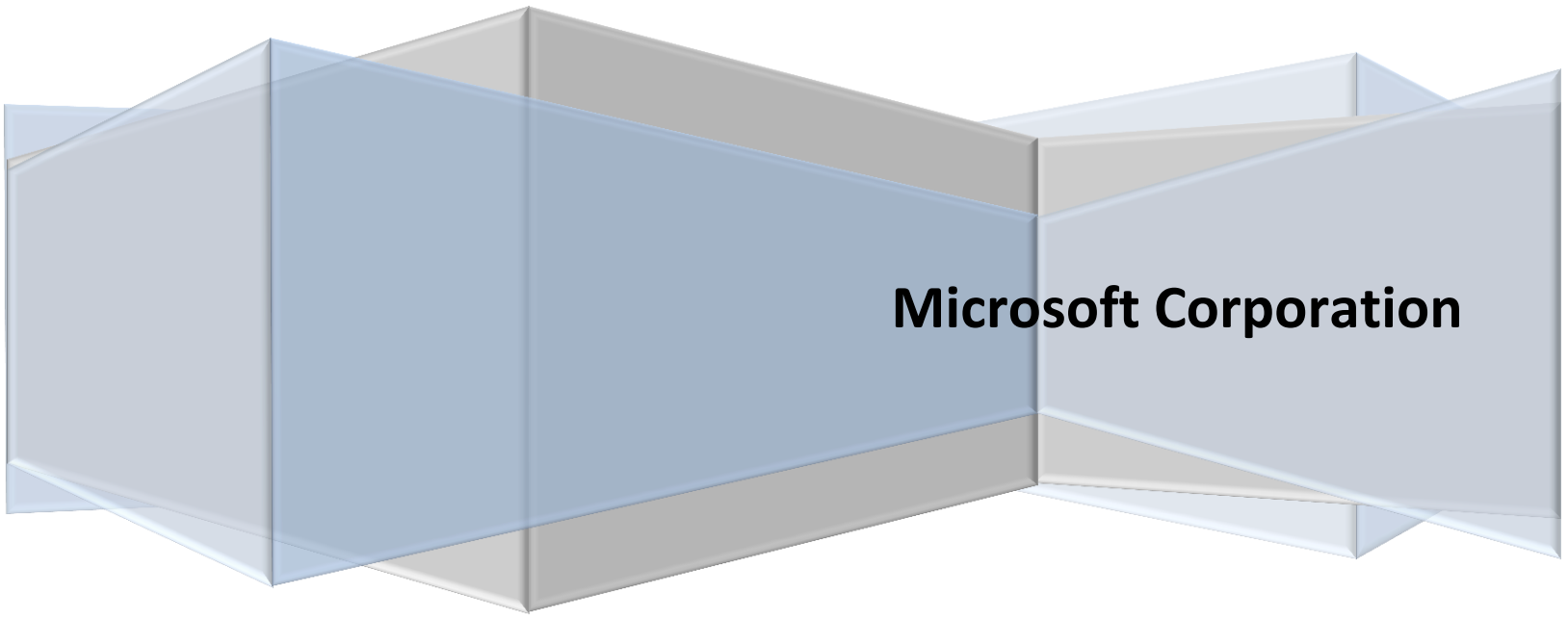
# Enhanced Mitigation Experience Toolkit v3.0

## User Guide



Microsoft Corporation

# Table of Contents

# 1. Introduction

The Enhanced Mitigation Experience Toolkit (EMET) is designed to help prevent hackers from gaining access to your system.

Software vulnerabilities and exploits have become an everyday part of life. Virtually every product has to deal with them and consequently, users are faced with a stream of security updates. For users who get attacked before the latest updates have been applied or who get attacked before an update is even available in cases such as 0day attacks, the results can be devastating: malware, loss of PII, loss of business data etc.

Security mitigation technologies are designed to make it impossible or more difficult for an attacker to exploit vulnerabilities in a given piece of software. EMET allows users to leverage these technologies on their system and provides several unique benefits:

1. **No source code needed:** Until now, several of the available mitigations (such as Data Execution Prevention) have required for an application to be manually opted in and recompiled. EMET changes this by allowing a user to opt-in applications without recompilation. This is especially useful for deploying mitigations on software that was written before the mitigations were available, and when source code is not available.

2. **Highly configurable:** EMET provides a higher degree of granularity by allowing mitigations to be individually applied on a per process basis. There is no need to enable an entire product or suite of applications. This is helpful in situations where a process is not compatible with a particular mitigation technology. When that happens, a user can simply turn that mitigation off for that process.

3. **Helps harden legacy applications:** It's not uncommon to have a hard dependency on old legacy software that cannot easily be rewritten and needs to be phased out slowly. Unfortunately, this can easily pose a security risk as legacy software is notorious for having security vulnerabilities. While the real solution to this is migrating away from the legacy software, EMET can help manage the risk while this is occurring by making it harder to hackers to exploit vulnerabilities in the legacy software.

4. **Ease of use:** The policy for system wide mitigations can be seen and configured with EMET's graphical user interface, the command line tool or via Group Policy. There is no need to locate up and decipher registry keys or run platform dependent utilities. With EMET you can adjust settings with a consistent interface regardless of the underlying platform.

5. **Ongoing improvement:** EMET is a living tool designed to be updated as new mitigation technologies become available. This provides a chance for users to try out and benefit from cutting edge mitigations. The release cycle for EMET is also not tied to any product. EMET updates can be made dynamically as soon as new mitigations are ready.

   The toolkit includes several pseudo mitigation technologies aimed at disrupting current exploit techniques. These pseudo mitigations are not robust enough to stop future exploit techniques, but can help prevent users from being compromised by many of the exploits currently in use. The mitigations are also designed so that they can be easily updated as attackers start using new exploit techniques.

## 1.1. Capabilities

EMET 3.0 allows users to both configure the system policy for mitigations as well as to configure mitigations on a per executable basis.

The first option allows the user to set the defaults for system supported mitigaitons; for instance choosing whether a mitigation should be enabled for all processes, enabled for only those that chose to opt-in, disabled entirely etc.

The second option allows the user to enable an EMET supported mitigation on an arbitrary executable. Any one of the supported mitigations can independently be turned on and off for any executable residing on the system. Next time one of the configured executables runs, the specified mitigations will be applied to it. Combining these two options gives the user a high degree of control over the mitigations available on a system and how they get used.

EMET doesn't run as a service, or attach to an application like a debugger. Instead, behind the scenes, in order to enable mitigations for applications, EMET is leveraging a shim infrastructure in Windows called the Application Compatibility Framework. This is a highly optimized low level interface and as such, EMET presents no additional resource overhead on the protected applications. A high-level overview of this infrastructure and the toolkit that accompanies it can be found in this blog post.

**NOTE: Before continuing, please be aware that some security mitigation technologies may break some applications. It is important to thoroughly test EMET in all target use scenarios before rolling it out to a production environment.**

## 1.2. EMET Provided Mitigations

The current version supports seven different mitigation technologies. In this section, we will outline the different mitigations and the protections they provide.

### Structure Exception Handler Overwrite Protection (SEHOP)

This protects against currently the most common technique for exploiting stack overflows in Windows. This mitigation has shipped with Windows since Windows Vista SP1. Recently with Windows 7, the ability to turn it on and off per process was added. With EMET, we provide the Windows 7 capabilities on any platform back though Windows XP. For more information, take a look at the SEHOP Overview and Windows 7 SEHOP Changes blog posts.

Without EMET in place an attacker can overwrite, with a controlled value, the handler pointer of an exception record on the stack. Once an exception happens, the OS will walk the exception record chain and call all the handlers on each exception record. Since the attacker controls one of the records, the OS will jump to wherever the attacker wants, giving the attacker control the flow of execution. See figure 1 for an illustration of this.

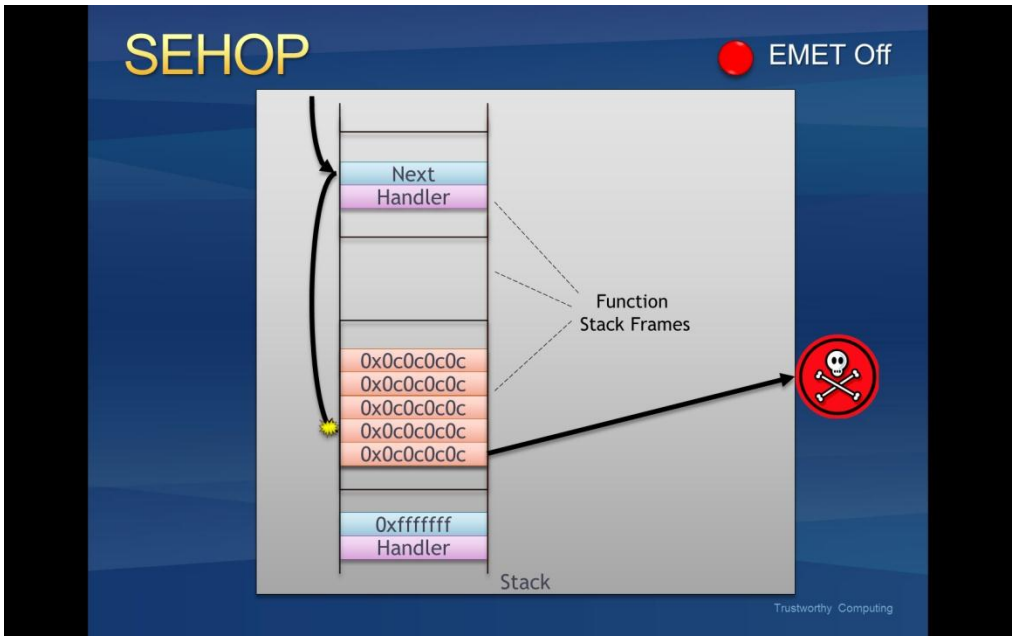



**Figure 1: An exception handler hijack**

With EMET in place, before the OS calls any exception handlers, it will validate the exception record chain. This involves checking if the final exception contains a predefined one. If the chain is corrupted, EMET will terminate the process without calling any of the handlers. Figure 2 illustrates what this looks like.

