Enhanced Mitigation Experience Toolkit 4.0

User Guide

Microsoft Corporation

Table of Contents

1   Introduction

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

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

Security mitigation technologies are designed to make more difficult for an attacker to exploit vulnerabilities in a given piece of software. EMET allows to leverage these technologies on their system and provides 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 4.0 introduces a new feature to enforce a set of pinning rules that can verify SSL certificates of specified domains against their issuing Root CA (configurable certificate pinning).

- **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.

1.1 Capabilities

EMET 4.0 allows to both configure the system policy for mitigations as well as to configure mitigations on a per executable basis. Furthermore, EMET 4.0 introduces the capability of validating SSL certificates against a set of configurable "pinning" rules and 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 these two options 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 pinning rules to validate digitally signed certificates (SSL certifcates) while browsing. These rules are designed to bind specific domains' SSL certificates with the corresponding 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.

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.

1.2   Mitigations

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

1.2.1   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 on any platform back through Windows XP. For more information, take a look at the SEHOP Overview and Windows 7 SEHOP Changes blog posts.

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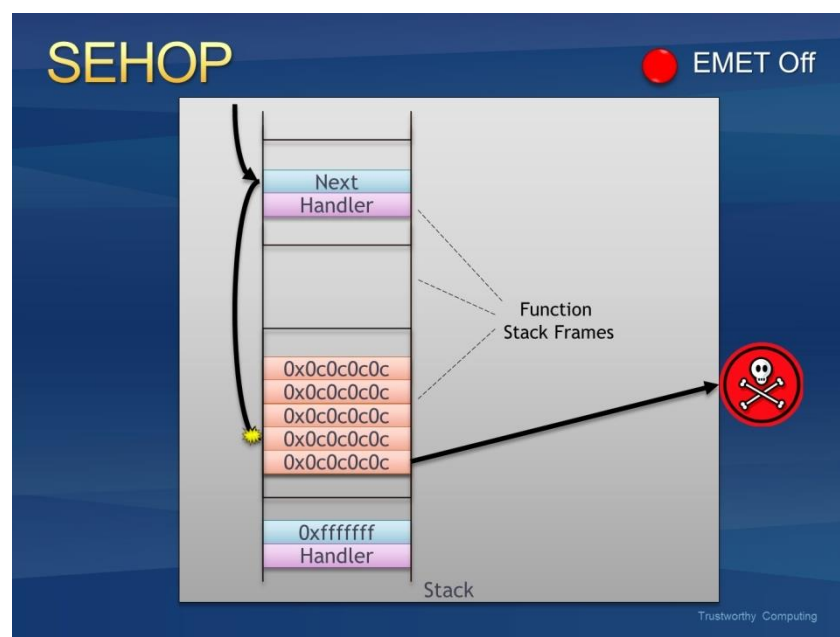
Figure 1: An exception handler hijack

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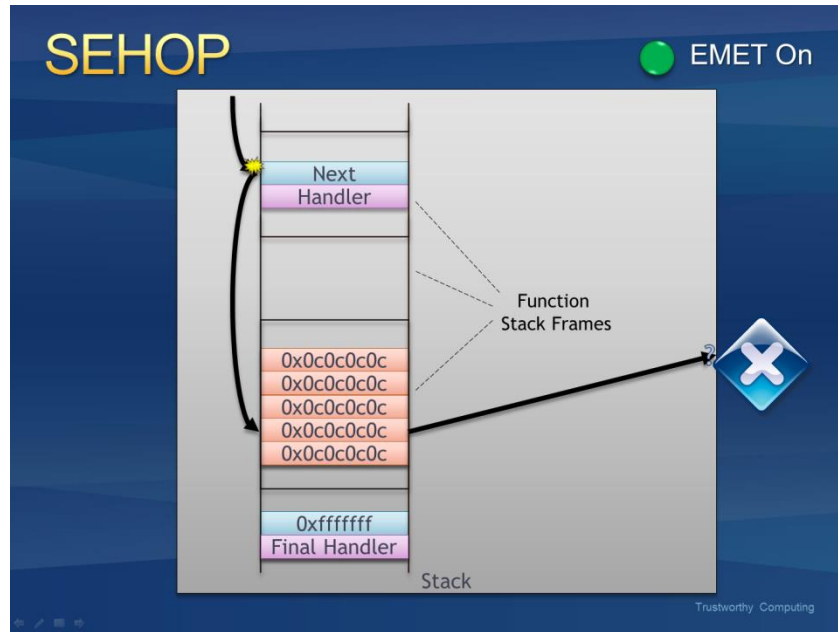
Figure 2: EMET stopping an exception handler hijack

