GOBIERNO DE ESPAÑA
MINISTERIO DE LA PRESIDENCIA Y PARA LAS ADMINISTRACIONES TERRITORIALES

CCN
CENTRO CRIPTOLÓGICO NACIONAL

Guía de Seguridad de las TIC
CCN-STIC 950

RECOMENDACIONES DE EMPLEO DE LA HERRAMIENTA EMET

Abril 2017

Edita:

GOBIERNO DE ESPAÑA
MINISTERIO DE LA PRESIDENCIA Y PARA LAS ADMINISTRACIONES TERRITORIALES

CCN Centro Criptológico Nacional

CENTRO CRIPTOLOGICO NACIONAL
2.5.4.13=Qualified Certificate: AAPP-SEP-M-SW-KPSC, ou=sello electrónico, serialNumber=S2800155J, o=CENTRO CRIPTOLOGICO NACIONAL, cn=CENTRO CRIPTOLOGICO NACIONAL, c=ES
2017.05.03 16:20:40 +02'00'

NIPO:785-17-034-6

Fecha de Edición: abril de 2017

PRÓLOGO

El uso masivo de las tecnologías de la información y las telecomunicaciones (TIC), en todos los ámbitos de la sociedad, ha creado un nuevo espacio, el ciberespacio, donde se producirán conflictos y agresiones, y donde existen ciberamenazas que atentarán contra la seguridad nacional, el estado de derecho, la prosperidad económica, el estado de bienestar y el normal funcionamiento de la sociedad y de las administraciones públicas.

La Ley 11/2002, de 6 de mayo, reguladora del Centro Nacional de Inteligencia (CNI), encomienda al Centro Nacional de Inteligencia el ejercicio de las funciones relativas a la seguridad de las tecnologías de la información en su artículo 4.e), y de protección de la información clasificada en su artículo 4.f), a la vez que confiere a su Secretario de Estado Director la responsabilidad de dirigir el Centro Criptológico Nacional (CCN) en su artículo 9.2.f).

Partiendo del conocimiento y la experiencia del CNI sobre amenazas y vulnerabilidades en materia de riesgos emergentes, el Centro realiza, a través de su Centro Criptológico Nacional, regulado por el Real Decreto 421/2004, de 12 de marzo, diversas actividades directamente relacionadas con la seguridad de las TIC, orientadas a la formación de personal experto, a la aplicación de políticas y procedimientos de seguridad, y al empleo de tecnologías de seguridad adecuadas.

El Real Decreto 3/2010, de 8 de enero, por el que se regula el Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica (ENS, en adelante), al que se refiere el apartado segundo del artículo 156 de la Ley 40/2015, de 1 de octubre, de Régimen Jurídico del Sector Público, establece la política de seguridad en la utilización de medios electrónicos que permita una protección adecuada de la información.

Precisamente el Real Decreto 3/2010 de 8 de Enero, modificado por el Real Decreto 951/2015, de 23 de octubre, fija los principios básicos y requisitos mínimos así como las medidas de protección a implantar en los sistemas de la Administración, y promueve la elaboración y difusión de guías de seguridad de las tecnologías de la información y las comunicaciones (STIC) por parte de CCN para facilitar un mejor cumplimiento de dichos requisitos mínimos.

En definitiva, la serie de documentos CCN-STIC se elabora para dar cumplimiento a los cometidos del Centro Criptológico Nacional y a lo reflejado en el Esquema Nacional de Seguridad, conscientes de la importancia que tiene el establecimiento de un marco de referencia en esta materia que sirva de apoyo para que el personal de la Administración lleve a cabo su difícil, y en ocasiones, ingrata tarea de proporcionar seguridad a los sistemas de las TIC bajo su responsabilidad.

Abril de 2017

Félix Sanz Roldán
Secretario de Estado
Director del Centro Criptológico Nacional

ÍNDICE

1. INTRODUCCIÓN

La herramienta Enhanced Mitigation Experience Framework (EMET) es una herramienta desarrollada por Microsoft que ayuda a prevenir la explotación de ciertas vulnerabilidades de software de manera exitosa. EMET en sí misma es una herramienta que permite configurar, de manera centralizada y con diferentes granularidades, los diferentes mecanismos de seguridad que incorpora el sistema operativo Windows para defenderse de posibles ataques de explotación de vulnerabilidades. Es interesante destacar, como el propio Microsoft indica [1], **que estos mecanismos de seguridad no garantizan que las vulnerabilidades no sean explotables, sino que dificultan al máximo su éxito.**

La última versión del software EMET es la versión 5.52, liberada en noviembre de 2016. Esta herramienta es gratuita y de libre descarga en https://www.microsoft.com/en-us/download/details.aspx?id=54264. Como software adicional requerido para el funcionamiento de EMET, se necesita el framework 4.5 de .NET. Además, si se desea que EMET funcione sobre Internet Explorer 10 en Windows 8, ha de instalarse el parche de compatiblidad KB2790907 [2].

Cabe comentar que Microsoft anunció recientemente que no va a dar soporte (es decir, nuevas actualizaciones) a la herramienta EMET. A pesar de que inicialmente la nota de prensa databa su final de vida el 27 de enero de 2017, debido a quejas de usuarios se decidió extender esta fecha hasta el 31 de julio de 2018. Por tanto, a partir de esta citada fecha, la herramienta EMET dejará de tener soporte oficial de Microsoft.

En esta guía se repasa la herramienta EMET a nivel de qué protecciones ofrece, cómo se ha de configurar, y se comprueba su funcionamiento bajo escenarios de ataque diseñados específicamente para las pruebas. En concreto, estos escenarios tratan de conseguir explotar de manera exitosa una vulnerabilidad de desbordamiento de búffer en una aplicación diseñada para las pruebas. En estos escenarios se prueban las medidas de mitigación introducidas por EMET. Se pretende mostrar cómo EMET es capaz de detectar estas situaciones, evitando así explotaciones exitosas de vulnerabilidades en el sistema.

La estructura de este documento es como sigue. En la Sección 3 se describe en detalle las técnicas de mitigación que ofrece EMET, tanto a nivel de sistema como a nivel de aplicación. La Sección 4 explica el proceso de instalación y configuración de la herramienta en detalle. La Sección 5 muestra unas pruebas de concepto para verificar que efectivamente la aplicación EMET realiza su cometido con algunas de los métodos de protección que aporta. Finalmente, la Sección 6 concluye esta guía y establece unas recomendaciones finales.

2. Descripción de EMET

La herramienta EMET presenta varias ventajas como herramienta de seguridad ante ataques de explotación de vulnerabilidades. Por ejemplo, es capaz de aplicar las mitigaciones que ofrece a cualquier aplicación que se esté ejecutando en el sistema operativo (siempre que no existan problemas de compatibilidad durante la ejecución) sin ser necesario el código fuente para recompilar la aplicación con opciones de seguridad adicionales. Además, EMET es altamente configurable, permitiendo seleccionar protecciones a nivel de sistema operativo o a nivel de procesos concretos. EMET también permite protegerse de ataques *man-in-the-middle* por el uso de certificados SSL fraudulentos.

Como resumen global, las opciones que permite EMET son:

- **Mitigación, a nivel de sistema** (es decir, aplicable a cualquier ejecutable de Windows), de diversas técnicas normalmente usadas para la explotación de vulnerabilidades en sistemas Windows. Las protecciones a nivel de sistema son, en concreto, DEP (Sección 0), SEHOP (Sección 0) y ASLR (Sección 0). Estas protecciones se activan como se explica en la Sección 3.2.1.1.
- **Mitigación, a nivel de proceso,** de las mismas técnicas anteriores y algunas adicionales. Estas protecciones son las que describen de la Sección 0 a la Sección 2.10. Estas protecciones se activan como se explica en la Sección 3.2.1.2.
- **Certificados de confianza**, que permite validar los certificados SSL digitalmente firmados comprobando la autoridad certificadora (Root Certificate Authority, Root CA) para detectar posibles ataques de man-in-the-middle mientras se navega en Internet. Este tipo de protección se describe en más detalle en la Sección 2.12.

A continuación, se detallan todas las protecciones que ofrece EMET, a nivel de mitigación, explicando además en qué consisten cada una de estas protecciones. La mayor parte de esta información se encuentra también en [5].

2.1 Data Execution Prevention (DEP)

El mecanismo de DEP se introdujo a partir de Windows XP. Este mecanismo de protección se diseñó como mecanismo de defensa de las explotaciones de vulnerabilidades de desbordamientos de búffer. Estas vulnerabilidades se basan en colocar el código que se ejecutará de forma arbitraria en la pila. Sin embargo, la pila no tiene por qué encontrarse en una zona de memoria que tiene permisos de escritura y de ejecución. Así, se introdujo el mecanismo Data Execution Prevention (DEP) para

controlar que las zonas de memoria que contienen información temporal (e.g., la pila, el heap), que deben de ser de sólo lectura o escritura, nunca tengan permisos de ejecución.

DEP se habilita en aplicaciones de manera individual siempre y cuando estas aplicaciones hayan sido compiladas con soporte para DEP, indicándolo mediante una flag específica. La herramienta EMET permite que aplicaciones compiladas sin soporte para DEP puedan tener DEP habilitado.

Cabe destacar que existen sistemas, como las máquinas virtuales, donde DEP no está soportado. Por tanto, a pesar de que EMET permite configurarlo, su configuración en estos sistemas no tendrá efecto alguno.

La protección DEP de EMET marca como no ejecutables tanto las páginas de memoria que contienen la pila como la zona de heap. Nótese que algunas aplicaciones pueden no funcionar correctamente con esta protección activada (por ejemplo, aplicaciones que tengan un motor *just-in-time*).

2.2 Structured Exception Handler Overwrite Protection (SEHOP)

Este tipo de mitigación trata de proteger una de las técnicas más comunes para la explotación de desbordamiento de la pila en Windows. SEHOP está disponible en Windows desde Windows Vista SP1. En Windows 7 y posteriores, se permite habilitar/deshabilitar SEHOP a nivel de proceso.

Para poder entender cómo funciona SEHOP, primero ha de entenderse la explotación de desbordamiento de pila, y el uso de los manejadores de excepciones. La pila de un ejecutable en Windows se usa para, entre otras cosas, almacenar las variables locales cuando se entra en un procedimiento o función, junto con la dirección de retorno. A esta dirección de retorno es donde el flujo de ejecución del programa retorna tras ejecutar el procedimiento/función.

Típicamente, el desbordamiento de búffer ocurre en programas donde existe alguna función vulnerable que actúa sobre variables locales de manera insegura. Estas funciones suelen estar relacionadas con manejo de cadenas, escritura, lectura de datos del usuario, etc: al realizar operaciones sobre búffers de manera no controlada, es decir, sin controlar el número de bytes que se leen/escriben/copian, se es capaz de controlar la escritura sobre la pila introduciendo más bytes de los que normalmente se esperan. Así, si un atacante consigue sobreescribir la dirección de retorno (que se guarda en la pila), éste consigue redirigir el flujo de ejecución a un lugar arbitrario cuando el ejecutable vaya a retornar la ejecución a la dirección marcada por la pila.

A modo de ejemplo, considérese el código de ejemplo, escrito en C, de la Figura 1, que se usará como prueba de concepto en esta guía. La función "*readCredentials*" es

vulnerable a desbordamiento de búffer. Concretamente, existe una variable de tipo cadena, "*username*", de 16 bytes de longitud. Esta variable toma valor a través de la función "*scanf*", que recoge el valor introducido por el usuario. Obsérvese, sin embargo, que esta lectura del usuario se realiza de forma insegura: no están acotados los bytes que se leen. Así pues, en el caso de que el usuario introdujera más bytes de los declarados en la variable, se estaría sobreescribiendo la zona de la pila adyacente. Por lo tanto, si se realiza una entrada de datos de una longitud calculada, se es capaz de escribir la dirección de retorno en la pila.

La Figura 2 muestra la pila del programa durante la ejecución de la citada función. La dirección de retorno tras la función está representada por @L1. Si se introdujeran 32 bytes, se llega a sobreescribir @L1, con lo que al acabar la función se consigue controlar la dirección de la siguiente instrucción a ejecutar.

```c
#include <stdio.h>
#include <windows.h>

void readCredentials()
{
    /* Create an array for storing some dummy data */
    char username[16];
    printf ("Enter your username for login, and then press <Enter>: ");
    scanf ("%s", username);

    printf("Hi %s, welcome back! Well coding!\n", username);

    return;
}

int main(void)
{
    printf("$: Welcome aboard!\n");
    readCredentials();
    printf("$: C U soon!\n");

}
```

Figura 1. Código de aplicación vulnerable a desbordamiento de búffer.

