Measuring similarity between cyber security incident reports

Zach Kurtz & Sam Perl

June 12, 2017

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Measuring similarity between cyber security incident reports

This material is based upon work funded and supported by the Department of Homeland Security and Intelligence Advanced Research Projects Activity (IARPA) under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

DM17-0154

Measuring similarity between cyber security incident reports

Why measure similarity between reports?

Basic similarities

Clustering reports

Evaluating the fortune tellers

Soft Jaccard similarity

Measuring similarity between cyber security incident reports

Why measure similarity between reports?

Why measure similarity between reports?

Support active investigations

- Understand a nonstandard cyber attack.
- *Identifying records of similar attacks lets you build on previous work.*

Identify campaigns or broader patterns

- Cluster events to see the bigger picture.
- Evaluate the meaning of existing taxonomies.
- *Most advanced clustering algorithms rely on some form of similarity.*

Evaluate cyber warning systems

- The IARPA CAUSE program develops early warning systems.
- *Usefulness of a warning depends on similarity against real events.*

Measuring similarity between cyber security incident reports

Basic Similarities

Basic Similarities – Features important for similarity

Two incidents are typically more "similar" if

- they happened close together in time
- they are of a similar "event type" in some cyber-incident taxonomy (watering hole, DDOS, phishing ...)
- they triggered similar alerts
- they targeted similar victims or vulnerabilities
- they contain similar indicators of compromise (IOCs)
- ... anything else you might have data on

A weighted comprehensive similarity function:

$$sim(e_1, e_2) = w_1 sim_{time}(e_1, e_2) + w_2 sim_{type}(e_1, e_2) + w_3 sim_{IOC}(e_1, e_2) + \ldots$$

Choose your weights & choose your similarities!

Basic Similarities – Compare two times or numbers

Suppose x_1, x_2 are two measurements. $sim(x_1, x_2)$ ranges from 0 to 1.

Simplest similarity:

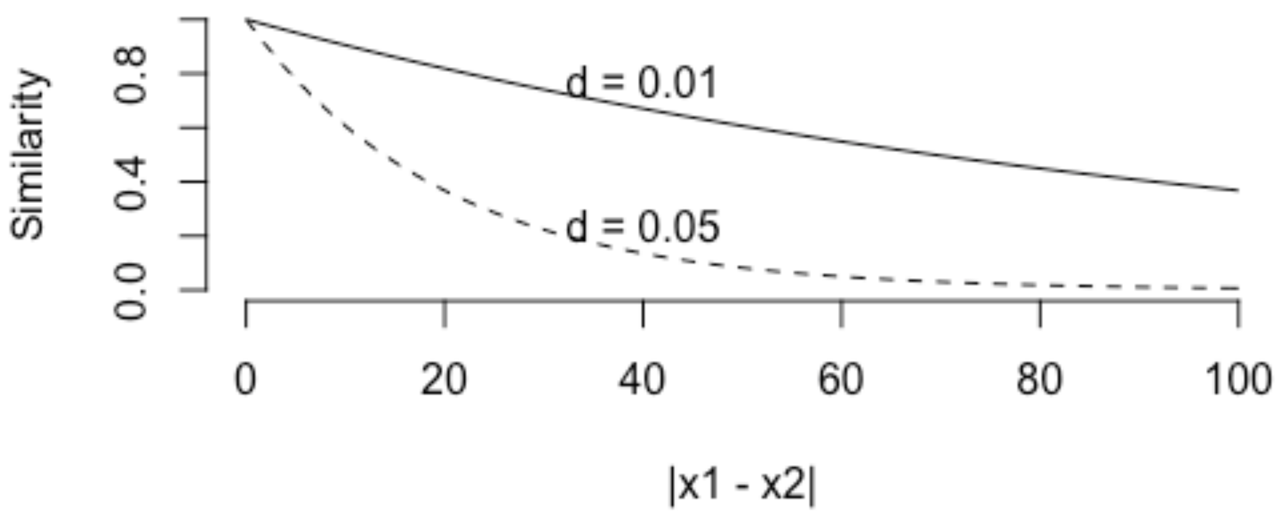
$$sim(x_1, x_2) = 1 \text{ if } x_1 = x_2 \text{ and } 0 \text{ otherwise}$$

Boxcar similarity (with “diameter” $d > 0$):

$$sim(x_1, x_2) = 1 \text{ if } |x_1 - x_2| < d \text{ and } 0 \text{ otherwise}$$

