Classifying DNS Heavy User Traffic by using Hierarchical Aggregate Entropy

2012/3/5
Keisuke Ishibashi, Kazumichi Sato
NTT Service Integration Labs
Motivation

• Network resources are consumed by a small number of heavy users.
• Controlling traffic from heavy users is a crucial task for efficient use of network.
  • Filtering, rate limiting, charging.
• Before controlling the heavy user traffic, we need to understand what type of traffic they send.
• If heavy user traffic are mostly anomalous, then filtering such traffic is rather acceptable.
  – Anomalous traffic: DDoS attack, spam, illegal file exchange etc.
• Thus, we need to classify heavy user traffic whether normal or abnormal.
• In this talk, we focus on heavy users in DNS traffic, one of the most important control traffic in the Internet.
Bogus traffic in DNS

- DNS: mainly used for mapping domain name to IP address or vice versa
- Two types of servers: caching server and authoritative server
- Bogus queries are consuming resources of both DNS authoritative servers and caching servers
  - repeated queries for a single name (bug?)
  - scanning queries for non-existing names (worm?)

Motivation cont’d

• Most of bogus queries are sent by small number of heavy clients [toyono]
  – Filtering queries sent by those heavy clients is efficient to protect DNS server resources

![Diagram showing percentages of different types of queries at varying rates]

• However, not all queries from heavy clients are bogus
  • PTR queries from web servers (analog)
  • Aggregated queries from DNS proxies
Normal heavy user

• DNS prefetch
  – resolves all the domain names of the URLs in a browsed web page **before** the URL is actually clicked
  – Faster web but burst (unnecessary) queries for a page that contains huge URLs

• Log analyzer
  – Log analyzers in web server send reverse queries (resolve domain names for IP addresses) for addresses in their access logs
  – What organizations access our web servers?
Entropy based classification

• Needs to classify heavy clients into normal users and abnormal users
  ⇒ Classify heavy clients by their query pattern
• How to capture query patterns?
  ⇒ Use of entropy of queries in domain name spaces

• Entropy of legitimate queries: expected to lie between them
• Does not have information on spatial characteristics
  – Independent on where queries concentrate or diverse in domain name spaces
  – Only depends on how queries concentrate or diverse

• Hierarchical Aggregate Entropy
  – Aggregating queries accordance to its hierarchical structure and calculate entropy for each hierarchy
Hierarchical Aggregate Entropy (2/2)

- **DNS:** tree based hierarchical structure
  - Fully qualified domain name (FQDN): www.google.com, www.ntt.co.jp
    \[ H(D^{(0)} \mid D^{(1)}) \]: deviation in www.example.org level
  - Second level domain (SLD): .google.com
    \[ H(D^{(1)} \mid D^{(2)}) \]: deviation in example.org level
  - Top level domain (TLD): .com, .net, .jp...
    \[ H(D^{(2)}) \]: dispersion in queries for com, .net, .jp...
    \[ H(D^{(0)}) = H(D^{(2)}) + H(D^{(1)} \mid D^{(2)}) + H(D^{(0)} \mid D^{(1)}) \]

Domain Tree  Query Distribution

Identify the deviation occurs intra TLD or inter TLD.
Experimental results

- Calculate entropies of top 10,000 heavy clients for DNS traffic monitored at DNS caching servers.
- Entropies from normal clients concentrated in a specific region.

⇒ Clients whose entropies are out of the region can be expected to be abnormal.
• Extract normal domain by using SVM
  – Training Data: manually labeled data for host sending over 1 query per second
  – SVM (Support Vector Machine): generates boundary between normal region and abnormal region based on the training data
Accuracy of classification

• Evaluate the accuracy with 10 cross-fold validation
  – Separate training data into 10 groups.
  – Classify host in a group by using training data of the rest of nine groups and compare the classification results and manual label.
• 10% improvement can be achievement by using hierarchical aggregate entropy

<table>
<thead>
<tr>
<th>Entropy</th>
<th>Mis-classification ratio (FP+NP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchical Aggregate Entropy</td>
<td>8.7%</td>
</tr>
<tr>
<td>FQDN Entropy</td>
<td>18.9%</td>
</tr>
<tr>
<td>SLD Entropy</td>
<td>23.3%</td>
</tr>
<tr>
<td>TLD Entropy</td>
<td>19.8%</td>
</tr>
</tbody>
</table>
Reverse queries (IP addr -> FQDN) have common SLD and TLD -> entropy is zero.
- 1.0.168.192.in-addr.arpa.

Apply our hierarchical aggregate entropy to IP address part
- 1.0.168.192.in-addr.arpa.

Confirm concentration of entropy of normal clients to a domain
Effect of DNS prefetch

- Extract Firefox users, and compare their entropies before and after Firefox implements DNS prefetch.
- After the implementation, ratio of Firefox users among heavy users increases, and that of normal heavy users increases as well.
- Filtering queries from heavy users may impede Internet access of normal users.
Conclusion

• Propose the use of hierarchical aggregate entropies to classify DNS heavy clients
• Can capture spatial dispersion of queries among domain name spaces
• Entropies from normal clients concentrated in a specific region
• Experimental results show that the proposed method achieve 10 % improvement in classification accuracy