

URL Analysis

At Scale

Josh Pyorre

Security Researcher

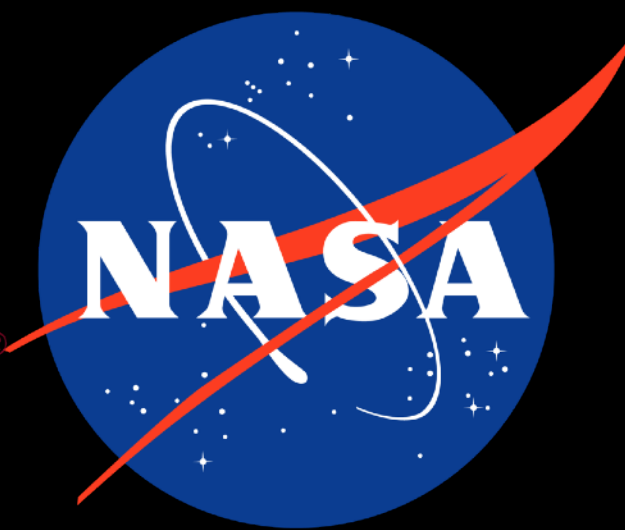


CISCO

TALOS

Senior Security Researcher

Previously:



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

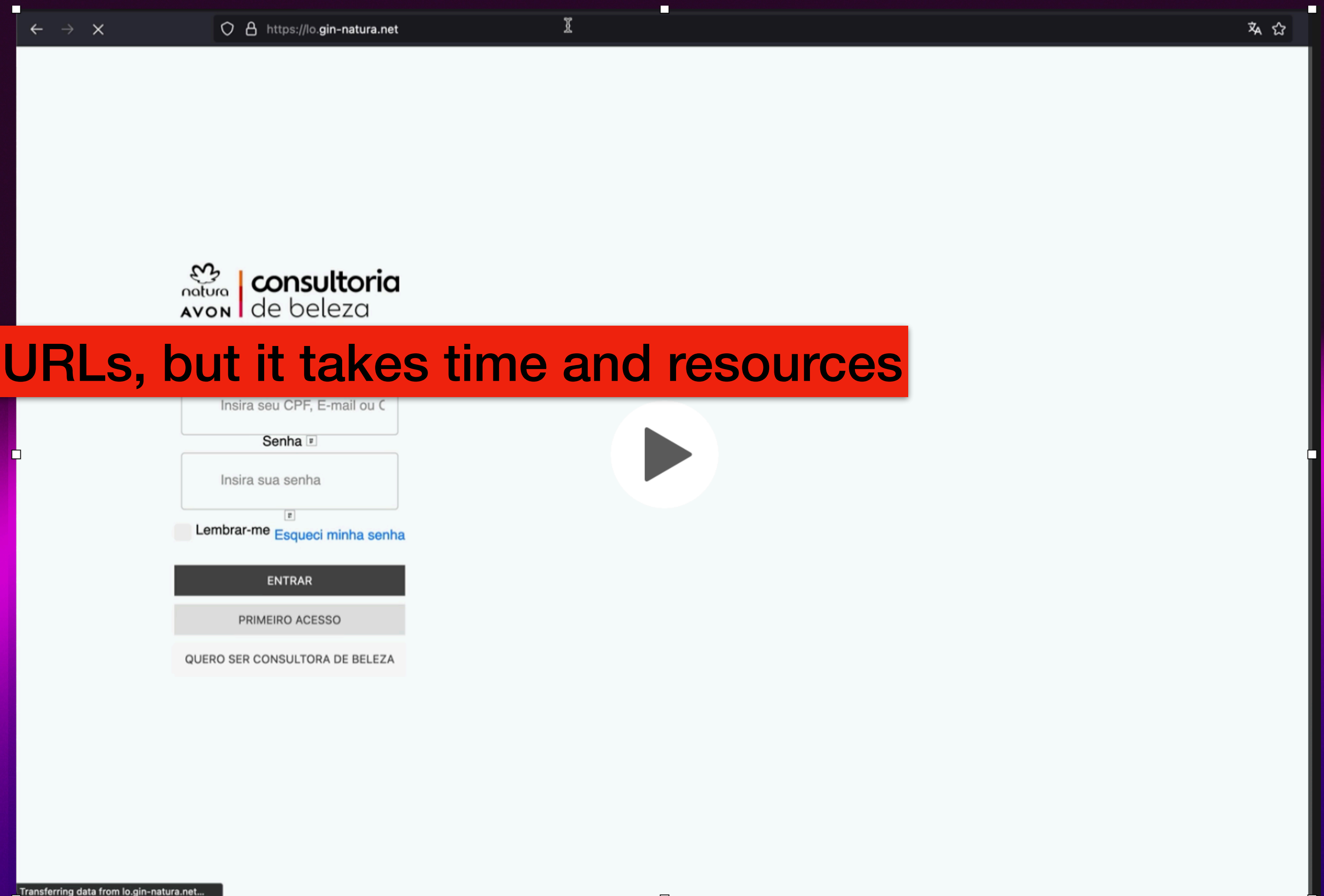
Third Party APIs

Vendor Feeds

Endpoint Detection

Log Analysis

You can crawl URLs, but it takes time and resources



Analyzing URLs

Lots of (*expensive*) options

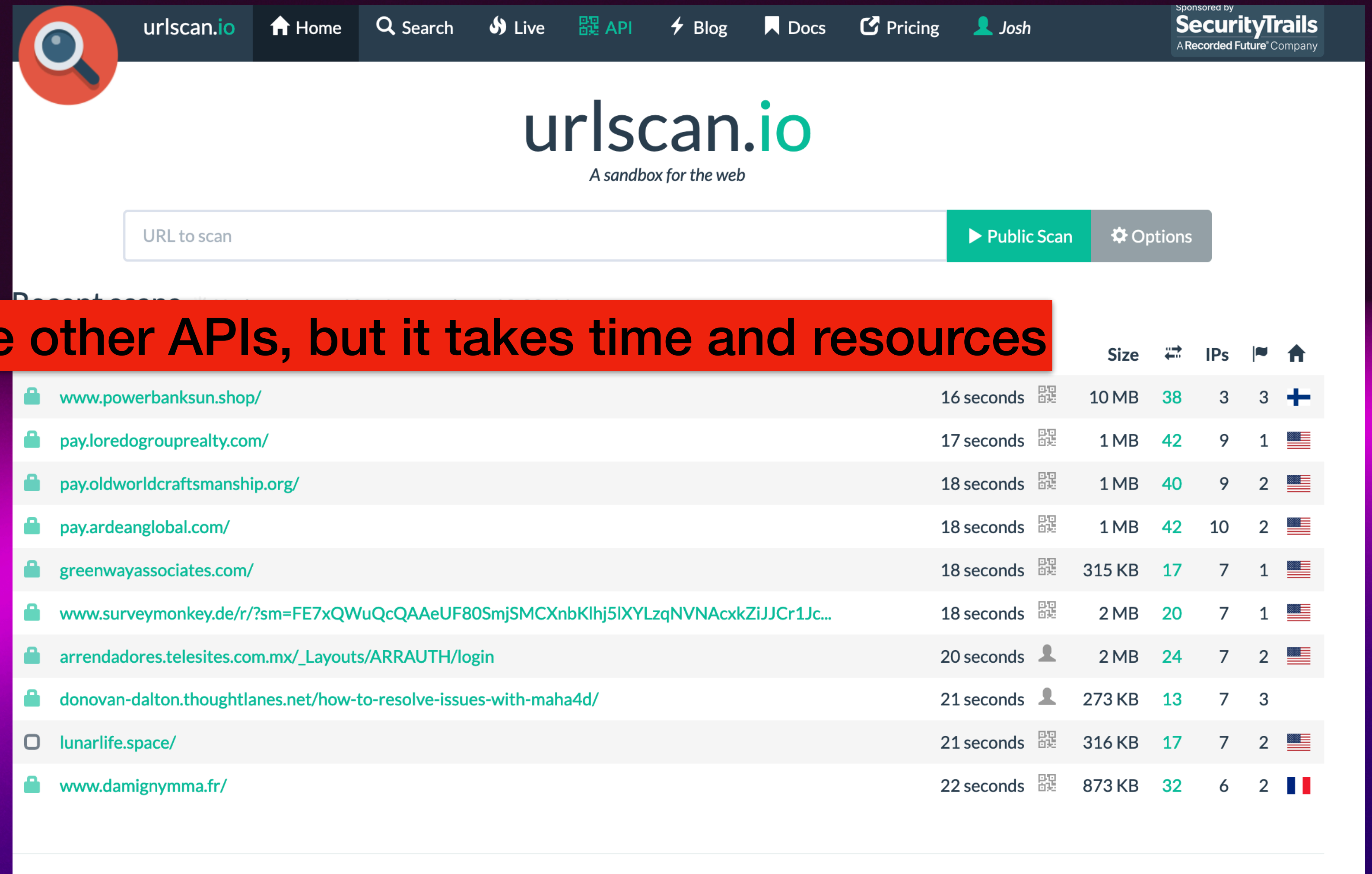
Third Party APIs

Vendor Feeds

Endpoint Detection

Log Analysis

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



The screenshot shows the urlscan.io website interface. At the top, there is a navigation bar with links for Home, Search, Live, API, Blog, Docs, Pricing, and a user profile for Josh. A search bar is prominently displayed with the text "URL to scan" and buttons for "Public Scan" and "Options". Below the search bar, a table lists scan results for various URLs. The table columns include the URL, scan time, size, status (indicated by a lock icon), number of IPs, and a flag icon. A red text box is overlaid on the table, stating "You can use other APIs, but it takes time and resources".

URL	Time	Size	Status	IPs	Flag
www.powerbanksun.shop/	16 seconds	10 MB	38	3	3
pay.loredogrouprealty.com/	17 seconds	1 MB	42	9	1
pay.oldworldcraftsmanship.org/	18 seconds	1 MB	40	9	2
pay.ardeanglobal.com/	18 seconds	1 MB	42	10	2
greenwayassociates.com/	18 seconds	315 KB	17	7	1
www.surveymonkey.de/r/?sm=FE7xQWuQcQAAeUF80SmjSMCXnbKIhj5IXYLzqNVNAcxkZiJJCr1Jc...	18 seconds	2 MB	20	7	1
arrendadores.telesites.com.mx/_Layouts/ARRAUTH/login	20 seconds	2 MB	24	7	2
donovan-dalton.thoughtlanes.net/how-to-resolve-issues-with-maha4d/	21 seconds	273 KB	13	7	3
lunarlife.space/	21 seconds	316 KB	17	7	2
www.damignymma.fr/	22 seconds	873 KB	32	6	2

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

The screenshot shows a web interface for URL analysis. At the top, there is a search bar containing the URL `https://innovapakistan.com/inventoreet/i.exe` and a blue button labeled "INVESTIGATE". Below the search bar, there are three circular icons for "Talos", "Google", and "VirusTotal". A red banner with white text is overlaid on the interface, stating "You can use other APIs, but it takes time and resources". Below the banner, there is a red box with a white "x" icon and the text "Security Categories: Malware". Underneath, the text "DETAILS FOR https://innovapakistan.com/inventoreet/i.exe" is displayed. A section titled "Subdomains" contains a table with the following data:

Name	First Seen	Category
mail.innovapakistan.com	02/17/2021 05:14 AM	Malware
www.innovapakistan.com	02/05/2021 02:23 AM	Malware

At the bottom right of the table, there is a pagination indicator "1 - 2 of 2" with left and right arrow symbols.

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.
3. **Strike Falcon** Intelligence service as
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

EDR takes resources and maintenance

The screenshot shows a Cisco website page with the following elements:

- Header:** Cisco logo, navigation links for Products and Services, Solutions, Support, and Learn. A 'How to Buy' link and a 'Partners' link are also visible in the top right corner.
- Hero Section:** A dark blue banner with the text 'Cisco Secure Endpoint garners industry recognition.' and a 'Read surveys' button.
- Breadcrumbs:** Products & Services / Security / Endpoint Security /
- Title:** 'What Is Endpoint Detection and Response (EDR)?'
- Image:** A photograph of a man with glasses working at a computer in an office setting.
- Text:** A paragraph explaining that endpoint detection and response (EDR) solutions detect threats across your environment, investigate the entire lifecycle of the threat, and provide insights into what happened, how it got in, where it has been, what it's doing now, and what to do about it. It states that by containing the threat at the endpoint, EDR helps eliminate the threat before it can spread.
- Buttons:** Two buttons are present: 'EDR deployment (2:18)' and 'EDR best practices'.
- Footer:** Navigation links for 'Why Use EDR?', 'EDR Capabilities', and 'Related Topics'. A 'Contact Cisco' button is also visible.

