Creación de tools de Inteligencia

Desde localizar, analizar y enriquecer el dato hasta automatizar el proceso

The better the question. The better the answer.
The better the world works.

JORNADAS DE SEGURIDAD Y CIBERDEFENSA CIBERSEG17

EY
Building a better working world

JORNADAS DE SEGURIDAD Y CIBERDEFENSA CIBERSEG17

Contenido …

- ¿Quiénes somos?
- Definiciones
- Búsqueda de información en OSINT
- Automatización de scripts de Inteligencia
- Arquitectura de una plataforma de Inteligencia

25 de Enero del 2017 / Universidad de Alcalá

EY
Building a better working world

¿Quiénes somos?

- ¿Quiénes somos?
- Definiciones
- Búsqueda de información en OSINT
- Automatización de scripts de Inteligencia
- Arquitectura de una plataforma de Inteligencia

JORNADAS DE SEGURIDAD Y CIBERDEFENSA CIBERSEG17

EY
Building a better working world

Wiktor Nykiel

Cyber Intelligence & Security Expert
Senior Manager - EY

wiktornykiel

wiktornykiel

wiktor.nykiel@es.ey.com

Iván Portillo Morales

Cyber Intelligence Expert
Senior Analyst - EY

ivanportillomorales

ivanPorMor

ivan.portillomorales@es.ey.com

Definiciones

JORNADAS DE SEGURIDAD Y CIBERDEFENSA CIBERSEG17

- ¿Quiénes somos?
- Definiciones
- Búsqueda de información en OSINT
- Automatización de scripts de Inteligencia
- Arquitectura de una plataforma de Inteligencia

EY
Building a better working world

Definiciones

OSINT	Inteligencia de Fuentes Abiertas. Información cuya procedencia se origina en fuentes públicas.
RISP	Reutilización de la Información del Sector Publico. Iniciativa de Datos Abiertos promovido por red.es, la cual permite obtener información relacionada con dominios de manera pública.
RIPE	Es el registro regional de internet encargado de supervisar y registrar los recursos de internet (Direcciones IP y AS) en las zonas de Europa, Oriente Medio y Asia Central.
TLD	Dominio de nivel superior: com, net, cn, info, es, jp, gov, edu, mil, uk, de …
Docker	Proyecto de código abierto que permite automatizar el despliegue de aplicaciones dentro de contenedores virtualizados, proporcionando una abstracción con el resto de contenedores.
Dockerfile	Archivo de configuración con los parámetros necesarios para crear una imagen en Docker.

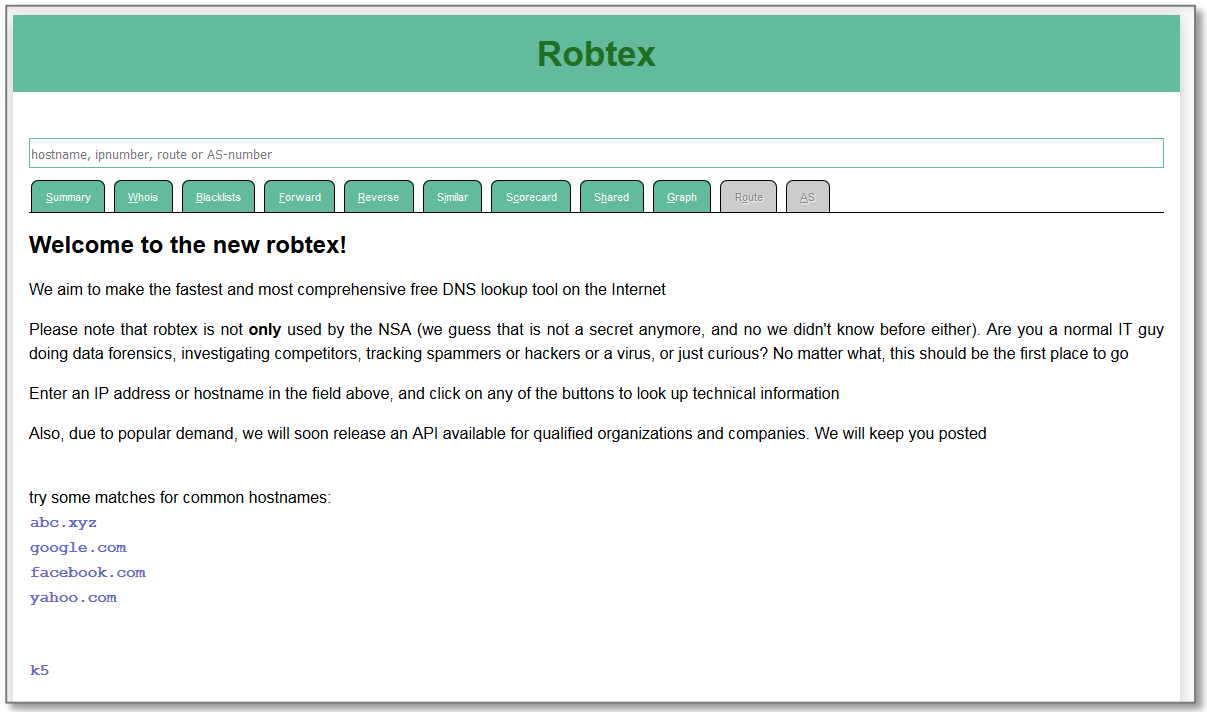
Búsqueda de información en OSINT

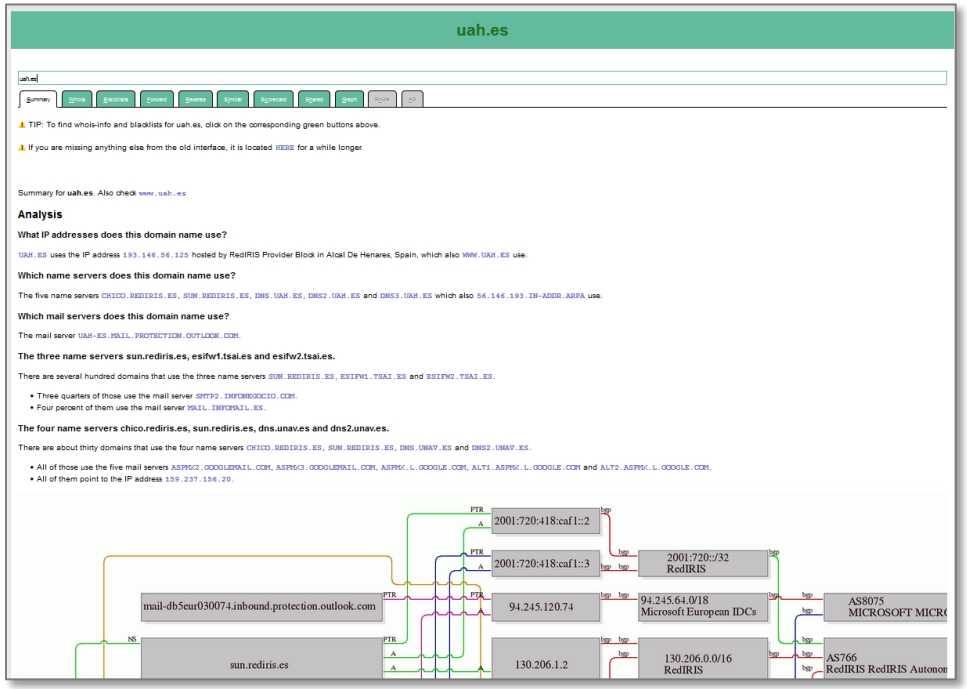
JORNADAS DE SEGURIDAD Y CIBERDEFENSA CIBERSEG17

- ¿Quiénes somos?
- Definiciones
- Búsqueda de información en OSINT
- Automatización de scripts de Inteligencia
- Arquitectura de una plataforma de Inteligencia

EY
Building a better working world

Búsqueda de información en OSINT - Robtex

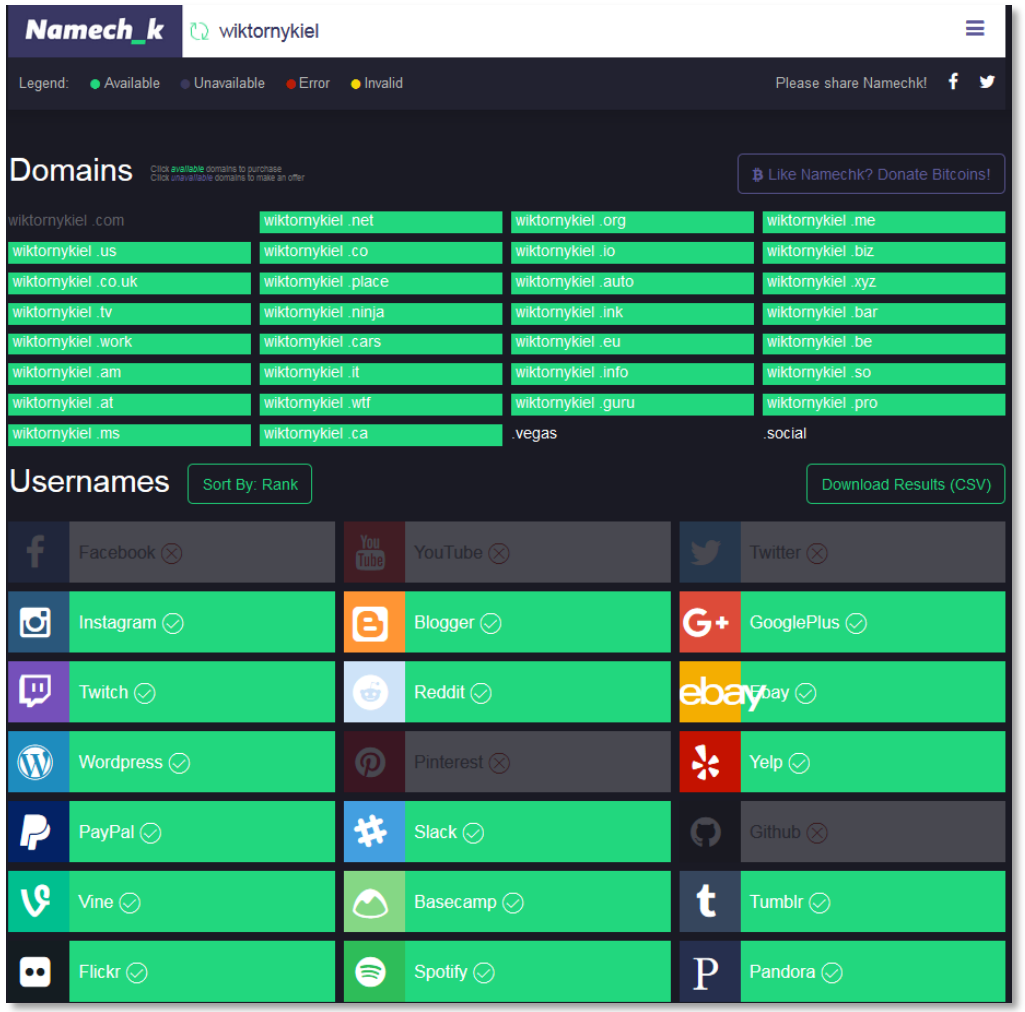




https://www.robtex.com

Herramienta de footprinting pasiva que proporciona información relacionada con direcciones IP, dominios, subdominios, DNS, servidores de correo y blacklist.

Búsqueda de información en OSINT - Namech_k

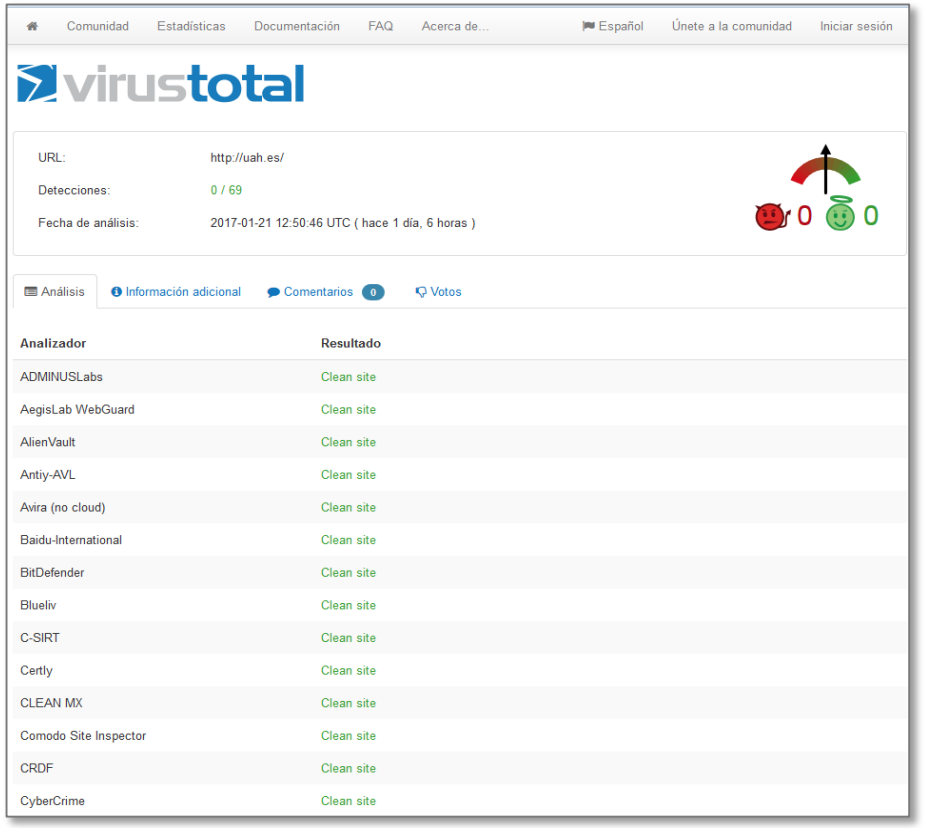


	A	B	C	D	E	F
1	username,status,url					
2	wiktornykiel,available,http://www.google.com/					
3	wiktornykiel,unavailable,https://facebook.com/wiktornykiel					
4	wiktornykiel,unavailable,https://www.youtube.com/wiktornykiel					
5	wiktornykiel,available,https://plus.google.com/					
6	wiktornykiel,available,http://instagram.com					
7	wiktornykiel,available,http://www.twitch.tv/					
8	wiktornykiel,available,http://www.ebay.com/					
9	wiktornykiel,available,http://www.yelp.com/					
10	wiktornykiel,unavailable,http://pinterest.com/wiktornykiel/					
11	wiktornykiel,available,https://slack.com/					
12	wiktornykiel,unavailable,https://github.com/wiktornykiel					
13	wiktornykiel,available,https://www.paypal.me/					
14	wiktornykiel,available,https://vine.co/					
15	wiktornykiel,available,http://www.tumblr.com/					
16	wiktornykiel,available,https://basecamp.com					
17	wiktornykiel,available,http://www.reddit.com/					
18	wiktornykiel,available,https://www.spotify.com/					
19	wiktornykiel,unavailable,https://www.producthunt.com/@wiktornykiel					
20	wiktornykiel,available,http://www.flickr.com/					
21	wiktornykiel,available,http://www.pandora.com					
22	wiktornykiel,available,http://www.myspace.com/					
23	wiktornykiel,unavailable,https://foursquare.com/wiktornykiel					
24	wiktornykiel,available,http://steamcommunity.com/					
25	wiktornykiel,available,https://www.okcupid.com/					
26	wiktornykiel,available,http://vimeo.com/					
27	wiktornykiel,unavailable,https://flipboard.com/@wiktornykiel					

https://namechk.com

Herramienta que permite realizar búsquedas de un usuario en todas las redes sociales y comprobar coincidencias.