1.2.2 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.
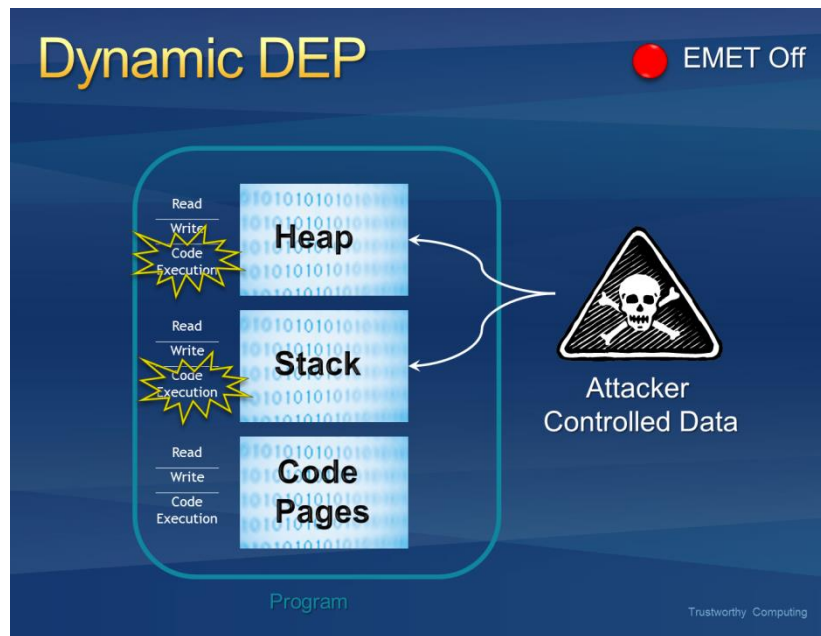
Figure 3: Running shellcode from attacker controlled locations

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.
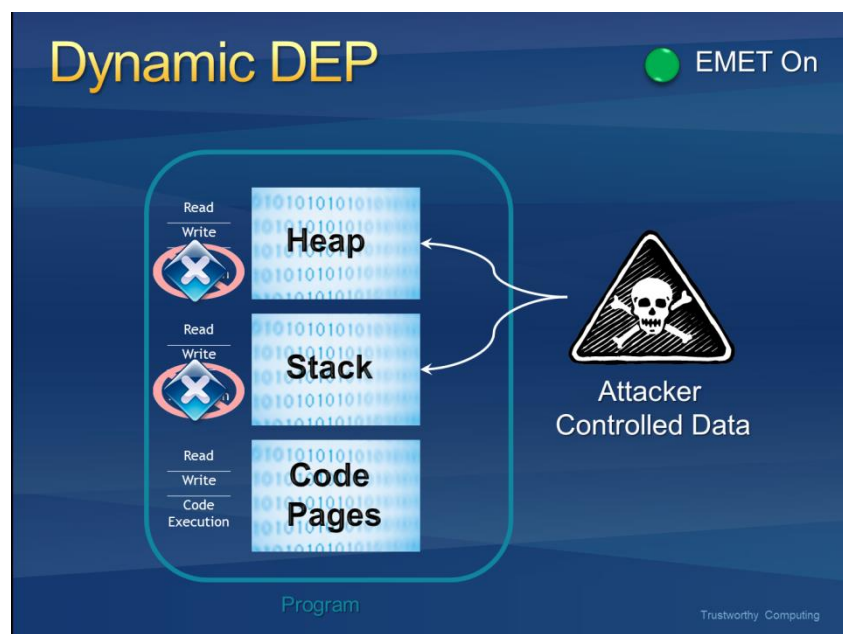
Figure 4: DEP blocking shellcode from running

1.2.3   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.
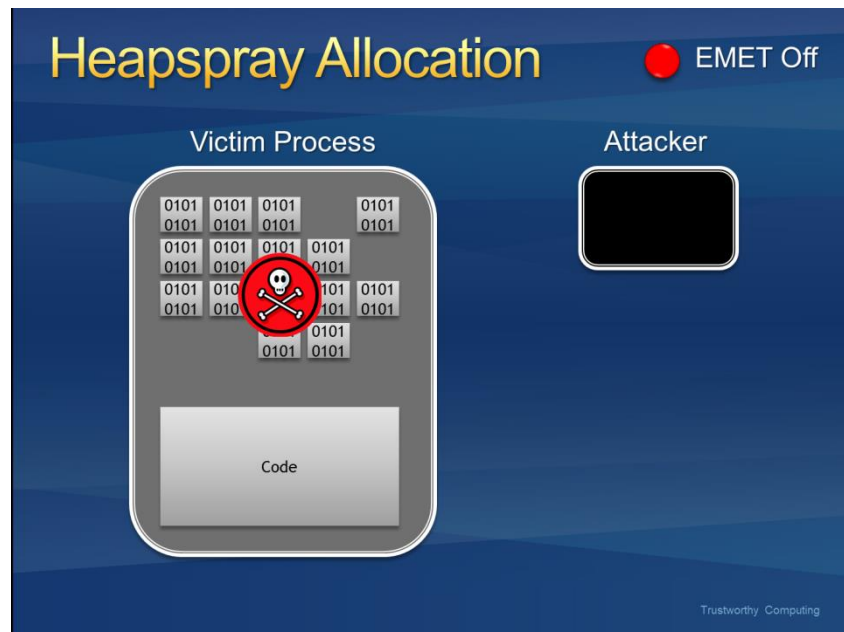
Figure 5: Using heapspray in an exploit

