

# Enhanced Mitigation Experience Toolkit 5.5

## User Guide

January 2016

[www.microsoft.com/emet](http://www.microsoft.com/emet)

# Table of Contents

---

Introduction.....	1
Capabilities .....	2
Mitigations.....	2
Structured Exception Handler Overwrite Protection (SEHOP) .....	2
Data Execution Prevention (DEP).....	4
Heapspray Allocations .....	5
Null page allocation.....	7
Mandatory Address Space Layout Randomization (ASLR).....	7
Export Address Table Access Filtering (EAF) .....	8
Export Address Table Access Filtering Plus (EAF+).....	9
Bottom-up randomization .....	9
ROP mitigations.....	9
Attack Surface Reduction (ASR) .....	10
Advanced Mitigations for ROP .....	10
Certificate Trust (configurable certificate pinning).....	11
Untrusted font mitigation.....	11
Reporting.....	12
Supported Operating Systems and software requirements .....	13
EMET Configuration .....	15
EMET Protection Profiles.....	16
EMET Graphical User Interface .....	17
EMET Command Line Tool.....	23
Deploying EMET .....	26
Microsoft System Center Configuration Manager.....	26
Group Policy.....	28
Other Options.....	30

- Advanced Options .....30
  - Enabling Unsafe Configurations .....30
  - Configuring custom message for user reporting .....31
  - Configuring Certificate Trust feature for third party browsers .....31
  - Configuring Local Telemetry .....31
  - Configuring EMET Agent icon visibility .....31
- Mitigation Caveats .....31
  - System Settings .....32
  - Application Specific Settings .....32
- Frequently Asked Questions .....33
  - Lifecycle Policy .....33
  - EMET 4 Questions .....33
  - General Mitigation Questions .....33
  - Troubleshooting Problems with Mitigations .....34
  - General Questions .....35
- Support .....35
- Appendix A: EMET Compatibility .....36
- Appendix B: EMET 5.5 Release Notes .....36

# Introduction

The Enhanced Mitigation Experience Toolkit (EMET) is designed to help prevent attackers from gaining access to computer systems. EMET anticipates the most common techniques attackers might use to exploit vulnerabilities in computer systems, and helps protect by diverting, terminating, blocking, and invalidating those actions and techniques. EMET protects computers even before new and undiscovered threats are addressed by security updates and antimalware software. It helps enterprises and all PC users by protecting against security threats and privacy breaches that can disrupt businesses and daily lives.

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 zero day attacks, the results can be devastating: malware infections, loss of Personally Identifiable Information (PII), loss of business data, 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 customers to leverage these security mitigation technologies onto their systems that provide several unique benefits:

**No source code needed:** 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.

**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.

**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.

**Helps verifying SSL certificates trust while surfing websites:** Since incidents concerning Certificate Authorities allowing the creation of fraudulent SSL certificates to perform man-in-the-middle attacks are becoming a recurrent problem, EMET offers the possibility to enforce a set of pinning rules that can verify SSL certificates of specified domains against their issuing Root CA (configurable certificate pinning).

**Allows granular plugin blacklisting within applications:** Modules and plugins, when loaded into an application, increase its exposure to vulnerabilities and, consequentially, to potential attacks. EMET allows to blacklist modules and plugins that are loaded within an application.

**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 and decipher registry keys, or run platform dependent utilities. With EMET it is possible to adjust settings with a consistent interface regardless of the underlying platform.

**Ongoing improvement:** EMET is a living tool designed to be updated as new mitigation technologies become available. This provides a chance 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 systems 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.

## Capabilities

EMET allows to both configure the system policy for mitigations as well as to configure mitigations on a per executable basis. Furthermore, EMET offers the capability of validating SSL certificates against a set of configurable “pinning” rules, and is able to detect fraudulent ones.

The **system mitigation** policies allow the user to set the defaults for system supported mitigations; for instance choosing whether a mitigation should be enabled for all processes, enabled for only those that chose to opt-in, or disabled completely.

The **mitigations per executable** option allows the user to enable an EMET supported mitigation on an application. Any one of the supported mitigations can independently be turned on and off for any application residing on the system. Next time one of the configured applications runs, the specified mitigations will be applied to it. Combining this with the possibility to configure system mitigations give the user a high degree of control over the mitigations available on a system and how they get used.

The **Certificate Trust** feature allows to configure a set of certificate pinning rules to validate digitally signed certificates (SSL certificates) while browsing. These rules are designed to bind specific domains’ SSL certificates with the legitimate Root Certificate Authority (Root CA) that issued the certificate. When EMET detects the variation of the issuing Root CA for a specific SSL certificate configured for a domain, it will report this anomaly as potential symptom of an ongoing man-in-the-middle attack, and it allows users to block the connection.

EMET mitigation module does not run as a service, or attaches to an application like a debugger. Instead, behind the scenes, in order to enable mitigations for applications, EMET is leveraging an infrastructure in Windows called the Application Compatibility Framework. 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 have compatibility issues with some applications while executed. It is important to thoroughly test EMET in all target use scenarios before rolling it out to a production environment.

## Mitigations

EMET supports multiple mitigation technologies. In this section, we will outline the different mitigations and the protections they provide.

### **Structured 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. With Windows 7 and later versions of Windows, the ability to turn it on and off per process was added. With EMET, we provide the same

capabilities as recent versions of Windows also on older versions. 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.

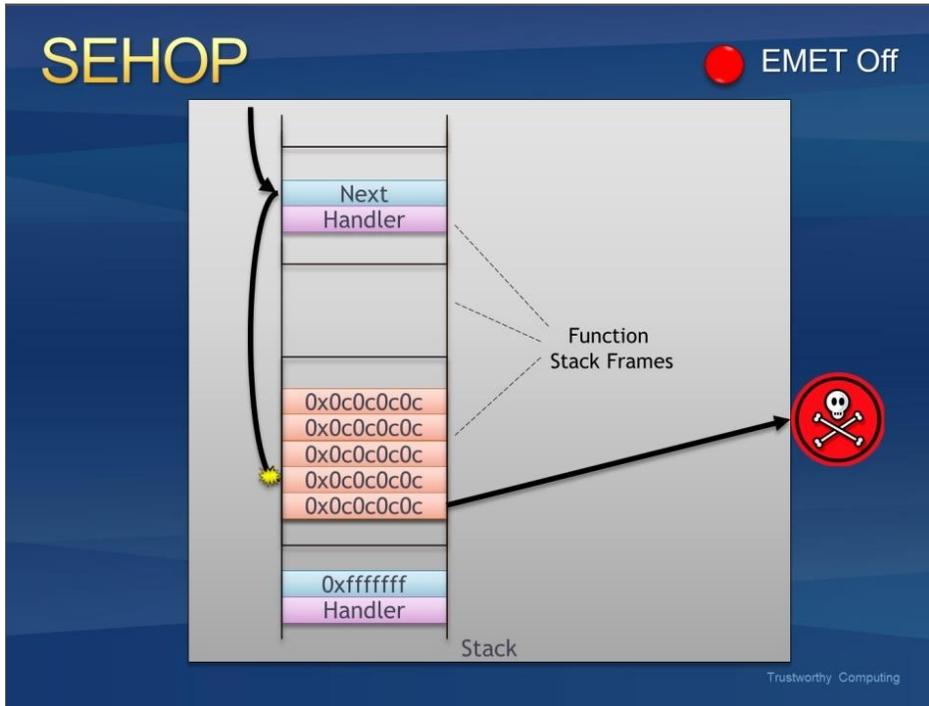


Figure 1: An exception handler hijack

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.

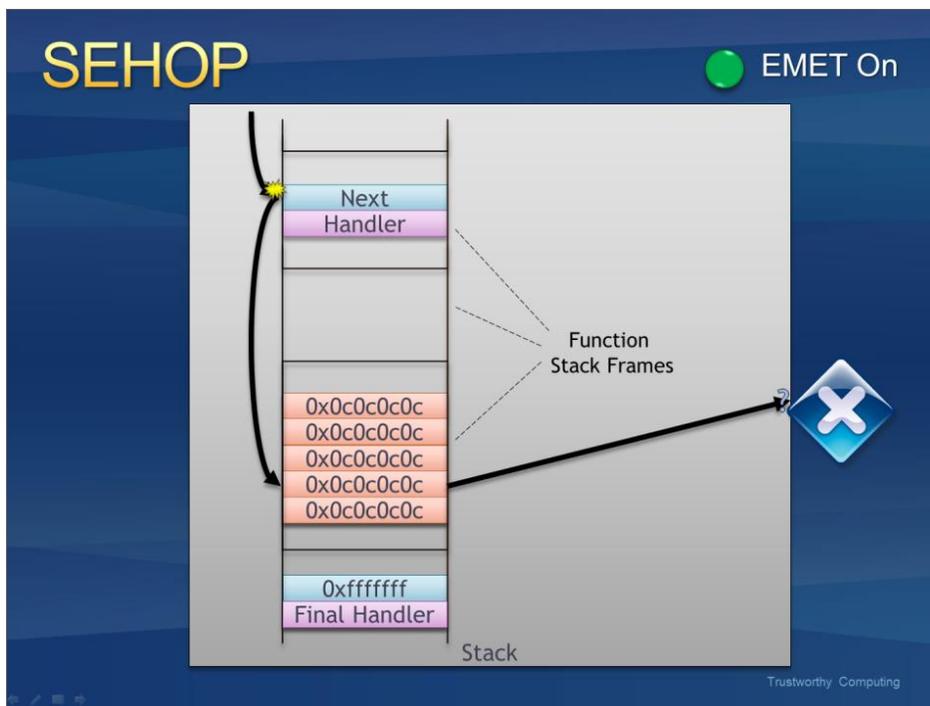


Figure 2: EMET stopping an exception handler hijack

NOTE: With Windows 7 and newer versions of Windows, EMET configures the native SEHOP provided by the operating system for the selected applications.

