SHODAN API and CODING SKILLS

Laura García @ RootedCON2019

$ whoami

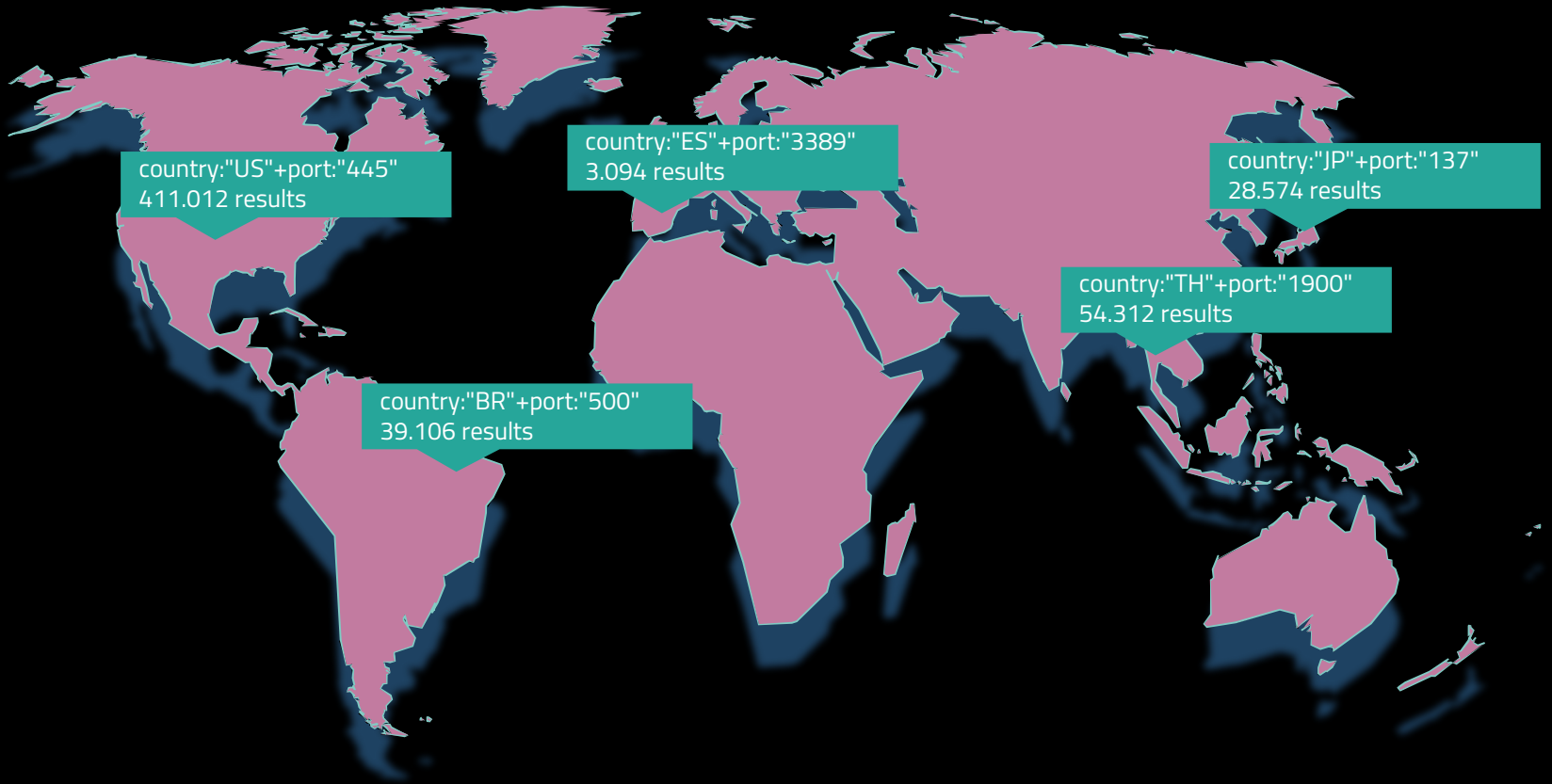
I am Laura García

Computer Engineer & Cybersec Master @Polytechnic_University_of_Madrid

Security Architect / Pentester @Deloitte_Hack_Team

Speaker at RootedCON Madrid 2016

The large number of assets published on the Internet, increase the probability of services exposed that could put them at risk.


