KEYSIGHT

Hunting / Scraping with Favicons

Mihai Vasilescu
Andrei Niculae

About us

- Security Research Engineer
- Sandboxing / honeypots / phishing
- @_mihaiv_

4th year student

Threat Intelligence / ML

github.com/nan-dre

Disclaimer

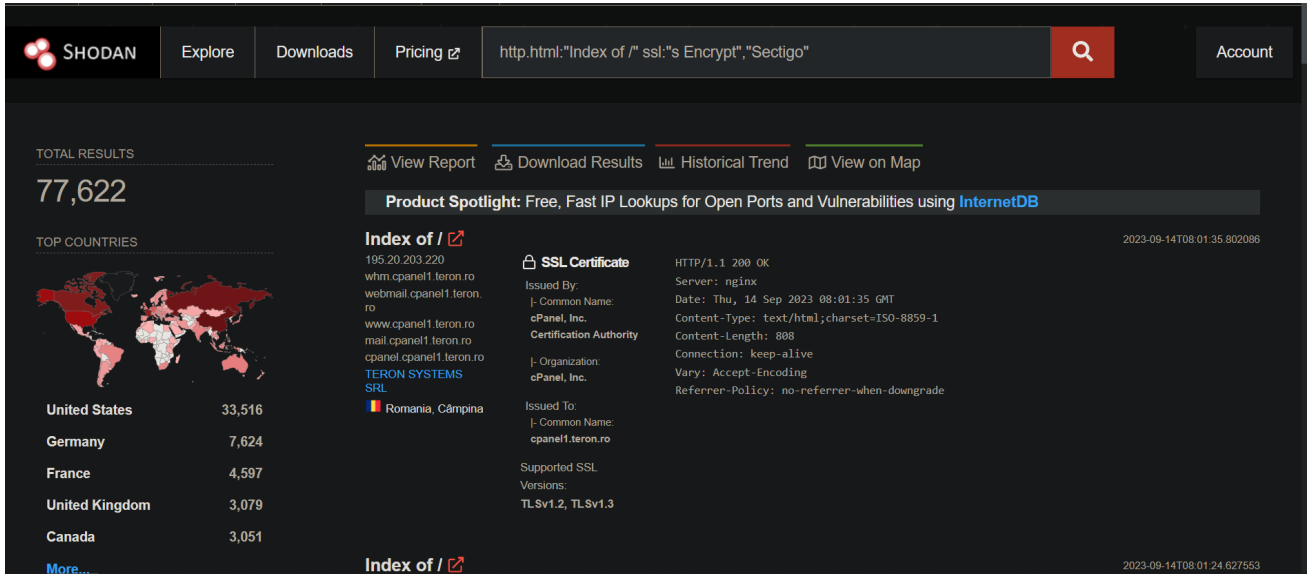
- NOT about phishing detection
- (see Keysight DefCamp presentations 2021 & 2018)

- This is about crawling / hunting

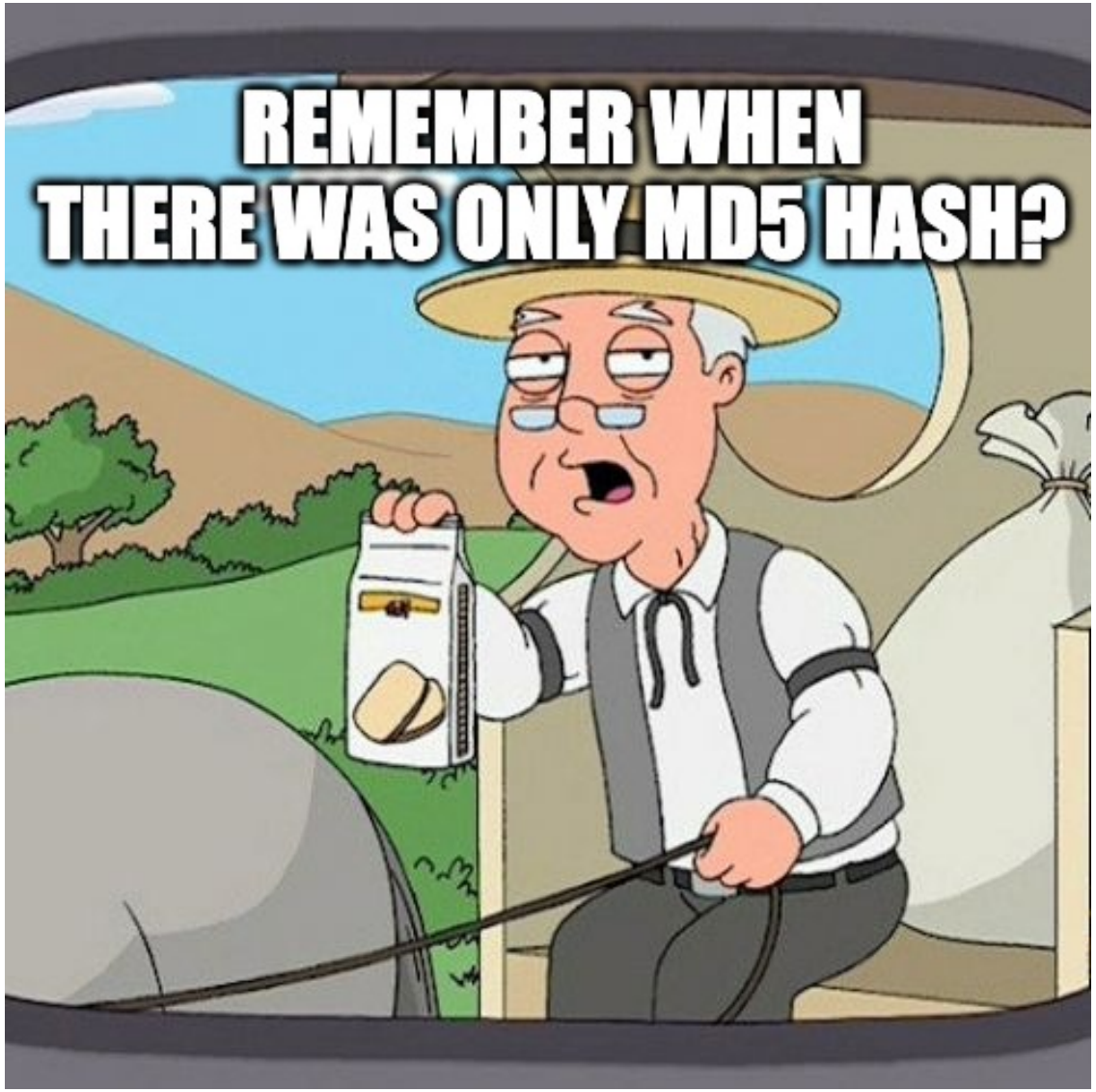
Idea

- Use favicon hashes and other relevant filters to search for potentially malicious websites on shodan.io

- Gain insights from previous detections to identify similarities between phishing sources

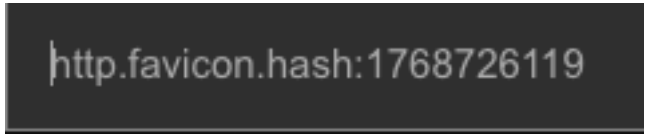


Idea



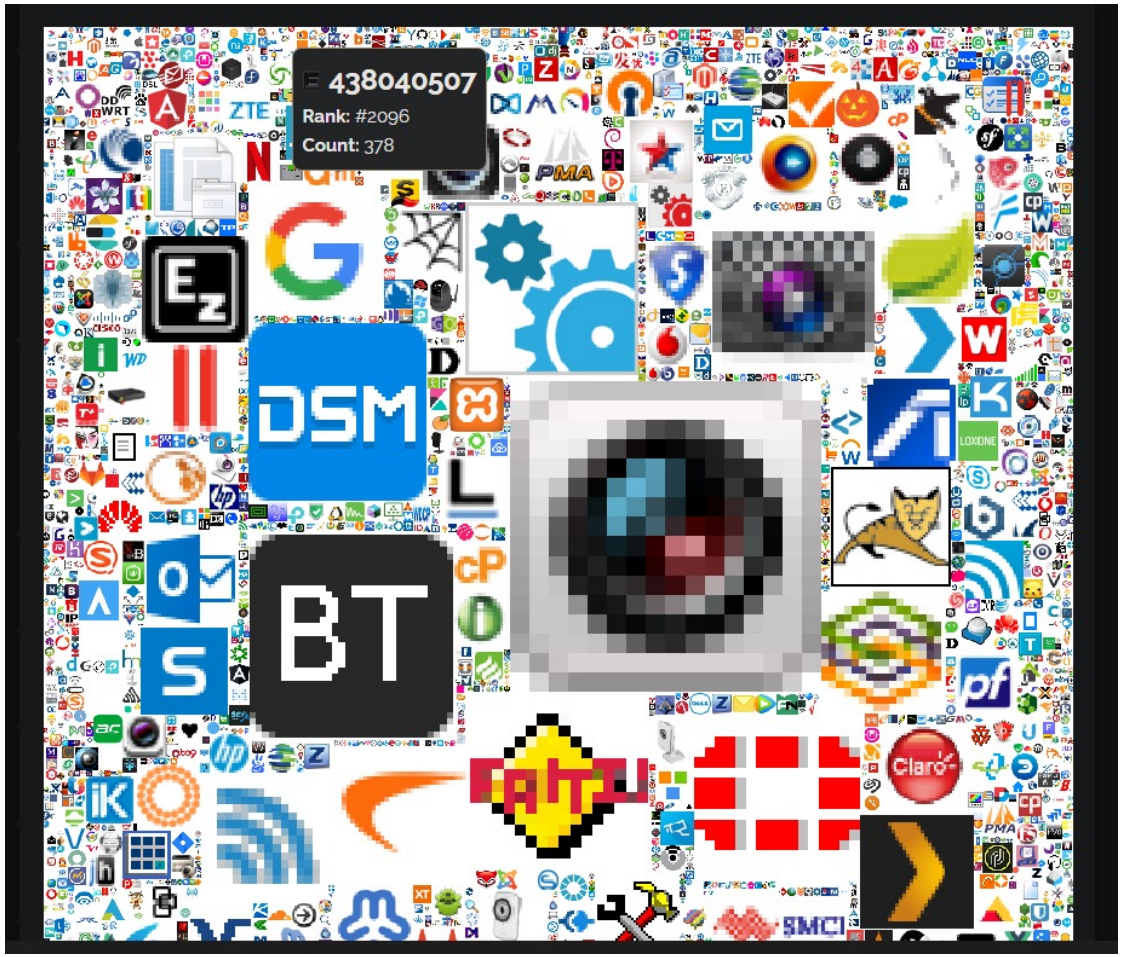
Idea

- Shodan uses murmur hash for favicons
- non-cryptographic hash functions
- https://pypi.org/project/mmh3/



