Domain generation and blacklists – a problem?

Leigh Metcalf, Jonathan Spring

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense 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.

DM19-0108

Intro

Domain Generation Algorithms (DGA):

- Algorithmic generation of domain names, generally to avoid blacklisting
- Made famous by conficker back in 2008

We did a comparative analysis of open-source DGA detectors

- Will provide conceptual results

We followed up on the question as to what impact DGA has on the shape of the blacklist ecosystem as a whole

- Thanks to the bright folks at NASK for first raising this question

Blacklist results in question

Each blacklist is context and topic specific

- This is based on conversations here as well case studies

Therefore blacklists do not overlap

This is our explanation; the non-overlap is well documented

- Go to https://resources.sei.cmu.edu/library/results.cfm and search “blacklist”
- Reports covering Jan 2012 through Dec 2017
- “Do not” means
    - of the roughly 35 million indicators every 6 months
    - 95% to 98% are unique to one list
        - *(if we account for IP address churn)

95% to 98% are unique to one list

One natural question is

- Is this uniqueness caused by blacklisting generated domains?

We find DGAs do not interfere with this conclusion

What we mean by DGA

An algorithmically generated set of names, generally used to mean generated with the intent of abuse

Effective second-level domains only

- Example.co.uk or example.com
- NOT looking for internal or organizational weird management

Really, though, we wondered how many DGA detection algorithms there are and whether they agree with each other

DGADA!

Study investigates 17 DGA detection algorithms

Requirements

- Open source
- Implementable in our environment

Research question:

Given known sets of DGA names, how do the DGADA compare?

The point:

- If the DGADA are distinct and blacklists use different ones
- then blacklists differ by design (as expected, along with context)
  - not by random sampling chance / error

Surprise!

DGAs, and thus DGADAs, are also context and topic specific

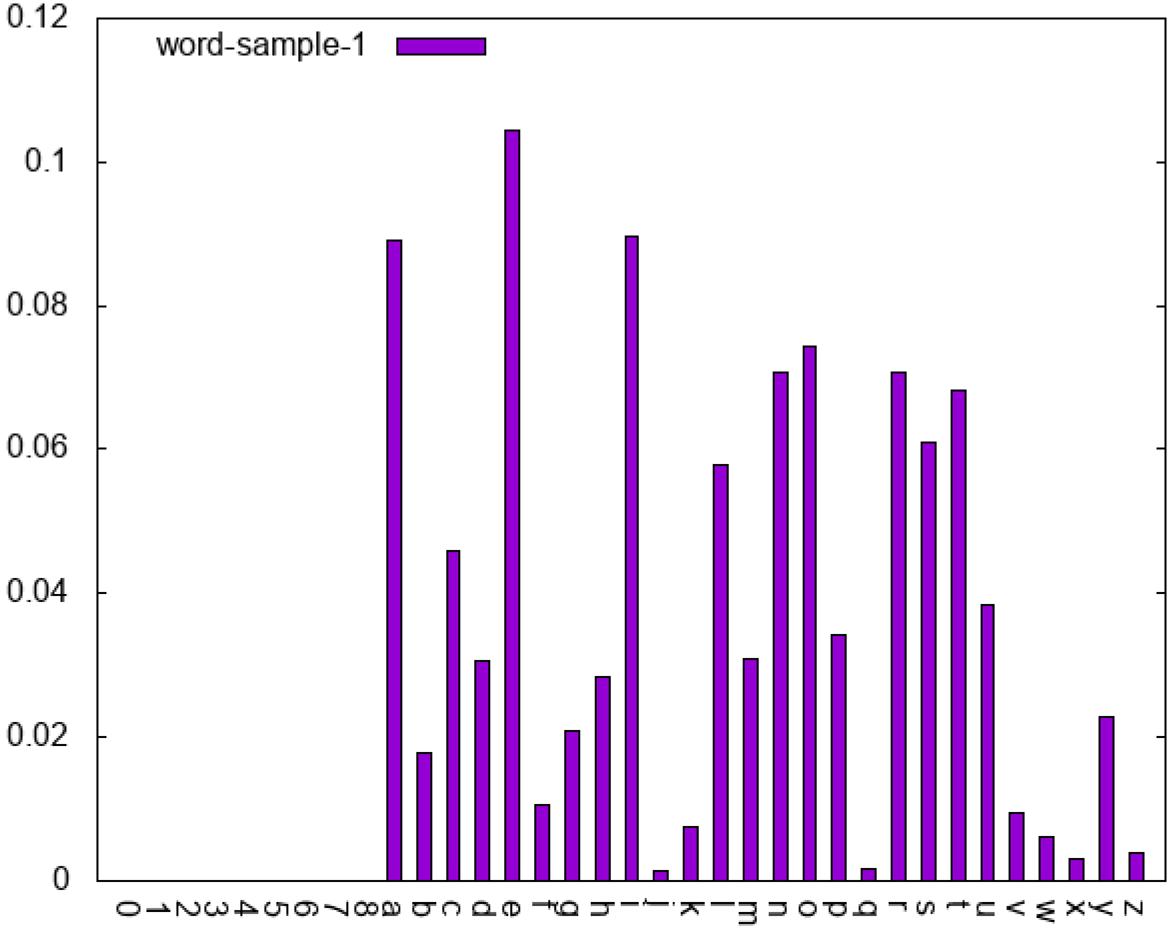
To test, we collected about 30 malware-generated sets of DGAs

- As well as domain names from alexa, English text, blacklists, and sampling from a uniform distribution over characters

