twosix
TECHNOLOGIES

Campaign Detection

Automatic Cyber-attack campaign detection using network traffic data

Emily Gray emily.gray@twosixtech.com
Chae Clark chae.clark@twosixtech.com
Robert Gove robert.gove@twosixtech.com

Campaign Detection

Why do we care?

Cyber-attacks on the internet don’t occur in a vacuum. Events are related. Only seeing one (or a few) parts can lead to incomplete or incorrect assumptions. Providing analysts with an automated way to assess which occurrences are related to each other can help improve both speed and quality of cybersecurity work.

There are lots of things on the internet, and trying to detect campaigns by hand is infeasible. Automation of campaign detection is a possible solution.

Presentation summary

What

Detect campaigns by determining whether pairs of log lines are from the same attack

Aggregate over those pairs to discover the whole campaign

How

Use generated data to determine expectations

Use project-specific data to test algorithm viability

Results

Generated data: can achieve F1 scores of 0.75-0.95, depending on data used

With project data, can determine cutoff score to identify incident reports belonging to same attack

Background work

Challenges to think about

A Naive Approach

Measuring distance between sets (modified Jaccard distance)

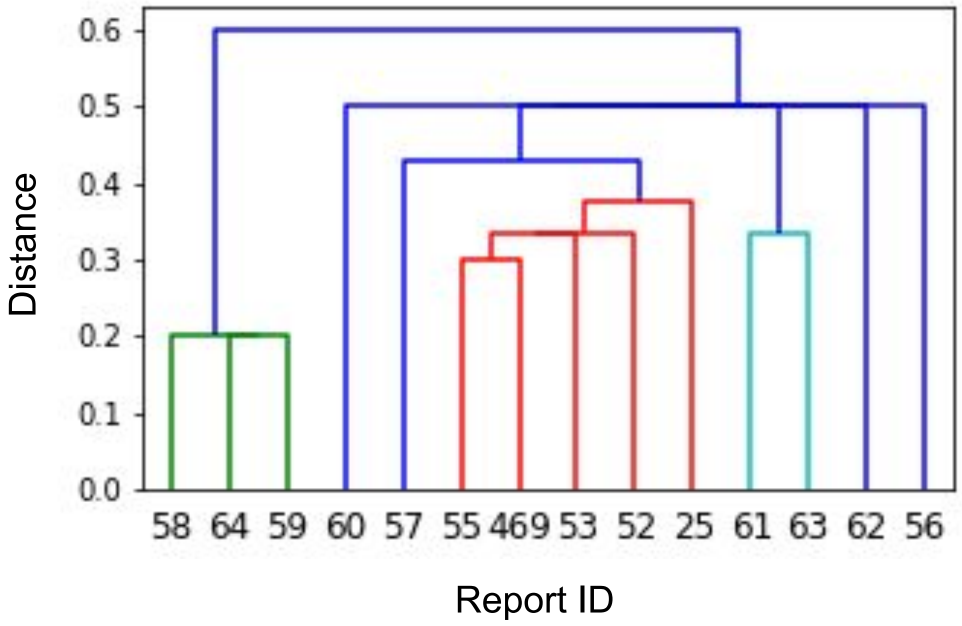
A: set of entities from one attack

B: set of entities from another attack

Measure overlap: $1 - \frac{|A \cap B|}{\min(|A|,|B|)}$

Cutoff of 0.5-0.6 could indicate same campaign

Problem: not enough data to prove or test rigorously



Challenges

Usual challenges for machine learning with cyber data apply:

1. Variability of data, format and content
2. Limited labeled data
3. Volume of data

Algorithm design

Algorithm Design

Design is relatively simple

- Divide data into manageable log lines
- Pair log lines randomly
- Label pairs as from ‘same’ or ‘different’ attacks
- String vectorization and AutoKeras’ Automodel

Addresses problem #1 (variability of data)

String vectorization allows for multiple data formats to be used simultaneously

Testing Strategy

Step 1: will this work?

