3

BUSCADORES ESPECIALIZADOS

3.1 INTRODUCCIÓN

La seguridad informática es un campo en constante evolución, donde los atacantes utilizan cada vez más herramientas sofisticadas para explotar vulnerabilidades. Para mantenerse un paso adelante, los profesionales de la seguridad necesitan herramientas que les permitan descubrir y analizar las amenazas de manera proactiva. Los buscadores especializados como Shodan, Censys, BinaryEdge, Fofa y ZoomEye desempeñan un papel fundamental en este sentido.

Al indexar una gran cantidad de dispositivos y servicios en línea, estos buscadores permiten a los investigadores identificar sistemas vulnerables, rastrear la actividad de ciberatacantes y obtener una visión general de la superficie de ataque de una organización. Estas herramientas son indispensables para realizar pruebas de pentesting, análisis de riesgos y otras tareas relacionadas con la seguridad informática.

3.2 SHODAN

Shodan https://www.shodan.io corresponde al acrónimo de **Sentient Hyper Optimized Data Access Network**. A diferencia de los motores de búsqueda tradicionales que rastrean el sitio web para mostrar los resultados, Shodan ayuda a encontrar los servicios vulnerables en un servidor web.

Shodan es un motor de búsqueda para encontrar dispositivos y funciona escaneando y analizando los banners que devuelven los dispositivos. Utilizando esa información, Shodan puede devolver datos como qué servidor web (y versión) es más popular, o cuántos servidores FTP anónimos existen en una ubicación determinada.

Es de particular utilidad para la investigación de seguridad en Internet de las cosas, ya que actualmente podemos encontrar millones de dispositivos conectados a Internet que tienen vulnerabilidades específicas y pueden identificarse rápidamente mediante la información devuelta por el banner.

Cuando le pido a Shodan que me muestre todos los servidores Microsoft IIS versión 8 que se ejecutan en el puerto 8080 de TCP en China, me muestra los que el sistema tiene registrado, pero desde el punto de vista de privacidad del usuario ninguno de esos servidores sabe qué usuario es el que buscó esta información.

3.2.1 DIFERENCIAS ENTRE SHODAN Y GOOGLE

Se podría decir que Shodan es un buscador como Google, pero tiene una muy importante diferencia y es que Google va recorriendo Internet y cacheando o almacenando la información web, es decir la que se puede visualizar a través de un navegador.

Google trabaja a través de sus spiders, crawlers y robots recolectando información que va indexando, para luego organizarla y mostrarnos la misma en el momento que la necesitemos Shodan, sin embargo, recorre internet escaneando cada dirección IP, obteniendo los servicios en ejecución y los puertos que tiene abiertos, capturando el banner que le devuelve cada servicio. Esto nos permite realizar búsquedas en la información retornada por los servicios, de tal forma que permite encontrar, por ejemplo, todos los servidores apache de una versión en concreto.

3.2.2 FILTROS Y BÚSQUEDAS PERSONALIZADAS EN SHODAN

Shodan incorpora diferentes tipos de filtros cuales ayudan a realizar una búsqueda puntual valiéndose para ello de los metadatos que los dispositivos o servicios otorgan. Al igual que con cualquier motor de búsqueda, la potencia real viene con consultas personalizadas. Estos son los filtros de búsqueda que puede usar:

- **city**: buscar dispositivos en una ciudad en particular.
- **country**: buscar dispositivos en un país en particular.
- **geo**: puedes pasar las coordenadas latitud y longitud.
- **hostname**: busca valores que coincidan con el nombre de host.
- **net**: búsqueda basada en una dirección IP o rango de red.
- **os**: búsqueda basada en el sistema operativo.
- **port**: permite filtrar los resultados por puerto.

Ejemplos de búsquedas:

- **Servidores Apache en San Francisco**: apache city:“San Francisco”
- **Servidores Nginx en UK**: nginx country:“UK”
- **Buscar servidores GWS**: “Server: gws” hostname:“google

También es posible combinar filtros. Por ejemplo, si nos interesa buscar servidores Nginx en San Francisco, que se ejecutan en el puerto 8080, que también ejecutan Tomcat, podríamos realizar la siguiente búsqueda:

```
Apache city:“San Francisco” port:“8080” product:“Apache Tomcat/Coyote JSP
engine”
```

Entre las principales **funcionalidades** que ofrece el servicio podemos destacar:

- **Exportación de datos:** es posible exportar los resultados en varios formatos usando el menú superior al realizar una búsqueda.
- **Cuenta gratuita de Shodan:** es posible crear e iniciar sesión en su cuenta gratuita cuando realice una búsqueda.
- **Cuentas Premium:** una cuenta premium ofrece un mayor acceso a la API. Los detalles completos y los documentos están disponibles en https://developer.shodan.io.