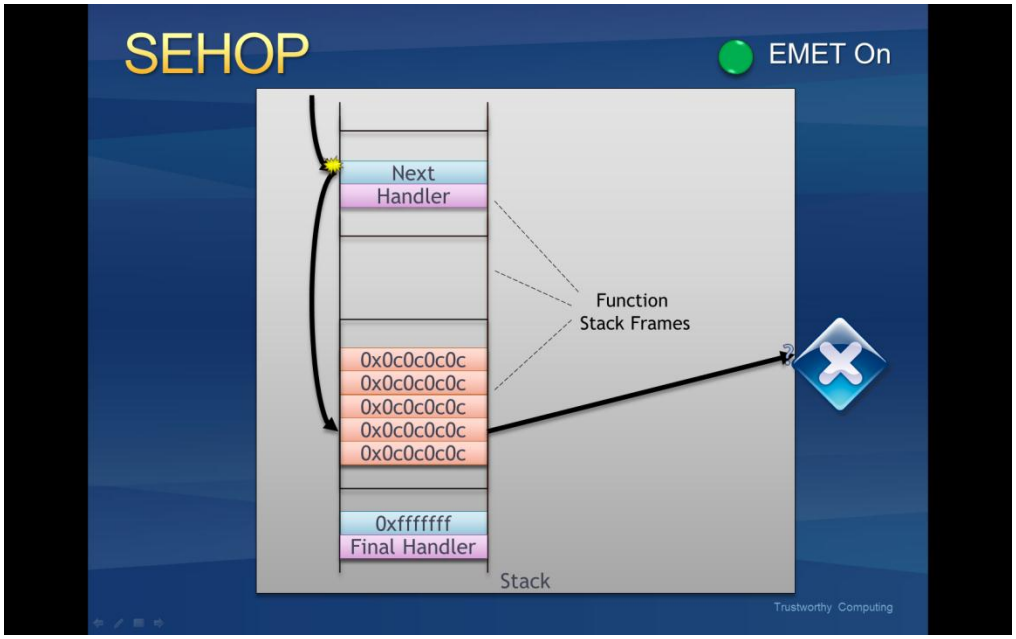



**Figure 2: EMET stopping an exception handler hijack**

## Dynamic Data Execution Prevention (DEP)

DEP has been available since Windows XP. However, current configuration options don't allow applications to be opted in on an individual basis unless they are compiled with a special flag. EMET allows applications compiled without that flag to also be opted. For more information on what DEP is and how it works, take a look at Part 1 and Part 2 of our two-part SRD blog post on it.

Without EMET in place, an attacker can attempt to exploit a vulnerability by jumping to shellcode at a memory location where attacker controlled data resides such as the heap or stack. Since these regions are marked as executable the malicious code will be able to run.

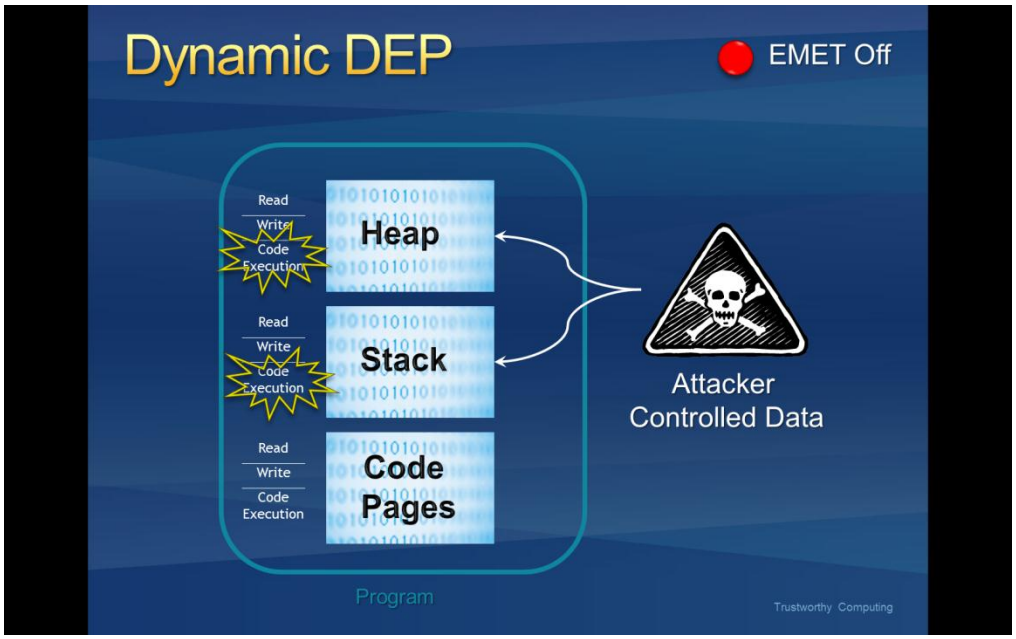



**Figure 3: Running shellcode from attacker controlled locations**

Turning EMET on will enable DEP for a process. Once this happens, the stack and heap will be marked as non-executable and any attempt to execute malicious code from these regions will be denied at the processor level.

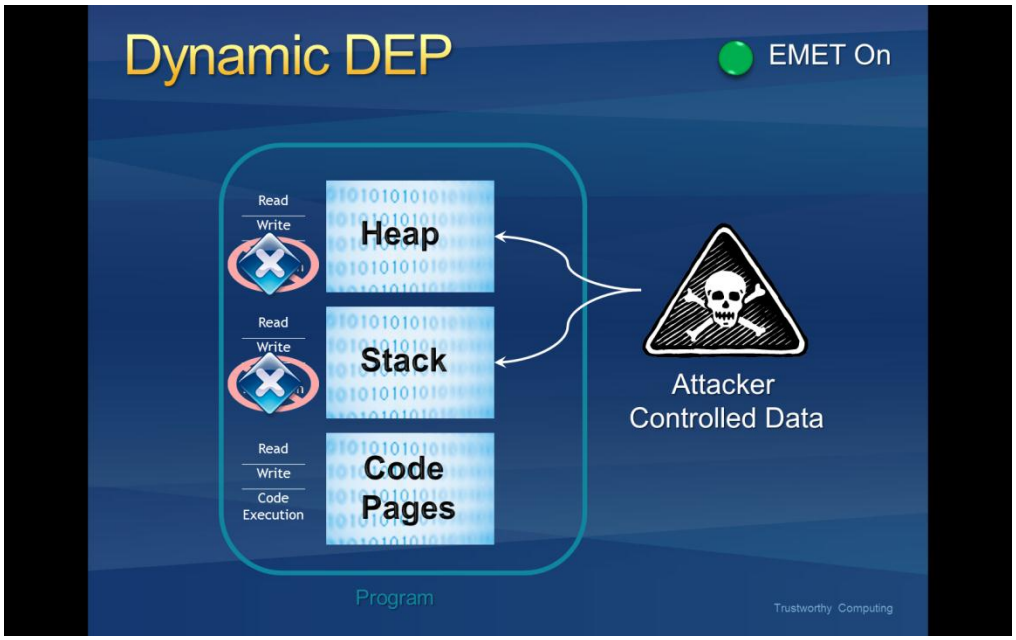



**Figure 4: DEP blocking shellcode from running**

### Heapspray Allocations

When an exploit runs, it often cannot be sure of the address where its shellcode resides and must guess when taking control of the instruction pointer. To increase the odds of success, most exploits now use heapspray techniques to place copies of their shellcode at as many memory locations as possible. Figure 5 shows an illustration of what this looks like in a victim process.