- Start with generated data
- Can make data with whatever labels and attacks are desired

Addresses problem #2 (lack of labeled data)

Generated data allows creation of exactly the format and type desired

Step 2: project specific data

- Run as second pass after threat detectors identify traffic of interest
- However, the more limited data makes testing more difficult

Addresses problem #3 (data volume), but reintroduces #2

Step 1: Generated data

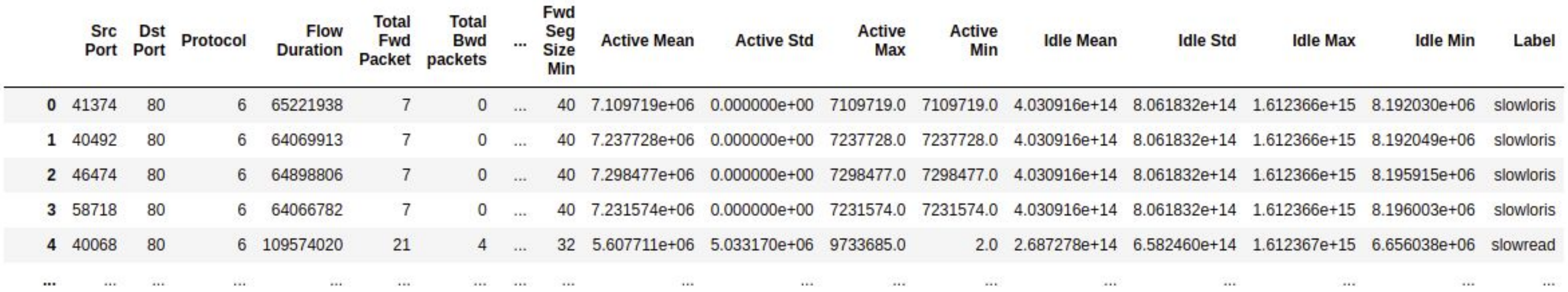
Controlling training data to assess viability

Generating data

Use MagicWand[1] - open source tool that generates high quality reproducible DDoS data from a variety of attacks

Generate data from each of 7 available attacks

Remove features that are indicative of generation differences (IP addresses, timestamps)



	Src Port	Dst Port	Protocol	Flow Duration	Total Fwd Packet	Total Bwd packets	...	Fwd Seg Size Min	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	41374	80	6	65221938	7	0	...	40	7.109719e+06	0.000000e+00	7109719.0	7109719.0	4.030916e+14	8.061832e+14	1.612366e+15	8.192030e+06	slowloris
1	40492	80	6	64069913	7	0	...	40	7.237728e+06	0.000000e+00	7237728.0	7237728.0	4.030916e+14	8.061832e+14	1.612366e+15	8.192049e+06	slowloris
2	46474	80	6	64898806	7	0	...	40	7.298477e+06	0.000000e+00	7298477.0	7298477.0	4.030916e+14	8.061832e+14	1.612366e+15	8.195915e+06	slowloris
3	58718	80	6	64066782	7	0	...	40	7.231574e+06	0.000000e+00	7231574.0	7231574.0	4.030916e+14	8.061832e+14	1.612366e+15	8.196003e+06	slowloris
4	40068	80	6	109574020	21	4	...	32	5.607711e+06	5.033170e+06	9733685.0	2.0	2.687278e+14	6.582460e+14	1.612367e+15	6.656038e+06	slowread
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

