Data-driven models of reputation in cyber-security

Pierre Lison
Norwegian Computing Center (NR)

Second Workshop on Text Analytics for Cybersecurity and Online Safety (TA-COS)

11th International Conference on Language Resources and Evaluation (LREC 2018)

12/05/2018
Introduction

- Blacklists and whitelists (= reputation lists) often employed to filter network traffic
- Manually curated by security experts
Introduction

► Shortcomings of blacklists and whitelists:
  ▪ Slow reaction time
  ▪ Maintenance is difficult and time-consuming
  ▪ Limited coverage
  ▪ Static (can be circumvented through techniques such as domain flux and fast flux networks)
Introduction

Can we use **machine learning** to automatically predict the reputation of end-point hosts?

1. Predictions in real-time, without human intervention
2. Less vulnerable to human errors and omissions
3. Full coverage of end-point hosts
Can we use **machine learning** to automatically predict the reputation of end-point hosts?

- Detecting domain names generated by malware with RNNs
- Predicting the reputation of domains and IP addresses from passive DNS data
Part 1: Detecting domain names generated by malware
Domain-generating malware

- Most malware must connect compromised machines with a command and control (C2) server for their operations.

**Attacker**

**Cyber-attack**
(through e.g. spear phishing)

**C2 server**
5.35.225.127
Domain-generating malware

- Most malware must connect compromised machines with a command and control (C2) server for their operations.

Static domains or IP addresses can be used…
… but are easy to block (with e.g. blacklists).
Domain-generating malware

- With domain-generation algorithms (DGA), compromised machines will attempt to connect to a large number of pseudo-random domain names...

- The attacker can then simply register a few of these artificial domains to establish a rendez-vous point.

Register toyvsgu.com As 5.35.225.127
Domain-generating algorithms (DGAs)

- DGAs increasingly popular as command-and-control (C2) rendez-vous mechanism in botnets
  - First observed in the Kraken botnet (2008)
- DGAs generate a large number of seemingly random domain names based on a *shared secret* (seed)
- Highly *asymmetric* situation:
  - Malicious actors only need to register a single domain to establish a C2 communication channel
  - But security professionals must control the full range of potential domains to contain the threat (*counter-measures*: pre-registering, blacklists, or sinkholes)
Taxonomy of DGAS

► **Time dependence:**
  - Are the seeds fixed or are they only valid for a specific period (by including a time source in their calculation?)

► **Determinism:**
  - Are the seeds computed through a deterministic procedure, or do they include unpredictable factors (weather forecasts, stock markets prices, etc.)

► **Generation scheme:**
  - How are the domains generated from the seeds? Popular techniques include alphanumeric combinations, hash-based techniques, wordlists and permutations.
Detection of DGAs

- **Recurrent neural network** trained on a large dataset of benign & malicious domains
  - Ability to learn complex sequential patterns

- Purely data-driven – easy to apply and update

![Diagram of a recurrent neural network](image)
Architecture

Recurrent layer builds up a representation of the character sequence as a dense vector.

First layer encode each character as a "one-hot" vector.

Domain name is fed to the neural network character by character.

Final vector is used to predict whether the domain is DGA.

Recurrent layer (LSTM or GRU)

One-hot layer

Input layer (characters)

Output (probability of being generated by malware)

domain.com

t oyvs g u . c om

Domain name is fed to the neural network character by character.
Extensions

- Embeddings
- Bidirectionality
- Hidden layer
- Multi-task learning

Recurrent layer (LSTM or GRU)
Learned embeddings
Input layer (characters)

Output (probability of being generated by malware)

```
t o y v s g u . c o m
```
Extensions

- Embeddings
- **Bidirectionality**
- Hidden layer
- Multi-task learning

![Diagram of a neural network with layers labeled as Recurrent layer (right-to-left), Recurrent layer (left-to-right), One-hot layer, Input layer (characters), and Output (probability of being generated by malware).]
Extensions

- Embeddings
- Bidirectionality
- Hidden layer
- Multi-task learning

![Diagram of a neural network model with layers: Input layer (characters), One-hot layer, Recurrent layer (LSTM or GRU), Dense layer (linear combination + non-linear activation), and Output (probability of being generated by malware).]
Extensions