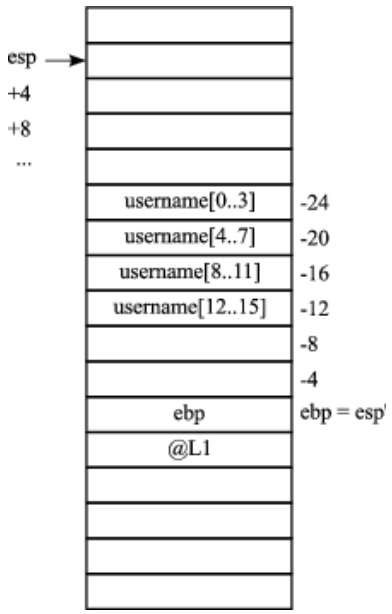


Figura 2. Estado de la pila en la función "*readCredentials*".

Este ejemplo anteriormente descrito es el clásico desbordamiento de búffer explotado. Basado en la misma idea, se puede explotar también esta vulnerabilidad a través de la cadena de menajeadores de excepciones. Un manejador de excepciones es lo que se ejecuta cuando un programa provoca una excepción, con el fin de solucionarla y poder continuar normalmente la ejecución. Los manejadores de excepciones en Windows se mantienen en una lista enlazada, de modo que cuando una excepción ocurre, entra en funcionamiento el primer manejador. En caso de que este manejador no controle la excepción, ésta se pasa al siguiente manejador en la lista. Puede ocurrir que ningún manejador controle la excepción, y entonces el programa (normalmente) es incapaz de continuar su ejecución, acabando con error.

Estos manejadores de excepciones se llaman en Windows *stack-based exception frames*, ya que se encuentran ubicados en la pila como se muestra en la Figura 3 (SEH record, o registro SEH). Un registro SEH tiene un tamaño de 8 bytes, y se compone de dos campos: un puntero al manejador de excepciones actual, y otro puntero al siguiente manejador de la lista.

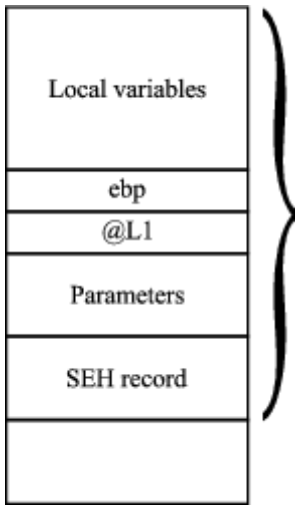


Figura 3. Localización de un registro de manejador de excepción.

Los manejadores de excepciones, entonces, se pueden explotar del mismo modo que se ha explicado anteriormente: sobreescribiendo el registro SEH mediante la escritura controlada en la pila. Sin embargo, la sobreescritura de este registro ha de realizarse siguiendo un esquema concreto como se detalla en la Figura 4.

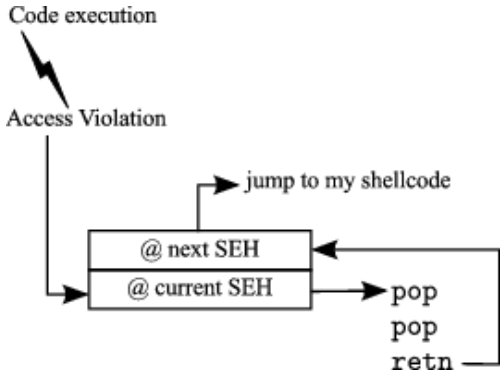
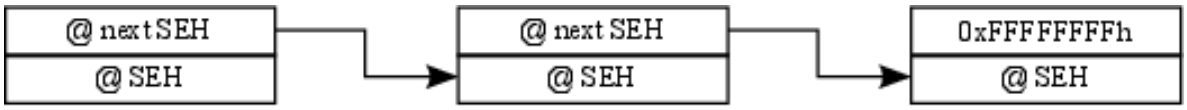


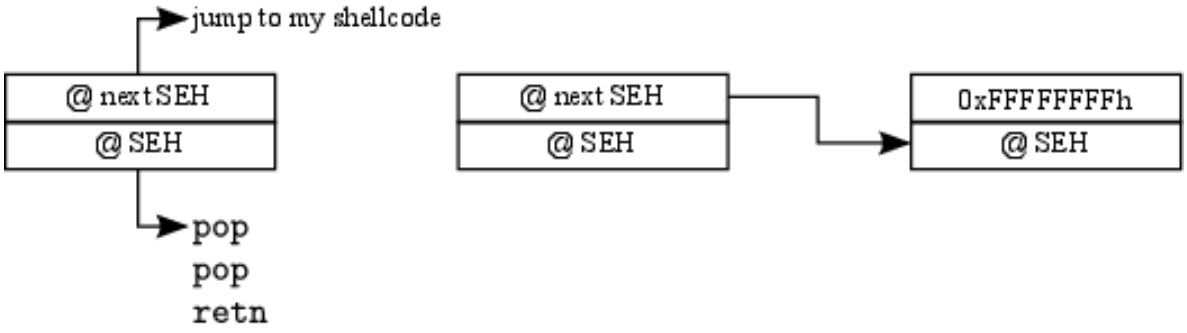
Figura 4. Explotación de manejador de excepciones.

Así, se ha de sobreescribir el puntero del manejador de excepciones actual a una parte del código del ejecutable (o cualquier librería adicional que esté cargada junto con el ejecutable) a una secuencia de instrucciones **pop**, **pop**, **retn**. Estas instrucciones indican al sistema operativo Windows que este manejador de excepciones no ha controlado la excepción, y ha de pasarse al siguiente manejador. Además, el puntero al siguiente manejador se ha de sobreescribir previamente apuntando al código donde se quiere llevar la ejecución, consiguiendo así tener de nuevo el control de la ejecución de la aplicación.

Para conseguir que se ejecute el manejador de excepciones, cabe destacar que después de la sobreescritura controlada del registro SEH será necesario provocar una excepción en la aplicación para conseguir que se ejecute el manejador de excepciones modificado.



(a) Cadena de manejadores de excepciones normal



(b) Cadena de excepciones explotada

Figura 5. Cadena de excepciones (a) normal y (b) explotada.

La protección SEHOP trata de detectar cuando ha habido una modificación en la cadena de manejadores de excepciones, como se muestra en la Figura 5. En caso de

que en un momento de ejecución de los manejadores de excepciones se detecte que la cadena normal (Figura 5(a)) se encuentre modificado (Figura 5(b)), SEHOP para la ejecución de la aplicación sospechosa de haber sido explotada. Sin embargo, la idea de prevención aportada por SEHOP se puede evitar, tal y como se explica en [3].

Por último, cabe destacar que a partir de Windows 7 (en adelante) SEHOP está implementado en el propio sistema operativo (tanto para todo el sistema como por aplicación concreta), por lo que EMET no es capaz de detectar cuándo SEHOP entra en acción, y será el sistema operativo quien cierra la aplicación sospechosa, y reporta él mismo el fallo en vez de EMET. Existen, además, ciertas aplicaciones que pueden ser incompatibles con esta opción de mitigación. En estos casos, se recomienda usar EMET para deshabilitar SEHOP en estas aplicaciones, dejándolo activo en el resto de casos.

2.3 Mandatory Address Space Layout Randomization (ASLR)

En las técnicas de explotación vistas anteriormente es necesario disponer de una dirección de memoria que contenga ciertas instrucciones necesarias para poder completar de forma exitosa una explotación de una vulnerabilidad. Por defecto, esta dirección es fija e idéntica para todas las versiones de la librería o ejecutable que la contenga.

La técnica de mitigación *Address Space Layout Randomization* (ASLR), o aleatorización del mapa de espacio de direcciones, trata de evitar que estas direcciones estén predefinidas y por tanto conocidas, mediante una aleatorización de las direcciones efectivas donde se cargan las librerías de un ejecutable. Esto evita así que las direcciones sean predecibles de antemano.

Esta técnica, al igual que DEP, se ha de habilitar en tiempo de compilación mediante una flag concreta. EMET permite forzar la aleatorización de las direcciones base de los módulos, a pesar de que estos no estén compilados con la flag necesaria. Así, se consigue evitar los intentos de explotación de vulnerabilidades que usan direcciones predefinidas para conseguirlo.

Según la documentación oficial [5], ASLR de forma obligatoria puede ser contraproducente en ciertos sistemas que pueden incluso no arrancar debido a que algunos módulos gráficos no soportan ASLR, provocando un error irrecuperable, y por ende, el sistema no es capaz de arrancar. En el caso de Windows XP y Windows Server 2003 ASLR a nivel de sistema no está soportado, por lo que esta opción de mitigación está desactivada en estos sistemas. A partir de Windows 8 y versiones posteriores, esta mitigación no se aplica en aplicaciones que tienen activado el mecanismo de ASLR forzado por el sistema operativo de manera nativa.

2.4 *Heapspray Allocations*

Como se ha descrito antes, cuando una vulnerabilidad es explotada de manera exitosa el flujo de ejecución se lleva a una dirección arbitraria donde el atacante ha colocado el código a ejecutar. Sin embargo, la localización de esa dirección no puede ser siempre asegurada. Así, un atacante puede usar lo que se conoce como *heapspray allocation*, que consiste en replicar tantas veces como sea posible su código a ejecutar en diferentes sitios de la memoria, a fin de incrementar las posibilidades de éxito.

Cuando EMET está funcionando, las páginas de memoria están pre-reservadas, con lo que un intento de controlar estas páginas fallaría.

2.5 *Null Page Allocation*

Según se documenta en la documentación oficial de EMET v5.52 [5], esta técnica es similar a la anterior, pero se basa en evitar posibles punteros nulos que puedan ser explotados. Actualmente, no se conocen técnicas de explotación usando esta característica.

2.6 Export Address Table Access Filtering (EAF)

Una explotación de cualquier vulnerabilidad necesita conocer dónde se encuentran ciertas funciones de Windows para conseguir su objetivo de explotación. Esta técnica de mitigación trata de proteger la tabla de exportaciones de las librerías de los módulos cargados, de modo que cuando un código intenta leerla o escribirla es bloqueado, inhabilitando así la explotación.

Es conveniente destacar que esta protección puede ser problemática con ciertas aplicaciones que realicen actividades de depuración, o que usen técnicas de anti-depuración, como ciertos mecanismos de protección software (antivirus, firewalls, sandbox, etc.), DRMs o protectores/compresores software (conocidos como *packers*). Tampoco se puede usar en algunas máquinas virtuales, si éstas no soportan registros de debug.

2.7 Export Address Table Access Filtering Plus (EAF+)

Esta técnica de mitigación, introducida a partir de la versión de EMET 5.0, es una extensión de la técnica anterior que puede ser usada tanto de manera independiente como combinada con la anterior.

EAF+ provee una serie de mecanismos de protección de módulos de bajo nivel, previniendo así el uso de ciertas técnicas usadas para construir gadgets ROP de manera dinámica usando las tablas de exportación de los binarios. Esta técnica se

puede activar desde la ventana de "Aplicaciones" del menú de "Configuración" de la ventana de EMET (véase la Sección 3.2.1.2).

Las técnicas de mitigación que incorpora, en concreto, son las siguientes:

- Realiza comprobaciones de integridad adicionales a los registros de pila y los límites de pila, comprobando si los límites están dentro de los permitidos o si existe alguna diferencia entre los registros de pila y los registros de punteros a marco de pila.
- Añade protección en las exportaciones de la librería KERNELBASE, además de las existentes a NTDLL y KERNEL32; en concreto, detectando accesos de lectura de memoria a los punteros de la tabla de exportación de estas librerías originados desde módulos concretos (este tipo de técnica se usa típicamente para la explotación de vulnerabilidades de corrupción de memoria).
- Detecta accesos de lectura de memoria a la cabecera del ejecutable de Windows desde módulos concretos (del mismo modo, este tipo de técnica se usa típicamente para la explotación de vulnerabilidades de corrupción de memoria).

Los dos puntos anteriores requieren al usuario especificar un conjunto de módulos que actuarán como una lista blanca, es decir, el usuario especifica qué módulos son los que se validarán. Si no se especifica ningún módulo, estas dos mitigaciones serán ignoradas.

2.8 Bottom-up randomization

Esta técnica de mitigación trata de aleatorizar (al estilo de ASLR), mediante el uso de 8 bits de entropía, la dirección base de cualquier reserva de memoria que realice la aplicación, incluyendo además la pila y el heap.