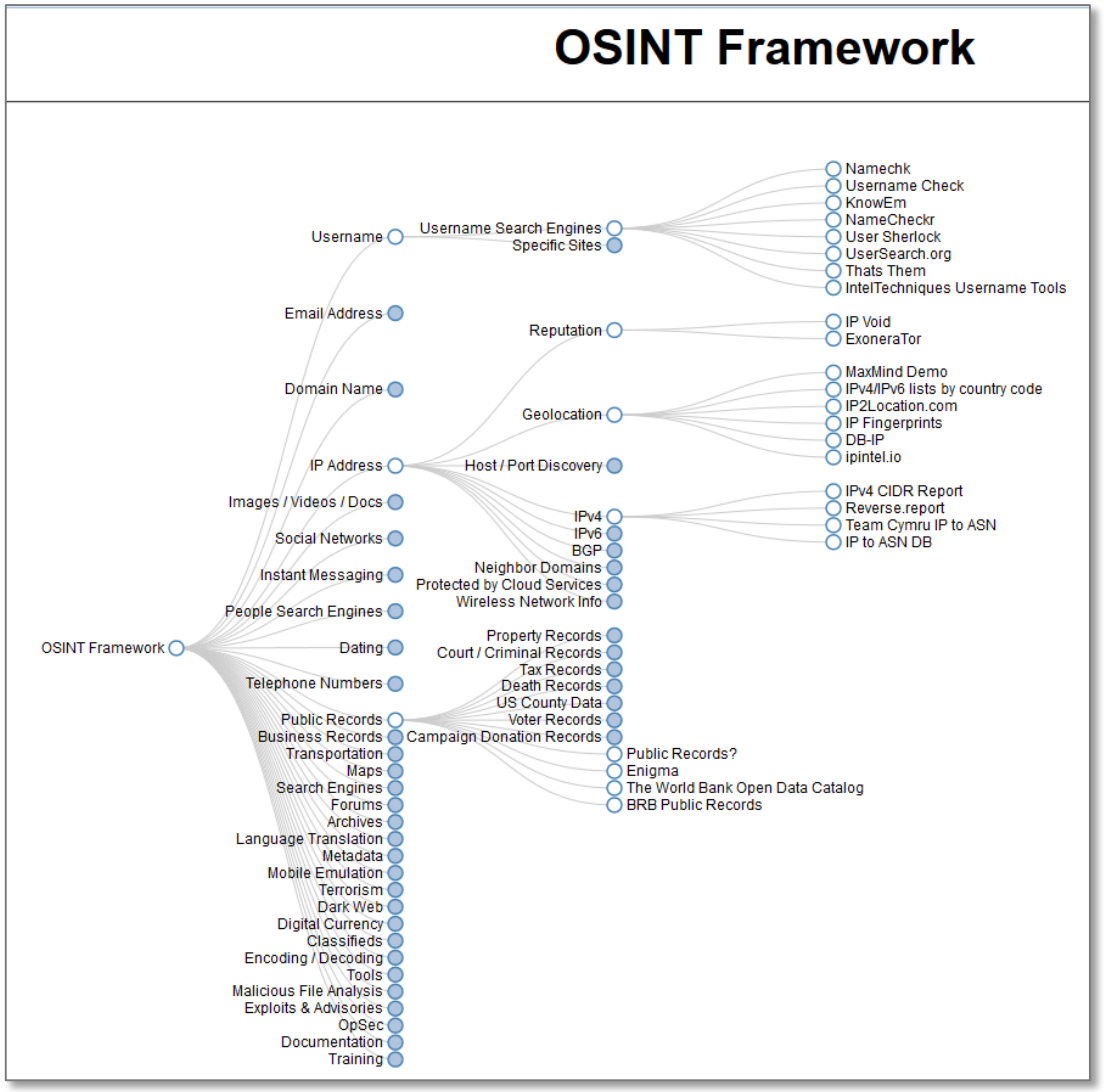
Búsqueda de información en OSINT - Virustotal



https://www.virustotal.com

Herramienta que posee una base de datos de todo tipo de malware y que permite comprobar si un fichero o una web está infectada y presenta amenazas.

Búsqueda de información en OSINT - OSINT Framework



http://osintframework.com

Web que posee una base de datos con enlaces URL de herramientas OSINT agrupadas por categorías y subcategorías en formato árbol.

Búsqueda de información en OSINT - SpiderFoot

http://spiderfoot.net/download/

Sistema Operativo
Linux y Windows

Dependencias Linux
pip install lxml netaddr M2Crypto cherrypy mako

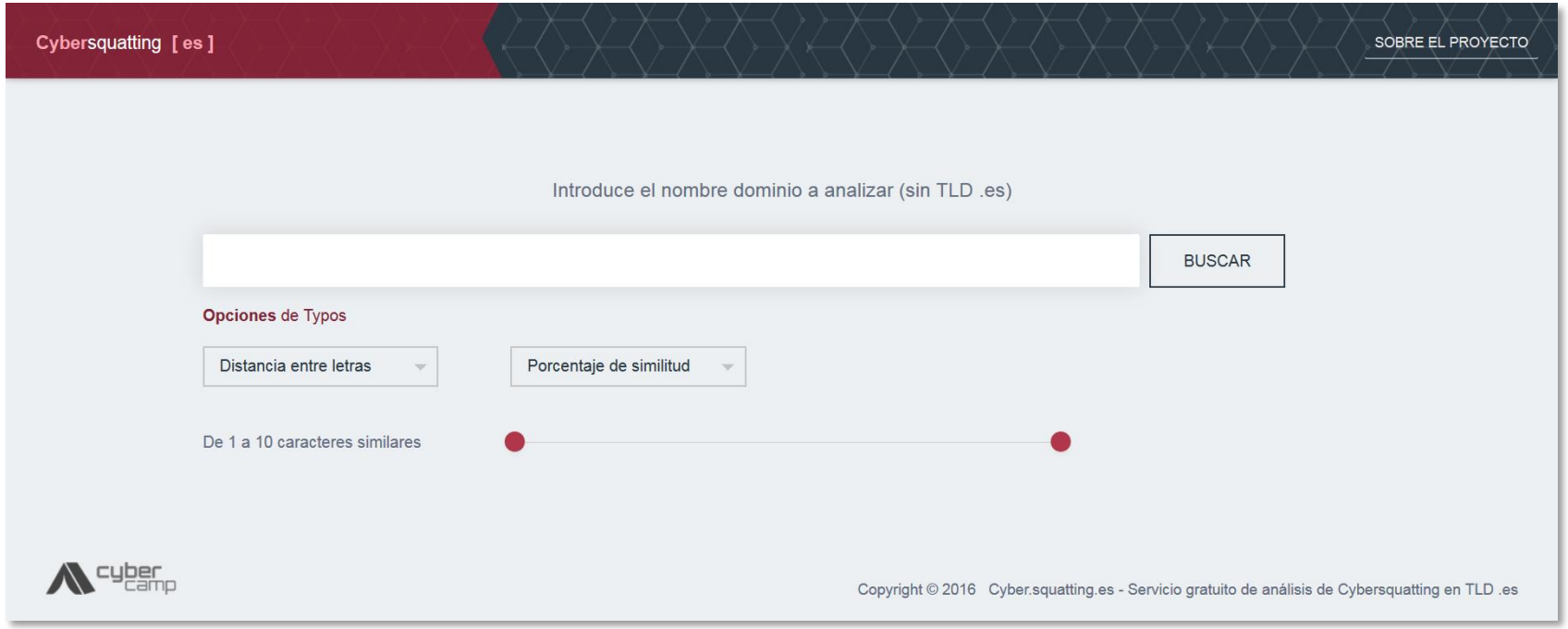
Ejecución en Linux
python ./sf.py 0.0.0.0:5001

Ejecución en Windows ejecutar fichero sf.exe

Acceso web a herramienta localhost:5001

Herramienta de inteligencia de código abierto, que permite recopilar información sobre un objetivo concreto de manera automatizada.

Buscador de libre acceso, de desarrollo propio, para localizar amenazas en dominios con TLD .es relacionado con CyberSquatting y TypoSquatting.



http://cyber.squatting.es

Automatización de scripts de Inteligencia

JORNADAS DE SEGURIDAD Y CIBERDEFENSA CIBERSEG17

- ¿Quiénes somos?
- Definiciones
- Búsqueda de información en OSINT
- **Automatización de scripts de Inteligencia**
- Arquitectura de una plataforma de Inteligencia

EY Building a better working world

Dominios_es https://www.dominios.es/

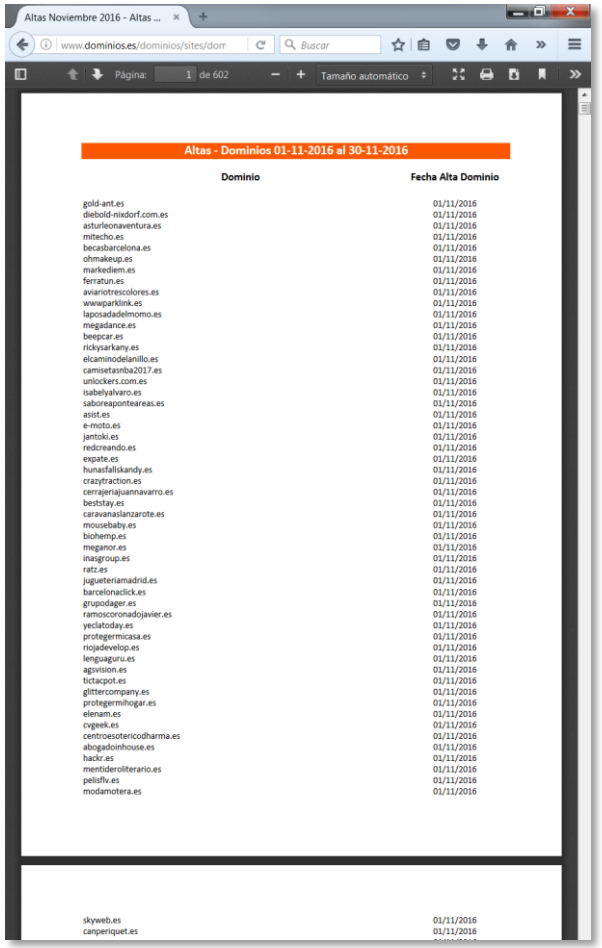
- Pdf – wget – PdfExtract
- RISP

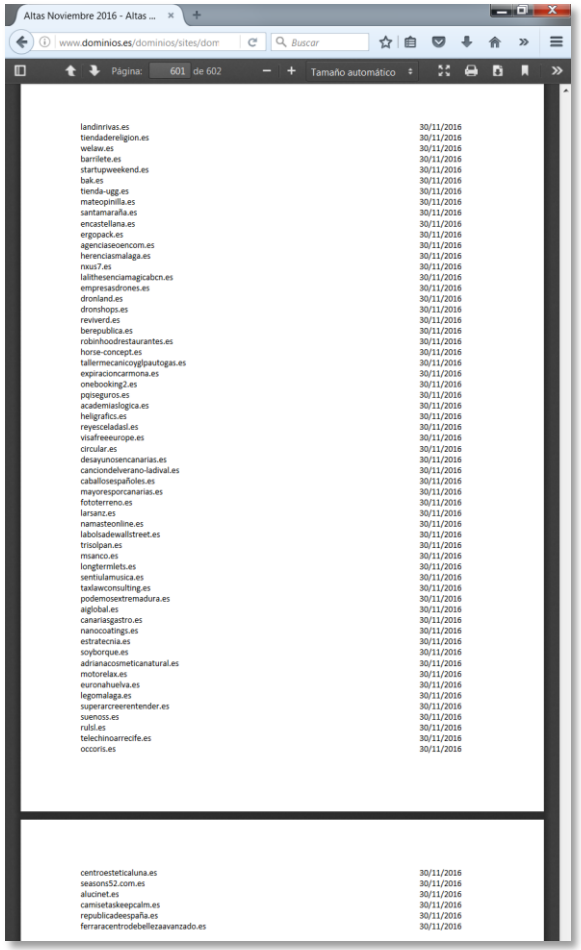
RIPE https://www.ripe.net/

Extracción Estadísticas

Se obtienen listados del alta de dominios .es registrados mensualmente desde el año 2007.