### 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, see [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.

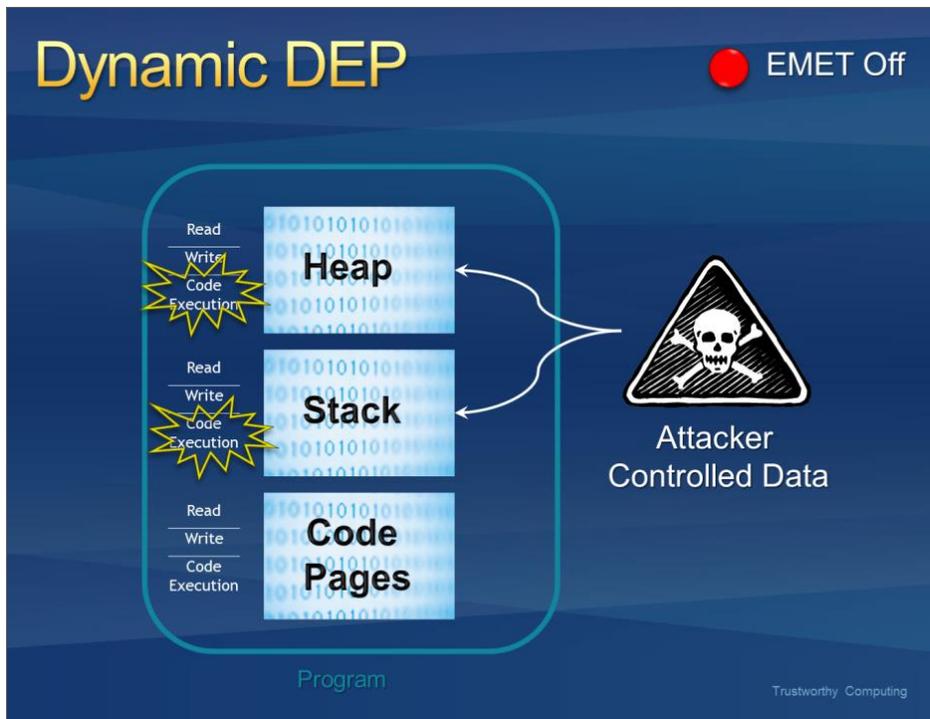


Figure 3: Running shellcode from attacker controlled locations

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.

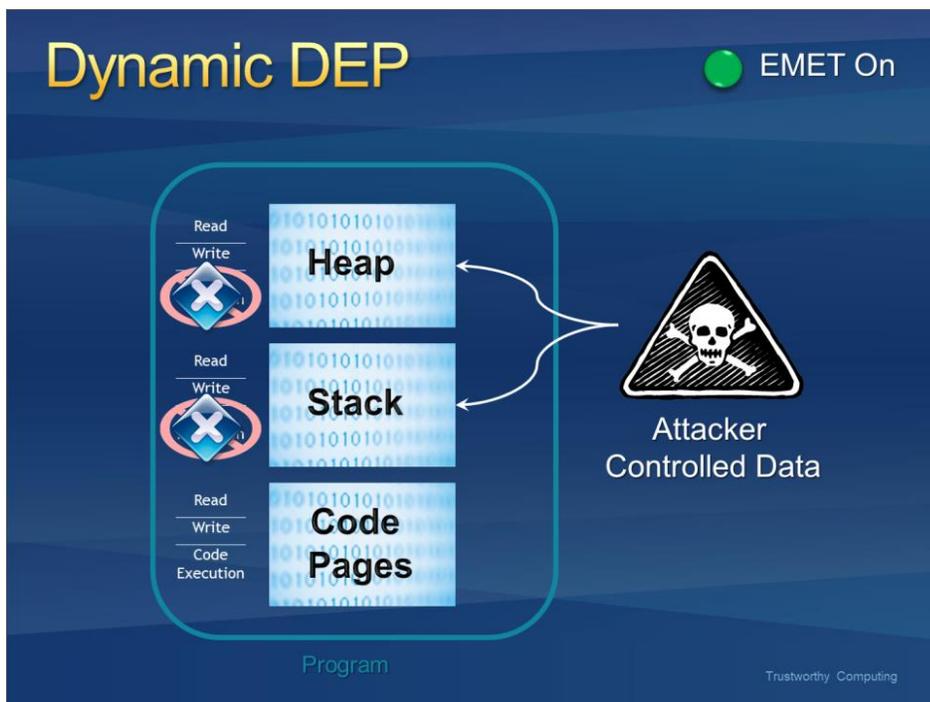


Figure 4: DEP blocking shellcode from running

### 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.

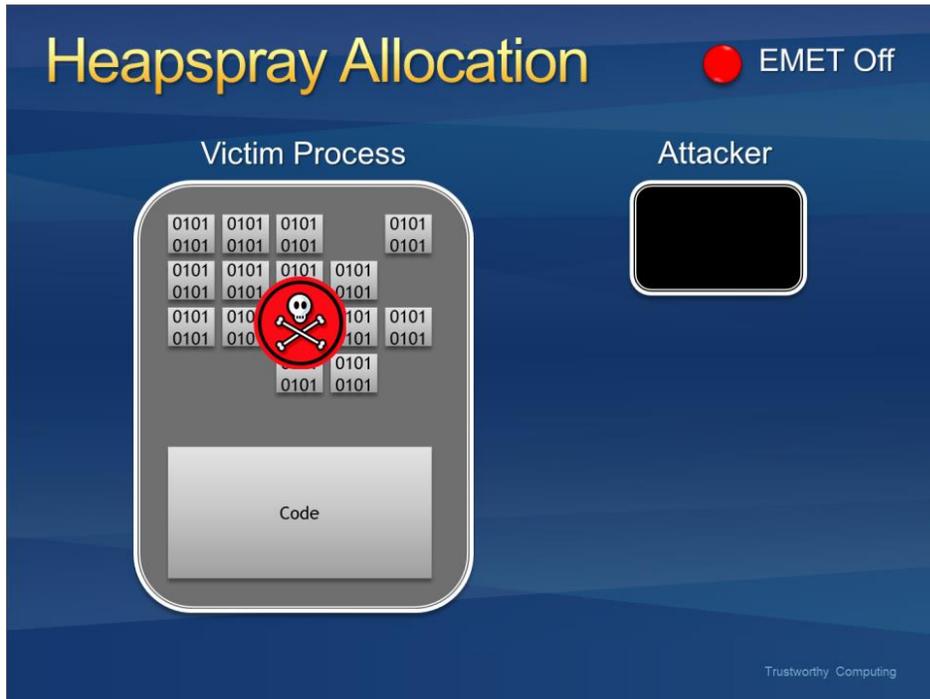


Figure 5 Using heapspray in an exploit

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.

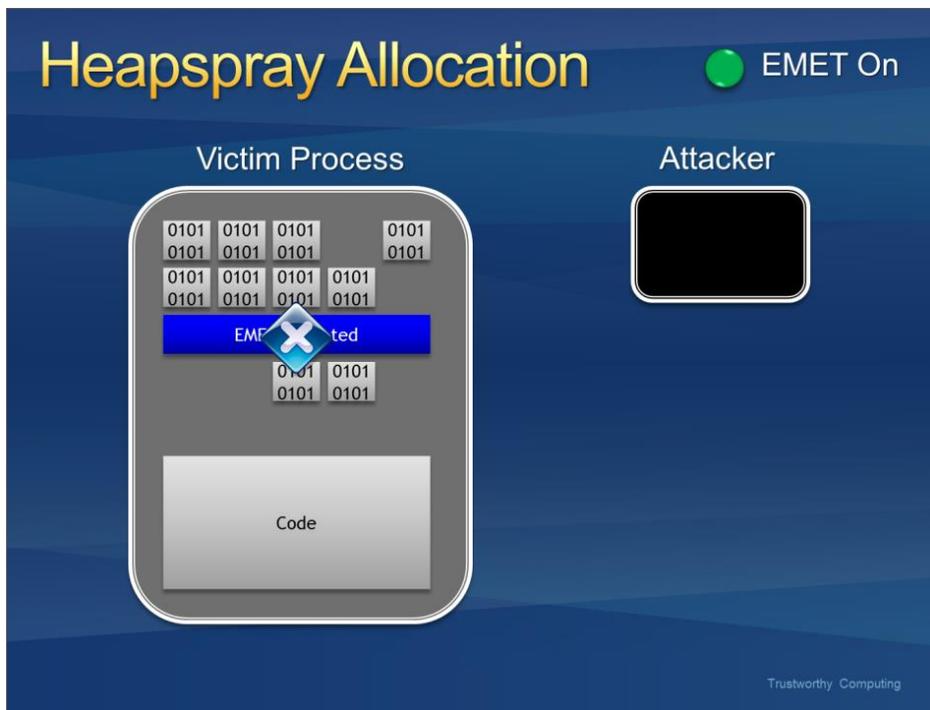


Figure 6: Blocking an attack that uses heapspray

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. 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.

## 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 through a known technique called return oriented programming (ROP).

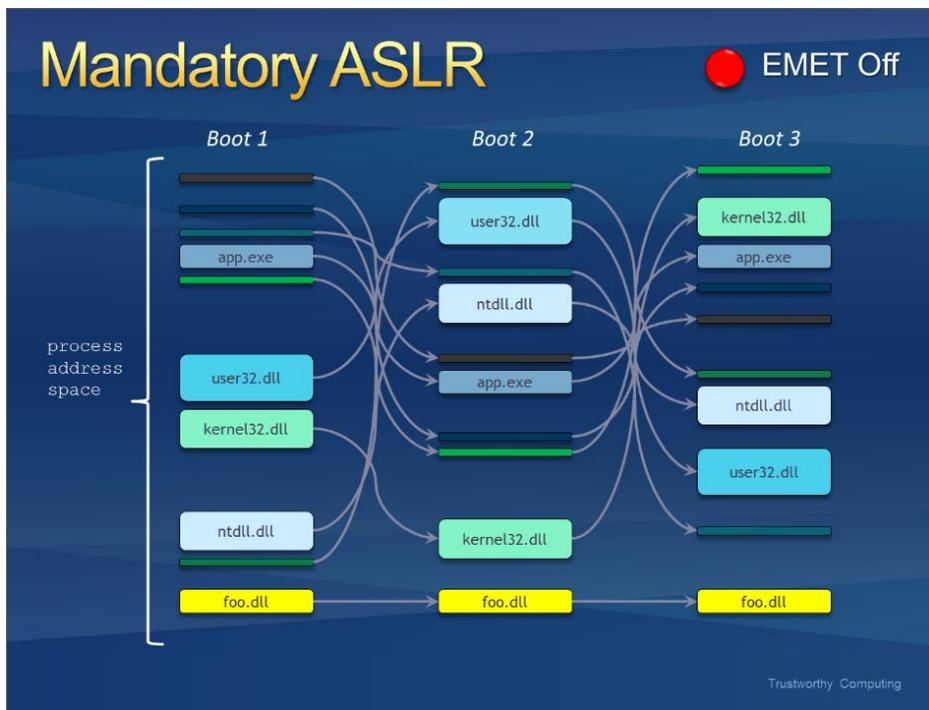


Figure 7: A module being loaded at a predictable location

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.