3.2.3 SHODAN API REST

Con Shodan también tenemos disponible un API REST https://developer.shodan.io/api para realizar búsquedas, escaneos y consultas. Podríamos utilizar el API utilizando los endpoints que aparecen en la documentación y concatenando como parámetros el api key y la consulta a realizar. Por ejemplo, con el método de consulta **/shodan/host** podemos obtener la información de un host a partir de la dirección IP.

https://api.shodan.io/shodan/host/8.8.8.8?key=rszBpYTnMfVgHPKc9F50YY84iEgqoeEG&minify=True

```
{“region_code”: “CA”, “tags”: [], “ip”: 134744072, “area_code”: null, “domains”:
[“dns.google”], “hostnames”: [“dns.google”], “country_code”: “US”, “org”:
“Google LLC”, “data”: [], “asn”: “AS15169”, “city”: “Mountain View”, “latitude”:
37.4056, “isp”: “Google LLC”, “longitude”: -122.0775, “last_update”: “2025-01-
12T07:06:17.055902”, “country_name”: “United States”, “ip_str”: “8.8.8.8”, “os”:
null, “ports”: [443, 53]}
```

En este punto podríamos desarrollar un script en Python en el que a partir de una dirección IP obtenga esta información en formato json. Para probarlo habría que añadir nuestra API_KEY en la línea SHODAN_API_KEY.

shodan_info_ip.py

```
#!/usr/bin/env python

import requests

SHODAN_API_KEY = “SHODAN_API_KEY”
ip = ‘8.8.8.8’
```

```
def ShodanInfo(ip):
    try:
        result = requests.get('https://api.shodan.io/shodan/
host/'+ip+'?key='+SHODAN_API_KEY+'&minify=True').json()
    except Exception as exception:
        result = {"error":"Informacion no disponible."}
    return result

print(ShodanInfo(ip))
```

3.2.4 UTILIDADES SHODAN

Basado en su función principal como motor de búsqueda, Shodan nos ofrece distintas utilidades entre los cuales se destacan los siguientes:

- **Shodan explore** https://www.shodan.io/explore ofrece un acceso directo a búsquedas y categorías predefinidas.

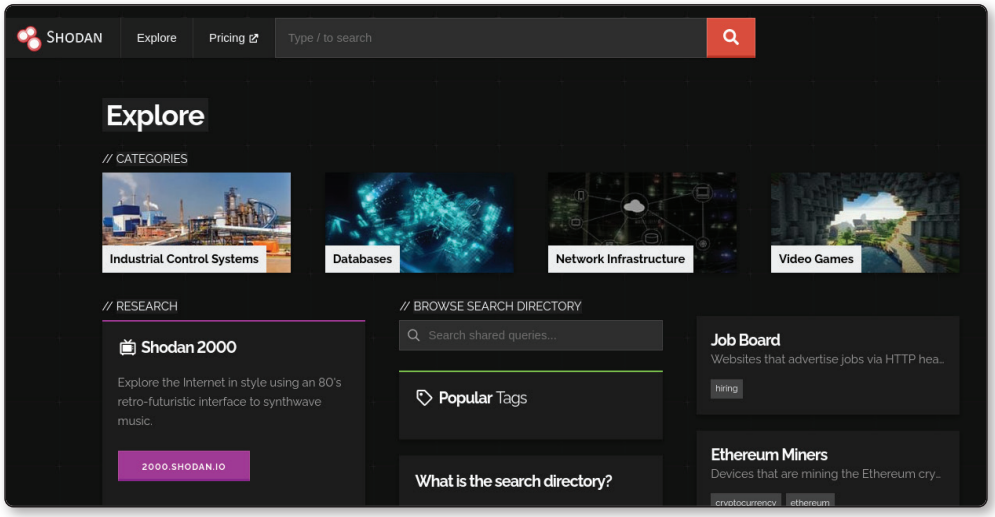

Figura 3.1. Shodan explore

- **Shodan Exposure**: Shodan dispone de un dashboard https://exposure.shodan.io donde podemos ver un resumen de los puertos y servicios expuestos para un país concreto. Por ejemplo podemos ver la cantidad de ICS (Sistema de Control Industrial) que están expuestos en internet para un país concreto.