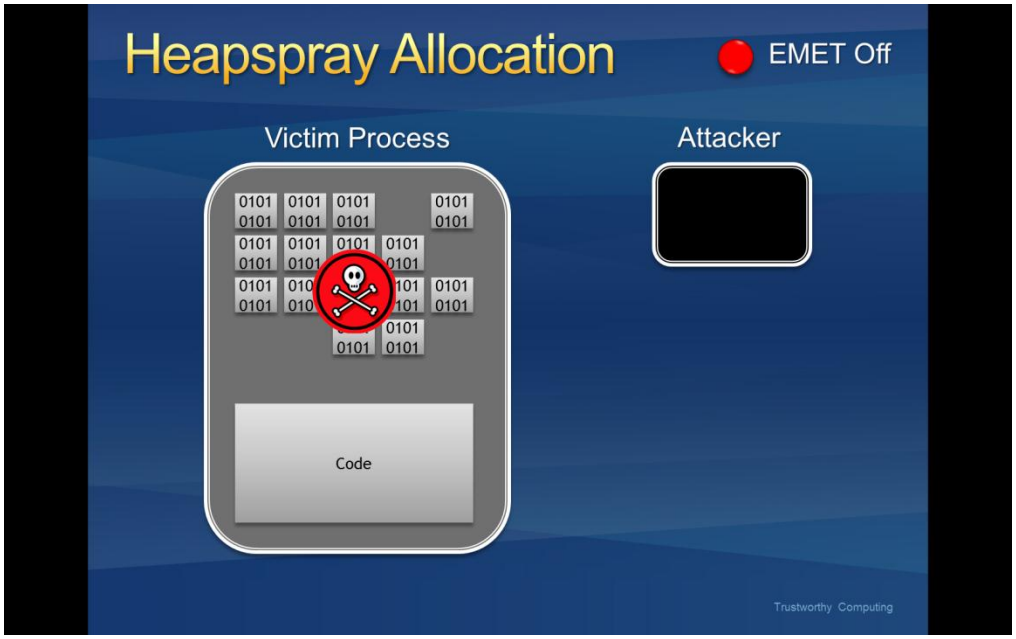



**Figure 5: Using heapspray in an exploit**

With EMET in place some commonly used pages are pre-allocated. Exploits that rely on controlling these pages (and then jumping into them) will fail.

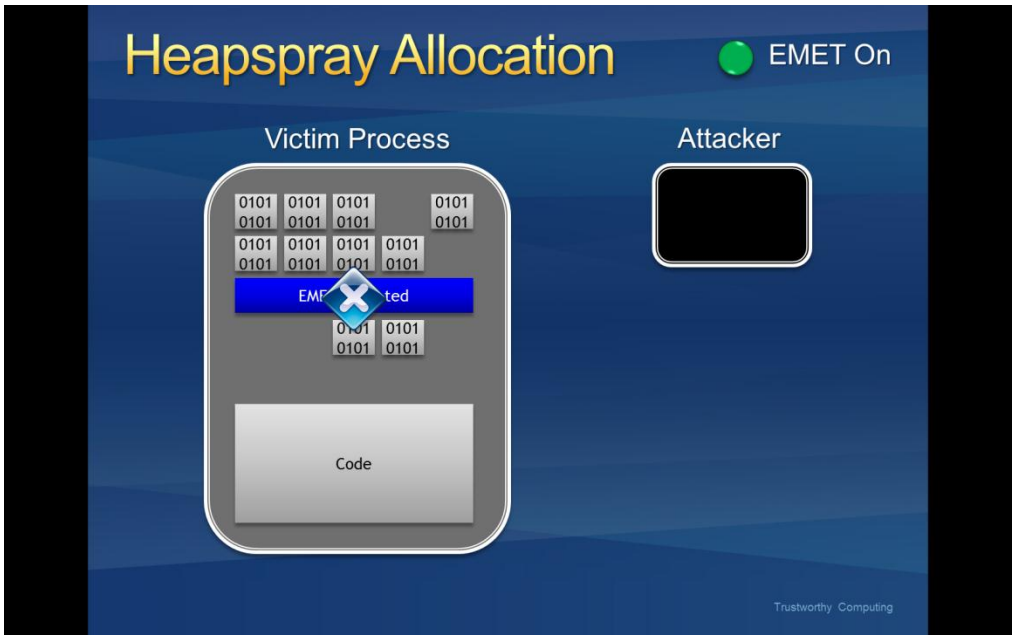



**Figure 6: Blocking an attack that uses heapspray**

*Please note this is a pseudo mitigation designed to break current exploit techniques. It is not designed to break future exploits as well. As exploit techniques continue to evolve, so will EMET.*

### Null page allocation

This is similar technology to the heap spray allocation, but designed to prevent potential null dereference issues in user mode. Currently there are no known ways to exploit them and thus this is a defense in depth mitigation technology.

### Mandatory Address Space Layout Randomization (ASLR)

ASLR randomizes the addresses where modules are loaded to help prevent an attacker from leveraging data at predictable locations. The problem with this is that all modules have to use a compile time flag to opt into this.

Without EMET in place, attackers can take advantage of a predictable mapping of those dlls and could use them in order to bypass DEP though a known technique called return oriented programming (ROP).

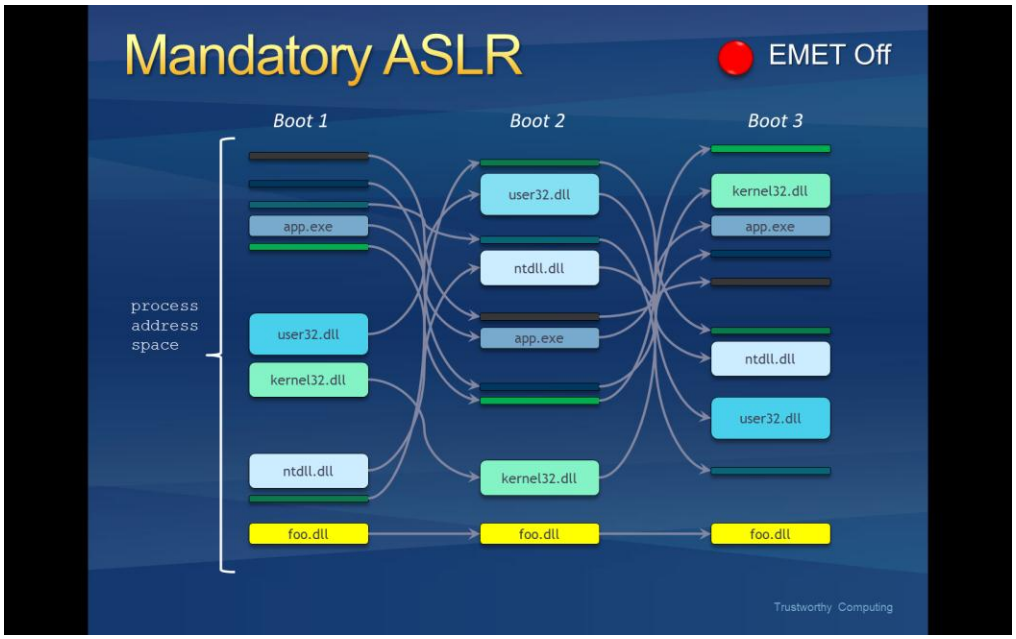



**Figure 7: A module being loaded at a predictable location**

With EMET in place, we force modules to be loaded at randomized addresses for a target process regardless of the flags it was compiled with. Exploits using ROP and relying on predictable mappings will fail.

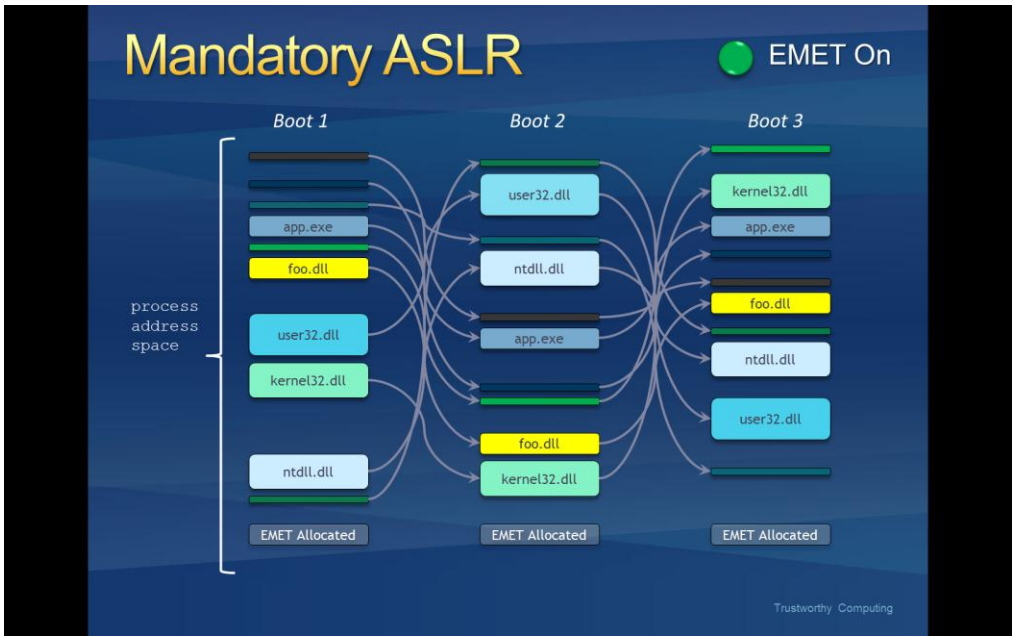



**Figure 8: A module being forced to load at a random address**

### Export Address Table Access Filtering (EAF)

In order to do something "useful", shellcode generally needs to call Windows APIs. However, in order to call an API, shellcode must first find the address where that API has been loaded. To do this the vast majority of shellcode iterates through the export address table of all loaded modules, looking for modules that contain useful APIs. Typically this involves kernel32.dll or ntdll.dll. Once an interesting module has been found, the shellcode can then figure out the address where an API in that module resides.

This mitigation filters accesses to the Export Address Table (EAT), allowing or disallowing the read/write access based on the calling code. With EMET in place, most of today's shellcode will be blocked when it tries to lookup the APIs needed for its payload.

