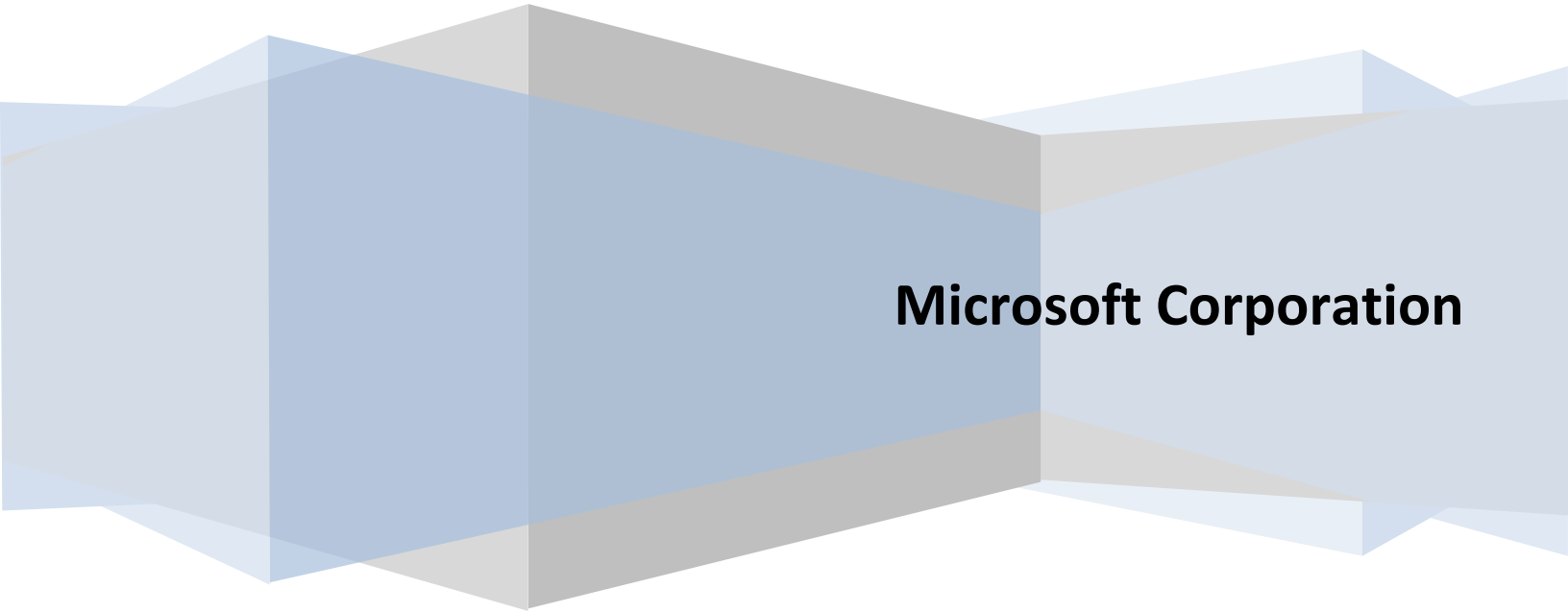
Enhanced Mitigation Experience Toolkit 2.0.0

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, the results can be devastating: malware, loss of PII, etc.

Security mitigation technologies are designed to make it more difficult for an attacker to exploit vulnerabilities in a given piece of software. EMET allows users to manage 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 handy 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. There is no need to locate up and decipher registry keys or run platform dependent utilities. With EMET you can adjust setting with a single 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 2.0.0 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 one 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 idependently 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.

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

1.2. Supported mitigations

The current version supports six different mitigation technologies. A training video covering many of the mitigations is available here: http://technet.microsoft.com/en-us/security/ff859539.aspx. The remainder of this section 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 Window 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 handler 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.

