Outline

● Motivation
  - original work
  - algorithm
  - DNS specifics

● Method description
  - original work

● Experiments
  - set-up
  - results

● Conclusion
Motivation

- Most of the internet communication starts with a DNS query.

- There is a possibility to track communication at a certain level of DNS hierarchy.

- There is a possibility to track communication at a certain level of DNS hierarchy.

- Will the tool be able to detect something at a ccTLD?

- We want a tool that is able to:
  - detect low volume anomalies
  - scan high volume traffic
  - detect suspicious behaviour
  - works in real-time = low computation cost
  - does not need any initial knowledge about the analysed traffic
  - e.g. for intrusion detection, botnet discovery

- Does not need any initial knowledge about the analysed traffic
Blindly analyses large-scale packet trace databases.

• Promises a very low computation cost.
• Detection method is sensitive to statistical characteristics.
• Able to detect short-lived anomalies as well as longer ones.

G. Dewaele, K. Fukuda, P. Borgnat, P. Abry, K. Cho

Extracting Hidden Anomalies using Sketched and Non Gaussian Multiresolution Statistical Detection Procedures by G. Dewaele,

Original Work
Method description

The algorithm analyses the traffic using a sliding time-window.

The analysis iterates over the following steps:

1. Random projection - sketches
2. Data aggregation - sketches
3. Gamma distribution estimation
4. Reference values computation
5. Distance from reference evaluation
6. Sketch combination and anomaly identification

The algorithm analyses the traffic using a sliding time-window within which the analysis is performed.
Random projections

- A fixed size time-window of captured traffic is split into sketches using a hash function.
- Selected packet attribute (policy) serves as hash key.
- Hash table size is fixed.
The sketches are aggregated jointly over a collection of aggregation levels to form a series of packet counts which arrived during an aggregation period. Aggregation levels transform the time-scale granularity. Data from the aggregated time series are modelled using Gamma distribution. Shape ($\alpha$) and scale ($\beta$) Gamma distribution parameters are computed for each aggregation level. Aggregation levels transform the time-scale granularity.

- The sketches are aggregated jointly over a collection of parameters.
Reference values, identification of anomalous sketches

- For each aggregation level across all sketches, standard sample mean and variance of the computed Gamma parameters are computed.
- For each sketch, the average Mahalanobis distance to the 'centre of gravity' is computed. Sketches with their average distance exceeding a given threshold are marked as anomalous.
- For each sketch, the average Mahalanobis distance to the 'centre of gravity' is computed.
Anomaly identification

- All packet attributes (hash keys) contained in an anomalous sketch are considered suspicious.
- Using a different hash function provides a different mapping into sketches resulting in various anomalous sketches.
- A list of attributes corresponding to detected anomalies is obtained by combining the results for several hash functions and computing the intersection of anomalous sketches.

Anomaly Identification
The method was designed to analyse the whole TCP/IP traffic.

- Works with TCP/IP connection identifiers (src/dst port/address).
- Help finding suspicious traffic from legitimate sources.
- First domain name of the query is extracted and used as hash key.

We extended it to meet DNS traffic specifics.

- Query name policy
  - First domain name of the query is extracted and used as hash key.

- IP address policy
  - Works with TCP/IP connection identifiers (src/dst port/address).
  - Help finding suspicious traffic from legitimate sources.

Policies:
- Supports IPv4 and IPv6.
- Based on original paper, uses the TCP/IP connection identifiers.
- Helps finding suspicious traffic sources.
- Helps finding suspicious traffic sources.

Modification for DNS

The method was designed to analyse the whole TCP/IP traffic.
The tool

- Standalone application is freely available at
  git://git.nic.cz/dns-anomaly/

**Command line parameters:**

- Count of aggregation levels
- Window size + detection interval
- Aggregation steps are power of 2 in seconds (i.e. 1, 2, 4, 8, ...)
- Analyse shape, scale or both
- Detection threshold
- Policy
- Hash function count
- Sketch count (hash table size)
Experiments

Tested on DITL 2011 data collected in April 2011 on .cz authoritative DNS servers.