*Please note this is a pseudo mitigation designed to break current exploit techniques. It is not designed to break future exploits as well. As exploit techniques continue to evolve, so will EMET.*

### Bottom-up randomization

This mitigation randomizes (8 bits of entropy) the base address of bottom-up allocations (including heaps, stacks, and other memory allocations) once EMET has enabled this mitigation but not for previous allocations.

## 1.3. Supported Operating Systems

EMET 3.0 supports the following operating systems and service pack levels:

**Client Operating Systems**

- Windows XP service pack 3 and above
- Windows Vista service pack 1 and above
- Windows 7 all service packs

**Server Operation Systems**

- Windows Server 2003 service pack 1 and above
- Windows Server 2008 all service packs
- Windows Server 2008 R2 all service packs

Please note that not all mitigations are supported on each operating system.

| | Mitigation | XP | Server 2003 | Vista | Server 2008 | Win7 | Server 2008 R2 |
|---|---|---|---|---|---|---|---|
| **System Settings** | DEP | Y | Y | Y | Y | Y | Y |
| | SEHOP | N | N | Y | Y | Y | Y |
| | ASLR | N | N | Y | Y | Y | Y |
| **Application Settings** | DEP | Y | Y | Y | Y | Y | Y |
| | SEHOP | Y | Y | Y | Y | Y | Y |
| | NULL Page | Y | Y | Y | Y | Y | Y |
| | Heap Spray | Y | Y | Y | Y | Y | Y |
| | Mandatory ASLR | N | N | Y | Y | Y | Y |
| | EAF | Y | Y | Y | Y | Y | Y |
| | Bottom-up | Y | Y | Y | Y | Y | Y |

Additionally, on 64 bit systems, some application specific mitigations are only applicable when running on 32 bit processes. For details, refer to the following table.

| | Mitigation | 32 bit Processes | 64 bit Processes |
|---|---|---|---|
| **Application Settings** | DEP | Supported by EMET | Already Mandatory without EMET |
| | SEHOP | Supported by EMET | Not Applicable |
| | NULL Page | Supported by EMET | Supported by EMET |
| | Heap Spray | Supported by EMET | Supported by EMET |
| | Mandatory ASLR | Supported by EMET | Supported by EMET |
| | EAF | Supported by EMET | Supported by EMET |
| | Bottom-up | Supported by EMET | Supported by EMET |

# 2. Configuring EMET

Without configuring EMET, there is no benefit from having it installed. To configure EMET, settings have to be specified that:

- Describe which system mitigations should be enabled.
- Describe which applications should be protected with which mitigations.

Both system and application mitigations can be configured via the EMET Graphical User Interface or via the EMET Command Line Tool. Refer to Sections 2.2 and 2.3 of this guide for further instructions on how to use those tools to achieve this.

You can also use Group Policy to configure system and application mitigations for EMET. Group Policy support is explained in Section 3.2.

Another option for configuring EMET is using Protection Profiles.

EMET application mitigations are saved in the registry at HKLM\SOFTWARE\Microsoft\EMET.

## 2.1. EMET Protection Profiles

EMET 3.0 comes with three default Protection Profiles. Protection Profiles are XML files that contain pre-configured EMET settings for common Microsoft and 3rd party applications. Under EMET's installation directory, these files are in the Deployment\Protection Profiles folder. Users are encouraged to enable them as-is, to modify them, or to create new protection profiles based on them.

The three profiles that ship with EMET are

- Internet Explorer.xml: Enables mitigations for supported versions of Microsoft Internet Explorer.
- Office Software.xml: Enables mitigations for supported versions of Microsoft Internet Explorer, applications that are part of Microsoft Office suite, Adobe Acrobat 8-10 and Adobe Acrobat Reader 8-10.
- All.xml: Enable mitigations for common home and enterprise applications, including Microsoft Internet Explorer and Microsoft Office.

Let's look at some rules from All.xml.

```
<Product Name="Internet Explorer">
<Version Path="*\Internet Explorer\iexplore.exe"/>
```

```
</Product>
```

The rule above is simple. It tells EMET to protect Internet Explorer with the default mitigation settings. By default, all seven mitigations are enabled for all applications in a protection profile. This can be changed by editing the DefaultConfig node in the profile file. In short, this rule configures EMET to enable all the mitigations for Internet Explorer.

```
<Product Name="Windows Media player">
<Version Path="*\Windows Media Player\wmplayer.exe">
<Mitigation Enabled="false" Name="MandatoryASLR"/>
</Version>
</Product>
```

With this rule, we enable all mitigations for Windows Media Player, except Mandatory ASLR. Since there is a known compatibility issue, like in this case, we disable that mitigation for that application in the profile.

Another important information is the Path. We have for instance "*\Windows Media Player\wmplayer.exe". The path is what EMET uses to register its mitigations for an application. It has to match the target application's path for the mitigations to be effective.

To specify the Path, you can use the full path name to the application. You can also use wildcards, namely * or ?. Another option is to just use the executable name without the path, such as wmplayer.exe.

Please note that wildcards are only accepted in the path portion, and are not valid in the executable image name itself. For instance "wmplayer.exe" or "*\wmplayer.exe" are valid paths, while "*player.exe" or "*wmplayer.exe" aren't. This is due to a limitation of the Application Compatibility Framework in Windows that EMET relies on.

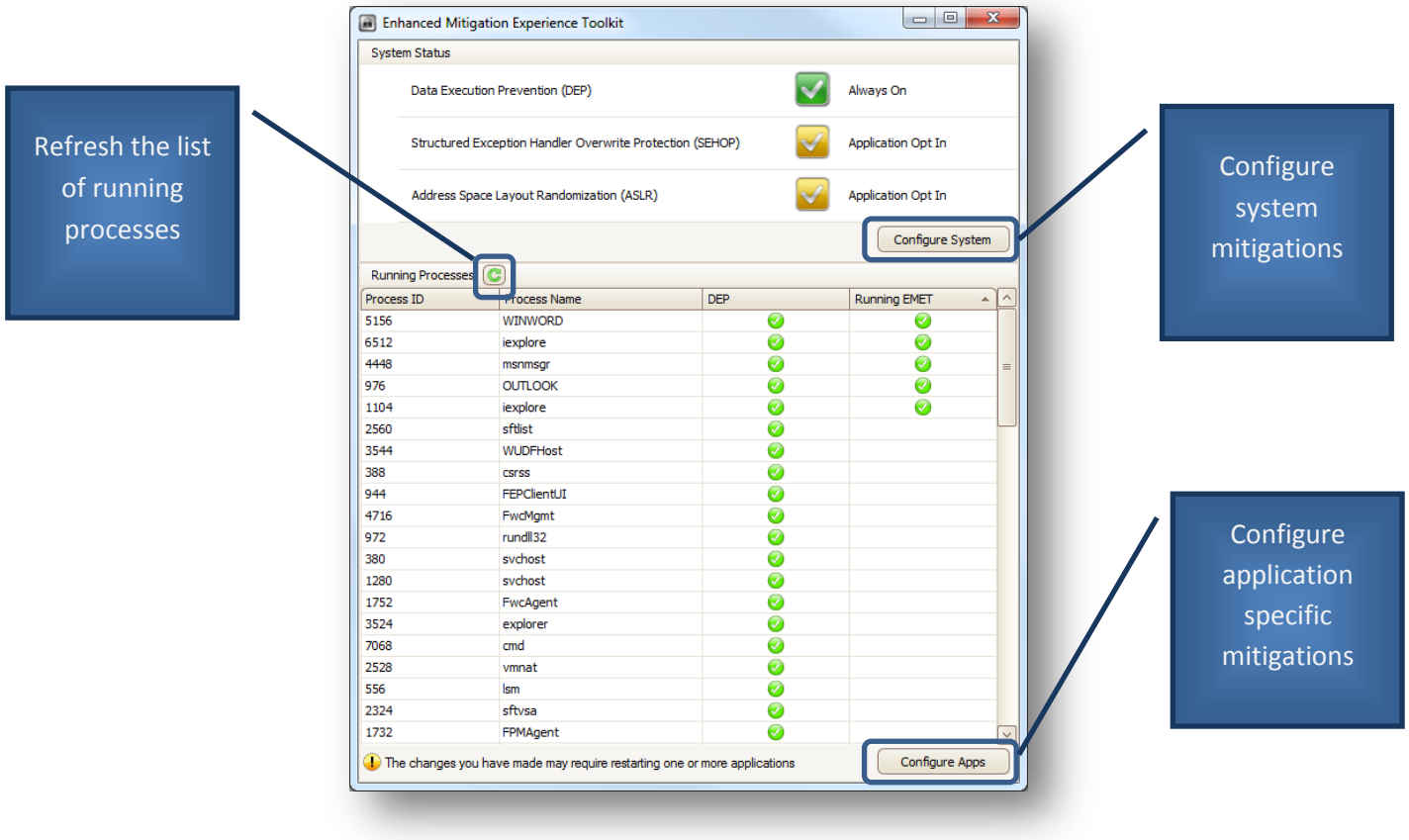
The protection files are well commented themselves. Reading them is a great way to learn more about this feature.

Protection Profiles can be enabled via the EMET Graphical User Interface, the EMET Command Line Tool or via Group Policy.

