Revealing Botnet Membership using DNSBL Counter-Intelligence

Nick Feamster
Anirudh Ramachandran
David Dagon
Georgia Tech
Motivation for this work

- **Fact**: Bot-herds advertise and sell their “clean” bots at a premium

- **Insight**: If the claims are true, they must be looking up their bots’ status in some blacklist!

- **Opportunistic Application**: Might it be possible to mine DNS Blacklist *queries* to reveal such *reconnaissance* activity?
Detecting Reconnaissance

- **Key Requirement**: Distinguish reconnaissance queries from queries performed by legitimate mail servers

- **Our Approach**: Develop heuristics based on the spatial and temporal properties of a DNSBL Query Graph

- We focus (mostly) on spatial heuristics
Legit Queries vs. Reconnaissance

- Legitimate queriers are also the targets of queries
- Reconnaissance queriers are usually not queried themselves

$$\text{lookup} \quad \text{mx.b.com}$$

**DNS-Based Blacklist**

**Legit Mail Server A**
mx.a.com

email to mx.b.com

**Legit Mail Server B**
mx.b.com

e-mail to mx.a.com

**DNS-Based Blacklist**

**Reconnaissance host**
Measurement Approach

• Log Spamhaus queries

• Construct querier/queried graph

• Prune graph: only nodes in the Bobax trace

• Examine nodes with high out-degree
  – **Hypothesis:** targets of nodes with high out-degree likely bots
Applying the Spatial Heuristic

• Construct the directed DNSBL Query Graph $G$

\[ A \xrightarrow{\text{lookup } B} \text{DNSBL} \xrightarrow{} \text{Add } E(A, B) \text{ to } G \]

• Extract nodes (and their connected components) with the highest values of the spatial metric $\lambda$, where $\lambda = \text{(Out-degree/In-degree)}$
Third-Party Reconnaissance

- *Third-party performs reconnaissance query*

- Relatively easy to detect using the spatial metric
Other Techniques

• Self-Reconnaissance
  – Each bot looks itself up
  – This should not happen normally (at least, not *en-masse*) – thus, easy to detect

• Distributed Reconnaissance
  – Bots perform lookups for other bots
  – Complex to deploy and operate
  – *We witnessed evidence of this technique*
Distributed Reconnaissance

- The botmaster, on behalf of the bots
- The bots, on behalf of themselves
- The bots, on behalf of each other

Known bobax drone!

Implication: Use a “seed” set to bootstrap?

<table>
<thead>
<tr>
<th>ASN of Node</th>
<th>Out-degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Everyone’s Internet (AS 13749)</td>
<td>36,875</td>
</tr>
<tr>
<td>IQquest (AS 7332)</td>
<td>32,159</td>
</tr>
<tr>
<td>UUNet (AS 701)</td>
<td>31,682</td>
</tr>
<tr>
<td>UPC Broadband (AS 6830)</td>
<td>26,502</td>
</tr>
<tr>
<td>E-xpedient (AS 17054)</td>
<td>19,530</td>
</tr>
</tbody>
</table>
Prevalence of Reconnaissance

- **Long tail** – Bot-herds might already have the capability to distribute reconnaissance among many bots.

- **A few high out-degree nodes** – multiple vantage points might help identify “prominent players”
Implications

• Bad news! Bot reconnaissance techniques are pretty advanced

• Good news, too
  – Can use these spatial dependencies to opportunistically identify new bots
Opportunistic Bot Detection

• Many sources of data for *bootstrapping* passive botnet detection (*i.e.*, to compile a ‘seed’ list) like
  – SMTP/Spam logs,
  – Portscan logs from Intrusion Detection Systems

• Knowledge of botnet membership → ability to stop attacks closer to the source

• Multiple vantage points increase confidence and reduce risk of false positives.
Some Problems with Counter-Intel

• Constructing the query graph is intensive
  – Computationally
  – Storage-wise

• Initially pruning the graph with IP addresses of known suspects (e.g., spammers) could help