URL Analysis

At Scale

Josh Pyorre, November 16, 2023

Josh Pyorre

Security Researcher

CISCO TALOS

Senior Security Researcher

Previously:

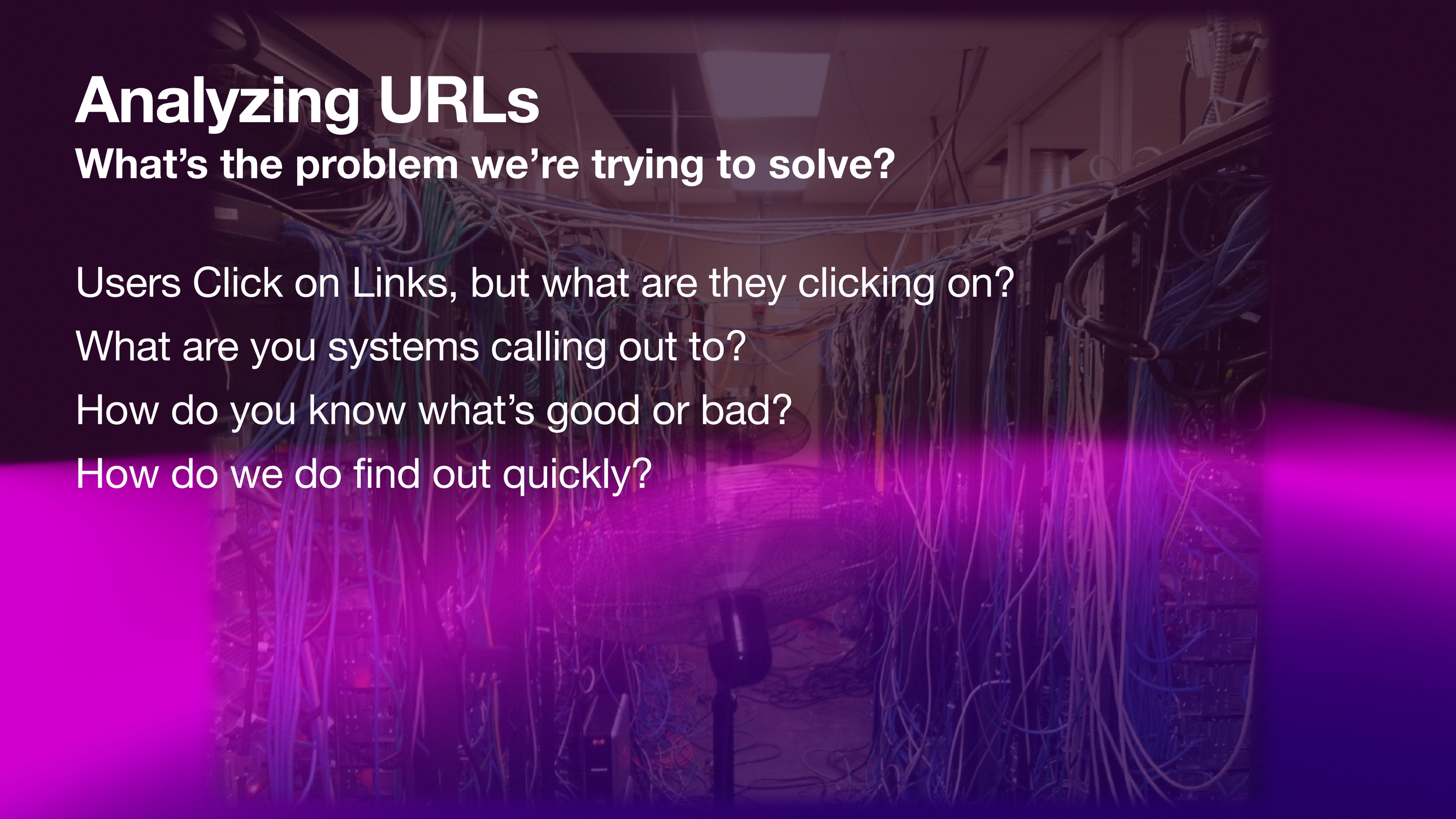
zscaler

MANDIANT

NASA

Agenda

- The Problem
- URL Analysis Techniques
- Optimization Methods
- The Interface and Code



Analyzing URLs

What's the problem we're trying to solve?

Users Click on Links, but what are they clicking on?

What are you systems calling out to?

How do you know what's good or bad?

How do we do find out quickly?

