



GPenT0ols

Aquest projecte esta creat per en Marc Hortelano en el curs de Grau Superior

Abril 2021

Tutor: Victor Marquina

-
Grau Superior - Administració de Sistemes Informàtics en Xarxa
especialitat en Ciberseguretat

Índice

1. Abstract	2
2. Introducció	3
3. Objectius	4
4. Anàlisi de la programació	5
5. Resultats	6
5.1. Instal·lació	6
5.1.1. Per utilitzar python3	6
5.1.2. Docker (Recomenada)	7
5.2. Funcionament	8
5.2.1. Codi	8
5.3. Utilització	28
5.4. Exemples d'ús	31
6. Conclusions	46
6.1. Línies de futur	46
6.2. Problemes trobats	46

1. Abstract

When you start a project you look for precision, but I think the most important thing for me is how you do it. It's how you do it. My main goal is to enjoy learning about this subject as much as possible. The topic I have chosen has been very complicated to define because it can be very broad and sometimes you can get saturated with so much information you find on the Internet. First of all I am going to write about my motivations. Why I have chosen this final project and not another one, since I have had in mind several alternatives and all of them very similar.

Then I would like to show how I have done the project, the steps I have followed and show the final result. Add that my goal of this project is that other people can use it as they want. Which I will show how I have planned it so that it has no security problems as well as how to use it.

As I said before I hope to enjoy and assimilate as much as I can doing this kind of work because I think it helps me a lot to get out of my comfort zone and this kind of challenges makes learning much more fun.

2. Introducció

Feia temps que volia crear-me una eina automatitzada per poder realitzar pentestings, a més a més treballar una mica el desenvolupament web i conèixer noves tecnologies també estava dins el meu pla. La idea del projecte es realitzar un entorn de pentesting de forma gràfica mitjançant una web. Aquest entorn és una infraestructura en Docker on recolliré totes les eines que son útils per a mi. Em permetrà realitzar una auditoria de pentesting i desplegar-la de forma còmode i ràpida.

Un pentest consisteix en realitzar un procés d'avaluació coordinat. La prova implica una varietat d'elements, però per a simplificar l'explicació, un individu o equip contractat accedeix al sistema, avalua tot el sistema cercant vulnerabilitats o febleses a través d'una metodologia predefinida, aquestes vulnerabilitats són explotades de manera controlada i permet identificar el risc per a l'organització.

La idea s'origina en la necessitat d'automatitzar i les ganes d'aprendre o millorar un llenguatge de programació, concretament Python. Com he comentat anteriorment aquest projecte no ha set la primera idea que he tingut, ja que abans han passat varies molt similars i molt interessants. Una d'aquestes i que m'ha ajudat molt a obtenir aquesta idea final, era crear un plugin del Framework caldera. Caldera és un framework que utilitza les tàctiques de MITRE ATT&CK, i s'utilitza tant per red team com per blue. Aquest inici em va permetre entendre una mica el funcionament de com estava programat. Però aquesta idea no m'acabava de convecer, ja que principalment volia crear una eina per realitzar pentesting i no de red team. Molta gent ho engloba tot i realment són metodologies diferents. Més endavant explicaré que signifiquen aquests conceptes, però detallar aquesta part ho he trobat important perquè ha set de gran ajuda per obtenir la idea final. Finalment vaig decidir crear la meva pròpia eina des de zero, anomenada GPenTools. Com he dit, ho escriure principalment amb Python tant l'entorn gràfic com la majoria de scripts. El que faré serà una web on hi hagi una shell interactiva i tindrè totes les eines de pentesting que consideri. Aquesta web et permetrà modificar ràpidament el script i també crear tasques automatitzades. Per la creació d'aquesta eina m'ha servit d'inspiració altres eines com [1] [crackmapexec](#) i [2] [autorecon](#).

Tota la informació la compartiré en el meu repositori de Github, ja que ho vull fer de forma pública i que qualsevol persona la pugui utilitzar.

Enllaç del repositori: <https://github.com/Th3FirstAvenger/GPenTools>

3. Objectius

Per poder executar aquest projecte, ha set necessari complir uns requisits mínims. Aquest requisit seria comptar amb les bases de Linux i tenir uns coneixements de pentesting. Els objectius que m'agradaria assolir serien aconseguir una primera experiència desenvolupant una eina automatitzada totalment pròpia amb el llenguatge de programació de python, poder aprofitar aquesta eina per implementar-la en les auditories i seguir treballant amb ella per aconseguir millores. Per complir els objectius ha set necessari una primera fase de planificació, aquesta part consider-ho que ha set la més important ja que soc una persona que li agrada realitzar inversions a llarg termini i tenir una bona preparació.

Com bé deia Abraham Lincoln "Give me six hours to chop down a tree and I will spend the first four sharpening the ax. "Dona'm sis hores per tallar un arbre i em passaré les quatre primeres afilant la destrat.

Al tenir poca experiència en desenvolupar un projecte de programació a llarg termini, com es normal he comès alguns errors i imprevistos. Sent veritat, que gràcies a l'haver planificat una idea inicial i ben estructurada he rectificat a temps sense que hagi complicat molt el projecte. Quan dic que no tinc experiència vull fer referència que jo en l'àmbit de programació, només he realitzat scripts automatitzats i creació d'algun exploit web. Aquests a nivell bàsic.

4. Anàlisi de la programació

La planificació del projecte ha estat estructurada de la següent forma:

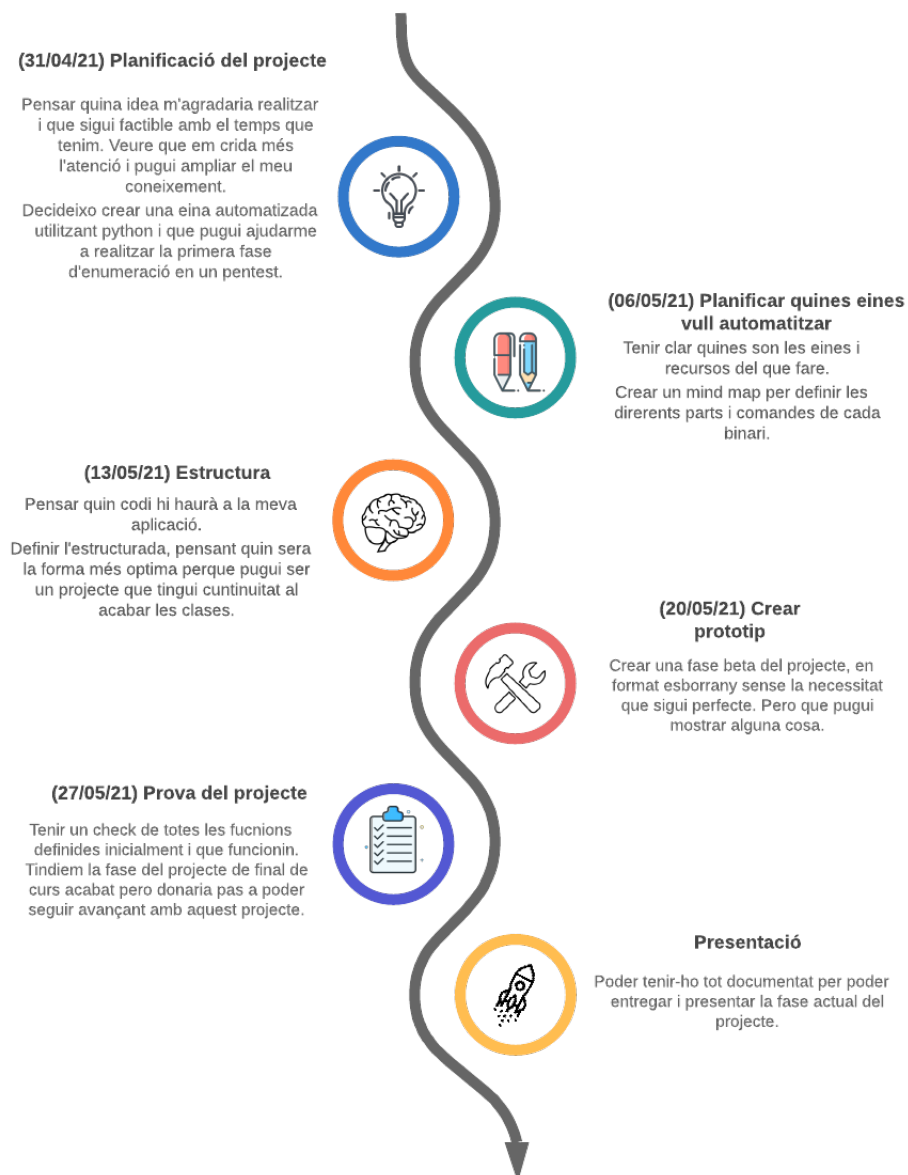


Figura 1: Cronograma

5. Resultats

En aquest apartat compartiré el codi i les instruccions per poder utilitzar la eina GPenT0ols. Tota la informació es troba en el meu repositori de github que he anunciat anteriorment. Per realitzar la instal·lació recomano que es faci amb docker ja que et permet tenir la última versió del projecte amb totes les eines i llibreries necessàries. Apart aquesta eina únicament ocupa 3 GB en la que cada vegada que es utilitzada s'elimina automàticament a no se que utilitzis configuracions que no s'indiquen a la instal·lació.