What we did

Favicon map

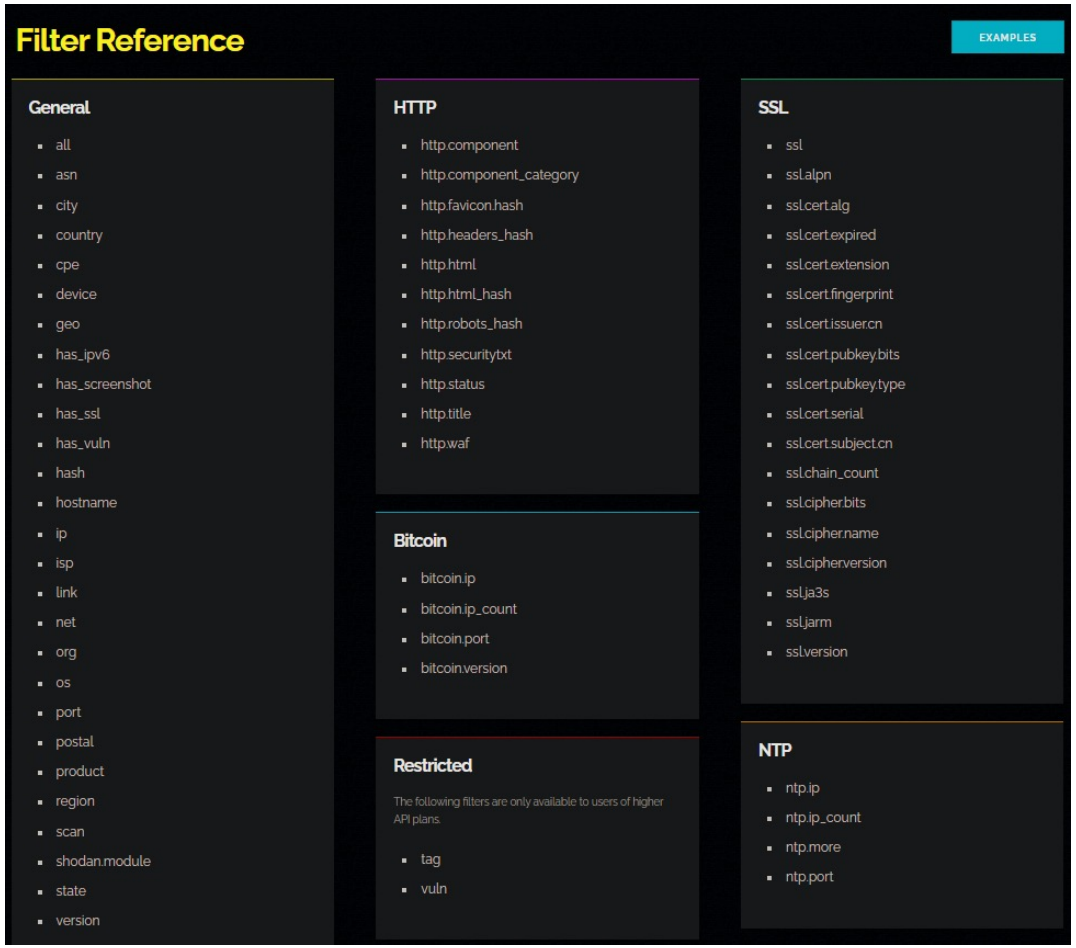


faviconmap.shodan.io/data/favicons.json

Challenges

Need for filtering

- Limited shodan query credits
- Need to restrict the seach queries
- What filters to use?

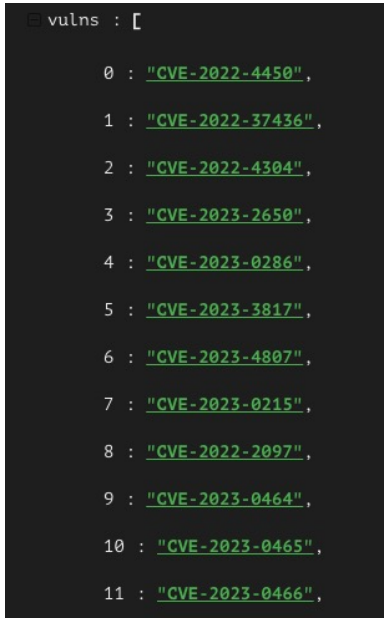


What we did

- ~1/2 of hosts had either ports 21, 53, 993, 995, 2087 open
- ~1/3 of hosts had either ports 22, 110, 2082, 2083, 2086 open
- ~1/3 hosts are affected by CVE-2016-20012, CVE-2021-36368, CVE-2020-15778, CVE-2019-6110 (because they used openssh < 7.9)
- ~1/2 of hosts had SSL certificates from either Sectigo Limited, Let's Encrypt or cPanel.

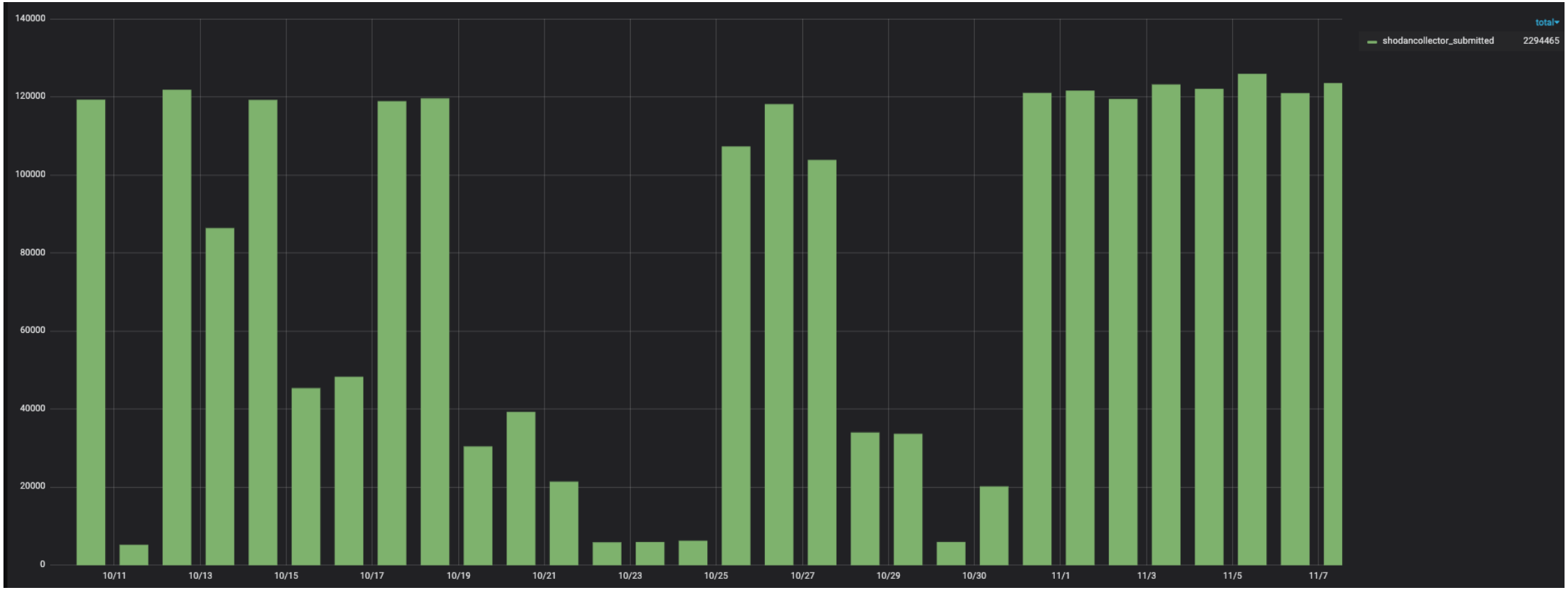
What we did

- Use all 5000 icons from shodan’s icon map
- Filter out all organization (field in ssl certificate) that had > 50 results
- Filter out all ssl certificate providers that had > 200 results
- BUT include the suspicious ssl certificate providers (Let’s encrypt, ZeroSSL etc.)
- Exclude all queries that had 0 results after applying filters
- Hosts with vulnerabilities are a prime target