Analyzing URLs

Lots of (*expensive*) options

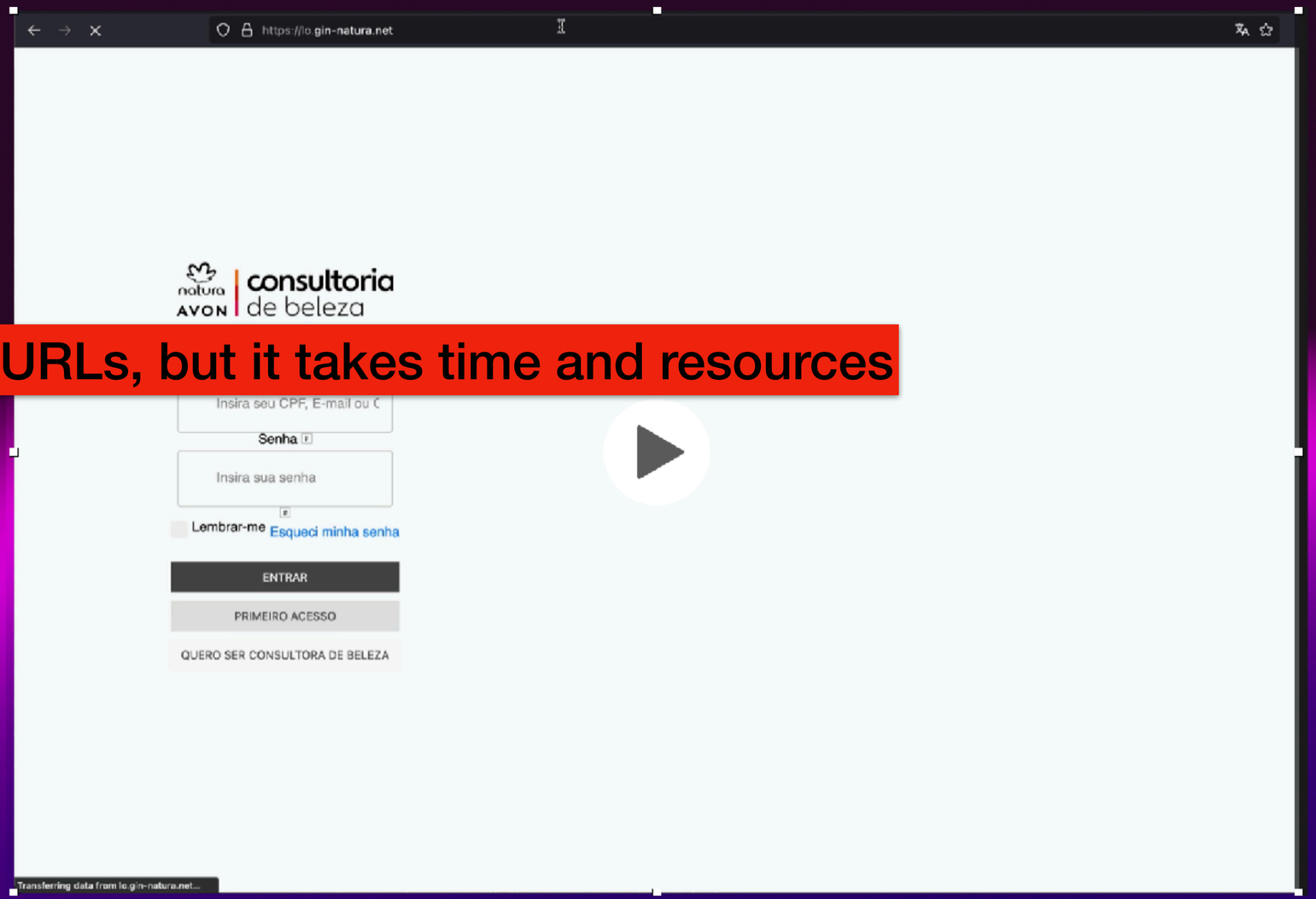
Crawl URLs

Third Party APIs

Vendor Feeds

Endpoint Detection

Log Analysis



You can crawl URLs, but it takes time and resources

Analyzing URLs

Lots of (*expensive*) options

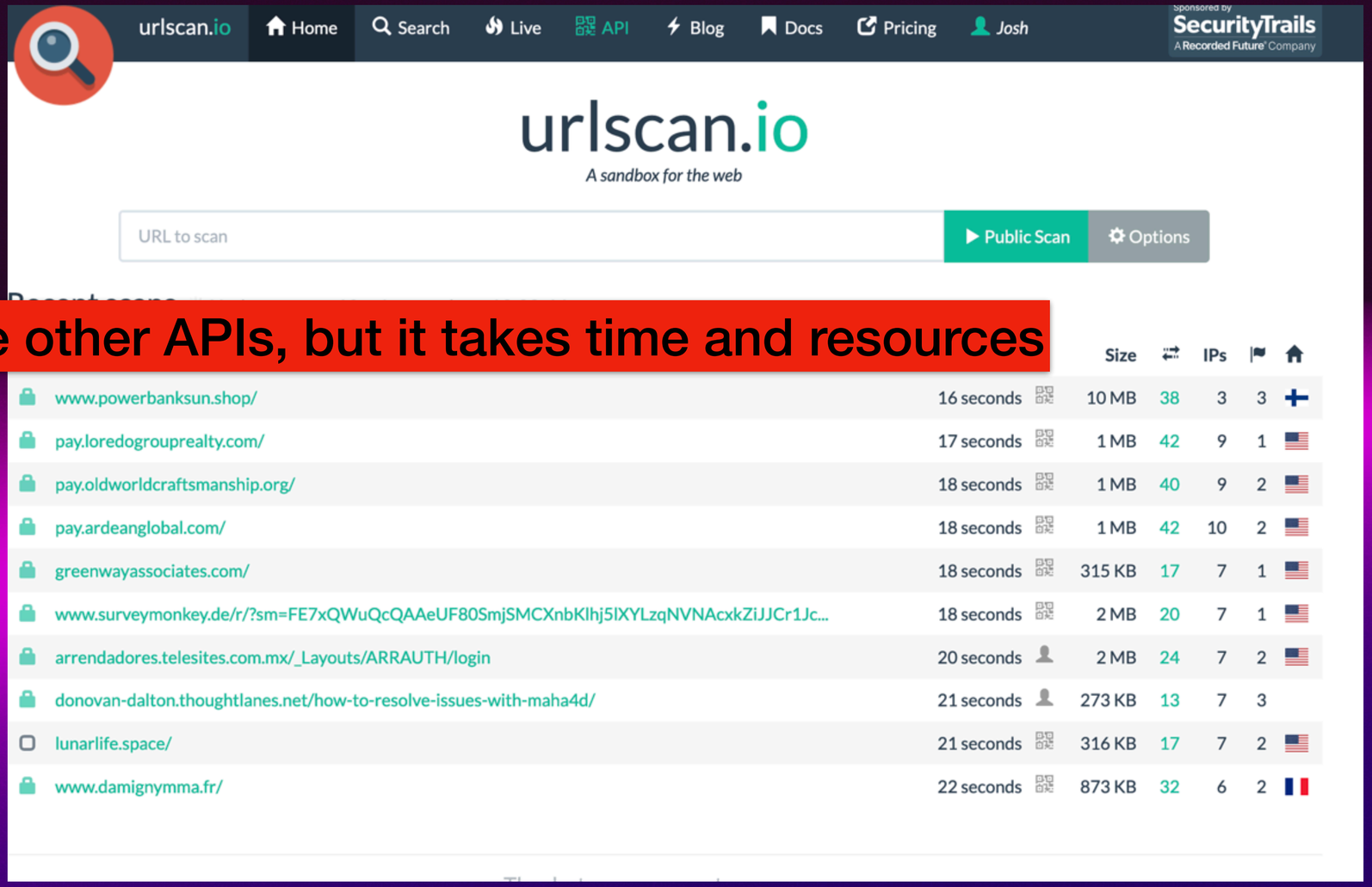
Third Party APIs

Vendor Feeds

Endpoint Detection

Log Analysis

You can use other APIs, but it takes time and resources



Analyzing URLs

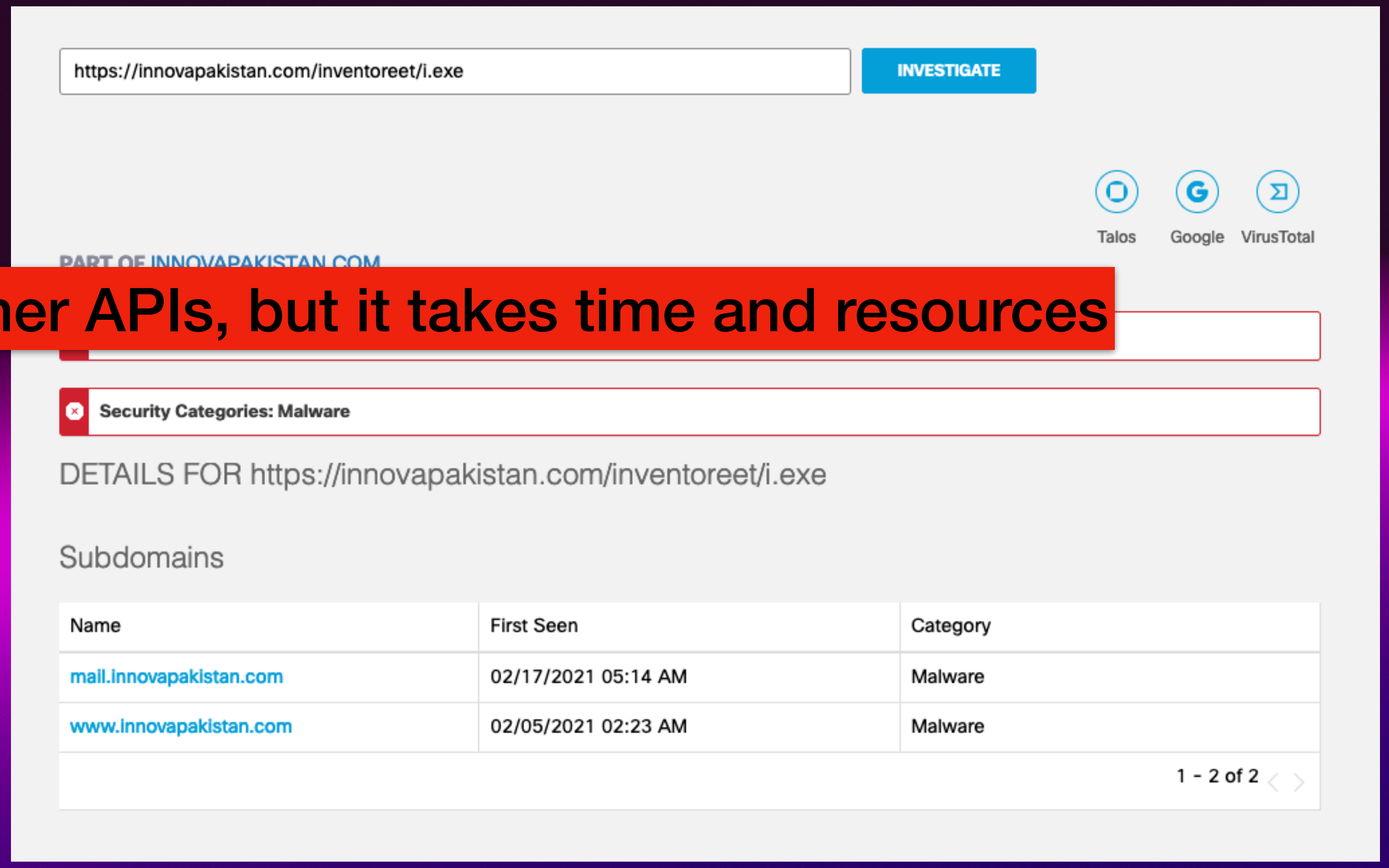
Lots of (*expensive*) options

Third Party APIs

Vendor Feeds

Endpoint Detection

Log Analysis



You can use other APIs, but it takes time and resources

Analyzing URLs

Lots of (*expensive*) options

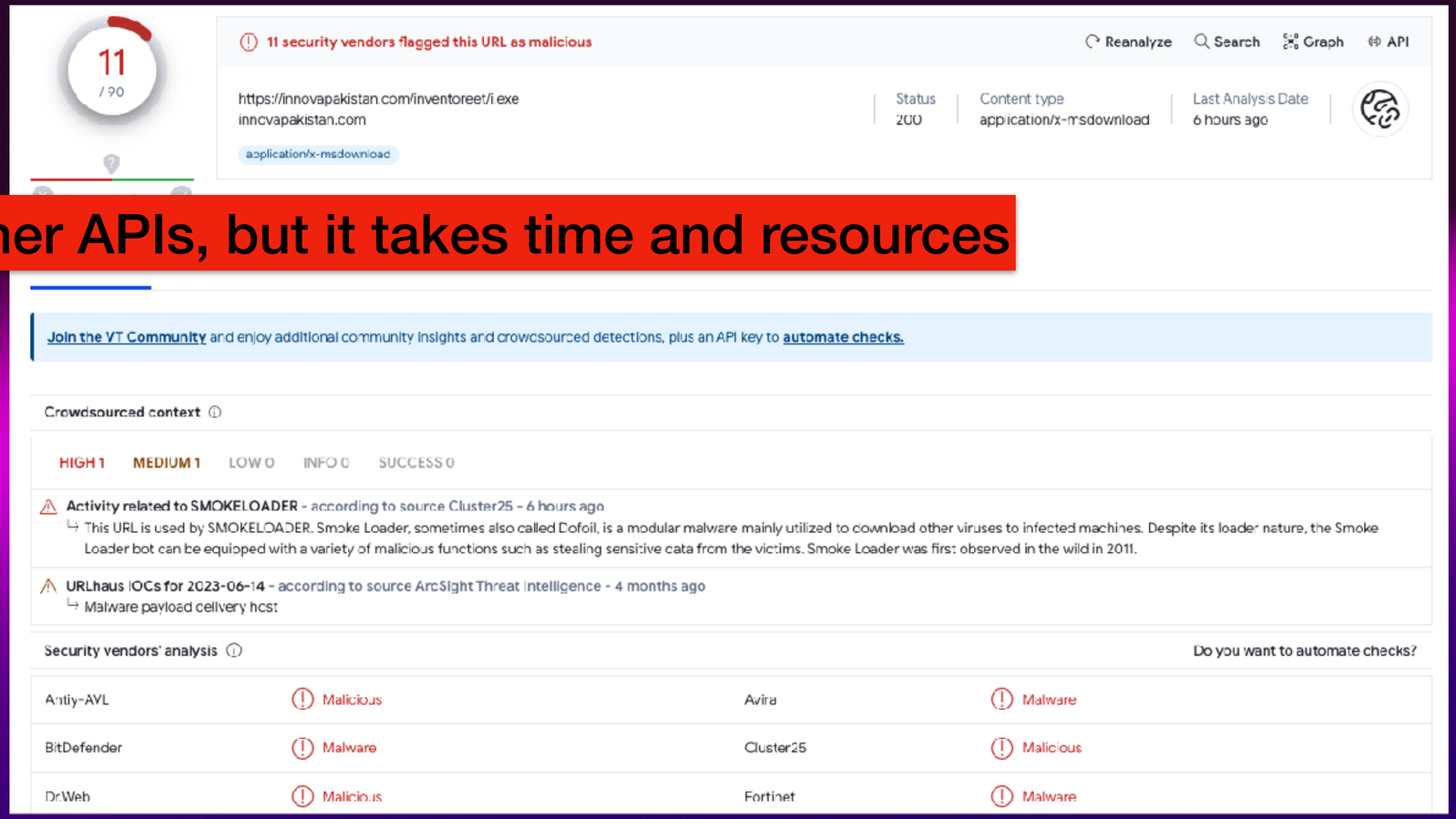
Third Party APIs

Vendor Feeds

Endpoint Detection

Log Analysis

You can use other APIs, but it takes time and resources



Analyzing URLs

Lots of (*expensive*) options

Vendor Feeds

Endpoint Detection

Log Analysis

You can use feeds from vendors, but it takes resources

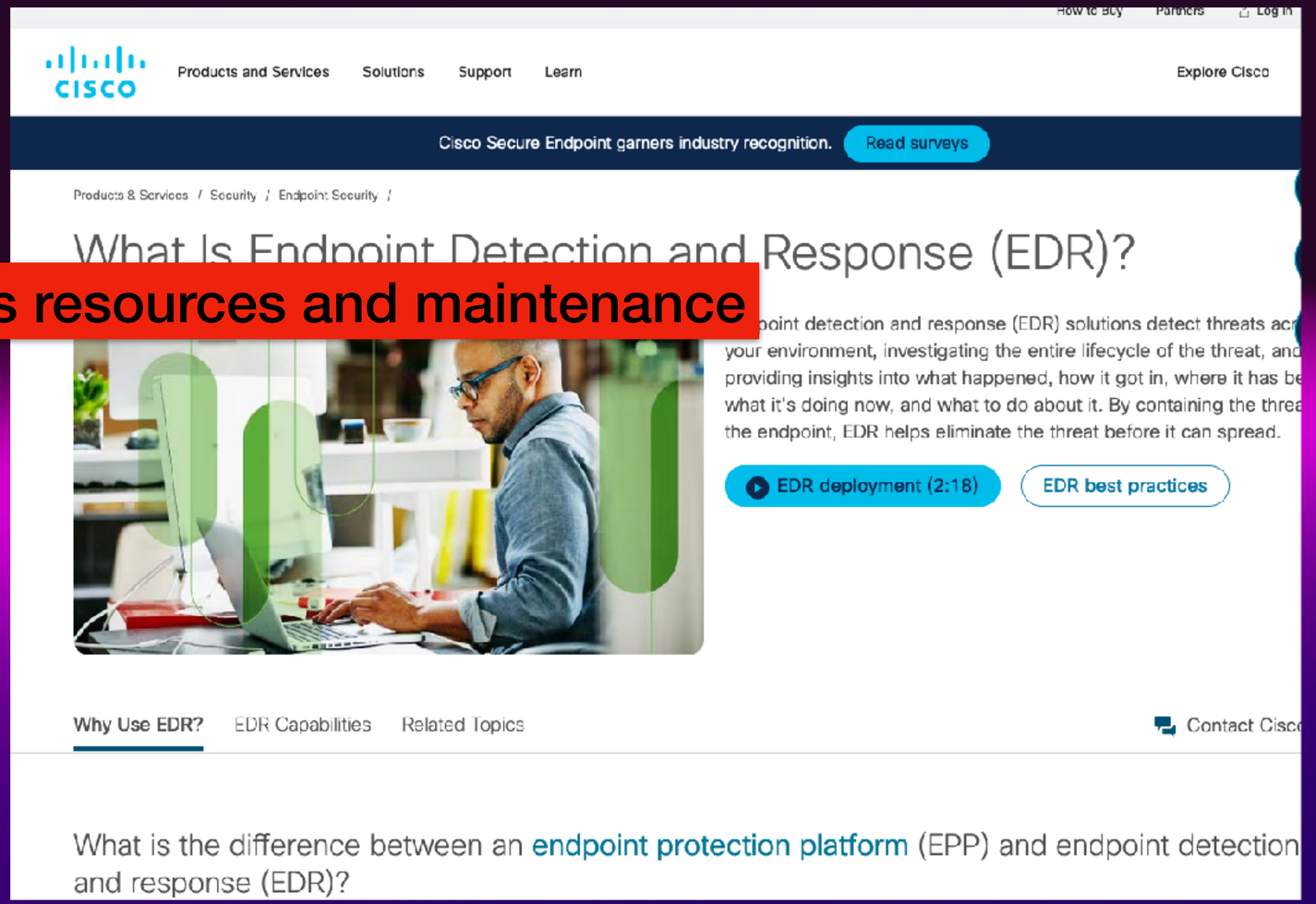
1. **AlienVault Open Threat Exchange** This is the original crowd-sourced threat intelligence collection, and it is probably still the best, processing more than 19 million new IoC records every day. The service is free to use and can deliver threat intelligence in various formats, including STIX, OpenIoC, MAEC, JSON, and CSV formats. Each feed instance is called a "pulse." You can define your requirements, getting specific pre-filtered data, and there is also an opportunity to get tailored feeds per device type, such as endpoints. If related data lies outside of the parameters of your feed, that extra data will be linked to within the records that are delivered to you.
2. **FBI InfraGard** A threat intelligence feed from the FBI carries a lot of authority, and it is free to access. Feeds are categorized by industry according to the definition of the Cybersecurity and Infrastructure Security Agency. So, this is a filtered list of IoCs according to the activity sector. Joining the service also enroll you in a local chapter, which is an excellent opportunity to network with other local business leaders.
gence service as
trike Falcon Intelligence is available in three plan levels. This service mainly aims to enhance the performance of the media XDR and SIEM systems. However, they can also be linked to third-party security tools. Falcon Intelligence provides human-readable reports plus automated feeds sent straight to security services.
4. **Anomali ThreatStream** This aggregator service consolidates threat intelligence feeds from multiple sources down to one. The service uses AI to filter out false positives and irrelevant warnings. It handles TTP data and IoCs, and it will produce an automated feed for your security software and a human-readable report. The tool can be run on-premises as a virtual machine or accessed as a SaaS. The package will also upload reports from your system to threat databases and circulate activity warnings from each of your network devices to all of the others.
5. **Mandiant Threat Intelligence** This threat Intelligence service is highly respected and offers regular feeds in various formats, including reports for analysts and inputs for software. Information covers both IoCs and TTPs. There is a free version of this service.

Analyzing URLs

Lots of (*expensive*) options

Endpoint Detection

Log Analysis

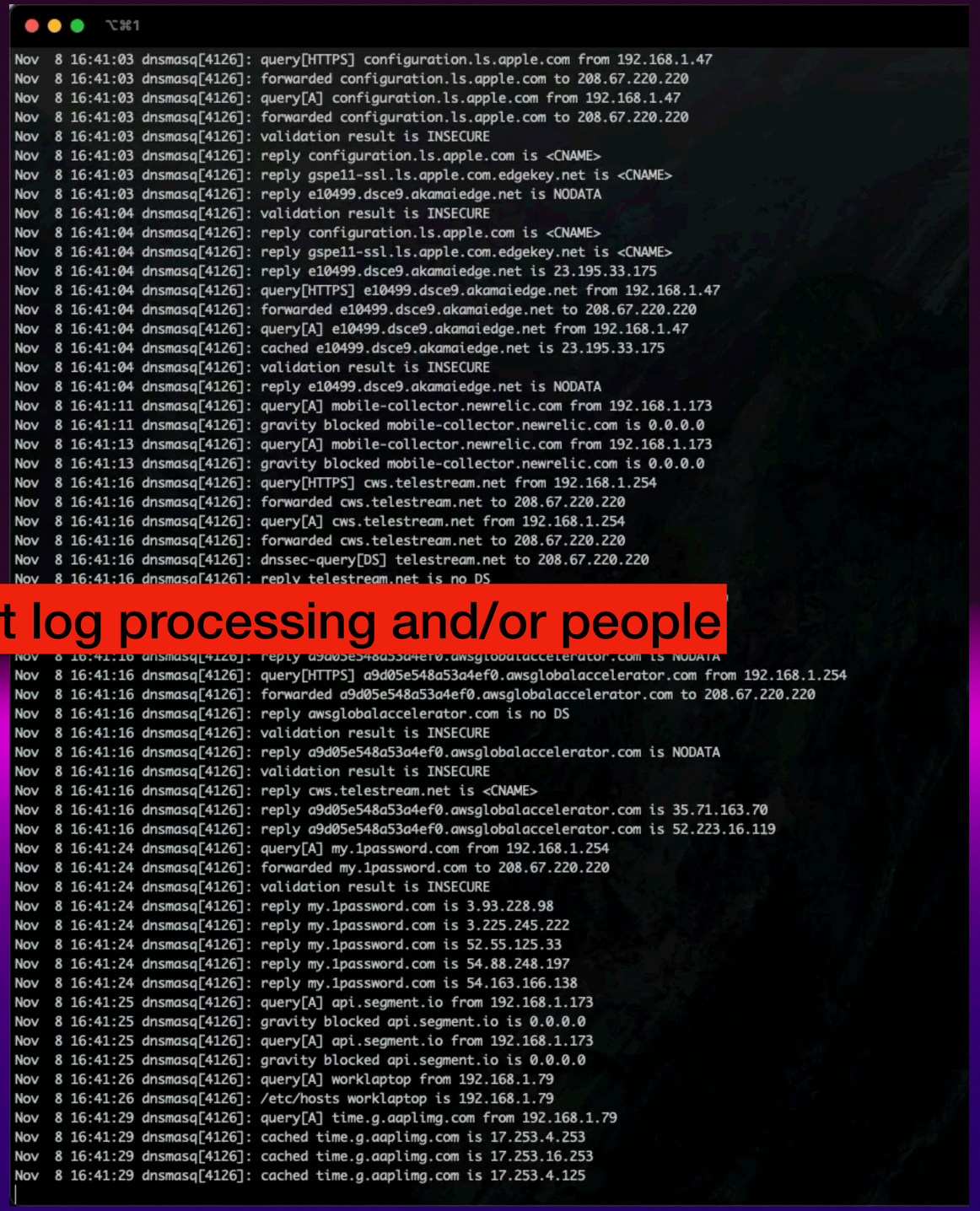


Analyzing URLs

Lots of (*expensive*) options

Log Analysis

Log Analysis requires smart log processing and/or people



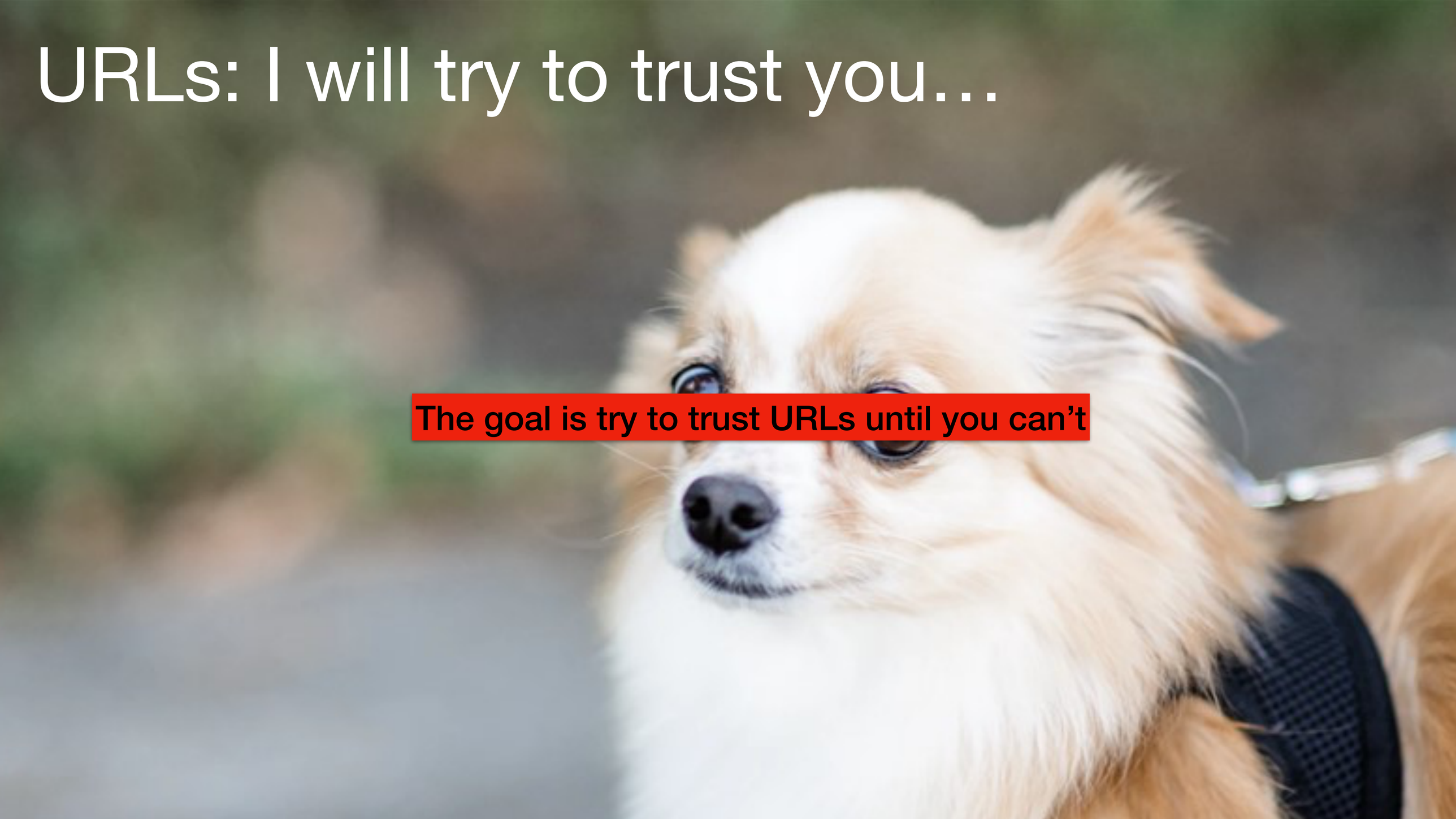
Collecting URLs

From your environment

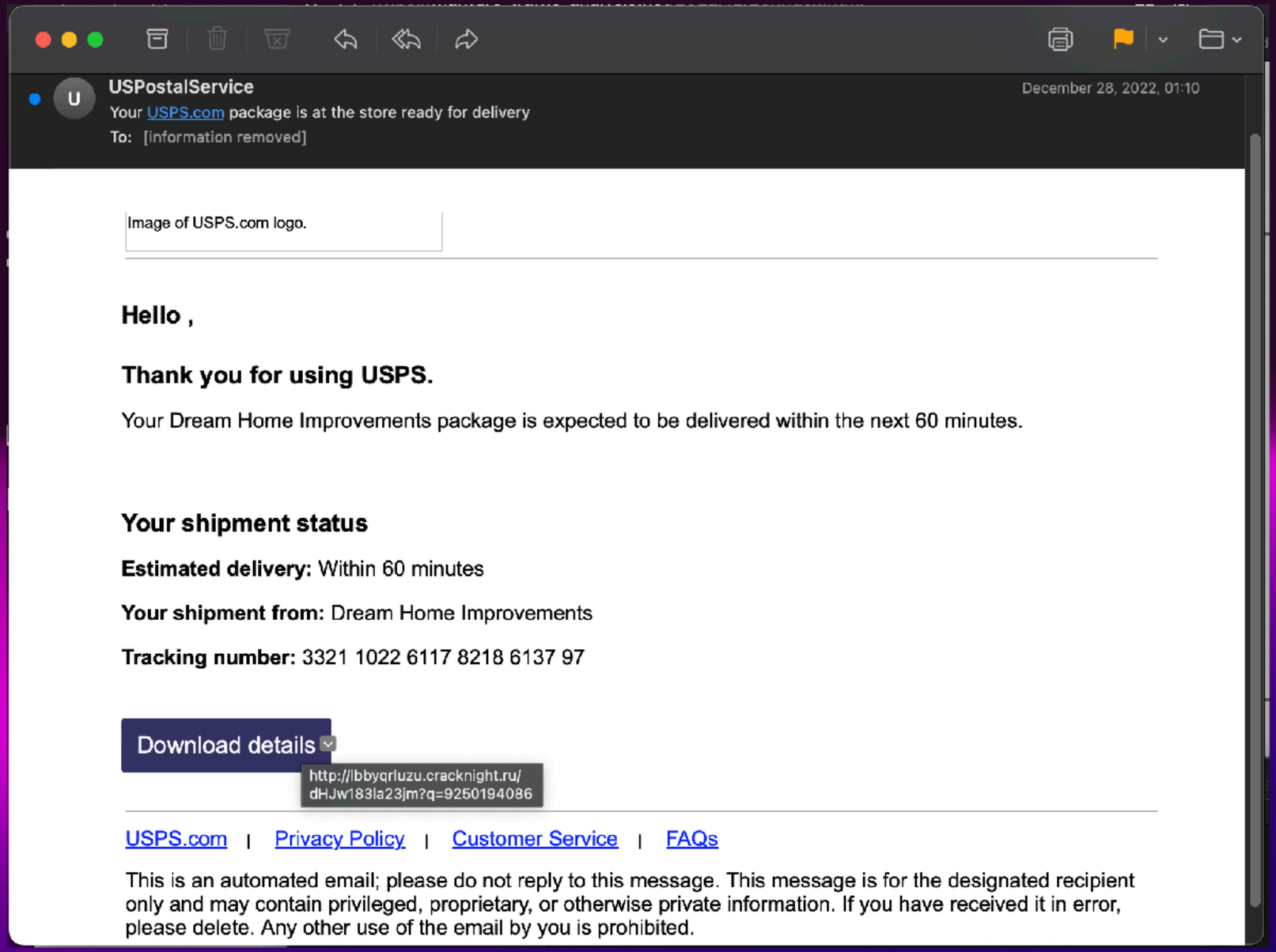
- Logs
- Checking if it was clicked (DNS responses)

Getting data to work with:
From proxy or DNS logs

URLs: I will try to trust you…



What is a bad URL?



U

USPostalService

December 28, 2022, 01:10

Your USPS.com package is at the store ready for delivery

To: [information removed]

Image of USPS.com logo.

Hello ,

Thank you for using USPS.

Your Dream Home Improvements package is expected to be delivered within the next 60 minutes.

Your shipment status

Estimated delivery: Within 60 minutes

Your shipment from: Dream Home Improvements

Tracking number: 3321 1022 6117 8218 6137 97

Download details

http://lbbyqrluzu.cracknight.ru/dHJw183la23jm?q=9250194086

USPS.com | Privacy Policy | Customer Service | FAQs

This is an automated email; please do not reply to this message. This message is for the designated recipient only and may contain privileged, proprietary, or otherwise private information. If you have received it in error, please delete. Any other use of the email by you is prohibited.

acking number: 3321 1022 6117 8218 6137 97

Download details

URL in email doesn't look right…

http://lbbyqrluzu.cracknight.ru/
dHJw183la23jm?q=9250194086

SPS.com | Privacy Policy | Customer Service | FA

his is an automated email; please do not reply to this messa
nly and may contain privileged, proprietary, or otherwise priv
ease delete. Any other use of the email by you is prohibited

15
/ 90

Community Score

15 security vendors flagged this URL as malicious

Reanalyze | Search | Graph | API

http://lbbyqrluzu.cracknight.ru/dHJw183la23jm?q=9250194086
lbbyqrluzu.cracknight.ru

text/html

Status
200

Last Analysis Date
8 months ago

DETECTION | DETAILS | COMMUNITY

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

It's seen as 'bad' from VirusTotal

Security vendors' analysis

Do you want to automate checks?

alphaMountain.ai	Malicious	Antiy-AVL	Malicious
Avira	Malware	BitDefender	Malware
CRDF	Malicious	CyRadar	Malicious
ESET	Phishing	ESTsecurity	Malicious
Forcepoint ThreatSeeker	Malicious	Fortinet	Malware
G-Data	Malware	Lionic	Malicious
Seclookup	Malicious	Sophos	Malware

Ok, I can do this. It’s easy to tell if a URL is bad!