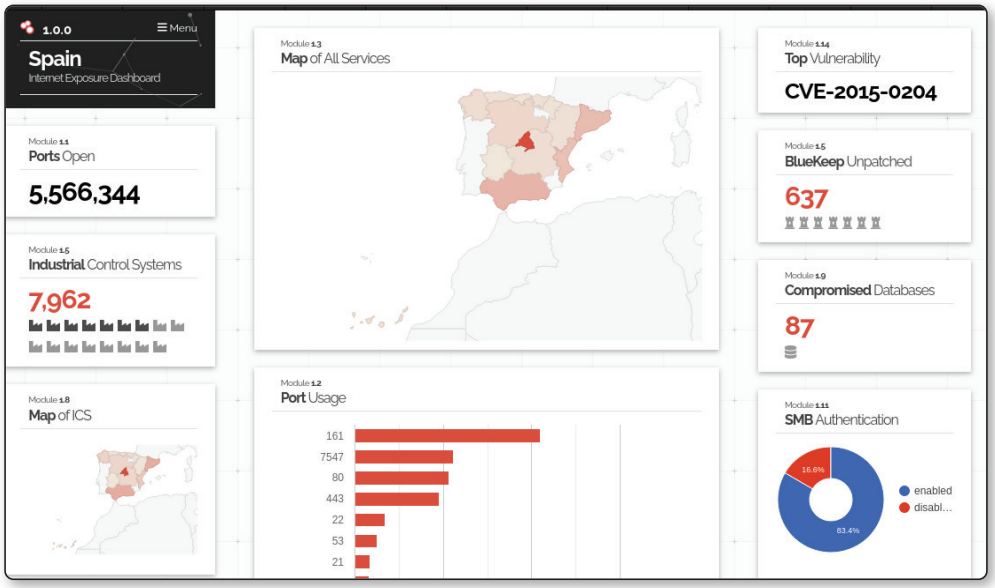

Figura 3.2. Shodan exposure

- **FireShodanMap** https://github.com/Warflop/FireShodanMap es un mapa en tiempo real que integra Firebase y Shodan. Se realiza una búsqueda utilizando Shodan buscando dispositivos vulnerables y se muestran en el mapa para su análisis. Todos los datos actualizados en Firebase son en tiempo real.
- **HoneyScore** https://honeyscore.shodan.io permite comprobar si una determinada IP es un honeypot, es decir, tiene un mecanismo configurado para detectar los intentos de uso y acceso no autorizado a los sistemas.
- **Shodan Monitor** https://monitor.shodan.io es una herramienta diseñada para permitir a sus usuarios llevar un seguimiento directo de sus dispositivos, recibiendo recomendaciones de seguridad y notificaciones en tiempo real cuando uno de sus dispositivos se exponga en la red. Shodan Monitor actúa a grandes rasgos como un escáner de puertos masivo que podemos integrar fácilmente con las conocidas herramientas Nmap, Metasploit, Maltego y FOCA, así como usarlo directamente desde Google Chrome y Firefox.
- **Shodanploit** es un script en Python https://github.com/ismailtasdelen/shodanploit que contiene todas las llamadas a la api de shodan desde el terminal. Con esta herramienta se puede tener todas las llamadas que hacemos a Shodan desde nuestra terminal. Lo único que tenemos que hacer es añadir la API Key de Shodan. La herramienta también la tenemos disponible en el repositorio Docker Hub donde encontramos la imagen que permite ejecutarlo https://hub.docker.com/r/ismailtasdelen/shodansploit

```
$ docker run --rm -it ismailtasdelen/shodansploit                                ✓
Unable to find image 'ismailtasdelen/shodansploit:latest' locally
latest: Pulling from ismailtasdelen/shodansploit
e7c96db7181b: Pull complete
df46343d112d: Pull complete
377f42e2152c: Pull complete
26b93f1788d5: Pull complete
e6f344bed9dd: Pull complete
Digest: sha256:8e422be760aa85a0d114c013fc5a11b4bf38b4476deb05c71e14d82f7
6b04965
Status: Downloaded newer image for ismailtasdelen/shodansploit:latest
[*] Please enter a valid Shodan.io API Key:
  ___| |__   ___   __| | __ _ _ __  ___ _ __ | | ___ (_) |_
 / __| '_ \ / _ \ / _` |/ _` | '_ \/ __| '_ \| |/ _ \| | __|
 \__ \ | | | (_) | (_| | (_| | | | \__ \ |_) | | (_) | | |_
 |___/_| |_|\___/ \__,_|\__,_|_| |_|___/ .__/|_|\___/|_|\__|
                                       |_|          v1.2.0
    Author : Ismail Tasdelen
    GitHub : github.com/ismailtasdelen
    Linkedin : linkedin.com/in/ismailtasdelen
     Twitter : twitter.com/ismailtsdln


[1] GET > /shodan/host/{ip}
[2] GET > /shodan/host/count
[3] GET > /shodan/host/search
[4] GET > /shodan/host/search/tokens
[5] GET > /shodan/ports

[6] GET > /shodan/exploit/author
[7] GET > /shodan/exploit/cve
[8] GET > /shodan/exploit/msb
[9] GET > /shodan/exploit/bugtraq-id
[10] GET > /shodan/exploit/osvdb
[11] GET > /shodan/exploit/title
[12] GET > /shodan/exploit/description
[13] GET > /shodan/exploit/date
[14] GET > /shodan/exploit/code
[15] GET > /shodan/exploit/platform
[16] GET > /shodan/exploit/port

[17] GET > /dns/resolve
[18] GET > /dns/reverse
[19] GET > /labs/honeyscore/{ip}

[20] GET > /account/profile
[21] GET > /tools/myip
[22] GET > /tools/httpheaders
[23] GET > /api-info
```