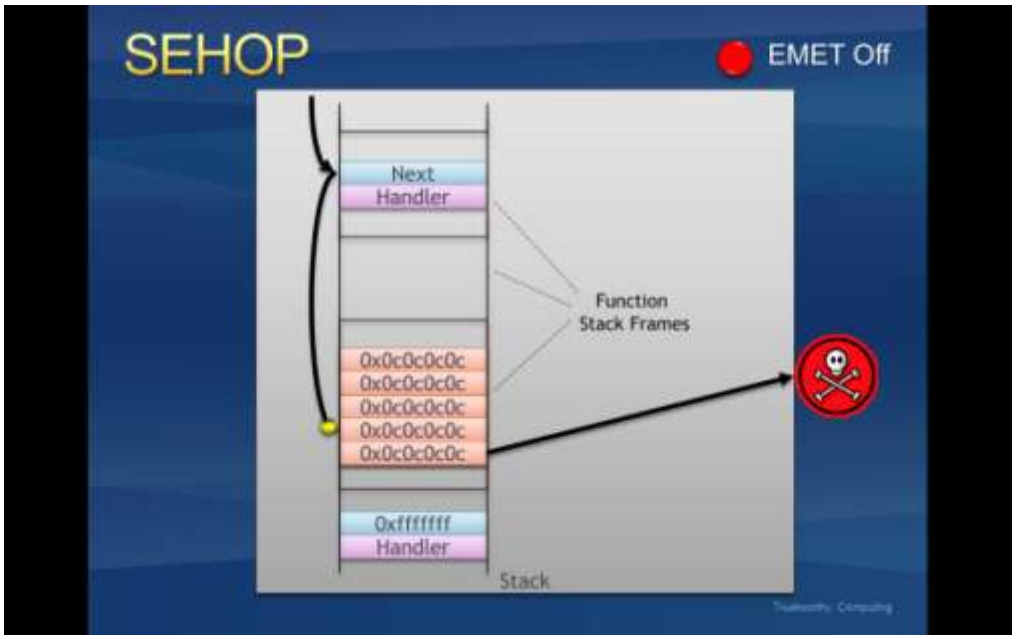


Figure 1: An exception handler hijack

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.

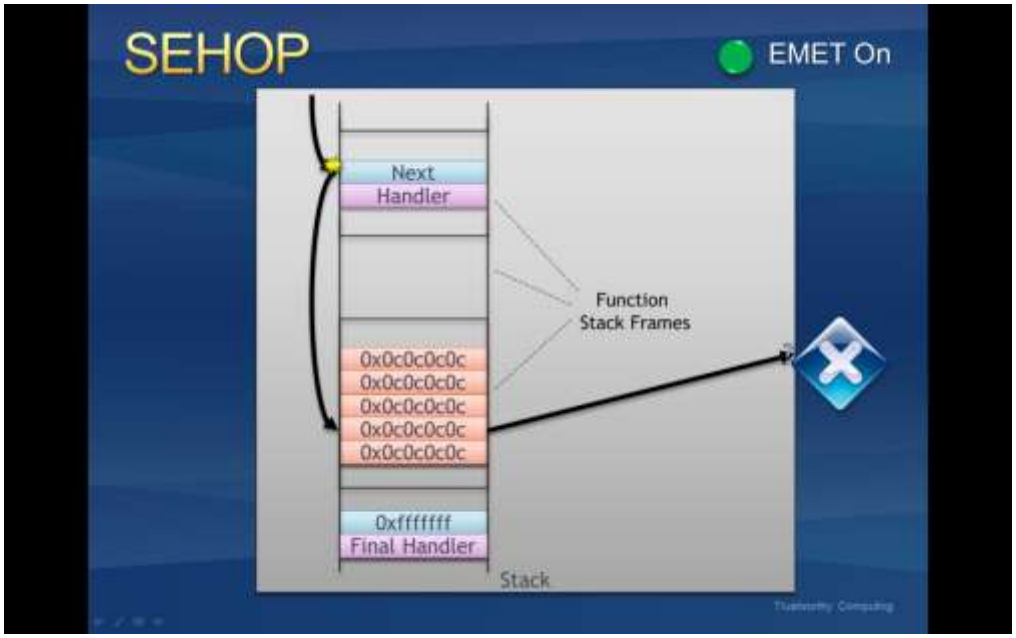


Figure 2: EMET stopping an exception handler hijack

Dynamice 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.