Analyzing URLs

Lots of (*expensive*) options

Log Analysis

Log Analysis requires smart log processing and/or people

```
Nov 8 16:41:03 dnsmasq[4126]: query[HTTPS] configuration.ls.apple.com from 192.168.1.47
Nov 8 16:41:03 dnsmasq[4126]: forwarded configuration.ls.apple.com to 208.67.220.220
Nov 8 16:41:03 dnsmasq[4126]: query[A] configuration.ls.apple.com from 192.168.1.47
Nov 8 16:41:03 dnsmasq[4126]: forwarded configuration.ls.apple.com to 208.67.220.220
Nov 8 16:41:03 dnsmasq[4126]: validation result is INSECURE
Nov 8 16:41:03 dnsmasq[4126]: reply configuration.ls.apple.com is <CNAME>
Nov 8 16:41:03 dnsmasq[4126]: reply gspe11-ssl.ls.apple.com.edgekey.net is <CNAME>
Nov 8 16:41:03 dnsmasq[4126]: reply e10499.dsce9.akamaiedge.net is NODATA
Nov 8 16:41:04 dnsmasq[4126]: validation result is INSECURE
Nov 8 16:41:04 dnsmasq[4126]: reply configuration.ls.apple.com is <CNAME>
Nov 8 16:41:04 dnsmasq[4126]: reply gspe11-ssl.ls.apple.com.edgekey.net is <CNAME>
Nov 8 16:41:04 dnsmasq[4126]: reply e10499.dsce9.akamaiedge.net is 23.195.33.175
Nov 8 16:41:04 dnsmasq[4126]: query[HTTPS] e10499.dsce9.akamaiedge.net from 192.168.1.47
Nov 8 16:41:04 dnsmasq[4126]: forwarded e10499.dsce9.akamaiedge.net to 208.67.220.220
Nov 8 16:41:04 dnsmasq[4126]: query[A] e10499.dsce9.akamaiedge.net from 192.168.1.47
Nov 8 16:41:04 dnsmasq[4126]: cached e10499.dsce9.akamaiedge.net is 23.195.33.175
Nov 8 16:41:04 dnsmasq[4126]: validation result is INSECURE
Nov 8 16:41:04 dnsmasq[4126]: reply e10499.dsce9.akamaiedge.net is NODATA
Nov 8 16:41:11 dnsmasq[4126]: query[A] mobile-collector.newrelic.com from 192.168.1.173
Nov 8 16:41:11 dnsmasq[4126]: gravity blocked mobile-collector.newrelic.com is 0.0.0.0
Nov 8 16:41:13 dnsmasq[4126]: query[A] mobile-collector.newrelic.com from 192.168.1.173
Nov 8 16:41:13 dnsmasq[4126]: gravity blocked mobile-collector.newrelic.com is 0.0.0.0
Nov 8 16:41:16 dnsmasq[4126]: query[HTTPS] cws.telestream.net from 192.168.1.254
Nov 8 16:41:16 dnsmasq[4126]: forwarded cws.telestream.net to 208.67.220.220
Nov 8 16:41:16 dnsmasq[4126]: query[A] cws.telestream.net from 192.168.1.254
Nov 8 16:41:16 dnsmasq[4126]: forwarded cws.telestream.net to 208.67.220.220
Nov 8 16:41:16 dnsmasq[4126]: dnssec-query[DS] telestream.net to 208.67.220.220
Nov 8 16:41:16 dnsmasq[4126]: reply telestream.net is no DS
Nov 8 16:41:16 dnsmasq[4126]: reply a9d05e548a53a4ef0.awsglobalaccelerator.com is NODATA
Nov 8 16:41:16 dnsmasq[4126]: query[HTTPS] a9d05e548a53a4ef0.awsglobalaccelerator.com from 192.168.1.254
Nov 8 16:41:16 dnsmasq[4126]: forwarded a9d05e548a53a4ef0.awsglobalaccelerator.com to 208.67.220.220
Nov 8 16:41:16 dnsmasq[4126]: reply awsglobalaccelerator.com is no DS
Nov 8 16:41:16 dnsmasq[4126]: validation result is INSECURE
Nov 8 16:41:16 dnsmasq[4126]: reply a9d05e548a53a4ef0.awsglobalaccelerator.com is NODATA
Nov 8 16:41:16 dnsmasq[4126]: validation result is INSECURE
Nov 8 16:41:16 dnsmasq[4126]: reply cws.telestream.net is <CNAME>
Nov 8 16:41:16 dnsmasq[4126]: reply a9d05e548a53a4ef0.awsglobalaccelerator.com is 35.71.163.70
Nov 8 16:41:16 dnsmasq[4126]: reply a9d05e548a53a4ef0.awsglobalaccelerator.com is 52.223.16.119
Nov 8 16:41:24 dnsmasq[4126]: query[A] my.1password.com from 192.168.1.254
Nov 8 16:41:24 dnsmasq[4126]: forwarded my.1password.com to 208.67.220.220
Nov 8 16:41:24 dnsmasq[4126]: validation result is INSECURE
Nov 8 16:41:24 dnsmasq[4126]: reply my.1password.com is 3.93.228.98
Nov 8 16:41:24 dnsmasq[4126]: reply my.1password.com is 3.225.245.222
Nov 8 16:41:24 dnsmasq[4126]: reply my.1password.com is 52.55.125.33
Nov 8 16:41:24 dnsmasq[4126]: reply my.1password.com is 54.88.248.197
Nov 8 16:41:24 dnsmasq[4126]: reply my.1password.com is 54.163.166.138
Nov 8 16:41:25 dnsmasq[4126]: query[A] api.segment.io from 192.168.1.173
Nov 8 16:41:25 dnsmasq[4126]: gravity blocked api.segment.io is 0.0.0.0
Nov 8 16:41:25 dnsmasq[4126]: query[A] api.segment.io from 192.168.1.173
Nov 8 16:41:25 dnsmasq[4126]: gravity blocked api.segment.io is 0.0.0.0
Nov 8 16:41:26 dnsmasq[4126]: query[A] worklaptop from 192.168.1.79
Nov 8 16:41:26 dnsmasq[4126]: /etc/hosts worklaptop is 192.168.1.79
Nov 8 16:41:29 dnsmasq[4126]: query[A] time.g.aaplimg.com from 192.168.1.79
Nov 8 16:41:29 dnsmasq[4126]: cached time.g.aaplimg.com is 17.253.4.253
Nov 8 16:41:29 dnsmasq[4126]: cached time.g.aaplimg.com is 17.253.16.253
Nov 8 16:41:29 dnsmasq[4126]: cached time.g.aaplimg.com is 17.253.4.125
```

Collecting URLs

From your environment

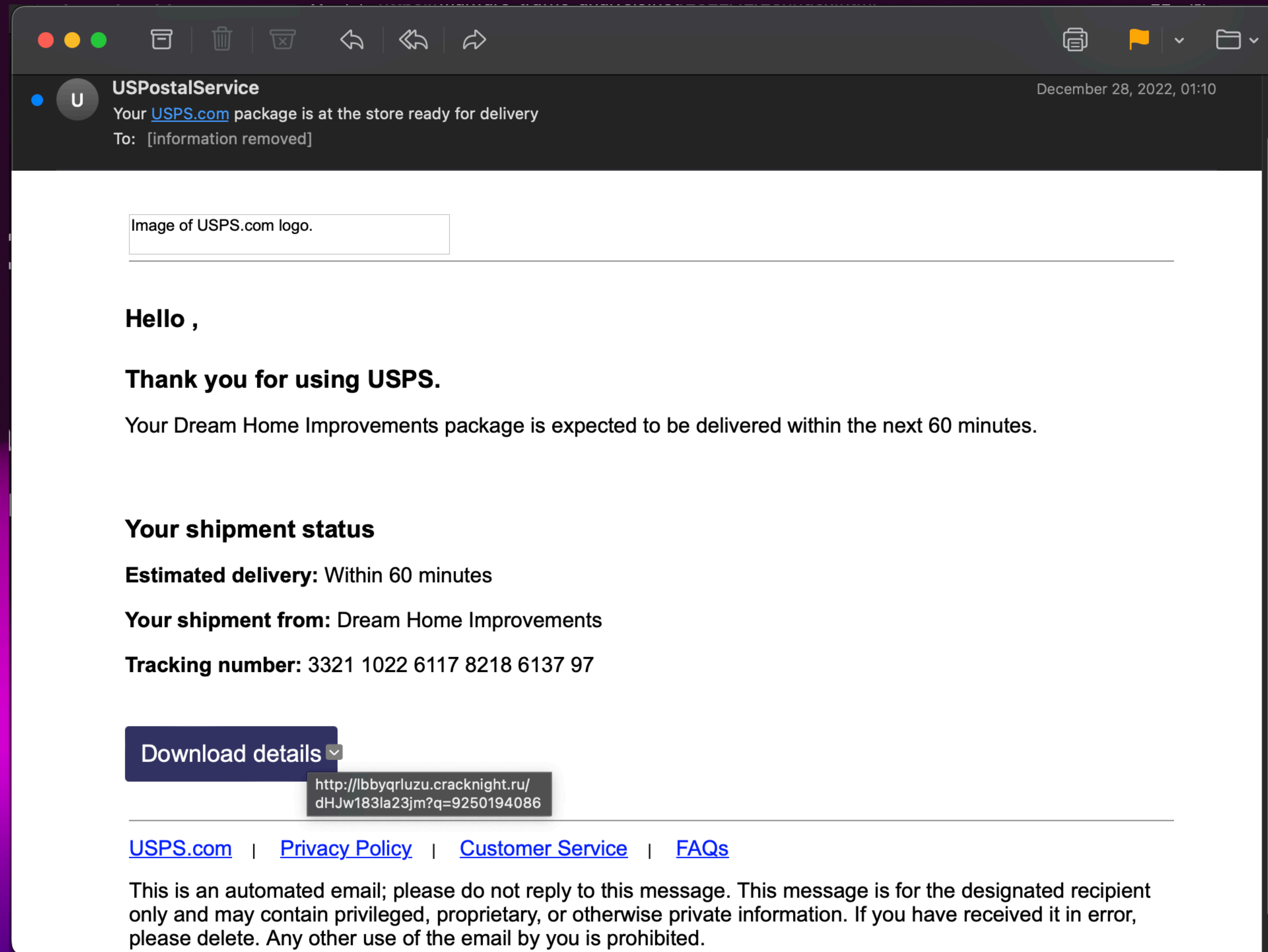
- Logs
- Checking if it was clicked (DNS responses)

Getting data to work with:
From proxy or DNS logs

URLs: I will try to trust you...

The goal is try to trust URLs until you can't

What is a bad URL?



acking number: 3321 1022 6117 8218 6137 97

URL in email doesn't look right...

Download details

[http://lbbyqrluzu.cracknight.ru/
dHJw183la23jm?q=9250194086](http://lbbyqrluzu.cracknight.ru/dHJw183la23jm?q=9250194086)

[SPS.com](#) | [Privacy Policy](#) | [Customer Service](#) | [FA](#)

This is an automated email; please do not reply to this message. This message may contain privileged, proprietary, or otherwise private information. Any other use of the email by you is prohibited.