Results

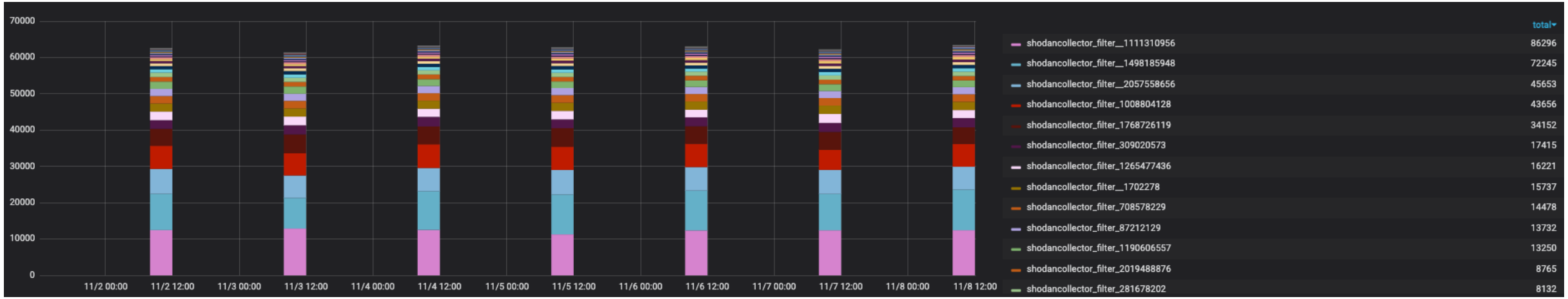
- 2.3 million URLs collected / week



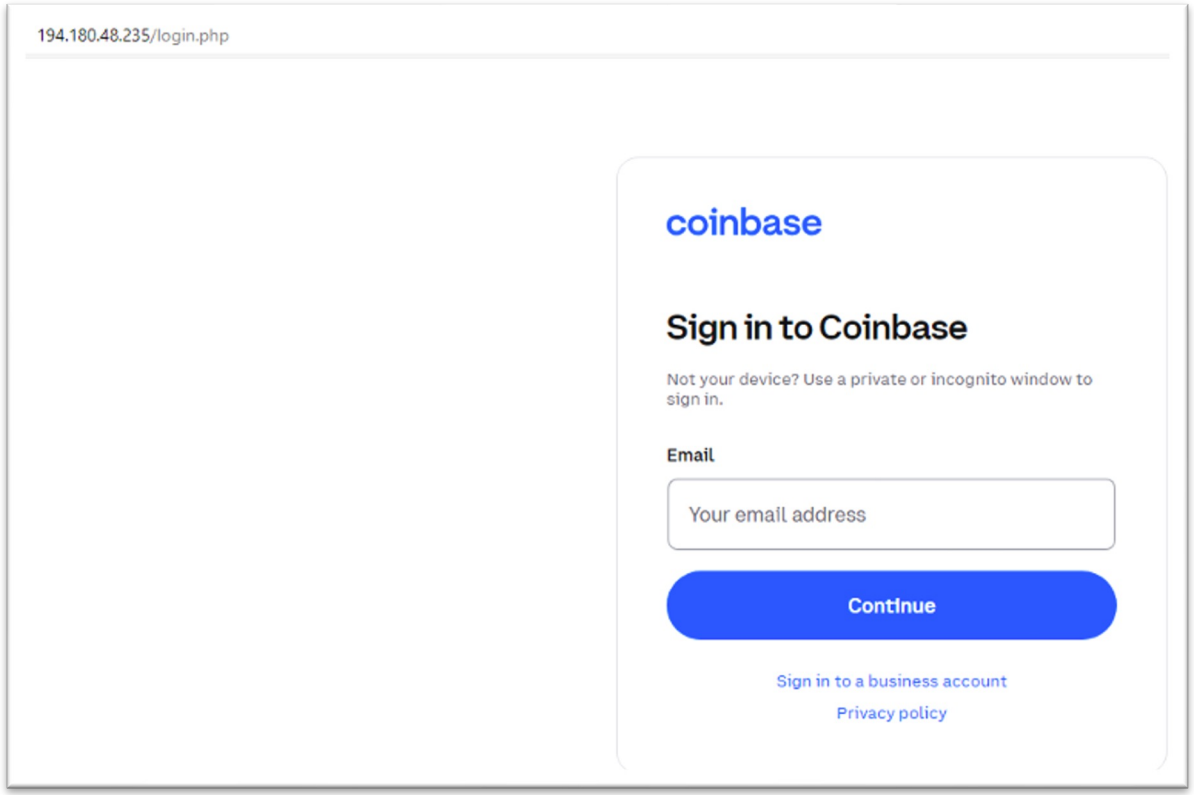
Results

Distribution per favicon/brand

- Ebay
- Apple
- Microsoft
- Adobe
- Outlook

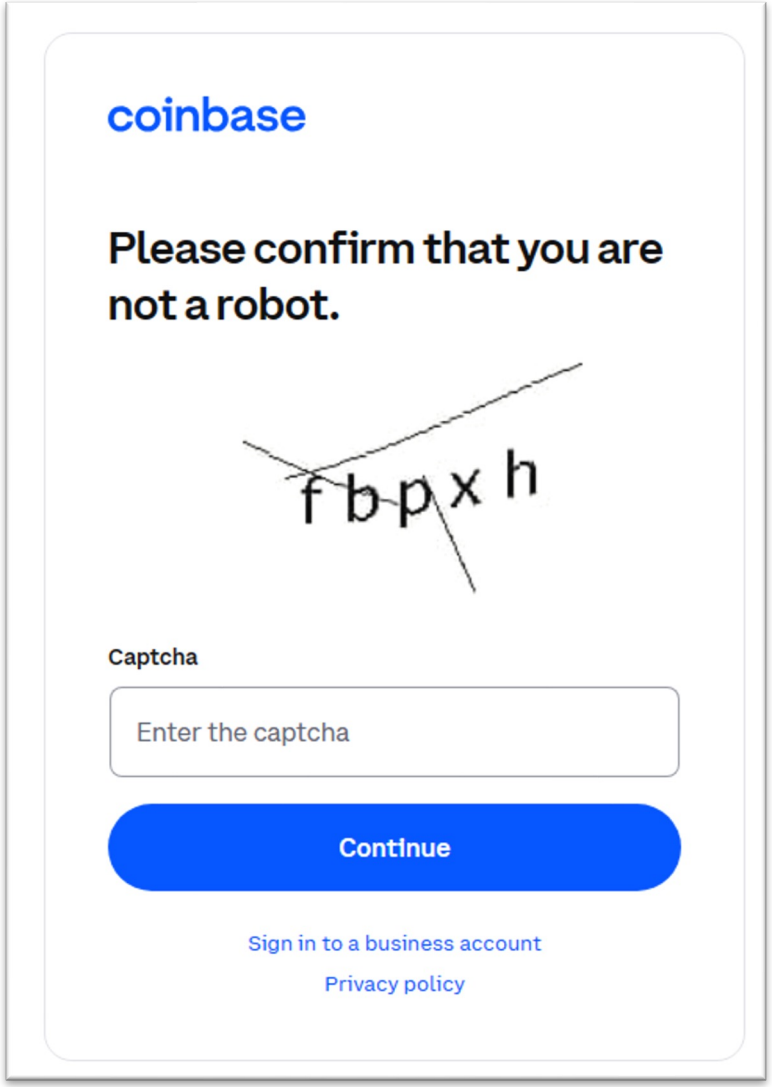


Results

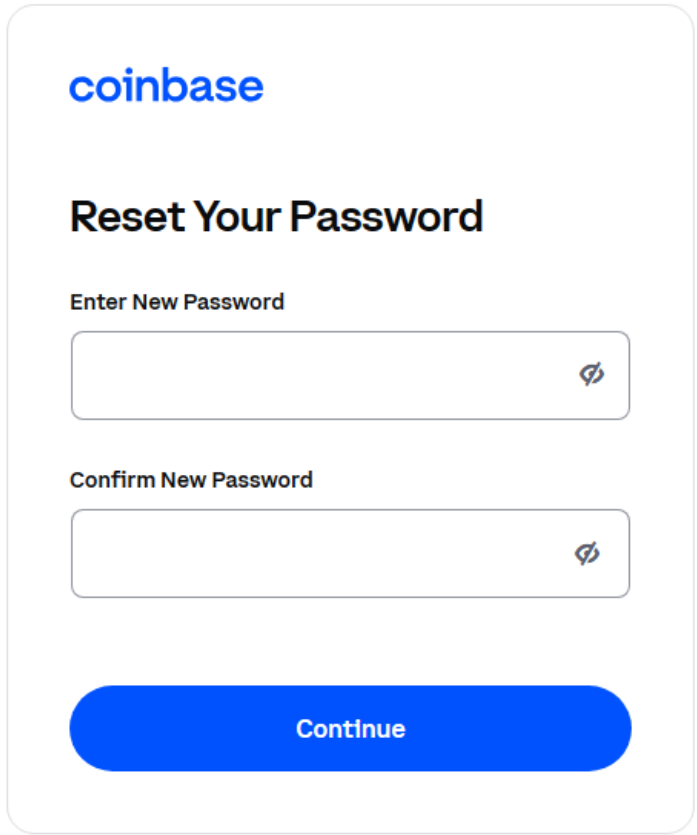

194.180.48.235/login.php
coinbase
Sign in to Coinbase
Not your device? Use a private or incognito window to sign in.
Email
Your email address
Continue
Sign in to a business account
Privacy policy