5.1. Instal·lació

Primer de tot es necessari tenir-ho descarregat

```
git clone https://github.com/Th3FirstAvenger/GPenT0ols.git /opt/GPenT0ols
cd /opt/GPenT0ols
```

5.1.1. Per utilitzar python3

En aquest cas ja tindriem l'eina descarregada. Faltaria descarregar i instal·lar les dependències mínimes per poder utilitzar l'eina. Per tenir-ho es pot fer us de la següent comanda:

** Aquesta comanda es valida per kali linux, en altres distribucions es probable que hagi de realitzar una instal·lació manual.*

```
apt-get update && \
    apt-get install -y \
    python3 \
    python3-pip \
    git \
    nmap \
    masscan \
    smbmap \
    whatweb \
    snmp \
    wget \
    nbtscan \
    wpscan \
    enum4linux \
    nikto \
    ffuf \
    golang \
    python3-venv \
    crackmapexec \
    seclists
```

El següent pas ja es realitzar l'execució.

```
python3 AutoGPenT0ols.py -h
```

5.1.2. Docker (Recomenada)

A continuació mostraré els passos per instal·lar l'eina amb docker. Per la creació de la imatge haurem d'executar la següent comanda.

```
docker build -t capitanj4ck/gpentools .
```

Per fer l'execució es bestant facil i com he indicat abans d'aquesta forma l'eina no es guarda i s'elimina quan acaba l'execució. Per iniciar utilitzariem la següent comanda:

```
docker run --rm -it -v /tmp/gpt_report:/tmp/gpt_report capitanj4ck/gpentools -h
```

Recomano afegir el següent al·lies per fer l'execució d'una forma més ràpida:

```
alias gpt="docker run --rm -it -v /tmp/gpt_report:/tmp/gpt_report capitanj4ck/gpentools"
```

5.2. Funcionament

Primer de tot explicaré com esta estructurada l'eina. M'agradaria donar molta més énfasi a l'apartat del codi ja que la web encara enstà en desenvolupament. He prioritzat aprendre el funcionament de python abans que flask, l'objectiu inicial era tenir una eina propia.

A l'executar l'eina ens permet realitzar diferents funcions. Com he dit anteriorment, el que ens permet realitzar aquesta eina es enumerar qualsevol equip de forma bàsica. Com que la meua intenció es seguir treballant amb aquesta eina, inicialment vaig planificar una estructura que em permetés poder desenvolupar el projecte a llarga durada i no fos únicament un fitxer python que ho realitzes tot. Un dels objectius era entendre com desenvolupar una eina de forma professional i no únicament teclejar i entendre el codi. Per aquest motiu vaig estar preguntat a professionals i veien diferents programes realitzats en python. D'aquesta forma em van servir d'inspiració diferents programes i amb aquesta ajuda vaig aconseguir planificar una estructura que fins el dia d'avui no he tingut molt problemes.

Per tant procediré en mostrar com la tinc estructurada.

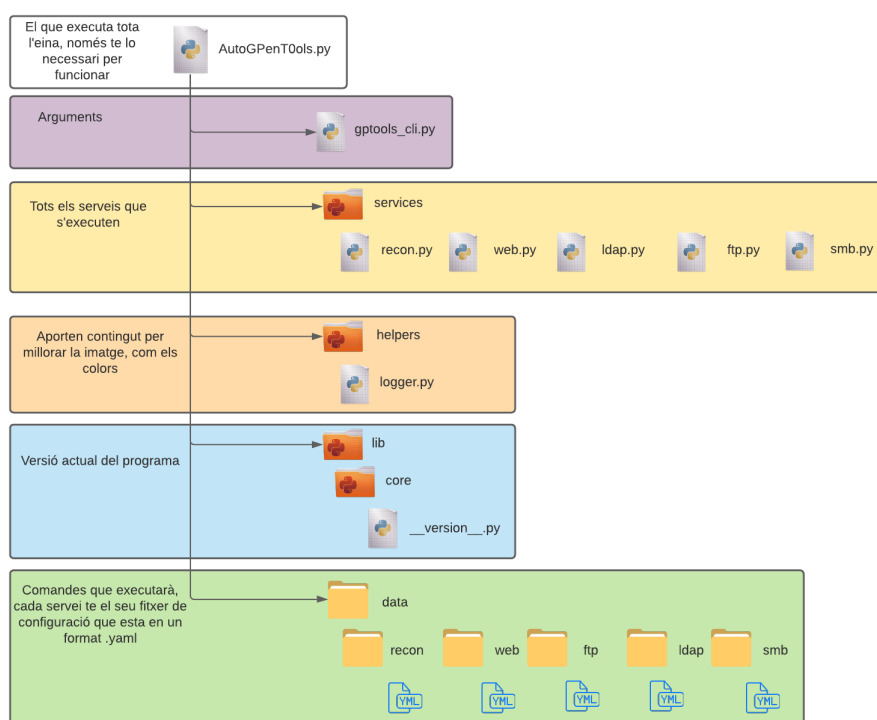


Figura 2: Estructura de l'eina

5.2.1. Codi

Veiem que inicialment tenim un fitxer main, aquest em permet cridar les diferents llibreries creades i cada llibreria té les llibreries necessàries i així no omplo el primer fitxer de codi que únicament em pot servir per a un servei. El contingut d'aquest fitxer seria el següent: AutoGPenT0ols.py

```

1  #!/usr/bin/python3
2  ##
3
4      #
5      ##
6      #####
7      #####
8      #####
9      #####
10     #
11     # Author : CapitanJ4ck
12     ##
13
14     import signal
15     from sys import exit
16     import os
17     import time
18     import subprocess
19     from pwn import *
20     from gptools_cli import gen_cli_args
21     from services.recon import recon
22     from services.web import web
23     from services.smb import smb
24     from services.ldap import ldap
25     from services.ftp import ftp
26
27     ## Detect Contrl + C
28     def signal_handler(key, frame):
29         # Handle any cleanup here
30         exit = log.progress("SIGINT or CTRL-C detected.")
31         exit.status("Exiting...")
32         time.sleep(1)
33         exit.failure("Exiting gracefully")
34         sys.exit(1)
35
36     signal = signal.signal(signal.SIGINT, signal_handler)
37
38
39     # Vars
40
41     # Make directories
42
43     def mkdir(dir_name):
44
45         directory_created = True

```

```

46
47     try:
48         os.makedirs(dir_name)
49
50     except OSError:
51         directory_created = False
52
53     return directory_created
54
55     # Managment function, we can build directories for save outputs
56
57     def build_infraestructure(dir_name,output):
58
59         infra = log.progress("Managment")
60
61         infra.status("Building structure on ")
62
63
64         if not os.path.exists(dir_name):
65             directory_created = mdir(dir_name)
66             if directory_created:
67                 infra.success("Succesfully created the directory {}".format(dir_name))
68             else:
69                 infra.failure("Creation of the directory {} failed".format(dir_name))
70         else:
71             infra.success("Directory {} already exist".format(dir_name))
72
73     def run(dir_file, command,service,debug):
74         out = dir_file + "out.txt"
75         err = dir_file + "err.txt"
76         #   parsed_command = command ## Util working other commands
77         parsed_command = []
78
79         for c in command:
80             parsed_command.append(c.replace('.', ' '))
81
82
83         p = log.progress(service)
84
85         with open(out,'w+') as fout:
86             with open(err,'w+') as ferr:
87
88                 p.status("Running")
89                 try:
90                     out=subprocess.call(parsed_command,stdout=fout,stderr=ferr)
91                     # reset file to read from it

```

```
92         fout.seek(0)
93         # save output (if any) in variable
94         output=fout.read()
95         if debug:
96             print(output)
97
98         # reset file to read from it
99         ferr.seek(0)
100        # save errors (if any) in variable
101        errors = ferr.read()
102        if out != 0:
103            p.failure("Something wrong")
104        else:
105            p.success("Succesfully!")
106    except:
107        p.failure("Command timeout")
108
109 def main():
110
111     # Get args from Namespace type
112     args = vars(gen_cli_args())
113
114     # Get service
115     service = args['services']
116     target = args['target']
117     debug = args['verbose']
118     out_path = os.path.join(args['path'],(os.path.join(args['target'],service)))
119     config_path = os.path.join(os.getcwd(), os.path.join("data",service))
120
121     web_path = out_path # CHANGE WHEN WEB WORKS
122
123     # Build infraestructure for save output
124     build_infraestructure(out_path,config_path)
125
126     #print(args) # debug
127     # Start progress
128     service_progress = log.progress(service)
129
130     ## Recon scanner
131     if 'recon' == service:
132         scanner = recon(args,config_path,out_path)
133     ## Web scanner
134     elif 'web' == service:
135         scanner = web(args,config_path,out_path)
136     ## smb scanner
137     elif 'smb' == service:
```

```

138     scanner = smb(args,config_path,out_path)
139     ## ldap scanner
140     elif 'ldap' == service:
141         scanner = ldap(args,config_path,out_path)
142     ## ftp scanner
143     elif 'ftp' == service:
144         scanner = ftp(args,config_path,out_path)
145
146     print("--  --")
147     for description, command in scanner.items():
148         service_progress.status("{}".format(command))
149         if args['show_commands']:
150             print(command)
151             time.sleep(2) # Check without exec
152         else:
153             run(web_path,command.split(),description,debug)
154
155 if __name__ == '__main__':
156     main()

```

En aquest fitxer mostrat anteriorment el que fa es agafar els arguments passats pel fitxer gpentools_cli.py. A partir del que passa l'usuari executarà una cosa o altra. Més endavant mostraré el que mostra i el seu funcionament.

gpentools_cli.py

Aquest fitxer fa ús del mòdul argparse, aquest mòdul em permet agafar els arguments de forma fàcil i compatible de moltes formes. També he escollit aquest mòdul perquè em permet crear com submenús i aprofundir més en detall cada servei. Aquest mòdul el comparteixo a la webgrafia, ja que ha set complexa entendre el seu funcionament però finalment he aconseguit implementar-ho.