## 2.2. EMET Graphical User Interface

One method of interacting with EMET is through the graphical user interface (GUI). You can launch this program through the start menu icon created during the EMET installation. In this section we will walk you through the various windows of this interface.

When EMET is launched the following GUI is be presented to the user.

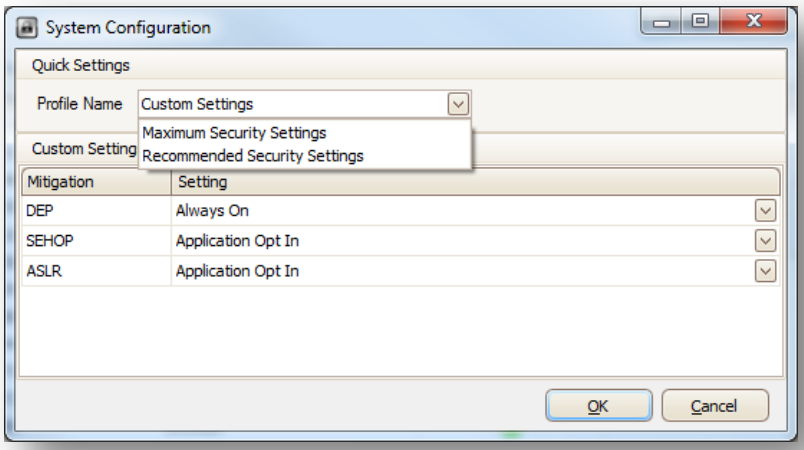



Through this initial window a user will be able to display the status of the different system mitigations and whether or not any of the running processes have been opted-in to EMET. Please note the list of running processes is only updated every 30 seconds. To get updated information on demand, click the button next to “Running Processes”.

A user can also click on either of the two buttons to configure system mitiations or to opt-in an application to the EMET supported mitigations.

### 2.2.1. Configuring System Mitigations

Users can configure system wide mitigations in two different ways. Either they can select one of the two system mitigation profiles (“Maximum Security Settings” and “Recommended Security Settings”) or set the mitigation configuration individually.




Please note some configuration changes will require rebooting the operating system. EMET's GUI provides notification of this when it happens.

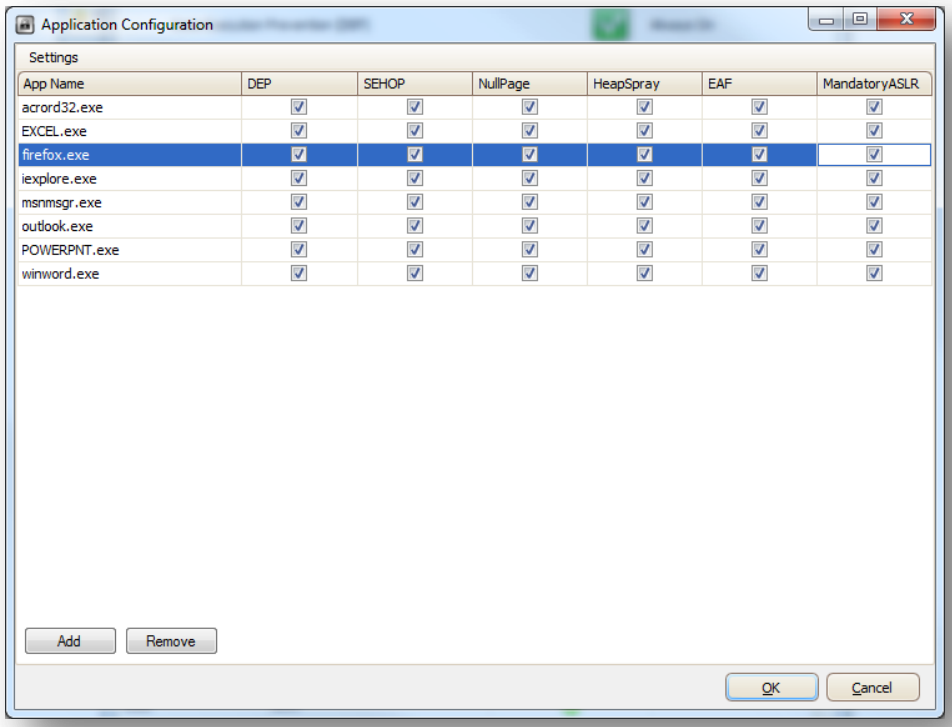
You will notice that the list of available system mitigations vary between different versions of Windows. This is because some system mitigations are not available on every supported operating system. Refer to Section 1.3 for more information about mitigation support in different Windows versions.

### 2.2.2. Configuring Mitigations for Applications

Users will be able to configure specific applications to opt-in to the mitigations supported by EMET. Additionally, mitigations can be individually enabled or disabled on a per application basis.

For example, a user will be able configure iexplore.exe to opt-in to all EMET's mitigation and, at the same time, opt-in firefox.exe only for SEHOP and Mandatory ASLR.

Users will be able to Add and Remove applications from the list by clicking the corresponding buttons. When adding an application, a user will get prompted with the regular open file dialog and once having selected one it will get added to this list. Then, user will be able to configure it.




EMET will only be in place with the selected configuration after you click the Ok button and after you restart the newly added/configured application(s).

## 2.3. EMET Command Line Tool

An alternate way to configure EMET is to use EMET_Conf.exe. This command line utility can be found at the location where EMET is installed.

Below are configuration related options the EMET Command Line Tool supports.

**Add an application to EMET**

*EMET_Conf --set [--force] < path to executable> [(+|-)Mitigation …]*

<path to executable> can be the full path name to the application. You can also use wildcards, namely * or ?. Another option is to just use the executable name without the path, such as wmplayer.exe.

Please note that wildcards are only accepted in the path portion, and are not valid in the executable image name itself. For instance “wmplayer.exe” or “*\wmplayer.exe” are valid paths, while

“*player.exe” or “*wmplayer.exe” aren’t. This is due to a limitation of the Application Compatibility Framework in Windows that EMET relies on.

The --force option is used to configure EMET for an application that is not currently installed on a system.

Example usage:

EMET_Conf --set program.exe : enables all mitigations for program.exe.

EMET_Conf --set program.exe -DEP: enables all mitigations except DEP for program.exe.

**List which applications EMET has been enabled for**

*EMET_Conf --list*

Display all the application mitigation settings for EMET. Lines starting with “<” are settings managed via Group Policy.

**Remove an application from EMET**

*EMET_Conf --delete <path to executable>*

<path to executable> can be a full path, a path with wildcards or just the executable name. It should match the <path to executable> used to add the application to EMET.

**Remove all applications from EMET**

*EMET_Conf --delete_all*

This will remove all the EMET application mitigation settings. Please note that this does not remove application mitigation settings configured via Group Policy.

**Modify a system mitigation**

*EMET_Conf --system [--force] <SysMitigation=State> [SysMitigation=State …]*

The --force option is needed to set a mitigation to an unsafe state. For more information on this, refer to Section 5 about Advanced Options. By default unsafe options are not visible through either the command-line utility or the UI.

**Import/Export application settings from an xml file**

*EMET_Conf --import* *<xml file>*

Imports previously exported application settings. This command can also be used to import and enable a Protection Profile, e.g. EMET_Conf --import “Deployment\Protection Profiles\All.xml”

*EMET_Conf --export* *<xml file>*

Running the EMET Command Line Tool without any arguments will display a usage screen including all currently supported application specific mitigations as well as the supported system mitigations.

# 3. Deploying EMET

With EMET 3.0, enterprise users can take advantage of their existing management infrastructure with to deploy and configure EMET at a large scale. In this section, we talk about how to use System Center Configuration Manager and Group Policy to deploy and manage EMET across enterprise networks.

## 3.1. System Center Configuration Manager

EMET 3.0 is easily integrated into System Center Configuration Manager for deployment and configuration purposes.

### 3.1.1. Creating the Application to Deploy EMET to Clients

The first step in deploying EMET is to download the EMET 3.0 MSI. After you have the MSI, follow the steps below. In this example, we are going to reference building an application in Configuration Manager 2012, but the same thing could be accomplished with packages, programs, and advertisements using Configuration Manager 2007.

1. From Software Library | Application Management | Applications, choose to Create Application.
2. Keep the default type as Windows Installer (Native) and browse to the source UNC path for the EMET Setup MST file, which you downloaded previously (*).
3. The application details will be automatically derived from the MSI, along with MSI product code (on the Import Information page).
4. On the General Information page, you will be able to add any additional details for this application, and you'll see a pre-populated command next to Installation program, that has details on the MSI-based install of EMET. Edit the installation line to read: **msiexec /i "EMET Setup.msi" /qn /norestart**
5. Change install behavior to **Install for system**.
6. Complete the wizard.
7. From the application you just created, choose Deploy.
8. Browse to the collection you want to target.
9. On the content page, choose your distribution points.
10. On the deployment settings page, choose the intended install settings (most likely this will be required, unless you are just testing the deployment).
11. Configure the deployment scheduled, user experience, and alerts, then complete the wizard.
12. You are now in the process of deploying the EMET client silently to all targeted clients. You can monitor the deployment progress of this application in Monitoring | Deployments.