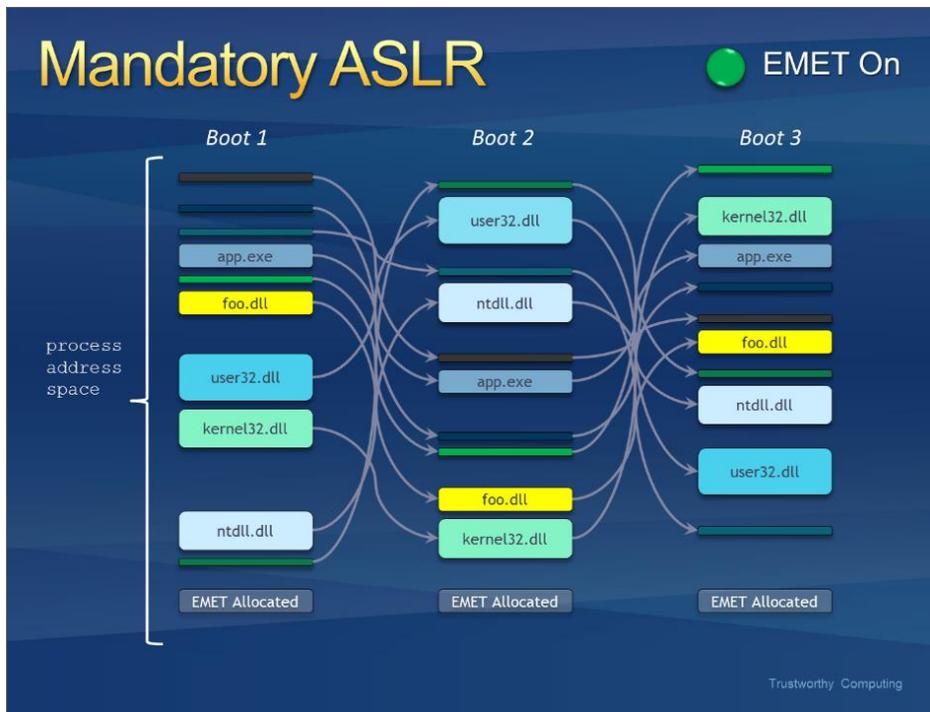


Figure 8: A module being forced to load at a random address

NOTE: With Windows 8 and newer versions of Windows, EMET will not use this mitigation if the native enforced ASLR provided by the operating system is already activated for an application.

### 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, ntdll.dll or kernelbase.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 read accesses to the Export Address Table (EAT), allowing or disallowing a read/write access based on whether the calling code originated from a shellcode. With EMET in place, most of today’s shellcode will be blocked when it tries to look up the APIs needed for its payload.

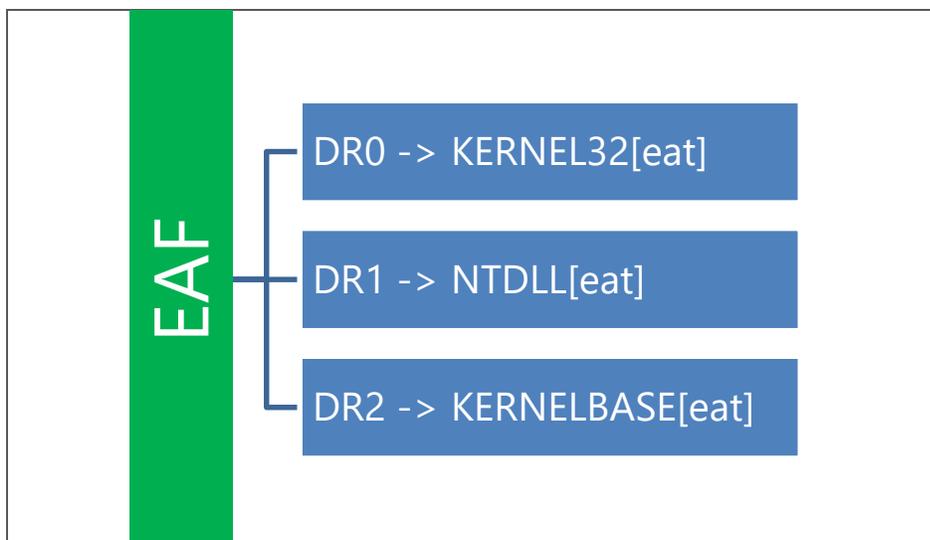


Figure 9: How EAF monitors accesses to critical modules' Export Address Table in old versions of EMET; new versions of EMET may employ different mechanisms to implement the protection.

This mitigation may have compatibility issues with software such as debuggers, software behaving like debuggers, or that use anti-debugging techniques. Examples include protection mechanisms for videogames, sandboxing solutions, DRM, debugging/tracing tools, and unpackers.

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

### Export Address Table Access Filtering Plus (EAF+)

The EAF+ mitigation is an extension of EAF that can be used independently or in combination with EAF itself. Following is the list of actions that this mitigation performs:

- Detects if the stack register is out of the allowed boundaries
- Detects mismatch of stack and frame pointer registers;
- Detects memory read access to export table pointers of KERNEL32, NTDLL and KERNELBASE originated from specific modules (typically used during the exploitation of memory corruption vulnerabilities);
- Detects memory read accesses to the MZ/PE header of specific modules (typically used during the exploitation of memory corruption vulnerabilities).

The actions described in the last two bullet points require users to specify a set of modules that will be used for validation; if no modules are specified, these two actions will be ignored.

### 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.

### ROP mitigations

EMET offers several experimental anti Return Oriented Programming (ROP) mitigations that aim to block any exploitation relying on this technique. ROP is an exploitation technique that facilitates the execution of code when mitigations like the Data Execution Prevention are in place. In order to do that, the ROP

technique uses snippets of code that are already present in the memory region of the attacked application.

Please note that all ROP mitigations are available and applicable to 32-bit, and some are available and applicable to 64-bit processes.

The following is a high-level description of the available ROP mitigations:

**Load library checks:** EMET monitors all calls to the LoadLibrary API and prevents loading libraries from UNC paths (i.e. \\evilsite\bad.dll). It is possible to disable this option if a program is known to legitimately load DLLs from UNC paths or remote servers. This mitigation is available for 32 and 64 bit processes.

**Memory protection checks:** EMET disallows making the stack area executable. Such activity is usually used by shellcode or ROP gadgets. This mitigation is available for 32 and 64 bit processes.

**Caller checks:** EMET makes sure that when a critical function is reached, it is reached via a CALL instruction rather than a RET instruction. This is a very useful mitigation and breaks many ROP gadgets. This mitigation may be incompatible with some applications. This mitigation is available for 32 bit processes.

**Simulate execution flow:** This feature tries to detect ROP gadgets following a call to a critical function. Like the "Caller checks", this feature may not be compatible with some applications. This mitigation is available for 32 bit processes.

**Stack pivot:** This mitigation is used to detect if the stack has been pivoted. This mitigation also validates the stack register present in the context structure of certain APIs. It is compatible with most programs. This mitigation is available for 32 and 64 bit processes.

### Attack Surface Reduction (ASR)

The Attack Surface Reduction (ASR) helps reduce the exposure of applications at risk for attacks by blocking the usage of specific modules or plugins within the target application. For example, EMET can be configured to prevent Microsoft Word from loading the Adobe Flash plugin, or, with the support of [Security Zones](#), can be configured to prevent Internet Explorer from loading the Oracle Java plugin on a website in the Internet Zone while continuing to allow Java on Intranet Zone websites. The mechanism simply prevents DLL loading on a per-process base, and it essentially adds the benefit to "killbit" specific modules in specific applications.

### Advanced Mitigations for ROP

EMET offers additional mitigation options that apply to all configured software. The currently available additional mitigations are related to the ROP mitigations, and when enabled or disabled they affects all the programs that have any of the ROP mitigations configured.

Following is a summary of what these advanced mitigations are:

---

Requires ROP	<p><b>Deep hooks:</b> EMET protects critical APIs and the subsequent lower level APIs used by the top level critical API. For example, EMET not only hooks and protects <i>kernel32!VirtualAlloc</i> but also the related lower level functions, <i>kernelbase!VirtualAlloc</i> and <i>ntdll!NtAllocateVirtualMemory</i>.</p> <p><b>Anti detours:</b> Some exploits attempt to evade the hooks by executing a copy of the hooked function prologue and then jump to the function past the prologue. With "Anti detours" option enabled, common shellcode using this technique are not effective.</p>
--------------	--

---

---

**Banned functions:** By enabling this option, EMET will block calls to *ntdll!LdrHotPatchRoutine* to mitigate potential exploits abusing this API.

---

### Certificate Trust (configurable certificate pinning)

EMET provides a mechanism that adds additional checks during the certificate chain trust validation process, with the goal to detect man-in-the-middle attacks over an encrypted channel. Each time a certificate chain trust is built by Internet Explorer for a SSL certificate while browsing to an HTTPS website for which certificate pinning is configured, EMET will validate the end-entity SSL certificate and verify that it chains to the Root CA specified by the pinning rule configured for that site.

