Módulo 3.4

Redes y Seguridad
Gr. Ing. Informática
J.L. Vázquez Poletti

Objetivo

En este módulo manejaremos distintos aspectos de la seguridad en Windows: utilizaremos un troyano y un programa de gestión remota, emplearemos un pincho USB para desplegar el primero y alteraremos el funcionamiento ("crackearemos") de un sencillo programa.

IMPORTANTE: Es imprescindible traer **un pincho USB propio y vacío** para realizar esta práctica.

Punto de partida

Nuestro objetivo es una de las máquinas de Industrias Abstergo, una empresa dedicada al sector farmacéutico y el entretenimiento, así como al desarrollo de proyectos de alta tecnología aplicables a otros ámbitos. Dicha máquina almacena información vital y su sistema operativo es **Windows XP**.

Para acceder a la misma usaremos un troyano cuya instalación se realizará en una misión de infiltración. El troyano se llevará en un pincho USB que se conectará a la máquina objetivo.

Escenario

Las dos máquinas están conectadas a la misma red pero falta asignarles direcciones IP estáticas:

- **RyS-TheFarm** tendrá la IP 192.168.1.2.
- **RyS-Abstergo** tendrá la IP 192.168.1.3.

Para asignarles la IP hay que:

1. Acceder al **Panel de Control** y de ahí, a las **Conexiones de Red**.
2. Hacer doble click en la conexión existente y después en las **Propiedades**.
3. Marcar **TCP/IP** y mostrar sus **Propiedades**.
4. Indicar que se conectará con la IP que se introducirá.
5. Introducir la IP (la máscara se introducirá sola) y aceptar.

BO2K: Configuración

La máquina con la que se trabajará inicialmente es **RyS-TheFarm**. En ella se preparará el troyano escogido, **Back Oriffice 2000 (BO2K)** para luego copiarlo al pincho USB que se usará durante la infiltración física.

Éstas son algunas de las características del troyano:

- Registro de pulsaciones del teclado.
- Navegación del sistema de ficheros y transferencia con restricciones personalizadas.
- Edición directa del registro de Windows.
- Actualización, instalación y desinstalación remotas.

Máquinas Virtuales

- RyS-TheFarm
- RyS-Abstergo

Más información

BO2K: http://www.bo2k.com/

UltraVNC: http://www.uvnc.com/

OllyDbg: http://www.ollydbg.de/

Tutorial OllyDbg: http://ricardonarvaja.info/

- Soporte multimedia (captura de audio y vídeo).
- Gestión de procesos.
- Reinicio remoto.
- Plugins que añaden funcionalidades.

Los archivos del troyano se encuentra en la carpeta **Piri Reis** del **Escritorio**:

- El servidor (**bo2k**), que **bajo ningún concepto se debe ejecutar en esta máquina**.
- El cliente (**bo2kgui**), que sirve para conectarse al servidor.
- El configurador de servidores (**bo2kcfg**), que se usará para personalizar el servidor.
- Una carpeta con diferentes **plugins**, que sirven para mejorar el servidor.

1. Hacer una copia de **bo2k** y llamarla **HiddenBlade**.
2. Ejecutar **bo2kcfg** y abrir el servidor **HiddenBlade**.
3. Insertar **plugins** y configurarlos para que el servidor tenga las siguientes características:
    a. La conexión sea empleando el protocolo TCP a través del puerto 9090.
    b. Tanto la encriptación como la autenticación deber ser nula.

¿Qué diferencias hay entre TCP y UDP?

4. Las categorías **Startup** y **Stealth** forman parte del núcleo del servidor y los valores de sus variables tienen que ser actualizadas/revisadas.
    a. **Init Cmd Net Type** tiene que ser TCPIO.
    b. **Init Cmd Bind Str** debe contener el puerto escogido para la conexión.
    c. **Init Cmd Encryption** y **Init Cmd Auth** deben estar a NULL (o NULLAUTH en la segunda variable).
5. Añadir los siguientes **plugins**:
    a. **srv_control**. Permite las funciones básicas del servidor.
    b. **srv_regfile**. Modificar/revisar las variables de tipo de red, encriptación y autenticación para que coincidan con lo ya introducido antes.
    c. **srv_system**. Permite navegar por el sistema de ficheros remoto y usar ejecutables.
    d. **srv_legacy**. Mecanismos nativos de comprensión de ficheros muy útiles cuando transfieren grandes cantidades de datos.
6. Salvar los cambios en el servidor y salir.
7. Copiar el ejecutable en el pincho USB y extraerlo.

BO2K: Inoculación

Ahora es el momento de arrancar la máquina **RyS-Abstergo** en la que se simulará una infiltración física.

1. Introducir el pincho USB.

2. Ejecutar **HiddenBlade**.
3. Indicar al **Firewall de Windows** que se desea **no bloquear** el programa.
4. Extraer el pincho USB.
5. Ejecutar en una terminal:
    a. **netstat -an** para verificar que el troyano ha abierto el puerto deseado.
    b. **ipconfig** para anotar la IP.