### 3.1.2. Creating the Package and Program to Configure EMET

Now that you have EMET deployed (or the deployment in progress), you will need to configure it for enhanced mitigation of your specified applications. Without configuring EMET, the base client does nothing standalone to offer enhanced application protection. Here we'll create a collection of clients reporting EMET client installed, and we'll target those with the configuration package.

#### Create the EMET Configuration Target Collection

1. From Assets and Compliance | Device Collections choose to Create Device Collection.
2. Name the Device Collection (Clients with EMET Installed), and choose the limiting collection.
3. On the membership rules page, click Add Rule, and choose a Query Rule.
4. Name the query, and choose Edit Query Statement.
5. In the criteria tab, click the yellow star.
6. In Criterion Properties, keep the type as Simple value, and choose select.
7. Choose Installed Applications as the attribute class.
8. Choose Display Name as the Attribute.
9. After clicking OK, click the Value button.
10. Choose EMET from the list of values. NOTE: At least one system will have to have reported its hardware inventory up post-EMET client install for this value to be populated. If it's not in the list, simply type the value in.
11. After completing the query rule, choose how often you want to evaluate this collection. We will be targeting EMET configuration to this collection, so evaluate it as often as you want clients with EMET newly installed, to have it configured. Also, keep in mind that this collection will only be populated when inventory information from clients (with EMET installed), is sent to the server. By default, inventory is sent every 7 days.

#### Create the EMET Configuration Package and Program

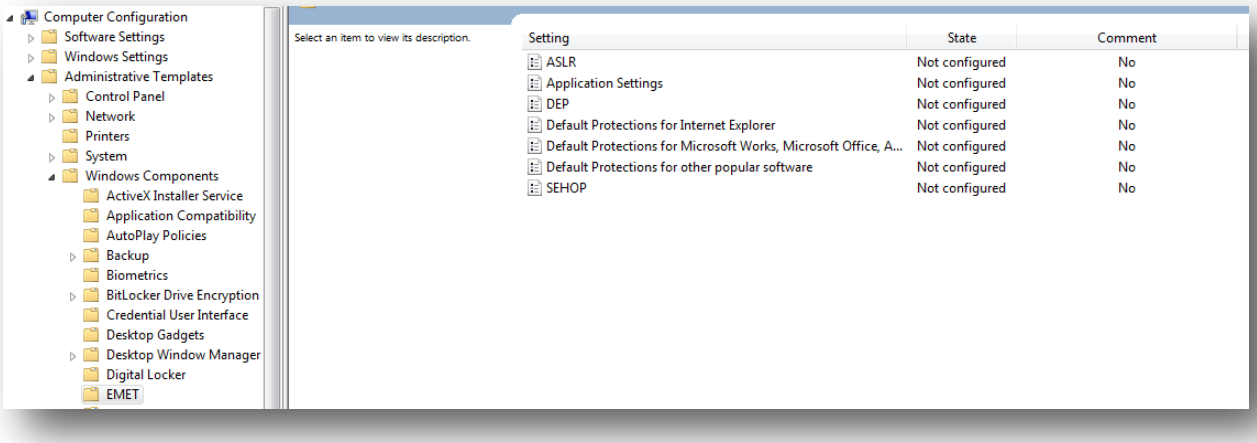
1. Place the following 4 files in a source directory that you will use as the source for the EMET configuration package. You can get these files from the source directory of the EMET client after you've installed the MSI on a client. NOTE: If you don't include all of these files, EMET configuration will not work.
    a. All.XML (from the source \program files (x86)\EMET\Deployment\Protection Profiles)
    b. EMET_Conf.exe (from the source \program files (x86)\EMET)
    c. EMET_notifier.exe (from the source \program files (x86)\EMET)
    d. MitigationInterface.dll (from the source \program files (x86)\EMET)
2. From Software Library | Packages choose to Create Package.

3. Name the package, and choose this package contains source files. Provide the path where you are sourcing the four files referenced in step 1.
4. Choose standard program.
5. Name the program, and set the command line to be EMET_Conf.exe --import All.xml. NOTE: This is just an example, using the "All" protection profile provided by the EMET team. You can modify this profile to your own preferences, or use one of the other protection profiles provided by EMET. You simply need to reference the file to be imported, and include it in your EMET configuration package.
6. Set the program to run hidden, and whether or not a user is logged on.
7. Complete the wizard.
8. After the package and program are complete, choose to deploy it.
9. Pick the collection we created earlier as the target collection, and complete the wizard with your desired settings.

(*) More information and the downloadable Configuration Manager packages can be found at the Configuration Manager Team Blog here.

## 3.2. Group Policy

EMET 3.0 comes with group policy support. When you install EMET, EMET.admx and EMET.adml files are also installed to the "Deployment\Group Policy Files" folder. These files can then be copied onto \Windows\PolicyDefinitions and \Windows\PolicyDefinitions\en-US folders respectively. Once this is done, EMET system and application mitigation settings can be configured via Group Policy.



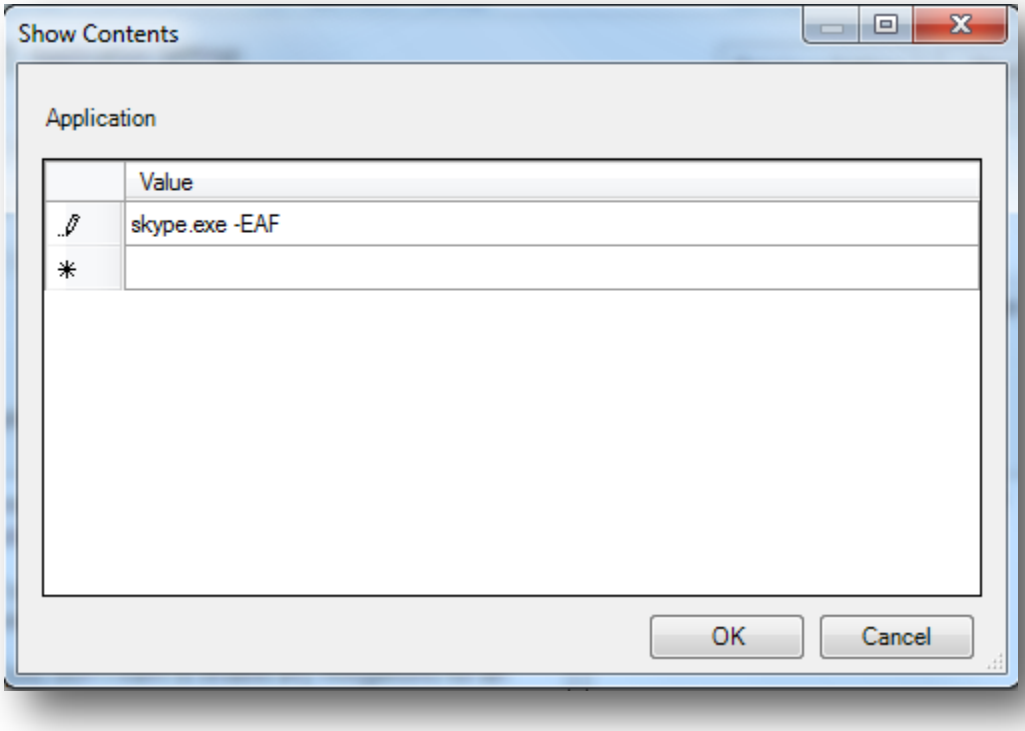
There are three sets of policies that EMET exposes. Below is a description of each. More information can be found at the policy editor for each policy.

1. System Mitigations: Named ASLR, DEP and SEHOP, these policies are used to configure system mitigations. Please note that modifying system mitigation settings may require a reboot to be effective.

2. Default Protection Profiles: There are three: Internet Explorer, Office applications and other popular software. Protection Profiles are pre-configured EMET settings that cover common home and enterprise software. Apply these policies to enable them.

3. Application Settings: This leads to a freeform editor where you can configure any additional applications not part of the default protection profiles. The syntax is application executable name followed by an optional list of mitigations you don't want to enable. If you don't specify any mitigation, all seven EMET application mitigations will be enabled.

   Below example enables all mitigations for Skype, with the exception of Export Address Table Filtering.



Once you enable EMET Group Policies, they will be written out to the registry at HKLM \SOFTWARE\Policies\Microsoft\EMET. To make them effective in EMET, you have to run the following command using the EMET Command Line Tool.

EMET_Conf --refresh

Please note that when you apply a Group Policy in Windows, there is often a short delay before Group Policy writes them out to the registry.

You can run this command separately, at startup or at logon time according to your deployment strategy.

To view the Group Policy controlled EMET settings, run the following command using the EMET Command Line Tool.

EMET_Conf --list

As can be seen below, this command lists two EMET rules. The settings controlled by Group Policy start with the '>' character. In this example, we have 2 settings and the Internet Explorer one has been configured by Group Policy, while the program.exe setting has been configured either through the EMET Graphical User Interface or the EMET Command Line Tool.

```
C:\Program Files\EMET>EMET_Conf.exe --list

Executable    Path                Mitigations
----------    ----                -----------
>iexplore.exe *\Internet Explorer DEP SEHOP NullPage HeapSpray EAF MandatoryASLR BottomUpASLR
program.exe   *                   DEP SEHOP NullPage HeapSpray EAF MandatoryASLR BottomUpASLR
```

It is important to note that the settings configured via Group Policy take precedence over the settings configured locally using the EMET GUI or the EMET Command Line Tool. Also, Group Policy controlled settings can only be modified or deleted via Group Policy. For example, running

EMET_Conf --delete_all

in the situation above would only delete the program.exe settings, and leave Internet Explorer settings intact.