I can do this

Human instinct based on experience suggest this URL is not good

http://lbbyqrluzu.cracknight.ru/dHJw183la23jm?q=9250194086

Human instinct based on experience suggest this URL is not good

lbbyqrluzu cracknight ru/dHJw183la23jm?q=9250194086

http

Not using SSL

Human instinct based on experience suggest this URL is not good

cracknight ru/dHJw183la23jm?q=9250194086

http lbbyqrluzu

Weird spelling…

Human instinct based on experience suggest this URL is not good

/dHJw183la23jm?q=9250194086

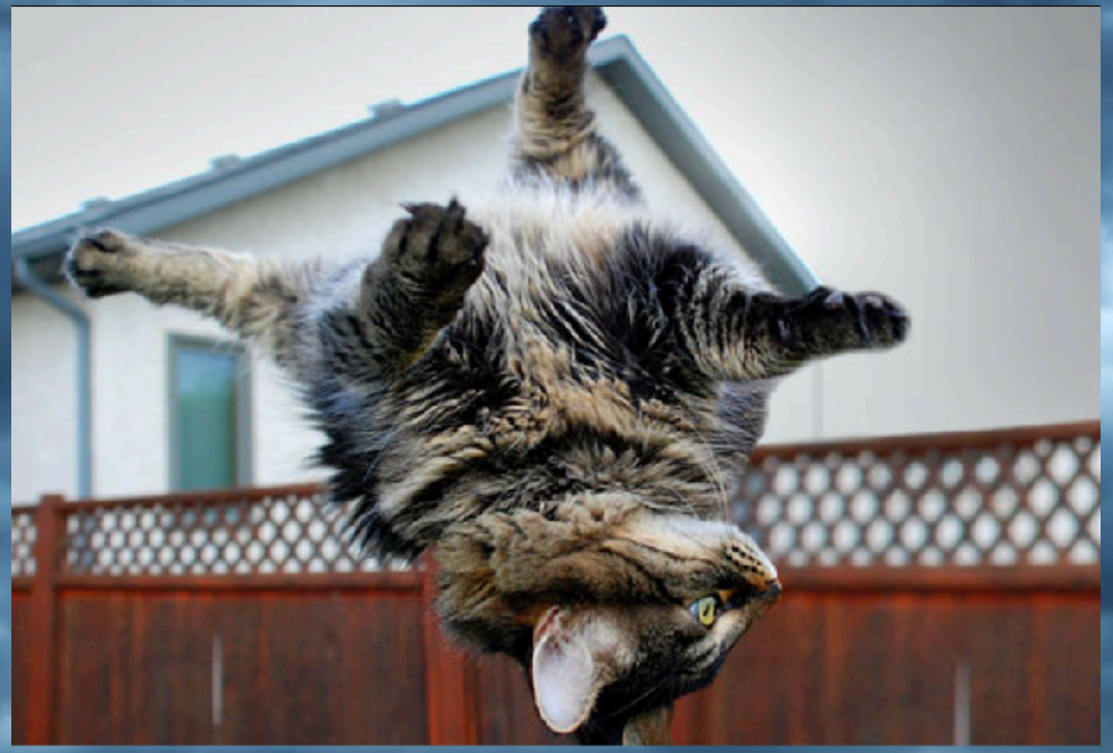
http lbbyqrluzu cracknight ru

Human instinct based on experience suggest this URL is not good

http lbbyqrluzu cracknight ru dHJw183la23jm?q=9250194086

I need help - one person or team can’t keep up when looking at millions of URLs

I can’t do this



URL Analysis Techniques

Various rule-based detection methods (some of which are employed here)

Features

Identifying a malicious URL using Rules

- Spelling
- Incorrect Infrastructure
- Brand Components in URLs
- Is unusually long
- Is just an IP address
- Popularity is low
- Is new
- Has a low TTL
- Low Pagerank (https://en.wikipedia.org/wiki/PageRank)

Clean the URLs first

URL Analysis Techniques

Getting Everything Ready

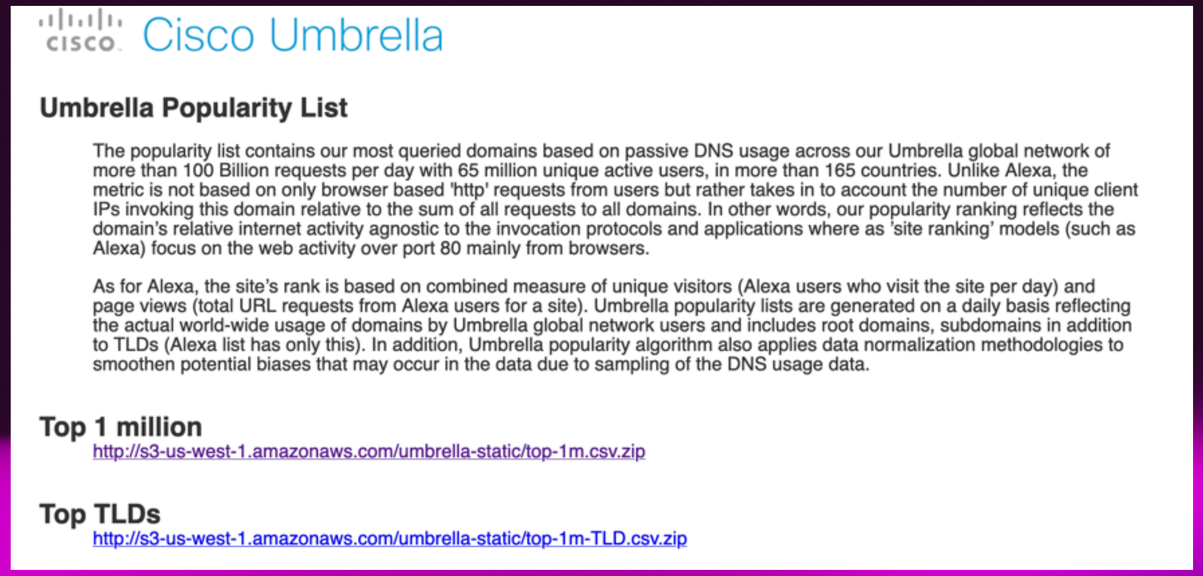
- Remove Popular Domains
- Separate redirects, deobfuscate
- Remove Protocol, www, convert to lowercase
- Tokenize and Collect Words

Remove Popular

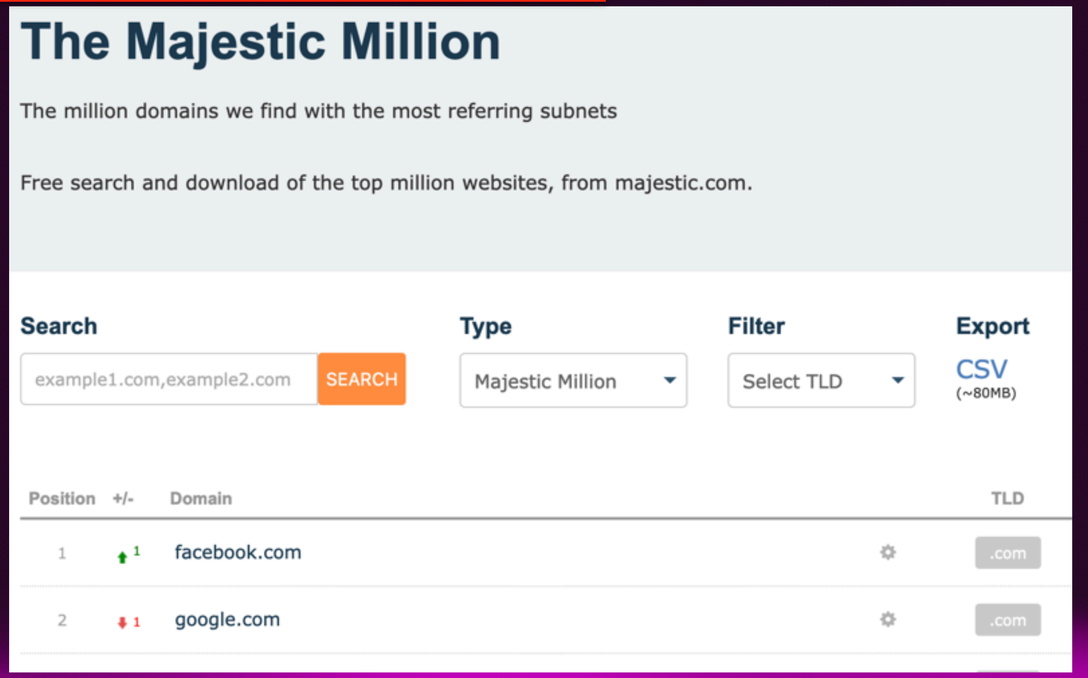
There are many options for finding popular domains

Remove Popular

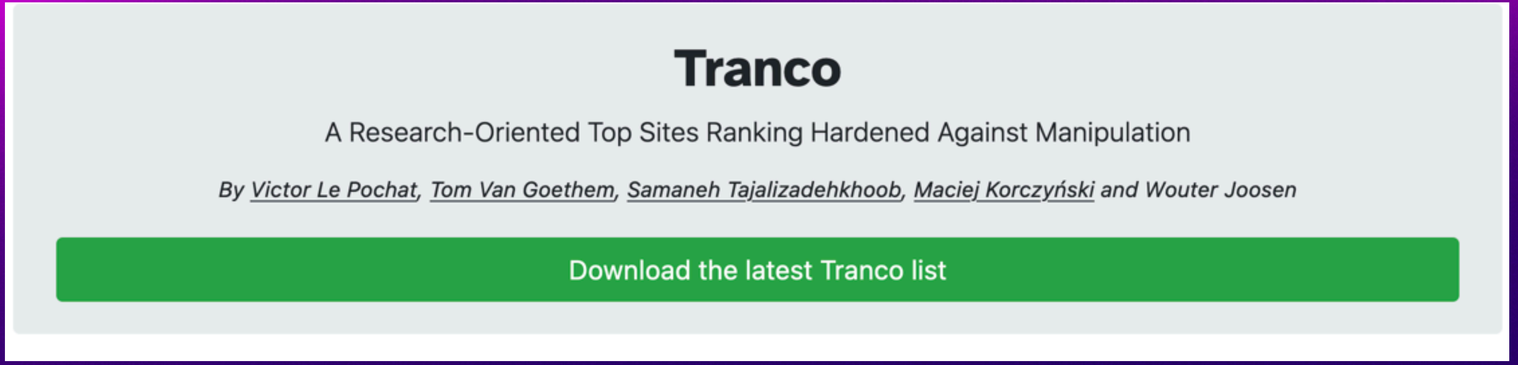
Removing URLs at Popular Domains



http://s3-us-west-1.amazonaws.com/umbrella-static/index.html

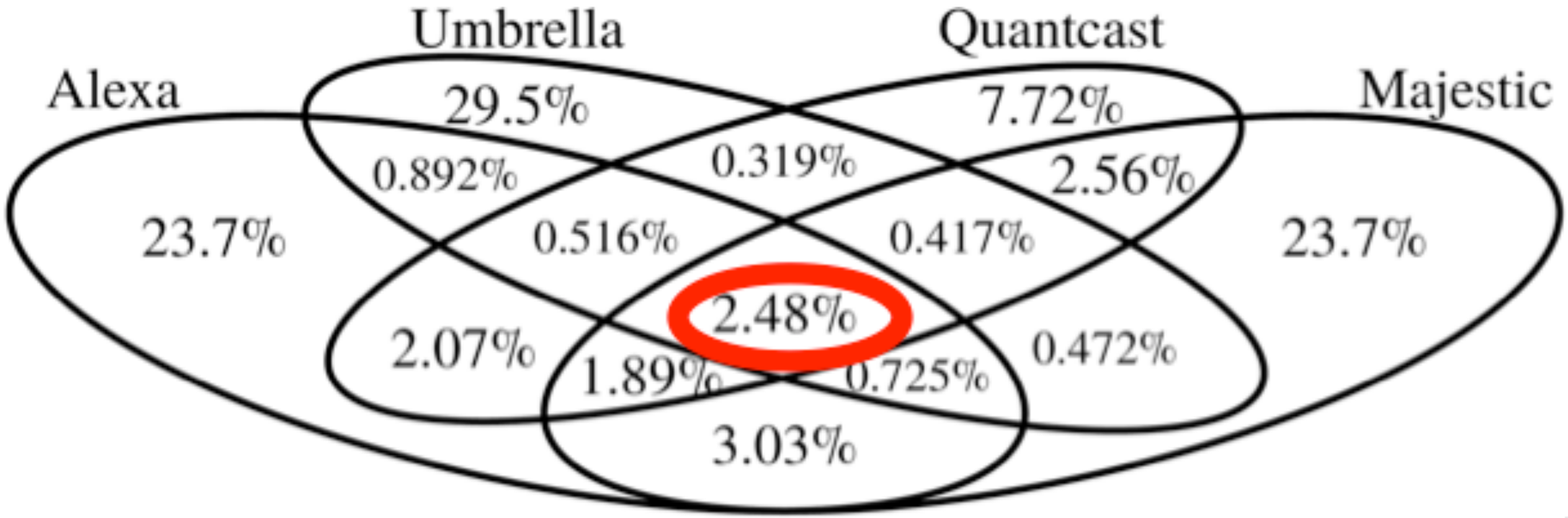


https://majestic.com/reports/majestic-million



https://tranco-list.eu/

They all seem to be about the same, so pick whatever suits you



The daily average intersection between the four lists from January 2018 to November 2019

https://labs.ripe.net/author/samaneh_tajalizadehkhoob_1/the-tale-of-website-popularity-rankings-an-extensive-analysis/

Separate Redirects
&
Deobfuscate

Some URLs redirect to other URLs. Security products do this to make the URL safe.

Unquote & Separate Redirections

Security Products on top of Security Products

https://hes32-ctp.trendmicro.com/wis/clicktime/v1/query?
url=https%3a%2f%2fnam10.safelinks.protection.outlook.com%2f%3furl%3dhttps%253A%252F%252F
drive.google.com%252Ffile

Separate the URLs and remove URL Encoding

Unquote & Separate Redirections

Security Products on top of Security Products

1: https://hes32-ctp.trendmicro.com/wis/clicktime/v1/queryurl=

2: https%3a%2f%2fnam10.safelinks.protection.outlook.com%2f%3furl%3d

3: https%253A%252F%252Fdrive.google.com%252Ffile

End up with the URL you want to analyze

Unquote & Separate Redirections

Security Products on top of Security Products

~~1: https://hes32-ctp.trendmicro.com/wis/clicktime/v1/queryurl=~~

~~2: https://nam10.safelinks.protection.outlook.com/?url=~~

3: https://drive.google.com/file

Unquote & Separate Redirections

Actual redirections

http://elinux-software-news-tutorialsts.adjust.com/izw3imq?
redirect=https%3A%2F%2Fdentalfunzoneelpaso.com

Some URLs redirect as a feature, and some as a misconfiguration (openredirects)
In this case, check the redirecting AND redirected URL. The redirecting URL might be compromised and the redirected might be malicious.

Unquote & Separate Redirections

Actual redirections

http://elinux-software-news-tutorialsts.adjust.com/izw3imq?redirect=https://dentalfunzoneelpaso.com

Some URLs redirect as a feature, and some as a misconfiguration (openredirects)
In this case, check the redirecting AND redirected URL. The redirecting URL might be compromised and the redirected might be malicious.

Collect Words

Take all the words out of the URL and add to a list for analysis

Tokenize & Collect Words

Split URLs, create dictionaries, set score

```
{
  ‘url': ‘benkofmaerical.com/benking',
  ‘tokens': [‘benkofmaerical', ‘com', ‘benking’],
  ‘score’: 0,
}
```

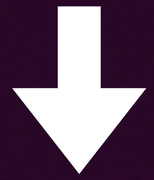
Here, we’re also creating a dictionary to start saving data. We’re also setting a score at 0, which might change if the URL is bad

Exact Searches

Exact Searches

Build a wordlist from known-bad URLs

sikayetvar.com/finans/bankacilik/hsbc-bank/internet-bankaciligi



['sikayetvar', 'finans', 'bankacilik', 'hsbc-bank', 'internet-bankaciligi']

Finding Exact Matches

Using those words

['sikayetvar', 'finans', 'bankacilik', 'hsbc-bank', 'internet-bankaciligi']

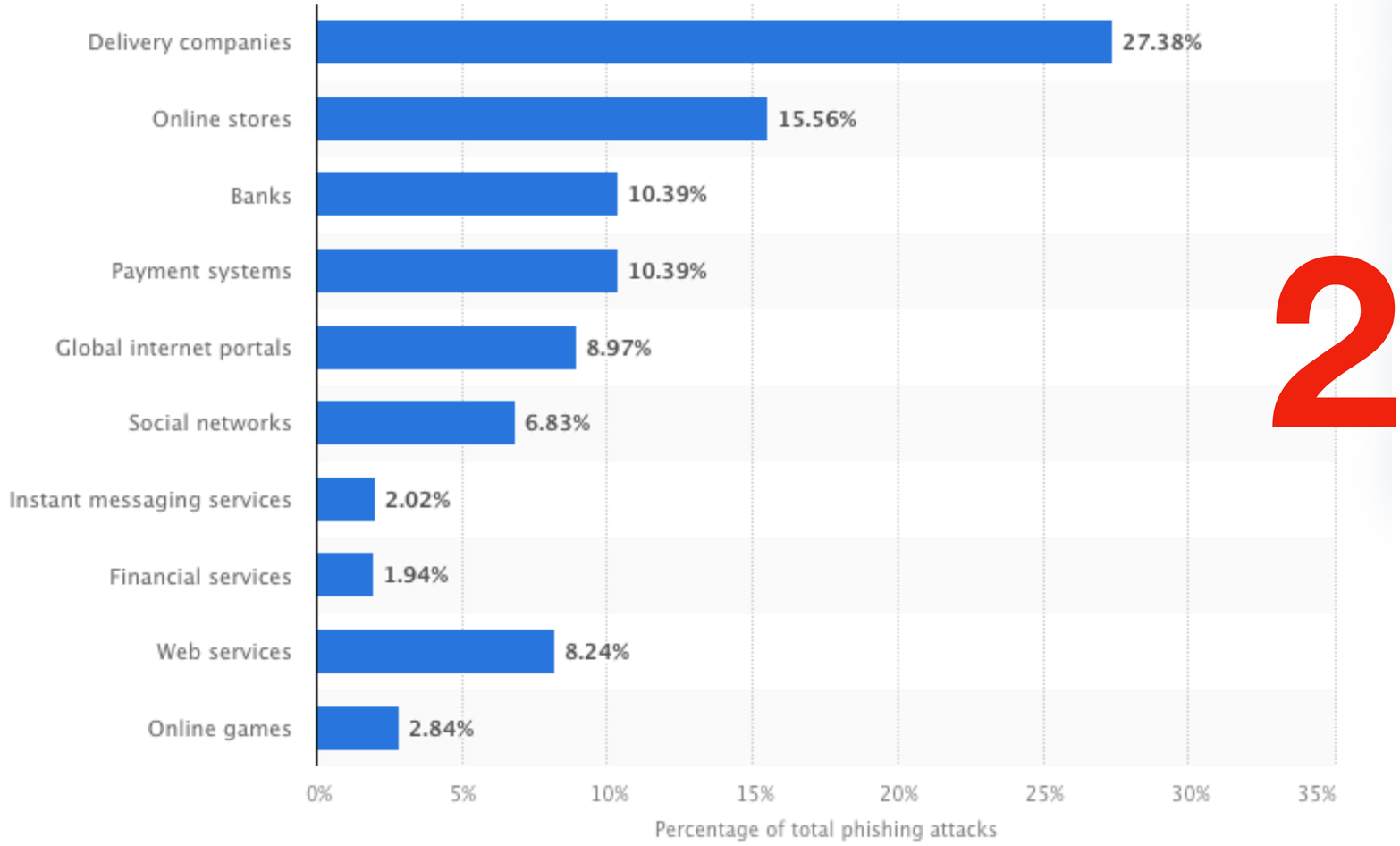
superbadsite.com/financez/banking.exe

Find Commonly Spoofed Brands

Maintain a list of brands used in phishing to search in your URLs

Brand Components in URLs

Worldwide organizations most targeted by phishing attacks in 2022, by industry



2021

https://www.statista.com/statistics/420442/organizations-most-affected-by-phishing-by-industry/

Maintain a list of brands used in phishing to search in your URLs

Brand Components in URLs

Top phishing brands in Q1 2023

Below are the top brands ranked by their overall appearance in brand phishing attempts:

1. Walmart (relating to 16% of all phishing attacks globally)
2. DHL (13%)
3. Microsoft (12%)
4. LinkedIn (6%)
5. FedEx (4.9%)
6. Google (4.8%)
7. Netflix (4%)
8. Raiffeisen (3.6%)
9. PayPal (3.5%)

https://www.checkpoint.com/press-releases/retail-giant-walmart-ranks-first-in-list-of-brands-most-likely-to-be-imitated-in-phishing-attempts-in-q1-2023/

Maintain a list of brands used in phishing to search in your URLs

Brand Components in URLs

Find matches based on brand

```
[‘walmart','citibank','hsbc','chase','wellsfargo','citi'
,'bankofamer']
```

Search for those words in the tokenized words

Building a list of ‘bad’ words

Using NLTK

Get the English Words Corpus

You only have to do this once

```
python3
import nltk
# Download the English words corpus
nltk.download("words")
```

Using the English Words Corpus

- We can find actual words
- But what about misspellings and words that sound similar?

```
apple-checker.org
apple-iclods.org
apple-uptoday.org
apple-search.info
```

Take words from known bad URLs (phishtank in this example) and use that as a ‘bad’ wordlist

Increase the list using known bad URLs

Word Lists and Lexicons

The NLTK data package also includes a number of lexicons and word lists. These are accessed just like text corpora. The following examples illustrate the use of the wordlist corpora:

```
>>> from nltk.corpus import names, stopwords, words
>>> words.fileids()
['en', 'en-basic']
>>> words.words('en')
['A', 'a', 'aa', 'aal', 'aalii', 'aam', 'Aani', 'aardvark', 'aardwolf', ...]
```

https://www.nltk.org/howto/corpus.html#word-lists-and-lexicons

```
site
index
site
square
hold
page
start
game
life
meta
work
page
manager
site
click
store
payment
millennium
veri
login
domain
manager
link
square
site
luxury
road
import
import
road
luxury
```

Also keep misspelled words

Word Lists and Lexicons

The NLTK data package also includes a number of lexicons and word lists. These are accessed just like text corpora. The following examples illustrate the use of the wordlist corpora:

```
>>> from nltk.corpus import names, stopwords, words
>>> words.fileids()
['en', 'en-basic']
>>> words.words('en')
['A', 'a', 'aa', 'aal', 'aalii', 'aam', 'Aani', 'aardvark', 'aardwolf', ...]
```