BO2K: Conexión y uso

Vuelta a **RyS-TheFarm**, se procederá a configurar el cliente del troyano para poder utilizarlo.

1. Ejecutar **bo2kgui**.
2. En la sección de **plugins**, insertar los empleados en el servidor para definir la encriptación, la autenticación y el protocolo de conexión (modificar convenientemente el puerto).
3. Añadir el plugin **cli_botools** y modificar/revisar las variables.

Ahora es el momento de crear una conexión con la máquina objetivo.

1. Ir a **File->New Server** e introducir los siguientes datos:
    a. Nombre de la máquina: Abstergo.
    b. Dirección: la IP anotada anteriormente.
2. Al hacer doble click sobre el servidor que se acaba de añadir, aparecerán todas las opciones para el mismo. Conectar con la máquina objetivo.
3. A través de las opciones nuevas que aparecen, obtener:
    a. Información de la máquina remota.
    b. Procesos en ejecución.
    c. Plugins del servidor.
4. Se desactivará el cortafuegos remoto usando la opción de **comenzar un nuevo proceso**.
    a. El proceso será oculto.
    b. El comando asociado al proceso será: **netsh firewall set opmode disable**.

    Nota: de no realizarse esta acción, las siguientes operaciones, en las que se usan puertos distintos al 9090 se verán bloqueadas por el cortafuegos.

VNC: Despliegue

La bajada del cortafuegos puede hacer saltar algunas alarmas, por lo que será necesario cambiar la arquitectura del ataque para que las conexiones se originen desde la máquina objetivo.

Ahora se hará uso de las **BO Tools**, que se cargaron en el cliente con el plugin correspondiente, para instalar un servidor **Virtual Network Computing (VNC)**.

1. Acceder al navegador del sistema de archivos (menú **Plugins**).
2. Conectar con la máquina objetivo.

3. Comenzar a navegar en la unidad **c:** y acceder a la carpeta **Animus**, ubicado en el **Escritorio** del usuario **John**.
4. Subir el ejecutable **winvnc_SCIII.exe** que puede encontrarse en la carpeta **vnc**, que a su vez se encuentra en la carpeta **Piri Reis**.

El siguiente paso será utilizar el servidor VNC para que actúe de forma inversa, esto es, que en vez de aceptar conexiones, sea él quien conecte al cliente.

1. En local, arrancar desde la terminal el ejecutable **vncviewer_ssl.exe**, ubicado también en la carpeta **VNC**, pasándole como parámetro: **-listen**.
2. **No emplear las BO Tools**, ya que no permiten la ejecución con parámetros.
3. Volver a subir el cortafuegos.
4. Ejecutar el servidor VNC usando la misma funcionalidad empleada para bajar el cortafuegos:
    a. El proceso será oculto.
    b. El parámetro será: **-connect IP:5500**, siendo IP la de la máquina en la que se inició el cliente.

VNC: Conexión y uso

Una vez arrancado el servidor, éste se conectará al cliente, saliendo un diálogo que se deberá aceptar. Enseguida cargará una visualización del escritorio de la máquina objetivo, así como una barra de herramientas en la parte de arriba.

1. Visualizar el contenido de la carpeta **Profiles**, dentro de la carpeta **Animus**.
2. A través del **gestor de transferencias de ficheros**:
    a. Copiar en una carpeta (creada fuera de la aplicación) los archivos de la carpeta remota **Profiles**. Borrar los archivos de la ubicación remota.
    b. Copiar **AnimusPatch.EXE**.

Borrado de huellas

Ahora toca borrar los ejecutables y archivos asociados a **bo2k** y **VNC**. Esto se hará a través del primero.

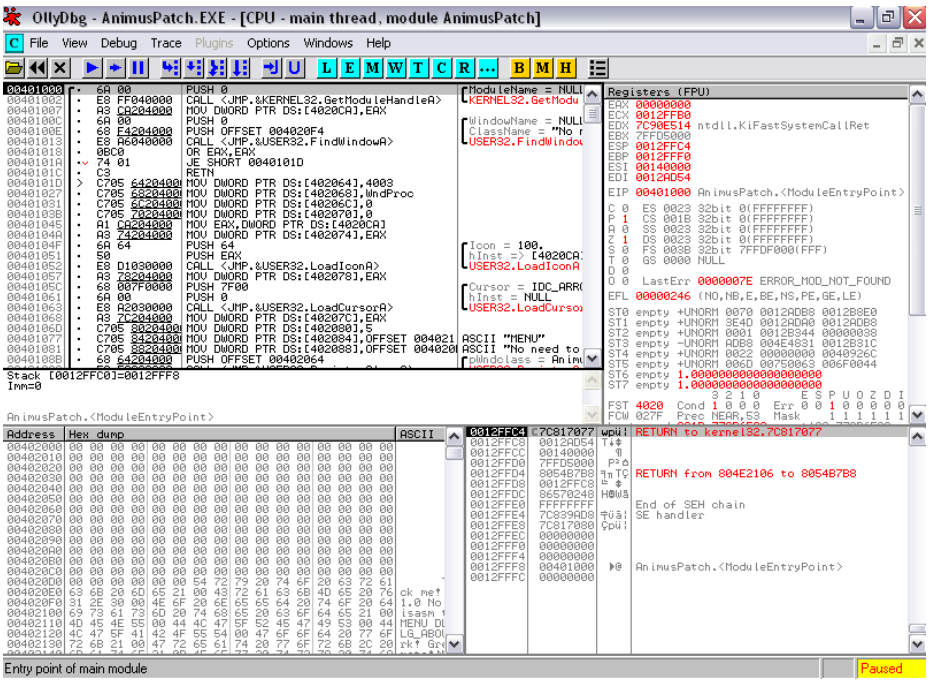
1. Salir de **VNC**.
2. A través de las **BO Tools**, borrar el servidor VNC.
3. A través del cliente estándar de **bo2k**, apagar el servidor (**Shutdown**), indicando que se quiere **erradicar**.