http://dominios.es/dominios/es/todo-lo-que-necesitas-saber/estadisticas/





Extracción Estadísticas

Se utiliza wget para descargar todos los PDFs de dominios registrados mensualmente

wget –r –A .pdf http://www.dominios.es/dominios/es/todo-lo-que-necesitas-saber/estadisticas

```
(ui                    ~/public_html/seguridad/dominiosES/dominios/cybercamp$ wget -A .pdf -r http://www.dominios.es/dominios/es/todo-lo-que-necesitas-saber/estadisticas

converted 'http://www.dominios.es/dominios/es/todo-lo-que-necesitas-saber/estadisticas' (ANSI_X3.4-1968) -> 'http://www.dominios.es/dominios/es/todo-lo-que-necesitas-s
aber/estadisticas' (UTF-8)
converted 'http://www.dominios.es/dominios/es/todo-lo-que-necesitas-saber/estadisticas' (ANSI_X3.4-1968) -> 'http://www.dominios.es/dominios/es/todo-lo-que-necesitas-s
aber/estadisticas' (UTF-8)
--2016-12-01 23:51:35--  http://www.dominios.es/dominios/es/todo-lo-que-necesitas-saber/estadisticas
Resolving www.dominios.es (www.dominios.es)... 194.69.254.120, 2001:67c:21cc:2000::64:120
Connecting to www.dominios.es (www.dominios.es)|194.69.254.120|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'www.dominios.es/dominios/es/todo-lo-que-necesitas-saber/estadisticas'

www.dominios.es/dominios/es/todo-lo-que-n     [ <=>                                                                                ]  71.44K  --.-KB/s    in 0.1s

2016-12-01 23:51:37 (695 KB/s) - 'www.dominios.es/dominios/es/todo-lo-que-necesitas-saber/estadisticas' saved [73150]

converted 'http://www.dominios.es/dominios/sites/dominios/themes/redesms/favicon.ico' (ANSI_X3.4-1968) -> 'http://www.dominios.es/dominios/sites/dominios/themes/redesm
s/favicon.ico' (UTF-8)
```

```
./www.dominios.es/dominios/sites/dominios/files/files:
total 67524
5381780802   12 drwx---r-x 2                    8192 Dec  2 00:01 .
4296657645    8 drwx---r-x 4                    4096 Dec  1 23:59 ..
5369132768  632 -rw----r-- 1                  645965 Nov 13  2014 1-15  10  2007.pdf
5408502494  400 -rw----r-- 1                  406412 Mar 11  2013 Agentes Registradores Febrero 2013(1).pdf
5408513564  348 -rw----r-- 1                  355015 Dec 18  2012 Agentes Registradores Noviembre 2012.pdf
5408513568  352 -rw----r-- 1                  360047 Nov 16  2012 Agentes Registradores Octubre 2012.pdf
5408513559  348 -rw----r-- 1                  354482 Jan 15  2013 Agentes Registradores diciembre 2012.pdf
5408513552  348 -rw----r-- 1                  354053 Feb 11  2013 Agentes Registradores enero 2013.pdf
5383930970  896 -rw----r-- 1                  915420 May  5  2014 Altas Abril 2014 (espa??ol).pdf
5383610266 1220 -rw----r-- 1                 1245354 May  4  2015 Altas Abril 2015 (espa??ol).pdf
5408502476  704 -rw----r-- 1                  717582 Sep  4  2013 Altas Agosto 2013.pdf
5383930959 1452 -rw----r-- 1                 1483120 Sep  4  2014 Altas Agosto 2014(espa??ol)(1).pdf
5382381106  952 -rw----r-- 1                  972497 Sep  1  2015 Altas Agosto 2015 (espa??ol).pdf
5408513560 1492 -rw----r-- 1                 1524158 Jan 15  2013 Altas Diciembre 2012.pdf
5384162067  780 -rw----r-- 1                  795228 Jan  9  2014 Altas Diciembre 2013 (espa??ol)(1).pdf
5383930947  620 -rw----r-- 1                  634235 Jan  8  2015 Altas Diciembre 2014 (espa??ol).pdf
5408513556 1404 -rw----r-- 1                 1435928 Feb  4  2013 Altas Enero 2013 (espa??ol).pdf
5384162055  924 -rw----r-- 1                  942937 Feb  3  2014 Altas Enero 2014 (espa??ol).pdf
5383610286 1376 -rw----r-- 1                 1405625 Feb  4  2015 Altas Enero 2015 (espa??ol)(1).pdf
5408502496 1208 -rw----r-- 1                 1233310 Mar  4  2013 Altas Febrero 2013 (espa??ol).pdf
```

ls -lisaR

Extracción Estadísticas

Se parsea cada documento PDF a texto con *PDF Parser*: https://github.com/smalot/pdfparser

PdfParser

Pdf Parser, a standalone PHP library, provides various tools to extract data from a PDF file.

build passing downloads 164.11 k Current Version HHVM Partial

Website : http://www.pdfparser.org

Test the API on our demo page.

This project is supported by Actualys.

Features

Features included :

- Load/parse objects and headers
- Extract meta data (author, description, ...)
- Extract text from ordered pages
- Support of compressed pdf
- Support of MAC OS Roman charset encoding
- Handling of hexa and octal encoding in text sections
- PSR-0 compliant (autoloader)
- PSR-1 compliant (code styling)

Currently, secured documents are not supported.

This Library is still under active development. As a result, users must expect BC breaks when using the master version.

PDF Parser

PHP library to parse PDF files and extract elements like text.

build passing downloads 164.11 k stable v0.9.26

Home / Demo

Demo

Demo form

Select a file to be parsed with the PDF Parser API. Text will be extracted from each page and rendered below.

Examinar... No se ha seleccionado ningún archivo.

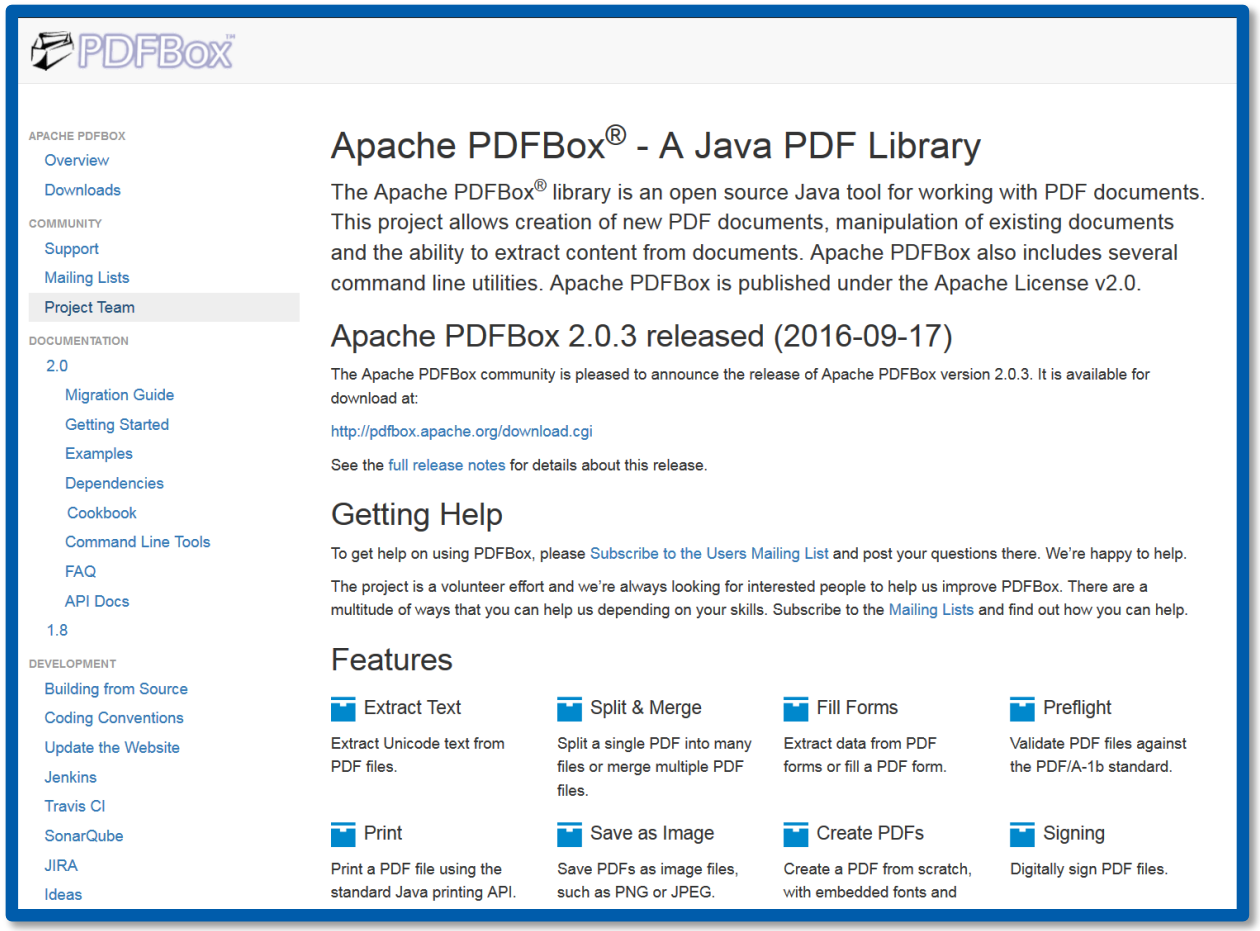
Submit

Extracción Estadísticas

Para convertir PDF a Texto por CLI se utilizará la librería software *Apache PDFBox 2.0.3 (o superior).*

https://pdfbox.apache.org/

http://pdfbox.apache.org/download.cgi



Extracción Estadísticas

java -jar pdfbox-app-2.0.3.jar ExtractText Altas-Noviembre-2016.pdf AltasNoviembre.txt

pdfbox-app-2.0.3.jar

Altas-Noviembre-2016.pdf

Altas - Dominios 01-11-2016 al 30-11-2016

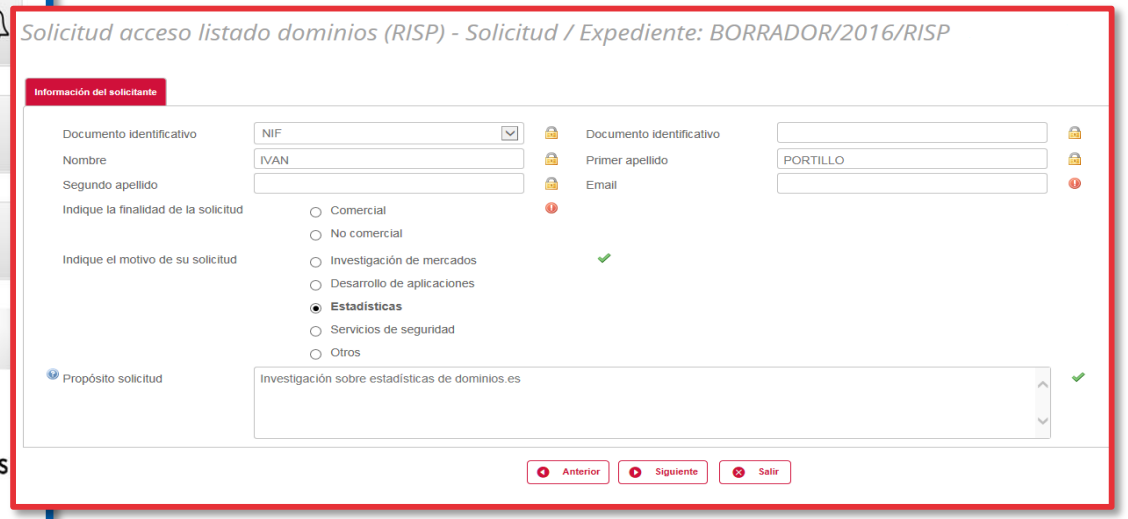
Dominio	Fecha Alta Dominio
gold-ant.es	01/11/2016
diebold-nixdorf.com.es	01/11/2016
asturleonaventura.es	01/11/2016
mitecho.es	01/11/2016
becasbarcelona.es	01/11/2016
ohmakeup.es	01/11/2016
markediem.es	01/11/2016
ferratun.es	01/11/2016
aviariotrescolores.es	01/11/2016
wwwparklink.es	01/11/2016
laposadadelmomo.es	01/11/2016
megadance.es	01/11/2016
beepcar.es	01/11/2016
rickysarkany.es	01/11/2016
elcaminodelanillo.es	01/11/2016
camisetasnba2017.es	01/11/2016
unlockers.com.es	01/11/2016
isabelyalvaro.es	01/11/2016
saboreapontеareas.es	01/11/2016
asist.es	01/11/2016
e-moto.es	01/11/2016
jantoki.es	01/11/2016
redcreando.es	01/11/2016
expate.es	01/11/2016
hunasfallskandy.es	01/11/2016
crazytraction.es	01/11/2016
cerrajeriajuannavarro.es	01/11/2016
beststay.es	01/11/2016
caravanaslanzarote.es	01/11/2016
mousebaby.es	01/11/2016
biohemp.es	01/11/2016
meganor.es	01/11/2016
inasgroup.es	01/11/2016
ratz.es	01/11/2016
jugueteriamadrid.es	01/11/2016
barcelonaclick.es	01/11/2016
grupodager.es	01/11/2016
ramoscoronadojavier.es	01/11/2016
yeclatoday.es	01/11/2016
protegermicasa.es	01/11/2016
riojadevelop.es	01/11/2016
lenguaguru.es	01/11/2016
agsvision.es	01/11/2016
[illegible]kany.es	01/11/2016
[illegible]hogar.es	01/11/2016
[illegible]	01/11/2016
[illegible]	01/11/2016
centroesotericodharma.es	01/11/2016
abogadoinhouse.es	01/11/2016
hackr.es	01/11/2016
mentideroliterario.es	01/11/2016
pelisflv.es	01/11/2016
modamotera.es	01/11/2016

AltasNoviembre.txt

```
1  Dominio Fecha Alta Dominio
2  gold-ant.es 01/11/2016
3  diebold-nixdorf.com.es 01/11/2016
4  asturleonaventura.es 01/11/2016
5  mitecho.es 01/11/2016
6  becasbarcelona.es 01/11/2016
7  ohmakeup.es 01/11/2016
8  markediem.es 01/11/2016
9  ferratun.es 01/11/2016
10 aviariotrescolores.es 01/11/2016
11 wwwparklink.es 01/11/2016
12 laposadadelmomo.es 01/11/2016
13 megadance.es 01/11/2016
14 beepcar.es 01/11/2016
15 rickysarkany.es 01/11/2016
16 elcaminodelanillo.es 01/11/2016
17 camisetasnba2017.es 01/11/2016
18 unlockers.com.es 01/11/2016
19 isabelyalvaro.es 01/11/2016
20 saboreapontеareas.es 01/11/2016
21 asist.es 01/11/2016
22 e-moto.es 01/11/2016
23 jantoki.es 01/11/2016
24 redcreando.es 01/11/2016
25 expate.es 01/11/2016
26 hunasfallskandy.es 01/11/2016
27 crazytraction.es 01/11/2016
28 cerrajeriajuannavarro.es 01/11/2016
29 beststay.es 01/11/2016
30 caravanaslanzarote.es 01/11/2016
31 mousebaby.es 01/11/2016
32 biohemp.es 01/11/2016
33 meganor.es 01/11/2016
34 inasgroup.es 01/11/2016
35 ratz.es 01/11/2016
36 jugueteriamadrid.es 01/11/2016
37 barcelonaclick.es 01/11/2016
38 grupodager.es 01/11/2016
39 ramoscoronadojavier.es 01/11/2016
40 yeclatoday.es 01/11/2016
41 protegermicasa.es 01/11/2016
42 riojadevelop.es 01/11/2016
43 lenguaguru.es 01/11/2016
```

RISP

Se accede a *Solicitud acceso listado dominios (RISP)* dentro del sitio oficial de red.es y se solicita la última Base de Datos con todos los dominios registrados con TLD .es.