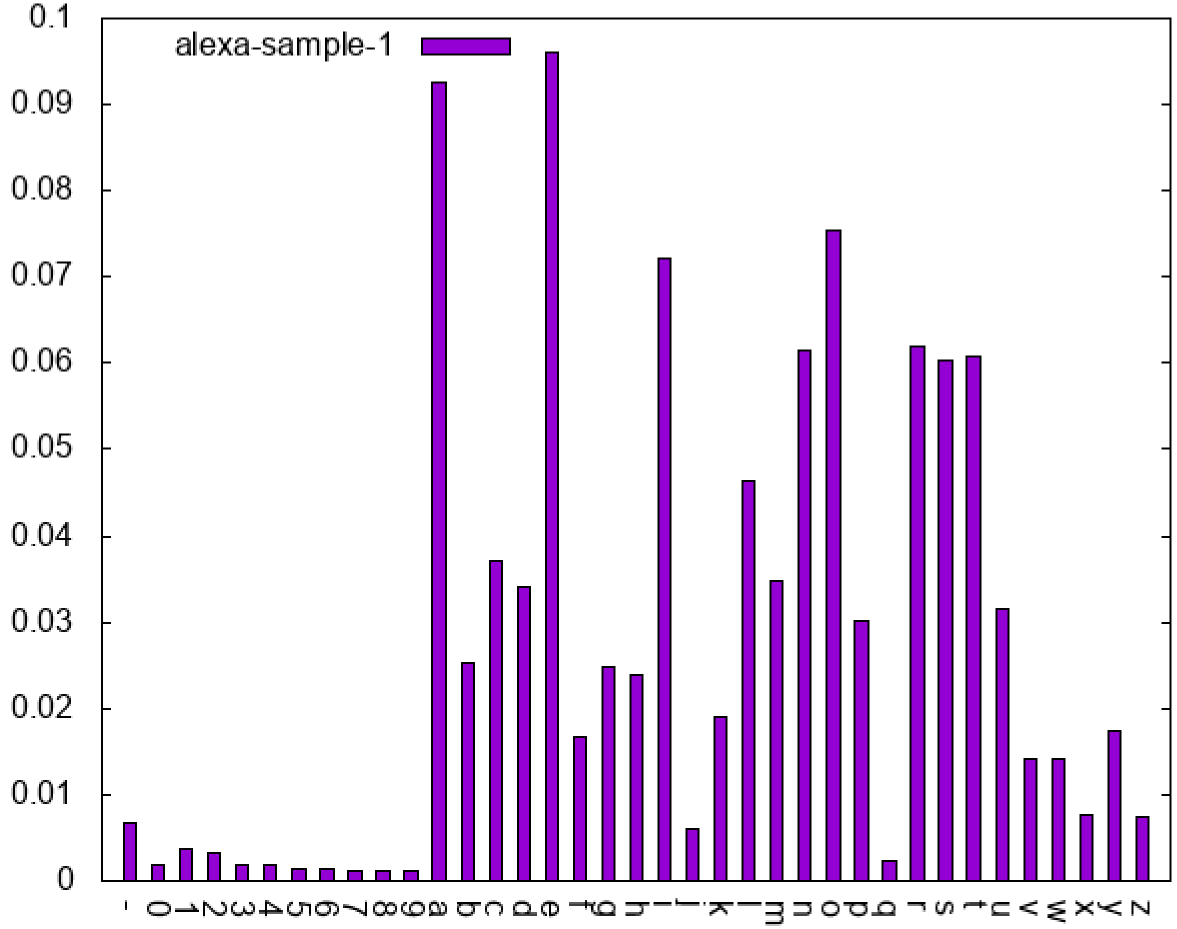
DGAs largely have distinct probability distributions over characters from which they draw

This is easier to visualize

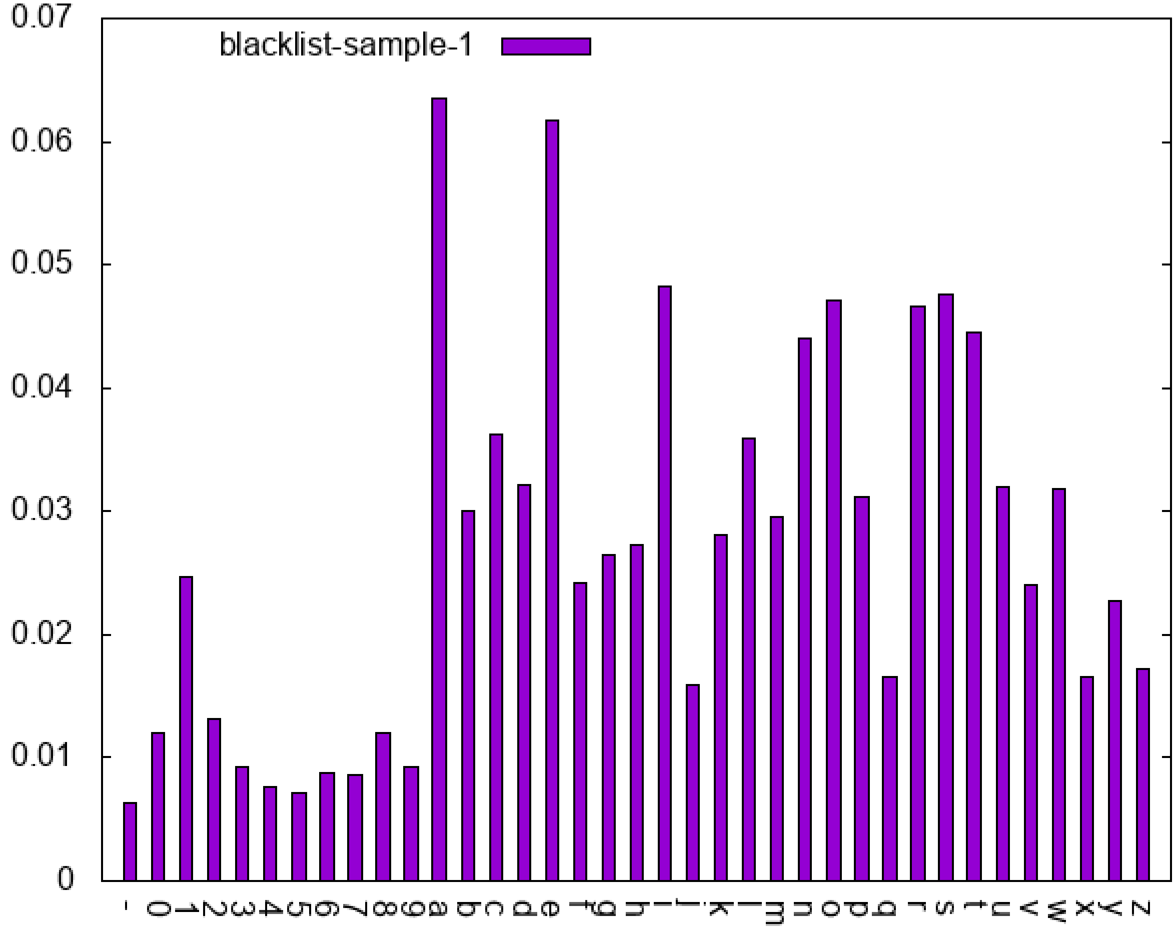
Character distribution in English dictionary words



