Scrutinizing a Country using Passive DNS and Picviz

or how to analyze big dataset without loosing your mind

PICVIZ®
LABS

CIRCL
Computer Incident
Response Center
Luxembourg

Sebastien Tricaud,
Alexandre Dulaunoy

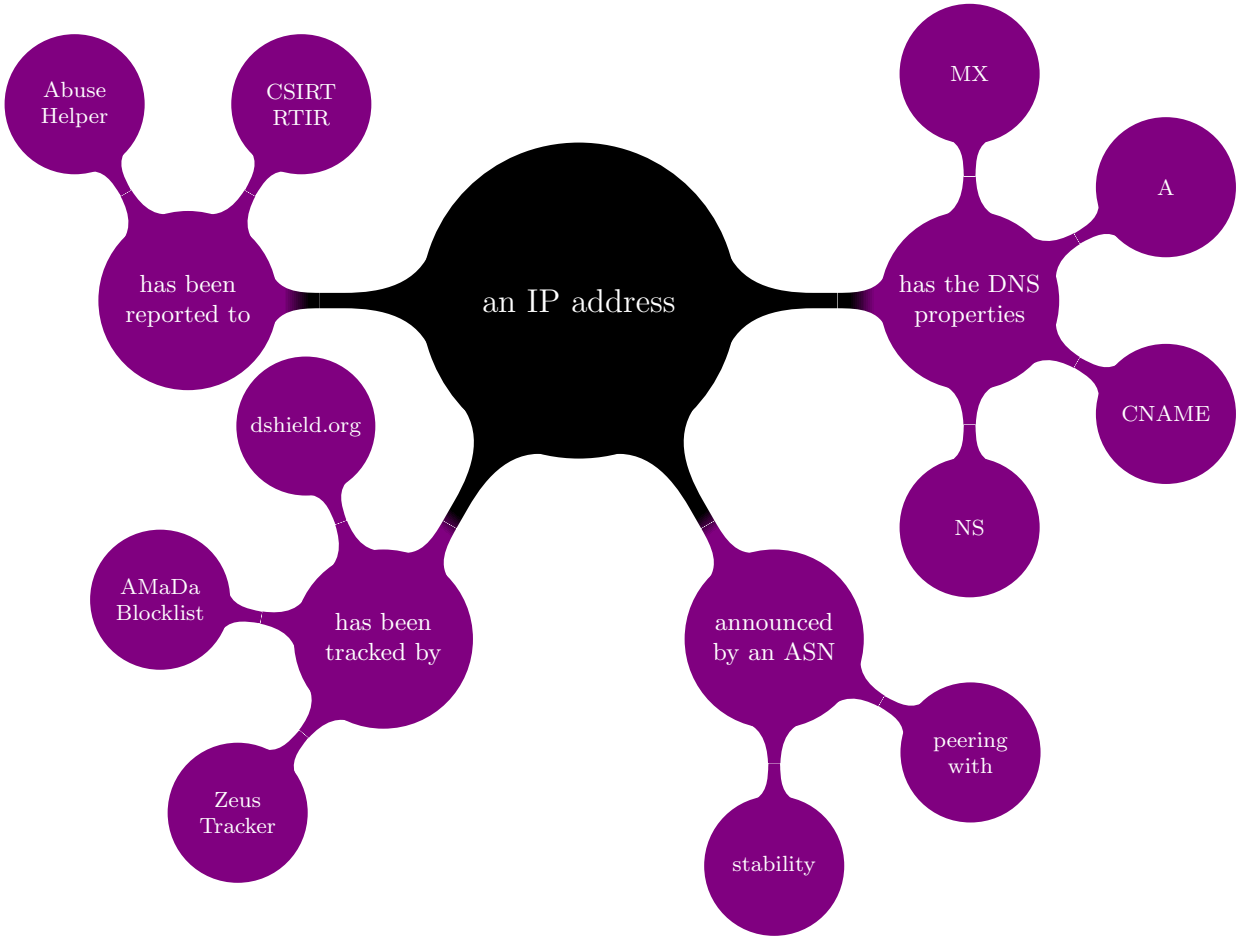
March 10, 2012

Disclaimer

- Passive DNS is a technique to collect only valid answers from caching/recursive nameservers and authoritative nameservers
- By its design, privacy is preserved (e.g. no source IP addresses from resolvers are captured[1])
- The research is done in the sole purpose to detect malicious IP/domains or content to better protect users

[1] Except if the web application abused DNS answers to track back their users.