Exponential decay similarity:

$$sim(x_1, x_2) = e^{-d|x_1 - x_2|}$$

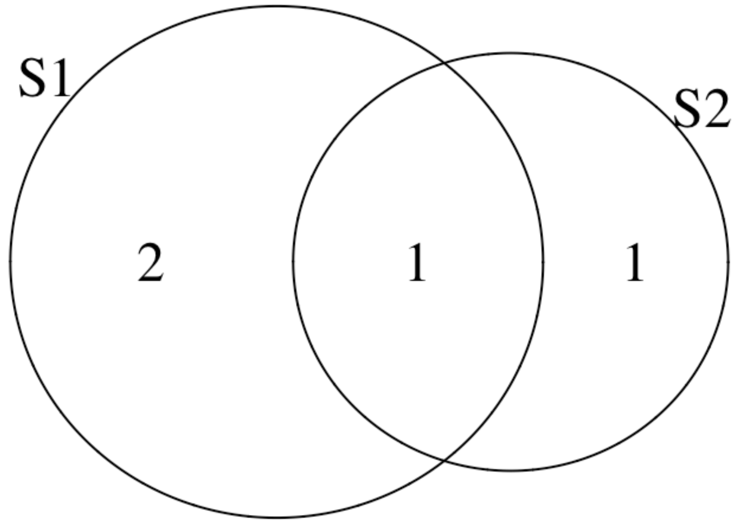


Basic Similarities - Similarity in IOCs

Suppose two reports contain some IOCs:

$$S_1 = \{ip_1, ip_2, phish.sender_1\}$$

$$S_2 = \{ip_2, url_1\}$$



How “similar” are the two sets?

Union and intersection notation:

$$\text{Union} = S_1 \cup S_2 = \{ip_1, ip_2, phish.sender_1, url_1\}$$

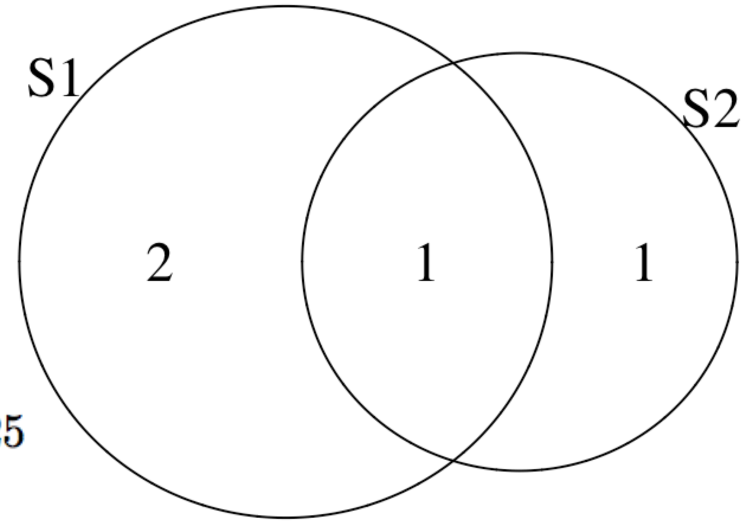
$$\text{Intersection} = S_1 \cap S_2 = \{ip_2\}$$

Basic Similarities - Similarity in IOCs

Suppose two reports contain some IOCs:

$$S_1 = \{ip_1, ip_2, phish.sender_1\}$$

$$S_2 = \{ip_2, url_1\}$$



Jaccard similarity:

$$Jaccard(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = \frac{1}{2+1+1} = 0.25$$

What if some types of indicators are more meaningful than others? Break out the various types: $S = S(ip) \cup S(url) \cup S(phish.sender) \cup ...$

Choose some weights:

$$1 = w_{ip} + w_{url} + w_{phish.sender} + ...$$

Make a weighted average:

$$sim(S_1, S_2) = w_{ip} sim(S_1(ip), S_2(ip)) + w_{url} sim(S_1(url), S_2(url)) + ...$$

