botnet?

- An army of compromised hosts ("bots") coordinated via a command and control center (C&C). The perpetrator is usually called a "botmaster."

"A botnet is comparable to compulsory military service for windows boxes"

-- Bjorn Stromberg

Typical (IRC) infection cycle

Bots usually require some form of authentication from their botmaster

Fabian Monrose
Botnet Analysis & Defense

- Study of botnet phenomena
  - How prevalent are botnets?
    - How many botnets are there?
    - What are their sizes?
  - What techniques/tactics do attackers use?
  - What are botnets used for?
  - What are the trends for botnets?

- Detect & defend against live botnets
  - What methods can we devise?

What Methods Can you Design to Study/Measure Botnet Phenomena?

- HoneyX to entice attackers
  - HoneyNet/honeypots
  - Honey email accounts
  - HoneyMonkey
    - Craw the web to find drive-by downloads, etc.

- Botware analysis
  - Gray-box/black-box testing
  - Binary analysis

- Live tracking
  - IRC tracking
  - DNS cache probing

You Can Build a HoneyKingdom in Your Garage

- A local darknet + 14 PlanetLab nodes
  - record ~1 GB of traffic daily
  - over 4000 “unique” binaries over months

- Even easier to set up Honey email accounts
How much botnet traffic is out there?

- From a two week snapshot of total incoming SYN packets to darknet, 27% can be attributed to known botnet spreaders

~20,000 connection attempts every 10 mins

Fabian Monrose

Botware Analysis

- A wide range of technical skills in the botmasters
- Bot software is fairly advanced

<table>
<thead>
<tr>
<th>Utility Software Thread</th>
<th>Frequency (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTP Server</td>
<td>85</td>
</tr>
<tr>
<td>TFTP Server</td>
<td>85</td>
</tr>
<tr>
<td>AV/FW Killer</td>
<td>55</td>
</tr>
<tr>
<td>System Security Monitor</td>
<td>49</td>
</tr>
<tr>
<td>Registry Monitor</td>
<td>41</td>
</tr>
<tr>
<td>Identi Server</td>
<td>31</td>
</tr>
<tr>
<td>CerenetBack Buhodoo</td>
<td>9</td>
</tr>
</tbody>
</table>

Fabian Monrose

IRC-Tracking:

What are botnets being used for?

- Stealing CD keys

```
Stealing CD keys:
```

```
BGR|0981901486!nmavmkmyam@212.91.170.57
PRIVMSG #atta :BGR|0981901486 $getcdkeys
```

```
BGR|0981901486!nmavmkmyam@212.91.170.57
PRIVMSG #atta :Microsoft Windows Product ID CD Key: (55274-648-5295662-23992).
```

```
BGR|0981901486!nmavmkmyam@212.91.170.57
PRIVMSG #atta :[CDKEYS]: Search completed.
```

- Reading a user's clipboard

```
Reading a user's clipboard:
```

```
Ch3m|784318!~zbhibvn@xxx-7CCCB7AA.click-network.com
PRIVMSG ##chem## :~getclip
```

```
Ch3m|784318!~zbhibvn@xxx-7CCCB7AA.click-network.com
PRIVMSG ##chem## :
```

- DDoS someone:

```
DDoS someone:
```

```
devil!evil@admin.of.hell.network.us
PRIVMSG #t3rr0r0Fc1a :!pflood 82.147.217.39
```

```
PRIVMSG #t3rr0r0Fc1a :Packets
```

```
PRIVMSG #t3rr0r0Fc1a :flooding....
```

- Set up a web-server (presumably for phishing)

```
Set up a web-server (presumably for phishing):
```

```
[DeXTeR]!alexo@l85-130-136-193.broadband.actcom.net.il
PRIVMSG [Del]29466 .http 7564 c:\
```

```
[Del]38628!zaazbob@born113.athome233.wau.nl
PRIVMSG _[DeXTeR] :[HTTPD]: Server listening on IP: 10.0.2.100:7564, Directory: c:\.
```

Fabian Monrose
DNS Cache Probing

- Visiting a domain that has a botnet
- Malicious botnet master
- DNS cache probing

Visualizing botnet footprints via DNS Cache Probing

- Infections in 11% of 800,000 DNS domains

Live Botnet Detection & Defense

- Vantage Point
  - Enterprise perimeter/egress point monitoring
    - BotHunter
  - Internet wide-scale monitoring
    - AT&T Wide-scale Botnet Detection & Characterization
BotHunter: Dialog-based Correlation

BotHunter employs an Infection Lifecycle Model to detect host infection behavior.