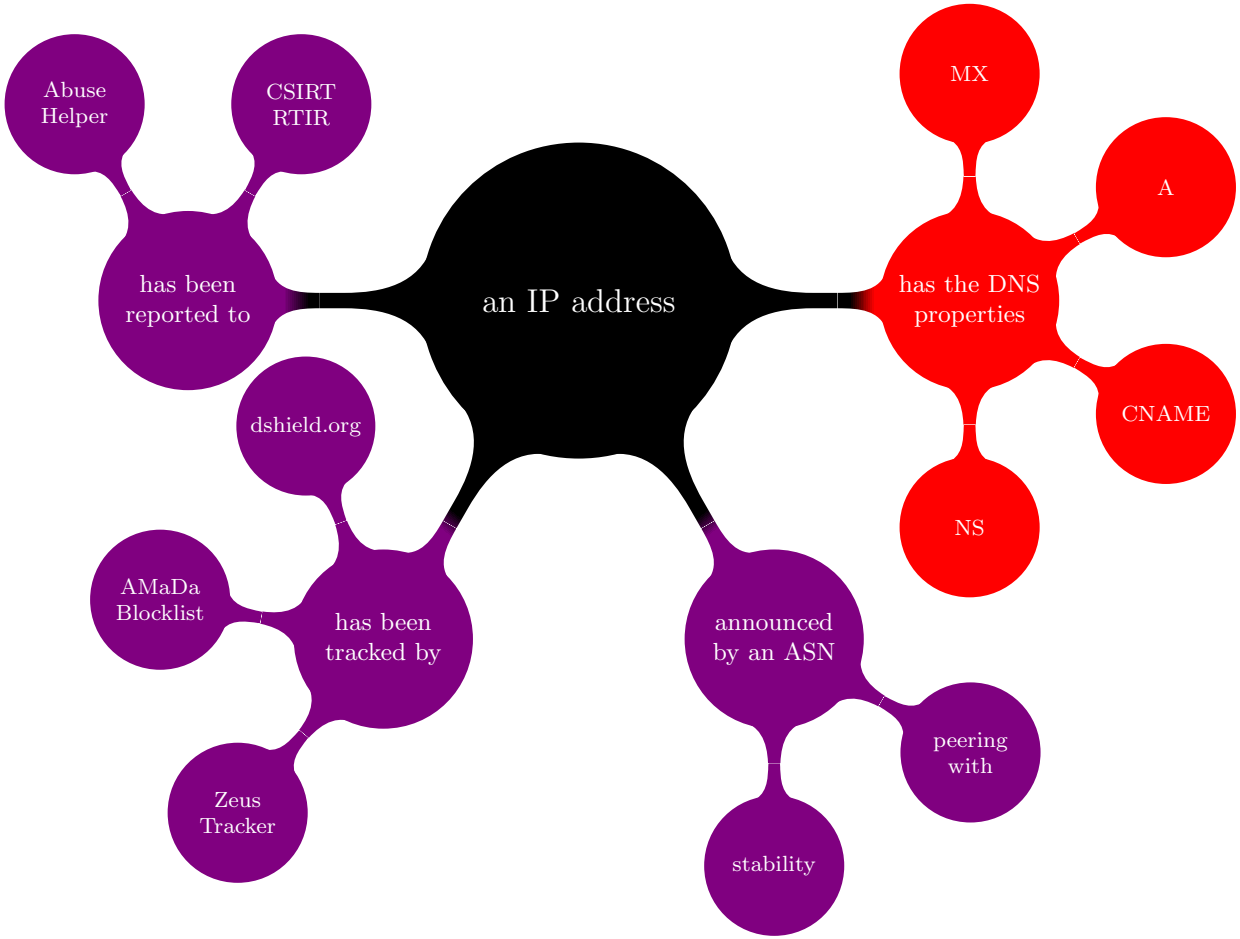
IP overview - some properties



Introduction or Problem Statement

- Datasets become larger and larger (even for a small country)
- Malicious (and non malicious) activities are distributed across IP addresses or domain names
- Time to live of Internet resources (especially the malicious ones) is low
- $\rightarrow$ Attackers abuse and benefit from these facts

Passive DNS

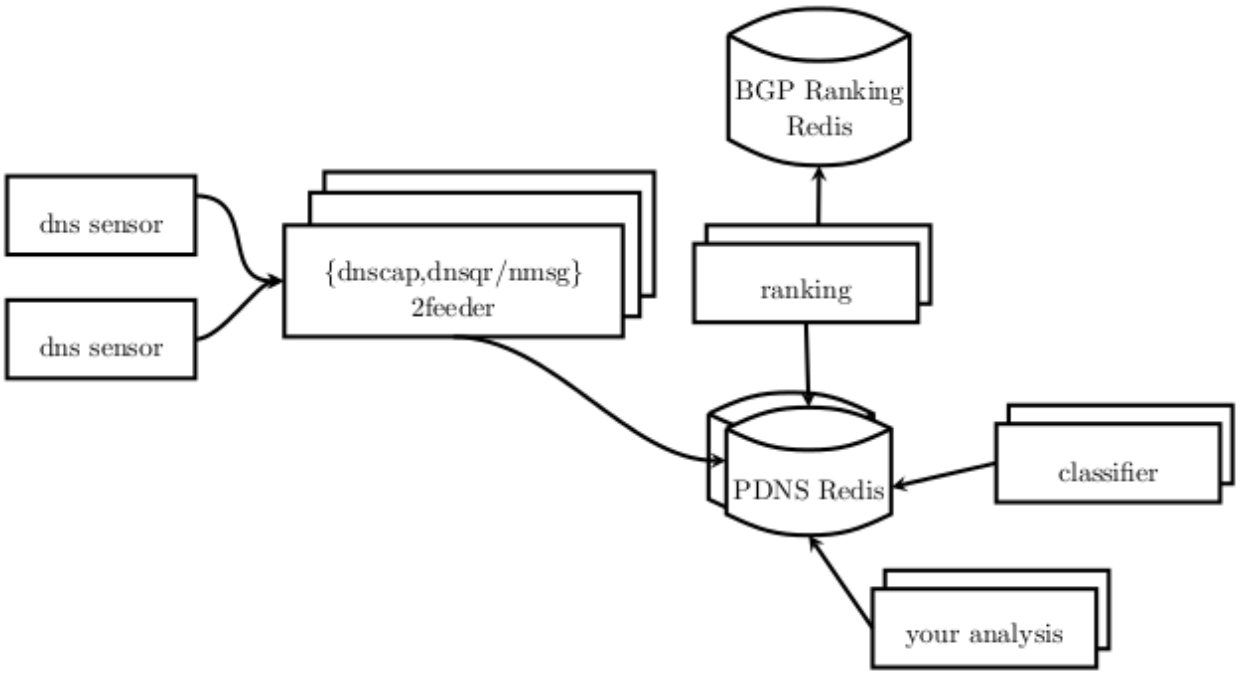

Abuse Helper
CSIRT RTIR
has been reported to
an IP address
MX
A
has the DNS properties
CNAME
NS
dshield.org
AMaDa Blocklist
has been tracked by
Zeus Tracker
announced by an ASN
peering with
stability

Storing Passive DNS or how to do trial and error?

- Implementing the storage of a Passive DNS can be challenging
- Starting from standard RDBMS to key-value store
- We learned to hate[2] hard disk drive and to love random access memory
- Loving memory is great especially when it's now cheap and addressable in 64bits

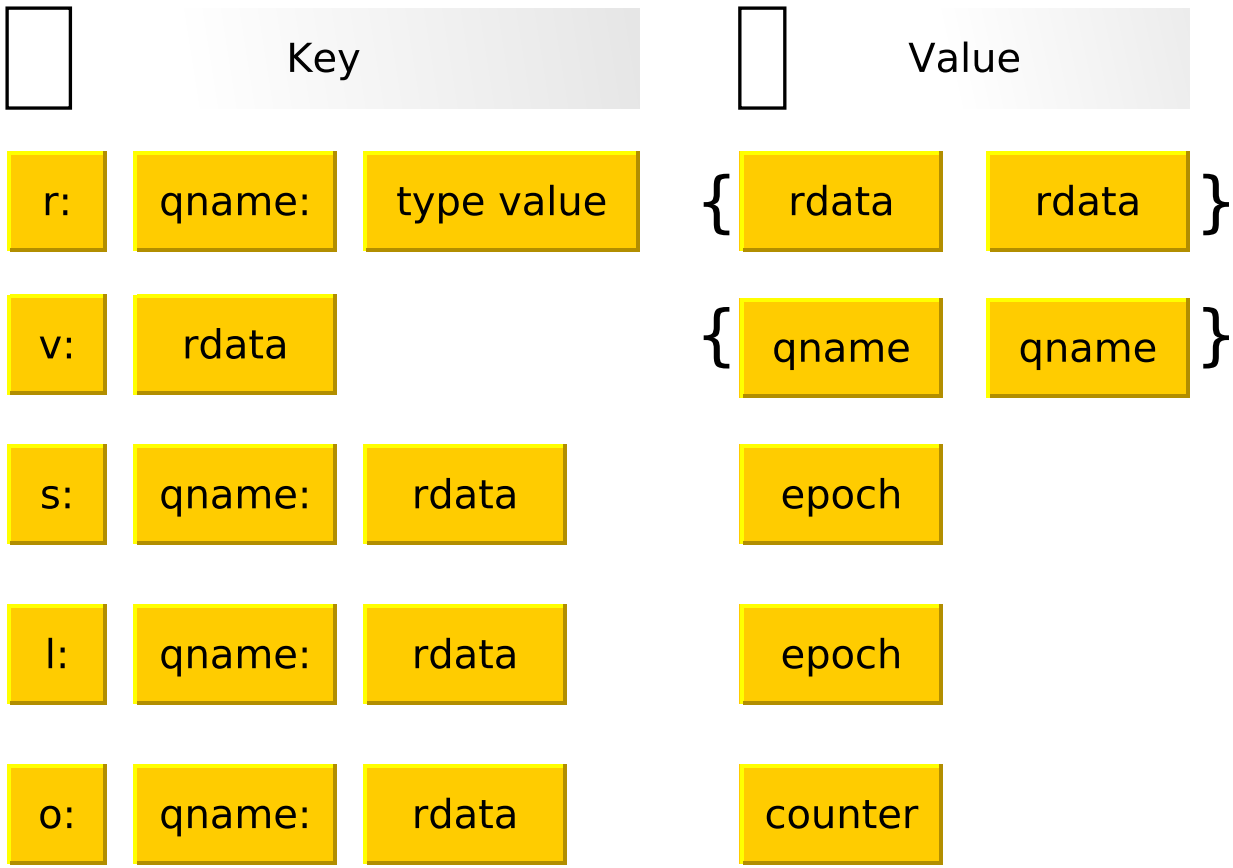
[2] exception $\rightarrow$ only used for data store snapshot