## 3.3. Other Options

If you have a custom management solution not relying on either System Center Configuration Manager or Group Policy, it is recommended that you leverage the Protection Profiles feature presented in Section 2.1.

# 4. EMET Reporting

EMET 3.0 has reporting capability provided through an additional feature called the EMET Notifier. Once you install EMET, this lightweight component is set to automatically start with Windows. It will show up in the notification area of your taskbar with an EMET icon.

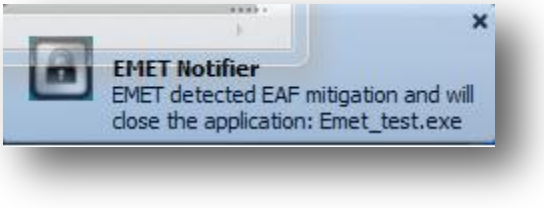
EMET Notifier has two duties:

**Write events out to the Windows Event Log**

EMET events are logged via the event source called EMET. These logs can be found at the Application log. There are three levels: Information, Warning and Error. Information messages are used for logging usual operation such as the EMET Notifier starting. Warning messages are used when EMET settings change. Error messages are used for logging cases where EMET stopped an application with one of its mitigations and this means an active attack was prevented.

**Show important events via a tooltip in the taskbar notification area**

Similar in severity to the error messages written to the Windows Event Log, when EMET crashes an application due to one of the mitigations, a message is displayed for the user, stating which application is being stopped and which mitigation is causing EMET to stop it.




Please note that it may not always be possible to provide meaningful information immediately when an application is crashing due to EMET. Thus, careful inspection of other log messages is also valuable when analyzing attacks stopped by EMET.

It is possible to turn off EMET reporting if you need to. For more information on how to turn on the unsafe ASLR setting, refer to Section 5 “Advanced Options”.

# 5. Advanced Options

**Enabling Unsafe Configurations**

By default, EMET hides configuration options considered to be unsafe. These are options that have shown to cause system instability in common use scenarios. For users still wishing to configure these options, there is a registry override. After the override is applied, EMET will display the unsafe options, but will also warn the user whenever one of them is selected.

The override can be found in registry at *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\EMET*. If you do not see this key, launch the EMET GUI and refresh your view of the registry. Inside the key, there is a DWORD value called *EnableUnsafeSettings*. By default it has a value of 0. By setting it to 1 and restarting the EMET GUI, you can see the unsafe options.

With EMET 3.0, there is currently one unsafe option: the "Always On" setting for the system ASLR setting. Depending on your operating system configuration, setting the system ASLR setting to "Always On" could make your operation system blue screen during boot. Recovering from this will require booting the system in safe mode and setting the system ASLR setting to either "Opt In" (recommended) or "Disabled".

**Disabling EMET Reporting**

Reporting is by default enabled in EMET. It is possible to disable it if needed. To do this, create a new DWORD called NotifierLogLevel under HKLM\SOFTWARE\Microsoft\EMET and set it to 0.

# 6. Mitigation Caveats

There are a few things you should be aware of when configuring the various mitigations available through EMET. In the following sections we discuss the caveats broken down by the system settings and application specific settings.

## 6.1. System Settings

### DEP

1. Configuring the system setting for DEP changes a boot option for Windows. For systems using BitLocker, this will cause BitLocker to detect that "system boot information has changed" and you will be forced to enter your recovery key the next time you boot Windows. It is highly recommended that you have your recovery key ready before changing the system configuration setting for DEP on a system with BitLocker enabled.

2. Not all systems, including virtual machines, support DEP. However, this option will still be available for configuration even if EMET is being run on a machine that doesn't support it. Setting this option on those systems will have no effect. Be aware of the limitations of your systems when configuring DEP.

### SEHOP

*None*

### ASLR

1. There is an unsafe option for the ASLR setting called "Always On". This setting will force address space randomization for binaries that do not specifically support it. This setting is not visible by default due to the risk of introducing system instability.

   In our tests we encountered issues in a common use scenario where having ASLR set to "Always On" would cause a system to blue screen during boot. This occurred because the address space for certain third party video drivers was being randomized. These drivers had not been built to support this randomization and subsequently crashed, causing the whole system to crash as well. Recovering from this issue requires booting into safe mode and switching the system ASLR setting to either "Opt in" or "Disabled".

   For more information on how to turn on the unsafe ASLR setting, refer to Section 5 "Advanced Options".

## 6.2. Application Specific Settings

### DEP

1. Not all systems, including virtual machines, support DEP. However, this option will still be available for configuration even if EMET is being run on a machine that doesn't support it. Setting this option on those systems will have no effect. Be aware of the limitations of your systems when configuring DEP.

### SEHOP

*None*

### Null Page

*None*

### Heap Spray

*None*

### Bottom-up randomization

*None*

### EAF

1. Systems configured with the /debug boot option need to have a debugger attached when running EAF enabled applications. If the /debug boot option is enabled and a debugger is not attached, the system will become unresponsive when an application with EAF enabled starts. This happens because the EAF mitigation relies on debug registers. If Windows has been configured to use a kernel debugger, Windows will try to inform the debugger whenever one of several memory addresses has been accessed. Windows will then wait for a response from the debugger. If a debugger does not respond, the system will appear unresponsive.

2. Some virtual machines do not support debug registers (and consequently EAF). However, the EAF option will still be available for configuration even if EMET is being run on a machine that doesn't support debug registers. Setting this option on those machines will have no effect. Be aware of this limitation when configuring EAF.

### Mandatory ASLR

1. EMET's mitigations only become active after the address space for the core process and the static dependencies has been set up. Mandatory ASLR does not force address space randomization on any of these. The main focus of Mandatory ASLR is to protect dynamically linked modules, such as plug-ins.

2. Windows XP and Windows Server 2003 do not support randomization. Since Mandatory ASLR does not protect the core process or static imports (see #1 above), they will always be at predictable addresses. Consequently, Mandatory ASLR is unable to provide any meaningful protection against attacks on these platforms and is therefore disabled. For more information on which platforms support which mitigations, see Section 1.3 Supported operating systems.

# 7. Frequently Asked Questions

## 7.1. EMET 2.1 Questions

- **Will my configuration from EMET 2.1 be carried over to EMET 3.0?**

  Yes, EMET 2.1 settings and exported setting files are compatible with EMET 3.0.

- **I have EMET 2.1 installed. Should I uninstall it before installing the new version?**

  You do not explicitly need to uninstall it before installing the new version. The EMET 3.0 installer will upgrade the older version of EMET as needed.

  If you have trouble upgrading, it is recommended that you uninstall EMET 2.1 and then install EMET 3.0

  In both of these cases, i.e. upgrading or removing 2.1 and installing 3.0, your existing EMET configuration will not be modified. It will carry over to the new version.

- **Are older versions (i.e. EMET 1.x) still supported?**

  As of EMET 3.0, we no longer support EMET 1.x.

## 7.2. General Mitigation Questions

- **In Process Explorer, the ASLR column for a process is blank even though EMET is configured for use with that application.**

  EMET does not take advantage of the OS implementation of ASLR. It will not show up in Process Explorer even when it is turned on because Process Explorer only queries the OS implementation of ASLR.

## 7.3. Troubleshooting Problems with Mitigations

- **I've modified the system setting for DEP and rebooted. Now BitLocker is asking me for the recovery key. Why is that and how can I stop it from asking me?**

  Modifying the system setting for DEP changes the boot options for the operating system. BitLocker cannot prevent an attacker from tampering with these options and instead monitors them for change. When they change, BitLocker asks for the recovery key to ensure the changes are legitimate.

To prevent BitLocker from continually asking for your recovery key, you will need to disable BitLocker (and decrypt the drive). Afterward, you can re-enable it (and re-encrypt the drive). This will cause BitLocker to record the new boot options.

- **My system hangs if the Export Address Filtering (EAF) mitigation is enabled.**

  This generally occurs when the system is running under DEBUG mode (the /debug boot option has been specified). Due to the nature of the EAF mitigation (involving debug registers and single step events) the hang occurs because the system waiting for a response from the debugger before continuing the execution of the application.

  To prevent this from happening, you can do one of the following:

  a) Remove the /debug boot option and reboot the system
  b) Attach a debugger and have it respond to the system.

- **One of my applications <u>always</u> crashes when I launch it after I configure EMET to protect it.**

  This generally occurs because the application is not compatible with one of EMET's mitigations. One way to figure out which mitigation is causing this is to start with all the mitigations enabled and disable them one by one until the application starts launching correctly without crashing. Once you determine the offending mitigation, you can disable it and still have the rest of the mitigations enabled.

  Please note the emphasis on "always" in the bold text above. A crash that happens 100% of the time no matter the nature of the user input is more likely to be an application compatibility issue if the application is coming from a vendor you consider to be trusted.

  Crashes that happen every now and then or crashes that happen based on external input such as crashes that happen only when opening a certain document with a reader or crashes that happen in applications that may come from untrusted sources should be treated differently. For these applications, EMET mitigations should not be deliberately disabled until the root cause of the crash is understood in order to avoid a security incident.