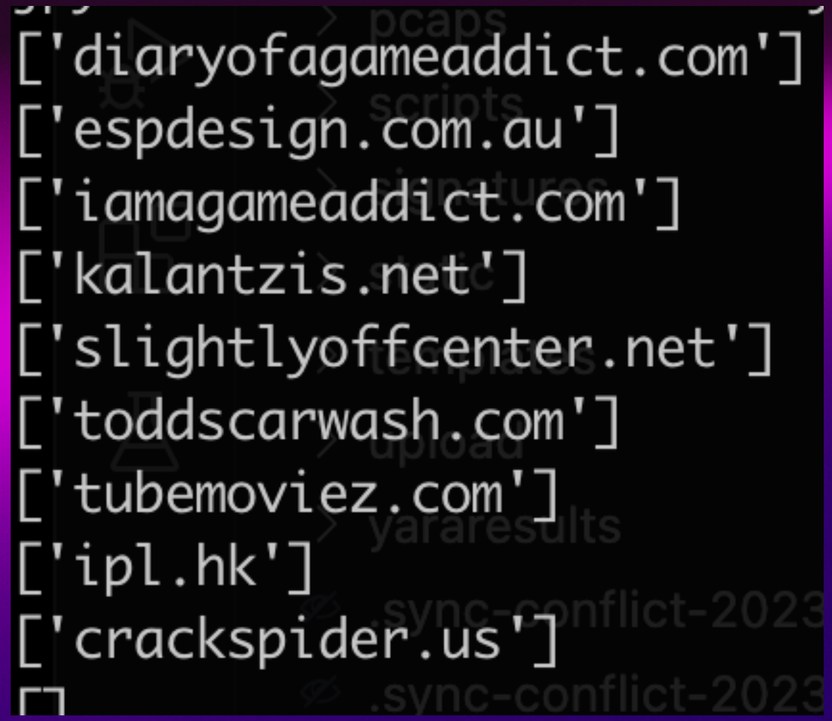
https://www.nltk.org/howto/corpus.html#word-lists-and-lexicons

```
site
index
site
square
hold
page
start
game
life
meta
work
page
manager
site
click
store
payment
millennium
veri
login
domain
manager
link
square
site
luxury
road
import
import
road
luxury
```

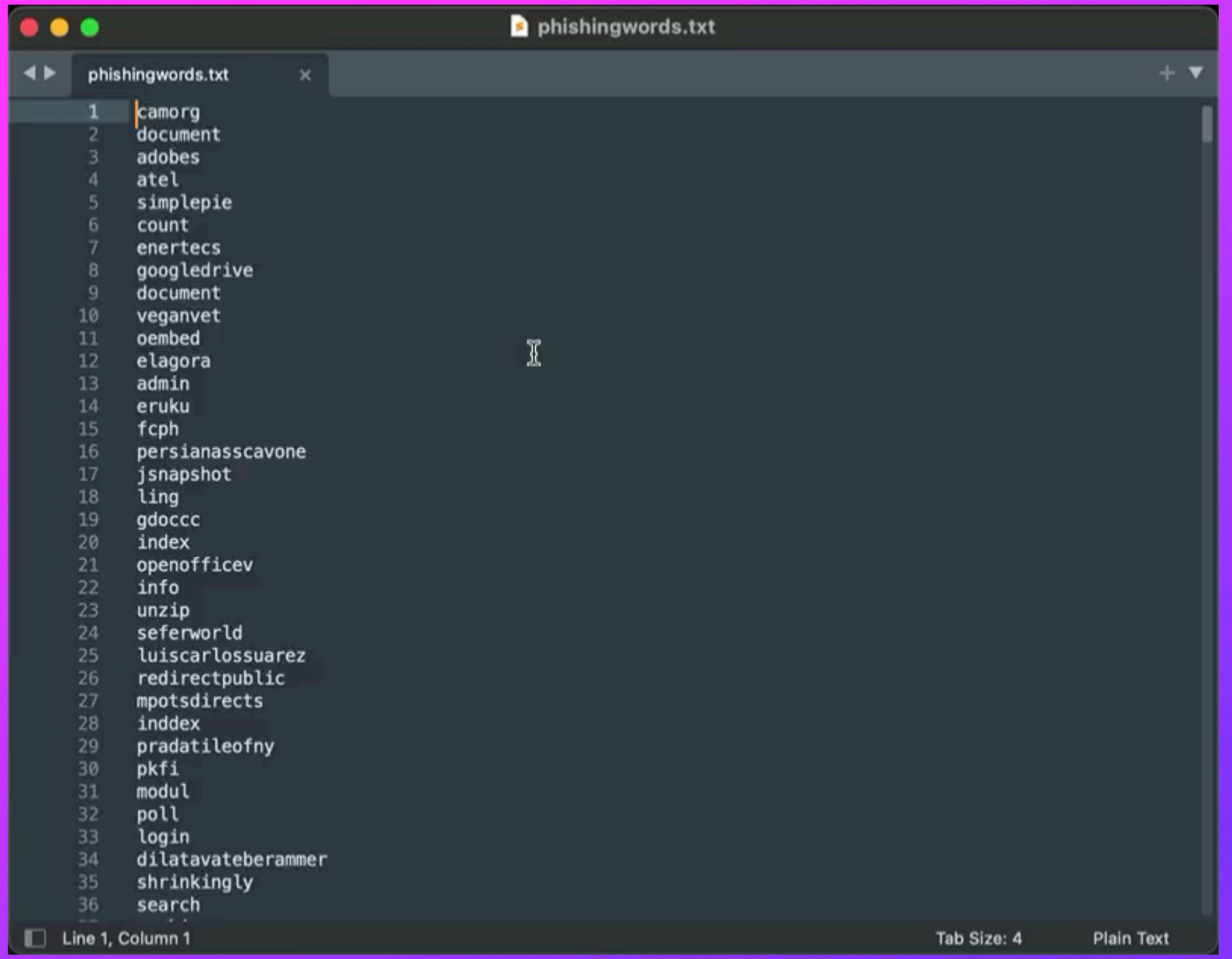
Check Spelling

pip3 install pyspellchecker

```python
from spellchecker import SpellChecker
spell = SpellChecker()
def find_unusual_words(text):
    words = text.split()
    unusual_words = [word for word in words if not spell.correction(word.lower())]
    return unusual_words
```



Increase the list using known bad URLs



Increase the list using known bad URLs

```
acount
acpaeqypt
```

```
actisnet
activaction
activar
```

```
activatemywebsetup
activaterequest
```

```
bankacilik
bankaustria
bankdanych
banker
bankershandbook
bankguaranteefacts
bankiing
bankilineitau
banking
bankingindiaupdate
bankingupdate
bankinnovation
banklogin
banknet
bankofamerica
bankofamericaonline
bankofamericasucks
```

I’ve (partially) done this

GitHub link at the end

Fuzzy Searches

Find misspelled words

Spellchecker python module

```
{
   ‘url': ‘benkofmaerical.com/benking',
   ‘tokens': [‘benkofmaerical', ‘com', ‘benking’],
   ‘possible_actual_words’: [‘benkofmaerical’, ‘benking’],
   ‘score’: 0,
}
```

Use Levenshtein distance to get fuzzy with the search

Levenshtein Distance

Use the list of ‘good’ words

```
wordlist =

['banking', 'pharma', ‘buy',
'walmart','citibank','hsbc','chase','wellsfargo','citi','bank
ofamer']
```

Use Levenshtein distance to get fuzzy with the search

Levenshtein Distance

Find potential matches

```
school76.irkutsk.ru/language/wellsfargo/online.htm, wellsfargo: wellsfargo
dogtassomine.com/wellsfargo/wellsfargo/identity.php, wellsfargo: wellsfargo
security.wellson.wellsfarg0.work/wells8581/wells/onlinesession/kk/, wellsfargo: wellsfarg0
usgiftwholesale.com/jewelrywatch/update.citibank.com.my/home/, citibank: citibank
hghhgggj.online/chase2/index(1).html, chase: chase2
charles1122.com/walmart/walmart.com/, walmart: walmart
```

Levenshtein Distance

Increase with commonly used and misspelled words

https://01.walmart-shop.store/

```
def find_suspicious_words_levenshtein(self, urls):
    wordlist = ['banking', 'pharma', 'buy', 'walmart', 'citibank', 'hsbc', 'chase', 'wellsfargo', 'citi', 'bankofamer']
```

https://account-update.tn2buy.cn

https://httpglobalbank-servecomve.cliente88282.repl.co

Add more info to the dictionary blob that is being used to save info

Levenshtein Distance

Add to dictionary blob

```
{
   ‘url': ‘benkofmaerical.com/benking',
   ‘tokens': [‘benkofmaerical', ‘com', ‘benking’],
   ‘possible_actual_words’: [‘benkofmaerical’, ‘benking’],
   ‘levenshtein_distance’:
      {
         ‘distance': 1,
         ‘match': ‘banking',
         ‘word': ‘benking’
      },
   ‘score’: 0,
}
```

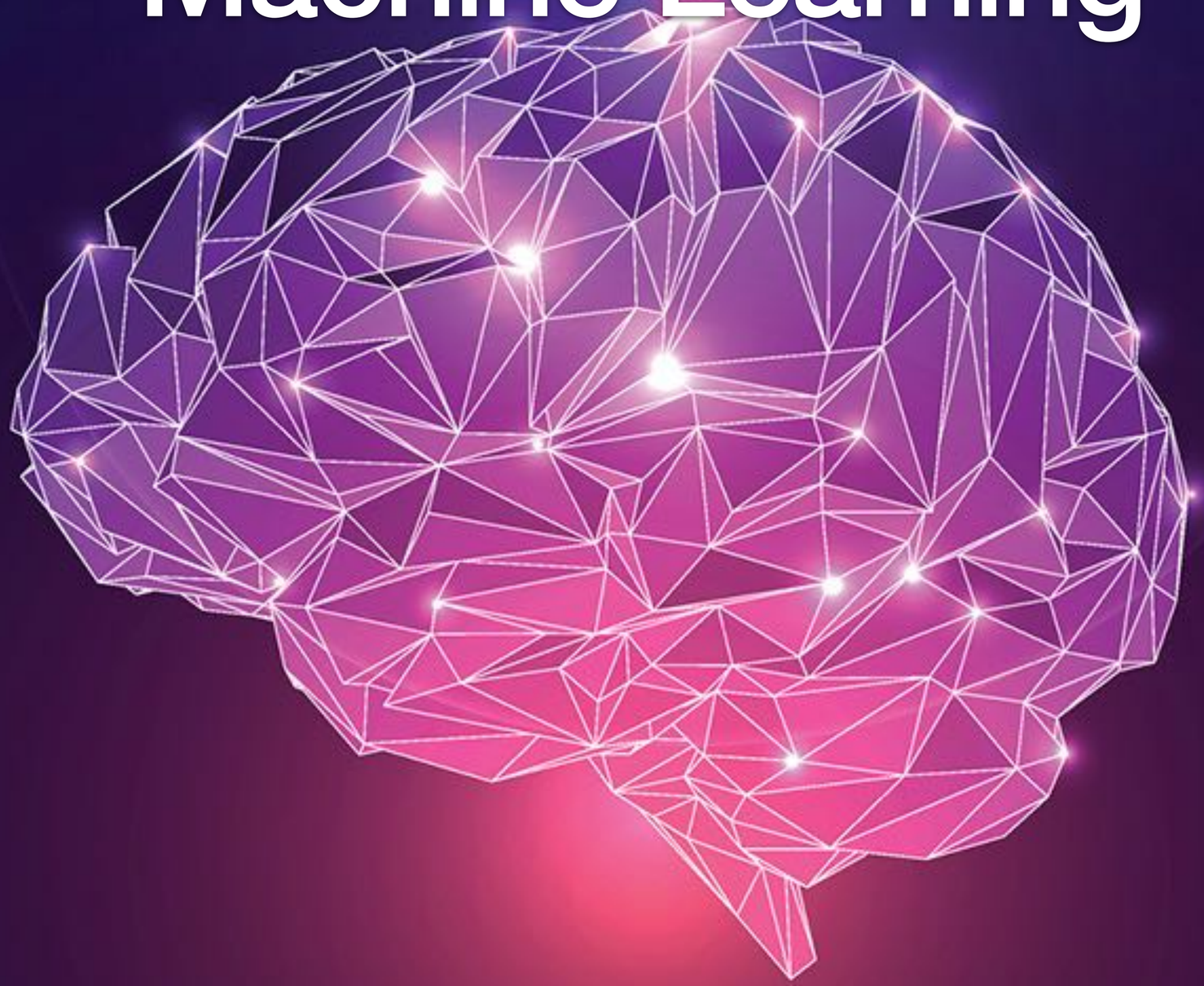
Levenshtein hit, suggesting it’s bad. Set the score down by 1

Levenshtein Distance

Lower URL Score

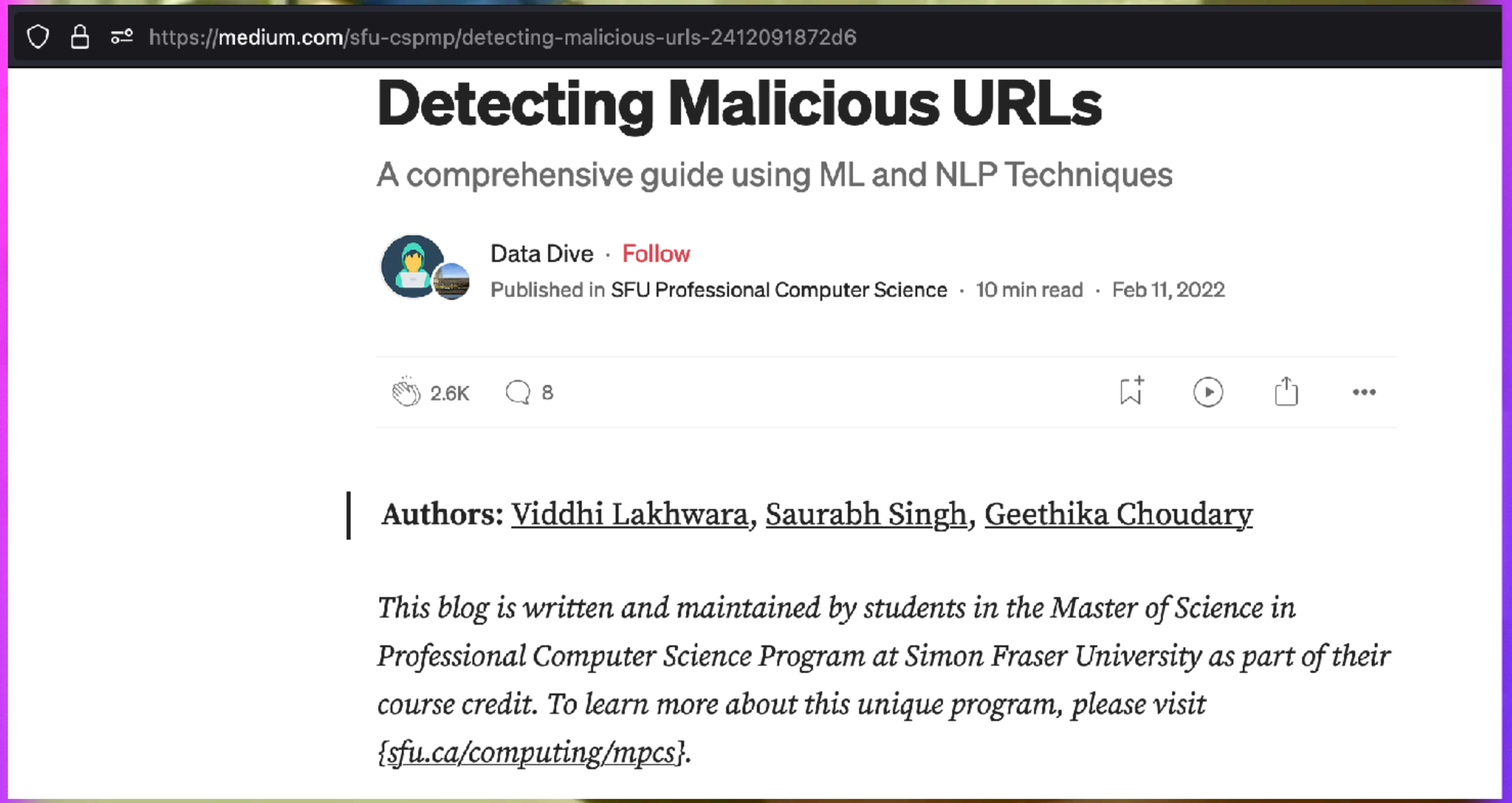

```
{
  ‘url': ‘benkofmaerical.com/benking',
  ‘tokens': [‘benkofmaerical', ‘com', ‘benking’],
  ‘possible_actual_words’: [‘benkofmaerical’, ‘benking’],
  ‘levenshtein_distance’:
    {
      ‘distance': 1,
      ‘match': ‘banking',
      ‘word': ‘benking’
    },
  ‘score’: -1,
}
```

Machine Learning



~~Wheel Reinvention~~

Thank you, other people!



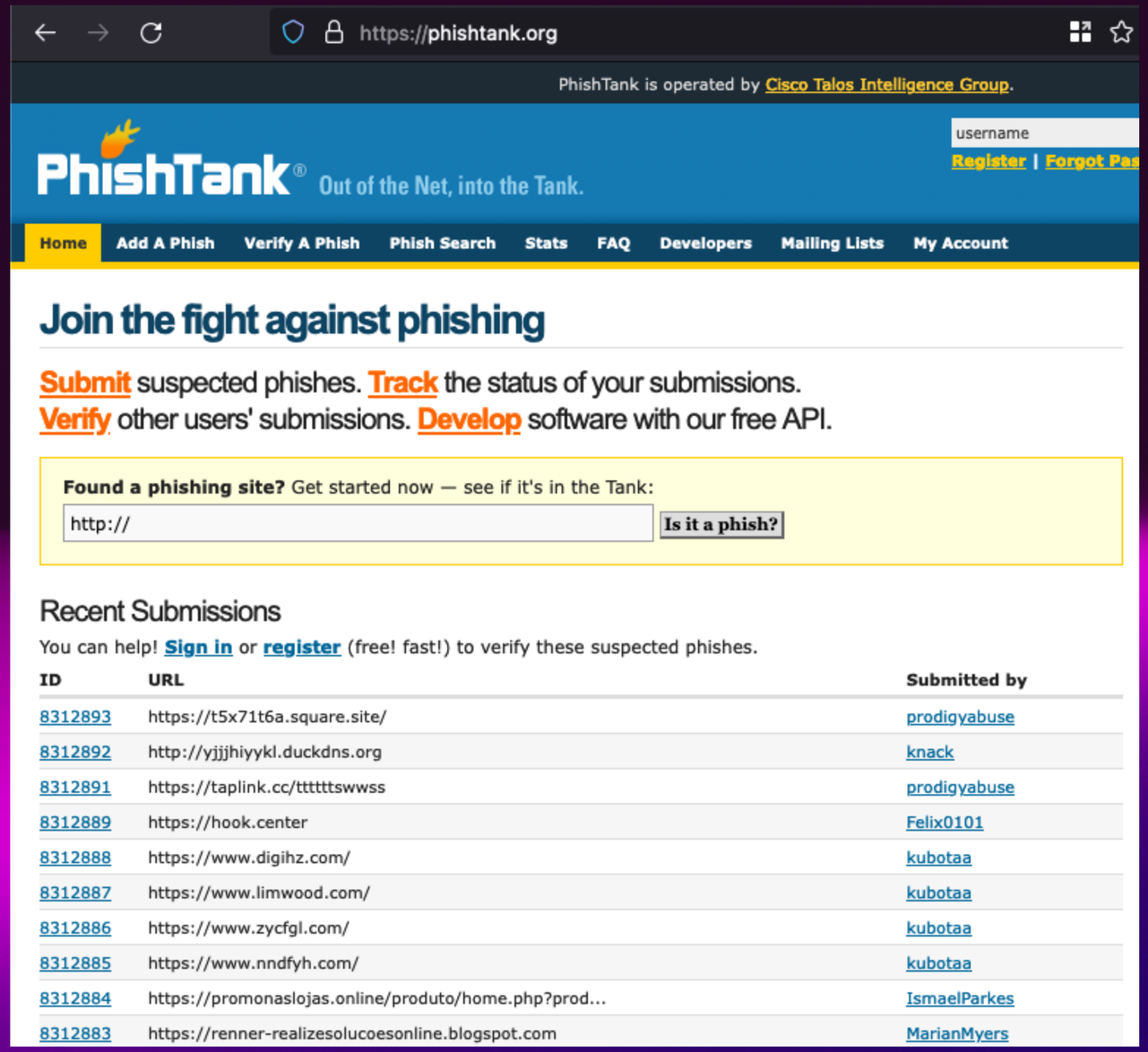
https://medium.com/sfu-cspmp/detecting-malicious-urls-2412091872d6

Collect Data Using Existing Datasets

ska.energia.cz/download/imer.up	bad
obyz.de/webproxytest.txt	bad
callingcardsinstantly.com/webalizer/050709wareza/crack=17=keygen=serial.html	bad
dawnframing.com/webalizer/050709wareza/crack=17=keygen=serial.html	bad
free-crochet-pattern.com/webalizer/050709wareza/crack=17=keygen=serial.html	bad
js.tongji.linezing.com/1189582/tongji.js	bad
oiluk.net/cache/cache_94afbfb2f291e0bf253fcf222e9d238e_180836f9b956ab9d91a50f9add968699	bad
adgallery.whitehousedrugpolicy.gov/members/Miley-Cyrus-Nude/default.aspx	bad
vvvic.com	bad
sportsulsan.co.kr/poll/aipi/id.txt	bad