[1] https://joss.theoj.org/papers/10.21105/joss.03032

Using the data

Randomly pair logs

- Some pruning required to prevent more prolific attacks from dominating

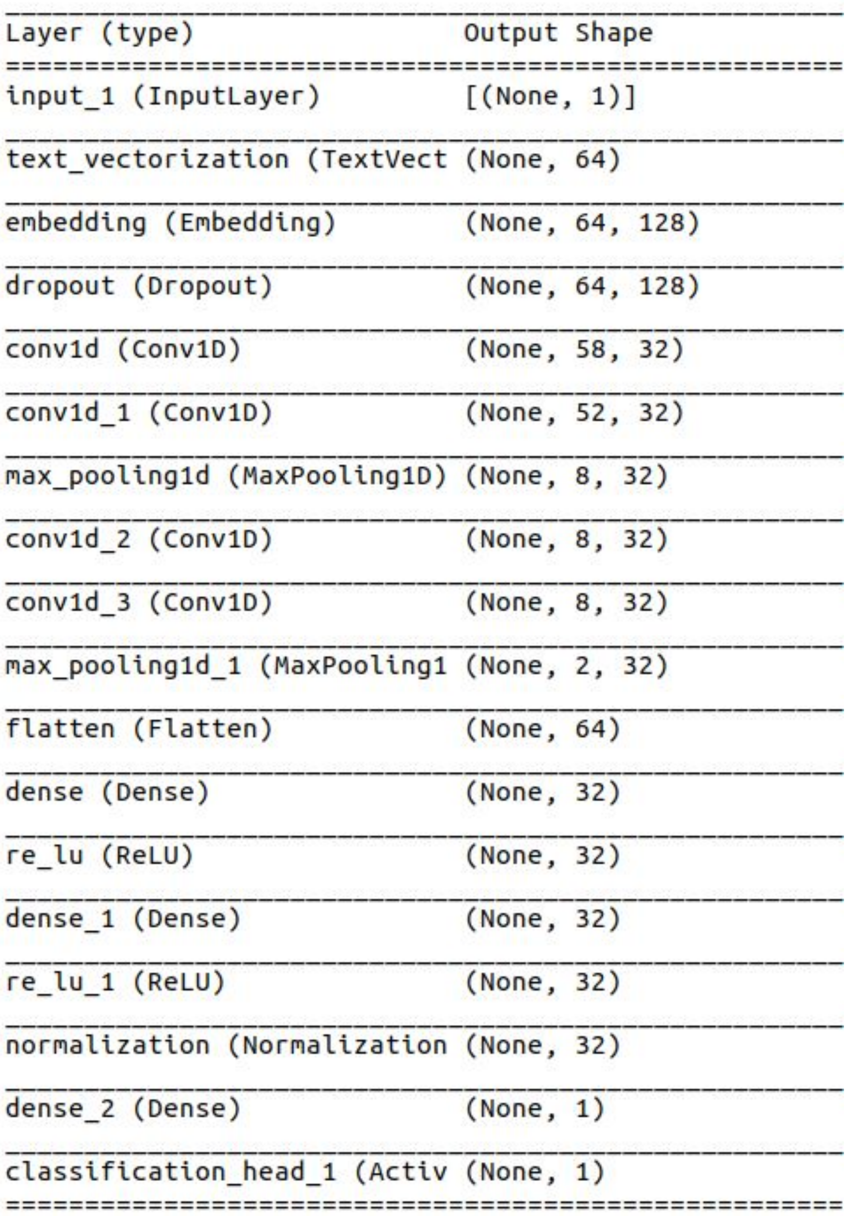
Label pairs as ‘same’ or ‘different’

- Not an attack classification problem
- Regardless of which attack, only concerned with equivalence of pairs

Train model using AutoKeras to predict whether pairs are from same or different attack

Test on holdout dataset

The Model



```
_____________________________________________________
Layer (type)                 Output Shape
=====================================================
input_1 (InputLayer)         [(None, 1)]
_____________________________________________________
text_vectorization (TextVect (None, 64)
_____________________________________________________
embedding (Embedding)        (None, 64, 128)
_____________________________________________________
dropout (Dropout)            (None, 64, 128)
_____________________________________________________
conv1d (Conv1D)              (None, 58, 32)
_____________________________________________________
conv1d_1 (Conv1D)            (None, 52, 32)
_____________________________________________________
max_pooling1d (MaxPooling1D) (None, 8, 32)
_____________________________________________________
conv1d_2 (Conv1D)            (None, 8, 32)
_____________________________________________________
conv1d_3 (Conv1D)            (None, 8, 32)
_____________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 2, 32)
_____________________________________________________
flatten (Flatten)            (None, 64)
_____________________________________________________
dense (Dense)                (None, 32)
_____________________________________________________
re_lu (ReLU)                 (None, 32)
_____________________________________________________
dense_1 (Dense)              (None, 32)
_____________________________________________________
re_lu_1 (ReLU)               (None, 32)
_____________________________________________________
normalization (Normalization (None, 32)
_____________________________________________________
dense_2 (Dense)              (None, 1)
_____________________________________________________
classification_head_1 (Activ (None, 1)
=====================================================
```