A minimalist and scalable implementation of a passive DNS



- Our passive DNS implementation is a toolkit for experimenting classification or visualization techniques

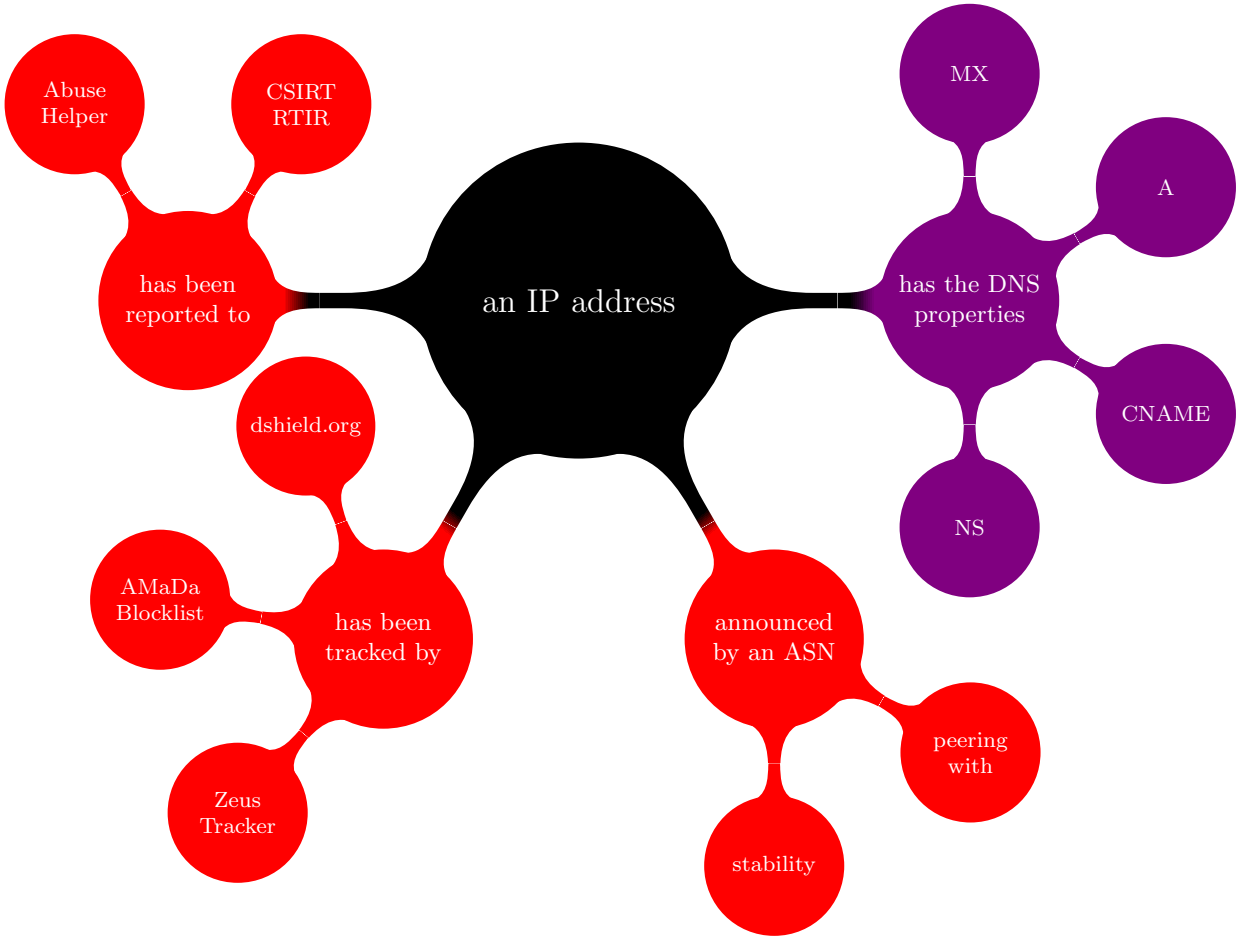
Redis - Passive DNS data structure



Redis - a sample query

```
redis> SMEMBERS "r:www.linkedin.com:5"
1) "dub.linkedin.com"
redis> SMEMBERS "r:dub.linkedin.com:1"
1) "91.225.248.80"
redis> SMEMBERS "v:dub.linkedin.com"
1) "www.linkedin.com"
redis> GET "s:www.linkedin.com:dub.linkedin.com"
"1331057300"
redis> GET "l:www.linkedin.com:dub.linkedin.com"
"1331057412"
redis> GET "o:www.linkedin.com:dub.linkedin.com"
"3"
```

BGP Ranking on IP attributes



AS Ranking Calculation

Formula

$$AS_{rank} = 1 + \left(\frac{\left(\sum_{s=1}^{\#s} (\text{Occ} \;\; S_{impact}) \right)}{AS_{size}} \right)$$

- Number of malicious occurrence per unique IP (Occ)
- Weight of the blacklist source (S_{impact})
- Grand total of IP addresses announced by the ASN (AS_{size})
- Each iteration of the Occ sum is saved (e.g. to discard a source blacklist from the ranking calculation)

Why Ranking ISPs?

- CSIRTs can assess the level of trust per ISPs (e.g. know to host drive-by-download website, reactive to abuse handling, ...)
- Improve assessment between ISPs (e.g. IP peering policies)
- Detecting common suspicious activities among ISPs/ASN
- Can be used as an additional weight factor to abuse handling (e.g. detect outliers in large set of IP addresses)

A daily use: ease your log analysis

- 300 million lines of proxy logs? You have 30 minutes to find out what's happened? or discarding the noise of "known" malware communication?
- Prefix the ranking AS15169,1.00273578519859,74.125.... to the log file
- logs-ranking $\rightarrow$ sort -r -g -t"," -k2 proxy.log-ranked

A daily use: ease your memory dump analysis

- During large incident, we got many memory dumps in a single day
- Dumping all the memory per process and we extracted all URLs and IPs from each memory dump
- Ranking URLs and IPs, and analyzing the processes with the higher malicious rank
- Ranking can be used for a lot of reverse analysis techniques (from finding malicious process to artefacts of antivirus in memory)

Ranked domains - Where Picviz can help

- Now, we have 50 millions lines of ranked hostname...

```
...
www.stopacta.info. = 1.0
www.vista-care.com. = 1.0
breadworld.com. = 1.00002301767
o-o.resolver.A.B.C.D.5xevqnwsds5zdq34.metricz.\
l.google.com. = 1.00303388648
www.thechinagarden.com. = 1.00009822292
smtp10.dti.ne.jp. = 1.00010586629
...
```

Detection of multi-homed compromised systems

- Regularly malicious links are posted on compromised systems
- Ranking increased for the ASN and its announced subnet
- Passive DNS collects associated hostnamed to a subnet (usually filling the gap in the subnet)
- But how to find thoses cases?