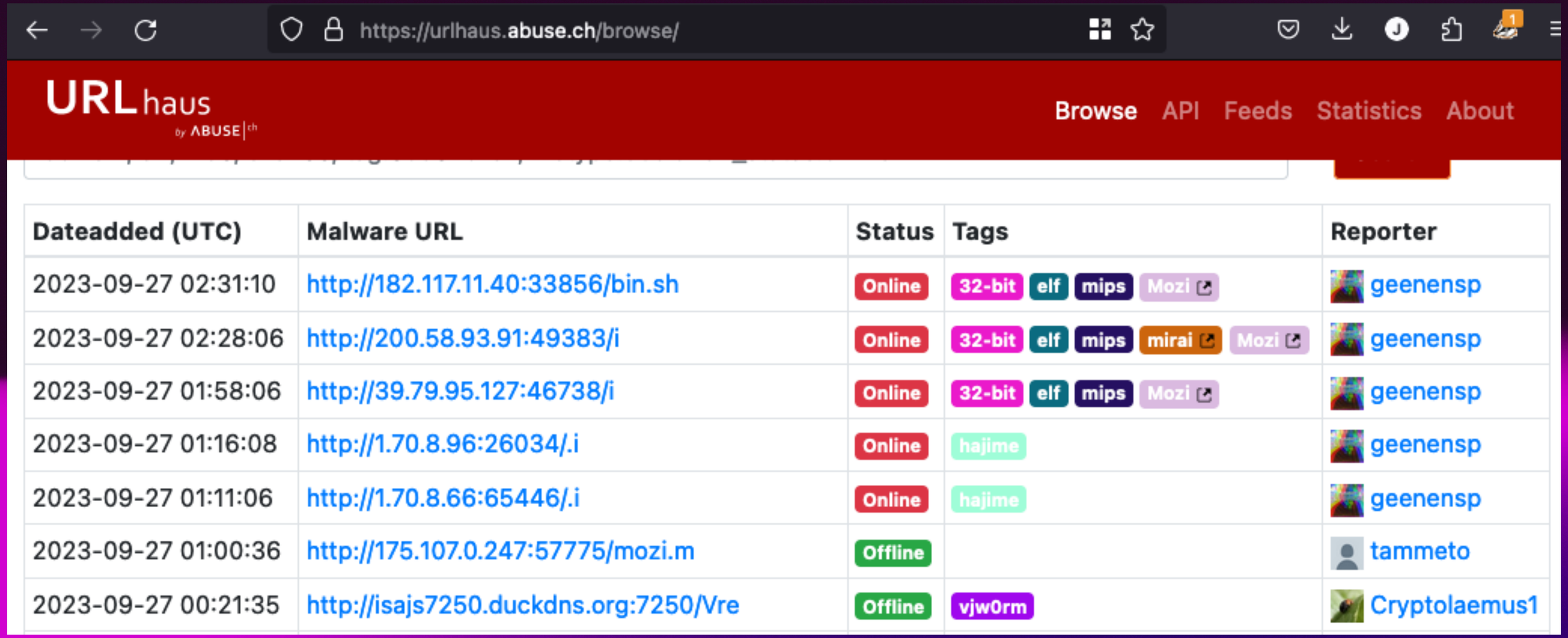
https://www.kaggle.com/datasets/antonyj453/urldataset

Collect Data Using Existing Datasets



https://phishtank.org/

Collect Data Using Existing Datasets



Dateadded (UTC)	Malware URL	Status	Tags	Reporter
2023-09-27 02:31:10	http://182.117.11.40:33856/bin.sh	Online	32-bit elf mips Mozi	geenensp
2023-09-27 02:28:06	http://200.58.93.91:49383/i	Online	32-bit elf mips mirai Mozi	geenensp
2023-09-27 01:58:06	http://39.79.95.127:46738/i	Online	32-bit elf mips Mozi	geenensp
2023-09-27 01:16:08	http://1.70.8.96:26034/.i	Online	hajime	geenensp
2023-09-27 01:11:06	http://1.70.8.66:65446/.i	Online	hajime	geenensp
2023-09-27 01:00:36	http://175.107.0.247:57775/mozi.m	Offline		tammeto
2023-09-27 00:21:35	http://isajs7250.duckdns.org:7250/Vre	Offline	vjw0rm	Cryptolaemus1

https://urlhaus.abuse.ch

Pre-process

Getting the data ready

Remove empty lines and newlines

```
url.strip()
```

Remove http:// and https://

```
url.replace(“https://“,””)
url.replace(“http://“,"")
```

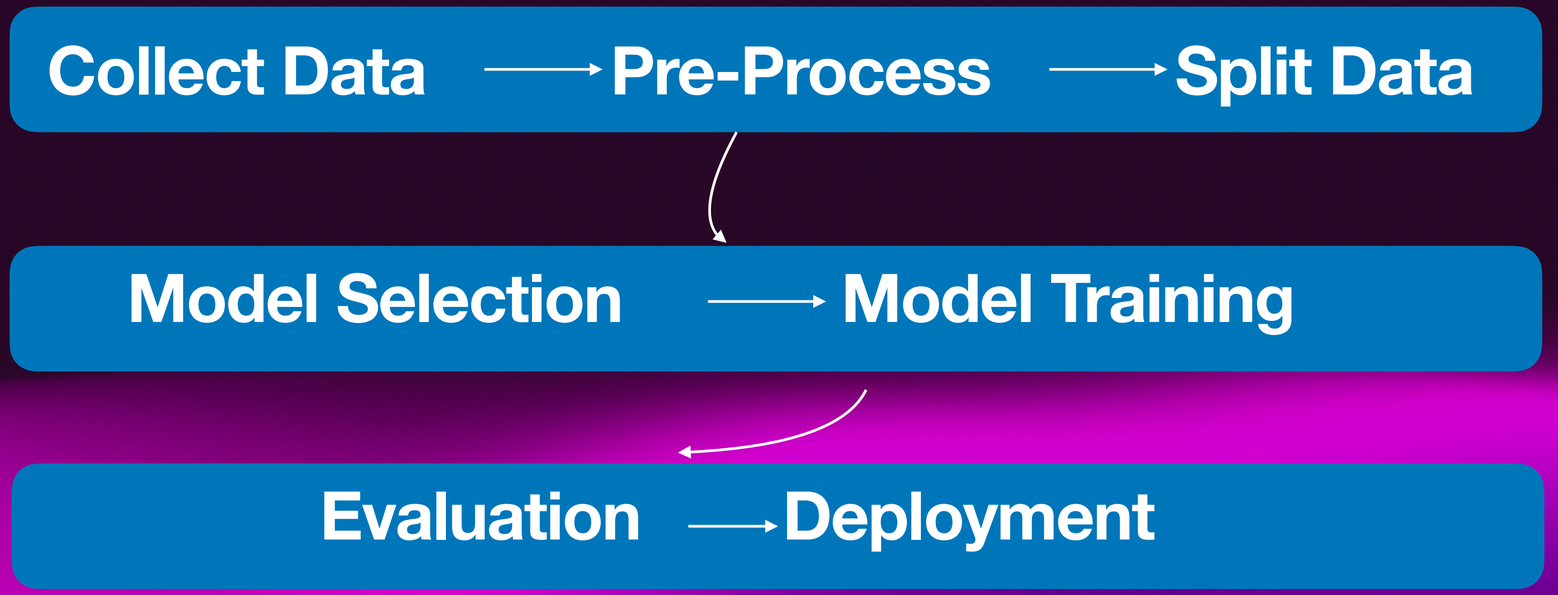
Remove file extensions

```
url = re.sub(r'\.[A-Za-z0-9]+/*','',url)
```

Split Data

20% Training

80% Testing/Validation


Collect Data
Pre-Process
Split Data
Model Selection
Model Training
Evaluation
Deployment

We'll load the pre-trained joblib file so classification can be done quickly

Saving the classifier

```
1 # Save the trained classifier to a file
2 classifier_filename = 'url_maliciousness_trained_classifier.joblib'
3 dump(classifier, classifier_filename)
```

DGA’s

README.md

DGA-Detection

More and more malware is being created with advanced blocking circumvention techniques. One of the most prevalent techniques being used is the use of Domain Generation Algorithms which periodically generates a set of Domains to contact a C&C server. The majority of these DGA domains generate random alphanumeric strings which differ significantly in structure to a standard domain. By looking at the frequency that a set of bigrams in a domain occur within the Alexa top 1M, we were able to detect whether a domain was structured with a random string or if it was a legitimate human readable domain. If a domain is comprised nearly entirely of low frequency bigrams which occurred rarely within the Alexa top 1m then the domain would more likely be a random string. Bigrams of a vowel and constants occurred the most frequent whereas characters and integers occurred the least frequent. The script was ran against 100,000 GameoverZeus domains and had a detection rate of 100% and a false positive rate against the Alexa top 1m of 8% without any domain whitelisting being applied.

This System has been tested on Ubuntu and RaspberryPi. Currently I have my raspberrypi setup as a DNS server using Bind9. The DGA-Detection script is also run on the raspberrypi and reads the requests. The requests are then processed to determine if they are a potential DGA or not.

https://github.com/philarkwright/DGA-Detection

README.md

DGA Detector

DGA Domains detection

DGA domain detection is based on ngram analysis with trained markov chain model. It is incorporate code by https://github.com/rrenaud/Gibberish-Detector

The decision is based solely on results by this check.

In addition to ngram analysis it is also provide additional methods:

- entropy - High entropy is another indicator of DGA domain. Threshold is 3.8
- consonants - High consonants count is an indicator of DGA domain. Threshold is 7
- length - High domain length can also indicate DGA. Threshold is 12.

https://github.com/exp0se/dga_detector

Punycode

- **Domain:** xn--example-c3d.com

  URL: xn--example-c3d.com

  URL Length: 18

  Possible Actual Words:

  Reason: punycode

  Score: -1

  Punycode Match:

    - **Match:** exam´ple

Starting to Set a Score

- **Domain:** bezproudoff.cz
  URL: bezproudoff.cz
  URL Length: 13
  Possible Actual Words:
    - bezproudoff
  Reason: Umbrella Investigate, virustotal
  Score: -1

- **Domain:** westlifego.com
  URL: westlifego.com/js/jquery-1.3.2.min.js
  URL Length: 30
  Possible Actual Words:
    - westlifego
  Reason: virustotal
  Score: -1

- **Domain:** benkofmaerical.com

  URL: benkofmaerical.com/benking

  URL Length: 24

  Possible Actual Words:
    - benkofmaerical
    - benking

  Reason: levenshtein

  Score: -1

  Levenshtein Match:
    - **Distance:** 1

      Match: banking

      Word: benking

1000 '*maybe* Good' URLs

Score by Detection Method	Count of url
-2	**66**
ML SVM Model, DGA Detection	12
ML SVM Model, Virustotal	54
-1	**225**
DGA Detection	66
ML SVM Model	137
Virustotal	22
0	**629**
benign	629
Grand Total	**920**

920 Unique URLs

629 Benign

291 Malicious (potential FP)

1000 '*probably* bad' URLs

Score by Detection Method	Count of url
-4	**3**
ML SVM Model, DGA Detection, Investigate, Virustotal	3
-3	**186**
DGA Detection, Investigate, Virustotal	11
ML SVM Model, DGA Detection, Virustotal	159
ML SVM Model, Investigate, Virustotal	16
-2	**284**
DGA Detection, Investigate	3
DGA Detection, Virustotal	21
Investigate, Virustotal	118
ML SVM Model, DGA Detection	8
ML SVM Model, Investigate	9
ML SVM Model, Virustotal	125
-1	**362**
DGA Detection	12
Investigate	51
ML SVM Model	86
Virustotal	213
0	**155**
benign	155
Grand Total	**990**

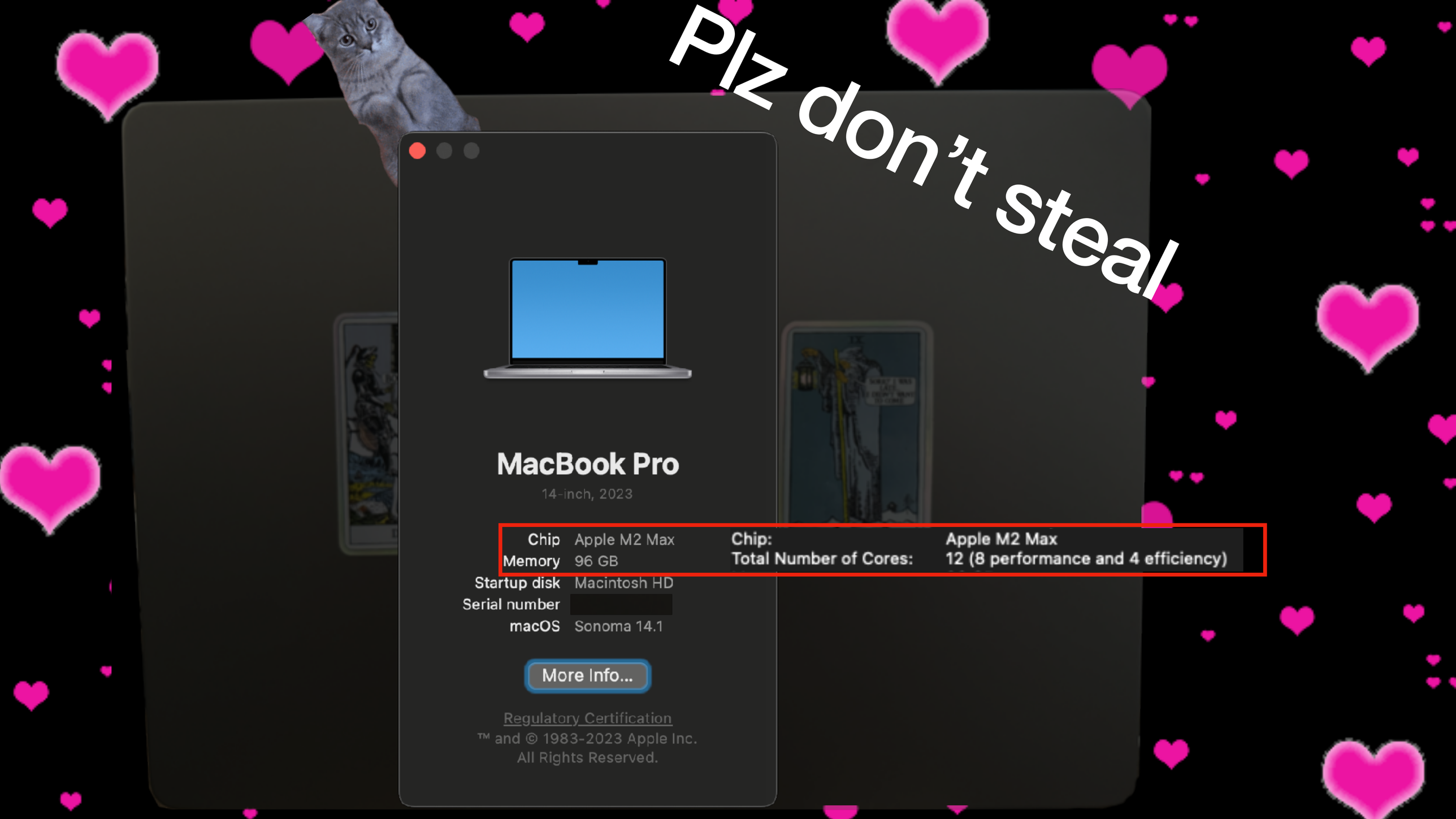
990 Unique URLs

835 Malicious

155 Benign (potential FN)

Optimization

My Laptop


Plz don't steal
MacBook Pro
14-inch, 2023
Chip Apple M2 Max
Memory 96 GB
Startup disk Macintosh HD
Serial number
macOS Sonoma 14.1
More Info...
Regulatory Certification
™ and © 1983-2023 Apple Inc.
All Rights Reserved.
Chip: Apple M2 Max
Total Number of Cores: 12 (8 performance and 4 efficiency)

Things are still a little slow

Things are slow

Sending one URL at a time takes time

```
URL: gooflashcorp.com/demo/k231/y/yh/cameo.php?continue=to&amp;inbox=xclus
iv-3d
TOKENS: ['gooflashcorp', 'com', 'demo', 'k231', 'y', 'yh', 'cameo', 'php?c
```

Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	Kind	% GPU	GPU
Python	100.0	29.48	12	0	Apple	0.0	
Python	0.0	3.59	12	0	Apple	0.0	

```
Total Number of URLs with Suspicious Words via levenshtein: 28
----------
Script execution time: 30.67 seconds
```

Multi-threading

Multi-threading

Sending one URL at a time, but via multithreading is a little faster

```
URL: rdlbd.net/dropbox/us-mg5.mail.yahoo.com/pass.php?neo.launch?.rand=1qh
ua0f7o2jut
TOKENS: ['rdlbd', 'net', 'dropbox', 'us-mg5', 'mail', 'yahoo', 'com', 'pas
s', 'php?neo', 'launch?', 'rand=1qhua0f7o2jut']
POSSIBLE WORDS: ['rdlbd', 'dropbox', 'mail', 'yahoo', 'pass']
```

Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	Kind	%
Python	277.9	15.11	72	261	Apple	
Python	0.0	3.59	12	0	Apple	

```
Total Number of URLs with Suspicious Words via levenshtein: 28
----------
Script execution time: 25.47 seconds with chunk size of 59
```

Multi-threading

Sending one URL at a time, but via multithreading is a little faster

```
URL: rdlbd.net/dropbox/us-mg5.mail.yahoo.com/pass.php?neo.launch?.rand=1qh
ua0f7o2jut
TOKENS: ['rdlbd', 'net', 'dropbox', 'us-mg5', 'mail', 'yahoo', 'com', 'pas
s', 'php?neo', 'launch?', 'rand=1qhua0f7o2jut']
POSSIBLE WORDS: ['rdlbd', 'dropbox', 'mail', 'yahoo', 'pass']
```

Saved 5 seconds!

Process Name			Wake Ups	Kind
Python	277.9	15.11	72	261	Apple
Python	0.0	3.59	12	0	Apple

```
Total Number of URLs with Suspicious Words via levenshtein: 28
----------
Script execution time: 25.47 seconds with chunk size of 59
```

Multi-threading

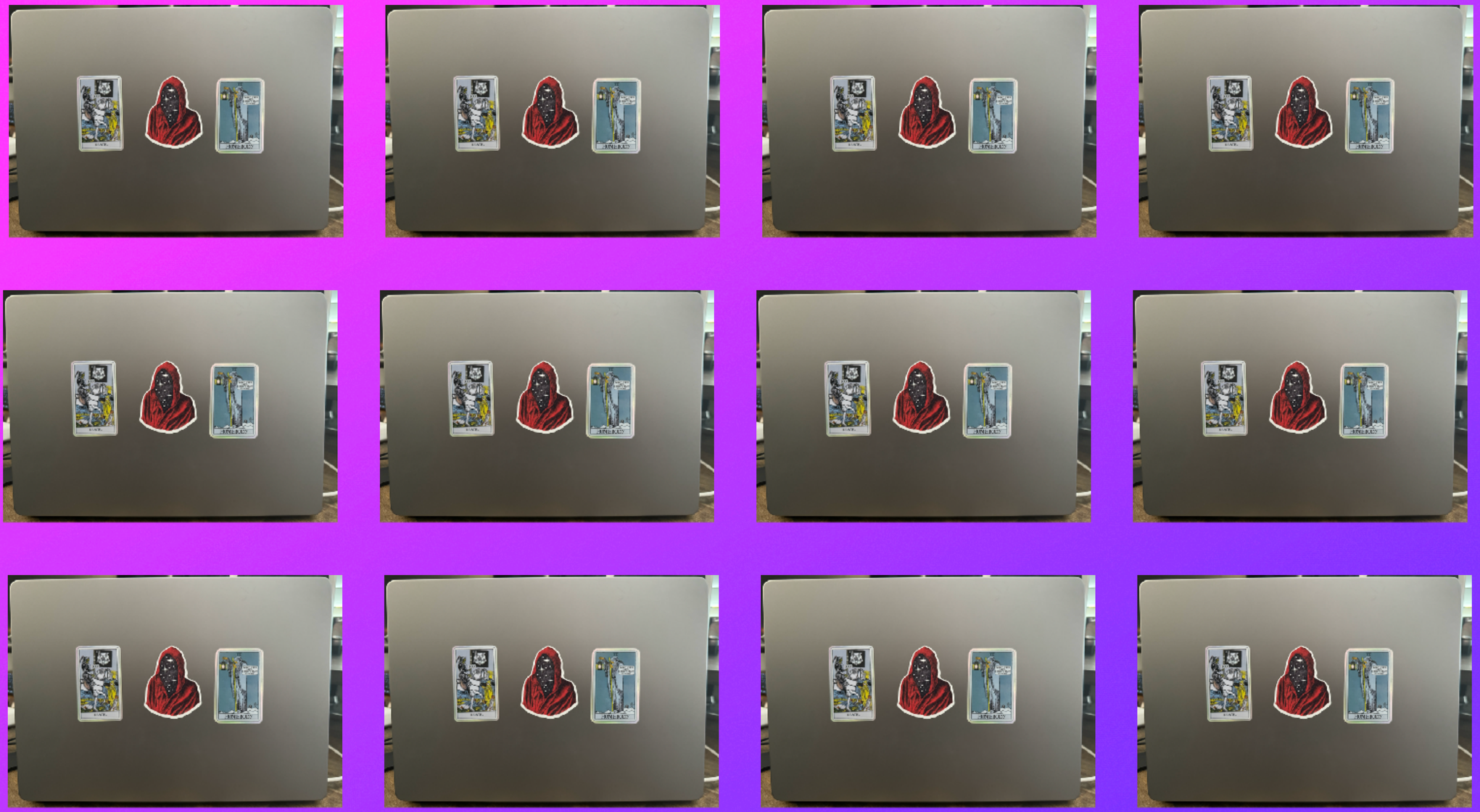
Sending 1000 URLs at a time, but via multithreading is a little faster.

```
Script execution time: 27.92 seconds with chunk size of 1000
```

Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	Kind	% GPU	GPU Time
Python	239.7	19.33	998	227	Apple	0.0	0.00
Python	0.0	3.59	12	0	Apple	0.0	0.00

Multi-Processing

12 of my laptops!