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.
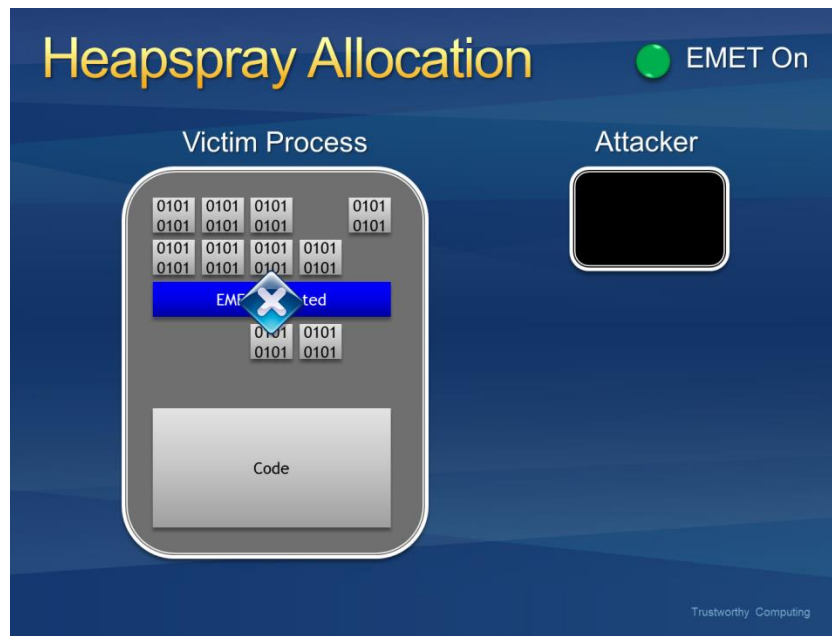
Figure 6: Blocking an attack that uses heapspray

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.

1.2.4   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.

1.2.5   Mandatory Address Space Layout Randomization (ASLR)

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

Without EMET in place, attackers can take advantage of a predictable mapping of those dlls and could use them in order to bypass DEP though a known technique called return oriented programming (ROP).
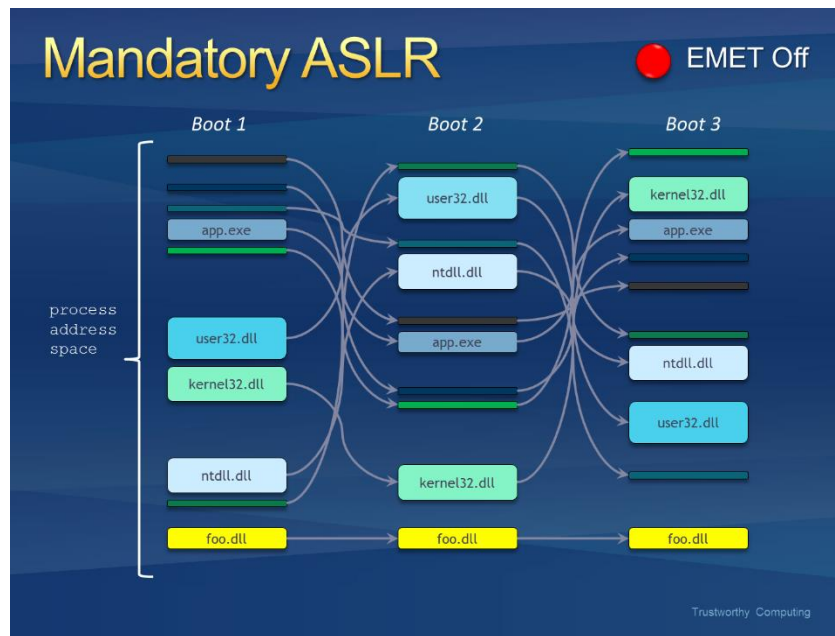
Figure 7: A module being loaded at a predictable location