DEEPSEC



Community Score

15 security vendors flagged this URL as malicious

Reanalyze Search Graph API

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

Status
200

Last Analysis Date
8 months ago



text/html

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
	Malicious	Sophos	Malware

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

I can do this

HANG IN THERE



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

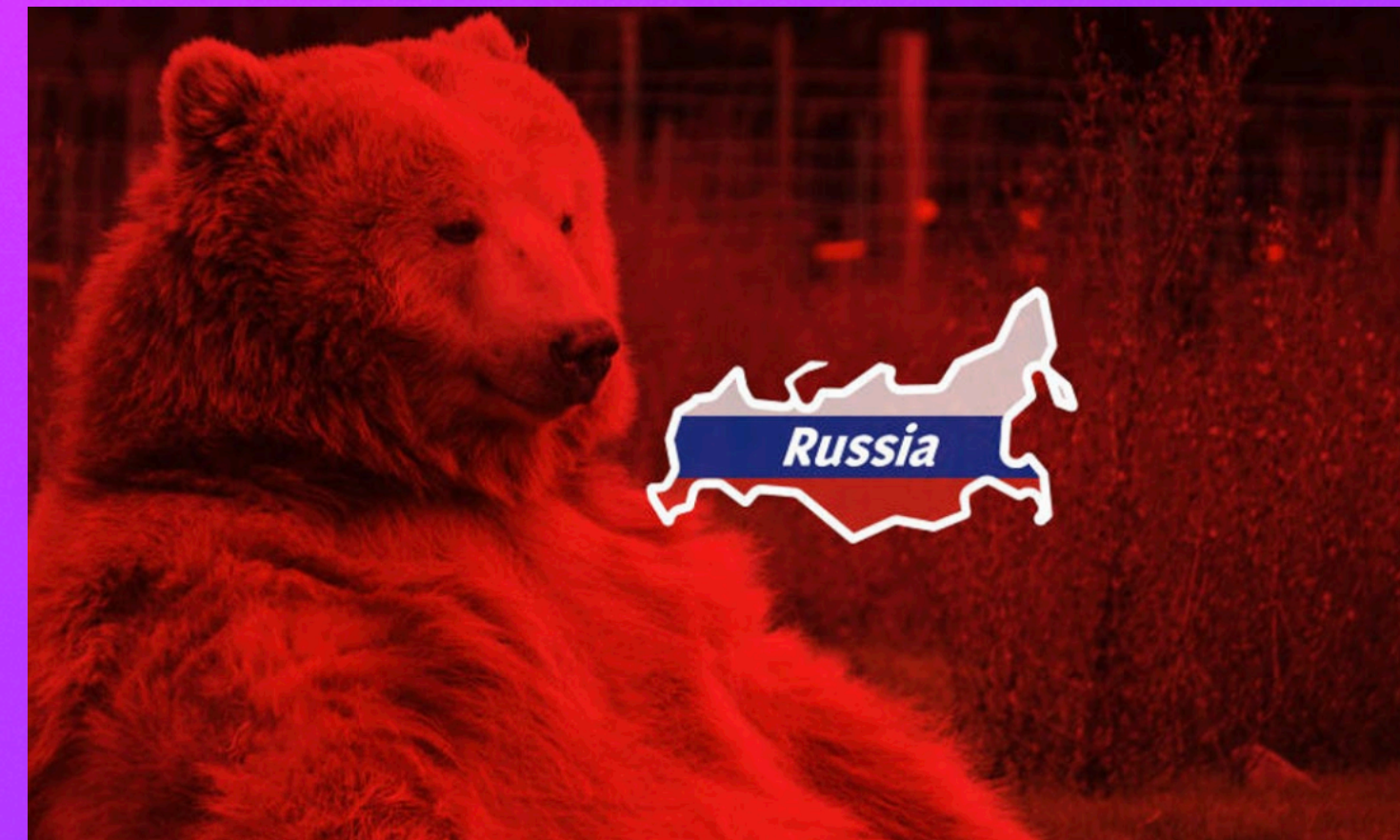
http lbbyqrluzu

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



Umbrella Popularity List

The popularity list contains our most queried domains based on passive DNS usage across our Umbrella global network of more than 100 Billion requests per day with 65 million unique active users, in more than 165 countries. Unlike Alexa, the metric is not based on only browser based 'http' requests from users but rather takes in to account the number of unique client IPs invoking this domain relative to the sum of all requests to all domains. In other words, our popularity ranking reflects the domain's relative internet activity agnostic to the invocation protocols and applications where as 'site ranking' models (such as Alexa) focus on the web activity over port 80 mainly from browsers.

As for Alexa, the site's rank is based on combined measure of unique visitors (Alexa users who visit the site per day) and page views (total URL requests from Alexa users for a site). Umbrella popularity lists are generated on a daily basis reflecting the actual world-wide usage of domains by Umbrella global network users and includes root domains, subdomains in addition to TLDs (Alexa list has only this). In addition, Umbrella popularity algorithm also applies data normalization methodologies to smoothen potential biases that may occur in the data due to sampling of the DNS usage data.

Top 1 million

<http://s3-us-west-1.amazonaws.com/umbrella-static/top-1m.csv.zip>

Top TLDs

<http://s3-us-west-1.amazonaws.com/umbrella-static/top-1m-TLD.csv.zip>

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

The Majestic Million

The million domains we find with the most referring subnets

Free search and download of the top million websites, from majestic.com.

Search

example1.com,example2.com

SEARCH

Type

Majestic Million

Filter

Select TLD

Export

CSV
(~80MB)

Position	+/-	Domain	TLD
1	↑ 1	facebook.com	.com
2	↓ 1	google.com	.com

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

Tranco

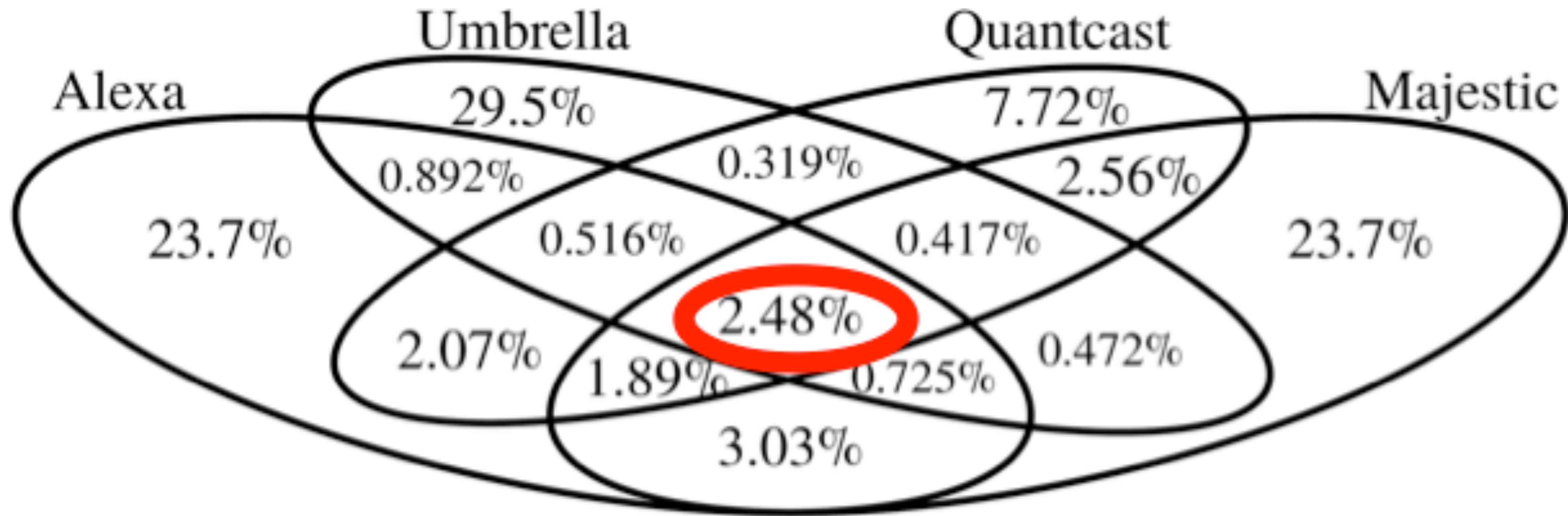
A Research-Oriented Top Sites Ranking Hardened Against Manipulation

By [Victor Le Pochat](#), [Tom Van Goethem](#), [Samaneh Tajalizadehkhoob](#), [Maciej Korczyński](#) and [Wouter Joosen](#)

Download the latest Tranco list

<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

Tokenize & Collect Words

Split URLs, create dictionaries, set score

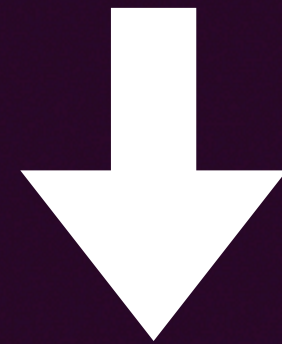
```
{  
  'url': 'benkofmaerical.com/benking',  
  'tokens': ['benkofmaerical', 'com', 'benking'],  
  'score': 0,  
}
```

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

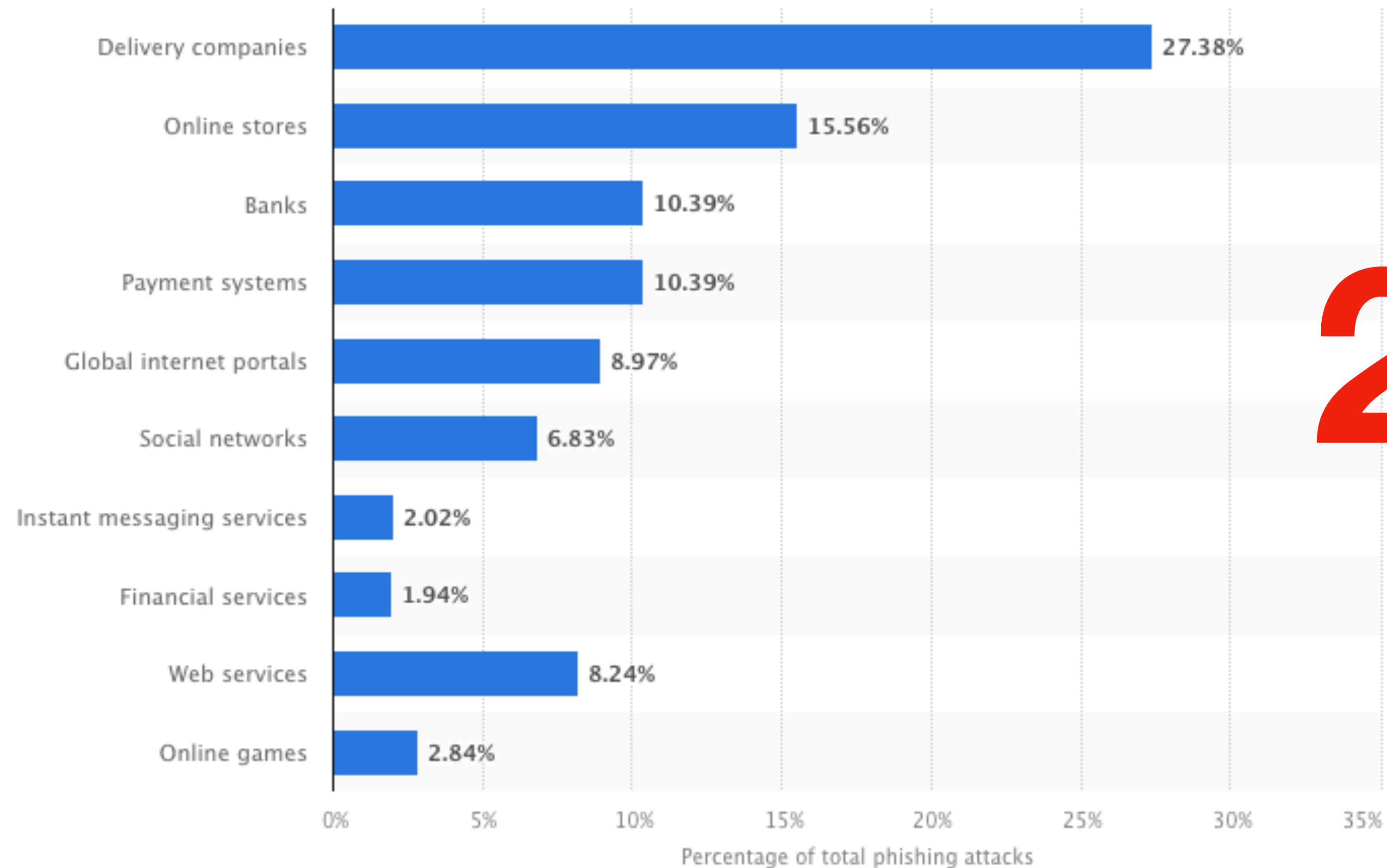
['sikayetvar', 'finans', 'bankacilik', 'hsbc-bank', 'internet-bankaciligi']

superbadsite.com/financez/banking.exe

Find Commonly Spoofed Brands

Brand Components in URLs

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



2021

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/>

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
```

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
link
```

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
link
```

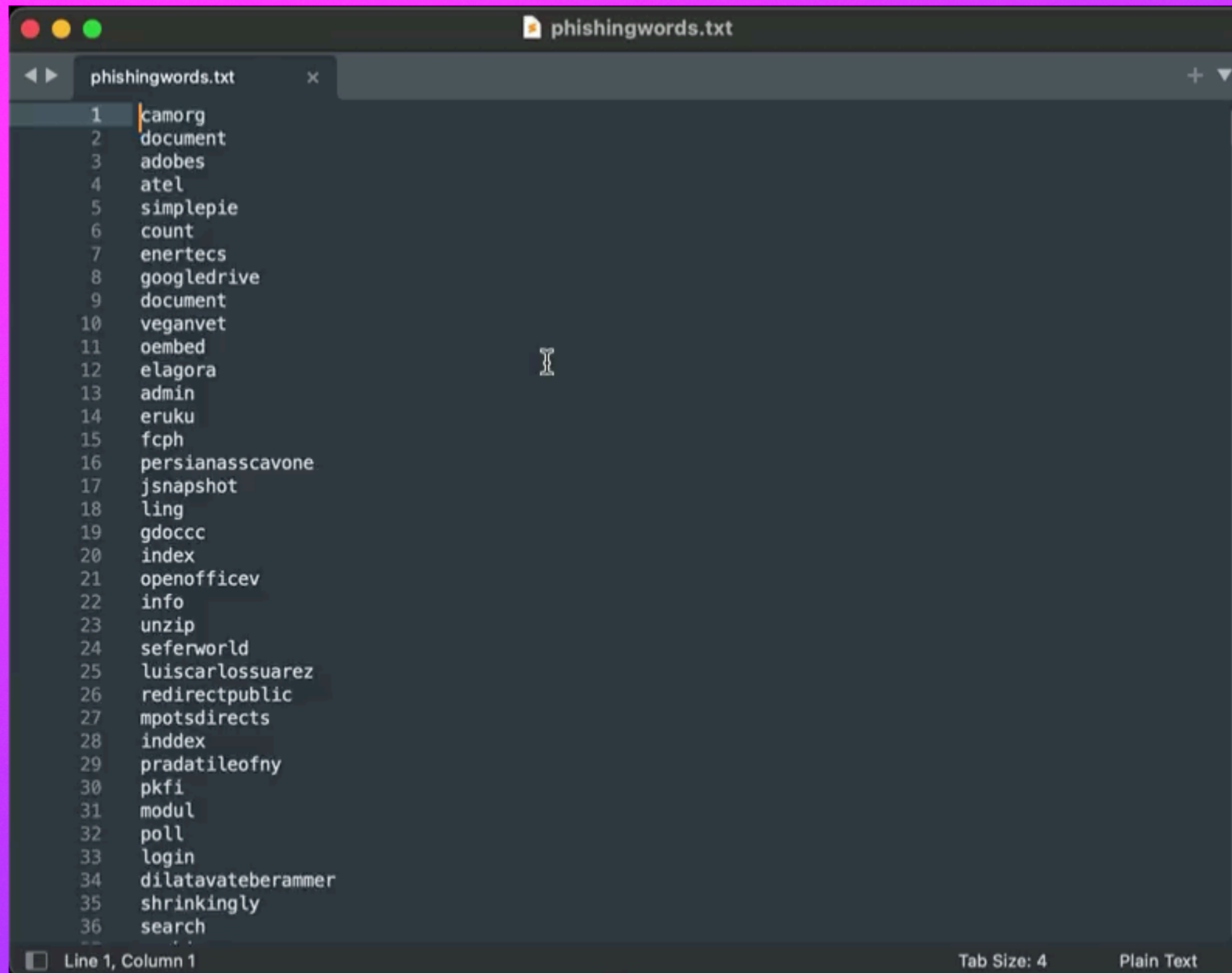

Check Spelling

`pip3 install pyspellchecker`