Al usar la herramienta nos va solicitando desde la consola aquella información que necesitamos para realizar las consultas. Por ejemplo, realizar consultas a partir de la dirección IP.

```
Which option number : 1
Shodan Host Search : 8.8.8.8
{
  “area_code”: null,
  “asn”: “AS15169”,
  “city”: “Mountain View”,
  “country_code”: “US”,
  “country_name”: “United States”,
  “data”: [
    {
      “_shodan”: {
        “crawler”: “573d97bfe40e7570d3425ce781e4e461309e892d”,
        “id”: “460ffeed-08b1-4b5d-b829-687e4e58e8da”,
        “module”: “dns-tcp”,
        “options”: {},
        “region”: “na”
      },
      “asn”: “AS15169”,
      “data”: “\nRecursion: enabled”,
      “dns”: {
        “recursive”: true,
        “resolver_hostname”: null,
        “resolver_id”: null,
        “software”: null
      },
      “domains”: [
        “dns.google”
      ],
      “hash”: -553166942,
      “hostnames”: [
        “dns.google”
      ],
      “ip”: 134744072,
      “ip_str”: “8.8.8.8”,
      “isp”: “Google LLC”,
      “location”: {
        “area_code”: null,
        “city”: “Mountain View”,
        “country_code”: “US”,
        “country_name”: “United States”,
        “latitude”: 37.4056,
        “longitude”: -122.0775,
        “region_code”: “CA”
      },
      “opts”: {},
      “org”: “Google LLC”,
      “os”: null,
      “port”: 53,
      “timestamp”: “2025-01-11T19:37:48.596168”,
      “transport”: “tcp”
    }
```

3.2.5 SHODAN CLI

Shodan CLI https://cli.shodan.io ofrece la posibilidad de ejecutar Shodan desde línea de comandos. También se necesita iniciar el programa con una API key que se obtiene al registrarse en el sitio web de Shodan. La interfaz de línea de comandos (CLI) de Shodan está empaquetada con la librería oficial de Python para Shodan, lo que significa que si estás ejecutando la última versión de la librería, deberías tener acceso a la consola. Para instalar la herramienta desde nuestra terminal lo podemos hacer con el comando:

```
$ pip install shodan
```

```
$ shodan
Usage: shodan [OPTIONS] COMMAND [ARGS]...

Options:
  -h, --help  Show this message and exit.

Commands:
  alert       Manage the network alerts for your account
  convert     Convert the given input data file into a different format.
  count       Returns the number of results for a search
  data        Bulk data access to Shodan
  domain      View all available information for a domain
  download    Download search results and save them in a compressed JSON...
  honeyscore  Check whether the IP is a honeypot or not.
  host        View all available information for an IP address
  info        Shows general information about your account
  init        Initialize the Shodan command-line
  myip        Print your external IP address
  org         Manage your organization's access to Shodan
  parse       Extract information out of compressed JSON files.
  radar       Real-Time Map of some results as Shodan finds them.
  scan        Scan an IP/ netblock using Shodan.
  search      Search the Shodan database
  stats       Provide summary information about a search query
  stream      Stream data in real-time.
  trends      Search Shodan historical database
  version     Print version of this tool.
```

Para inicializar la herramienta es necesario indicarle el API KEY con el comando:

```
$ shodan init <API_key>
```

El comando "**search**" permite buscar en Shodan y visualizar los resultados en la terminal de manera amigable. Por defecto muestra la dirección IP, puerto, nombres del host y datos. Se puede utilizar el parámetro "--fields" para imprimir los campos en los cuales estamos interesados.

```
$ shodan search --fields ip_str,port,org,hostnames apache tomcat
```

También podríamos utilizar el API de python para obtener esta información de forma programática. Para buscar información sobre una determinada dirección IP usaremos el método **Shodan.host()**.

shodan_info.py

```
import shodan
api = shodan.Shodan('API_KEY')
info = api.host('8.8.8.8')
print(info)
```

El código anterior solicita información sobre la resolución DNS 8.8.8.8 de Google y la almacena en la variable info. El resultado de **api.host()** contiene información sobre los servicios que ejecuta, dónde se encuentra y el proveedor de alojamiento. Entre las propiedades que devuelve la llamada podemos destacar:

- **data**: devuelve una lista de banners que proporcionan detalles sobre los servicios que están en ejecución en ese host.
- **port**: obtiene una lista de puertos abiertos para la dirección ip proporcionada.
- **tags**: Shodan hace validación extra para algunos servicios/dispositivos y tiene etiquetas especiales para facilitar la identificación de ciertos tipos de dispositivos (por ejemplo, la etiqueta "ics" nos permite identificar sistemas de control industrial).

El siguiente ejemplo combina el uso del API REST junto con el módulo de Shodan en Python para realizar una resolución de nombres con el API donde se obtiene la dirección IP a partir del nombre de dominio y posteriormente obtiene información a partir de la dirección IP utilizando el método **api.host()** que ofrece la librería de Shodan.

shodan_info_host.py

```
#!/usr/bin/env python

import shodan
import requests

SHODAN_API_KEY= "API_KEY"
api = shodan.Shodan(SHODAN_API_KEY)

dominio = 'www.google.com'

dnsResolve = 'https://api.shodan.io/dns/resolve?hostnames=' + dominio + '&key='
+ SHODAN_API_KEY
```

```
try:
    # Primero necesitamos resolver nuestro dominio a una IP
    resolved = requests.get(dnsResolve)
    print(resolved)
    hostIP = resolved.json()[dominio]


    # Entonces necesitamos hacer una busqueda de Shodan en esa IP
    host = api.host(hostIP)
    print("IP: %s" % host['ip_str'])
    print("Organization: %s" % host.get('org', 'n/a'))
    print("Operating System: %s" % host.get('os', 'n/a'))


    # Imprimir todos los banners
    for item in host['data']:
        print("Port: %s" % item['port'])
        print("Banner: %s" % item['data'])

except shodan.APIError as exception:
        print('Error: %s' % exception)
```