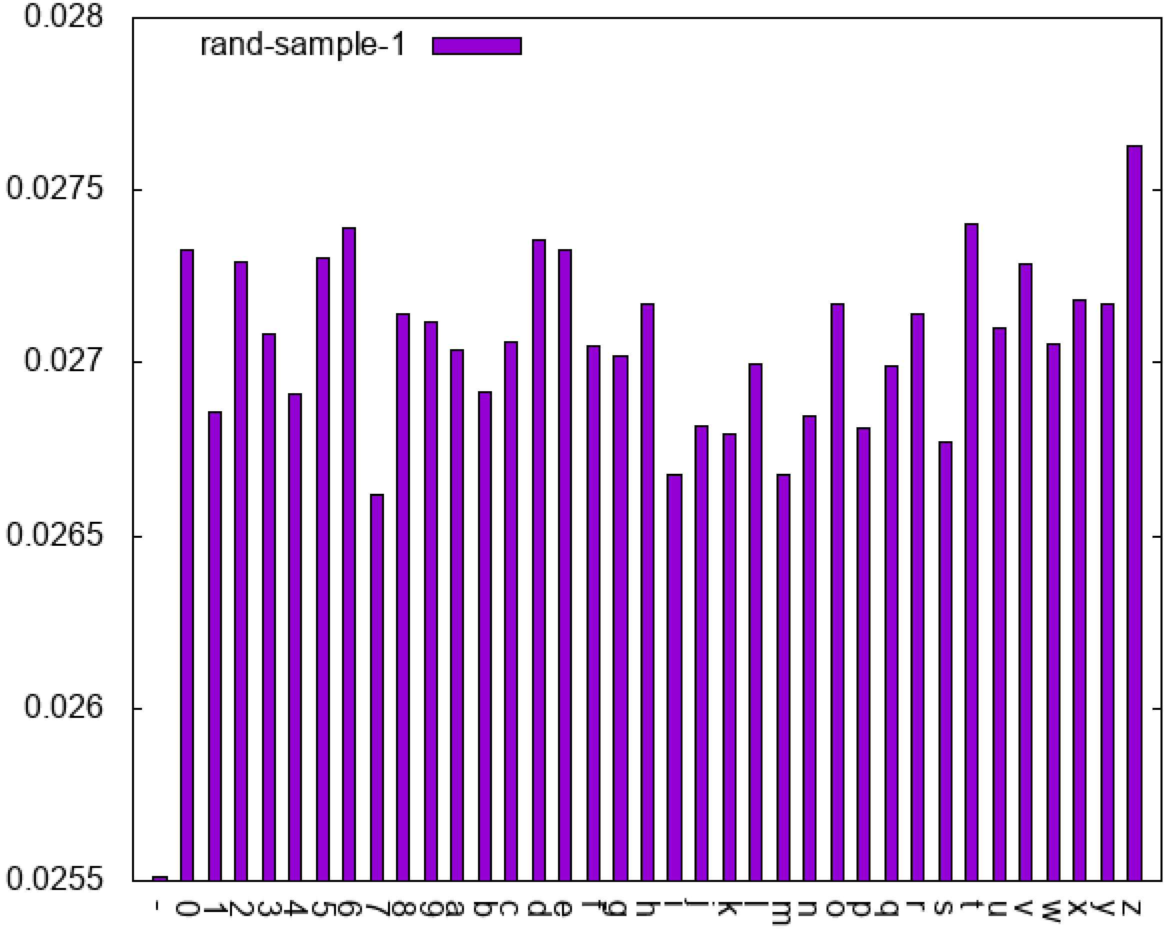
Distribution – Alexa 1M domains (2017 H1)



Distribution, ~30 blacklists over 6 months



Sample characters as drawn from a uniform random distribution



Random is not all the same

It’s old news that English text does not have a uniform distribution of characters

- Shannon CE. Prediction and entropy of printed English. Bell system technical journal. 1951 Jan;30(1):50-64.
- Colloquially, English letters are not random

But “random” does not have a solid meaning in statistics

We “draw from a distribution”

- Distributions come in lots of shapes
- “random” ≈ drawn from the uniform distribution
- But there are lots of other distributions that are different from both the uniform and the usual English distribution
  - Alexa list, for example