- Embeddings
- Bidirectionality
- Hidden layer
- Multi-task learning

Recurrent layer (LSTM or GRU)

One-hot layer

Input layer (characters)

... t o y v s g u . c o m ...
Data

► The parameters of the neural model must be estimated from training data

► **Negative examples** (benign domains):
  ▪ Snapshots from the Alexa top 1 million domains
  ▪ Total: over 4 million domains

► **Positive examples** (malware DGAs)
  ▪ DGA lists from the DGArchive (63 types of malware)
  ▪ Feeds from Bambenek Consulting
  ▪ Domain generators for 11 DGAs
  ▪ Total: 2.9 million domains
# Data

<table>
<thead>
<tr>
<th>Malware</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>bamital</td>
<td>40 240</td>
</tr>
<tr>
<td>banjori</td>
<td>89 984</td>
</tr>
<tr>
<td>bedep</td>
<td>15 176</td>
</tr>
<tr>
<td>beebone</td>
<td>420</td>
</tr>
<tr>
<td>blackhole</td>
<td>732</td>
</tr>
<tr>
<td>bobax</td>
<td>19 288</td>
</tr>
<tr>
<td>conficker</td>
<td>400 000</td>
</tr>
<tr>
<td>corebot</td>
<td>50 240</td>
</tr>
<tr>
<td>cryptolocker</td>
<td>55 984</td>
</tr>
<tr>
<td>cryptowall</td>
<td>94</td>
</tr>
<tr>
<td>dircrypt</td>
<td>11 110</td>
</tr>
<tr>
<td>dnschanger</td>
<td>40 000</td>
</tr>
<tr>
<td>downloader</td>
<td>60</td>
</tr>
<tr>
<td>dyre</td>
<td>47 998</td>
</tr>
<tr>
<td>ekforward</td>
<td>1 460</td>
</tr>
<tr>
<td>emotet</td>
<td>40 576</td>
</tr>
<tr>
<td>feodo</td>
<td>192</td>
</tr>
<tr>
<td>fobber</td>
<td>2 600</td>
</tr>
<tr>
<td>gameover</td>
<td>80 000</td>
</tr>
<tr>
<td>gameover_p2p</td>
<td>41 000</td>
</tr>
<tr>
<td>gozi</td>
<td>105 631</td>
</tr>
<tr>
<td>hesperbot</td>
<td>370</td>
</tr>
<tr>
<td>locky</td>
<td>179 204</td>
</tr>
<tr>
<td>madmax</td>
<td>192</td>
</tr>
<tr>
<td>matsnu</td>
<td>12 714</td>
</tr>
<tr>
<td>modpack</td>
<td>52</td>
</tr>
<tr>
<td>murofet</td>
<td>53 260</td>
</tr>
<tr>
<td>murofet(_w)</td>
<td>40 000</td>
</tr>
<tr>
<td>necur</td>
<td>40 000</td>
</tr>
<tr>
<td>necurs</td>
<td>36 864</td>
</tr>
<tr>
<td>nymaim</td>
<td>186 653</td>
</tr>
<tr>
<td>oderoor</td>
<td>3 833</td>
</tr>
<tr>
<td>pacrypt</td>
<td>35 616</td>
</tr>
<tr>
<td>proslife(_k)</td>
<td>75 270</td>
</tr>
<tr>
<td>pushdo</td>
<td>176 770</td>
</tr>
<tr>
<td>pushdotid</td>
<td>6 000</td>
</tr>
<tr>
<td>pykspa</td>
<td>424 215</td>
</tr>
<tr>
<td>pykspa(_2)</td>
<td>24 322</td>
</tr>
<tr>
<td>qadars</td>
<td>40 400</td>
</tr>
<tr>
<td>qakbot</td>
<td>90 000</td>
</tr>
<tr>
<td>ramdo</td>
<td>15 984</td>
</tr>
<tr>
<td>ramnit</td>
<td>90 000</td>
</tr>
<tr>
<td>ranbyu</td>
<td>40 000</td>
</tr>
<tr>
<td>ranbyus</td>
<td>12 720</td>
</tr>
<tr>
<td>rovnix</td>
<td>40 000</td>
</tr>
<tr>
<td>shifu</td>
<td>4 662</td>
</tr>
<tr>
<td>simda</td>
<td>38 421</td>
</tr>
<tr>
<td>sison</td>
<td>5 936</td>
</tr>
<tr>
<td>suppobox</td>
<td>41 014</td>
</tr>
<tr>
<td>sutra</td>
<td>9 882</td>
</tr>
<tr>
<td>symmi</td>
<td>40 064</td>
</tr>
<tr>
<td>szribi</td>
<td>16 007</td>
</tr>
<tr>
<td>tempedreve</td>
<td>453</td>
</tr>
<tr>
<td>tinba</td>
<td>80 000</td>
</tr>
<tr>
<td>torpig</td>
<td>40 000</td>
</tr>
<tr>
<td>tsifiri</td>
<td>59</td>
</tr>
<tr>
<td>urlzone</td>
<td>34 536</td>
</tr>
<tr>
<td>vawtrak</td>
<td>1 050</td>
</tr>
<tr>
<td>virut</td>
<td>400 600</td>
</tr>
<tr>
<td>volatilecedar</td>
<td>1 494</td>
</tr>
<tr>
<td>xxhex</td>
<td>4400</td>
</tr>
</tbody>
</table>