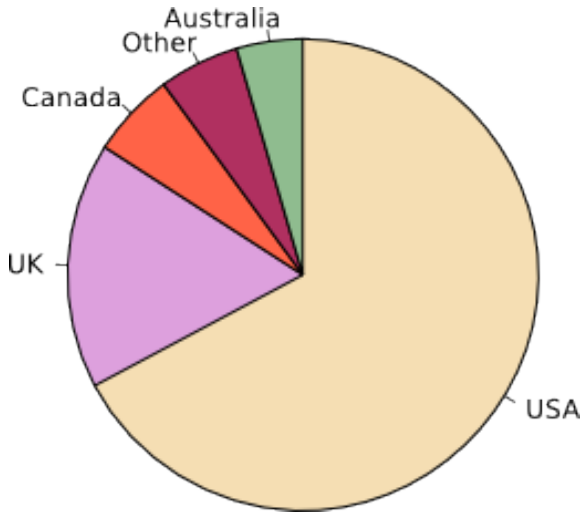
Ooops wrong visualization

- For the ones who were at the party ;-)

Why visualization?

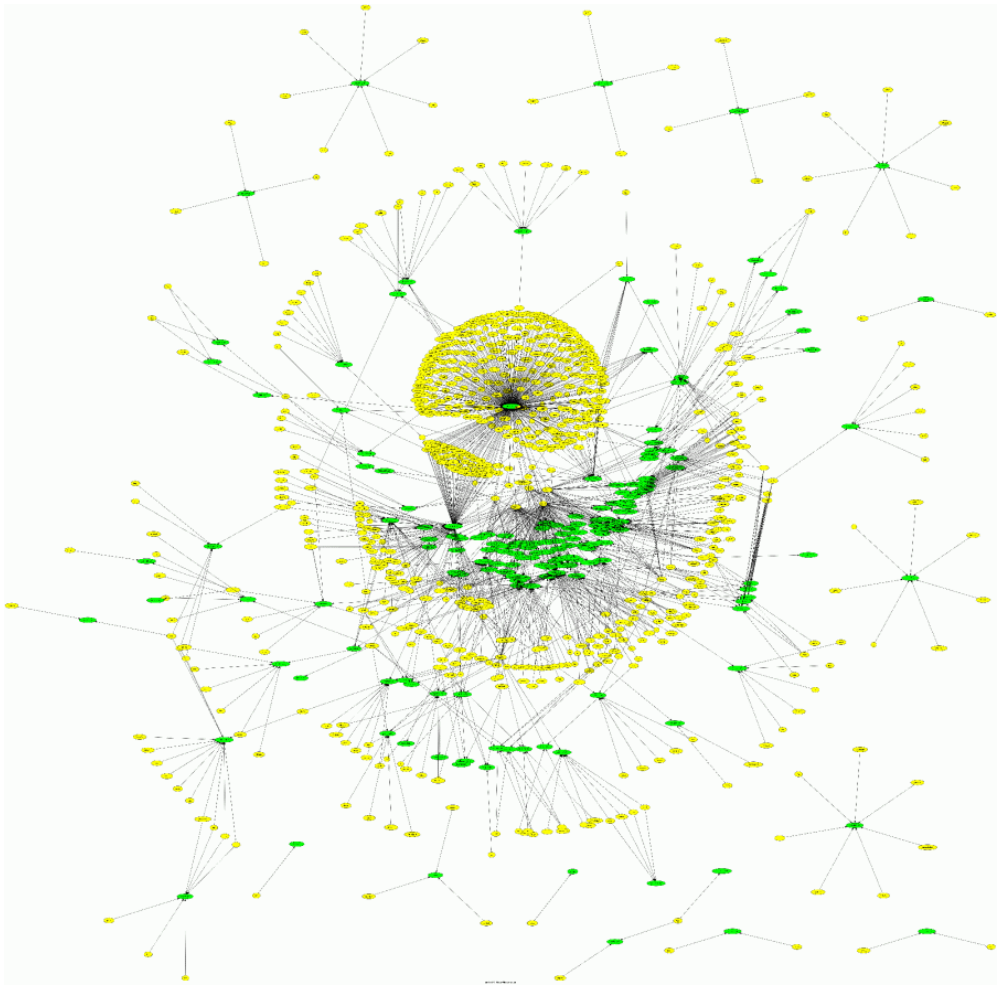
- Understand big data
- Find stuff we cannot guess

Problem with usual visualizations

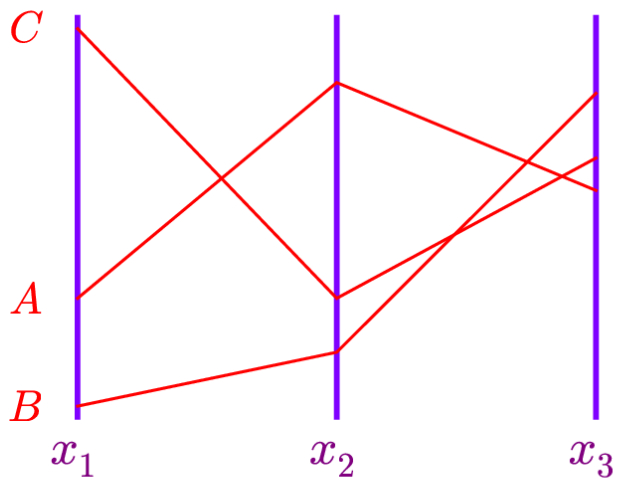


- Limited
  - Top 10 (!)
  - Just to display tendencies...
  - Hide most of information
- Hard to get meaningful/useful information
- Folks mostly use it to display stuff in a different way

Problem with usual visualizations

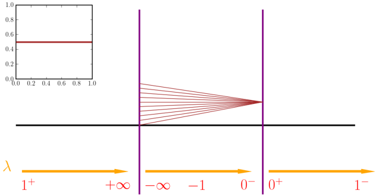


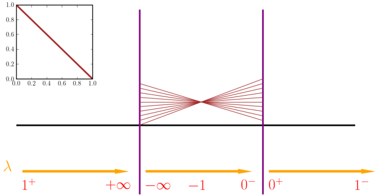
Choosing Parallel Coordinates

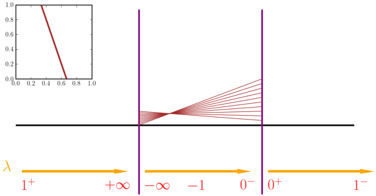


- Display as much dimensions wanted (yes, **as many**)
- Display as much data wanted (I mean it!)

