A Survey of Botnet Technology and Defenses

Michael Bailey, Evan Cooke, Farnam Jahanian, Yunjing Xu, Manish Karir

Cybersecurity Applications & Technology Conference for Homeland Security (CATCH 2009)

Presented by Gaspar Modelo-Howard

PURDUE UNIVERSITY

Objective

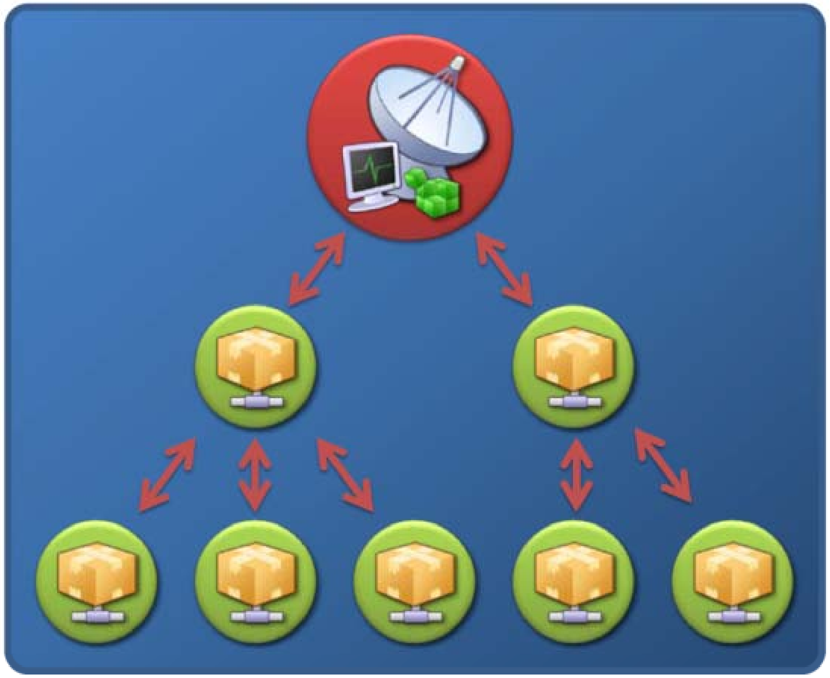
- Provide a **brief look at** how *existing botnet research*, *evolution and future of **botnets***, as well as the *goals and visibility of today's networks* intersect

Agenda

- Botnets
- Data Sources
- Research Studies
- Conclusions

Botnets (1)

- A botnet consists of:
  - Zombies: Pool of compromised computers
  - Bot: Software to enable operator to remotely control zombies
    - Bots are a hybrid of previous threats (virus, worms)
    - Its construction is (usually) a cooperative effort
- Predominant in today's networks and can be very large (100K)



Botnets (2)

- Design requirement 1: how to make owners "accept" usage of computers for malicious purposes
- Botnet attackers have migrated from
  - Single, manual propagation method to multiple automated propagation
  - Random scanning to robust "hitlists"
  - Vulnerable services to "vulnerable" users (social engineering)

Table 1. Propagation Mechanisms

Propagation Methodology		Design Complexity	Detectability	Propagation Speed	Population Size
Exploit:	Operating System	*Medium*	*High*	*Low*	*High*
	Services	*Medium*	*Medium*	*Medium*	*Medium*
	Applications	*High*	*Low*	*High*	*Low*
Social Engineering		*Low*	*Medium*	*Low*	*High*

Botnets (3)

- Design Requirement 2: how to communicate with each bot instance without being detected
- Three botnet topologies identified:
  - Centralized: central point forwarding messages between clients, low latency, easier to detect, central location can compromise whole system
  - P2P: no central point/hierarchy, harder to disrupt, more complex design, no delivery or latency guarantees
  - Unstructured: completely random P2P, messages encrypted, random Internet scan, simple design, high latency, no delivery guarantee

Botnets (4)

- Design Requirement 2: how to communicate with each bot instance without being detected

Table 2. Command and Control Topologies

Topology	Design Complexity	Detect-ability	Message Latency	Surviv-ability
Centralized	*Low*	*Medium*	*Low*	*Low*
Peer-to-Peer	*Medium*	*Low*	*Medium*	*Medium*
Unstructured	*Low*	*High*	*High*	*High*

Botnets (5)

- Design Requirement 3: how to extract value from a bot infected node
  - Attackers moving from DoS attacks, punish IRC users or gain status to create value and even extract real monetary gain
    - Agobot can initiate DDoS attacks
    - SDBot includes advanced key logging techniques
    - Storm botnet has interface for conducting Spam campaigns