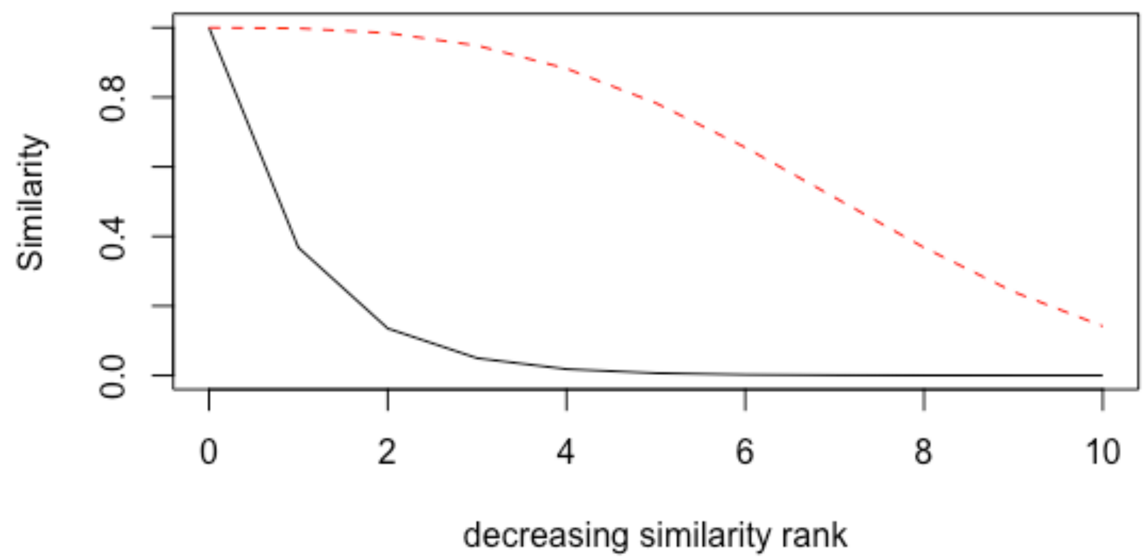
Basic Similarities – Finding the most similar incidents

Suppose “target record” is an incident record of interest in some database.

To find the other records most similar to R: sort records by descending similarity.

ID	Similarity with target
target record	1
id1	0.7
id2	0.4
…	…
idN	0.0001

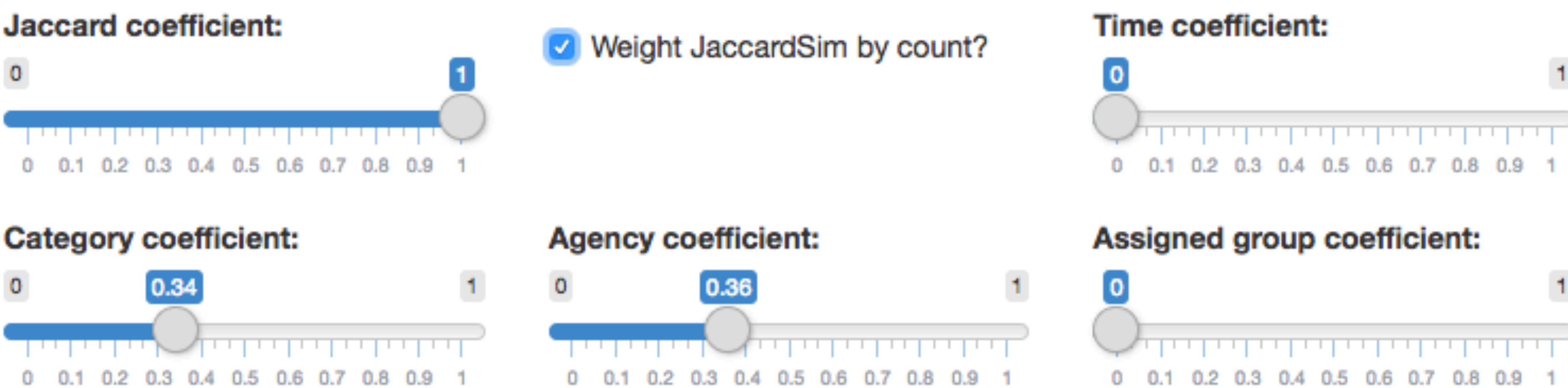
The "rate of decay" of similarity for the most-similar reports can look very different depending on the similarity and the data:



Basic Similarities – Importance of weights

Screenshot from a R-Shiny tool developed for browsing US-CERT data:

The global similarity is a linear combination of the similarities computed for each dimension, where the combination coefficients are controlled by the sliders.



Show 10 entries

Incident	Reported	Agency	Category	Assigned	JaccardSim	Obs Count
id1 = ref. id	time1	ag1	DDOS	group1	1	na
id2	time1	ag2	Phishing	group2	0.6	na
…	…	…	…	…	…	…

Measuring similarity between cyber security incident reports

Clustering reports

Clustering Reports – Role of similarity

What I'm not going to do ...