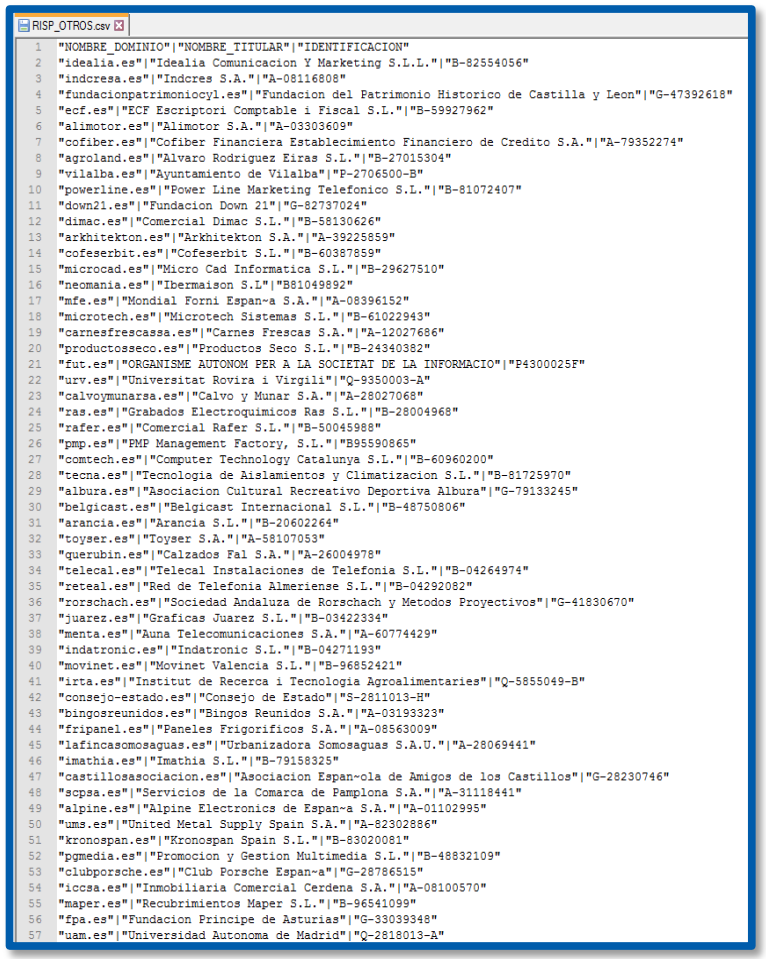


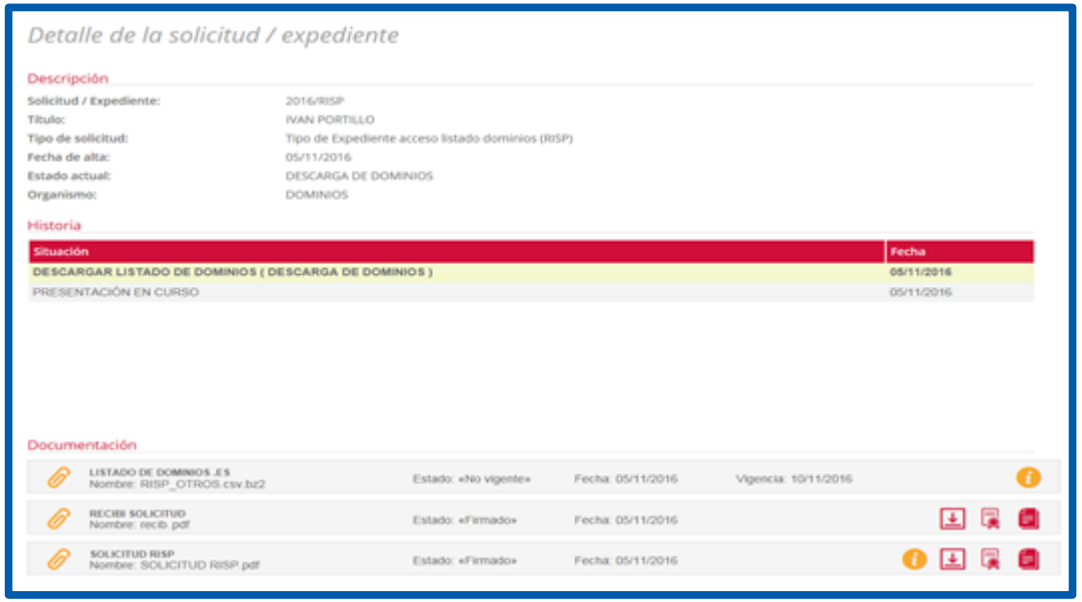
Se necesita:

- URL https://sede.red.gob.es
- Certificado Digital

RISP

Se accede al último registro recibido en la solicitud abierta y se descarga el fichero CSV.





Pasos a seguir

- Acceso CSV durante 5 días.
- Control de Base de Datos de 1.842.568 de dominios .es.

Descargando fichero

Index of /ripe/dbase

Name	Last modified	Size	Description
Parent Directory		-	
RIPE.CURRENTSERIAL	20-Jan-2017 00:00	8	
ripe.db.gz	20-Jan-2017 01:20	336M	
split/	20-Jan-2017 00:00	-	

http://ftp.ripe.net/ripe/dbase/

Index of /ripe/dbase/split

Name	Last modified	Size	Description
Parent Directory		-	
ripe.db.as-block.gz	20-Jan-2017 01:20	9.4K	
ripe.db.as-set.gz	20-Jan-2017 01:20	2.0M	
ripe.db.aut-num.gz	20-Jan-2017 01:20	7.5M	
ripe.db.domain.gz	20-Jan-2017 01:20	19M	
ripe.db.filter-set.gz	20-Jan-2017 01:20	395K	
ripe.db.inet-rtr.gz	20-Jan-2017 01:20	11K	
ripe.db.inet6num.gz	20-Jan-2017 01:20	23M	
ripe.db.inetnum.gz	20-Jan-2017 01:20	228M	
ripe.db.irt.gz	20-Jan-2017 01:20	27K	
ripe.db.key-cert.gz	20-Jan-2017 01:20	15M	
ripe.db.mntner.gz	20-Jan-2017 01:20	1.6M	
ripe.db.organisation.gz	20-Jan-2017 01:20	4.5M	
ripe.db.peering-set.gz	20-Jan-2017 01:20	72K	
ripe.db.person.gz	20-Jan-2017 01:20	409	
ripe.db.poem.gz	20-Jan-2017 01:20	29K	
ripe.db.poetic-form.gz	20-Jan-2017 01:20	1.3K	
ripe.db.role.gz	20-Jan-2017 01:20	2.7M	
ripe.db.route-set.gz	20-Jan-2017 01:20	229K	
ripe.db.route.gz	20-Jan-2017 01:20	8.2M	
ripe.db.route6.gz	20-Jan-2017 01:20	713K	
ripe.db.rtr-set.gz	20-Jan-2017 01:20	2.4K	

http://ftp.ripe.net/ripe/dbase/split/

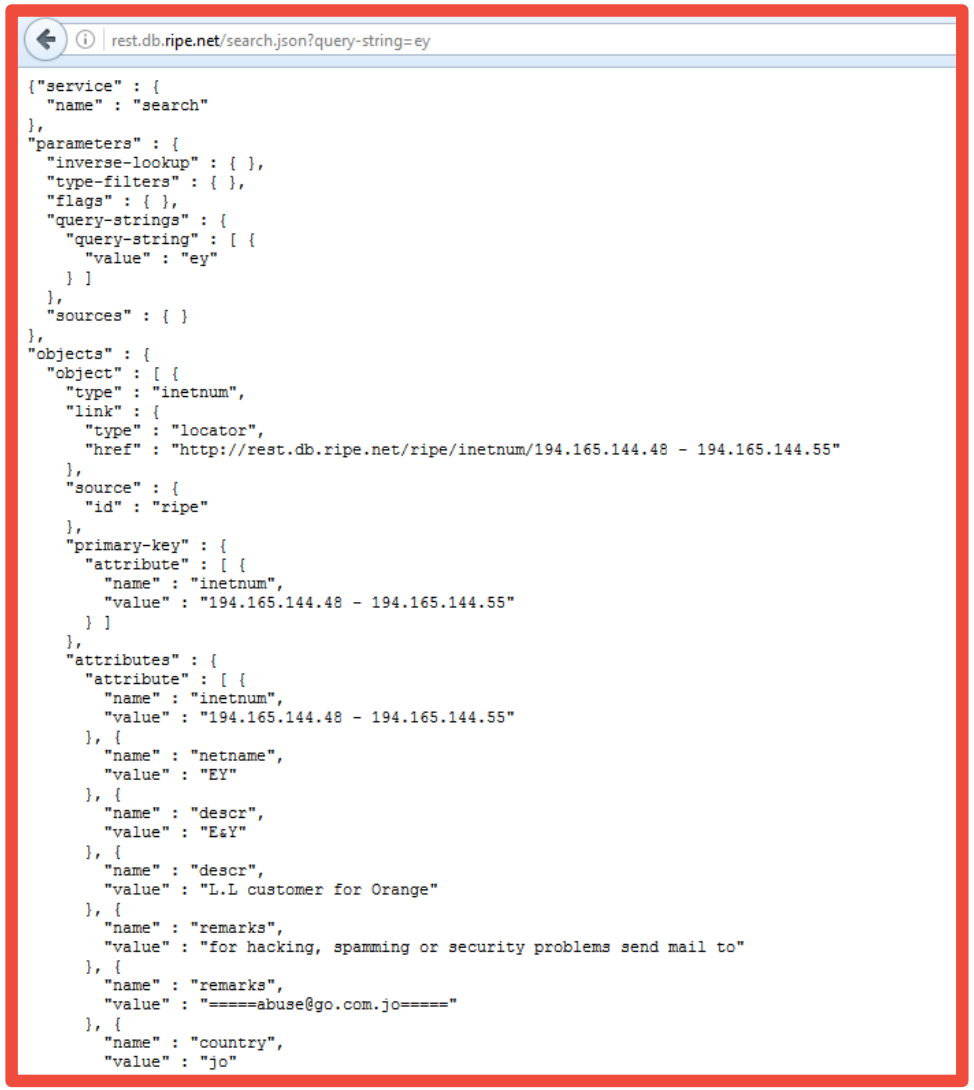
```
root@prometheus002:~/taller_ciberseg# cat ripe.db.inetnum
#
# The contents of this file are subject to
# RIPE Database Terms and Conditions
#
# http://www.ripe.net/db/support/db-terms-conditions.pdf
#

inetnum:        80.16.151.184 - 80.16.151.191
netname:        NETECONOMY-MG41731
descr:          TELECOM ITALIA LAB SPA
country:        IT
admin-c:        DUMY-RIPE
tech-c:         DUMY-RIPE
status:         ASSIGNED PA
notify:         neteconomy.rete@telecomitalia.it
mnt-by:         INTERB-MNT
created:        1970-01-01T00:00:00Z
last-modified:  2001-09-21T22:08:01Z
source:         RIPE
remarks:        ****************************
remarks:        * THIS OBJECT IS MODIFIED
remarks:        * Please note that all data that is generally regarded as personal
remarks:        * data has been removed from this object.
remarks:        * To view the original object, please query the RIPE Database at:
remarks:        * http://www.ripe.net/whois
remarks:        ****************************

% Tags relating to '80.16.151.184 - 80.16.151.191'
% RIPE-USER-RESOURCE

inetnum:        80.16.151.180 - 80.16.151.183
netname:        NETECONOMY-MG41731
descr:          TELECOM ITALIA LAB SPA
country:        IT
admin-c:        DUMY-RIPE
tech-c:         DUMY-RIPE
status:         ASSIGNED PA
notify:         neteconomy.rete@telecomitalia.it
mnt-by:         INTERB-MNT
created:        1970-01-01T00:00:00Z
last-modified:  2001-09-21T22:08:01Z
source:         RIPE
remarks:        ****************************
remarks:        * THIS OBJECT IS MODIFIED
remarks:        * Please note that all data that is generally regarded as personal
remarks:        * data has been removed from this object.
remarks:        * To view the original object, please query the RIPE Database at:
remarks:        * http://www.ripe.net/whois
```

Contenido de archivo parcial de ripe.db.inetnum

API

Application Programming Interface
http://rest.db.ripe.net/search.json?query-string=EY

Resultados en formato JSON



Automatización de scripts de Inteligencia - RIPE

API

script_ripe_api.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
from urllib2 import Request, urlopen
import json

def copy_csv(final_result):
        csv = open('ripe_csv.csv','a+')
        print final_result

        csv.write('inetnum;netname;descr;descr2;descr3\n')

        for dict1 in final_result:
                for dict2 in dict1:
                        for key,value in dict2.items():
                                csv.write(value+';')
                csv.write('\n')

        csv.close()

def search_ripe():
        inetnum=[]
        request = Request("http://rest.db.ripe.net/search.json?query-string=EY")
        response_body = urlopen(request).read()
        tmp = json.loads(response_body)
        ripe=tmp['objects']['object']
        final_result=[]
        for i in ripe:
                if 'inetnum' in i['type']:
                        total=i
                        line_result=[]
                        for line in total['attributes']['attribute']:

                                if "inetnum" in line['name'] or "netname" in line['name'] or "descr" in line['name']:

                                        file={str(line['name']):str(line['value'])}
                                        line_result.append(file)