Ejecución:

```
<Response [200]>
IP: 142.251.40.196
Organization: Google LLC
Operating System: None
Port: 80
Banner: HTTP/1.1 301 Moved Permanently
Location: http://www.google.com/
Content-Type: text/html; charset=UTF-8
Content-Security-Policy-Report-Only: object-src 'none';base-uri 'self';script-
src 'nonce-41MsVxdDS3SQdcH3p6D7KQ' 'strict-dynamic' 'report-sample' 'unsafe-
eval' 'unsafe-inline' https: http:;report-uri https://csp.withgoogle.com/csp/
gws/other-hp
Permissions-Policy: unload=()
Date: Sat, 11 Jan 2025 15:08:23 GMT
Expires: Mon, 10 Feb 2025 15:08:23 GMT
Cache-Control: public, max-age=2592000
Server: gws
Content-Length: 219
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN


Port: 443
Banner: HTTP/1.1 301 Moved Permanently
```

```
Location: http://www.google.com/
Content-Type: text/html; charset=UTF-8
Content-Security-Policy-Report-Only: object-src 'none';base-uri 'self';script-
src 'nonce-NlPoqqICWryMo2pyZFV5xA' 'strict-dynamic' 'report-sample' 'unsafe-
eval' 'unsafe-inline' https: http:;report-uri https://csp.withgoogle.com/csp/
gws/other-hp
Cross-Origin-Opener-Policy: same-origin-allow-popups; report-to="gws"
Report-To: {"group":"gws","max_age":2592000,"endpoints":[{"url":"https://csp.
withgoogle.com/csp/report-to/gws/other"}]}
Permissions-Policy: unload=()
Date: Tue, 07 Jan 2025 22:00:11 GMT
Expires: Thu, 06 Feb 2025 22:00:11 GMT
Cache-Control: public, max-age=2592000
Server: gws
Content-Length: 219
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
```

3.3 CENSYS

Del mismo modo que ocurre con Shodan, con Censys es posible realizar búsquedas muy concretas sobre dispositivos, servidores y redes que se encuentran en Internet, siendo un servicio muy similar y directa competencia de Shodan. El funcionamiento de Censys se basa en el uso de una herramienta llamada Zmap https://zmap.io, un escáner que permite realizar un escaneo completo contra el rango de direcciones IPv4 y que con los recursos de cómputo adecuados, permite el escaneo de Internet en un tiempo bastante razonable https://censys.com/es/internet-map. Al final con Censys tenemos instantáneas con periodicidad de cómo los equipos y sitios web están configurados. Para interactuar con Censys se puede hacer a través de:

- Buscador en la página web.
- API para integraciones de plugins.
- Descargando en modo "Raw data" las bases de datos, las cuales son ficheros JSON bastante grandes con gigas de información.

3.3.1 ARQUITECTURA DE CENSYS

Se puede encontrar más información acerca de la arquitectura sobre la que Censys se encuentra montado y sus funcionalidades en el artículo que sus autores publicaron, con el título "**A Search Engine Backed by Internet-Wide Scanning**", Durumeric, Zakir, David Adrian, Ariana Mirian, Michael Bailey and J. Alex Halderman. "A Search Engine Backed by Internet-Wide Scanning." Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (2015).

La arquitectura de Censys tiene el concepto de Worker y el scheduler es el elemento que reparte el trabajo entre los diferentes Workers. En el caso de Censys, el Worker realiza operaciones de fingerprinting a través de la ejecución de ZMap, reportando los resultados en una base de datos Google Cloud Storage.

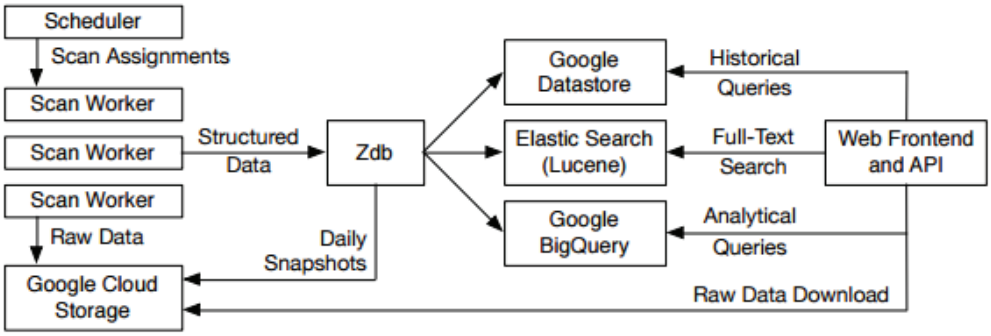

Figura 3.3. Arquitectura de Censys

3.3.2 OPCIONES DEL BUSCADOR

Desde el propio buscador https://search.censys.io/ podemos elegir qué tipo de búsqueda realizar, por ejemplo, si queremos realizar una búsqueda sobre los resultados de los escaneos sobre máquinas Hosts podemos elegir la opción "**Hosts**".