El contingut del fitxer gpentools_cli és el següent:

```

1  import argparse
2  import sys
3  from argparse import RawTextHelpFormatter
4  from lib.core.__version__ import __version__
5  from helpers.logger import highlight
6
7  def gen_cli_args():
8
9      VERSION = __version__
10     CODENAME = 'CapitanJ4ck'
11
12
13     parser = argparse.ArgumentParser(description="""
14                                     #
15                                     ##                                     ###

```

```

16 #####
17 #####
18 #####
19 #####
20 #####
21 #
22 {}: {}
23 {}: {}
24 """.format(highlight('Version', 'red'),
25             highlight(VERSION),
26             highlight('Codename', 'red'),
27             highlight(CODENAME)),
28
29             formatter_class=RawTextHelpFormatter,
30             epilog="We are in... Let the hacking begin!")
31
32 parser.add_argument("-t", type=int, dest="threads", default=100,
33                     help="set how many concurrent threads to use (default: 100)")
34 parser.add_argument("--verbose", action='store_true', help="enable verbose output")
35 parser.add_argument("--show-commands", action='store_true', help="Just show commands")
36 parser.add_argument("--path", dest="path", default='/tmp/gpt_report/',
37                     help="Destination path (default: /tmp/gpt_report)")
38
39 std_parser = argparse.ArgumentParser(add_help=False)
40 std_parser.add_argument("target", nargs='?', type=str,
41                         help="(Target Required *) The target IP(s), range(s),
42                         CIDR(s), hostname(s), FQDN(s), file(s) containing a list of targets ")
43 # Type scanner group
44 scan_parser = argparse.ArgumentParser(add_help=False)
45 scan_parser.add_argument(
46     '--scanner',
47     help='Select scanner (default : full_scanner)',
48     nargs='?',
49     default = 'full_scanner'
50 )
51
52 # Introduce what info you have
53 have_info_parser = argparse.ArgumentParser(add_help=False)
54 have_info_parser.add_argument(
55     '--tags',
56     help='What do you have? [Creds, NoCreds, Hash, Shell] (Default: NoCreds)',
57     default = 'NoCreds',
58     nargs='?'
59 )
60
61 # wordlist group

```

```

62     wlist_parser = argparse.ArgumentParser(add_help=False)
63     wlist_parser.add_argument("-w", metavar="WORDLIST", dest='wordlist',
64                             nargs='+', help="set wordlist (Default SecList wordlist)")
65
66     # ssl group
67     ssl_parser = argparse.ArgumentParser(add_help=False)
68     ssl_parser.add_argument(
69         '--ssl',
70         help='usage of SSL/TLS requests',
71         action='store_true'
72     )
73
74     # credentials group
75     cred_parser = argparse.ArgumentParser(add_help=False)
76     cred_parser.add_argument("-u", metavar="USERNAME", dest='username',
77                             nargs='?', default=[], help="username(s) or file(s) containing usernames")
78     cred_parser.add_argument("-p", metavar="PASSWORD", dest='password',
79                             nargs='?', default=[], help="password(s) or file(s) containing passwords")
80     cred_parser.add_argument("-H", metavar="HASH", dest='HASH',
81                             nargs='?', default=[], help="Pass The hash")
82
83     subparsers = parser.add_subparsers(title='services', dest='services',
84                                       description='available options')
85
86
87     # Arguments Recon
88
89     recon = subparsers.add_parser('recon', help='Initial recon',
90                                  parents=[std_parser, scan_parser])
91     ## Get new arguments and can introduce std_parser arguments
92     recon.add_argument(
93         '--all-ports',
94         help='scan all ports',
95         action='store_true'
96     )
97
98     recon.add_argument(
99         '--ports',
100        help='scan specific ports',
101        nargs='?'
102    )
103
104    recon.add_argument(
105        '--full',
106        help='Full recon scan',
107        action='store_true'

```

```

108         )
109
110
111     # Arguments WEB
112
113     web = subparsers.add_parser('web', help='Web server scanner',
114                                parents = [cred_parser, std_parser, wlist_parser, scan_parser, ssl_parser])
115
116     web.add_argument(
117         '--port',
118         help='scan specific port (Default 80)',
119         nargs='?',
120         default = '80'
121     )
122
123     web.add_argument(
124         '--file-path',
125         help='Specify to find the requested resource and start the enumeration with that route (
126         nargs='?',
127         default = '/'
128     )
129
130
131     web.add_argument(
132         '--cms',
133         help='What do you have? [Wordpress, Joomla, Drupal] (Default: NoCreds)',
134         default = 'NoCreds',
135         nargs='?'
136     )
137     # Arguments SMTP
138
139     smtp = subparsers.add_parser('smtp', help='smtp enumeration')
140
141     # Arguments SMB
142
143     smb = subparsers.add_parser('smb', help='Enum smb', parents = [cred_parser, std_parser, have_info_
144
145     smb.add_argument(
146         '--port',
147         help='scan specific port (Default 445)',
148         nargs='?',
149         default = '445'
150     )
151
152
153     # Arguments FTP

```



```

154
155     ftp = subparsers.add_parser('ftp', help='FTP enum',
156                               parents = [cred_parser, std_parser, have_info_parser, ssl_parser])
157
158     ftp.add_argument(
159         '--port',
160         help='scan specific port (Default 21)',
161         nargs='?',
162         default = '21'
163     )
164
165
166     # Arguments LDAP
167
168     ldap = subparsers.add_parser('ldap', help='LDAP enum',
169                                 parents = [cred_parser, std_parser, have_info_parser])
170
171     ldap.add_argument(
172         '--port',
173         help='scan specific port (Default 389)',
174         nargs='?',
175         default = '389'
176     )
177
178     # Arguments SNMP
179
180     snmp = subparsers.add_parser('snmp', help='Enum SNMP', parents = [cred_parser, std_parser])
181
182     if len(sys.argv) == 1:
183         parser.print_help()
184         sys.exit(1)
185
186     args = parser.parse_args()
187
188     check = vars(args)
189
190
191     if not check['target']:
192         if check['services'] == 'recon':
193             recon.print_help()
194         elif check['services'] == 'web':
195             web.print_help()
196         elif check['services'] == 'smb':
197             smb.print_help()
198         elif check['services'] == 'smtp':
199             smtp.print_help()

```

```
200         elif check['services'] == 'ftp':
201             ftp.print_help()
202         elif check['services'] == 'ldap':
203             ldap.print_help()
204         elif check['services'] == 'snmp':
205             snmp.print_help()
206         else:
207             parser.print_help()
208         sys.exit(1)
209
210     return args
```

Una vegada s'han passat els arguments de forma correcta ja passen a executar els serveis. Cada servei té el seu fitxer i està creat com una llibreria. En el fitxer Main, veiem que crida unes funcions i aquestes funcions són les que es troben dins de cada servei. Sí que la metodologia és molt similar a cada fitxer però a poc a poc aniré implementant millores. Per exemple els fitxers recon i web són molt similars en canvi amb samba i ldap ja són diferents. El motiu es que he estat realitzant proves per veure quina seria la més eficaç i de moment crec que l'estructura que mantindré es la forma que estan configurats la segona opció. En primer lloc la primera opció, fa ús dels fitxers de configuració .yaml però l'estructura es diferent. Podem observar que la primera opció s'ha de passar un argument i serà el mateix que cridarà en el fitxer yaml, en canvi la segona opció utilitzem sempre els mateixos tags i busca a través d'aquest tags. Si que pel servei de recon la millor opció es l'actual pero per l'apartat web seria modificar-ho i utilitzar els tags, però aquesta part la fa molt més complexa, ja que s'hauria de planificar quins serien aquest tags i com ho organitzaria per aquest motiu de moment ho mantinc com està.

Seguidament mostraré el codi dels serveis. Aquests serveis són importats en l'inici del codi Main i permet fe ús de les funcions que es troben dins d'aquestes llibreries. Les funcions són passar-l'hi la comanda que haurà d'executar, retorna un array amb la descripció de la comanda i la comanda. La informació l'agafa del seu fitxer de configuració, el motiu que ho he realitzat d'aquesta forma es perquè em permet ampliar les comandes sempre que ho vegi convenient i no tindrà cap problema, ja que al final retorno un array amb tota la informació.

Inicio la part dels serveis mostrant el codi i el seu fitxer de configuració.

recon.py

```
1  #!/usr/bin/env python3
2  ### This script enum all ports, services and vulnerabilities with nmap
3  import yaml
4  import os
5
6
7
8  def recon(all_info, recon_path, out_path):
9      target = all_info['target']
10     info_data = all_info['scanner']
```

```

11
12     # get path information
13     recon_data = os.path.join(recon_path, "recon_config.yaml")
14     command_info = {}
15     with open(recon_data, 'r') as unparsed:
16         try:
17             recon_data = yaml.safe_load(unparsed)
18         except yaml.YAMLError as exc:
19             print(exc)
20
21     list_data = recon_data[info_data].keys()
22
23     for options in list_data:
24         descr = recon_data[info_data][options]['description']
25         out_file = os.path.join(out_path, '.'.join((options, 'txt')))
26         cmd = (recon_data[info_data][options]['commands'].replace('${ out_dir }',
27             out_file).replace('${ target }', target))
28         command_info[descr] = cmd
29
30     return command_info

```

recon_config.yaml

La configuració veurem que hi ha unes variables `${ nom }` que em permeten fer el replace en el codi de python3. He trobat varies formes pero aquesta era la que més s'ajustaba el que volia.

```

1  ```yaml
2  full_scanner:
3      nmap_quick_open:
4          description: "Quick open ports with nmap"
5          commands: |
6              nmap -F --open -oN ${ out_dir } ${ target }
7
8      nmap_quick_versions:
9          description: "Quick scan with nmap"
10         commands: |
11             nmap -sV -sC --open --version-all -oN ${ out_dir } ${ target }
12
13     nmap_full:
14         attack_technique: nmap
15         description: "Scanning all ports with nmap and return service information"
16         commands: |
17             nmap -sS -sV -sC -O -p- --min-rate 5000 -n -oN ${ out_dir } ${ target }
18     nmap_fast_udp:
19         attack_technique: nmap
20         description: "Scanning udp ports"

```

```

21     commands: |
22         nmap -F -sU -sV -T 4 -oN ${out_dir} ${target}
23
24     masscan_full:
25         attack_technique: masscan
26         description: "Scanning all ports with masscan"
27         commands: |
28             masscan --rate 10000 -p1-65535 --only-open -oL ${out_dir} ${target}
29     ## Not ready yet
30     nmap_custom:
31         nmap_allports:
32             attack_technique: nmap
33             description: "Scanning all ports with nmap"
34             commands: |
35                 nmap -p- --min-rate 5000 -n -oG ${out_dir} ${target}
36
37         nmap_ports:
38             attack_technique: nmap
39             description: "Scanning specific ports with nmap"
40             commands: |
41                 nmap -p ${ports} -sC -sV ${out_dir} ${target}
42     ...