Now, for some malware DGAs

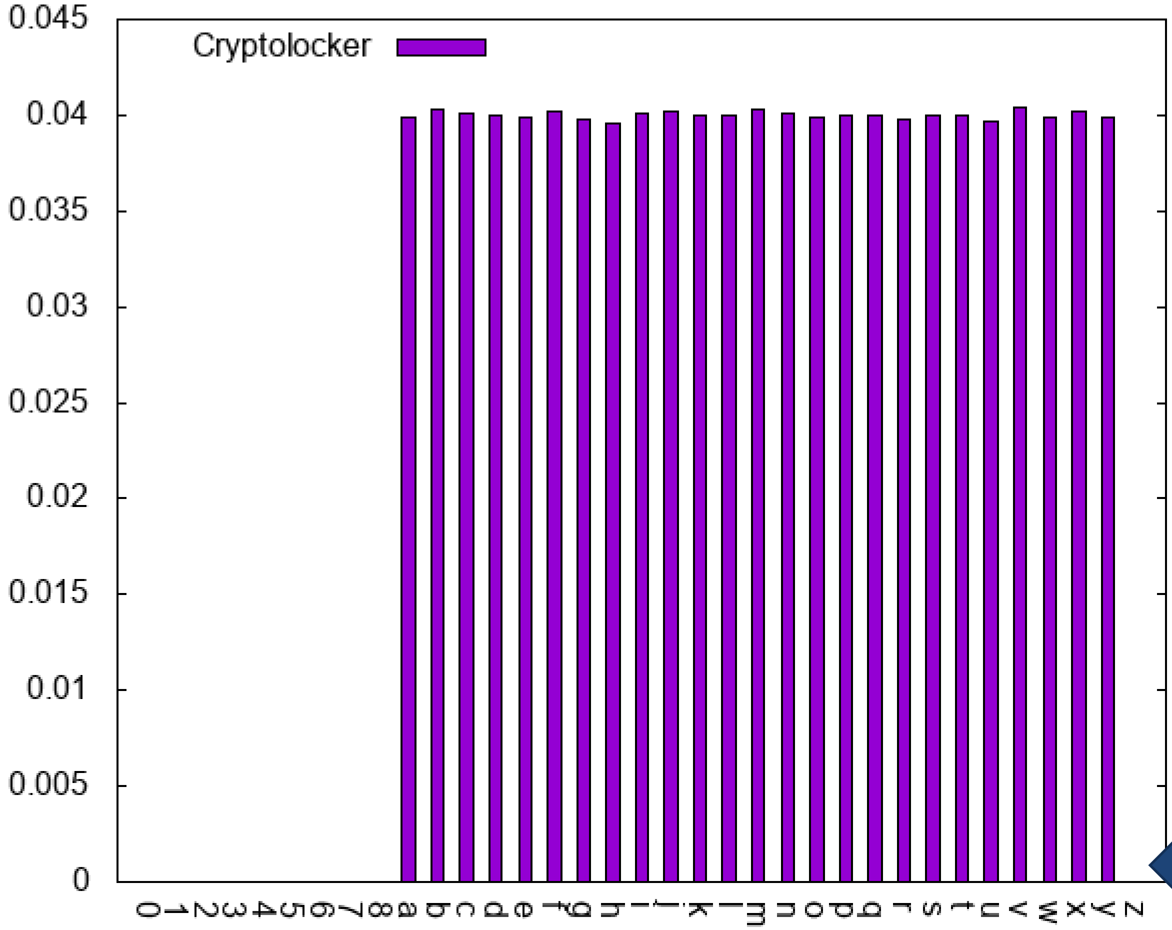
For example, let's look at:

- Cryptolocker
- Dyre
- Murofet
- Suppobox

Try to think of what is both in common among these, and also different from the alexa / English distributions

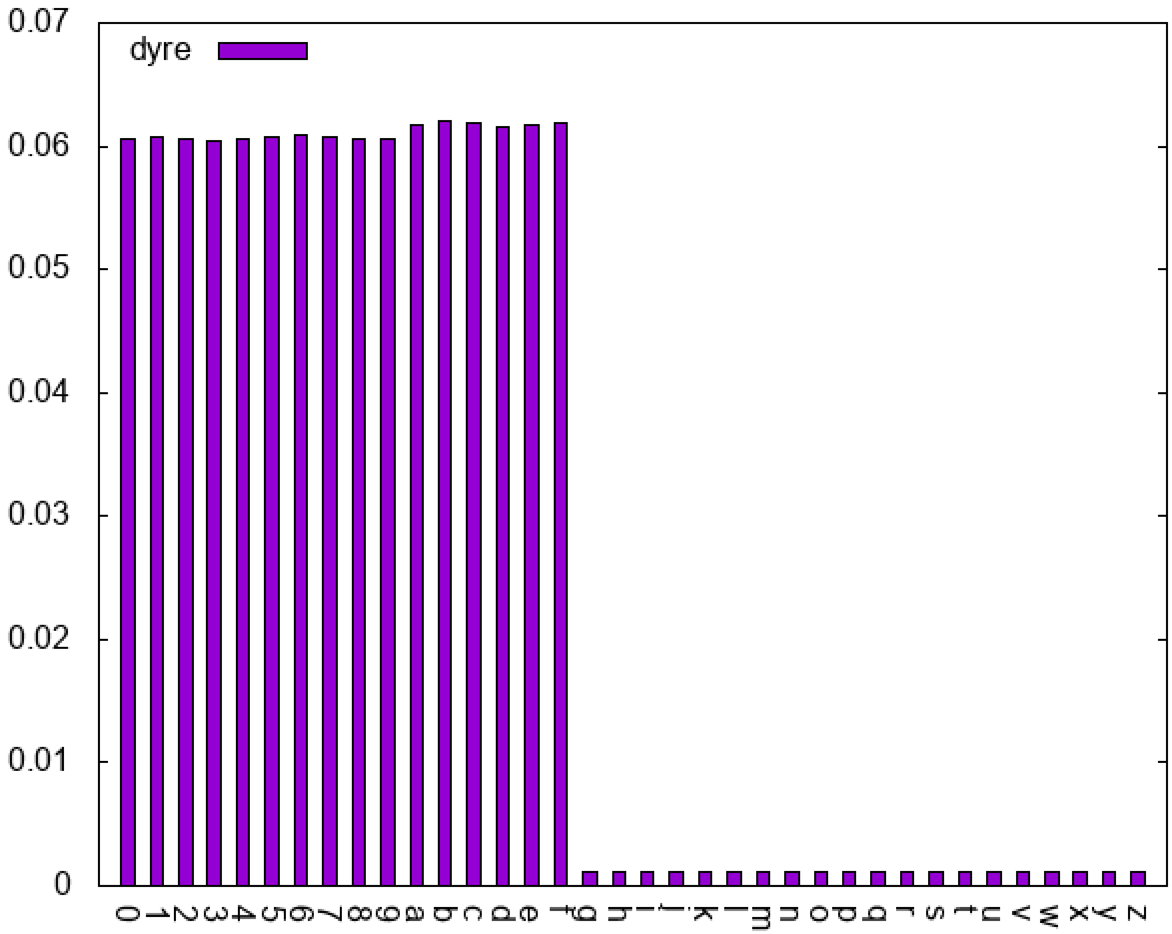
- This is what a general DGADA would need to do