Interesting patterns



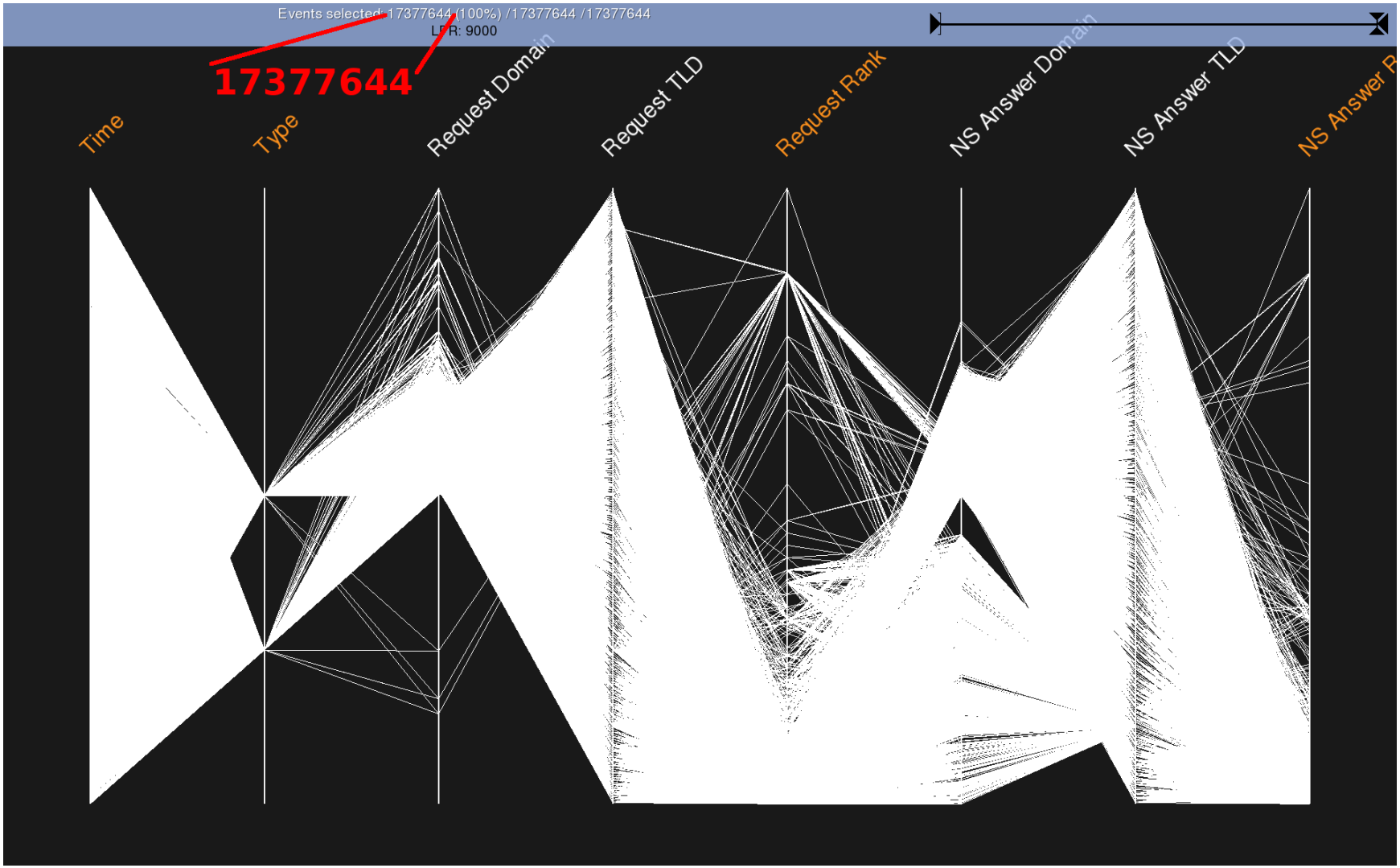




Dataset

1313716097	s	careercloset.org	0	dns1.name-services.com	1.00129821958
1313250080	s	thorstenschimmel.com	1.00059721139	ns2.webmailer.de	1.00098722699
1303867730	s	170.161.119.in-addr.arpa	0	ns1.shoukedns.com	1.0
1318350101	s	205.182.198.in-addr.arpa	0	dns2.lsus.edu	0
1318243614	s	203.131.177.122.sbl-xbl.spa	0	127.0.0.4	0
1313389794	s	snocoaa.com	1.00229779412	ns1.lunarservers.com	1.0013560557
1314793983	s	bree.hellocotton.com	1.00190723953	69.175.88.42	0
1313511298	s	allmarks.com	1.00119609198	75.125.189.194	0
1327083205	s	a1.sphotos.ak.fbcdn.net	1.00005667589	a1.sphotos.ak.fbcdn.net.e	1.00109021195
1319552814	s	230.25.151.in-addr.arpa	0	ns2.libero.it	1.00024327551

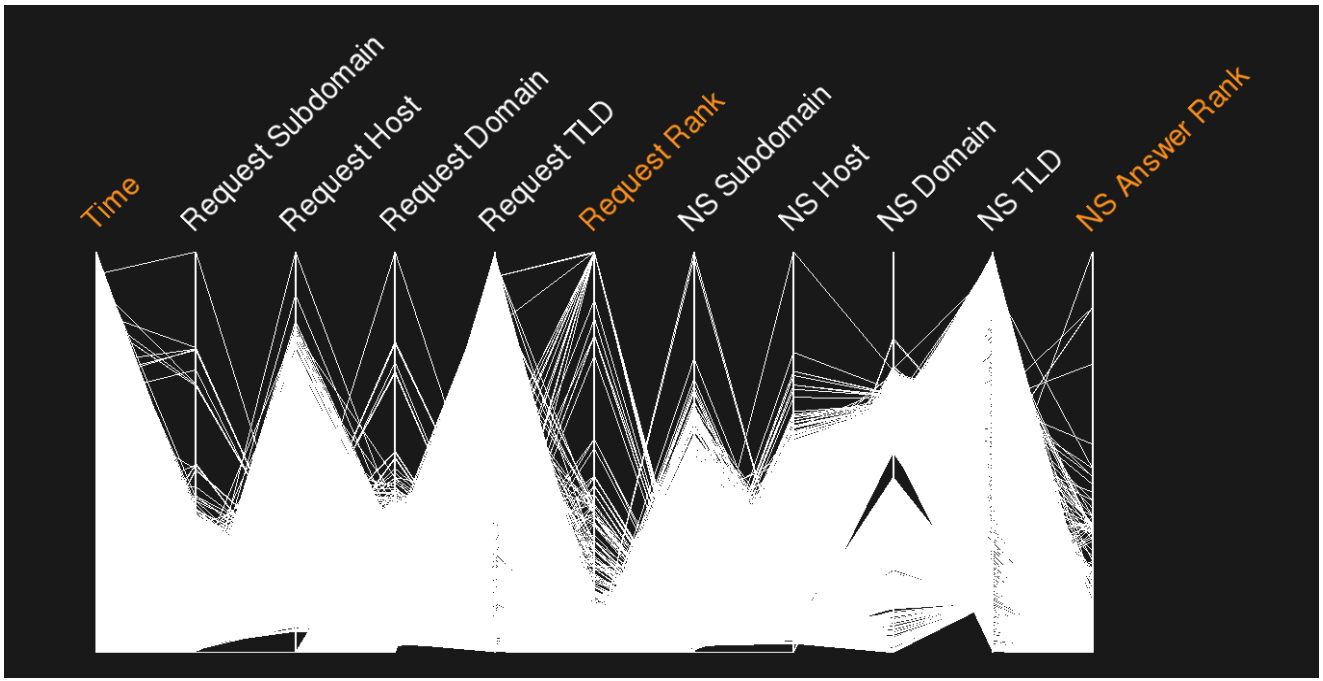
Picvizing the whole dataset



Splitting the URL

- We want to get the TLD, subdomains etc. . .
- A regex does not work: 192.168.0.1, http://localhost, google.com, www.slashdot.org:80, . . .
- We simply put them according to their ascii value
  - a is at the axis bottom
  - zzzzzzzzzzzzzzzzzzz{500} is on the very top