```

A continuació mostro el que fa la funció [3] `argsparse`, i ens retorna la següent llista:

```

'threads': 100, 'verbose': False, 'path': '/tmp/autorecon/', 'services': 'recon', 'target': ['127.0.0.1'],
'all_ports': True, 'ports': None

```

web.py

En l'apartat web, trobem que el funcionament es molt similar. L'únic que prèviament pots passar-li el parametre `ssl` i et transforma el `http` a `https`.

```

1  #!/usr/bin/env python3
2  ### This script enum all ports, services and vulnerabilities with nmap
3  import yaml
4  import os
5  ## need port, user, passwd, type scanner, CMS, fuzzing, layer (http/https), wordlist, url
6
7  def web(all_info, recon_path, out_path):
8      target = all_info['target']
9      info_data = all_info['scanner']
10     port = all_info['port']
11     url_path = all_info['file_path']
12     cms = all_info['cms']
13
14
15     # Check https or http scheme
16     scheme = 'http'

```

```

17     if all_info['ssl']:
18         scheme = 'https'
19
20     url = '{0}://{1}:{2}{3}'.format(scheme,target, port, url_path)
21
22     ## Set wordlist
23     wordlist = all_info['wordlist']
24
25     if wordlist == None:
26         wordlist = '/usr/share/seclists/Discovery/Web-Content/big.txt'
27
28
29     # get path information
30     web_data = os.path.join(recon_path, "web_config.yaml")
31     command_info = {}
32
33
34
35     with open(web_data, 'r') as unparsed:
36         try:
37             web_data = yaml.safe_load(unparsed)
38         except yaml.YAMLError as exc:
39             print(exc)
40
41     list_data = web_data[info_data].keys()
42
43     for options in list_data:
44         descr = web_data[info_data][options]['description']
45         out_file = os.path.join(out_path, '.'.join((options, 'txt')))
46         cmd = (web_data[info_data][options]['commands'].replace('${ out_dir }',
47             out_file).replace('${ target }', target).replace('${ port }',
48             port).replace('${ url }', url).replace('${ wordlist }',wordlist))
49         command_info[descr] = cmd
50
51     return command_info

```

web_config.yaml

```

1  ```yaml
2  full_scanner:
3      nmap_scan:
4          attack_technique: nmap
5          description: "Scanning nmap with scripts banner,(http* or ssl*) and not
6              (brute or broadcast or dos or external or http-slowloris* or fuzzer)"
7          commands: |
8              nmap -sV -p ${ port } --script="banner,(http*.or.ssl*).and.not.

```

```

9         (brute.or.broadcast.or.dos.or.external.or.http-slowloris*.or.fuzzer)" -oN
10         ${ out_dir }} ${ target }}
11
12     curl_index:
13         attack_technique: curl-index
14         description: "Enum index web content"
15         commands: |
16             curl -sSik ${ url }} -m 10 | tee ${ out_dir }}
17
18     whatweb_index:
19         attack_technique: whatweb
20         description: "Enum versions webs"
21         commands: |
22             whatweb -v -a 3 ${ target }}
23     nikto_scaner:
24         attack_technique: nikto
25         description: "Nikto detect misconfiguration, risky files, etc."
26         commands: |
27             nikto -h ${ url }} | tee ${ out_dir }}
28
29     fuzzing_ffuf:
30         attack_technique: ffuf
31         description: "Fuzzing directories and files"
32         commands: |
33             ffuf -w ${ wordlist }} -u ${ url }}FUZZ -c -e '.php,.asp,.aspx,.txt,.html,.zip'
34             -o ${ out_dir }}
35     wp:
36         wordpres_scanner:
37             attack_technique: wpscan
38             description: "Fuzzing directories and files"
39             commands: |
40                 wpscan --url ${ url }} -e vp,vt,tt,cb,dbe,u,m --plugins-detection aggressive
41                 --plugins-version-detection aggressive -o ${ out_dir }}
42     ...

```

smb.py

Aquest script ja veurem que el que fa és diferent de la resta. Per arguments és passen uns tags que seran buscats en el fitxer de configuració. Fa referència a la informació que té l'atacant. Es a dir, si l'atacant no té informació, el més normal seria iniciar per una sessió nul·la. Una vegada ja es te credencials doncs és passa a enumerar aquesta informació amb l'usuari i contrasenya. I també hi ha l'opció de passar-li el hash, aquesta et permet de la mateixa forma executar comandes de forma remota com si tinguessis la contrasenya.

```

1  #!/usr/bin/env python3
2  import yaml
3  import os

```

```
4
5  # list only data passed
6  def get_tags(smb_data, tags):
7
8      list_data = smb_data.keys()
9      final_services = []
10
11     for parameter in tags:
12         for technique in list_data:
13             total = 0
14             for opt in smb_data[technique]['tags']:
15                 ecual = False
16                 if parameter.lower() == opt.lower():
17                     ecual = True
18                     total +=1
19                 if not ecual:
20                     break
21             if total == len(tags):
22                 final_services.append(technique)
23
24     return final_services
25
26  #return command
27
28  def smb(all_info, recon_path, out_path):
29      target = all_info['target']
30      port = all_info['port']
31      tags = all_info['tags'].split(',')
32      username = all_info['username']
33      password = all_info['password']
34
35      if len(username) == 0 or len(password) == 0:
36          username = ''
37          password = ''
38
39      # get path information
40      smb_data = os.path.join(recon_path, "smb_config.yaml")
41      command_info = {}
42
43      with open(smb_data, 'r') as unparsed:
44          try:
45              smb_data = yaml.safe_load(unparsed)
46          except yaml.YAMLError as exc:
47              print(exc)
48
49      services = get_tags(smb_data, tags)
```

```

50
51     for options in services:
52         descr = smb_data[options]['description']
53         out_file = os.path.join(out_path, '.'.join((options, 'txt')))
54         cmd = (smb_data[options]['commands'].replace('${ out_dir }',
55         out_file).replace('${ target }', target).replace('${ port }',
56         port).replace('${ username }', username).replace('${ password }', password))
57         command_info[descr] = cmd
58
59     return command_info

```

smb_config.yaml

```

1  ```yaml
2  smbc_null_session:
3      attack_technique: smbclient
4      description: "Samba null sessions enum directories"
5      tags:
6          - nocreds
7      commands: |
8          smbclient -L \\\\${{ target }}\ -N -p ${{ port }}
9  cme_null_session:
10     attack_technique: crackmapexec
11     description: "Enum null sessions with Crackmapexec"
12     tags:
13         - nocreds
14     commands: |
15         crackmapexec smb ${{ target }} -u '' -p '' --shares --timeout 5
16  enum4linux_null_session:
17     attack_technique: Enum4linux
18     description: "Enum null sessions with enum4linux"
19     tags:
20         - nocreds
21     commands: |
22         enum4linux -a ${{ target }} | tee ${{ out_dir }}
23  cme_dsa_session:
24     attack_technique: crackmapexec
25     description: "Enum directories and see if have shell"
26     tags:
27         - creds
28     commands: |
29         crackmapexec smb ${{ target }} -u '${{ username }}' -p '${{ password }}'
30         --shares --timeout 5
31  cme_passthehash:
32     attack_technique: crackmapexec
33     description: "Enum directories and see if have shell"

```



```

34     tags:
35         - hash
36     commands: |
37         crackmapexec smb ${ target } -u '${ username }' -H '${ hash }'
38         --shares --timeout 5
39
40     ...

```

ldap.py

```

1     #!/usr/bin/env python3
2     import yaml
3     import os
4     import ldap3
5
6     # Not ready yet
7     def enum_ldap(server, port):
8         server = ldap3.Server(server, port, use_ssl = True)
9         connection = ldap3.Connection(server)
10        if connection.bind():
11            info = server.info
12        else:
13            info = False
14
15    def get_tags(ldap_data, tags):
16
17        list_data = ldap_data.keys()
18        final_services = []
19
20        for parameter in tags:
21            for technique in list_data:
22                total = 0
23                for opt in ldap_data[technique]['tags']:
24                    ecual = False
25                    if parameter.lower() == opt.lower():
26                        ecual = True
27                        total +=1
28                    if not ecual:
29                        break
30                if total == len(tags):
31                    final_services.append(technique)
32
33        return final_services
34
35    def ldap(all_info, recon_path, out_path):
36        target = all_info['target']

```

```

37     port = all_info['port']
38     tags = all_info['tags'].split(',')
39     username = all_info['username']
40     password = all_info['password']
41
42     if len(username) == 0 or len(password) == 0:
43         username = ''
44         password = ''
45
46     # get path information
47     ldap_data = os.path.join(recon_path, "ldap_config.yaml")
48     command_info = {}
49
50     with open(ldap_data, 'r') as unparsed:
51         try:
52             ldap_data = yaml.safe_load(unparsed)
53         except yaml.YAMLError as exc:
54             print(exc)
55
56     services = get_tags(ldap_data, tags)
57
58     for options in services:
59         descr = ldap_data[options]['description']
60         out_file = os.path.join(out_path, '.'.join((options, 'txt')))
61         cmd = (ldap_data[options]['commands'].replace('${ out_dir }',
62             out_file).replace('${ target }', target).replace('${ port }',
63             port).replace('${ username }', username).replace('${ password }', password))
64         command_info[descr] = cmd
65
66     return command_info