                        final_result.append(line_result)

        copy_csv(final_result)

if __name__ == '__main__':
  search_ripe()
```

Se automatiza el proceso y se genera un CSV con los datos, el script consiste en la obtención de los rangos, nombre y descripciones por la que está registrado una organización buscada dentro de RIPE.

Ripe_csv.csv

	A	B	C	D	E
1	inetnum	netname	descr	descr	descr
2	194.165.144.48 - 194.165.144.55	EY	E&Y	L.L customer for Orange	
3	194.165.144.224 - 194.165.144.231	EY	Ernst & Young	Amman- Jordan	LL CUSTOMER FOR ORANGE
4	213.178.229.128 - 213.178.229.135	EY	Ernst & Young branch in Damascus Syria.		

Automatización de scripts de Inteligencia - Whois

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import pythonwhois
import argparse
import os.path
def copy_csv(final_result,csv,domain):
        heads_csv=['city','fax','name','state','phone','street','country','postalcode','organization','email']
        head={}
        head['domain']=domain
        head['city']=""
        head['fax']=""
        head['name']=""
        head['state']=""
        head['phone']=""
        head['street']=""
        head['country']=""
        head['postalcode']=""
        head['organization']=""
        head['email']=""

        for key in final_result:
                for key2,value in key.items():
                        for head_csv in heads_csv:
                                if key2 in head_csv:
                                        head[head_csv]=value


        csv.write(head['domain']+';'+head['city']+';'+head['fax']+';'+head['name']+';'+head['state']+';'+head['phone']+';'+head['street']+';'+head['country']+';'+head['postalcode']+';'+head['organization']+';'+head['email']+"\n")


def find_csv(final_result,domain):

        if os.path.exists('whois_csv.csv') is False:
                csv = open('whois_csv.csv','a+')
                csv.write('Domain;City;Fax;name;State;Phone;Street;Country;Postal Code;Organization;Email\n')
                copy_csv(final_result,csv,domain)
                csv.close()
        else:
                csv = open('whois_csv.csv','a+')
                copy_csv(final_result,csv,domain)
                csv.close()

def search_whois(domain):
        print domain
        final_result=[]
        domain_whois = pythonwhois.get_whois(domain)
        result_domain=domain_whois['contacts']


        for key,value in result_domain.items():
                if 'registrant' in key:

                        print value
                        final_result.append(value)


        find_csv(final_result,domain)

if __name__ == '__main__':
        parser = argparse.ArgumentParser()
        parser.add_argument("-d", "--domain", help="insert domain")
        args = parser.parse_args()
        if not args.domain:
                print "Introduce domain with -d or --domain"
        else:

                search_whois(args.domain)
```

script_whois.py

Paquetes necesarios

pip install python-whois

El script consiste en la obtención de la información relacionada con el whois de un objetivo concreto.

Se automatiza el proceso y se genera un CSV con los datos.

	A	B	C	D	E	F	G	H	I	J	K
1	Domain	City	Fax	name	State	Phone	Street	Country	Postal Code	Organization	Email
2	marca.com	Madrid		UNIDAD EDITORIAL INFORMACION DEPORTIVA S L U		34.914.435.907	Avda San Luis, 25	ES	28033	UNIDAD EDITORIAL INFORMACION DEPORTIVA, S.L.U	dominios@herrero.es

```python
#!/usr/bin/python
# -*- coding: utf-8 -*-
import socket
import argparse
import os
import json
from geoip import geolite2
def copy_csv(geoip_result,csv,domain):

        domain_result=domain
        country=str(geoip_result.country)
        continent=str(geoip_result.continent)
        timezone=str(geoip_result.timezone)
        location=str(geoip_result.location)

        csv.write(domain_result+';'+country+';'+continent+';'+timezone+';'+location+';\n')


def find_csv(geoip_result,domain):

        if os.path.exists('geoip_csv.csv') is False:
                csv = open('geoip_csv.csv','a+')
                csv.write('Domain;Country;Continent;Timezone;Location\n')
                copy_csv(geoip_result,csv,domain)
                csv.close()
        else:
                csv = open('geoip_csv.csv','a+')
                copy_csv(geoip_result,csv,domain)
                csv.close()

def search_geoip(domain):
        print domain

        ip = socket.gethostbyname(domain)
        geoip_result = geolite2.lookup(ip)
        find_csv(geoip_result,domain)


if __name__ == '__main__':
        parser = argparse.ArgumentParser()
        parser.add_argument("-d","--domain", help="insert domain")
        args = parser.parse_args()
        if not args.domain:
                print "Introduce domain with -d or --domain"
        else:

                search_geoip(args.domain)
```

script_geoip.py

	A	B	C	D	E
1	Domain	Country	Continent	Timezone	Location
2	google.com	US	NA	America/Los_Angeles	(37.419200000000004, -122.0574)
3	marca.com	ES	EU	None	(40.4, -3.6833)

Paquetes necesarios pip install python-geoip-geolite2

El script consiste en la obtención de información relacionada con la geolocalización de un dominio concreto. Se automatiza el proceso y se genera un CSV con los datos.

Automatización de scripts de Inteligencia – Obtención de A, NS, MX

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import argparse
import os
import json
import dns.resolver
def copy_csv(domain,a,ns,mx,csv):
        list_ns={}
        list_mx={}
        list_ns["ns1"]=""
        list_ns["ns2"]=""
        list_ns["ns3"]=""
        list_ns["ns4"]=""
        list_mx["mx1"]=""
        list_mx["mx2"]=""
        list_mx["mx3"]=""
        list_mx["mx4"]=""
        cont_ns=0
        cont_mx=0

        for i in ns:
                cont_ns=cont_ns+1
                list_ns['ns'+str(cont_ns)]=i

        for i in mx:
                cont_mx=cont_mx+1
                list_mx['mx'+str(cont_mx)]=str(i)

        csv.write(domain+';'+a[0]+';'+list_ns["ns1"]+';'+list_ns["ns2"]+';'+list_ns["ns3"]+';'+list_ns["ns4"]+';'+list_mx["mx1"]+';'+list_mx["mx2"]+';'+list_mx["mx3"]+';'+list_mx["mx4"]+';\n')

def find_csv(domain,a,ns,mx):

        if os.path.exists('domains_csv.csv') is False:
                csv = open('domains_csv.csv','a+')
                csv.write('Domain;A;NS1;NS2;NS3;NS4;MX1;MX2;MX3;MX4\n')
                copy_csv(domain,a,ns,mx,csv)
                csv.close()
        else:
                csv = open('domains_csv.csv','a+')
                copy_csv(domain,a,ns,mx,csv)
                csv.close()

def search_ns(domain):

        list_ns=[]
        num_ns=0
        try:
                answers_ns = dns.resolver.query(domain, 'NS')
                for i in answers_ns:
                        num_ns=num_ns+1
                        if num_ns<=4:
                                if num_ns==1:
                                        ns1=i.to_text()
                                        list_ns.append(ns1)

                                elif num_ns==2:
                                        ns2=i.to_text()
                                        list_ns.append(ns2)
                                elif num_ns==3:
                                        ns3=i.to_text()
                                        list_ns.append(ns3)
                                elif num_ns==4:
                                        ns4=i.to_text()
                                        list_ns.append(ns4)
        except:
                print "error"
        return list_ns
```

script_mx_ns.py

```
def search_mx(domain):
        list_mx=[]
        num_mx=0
        try:
                answers_mx = dns.resolver.query(domain, 'MX')
                for i in answers_mx:
                        num_mx=num_mx+1
                        if num_mx<=4:
                                if num_mx==1:
                                        mx1=i.exchange
                                        list_mx.append(mx1)
                                elif num_mx==2:
                                        mx2=i.exchange
                                        list_mx.append(mx2)
                                elif num_mx==3:
                                        mx3=i.exchange
                                        list_mx.append(mx3)
                                elif num_mx==4:
                                        mx4=i.exchange
                                        list_mx.append(mx4)
        except:
                mx1=""
                mx2=""
                mx3=""
                mx4=""
        return list_mx

def search_domains(domain):
        print domain
        a=[]
        try:
                answer = dns.resolver.query(domain, 'A')
                for i in answer:
                        a.append(str(i))

        except:
                a=[]
        print a
        ns=search_ns(domain)
        mx=search_mx(domain)
        print a,domain,ns,mx
        find_csv(domain,a,ns,mx)


if __name__ == '__main__':
        parser = argparse.ArgumentParser()
        parser.add_argument("-d","--domain", help="insert domain")
        args = parser.parse_args()
        if not args.domain:
                print "Introduce domain with -d or --domain"
        else:
                search_domains(args.domain)
```

El script consiste en la obtención de la información relacionada con la IP, nombres de dominio y servidores de correo de un dominio concreto.

Se automatiza el proceso y se genera un CSV con los datos.

	A	B	C	D	E	F	G	H	I	J
1	Domain	A	NS1	NS2	NS3	NS4	MX1	MX2	MX3	MX4
2	google.com	216.58.214.174	ns2.google.com.	ns1.google.com.	ns4.google.com.	ns3.google.com.	alt4.aspmx.l.google.com.	alt1.aspmx.l.google.com.	aspmx.l.google.com.	alt2.aspmx.l.google.com.

Automatización de scripts de Inteligencia - Combinatoria con TLDs

El script consiste en la generación de un listado de un dominio, combinándolo con diferentes TLDs para obtener de manera automatizada todos los registros A, NS y MX de cada uno de ellos. Permitirá comprobar si cada unos de los dominios tiene relación entre si.

```python
#!/usr/bin/python
# -*- coding: utf-8 -*-
import pythonwhois
import argparse
import os.path
from script_tlds_mx_ns import search_domains


def search_tld(domain,file):
        final_result=[]
        domain_all=[]
        domain_sintld=domain.split('.')

        tlds=open(file,'r')
        for tld in tlds.read().split('\r\n'):
                domain_all.append(domain_sintld[0]+tld)

        for domain in domain_all:
                print domain
                search_domains(domain)



if __name__ == '__main__':
        parser = argparse.ArgumentParser()
        parser.add_argument("-d","--domain", help="insert domain")
        parser.add_argument("-f","--file", help="insert file")
        args = parser.parse_args()
        if not args.domain:
                print "Introduce domain with -d or --domain"
        elif not args.file:
                print "Introduce file with -f or --file"
        else:

                search_tld(args.domain,args.file)
```

script_combinatoria_tlds.py

script_tlds_mx_ns.py

	A	B	C	D	E	F	G	H	I	J
1	Domain	A	NS1	NS2	NS3	NS4	MX1	MX2	MX3	MX4
2	uah.it	['80.92.84.139']	ns4.eurodns.com.	ns1.eurodns.com.	ns2.eurodns.com.	ns3.eurodns.com	mail.b-io.co.			
3	uah.biz	['176.31.179.131', '37.187.83.72']	dns2.ru-tld.ru.	dns1.ru-tld.ru.						
4	uah.eu	['109.106.167.8']	dns4.blixem.nl.	dns5.blixem.nl.			postduif.natuurpark.nl.			
5	uah.net	['185.53.179.8']	ns2.parkingcrew.net.	ns1.parkingcrew.net.			mail.h-email.net.			
6	uah.org	['75.119.201.101']	ns.xmission.com.	ns1.xmission.com.	ns2.xmission.com.		mx.xmission.com.			
7	uah.com	['208.73.211.178', '208.73.210.214', '208.73.210.217', '208.73.210	ns1.dsredirection.com.	ns2.dsredirection.com.						
8	uah.info	['37.187.83.72', '176.31.179.131']	dns2.ru-tld.ru.	dns1.ru-tld.ru.						
9	uah.gov.it	[]								
10	uah.ch	['72.52.4.119']	ns1.sedoparking.com.	ns2.sedoparking.com.			localhost.			
11	uah.ru	['193.124.57.181']	ns.sutel.ru.	ns.66p.su.			emx.mail.ru.			
12	uah.us	['69.172.201.153']	ns1.uniregistrymarket.link.	ns2.uniregistrymarket.link.						
13	uah.post	[]								
14	uah.es	['193.146.56.125']	dns.uah.es.	dns3.uah.es.	chico.rediris.es.	sun.rediris.es.	uah-es.mail.protection.outlook.com.			
15	uah.fr	['52.58.78.16']	ns1.undeveloped.com.	ns2.undeveloped.com.			mx.uah.fr.			
16	uah.int	[]								
17	uah.pro	[]								
18	uah.tel	[]								
19	uah.co	['184.168.221.96']	ns01.cashparking.com.				smtp.secureserver.net.	mailstore1.secureserver.net.		
20	uah.uk	[]								
21	uah.su	['82.146.61.215']	ns110.inhostedns.com.	ns310.inhostedns.org.	ns210.inhostedns.net.		mx15.ukraine.com.ua.	mx20.ukraine.com.ua.		
22	uah.ar	[]								
23	uah.at	['109.235.63.103']	ns1.golem.eu.	ns2.golem.eu.	ns3.golem.eu.					
24	uah.ae	[]								
25	uah.be	['213.132.196.245']	ns.nlhosting.net.	ns1.nlhosting.net.			mailfilter1.webguru.nl.			
26	uah.br	[]								
27	uah.ca	['208.73.210.212', '208.73.211.163', '208.73.211.236', '208.73.210	ns1.dsredirection.com.	ns2.dsredirection.com.						
28	uah.cl	['190.96.232.125']	secundario.nic.cl.	ns.uahurtado.cl.			proxmox.uahurtado.cl.	mx2.uahurtado.cl.		
29	uah.cy	[]								
30	uah.cz	['54.72.9.51']	ns2.parkingcrew.net.	ns1.parkingcrew.net.			mail.h-email.net.			
31	uah.de	['81.169.145.72']	docks05.rzone.de.	shades15.rzone.de.			smtpin.rzone.de.			
32	uah.dk	['46.30.213.154']	ns02.one.com.	ns01.one.com.			mxcluster2.one.com.	mxcluster1.one.com.		
33	uah.ee	[]								
34	uah.gr	[]								
35	uah.fi	['54.72.9.51']	ns1.parkingcrew.net.	ns2.parkingcrew.net.			mail.h-email.net.			
36	uah.hk	[]								
37	uah.hr	['88.99.30.30']	ns3.totohost.com.	ns2.totohost.com.	ns1.totohost.com.	ns4.totohost.com	uah.hr.			
38	uah.lu	[]								
39	uah.mc	[]								
40	uah.me	['122.10.94.116']	f1g1ns2.dnspod.net.	f1g1ns1.dnspod.net.						
41	uah.nl	['109.106.167.8']	dns1.blixem.nl.	dns2.blixem.nl.	dns3.blixem.nl.		postduif.natuurpark.nl.			
42	uah.no	['194.63.248.52']	ns1.hyp.net.	ns2.hyp.net.	ns3.hyp.net.					
43	uah.pt	[]								
44	uah.pl	['80.92.84.139']	ns2.eurodns.com.	ns3.eurodns.com.	ns4.eurodns.com.	ns1.eurodns.com.	mail.b-io.co.			
45	uah.py	[]								
46	uah.se	['72.52.4.119']	ns2.sedoparking.com.	ns1.sedoparking.com.			localhost.			
47	uah.si	[]								
48	uah.ua	[]								