Cryptolocker



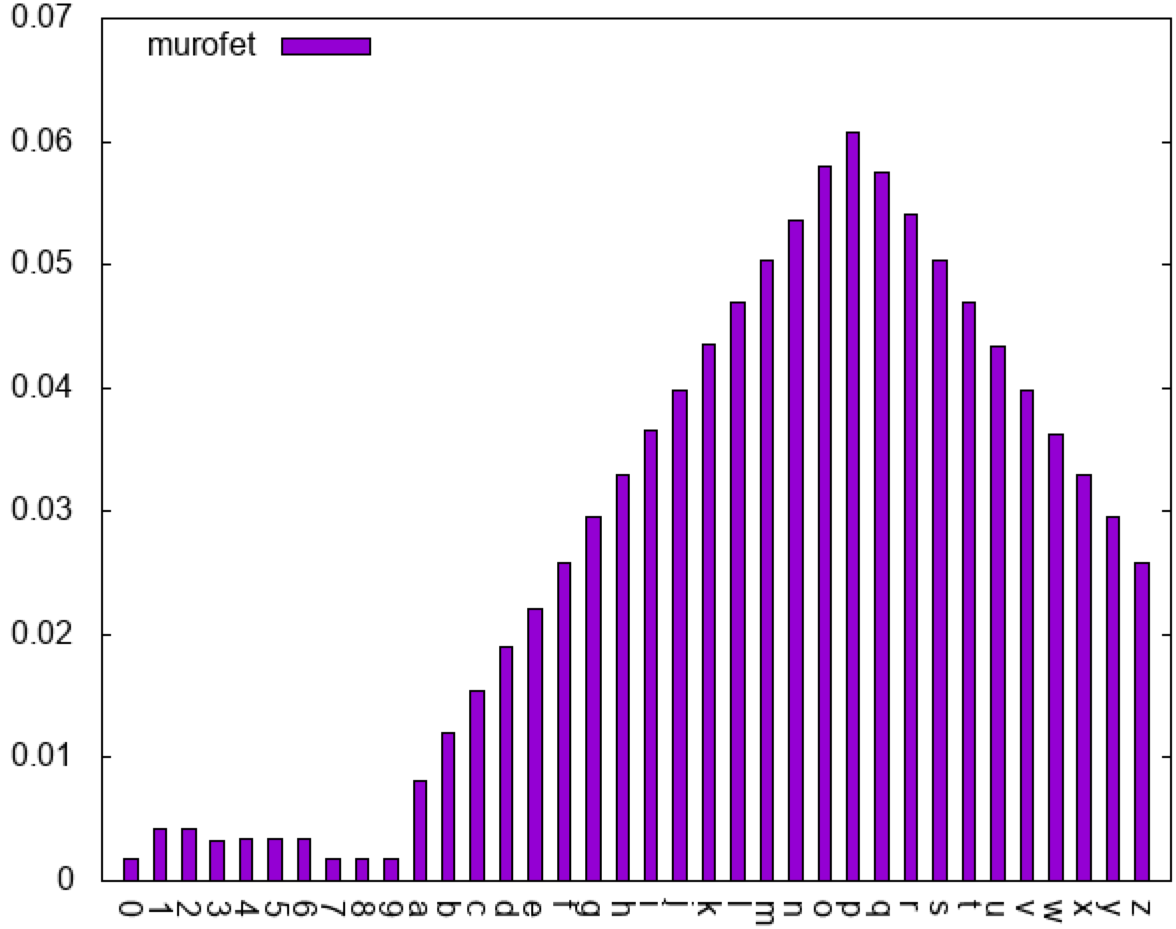
Note: no Z

Dyre



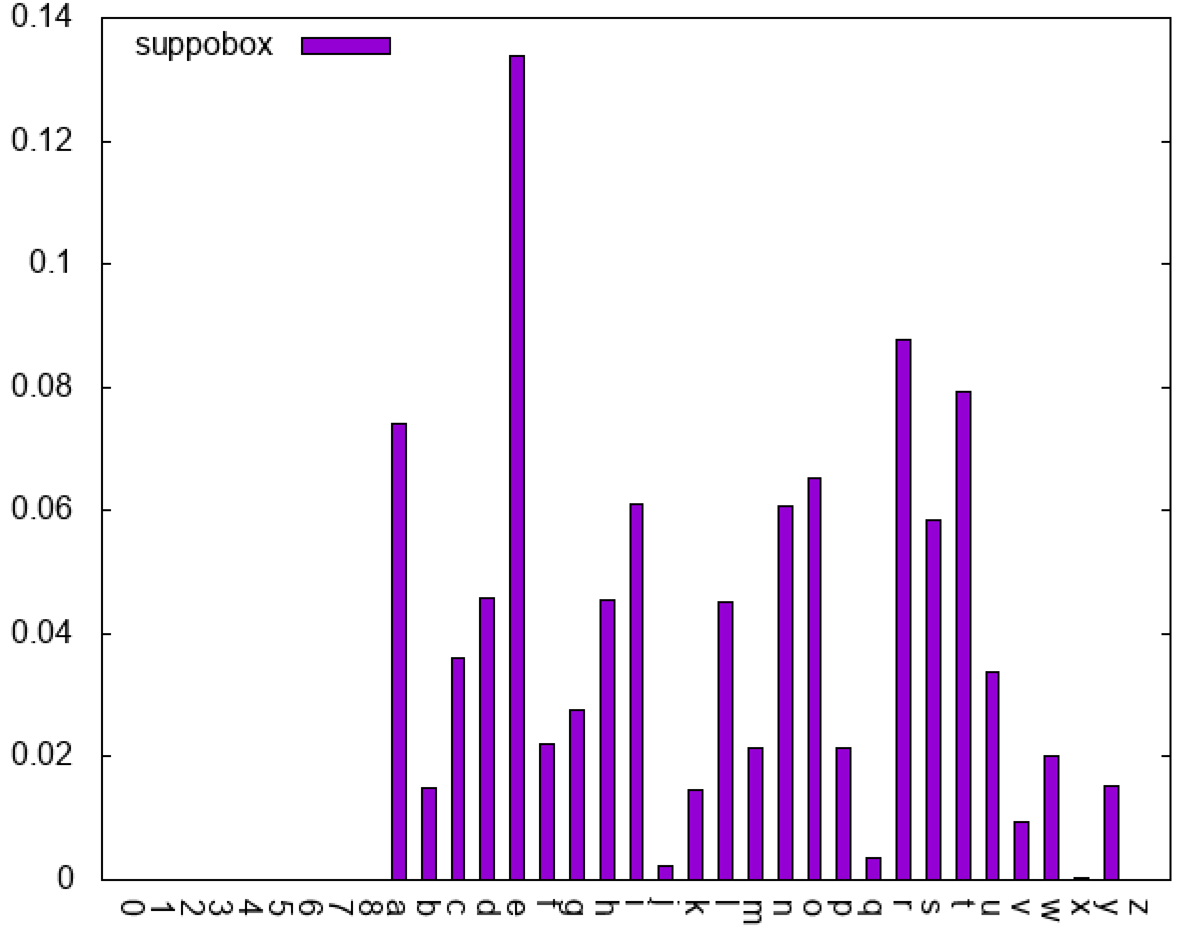
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
dyre
0 1 2 3 4 5 6 7 8 9 a b c d e f g h i j k l m n o p q r s t u v w x y z

