Peer-to-Peer Botnets Taking Over the New Voronoi-Overlay

J.J. Montgomery
REU 2010 USU

Abstract

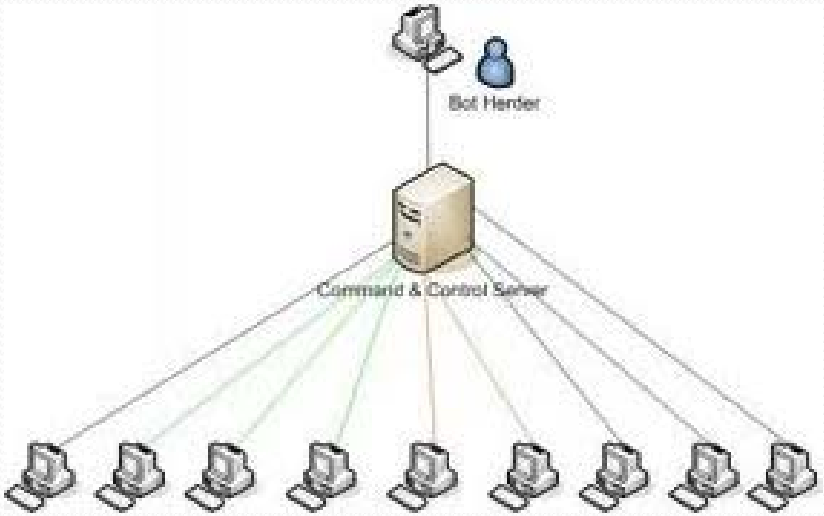
- Most significant threat on Internet is P2P botnets
- Security community always one step behind botnets
- Voronoi overlay is being researched for use in:
  - Networked Virtual Environments (NVEs) and
  - Massively Multiplayer Online Games (MMOGs)
- Much research has been done into P2P botnets and the Voronoi overlay
- Nothing has been published regarding how a Voronoi-based network could be affected by a new botnet
- This project proposes a modification to a current Voronoi-based NVE simulator to study the effects of a botnet infestation using this new technique.

Intro

- Botnets
- Peer-to-Peer (P2P) Botnets
- Voronoi Overlay for P2P Networks
- Related Works
- Proposed Approach
- Results

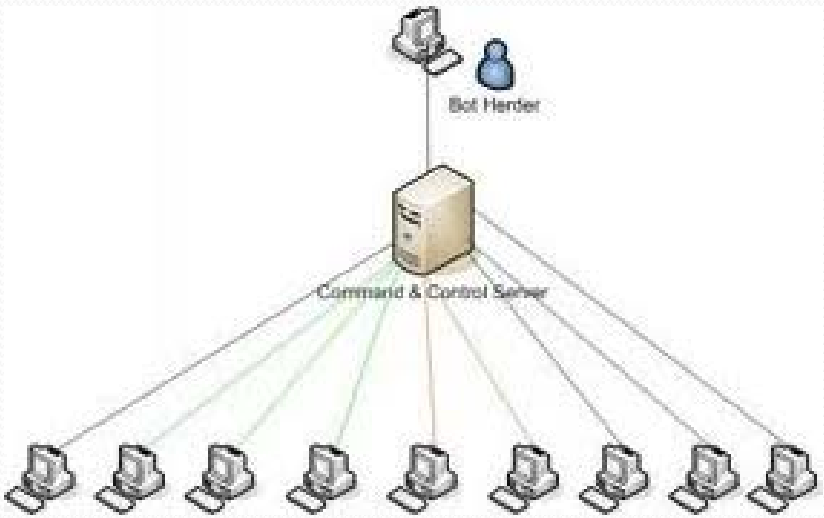
Botnets

- Origins from IRC
- Controlled by a "Botmaster" or "Bot Herder"
- Used for:
  - Spamming
  - Identity theft
  - Click fraud
  - DDoS attacks



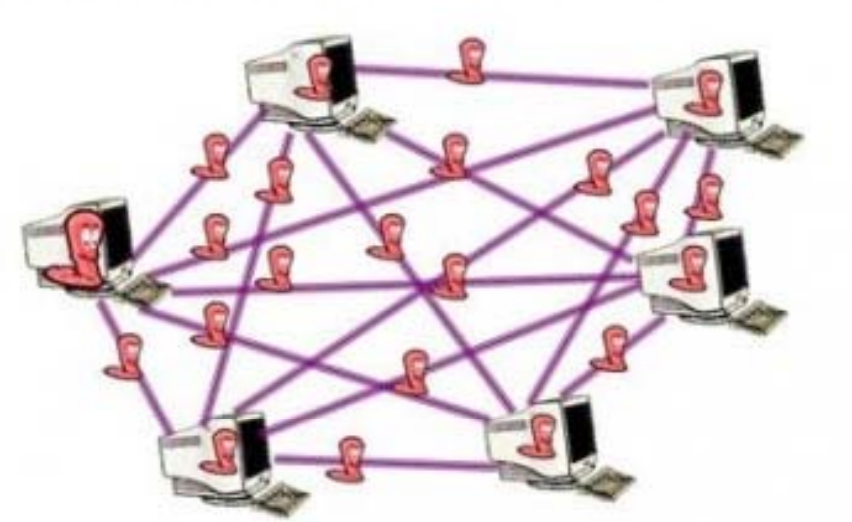
Botnets

- Most-threatening malware on the Internet
- Old Command & Control Model
  - Stems from IRC origins
  - Single point of failure (server)
  - Easily mitigated