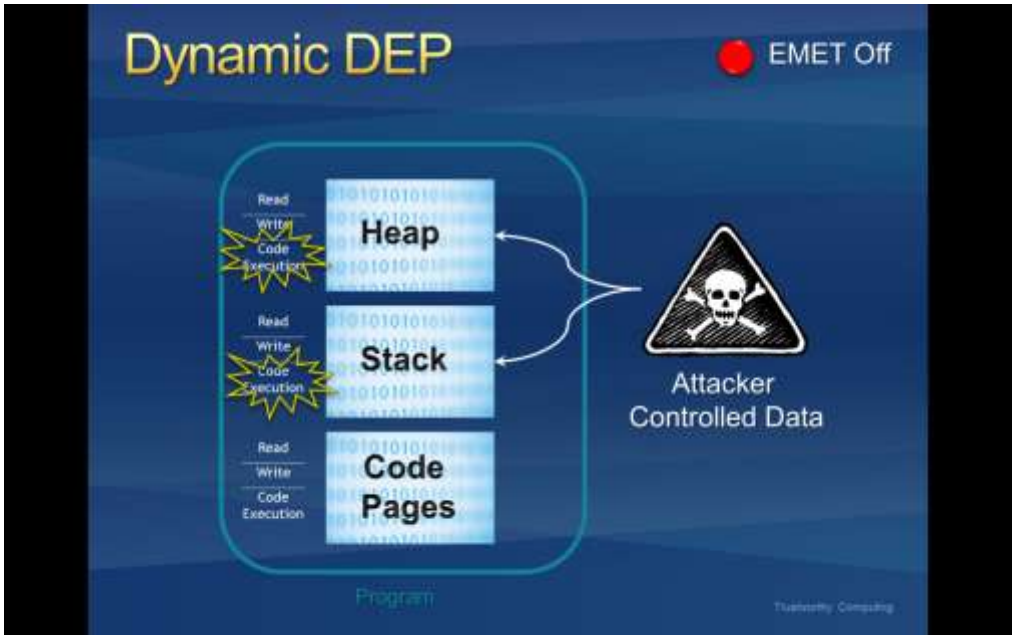


Figure 3: Running shellcode from attacker controlled locations

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.

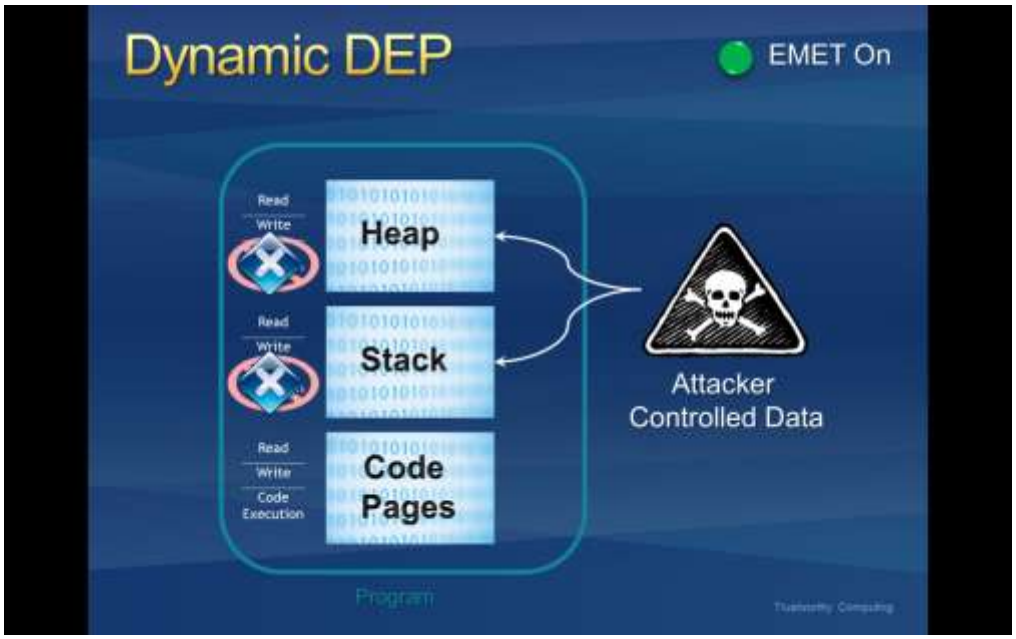


Figure 4: DEP blocking shellcode from running

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.

