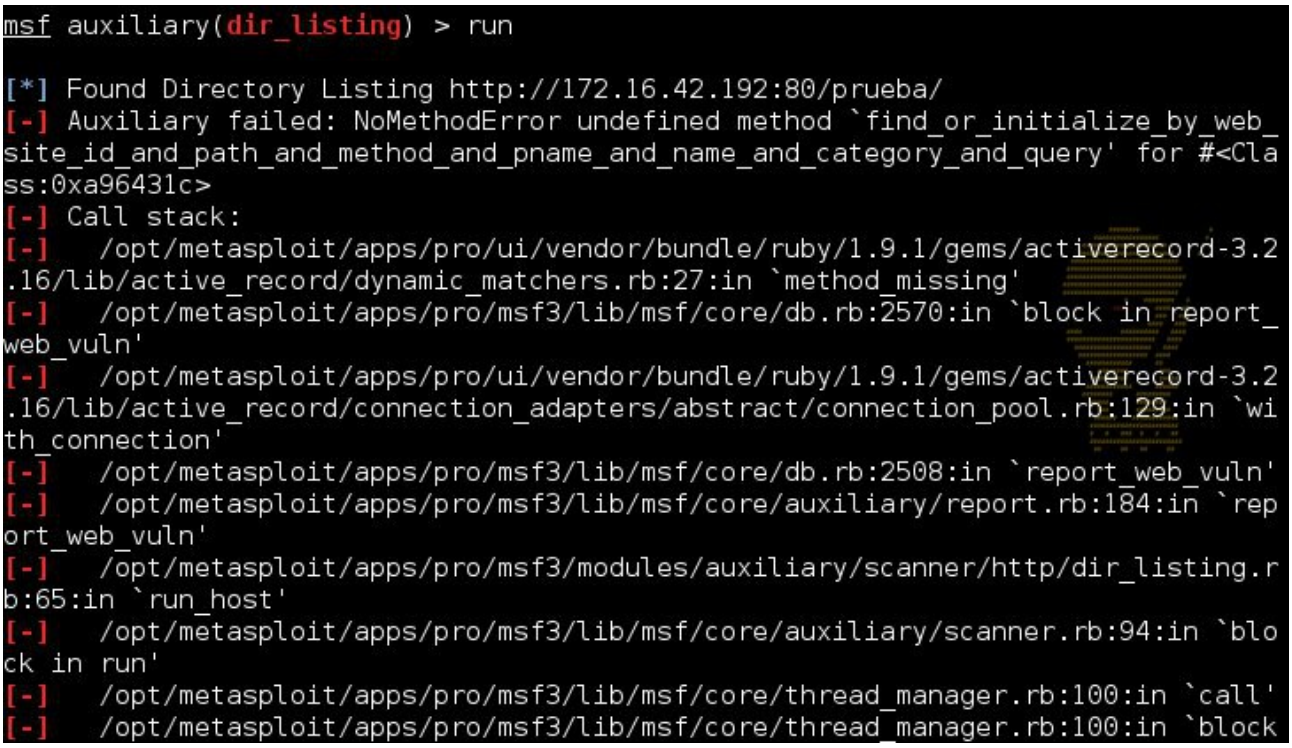
Empezando con Wmap

@luisco100

El objetivo de este tutorial es dar a conocer lo pasos necesarios para utilizar la herramienta de escaneo de vulnerabilidades en sitios web “Wmap”.

Primero que todo un reconocimiento a la gran labor de Efrain Torres, especialista de seguridad Colombiano, creador del plugin para el framework Metasploit.

Al empezar a utilizar el plugin Wmap me encontré con el problema “NoMethodErrro undefined method” el cual hace referencia al método “report_web_vuln”.



Investigando me encontré que este problema se debe a incompatibilidad del framework **Metasploit Community** y la base de datos que se instala por defecto (Pro). Esto se debe a modificaciones de la tabla Mdm::WebVuln que permite mejoras en la caracterización de temas relacionados con OWASP 2010 y 2013.

Como el problema es con la base de datos por defecto, lo primero que tenemos que hacer es crear una nueva base de datos donde vamos a almacenar la información de los escaneos con Wmap.

Para lo cual desde la consola ejecutamos el comando:

sudo su postgres

En este caso no me pide contraseña dado que estoy trabajando con la configuración por defecto que trae la distribución Kali linux.

Creamos el usuario y digitamos la contraseña que se le asignara a este nuevo usuario

createuser wmap -P

Por ultimo creamos la base de datos y le asignamos como propietario el usuario creado anteriormente.

createdb –owner=wmap wmap

```
root@kali:~# sudo su postgres
postgres@kali:/root$ createuser wmap -P
Ingrese la contraseña para el nuevo rol:
Ingrésela nuevamente:
¿Será el nuevo rol un superusuario? (s/n) s
postgres@kali:/root$ createdb --owner=wmap wmap
postgres@kali:/root$ quit
bash: quit: no se encontró la orden
postgres@kali:/root$ exit
exit
```

Una vez creada la base de datos ejecutamos la consola de Metasploit.