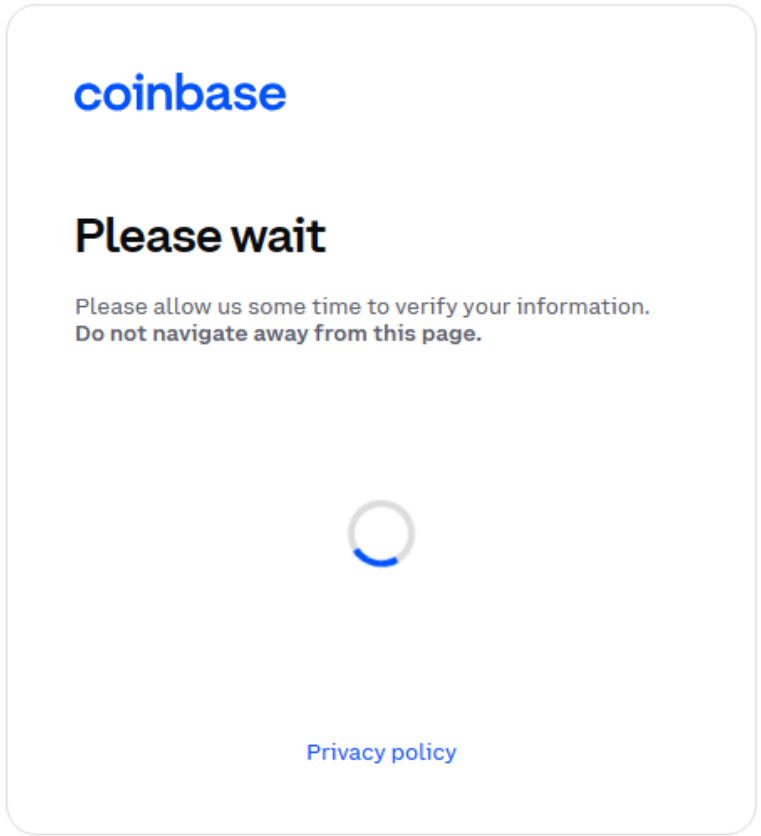
Results

Captcha doesn’t really work

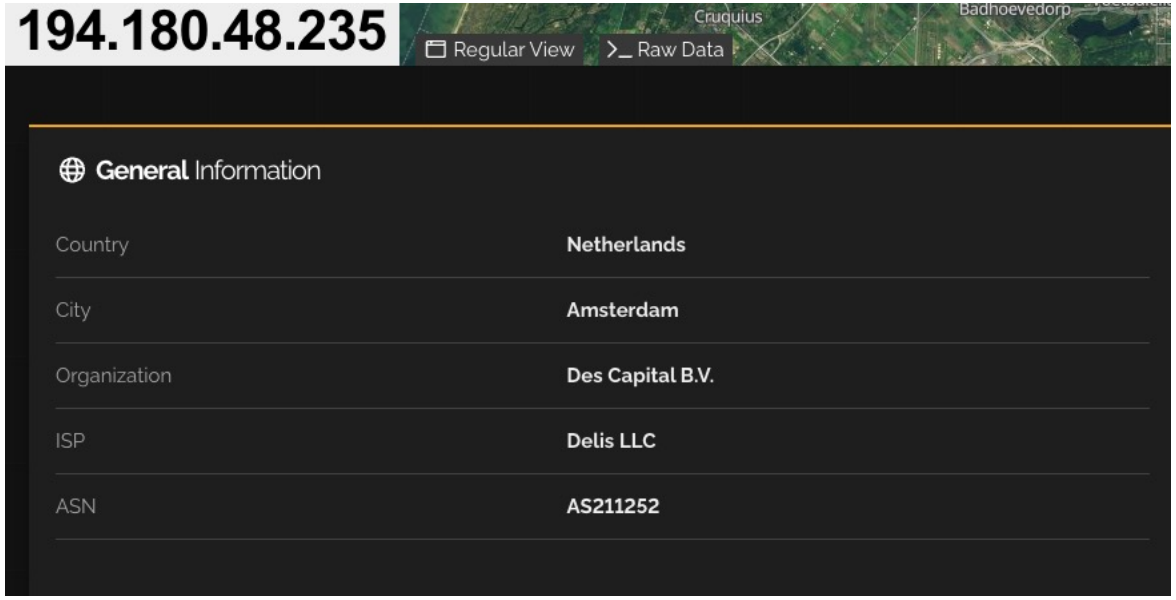


Results

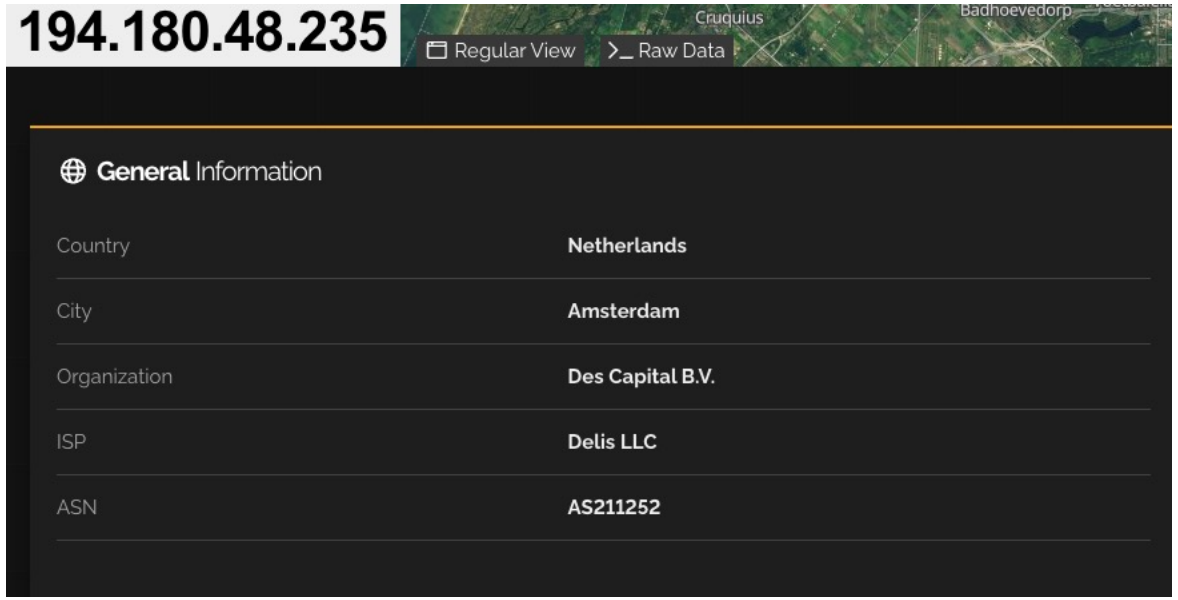

coinbase
Reset Your Password
Enter New Password
Confirm New Password
Continue

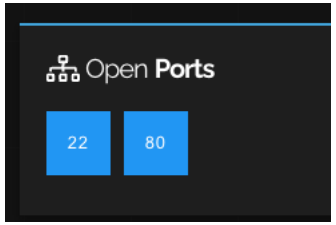

coinbase
Please wait
Please allow us some time to verify your information.
Do not navigate away from this page.
Privacy policy

Results

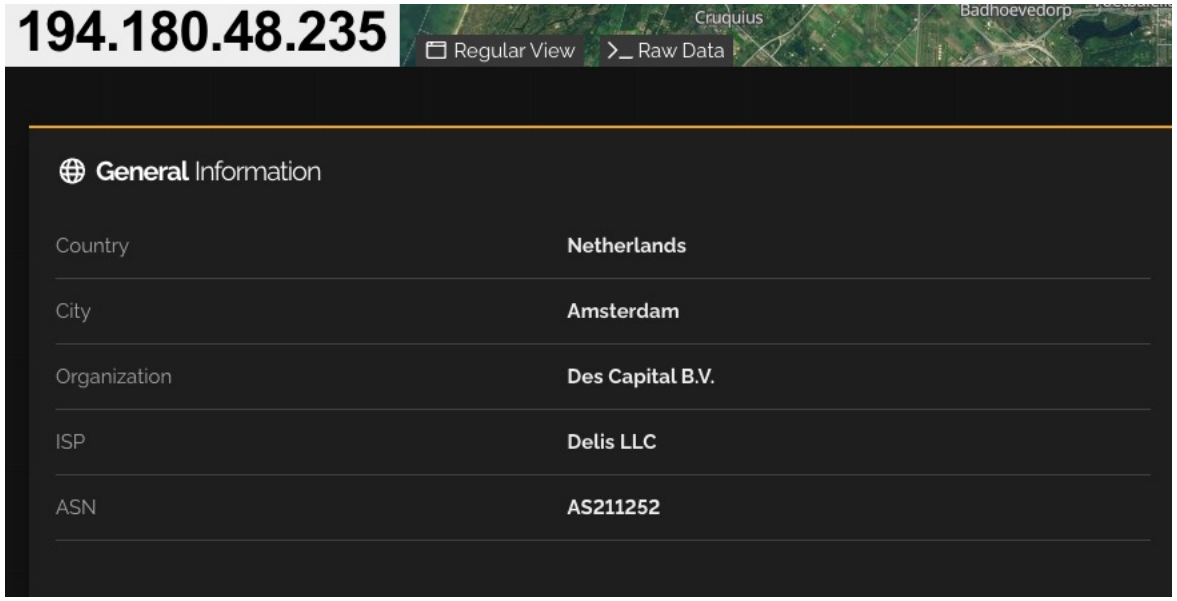

194.180.48.235
Regular View
Raw Data
Cruquius
Badhoevedorp
General Information
Country
Netherlands
City
Amsterdam
Organization
Des Capital B.V.
ISP
Delis LLC
ASN
AS211252