Murofet



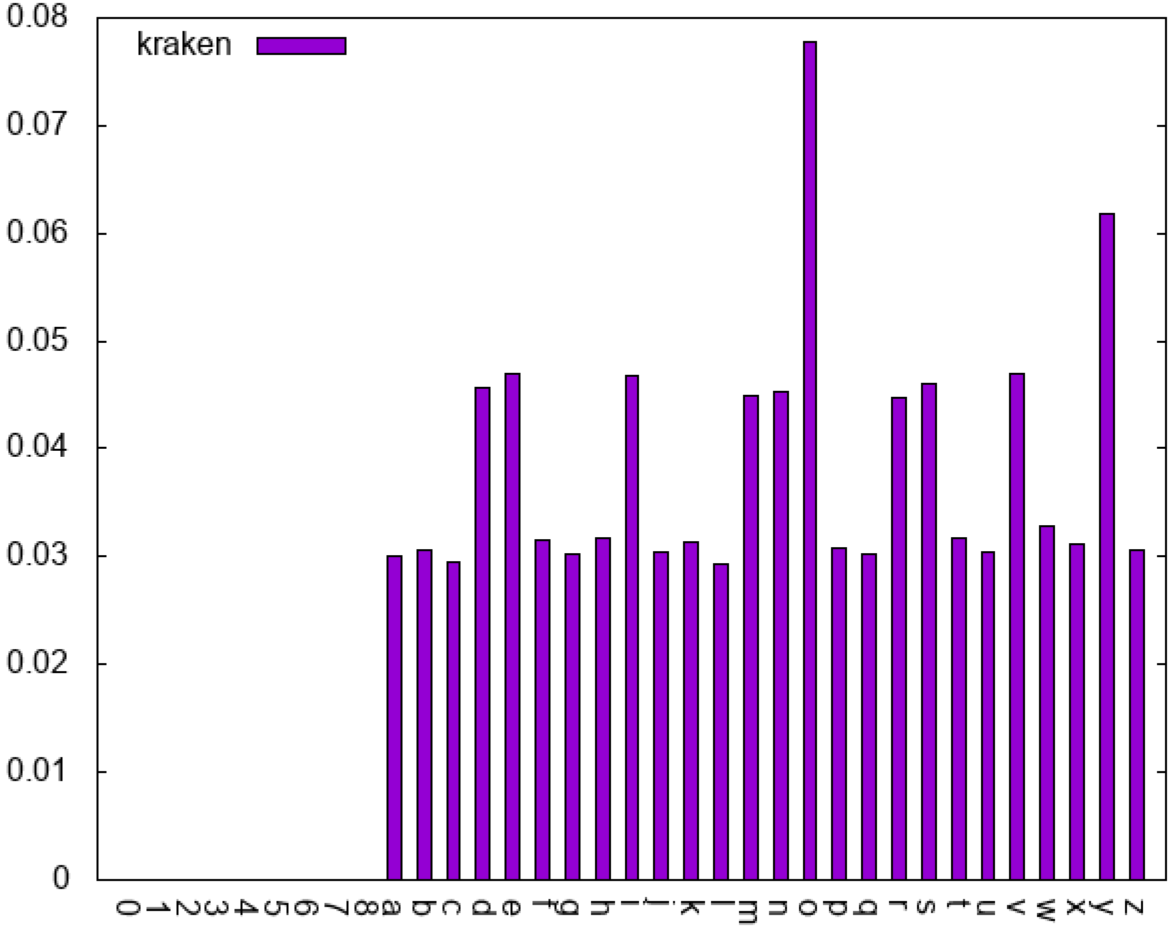
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
murofet
0 1 2 3 4 5 6 7 8 9 a b c d e f g h i j k l m n o p q r s t u v w x y z