country:"US"+port:"445"
411.012 results
country:"ES"+port:"3389"
3.094 results
country:"JP"+port:"137"
28.574 results
country:"TH"+port:"1900"
54.312 results
country:"BR"+port:"500"
39.106 results

SHODAN SEEKER

Business Logic

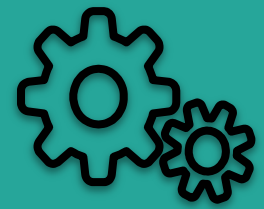
Fully Customizable

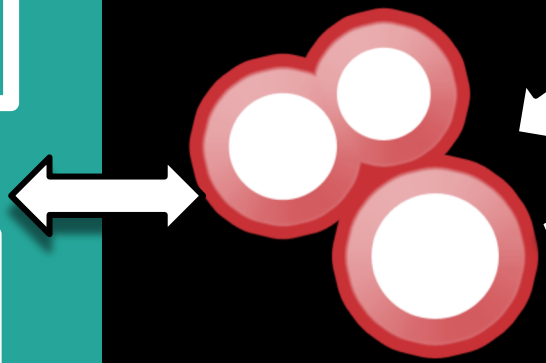
Reporting

Python

Mailing list

Logs





REST API

STREAMING API

Official Shodan Library for Python

```
shodan-seeker $ ./shodanseeker

Usage: python shodanseeker [options]

Options:
  -h, --help                    show this help message and exit
  --mail=MAIL                   Send email with results and alerts
  -a                            Attach csv results to an email

  Scanning Options:
                --si=SCANINPUT  Scan an IP/netblock
                --sf=SCANFILE   Scan an IP/netblock from file
                --force         Force Shodan to re-scan the provided IPs
                -l              List previously submitted scans

  Searching Options:
                -i GETINFO      Get all information of an IP/netblock
                -f GETINFOFROMFILE
                                Get all information of an IP/netblock from file
                --history       Get all Historical Banners
                --diff          Detect New Services Published
                --output=OUTPUT
                                Output results in csv format

  Monitoring in Real-Time:
                --ca=ADDALERT   Create network alerts for the IP/netblock
                --cf=ADDALERTFILE
                                Create network alerts from file
                --la            List of all the network alerts activated
                --da=DELALERT   Remove the specified network alert
                --subs=SUBSALERTS
                                Subscribe to the Private Horse Streaming
                --monport=MONPORT
                                Monitoring for High Risk Services
                --mondiff       Monitoring for New Services Published
                --montag=MONTAG
                                Tags (ex: compromised, doublepulsar, self-signed,...)
                --get=GET       Protocols, services, ports and tags supported
```

EXAMPLES:

```
./shodanseeker --si 'X.X.X.X Y.Y.Y.Y/24'                                  # Scan IPs/netblocks
./shodanseeker --sf 'pathfilename'                                        # Scan IPs/netblocks from a file
./shodanseeker -l                                                         # List previously submitted scans

./shodanseeker -i 'X.X.X.X Y.Y.Y.Y/24 Z.Z.Z.Z'                            # Get all information of IP/netblocks
./shodanseeker -f 'pathfilename'                                          # Get all information from a file of IPs/netblocks
./shodanseeker -i 'X.X.X.X' --history                                     # Get all historical banners
./shodanseeker -i 'X.X.X.X' --diff                                        # Detect new services published
./shodanseeker -f 'pathfilename' [--history|--diff] --output csv          # Output results in csv format
./shodanseeker -i 'X.X.X.X' --diff --output csv --mail toaddr -a          # Send email with csv results attached

./shodanseeker --ca Name 'X.X.X.X Y.Y.Y.Y/24'                             # Create network alerts for the IP/netblock
./shodanseeker --cf Name 'pathfilename'                                   # Create network alerts from file
./shodanseeker --la                                                       # List of all the network alerts activated on the account
./shodanseeker --da [alertid|all]                                         # Remove the specified network alert
./shodanseeker --subs [alertid|all] --monport '3389 22' [--mail toaddr]   # Subscribe to the Streaming for high risk services
./shodanseeker --subs [alertid|all] --mondiff [--mail toaddr]             # Subscribe to the Streaming for new services published
./shodanseeker --subs [alertid|all] --montag 'compromised' [--mail toaddr] # Subscribe to the Streaming and monitoring for tags
./shodanseeker --get [protocols|services|ports|tags]                      # List of (protocols,services,ports,tags) supported
```