Using these settings the analyser is able to process 10 minutes of traffic (126MB) in 1.8 second on an E5400@2.7G.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>distance threshold</td>
<td>0.8</td>
</tr>
<tr>
<td>aggregation levels</td>
<td>8</td>
</tr>
<tr>
<td>hash table size</td>
<td>32</td>
</tr>
<tr>
<td>hash function count</td>
<td>25</td>
</tr>
<tr>
<td>time-window size</td>
<td>10 minutes</td>
</tr>
<tr>
<td>detection interval</td>
<td>10 minutes</td>
</tr>
</tbody>
</table>

Experiments
Results

Types of traffic labelled as anomalies:

- Traffic from legitimate sources (exhibiting specific patterns)
  - e.g. large recursive resolvers, web crawlers

- Domain enumeration
  - With the knowledge of the content (little or no NXDOMAIN replies)
    - e.g. alteration for given words – e.g. bank or various trademarks
    - Blind or dictionary based (gTLD domain, prefix and postfix)

- Suspicious
  - Traffic generated by broken resolvers or testing scripts.
    - e.g. bursts of queries for the same name from single host
    - Repeated queries due to short TTL
      - e.g. bursts of queries for the same name from single host

- Traffic from legitimate sources (exhibiting specific patterns)
  - Traffic of traffic labelled as anomalies:
Originates at webhosting/ISP. The pattern is very.

Web crawler farm

Possibly web crawlers. They generate lots of references whenever they encounter sites with many references.

Recursive resolver

Many references.
When analysing the DNS queries a pattern emerged – prefixes and postfixes variation using well-known trademarks.

The source must have a very good knowledge about the content of the domain. Very few NXDOMAIN replies are generated.

**Known domain enumeration**

**Blind domain enumeration**
Hundreds of queries for a single record are generated in less than two seconds. The pattern is visible throughout the entire tested period - always as characteristic spikes.

Multiple hosts are generating bursts of queries for the same record. Multiple hosts evenly distributed around the world are querying the same MX record.

Possible spam attack

Broken resolver

Other suspicious
The tool is able to pinpoint low- and high-volume anomalies.

Two policies implemented with different effect:
- IP policy serves best for domain enumeration detection.
- Query name policy divulges domain-related events.
- e.g. presence of short TTL domains (fast flux)

The classification of the anomalies is currently left to be done manually.

Future work: automate this process.
A system under development

- Network interface
- Archived traffic
- Statistical anomaly classifier
- Anomaly detector
- Loadable module layer
- Python script execution
- DNS traffic archiver
- Exporting raw DNS packets
- TCP connection reconstruction
- De-fragmentation
- Network interface

A system under development
DNS anomaly classifier

- Used to classify output from the DNS anomaly detector.

- Random forest classifier is being used because:
  - Highly accurate classifier
  - Efficient run on large data sets
  - Gives an estimate of what variables are important in the classification
  - Soft decisions

- Classifiers can be saved for future use

- Classifiers are sensitive to the source of classified data. Different sources need to have separate classifiers.
Input variables – statistical data

- 62 variables serve for classification of anomalous data
- Relative and absolute measures regarding the volume of query types, return codes, TTL
- Penetration of various selected identifiers
- BGP prefixes, ASN, IP addresses, country of origin, query names
- Also takes into account query time, total traffic volume, server response time
Classifier performance and accuracy

- The training set contains approximately 1 million classified samples (six days of anomalous traffic).
- Training a classifier containing 200 trees each containing 15 nodes lasts about 2 hours.
- Classifying such a large data set using the trained forest lasts about 10 minutes.
- Classifying out-of-bag data yields 80% accuracy.
- The accuracy was determined by comparing classifier results with hand classified data.

The training set contains approximately 1 million classified samples (six days of anomalous traffic).
Work in progress

- Statistical collector module – replacement for DSC?
- Performance improvements in the random forest classifier
- Communication protocol
- Change collector settings, module reloading, attaching/detaching network devices
- Communication consumption – increasing accuracy, increasing speed, reducing memory consumption


Questions?

Thank you for your attention.

The End