Crack de la aplicación

Entre el botín obtenido, se encuentra **AnimusPatch**. Se procederá a ejecutarlo. Yendo a **Help->Register** se verá que es necesario introducir un nombre y un número de serie que no se dispone, por lo que será necesario desensamblarlo.

Para ello, se usará **OllyDbg**, el cual se encuentra en la carpeta **Leonardo** del **Escritorio**.

1. Abrir el ejecutable en **OllyDbg**.
2. La interfaz está divivido en 4 sectores:



En la primera fila está el **desensamblado** y el contenido de los **registros**. En la segunda, el **volcado** y la **pila**.

3. Resaltar los saltos y las llamadas en el **desensamblado** (**botón derecho->Appearance->Highlighting**).
4. Ir al **EBP** y después al **ESP** en la **pila** usando las direcciones almacenadas en los **registros**.
5. Ir al **EBP** y después al **ESP** en la **pila** usando la opción que aparece al pulsar el **botón derecho**.

¿A qué corresponde el registro **EIP**?

6. Avanzar la ejecución en 1 instrucción con **F7** (**step into** en la barra de herramientas). Observar la variación en los registros a medida que se pulsa.
7. Ejecutar todo el programa con **F9** (**Run**). Arrancará **AnimusPatch** (“CrackMe v1.0”). Cerrarlo.
8. Cerrar el análisis con **ALT+F2** (icono de la **X**, tercero de la izquierda en

la barra de herramientas). Volver a abrir **AnimusPatch** para el análisis.

La instrucción **NOP** es una que no hace nada y se suele usar para alterar la lógica del programa a base de reescribir ciertas instrucciones clave.

1. Pinchar en la primera instrucción (**PUSH 0**) y pulsar **espacio** para invocar la ventana de ensamblado.
2. Sustituir la instrucción existente con **NOP**.

¿Cuántos **NOP** aparecen? ¿Por qué? **PISTA: L**a instrucción **NOP** ocupa 1 byte

1. Ir a la instrucción ubicada en la dirección **401364** (usar la opción **Go To->Expression…**).
2. Añadir un **breakpoint** en esa dirección (**F2**).
3. Ejecutar todo el programa. Nuevamente **AnimusPatch** arrancará.
4. Introducir un nombre y un número de serie al azar.

¿Por qué se ha pausado **OllyDbg**?

1. Comenzar con el análisis desde cero.
2. Cambiar la primera instrucción por **CALL 401245**. Indicar que no se desea que mantenga el tamaño.
3. Ejecutar todo el programa.

¿Qué ha habido diferente de la ejecución anterior?

1. Localizar en el **desensamblado** la rutina de API **&USER32.MessageBoxA** (a la que llama **AnimusPatch**).
2. Indicar que se desea seguir esta importación.

¿A qué dirección se ha movido el **desensamblador**?

¿Está en el espacio de direcciones de **AnimusPatch**? ¿Por qué?

3. Buscar referencias a esa llamada.
4. Pulsar sobre las mismas.

¿En qué dirección comienza el mensaje de error? ¿y el de éxito?

5. Anotar la dirección de referencia de la primera instrucción del mensaje de error ("**Local call from"**).
6. Ir a la llamada o función que la referenció ("**Jump or call to selection…**").

A este punto, la situación es la siguiente:

- Hay una instrucción de salto (**JE**) que compara los valores del nombre y el número de serie.
- Por defecto éste no se tomará (a no ser que se haya sido tan afortunados con la combinación).
- La decisión se toma en función del **Flag Z** mostrado en el apartado de los registros.

1. Poner un **breakpoint** en el salto.
2. Ejecutar todo el programa. Introducir en **AnimusPatch** cualquier nombre y número de serie.
3. Cambiar el bit del **Flag Z** a 1.
4. Ejecutar todo el programa y comprobar el resultado.

¿Qué indica el **Flag Z** y en base a qué obtiene su valor en el programa?
Crackear el programa con instrucciones **NOP**