The simplest "out of the box" clustering algorithms (like k-means) rely on a numeric feature matrix:

ID	f1	…	fM
1	f11	…	f1M
2	f21	…	f2M
…	…	…	…
N	fN1	…	fNM

k-means relies on the NxN distance matrix, typically the Euclidian distance:

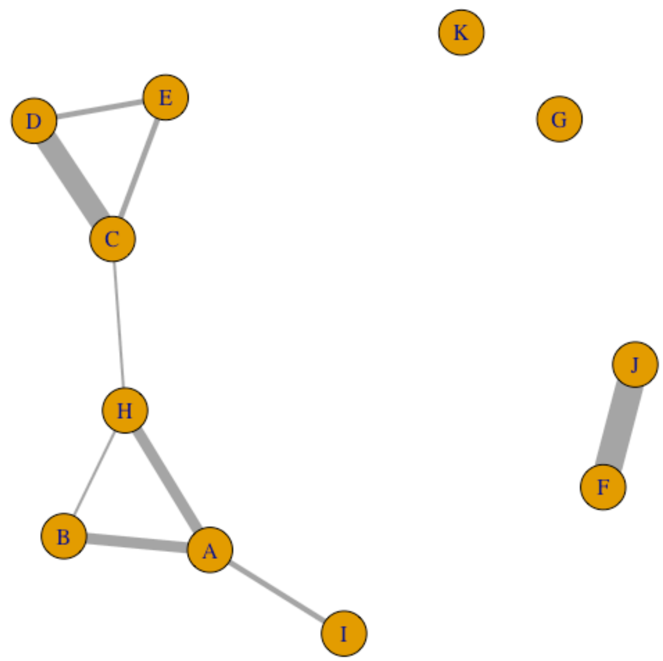
$$d(i,j) = \sqrt{(f_{i1} - f_{j1})^2 + ... + (f_{iM} - f_{jM})^2}$$

Problem: cyber incident data is extremely high-dimensional, and not "naturally" numeric

Clustering Reports – Role of similarity

Now similarities instead of distances

Think of similarities as “friendships” in a social network graph, representing a set of similarities.



Similarities:

$$sim(K, \cdot) \approx 0$$

$$sim(F, J) \approx 1$$

Sparsity: 9 similarities represent the whole network, much less than $O(n^2)$

Cluster sizes depend on a similarity threshold:

Clear-cut: {F,J}

Ambiguous: {C,D} or {C,D,E} or {A,B,C,D,E,H,I}?

Used igraph -- graph-based clustering (or community detection):

- guiding principle = modularity, or high within-cluster connectedness and low between-cluster connectedness
- many algorithms; trade-offs between speed and accuracy

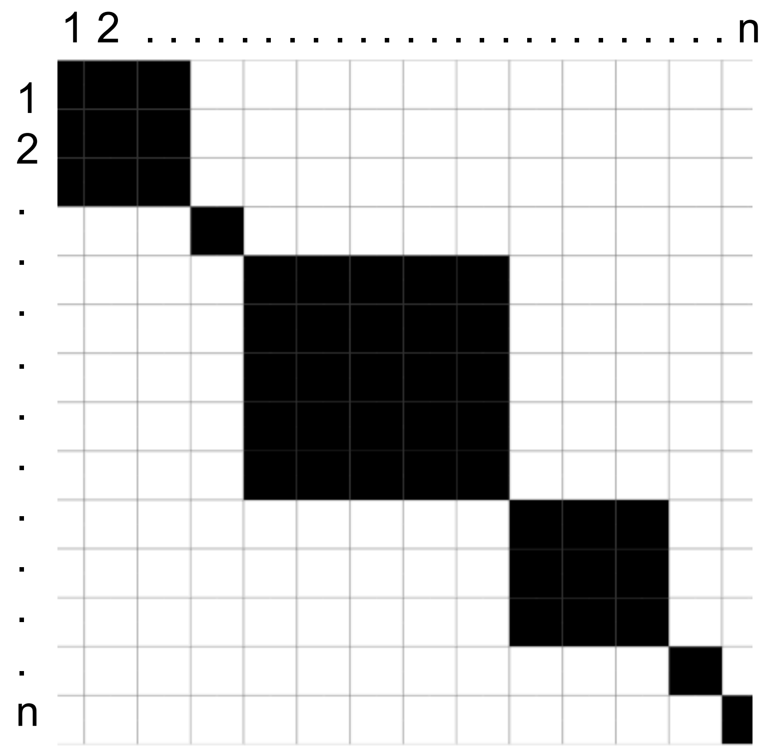
Clustering Reports – Computational cost

Problem: With n incident reports, we need $O(n^2)$ similarities

- n = 10 -> ~50 similarity computations
- n = 1 thousand -> ~500 thousand
- n = 100 thousand -> ~5 billion

Solution: Reduce search space with some form of *blocking*

- Intuition: dissimilarity on a single feature sometimes is strong evidence of dissimilarity overall
- Suppose there are 6 “event types” -> relatively few within-type similarities to compute
- Block on multiple variables separately to help avoid accidental exclusions
- How to block on set-valued variables, like the set of IOCs per record? -- Minhash



Clustering Reports – Random blocking in the set similarity

Suppose S_1 and S_2 are sets of IOCs in two incident reports

Experiment: Pick a random IOC $x \in S_1 \cup S_2$ and check whether $x \in S_1 \cap S_2$

What is $P(x \in S_1 \cap S_2)$?