```
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
```

```
['diaryofagameaddict.com']
['espdesign.com.au']
['iamagameaddict.com']
['kalantzis.net']
['slightlyoffcenter.net']
['toddsicarwash.com']
['tubemoviez.com']
['ipl.hk']
['crackspider.us']
[]
```

Increase the list using known bad URLs



A screenshot of a text editor window titled "phishingwords.txt". The window displays a list of 36 domain names, each on a new line, numbered from 1 to 36. The domains are: camorg, document, adobes, atel, simplepie, count, enertecs, googledrive, document, veganvet, oembed, elagora, admin, eruku, fcph, persianasscavone, jsnapshot, ling, gdoccc, index, openofficev, info, unzip, seferworld, luisarlosuarez, redirectpublic, mpotsdirects, inddex, pradatileofny, pkfi, modul, poll, login, dilatavateberammer, shrinkingly, and search. The cursor is positioned at the end of line 12, "elagora". The status bar at the bottom indicates "Line 1, Column 1", "Tab Size: 4", and "Plain Text".

```
1 camorg
2 document
3 adobes
4 atel
5 simplepie
6 count
7 enertecs
8 googledrive
9 document
10 veganvet
11 oembed
12 elagora
13 admin
14 eruku
15 fcph
16 persianasscavone
17 jsnapshot
18 ling
19 gdoccc
20 index
21 openofficev
22 info
23 unzip
24 seferworld
25 luisarlosuarez
26 redirectpublic
27 mpotsdirects
28 inddex
29 pradatileofny
30 pkfi
31 modul
32 poll
33 login
34 dilatavateberammer
35 shrinkingly
36 search
```

Increase the list using known bad URLs

```
account  
acount  
acpaeqgypt
```

```
actisnet  
activation  
activar
```

```
activatemywebsetup  
activaterequest
```

```
bankacilik  
bankaustria  
bankdanych  
banker  
bankershandbook  
bankguaranteefacts  
banking  
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

With a 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, wells Fargo: wells Fargo  
dogtassomine.com/wellsfargo/wellsfargo/identity.php, wells Fargo: wells Fargo  
security.wellson.wellsfarg0.work/wells8581/wells/onlineSession/kk/, wells Fargo: wells Fargo  
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
```


Increase the list with commonly abused 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!

The screenshot shows a Medium article page. At the top, the URL is <https://medium.com/sfu-cspmp/detecting-malicious-urls-2412091872d6>. The article title is "Detecting Malicious URLs" with a subtitle "A comprehensive guide using ML and NLP Techniques". The author is "Data Dive" with a "Follow" button. It is published in "SFU Professional Computer Science", is a "10 min read", and was published on "Feb 11, 2022". The article has 2.6K claps and 8 comments. Below the article content, there is a section for authors: "Authors: Viddhi Lakhwara, Saurabh Singh, Geethika Choudary". A paragraph follows: "This blog is written and maintained by students in the Master of Science in Professional Computer Science Program at Simon Fraser University as part of their course credit. To learn more about this unique program, please visit sfu.ca/computing/mpcs." The URL <https://medium.com/sfu-cspmp/detecting-malicious-urls-2412091872d6> is repeated at the bottom of the screenshot.

<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
vvic.com	bad
sportsulsan.co.kr/poll/aipi/id.txt	bad

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

Collect Data Using Existing Datasets

PhishTank is operated by [Cisco Talos Intelligence Group](#).

PhishTank® Out of the Net, into the Tank.

username [Register](#) | [Forgot Pas](#)

[Home](#) [Add A Phish](#) [Verify A Phish](#) [Phish Search](#) [Stats](#) [FAQ](#) [Developers](#) [Mailing Lists](#) [My Account](#)

Join the fight against phishing

[Submit](#) suspected phishes. [Track](#) the status of your submissions.
[Verify](#) other users' submissions. [Develop](#) software with our free API.

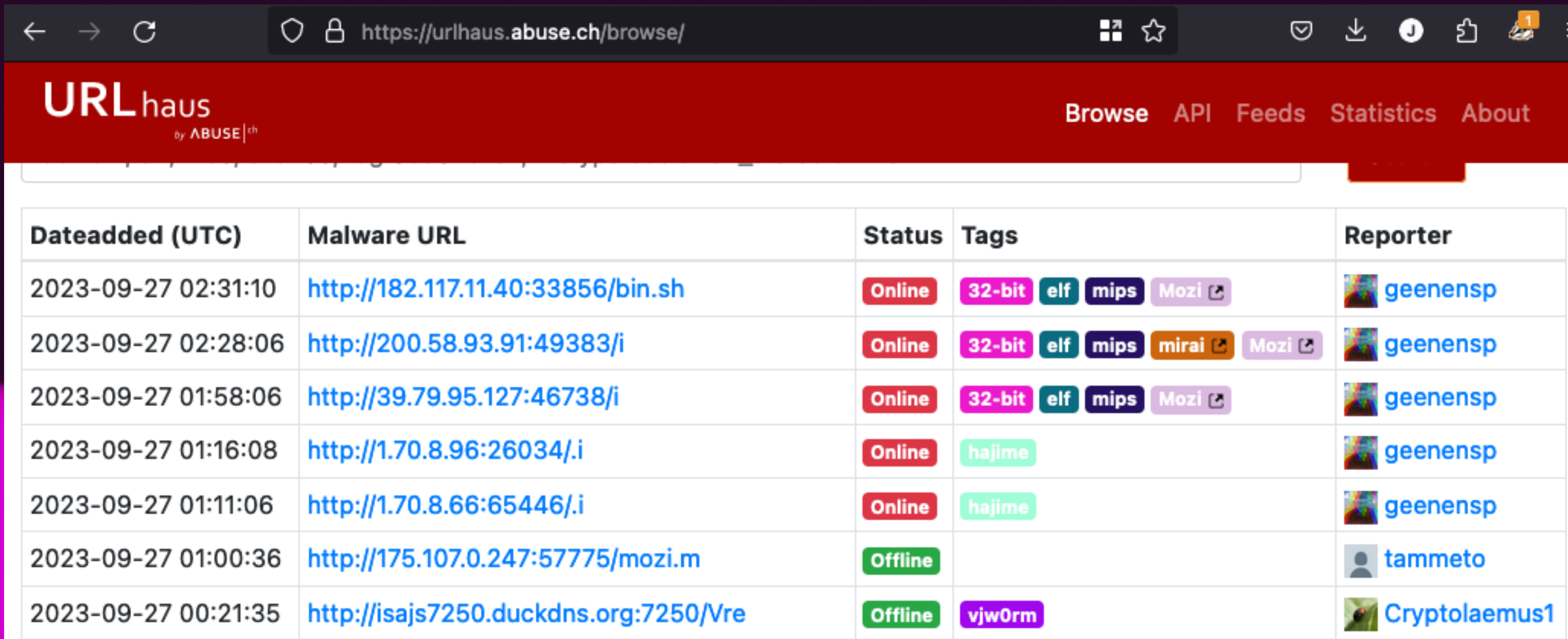
Found a phishing site? Get started now — see if it's in the Tank:

Recent Submissions

You can help! [Sign in](#) or [register](#) (free! fast!) to verify these suspected phishes.

ID	URL	Submitted by
8312893	https://t5x71t6a.square.site/	prodigyabuse
8312892	http://yjjhiyykl.duckdns.org	knack
8312891	https://taplink.cc/tttttswss	prodigyabuse
8312889	https://hook.center	Felix0101
8312888	https://www.digihz.com/	kubotaa
8312887	https://www.limwood.com/	kubotaa
8312886	https://www.zycfgl.com/	kubotaa
8312885	https://www.nndfyh.com/	kubotaa
8312884	https://promonaslojas.online/produto/home.php?prod...	IsmaelParkes
8312883	https://renner-realizesolucoesonline.blogspot.com	MarianMyers

Collect Data Using Existing Datasets



The screenshot shows the URLhaus website interface. The browser address bar displays the URL <https://urlhaus.abuse.ch/browse/>. The website header includes the URLhaus logo and navigation links for Browse, API, Feeds, Statistics, and About. The main content area features a table with the following columns: Dateadded (UTC), Malware URL, Status, Tags, and Reporter. The table lists seven entries, each with a unique URL, status (Online or Offline), associated tags (such as 32-bit, elf, mips, Mozi, mirai, hajime, vjw0rm), and the name of the reporter.

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

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

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>

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	Chip:	Apple M2 Max
Memory	96 GB	Total Number of Cores:	12 (8 performance and 4 efficiency)
Startup disk	Macintosh HD		
Serial number	[REDACTED]		
macOS	Sonoma 14.1		

[More Info...](#)

[Regulatory Certification](#)
™ and © 1983-2023 Apple Inc.
All Rights Reserved.

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&inbox=xclusiv-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: rd1bd.net/dropbox/us-mg5.mail.yahoo.com/pass.php?neo.launch?.rand=1qh  
ua0f7o2jut  
TOKENS: ['rd1bd', 'net', 'dropbox', 'us-mg5', 'mail', 'yahoo', 'com', 'pas  
s', 'php?neo', 'launch?', 'rand=1qhua0f7o2jut']  
POSSIBLE WORDS: ['rd1bd', '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: rd1bd.net/dropbox/us-mg5.mail.yahoo.com/pass.php?neo.launch?.rand=1qh  
ua0f7o2jut  
TOKENS: ['rd1bd', 'net', 'dropbox', 'us-mg5', 'mail', 'yahoo', 'com', 'pas  
s', 'php?neo', 'launch?', 'rand=1qhua0f7o2jut']  
POSSIBLE WORDS: ['rd1bd', 'dropbox', 'mail', 'yahoo', 'pass']
```

Saved 5 seconds!

Process Name	Resident Memory Size	CPU Time	Threads	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_VJOXI
kswarellc', 'com']
POSSIBLE WORDS: ['soloseg', 'products']
```

```
URL: gooflashcorp.com/demo/k231/y/yh/cameo.php?continue=to&inbox=Xclusiv-3Dl&login=
TOKENS: ['gooflashcorp', 'com', 'demo', 'k231', 'y', 'yh', 'cameo', 'php?continue=to&in
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	2.60	4	0

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!