Depending on the pinning rule for a specific domain, when EMET detects that the server authentication certificate chains to a different root certificate than the one specified in that domain's pinning rule, EMET can either simply warn the user but allow the connection, or block the connection entirely. By default EMET warns the user with a "toast" notification near the taskbar's notification area and does not stop the connection. It's a simple checkbox to turn the rule into a blocking rule. Pinning rules are implemented in terms of a root CA certificate's thumbprint (sometimes called a "fingerprint"). A user or IT admin defines a pinning rule by picking one or more root certificates from the configuring machine's root certificate store. When a pinning rule is validated, the server's root CA certificate's thumbprint then must match one of the thumbprints associated with the pinning rule. (Note that this is a change from earlier versions of EMET that matched certificate subject name and serial number, or alternatively a hash of its public key.) EMET also matches the certificate subject name (CN) of the SSL certificate, including the alternative names if available, against the website name configured in the pinning rules. A set of trusted Root CA can be defined importing their certificates from the Windows [Trusted Root Certification Authorities](#) store. By default, the root store contains only a subset of all trusted root CAs. You can download the full set of trusted root CA certs by running this command: **certutil -SyncWithWU directory**. You can then import any or all of those certificates into the local root store with the **certutil -addStore** command.

A pinning rule for a specific domain can also be temporarily marked inactive, so that the rule is not enforced.

Note that exceptions to pinning rules that were provided in earlier versions of EMET and based on key size, hashing algorithm, or issuer country are no longer implemented in EMET, although they still appear in the configuration UI.

### Untrusted font mitigation

Windows 10 added a Blocking Untrusted Fonts feature to protect users from attacks originating from untrusted or attacker-controlled font files. (You can learn more about the feature by reading the "[Block untrusted fonts in an enterprise](#)" Technet article.)

EMET provides a way to enable untrusted font protection system-wide, and to exempt specific apps from that protection.

. Untrusted fonts are any font installed outside of the %windir%/Fonts directory. Blocking untrusted fonts helps prevent both remote (web-based or email-based) and local EOP attacks that can happen during the font file-parsing process.

# Reporting

EMET has reporting capability provided through a Windows Service called "Microsoft EMET Service". Once EMET is installed, the service is set to automatically start with Windows. The EMET Service is responsible to dispatch the EMET Agent, which will show up in the system tray area of the taskbar with an EMET icon. The visibility of the EMET Agent icon in tray area can be configured via Group Policy or via the command line tool).

The EMET Service performs the following tasks:

**Write events in the Windows Event Log:** EMET events are logged via the event source called EMET. These logs can be found in the Application Log. There are 3 different levels of logging: Information, Warning and Error. Information messages are used for logging usual operation such as the EMET Agent starting. Warning messages are used when EMET settings change or to report Certificate Trust detections of SSL certificates validated by an exception rule. Error messages are used for logging cases where an untrusted SSL certificate is detected or where EMET stopped an exploit with one of its mitigations and this means a possible active attack was prevented. The list of possible EventIDs associated with EMET reporting is presented below; users should be also aware that some mitigations may not be fully logged by EMET when they are configured as System mitigations and are natively provided by the operating system.

Table 1: Event ID formats

Event Level	Event ID
Information	[S]0
Warning	[S]1
Error	[S]2

[S] is a number used to identify the subsystem sending the log event (possible values: 0-4)

Table 2: possible EventIDs

Event Level	Mitigation	GUI	Command Line	Agent	Certificate Trust
Information	00	10	20	30	40
Warning	01	11	21	31	41
Error	02	12	22	32	42

Table 3: EMET mitigations available for event logging

Man dator y ASLR	DEP	SEH OP	EAF	EAF+	Heap Spra y	Bott om Up	Null Page	Load Libra ry	Mem Prot	Sim Exec Flow	Stack Pivot	ASR
✓ *	✓ *	✓ *	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓

(\*) when configured as System mitigation, it may not generate a log entry

**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 stops an exploit due to one of the mitigations or detects an untrusted SSL certificate, a message is displayed for the user, stating which application is being

stopped and which mitigation has been used to stop the exploit. In case of a Certificate Trust violation, it shows details about the untrusted SSL certificate on the current HTTPS connection.

**Perform certificate trust validation tasks:** SSL certificates, Root CA certificates and pinning rules are enforced and validated only when the EMET Service is active and running.

**Send reports for the Early Warning Program:** EMET offers the “Early Warning Program” reporting feature. When an exploitation attempt is detected and blocked by EMET, a set of information related to the attack will be sent back to Microsoft through the standard Windows Error Reporting channel. This information will help Microsoft to obtain information related to Oday exploits and will facilitate the remediation of the issue before it becomes a large scale threat. If the vulnerability is related to a software from a third party vendor, Microsoft will work with the affected vendor through the Microsoft Vulnerability Research program to remediate the issue.

The Early Warning Program reporting feature also sends back to Microsoft information related to suspicious SSL certificates related to Microsoft online services. Please refer to the “Privacy Statement.rtf” file, available also through the “Help” ribbon in EMET GUI, or at <http://aka.ms/EMETips>, for more information on the type of data that will be sent to Microsoft.

## Supported Operating Systems and software requirements

### Supported Operating Systems and Applications

EMET 5.5 supports the following operating systems and service pack levels (or latest, as long as they continue to be within the [Windows Support Lifecycle](#)):

#### Client Operating Systems

- Windows 10
- Windows 8.1
- Windows 8
- Windows 7 Service Pack 1
- Windows Vista Service Pack 2

#### Server Operating Systems

- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 Service Pack 1
- Windows Server 2008 Service Pack 2

Please note that not all mitigations are supported on all operating systems.

Table 4: system mitigations compatibility matrix

Mitigation Type	Mitigation	Vista, Windows 8, 8.1/Server 2008 and newer	Windows 10
System-wide	DEP	✓	✓
	SEHOP	✓	✓
	ASLR	✓	✓
	Untrusted fonts	✗	✓

Application	DEP	✓	✓
	SEHOP	✓	✓
	NULL Page	✓	✓
	Heap Spray	✓	✓
	Mandatory ASLR	✓	✓
	EAF	✓	✓
	EAF+	✓	✓
	Bottom-up ASLR	✓	✓
	Load library	✓	✓
	Memory protection	✓	✓
	Simulate execution flow	✓	✓
	Stack pivot	✓	✓
	Caller checks	✓	✓
	ASR	✓	✓
	Untrusted fonts	✗	✓

Additionally, on 64 bit systems, some mitigations are only applicable with 32 bit processes. For details, refer to the following table:

Table 5: application mitigations compatibility matrix

Mitigation Type	Mitigation	32-bit Processes	64-bit Processes
Application	DEP	✓	✓
	SEHOP	✓	✗
	NULL Page	✓	✓
	Heap Spray	✓	✓
	Mandatory ASLR	✓	✓
	EAF	✓	✓
	EAF+	✓	✓
	Bottom-up ASLR	✓	✓
	Load library	✓	✓
	Memory protection	✓	✓
	Simulate execution flow	✓	✗
	Stack pivot	✓	✓
	Caller checks	✓	✗

ASR	✓	✓
Untrusted fonts	✓	✓

EMET can be installed and used in virtual machines, however virtualized applications such as Microsoft App-V or VMware ThinApp™ are not supported.

The Certificate Trust feature is supported for Internet Explorer only, but it can be configured for other browsers with an experimental setting. Please see the paragraph Configuring Certificate Trust feature for third party browsers for more information.

### Software requirements

EMET requires the Microsoft .NET Framework 4. In addition, in order for EMET to work properly on Windows 8 and Windows Server 2012, [Microsoft KB 2790907 – Compatibility update is available for Windows 8 and Windows Server 2012](#) or a more recent version of the compatibility update must be installed.

# EMET Configuration

EMET must be configured after the EMET installation, for the security mitigations to be enabled. To configure EMET, the following settings have to be specified:

- Which system mitigations should be enabled.
- Which applications should be protected with which mitigations.
- What SSL/TLS certificate pinning rules to adopt.

Both system and application mitigations can be configured via the EMET Graphical User Interface or via the EMET Command Line Tool. The Certificate Trust feature for SSL/TLS connections can be configured only via the EMET Graphical User Interface. Refer to paragraphs EMET Graphical User Interface and EMET Command Line Tool for further instructions on how to use these interfaces.

It is also possible to use Group Policy to configure system and application mitigations for EMET. Group Policy support is explained in the Group Policy paragraph.

Another option for configuring EMET is using Protection Profiles. Refer to paragraph 2.1 for details on what is contained in these Protection Profiles.

An additional, and the easiest, way to configure EMET is through the Configuration Wizard. At the end of the installation, the Configuration Wizard will ask to apply a set of recommended settings. In case a manual configuration is preferred, the Configuration Wizard can be ignored. For more information about the Configuration Wizard see the paragraph Configuration Wizard.

Normal EMET configuration settings are saved in the registry sub-key *HKLM\SOFTWARE\Microsoft\EMET* and some limited user-specific settings are saved also in *HKCU\SOFTWARE\Microsoft\EMET*. EMET settings configured via Group Policies are saved in the registry sub-key *HKLM\SOFTWARE\Policies\Microsoft\EMET*. Some per-application settings may also exist in the *HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options*. Note that on 64-bit platforms, the settings may be saved in the both 64-bit hives and their 32-bit counterparts in the registry (e.g. *Wow6432Node*).

## EMET Protection Profiles

EMET comes with two default Protection Profiles for applications and one protection profile for Certificate Trust. Protection Profiles are XML files that contain pre-configured EMET settings for common Microsoft and 3<sup>rd</sup> party applications. In the EMET installation directory, these files are in the Deployment\Protection Profiles folder. They can be enabled as-is, modified, or used to create new protection profiles.

The profiles that are included with EMET are

**Recommended Software.xml:** Enables mitigations for Microsoft Internet Explorer, WordPad, applications that are part of the Microsoft Office suite, Adobe Acrobat, Adobe Reader, and Oracle Java.

**Popular Software.xml:** Enables mitigations for other common applications.

**CertTrust.xml:** Enables certificate pinning rules for the login services of Microsoft Account, Microsoft Office 365, and Skype, and other popular online services such as Twitter, Facebook, and Yahoo!.