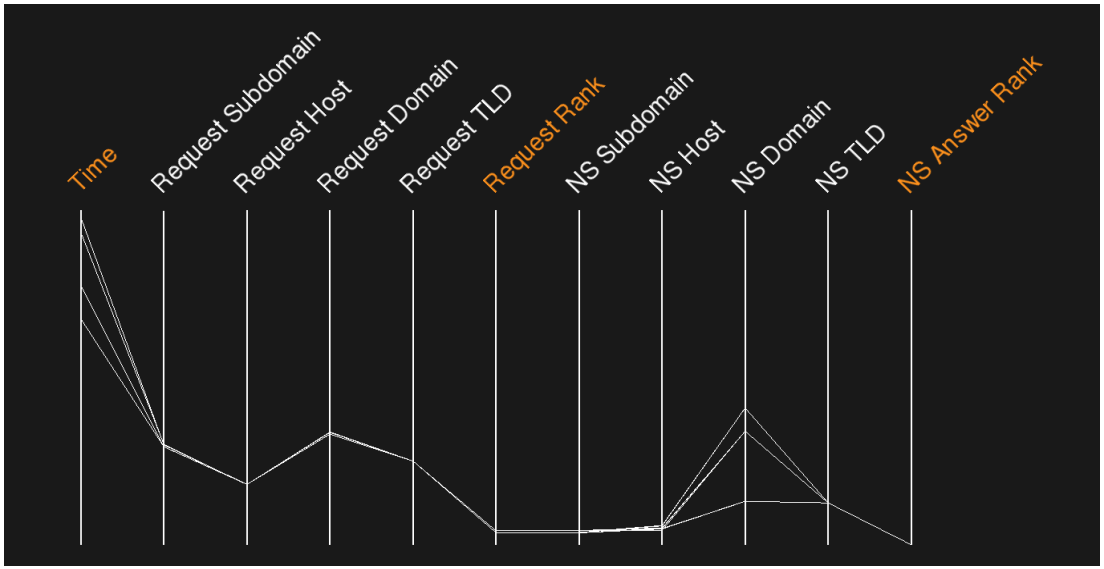
Picviz with the whole url split



Reward: highest is youtube

Subdomain entropy

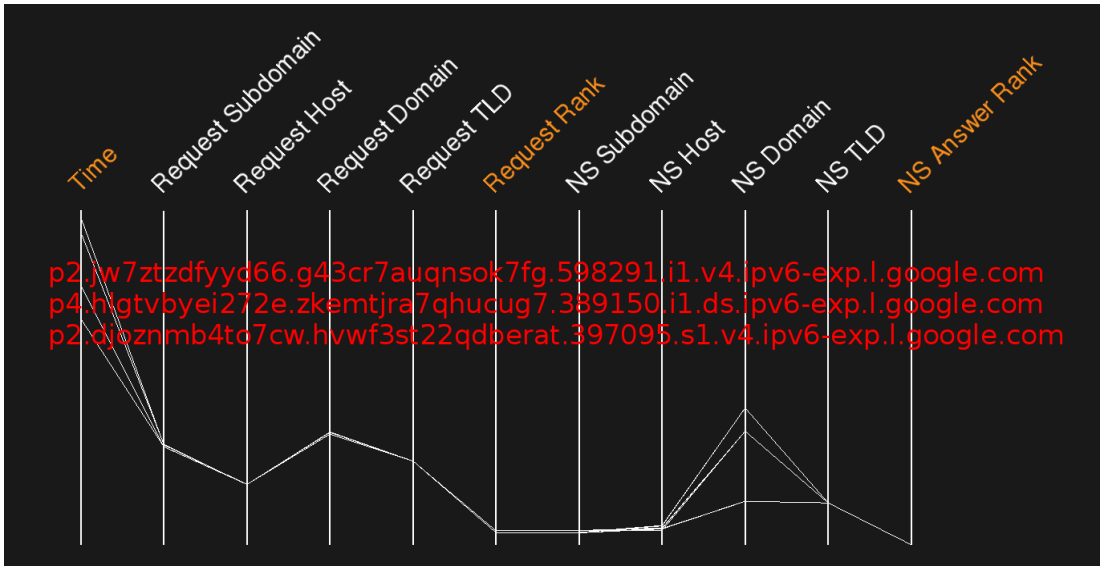
Only one sub-domain has an entropy[3] >4.8