Multi-Processing

Sending number of URLs/processor cores at a time, but via multiprocessing is super fast!
…my fan also turns on

```
URL: soloseg.com/qq/hbb/hbb/hbb/products/id.php?l=_JeHFUq_VJOXK0QWHtoGYDw_Product-UserID&amp
TOKENS: ['soloseg', 'com', 'qq', 'hbb', 'hbb', 'hbb', 'products', 'id', 'php?l=_JeHFUq_VJOXK
kswarellc', 'com']
POSSIBLE WORDS: ['soloseg', 'products']

URL: gooflashcorp.com/demo/k231/y/yh/cameo.php?continue=to&amp;inbox=Xclusiv-3D|&amp;login=
TOKENS: ['gooflashcorp', 'com', 'demo', 'k231', 'y', 'yh', 'cameo', 'php?continue=to&amp;inb
POSSIBLE WORDS: ['gooflashcorp', 'demo', 'cameo']



Total Number of URLs: 4,999
Total Number of Words: 31,543
Script execution time: 6.29 seconds with chunk size of 12
jpyorre@dievortex URL Analysis with ML Research %
```

Multi-Processing

Sending number of URLs/processor cores at a time, but via multiprocessing is super fast!
…my fan also turns on

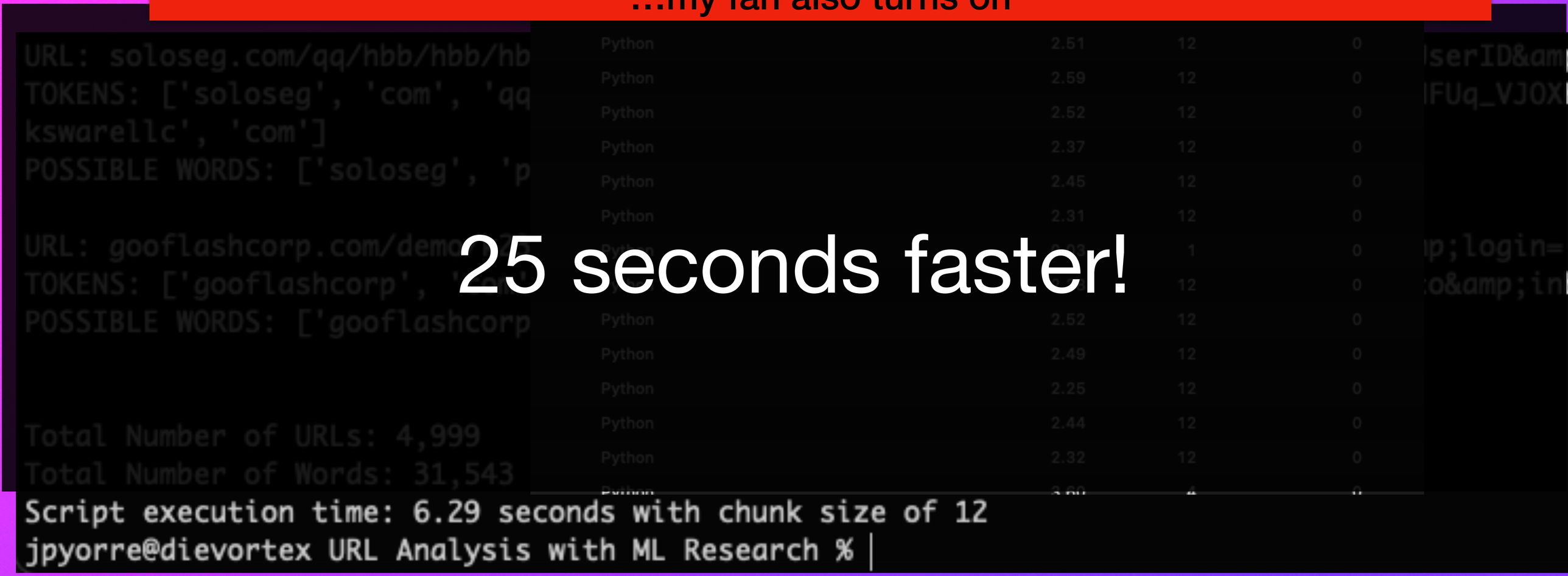
```
URL: soloseg.com/qq/hbb/hbb/hb
TOKENS: ['soloseg', 'com', 'qq
kswarellc', 'com']
POSSIBLE WORDS: ['soloseg', 'p

URL: gooflashcorp.com/demo/k23
TOKENS: ['gooflashcorp', 'com'
POSSIBLE WORDS: ['gooflashcorp


Total Number of URLs: 4,999
Total Number of Words: 31,543
Script execution time: 6.29 seconds with chunk size of 12
jpyorre@dievortex URL Analysis with ML Research %
```

Python	2.51	12	0
Python	2.59	12	0
Python	2.52	12	0
Python	2.37	12	0
Python	2.45	12	0
Python	2.31	12	0
Python	0.03	1	0
Python	2.38	12	0
Python	2.52	12	0
Python	2.49	12	0
Python	2.25	12	0
Python	2.44	12	0
Python	2.32	12	0
Python	3.60	4	0

```
serID&amp
FUq_VJOX
p;login=
o&amp;inl
```

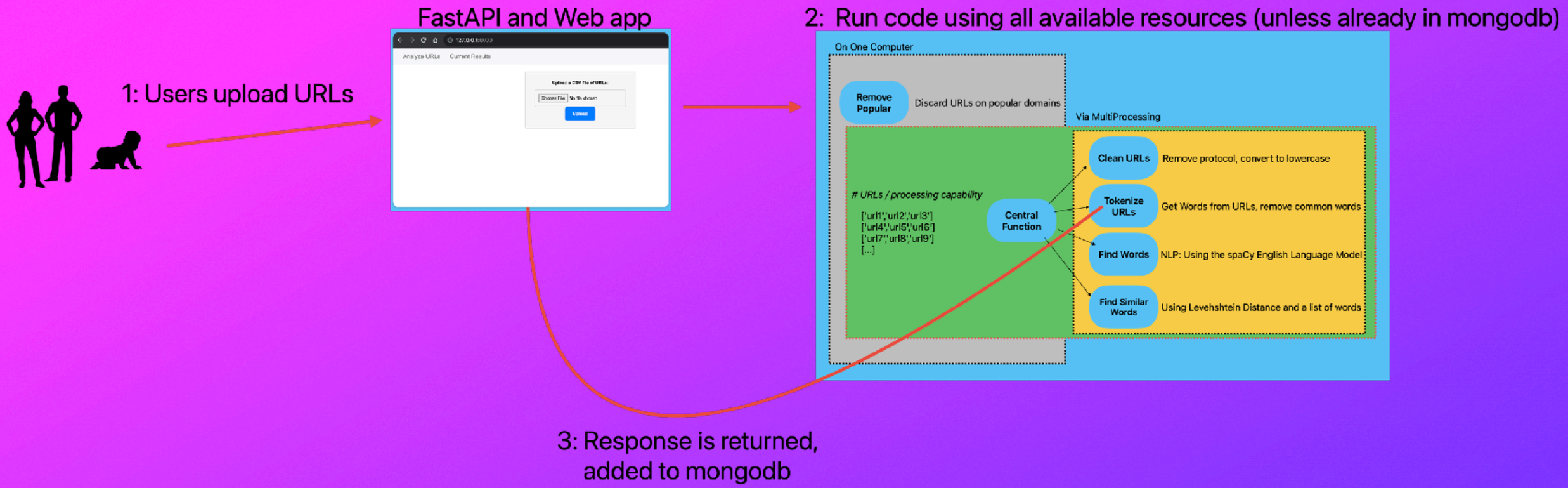
Multi-Processing

Sending number of URLs/processor cores at a time, but via multiprocessing is super fast!
…my fan also turns on



25 seconds faster!

The architecture using one machine and multiprocessing



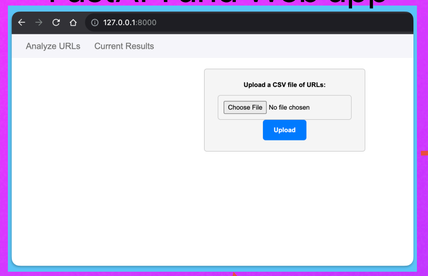
How about ‘no machines’?

Lambdas

The architecture using one machine for the web app and AWS Lambda for the functions

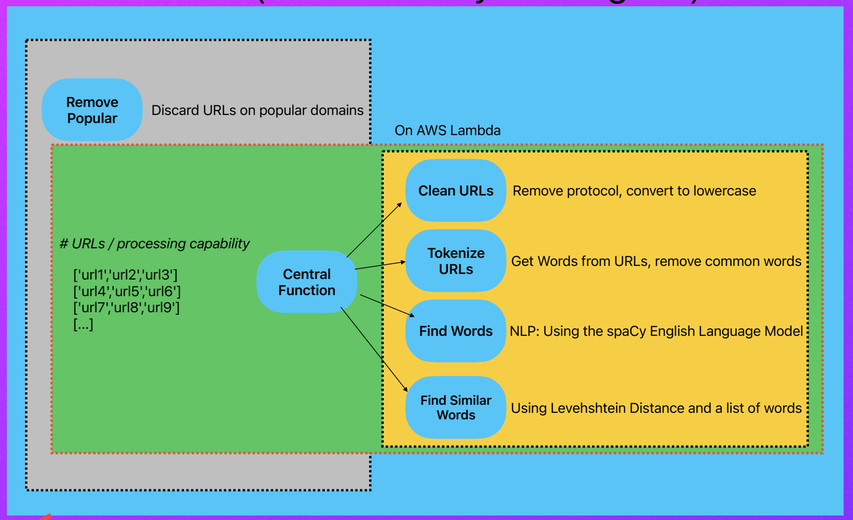
1: Users upload URLs

FastAPI and Web app



2: URLs cleaned and sent to Lambda

AWS Lambdas (unless already in mongodb)



3: Response is returned, added to mongodb

AWS Lambda: Testing one function

Environment Vari × lambda_function ×

```python
import json
import Levenshtein

def lambda_handler(event, context):
    wordlist = ['banking', 'pharma', 'buy', 'walmart','citibank','hsbc','chase','wellsfargo','citi','bankofamer']
    data = json.loads(event["body"])

    threshold = 2
    returnlist = []
    for w in data:
        listkeeper = []
        for i in w['possible_actual_words']:
            for word in wordlist:
                if word == 'pharm':
                    threshold = 2
                else:
                    threshold = 1
                distance = Levenshtein.distance(word, i)
                if distance <= threshold and len(i) > 5:
                    w['levenshtein_distance'] = {}
                    w['levenshtein_distance']['word'] = i
                    w['levenshtein_distance']['match'] = word
                    w['levenshtein_distance']['distance'] = distance
                    listkeeper.append(w)
                    break  # Break out of the inner loop to include the entire word
        if listkeeper:
            for item in listkeeper:
                if len(item) > 1:
                    returnlist.append(item)

    return returnlist
```

Lambda vs Local: *5000 domains, reduced to 4,575*

```
{'levenshtein_distance': {'distance': 1, 'match': 'banking', 'word': 'benking'},
 'possible_actual_words': ['benkofmaerical', 'benking'],
 'score': -1,
 'tokens': ['benkofmaerical', 'com', 'benking'],
 'url': 'benkofma
 'url_length': 24
```

4,575 Total Domains
0 Popular URLs
4,575 Unpopular URLs
4,553 Benign URLs (so far)

22 Bad URLs
22 Suspicious URLs via Levenshtein match

Lambda:

Script execution time: 10.58

Local:

Script execution time: 10.47

How much is it? Nobody knows….

Lambda Cost:

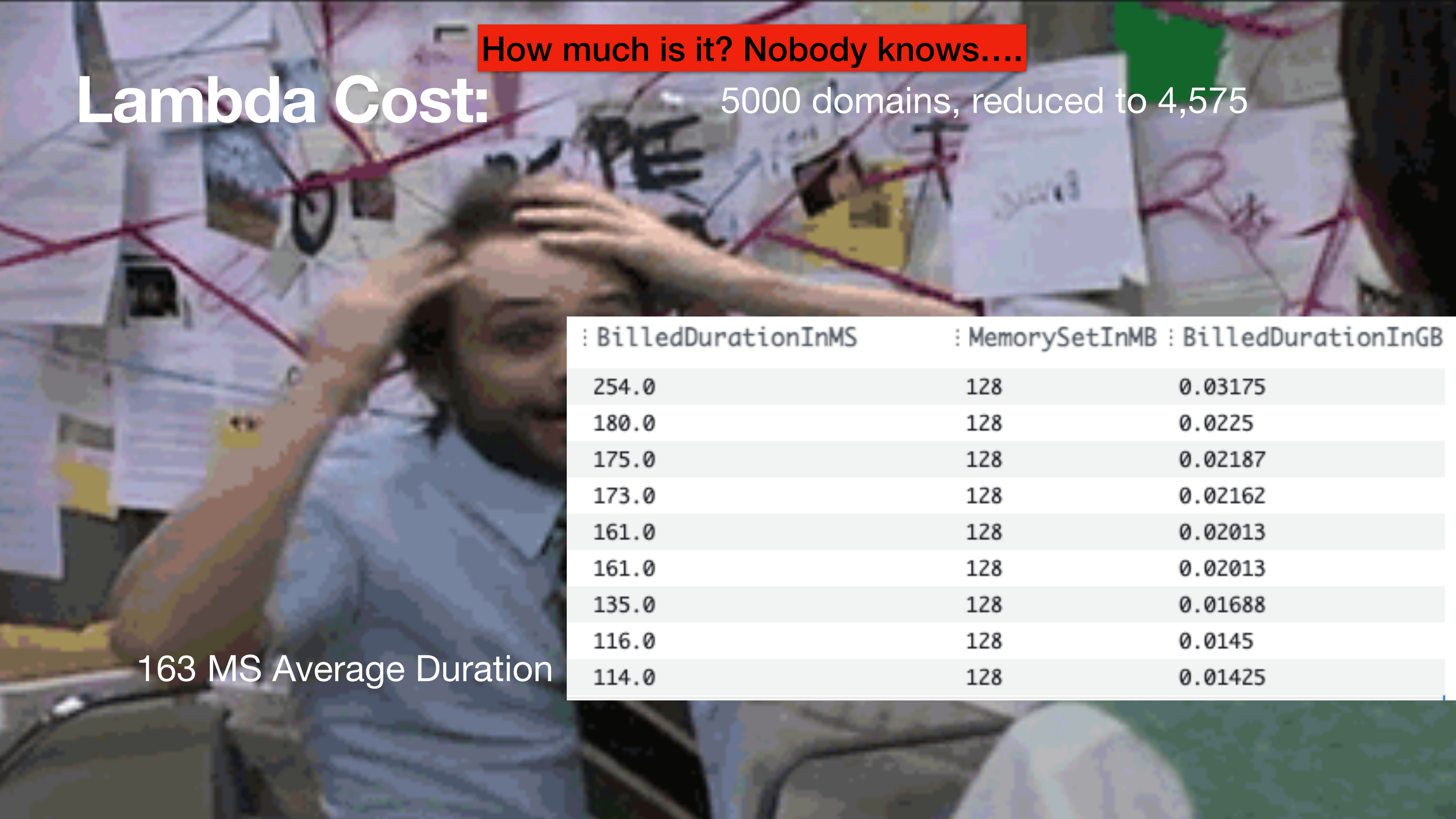
5000 domains, reduced to 4,575

Last event time
2023-10-11 12:08:34 (UTC-07:00)
2023-10-11 12:08:34 (UTC-07:00)
2023-10-11 12:08:34 (UTC-07:00)
2023-10-11 12:08:34 (UTC-07:00)
2023-10-11 12:08:33 (UTC-07:00)
2023-10-11 12:08:33 (UTC-07:00)
2023-10-11 12:08:33 (UTC-07:00)
2023-10-11 12:08:33 (UTC-07:00)
2023-10-11 12:08:33 (UTC-07:00)

4,575 Domains:

9 Function Calls

(508 domains each)



Lambda Cost:

How much is it? Nobody knows….

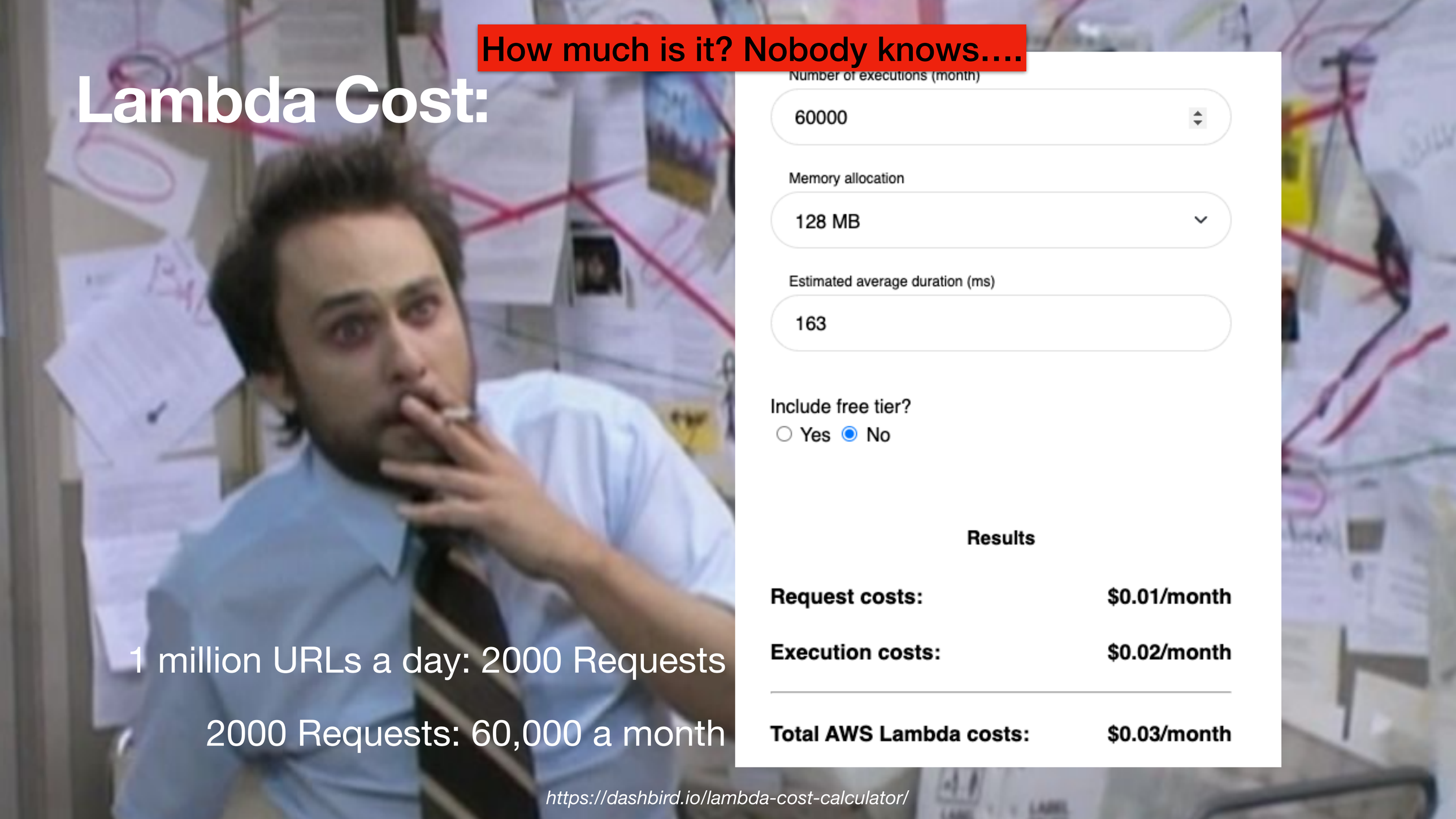
5000 domains, reduced to 4,575

BilledDurationInMS	MemorySetInMB	BilledDurationInGB
254.0	128	0.03175
180.0	128	0.0225
175.0	128	0.02187
173.0	128	0.02162
161.0	128	0.02013
161.0	128	0.02013
135.0	128	0.01688
116.0	128	0.0145
114.0	128	0.01425

163 MS Average Duration

Lambda Cost:

How much is it? Nobody knows….