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

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

```
Total Number of URLs: 4,999
Total Number of Words: 31,543
```

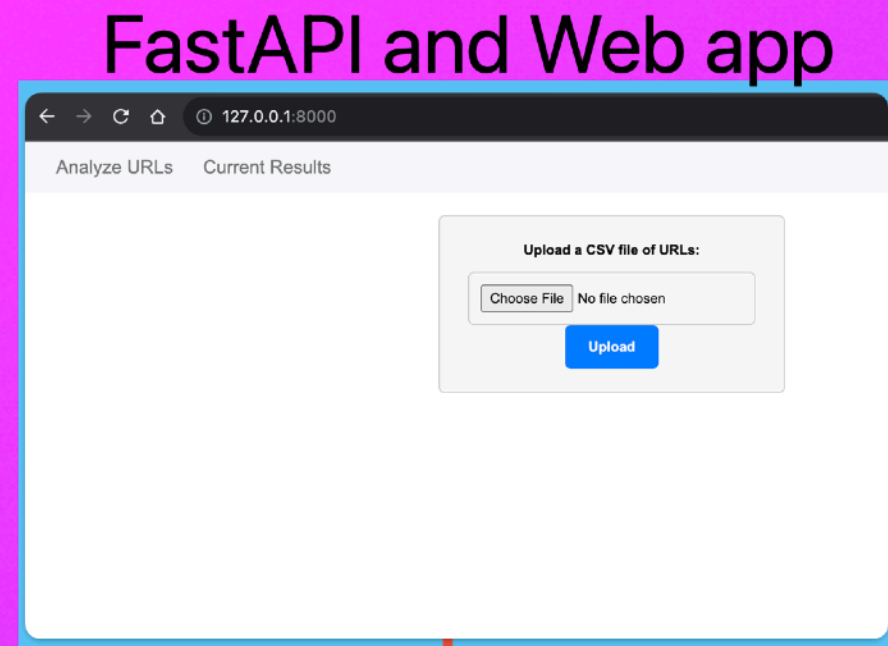
```
Script execution time: 6.29 seconds with chunk size of 12
jpyorre@dievortex URL Analysis with ML Research % |
```

Python	2.51	12	0	UserID&am
Python	2.59	12	0	IFUq_VJ0X
Python	2.52	12	0	
Python	2.37	12	0	
Python	2.45	12	0	
Python	2.31	12	0	
Python	2.33	1	0	ip;login=
Python	2.39	12	0	o&in
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	2.60	4	0	

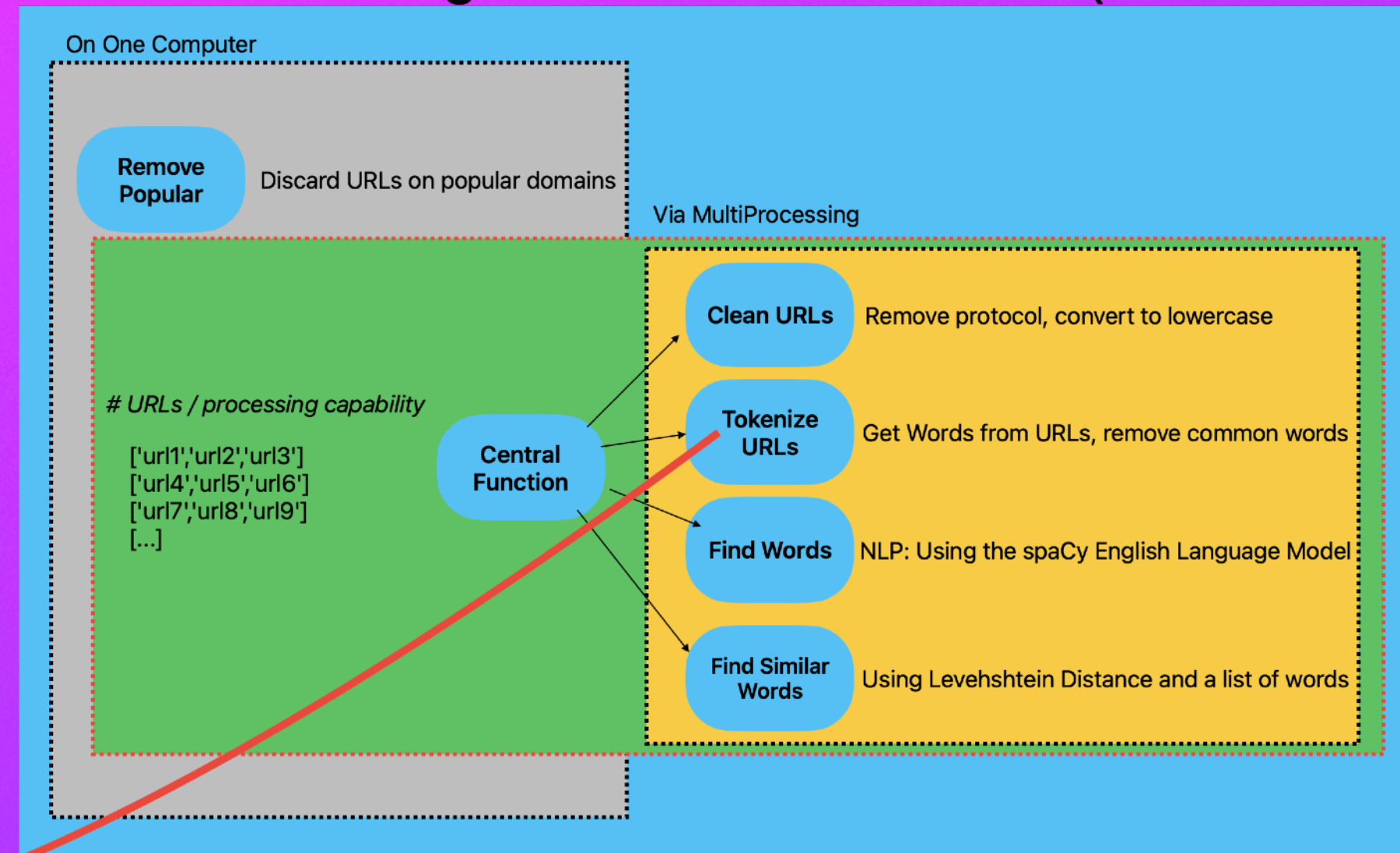
The architecture using one machine and multiprocessing



1: Users upload URLs



2: Run code using all available resources (unless already in mongodb)

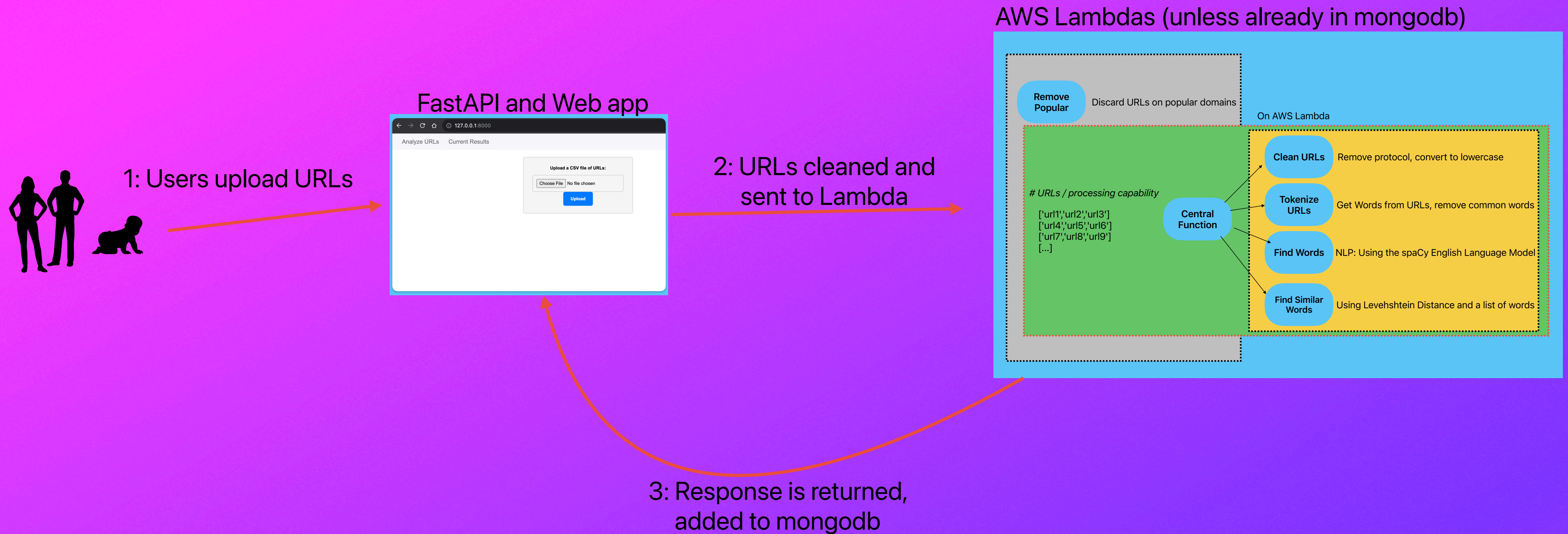


3: Response is returned, added to mongodb

How about 'no machines'?