2.9 Mitigaciones de ataques Return-Oriented Programming (ROP)

La técnica de explotación ROP trata de aprovecharse de trozos de código presentes tanto en la aplicación como en los módulos cargados de la misma. Estos trozos permiten, encadenándolos, que se alcance el código que el atacante quiere o las condiciones que le interesen en el flujo de ejecución. EMET incluye una serie de mitigaciones para evitar ROP.

Las mitigaciones de ROP que ofrece EMET son, en este momento, las siguientes: (i) chequeo de carga de librerías, evitando la carga de librerías localizadas en red (i.e., usando rutas UNC o en servidores remotos); (ii) protección de las zonas de memoria, haciéndola no ejecutable; (iii) comprobación del origen de una llamada, asegurando que se llegó a ella con una instrucción de llamada y no con una instrucción de

retorno/final de procedimiento (RETN); (iv) simulación de flujo de ejecución, que trata de detectar exploits basados en ROP que hacen llamadas a funciones críticas; (v) pivotación de pila, para comprobar si se ha pivotado (cambiado) la pila. Este tipo de técnica se usa durante la explotación para conseguir cambiar los valores de registro de pila de manera adecuada para el atacante.

2.10 Attack Surface Reduction (ASR)

La mitigación introducida por Attack Surface Reduction (ASR), o reducción de la superficie de ataque, en español, consiste en permitir el bloqueo de ciertos módulos específicos o incluso plugins (extensiones) dentro de la aplicación a proteger. Se consigue, por tanto, reducir los posibles vectores de ataque de que dispone un atacante. En concreto, se puede decir que esta técnica habilita un control para determinar qué módulo/extensión debe de cargarse y cuál no, a nivel de proceso.

Por ejemplo, se puede configurar para evitar que la aplicación de Microsoft Word cargue la extensión para reproducir de Adobe Flash (vector de ataque habitual en este programa) o el incluso evitar que Internet Explorer cargue el plugin de Java en una web de Internet o para permitirlo en caso de ser una web de la intranet. También es posible evitar la carga de ficheros VBScript, frecuentemente utilizado como vector de ataque por el ransomware [9] (por ejemplo, en el ataque conocido como “VBScript God Mode”). En la Figura 6, extraída de [7], se muestra un ejemplo de este ASR funcionando sobre Windows Word, y cómo se notifica el bloqueo de la carga del plugin Adobe Flash Player r12.0.

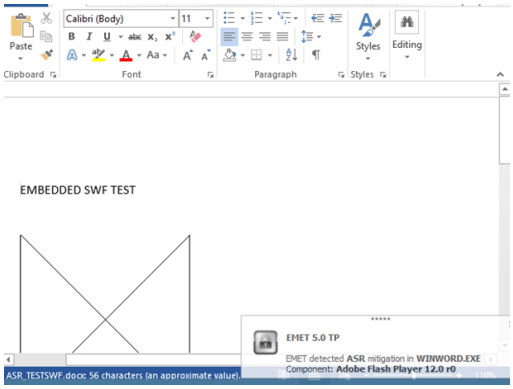


Figura 6. Bloque de carga de plugin en Microsoft Word y notificación por ASR en EMET (extraía de [7]).

2.11 Otras mitigaciones avanzadas de ROP (a nivel de funciones)

Existen también otras técnicas de mitigación que incorpora EMET y se pueden aplicar a cualquier aplicación, siempre que esta lo soporte. Estas técnicas son: (i) deep hooks, que derivan ciertas funciones críticas de Windows y las protegen de ser usadas para fines de explotación (funciones como VirtualAlloc o NtAllocateVirtualMemory); (ii) anti detours, que permite detectar cuando una explotación de una vulnerabilidad replica las primeras instrucciones de una función de Windows (llamado prólogo de la función), y después ejecuta dicha función a partir del prólogo; y (iii) funciones prohibidas, que permite configurar una lista de las funciones que se quieren bloquear (como por ejemplo, LdrHotPatchRoutine).

Nótese que todas estas técnicas avanzadas de mitigación funcionan a nivel de las funciones o APIs de Windows (es decir, es una mitigación que afecta a todas las aplicaciones que se ejecuten de forma protegida). Estas técnicas se activan en cualquier aplicación que tenga una de las opciones de anti-ROP activadas.

Por último, cabe destacar que estas técnicas adicionales aquí descritas funcionarán cuando estén activadas sobre todos aquellos programas que tengan alguna de las mitigaciones de ROP configuradas.

2.12 Certificados de confianza (*certificate pinning*)

La herramienta EMET permite la comprobación de la cadena de certificados SSL durante una nevegación web, con el fin de detectar un posible ataque de man-in-the-middle. Un atacante podría colocarse en medio de una comunicación y suplantar la identidad de la web destino. Para ello, EMET valida el certificado SSL de la entidad final y la autoridad certificadora raíz (Root CA) con una regla de flitrado configurada por el usuario.

Nótese que EMET por defecto sólo produce un aviso cuando se detecta una conexión a página web con alguna anomalía en su certificado, pero en ningún momento bloquea la conexión. Se puede, sin embargo, configurar para que ésta se bloquee (véase la Sección 3.2.1.3).

2.13 Otras mejoras introducidas en EMET

Además de todas las protecciones anteriores, existen también otros mecanismos de mitigación integrados en EMET. Un ejemplo de ellos es Control Flow Guard (CFG) [8], una característica de compilación de Visual Studio 2015 (con soporte en Windows 8.1 y Windows 10) que ayuda a prevenir la aparición de vulnerabilidades de corrupción de memoria. Esta característica se introdujo a partir de EMET 5.2. Todas los ficheros DLL que se distribuyen con EMET desde entonces están compilados con esta protección.

Respecto a las posibilidades de alerta y reporte de EMET, a partir de la versión 5.1 se introdujo la característica de “Local Telemetry” (telemetría local), que permite almacenar de manera local volcados de memoria cuando se produce una mitigación por EMET. Estos ficheros pueden ser útiles posteriormente para un análisis forense del incidente. EMET también incorpora soporte de alerta y reporte total para la aplicación de Modern Internet Explorer o Desktop IE con el modo Enhanced Protected mode activado [11].

A partir de la versión 5.51, además, se ha mejorado la configuración de varias mitigaciones posibles a través de políticas de grupo Windows GPO. Se ha mejorado además la activación de estas técnicas mediante el registro de Windows, lo que permite una integración sencilla con herramientas de gestión mediante políticas GPO. En esta última versión, también se ha añadido una mitigación para prevenir otro vector de ataque común, el relacionado con archivos de fuentes de texto [12] (únicamente para Windows 10).

3. Instalación y configuración

En esta sección se detallan la instalación y configuración de la herramienta. En primer lugar, se describe la instalación. A continuación, se describe la configuración de EMET, respondiendo a qué mitigaciones se deben de habilitar a nivel de sistema, a nivel de aplicación, y qué reglas sobre los certificados SSL/TLS deben de comprobarse.

3.1 Instalación

La instalación de la herramienta EMET es muy sencilla. Del enlace [4] se puede descargar el instalador de la herramienta. Cabe recordar que será necesario tener instalado también en el sistema el framework .NET versión 4.5 o superior.

La Tabla 1 resume las versiones del sistema operativo Windows en las cuales se puede instalar EMET versión 5.52. Nótese que no todas las técnicas de mitigación explicadas en la Sección 3 están disponibles en todos los sistemas operativos. En particular la Tabla 2 recoge, para cada sistema operativo, qué técnicas de mitigación son aplicables. Esta información se ha extraído del manual de usuario de EMET versión 5.52 [4]. La última versión de EMET disponible da compatibilidad de todas las técnicas de mitigación proporcionadas por EMET con todas las versiones del sistema operativo Windows, tanto a nivel de escritorio como de servidor, excepto la técnica de mitigación de “Untrusted fonts”, que sólo aplica a Windows 10. Respecto a la arquitectura del sistema soportado, en arquitecturas de 32 bits todas las técnicas de mitigación a nivel de aplicación son compatibles. En arquitecturas de 64 bits, por el contrario, no se soportan la técnica de SEHOP (Sección 0) ni algunas de las técnicas de mitigación contra ROP (en particular, ni la protección de comprobación del origen de una llamada ni la de simulación de flujo de ejecución, véase la Sección 0).

Sistemas operativos de escritorio	Sistemas Operativos de servidor
Windows Vista SP 2	Windows Server 2008 SP2
Windows 7 SP1	Windows Server 2008 R2 SP1
Windows 8 y Windows 8.1	Windows Server 2012
Windows 10	Windows Server 2012 R2

Tabla 1. Versiones mínimas de Windows para el funcionamiento de EMET 5.52.

	Técnica de mitigación	Windows Vista, 7, 8, 8.1 / Server 2008 and Server 2012	Windows 10
A nivel de sistema	SEHOP	✓	✓
	DEP	✓	✓
	ASLR	✓	✓
	Untrusted fonts		✓
A nivel de aplicación	SEHOP	✓	✓
	DEP	✓	✓
	Heapspray	✓	✓
	Nullpages	✓	✓
	Mandatory ASLR	✓	✓
	EAF	✓	✓
	EAF+	✓	✓
	Bottom-up ASLR	✓	✓
	Anti-ROP	✓	✓
	ASR	✓	✓
	Untrusted fonts		✓

Tabla 2. Compatibilidad entre versiones de Windows y técnicas de mitigación (EMET 5.52).

A continuación se describe el proceso de instalación de EMET 5.52 en un sistema operativo Windows 7. El instalador pregunta sobre el directorio de instalación, como se observa en la Figura 7, y sobre la aceptación de una licencia de uso del software. Antes de instalar, se debe de aceptar la ventana de Windows UAC para permitir al instalador hacer cambios en el equipo. Una vez instalada la aplicación, ésta se lanza mostrando la posibilidad de configurarla de forma automática o de forma manual. Dado que se explica más adelante el proceso manual de configuración, se recomienda optar por la segunda opción ("Configure manually later", Figura 8). Finalmente, tras acabar el proceso de instalación se observa que la aplicación EMET está ejecutándose en segundo plano al localizar en la barra de notificaciones de Windows un icono de un candado negro sobre fondo gris (véase la Figura 9).

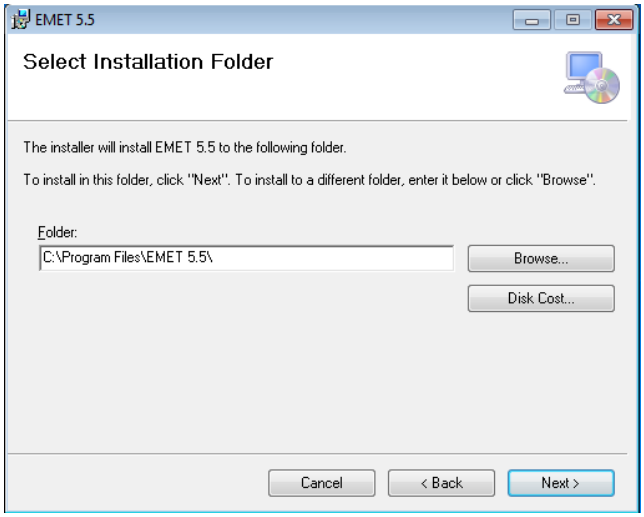


Figura 7. Instalación de EMET 5.21.

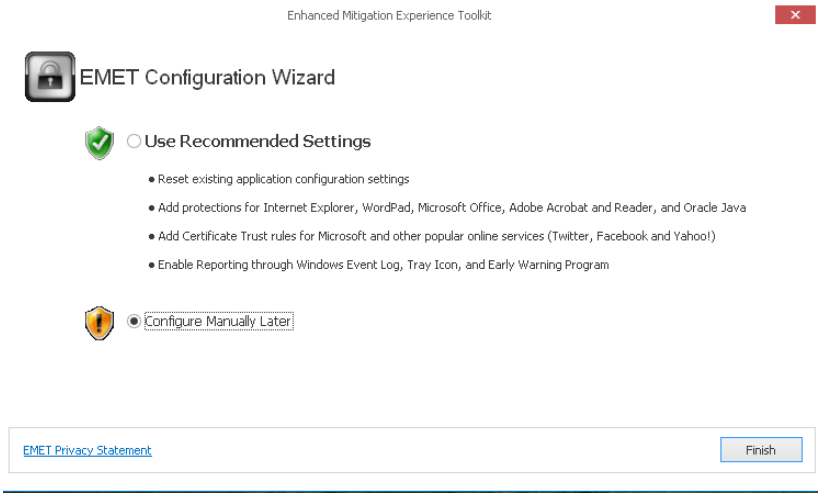


Figura 8. Opciones de configuración de EMET, tras instalación.

Figura 9. Aplicación EMET en la barra de notificaciones, ejecutándose en segundo plano.