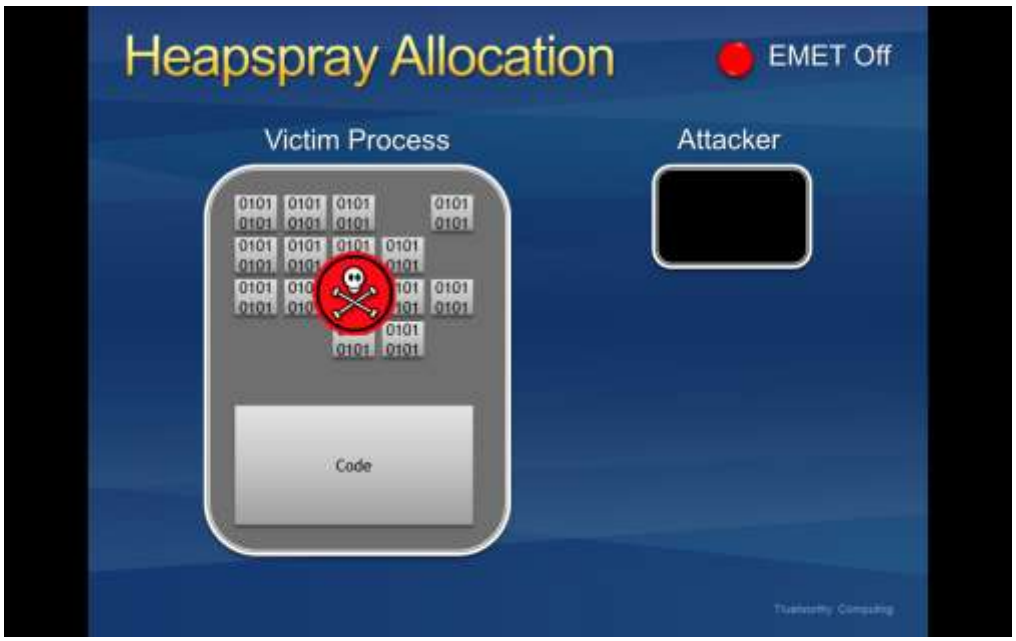


Figure 5: Using heapspray in an exploit

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.

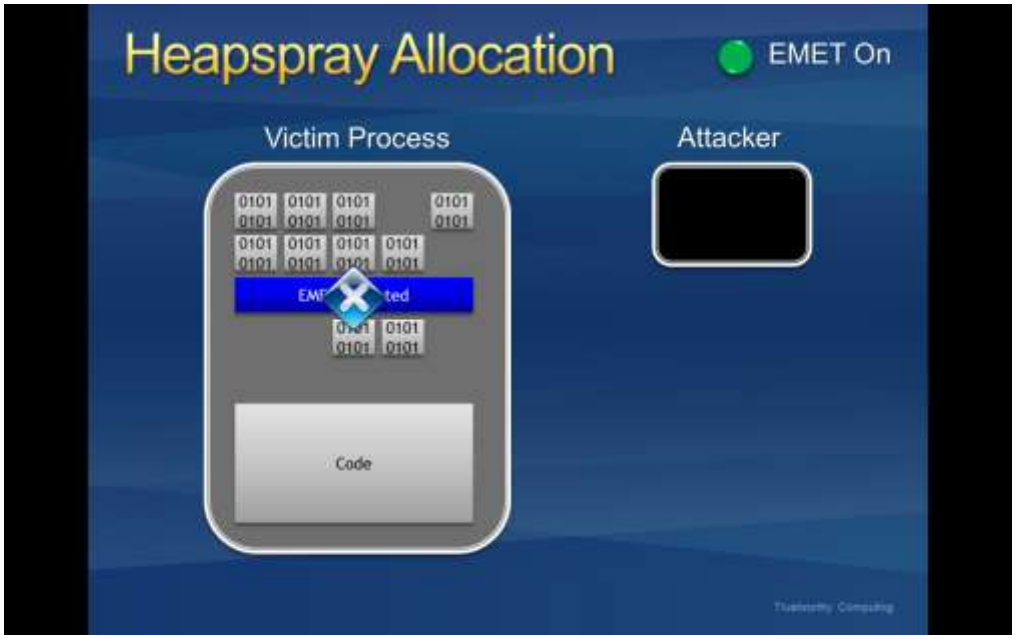


Figure 6: Blocking an attack that uses heapspray

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).