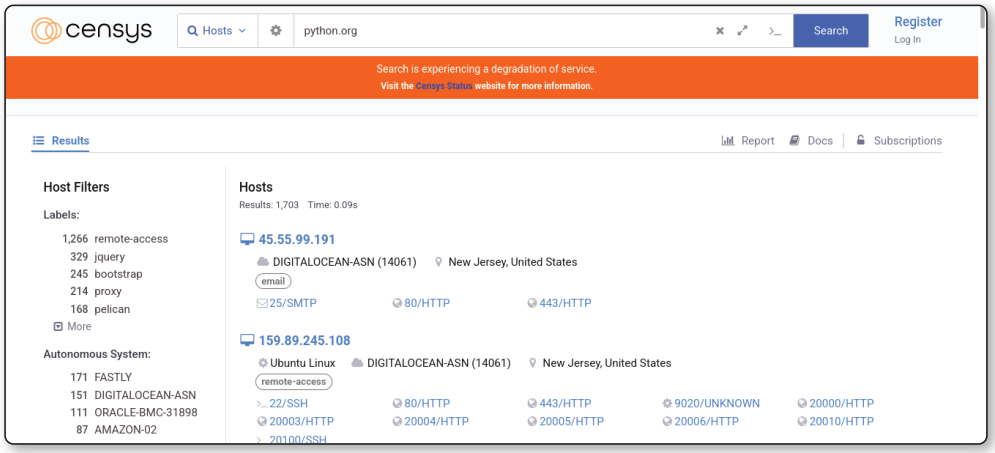

Figura 3.4. Búsqueda de hosts en Censys

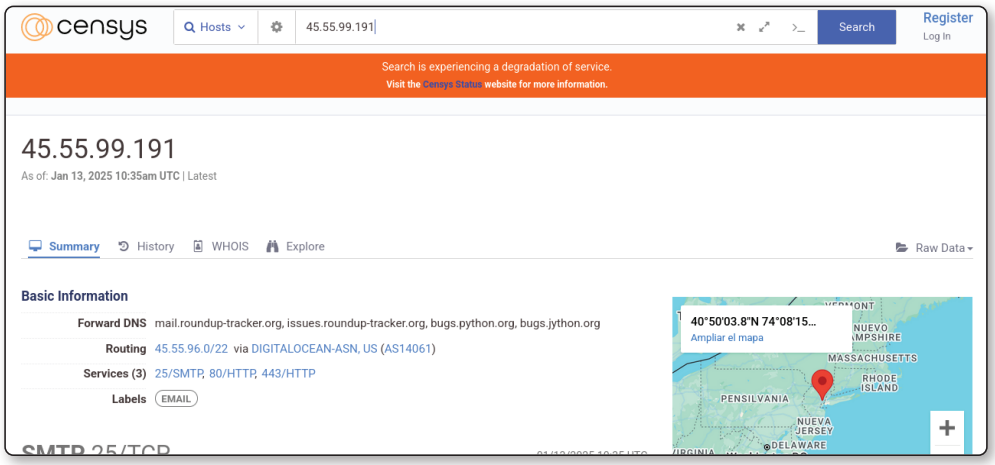

Figura 3.5. Detalle de un host en Censys

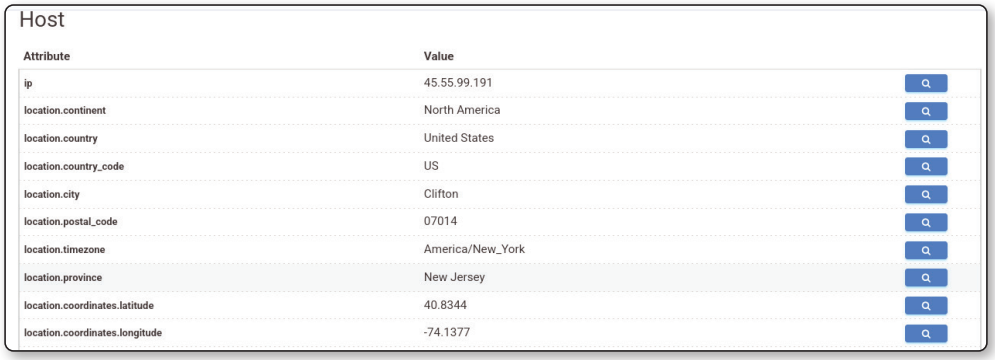

Figura 3.6. Detalle de un host en Censys

3.3.3 FILTROS EN CENSYS

Censys permite filtrar por campos específicos e introducir en el campo de búsqueda los siguientes criterios como realizar la búsqueda por un rango de direcciones IP o la búsqueda por nombre de servicio y puerto:

```
ip: {“23.0.0.0/8”, “8.8.8.0/24”}
```