1 million URLs a day: 2000 Requests

2000 Requests: 60,000 a month

https://dashbird.io/lambda-cost-calculator/

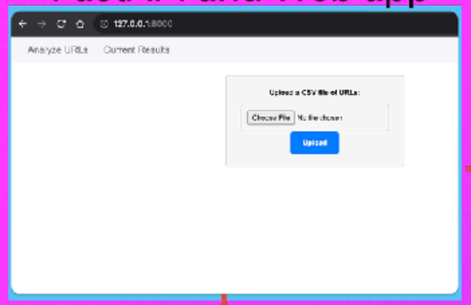
Distributed Computing

Expensive, then cheap (for a little while)

Lots of Machines!

1: Users upload URLs

FastAPI and Web app



2: URLs cleaned and sent to workers

Rabbit MQ Worker

Rabbit MQ Worker

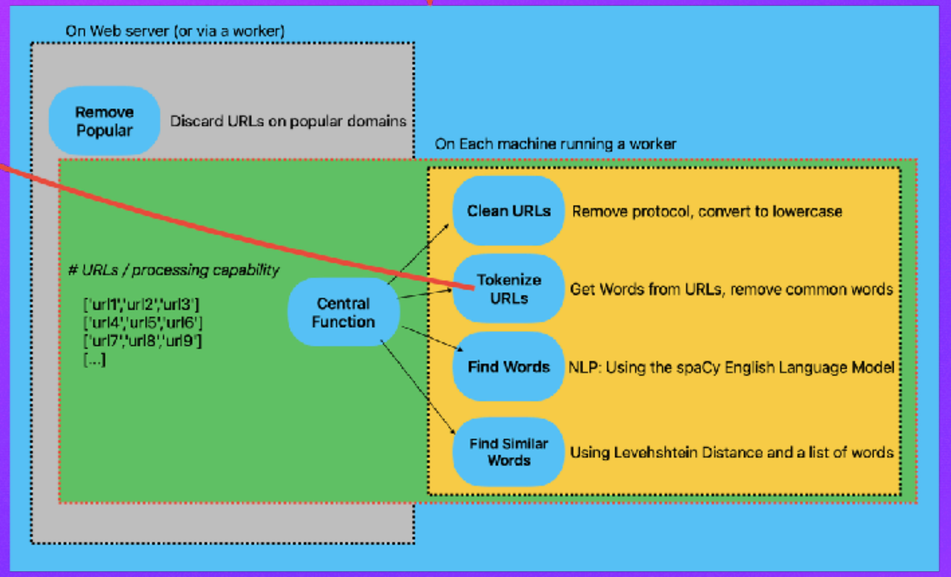
Rabbit MQ Worker

Rabbit MQ Worker

4: Response is Returned.

The architecture using one machine for the web app and other machines with multiprocessing for the functions

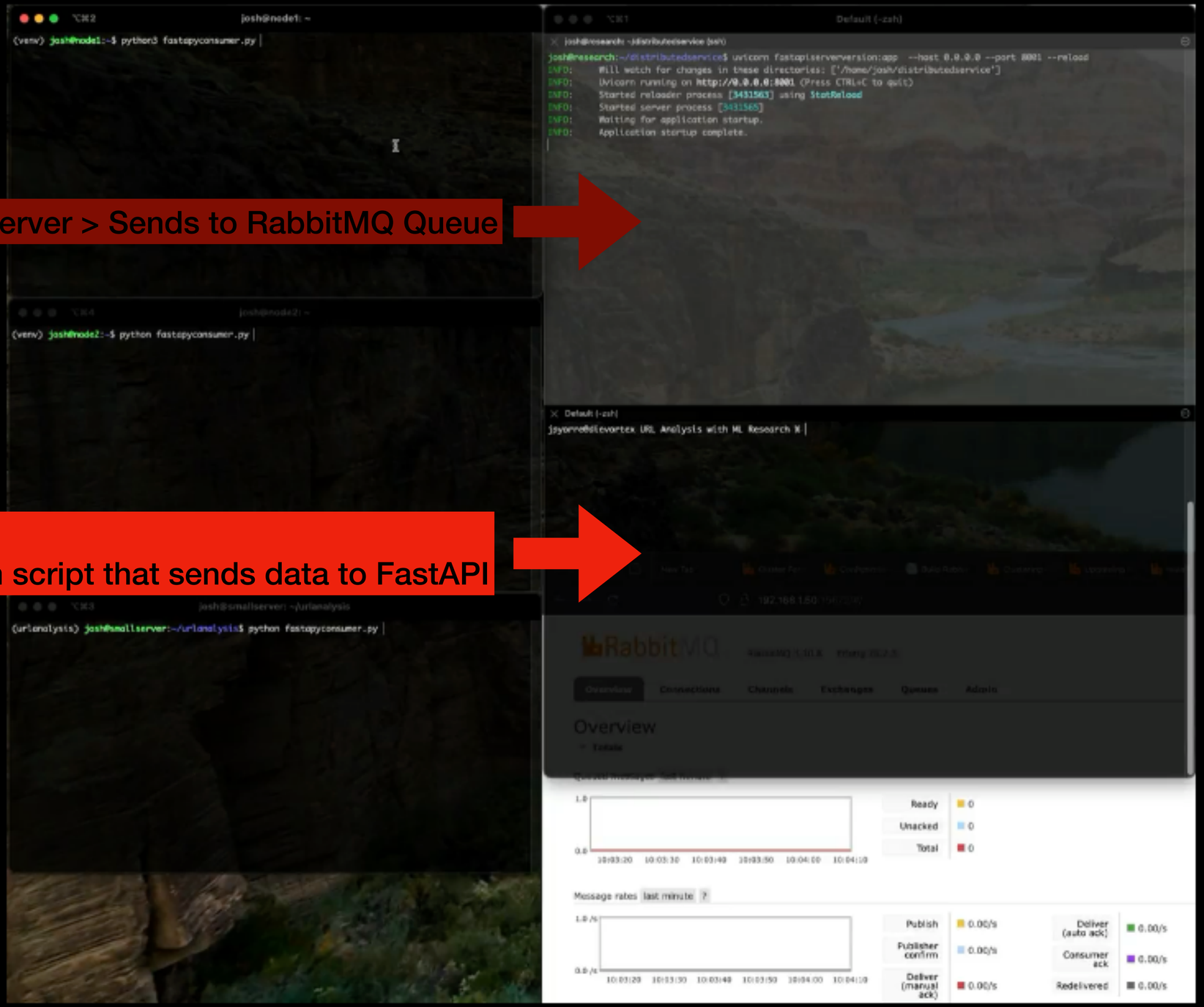
3: Workers run code *(Unless already in mongodb)*

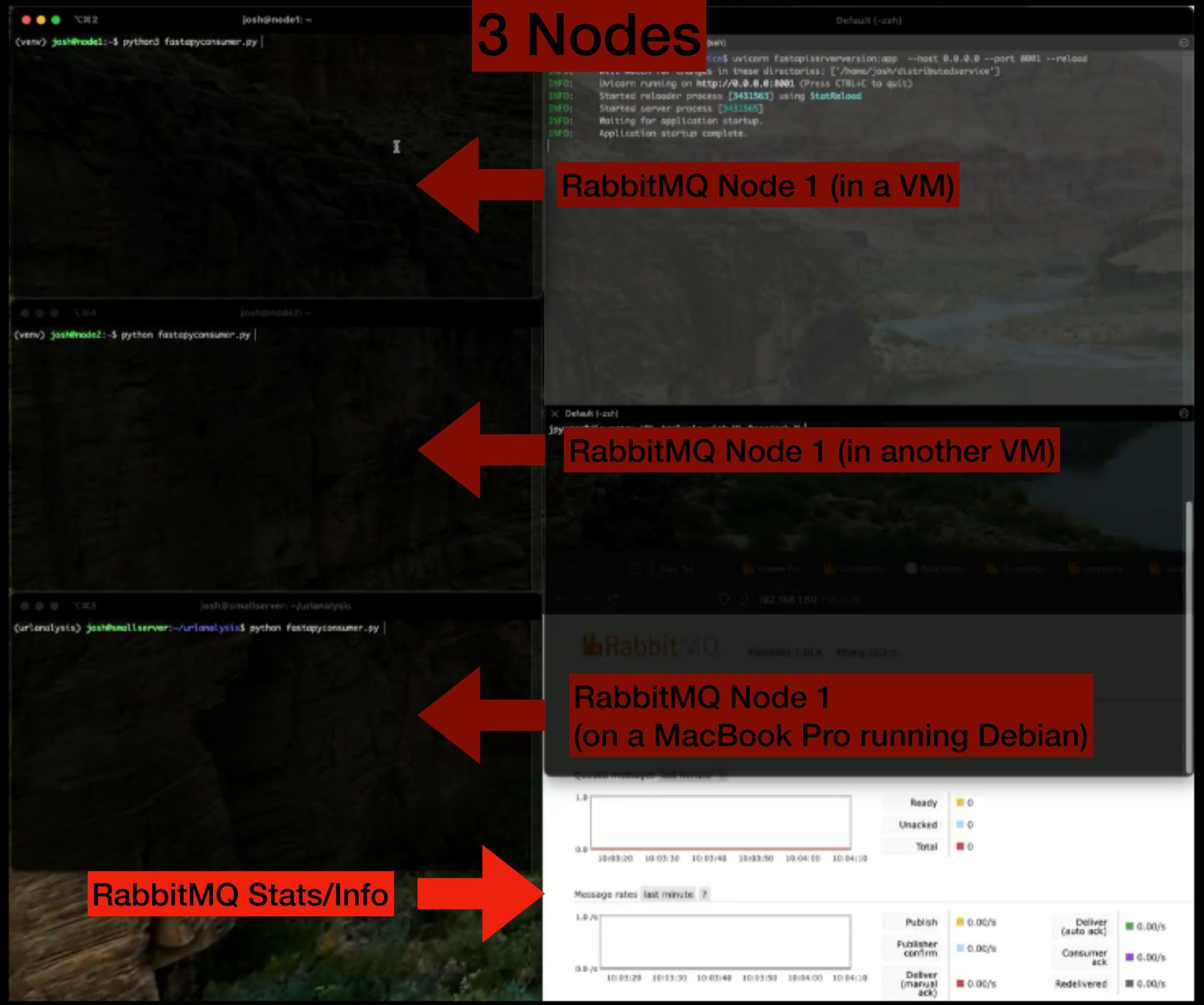


How RabbitMQ Works

Fast API Web Server > Sends to RabbitMQ Queue

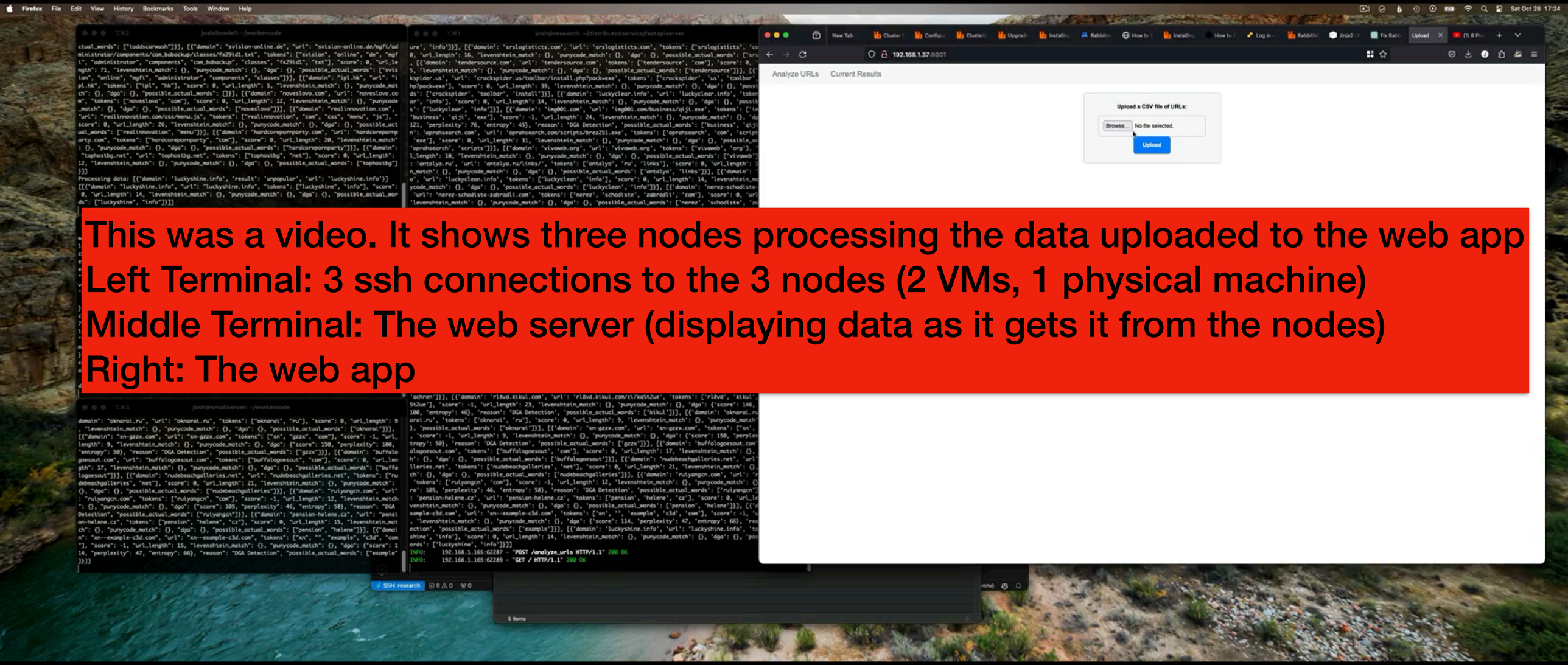
My computer,
Running a python script that sends data to FastAPI




3 Nodes
(venv) josh@node1:~$ python3 fastapyconsumer.py
(venv) josh@node2:~$ python fastapyconsumer.py
(urlanalysis) josh@smallserver:~/urlanalysis$ python fastapyconsumer.py
INFO: Uvicorn running on http://0.0.0.0:8001 (Press CTRL+C to quit)
INFO: Started reloader process [3431563] using StatReload
INFO: Started server process [3431565]
INFO: Waiting for application startup.
INFO: Application startup complete.
RabbitMQ Node 1 (in a VM)
RabbitMQ Node 1 (in another VM)
RabbitMQ Node 1
(on a MacBook Pro running Debian)
RabbitMQ Stats/Info
Ready
Unacked
Total
Message rates
Publish
Publisher confirm
Deliver (manual ack)
Deliver (auto ack)
Consumer ack
Redelivered

How it looks for this Process

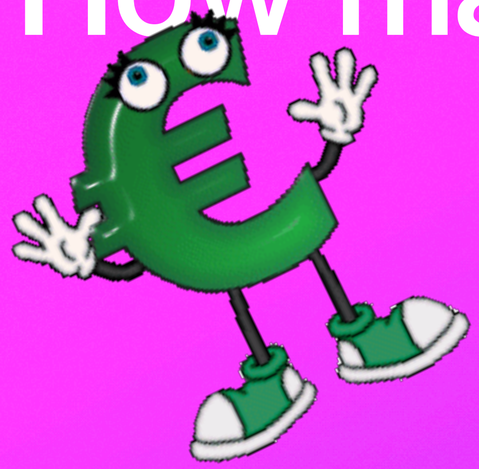
Distributed Computing: 3 nodes, 5000 URLs

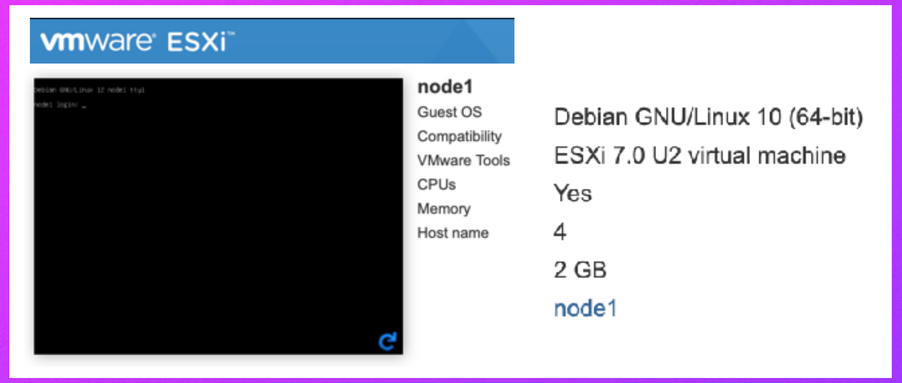


This was a video. It shows three nodes processing the data uploaded to the web app
Left Terminal: 3 ssh connections to the 3 nodes (2 VMs, 1 physical machine)
Middle Terminal: The web server (displaying data as it gets it from the nodes)
Right: The web app

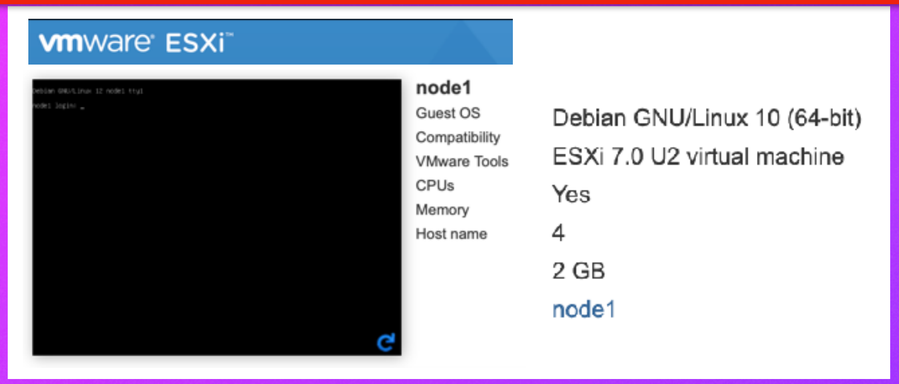
Distributed Service Concerns

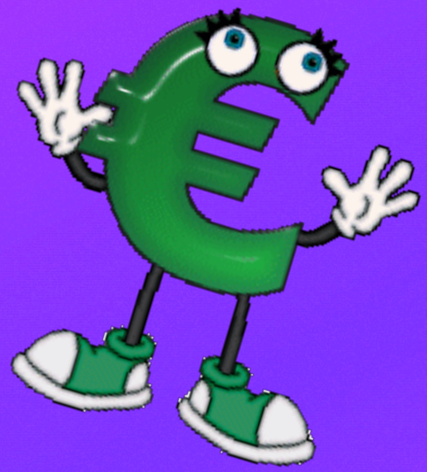
How many machines can you afford?