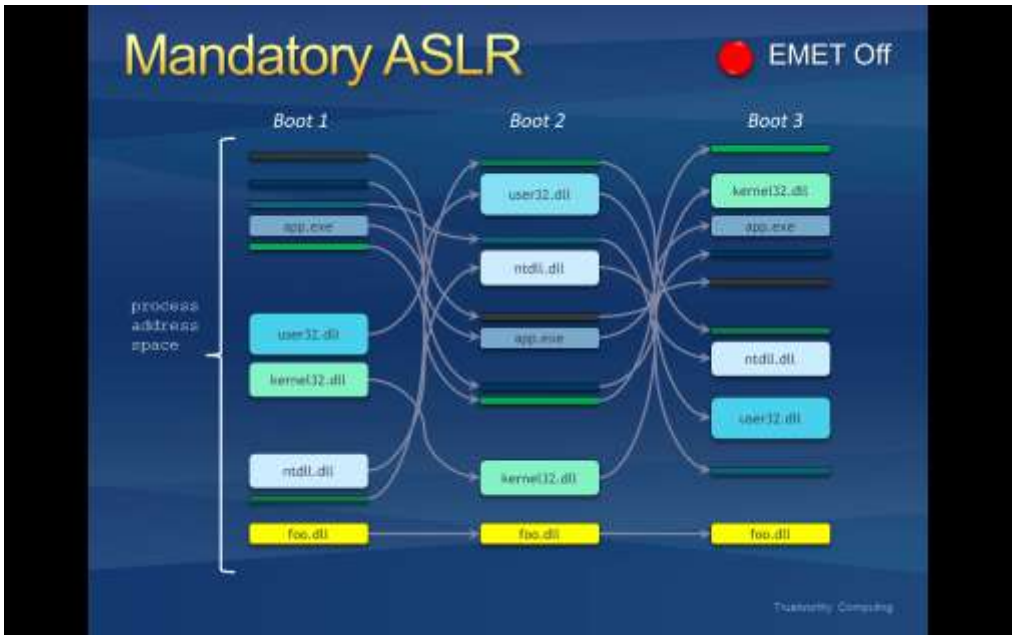


Figure 7: A module being loaded at a predictable location

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.