Cuando se inicia Metasploit por defecto se conecta a la base de datos **msf3**, pero en este caso necesitamos que se conecte a la nueva base de datos creada y se creen las tablas.

Para esto ejecutamos el comando

db_disconnect

y luego el comando

db_connect wmap:pass@127.0.0.1/wmap

```
msf > db_connect wmap:wmap123@127.0.0.1/wmap
NOTICE:  CREATE TABLE will create implicit sequence "hosts_id_seq" for serial column "hosts.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "hosts_pkey" for table "hosts"
NOTICE:  CREATE TABLE will create implicit sequence "clients_id_seq" for serial column "clients.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "clients_pkey" for table "clients"
NOTICE:  CREATE TABLE will create implicit sequence "services_id_seq" for serial column "services.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "services_pkey" for table "services"
NOTICE:  CREATE TABLE will create implicit sequence "vulns_id_seq" for serial column "vulns.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "vulns_pkey" for table "vulns"
NOTICE:  CREATE TABLE will create implicit sequence "refs_id_seq" for serial column "refs.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "refs_pkey" for table "refs"
NOTICE:  CREATE TABLE will create implicit sequence "notes_id_seq" for serial column "notes.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "notes_pkey" for table "notes"
NOTICE:  CREATE TABLE will create implicit sequence "wmap_targets_id_seq" for serial column "wmap_targets.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "wmap_targets_pkey" for table "wmap_targets"
NOTICE:  CREATE TABLE will create implicit sequence "wmap_requests_id_seq" for serial column "wmap_requests.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "wmap_requests_pkey" for table "wmap_requests"
```

Para utilizar Wmap es necesario contar con la url del sitio y sus respectivos directorios. Para esto podemos utilizar el modulo auxiliar Crawler de metasploit. El modulo Crawler hace un recorrido e indexación de las paginas del sitio.

Ademas del modulo auxiliar Crawler de Metasploit podemos importar la información de escaneos realizados con herramientas tales como:

- Acunetix
- Amap
- Appscan
- Burp

Comandos para realizar la importacion de otras herramientas :

msf > db_import
Usage: db_import <filename> [file2...]

Para este ejercicio vamos a utilizar el modulo Crawler con un máximo de una pagina, pues la prueba consiste en verificar la vulnerabilidad de listados de directorios y el sitio solo tiene una carpeta de prueba.

La idea de esta prueba es verificar que tenemos nuestro ambiente perfectamente configurado y listo para registrar las vulnerabilidades.

Cargamos el modulo y configuramos las variables de RHOST, URI, MAX_PAGES, como lo muestra la imagen a continuación.

```
msf > use auxiliary/scanner/http/crawler
msf auxiliary(crawler) > set RHOST 172.16.42.192
RHOST => 172.16.42.192
msf auxiliary(crawler) > set URI /prueba
URI => /prueba
msf auxiliary(crawler) > show options

Module options (auxiliary/scanner/http/crawler):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   DOMAIN       WORKSTATION      yes       The domain to use
   MAX_MINUTES  5                yes       The maximum number
   MAX_PAGES    500              yes       The maximum number
   MAX_THREADS  4                yes       The maximum number
   PASSWORD                      no        The HTTP password
   RHOST        172.16.42.192    yes       The target address
   RPORT        80               yes       The target port
   URI          /prueba          yes       The starting page
   USERNAME                      no        The HTTP username
   VHOST                         no        HTTP server virtua

msf auxiliary(crawler) > set MAX_PAGES 1
```

Por ultimo ejecutamos el modulo con el comando **run.**

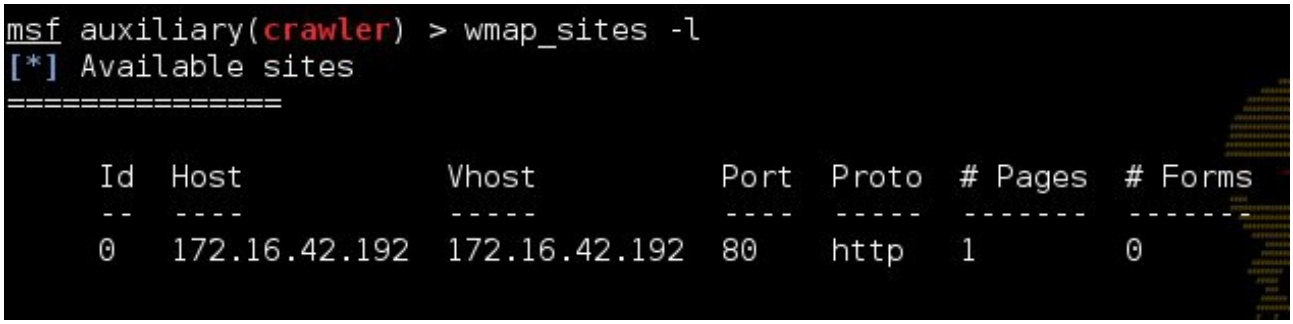
Una vez ejecutado el modulo auxiliar Crawler, cargamos el plugin “Wmap”.

load wmap

Verificamos que la información del escaneo con el auxiliar Crawler se encuentre registrada en la tabla de sitios de “Wmap”.