Lambdas

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



AWS Lambda: Testing one function

```
Environment Vari x lambda_function x (+)
1 import json
2 import Levenshtein
3
4 def lambda_handler(event, context):
5     wordlist = ['banking', 'pharma', 'buy', 'walmart', 'citibank', 'hsbc', 'chase', 'wellsfargo', 'citi', 'bankofamer']
6     data = json.loads(event["body"])
7
8     threshold = 2
9     returnlist = []
10    for w in data:
11        listkeeper = []
12        for i in w['possible_actual_words']:
13            for word in wordlist:
14                if word == 'pharm':
15                    threshold = 2
16                else:
17                    threshold = 1
18                distance = Levenshtein.distance(word, i)
19                if distance <= threshold and len(i) > 5:
20                    w['levenshtein_distance'] = {}
21                    w['levenshtein_distance']['word'] = i
22                    w['levenshtein_distance']['match'] = word
23                    w['levenshtein_distance']['distance'] = distance
24                    listkeeper.append(w)
25                    break # Break out of the inner loop to include the entire word
26        if listkeeper:
27            for item in listkeeper:
28                if len(item) > 1:
29                    returnlist.append(item)
30
31    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': 'benkofmaerical.com',  
'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

4,575 Domains:
9 Function Calls
(508 domains each)

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)

How much is it? Nobody knows....

Lambda Cost:

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

How much is it? Nobody knows....

Lambda Cost:

1 million URLs a day: 2000 Requests
2000 Requests: 60,000 a month

Number of executions (month)

60000

Memory allocation

128 MB

Estimated average duration (ms)

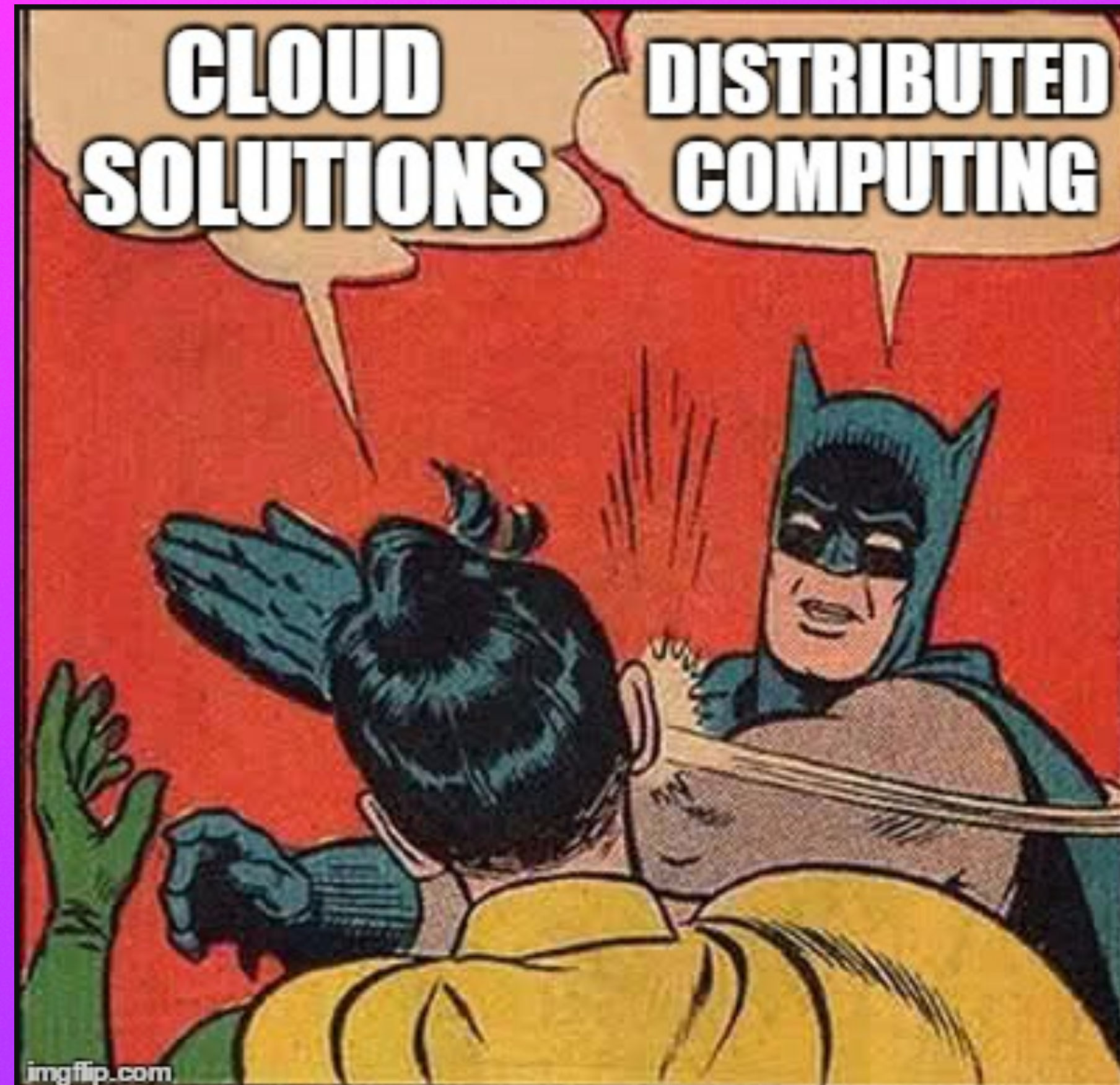
163

Include free tier?
 Yes No

Results

Request costs:	\$0.01/month
Execution costs:	\$0.02/month
<hr/>	
Total AWS Lambda costs:	\$0.03/month

Distributed Computing

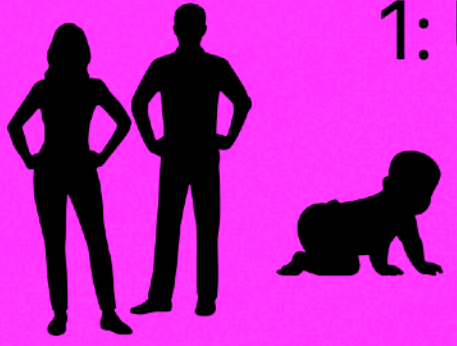


Expensive, then cheap (for a little while)

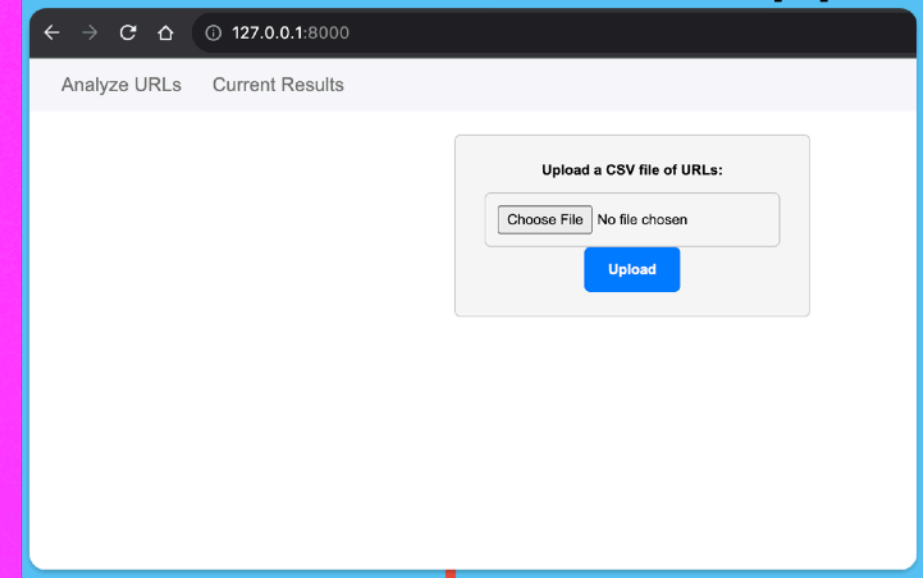
Lots of Machines!



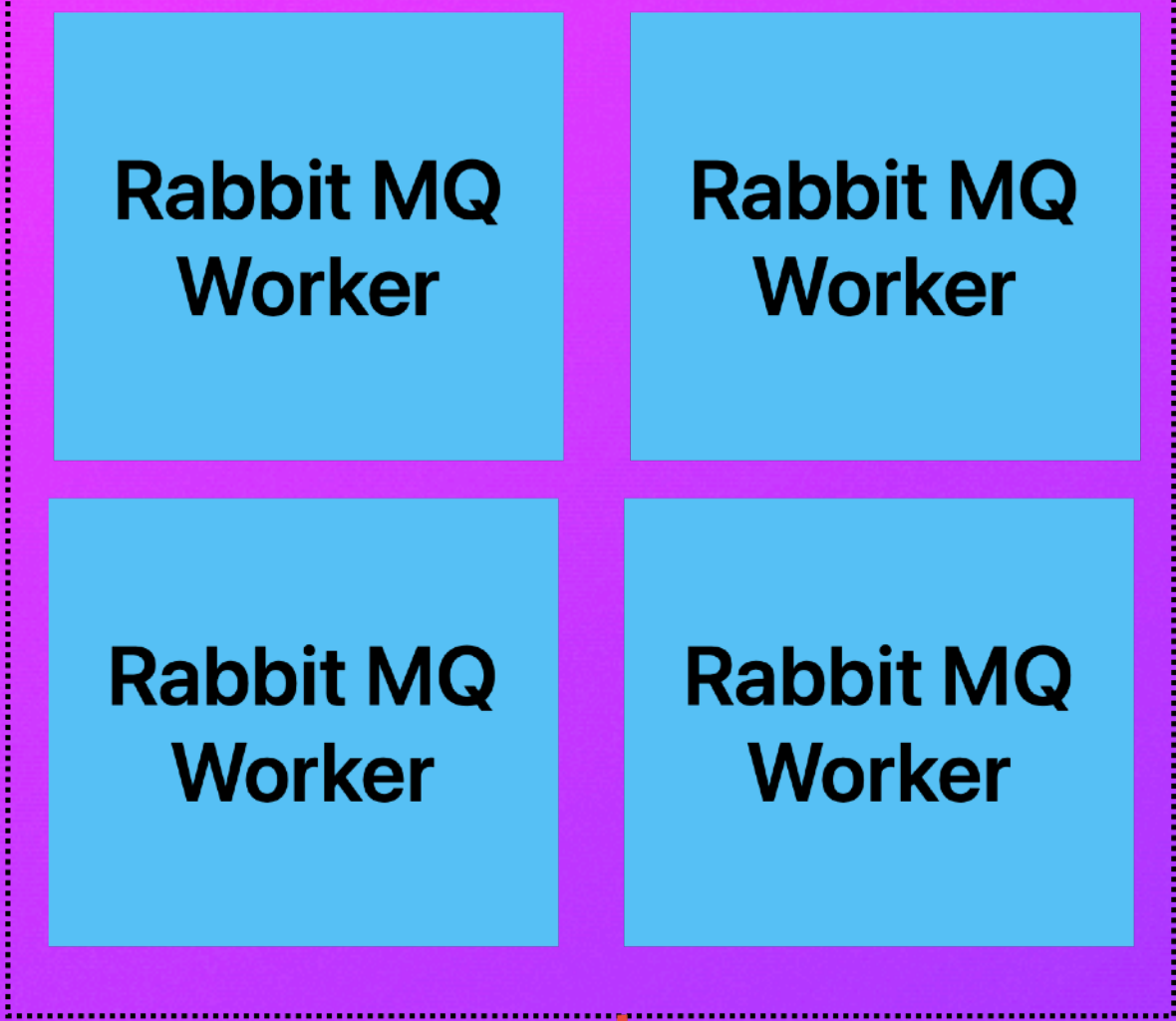
FastAPI and Web app



1: Users upload URLs



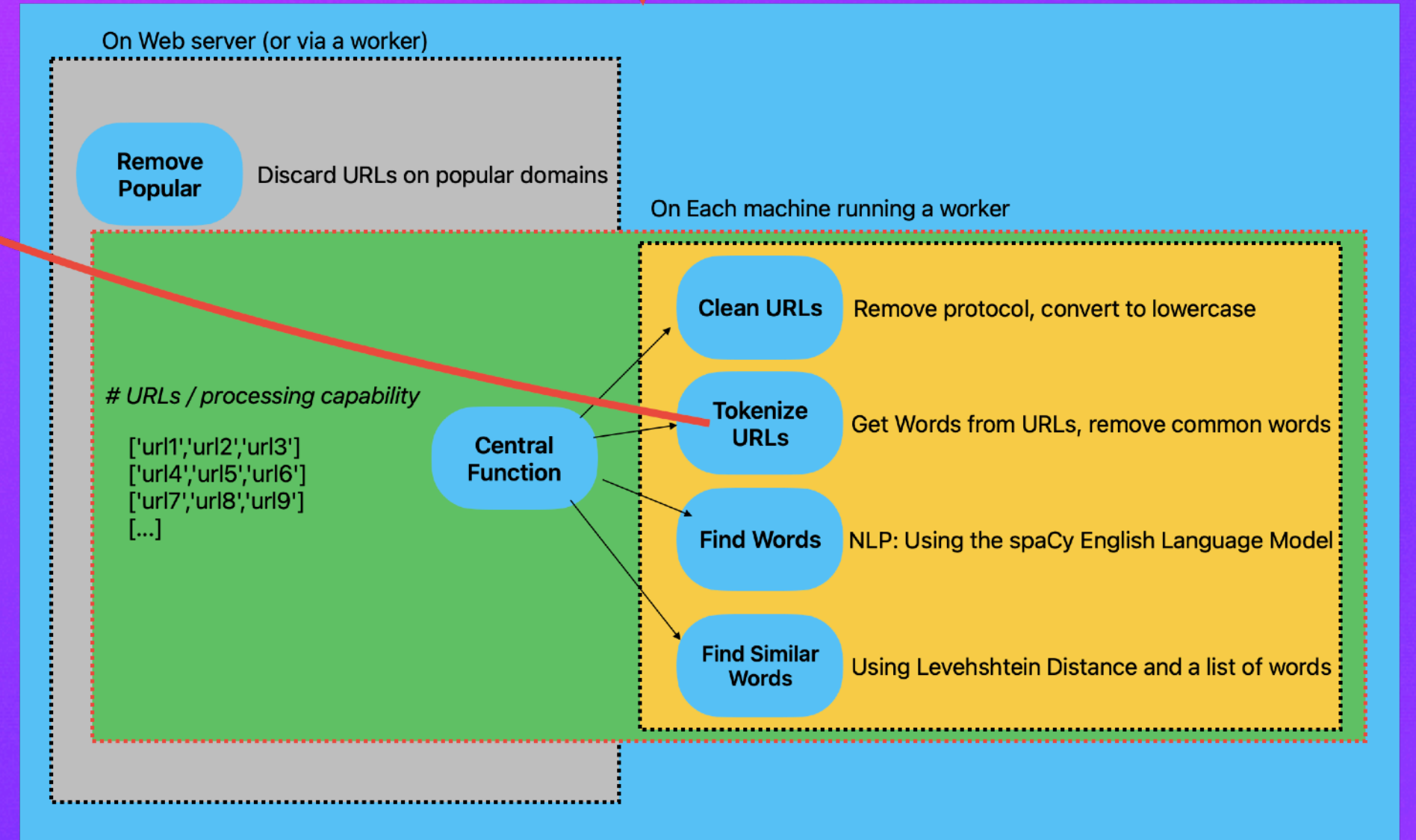
2: URLs cleaned and sent to workers



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

The image displays a workflow for data processing. On the left, three terminal windows show the execution of a Python script named `fastapycrawler.py` on different machines: `node1`, `node2`, and `smallserver`. On the right, a terminal window shows the FastAPI web server starting up using `uvicorn` on `0.0.0.0:8001`. Below the terminals is a screenshot of the RabbitMQ management interface, showing the 'Overview' page for a queue. The interface includes a 'Queue overview' section with a graph of message rates and a 'Message rates' section with a graph and status indicators for various message states.

Message State	Count
Ready	0
Unacked	0
Total	0
Publish	0.00/s
Publisher confirm	0.00/s
Deliver (manual ack)	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	0.00/s
Redelivered	0.00/s

3 Nodes

RabbitMQ Node 1 (in a VM)

RabbitMQ Node 1 (in another VM)

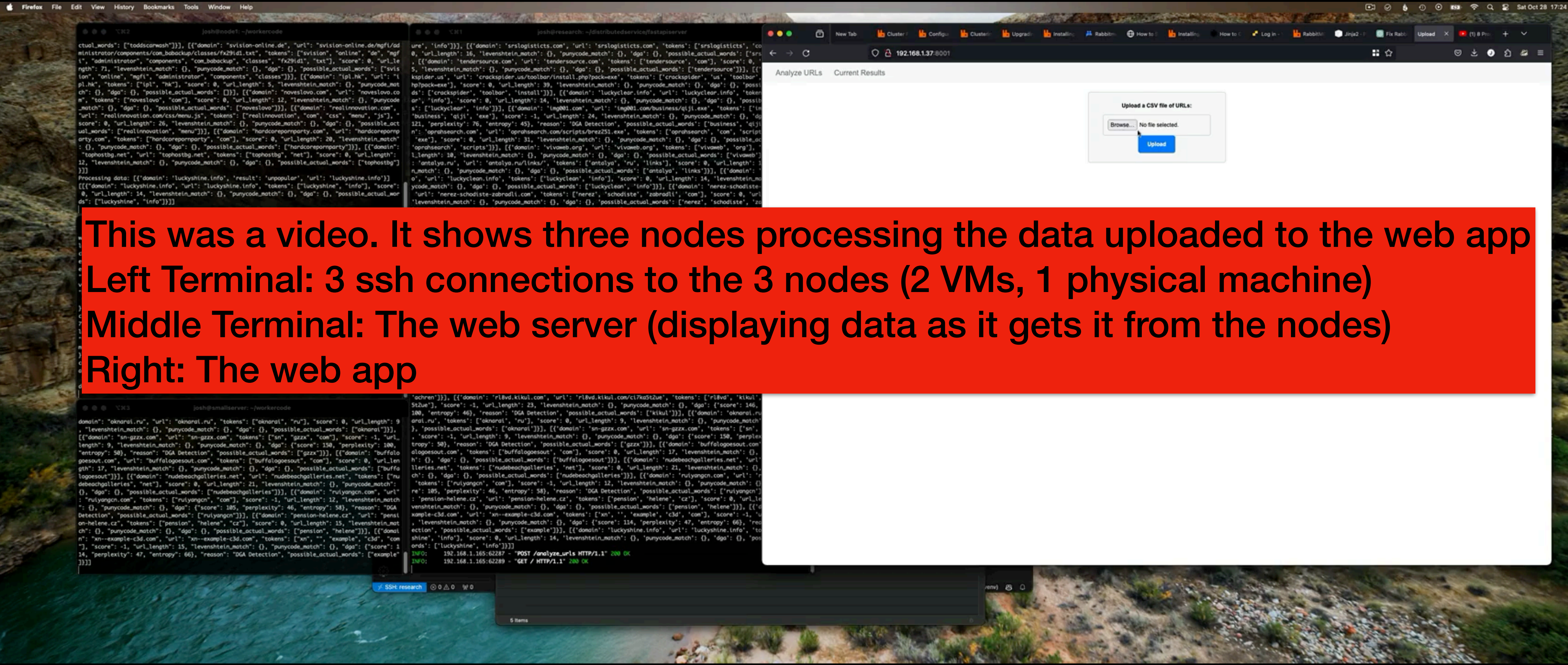
RabbitMQ Node 1 (on a MacBook Pro running Debian)

RabbitMQ Stats/Info

The image displays a setup for three RabbitMQ nodes. On the left, three terminal windows are shown, each running the command `python fastapyconsumer.py` on a different node: `josh@node1`, `josh@node2`, and `josh@smallserver`. On the right, the RabbitMQ management interface is visible, showing the 'RabbitMQ' logo and various performance metrics and graphs. The interface includes a 'Ready' status bar with 0 items, a 'Message rates' section for the last minute, and a 'Message rates' graph. The 'Message rates' section shows: Publish (0.00/s), Publisher confirm (0.00/s), Deliver (manual ack) (0.00/s), Deliver (auto ack) (0.00/s), Consumer ack (0.00/s), and Redelivered (0.00/s). The 'Message rates' graph shows a flat line at 0.00/s for the last minute.

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

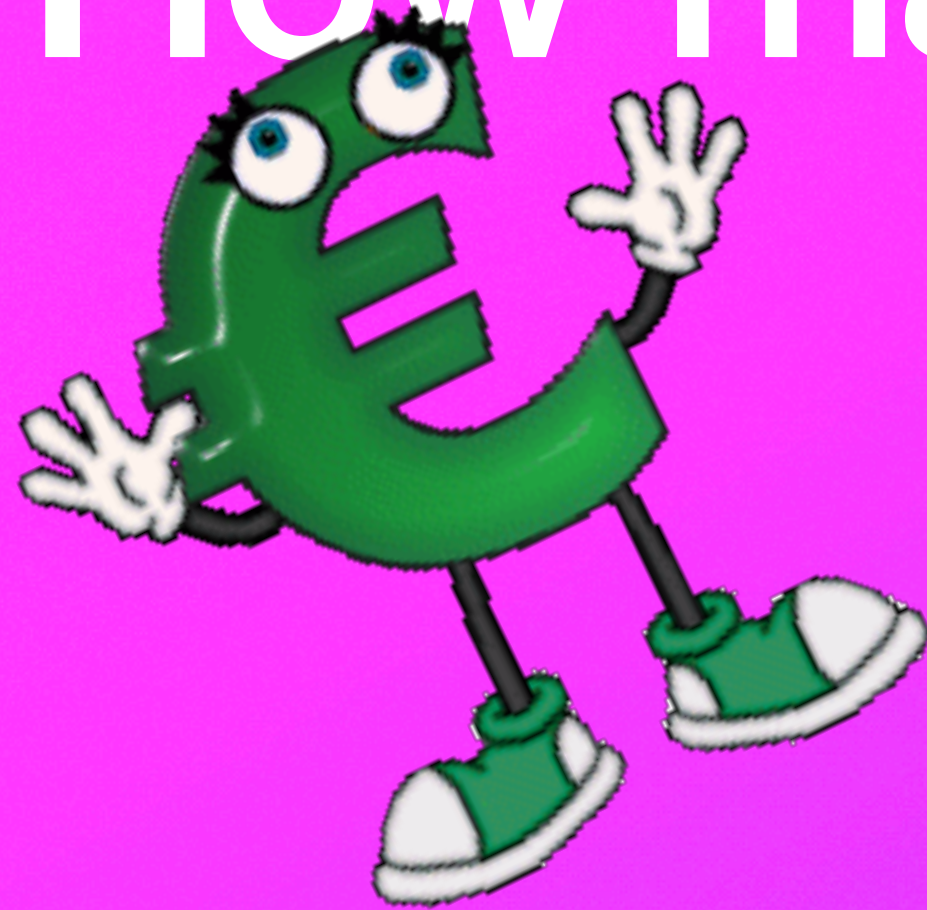
Left Terminal: 3 ssh connections to the 3 nodes (2 VMs, 1 physical machine)

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?



vmware ESXi™