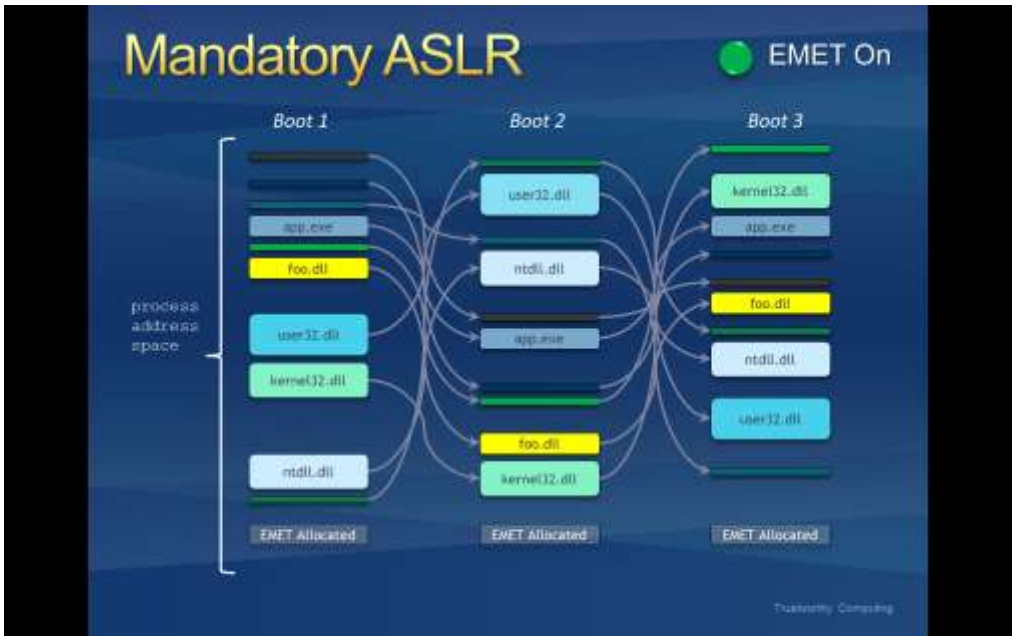


Figure 8: A module being forces to load at a random address

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.

1.3. Supported operating systems

EMET 2.0.0 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

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.

Application Settings

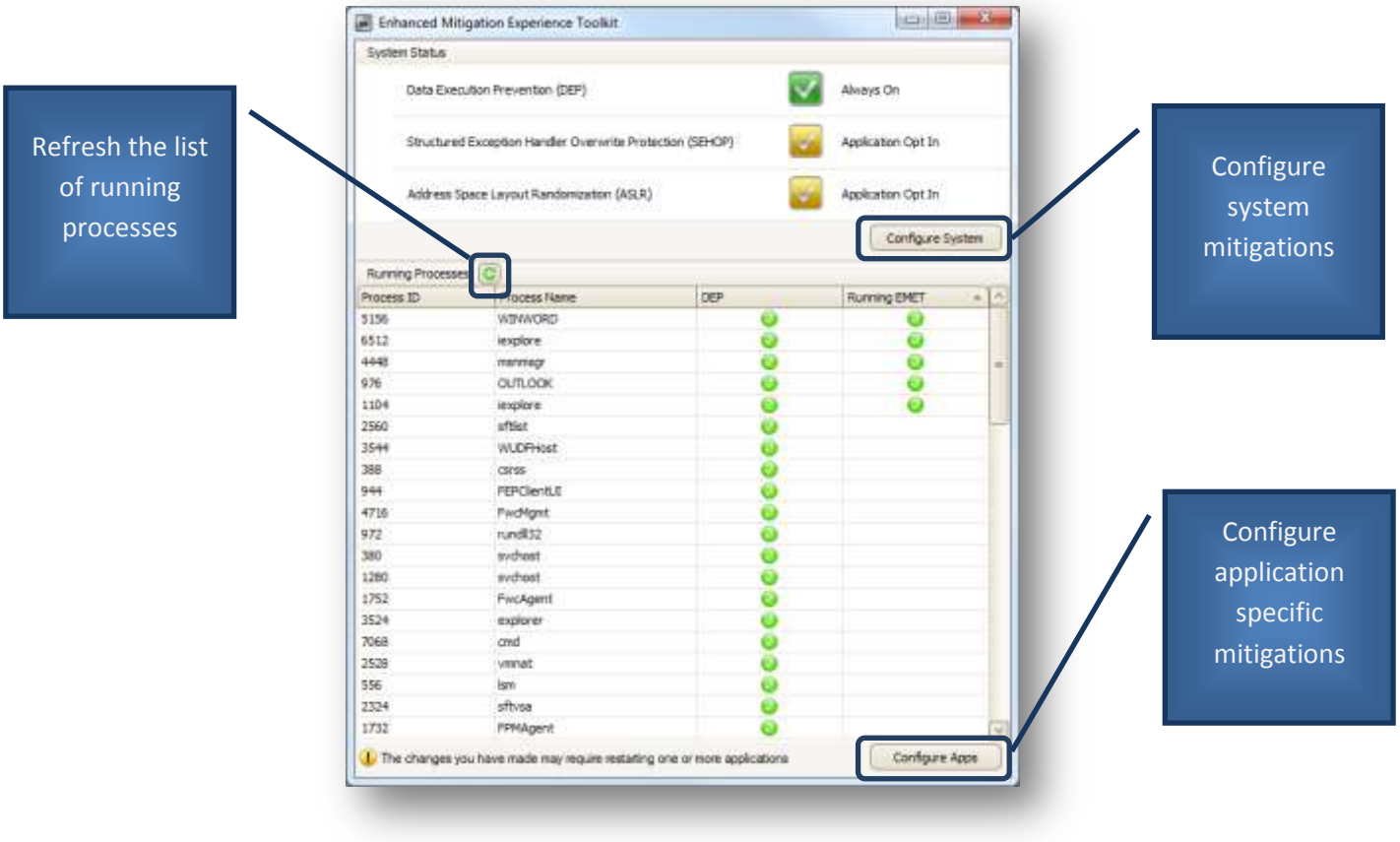
Mitigation	32 bit Processes	64 bit Processes
DEP	Supported by EMET	Already Mandatory without EMET
SEHOP	Supported by EMET	Not Applicable
NULL Page	Supported by EMET	Supported by EMET
Heap Spray	Supported by EMET	Supported by EMET
Mandatory ASLR	Supported by EMET	Supported by EMET
EAF	Supported by EMET	Not Supported