Technical issues

- Request rate limit reached (1 request/second) in API calls (REST API).
  - sleep(0.5s)
- Connection between the script and the server got broken (Streaming API).
  - ./sh to respawn shodan-seeker.py script.
  - ChunkedEncodingError exception: call self.function.
- JSON output contains blank fields (“ports”).
- Shodan takes a few hours, since on-demand scanning is launched, to update its databases with the results.
- IPs found with open ports did not appear in their database.

Diffing implementation

Request:

GET/shodan/host/{ip}&history=true

Response:

ip	
last_update	
port1	timestamp1
port1	timestamp2
port2	timestamp3
port3	timestamp1
port3	timestamp3

list_port_uniq:

port1	port2	port3

list_timestamp_host_sort_uniq:

timestamp1
timestamp2
timestamp3

timestamp_adjustment:

timestamp_ajustment = datetime.now() + days=32

```
for port in port_uniq:
    var list_timestamp_port:
        timestamp1
        timestamp3

    for timestamp in list_timestamp_port:
        if (last_update == timestamp) and (date <= timestamp_adjustment):
            if (len (list_timestamp_port) == 1:
                print “diff port open”
            else:
                next_timestamp_port = list_timestamp_port[1]
                next_timestamp_host = list_timestamp_host_sort_uniq[1]
                if  (next_timestamp_port != next_timestamp_host):
                    print “diff port open”
```

Pros and Cons

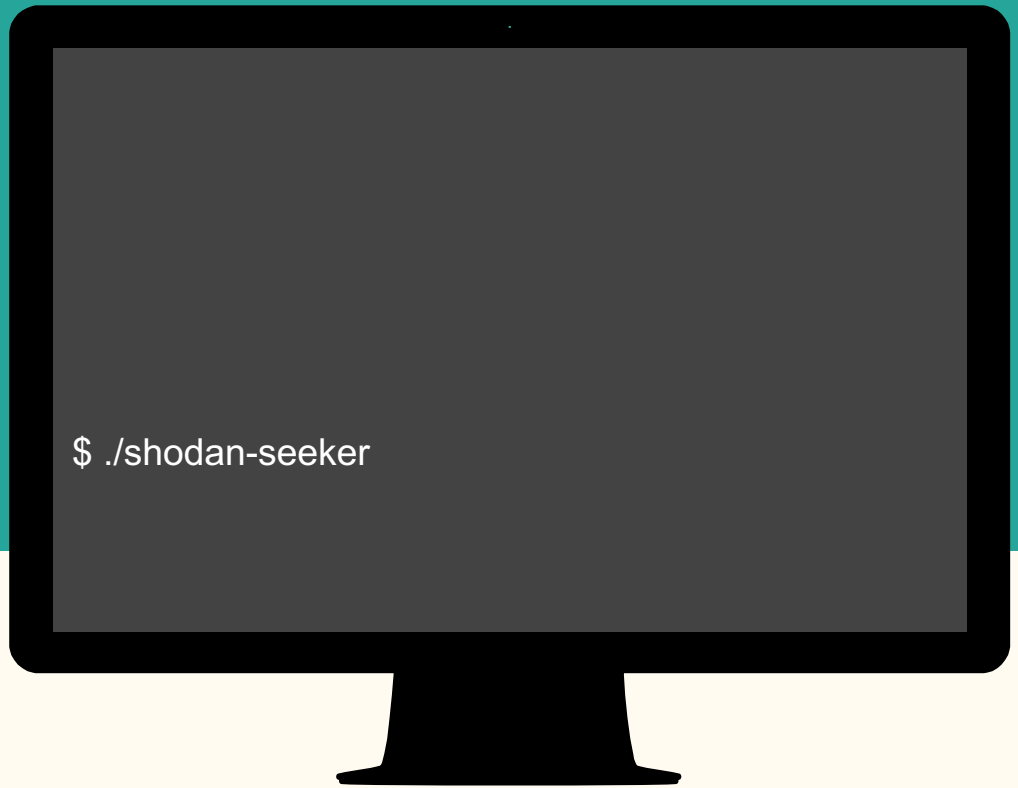
- Diffing implementation for assets discovery on Real-Time or REST API approach.
- Getting information (History, Diffing) without consuming credits.
- Generating results on csv format without consuming credits.
- Reports easily integrated with Business Data Analytics frameworks.
- Fully customizable:
  - Input data via command-line or files.
  - Different output modes.
  - Send alerts and output results to different mailing lists.
- Monitoring all tags supported by Shodan.