- **One of my applications <u>always</u> crashes when I launch it after I enable the EAF mitigation.**

  Similar in vein to the previous question, some applications might not work with the EAF mitigation. This is often caused by defenses that the application is implementing to protect intellectual property. We sometimes see that approach in video players, converters, VOIP programs etc. If you see a crash 100% of the time when the application is launching due to EMET's EAF mitigation in such

an application, you can disable EAF mitigation and still have the remaining mitigations in place for that application.

## 7.4. General Questions

- **I get the error "app failed to initialize properly" when attempting to launch the graphical user interface. How can I fix this?**

  The GUI requires that .NET 2.0 is installed on the system. If you get this error after copying the binaries form another machine, try running the installer on the local machine. It will direct you to a location where you can download the .NET 2.0 redistributable.

- **Does EMET work on 64 bits applications? It is installed in the 32bit program files directory.**

  Yes, EMET supports 64 bit applications. The installer is designed to work on both 64 bit systems and 32 bit systems. A side effect of this is that the binaries are placed in the 32 bit directory.

  However, please note there could be some mitigation that is not available or applicable to 64 bit applications. Refer to Section 2.3 Supported Operating Systems for more details.

- **I have a Beta version of EMET v3 installed. How do I upgrade to EMET v3 release?**

  It is recommended that you first uninstall the Beta version via Windows Control Panel, and then manually delete the HKLM\Software\Microsoft\EMET and HKLM\Software\Policies\Microsoft\EMET keys prior to running the EMET v3 installer.

- **How can I know if my application is compatible with EMET?**

  Testing was done only for the applications included in the default Protection Profiles. For any other applications, it is recommended to thoroughly test them on a staging environment prior to deploying EMET protections for those applications on a live system.

- **Will plugins also be protected when I protect an application?**

  Yes, the mitigations apply to plugins such as ActiveX controls or other 3rd party add-ins that get loaded into an EMET protected process.

- **My antivirus application is complaining about EMET GUI.**

EMET GUI queries all the processes to get their DEP status.  We are aware that rarely, antivirus software may flag this behavior when they detect it on their own process.

EMET is not trying to do anything harmful so you can just allow this and EMET will still work.

# 8. Support

EMET 3.0 is an officially supported and actively developed Microsoft product.

The primary channel for support is through the TechNet forums at http://go.microsoft.com/fwlink/?LinkID=213962&clcid=0x409.

Users can also send email to switech@microsoft.com with questions, ideas and suggestions.

# A. Appendix: EMET Compatibility

Thinking about EMET compatibility is an important part of the deployment process. Compatibility in this context means "being able to run an application with all the EMET mitigations enabled without any loss of functionality".

EMET doesn't do anything harmful and it refrains from doing anything that would cause a high amount of incompatibility. This means that most applications will be compatible with it. It is however strongly recommended that you do enough application compatibility testing on your applications prior to deploying EMET protections for them.

In EMET 3.0, application compatibility testing was done on all the Microsoft and 3rd party applications that are part of the EMET protection profiles on all the supported platforms. A list of these applications and identified compatibility issues can be found in the table below.

Please note that whenever the version is not specified, the latest version can be assumed.

Y: Compatible / N: Not Compatible

| Application | Mandatory ASLR | DEP | SEHOP | EAF | Heap Spray Protection | Bottom Up Randomization | Null Page Allocation |
|---|---|---|---|---|---|---|---|
| **Internet Explorer 7-9** | Y | Y | Y | Y | Y | Y | Y |
| **Windows Media Player** | N | Y | Y | Y | Y | Y | Y |
| **Microsoft Works** | Y | Y | Y | Y | Y | Y | Y |
| **Microsoft Works Calendar** | Y | Y | Y | Y | Y | Y | Y |
| **Microsoft Works Calendar Reminder** | Y | Y | Y | Y | Y | Y | Y |
| **Skype** | Y | Y | Y | N | Y | Y | Y |

| Application | Mandatory ASLR | DEP | SEHOP | EAF | Heap Spray Protection | Bottom Up Randomization | Null Page Allocation |
|---|---|---|---|---|---|---|---|
| **Lync Communicator** | Y | Y | Y | Y | Y | Y | Y |
| **Live Writer** | Y | Y | Y | Y | Y | Y | Y |
| **Windows Live Mesh Operating Environment** | Y | Y | Y | Y | Y | Y | Y |
| **Windows Live Sync** | Y | Y | Y | Y | Y | Y | Y |
| **MSN Messenger** | Y | Y | Y | Y | Y | Y | Y |
| **Visio Viewer XP** | Y | N | Y | Y | Y | Y | Y |
| **PowerPoint Viewer XP** | Y | N | Y | Y | Y | Y | Y |
| **Visio XP** | Y | N | Y | Y | Y | Y | Y |
| **Access XP** | Y | N | Y | Y | Y | Y | Y |
| **Excel XP** | Y | N | Y | Y | Y | Y | Y |
| **Outlook XP** | Y | N | Y | Y | Y | Y | Y |
| **PowerPoint XP** | Y | N | Y | Y | Y | Y | Y |
| **Word XP** | Y | N | Y | Y | Y | Y | Y |
| **Publisher XP** | Y | N | Y | Y | Y | Y | Y |

| Application | Mandatory ASLR | DEP | SEHOP | EAF | Heap Spray Protection | Bottom Up Randomization | Null Page Allocation |
|---|---|---|---|---|---|---|---|
| **InfoPath XP** | Y | N | Y | Y | Y | Y | Y |
| **Visio Viewer 2003-2007-2010** | Y | Y | Y | Y | Y | Y | Y |
| **PowerPoint Viewer 2003-2007-2010** | Y | Y | Y | Y | Y | Y | Y |
| **Visio 2003-2007-2010** | Y | Y | Y | Y | Y | Y | Y |
| **Access 2003-2007-2010** | Y | Y | Y | Y | Y | Y | Y |
| **Excel 2003-2007-2010** | Y | Y | Y | Y | Y | Y | Y |
| **Outlook 2003-2007-2010** | Y | Y | Y | Y | Y | Y | Y |
| **PowerPoint 2003-2007-2010** | Y | Y | Y | Y | Y | Y | Y |
| **Word 2003-2007-2010** | Y | Y | Y | Y | Y | Y | Y |
| **Publisher 2003-2007-2010** | Y | Y | Y | Y | Y | Y | Y |
| **InfoPath 2003-2007-2010** | Y | Y | Y | Y | Y | Y | Y |
| **Google Chrome** | Y | Y | Y | Y | Y | Y | Y |
| **Google Talk** | Y | Y | Y | Y | Y | Y | Y |
| **Mozilla Firefox** | Y | Y | Y | Y | Y | Y | Y |

| Application | Mandatory ASLR | DEP | SEHOP | EAF | Heap Spray Protection | Bottom Up Randomization | Null Page Allocation |
|---|---|---|---|---|---|---|---|
| **Mozilla Firefox Plugin Container** | Y | Y | Y | Y | Y | Y | Y |
| **Thunderbird** | Y | Y | Y | Y | Y | Y | Y |
| **Thunderbird Plugin Container** | Y | Y | Y | Y | Y | Y | Y |
| **Adobe Photoshop CS2/CS3/CS4/CS5/CS5.1** | Y | Y | Y | Y | Y | Y | Y |
| **Adobe Acrobat Reader 8-9-10** | Y | Y | Y | Y | Y | Y | Y |
| **Adobe Acrobat 8-9-10** | Y | Y | Y | Y | Y | Y | Y |
| **Winamp** | Y | Y | Y | Y | Y | Y | Y |
| **Opera** | Y | Y | Y | Y | Y | Y | Y |
| **WinRAR (winrar.exe, unrar.exe, rar.exe)** | Y | Y | Y | Y | Y | Y | Y |
| **WinZip** | Y | Y | Y | Y | Y | Y | Y |
| **VLC Player** | Y | Y | Y | Y | Y | Y | Y |
| **RealPlayer (realplay.exe, realconverter.exe)** | Y | Y | Y | Y | Y | Y | Y |
| **mIRC** | Y | Y | Y | Y | Y | Y | Y |
| **7-Zip (7z.exe, 7zG.exe, 7zFM.exe)** | Y | Y | Y | Y | Y | Y | Y |

| Application | Mandatory ASLR | DEP | SEHOP | EAF | Heap Spray Protection | Bottom Up Randomization | Null Page Allocation |
|---|---|---|---|---|---|---|---|
| **Safari** | Y | Y | Y | Y | Y | Y | Y |
| **QuickTime** | Y | Y | Y | Y | Y | Y | Y |
| **iTunes** | Y | Y | Y | Y | Y | Y | Y |
| **Pidgin** | Y | Y | Y | Y | Y | Y | Y |
| **Java (java.exe, jawaw.exe, jawaws.exe)** | Y | Y | Y | Y | Y | Y | y |

When an incompatibility is found, the next step is to determine which mitigation is causing it. This can be done by running the application with all EMET mitigations enabled to reproduce the issue. This should be followed by removing mitigations one by one until the issue doesn't reproduce anymore. Once the offending mitigation is identified via this test process, it is recommended to still enable the non-offending mitigations in deploy-time to leverage EMET protections as much as possible.

Please feel free to contact us via the information in Section 8- Support to let us know of any incompatibilities you have encountered.