NOTE: EMET protection profiles are optimized configurations that take into consideration the known compatibility issues of some applications. The default Certificate Trust rules available with EMET are configured with specific expiration dates that will de-activate each rule before the expiration of the protected SSL certificate.

Let's look at some rules from Popular Software.xml.

```
<App Name="Word" Path="*\OFFICE1*\WINWORD.EXE">
<Mitigation Enabled="true" Name="ASR">
<asr_modules>flash*.ocx</asr_modules>
</Mitigation>
</App>
```

This rule tells EMET to protect Microsoft Word with all default memory mitigations, and to add ASR to block the loading of any module matching "flash\*.ocx". By default, all mitigations, with the exception of ASR and EAF+ that require specific application settings, are enabled for all applications in a protection profile. This can be changed by editing the DefaultConfig node in the profile file.

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

This rule enables all mitigations for Windows Media Player, except Mandatory ASLR, EAF, and SEHOP. Another important information is the Path. We have "\*\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.

The full path name to the application must be specified. Wildcards can also be used, such as \* 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 not in the executable name itself. For instance "wmplayer.exe" or "\*\wmplayer.exe" are valid paths, while "\*player.exe" or "\*wmplayer.exe" are not. This is due to a limitation of the Application Compatibility Framework in Windows that EMET relies on.

If a newly-started process matches multiple path specifications, EMET chooses the rule with the longer path specifier. This approximates picking the most specific match.

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.

## EMET Graphical User Interface

One method of interacting with EMET is through the graphical user interface (GUI). It can be launched through the start menu/window icon created during the EMET installation. In this section the various windows and sections will be described.

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

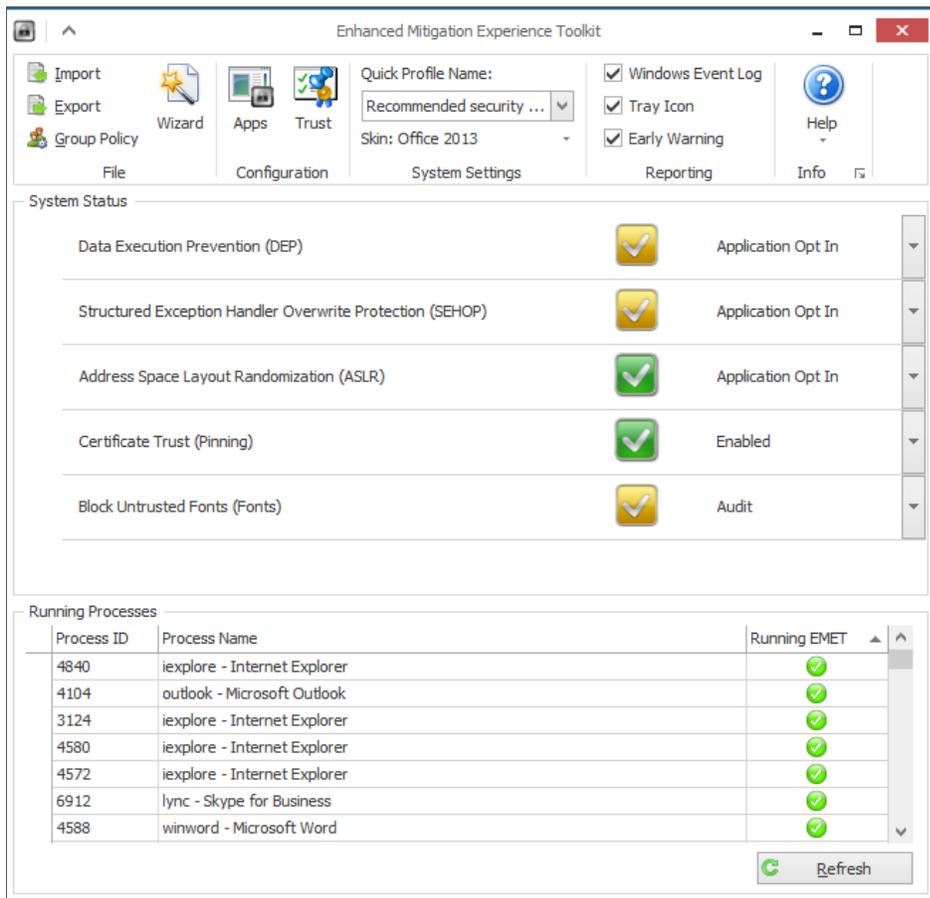


Figure 10: EMET GUI main window

The EMET GUI is divided in three sections. From the top to the bottom:

### Ribbon:

- **File:** This group allows to "Import" (*Ctrl+Shift+I*) or "Export" (*Ctrl+Shift+E*) EMET's configuration, and allows to run the EMET Configuration Wizard (*Ctrl+Shift+W*). See Configuration Wizard for additional details. This group also enables you to configure settings through Group Policy instead of through registry configuration. You must have Remote Server Administration Tools (RSAT) installed to configure local or Active Directory Group Policy.

- **Configuration:** This group allows to access the “Application Configuration” window by clicking on “Apps” (*Ctrl+Shift+A*), and the “Certificate Trust Configuration” window by clicking on “Trust” (*Ctrl+Shift+T*). See *Configuring Mitigations for Applications* and *Configuring Certificate Trust* (pinning rules) for additional details.
- **System Settings:** This group allows to apply a Quick Profile for the system, as well as select a Skin for EMET GUI. See *Configuring System-wide Settings* for additional details.
- **Reporting:** This group allows to toggle the Reporting options. See *Configuring Reporting* for additional details.
- **Help:** This group allows to access to help resources, such as the Support Forums, and the User Guide (*Ctrl+Shift+F1*), and to access to the EMET Privacy Statement.

**System Status:** This section shows the current status of system mitigations (DEP, SEHOP, and ASLR), and the status of the Certificate Trust feature. These settings can be changed directly from this section.

**Running Processes:** This section shows a list of currently running applications and which ones are protected by EMET. The application list is refreshed every 30 seconds, and can be manually refreshed by clicking on the “Refresh” button. Also, with the keyboard combination *CTRL+F* it’s possible to search for a specific application in the list.

### Configuration Wizard

The Configuration Wizard is displayed at the end of the EMET installation, and in case of a clean EMET installation it allows to either apply the recommended settings or to configure EMET manually. In case of an upgrade from a previous version of EMET, it allows to keep the previous EMET settings.

We strongly suggest to always apply the Recommended Settings for both new installation and upgrade scenarios, and then toggle EMET’s configuration as needed.

### Wizard options

**Use Recommended Settings:** This option deletes existing settings and apply the recommended settings:

- **Application Configuration:**
  - Adds protections for Internet Explorer, WordPad, Microsoft Office, Adobe Acrobat and Reader, and Oracle Java.
  - Configures EAF+ with Internet Explorer with the Microsoft Trident engine, the Adobe Flash plugin, the Microsoft VML plugin, the Microsoft VBScript engine, and the Microsoft JavaScript engine. Configures Adobe Acrobat and Adobe Reader with the Lotus Notes Field Exchange Module for Adobe Acrobat, the Adobe Reader engine, and the Adobe Acrobat Forms.
  - Blocks the Adobe Flash plugin from running in Microsoft Excel, PowerPoint, and Word, and blocks the Oracle Java, Microsoft VML, Microsoft MSXML 4.0, Windows Script Host Runtime, and Microsoft Scripting Runtime plugins from running in Internet Explorer in websites not belonging to the Trusted Sites or Intranet zones.
- **Certificate Trust:** Adds rules for Microsoft and other 3<sup>rd</sup> party popular online services.
- **Reporting:** Enables all Reporting mechanisms (Windows Event Log, Tray Icon, and Early Warning Program).

**(New installation) Configure Manually Later:** This option does not configure EMET.

**(Upgrade from previous version) Keep Existing Settings:** This option keeps the existing EMET configuration. Two optional settings, related to EMET's new features, can be automatically configured:

- **Certificate Trust:** Adds rules for Microsoft and other 3<sup>rd</sup> party popular online services.
- **Reporting:** Enables the Early Warning Program.

### Configuring System-wide Settings

It is possible to configure system wide settings in two different ways. It is either possible to select one of the two system mitigation profiles ("Maximum Security Settings" and "Recommended Security Settings") from the "System Settings" ribbon group or to set the mitigation configuration individually.

Please note some configuration changes will require rebooting the system. EMET GUI provides notification as needed.

The list of available system mitigations varies between different versions of Windows. This reason is that some system mitigations are not available on all versions of Windows. The paragraph Supported Operating Systems and software requirements contains more information about system mitigations support in different Windows versions.

The Certificate Trust feature can be enabled or disabled by changing the related entry. In addition, Internet Explorer must be added in the "Application Configuration".

### Configuring Mitigations for Applications

It is possible to configure specific applications to opt-in to the mitigations provided by EMET. Additionally, mitigations can be individually enabled or disabled on a per application basis.

For example, it is possible to configure iexplore.exe to opt-in to all EMET's mitigation and, at the same time, opt-in winword.exe only for SEHOP and Mandatory ASLR.

Note that checking the Fonts checkbox for an app *exempts* that app from the Windows 10 untrusted font mitigation. Doing so is discouraged, but may be valuable if you have a specific app that must use fonts that are not installed in the standard Fonts directory.

It is possible to Add (*Ctrl+Shift + Plus*) and Remove (*Ctrl+ Minus*) applications from the list by clicking the corresponding buttons. When adding an application, a user will get prompted with the regular open file dialog. An application can then be selected and added to the list. The "Add Wildcard" (*Ctrl+Shift +Asterisk*) button allows to configure an application by adding wildcards in its path.

It is also possible to enable/disable multiple mitigations by right clicking on the mitigation name column, or the application row.

The configuration is applied only when the "Ok" button is clicked.

### *Additional mitigations configuration*

It is possible to configure additional settings for EMET mitigations. These settings are reachable from the "Application Configuration" window.