```python
#!/usr/bin/python
# -*- coding: utf-8 -*-
import argparse
import os
import json
import shodan
from conf import API_SHODAN
def copy_csv(activos,rango,csv):

        for activo in activos:
                csv.write(rango+';'+str(activo['ip'])+';'+str(activo['asn'])+';'+str(activo['isp'])+';'+str(activo['puertos'])+';'+str(activo['module'])+';'+str(activo['org'])+';'+
                        +str(activo['domains'])+';'+str(activo['location'])+';'+str(activo['hostnames'])+';'+str(activo['os'])+';'+str(activo['cpe'])+';'+str(activo['product'])+
                        +';'+str(activo['vuln'])+','+str(activo['ssh'])+';\n')

def find_csv(activo,rango):

        if os.path.exists('shodan_csv.csv') is False:
                csv = open('shodan_csv.csv','a+')
                csv.write('rango;ip;asn;isp;puertos;module;org;domains;location;hostnames;os;cpe;product;vuln;ssh\n')
                copy_csv(activo,rango,csv)
                csv.close()
        else:
                csv = open('shodan_csv.csv','a+')
                copy_csv(activo,rango,csv)
                csv.close()
```

```python
def search_shodan(rango):
        api = shodan.Shodan(API_SHODAN)
        dict_shodan={}
        dict_shodan["rango"]=rango
        activo=[]
        print dict_shodan["rango"]
        try:
                # busqueda de rangos IP en Shodan
                host = api.search('net:"%s"' %(rango))

                # recorrido de cada resultado de shodan
                for i in host['matches']:
                        dict_shodan["ips"]={}
                        dict_shodan["ips"]["asn"]=""
                        dict_shodan["ips"]["puertos"]=""
                        dict_shodan["ips"]["isp"]=""
                        dict_shodan["ips"]["org"]=""
                        dict_shodan["ips"]["domains"]=""
                        dict_shodan["ips"]["hostnames"]=""
                        dict_shodan["ips"]["location"]=""
                        dict_shodan["ips"]["ip"]=""
                        dict_shodan["ips"]["os"]=""
                        dict_shodan["ips"]["cpe"]=[]
                        dict_shodan["ips"]["product"]=""
                        dict_shodan["ips"]["module"]=[]
                        dict_shodan["ips"]['vuln']=[]
                        dict_shodan["ips"]['ssh']=[]
                        host_buscar=api.host(i['ip_str'])
                        if 'ssh' in i:
                                dict_shodan["ips"]['ssh']=i['ssh']
                        if 'cpe' in i:
                                dict_shodan["ips"]["cpe"]=i['cpe']
                        dict_shodan["ips"]["os"]=i['os']
                        dict_shodan["ips"]["module"]=i['_shodan']['module']
                        dict_shodan["ips"]["domains"]=i["domains"]

                        for it in host_buscar['data']:
                                if 'org' in it:
                                        dict_shodan["ips"]["org"]=it['org'].decode('utf-8')
                                elif 'asn' in it:
                                        dict_shodan["ips"]["asn"]=it['asn'].decode('utf-8')
                        if 'vuln' in host_buscar:
                                dict_shodan["ips"]['vuln']=host_buscar['vuln']
                        # Imprimimos los banners
                        dict_shodan["ips"]["ip"]=host_buscar['ip_str']
                        print(host_buscar['ip_str'])
                        if 'product' in i:
                                dict_shodan["ips"]["product"]=i['product']
                        dict_shodan["ips"]["puertos"]=host_buscar['ports']
                        dict_shodan["ips"]["isp"]=host_buscar['isp'].decode('utf-8')
                        dict_shodan["ips"]["location"]=host_buscar['country_name']
                        activo.append(dict_shodan["ips"])
                        print activo
                        find_csv(activo,rango)
                        print '\n----------------------\n'


        except Exception as e:
                print 'Ha ocurrido un error: %s' % e
```

```python
if __name__ == '__main__':
        parser = argparse.ArgumentParser()
        parser.add_argument("-r", "--rango", help="inserte rango")
        args = parser.parse_args()
        if not args.rango:
                args.rango='194.165.144.224-194.165.144.231'

        search_shodan(args.rango)
```

script_shodan.py

Paquetes necesarios: pip install shodan

El script consiste en la obtención de la información relacionada con los datos que shodan posee sobre el objetivo a partir de un rango dado. Se automatiza el proceso y se genera un CSV con los datos.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	rango	ip	asn	isp	puertos	module	org	domains	location	hostnames	os	cpe	product	vuln	ssh
2	194.165.144.224-194.165.144.231	194.165.144.226	AS8376	Jordan Data Communications Company LLC	[500, 4500]	ike	Jordan Data Communications Company LLC	[]	Jordan		None	[]		[]	[]
3	194.165.144.224-194.165.144.231	194.165.144.232	AS8376	Jordan Data Communications Company LLC	[2123, 2152]	gtp-v1	Jordan Data Communications Company LLC	[]	Jordan		None	[]	GPRS Tunneling Protocol	[]	[]
4	194.165.144.224-194.165.144.231	194.165.144.226	AS8376	Jordan Data Communications Company LLC	[500, 4500]	ike-nat-t	Jordan Data Communications Company LLC	[]	Jordan		None	[]		[]	[]
5	194.165.144.224-194.165.144.231	194.165.144.232	AS8376	Jordan Data Communications Company LLC	[2123, 2152]	gtp-v1	Jordan Data Communications Company LLC	[]	Jordan		None	[]	GPRS Tunneling Protocol	[]	[]
6	194.165.144.224-194.165.144.231	194.165.144.229	AS8376	Jordan Data Communications Company LLC	[123]	ntp	Jordan Data Communications Company LLC	[]	Jordan		None	[]		[]	[]

Automatización de scripts de Inteligencia - Nmap

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import argparse
import os.path
import nmap
def copy_csv(final_result,csv):

        for key,value in final_result.items():
                ip=key
                ports=""
                product=""
                state=""
                version=""
                extra_info=""
                cpe=""

                if value.has_key('status'):
                        listaProtocolos = ["udp", "tcp", "icmp"]
                        for protocol in listaProtocolos:
                                if value.has_key(protocol):
                                        list_ports = value[protocol].keys()
                                        for port in list_ports:
                                                ports=port
                                                product = value[protocol][port]['product']
                                                state = value[protocol][port]['state']
                                                version = value[protocol][port]['version']
                                                extra_info = value[protocol][port]['extrainfo']
                                                cpe = value[protocol][port]['cpe']
                                                print str(key)+';'+str(ports)+';'+str(product)+';'+str(state)+';'+str(version)+';'+str(extra_info.replace(';',','))+';'+str(cpe)+"\n"
                                                csv.write(str(key)+';'+str(ports)+';'+str(product)+';'+str(state)+';'+str(version)+';'+str(extra_info.replace(';',','))+';'+str(cpe)+"\n")


def find_csv(final_result):

        if os.path.exists('nmap_csv.csv') is False:
                csv = open('nmap_csv.csv', 'a+')
                csv.write('ip;Ports;Product;State;Version;Extra Info;cpe\n')
                copy_csv(final_result,csv)
                csv.close()
        else:
                csv = open('nmap_csv.csv','a+')
                copy_csv(final_result,csv)
                csv.close()

def ejecutar_nmap(host, argumentos, puertos=""):  # Ejecutar comandos de nmap
    try:
        escaneo = nmap.PortScanner()                        # Abre el portscanner de nmap
        escaneo.scan(host, arguments=argumentos)          # Ejecuta el comando con los argumentos pasados
        return escaneo
    except Exception:
        RuntimeWarning

def search_nmap(ip):
        print ip
        final_result=[]

        nmap_result=ejecutar_nmap(ip,"-sS -O -sV")

        for clave, valor in list(nmap_result.__dict__.items()):  # recorre el resultado
           if isinstance(valor, dict):
                                print valor['scan']
                                find_csv(valor['scan'])


        #find_csv(final_result,domain)

if __name__ == '__main__':
        parser = argparse.ArgumentParser()
        parser.add_argument("-ip","--ip", help="insert IP")
        args = parser.parse_args()
        if not args.ip:
                print "Introduce IP with -ip or --ip"
        else:

                search_nmap(args.ip)
```

script_nmap.py

Paquetes necesarios: pip install python-nmap

El script consiste obtener información de los servicios activos de una dirección IP como objetivo. Se automatiza el proceso y se genera un CSV con los datos.

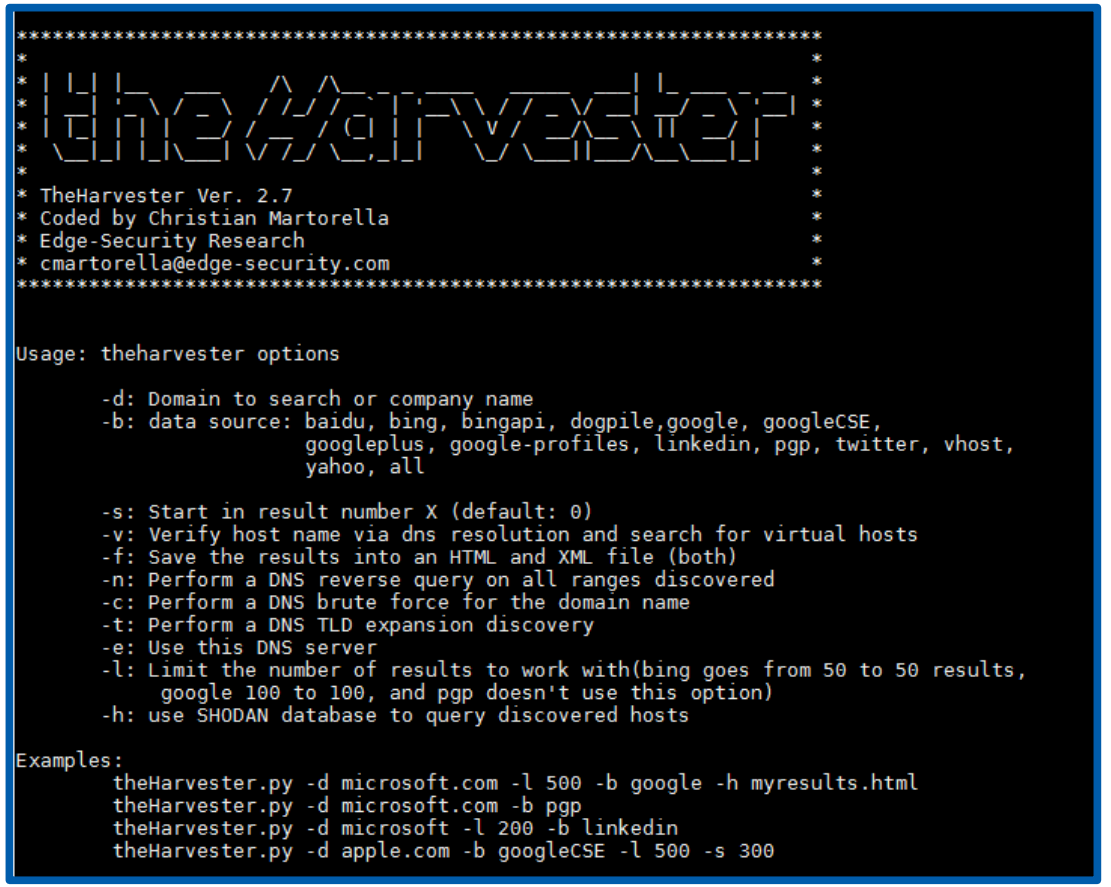
	A	B	C	D	E	F	G
1	ip	Ports	Product	State	Version	Extra Info	cpe
2	127.0.0.1	22	OpenSSH	open	7.2p2 Ubuntu 4ubuntu2.1	Ubuntu Linux, protocol 2.0	cpe:/o:linux:linux_kernel

Automatización de scripts de Inteligencia - The Harvester

The Harvester es una herramienta desarrollada en Python, que permite recopilar información pública a través de los buscadores. Se pueden obtener emails, subdominios, IPs, nombre de perfiles de usuario en redes sociales.

https://github.com/laramies/theHarvester

python theHarvester.py -d uah.es -l 300 -b all -f result.html



Listado de Emails

Listado de Subdominios e IPs

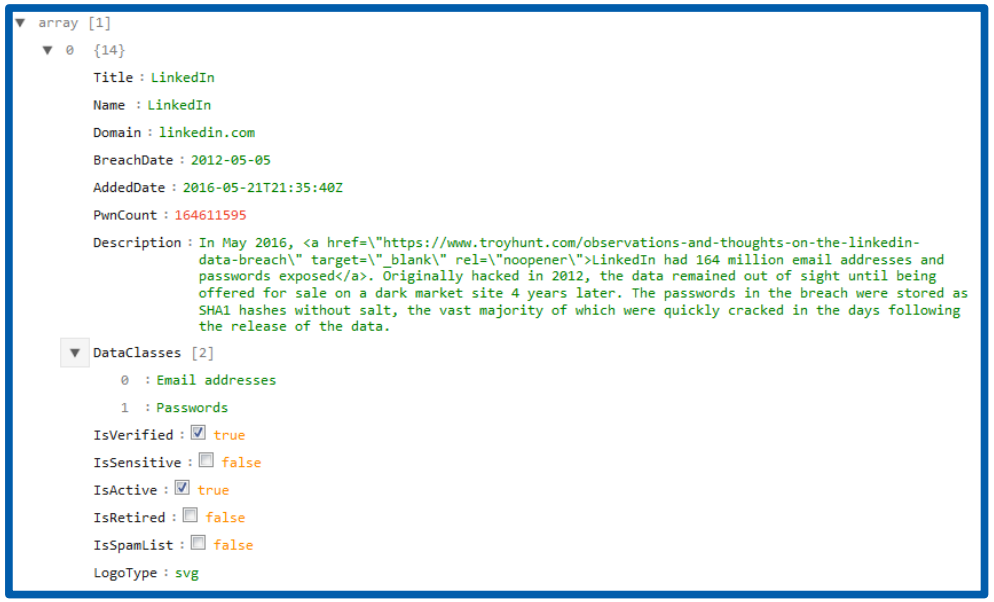
Listado de nombres de usuarios de Redes Sociales

Automatización de scripts de Inteligencia - HaveibeenPwned (I)

HaveIbeenPwned es una herramienta web que permite comprobar si un email se encuentra en alguno de las base de datos de leaks. Se pretende desarrollar una herramienta que permita consultar mediante su API, que emails se encuentran filtrados en algún leak, utilizando el listado obtenido de la herramienta anterior.