Suppobox



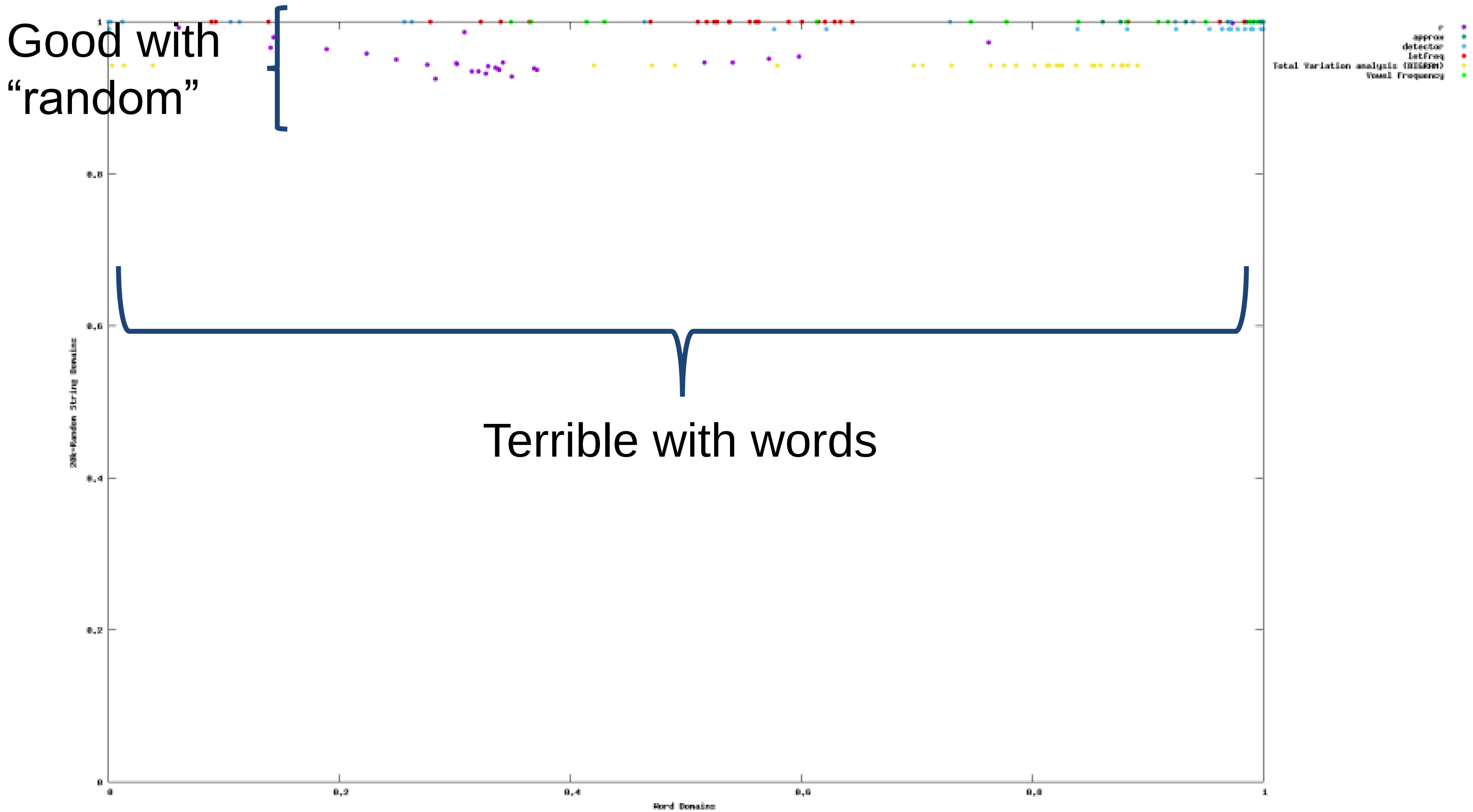
suppobox
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0 1 2 3 4 5 6 7 8 a b c d e f g h i j k l m n o p q r s t u v w x y z

Kraken



kraken
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0 1 2 3 4 5 6 7 8 a b c d e f g h i j k l m n o p q r s t u v w x y z

So how well do the 17 DGADAs do?



Positive Likelihood Ratio

Measures the rate of accurate alerts

For example, a LR+ of 3 means:

- For every three domains correctly identified as DGA
- One domain is incorrectly identified as DGA

Does not include how many DGA domains a detector misses

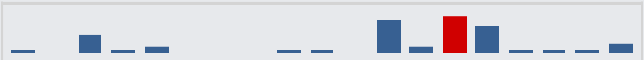
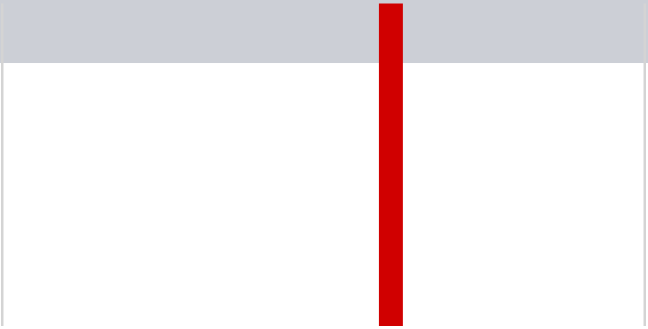
So how well do the 18 DGADAs do?

Select average positive likelihood ratios for DGADAs across 30 malware DGA samples

DGA detection algorithm	Average likelihood ratio
Bayesian_analysis_UNIGRAM	8.000
Entropy_analysis_BIGRAM	70.368
entropy	7.384
detector	2325.061
ngram	53.271
Probability_analysis_UNIGRAM	12.571

Are some malware DGAs harder to detect?

Select average positive likelihood ratios against DGAs across 17 detection algorithms

Malware DGA	Average likelihood ratio	Distribution of likelihood ratios
Cryptolocker	29.781	
dyre	30.178	
murofet	28.252	
suppobox	0.779	
kraken	426.543	

Some are better than others

Wide range on which is best per malware

Is this a good ratio?

It depends on the base-rate at which these DGAs occur in the DNS

Consider this example

- My DNS server sees 10,000 unique domain requests
- 334 of them are Cryptolocker domains
- With a LR+ ≈ 30
- You'll get about 660 alerts on Cryptolocker DGA domains
- Half are erroneous alerts

1/2 is much better than the base rate of 3/100

But is it enough to take incident response actions?