```

ldap_config.yaml

```

1  ```yaml
2  ldap_nmap_session:
3      attack_technique: nmap
4      description: "Enum null sessions with nmap"
5      tags:
6          - nocreds
7      commands: |
8          nmap -sV -p ${ port } --script="banner,(ldap* or ssl*) and not
9              (brute or broadcast or dos or external or fuzzer)" -oN "${ out_dir }" ${ target }
10  ```

```

ftp

```
1      #!/usr/bin/env python3
2  import yaml
3  import os
4
5  def get_tags(ftp_data, tags):
6
7      list_data = ftp_data.keys()
8      final_services = []
9
10     for parameter in tags:
11         for technique in list_data:
12             total = 0
13             for opt in ftp_data[technique]['tags']:
14                 ecual = False
15                 if parameter.lower() == opt.lower():
16                     ecual = True
17                     total +=1
18                 if not ecual:
19                     break
20             if total == len(tags):
21                 final_services.append(technique)
22
23     return final_services
24
25 def ftp(all_info, recon_path, out_path):
26     target = all_info['target']
27     port = all_info['port']
28     tags = all_info['tags'].split(',')
29     username = all_info['username']
30     password = all_info['password']
31
32     if len(username) == 0 or len(password) == 0:
33         username = 'anonymous'
34         password = ''
35
36     # Check ftp or ftps scheme
37     scheme = 'ftp'
38     if all_info['ssl']:
39         scheme = 'ftps'
40
41     # get path information
42     ftp_data = os.path.join(recon_path, "ftp_config.yaml")
43     command_info = {}
44
45     with open(ftp_data, 'r') as unparsed:
```

```

46     try:
47         ftp_data = yaml.safe_load(unparsed)
48     except yaml.YAMLError as exc:
49         print(exc)
50
51     services = get_tags(ftp_data, tags)
52
53     for options in services:
54         descr = ftp_data[options]['description']
55         out_file = os.path.join(out_path, '.'.join((options, 'txt')))
56         cmd = (ftp_data[options]['commands'].replace('${ out_dir }',
57         out_file).replace('${ target }', target).replace('${ port }',
58         port).replace('${ username }', username).replace('${ password }',
59         password).replace('${ scheme }', scheme))
60         command_info[descr] = cmd
61
62     return command_info

```

ftp-config.yaml

```

1  ```yaml
2  ftp_anon_session:
3      attack_technique: nmap_ftp
4      description: "check anonymous user ftp"
5      tags:
6          - nocreds
7      commands: |
8          nmap -sC -sV -p ${ port } -on ${ out_dir } ${ target }
9  download_anonymous_session:
10     attack_technique: download_files
11     description: "downloading all files"
12     tags:
13         - nocreds
14     commands: |
15         wget -r ${ scheme }://${ target }:${ port } -P ${ out_dir }
16  download_creds_session:
17     attack_technique: download_files
18     description: "downloading all files"
19     tags:
20         - creds
21     commands: |
22         wget -r ${ scheme }://${ username }:${ password }@${ target
23         }:${ port } -P ${ out_dir }
24  ```

```



```
usage: AutoGPentTools.py recon [-h] [--scanner [SCANNER]]
[--all-ports] [--ports [PORTS]] [--full] [target]
```

positional arguments:

target (Target Required *) The target IP(s), range(s),
CIDR(s), hostname(s), FQDN(s), file(s) containing a list of targets

optional arguments:

-h, --help show this help message and exit
--scanner [SCANNER] Select scanner (default : full_scanner)
--all-ports scan all ports
--ports [PORTS] scan specific ports
--full Full recon scan

Una vegada tenim la primera informació podem porcedir en enumerar els serveis oberts. Jo inicio l'explicació en funcio del port, de menys a més gran. Iniciem pel port 22 - FTP. En aquest cas veurem que ens permet realitzar les següents operacions.

```
gpt ftp -h
```

```
usage: AutoGPentTools.py ftp [-h] [-u [USERNAME]] [-p [PASSWORD]] [-H [HASH]]
[--tags [TAGS]] [--ssl] [--port [PORT]] [target]
```

positional arguments:

target (Target Required *) The target IP(s), range(s),
CIDR(s), hostname(s), FQDN(s), file(s) containing a list of targets

optional arguments:

-h, --help show this help message and exit
-u [USERNAME] username(s) or file(s) containing usernames
-p [PASSWORD] password(s) or file(s) containing passwords
-H [HASH] Pass The hash
--tags [TAGS] What do you have? [Creds, NoCreds, Hash, Shell] (Default: NoCreds)
--ssl usage of SSL/TLS requests
--port [PORT] scan specific port (Default 22)

Seguim amb el port 80/443 - Web

El que ens permet es escollir el tipo de scanner en aquest cas per defecte hi ha el full_scanner pero podriem escollir el wp que es per una web wordpress. També veiem que ens permet passar la les credencials i la wordlist. Per defecte ja estan configurades aquestes variables per evitar errors. I el parametre ssl et permet afegir el https en la url.

```
gpt web -h
```

```
usage: AutoGPentTools.py web [-h] [-u [USERNAME]] [-p [PASSWORD]] [-H [HASH]] [-w WORDLIST [WORDLIST]]
[--scanner [SCANNER]] [--ssl] [--port [PORT]] [--file-path [FILE_PATH]] [--cms [CMS]] [target]
```

positional arguments:

target (Target Required *) The target IP(s), range(s),
CIDR(s), hostname(s), FQDN(s), file(s) containing a list of targets

optional arguments:

-h, --help show this help message and exit
 -u [USERNAME] username(s) or file(s) containing usernames
 -p [PASSWORD] password(s) or file(s) containing passwords
 -H [HASH] Pass The hash
 -w WORDLIST [WORDLIST ...] set wordlist (Default SecList wordlist)
 --scanner [SCANNER] Select scanner (default : full_scanner)
 --ssl usage of SSL/TLS requests
 --port [PORT] scan specific port (Default 80)
 --file-path [FILE_PATH] Specify to find the requested resource and start the enumeration with that route (Default /)
 --cms [CMS] What do you have? [Wordpress, Joomla, Drupal] (Default: None)

El servei LDAP, ens permet obtenir molta informació de la víctima. Aquest protocol es molt utilitzat per Controladors de Domini.

```
gpt ldap -h
```

```
usage: AutoGPentTools.py ldap [-h] [-u [USERNAME]] [-p [PASSWORD]]
                             [-H [HASH]] [--tags [TAGS]] [--port [PORT]] [target]
```

positional arguments:

target (Target Required *) The target IP(s), range(s),
CIDR(s), hostname(s), FQDN(s), file(s) containing a list of targets

optional arguments:

-h, --help show this help message and exit
 -u [USERNAME] username(s) or file(s) containing usernames
 -p [PASSWORD] password(s) or file(s) containing passwords
 -H [HASH] Pass The hash
 --tags [TAGS] What do you have? [Creds, NoCreds, Hash, Shell] (Default: NoCreds)
 --port [PORT] scan specific port (Default 389)

Finalment trobem el protocol samba, aquest es el que s'utilitza per les carpetes compartides. El funcionament es similar els anteriors. Pases un tag, en cas que no tinguis informació previa ho deixes per defecte i extraurà tota la informació possible d'aquest protocol.

```
gpt smb -h
```

```
usage: AutoGPentTools.py smb [-h] [-u [USERNAME]] [-p [PASSWORD]]
                             [-H [HASH]] [--tags [TAGS]] [--port [PORT]] [target]
```

positional arguments:

target (Target Required *) The target IP(s), range(s),
CIDR(s), hostname(s), FQDN(s), file(s) containing a list of targets

optional arguments:

-h, --help show this help message and exit
-u [USERNAME] username(s) or file(s) containing usernames
-p [PASSWORD] password(s) or file(s) containing passwords
-H [HASH] Pass The hash
--tags [TAGS] What do you have? [Creds, NoCreds, Hash, Shell] (Default: NoCreds)
--port [PORT] scan specific port (Default 445)