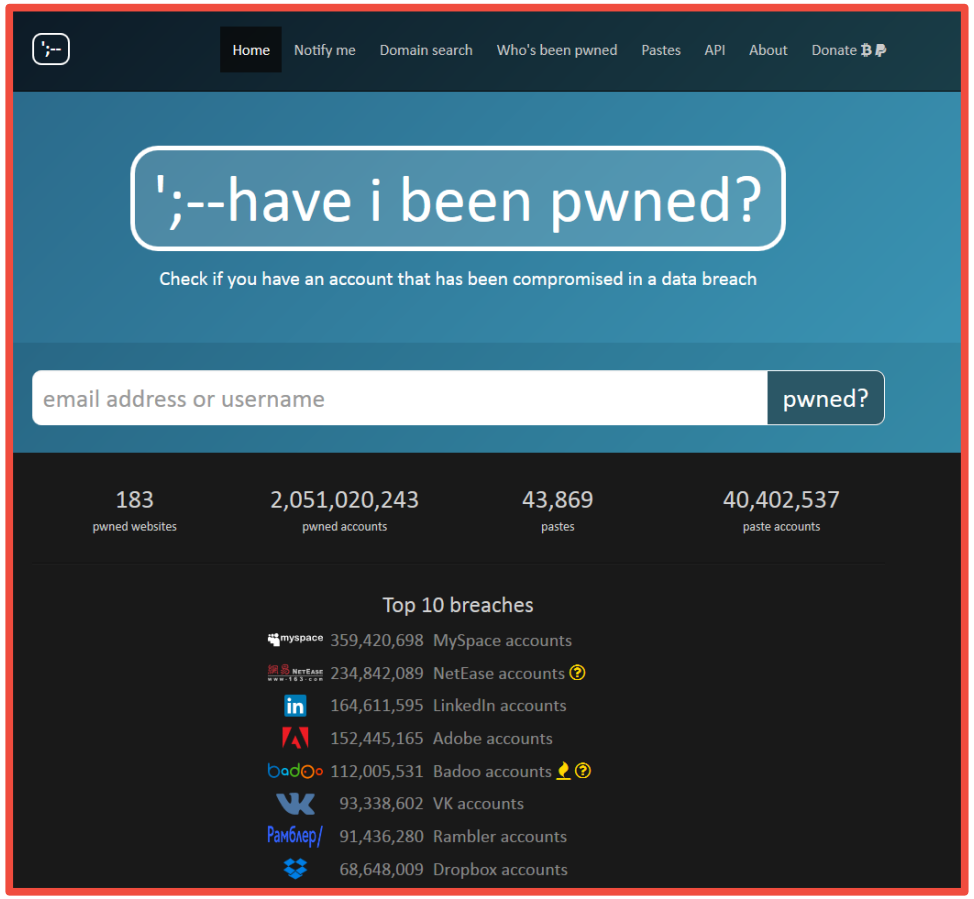
API https://haveibeenpwned.com/api/v2/breachedaccount/carlos.iglesias@uah.es

[{"Title":"LinkedIn","Name":"LinkedIn","Domain":"linkedin.com","BreachDate":"2012-05-05","AddedDate":"2016-05-21T21:35:40Z","PwnCount":164611595,"Description":"In May 2016, LinkedIn had 164 million email addresses and passwords exposed. Originally hacked in 2012, the data remained out of sight until being offered for sale on a dark market site 4 years later. The passwords in the breach were stored as SHA1 hashes without salt, the vast majority of which were quickly cracked in the days following the release of the data.","DataClasses":["Email addresses","Passwords"],"IsVerified":true,"IsSensitive":false,"IsActive":true,"IsRetired":false,"IsSpamList":false,"LogoType":"svg"}]



Automatización de scripts de Inteligencia – HaveibeenPwned (II)

https://haveibeenpwned.com/



script_leak.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import argparse
import json
import requests
import os
import time

def copy_csv(email,list_leak,csv):


        csv.write(email+';'+str(list_leak)+';\n')

def find_csv(email,list_leak):

        if os.path.exists('leak_csv.csv') is False:
                csv = open('leak_csv.csv','a+')
                csv.write('email;leak\n')
                copy_csv(email,list_leak,csv)
                csv.close()
        else:
                csv = open('leak_csv.csv','a+')
                copy_csv(email,list_leak,csv)
                csv.close()

def search_all_leaks(leak_user):

        leaks=[]
        url = "https://haveibeenpwned.com/api/v2/breachedaccount/"+leak_user
        response = requests.get(url)
        if response.status_code == 200:
                results = response.json()
                for result in results:
                        leaks.append(result['Name'])
        else:
                print "Error code %s" % response.status_code

        return leaks

def search_leak(file):

        emails_leak=[]

        emails=open(file,'r')
        for line in emails.read().split('\r\n'):
                emails_leak.append(line.encode('utf8'))

        for line_email in emails_leak:
                leak_dict={}
                leak_dict['user']=""
                leak_dict['leaks']=[]
                leak_dict['user']=line_email

                leak_dict['leaks']=search_all_leaks(leak_dict['user'])
                print leak_dict['user'] , leak_dict['leaks']
                find_csv(leak_dict['user'],leak_dict['leaks'])
                time.sleep(2)

if __name__ == '__main__':
        parser = argparse.ArgumentParser()
        parser.add_argument("-f","--file", help="insert file")
        args = parser.parse_args()
        if not args.file:
                print "Insert a File with emails"
        else:
                search_leak(args.file)
```

Automatización de scripts de Inteligencia - DNSTwist

DNSTwist es una librería open-source creada para el análisis de amenazas de Typosquatting y suplantación de dominios.

https://github.com/elceef/dnstwist

Comando ejecución

python dnstwist.py -b -w -g -m google.es

Arquitectura de una plataforma de Inteligencia

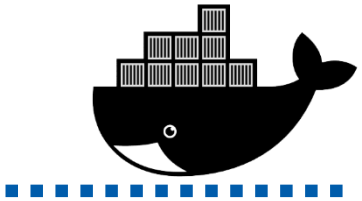
JORNADAS DE SEGURIDAD Y CIBERDEFENSA CIBERSEG17

- ¿Quiénes somos?
- Definiciones
- Búsqueda de información en OSINT
- Automatización de scripts de Inteligencia
- Arquitectura de una plataforma de Inteligencia

EY
Building a better working world

Instalación de Docker

```
echo "deb https://apt.dockerproject.org/repo ubuntu-xenial main"
>> /etc/apt/sources.list;  apt-get update;
sudo apt install docker-engine
```



Comandos Básicos

- docker images **Ver imágenes instaladas.**
- docker ps **Ver contenedores arrancados.**
- docker build -t nombre_imagen:versión **Crear una imagen de docker.**
- docker run -i -t nombre_imagen:versión --name nombre_contenedor /bin/bash **Arrancar un contenedor asociado a una imagen de docker.**
- docker rmi -f image_id / docker rm -f container id **Eliminar una imagen / contenedor de docker.**
- ctrl q + p **Salir de un contenedor de docker sin perder cambios.**
- docker attach nombre_contenedor **Acceder a contenedor.**
- docker commit nombre_contenedor nombre_imagen:versión **Guardar cambios en el contenedor y creación de una nueva imagen.**
- docker save nombre_imagen:version > imagen.tar **Exportar imagen en un tar.**
- docker load < imagen.tar **Importa la imagen desde un tar.**
- docker cp fichero_a_copiar hash_contenedor:/root/fichero_a_copiar **Copia un archivo desde máquina anfitriona a contenedor docker.**

```
FROM ubuntu:latest

# Instalamos las actualizaciones
RUN apt-get update
RUN apt-get upgrade -y

#Instalamos cliente y servidor de MySQL
RUN DEBIAN_FRONTEND=noninteractive apt-get -y install mysql-client mysql-server curl
RUN apt-get install -y inetutils-tools inetutils-ping nano wget
RUN apt-get install -y net-tools
#Se habilita el acceso remoto desde MySQL
RUN sed -i -e"s/^bind-address\s+=\s+127.0.0.1/bind-address = 0.0.0.0/" /etc/mysql/my.cnf


# Se crea el volumen
VOLUME ["/opt/mysql"]

# Añadimos el archivo con la base de datos y las tablas.
ADD ./bbdd.sql /var/db/bbdd.sql

#Añadimos la configuracion estandar
ENV user admin
ENV password admin
ENV url file:/var/db/bbdd.sql
ENV right WRITE

#Ejecutamos el script
ADD ./start.sh /usr/local/bin/start.sh
RUN chmod +x /usr/local/bin/start.sh

# Habilitamos el puerto 3306 de MySQL
EXPOSE 3306


#Iniciamos el motor de MySQL

CMD ["/usr/local/bin/start.sh"]
```

Archivo Dockerfile

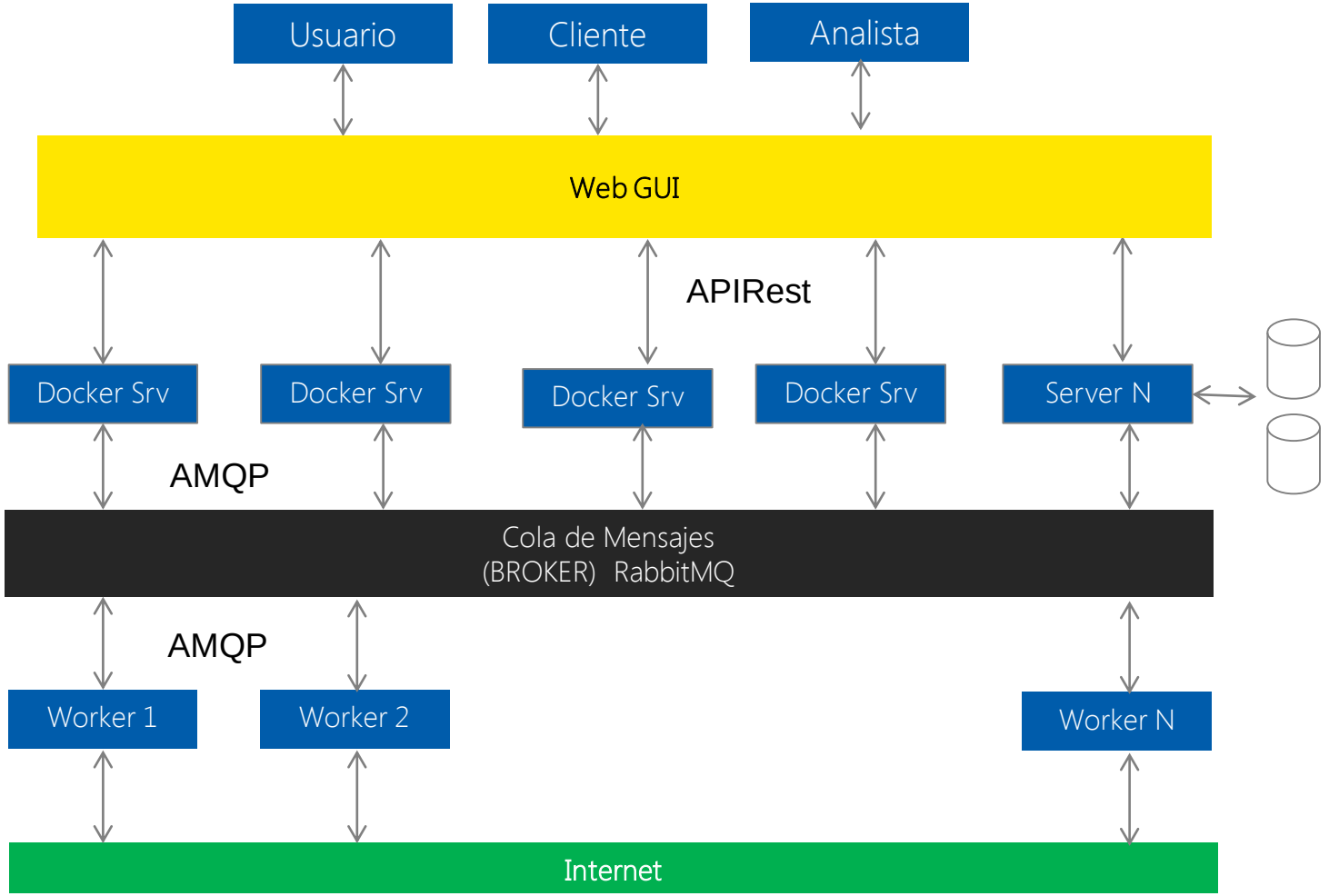
```
#!/bin/bash

# This script starts the database server.
echo "Creando el usuario $user para la base de datos desde la url $url"

# Import database if provided via 'docker run --env url="http:/ex.org/db.sql"'
echo "Añadiendo datos a MySQL"
service mysql start
#/usr/sbin/mysqld
sleep 5
curl $url -o import.sql

mysql --default-character-set=utf8 < import.sql
rm import.sql
mysqladmin shutdown
echo "finished"