General DGA detection?

No detector was uniformly best

Some detectors may be better or worse on average

A good DGA detector usually focuses on detecting a specific DGA's distribution

Good detection of one DGA does not seem to transfer to good detection of another

Constructing a detector of DGAs generally does not appear to work

DGA detector weaknesses

Each DGA detector has blind spots for existing malware DGAs

The cause is, roughly, that these are not the distributions you're looking for

Kristina Alexanderson, CC BY-NC-ND 2.0

Slightly mathy-er

The various DGAs have sufficiently different probability distributions it is not plausible to construct a representative distribution from which they all pull that can be used for general detection

AKA:

DGA detection algorithms do not substantially overlap

- Nicely mirrors the “blacklists do not substantially overlap” conclusion

Overlap between all blacklist domains and DGA detectors

We checked whether DGA broadly interferes with blacklist overlap

- For each DGA, we ran it against a set of 11 domain blacklists and checked the overlap among the domains marked as DGA

For 2017H2, the rate all blacklisted names are on multiple lists:

# lists	Names on X lists	% of total
1	39921011	97.7407%
2	725387	1.7760%
3	133892	0.3278%
4	37629	0.0921%
5	15395	0.0377%
6	7161	0.0175%
7	2658	0.0065%

Difference from DGA marking

Average difference of names on exactly one list:

- -1.61 percentage points (that is, 96.13%)
- Range: -0.34 to -2.75 percentage points
  - Out of names marked as DGA, so it should be independent of the number of names the detection algorithm marked as DGA

Most of the increase showed up as names on exactly 2 (of 11) lists

- Average increase: +1.26 percentage points (that is, 3.03%)
- Range: +0.26 to +2.17 percentage points

Difference from DGA marking

On the one hand, the largest impact of a DGA detector doubles the number of domains on more than one blacklist

One the other hand, it goes from 2.5% of names on more than one list to 5%.

Thus as a relative effect, it has an impact

As a question to whether the blacklist-non-overlap is real

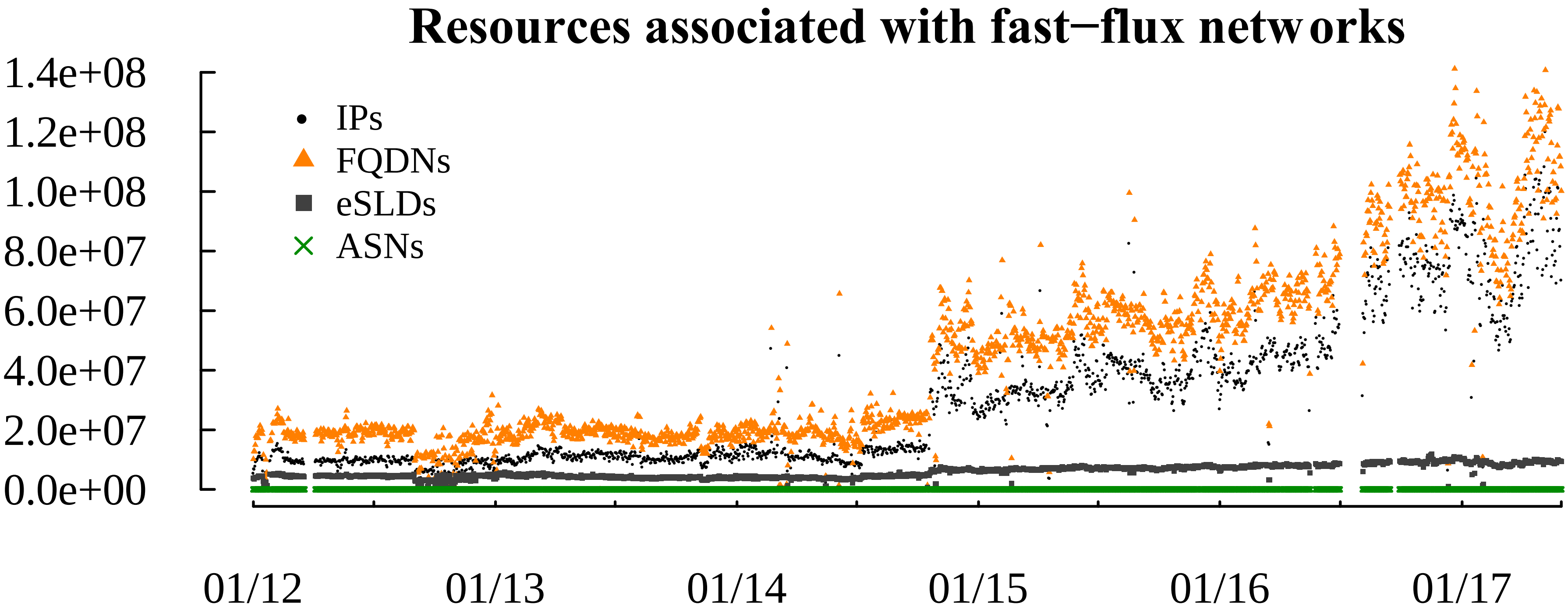
- Yes, it’s still real

A note on IPs, fast flux, and DGA

Variable DGA domains avoid IP blocking via fast flux, for example

Free fast flux detection tool: Analysis pipeline

- See “Open-source Measurement of Fast-flux Networks While Considering Domain-name Parking”



Summary

DGAs do not interfere with the conclusion that each blacklist is context and topic specific

- DGA detectors are also context and topic specific

Impact and recommendations

- “random” ≠ bad
- DGA detectors can work against specific malware’s DGA distributions
- There are too many domains even if we block algorithms (https://insights.sei.cmu.edu/cert/2014/10/domain-blocking-the-problem-of-a-googol-of-domains.html)
- We cannot totally fix blacklisting by accounting for DGA’s better
- Get all the blacklists and DGA detection algorithms and use them in the right places with the right context for the right purpose
- Use other protections in addition to blacklists

Questions?

Thanks for your time!