El proceso de instalación es igual en cualquier otro sistema operativo. A continuación, se describen las posibilidades de configuración de esta herramienta en más detalle.

3.2 Configuración

La configuración de EMET se puede realizar tanto a través de una interfaz gráfica, muy descriptiva e intuitiva, así como a través de línea de comandos o a través de políticas de grupo de Windows (GPO). Se procede a explicar en el citado orden estas opciones de configuración de EMET.

3.2.1 Configuración mediante la interfaz de usuario

La interfaz gráfica de la aplicación de EMET es la que se muestra en la Figura 10. Se distinguen claramente la parte relativa a la actual configuración de las técnicas de mitigación a nivel de sistema ("System Status"), junto con la opción de confianza de certificados, y la parte relativa a mitigación por procesos ("Running Processes"), que muestra qué procesos se están ejecutando y si se encuentran protegidos por EMET. También se encuentra visible las opciones de "Reporting" de la aplicación. En la imagen se muestra cuál es la configuración por defecto de la herramienta.

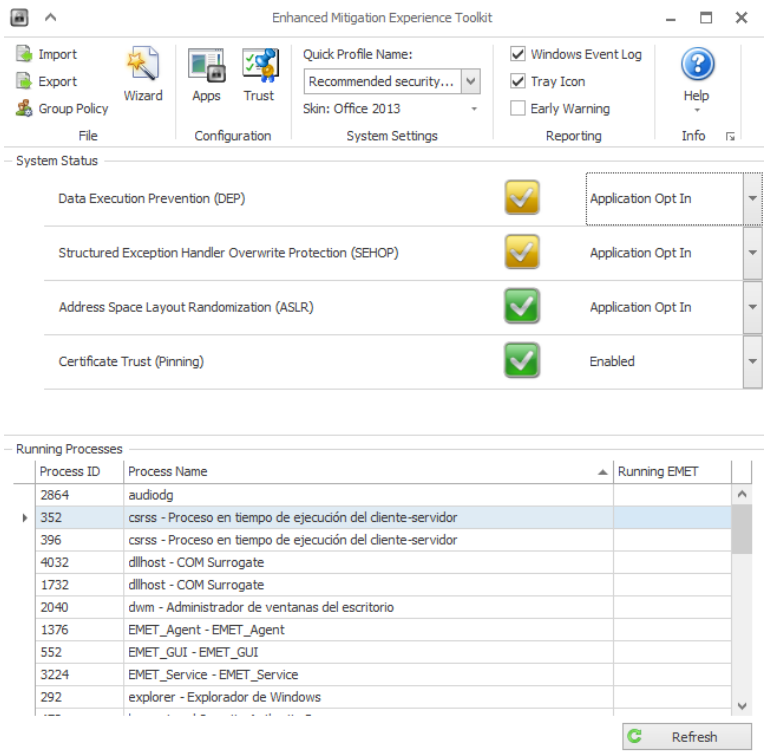


Figura 10. Interfaz gráfica de EMET 5.52.

La instalación de EMET trae dos perfiles de configuración definidos para aplicaciones, y uno para los certificados SSL/TLS. Estos ficheros se encuentran en el directorio de instalación de EMET, en el subdirectorio "Deployment\Protection Profiles". El contenido de este directorio se muestra en la Figura 11.

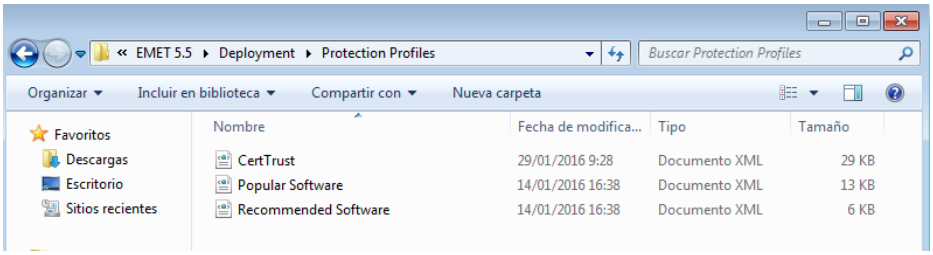


Figura 11. Contenido del directorio "Deployment\Protection Profiles".

Las aplicaciones protegidas por estos perfiles son los siguientes:

- **Recommended Software**: habilita todas las técnicas de mitigación posibles en EMET para las aplicaciones de Internet Explorer, WordPad (excepto la técnica de EAF+ y ASR), las suites de Office 2003, 2007, 2010, 2013 y Office365 (excepto EAF+, y ASR sólo habilitado en Word, Excel y PowerPoint para prevenir la cara de componentes Flash), Adobe Acrobat y Adobe Acrobat Reader 8, 9, 10 y 11 (se deshabilita la *protección* ASR), Oracle Java JRE 6 y 7 (se deshabilita la protección de *Heapspray allocation, EAF+* y *ASR*).
- **Popular Software**: habilita todas las técnicas de mitigación posibles en EMET para las aplicaciones comentadas anteriormente en "Recommended Software", así como en Windows Media Player (se deshabilita ASLR y EAF, SEHOP sólo se habilita si el sistema operativo es Windows 7 o superior), Skype (sólo en arquitecturas x86, se deshabilita EAF), Microsoft Lync Communicator (sólo en arquitecturas x86), Windows Photo Gallery, Microsoft Live Essentials 2012 (sólo en arquitecturas x86), Google Chrome (sólo en arquitecturas x86, activando EAF+ y con SEHOP activado a partir de Windows 7 o superior), Google Talk (sólo en arquitecturas x86, se deshabilita DEP y SEHOP se activa a partir de Windows 7 o superior), Mozilla Firefox (sólo en arquitecturas x86, EAF+ activado), Mozilla Thunderbird (sólo en arquitecturas x86), Adobe Photoshop CS, Winamp (sólo en arquitecturas x86), Opera (sólo en arquitecturas x86), WinRAR, WinZip, VLC (sólo en arquitecturas x86), RealPlayer (sólo en arquitecturas x86), mIRC (sólo en arquitecturas x86), 7-zip (se deshabilita EAF), Apple Safari (sólo en arquitecturas x86), QuickTime Player (sólo en arquitecturas x86), Apple iTunes, Pidgin (sólo en arquitecturas x86), Oracle Java JRE 6 y 7 (se deshabilita la protección de *Heapspray allocation*), FoxIt Reader (sólo en arquitecturas x86).
- **CertTrust**: configura como certificados de confianza las CAs de Yahoo, Facebook, Microsoft y Twitter. Se registran también las correspondientes URLs de inicio de sesión (login.microsoftonline.com, secure.skype.com, www.facebook.com, login.yahoo.com, login.live.com, login.skype.com, twitter.com).

Mediante la opción de "Import" en la ventana de EMET (también a través de línea de comandos o políticas de grupo) se pueden habilitar estos perfiles por defecto (véase la Figura 12). La opción de "Export" permite guardar la configuración realizada por el usuario a un fichero XML, para poder ser posteriormente importada.

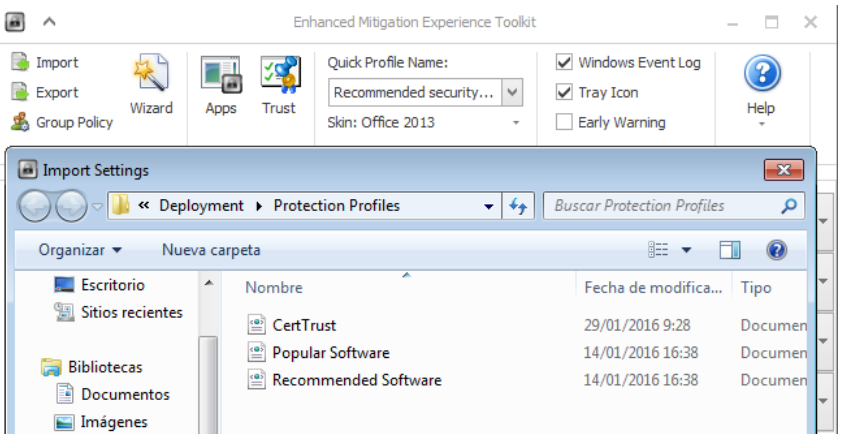


Figura 12. Importación de perfiles de protección.

3.2.1.1 Configuración a nivel del sistema

El desplegable de "Quick Profile Name" permite definir dos perfiles pre-configurados para el sistema: "Maximum Security Settings" y "Recommended Security Settings" (véase la Figura 13).

Figura 13. Desplegable de "*Quick Profile Name*".

Cualquier cambio en esta configuración requiere que se reinicie el sistema para hacerse efectiva. La opción de *"Maximum Security Settings"* habilita todas las protecciones al máximo nivel, como se observa en la Figura 14.

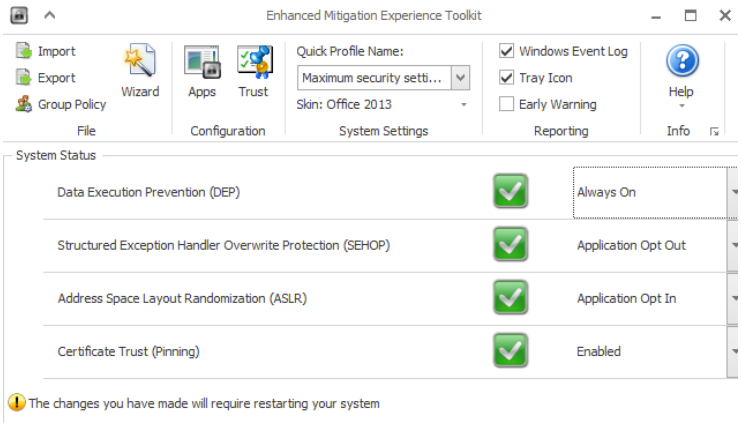


Figura 14. Opciones de mitigación habilitadas con "*Maximum Security Settings*" (en Windows 7).

La opción de "R*ecommended Security Settings*" habilita el nivel de protecciones que se muestra en la Figura 15.

Concretamente, las opciones posibles para DEP son:

- **Disabled**: deshabilitado.
- **Application Opt In**: por defecto no habilitado, excepto que la aplicación lo haya configurado como habilitado.
- **Application Opt Out**: por defecto habilitado, excepto que la aplicación lo haya configurado como deshabilitado.
- **Always On**: siempre habilitado.

Las opciones posibles de SEHOP son "*Application Opt In*" y "*Application Opt Out*", exclusivamente. Para ASLR, las opciones disponibles son *"Application Opt In"* o "*Disabled*".

Obsérvese que en el caso de "*Maximum Security Settings*", DEP está en *"Always On"*, SEHOP en "*Application Opt Out*" y ASLR en "*Application Opt In*".

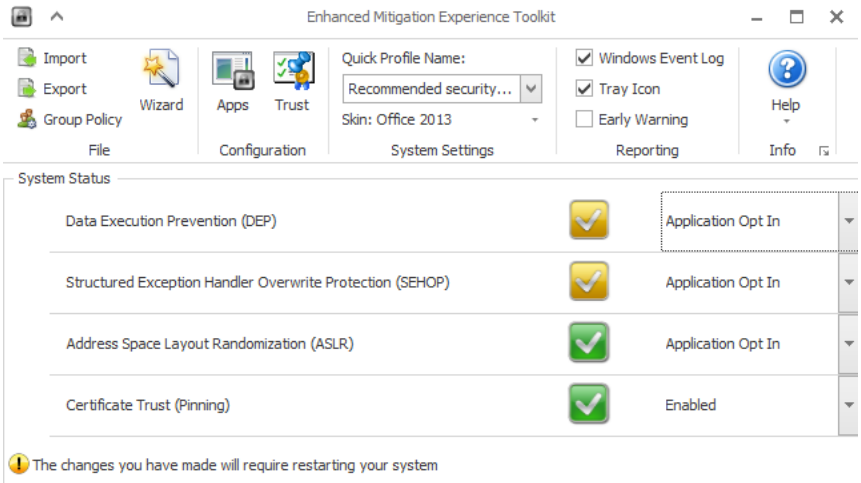


Figura 15. Opciones de mitigación habilitadas con "*Recommended Security Settings*".

En el caso del *Certificate Trust (Pinning)*, las únicas opciones disponibles son "*Enabled*" (habilitado) o "*Disabled*" (deshabilitado). Concretamente, esta opción de protección sólo funciona para Internet Explorer, aunque puede configurarse de manera experimental en otros navegadores. Se recomienda a este respecto consultar la parte relativa a configuración de esta protección en el manual oficial [5].

3.2.1.2 Configuración a nivel de las aplicaciones

Para acceder a la configuración de las aplicaciones, hay que presionar el icono de "Apps" en la interfaz gráfica de EMET (Figura 16), accediendo a la ventana que se muestra en la Figura 17.