# Now the provided user credentials are added
#/usr/sbin/mysqld &
service mysql start
sleep 5
echo "Creando usuario"
echo "CREATE USER '$user' IDENTIFIED BY '$password'" | mysql --default-character-set=utf8
echo "REVOKE ALL PRIVILEGES ON *.* FROM '$user'@'%'; FLUSH PRIVILEGES" | mysql --default-character-set=utf8
echo "GRANT SELECT ON *.* TO '$user'@'%'; FLUSH PRIVILEGES" | mysql --default-character-set=utf8
echo "Finalizado"

if [ "$right" = "WRITE" ]; then
echo "añadiendo permisos al usuario"
echo "GRANT ALL PRIVILEGES ON *.* TO '$user'@'%' WITH GRANT OPTION; FLUSH PRIVILEGES" | mysql --default-character-set=utf8
fi

# And we restart the server to go operational
mysqladmin shutdown
echo "Iniciando MySQL Server"
service mysql start
#/usr/sbin/mysqld
```

Archivo start.sh

Creación de una imagen con MySQL docker build -t taller_mysql:latest .

Creación del contenedor docker run -it -p 3306:3306 taller_mysql bash

Script a ejecutar dentro del contenedor sh /usr/local/bin/start.sh


Usuario
Cliente
Analista
Web GUI
APIRest
Docker Srv
Docker Srv
Docker Srv
Docker Srv
Server N
AMQP
Cola de Mensajes
(BROKER) RabbitMQ
AMQP
Worker 1
Worker 2
Worker N
Internet

Recursos para estudiantes

https://education.github.com/pack

GitHub Education

aws educate

DigitalOcean

UDACITY

Microsoft Imagine

axosoft GitKraken

CrowdFlower

FLATIRON SCHOOL

namecheap

SendGrid

DATADOG

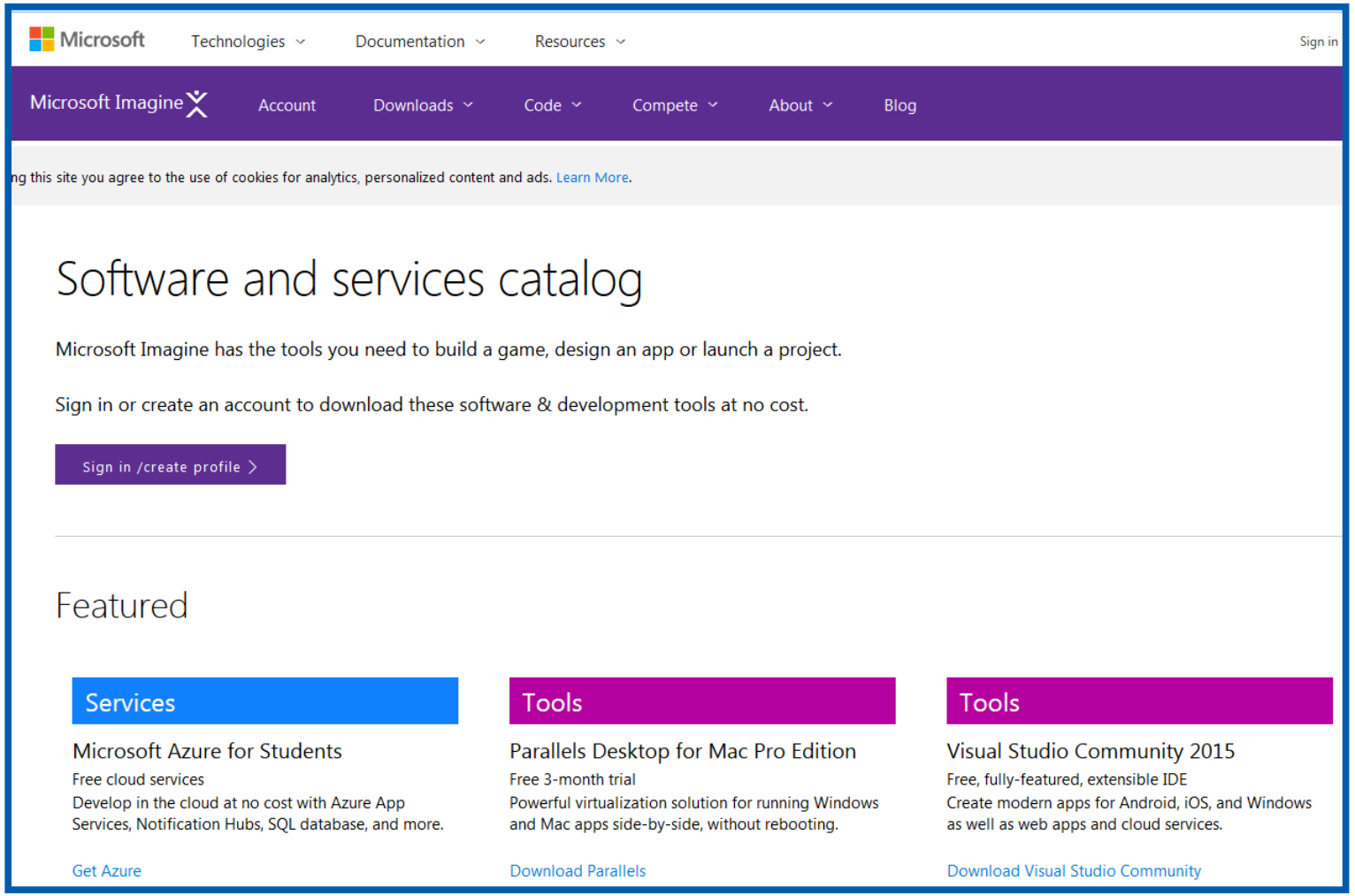
bitnami

ATOM

dnsimple

Recursos para estudiantes

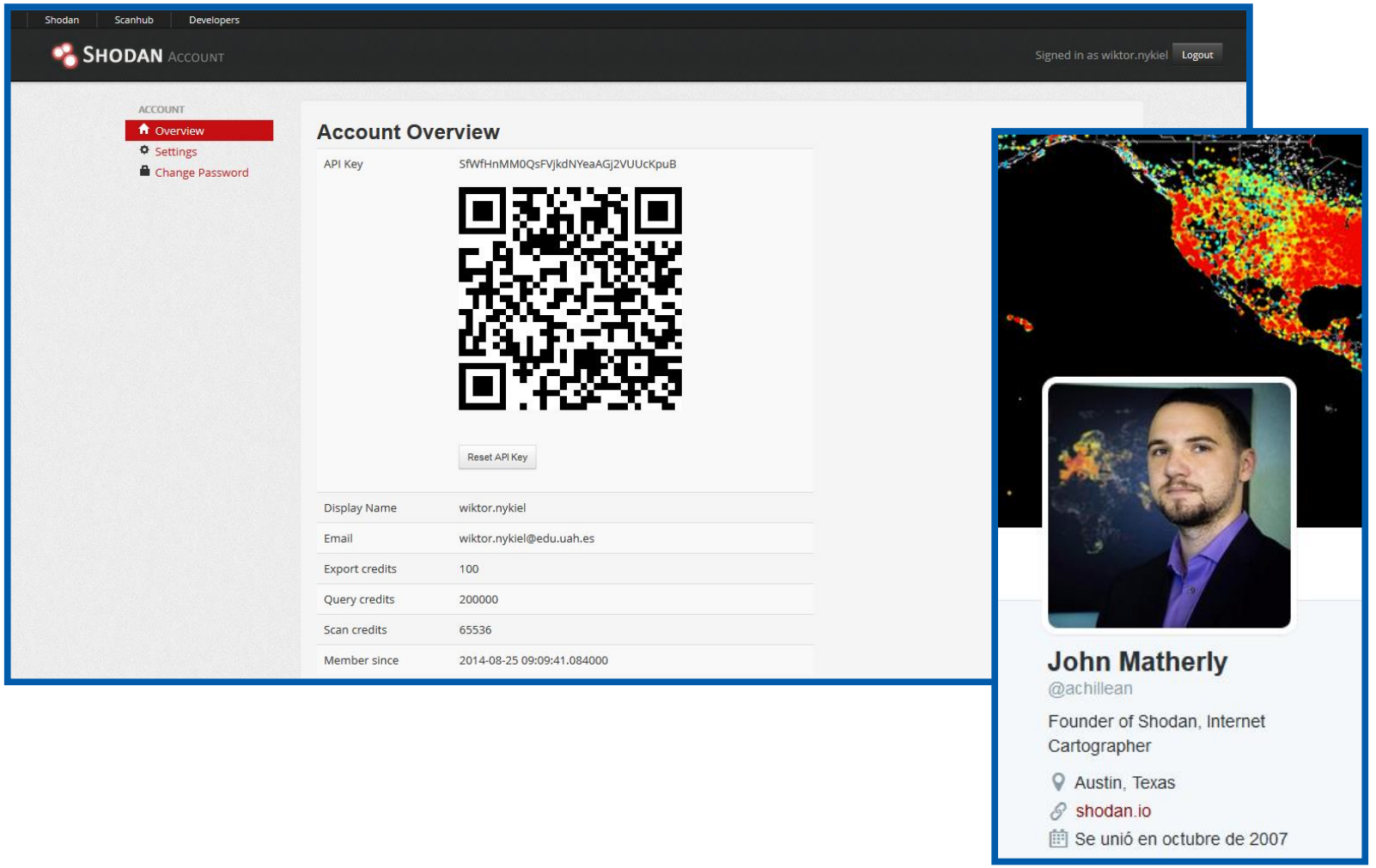
https://imagine.microsoft.com



SHODAN

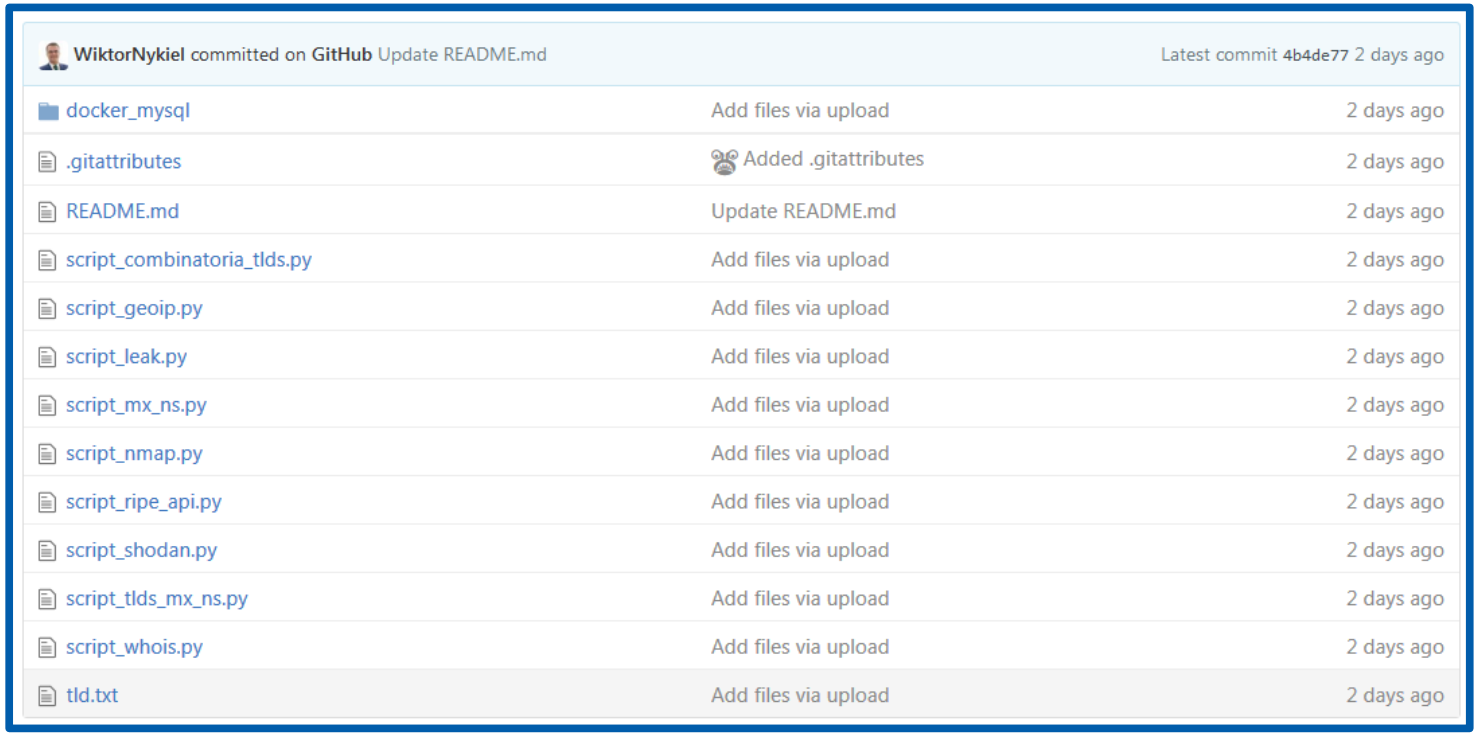
https://shodan.io

- Crear cuenta de Shodan.
- Solicitar upgrade por mail a jmath@shodan.io



GitHub

https://github.com/ey-ciberseg-wiktor-nykiel-ivan-portillo/scripts-taller-creacion-tools-de-inteligencia



WiktorNykiel committed on **GitHub** Update README.md		Latest commit 4b4de77 2 days ago
docker_mysql	Add files via upload	2 days ago
.gitattributes	Added .gitattributes	2 days ago
README.md	Update README.md	2 days ago
script_combinatoria_tlds.py	Add files via upload	2 days ago
script_geoip.py	Add files via upload	2 days ago
script_leak.py	Add files via upload	2 days ago
script_mx_ns.py	Add files via upload	2 days ago
script_nmap.py	Add files via upload	2 days ago
script_ripe_api.py	Add files via upload	2 days ago
script_shodan.py	Add files via upload	2 days ago
script_tlds_mx_ns.py	Add files via upload	2 days ago
script_whois.py	Add files via upload	2 days ago
tld.txt	Add files via upload	2 days ago

Talent Management

Manda tu CV a: eytalent@es.ey.com

http://www.ey.com/es/es/careers/students

Universidad de Alcalá

JORNADAS DE SEGURIDAD Y CIBERDEFENSA CIBERSEG17

EY
Building a better working world

Wiktor Nykiel

wiktor.nykiel@es.ey.com

wiktornykiel

@wiktornykiel

Iván Portillo

ivan.portillomorales@es.ey.com

ivanportillomorales

@ivanPorMor

Universidad de Alcalá

JORNADAS DE SEGURIDAD Y CIBERDEFENSA CIBERSEG17

EY Building a better working world