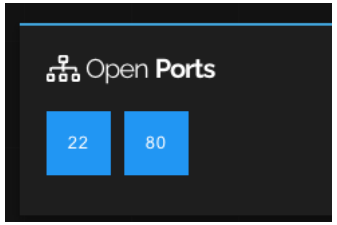
Results

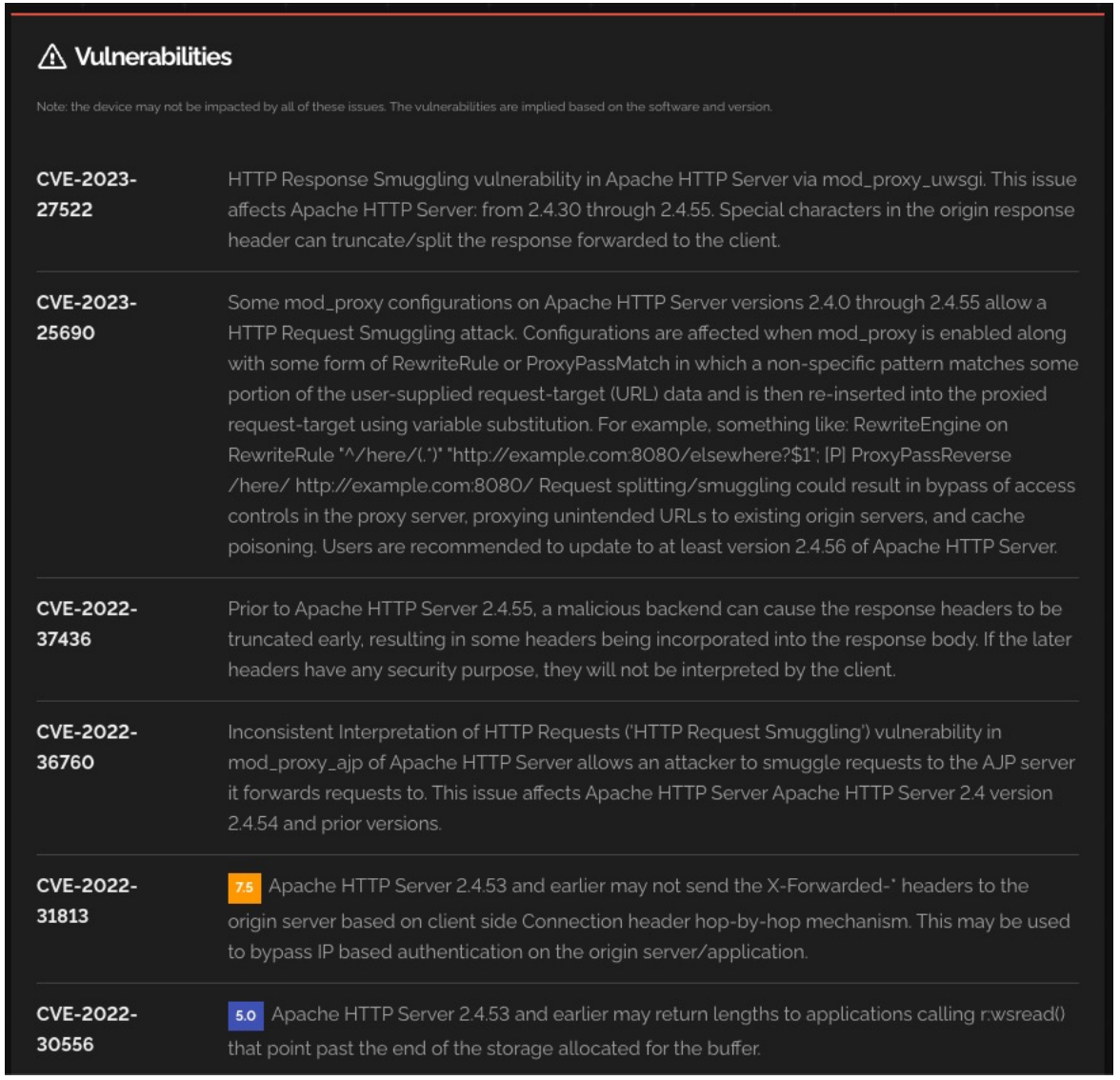

194.180.48.235
Regular View
Raw Data
Cruquius
Badhoevedorp
General Information
Country
Netherlands
City
Amsterdam
Organization
Des Capital B.V.
ISP
Delis LLC
ASN
AS211252


Open Ports
22
80

Results


194.180.48.235
Regular View
Raw Data
Cruquius
Badhoevedorp
General Information
Country
Netherlands
City
Amsterdam
Organization
Des Capital B.V.
ISP
Delis LLC
ASN
AS211252


Open Ports
22
80


Vulnerabilities
Note: the device may not be impacted by all of these issues. The vulnerabilities are implied based on the software and version.
CVE-2023-27522
HTTP Response Smuggling vulnerability in Apache HTTP Server via mod_proxy_uwsgi. This issue affects Apache HTTP Server: from 2.4.30 through 2.4.55. Special characters in the origin response header can truncate/split the response forwarded to the client.
CVE-2023-25690
Some mod_proxy configurations on Apache HTTP Server versions 2.4.0 through 2.4.55 allow a HTTP Request Smuggling attack. Configurations are affected when mod_proxy is enabled along with some form of RewriteRule or ProxyPassMatch in which a non-specific pattern matches some portion of the user-supplied request-target (URL) data and is then re-inserted into the proxied request-target using variable substitution. For example, something like: RewriteEngine on RewriteRule "^/here/(.*)" "http://example.com:8080/elsewhere?$1"; [P] ProxyPassReverse /here/ http://example.com:8080/ Request splitting/smuggling could result in bypass of access controls in the proxy server, proxying unintended URLs to existing origin servers, and cache poisoning. Users are recommended to update to at least version 2.4.56 of Apache HTTP Server.
CVE-2022-37436
Prior to Apache HTTP Server 2.4.55, a malicious backend can cause the response headers to be truncated early, resulting in some headers being incorporated into the response body. If the later headers have any security purpose, they will not be interpreted by the client.
CVE-2022-36760
Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') vulnerability in mod_proxy_ajp of Apache HTTP Server allows an attacker to smuggle requests to the AJP server it forwards requests to. This issue affects Apache HTTP Server Apache HTTP Server 2.4 version 2.4.54 and prior versions.
CVE-2022-31813
7.5
Apache HTTP Server 2.4.53 and earlier may not send the X-Forwarded-* headers to the origin server based on client side Connection header hop-by-hop mechanism. This may be used to bypass IP based authentication on the origin server/application.
CVE-2022-30556
5.0
Apache HTTP Server 2.4.53 and earlier may return lengths to applications calling r:wsread() that point past the end of the storage allocated for the buffer.

Results

Not secure | view-source:194.180.48.235/loading.pl

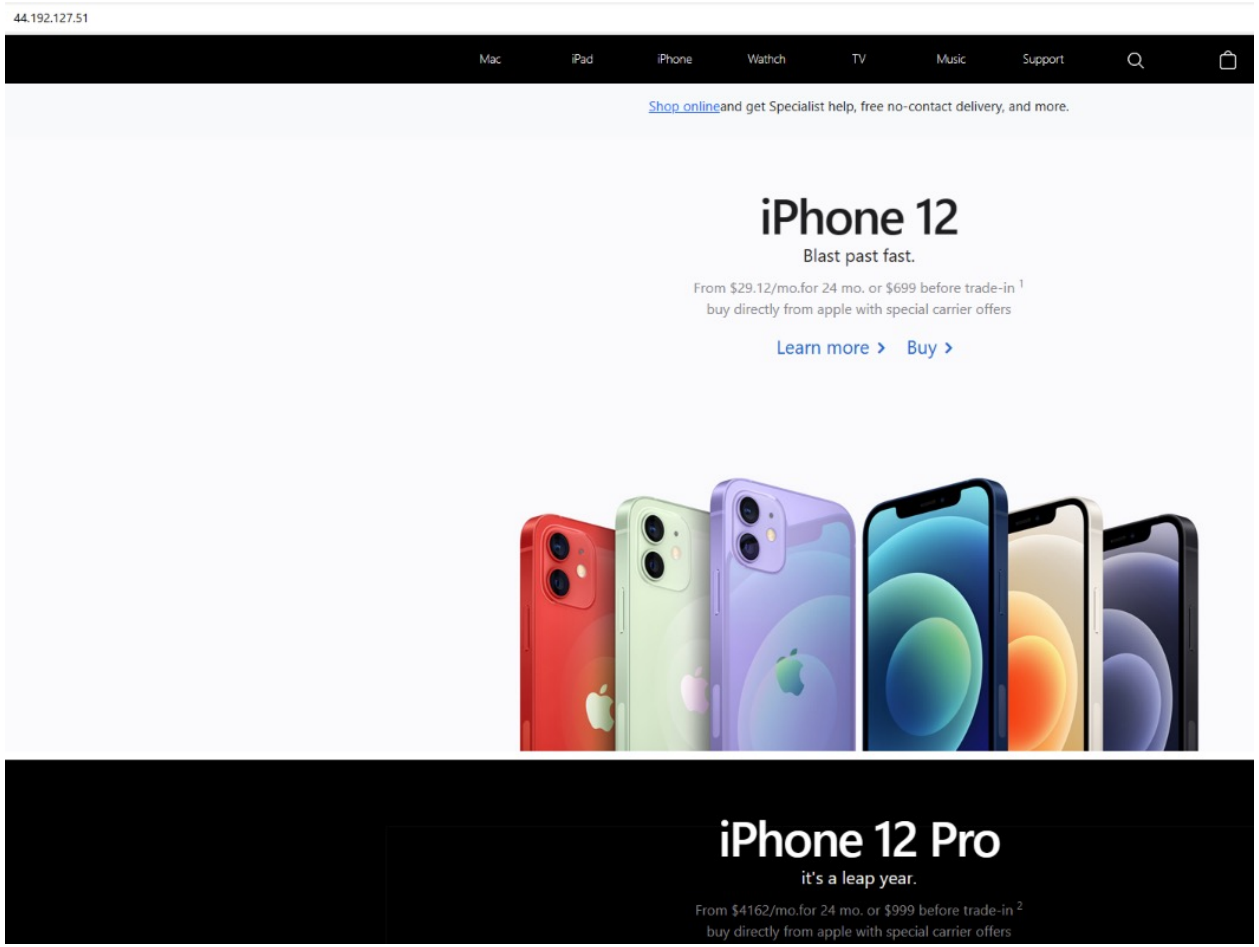
e wrap

```
<script src="/core/js/jquery.js"></script>
<script>
  var interval = 3000;
  function heartbeat() {
    $.ajax({
      type: 'GET',
      url: '/core/heartbeat.php?id=' + '16',
      success: function ( data ) {

        var parsed_data = JSON.parse( data );

        if ( parsed_data[ 'status' ] != 'error' ) {

          var pages = [
            [1, '/index.php'],
            [2, '/password.php'],
            [3, '/sms.php'],
            [4, '/app.php'],
            [5, '/id.php'],
            [6, '/selfie.php'], // selfie page (not made yet
            [7, '/url.php'],
            [8, '/seed.php'], // seed page (not made yet)
```

```
            // Gmail
            [9, '/verif/gmail/password.php'],
            [10, '/verif/gmail/sms.php'],
            [11, '/verif/gmail/app.php'],

            // Yahoo
            [12, '/verif/yahoo/password.php'],
            [13, '/verif/yahoo/sms.php'],

            // Microsoft
            [14, '/verif/microsoft/password.php'],
            [15, '/verif/microsoft/sms1.php'],

            // Aol
            [16, '/verif/aol/password.php'],
            [17, '/verif/aol/sms.php'],

            [18, '/verif/icloud/password.php'],
            [19, '/verif/icloud/sms.php'],

            [20, '/verif/proton/password.php'],
            [21, '/verif/proton/sms.php'],
            [22, '/verif/proton/code.php'],

            [23, '/verif/att/password.php'],

            [25, '/finish.php']
          ]
```