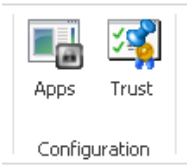


Figura 16. Acceso a la configuración de técnicas de mitigación a nivel de aplicación.

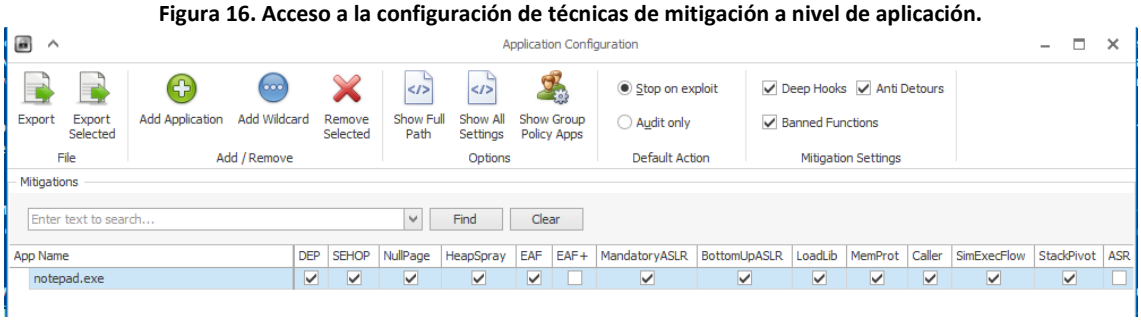


Figura 17. Ventana de configuración de técnicas de mitigación a nivel de aplicación (tras añadir la aplicación "notepad.exe").

Obsérvese que aquí también se puede configurar la acción por defecto a realizar cuando se detecta una explotación en una aplicación (véase la Figura 18). Se permiten dos acciones, "*Stop on exploit*", donde EMET informa del intento de explotación y termina la aplicación explotada, y "*Audit only*", donde EMET informa del intento de explotación pero no termina la aplicación explotada. Esta segunda opción no está soportada por todas las técnicas de mitigación. De hecho, sólo es soportada por EAF, EAF+, anti-ROP, SEHOP (en Windows Vista y Windows Server 2008) y ASR.

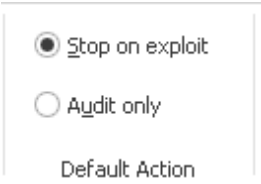


Figura 18. Configuración de acción por defecto cuando se detecta una explotación.

Desde esta venta se permite añadir una aplicación que se quiere proteger con EMET, así como definir las técnicas de mitigación a aplicar. A modo de ejemplo, se va a

añadir la aplicación de "*notepad.exe*". Para ello, hay que presionar el botón de "*Add Application"*, y seleccionar la aplicación que se desea proteger. En este caso, "notepad.exe" se encuentra en la ruta de "*C:\Windows\notepad.exe"*. Una vez seleccionada, aparece en la lista de aplicaciones protegidas, junto con todas las técnicas de mitigación anteriormente descritas (categorizadas todas ellas a través de las diferentes pestañas) y un *checkbox* para rápidamente habilitar o deshabilitar cada protección concreta, como se muestra en la Figura 17. Por defecto, se habilitan todas las protecciones excepto EAF+ y ASR.

3.2.1.3 Configuración de certificados de confianza

Para la configuración de los certificados basta con presionar el botón de "*Trust*" en la ventana principal de EMET (véase la Figura 16), accediendo así a la ventana de configuración de certificados de confianza que se muestra en la Figura 19.

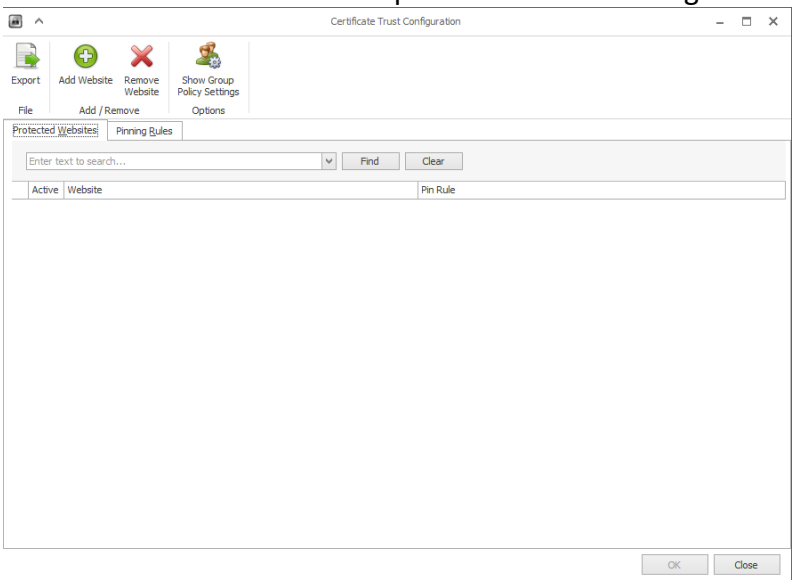


Figura 19. Ventana de configuración de Certificados de Confianza.

Para añadir un nuevo sitio con certificado de confianza, lo primero es definir una "*Pinning Rule*", seleccionando la citada pestaña y presionando el botón "*Add Rule*" en la nueva ventana (véase la Figura 20). Los datos que se piden configurar para la regla son los siguientes:

- **Nombre del certificado** ("*Name*")
- **Certificados** ("*Certificates*"), que permite añadir los certificados de la entidad autenticadora considerados de confianza.
- **Fecha de expiración** ("*Rule Expiration*"), para indicar una fecha a partir de la cual la regla no será más aplicable.

Estos tres valores son obligatorios. Existen unos datos adicionales y opcionales, que permiten además añadir comprobaciones de modo que incluso un sitio web cuyo CA no haya sido añadido a la lista de CAs seguros se aceptará como válido (siempre y cuando cumpla estas comprobaciones adicionales):

- **Mínimo tamaño de la clave** (“*Minimum Key Size*”): permite definir el tamaño mínimo de la clave del certificado, de modo que si ésta es igual o superior a la indicada, el certificado se considera válido.
- **País permitido** (“*Allowed Country*”): permite especificar el país proveedor del Root CA. Del mismo modo, en caso de que el Root CA no coincida, pero sí que lo haga su país proveedor con el especificado, éste se toma como certificado válido.
- **Hashes bloqueados** (“*Blocked Hashes*”): permite especificar los hashes de los certificados que se quieren bloquear. En caso de que un Root CA no tenga su hash bloqueado, se acepta como válido a pesar de que no esté definido como certificado válido.
- **Coincidencia con la clave pública** (“*PublicKey Match*”): con esta opción activada EMET sólo comprobará la clave pública del certificado, sin importar otra información como el sujeto y el número de serie.
- **Regla bloqueante** ("*Blocking Rule*”): con esta opción activada, EMET bloqueará la conexión totalmente con la página web que no cumpla la cadena de certificados. Nótese que por defecto EMET sólo avisa al usuario del problema en la cadena de certificados, pero permite que la conexión continúe.

Una vez añadida una regla, se puede agregar un dominio en la lista de direcciones web protegidas, seleccionando la regla recién creada como “*pinning rule*” a aplicar para el dominio protegido. Un ejemplo más detallado sobre la creación de reglas y sitios protegidos se puede encontrar en [6].

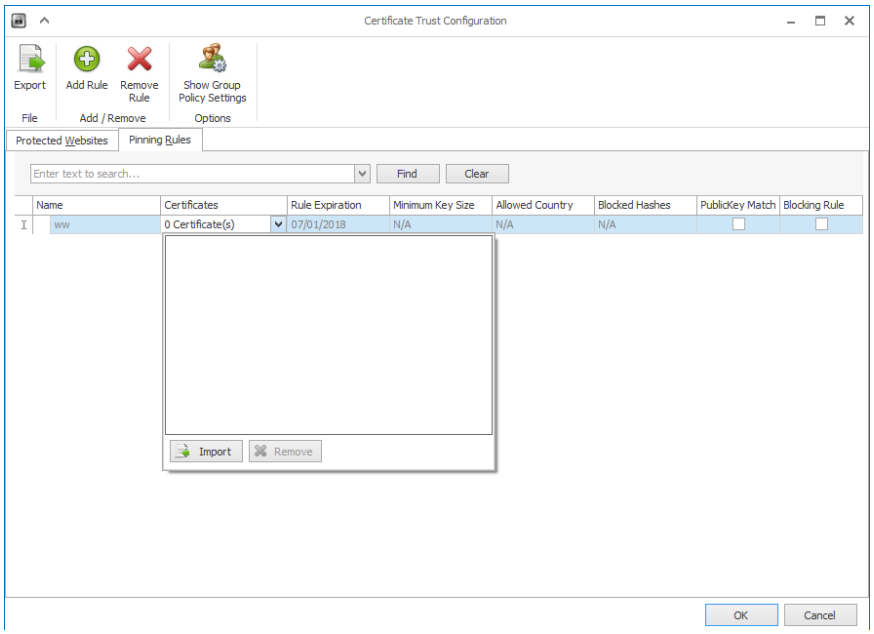


Figura 20. Ventana de configuración de una nueva “*pinning rule*”.

Por último, cabe comentar las posibilidades de registro y notificación de eventos que permite configurar EMET. Como se muestra en la Figura 21, existen tres posibilidades de notificación:

- **Registro de Eventos de Windows** ("*Windows Event Log*"): cuando está seleccionada esta opción, cualquier intento de explotación detectado por EMET queda registrado en el fichero de registro de eventos de Windows.
- **Icono de la barra de notificaciones** ("*Tray Icon*"): en este caso, cuando ocurre un intento de explotación, se informa al usuario mediante una ventana emergente en la barra de notificaciones de Windows.
- **Aviso Temprano de Windows** ("*Early Warning*"): cuando está seleccionado, se genera información relativa al ataque, como un volcado de la memoria y el tipo de mitigación que ha evitado el ataque, para ser enviada a Microsoft a través de su canal de comunicaciones de error. Cabe destacar que antes de realizar el envío se pregunta al usuario sobre su conformidad para el envío. Por último, nótese que esta opción no está disponible para la mitigación de certificados de confianza.

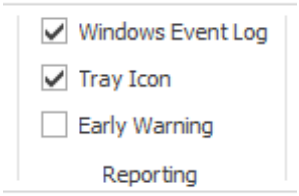


Figura 21. Posibilidades de configuración de las notificaciones de EMET.

3.2.2 Configuración mediante línea de comandos

La herramienta EMET contiene también un programa (*"EMET_Conf.exe"*, disponible en el directorio de instalación de EMET) para configurarla bajo línea de comandos. Todos los pasos vistos anteriormente, a través de diversas pantallas gráficas, se pueden realizar mediante una invocación de la línea de comandos con los parámetros acordes. En la Tabla 3 se muestran todos los posibles parámetros, junto con una breve descripción de su significado.

Comando	Descripción
--set [--force] <ruta a ejecutable>	Añade una aplicación, indicada por la ruta del ejecutable, a la lista de aplicaciones a proteger. La ruta ha de ser la ruta completa, permitiéndose máscaras (e.g. *) sólo en la ruta, y no en el nombre del ejecutable. También se permite incluir sólo el nombre del ejecutable a proteger. La opción --force indica que EMET incluya la aplicación en la lista de protegidas, a pesar de que ésta no se encuentre instalada en el sistema. **Ejemplos**: *EMET_Conf --set prueba.exe* (habilita todas las protecciones) *EMET_Conf --set *\prueba.exe --DEP* (habilita todas las protecciones menos DEP)

--list	Muestra todas las aplicaciones protegidas por EMET, así como la configuración de las políticas de grupo definidas.
--list_system	Muestra la configuración de la protección para el sistema habilitada por EMET, así como la configuración de las políticas de grupo definidas.
--list_certtrust	Muestra todos los sitios registrados como certificados de confianza, y sus reglas definidas.
--delete <ruta a ejecutable>	Elimina una aplicación de la lista de aplicaciones protegidas de EMET.
--delete_apps	Elimina todas las aplicaciones de la lista de aplicaciones protegidas de EMET.
--delete_certtrust	Elimina todos los sitios registrados como certificados de confianza, así como sus reglas definidas.
--delete_all	Elimina tanto las aplicaciones configuradas como los certificados de confianza de EMET. Es equivalente a --deleta_apps y --delete_certtrust.
--system [--force] <SysMitigation=State>	Permite configurar el nivel de las técnicas de mitigación a nivel de sistema. La opción de --force es necesaria cuando el cambio puede provocar inestabilidad en el sistema operativo, como por ejemplo colocar "Always On" en ASLR. Como se ha comentado anteriormente, esto puede provocar errores irrecuperables durante el arranque del sistema, provocando su inutilización.
--deephooks (enabled|disabled) **--antidetours (enabled|disabled)** **--eafplus (enabled|disabled)**	Permite activar o desactivar cada una de estos mecanismos avanzados de mitigación de manera independiente.
--import <fichero XML>	Importa la configuración de EMET de un fichero XML.
--export <fichero XML>	Exporta la configuración de EMET actual al fichero dado.
--reporting (+|-)(telemetry|eventlog|trayicon)	Especifica el modo de reporte de EMET que se desea: *telemetry*, para activar o desactivar el programa de Aviso Temprano de Windows; *eventlog*, para registrarlo en el sistema de Eventos de Windows; o *trayicon*, para informar mediante una notificación visual al usuario.
--exploitation (audit|stop)	Permite configurar el comportamiento de EMET ante una explotación: o bien terminar el programa (*stop*), o bien registrar la explotación y no terminar el proceso (*audit*).
--agentstarthidden (enabled|disabled)	Permite configurar la visibilidad del agente de EMET en la barra de tareas.