[3] Shannon entropy

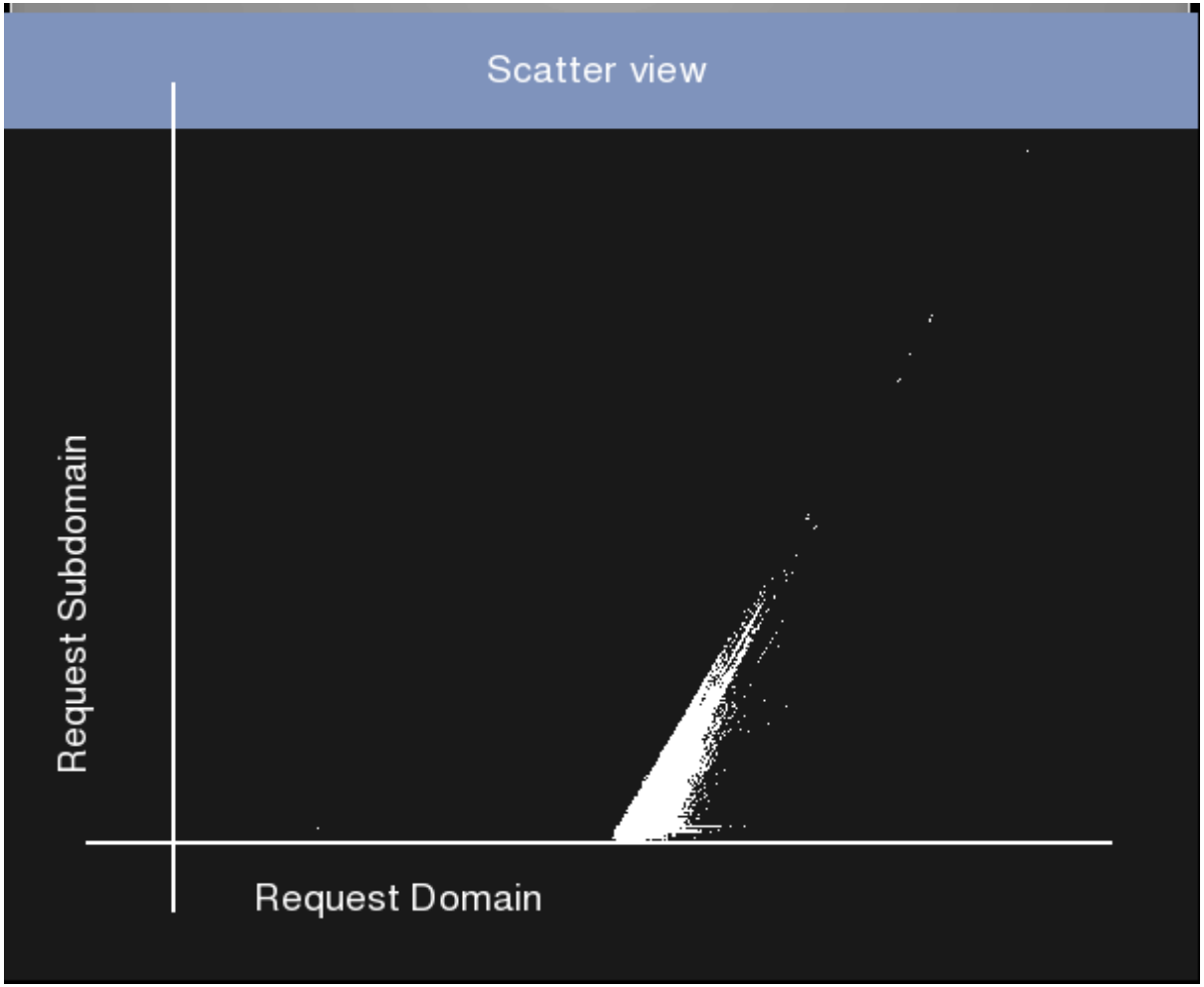
Subdomain entropy

Only one sub-domain has an entropy[4] >4.8

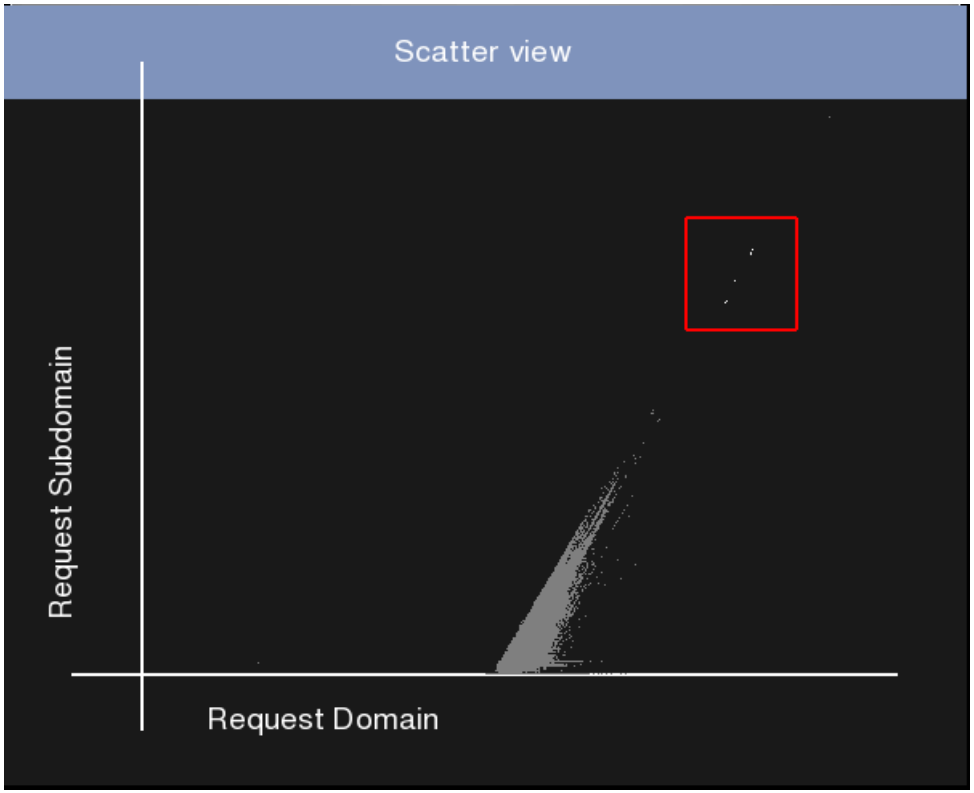


[4] Shannon entropy

Scatter plot - finding outliers



Scatter plot - finding outliers - covert channel?

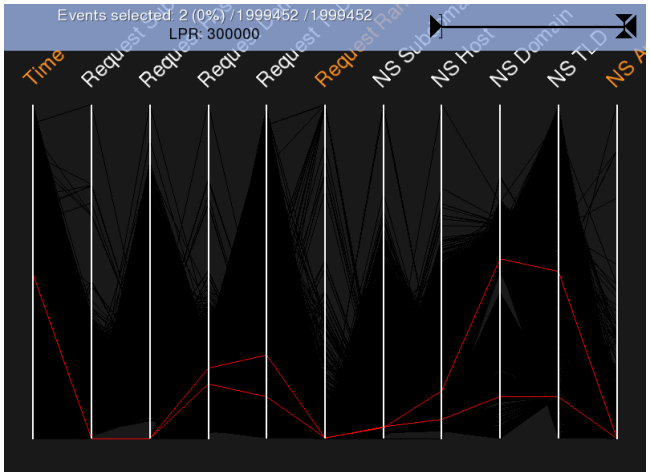


03006636366364393730653l[..].3639376431333365376 3.lbl8.mailshell.net
t10000.u1318235395163.s203679668[..]-1329.zv6lit-null.zrdtd-1311.zr6td-null.results.potaroo.net
030030643038316639653 86[..].643065613438373465 33.lbl8.mailshell.net