- Command-line friendly :)
- API plan subscription needed.

Downloads
Recommended clone from Git:

```
git clone https://github.com/laincode/shodan-seeker
cd shodan-seeker
./shodan-seeker   # just run this script
```

Demo



```
zeta:shodan-seeker lain$ ./shodanseeker
Usage: python shodanseeker [options]

Options:
  -h, --help            show this help message and exit
  --mail=MAIL           Send email with results and alerts
  -a                    Attach csv results to an email

  Scanning Options:
    --si=SCANINPUT      Scan an IP/netblock
    --sf=SCANFILE       Scan an IP/netblock from file
    --force             Force Shodan to re-scan the provided IPs
    -l                  List previously submitted scans

  Searching Options:
    -i GETINFO          Get all information of an IP/netblock
    -f GETINFOFROMFILE  Get all information of an IP/netblock from file
    --history           Return all Historical Banners
    --diff              Detect New Services Published
    --output=OUTPUT     Output results in csv format

  Monitoring in Real-Time:
    --ca=ADDALERT       Create network alerts for the IP/netblock
    --cf=ADDALERTFILE   Create network alerts from file
    --la                List of all the network alerts activated
    --da=DELALERT       Remove the specified network alert
    --subs=SUBSALERTS   Subscribe to the Private Horse Streaming
    --monport=MONPORT   Monitoring for High Risk Services
    --mondiff           Monitoring for New Services Published
    --montag=MONTAG     Tags (ex: compromised, doublepulsar, self-signed)
    --get=GET           Protocols, services, ports and tags supported

EXAMPLES:
  ./shodanseeker --si 'X.X.X.X X.X.X.X/24'                                  # Scan IPs/netblocks
  ./shodanseeker --sf 'pathfilename'                                        # Scan IPs/netblocks from a file
  ./shodanseeker -l                                                         # List previously submitted scans
  ./shodanseeker -i 'X.X.X.X X.X.X.X/24 Y.Y.Y.Y'                            # Get all information of IP/netblocks
  ./shodanseeker -f 'pathfilename'                                          # Get all information from a file of IPs/netblocks
  ./shodanseeker -i 'X.X.X.X' --history                                     # Get all historical banners
  ./shodanseeker -i 'X.X.X.X' --diff                                        # Detect new services
  ./shodanseeker -f 'pathfilename' [--history|--diff] --output csv          # Output results in csv format
  ./shodanseeker -i 'X.X.X.X' --diff --output csv --mail toaddr -a          # Send email with csv results attached
  ./shodanseeker --ca Name 'X.X.X.X X.X.X.X/24'                             # Create network alerts for the IP/netblock
  ./shodanseeker --cf Name 'pathfilename'                                   # Create network alerts from file
  ./shodanseeker --la                                                       # List of all the network alerts activated on the account
  ./shodanseeker --da [alertid|all]                                         # Remove the specified network alert
  ./shodanseeker --subs [alertid|all] --monport '3389 22' [--mail toaddr]   # Subscribe to the Streaming and monitoring for high risk services
  ./shodanseeker --subs [alertid|all] --mondiff [--mail toaddr]             # Subscribe to the Streaming and monitoring for new services published
  ./shodanseeker --subs [alertid|all] --montag 'compromised' [--mail toaddr] # Subscribe to the Streaming and monitoring for tags (ex: compromised, doublepulsar, self-signed)
  ./shodanseeker --get [protocols|services|ports|tags]                      # List of (protocols,services,ports,tags) supported

zeta:shodan-seeker lain$
```

Thanks!!

#hack.delite_es

-=Pwn&Swag=-

/RootedCON