```
Debian GNU/Linux 10 node1 tty1
node1 login: _
```

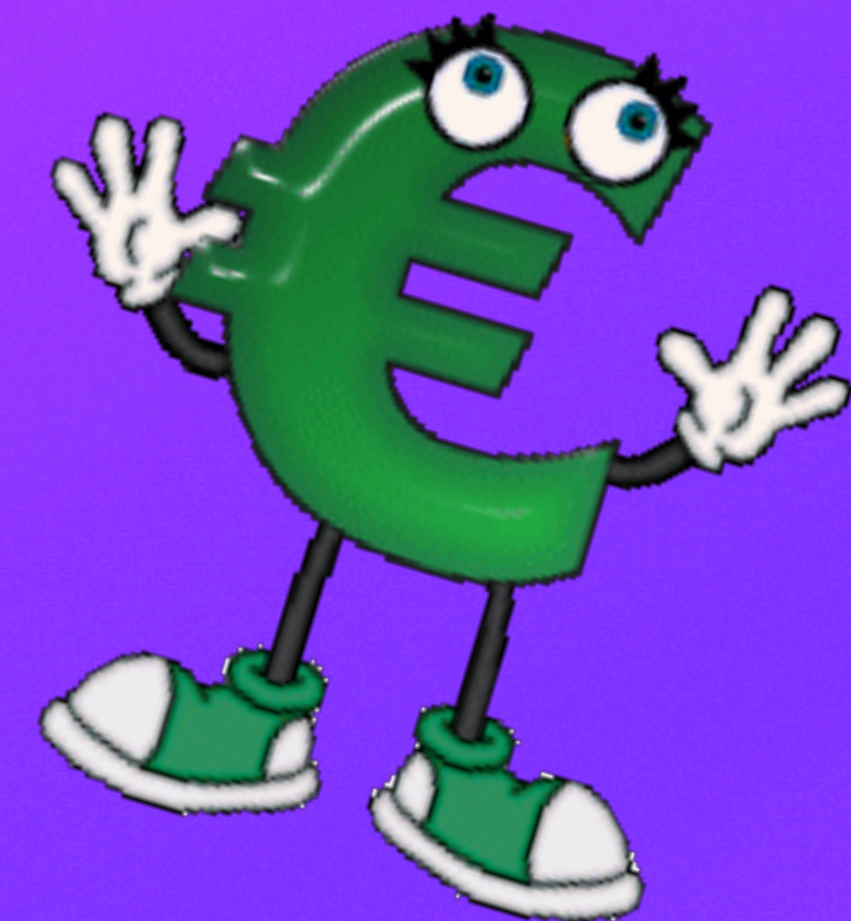
node1	
Guest OS	Debian GNU/Linux 10 (64-bit)
Compatibility	ESXi 7.0 U2 virtual machine
VMware Tools	
CPUs	Yes
Memory	4
Host name	2 GB
	node1

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

vmware ESXi™

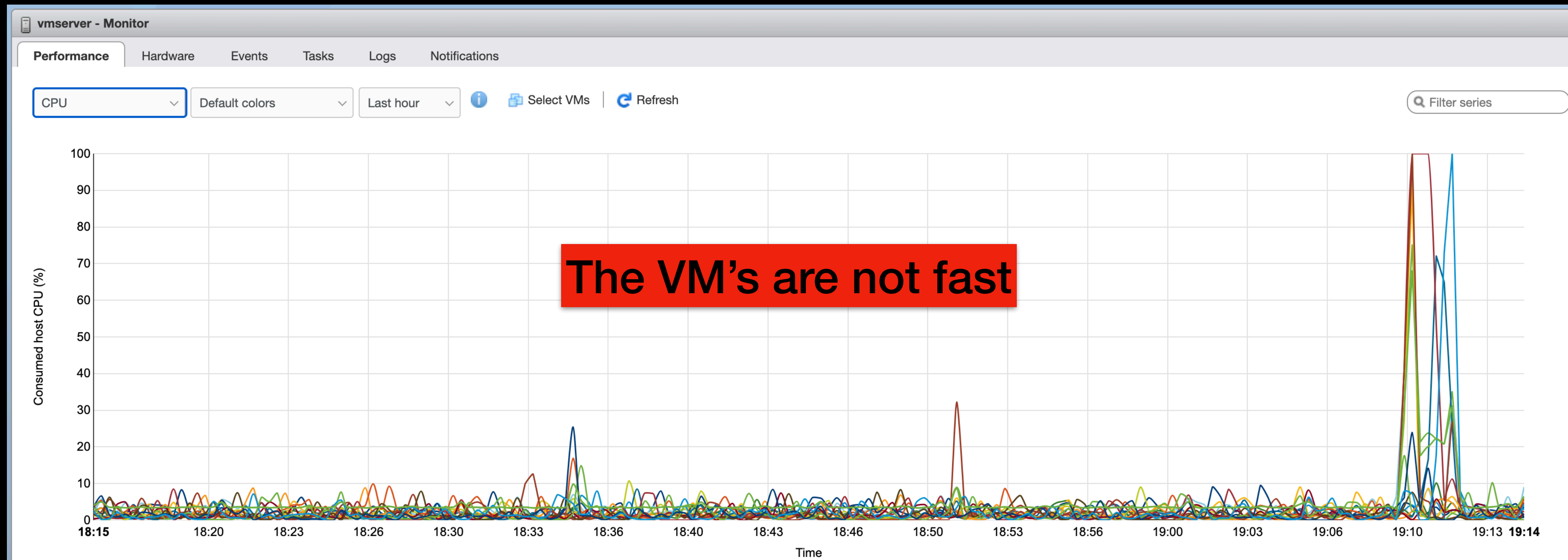
```
Debian GNU/Linux 10 node1 tty1
node1 login: _
```

node1	
Guest OS	Debian GNU/Linux 10 (64-bit)
Compatibility	ESXi 7.0 U2 virtual machine
VMware Tools	
CPUs	Yes
Memory	4
Host name	2 GB
	node1



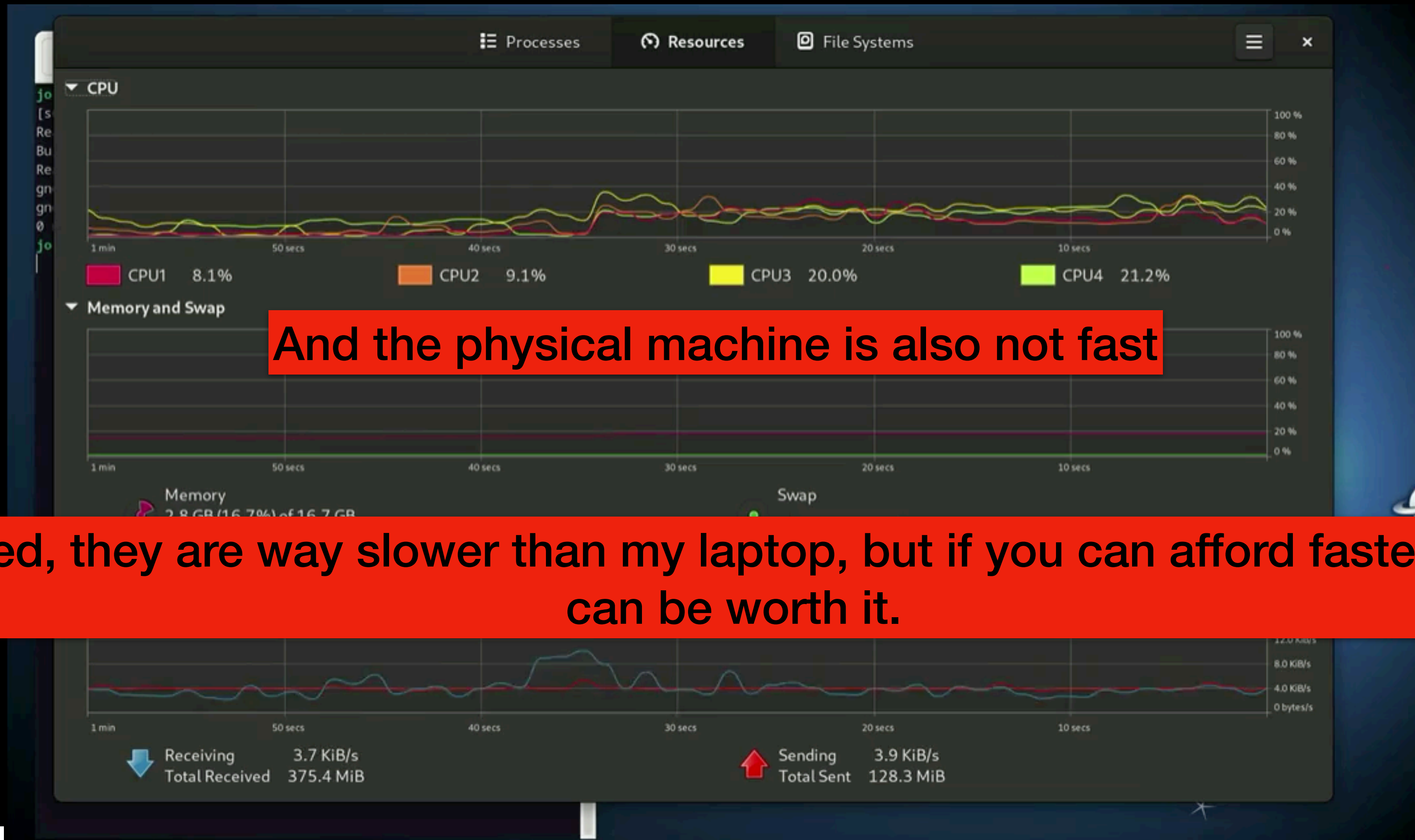
Distributed Computing: 3 nodes, 5000 URLs

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:



And the physical machine is also not fast

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
worldgympere.com	worldgympere.com	-2	Investigate, Virustotal	15		worldgympere
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 /walmart	benkofmaerical.com	-1	levenshtein	31	<ul style="list-style-type: none">• benking, banking• walmart, walmart	benkofmaerical, benking, walmart
diaryofagameaddict.com	diaryofagameaddict.com	-2	Investigate, Virustotal	21		diaryofagameaddict

Actions on Streaming URLs