The "Default Action" ribbon defines what action EMET will take when an exploit has been detected:

- **Stop on exploit:** EMET reports the exploitation attempt and terminates the process.
- **Audit only:** EMET reports the exploitation attempt and does not terminate the process. This mode is not applicable to all mitigations, since when some of them are detected the process is already in a state that cannot be recovered. The mitigations that support this feature are:
  - EAF

- EAF+
- ROP mitigations: LoadLib, MemProt, Caller, StackPivot, SimExecFlow
- SEHOP (only on Windows Vista and Windows Server 2008)
- ASR

The “Show All Settings” button in the “Options” ribbon allows to fine-tuning or to configure additional mandatory aspects of some mitigations. Currently the mitigations that can be configured through this panel are:

- **Heap Spray:** allows to configure the memory addresses to pre-allocate for this mitigation.
- **Simulate Exec Flow:** allows to define the number of simulated instructions to check.
- **EAF+:** allows to specify which modules are blocked by accessing the import/export tables of the protected application. This setting has no default values.

The “Modules” field define which modules are restricted from accessing the export and import table addresses of the protected application. The modules that should be defined here are ones that could be leveraged to resolve API’s during an exploit, and are different for each application. The default configuration contains the list of modules that have been observed being used, or that can be potentially being used, during exploitation of the protected application. Users can use wildcards in the module name, and specify multiple modules (maximum 8) by separating them with a semi-column (;). For example: module1.dll;module2.ocx;module3\*.dll

- **ASR:** allows to define the list of modules/plugins that are blocked from being loaded in the protected application. This setting has no default values and is mandatory to enable the mitigation.

The “Modules” field defines which modules will be blocked from being loaded within the protected application. Users can use wildcards in the module name, and specify multiple modules (maximum 8) by separating them with a semi-column (;). For example:  
module1.dll;module2.ocx;module3\*.dll

The “Internet Zone Exceptions” drop down defines for which Internet Security Zone the ASR does not apply (exclusion). This setting is validated only in applications that support the Internet Security Zones (i.e. the Trident engine). For example, selecting “Local Internet” and “Trusted sites” will allow to load the defined modules in these two zones.

### Configuring Certificate Trust (pinning rules)

To enable this feature the “Certificate Trust (Pinning)” must be enabled as described in the paragraph Configuring System-wide Settings, and the *explore.exe* process must be added in the list of protected applications, as described in the paragraph Configuring Mitigations for Applications. No other mitigations are required to be enabled to use the Certificate Trust feature.

It is possible to configure the SSL/TLS certificate pinning rules by clicking on the “Trust” (*Ctrl+Shift+T*) button in the “Configuration” ribbon group in the main EMET GUI window. From the “Certificate Trust Configuration” window it is possible to add or enumerate the websites that are protected (subject names of their SSL certificate) and assign an existing rule for each website. After clicking on “Add Website” (*Ctrl+Shift + Plus*) in the “Add / Remove” ribbon, type in the Fully Qualified Domain Name of the website as shown in its SSL certificate (NOTE: wildcards or other symbols are not accepted and the name must be unique within the list).

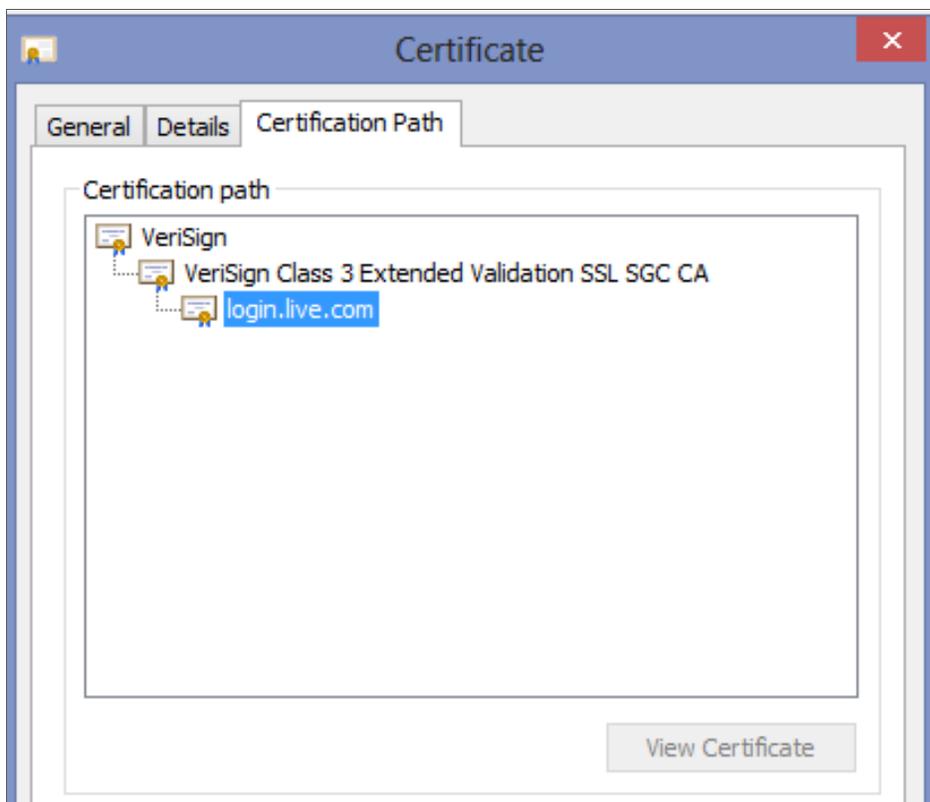


Figure 11: certificate trust chain for login.live.com

The next step is to assign a "Pin Rule" to that website. If there are no rules, click on the "Pinning Rules" tab. A window with the list of available rules will appear.

From this window it's possible to define the certificate pinning rules that can be assigned to websites. To generate a new rule, click on "Add Rule" (*Ctrl+Shift + Plus*) in the "Add / Remove" ribbon group and fill at least the first three parameters in the table with the appropriate values:

**Name:** is the unique identifier for the rule, to be accessed later from the "Protected Websites" tab.

**Certificates:** will open a window that allows to define and import a set of trusted Certification Authorities from the Trusted Root Certification Authorities folder in the User Certificate Store (certmgr.msc). It is possible to select one or more trusted Root CAs from this list. If a Root CA is not present in the list it will need to be imported in advance.

**Rule Expiration:** will establish when that rule will expire. When a rule is expired it will be ignored and a log event will be written at EMET Agent startup to notify the expiration of the rule.

**Blocking Rule:** if enabled, EMET will stop the connection when a mismatch of that rule is detected.

Note that EMET no longer uses the Minimum Key Size, Allowed Country, Blocked Hashes, and PublicKey Match attributes, even though they still appear in the EMET Certificate Trust Configuration dialog box.

Once the rule is defined, click on the "Protected Websites" tab and assign that rule to the desired websites. A website can have only one Pinning Rule, while a Pinning Rule can be assigned to multiple websites.

"Protected Websites" and "Pinning Rules" entries can be deleted by clicking on the entry in the table that needs to be deleted, and by clicking on "Remove Website" or "Remove Rule" (*Ctrl+Minus*) in the "Add / Remove" ribbon group afterwards. A pinning rule can be deleted only when not used by any website. The

protection for a specific "Protected Website" can be temporary disabled by unchecking the checkbox in the column "Active".

Once configured, if one of pinning rules is triggered while browsing, EMET will detect the SSL certificate not matching the configured rules, and will react according to the Reporting settings (the Early Warning reporting mechanism is not available for the Certificate Trust feature) and whether the triggered rule is a "Blocking Rule".

The EMET graphical user interface (EMET\_GUI.exe) provides an interface to configure the "Certificate Trust" entries. However, it is also possible to import a previously exported pinning rules configuration using either EMET\_GUI or EMET\_CONF.

An example of how to create Certificate Trust pinning rules is available at [this blog post](#) on the Security Research & Defense blog.

### Configuring Reporting

It is possible to configure the reporting of EMET alerts granularly. When EMET detects an exploitation attempt or a SSL certificate that violates one of the pinning rules, the EMET Service can be configured to perform one or more actions: writing to the Windows Event Log, display an alert to the user, and/or use the Early Warning Program. The Early Warning Program is only available for exploits detection.

It is possible to configure what actions EMET will perform when detecting an attack directly from the EMET GUI main window. The "Reporting" ribbon group contains three entries: Windows Event Log, Tray Icon, and Early Warning.

**Windows Event Log:** EMET writes to the Windows Events Log.

**Tray Icon:** The EMET Agent displays a pop-up that contains the details of the attack (targeted process and the mitigation used to detect and stop the attack).

**Early Warning:** EMET generates a set of information related to the attack, including a memory dump and the type of mitigation that has been used to detect and stop the attack, and sends this information to Microsoft through the standard Microsoft Error Reporting channel. Users have the opportunity to review the information sent to Microsoft in advance before the transmission occurs.

NOTE: please refer to Advanced Options for the advanced configuration of custom "Tray Icon" messages.

### Configuring Appearance

EMET offers the possibility to configure the look and feel of the EMET GUI and the various graphical components of both the EMET GUI and the EMET Agent. EMET's theme can be changed from the main EMET GUI window, by clicking on "Skin:" in the "System Settings" ribbon group.

### Accessibility

EMET GUI offers accessibility features that makes it more compliant with the ease of access features provided by Windows:

- Full keyboard navigation support
- Full High-Contrast support
- Full support for different text sizes, up to 200% larger than default
- Partial Narrator support

## EMET Command Line Tool

An alternative way to configure EMET is to use EMET\_Conf.exe. This command line utility can be found at the location where EMET is installed. Currently the command line tool does not allow to configure all EMET features. However, new sub-options have been added in EMET 5.5,, as shown below.

### Sub-options

They allow configuration of:

- ASR modules and zones
- EAF modules
- ROP Simulate Execution Flow number of simulated instructions

### Multiple commands at once

Multiple commands in one line. It is now possible to append any command in the same line. After syntax has been entirely and successfully verified, commands are executed sequentially from left to right.

Example usage:

```
EMET_Conf.exe --set prog1.exe +DEP -SEHOP --set prog2.exe -DEP +StackPoint -list
```

The above command changes settings for two programs then displays the new app configuration.