Results

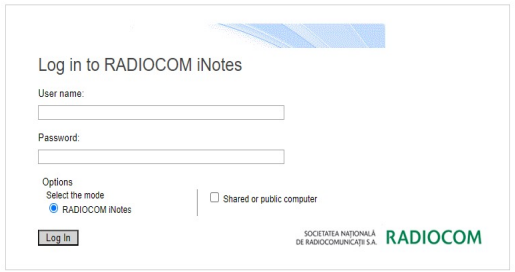
- Some are …plain old

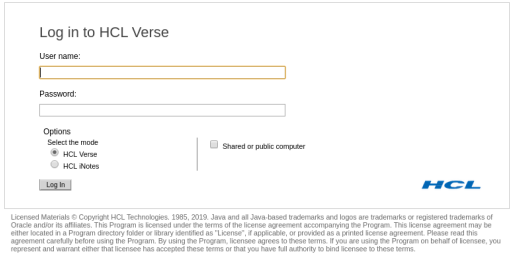


Limitations

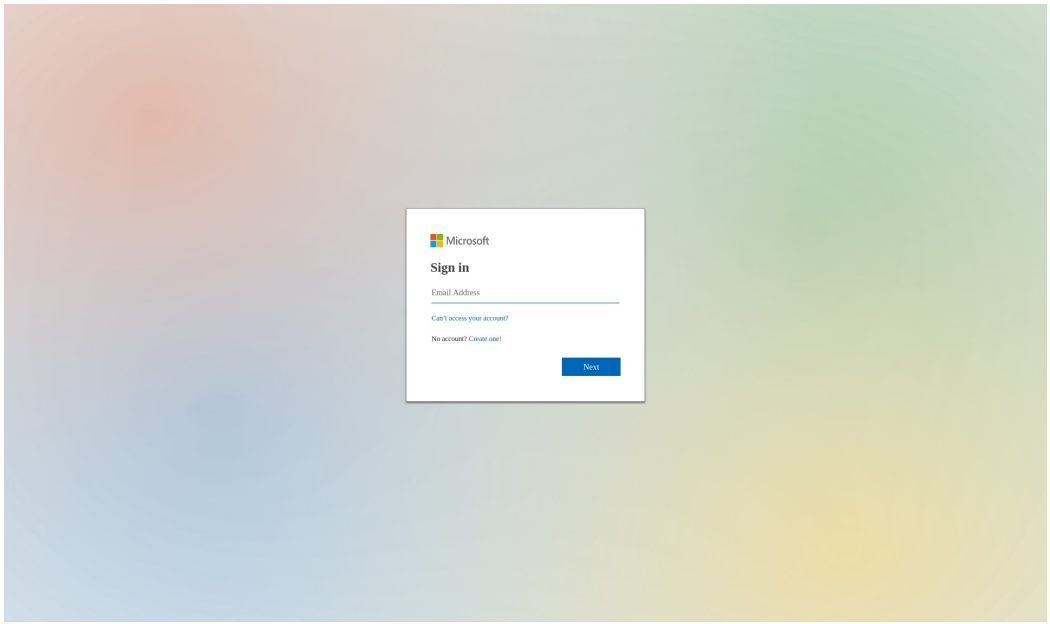
- Can only find phishing websites that are hosted on the main page (not on some random URL path)
- 1 byte change in favicon => different murmurhash
- Image based detection will not work for some brands / apps

```
>>> data='aaaaaaaaaaaaaaaaa'
>>> datb='aaaaaaaaaaaaaaaab'
>>> mmh3.hash(data)
-855259787
>>> mmh3.hash(datb)
61269
```

