The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets

Paper by Evan Cooke, Farnam Jahanian, and Danny McPherson

Oleg Ponomarev

Research seminar on trust, security and privacy in the Internet, 8.11.2006

Helsinki, HIIT

Overview

- Bots
- Botnets Today
- Botnets of Tomorrow
- Discussion

Introduction

- DoS
- Internet Worms

disrupt infrastructure

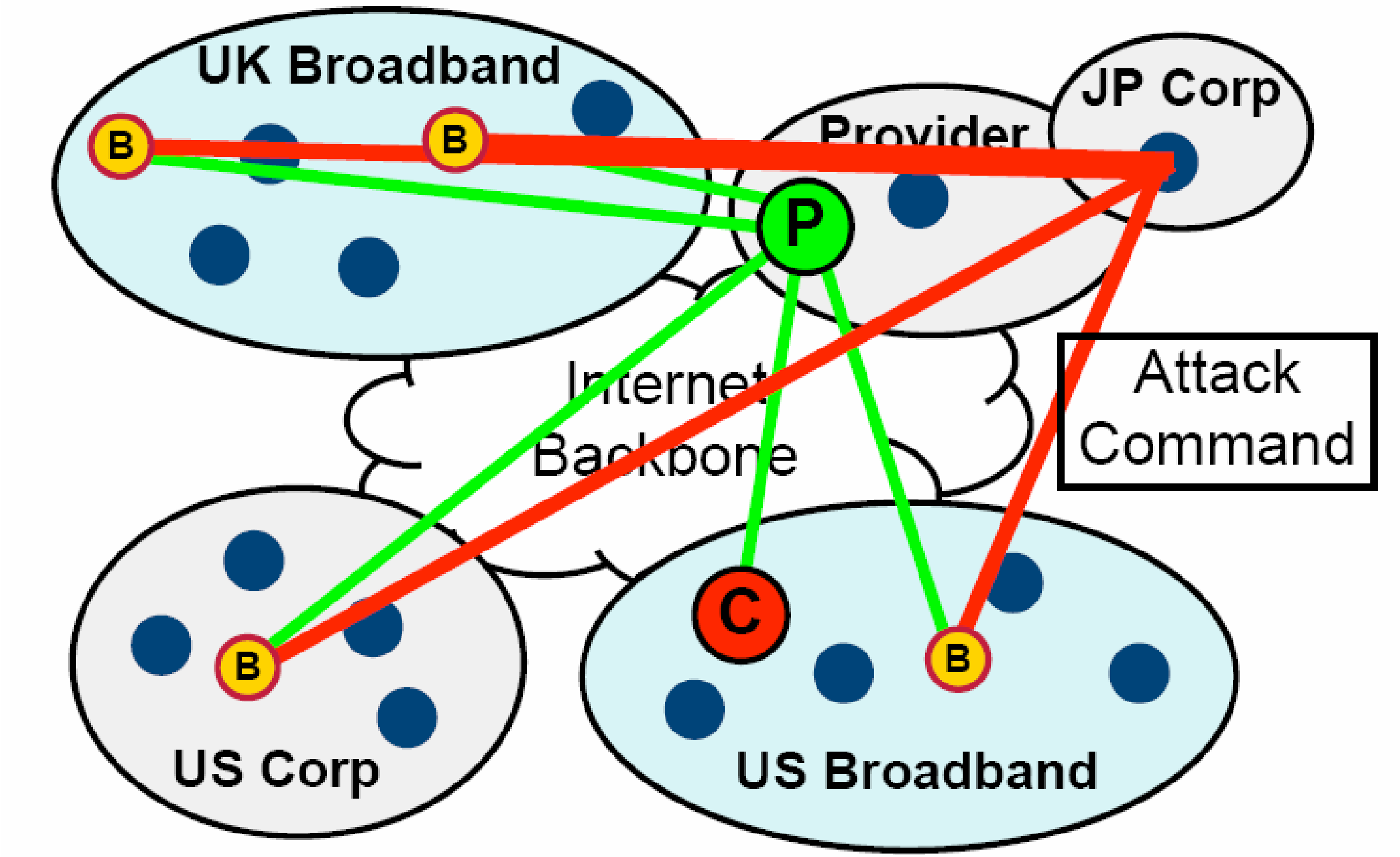
- SPAM
- Phishing
- ID Theft
- Spyware

directly target people

Bots (Rise of the Zombies)