```

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']}")

```

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)

```

for word in result.get('possible_dga_words', []):
    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("Punycod Match:")
for match in result.get('punycod_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']}")

```

```
Python
-1,dl.downf468.com/n/3.0.26/6068293/winrarfree.exe=0d=0ahttp://dl01.fabdmr.com/n/3.0.26/5034600/j_downloader.exe=,,ML SVM Model
-1,dl.downf468.com/n/3.0.26/187807/utorrent.exe=0d=0ahttp://dl01.facdmr.com/n/3.0.26/4993202/mediaget.exe=,,ML SVM Model
-1,directxex.com/uploads/144543902.rundll32.exe trojan,directxex.com,ML SVM Model
-1,directxex.com/uploads/1406101817.server.exe,directxex.com,ML SVM Model
-1,directxex.com/uploads/84937512.and.exe win32/injector.aujq,directxex.com,ML SVM Model
-1,0uk.net/zaaqw/pony.exe,0uk.net,DGA Detection
-1,directxex.com/uploads/662268336.bin.exe,directxex.com,ML SVM Model
-1,panazan.ro/online/libraries/pattemplate/pattemplate/modifier/html/im/o/z/3pingo/cfg.bin,panazan.ro,ML SVM Model
-1,dl.downf468.com/n/3.0.26/12050993/free+rar+extract+frog.exe,dl.downf468.com,ML SVM Model
-1,qualityindustrialcoatings.com/wm19l5st/index.html,qualityindustrialcoatings.com,ML SVM Model
-1,dl.downf468.com/n/3.0.26/11930758/file_installer.exe=0d=0ahttp://dl01.fabdmr.com/n/3.0.26/187807/utorrent.exe=,,ML SVM Model
-1,ttb.tbddl.com/download/request/51a9b7865f1c1eb81f000001/ctlli2yz?pubid=3457_2776&clickid=3247011638 pup.fakejava,ttb.tbddl.com,ML SVM Model
-1,dl.downf468.com/n/3.0.26/11930758/file_installer.exe=0d=0ahttp://dl.downf468.com/n/3.0.26/11928104/ares.exe=,,ML SVM Model
-1,download.ttrili.com:98/setup[11]-rl.exe,download.ttrili.com,ML SVM Model
-1,crackzone.net/data/super_mp3_download_version_3.3.4.6_serial_keys_gen-bee3afe71a.html,crackzone.net,ML SVM Model
-1,directxex.com/uploads/620509324.minimon.exe,directxex.com,ML SVM Model
-1,dl.downf468.com/n/3.0.26/11930758/file_installer.exe=0d=0ahttp://dl.softohqimjjedf0jq.net/n/3.0.26/12091048/skype.exe=,,ML SVM Model
-1,bj04.com/myimg/?img1507.jpg,bj04.com,DGA Detection
-1,dl.downf468.com/n/3.0.26/11930758/file_installer.exe=0d=0ahttp://dl.downf468.com/n/3.0.26/7088851/flv_media_player.exe=,,ML SVM Model
-1,miespaciopilates.com/7fy2fzng/index.html,miespaciopilates.com,ML SVM Model
-1,dl.downf468.com/n/3.0.24.1/12015256/windowsproductkeycodefinder2.20.exe,dl.downf468.com,ML SVM Model
-1,thefxarchive.com/downloads/wlmm/mmk_warp_variations.exe,thefxarchive.com,ML SVM Model
-1,dl.downf468.com/n/3.0.26/2105407/avs_media_player.exe=0d=0ahttp://dl.softohqimjjedf0jq.net/n/3.0.26/4351718/vlc_media_player.exe=,,ML SVM Model
-1,mobatory.com/5bj0eswiecc78rvp3eguf05xossn1segz4653xhs4?37078=46se8http%2f%3f%3ftrahic.ru%3f6h3m0gs9sgb8vvr70voqj1fa67&j68ljm=86b42bbis=t85660263&pqs
ubf2hr=1,,ML SVM Model
```

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

```
-1,sunny99.cholerik.cz/plugins/3yvprqfj.php,sunny99.cholerik.cz,DGA Detection
-1,directxex.com/uploads/777724411.safetycheck.exe,directxex.com,ML SVM Model
-1,pornstarss.tk/ntk/index.php?id=105828,pornstarss.tk,ML SVM Model
-1,directxex.com/uploads/939195944.newmine.exe msil/coinminer.ay,directxex.com,ML SVM Model
-1,dl.downf468.com/n/3.0.26.2/12014376/setup.exe=0d=0ahttp://dl.softpzivrubajjui.net/n/3.0.26.2/5565169/flvplayer.exe=,,ML SVM Model
-1,download.ttrili.com:98/setup%5b57%5d-rl.exe,download.ttrili.com,ML SVM Model
-1,win2150.vs.easily.co.uk/f49oj2tb/index.html,win2150.vs.easily.co.uk,ML SVM Model
-1,dl.downf468.com/n/3.0.26/2632028/avs_media_player.exe=0d=0ahttp://download.multiinstall.com.br/a75e4b51a7dfadaa4b8a88436b76af41/quadro_1600x1200.exe=
,,ML SVM Model
-1,dl.downf468.com/n/3.0.26.2/5738856/flv_media_player.exe=0d=0ahttp://dl.softpzivrubajjui.net/n/3.0.26.2/10064255/hitmanpro.exe=,,ML SVM Model
-1,feiyang163.com/soft/fyspeaker.exe,feiyang163.com,DGA Detection
-1,download.ttrili.com:98/setup%5b79%5d-rl.exe,download.ttrili.com,ML SVM Model
-1,hst-19-33.splius.lt,hst-19-33.splius.lt,DGA Detection
-1,puentaereo.info/fha5c5iw/index.html?s=883&lid=2231&elq=11f7b1b5179f45b09737bdf10d0fe61f,puentaereo.info,ML SVM Model
-1,download.ttrili.com:98/setup%5b75%5d-rl.exe,download.ttrili.com,ML SVM Model
-1,dl.downf468.com/n/3.0.26/4351718/vlc_media_player.exe=0d=0ahttp://dl.softohqimjjedf0jq.net/n/3.0.26/6708421/ares.exe=,,ML SVM Model
-1,formessengers.com/download.php?pn=mlp,formessengers.com,ML SVM Model
-1,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=,,ML SVM Model
-2,dl01.faddmr.com/n/e176d94e-d9b7-11e2-a752-00259033c1da/setup.exe?tid=102dccc4aa5799d2efb748b9dd0e4ffake,dl01.faddmr.com,ML SVM Model, DGA Detection
-1,cofeb13east.com/download.php?ln5/ca==,cofeb13east.com,ML SVM Model
-1,dl.downf468.com/n/3.0.24.1/12015430/ntkeyenterpriseedition3.80.exe,dl.downf468.com,ML SVM Model
-1,adserving.favorit-network.com/eas?camp=19320;cre=mu&grpId=1738&tag_id=618&nums=fgapbjfaaa,adserving.favorit-network.com,ML SVM Model
-1,tecslide.com/js/down/sbin1/ms0ftadapter.exe,tecslide.com,ML SVM Model
-1,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=,,ML SVM Mo
del
-1,directxex.com/uploads/2074531303.bin.exe win32/napolar.a,directxex.com,ML SVM Model
-1,dl.downf468.com/n/3.0.25/12023961/microsoft+hesap+makinesi++.exe,dl.downf468.com,ML SVM Model
```

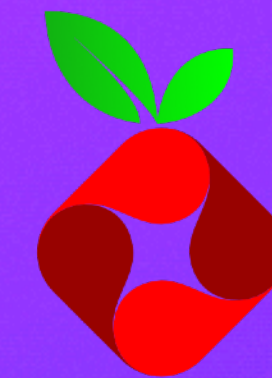
Bad URLs

```
punteaereo.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/avs_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

```
praxisww.com -1,scdsfdfgd
quinnwealth.com -1,praxisww.
cofeb13east.com -1,directxex
zyxyfy.com -1,directxex
silurian.cn -1,directxex
ns2ns1.tk -1,petplease
downloaddirect.com -1,textsex.t
reishus.de -1,directxex
lostartofbeingadame.com -1,vvps.ws/4
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
```



Pi-hole



The Code

The screenshot shows a GitHub repository page for 'URLAnalysis_at_Scale' by user 'jpyorre'. The repository is public and has 0 forks and 0 stars. The main branch is 'main'. The repository contains several files and folders: 'utilities', 'web_app', '.gitattributes', '.gitignore', and 'README.md'. The 'README.md' file is selected and its content is displayed. The content of the README includes a description of the project as a Flask web app running on FastAPI, which processes CSV files of URLs to determine if they are malicious. It also mentions that the project is in active development and provides a link to a presentation at 'https://pyosec.com'. A note at the bottom states that the project was built to run on Python 3.11. The right sidebar shows the 'About' section with no description, 'Releases' section with no releases published, 'Packages' section with no packages published, and a 'Languages' section showing the code is primarily Python (82.2%), with HTML (11.0%) and CSS (6.8%).

Navigation: <> Code, Issues, Pull requests, Actions, Projects, Security, Insights

Repository: jpyorre / URLAnalysis_at_Scale (Public)

Branches: main (1 branch), Tags: 0 tags

Files:

File/Folder	Commit Message	Time Ago
utilities	First Commit	5 hours ago
web_app	Moved Readme to the right location	5 hours ago
.gitattributes	Initial commit	6 hours ago
.gitignore	remove DS_Store	5 hours ago
README.md	formatting...	2 minutes ago

Commit: jpyorre formatting... (85bfdbb) 2 minutes ago, 5 commits

README.md

This is a Flask web app, running on top of FastAPI. It takes in a CSV file of URLs that it runs through various processes to determine if a URL is malicious or not.

It's in active development. A website will be set up soon for demo purposes, but you can easily set it up to test on your own using the following instructions.

To see a presentation on this, visit <https://pyosec.com>

Note: This was built using multiple methods for optimization. One version of this web app use AWS Lambdas for its functions while another version uses RabbitMQ to send URLs to process to multiple physical machines in other locations in order to make use of their multiprocessing. The code in this repository only uses multiprocessing. I will eventually document the setup for the other versions - it's just a little complex to put the three separate options in one repository and still make it easy for anyone to try running their own version of this web app/api.

This was built to run on python 3.11

About: No description, website, or topics provided.

Releases: No releases published

Packages: No packages published

Languages: Python 82.2%, HTML 11.0%, CSS 6.8%

Thank you!

<https://pyosec.com>

https://github.com/jpyorre/URLAnalysis_at_Scale