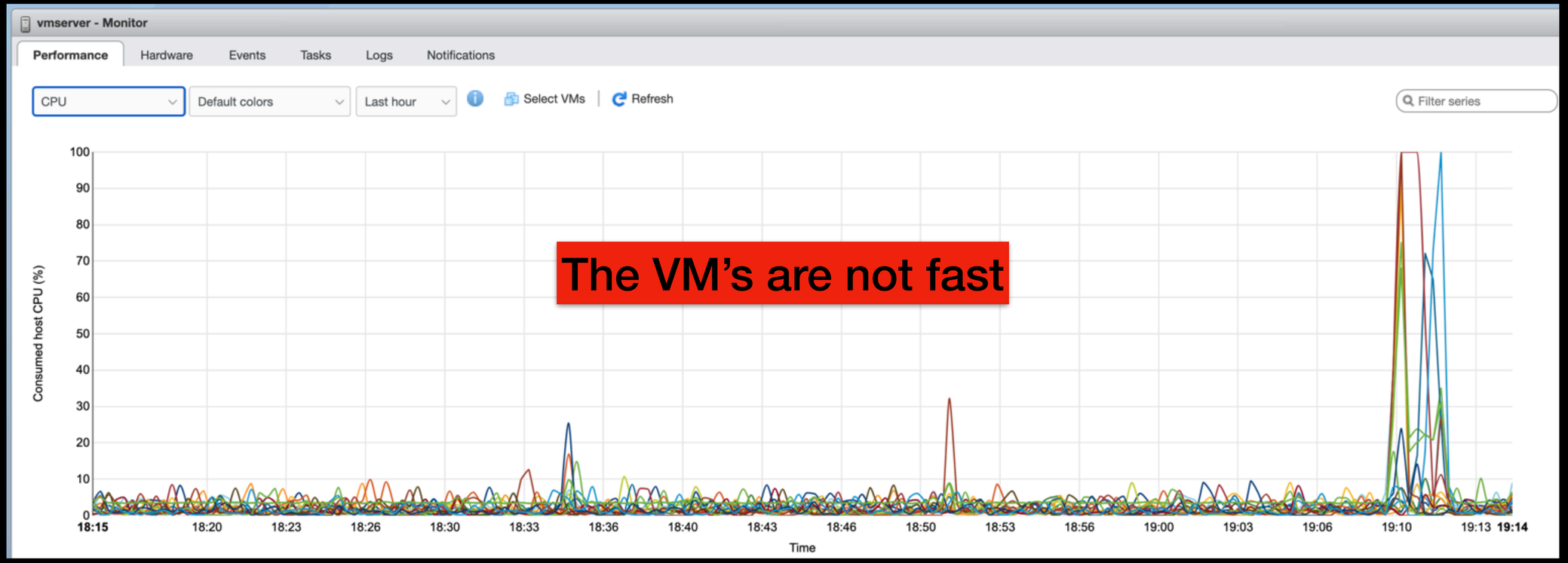
It turns out that my many machines model is only as fast as the machines you are running





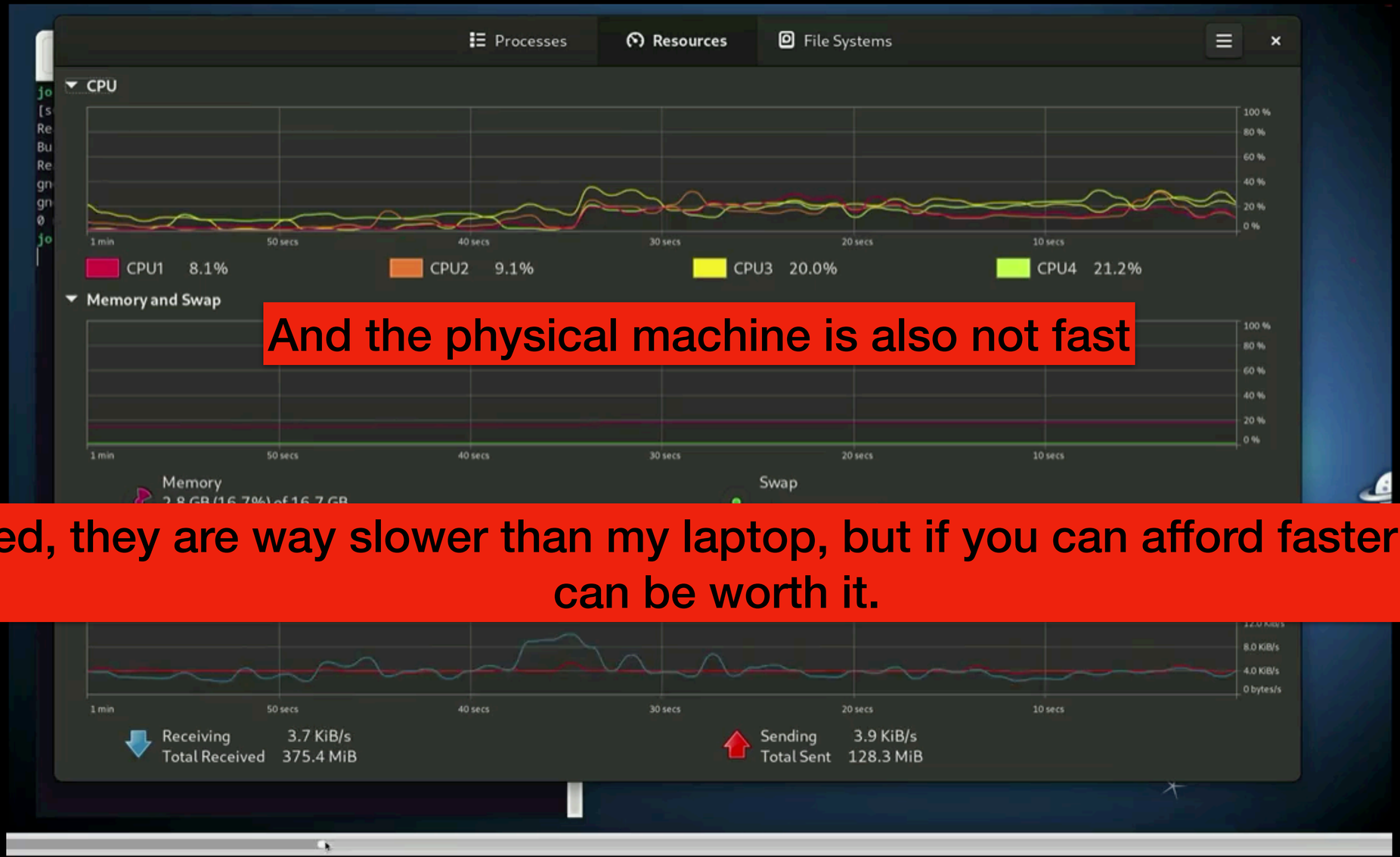
Distributed Computing: 3 nodes, 5000 URLs

Node 1 and 2: CPU Usage of the two VMs on my ESXI Server:



Distributed Computing: 3 nodes, 5000 URLs

Node 3: CPU Usage of the MacBook Pro running Debian:



All combined, they are way slower than my laptop, but if you can afford faster machines, it can be worth it.

The Interface

2.7 KB

100.csv

Execution Time: 16.42 seconds

URL	Domain	Score	Reason	URL Length	Levenshtein Matches	Possible Actual Words
luckysuccess.info	luckysuccess.info	-2	Investigate, Virustotal	16		luckysuccess, info
m2132.ehgaugysd.net/zyso.cgi?18	m2132.ehgaugysd.net	-1	Virustotal	27		ehgaugysd, zyso
associatesexports.com	associatesexports.com	-1	Investigate	20		associatesexports
oprahsearch.com/scripts/net19.exe	oprahsearch.com	-1	Virustotal	29		oprahsearch, scripts
pb-webdesign.net	pb-webdesign.net	-1	Investigate	15		
tophostbg.net	tophostbg.net	-2	Investigate, Virustotal	12		tophostbg
warco.pl	warco.pl	-2	Investigate, Virustotal	7		warco
freeserials.spb.ru/key/68703.htm	freeserials.spb.ru	0		27		freeserials
outporn.com	outporn.com	-2	Investigate, Virustotal	10		outporn
oknarai.ru	oknarai.ru	-2	Investigate, Virustotal	9		oknarai
worldgymperu.com	worldgymperu.com	-2	Investigate, Virustotal	15		worldgymperu
dimsnetwork.com	dimsnetwork.com	-1	Investigate	14		dimsnetwork
vocational-training.us	vocational-training.us	-2	Investigate, Virustotal	21		
sunlux.net/company/about.html	sunlux.net	-1	Virustotal	25		sunlux, company, about
nadegda-95.ru	nadegda-95.ru	-1	Investigate	12		
iamagameaddict.com	iamagameaddict.com	-2	Investigate, Virustotal	17		iamagameaddict

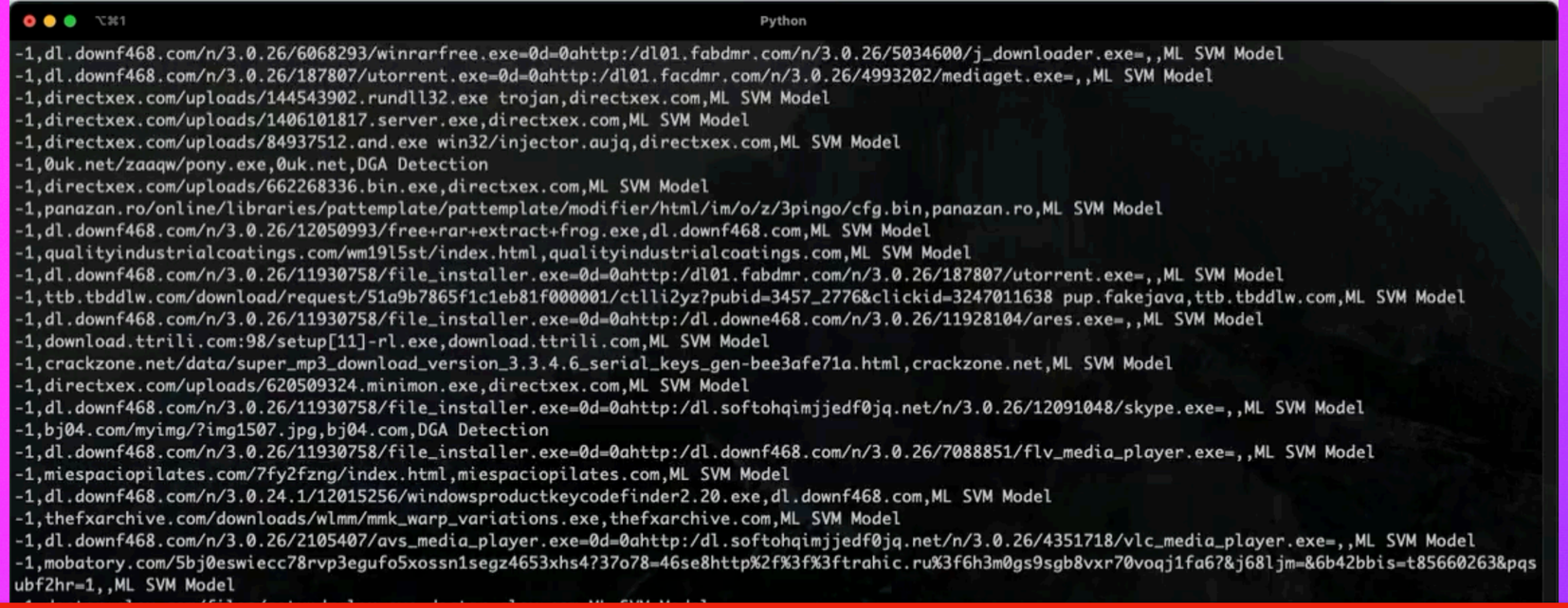
Execution Time: 7.13 seconds

URL	Domain	Score	Reason	URL Length	Levenshtein Matches	Possible Actual Words
testbored.com/testing	testbored.com	0		19		testbored, testing
svision-online.de/mgfi /administrator/components /com_babackup/classes /fx29id1.txt	svision-online.de	-1	ML SVM Model	72		mgfi, administrator, components, classes
xn--example-c3d.com	xn--example-c3d.com	-1	punycode	18		
benkofmaerical.com/benking /wilmart	benkofmaerical.com	-1	levenshtein	31	• benking,banking • wilmart,walmart	benkofmaerical, benking, wilmart
diaryofagameaddict.com	diaryofagameaddict.com	-2	Investigate, Virustotal	21		diaryofagameaddict

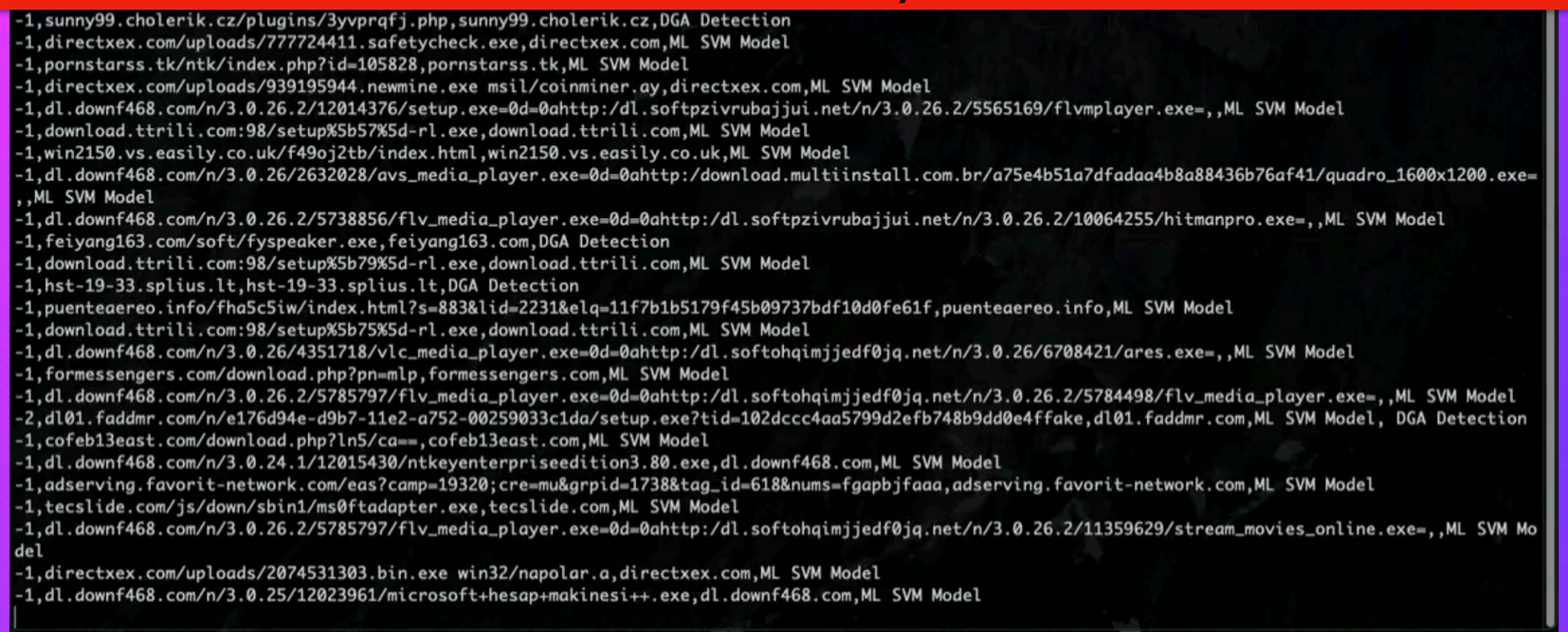
Actions on Streaming URLs

```python
def send_to_api(file_path):
    url = 'http://127.0.0.1:8000/analyze_urls_api'
    with open(file_path, 'rb') as f:
        files = {'file': (file_path, f, 'text/csv')}
        response = requests.post(url, files=files)

    if response.status_code == 200:
        result = response.json()
        filename = datetime.datetime.now().strftime('%Y%m%d') + '.json'
        with open('./{}'.format(filename),'w') as file:
            json.dump(result,file)
        return(result)
    else:
        return(False)

results = send_to_api(file_path)
if results != False:
    # Iterate through all items except the last one - it's always the execution time
    for item in results['url_results'][:-1]:  # Skip the last item
        for result in item:  # Now each item is a list of results, iterate through it
            print(f"Domain: {result['domain']}")
            print(f"URL: {result['url']}")
            print(f"Reason: {result['reason']}")
            print(f" - {word}")

            print("Tokens:")
            for token in result.get('tokens', []):
                print(f" - {token}")

            print("DGA Matches:")
            for key, value in result.get('dga', {}).items():
                print(f" - {key}: {value}")

            print("Levenshtein Matches:")
            for match in result.get('levenshtein_match', []):
                print(f" - {match}")

            print("Punycode Matches:")
            for match in result.get('punycode_match', []):
                print(f" - {match}")

            print("-" * 40)  # Print a divider for readability

    # Print the execution time
    execution_time_info = results['url_results'][-1]
    print(f"Execution Time: {execution_time_info['Execution Time']}")
```

Code demonstrating API use: Sending data from logs to the API and getting back a response that can be used to make a machine determination (block or not block)



Stream Response from the API that can be used to make a machine determination (block or not block)



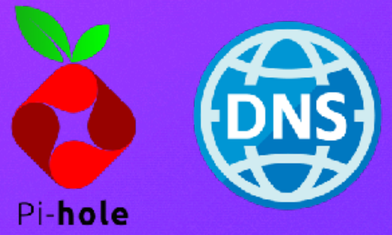
Bad URLs

```
puenteaereo.info/fha5c5iw/index.html?s=883&lid=2231&elq=11f7b1b5179f45b09737bdf10d0fe61f
download.ttrili.com:98/setup%5b75%5d-rl.exe
dl.downf468.com/n/3.0.26/4351718/vlc_media_player.exe=0d=0ahttp:/dl.softohqimjjedf0jq.net/n/3.0.26/6708421/ares.exe=
formessengers.com/download.php?pn=mlp
dl.downf468.com/n/3.0.26.2/5785797/flv_media_player.exe=0d=0ahttp:/dl.softohqimjjedf0jq.net/n/3.0.26.2/5784498/flv_media_player.exe=
dl01.faddmr.com/n/e176d94e-d9b7-11e2-a752-00259033c1da/setup.exe?tid=102dccc4aa5799d2efb748b9dd0e4ffake
cofeb13east.com/download.php?ln5/ca==
dl.downf468.com/n/3.0.24.1/12015430/ntkeyenterpriseedition3.80.exe
adserving.favorit-network.com/eas?camp=19320;cre=mu&grpid=1738&tag_id=618&nums=fgapbjfaaa
tecslide.com/js/down/sbin1/ms0ftadapter.exe
dl.downf468.com/n/3.0.26.2/5785797/flv_media_player.exe=0d=0ahttp:/dl.softohqimjjedf0jq.net/n/3.0.26.2/11359629/stream_movies_online.exe=
directxex.com/uploads/2074531303.bin.exe win32/napolar.a
dl.downf468.com/n/3.0.25/12023961/microsoft+hesap+makinesi++.exe
dl.downf468.com/n/3.0.24.1/12015256/windows+product+key+code+finder+2.20.exe
dl.downf468.com/n/3.0.26/2094912/qvs_media_player.exe=0d=0ahttp:/installsupdater.info/syshost.exe=
```

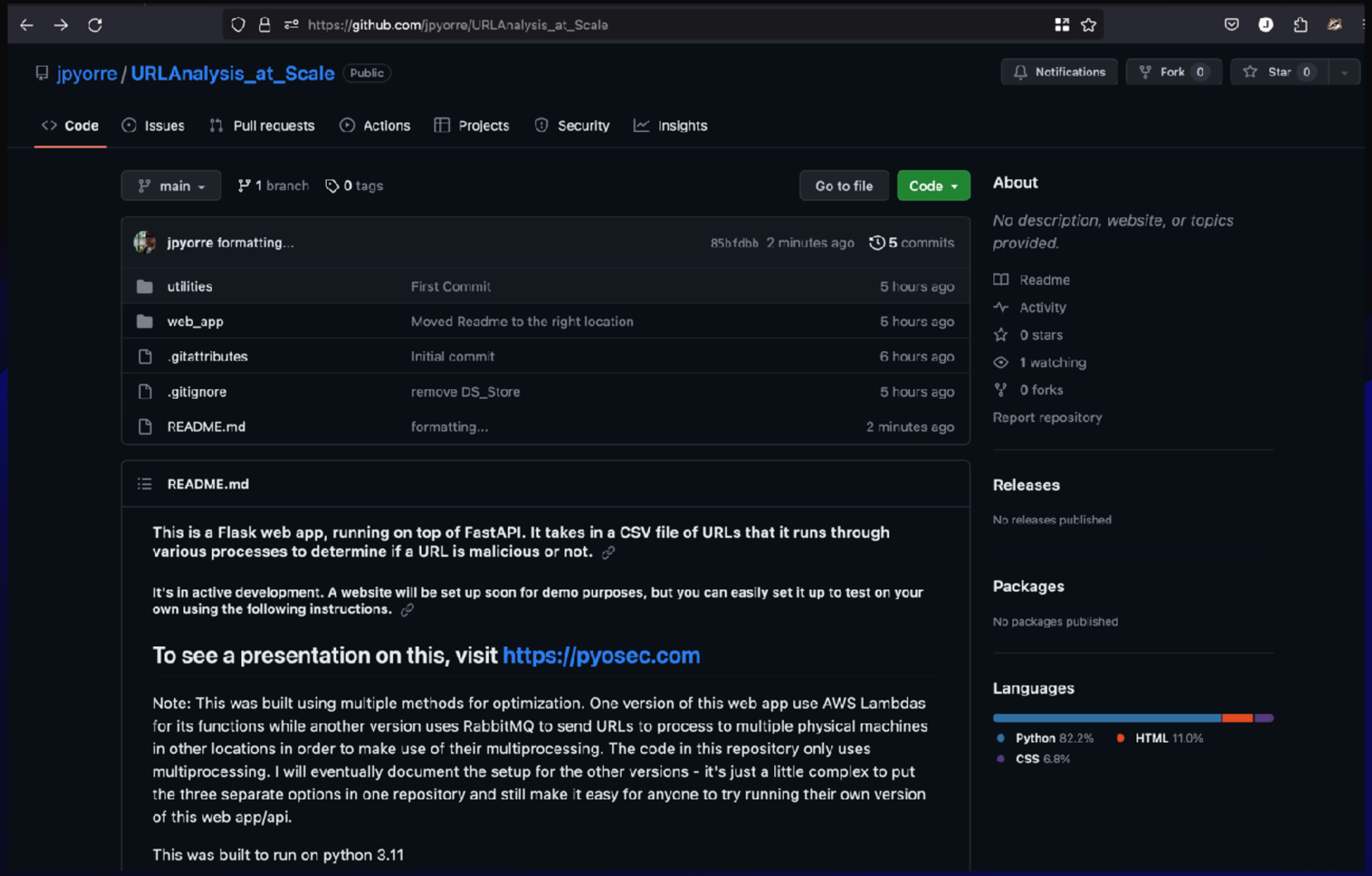
You can then decide to block URLs in a proxy or domains in DNS, or whatever seems appropriate to you

Ba

```
praxisww.com
quinnwealth.com
cofeb13east.com
zyxyfy.com
silurian.cn
ns2ns1.tk
downloaddirect.com
reishus.de
lostartofbeingadame.com
afa15.com.ne.kr
ip-182-50-129-181.ip.secureserver.net
hst-19-33.splius.lt
w4988.nb.host127-0-0-1.com
fgawegwr.chez.com
teameda.comcastbiz.net
ns1.updatesdns.org
win2150.vs.easily.co.uk
qualityindustrialcoatings.com
obkom.net.ua
a.update.51edm.net
zatzy.com
eldiariodeguadalajara.com
```



The Code



https://github.com/jpyorre/URLAnalysis_at_Scale

Thank you!
https://pyosec.com
https://github.com/jpyorre/URLAnalysis_at_Scale