2. The graphical user interface

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

2.1. The main window

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

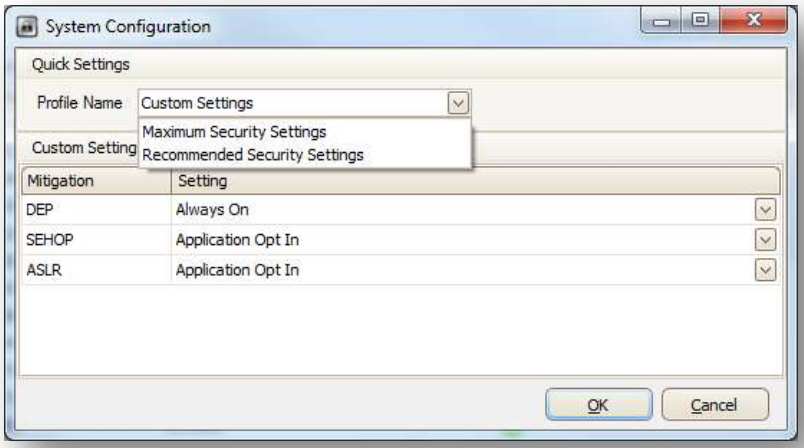


Though this initial window a user will be able to display the status of the different system mitigations and whether or not any of the running processes have been opted-in to EMET. Please note the list of running processes is only updated every 30 seconds. To get updated information on demand, click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

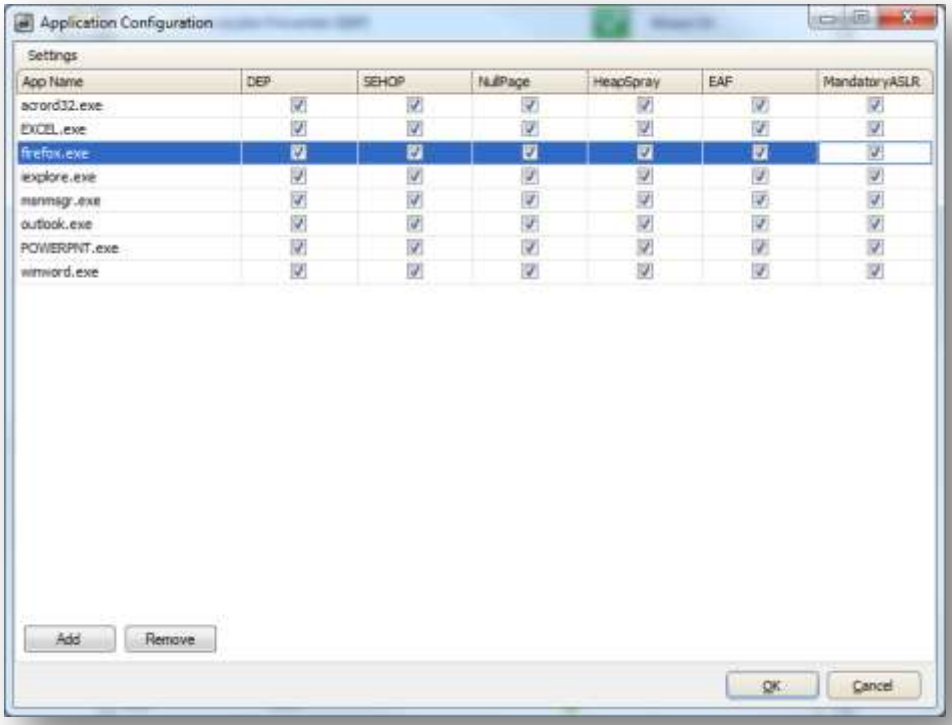
Users can configure system wide mitigations in two different ways. Either they can select one of the two profiles (“Maximum Security Settings” and “Recommended Security Settings”) or set the mitigation configuration individually.