**Total**: 2 925 168
Evaluation

- 10-fold cross validation on the full dataset
- **Baseline**: logistic regression on character bigrams
  - Toyvsgu.com → (to, oy, yv, vs, sg, gu, u., .c, co, om)
- Metrics: accuracy, precision, recall, F$_1$ score

\[
\text{precision} = \frac{\# \text{ correctly classified malware domains}}{\# \text{ domains classified as malware by model}}
\]

\[
\text{recall} = \frac{\# \text{ correctly classified malware domains}}{\# \text{ actual known malware domains}}
\]

\[
\text{F}_1 \text{ score} = \frac{2p \times r}{p + r} \quad \text{(harmonic mean of the two)}
\]
Model selection

► The use of embeddings, bidirectional layers, and additional hidden layers did not improve the performance

► Multi-task learning (i.e. simultaneously learning to detect DGAs and to classify them) yielded the same results as networks optimised for these two tasks separately
  - The two tasks can use a shared latent representation

► The recurrent layer used GRU units with dimension=512

► Model trained on GPU with a batch size of 256, two passes and RMSProp as optimisation algorithm
## Results

### Detection

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$ score</th>
<th>ROC AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram</td>
<td>0.915</td>
<td>0.927</td>
<td>0.882</td>
<td>0.904</td>
<td>0.970</td>
</tr>
<tr>
<td>Neural model</td>
<td>0.973</td>
<td>0.972</td>
<td>0.970</td>
<td>0.971</td>
<td>0.996</td>
</tr>
</tbody>
</table>

### Classification

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$ score</th>
<th>Micro</th>
<th>Macro</th>
<th>Micro</th>
<th>Macro</th>
<th>Micro</th>
<th>Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram</td>
<td>0.800</td>
<td>0.787</td>
<td>0.564</td>
<td>0.800</td>
<td>0.513</td>
<td>0.787</td>
<td>0.522</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neural model</td>
<td>0.892</td>
<td>0.891</td>
<td>0.713</td>
<td>0.892</td>
<td>0.653</td>
<td>0.887</td>
<td>0.660</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Area Under the Curve (AUC) of the ROC curve (see next slide)

- **Micro**: weighted averages over all classes
- **Macro**: unweighted averages
ROC curve

The ROC curve compares different models: Neural GRU with $d=512$ (AUC=0.996), Neural GRU with $d=128$ (AUC=0.994), and Bigram (AUC=0.970).
Discussion

- Neural model is also able to detect dictionary-based DGAs such as suppobox (recall of 93%, compared to only 12% for baseline) when given enough training examples
- Some DGAs still remain difficult to detect, such as matsnu (not enough training data to learn underlying wordlists)
- See our paper for detailed results for each malware family

Part 1: Predicting the reputation of domains and IP addresses from passive DNS data
Passive DNS

- Can we automatically predict the reputation of domain names and IP addresses from DNS data?