5.4. Exemples d'ús

Com he especificat anteriorment primer haurem d'executar l'opció de recon. Un exemple seria el següent:

```
[internetghost2] as jack in ~
gpt --verbose recon 192.168.1.12
[+] Managment: Directory /tmp/gpt_report/192.168.1.12/recon already exist
[] recon: nmap -sV -sC --open --version-all -oN
    /tmp/gpt_report/192.168.1.12/recon/nmap_quick_versions.txt 192.168.1.12
-- --
[+] Quick open ports with nmap: Succesfully!
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-30 08:28 UTC
Nmap scan report for internetghost2.home (192.168.1.12)
Host is up (0.000021s latency).
Not shown: 95 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds

[0] Quick scan with nmap: Running
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-30 08:28 UTC
Nmap scan report for internetghost2.home (192.168.1.12)
Host is up (0.000012s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Pure-FTPd
80/tcp    open  http         Apache/2.4.18 (Ubuntu)
```



```
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 302 Found
|     Location: https://http:443:///nice%20ports%2C/Tri%6Eity.txt%2ebak
|     Date: Sun, 30 May 2021 08:28:58 GMT
|     Content-Length: 5
|     Content-Type: text/plain; charset=utf-8
|     Found
|   GenericLines, Hello, Help, Kerberos, RTSPRequest, SSLSessionReq,
|   SSLv23SessionReq, TLSSessionReq, TerminalServerCookie:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
|     Request
|   GetRequest, HTTPOptions:
|     HTTP/1.0 302 Found
|     Location: https://http:443:///
|     Date: Sun, 30 May 2021 08:28:53 GMT
|     Content-Length: 5
|     Content-Type: text/plain; charset=utf-8
|_   Found
|_http-title: Did not follow redirect to https://internetghost2.home:443/
139/tcp   open  netbios-ssn Samba smbd 4.6.2
443/tcp   open  ssl/http   Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
| ssl-cert: Subject: commonName=TRAEFIK DEFAULT CERT
| Subject Alternative Name:
DNS:5fb207ee8c5e95258740a5c37d0c43c8.de0ab1663e5dab751b09fa03961c958a.traefik.default
| Not valid before: 2021-05-30T08:07:59
|_Not valid after:  2022-05-30T08:07:59
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|   h2
|   http/1.1
|_  acme-tls/1
| tls-nextprotoneg:
|   h2
|   http/1.1
|_  acme-tls/1
445/tcp   open  netbios-ssn Samba smbd 4.6.2
8090/tcp   open  http      PHP cli server 5.5 or later (PHP 7.4.9)
|_http-title: Login - Adminer
|_http-trane-info: Problem with XML parsing of /evox/about
30000/tcp open  tcpwrapped
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
```

```
SF-Port80-TCP:V=7.91%I=9%D=5/30%Time=60B34CC5%P=x86_64-pc-linux-gnu%r(GetR
SF:equest,9C,"HTTP/1\0x20302\x20Found\r\nLocation:\x20https://http:443:/
SF://\r\nDate:\x20Sun,\x2030\x20May\x202021\x2008:28:53\x20GMT\r\nContent-
SF:Length:\x205\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\n\r\nFo
SF:und")%r(HTTPOptions,9C,"HTTP/1\0x20302\x20Found\r\nLocation:\x20https
SF://http:443:///r\nDate:\x20Sun,\x2030\x20May\x202021\x2008:28:53\x20GM
SF:T\r\nContent-Length:\x205\r\nContent-Type:\x20text/plain;\x20charset=ut
SF:f-8\r\n\r\nFound")%r(RTSPRequest,67,"HTTP/1\1x20400\x20Bad\x20Request
SF:\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20clo
SF:se\r\n\r\n400\x20Bad\x20Request")%r(FourOhFourRequest,BF,"HTTP/1\0x20
SF:302\x20Found\r\nLocation:\x20https://http:443:///nice%20ports%2C/Tri%6E
SF:ity\0.txt%2ebak\r\nDate:\x20Sun,\x2030\x20May\x202021\x2008:28:58\x20GMT
SF:\r\nContent-Length:\x205\r\nContent-Type:\x20text/plain;\x20charset=utf
SF:-8\r\n\r\nFound")%r(GenericLines,67,"HTTP/1\1x20400\x20Bad\x20Request
SF:\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20clo
SF:se\r\n\r\n400\x20Bad\x20Request")%r(Hello,67,"HTTP/1\1x20400\x20Bad\x
SF:20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnectio
SF:n:\x20close\r\n\r\n400\x20Bad\x20Request")%r(Help,67,"HTTP/1\1x20400\
SF:x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nC
SF:onnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(SSLSessionReq,67,"
SF:HTTP/1\1x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20c
SF:harset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(T
SF:erminalServerCookie,67,"HTTP/1\1x20400\x20Bad\x20Request\r\nContent-T
SF:ype:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400
SF:\x20Bad\x20Request")%r(TLSSessionReq,67,"HTTP/1\1x20400\x20Bad\x20Req
SF:uest\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x2
SF:0close\r\n\r\n400\x20Bad\x20Request")%r(SSLv23SessionReq,67,"HTTP/1\1\
SF:x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf
SF:-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(Kerberos,67
SF:,"HTTP/1\1x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x2
SF:0charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request");
```

```
Host script results:
| smb2-security-mode:
| 2.10:
|_ Message signing enabled but not required
|_smb2-time: Protocol negotiation failed (SMB2)
```

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .
Nmap done: 1 IP address (1 host up) scanned in 266.04 seconds

```
[.] Scanning all ports with nmap and return service information: Running
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-30 08:33 UTC
Nmap scan report for 192.168.1.12
Host is up (0.00013s latency).
Not shown: 65517 closed ports
```

```

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Pure-FTPd
80/tcp    open  http
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 302 Found
|     Location: https://http:443:///nice%20ports%2C/Tri%6Eity.txt%2ebak
|     Date: Sun, 30 May 2021 08:33:25 GMT
|     Content-Length: 5
|     Content-Type: text/plain; charset=utf-8
|     Found
|   GenericLines, Help, Kerberos, LDAPSearchReq, LPDString, RTSPRequest,
|   SSLSessionReq, TLSSessionReq, TerminalServerCookie:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
|     Request
|   GetRequest, HTTPOptions:
|     HTTP/1.0 302 Found
|     Location: https://http:443:///
|     Date: Sun, 30 May 2021 08:33:20 GMT
|     Content-Length: 5
|     Content-Type: text/plain; charset=utf-8
|_   Found
|_http-title: Did not follow redirect to https://192.168.1.12:443/
139/tcp   open  netbios-ssn Samba smbd 4.6.2
443/tcp   open  ssl/http     Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
| ssl-cert: Subject: commonName=TRAEFIK DEFAULT CERT
| Subject Alternative Name:
DNS:5fb207ee8c5e95258740a5c37d0c43c8.de0ab1663e5dab751b09fa03961c958a.traefik.default
| Not valid before: 2021-05-30T08:07:59
|_Not valid after: 2022-05-30T08:07:59
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|   h2
|   http/1.1
|_ acme-tls/1
| tls-nextprotoneg:
|   h2
|   http/1.1
|_ acme-tls/1
445/tcp   open  netbios-ssn Samba smbd 4.6.2
8090/tcp  open  http         PHP cli server 5.5 or later (PHP 7.4.9)
|_http-title: Login - Adminer
|_http-trane-info: Problem with XML parsing of /evox/about

```

```

30000/tcp open  tcpwrapped
30001/tcp open  tcpwrapped
30002/tcp open  tcpwrapped
30003/tcp open  tcpwrapped
30004/tcp open  tcpwrapped
30005/tcp open  tcpwrapped
30006/tcp open  tcpwrapped
30007/tcp open  tcpwrapped
30008/tcp open  tcpwrapped
30009/tcp open  tcpwrapped
43677/tcp open  unknown
| fingerprint-strings:
|   NULL, RPCCheck:
|_  {"type": "Tier1", "version": "1.0"}
57621/tcp open  unknown
2 services unrecognized despite returning data. If you know the service/version,
please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port80-TCP:V=7.91%I=7%D=5/30%Time=60B34DD0%P=x86_64-pc-linux-gnu%r(GetR
SF:equest,9C,"HTTP/1\0\20302\20Found\r\nLocation:\20https://http:443:/
SF://\r\nDate:\20Sun,\2030\20May\202021\2008:33:20\20GMT\r\nContent-
SF:Length:\205\r\nContent-Type:\20text/plain;\20charset=utf-8\r\n\r\nFo
SF:und")%r(HTTPOptions,9C,"HTTP/1\0\20302\20Found\r\nLocation:\20https
SF://http:443:///r\nDate:\20Sun,\2030\20May\202021\2008:33:20\20GM
SF:T\r\nContent-Length:\205\r\nContent-Type:\20text/plain;\20charset=ut
SF:f-8\r\n\r\nFound")%r(RTSPRequest,67,"HTTP/1\1\20400\20Bad\20Request
SF:\r\nContent-Type:\20text/plain;\20charset=utf-8\r\nConnection:\20clo
SF:se\r\n\r\n400\20Bad\20Request")%r(FourOhFourRequest,BF,"HTTP/1\0\20
SF:302\20Found\r\nLocation:\20https://http:443:///nice%20ports%2C/Tri%6E
SF:ity\20.txt%2ebak\r\nDate:\20Sun,\2030\20May\202021\2008:33:25\20GMT
SF:\r\nContent-Length:\205\r\nContent-Type:\20text/plain;\20charset=utf
SF:-8\r\n\r\nFound")%r(GenericLines,67,"HTTP/1\1\20400\20Bad\20Request
SF:\r\nContent-Type:\20text/plain;\20charset=utf-8\r\nConnection:\20clo
SF:se\r\n\r\n400\20Bad\20Request")%r(Help,67,"HTTP/1\1\20400\20Bad\2
SF:0Request\r\nContent-Type:\20text/plain;\20charset=utf-8\r\nConnection
SF::\20close\r\n\r\n400\20Bad\20Request")%r(SSLSessionReq,67,"HTTP/1\1
SF:\20400\20Bad\20Request\r\nContent-Type:\20text/plain;\20charset=ut
SF:f-8\r\nConnection:\20close\r\n\r\n400\20Bad\20Request")%r(TerminalSe
SF:rverCookie,67,"HTTP/1\1\20400\20Bad\20Request\r\nContent-Type:\20t
SF:ext/plain;\20charset=utf-8\r\nConnection:\20close\r\n\r\n400\20Bad\2
SF:20Request")%r(TLSSessionReq,67,"HTTP/1\1\20400\20Bad\20Request\r\nC
SF:ontent-Type:\20text/plain;\20charset=utf-8\r\nConnection:\20close\r\
SF:n\r\n400\20Bad\20Request")%r(Kerberos,67,"HTTP/1\1\20400\20Bad\20
SF:Request\r\nContent-Type:\20text/plain;\20charset=utf-8\r\nConnection:
SF:\20close\r\n\r\n400\20Bad\20Request")%r(LPDString,67,"HTTP/1\1\204
SF:00\20Bad\20Request\r\nContent-Type:\20text/plain;\20charset=utf-8\r

```

```
SF:\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(LDAPSearchReq,6
SF:7,"HTTP/1\1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x
SF:20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port43677-TCP:V=7.91%I=7%D=5/30%Time=60B34DCA%P=x86_64-pc-linux-gnu%r(N
SF:ULL,22,"{\\"type\\":\\"Tier1\\",\\"version\\":\\"1\1.0\\"}\r\n")%r(RPCCheck,22,"
SF:{\\"type\\":\\"Tier1\\",\\"version\\":\\"1\1.0\\"}\r\n");
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
```

```
Host script results:
| smb2-security-mode:
| 2.10:
|_ Message signing enabled but not required
|_smb2-time: Protocol negotiation failed (SMB2)
```

