AN INSIDE LOOK AT BOTNETS

Paul Barford and Vinod Yagneswaran

As Condensed and Augmented by Christo Wilson

Table of Contents

Rationale

- Commercial network security mechanisms are reactive
- While these methods were sufficient in the past, they are quickly becoming ineffective
- Proactive security solutions are the future
- The first step towards building proactive security is understanding the fundamental properties of malicious software

The Code - Architecture

- Agobot/Phatbot – most sophistication family of bot, 20,000 lines of c/c++
  - 20,000 lines of C/C++
  - Monolithic architecture
  - Structured design with tightly controlled set of extensible structures and data types
  - Robust code documentation
  - "Just **grep** the source for RegisterCommand and get the whole command-list with a complete description of all features" – The Honeynet Project

The Code – Architecture (cont.)

- SDBot – simple, compact, GPL
  - 2,000 lines of C
  - Core source presents a simple IRC C&C architecture
  - Vast library of patches enables rolling custom bots to suit the specific needs of the bot-master
  - Patch based extension system provides coder anonymity (limited accountability) unlike controlled, monolithic architectures

The Code – Architecture (cont.)

- SpyBot – a slightly heftier fork of SDBot with pre-applied patches for scanning, exploiting, and DDoSing
- GT Bot – archaic mIRC based bot
  - Collection of mIRC scripts packaged with a hex-edited, cracked copy of mIRC
  - Optionally packaged with extra tools such as proxy servers and rootkits
  - No overall design specification, limited to individually modified instances

The Code – Remote Control

- All evaluated sources relied on IRC channels to communicate
- Agobot – relies on cvar.set and bot.* commands in the channel to change bot variables and execute behavior
  - New versions (Phatbot) include stripped down WASTE P2P connectivity [LURHQ]
- SDBot – listens for PRIVMSG, TOPIC IRC, and NOTICE messages
- SpyBot – subset of SDBot commands
- GT Bot – simplest IRC driven command language, high dependent on implementation version

The Code – Host Control

- Agobot – robust set of harvesting and patching commands
  - Commands to locate sensitive information inculding e-mails addresses, cd-keys, AOL passwords, Paypal passwords, etc
  - Remote registry access
  - Control over local filesystem, including download and execute capabilities
  - Process viewing and obstruction
  - Keylogger and network traffic sniffer based on pcap
  - Patches for common vulnerabilities such as RPC-DCOM (Blaster)
  - Closes open NetBIOS shares

The Code – Host Control (cont.)

- SDBot – limited to basic remote execution and information gathering
- SpyBot – similar functionality to Agobot (including the dangerous ability to flash the keyboard lights!!!)
- GT Bot – extremely limited base feature set; custom variants include expanded feature sets

The Code - Propagation

- Mainly comprised of horizontal (single port, ip-range) or vertical (single ip, port range) scans
- Agobot – scans across network prefix ranges or random addresses
  - Bots can be assigned specific network ranges
- SDBot – base version includes no propagation mechanisms
  - Variants do include, including some that can accept address ranges
- Spybot – limited to H and V scans of NetBIOS shares
- GT Bot – limited to H and V scans coupled with custom exploit programs

The Code – Exploits and Attacks

- Agobot – far reaching built in set of exploits and attacks
  - Includes a robust library of built in exploits to leverage (Dcom, Dameware, Radmin)
  - Can spread across common P2P networks like KaZaa, Grokster, and BearShare [Wikipedia]
  - NetBIOS support
  - Can automatically spread via previously installed open-door Trojan horses (Bagle, MyDoom, etc)
  - Password brute forcers for MS-SQL and Windows
  - Seven types of DDoS attacks: udp flood, syn flood, http flood, targa3, wonk flood, phat syn flood (?), ICMP flood

The Code – Exploits and Attacks

- SDBot
  - support for rudimentary UDP and ICMP floods
  - no built in exploits
- SpyBot
  - UDP, ICMP, and SYN flood support
  - NetBIOS attacks
- GTBot
  - Varies from version to version
  - Authors copy included ICMP flood and Dcom exploit attacks

The Code – Malware Delivery

- GT/SD/SpyBot all deliver exploit and malware simultaneously in a single package
- Agobot separates exploit from delivery
  - Exploit is used to open a remote shell
  - Shell is then instructed to download the payload via HTTP or FTP
- Agobot includes a shell encoder to obfuscate assembly and remove null bytes
  - Uses simple XOR encryption
  - Defeats or at least significantly complicates signature based detection

The Code - Obfuscations

- Agobot – includes a limited polymorphic engine with four different encoders (new versions have six [Wikipedia])
  - Polymorphic engine also tied to shell code encryption routine
- Other bots lack obfuscations
- No bot uses TCP obfuscation techniques (packet re-ordering attack)

The Code - Deceptions

- Agobot is the only bot with a consistent set of deception mechanisms
  - Some rootkit like measures for hiding processes and files
  - Anti debugging measures against OllyDebug, SoftIce and procdump
  - Tests for VMWare emulation
  - Attacks against common anti-virus applications via code injection
  - Remapping of anti-virus and update server DNS entries to localhost

Summary of Findings

- Botnet architecture is robust, modular
  - Facilitates extension (bad) and automated analysis (good)
- IRC is still the primary method of C&C (circa 2006)
  - Firewalls and traffic monitors will remain effective until Agobot maintainers read this paper
- Information harvesting capabilities of bot software would make the average marketer drool
  - Encryption of sensitive data on the desktop needs to be mandatory, not optional
- Exploits galore
  - Patch your box, or just buy software that was well written in the first place (ahem)

Summary of Findings (cont.)

- Ubiquitous DoS capabilities
  - Authors say availability of mechanisms should steer mitigation development. Yeah, whatever.
- Shell encoding and packing mechanisms are widespread, polymorphism is not
  - The AV industry can rest on it's laurels for the time being… or can it?
- Many bots include sophisticated methods for alluding detection
  - Better hope nobody circumvents PatchGuard on Vista (whoops, too late)
- Limited set of propagation algorithms
  - For now, modeling propagation is easy, until Agobot maintainers get around to reading Paxson's Flash Worm paper

The Code - Services

- Agobot includes several built in servers
  - Socks4 proxy
  - HTTP/HTTPS proxy
  - GRE redirect (protocol tunnel)
  - TCP port redirect
- Also interesting: Agobot http.visit command for committing click fraud

Conclusion

- Authors advocate source analysis and dynamic profiling of executables to evaluate malware and construct appropriate defensive measures
- Bots include a diverse array of information gathering and deception mechanisms
- C&C and propagation methods remain underdeveloped

THANKS FOR LISTENING

Now discuss!

Sources

- Paul Barford and Vinod Yagneswaran. An Inside Look at Botnets, In *Special Workshop on Malware Detection, Advances in Information Security, Springer*, 2006.
- Agobot (computer worm) on Wikipedia. http://en.wikipedia.org/wiki/Agobot
- Phatbot Trojan Analysis – LURHQ Threat Intelligence Group. 15 March 2004. http://www.lurhq.com/phatbot.html
- Paul Bacher, Thorsten Holz, Markus Kotter, Georg Wicherski. Know Your Enemy: Tracking Botnets. The Honeynet Project & Research Alliance. 13 March 2005. http://www.honeynet.org/papers/bots/