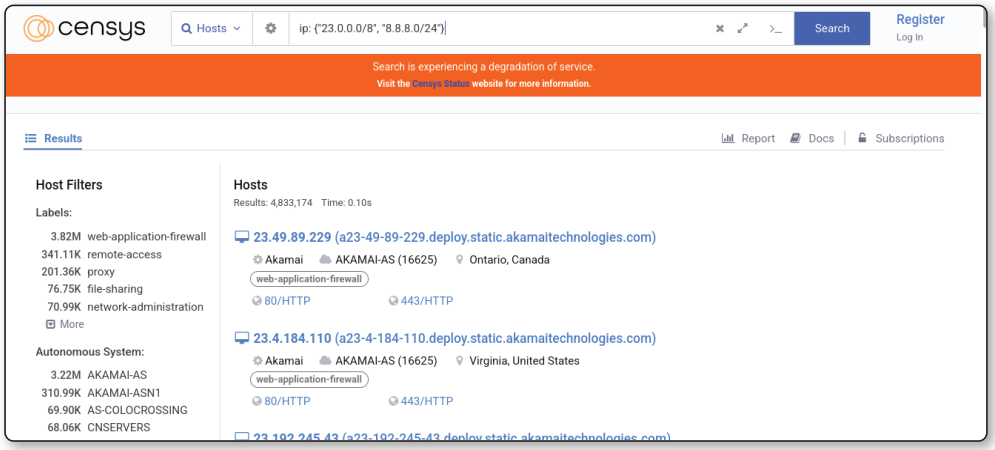

Figura 3.7. Búsqueda por rango de direcciones IP en Censys

```
services: (service_name: SSH and not port: {22, 2222})
```

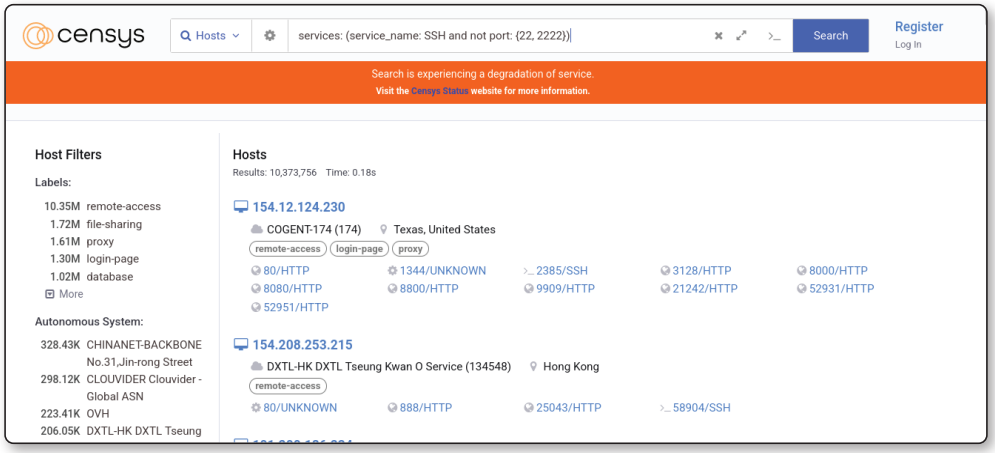

Figura 3.8. Búsqueda de servidores SSH en Censys

3.3.4 CENSYS API REST

Del mismo modo que ocurre con Shodan, es posible utilizar una API basada en servicios REST que pueden ser consultados fácilmente si se tiene una developer key, la cual se puede conseguir simplemente creando una cuenta en el servicio. Censys dispone de un API https://search.censys.io/api que proporciona una serie de endpoints. Dado que se trata de un servicio que se expone por medio de una API REST, es posible utilizar

cualquiera de las librerías disponibles en lenguajes de programación como Python, Ruby, C o Java, para consumir dichos servicios y endpoints.

Hay que tener en cuenta que de acuerdo a la documentación de la API REST de Censys, es necesario invocar a cada uno de los endpoints con un método HTTP concreto, una serie de parámetros obligatorios y que además, muchas de las peticiones reciben datos en formato JSON, con lo cual es necesario realizar las peticiones HTTP siguiendo estás especificaciones.