Clustering Reports – Random blocking in the set similarity

Suppose S_1 and S_2 are sets of IOCs in two incident reports

Experiment: Pick a random IOC $x \in S_1 \cup S_2$ and check whether $x \in S_1 \cap S_2$

What is $P(x \in S_1 \cap S_2)$?

$$P(x \in S_1 \cap S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = Jaccard(S_1, S_2)$$

Run the experiment M times:

- let $k =$ the number of times that the selected IOC is in the intersection
- then $k/M \approx Jaccard(S_1, S_2)$

Minhash gives a way to quickly run these experiments for all pairs of reports

Clustering Reports – Random blocking in the set similarity

Suppose we run the experiment M = 5 times for n sets

Minhash output looks something like

$$minhash(S_1) = [h_{11}, h_{12}, \cdots, h_{15}]$$
$$minhash(S_2) = [h_{21}, h_{22}, \cdots, h_{25}]$$
$$\vdots$$
$$minhash(S_n) = [h_{n1}, h_{n2}, \cdots, h_{n5}]$$

Each column represents the outcome of one experiment.

If $h_{12} = h_{22}$, this says that the random element chosen in the 2^{nd} experiment was in the intersection of S_1, S_2.

Adjacent identical rows are where the experimental ratio k/M = 5/5 = 1, evidence that Jaccard(S_1,S_2) >> 0

Block on the minhash by simply running a sort operation – exact duplicates are candidates for a high Jaccard similarity.

Measuring similarity between cyber security incident reports

Evaluating the fortune tellers

IARPA CAUSE – Introduction

CAUSE – Cyber Automated Unconventional Sensor Environment

Objective: Develop techniques that use *unconventional* data sources to predict cyber attacks

Performer teams:

- Charles River Analytics
- Leidos
- BAE Systems
- University of Southern California Information Sciences Institute

Data providers provide real events: shhhhhh

Events and warnings:

- CAUSE encodes real events as JSON with a standardized field structure
- a warning is just like a GT event, but with a timestamp in the future

Part of the evaluation task: How do you measure the accuracy of a warning?