Comando

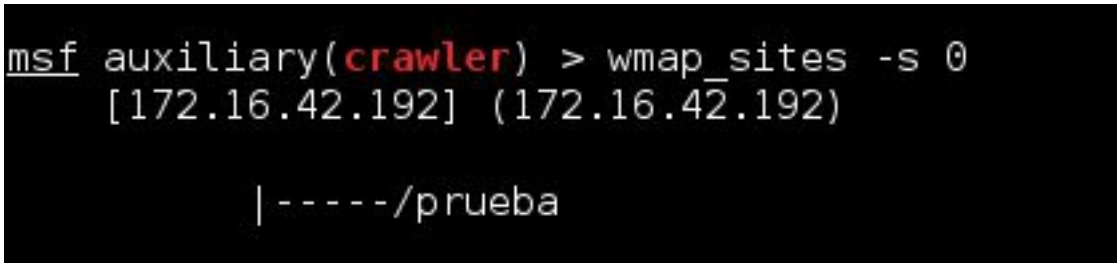
wmap_sites -l



La información se encuentra registrada, host, puerto, número de paginas y formularios.

Wmap nos da la posibilidad de listar los directorios y paginas en forma de arbol con el comando:

wmap_site -s 0



Una vez verificamos que tenemos el sitio correctamente indexado, lo siguiente es cargarlo como objetivo.

Para esto tenemos dos posibilidades, cargarlo de la lista de sitios o cargarlo directamente de forma manual.

Carga directa manual comando:

wmap_targets -t test.abcd.com,http://172.16.42.192/

Carga de obejtivo por la tabla de sitios:

wmap_targets -d 0

Donde el parámetro -d recibe el identificador del sitio, en este caso es 0.

Para listar los objetivos podemos utilizar el comando

wmap_targets -l

```
msf auxiliary(crawler) > wmap_targets -d 0
[*] Loading 172.16.42.192,http://172.16.42.192:80/.
msf auxiliary(crawler) > wmap_targets -l
[*] Defined targets
===============

     Id  Vhost          Host           Port  SSL    Path
     --  -----          ----           ----  ---    ----
     0   172.16.42.192  172.16.42.192  80    false      /
```

Ahora vamos a cargar los módulos que tiene Wmap, con el comando **wmap_modules -l**, cargamos y listamos los módulos.

```
msf auxiliary(crawler) > wmap_modules -l
[*] Loading wmap modules...
[*] 39 wmap enabled modules loaded.
[*] wmap_ssl
========

     Name                          OrderID
     ----                          -------
     auxiliary/scanner/http/cert   :last
     auxiliary/scanner/http/ssl    :last


[*] wmap_server
===========

     Name                                        OrderID
     ----                                        -------
     auxiliary/admin/http/tomcat_administration  :last
```

Por ultimo para ejecutar el escaneo, vamos a listar las opciones del comando wmap_run

wmap_run

```
msf auxiliary(crawler) > wmap_run -h
[*] Usage: wmap_run [options]
        -h                      Display this help text
        -t                      Show all enabled modules
        -m [regex]              Launch only modules that name match provided regex.
        -p [regex]              Only test path defined by regex.
        -e [/path/to/profile]   Launch profile modules against all matched targets.
                                (No profile file runs all enabled modules.)
```

Como podemos ver, se puede ejecutar el escanear con todos los módulos, con algunos modulos, algunas rutas o lanzar perfiles predefinidos.

Para este ejemplo vamos a utilizar la opción **-m** para ejecutar solo el modulo de **dir_listing**

```
msf auxiliary(crawler) > wmap_run -m dir_listing
[*] Using module dir_listing.
[-] NO WMAP NODES DEFINED. Executing local modules
[*] Testing target:
[*]     Site: 172.16.42.192 (172.16.42.192)
[*]     Port: 80 SSL: false
============================================================
[*] Testing started. 2014-04-09 21:07:53 -0500
[*]
=[ SSL testing ]=
============================================================
[*] Target is not SSL. SSL modules disabled.
[*]
=[ Web Server testing ]=
============================================================
[*]
=[ File/Dir testing ]=
============================================================
[*] Module auxiliary/scanner/http/dir_listing
[*] Path: /
[*] Path: /prueba

[*] Found Directory Listing http://172.16.42.192:80/prueba/
[*]
```

Como se puede observar el la imagen anterior, el escaneo termino satisfactoriamente y encontro lo vulnerabilidad en el directorio **/prueba**.
Se puede verificar las vulnerabilidades registradas con el comando:

wmap_vulns -l

```
msf auxiliary(crawler) > wmap_vulns -l
[*] + [172.16.42.192] (172.16.42.192): directory /prueba/
[*]     directory listing Directoy found allowing liting of its contents.
[*]     GET Res code: 200
```

Espero que este pequeño tutorial les sea de utilidad y de agrado.
@luisco100