- One of the original uses was to assist in IRC channel management (e.g. Eggdrop created in December 1993, remotely operated, easily used and expanded by scripting)
- Later they became used to attack other IRC users and IRC servers (e.g. by flooding)
- Large targets require more bots. Since very few users provide their resources voluntarily, attackers had to use trojans and other methods to infect computers (SubSeven Bot, Bionet Bot, Attack Bot, GTBot, EvilBot, SlackBot, …)
- Infection became automated: open file shares, Windows vulnerabilities, backdoors left by worms

Botnets Today (Zombie Army)



B – Bots, C – Commander, P – Proxy?

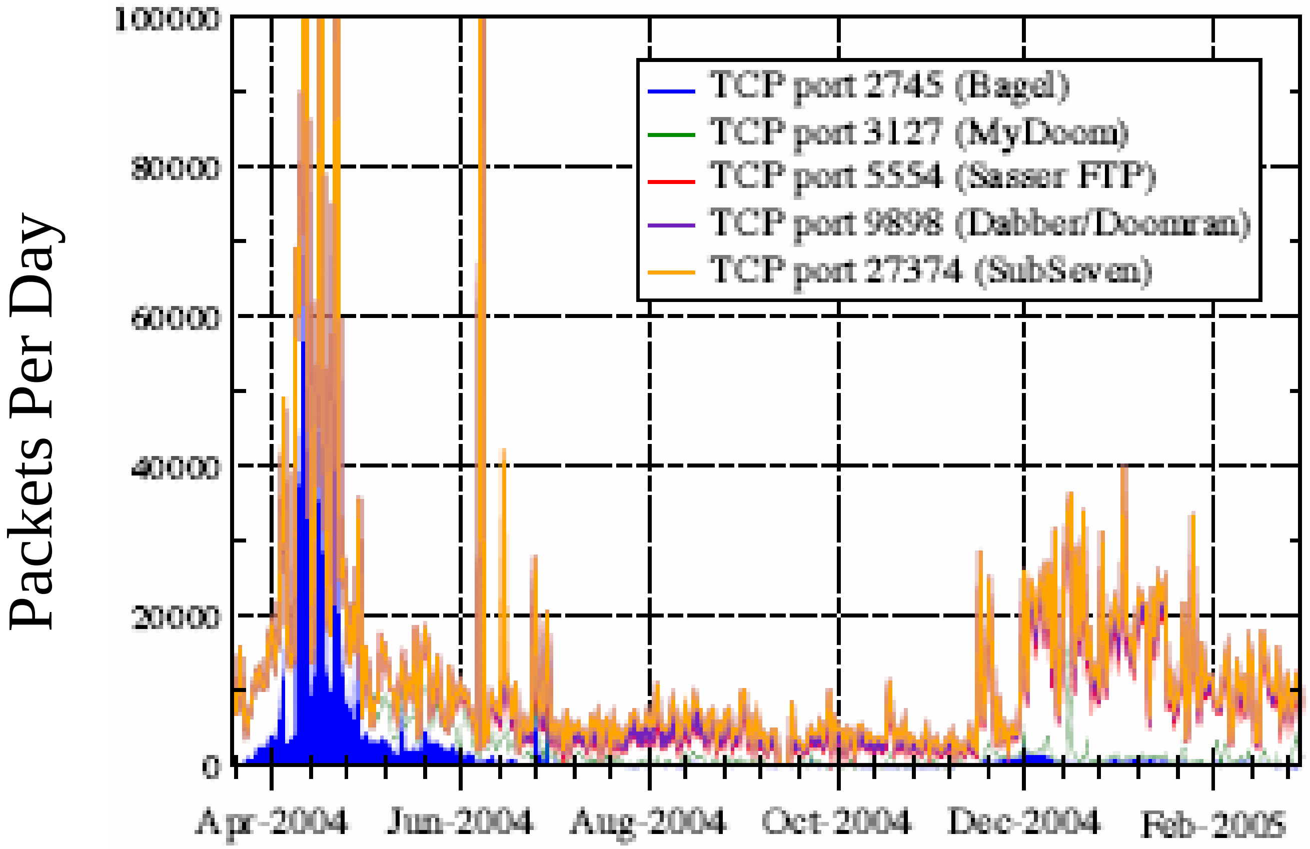
JP Corp – the attack target

Picture courtesy of the article authors

Botnet Propagation

Activity on worm/trojan backdoor ports

Over one year observed at /24 IMS Sensor



Picture courtesy of the article authors

Bot Honeypot

- Windows 2000/XP Honeypot placed behind a bridge:
  - Rate limit traffic 12KB/s
  - Disallow local network
  - 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
  - Only 2 worm (malware without C&C) detections

=> Bots are extremely prevalent

Detecting and Stopping Bots

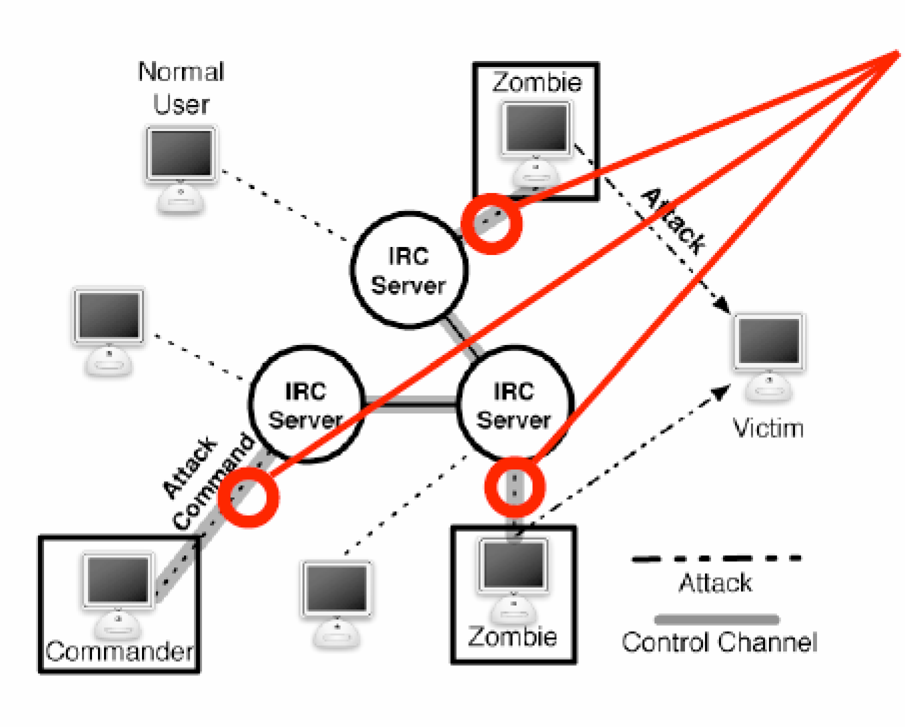
- Prevent systems from getting infected
  (Anti-viruses, Firewalls, Patching)

- Directly detect *bot* communications between *bots* and between *bots* and *bot controllers*

- Detect the secondary features of a *bot* infection like propagation or attacks

Detecting Command and Control

Many bots use IRC for Command and Control



Detect IRC Bot Commands

-Traffic encryption (some versions use SSL)

-Masking with random noise

-C&C communication is extremely flexible!

Botnets of Tomorrow

Topology	Design Complexity	Detectability	Message Latency	Survivability
Centralized	Low	Medium	Low	Low
Peer-to-Peer	Medium	Low	Medium	Medium
Random	Low	High	High	High

Advanced Botnet Detection

- Relying on detecting bot communication is *not* viable in the long term
- Leverage *all* available bot characteristics
- Use data from host detectors, network detectors, or a combination of both
  - System could use network detector to provide an alert on noisy behavior such as scanning or DoS activity
  - Using a host-based monitor, the packets could be correlated with the sending process, and the bot program identified
  - Other C&C channels related to that process could be identified

Challenges of Botnet Disruption

- Once you detect a bot how to shut it down?
- Botnets can be highly distributed, with nodes in hundreds of networks located in many different countries.
- Two goals
  - Take down the bot
  - Take down the botnet

Discussion?

- 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