Botnets

- Networks of compromised machines under the control of hacker, “bot-master”
- Used for a variety of malicious purposes:
  - Sending Spam/Phishing Emails
  - Launching Denial of Service attacks
  - Hosting Servers (e.g., Malware download site)
  - Proxying Services (e.g., FastFlux network)
  - Information Harvesting (credit card, bank credentials, passwords, sensitive data.)

After resolving the IP address for the IRC server, bot-infected machines CONNECT to the server, JOIN a channel, then wait for commands.

The botmaster sends a command to the channel. This will tell the bots to perform an action.

The IRC server sends (broadcasts) the message to bots listening on the channel.
The bots perform the command. In this example: attacking / scanning CNN.COM.

Unfortunately, the detection, analysis and mitigation of botnets has proven to be quite challenging.

- Supported by a thriving underground economy
  - Professional quality sophistication in creating malware codes
  - Highly adaptive to existing mitigation efforts such as taking down of central control server.

Traditional botnet communication
- Central IRC server for Command & Control (C&C)
- Single point of mitigation:
  - C&C Server can be taken down or blacklisted

Botnets with peer to peer C&C
- No single point of failure.
  - E.g., Waldedac, Storm, and Nugache
- Multi-layered Architecture to obfuscate and hide control servers in upper tiers.

Multi-Layered Command and Control Architecture Through P2P
- Each Supernode (server) publishes its location (IP address) under the key 1 and key 2
- Subcontrollers search for key 1
- Subnodes (workers) search for key 2 to open connection to the Supernodes
  - Asynchronous C&C

Current Approaches to Botnet
- Virus Scanner at Local Host
  - Polymorphic binaries against signature scanning
  - Not installed even though it is almost free
  - Rootkit
- Network Intrusion Detection Systems
  - Keeping states for network flows
  - Deep packet inspection is expensive
  - Deployed at LAN, and not scalable to ISP-level
  - Requires Well-Trained Net-Security SysAdmin
Conficker infections are still increasing after one year!!!

There are millions of computers on the Internet that do not have virus scanner nor IDS.

Botnet Enumeration Approach

- Used for spam blocking, firewall configuration, DNS rewriting, and alerting sys-admins regarding local infections.
- Fundamentally differs from existing Intrusion Detection System (IDS) approaches
  - IDS protects local hosts within its perimeter (LAN)
  - An enumerator would identify both local as well as remote infections
- Identifying remote infections is crucial
  - There are numerous computers on the Internet that are not under the protection of IDS-based systems.

How to Enumerate Botnet

- Need to know the method and protocols for how a bot communicates with its peers
- Using sand-box technique
  - Run bot binary in a controlled environment
  - Network behaviors are captured/analyzed
- Investigating the binary code itself
  - Reversing the binary into high level codes
  - C&C Protocol knowledge and operation details can be accurately obtained

Simple Crawler Approach

- Given network protocol knowledge, crawlers:
  1. collect list of initial bootstrap peers into queue
  2. choose a peer node from the queue
  3. send to the node look-up or get-peer requests
  4. add newly discovered peers to the queue
  5. repeat 2-5 until no more peer to be contacted
- Can't enumerate a node behind NAT/Firewall
- Would miss bot-infected hosts at home/office!

Passive P2P Monitor (PPM)

- Given P2P protocol knowledge that bot uses
- A collection of “routing-only” nodes that
  - Act as peer in the P2P network, but
  - Controlled by us, the defender
- PPM nodes can observe the traffic from the peer infected hosts
- PPM node can be contacted by the infected hosts behind NAT/Firewall

Crawler and Passive P2P Monitor (PPM)
DHCP
NATs
Non-uniform bot distribution
Churn
Most estimates put size of largest botnets at tens of millions of bots
Actual size may be much smaller if we account for all of the above

Botnets use a lot of newly-created domains for phishing and malware delivery
Fast flux: changing name-to-IP mapping very quickly, using various IPs to thwart defense attempts to bring down botnet
Single-flux: changing name-to-IP mapping for individual machines, e.g., a Web server
Double-flux: changing name-to-IP mapping for DNS nameserver too
Proxies on compromised nodes fetch content from backend servers

Advantages for the attacker:
- Simplicity: only one back end server is needed to deliver content
- Layers of protection through disposable proxy nodes
- Very resilient to attempts for takedown

Look for domain names where mapping to IP changes often
- May be due to load balancing
- May have other (non-botnet) cause, e.g., adult content delivery
- Easy to fabricate domain names
Look for DNS records with short-lived domain names, with lots of A records, lots of NS records and diverse IP addresses (wrt AS and network access type)
Look for proxy nodes by poking them

They have been known to fight back
- DDoS IPs that poke them (even if low workers are scanned)
They have been known to fabricate data for honeynets
- Honeynet is a network of computers that sits in otherwise unused (dark) address space and is meant to be compromised by attackers