IARPA CAUSE – Giving credit to warnings

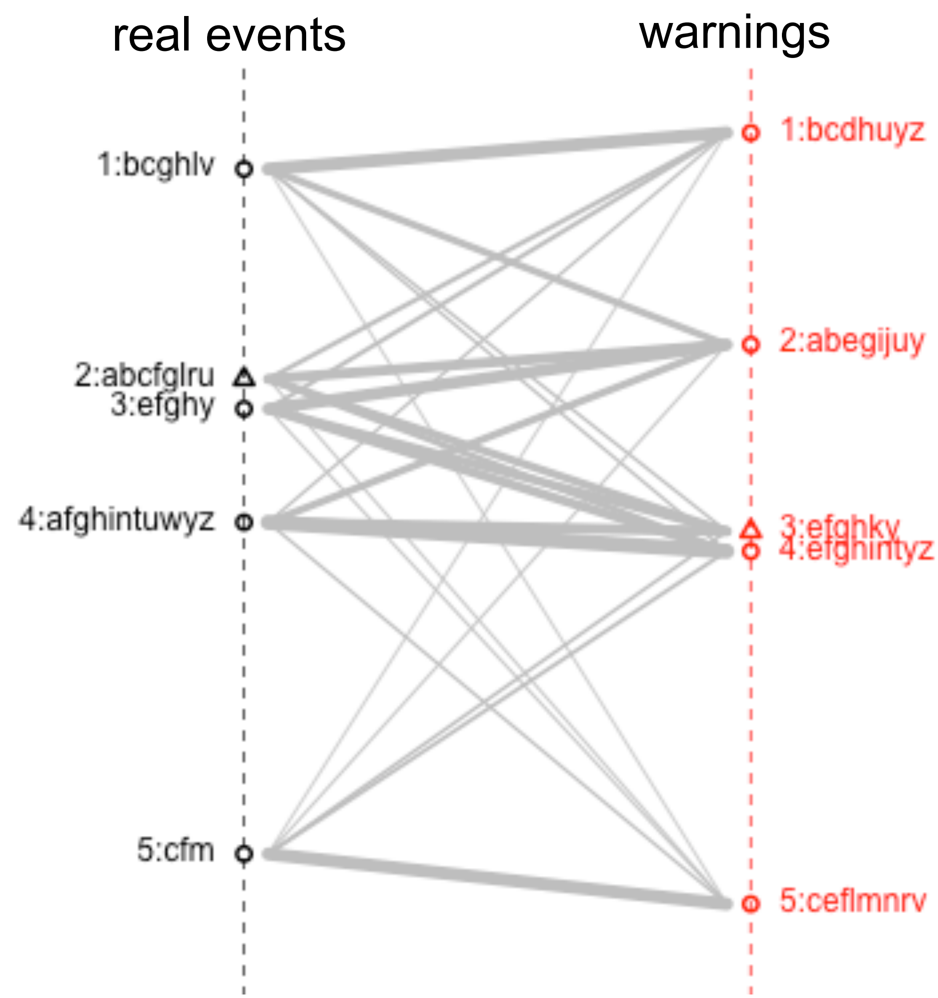
Warning evaluation simulator

Which warnings are potential matches for which GT events?

Reports have three attributes:

- timestamp (vertical axis)
- type (triangle, circle)
- details (random string)

Connection thickness = similarity



IARPA CAUSE – Giving credit to warnings

Warning evaluation simulator

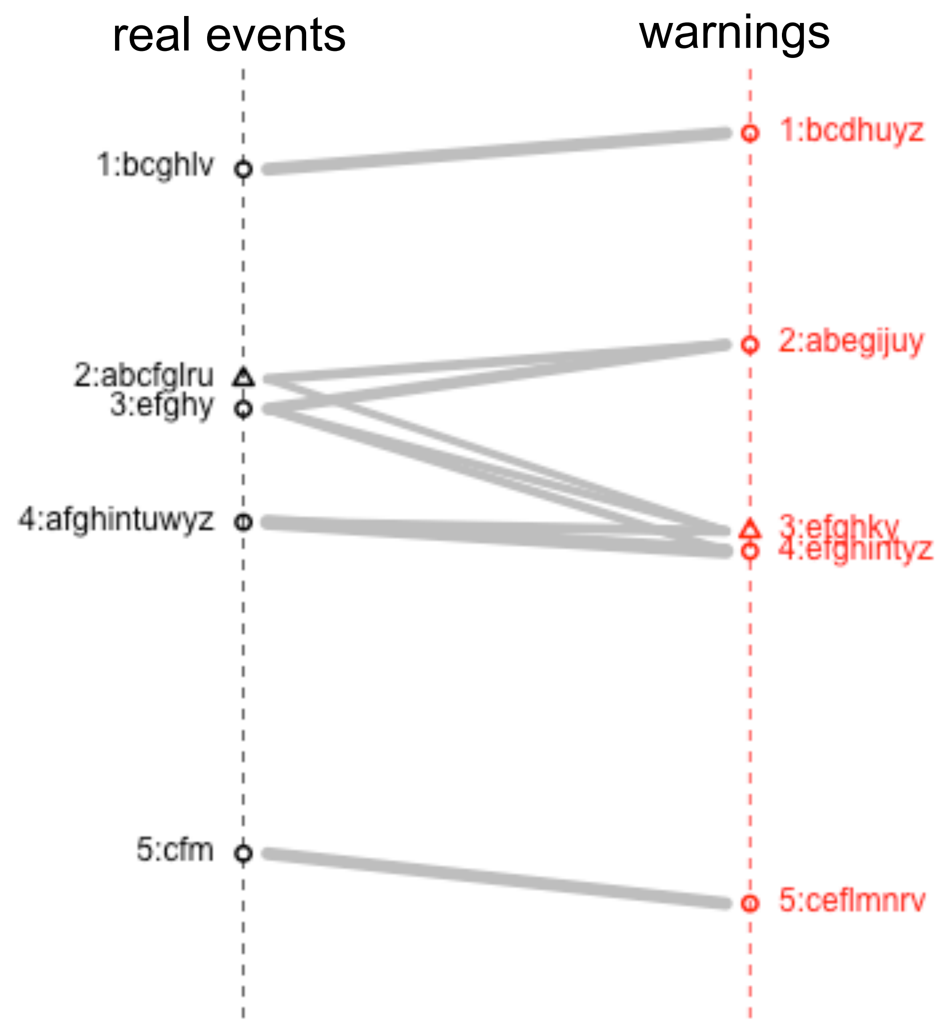
Which warnings are potential matches for which GT events?

Reports have three attributes:

- timestamp (vertical axis)
- type (triangle, circle)
- details (random string)

Connection thickness = similarity

Threshold removes weak connections



IARPA CAUSE – Giving credit to warnings

Warning evaluation simulator

Which warnings are potential matches for which GT events?