### Prefixes

Options can be prefixed by "--" (original EMET style), "-" (Unix style) or "/" (Windows style). Different prefixes can be used in the same command.

### Output and return value

By default, on success, EMET\_conf.exe returns 0 to the calling process and does not display anything.

Standard output is used for all commands that specifically request display (the "list" command, for example).

Error output (stderr) is use for everything else, mostly errors.

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

### Get help

```
EMET_Conf --help
```

```
EMET_Conf /?
```

Display a usage screen including all currently supported application specific mitigations as well as the supported system mitigations.

Running the EMET Command Line Tool without any arguments has the same effect than command help.

Example usage:

```
"EMET_Conf -help
```

### Parse only (no execution)

```
EMET_Conf -- parseonly
```

Syntax verification without execution. Can be set anywhere in the command, except between an option and its dependent sub-options.

Example usage:

```
"EMET_Conf -set myprog.exe +DEP -SEHOP -parseonly
```

### Add/modify an application to EMET

```
EMET_Conf --set [--force] < path to executable> [(+|-)Mitigation ... [modules|zones|number] ...]
```

<path to executable> can be the full path name to the application. Wildcards can also be used, namely \* or ?.

Another option is to just use the executable name without the path, such as wmpayer.exe.

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

Sub-options:

- Option "(+|-)ASR has two sub-options: --modules and --zones  
Example usage: *"EMET\_Conf /set prog.exe +EAF -modules malware.dll -zones "1;2" +DEP*
- Option "(+|-)EAF has one sub-option: --modules  
Example usage: *"EMET\_Conf /set prog.exe +ASR -modules suspicious.dll*
- Option "(+|-)SimExecFlow has one sub-option: --number (of simulated instructions)  
Example usage: *"EMET\_Conf /set prog.exe +SimExecFlow -number 12*

More example usages:

*"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, showing the settings configured locally (EMET\_GUI or EMET\_CONF) first, followed by the settings configured via Group Policy.

### List which system mitigations have been enabled by EMET

```
EMET_Conf --list_system
```

Display all the system mitigation settings, showing the settings configured locally (EMET\_GUI or EMET\_CONF) first, followed by the settings configured via Group Policy.

### List the Certificate Trust configuration

```
EMET_Conf --list_certtrust
```

Display all the Certificate Trust websites and pinning rules configured locally (EMET\_GUI or EMET\_CONF).

#### 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_apps
```

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

#### Remove all certificate trust configuration

```
EMET_Conf --delete_certtrust
```

This will remove all the Certificate Trust configuration from EMET.

#### Remove all EMET configuration

```
EMET_Conf --delete_all
```

This will remove all the EMET application mitigation settings and certificate trust configuration. It is equivalent to running both "--delete\_apps" and "--delete\_certtrust"

#### 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 the paragraph Advanced Options. By default unsafe options are not visible through either the command-line utility or the UI. The --force option is also needed when the system-wide DEP mitigation's configuration is changed and BitLocker is installed.

#### Modify advanced mitigation options

```
EMET_Conf --deephooks (enabled|disabled)
```

```
EMET_Conf --antidetours (enabled|disabled)
```

```
EMET_Conf --eafplus (enabled|disabled)
```

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

```
EMET_Conf --import [--force] <xml file>
```

Imports previously exported settings. This command can also be used to import and enable a Protection Profile or the entire configuration for Certificate Trust feature, e.g. `EMET_Conf --import "Deployment\Protection Profiles\Popular Software.xml"`

The --force option is needed to import an xml file that sets a mitigation to an unsafe state. For more information on this, refer to the paragraph Advanced Options. The --force option is also needed when the system-wide DEP mitigation's configuration is changed and BitLocker is installed.

```
EMET_Conf --export <xml file>
```

Exports the current configuration to the specified xml file.

#### Configuring the reporting settings

```
EMET_Conf --reporting (+|-)(telemetry|eventlog|trayicon)
```

This switch configures the way reporting takes place. The settings that can be toggled with this command are the following:

- **eventlog:** this keyword will turn on or off the recording of the attack in the Windows events system.
- **trayicon:** this keyword will turn on or off the visual notification for the user.
- **telemetry:** this keyword will turn on or off the Early Warning Program system.

An example on how to use this command is the following:

```
EMET_Conf --reporting -telemetry +eventlog +trayicon
```

Configuring the exploit action settings

```
EMET_Conf --exploitaction (audit|stop)
```

This switch configures how EMET should behave when an exploit happens:

- **Audit:** do not kill the process, when applicable, but just log the exploitation attempt
- **Stop:** Terminate the program when an exploitation attempt is detected

Configuring the visibility of EMET Agent icon in tray area

```
EMET_Conf --agentstarthidden (enabled|disabled)
```

This switch configures the visibility of the EMET Agent icon in the tray area.

## Deploying EMET

With EMET enterprises can take advantage of their existing management infrastructure 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.

### Microsoft System Center Configuration Manager

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

Creating the Application to Deploy EMET to Clients

The first step in deploying EMET is to download the EMET installer. Once the MSI package has been obtained, the steps below must be followed. 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 has been previously downloaded<sup>(1)</sup>.

---

<sup>1</sup> More information and the downloadable Configuration Manager packages can be found at the Configuration Manager Team Blog [here](#).

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, it is possible to add any additional details for this application, and a pre-populated command will be shown 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 just created application, select Deploy.
8. Browse to the collection to target.
9. On the content page, choose the distribution points.
10. On the deployment settings page, choose the intended install settings (most likely this will be required, unless it is just a test deployment).
11. Configure the deployment scheduled, user experience, and alerts, then complete the wizard.
12. The process of deploying the EMET client silently to all targeted clients has now started. Its progress can be monitored in Monitoring | Deployments.

### Creating the Package and Program to Configure EMET

Now that EMET is deployed, it must be configured for to protect the applications in the environment. 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. 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 to evaluate this collection. We will be targeting EMET configuration to this collection, so evaluate it as often as needed. Also, it has to be kept 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 will be used as the source for the EMET configuration package. These files can be gathered from the source directory of the EMET client after it has been installed on a system. If all of the files are not included EMET configuration will not work.
  - a. Popular Software.XML (from the applications folder \EMET\Deployment\Protection Profiles)
  - b. EMET\_Conf.exe (from the applications folder \EMET)
  - c. HelperLib.dll (from the applications folder \EMET)
  - d. MitigationInterface.dll (from the applications folder \EMET)
  - e. PKIPinningSubsystem.dll (from the applications folder \EMET)
  - f. SdbHelper.dll (from the applications folder \EMET)
2. From Software Library | Packages choose to Create Package.
3. Name the package, and choose this package containing the source files. Provide the path where the four files referenced in step 1 are sourced.
4. Choose standard program.
5. Name the program, and set the command line to be EMET\_Conf.exe --import "Popular Software.xml". This is just an example, using the "Popular Software" protection profile provided by the EMET team. It is possible to modify this profile or use one of the other protection profiles provided by EMET. The file to be imported needs just to be referenced and included in the 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 just created collection as the target collection, and complete the wizard with the desired settings.

### Group Policy

To deploy EMET installer through Group Policy, please follow the procedure described in [Microsoft KB Article 816102](#).

Group Policy administrators can configure EMET settings through Group Policy using the EMET GUI and using standard Group Policy editing tools. The EMET GUI is easier to use, but both options are available. Settings configured through Group Policy override settings configured in the local registry.

To use the EMET GUI to configure Group Policy, click the Group Policy button and select the policy store to configure, or click New to create a new one. EMET will open the Group Policy EMET Configuration dialog box, which is very similar to the main EMET GUI. The primary difference is that settings that you configure are written to Group Policy Objects rather than to the local registry. Note that the Import and Export buttons enable you to "transfer" configuration sets between Group Policy and regular registry configuration through XML files. Also note that there are some features that can be configured through

Group Policy using Windows Group Policy editors that are not exposed through the EMET GUI's Group Policy configuration GUI.

Once EMET is deployed, Group Policy administrators can use the provided template files for configuration. The EMET.admx and EMET.adml files are located in the "Deployment\Group Policy Files" folder. These files must be copied onto \Windows\PolicyDefinitions and \Windows\PolicyDefinitions\en-US folders respectively. Once this is done, EMET settings can be configured via Group Policy.

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

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

**Default Protections:** These are predefined sets of protection settings for groups of applications. The three profiles do not include each other, therefore to enable the most comprehensive list of predefined application protections you need to enable all of these. Enabling Internet Explorer and Recommended Software corresponds to the configuration that the Configuration Wizard applies when EMET is installed.

- **... for Internet Explorer:** Enables default recommended protections for Internet Explorer.
- **... for Recommended Software:** Enables default recommended protections for WordPad, applications that are part of the Microsoft Office suite, Adobe Acrobat, Adobe Reader, and Oracle Java.
- **... for Popular Software:** Enables default recommended protections for other popular software.

Note that the Default Protection options are *not* exposed in the EMET GUI Group Policy Configuration dialog. The settings that are preconfigured in the Default Protection options are the same ones that are in the *Popular Software.xml* and *Recommended Software.xml* protection profiles. If you import application protection profiles in the EMET GUI's Group Policy Configuration, they are added to Application Configuration, described next. If the same app is configured in Application Configuration and in one of the Default profiles, the Application Configuration settings take precedence.

**Application Configuration:** This leads to a two-column text editor where additional applications not part of the default protection profiles can be configured. For each row, the left column specifies the matching path rule for the application, and the right column specifies non-default configuration options. If you leave the right column empty, all the default mitigations are applied. You can enable or disable individual mitigations and supply parameters for ASR and EAF+ using the syntax described in the Help text displayed in the Windows Group Policy editor.

Note that if you import a protection profile in the EMET GUI Group Policy Configuration dialog, EMET adds the profile's settings to Application Configuration. You can edit the details of those settings in the EMET GUI or in the Windows Group Policy editor, although the former is easier. Application configuration settings are compatible with Group Policy definition schemas (ADMX), which enables reporting and compliance checks.

**Default Action and Mitigation Settings:** These settings are related to the advanced settings for the ROP mitigations, described in the paragraph Advanced Mitigations for ROP, and for the default action when an exploit is detected (Audit only or Stop).

**Certificate Pinning Configuration:** This setting configures EMET certificate pinning configuration through two separate two-column text editors. The first text editor specifies which sites to pin and the name of the rule to associate with each site. The second text editor associates rule names with the details

of their respective rules. The syntax for each is described in the Help text displayed in the Windows Group Policy editor. It is strongly recommended to use the EMET GUI to configure certificate pinning rather than the Windows Group Policy editor, as the former is far easier to use. Certificate pinning configuration settings are compatible with Group Policy definition schemas (ADMX), which enables reporting and compliance checks.

**EMET Agent Visibility:** This setting allows to automatically hide the EMET Agent icon in the tray area of the taskbar.

**EMET Agent Custom Message:** This entry allows to define a customized message that will be displayed in the alert that is shown when EMET detects an attack. The Tray Icon reporting setting must be turned on to display this message.

**Reporting:** This entry allows to toggle the reporting configuration for the Windows Event Log, the Tray Icon, and the Early Warning Program.

Once EMET Group Policies are enabled, they will be written out to the registry at *HKLM\SOFTWARE\Policies\Microsoft\EMET*. This registry key is monitored by the EMET Service, which will automatically apply the configuration locally.

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

```
EMET_Conf --list
```

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
```

will only clear the mitigations and SSL certificate pinning rules that have been defined through the EMET GUI or EMET\_Conf. The mitigation settings and SSL certificate pinning rules defined via GPO will be intact.