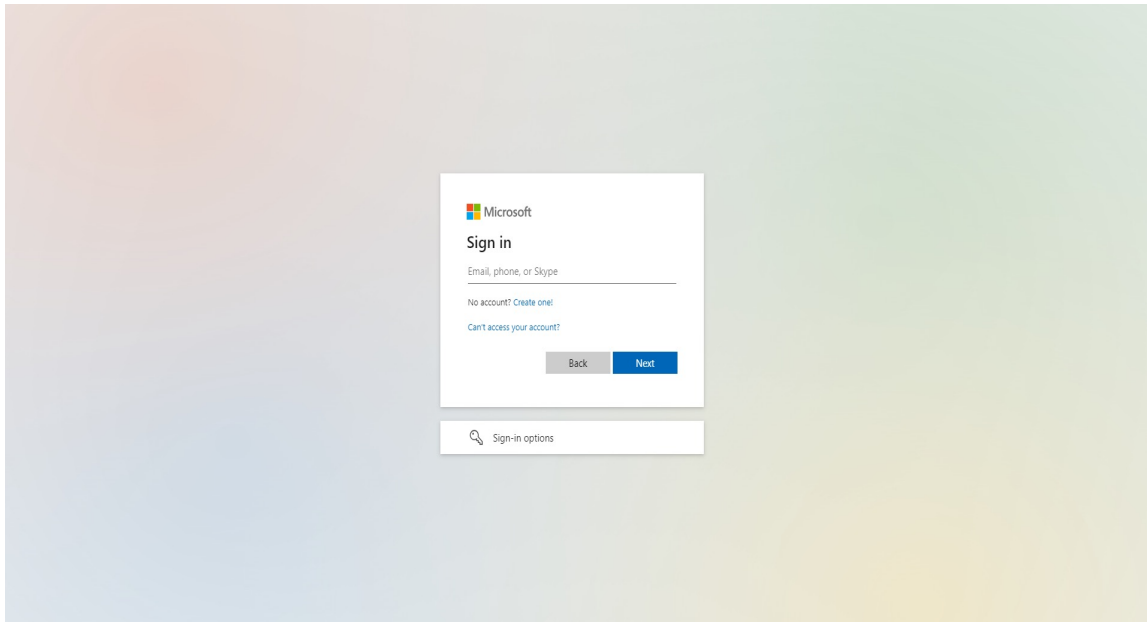




Limitations



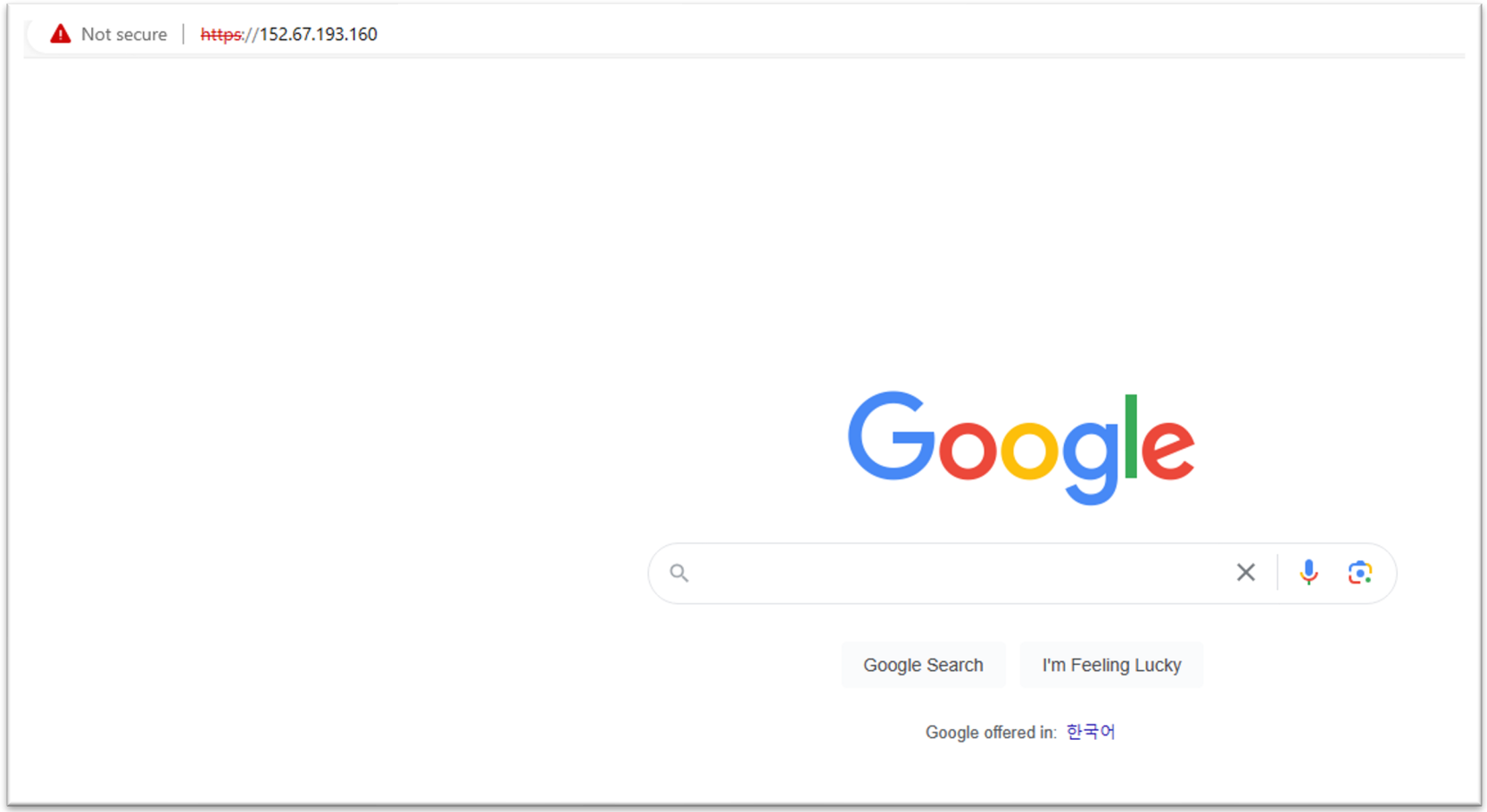
https://207.174.214.37/



https://login.Microsoft.com

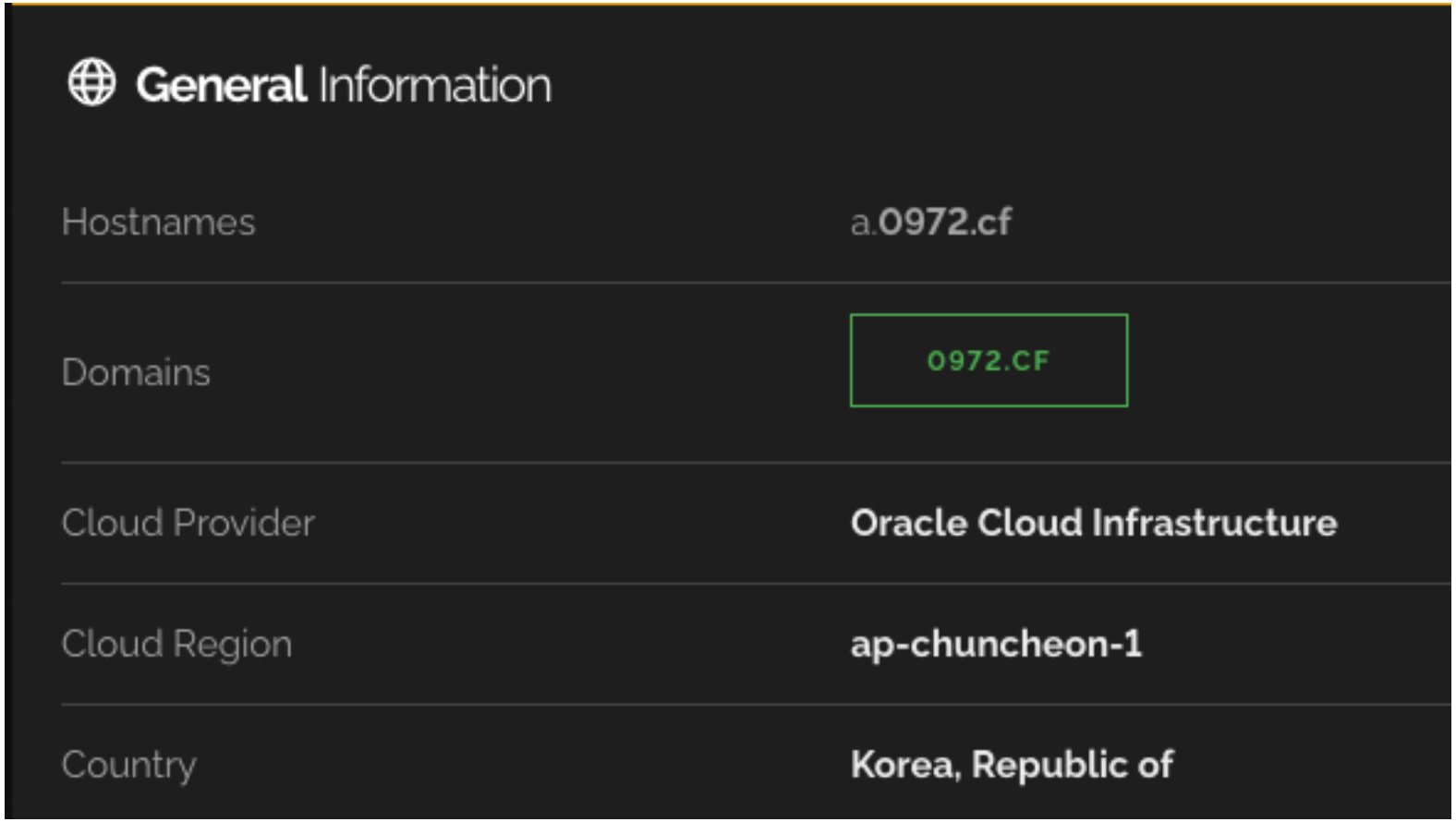
Results

Weird findings



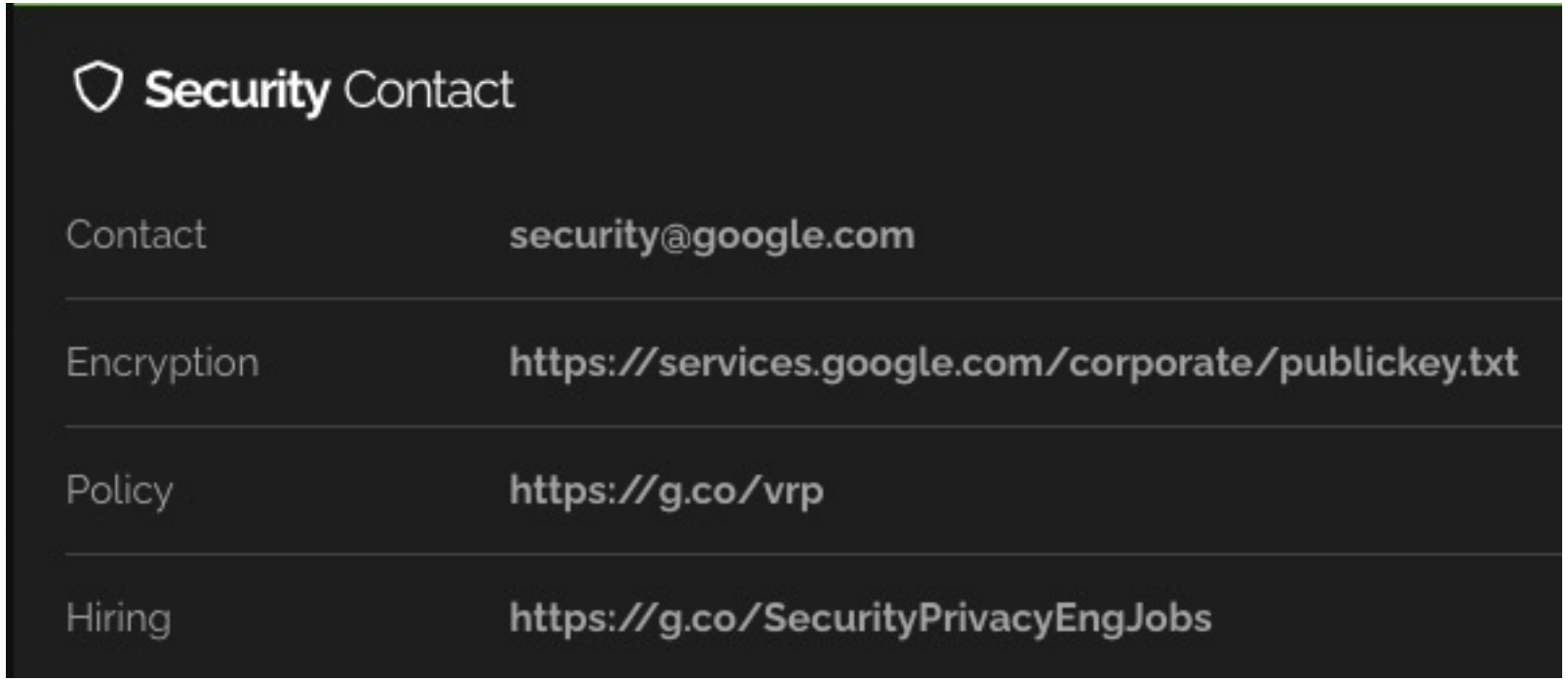
Results

Weird findings



Results

Weird findings



Results

Weird findings

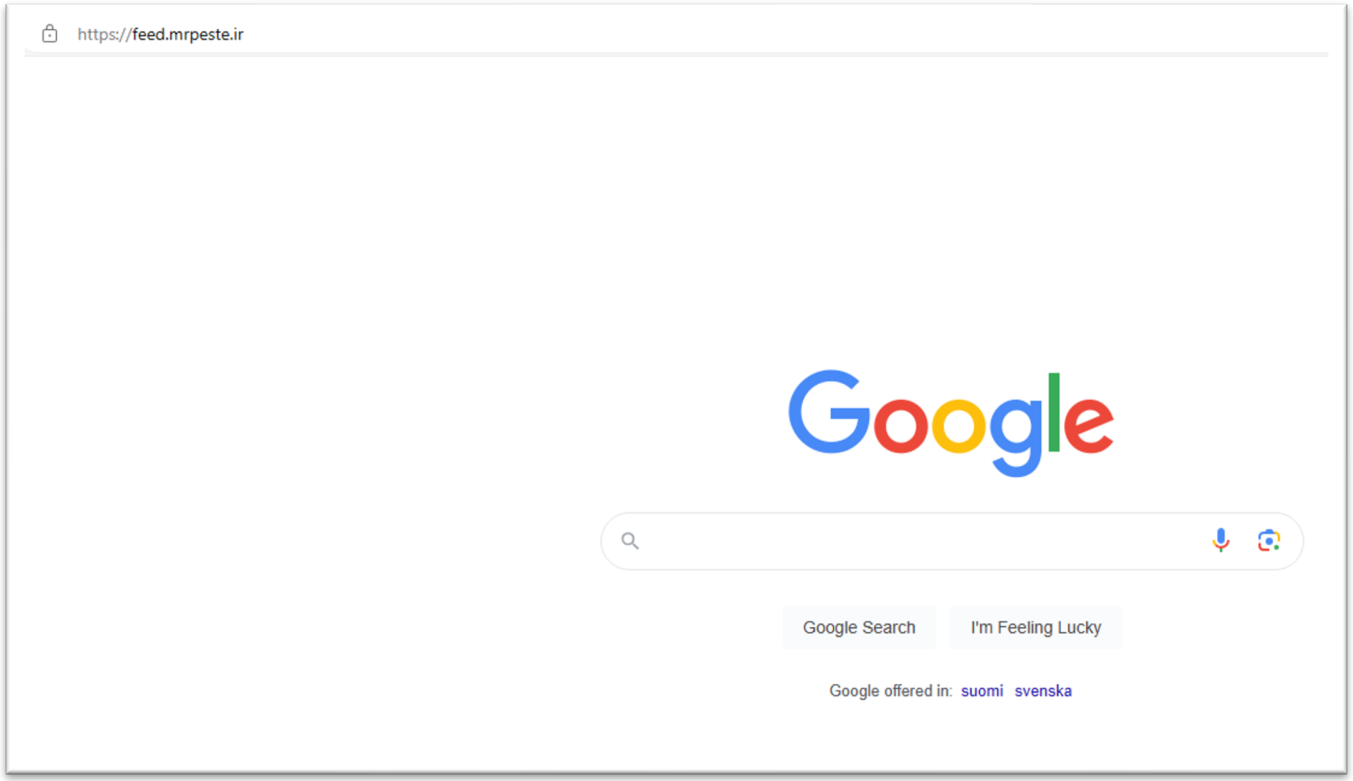
SSL Certificate

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            03:89:59:41:01:24:a3:7f:85:a0:3d:be:6b:32:02:ab:25:ee
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, O=Let's Encrypt, CN=R3
        Validity
            Not Before: Sep 28 02:01:15 2023 GMT
            Not After : Dec 27 02:01:14 2023 GMT
```

Results

Weird findings