Results

With appropriate training parameters, results are promising

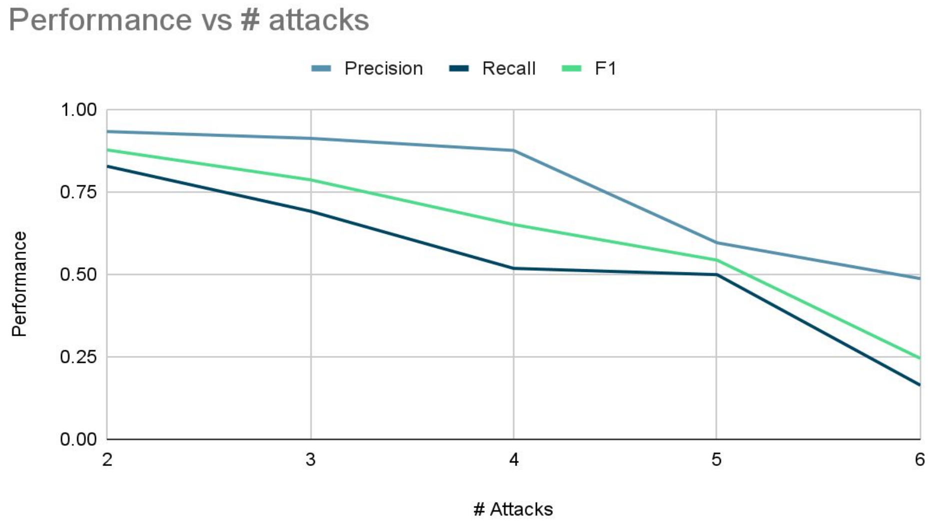
Training requirements vary based on data

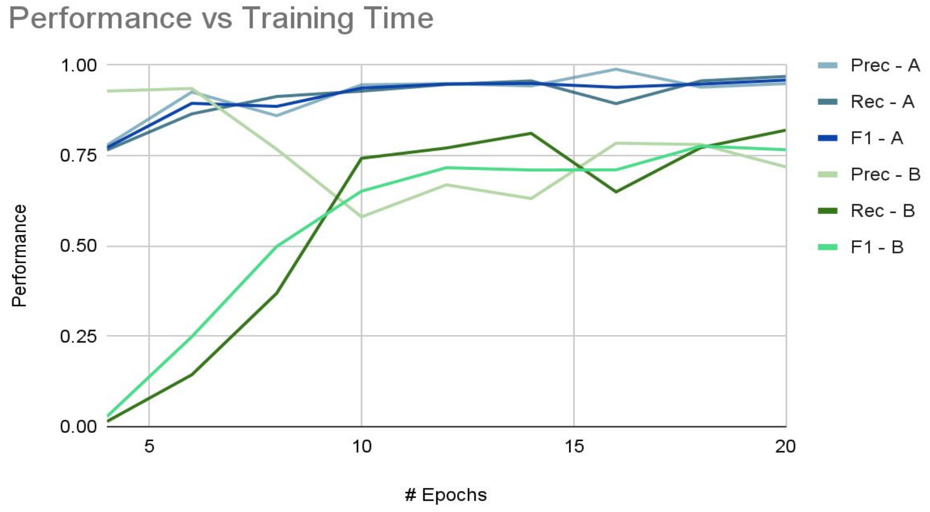
Need significantly longer time to distinguish between more types of attacks, more similar attacks (unsurprising)

Fig 1: Train for 2 trails, 8 epochs, vary number of attacks included in training/test set

Fig 2: Vary number of epochs.