### Other Options

If using a different management solution not relying on either System Center Configuration Manager or Group Policy, it is recommended to leverage the Protection Profiles feature presented in the paragraph EMET Protection Profiles.

## 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. It is still possible to configure these options by overriding a registry key. 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 this key is not present, run the EMET GUI and refresh the 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, unsafe options can be selected.

With EMET, there is currently one unsafe option: the “Always On” setting for the system ASLR setting. Depending on the operating system configuration, setting the system ASLR setting to “Always On” could make the operating system to crash at boot time. Recovering from this will require booting the system in safe mode and setting the system ASLR setting to either “Opt In” (recommended) or “Disabled”.

### Configuring custom message for user reporting

It is possible to configure a custom message for the reporting pop-up when an attack is detected. For EMET this setting can be configured via Group Policy or by creating a registry key.

In the hive *HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\EMET* create a new “String Value” named *TrayIconMsg*. The string specified in this will be displayed to the user in case EMET detects and stops an attack instead of the default notification.

### Configuring Certificate Trust feature for third party browsers

Advanced users can configure third party browsers to benefit the mitigation offered by the Certificate Trust feature. The browsers will need to use the Windows CryptoAPI and support the CAPI extensions. Furthermore, the third party browser must be added to the protected applications (even with no mitigations). Finally, the executable name of the third party browser must be appended to the registry value “EMET\_CE” in the registry hive *HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\EMET*, separated by “;”. For example “iexplore.exe;third\_party\_browser.exe”.

NOTE: This scenario is unsupported and experimental. Technically speaking, any program that uses Microsoft’s CryptoAPI for SSL chain trust validation can be enlisted in that registry value so that it works with EMET’s certificate trust feature.

### Configuring Local Telemetry

For troubleshooting purposes, we have added a “Local Telemetry” mode. When this mode is enabled, the information that would be sent through the “Early Warning” will be saved locally instead in a user-defined folder. To enable this mode, users need to create two entries in the registry hive *HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\EMET*:

- LocalTelemetryPath (string): path where to save the information (i.e. c:\emet\_local\_telemetry)

Optionally, you can create the following registry key to control what kind of MiniDump file to create:

- MiniDumpFlags (DWORD): 0x1ff (default value)

More information on the possible flags are available at this [MSDN article](#).

### Configuring EMET Agent icon visibility

Users can configure the visibility of the EMET Agent icon in the tray area by adding a DWORD registry key *AgentStartHidden* in the registry hive *HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\EMET*. The possible values are 1 to hide the icon and 0 to show it.

## Mitigation Caveats

There are a few things to consider 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.

## System Settings

### DEP

- 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 entering the recovery key at the next reboot will be required. It is highly recommended to have the recovery key ready before changing the system configuration setting for DEP on a system with BitLocker enabled.
- 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 the system when configuring DEP.

### SEHOP

- On Windows 7 and later versions, SEHOP (both system wide and per application) is implemented by the operating system. For this reason, when this mitigation is enabled and is detected, EMET will not be able to catch and notify that SEHOP was detected. Instead, the OS will terminate the process and write an event in the Applications event log.

### ASLR

- 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 the paragraph Advanced Options.

## Application Specific Settings

### DEP

- 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 the system when configuring DEP.

### SEHOP

- Various applications on Windows Vista and above are not compatible with EMET’s SEHOP, in this case it is advisable to disable SEHOP from EMET and use the System Mitigation’s SEHOP. Configure the system mitigation SEHOP to Applications Opt-Out

### EAF

- 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.

- 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.
- EAF mitigation should not be applied to: programs and libraries protected that use packers or compressors, DRM or software with anti-debugging code, debuggers, and security software such as antivirus, sandbox, firewalls, etc.

### Mandatory ASLR

- On Windows 8 and newer versions, some programs (e.g. Internet Explorer 11 or Office 2013) are configured to run with enforced ASLR natively provided by the operating system. For this reason, when the native ASLR from Windows is enabled on an application, EMET will not be able to provide notification and reporting for this mitigation. Instead, the OS will terminate the process and write an event in the Applications event log.
- 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.

## Frequently Asked Questions

### Lifecycle Policy

Are older versions still supported?

As of EMET 5.5, we no longer support EMET 4 or previous versions. EMET 5.5 will be supported until January 2017.

### EMET 4 Questions

Are my configurations from EMET 4 and EMET 5.2 compatible with EMET 5.5?

No. In EMET 5.5, the registry has been refactored to make it easier to manage. To convert settings from previous versions of EMET (including EMET 5.5 Beta), you must use the export feature to save your settings from the previous version of EMET, and then import the settings back with the use of the converter [PowerShell script](#) after EMET 5.5 is

I have EMET 4 installed. Should I uninstall it before installing the new version?

You don't need to uninstall EMET 4 before installing EMET 5.5. EMET 5.5 installer will automatically uninstall EMET 4 and install EMET 5.5. When upgrading, in order to keep your current settings, you must use the export feature to save your settings from the previous version of EMET, and then import the settings back with the use of the converter [PowerShell script](#) after EMET 5.5 is

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.

What applications should I apply EMET mitigations to?

We suggest to apply EMET mitigations to the applications that are typically targeted by exploits, which are ones that handle files or data coming from an untrusted source. Examples include web browsers, document readers, etc. Certain type of applications, like debuggers or applications that handle DRM files, are known to not get along very well with EMET. If you want to apply EMET mitigations to this type of applications expect to disable some mitigations. The [Knowledge Base article 2909257](#) provides guidelines related to EMET mitigations.

### 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 suspend BitLocker, apply the change and reboot the machine. After rebooting, you can resume it. 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:

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

One of my applications always 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 always 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.

### 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 4.0 is installed on the system. If you get this error after copying the binaries from another machine, try running the installer on the local machine. It will direct you to a location where you can download the .NET 4.0 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 the paragraph Supported Operating Systems and software requirements for more details.

I have an old version of EMET installed. How do I upgrade to EMET 5.5 ?

If you have a version of EMET older than 3, it is recommended that you first uninstall the old version via the Windows Control Panel, and then manually delete the HKLM\Software\Microsoft\EMET and HKLM\Software\Policies\Microsoft\EMET keys.

If you have EMET 3 or EMET 4 and want to upgrade to EMET 5.5 , you can run the installation package and follow the instructions to either maintain your old configuration or apply the recommended one.

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

Testing was done only for the applications included in the default Protection Profiles, in their default configuration. For any other applications or non-standard configurations, it is recommended to thoroughly test them in a dedicated environment prior to deploying EMET on production systems.

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.

## Support

Customers with a Premier or Professional support contract can leverage these channels to receive support.

Users can send email to [emet\\_feedback@microsoft.com](mailto:emet_feedback@microsoft.com) with feedback and suggestions. Please don't use this email with support requests, use the TechNet Forums or the official support channels instead.

# Appendix A: 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 to perform application compatibility testing on applications prior to deploying EMET protections for them.

With EMET, application compatibility testing was done on all the Microsoft and 3<sup>rd</sup> party applications that are part of the EMET protection profiles on all the supported platforms.

The known compatibility issues that some of the EMET mitigations have with certain applications are documented in the [Knowledge Base article 2909257](#), in the “Application compatibility list” section. The list is constantly updated and it is based to the latest publicly available version of EMET.

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 through the contact information in the paragraph Support to let us know of any incompatibilities that you might have encountered.

# Appendix B: EMET 5.5 Release Notes

Main changes brought with EMET 5.5:

- Full-featured GPO management, compatible with reporting and compliance requirements
- Command line: new syntax and options
- Implementation of certificate pinning now based on root CA thumbprints. Exceptions logic removed.
- Export and Import now memorize path
- EMET registry has been refactored. To convert settings from previous versions of EMET (including EMET 5.5 Beta), registry values must be saved in a file then imported back with the use of the converter [PowerShell script](#) after EMET 5.5 is installed. Here are the steps to follow:
  - a. Export settings. With elevated PowerShell, run the following command:  
**.\Migrate-EmetSettings.ps1 -RegFile .\NewEmetSettings.reg -MissingCertCsv .\MissingCerts.csv**  
PowerShell script Migrate-EmetSettings.ps1 is provided with EMET 5.5 RTM. It includes documentation about its usage.
  - b. Uninstall former version of EMET.
  - c. Install EMET 5.5 RTM. When asked to choose between “Use recommended settings” and “Configure manually later”, chose option “Configure manually later”.
  - d. Import settings. With elevated PowerShell, run the following command:  
**reg.exe import .\NewEmetSettings.reg**

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2014 Microsoft Corporation. All rights reserved.

Microsoft, list Microsoft trademarks used in your white paper alphabetically are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.