Searching for Zeus

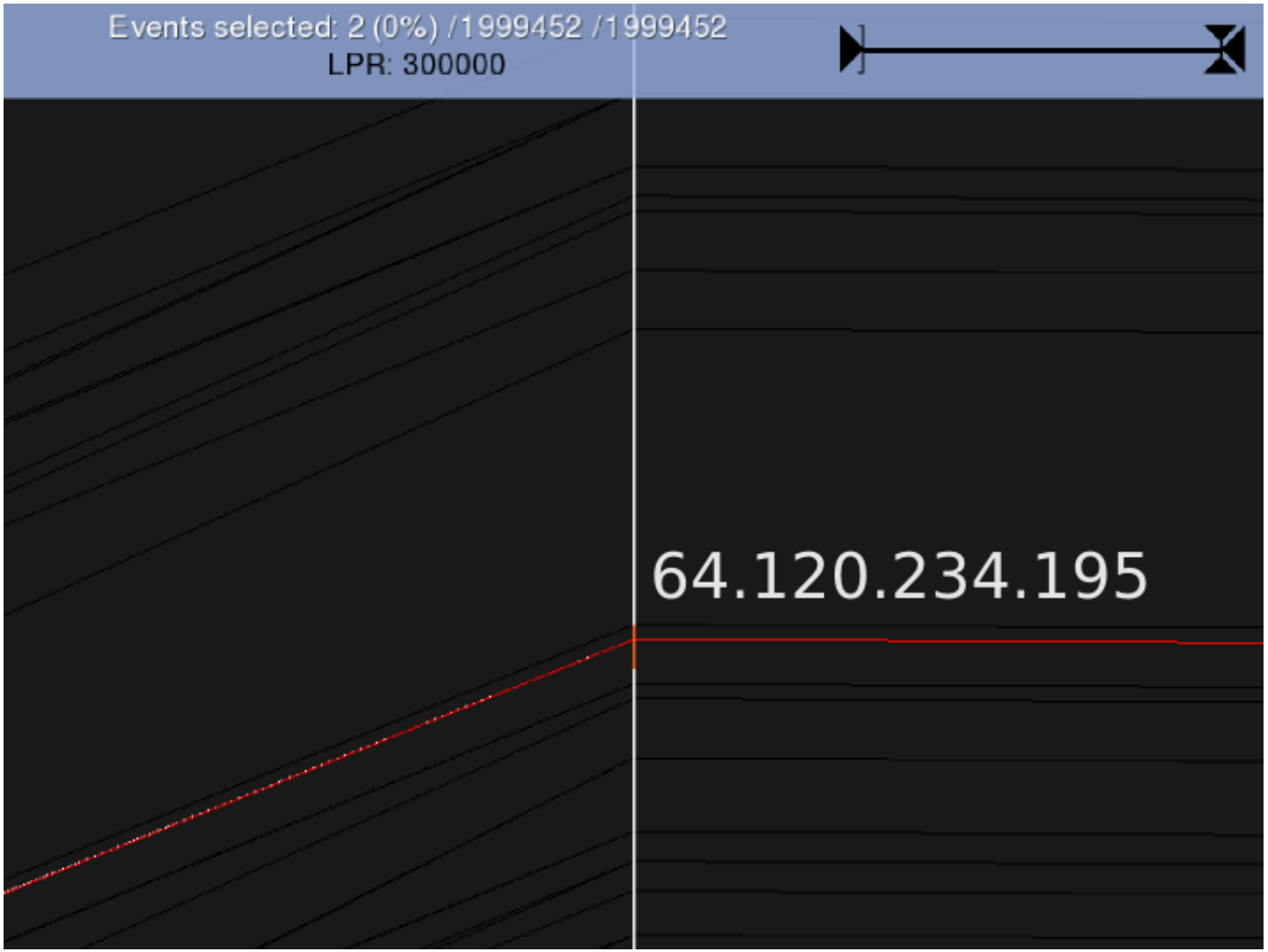
Using the broad Polish CERT regex

`[a-z0-9]{32,48}\.(ru|com|biz|info|org|net)`

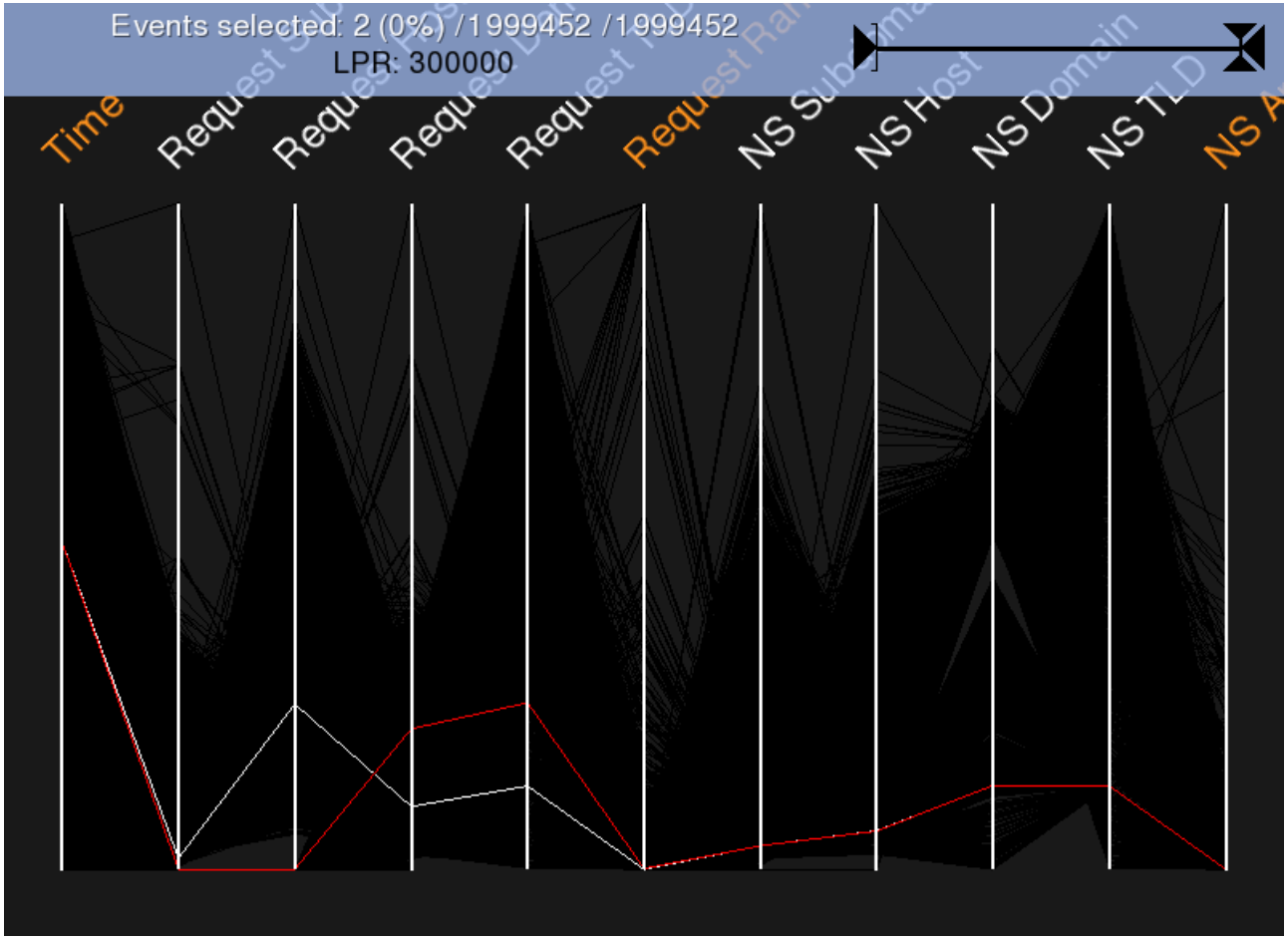


- We get some cool domains:
  - cg79wo20kl92doowfn01oqpo9mdieowv5tyj.com
  - eef795a4eddaf1e7bd79212acc9dde16.net
- but more important we got a visualization profile to find outliers not matching the regexp

Zoom on NS answer domain



Back to the global view



- request domain: ns2.speed-tube.net

Investigating ns2.speed-tube.net

- Grab cool stuff that are not ranked like:
  adsforadsense.co.cc;1.0;ns2.speed-tube.net;1.0
  extra-tube.net;1.0001125221;ns2.speed-tube.net;1.0 ...
- A recurring (reactivated or cached) malicious site:
  adsforadsense.co.cc rogue safebrowsing.clients.google.com
  20110315 20110125

Conclusion

- Passive DNS is an infinite source of security data mining
- The toolkit is now available on github and this is the basis for more research
- (adequate) Visualization is an appropriate way to discover unknown malicious or suspicious services
- This finally helps CSIRTs to act earlier on the incidents

Free Software

- BGP Ranking software
  `https://www.github.com/CIRCL/BGP-Ranking` - `http://bgpranking.circl.lu/`
- Passive DNS toolkit
  `https://www.github.com/adulau/pdns-viz/` - first commit for CanSecWest - more modules to come
- Domain Classification
  `https://www.github.com/adulau/DomainClassifier/`

Q&A

- @adulau - alexandre.dulaunoy@circl.lu
- @tricaud - sebastien@honeynet.org