OS and Service detection performed. Please report any incorrect results at <https://nmap.org/submit/>
Nmap done: 1 IP address (1 host up) scanned in 99.70 seconds

```
[+] Scanning udp ports: Succesfully!
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-30 08:34 UTC
Nmap scan report for internetghost2.home (192.168.1.12)
Host is up (0.00014s latency).
Not shown: 81 open|filtered ports
PORT      STATE SERVICE      VERSION
17/udp    closed qotd
69/udp    closed tftp
88/udp    closed kerberos-sec
139/udp   closed netbios-ssn
500/udp   closed isakmp
515/udp   closed printer
593/udp   closed http-rpc-epmap
1025/udp  closed blackjack
1029/udp  closed solid-mux
1433/udp  closed ms-sql-s
1813/udp  closed radacct
5000/udp  closed upnp
31337/udp closed BackOrifice
32769/udp closed filenet-rpc
32815/udp closed unknown
49154/udp closed unknown
49186/udp closed unknown
49193/udp closed unknown
```

49200/udp closed unknown

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .
Nmap done: 1 IP address (1 host up) scanned in 277.81 seconds

```
gpt --verbose smb 192.168.1.12 -h
usage: AutoGPentools.py smb [-h] [-u [USERNAME]] [-p [PASSWORD]] [-H [HASH]] [--tags [TAGS]]
                        [--port [PORT]]
                        [target]
```

positional arguments:

target (Target Required *) The target IP(s), range(s), CIDR(s), hostname(s), FQDN(s), file(s) containing a list of targets

optional arguments:

-h, --help show this help message and exit
-u [USERNAME] username(s) or file(s) containing usernames
-p [PASSWORD] password(s) or file(s) containing passwords
-H [HASH] Pass The hash
--tags [TAGS] What do you have? [Creds, NoCreds, Hash, Shell] (Default: NoCreds)
--port [PORT] scan specific port (Default 445)

```
[+] Managment: Directory /tmp/gpt_report/192.168.1.12/smb already exist
[/] smb: enum4linux -a 192.168.1.12 |
tee /tmp/gpt_report/192.168.1.12/smb/enum4linux_null_session.txt
[+] Managment: Directory /tmp/gpt_report/192.168.1.12/smb already exist
[/] smb: enum4linux -a 192.168.1.12 |
tee /tmp/gpt_report/192.168.1.12/smb/enum4linux_null_session.txt
-- --
[+] Samba null sessions enum directories: Succesfully!
```

Sharename	Type	Comment
-----	----	-----
marc	Disk	Carpeta de l'Alumne marc
michael	Disk	Carpeta de l'Alumne michael
ernest	Disk	Carpeta de l'Alumne ernest
cosmin	Disk	Carpeta de l'Alumne cosmin
isaac	Disk	Carpeta de l'Alumne isaac
jordi	Disk	Carpeta de l'Alumne jordi
fernando	Disk	Carpeta de l'Alumne fernando
victor	Disk	Carpeta de l'Alumne victor
lina	Disk	Carpeta de l'Alumne lina
[\\] jack	Disk	Carpeta de l'Alumne jack
ju	Disk	Carpeta de l'Alumne ju
andre	Disk	Carpeta de l'Alumne andre
anton	Disk	Carpeta de l'Alumne anton
juan	Disk	Carpeta de l'Alumne juan

```

pe          Disk      Carpeta de l'Alumne pe
IPC$        IPC       IPC Service (Samba Server)
SMB1 disabled -- no workgroup available

[-] Enum null sessions with Crackmapexec: Something wrong
[*] First time use detected
[*] Creating home directory structure
[*] Creating default workspace
[*] Initializing SSH protocol database
[*] Initializing WINRM protocol database
[*] Initializing MSSQL protocol database
[*] Initializing SMB protocol database
[*] Initializing LDAP protocol database
[*] Copying default configuration file
[*] Generating SSL certificate

[!] Enum null sessions with enum4linux: Running

SMB1 disabled -- no workgroup available

[-] Enum null sessions with Crackmapexec: Something wrong
[*] First time use detected
[*] Creating home directory structure
[*] Creating default workspace
[*] Initializing SSH protocol database
[*] Initializing WINRM protocol database
[*] Initializing MSSQL protocol database
[*] Initializing SMB protocol database
[*] Initializing LDAP protocol database
[*] Copying default configuration file
[-] Generating SSL certificate
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ )
on Sun May 30 08:28:05 2021

=====
|   Target Information   |
=====
Target ..... 192.168.1.12
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

=====
|   Enumerating Workgroup/Domain on 192.168.1.12   |

```

```

=====
[E] Can't find workgroup/domain

=====
|   Nbtstat Information for 192.168.1.12   |
=====
Looking up status of 192.168.1.12
No reply from 192.168.1.12

=====
|   Session Check on 192.168.1.12   |
=====
[+] Server 192.168.1.12 allows sessions using username '', password ''
[+] Got domain/workgroup name:

=====
|   Getting domain SID for 192.168.1.12   |
=====
Domain Name: MYGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

=====
|   OS information on 192.168.1.12   |
=====
[+] Got OS info for 192.168.1.12 from smbclient:
[+] Got OS info for 192.168.1.12 from srvinfo:
    1C144B1916EA   Wk Sv PrQ Unx NT SNT Samba Server
platform_id      :    500
os version       :    6.1
server type      :    0x809a03

=====
|   Users on 192.168.1.12   |
=====
index: 0x1 RID: 0x3f5 acb: 0x00000010 Account: juan   Name: Linux User   Desc:
index: 0x2 RID: 0x3f6 acb: 0x00000010 Account: pe    Name: Linux User   Desc:
index: 0x3 RID: 0x3e8 acb: 0x00000010 Account: marc   Name: Linux User   Desc:
index: 0x4 RID: 0x3ea acb: 0x00000010 Account: ernest  Name: Linux User   Desc:
index: 0x5 RID: 0x3ec acb: 0x00000010 Account: isaac   Name: Linux User   Desc:
index: 0x6 RID: 0x3e9 acb: 0x00000010 Account: michael Name: Linux User   Desc:
index: 0x7 RID: 0x3eb acb: 0x00000010 Account: cosmin  Name: Linux User   Desc:
index: 0x8 RID: 0x3ed acb: 0x00000010 Account: jordi   Name: Linux User   Desc:
index: 0x9 RID: 0x3ee acb: 0x00000010 Account: fernando Name: Linux User   Desc:
index: 0xa RID: 0x3ef acb: 0x00000010 Account: victor  Name: Linux User   Desc:

```



```
index: 0xb RID: 0x3f0 acb: 0x00000010 Account: lina      Name: Linux User      Desc:
index: 0xc RID: 0x3f1 acb: 0x00000010 Account: jack      Name: Linux User      Desc:
index: 0xd RID: 0x3f2 acb: 0x00000010 Account: ju       Name: Linux User      Desc:
index: 0xe RID: 0x3f3 acb: 0x00000010 Account: andre     Name: Linux User      Desc:
index: 0xf RID: 0x3f4 acb: 0x00000010 Account: anton     Name: Linux User      Desc:
```

```
user:[juan] rid:[0x3f5]
user:[pe] rid:[0x3f6]
user:[marc] rid:[0x3e8]
user:[ernest] rid:[0x3ea]
user:[isaac] rid:[0x3ec]
user:[michael] rid:[0x3e9]
user:[cosmin] rid:[0x3eb]
user:[jordi] rid:[0x3ed]
user:[fernando] rid:[0x3ee]
user:[victor] rid:[0x3ef]
user:[lina] rid:[0x3f0]
user:[jack] rid:[0x3f1]
user:[ju] rid:[0x3f2]
user:[andre] rid:[0x3f3]
user:[anton] rid:[0x3f4]
```

```
=====
|   Share Enumeration on 192.168.1.12   |
=====
```

Sharename	Type	Comment
marc	Disk	Carpeta de l'Alumne marc
michael	Disk	Carpeta de l'Alumne michael
ernest	Disk	Carpeta de l'Alumne ernest
cosmin	Disk	Carpeta de l'Alumne cosmin
isaac	Disk	Carpeta de l'Alumne isaac
jordi	Disk	Carpeta de l'Alumne jordi
fernando	Disk	Carpeta de l'Alumne fernando
victor	Disk	Carpeta de l'Alumne victor
lina	Disk	Carpeta de l'Alumne lina
jack	Disk	Carpeta de l'Alumne jack
ju	Disk	Carpeta de l'Alumne ju
andre	Disk	Carpeta de l'Alumne andre
anton	Disk	Carpeta de l'Alumne anton
juan	Disk	Carpeta de l'Alumne juan
pe	Disk	Carpeta de l'Alumne pe
IPC\$	IPC	IPC Service (Samba Server)