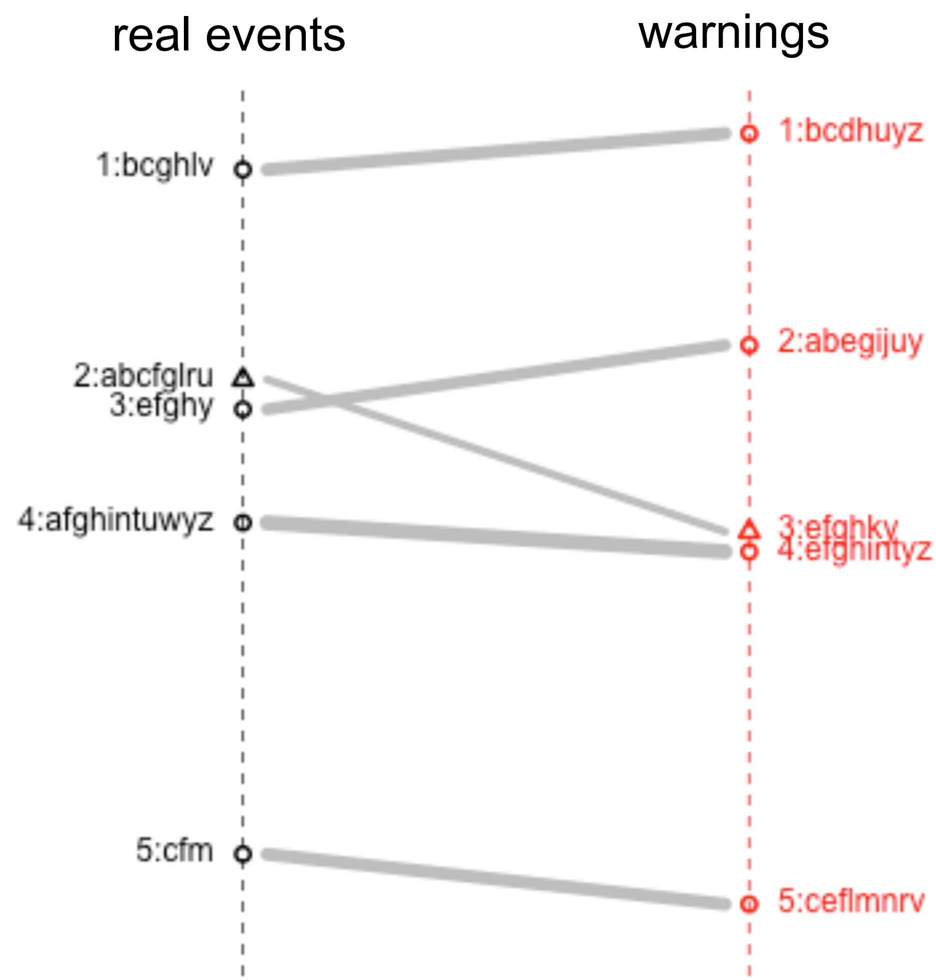
Reports have three attributes:

- timestamp (vertical axis)
- type (triangle, circle)
- details (random string)

Connection thickness = similarity

Threshold removes weak connections

Hungarian algorithm makes a final selection



IARPA CAUSE – Two approaches to computing recall

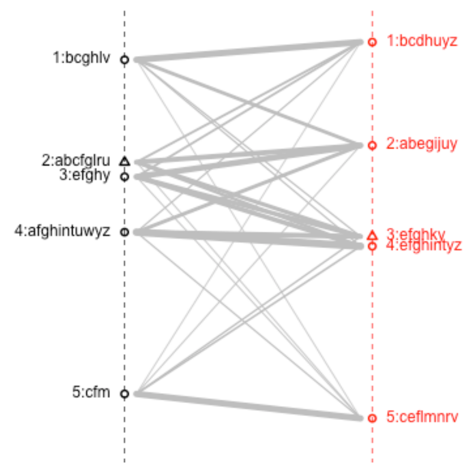
Suppose W is the set of warnings and E is the set of real events.

Recall is supposed to represent how much of E was warned about in W.

One-one matching

$$recall = \sum_{e \in E} 1_{matched}$$

Multi-way matching



$$recall = \sum_{e \in G} \max_{w \in W} sim(w, e)$$

Measuring similarity between cyber security incident reports

Soft Jaccard similarity

Soft Jaccard Similarity – Who needs soft?

An incident report can have many kinds of sets – not just one set of all its IOCs:

- set of phish recipient addresses
- set of ip-address IOCs
- set of file names
- set of event timestamps
- set of keywords used by analyst in free text comments

Suppose keywords A = {martian, martia, injection, numerous, c2}

keywords B = {mars, injected, repeated, remediated}

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{0}{9} = 0$$

But, intuitively, sim(A, B) > 0 .... ☹

Soft Jaccard Similarity – From hard to soft

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

What exactly in Jaccard is "hard"? What is there to soften?