Table 3. Attack Classes

Topology	Detectability	Design Complexity	Attack Value
Single Host DDoS	*High*	*Low*	*Low*
Multi Host DDoS	*Medium*	*Medium*	*Medium*
Identity Theft	*Low*	*High*	*Medium*
Spam	*Medium*	*Medium*	*High*
Phishing	*Medium*	*High*	*Medium*

PURDUE UNIVERSITY

Data Sources

- Issues of data sources available according to botnet detection and mitigation
  - Service provider networks: notification of malicious activity
  - Enterprise networks: cleaning hosts, preventing spread
- Types of Data
  - DNS: data to/from servers/resolvers to detect attack/communication behavior (spam)
  - Netflow: sampling traffic flows, identifies comm patterns and attacks, limited visibility
  - Packet Tap: switch/tap deployment, finer granularity, higher cost, encryption reduces visibility
  - Address Allocation: Identifies reconnaissance behavior, visibility generally reserved for enterprises
  - Host: wealth of info available, avoids visibility issues but faces scalability ones
  - Honeypot: insight into means and motives, does not involve production hosts, difficult for social engineering attacks

Research: Detection Techniques (1)

- Detection via cooperative behaviors
  - Bothunter: models bot infection phase to compare suspected events
  - Botsniffer: statistical algorithms to detect botnets using centralized topology
  - Botminer: extends Botsniffer, detection framework performing clustering C&C comm and malicious activities and cross-correlation on them
  - Karasaridis et al.: detection scheme to calculate distances between monitored flow data and pre-defined IRC traffic flow model
  - Akiyama et al.: three metrics to determine botnets cooperative behavior (relationship, response, synchronization)
  - Strayer et al.: temporal correlation algorithm in five-dimensional space about packet inter-arrival time and size
  - Chois et al.: studied anomaly group activities of botnets in DNS traffic
  - Ramachandram et al.: discovered identities of bots based "reconnaissance" lookups to determine bots' blacklist status

Research: Detection Techniques (2)

- Detection by signatures
  - Goebel et al.: used regular expressions, n-gram analysis and scoring systems to detect bots' conversations
  - Binkley et al.: grouped IP hosts in IRC channels with IP scanning activities to determine if they were malicious
- Detection of attack behaviors
  - Brodsky et al.: relied on behavior of botnets (send large number of data in short period of time) to detect spam
  - Xie et al.: used spam server traffic properties and spam payload to construct spam signature generation framework

Research: Detection Techniques (3)

- Detection via cooperative behaviors
- Detection by signatures
- Detection of attack behaviors

Table 4. The relationship between the network visibility, the botnet invariant behaviors, and various proposed techniques

		Bot Behaviors		
		Propagation	**Communication**	**Attack**
Data Sources	**Traffic Flows**	scan-detection [14, 15, 13, 3, 18, 26] binary-downloading-detection [14, 15, 13, 26]	control-protocols [14, 15, 13, 11, 3] [18, 1, 26]	ddos-detection [18, 1, 26] spam-detection [15, 13, 18, 4, 28] active-responder [25]
	Darknet Data	bot-informants [14, 13] scan-detection [14, 13]	bot-informants [14, 15, 13]	bot-informants [13]
	Packet Capture	vulnerability-signature [14]	control-signatures [18, 1, 11, 3]	
	DNS Logs		rendezvous-detection [18, 5]	spam-detection [15, 13, 4] reconnaissance-detection [24] active-responder [25]

Research: Measurement Studies (1)

- Size Estimation
  - Rajab et al. observed botnets using DNS, IRC, passive methods
  - Zhuang et al. grouped spam-generating bots by examining spam contents
  - Rajab et al. considered discrepancies in botnet size estimation
- Behavior Analysis
  - Gianvecchio et al. proposed two types of classifiers (entropy rate and ML) to differentiate human and IRC bots
  - Anderson et al. focused on scam hosting infrastructure and how it is shared
  - Dagon et al. noted time zones and locations play a critical role in malware propagation

Research: Measurement Studies (2)

- Peer-to-peer botnets
  - Grizzard et al. provided a history and overview of P2P botnets
  - Holz et al. presented case study on Storm with details on system and network-level behaviors
  - Kanich et al. estimated Storm botnet size by considering various types of noise (protocol aliasing)
  - Wang et al. proposed a hybrid (centralized and P2P) structured botnet that overcame individual disadvantages

Conclusions

- Botnets are moving targets
  – All aspects of life-cycle (propagation, C&C, and attacks) are evolving constantly
- No technique is perfect
  – Each detection algorithm has a set of tradeoffs (FP and FN)
- All networks are not the same
  – Different networks have different goals, visibility of botnet behaviors and data sources
- A successful botnet detection/mitigation solution should address these realities and their interactions with each other