Tabla 3. Opciones de configuración de EMET por vía de comandos.

3.2.3 Configuración mediante políticas de grupo (para entornos empresariales)

La herramienta de EMET provee también el soporte para políticas de grupo (aplicable a Windows 7 y superiores, o sus equivalentes en versión servidor). Junto con la herramienta se distribuyen dos ficheros, *EMET.admx* y *EMET.adml* (ambos en el directorio "*Deployment\Group Files Folder*"), que deben de ser copiados respectivamente a los directorios *"%WINDIR%\PolicyDefinitions"* y *"%WINDIR%\PolicyDefinitions\es-ES"* (en caso de disponer de que el sistema operativo esté configurado en otro idioma, deberá de copiarse en el directorio del idioma correspondiente).

Una vez instalado el componente, se puede acceder a él desde la ventana de configuración de políticas de grupo ejecutando el comando "*mmc*" en Windows 7. El panel de configuración de EMET se encuentra en "*Directiva Equipo local\Configuración del equipo\Plantillas administrativas\Componentes de Windows\EMET*". Desde aquí,

se tiene acceso a diversas opciones de configuración de EMET, como se muestra en la Figura 22.

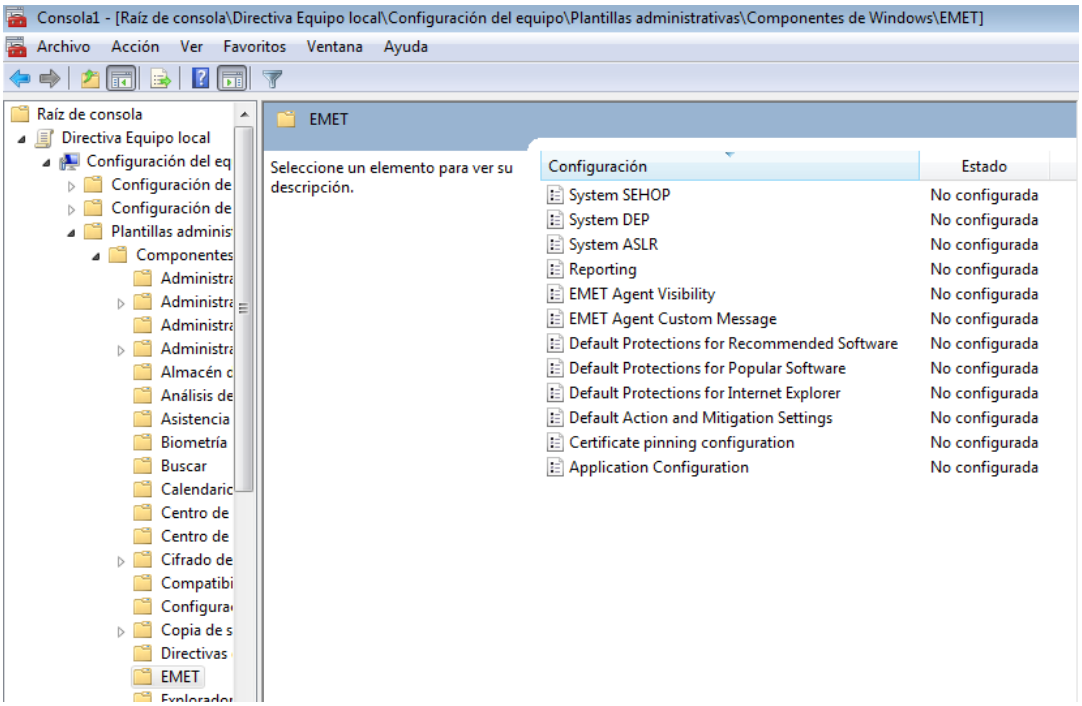


Figura 22. Configuración de EMET a través de directivas de Políticas de Grupo (Windows 7).

Como se observa, se permite configurar a través de esta pantalla las técnicas de mitigación a nivel de sistema de DEP, SEHOP y ASLR (*"System DEP", "System SEHOP" y "System ASLR"*, respectivamente), el tipo de reporte de EMET (*"Reporting"*), la visibilidad del agente de EMET ("*EMET Agent Visibility*"), configurar un mensaje propio para la notificación del agente de EMET ("*EMET Agent Custom Message*"), la configuración por defecto para el software recomendado o el software más popular ("*Default Protection for Recommended Software*" y "*Default Protection for Popular Software*", respectivamente), así como para Internet Explorer ("*Default Protection for Internet Software*"), las acciones y las técnicas de mitigación por defecto ("*Default Action and Mitigation Settings*"), la configuración de verificación de cadena de certificados ("*Certificate pinning configuration*") y la configuración a nivel de aplicación ("*Application Configuration*").

La configuración de aplicaciones desde aquí no es tan intuitiva como a partir de la interfaz gráfica, aunque sigue siendo igualmente sencillo. Una vez habilitado el componente, se ha de especificar las tuplas de valores, siguiendo la sintaxis de "aplicación [-<protección a deshabilitar>]". Por defecto, si no se especifica ninguna protección a deshabilitar, se encuentran todas activadas. En la Figura 23 se muestra un ejemplo de configuración de la aplicación de Inter*net Explor*er, *"iexplore.exe"*, con la opción de DEP desactivada, y la aplicación de *Google Chrom*e, *"chrome.exe"*, con las opciones de *SEHOP* y *EAF* desactivadas.

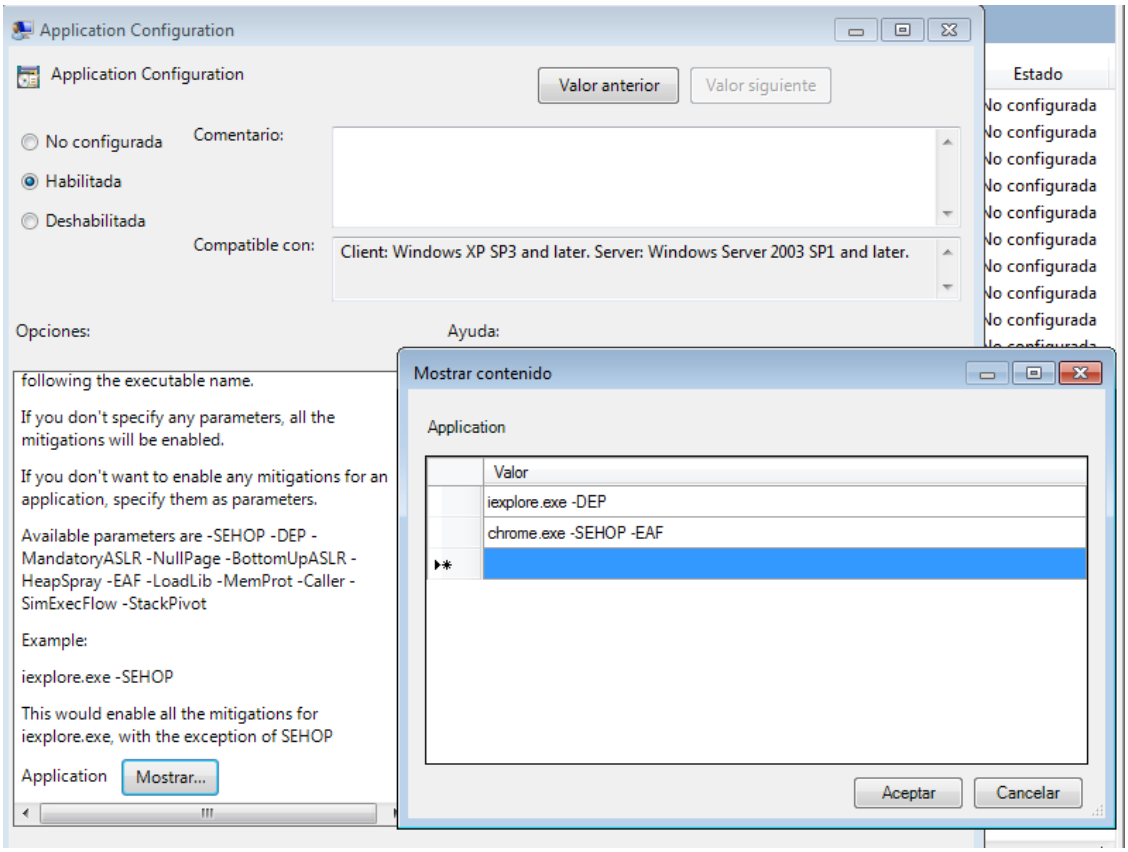


Figura 23. Ejemplo de configuración de aplicaciones a través de Políticas de Grupo.

4. Pruebas de Concepto

Como prueba de concepto, se procede a comprobar si se detecta la explotación de un programa que presenta una vulnerabilidad de desbordamiento de búffer. Este programa es el que se ha mostrado anteriormente en la Figura 1. Considérese un conjunto de bytes (denominado *shellcode* en el contexto de explotación) que contiene el código de explotación, y que es el que se obtiene tras la ejecución del código de Python que se muestra en la Figura 24.

Para este ejemplo en particular, la shellcode esqueleto usada se muestra en la Figura 25. Esta shellcode trata de ejecutar la calculadora de Windows a partir del código de la aplicación vulnerable. Así, cuando la explotación sea exitosa, se lanzará una ventana de la calculadora. En caso contrario (la explotación falla), no aparecerá la calculadora. Para las pruebas se ha usado el sistema operativo de Windows 7 ejecutándose en una máquina virtual Virtual Box.

```
bufLen = 28

buf='A'*bufLen

retn = "\xF7\xF8\x31\x76"           #kernel32.dll -- jmp esp
shellcode = "\x31\xC9"              #xor ecx, ecx
shellcode += "\x51"                 #push ecx
shellcode += "\x68\x2E\x65\x78\x65" #push 6578652E
shellcode += "\x68\x63\x61\x6C\x63" #push 636C6163
shellcode += "\x8B\xCC"             #mov ecx, esp
shellcode += "\x6A\x05"             #push SW_SHOW
shellcode += "\x51"                 #push ecx
shellcode += "\xBA\xFD\xE5\x2e\x76" #mov edx, kernel32.WinExec
shellcode += "\xFF\xD2"             #call edx
shellcode += "\xEB\xFE"             #jmp $EIP

print buf+retn+shellcode
```

Figura 24. Código Python para explotar el programa vulnerable de la Figura 1.

```
C:\poc-EMET>python shellcode.py
AAAAAAAAAAAAAAAAAAAAAAAAAAAA¸°1v1┌Qh.exehcalcï╎j♣Q╎²õ.v ÊÙ■

C:\poc-EMET>
```

Figura 25. Conjunto de bytes (shellcode) que explota el programa de la Figura 1.

En el caso de una configuración de EMET sin estar la aplicación vulnerable "*bof.exe*" preconfigurada ni ninguna otra protección del sistema activa, se observa que la explotación de la aplicación se realiza de forma exitosa (véase la Figura 26).