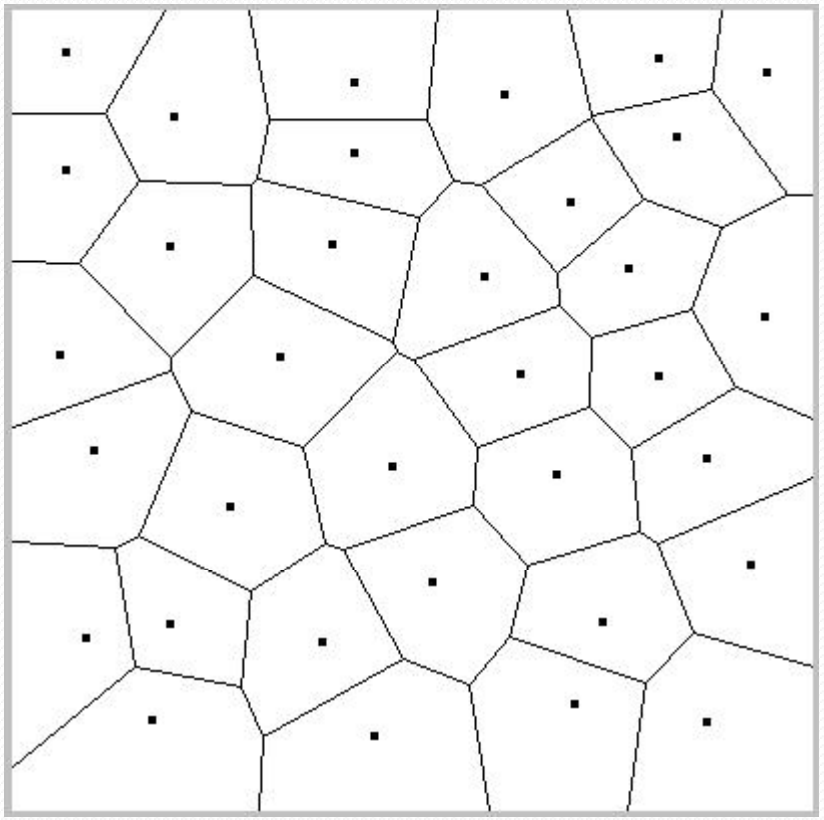
P2P Botnets

- Latest evolution
- Stronger
  - No single point of failure
  - Robust
  - Harder to take down
- Have already been seen in the wild:
  - Phatbot
  - Peacomm



Voronoi-Overlay for P2P Networks

- Voronoi Diagram:
  - Mathematic construct to the right ->
- Can be applied to P2P networks to distribute processing required for Networked Virtual Environments (NVEs) and Massively Multiplayer Online Games(MMOGs)



Related Works – Botnets

- Botnet classification and case studies:
  - "The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets" – Cooke, Jahanian, McPherson
  - "Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm" – Holz, Steiner, Dahl, Biersack, Freiling
  - "Peer-to-Peer Botnets: Overview and Case Study" – Grizzard, Sharma, Nunnery, Dagon
  - "Botnet: Classification, Attacks, Detection, Tracing, and Preventive Measures" – Liu, Xiao, Ghaboosi, Deng, Zhang

Related Works – Botnets

- Detecting Botnets:
  - "BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation" – Gu, Porras, Yegneswaran, Fong, Lee
  - "Detecting Covert Botnets Using Communication Patterns" – Sorensen, Sorensen, Feuz, Grigoriy, Kerzhner, Mano
  - "BotGAD: Detecting Botnets by Capturing Group Activities in Network Traffic" – Choi, Lee, Kim
  - "P2P Botnet Detection using Behavior Clustering & Statistical Tests" – Chang, Daniels

Related Works – Botnets

- Attacking Botnets:
  - "Your Botnet is My Botnet: Analysis of a Botnet Takeover" – Stone-Gross, Cova, Cavallaro, Gilbert, Szydlowski, Kemmerer, Kruegel, Vigna
- Forward-looking
  - "Overbot – A botnet protocol based on Kademlia" – Starnberger, Kruegel, Kirda

Related Works – Voronoi

- "Scalable Peer-to-Peer Networked Virtual Environments" – Hu, Liao
- "Voronoi-based Adaptive Scalable Transfer revisited" – Backhaus, Krause
- "Enhancing Neighborship Consistency for Peer-to-Peer Distributed Virtual Environments" – Jiang, Chiou, Hu
- "A Forwarding Model for Voronoi-based Overlay Network" – Chen, Lin, Chen, Hu
- Dean Mathias's hybrid P2P Networked Virtual Environments simulator "Aubrey"

Proposed Approach

- Modify Dean's program to simulate network traffic
- The current simulation runs on a single computer through a single application (no network traffic)
- With simulated network traffic, one can introduce bots into the Voronoi-based network and observe the results
- This information can then be used to incorporate better defenses against botnets into the Voronoi network

Proposed Approach

- "Aubrey" is written in C#
- Over 3,000 lines of code
- Network traffic will be simulated through the use of sockets
  - Each simulated "peer" will listen on a specific port for messages from its neighbors
  - The sending peer will connect to the receiving peer's socket and send the message to that peer's port at 127.0.0.1 (localhost)

Results

- Initially the task seemed much simpler than it actually was
- Modified the following classes in Dean's code:
  - class PeerPorts (modified PeerRandom)
  - class PeerFactoryPorts
  - class ScenarioPorts
- Major hurdles:
  - Learning C#
  - Large code base

Conclusions

- Further modifications will need to be made to introduce “infected” peers
- Network data will be collected to determine the impact and characteristics of the bots on the network
- This information can then be used to build in better defenses for a Voronoi-based network

Questions / Comments