- group A: 2 attack types
- group B: 4 attack types





Caveat

This is not a first pass algorithm

Computationally impossible to pair and test all internet traffic

Birthday problem: testing all pairs is way more comparisons than you think

Feasible as a second pass

Project data is already filtered through threat detection analytics

Pairing and testing the more limited data set is possible (100-200 incident reports per day)

Step 2: Project-specific data

Applying previous lessons to intended data

Project-specific incident reports

Automatically generated by project performers' threat detectors, but not yet checked by human analysts

Data presented in STIX format - standard format for describing cyber traffic

- IP addresses
- TTPs
- Timestamps
- Relationships between users

Not full-on log lines; just data intended for presentation to analyst

Less rich/dense than ZEEK logs

Training the Algorithm

Select incident reports generated from red-team event, and comparable non-red-team reports

Parse STIX into log lines (or log line equivalents)

Randomly pair up log lines, label appropriately (holding back test set)

Run through string vectorization and AutoModel

Resulting model has same layers as generated data model

Running the Algorithm

Pick two incident reports

Parse STIX

Randomly pair up log lines

Run algorithm on each pair, get scores (probability pair is from same attack)

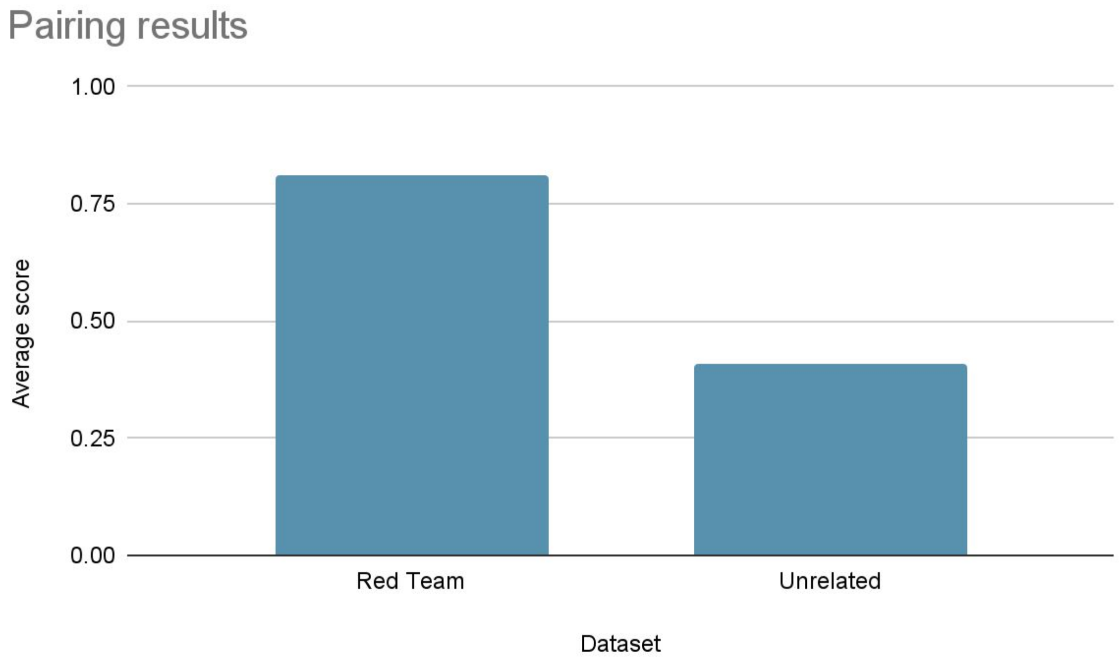
Aggregate (average) scores to determine whether incident reports are related

Results

Results are promising!

Average score for test incident reports: 0.81

Average score for unrelated incident reports: 0.41



What’s next?

Further work

Future Work

Expecting another round of project-specific data soon

- Will provide another campaign, expanding training data
- New data may cause different parameters to perform better
- May also reveal different data parsing results in better performance

Expanding to threat detectors/incident reports from other sources

- Will require complete retraining, with inclusion of other source formats
- Allow integration between multiple analyst tools

Conclusion

Campaign detection, via pairing log-line equivalents, is not viable as a first pass strategy on all internet traffic. However, as a second pass on data already identified by threat detectors, it shows promise at distinguishing whether two incident reports belong to the same cyber event.

Contact us

- **Emily Gray**
  emily.gray@twosixtech.com
- **Chae Clark**
  chae.clark@twosixtech.com
- **Robert Gove**
  robert.gove@twosixtech.com

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.
Distribution Statement "A" (Approved for Public Release, Distribution Unlimited)