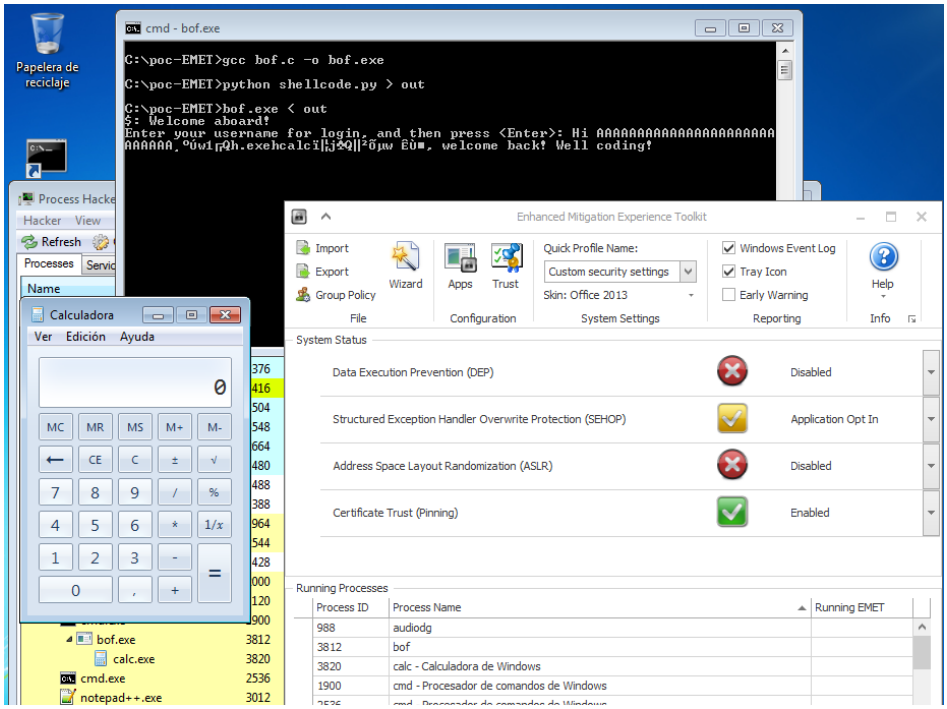


Figura 26. Explotación exitosa, sin seguridad predefinida (aplicación o sistema) en EMET.

Se procede ahora a configurar las técnicas de mitigación a nivel de aplicación en EMET, añadiendo la aplicación *"bof.exe"* a la lista de aplicaciones protegidas, como se muestra en la Figura 27. Nótese que todas las técnicas están activas.

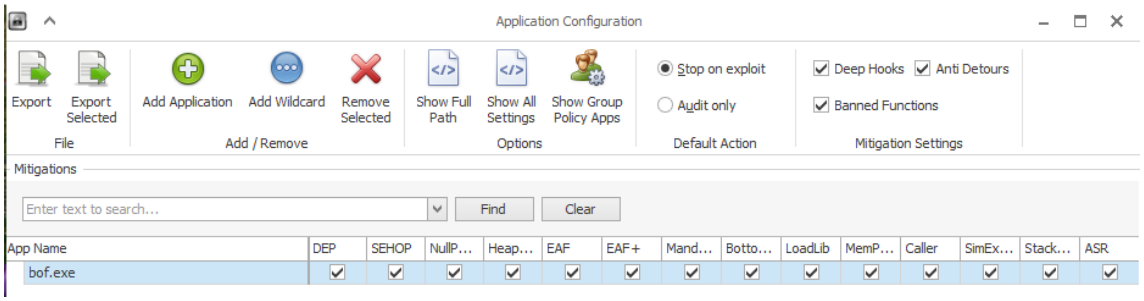


Figura 27. Configuración de técnicas de mitigación para "bof.exe".

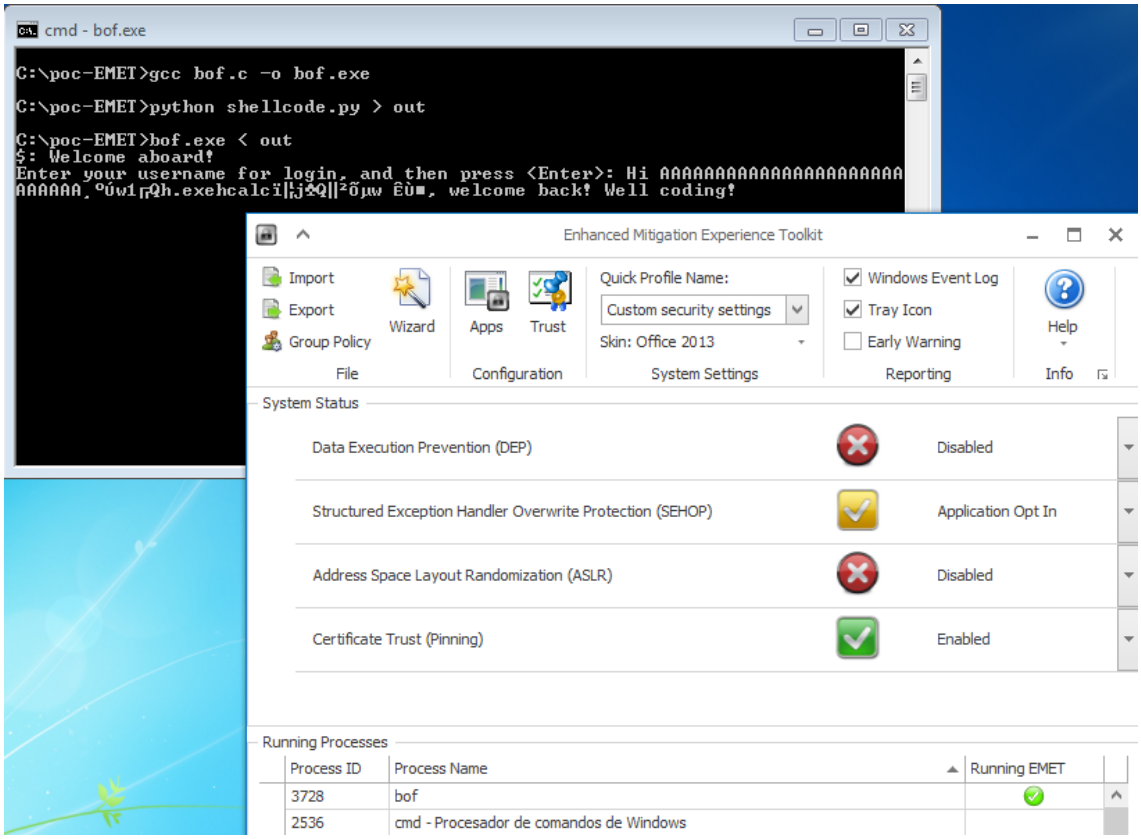


Figura 28. Aplicación "bof.exe" protegida con EMET.

En la Figura 28 se muestra que la explotación de la aplicación, estando protegida por EMET a nivel de aplicación, no funciona. De todas las técnicas de mitigación, en concreto, esta explotación necesita que la protección DEP no esté activa a nivel de aplicación para ser exitosa. De hecho, incluso usando técnicas de explotación para sobrepasar DEP estas técnicas no funcionan (es decir, se consigue evitar la explotación mediante EMET). Cabe decir que un aspecto importante es que **no existe notificación alguna al usuario**, ni similar, sobre que se ha evitado un intento de explotación.

Se procede a realizar ahora pruebas teniendo en cuenta únicamente las técnicas de mitigación a nivel de sistema.

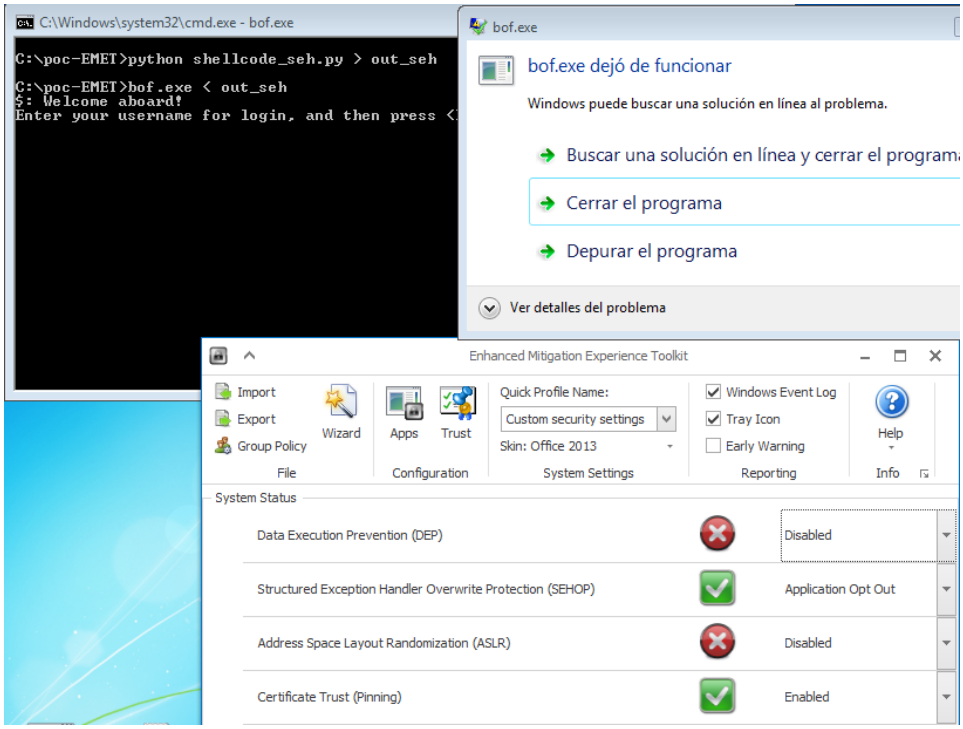


Figura 29. Activación, a nivel de sistema, de SEHOP en EMET.

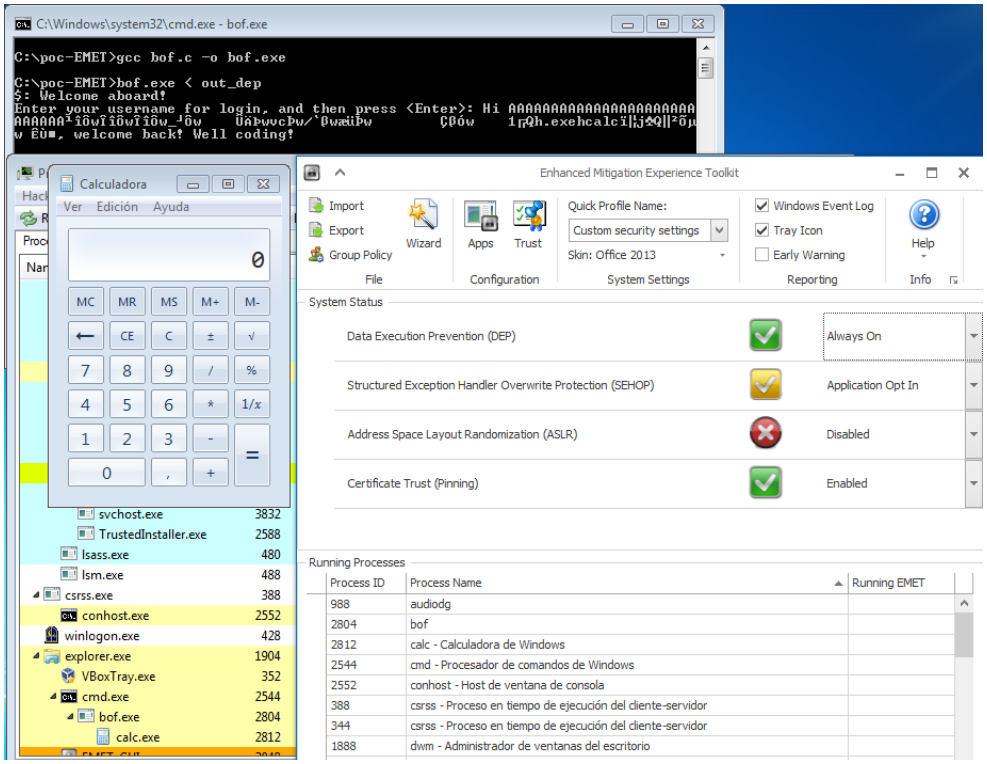


Figura 30. Activación, a nivel de sistema, de DEP en EMET.

