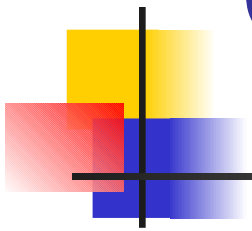
A Framework for Understanding Botnets

Justin Leonard, Shouhuai Xu, Ravi Sandhu

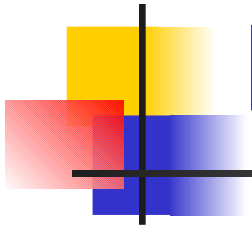
University of Texas at San Antonio

Overview



- Botnet Lifecycle
- Botnet Architecture
- Command and Control Mechanisms (C&C)
- Dynamic Graph Model
- Botnet Attributes

Botnet lifecycle

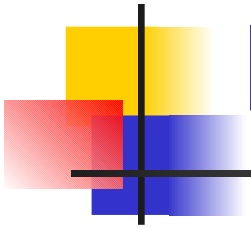


Formation – master compromises, recruits vulnerable machines, and assigns roles.

Command and Control (C&C) – master sends messages to bots

Attack – Bots launch attacks

Post-attack – bots are detected, cured, and new bots recruited.



Botnet Architecture

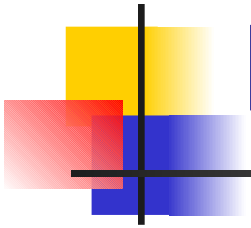
What roles are present in a botnet?

Master – human attacker(s)

Controllers – coordinates subset of bots, long term asset

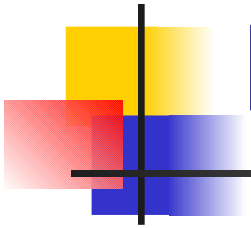
Intruders – disposable, high-risk of detection, may downgrade into a bot

Bots – responsible for attacks



Botnet C&C Mechanisms

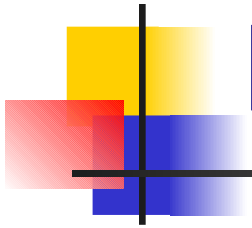
- Anonymous Channels
  - Sender anonymous channels
- Secret Handshakes
  - Privacy-preserving authentication
  - PKI-like infrastructure or group signatures
- Gossiping
  - Small fan-out of neighbors



Dynamic Graph Model

- Directed graph representation
- Vertex set represents bots
- Edge set represents “knows” relation – e.g., *(u,v)* implies *u* can spontaneous communication with *v*.
- Does capturing *u* imply exposure of *v?*
- Undirected graph is special case

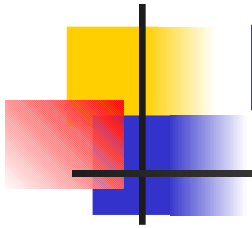
Dynamic Graph Model



Directed graph represents snapshot of graph over time.

Captures real network behavior – e.g., offline machines, detected and cured bots.

Implies attributes should be modeled as Random Variables instead of deterministic numbers.



Botnet Attributes

Robustness

Resilience

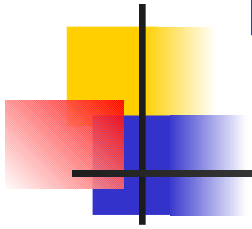
Sustainability

Exposedness

Bandwidth Consumption

Botnet Firepower

Robustness

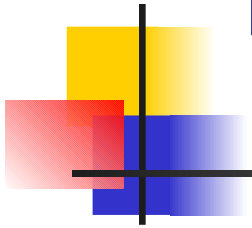


Minimum number of detections to trace every bot.

Random variable over time

Represents weakest or “best case” detection by defender

Resilience



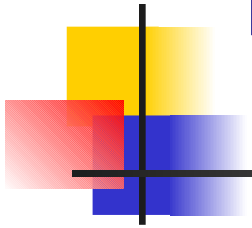
Captures consequence of exposure of a set of bots

Tracing uses “knows” relationship

Normalized by size of botnet

Intuitively captures how much a defender can achieve with fixed resources (e.g., subpoenas).

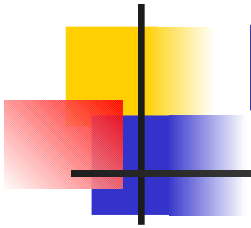
Resilience vs Robustness



Robustness establishes minimum number of captures, resilience the effects of a capture – the resilience for the corresponding robustness set is 0.

A set smaller than the robustness cannot capture all bots.

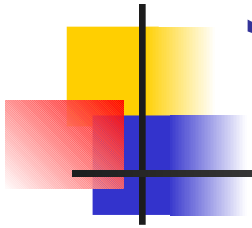
Known to attack *a priori*, defender has limited knowledge.



Dynamic Graph Model

- Directed graph representation
- Vertex set represents bots
- Edge set represents “knows” relation – e.g., *(u,v)* implies *u* can spontaneous communication with *v*.
- Does capturing *u* imply exposure of *v?*
- Undirected graph is special case

Sustainability

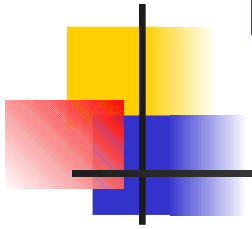


Captures effects of interactions between attacker and defender.

Uses a definition based on number of connected bots.

Reliability from the attacker's perspective against a “malicious” defender.

Exposedness

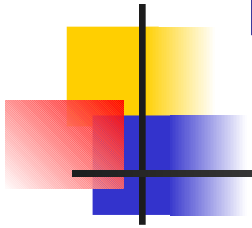


Worst-case probability a bot is detected by defender due to C&C.

Captures the effectiveness of the defenders IDS.

May be used to determine resilience set by using a “detection threshold”, above which we assume a bot is detected.

Bandwidth Consumption

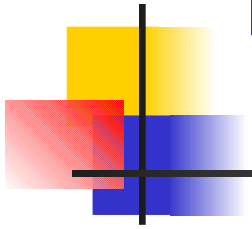


Captures the efficiency of the C&C mechanisms.

Gives an intuitive measure of the “noisiness” of the botnet.

Whole system point of view, as opposed to exposedness, which captures probability of detecting a particular bot based on C&C messages.

Botnet Firepower

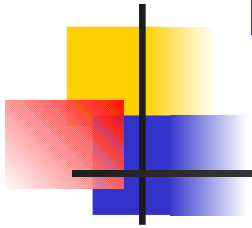


Captures the overall effectiveness of the botnet at launching an attack.

Simple measure is the size of the botnet.

Perhaps also weighted by available resources.

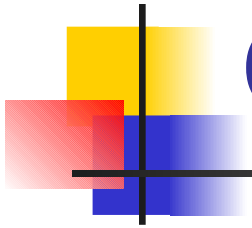
Future Research



Tying definitions to existing botnet case studies.

What strategies are effective at maximizing particular metrics?

Can we quantitatively compare attributes relative to a given defender capability?



Questions?