Entropy Based Worm and Anomaly Detection in Fast IP Networks

Arno Wagner
wagner@tik.ee.ethz.ch

Communication Systems Laboratory
Swiss Federal Institute of Technology Zurich (ETH Zurich)
Outline

- Dataset
- Entropy and Compression
- Observable Effects During Worm Outbreak
- Sampling, Compression Algorithm, Performance
- Summary
The DDoSvax Dataset

Project URL:
http://www.tik.ee.ethz.ch/~ddosvax/

- NetFlow v5
- From all SWITCH border routers
- About 60,000,000 flows/hour
- ~200k internal, ~800k external IPs/hour
- Unsampled
- Stored in full since March 2003
Entropy, Kolmogorov Complexity

- **Entropy:**
  Expected information in an object from a set with a specific selection probability for each element.

- **Kolmogorov Complexity:**
  Information in a specific (binary) object.

Kolmogorov Complexity cannot really be measured. Entropy can be estimated by compression.
Entropy Estimation by Compression

1. Represent data object in binary form
2. Compress
3. Entropy estimation [bit/bit] is

$$\min\left(1, \frac{\text{compressed size}}{\text{original size}}\right) \in ]0, 1]$$

- Relatively bad accuracy (worst case: encrypted data), but not that far off
- Usable for relative comparisons
Effects of Worm Outbreak

Normal traffic:
- Many contact few (servers)
  many contact many (P2P)
- Connections are mostly successful (bidirectional)
  ⇒ Flow set is mostly symmetric

Worm outbreak traffic:
- Few hosts contact many
- Most connections fail
  ⇒ Flow set is asymmetric

Generic properties of any scanning worm!
Compression statistics

Most promising (determined experimentally)

- Source IP
- Target IP
- Source port
- Target port

All 4 fields are converted to host byte-order and compressed individually per measurement interval of, e.g., 5 minutes.
Example 1: Blaster Worm

- First observed August 11th, 2003
- Tries TCP connection to port 135
- Random target selection with local preference
- Initially infected 200’000…500’000 hosts in 8 hours
Blaster: IP Compressibility

Inverse Compression Ratio (lzo)

Date and Time (UTC, 2003)

outbreak
Blaster: Port Compressibility

Inverse Compression Ratio (lzo)

Date and Time (UTC, 2003)

11.08. 08:00 11.08. 12:00 11.08. 16:00 11.08. 20:00 11.08. 24:00 12.08. 00:00 12.08. 04:00

source port

destination port

outbreak

Arno Wagner, ETH Zurich, SPRING 2006 – p.9
Example 2: Witty Worm

- First observed March 20th, 2004
- Infects a firewall product
- Random target selection
- Sends UDP packet with random target and fixed source port
- Initially infected ~12’000 computers in 75 minutes
Witty: IP Compressibility

Inverse Compression Ratio (Izo)

Date and Time (UTC, 2004)

outbreak

source IP

destination IP

Arno Wagner, ETH Zurich, SPRING 2006 – p.11
Sampling: Witty full vs. 5%
Compression Algorithms

- **lzo**: Lempel-Ziv variant
  very fast, bad compression

- **gzip**: Lempel-Ziv variant
  well-known GNU compressor, average in all respects

- **bzip2**: Burrows-Wheeler + Huffman coding
  slow, very good compression

- **Entropy estimation by value frequencies**: average speed, high memory needs
## Resource Comparison

<table>
<thead>
<tr>
<th>Method (Library)</th>
<th>CPU time / hour (Athlon 2800+) (60’000’000 flows/hour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bzip2 (libbz2-1.0)</td>
<td>169 s</td>
</tr>
<tr>
<td>gzip (zlib1g 1.2.1.1-3)</td>
<td>52 s</td>
</tr>
<tr>
<td>lzo1x-1 (liblzo1 1.08-1)</td>
<td>7 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Memory per compressor instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>bzip2</td>
<td>7600 kB</td>
</tr>
<tr>
<td>gzip</td>
<td>256 kB</td>
</tr>
<tr>
<td>lzo1x-1</td>
<td>64 kB</td>
</tr>
</tbody>
</table>
Summary

- Generic approach
- Works for any fast random scanning worm
- Scales linear for CPU and I/O, constant for memory
- Not suitable for slower worms
- Only limited information about worm details
Thank You!