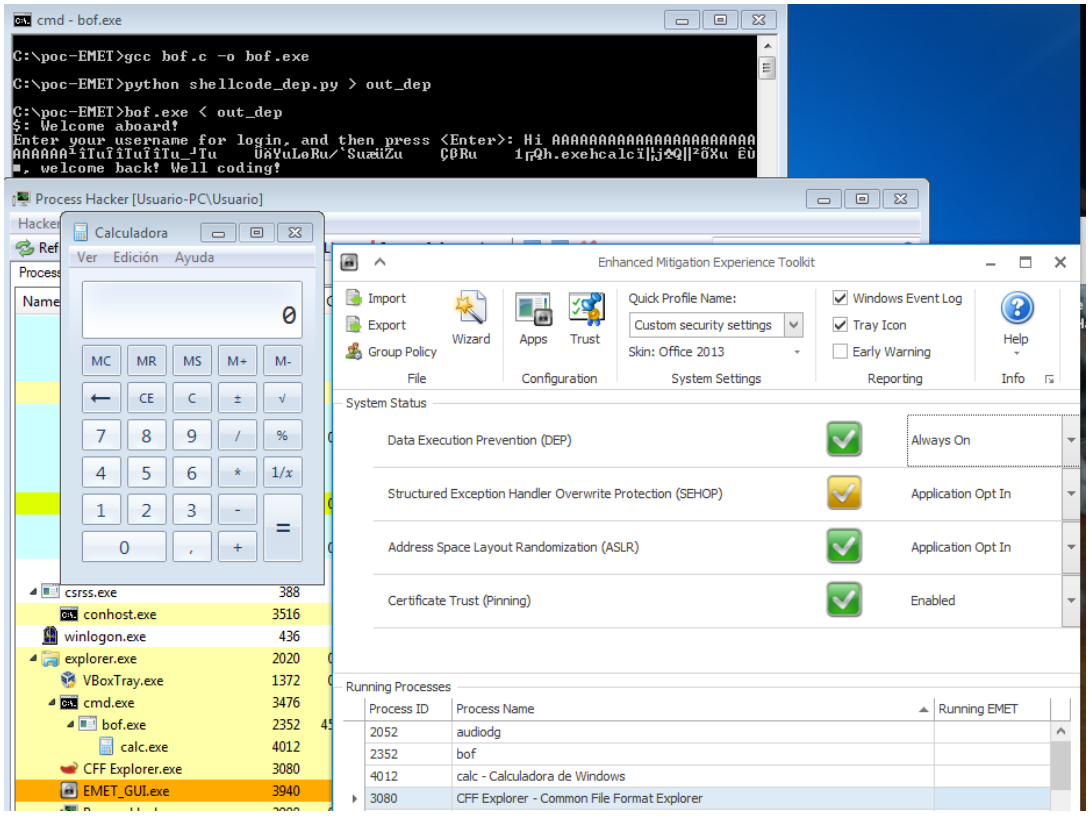


Figura 31. Activación, a nivel de sistema, de ASLR (con DEP también activado) en EMET.

La Figura 29 muestra la activación de SEHOP a nivel de sistema. Se ha modificado la shellcode anterior para conseguir que se explote a través de los manejadores de excepciones. Sin embargo, al tener activada la protección SEHOP, la explotación no funciona correctamente, como se muestra en la citada imagen. A pesar de que no se conocen hasta ahora ningún exploit funcional que rompa esta protección, sí que se ha estudiado el proceso de explotación de esta protección demostrando que es posible evitarla [3].

En la Figura 30, se muestra la activación de DEP a nivel de sistema. Como se observa, la explotación no se evita, logrando así la explotación de la vulnerabilidad. Para este caso, de nuevo, se ha modificado el exploit inicial para, mediante técnicas ROP, realice una desactivación de DEP sobre el propio proceso vulnerable. Del mismo modo, al habilitar la técnica de mitigación de ASLR a nivel de sistema, la explotación de la vulnerabilidad es exitosa, como se muestra en la Figura 31.

Nótese que ASLR únicamente aleatoriza las direcciones base de los módulos ejecutables, con lo que si existe una filtración de información (e.g., la dirección base del módulo “kernel32.dll”), un atacante sería capaz de preparar un exploit de manera exitosa. En este ejemplo, se han modificado las direcciones usadas por la shellcode para que el exploit funcione correctamente, asumiendo que el atacante era capaz de obtener las direcciones de memoria correctas.

Dado que se puede explotar esta vulnerabilidad sin hacer uso de los controladores de excepciones, se podría explotar de manera correcta la aplicación incluso con todas las protecciones de EMET a nivel de sistema activas (con ninguna protección a nivel de aplicación). Esta prueba se muestra en la Figura 32.

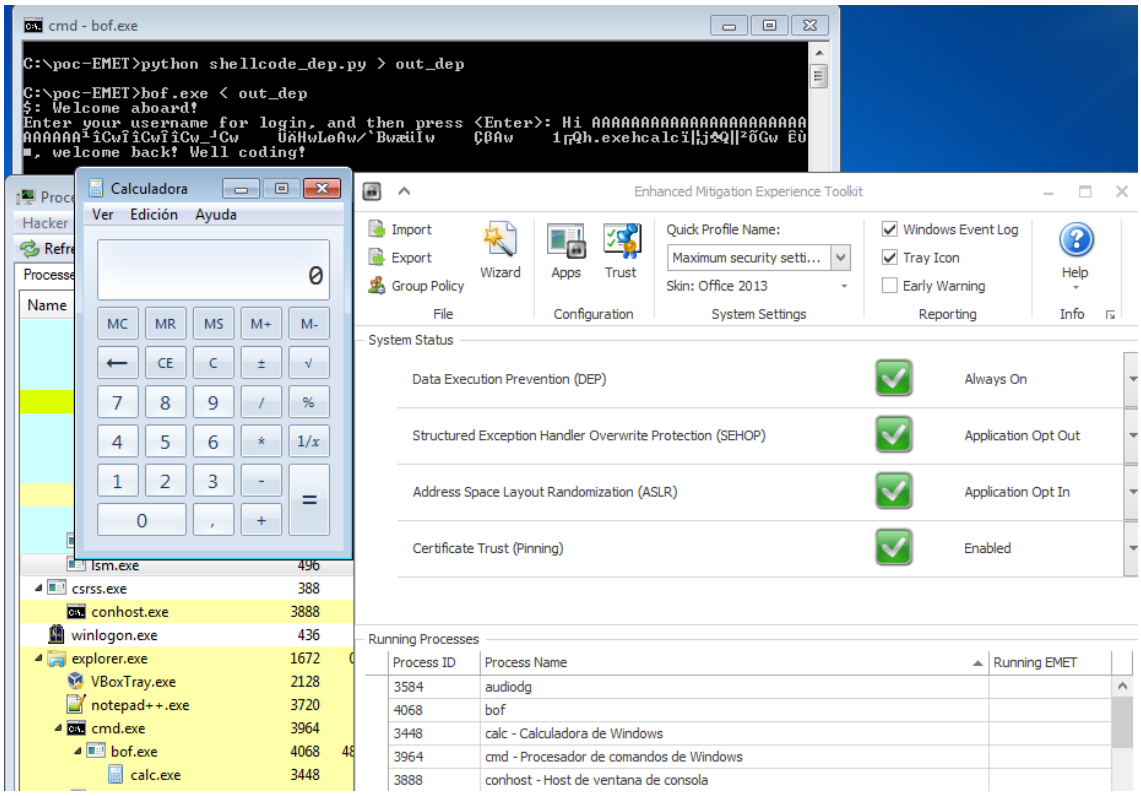


Figura 32. Activación, a nivel de sistema, de todas las protecciones en EMET.

Como se concluye, la seguridad aportada por EMET a nivel de sistema, en particular con las mitigaciones de ASLR y DEP, puede verse comprometida bajo ciertas condiciones de explotación (e.g., que el atacante obtenga información previa necesaria para la explotación por una filtración de memoria). Sin embargo, las técnicas de mitigación a nivel de aplicaciones introducidas por EMET realizan bien su trabajo, evitando la explotación de vulnerabilidades. Como se ha comprobado anteriormente, configurando la protección DEP a nivel de sistema se ha podido explotar de manera correcta (Figura 30), mientras que con la protección de DEP introducida a nivel de aplicación esto no ha sido posible (Figura 28).

En resumen, **se recomienda no sólo realizar la configuración a nivel de sistema sino también a nivel de aplicación**, para asegurarse de que en ningún caso un intento de explotación según los modelos actuales de explotación tenga éxito.

5. Conclusiones y recomendaciones finales

La herramienta Enhanced Mitigation Experience Framework (EMET) desarrollada por Microsoft permite configurar de manera centralizada y de una manera intuitiva todas las posibles técnicas de mitigación de explotación de vulnerabilidades que incorporan las diferentes versiones de los sistemas operativos de Microsoft Windows. Esta guía repasa la herramienta EMET, explicando las técnicas de mitigación que ofrece, así como su configuración. Además, se han realizado pequeñas pruebas de concepto para comprobar que efectivamente la herramienta EMET realiza su función.

EMET permite configurar no sólo las técnicas de mitigiación a aplicar a nivel de sistema (ASLR, DEP o SEHOP), si no también especificar por aplicación qué técnicas se desean aplicar (ASLR, DEP, SEHOP, anti-ROP, y heapspray allocations, entre otras). Esto permite poder tener más protegidas aquellas aplicaciones que son frecuentemente más explotadas por ataques, como por ejemplo Adobe Reader, Adobe Flash, Microsoft Office u Oracle Java. EMET también incorpora mecanismos de control sobre cadenas de certificados para asegurar que la conexión con páginas web cifradas es correcta y segura (protege de ataques tipo man-in-the-middle). Una desventaja de esta herramienta es que a pesar de ofrecer multitud de protecciones para las aplicaciones, puede que estas protecciones no sean compatibles con ciertas aplicaciones, con lo que será necesario deshabilitarlas de manera puntual a pesar del riesgo de seguridad que esto conlleva.

Sin embargo, EMET se ha visto como una buena herramienta para intentar proteger tanto los ordenadores personales como servidores en entornos corporativos, pudiendo además definir políticas de grupo para este tipo de entornos profesionales. Por último, cabe también destacar que EMET no es una solución 100% efectiva, debido a que las técnicas de explotación de vulnerabilidades continuamente evolucionan, por lo que habrá que mantener una actualización de la herramienta para conseguir detectar todos los ataques de explotación conocidos hasta la fecha (al menos, durante el período de vida de la aplicación).

Se recomienda así hacer uso de esta herramienta para garantizar al máximo la seguridad de los ordenadores en entornos corporativos y a nivel de usuario final. Previamente a su implantación en todo el entorno corporativo, **se recomienda estudiar qué las aplicaciones se usan y comprobar que todas las protecciones que habilita EMET son compatibles en ellas.** En caso de que alguna aplicación presente incompatiblidad con alguna mitigación, se tendrá que deshabilitar dicha técnica de mitigación para garantizar el correcto funcionamiento de la aplicación.

Por último, como se ha mostrado en la sección anterior se recomienda encarecidamente la activación de las técnicas de mitigación no sólo a nivel de sistema

si no también a nivel de aplicación, para dificultar al máximo las posibilidades de éxito de un atacante.

ANEXO A. REFERENCIAS

[1] **The Enhanced Mitigation Experience Toolkit**

Enlace web

http://support.microsoft.com/kb/2458544

[2] **Compatibility update available for Windows 8 and Wind. Server 2012**

Enlace web

http://support.microsoft.com/kb/2790907

[3] **Bypassing SEHOP**

S. Le Berre, D. Cauquil (artículo)

http://www.sysdream.com/sites/default/files/sehop_en.pdf

[4] **Enhanced Mitigation Experience Toolkit 5.52**

Enlace web

https://www.microsoft.com/en-us/download/details.aspx?id=54264

[5] **Enhanced Mitigation Experience Toolkit 5.52: User Guide**

Enlace web

https://www.microsoft.com/en-us/download/details.aspx?id=53355

[6] **EMET 4.0's Certificate Trust Feature**

Security Research & Defense blog (enlace web)

http://blogs.technet.com/b/srd/archive/2013/05/08/emet-4-0-s-certificate-trust-feature.aspx

[7] **Announcing EMET 5.0 Technical Preview**

Enlace web

https://blogs.technet.microsoft.com/srd/2014/02/25/announcing-emet-5-0-technical-preview/

[8] **Control Flow Guard**

Enlace web

https://msdn.microsoft.com/enus/library/windows/desktop/mt637065(v=vs.85).aspx

[9] **Informe de Amenazas CCN-CERT IA-01/16: MEDIDAS DE SEGURIDAD CONTRA RANSOMWARE**

Enlace web

https://www.ccn-cert.cni.es/seguridad-al-dia/comunicados-ccn-cert/3457-medidas-de-seguridad-contra-el-ransomware.html

[10] **Advanced Exploit Techniques Attacking the IE Script Engine**

Enlace web

https://blog.fortinet.com/2014/06/16/advanced-exploit-techniques-attacking-the-ie-script-engine

[11] **Understanding Enhanced Protected Mode**

Enlace web

https://blogs.msdn.microsoft.com/ieinternals/2012/03/23/understanding-enhanced-protected-mode/

[12] **Block untrusted fonts in an enterprise**
Enlace web
https://technet.microsoft.com/en-us/itpro/windows/keep-secure/block-untrusted-fonts-in-enterprise