SMB1 disabled -- no workgroup available

```
[+] Attempting to map shares on 192.168.1.12
//192.168.1.12/marc Mapping: DENIED, Listing: N/A
//192.168.1.12/michael Mapping: DENIED, Listing: N/A
//192.168.1.12/ernest Mapping: DENIED, Listing: N/A
//192.168.1.12/cosmin Mapping: DENIED, Listing: N/A
//192.168.1.12/isaac Mapping: DENIED, Listing: N/A
//192.168.1.12/jordi Mapping: DENIED, Listing: N/A
//192.168.1.12/fernando Mapping: DENIED, Listing: N/A
//192.168.1.12/victor Mapping: DENIED, Listing: N/A
//192.168.1.12/lina Mapping: DENIED, Listing: N/A
//192.168.1.12/jack Mapping: DENIED, Listing: N/A
//192.168.1.12/ju Mapping: DENIED, Listing: N/A
//192.168.1.12/andre Mapping: DENIED, Listing: N/A
//192.168.1.12/anton Mapping: DENIED, Listing: N/A
//192.168.1.12/juan Mapping: DENIED, Listing: N/A
//192.168.1.12/pe Mapping: DENIED, Listing: N/A
//192.168.1.12/IPC$ [E] Can't understand response:
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
```

```
=====
| Password Policy Information for 192.168.1.12 |
=====
```

```
[+] Attaching to 192.168.1.12 using a NULL share
```

```
[+] Trying protocol 139/SMB...
```

```
[!] Protocol failed: ('unpack requires a buffer of 1 bytes',
"When unpacking field 'SecurityMode | <B | b'[:1]'" )
```

```
[+] Trying protocol 445/SMB...
```

```
[+] Found domain(s):
```

```
[+] 1C144B1916EA
[+] Builtin
```

```
[+] Password Info for Domain: 1C144B1916EA
```

```
[+] Minimum password length: 5
[+] Password history length: None
[+] Maximum password age: 37 days 6 hours 21 minutes
[+] Password Complexity Flags: 000000
```

```
[+] Domain Refuse Password Change: 0
```

```
[+] Domain Password Store Cleartext: 0
[+] Domain Password Lockout Admins: 0
[+] Domain Password No Clear Change: 0
[+] Domain Password No Anon Change: 0
[+] Domain Password Complex: 0
```

```
[+] Minimum password age: None
[+] Reset Account Lockout Counter: 30 minutes
[+] Locked Account Duration: 30 minutes
[+] Account Lockout Threshold: None
[+] Forced Log off Time: 37 days 6 hours 21 minutes
```

```
[+] Retrieved partial password policy with rpcclient:
```

```
Password Complexity: Disabled
```

```
Minimum Password Length: 5
```

```
=====
|   Groups on 192.168.1.12   |
=====
```

```
[+] Getting builtin groups:
```

```
[+] Getting builtin group memberships:
```

```
[+] Getting local groups:
```

```
[+] Getting local group memberships:
```

```
[+] Getting domain groups:
```

```
[+] Getting domain group memberships:
```

```
=====
|   Users on 192.168.1.12 via RID cycling (RIDS: 500-550,1000-1050)   |
=====
```

```
[I] Found new SID: S-1-22-1
```

```
[I] Found new SID: S-1-5-21-2112836514-3653563689-2825713174
```

```
[I] Found new SID: S-1-5-32
```

```
[+] Enumerating users using SID S-1-5-32 and logon username '', password ''
```

```
S-1-5-32-544 BUILTIN\Administrators (Local Group)
```

```
S-1-5-32-545 BUILTIN\Users (Local Group)
```

```
S-1-5-32-546 BUILTIN\Guests (Local Group)
```

```
S-1-5-32-547 BUILTIN\Power Users (Local Group)
```

```

S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)
[+] Enumerating users using SID S-1-5-21-2112836514-3653563689-2825713174
and logon username '', password ''
S-1-5-21-2112836514-3653563689-2825713174-500 *unknown*\*unknown* (8)
S-1-5-21-2112836514-3653563689-2825713174-501 1C144B1916EA\nobody (Local User)
S-1-5-21-2112836514-3653563689-2825713174-513 1C144B1916EA\None (Domain Group)
S-1-5-21-2112836514-3653563689-2825713174-1000 1C144B1916EA\marc (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1001 1C144B1916EA\michael (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1002 1C144B1916EA\ernest (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1003 1C144B1916EA\cosmin (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1004 1C144B1916EA\isaac (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1005 1C144B1916EA\jordi (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1006 1C144B1916EA\fernando (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1007 1C144B1916EA\victor (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1008 1C144B1916EA\lina (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1009 1C144B1916EA\jack (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1010 1C144B1916EA\ju (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1011 1C144B1916EA\andre (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1012 1C144B1916EA\anton (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1013 1C144B1916EA\juan (Local User)
S-1-5-21-2112836514-3653563689-2825713174-1014 1C144B1916EA\pe (Local User)
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\marc (Local User)
S-1-22-1-1001 Unix User\michael (Local User)
S-1-22-1-1002 Unix User\ernest (Local User)
S-1-22-1-1003 Unix User\cosmin (Local User)
S-1-22-1-1004 Unix User\isaac (Local User)
S-1-22-1-1005 Unix User\jordi (Local User)
S-1-22-1-1006 Unix User\fernando (Local User)
S-1-22-1-1007 Unix User\victor (Local User)
S-1-22-1-1008 Unix User\lina (Local User)
S-1-22-1-1009 Unix User\jack (Local User)
S-1-22-1-1010 Unix User\ju (Local User)
S-1-22-1-1011 Unix User\andre (Local User)
S-1-22-1-1012 Unix User\anton (Local User)
S-1-22-1-1013 Unix User\juan (Local User)
S-1-22-1-1014 Unix User\pe (Local User)

=====
|   Getting printer info for 192.168.1.12   |
=====
Could not initialise spoolss. Error was NT_STATUS_OBJECT_NAME_NOT_FOUND

```

enum4linux complete on Sun May 30 08:28:39 2021

A nivell de web ve a mostra una mica el mateix, l'ús de l'eina no es molt complexa i extreu molta informació encara que sigui utilitzant enumeració bàsica.

```
[~] Enum index web content: Something wrong
HTTP/1.1 302 Found
Location: https://192.168.1.12:443/
Date: Sun, 30 May 2021 09:31:26 GMT
Content-Length: 5
Content-Type: text/plain; charset=utf-8
```

Found

```
[b] Enum versions webs: Running
WhatWeb report for http://192.168.1.12
Status      : 302 Found
Title       : <None>
IP          : 192.168.1.12
Country     : RESERVED, ZZ
```

Summary : RedirectLocation[https://192.168.1.12:443/]

Detected Plugins:

```
[ RedirectLocation ]
    HTTP Server string location. used with http-status 301 and
    302

    String      : https://192.168.1.12:443/ (from location)
```

HTTP Headers:

```
HTTP/1.1 302 Found
Location: https://192.168.1.12:443/
Date: Sun, 30 May 2021 09:31:27 GMT
Content-Length: 5
Content-Type: text/plain; charset=utf-8
Connection: close
```

WhatWeb report for https://192.168.1.12/

```
Status      : 404 Not Found
Title       : <None>
IP          : 192.168.1.12
Country     : RESERVED, ZZ
```

Summary : UncommonHeaders[x-content-type-options]

Detected Plugins:

```
[ UncommonHeaders ]
```

Uncommon HTTP server headers. The blacklist includes all the standard headers and many non standard but common ones. Interesting but fairly common headers should have their own plugins, eg. x-powered-by, server and x-aspnet-version. Info about headers can be found at www.http-stats.com

String : x-content-type-options (from headers)

HTTP Headers:

```
HTTP/1.1 404 Not Found
Content-Type: text/plain; charset=utf-8
X-Content-Type-Options: nosniff
Date: Sun, 30 May 2021 09:31:28 GMT
Content-Length: 19
Connection: close
```

[+] Nikto detect misconfiguration, risky files, etc.: Successfully!

- Nikto v2.1.6

```
-----
+ Target IP:          192.168.1.12
+ Target Hostname:    192.168.1.12
+ Target Port:        80
+ Start Time:         2021-05-30 09:31:30 (GMT0)
-----
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content
+ Root page / redirects to: https://192.168.1.12:443/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ 7915 requests: 0 error(s) and 3 item(s) reported on remote host
+ End Time:           2021-05-30 09:31:42 (GMT0) (12 seconds)
-----
+ 1 host(s) tested
```

[-] Fuzzing directories and files: Command timeout

[-] SIGINT or CTRL-C detected.: Exiting gracefully

6. Conclusions

El principal objectiu d'aquest treball era tenir una eina que em permetés automatitzar l'enumeració d'un pentest, també ha set aprendre i entendre com desenvolupar una eina. Considerant que havia posat el llisto molt alt però no comptava amb el poc temps que he tingut. De totes maneres em sento molt content d'haver escollit aquest projecte, ja que m'ha ajudat molt a millorar aspectes tècnics de la programació i entendre el funcionament d'altres eines.

Per això crec que és un bon inici per seguir desenvolupant aquestes habilitats, considero que crear-te les teves eines per automatitzar tasques et pot aportar molta més eficàcia i t'obliga a entendre molt més a baix nivell la tasca que s'ha de realitzar.

Se que encara li falta molt perquè l'eina deixi de ser una versió beta però realment he aconseguit donar-li forma i a poc a poc podré anar implementant les millores que m'he proposat.

6.1. Línies de futur

Com he dit, jo vull seguir millorant les habilitats adquirides en aquest projecte, per tant, seguiré treballant per teure'l de la fase beta i deixar-lo d'utilitzar en entorns de laboratori.

6.2. Problemes trobats

Per una banda, ha set bastant complicat el tema d'utilitzar alguns mòduls perquè no havia utilitzat mai, com ara el `argparse` i subproces. Aquest dos son els que més m'ha costat implementar. Finalment he aconseguit fer que funcionin però en algunes comandes no ho agafa del tot bé i no executa la comanda.

Un altre problema que vaig trobar que no hi contava era el tema de modificar els fitxers de configuració pensava que seria més fàcil i únicament hauria de buscar una llibreria que m'ho permetis però finalment ho vaig haver de preguntar quines alternatives podria utilitzar.

Referencias

- [1] CrackMapExec is developed by @byt3bl33d3r and maintained by @mpgn *Is a post-exploitation tool that helps automate assessing the security of large Active Directory networks* <https://mpgn.gitbook.io/crackmapexec/>
- [2] Tib3rius *Penetration tester with 10 years experience, specializing in web application security. I have an MSc in InfoSec, created AutoRecon and two popular PrivEsc courses for the OSCP labs/exam. He's stream on Twitch and create CyberSecurity* <https://github.com/Tib3rius/AutoRecon>
- [3] argparse Parser for command-line options, arguments and sub-commands <https://docs.python.org/3/library/argparse.html>
- [4] subprocess Subprocess management <https://docs.python.org/3/library/subprocess.html>
- [5] os.path Common pathname manipulations <https://docs.python.org/3/library/os.path.html>