- Egress point (internal – external)
- Search for duplex communication sequences that map to IL model
- Stimulus does not require strict ordering, but does require temporal locality

BotHunter: Architecture Overview

Bot Hunter Infection Profile:
- Confidence Score
- Victim IP
- Attacker IP List (by confidence)
- Coordination Center IP (by confidence)
- Full Evidence Trail: Sigs, Scores, Ports
- Infection Time Range

Limitations of BotHunter

- Alert generation
  - SLADE: statistical payload anomaly detection engine
    - Evasion?
  - Signature engine
    - E2 rulesets: exploit injection
    - E3 rulesets: download events
    - E4 rulesets: protocol, behavior & payload content signature for IRC & HTTP bot C&C
    - E1 & E5: scan detection
    - Evasion?
- Alert correlation
  - Can’t cover slow attacks
  - Scalability?
Internet Wide-scale Monitoring & Detection (AT&T)

- Identifying suspicious bot machines
  - Spam
  - Scanning
  - DDoS

- Identify candidate controller conversations
  - Identify hubs communicating with many bot machines
  - Identify IRC-like traffic with bot machines

- Analyze candidate controllers

- Limitations?

Comparison of Two Approaches

- Can you apply the AT&T method to enterprise networks?

- Can you apply BotHunter to large ISP networks?

Break Time

- This time we are really going to take a break :-}
A Generic Bot Cycle

How do you generalize the cycle?
1. Recruiting and taking control of bot machine
2. Communicating & obtaining commands through C&C
3. Conducting malicious tasks

Design Your Favorite Bot

- Desired properties
  - Strong survival ability
    - Stealthy
    - Die-hard/Recover/resurrect
  - Slavery
    - Robust communication to master
    - Receive orders ONLY from real-master

How to Achieve Desired Properties in Bot Cycle

- Bot Cycle:
  1. Recruiting and taking control of bot machine
  2. Communicating & obtaining commands through C&C
  3. Conducting malicious tasks

- Desired properties
  - Strong survival ability
    - Stealthy
    - Die-hard/Recover/resurrect
  - Slavery
    - Robust communication to master
    - Receive orders ONLY from real-master
Recruiting and taking control of bot machine (I)

- **Stealthy**
  - Gain control
    - Low rate scanning, polymorphic attacks, etc.
  - Hold control
    - Rootkits, VM-based rootkits
    - Memory-resident only (issues?)
    - Hide in other processes
    - Don’t bother users
- **Die-hard/Recover/resurrect**
  - Patch all the security holes
  - Watch attempts to kill bot & restart

Recruiting and taking control of bot machine (II)

- **Other tricks**
  - Making it hard to analyze bots
    - DoS attacks on analyzers
  - Making it hard to obtain bot footprint
    - Kill harddrive as soon as detecting any attempt to compromise nodes
  - Targeting low profiles
    - Avoid .mil, .gov, etc.

Communicating & Obtaining Commands through C&C---

- **How to Be Stealthy?**
  - Decentralized: e.g., p2p
  - Asynchronous C&C
  - Mimic legitimate communication profile
  - Add randomness in communication (no periodicity)
  - Encryption
  - Stegography
  - Hiding commander
    - Change topology often
    - Anonymous communication
    - Onion routing
    - Dining cryptographer network
  - Covert communication
    - ICMP, one-way communication
  - Ensure minimum loss of information about botnet structure given the loss of a node
Communicating & Obtaining Commands through C&C---
How to Be Robust?
- Very few students discussed this point
- Built-in redundancy
- Self-repairing in routing
- Secure routing
  - Even if some nodes are “compromised”

Conducting Malicious Tasks
- Stealthy
  - Low rate attacks
  - Different parts of botnet carry out different tasks
- Robust
  - Specific to different attacks

How to Defend against Joe’s Favorite Bot?
- Bot Cycle:
  1. Recruiting and taking control of bot machine
  2. Communicating & obtaining commands through C&C
  3. Conducting malicious tasks
- Desired properties
  - Strong survival ability
    - Stealthy
    - Die-hard/Recover/resurrect
  - Slavery
    - Robust communication to master
    - Receive orders ONLY from real-master
Preventing Recruiting and taking control of bot machine

- Does absolute host security solve the problem?
- Educating users?
- Any silver bullet?
  - Hopeless?
  - Bot programs don’t require root
  - With Web 2.0, running third-party code is more prevalent

Detecting & Destroying C&C

- What does it take?
  - Network monitoring for communications with suspicious nodes
    - Bots could deliberately communicate with legitimate nodes to make analysis even more difficult
  - Insider view
    - Doesn’t work for small botnets
- IP addr is not a trust-worthy/long term identifier
  - Will authenticated traffic help?
- How about ISP cutting off offending nodes?
  - Why should ISP do it?

Preventing Bots from Conducting Malicious Tasks

- Ideas?
  - Depending on different tasks
- Different angle
  - Reduce economic incentives
Summary

• Botnets is real, serious, & here to stay

• How to defend against it?
  – No single silver bullet
  – Need many pieces of the puzzle

• Next class
  – Privacy-breaching malware