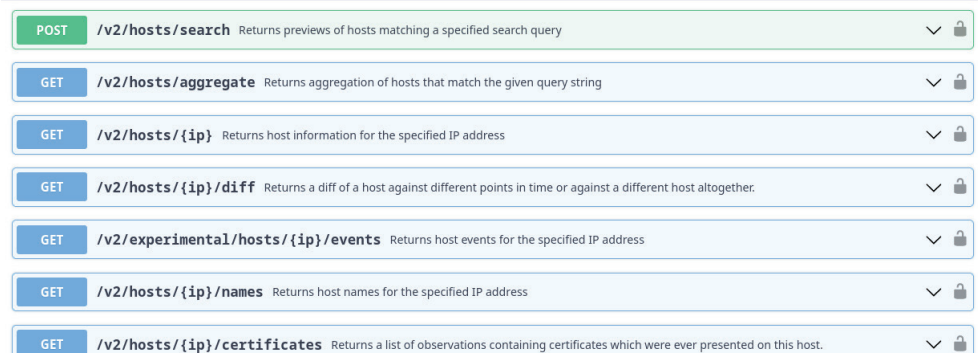

Figura 3.9. Operaciones disponibles en el API de Censys

En el siguiente ejemplo se obtiene la respuesta de la búsqueda para una determinada dirección IP:

```
{
  “code”: 200,
  “status”: “OK”,
  “result”: {
    “query”: “service.service_name: HTTP”,
    “total”: 50000,
    “hits”: [
      {
        “ip”: “8.8.8.8”,
        “services”: [
          {
            “port”: 53,
            “service_name”: “DNS”,
            “transport_protocol”: “UDP”
          },
          {
            “port”: 443,
            “service_name”: “HTTP”,
            “transport_protocol”: “TCP”,
            “certificate”:
“e58e89a726d80bb0219b218c3ab9d818b4be75d77959508400d660ebe1c1be3d”
          }
```

```
        ],
        "location": {
          "continent": "North America",
          "country": "United States",
          "country_code": "US",
          "postal_code": "48104",
          "timezone": "America/Michigan",
          "coordinates": {
            "latitude": "42.273",
            "longitude": "-83.751"
          },
          "registered_country": "United States",
          "registered_country_code": "US"
        },
        "autonomous_system": {
          "asn": 15169,
          "description": "GOOGLE",
          "bgp_prefix": "8.8.8.0/24",
          "name": "GOOGLE",
          "country_code": "US"
        }
      },
      {
        "name": "dns.google",
        "ip": "8.8.8.8",
        "services": [
          {
            "port": 53,
            "service_name": "DNS",
            "transport_protocol": "UDP"
          },
          {
            "port": 443,
            "service_name": "HTTP",
            "transport_protocol": "TCP",
            "certificate":
"e58e89a726d80bb0219b218c3ab9d818b4be75d77959508400d660ebe1c1be3d"
          }
        ],
        "location": {
          "continent": "North America",
          "country": "United States",
          "country_code": "US",
          "postal_code": "48104",
          "timezone": "America/Michigan",
          "coordinates": {
            "latitude": "42.273",
            "longitude": "-83.751"
          },
          "registered_country": "United States",
          "registered_country_code": "US"
```

```
        },
        "autonomous_system": {
          "asn": 15169,
          "description": "GOOGLE",
          "bgp_prefix": "8.8.8.0/24",
          "name": "GOOGLE",
          "country_code": "US"
        }
      }
    ]
  },
  "links": {
    "prev": "prevCursorToken",
    "next": "nextCursorToken"
  }
}
```


Censys dispone de una API escrita en Python https://github.com/censys/censys-python que podemos usar para ejecutar consultas contra los conjuntos de datos.

```
$ pip install censys
```

Con Censys podemos realizar la búsqueda de hosts y de subdominios.


```
$ censys search 'services.http.response.html_title=`Home Assistant` ' --index-
type hosts
     {
    "ip": "208.30.106.171",
    "services": [
        {
            "port": 443,
            "service_name": "HTTP",
            "certificate":
"c63cee2b816d7fcae974ce1d228650a492f45e771230ab7b748bc3df6694f6a3",
            "transport_protocol": "TCP"
        },
        {
            "port": 8888,
            "service_name": "HTTP",
            "certificate":
"c8054d622167443810d8827cb0fc703593ee995d3975da444aeef6178aa17ad9",
            "transport_protocol": "TCP"
        }
    ],
    "location": {
        "continent": "North America",
        "country": "U.S. Virgin Islands",
        "country_code": "VI",
        "timezone": "America/St_Thomas",
        "province": "Saint Croix Island",
        "coordinates": {
            "latitude": 17.7325,
```

```
            "longitude": -64.7376
        },
        "registered_country": "U.S. Virgin Islands",
        "registered_country_code": "VI"
    },
    "autonomous_system": {
        "asn": 14434,
        "description": "VIPNAS1",
        "bgp_prefix": "208.30.106.0/23",
        "name": "VIPNAS1",
        "country_code": "VI"
    },
    "last_updated_at": "2024-06-08T14:08:04.702Z"
}
$ censys subdomains censys.io --max-records 10
Found 3 unique subdomain(s) of censys.io
- nodomain.censys.io
- www.censys.io
- censys.io
```

El proyecto Censys también tiene la capacidad de obtener certificados SSL/TLS de múltiples fuentes. Una de las técnicas utilizadas es probar todas las máquinas en el espacio público de direcciones IPv4 en el puerto 443. Censys proporciona una manera de correlacionar el certificado SSL/TLS recopilado con los hosts IPv4 que proporcionaron el certificado. Podríamos escribir un script de Python que se conecte a la API de Censys para realizar consultas de certificados SSL/TLS para un dominio determinado y extraer los subdominios y direcciones de correo electrónico que pertenecen al dominio.

censys_search.py

```
import censys.search as c
import re

# Autenticación
c.search_api_id = "YOUR_CENSYS_API_ID"
c.search_api_secret = "YOUR_CENSYS_API_SECRET"

# Consulta
query = "services:https AND 443.tcp.https.certificate.parsed.names:example.com"

# Ejecutar consulta y extraer datos
results = c.SearchCursor(query)
for result in results:
    subdomains = result['443.tcp.https.certificate.parsed.names']
    email_pattern = r"\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b"
    emails = re.findall(email_pattern, str(result))
    print(f"IP: {result['ip']}")
    print(f"Subdominios: {subdomains}")
    print(f"Emails: {emails}")
    print("-"*20)
```