- $(a == b)$ is 0 or 1, binary, or "hard"
- $size(A) = |A|$ uses $(A_i == A_j) = 0$ whenever $i \neq j$.
- $|A \cap B| = \sum_{a \in A, b \in B} (a == b)$

Soft Jaccard general idea:

- replace $(a == b)$ with $sim(a,b)$ an *element similarity function*
- replace $|S|$ with effective set size $ESS(S; sim)$
- replace $|A \cap B|$ with effective intersection size $EIS(A, B; sim)$

Examples of element similarity functions for strings a, b:

python: `from Levenshtein import distance as d`

$$sim_1 = e^{-\beta d(a,b)}$$

$$sim_2 = e^{-\beta d(a,b)/(len(a) * len(b))}$$

Soft Jaccard Similarity – Effective set size

What is the size of this set? A = {martian, martia, injection, numerous, c2}

Let M(A) be the similarity matrix of A:

	martian	martia	injection	numerous	c2
martian	1.000	0.888	0.060	0.013	0.0
martia	0.888	1.000	0.013	0.025	0.0
injection	0.060	0.013	1.000	0.012	0.0
numerous	0.013	0.025	0.012	1.000	0.0
c2	0.000	0.000	0.000	0.000	1.0

Borrowing from notions of "effective sample size" in statistics,

(https://golem.ph.utexas.edu/category/2014/12/effective_sample_size.html)

define ESS(A) as the sum of all elements of the inverse of M(A):

sum(

	martian	martia	injection	numerous	c2
martian	4.784	-4.247	-0.232	0.047	0.0
martia	-4.247	4.770	0.194	-0.066	0.0
injection	-0.232	0.194	1.012	-0.014	0.0
numerous	0.047	-0.066	-0.014	1.001	0.0
c2	0.000	0.000	0.000	0.000	1.0

) = 3.93

Soft Jaccard Similarity – Effective intersection size

A = {martian, martia, injection, numerous, c2}

B = {mars, injected, repeated, remediated}

Compute the inter-set similarities:

	mars	injected	repeated	remediated
martian	0.07	0.00	0.00	0.01
martia	0.17	0.01	0.01	0.01
injection	0.00	0.54	0.03	0.01
numerous	0.03	0.02	0.02	0.02
c2	0.00	0.00	0.00	0.00

Define the effective intersection size (EIS) as a weighted mean of the inter-set similarities:

$$EIS(A, B) = \sum_{i,j} w_{i,j} sim(A_i, B_j)$$

Soft Jaccard Similarity – Effective intersection size

$$EIS(A,B) = \sum_{i,j} w_{i,j} sim(A_i, B_j)$$

EIS weights:

Define the "redundancy" $R(A_i)$ as the row sum of the ith row of $M(A)$.

Define the "uniqueness" $U(A_i)$ as $1/R(A_i)$.

Intuition: similarities involving the most unique elements should get more weight in the EIS:

$$w_{i,j} = ESS(A)ESS(B)\frac{U(A_i)U(B_j)}{\sum_{i,j} U(A_i)U(B_j)}$$

$$SoftJaccard(A,B) = \frac{EIS(A,B)}{ESS(A) + ESS(B) - EIS(A,B)}$$

Soft Jaccard Similarity – Examples

```
JaccardSimilarity(['goodbye', 'hello'], ['pandas', 'numpy'])
```

```
0.00
```

```
JaccardSimilarity(['goodbye', 'hello'], ['goodbyes', 'hellos'])
```

```
0.37
```

```
JaccardSimilarity(['a', 'b'], ['a', 'as', 'bs'])
```

```
0.25
```

```
JaccardSimilarity(['jonathan', 'johnathan', 'sally'], ['sally'])
```

```
0.46
```

```
JaccardSimilarity(['dog', 'cat'], ['dog', 'cat'])
```

```
1.00
```

Review

Reasons to measure similarity between incident reports include:

- Identifying records of similar attacks during active investigations
- Identifying campaigns or other groups of incidents
- Evaluate warnings for real events

First steps to building your own similarities:

- Pick the features that matter to you: free text key words, incident type, time of incident, IOC sets, etc.
- Define element similarity functions to compare any two specific items
- Define “set similarities” to compare sets of items
- Combine all of the component similarities above into a single weighted sum or other aggregate similarity

Contact Information

Presenter / Point of Contact
Zach Kurtz

Data Scientist

Telephone: +1 412.268.8689

Email: ztkurtz@sei.cmu.edu