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

2.3. 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 instance, 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.

It is important to note that the opt-in list is path dependent. A user must opt in the full path for each executable to be configured with EMET. Placing the mouse over an executable name will cause a tooltip to appear, showing the full path to the executable.

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).

3. The command line configuration tool

An alternate way to configure EMET is to use the EMET_conf command-line utility found at the location where EMET is installed. This utility is very similar to the one that shipped with EMET 1.0.2 with one notable exception. With 2.0.0, an application is configured based on the full path name to the executable, not just the executable file name. As with EMET 1.0.2, the EMET_conf utility requires administrator privileges. Be sure to run EMET_conf from an administrator command prompt.

The following table illustrates the options EMET_conf supports.

Task	**Command**
Add an application to EMET	*EMET_conf --add* *<full path to executable>*
List which applications EMET has been enabled for	*EMET_conf --list*
Remove an application from EMET	*EMET_conf --delete* *<full path to executable>*
Remove all applications from EMET	*EMET_conf --delete_all*

For example, to have EMET enable mitigations for foo.exe, run 'EMET_conf --add "c:\program files\app\foo.exe"'. To later disable the mitigations for foo.exe, run 'EMET_conf --delete "c:\program files\app\foo.exe"'.

The command-line utility does not support enabling and disabling the individual mitigations for an executable; all mitigations are enabled. The utility also does not support modifying the system mitigation settings. To do either of these tasks, please use the GUI.

4. Mitigation caveats

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

4.1. 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 would 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 the "Advanced options" section of this document.

4.2. 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

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 attack on the platforms and is therefore disabled. For more information on which platforms support which mitigations, see the "Supported operating systems" section of this document.

5. Advanced options

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 2.0.0, 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".

6. Frequently asked questions

6.1. EMET 1.0.2 questions

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

  The old version of EMET is incompatible with the new version. However, you do not explicitly need to uninstall it before installing the new version. The EMET 2.0.0 installer will disable the older version of EMET as needed.

- **I installed the new version of EMET, but my old settings disappeared. Can I get them back?**

  During the installation of EMET 2.0.0, the old EMET settings are disabled to prevent them from conflicting with the new version. However, they are not deleted. To find them simply open the registry editor and navigate to:

  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options

  For each key in there (e.g. calc.exe) open the key and look for a string value called "Debugger - from EMET 1.0.2". If you see that value, that executable was configured for use with EMET 1.0.2.

  Note: on 64 bit systems, you may also need to repeat the above step with the following path:

  HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\WindowsNT\CurrentVersion\Image File Execution Options

6.2. 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

- **With the previous version of EMET the DEP mitigation did not work on Windows Server. Does it work now?**

Yes, EMET 2.0.0 can now turn on DEP on Windows Server 2003. With the old version, the DEP policy was ineffective because Windows Server 2003 does not have the SetProcessDepPolicy() API. With the new version of EMET, DEP is enabled though a special process parameter

6.3. 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.

6.4. 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 as 32 bit systems. A side effect of this is that the binaries are place in the 32 bit directory.

However, please note several mitigations are not available on 64 bits. Refer to the “Supported operating systems” section for more details.

7. Support

EMET 2.0.0 is not currently an officially supported Microsoft product. We are working hard to establish the appropriate agreements to enable that. In the mean time, EMET is being released as an "AS-IS" product. That said, you can send your support questions to switech@microsoft.com and we will do our best to help you.