- Passive DNS data is highly useful:
  - Inter-server DNS messages captured by sensors
  - Less privacy concerns (not tied to personal information)
Passive DNS

► Collaboration with Mnemonic AS, a Norwegian cybersecurity company [www.mnemonic.no]
► Dataset of 720 million aggregated DNS queries collected over a period of four years
► Each entry is defined by:
  ► A record type (A, CNAME, etc.) 84% A records,
  ► A query and its answer, 11% CNAME
  ► A Time-to-Live (TTL) value 4% AAAA
  ► A number of occurrences
  ► Timestamps for the first and last occurrence of the entire
We enriched the passive DNS data with:

- Reputation labels from existing blacklists and whitelists
- IP location (geoname identifiers) and ISP data

Labelled dataset of **720 million** records
Data

Labelled dataset of 720 million records (102 M records labelled as benign, 8.2 M records as malicious and 614 K records as sinkhole)

- The reputations are associated with a confidence level (from the reputation source and description)
- Employed to derive reputations for DNS records (edges)
Graph inference

- Local neighbourhood is important for the reputation
- Traversal of bipartite graph to extract the number of neighbours and their reputations
- Experiments with adapted versions of PageRank
Features

► Numerical features derived from the records:
  ▪ Lifespan, number of queries (for record, domain or IP), number of distinct countries or ISP, TTL values, etc.

► Categorical features:
  ▪ ISP, geolocation, top-level domain, etc.

► Ranking features from Alexa

► Features extracted from neighbouring records
  ▪ Number of records at distance 1 and of reputation X

► Sequence of characters from the domain
Neural model

Output layer:

Dense layer 2:

Dense layer 1:

Input layer:

Numerical features

Categorical features

Domain name

Probability of being generated by malware

3 2.6 0.3 78 0

RU ISP65 G411 ORG
d o l … r g

benign malicious sinkhole
In this setting, the neural net is first trained on the labelled dataset and applied to predict the reputation of unlabelled records, which are then used to get better estimates of the "neighbour" features. The model is then trained again on these new feature values.

<table>
<thead>
<tr>
<th>Model</th>
<th>Benign</th>
<th></th>
<th>Malicious</th>
<th></th>
<th>Sinkhole</th>
<th></th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F&lt;sub&gt;1&lt;/sub&gt;</td>
<td>P</td>
<td>R</td>
<td>F&lt;sub&gt;1&lt;/sub&gt;</td>
<td>P</td>
</tr>
<tr>
<td>nb domain queries &lt; 10</td>
<td>0.98</td>
<td>0.44</td>
<td>0.61</td>
<td>0.10</td>
<td>0.87</td>
<td>0.19</td>
<td>0.0</td>
</tr>
<tr>
<td>Logistic regression</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.60</td>
<td>0.65</td>
<td>0.62</td>
<td>0.51</td>
</tr>
<tr>
<td>Neural net (with 1 hidden layer)</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.99</td>
</tr>
<tr>
<td>Neural net (with 2 hidden layers)</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>0.92</td>
<td>0.95</td>
<td>0.93</td>
<td>0.98</td>
</tr>
</tbody>
</table>
ROC curve

Neural net

Logistic regression
With the best performing model, we achieve a recall of:

- 0.74 for a false positive rate of 1:100K
- 0.86 for 1:10K
- 0.92 for 1:1000
- 0.97 for 1:100
- 0.99 for 1:10.
Future work

► Exploiting **semi-structured information sources**
  ► Security reports, alerts on cyber-security websites, etc.
  ► Knowledge discovery, information extraction necessary

► **Benefit**: go beyond simple reputation labels and understand *why* a host should or should not be trusted

► **Challenges**:
  ► Lack of annotated text data for this domain
  ► Inconsistent naming conventions for cyber-threats
Conclusion

► Neural networks can be successfully used to predict the reputation of end-point hosts
  ▪ Detection of DGA from the domain names
  ▪ Detection of malicious records from passive DNS

► Can be integrated in software tools for cyber-threat intelligence

► Current work:
  ▪ Consolidate experimental results
  ▪ Integration of unstructured data sources (i.e. textual data)