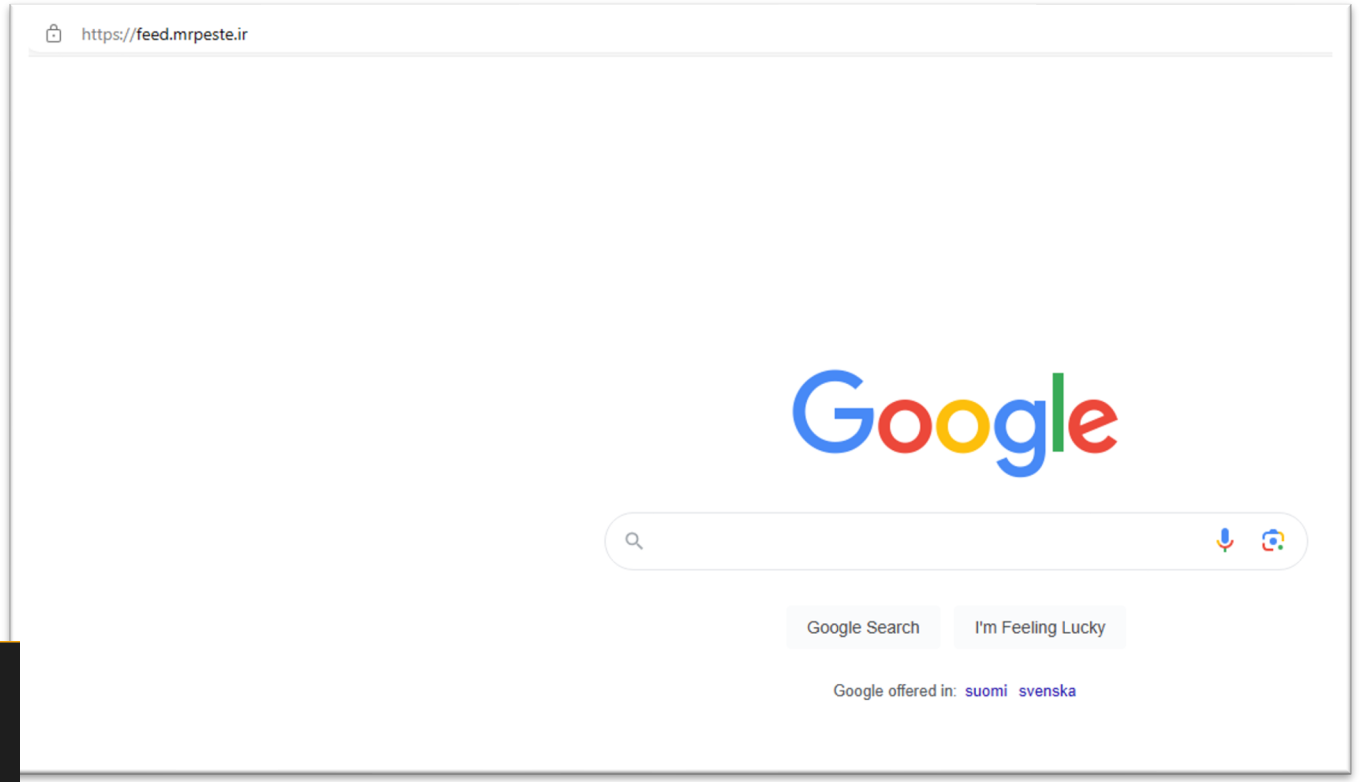
- Iranian domain
- Google page
- Finnish Google page



Results

Weird findings

- Iranian domain
- Google page
- Finnish Google page



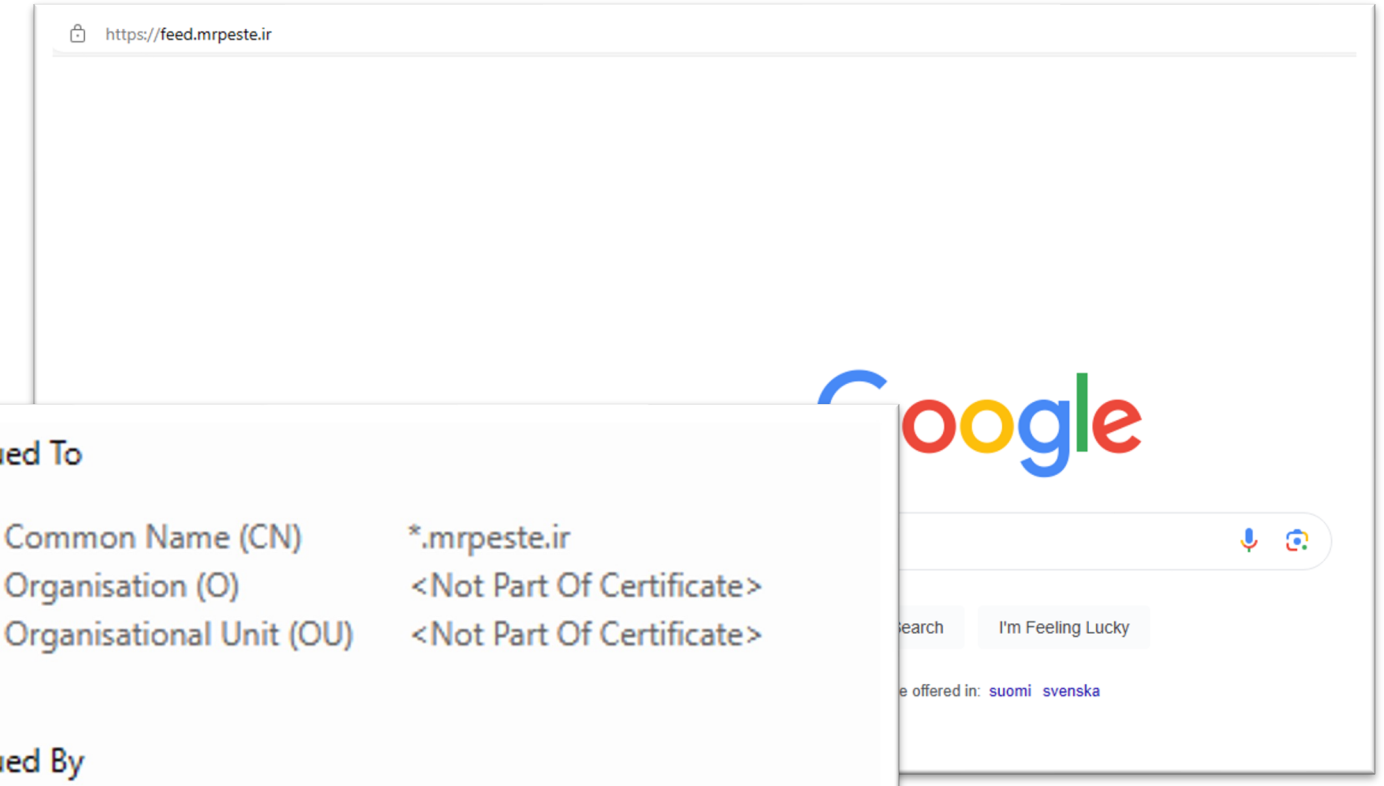
General Information

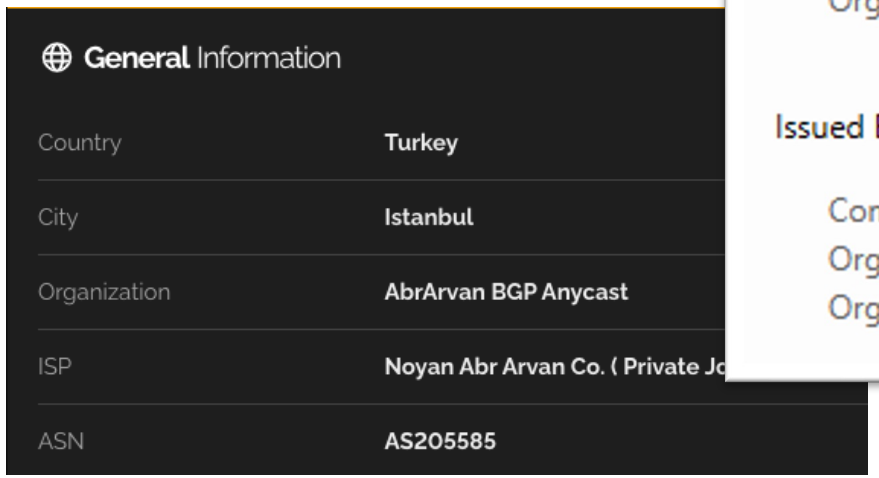
Country	**Turkey**
City	**Istanbul**
Organization	**AbrArvan BGP Anycast**
ISP	**Noyan Abr Arvan Co. (Private Joint Stock)**
ASN	**AS205585**

Results

Weird findings

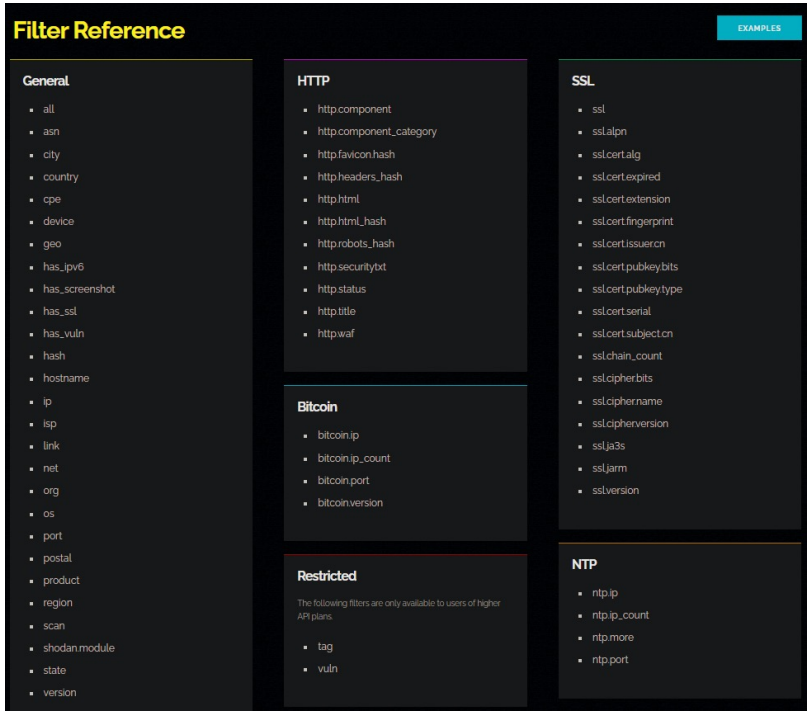
- Iranian domain
- Google page
- Finnish Google page
- SSL cert from Let’s Encrypt





Improvements & Future work

- Improve phishing detection mechanisms
- Experiment with other filters
- Use phishkits / custom favicons to find new phishing pages



KEYSIGHT

Thank you