With EMET in place, we force modules to be loaded at randomized addresses for a target process regardless of the flags it was compiled with. Exploits using ROP and relying on predictable mappings will fail.
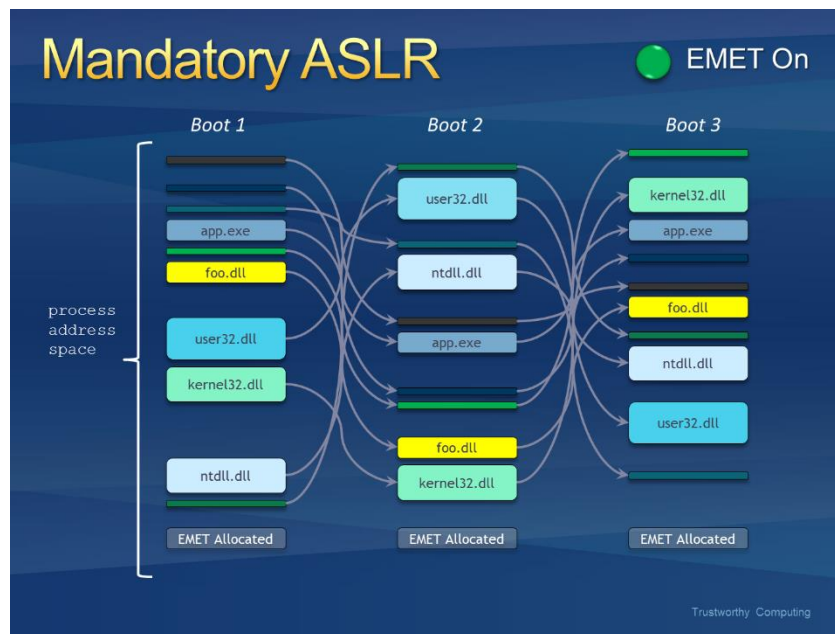
Figure 8: A module being forced to load at a random address

1.2.6   Export Address Table Access Filtering (EAF)

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

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

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, DRM, and unpackers.

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

1.2.7   Bottom-up randomization

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

1.2.8   ROP mitigations

EMET 3.5 Technical preview introduced several experimental anti Return Oriented Programming (ROP) mitigations that aim to block any exploitation relying on this technique. ROP is an exploitation technique that facilitate the execution of code in presence of mitigation like the Data Execution Prevention. In order to do that, the ROP technique use snippets of code that are already present in the application. With EMET 4.0 these mitigations have been enhanced, and many compatibility and performance issues have been solved.

Please note that ROP mitigations are only available and applicable to 32-bit processes. 64-bit processes are not protected with ROP in this version of EMET.

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

- **Load library checks:** EMET will monitor all calls to the LoadLibrary API and prevent loading libraries from UNC path (i.e. \\evilsite\bad.dll). It is possible to disable this option if a program is known to legitimately load DLLs from UNC path or remote servers
- **Memory protection checks:** EMET will disallow making the stack area executable. Such activity is usually used by shellcode or ROP gadgets.

- **Caller checks:** EMET will make sure that when a critical function is reached, it is reached via a call instruction rather than a "RET". This is a very useful mitigation and breaks many ROP gadgets. This mitigation may be incompatible with some programs.
- **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 programs.
- **Stack pivot:** This mitigation is used to detect if the stack has been pivoted. It is compatible with most programs.

1.2.9   Advanced Mitigations

EMET 4.0 introduces additional mitigation options that apply to all configured software. The additional mitigations introduced for this version are only for the ROP mitigations, and when enabled or disabled they will affect all the programs that have at least one ROP mitigation configured in EMET.

Following is a summary of what these advanced mitigations are:

- **Deep hooks:** EMET will protect critical APIs and the subsequent lower level APIs used by the top level critical API. For example, EMET will not only hook and protect *kernel32!VirtualAllo*c but also the related lower level functions, such as *kernelbase!VirtualAlloc* and *ntdll!NtAllocateVirtualMemory*.
- **Anti detours:** 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 will not be effective.
- **Banned functions:** By enabling this option, additional APIs, configured in a list, will be blocked when used. In this version only *ntdll!LdrHotPatchRoutine* is configured to mitigate potential exploits abusing this API.

1.3   Certificate Trust (configurable certificate pinning)

EMET 4.0 provides also a new 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, EMET will validate the end-entity SSL certificate and the Root CA that issued that certificate against the corresponding pinning rule configured by the user.

Depending on the configured rules for a specific domain, EMET will detect when a variation of the issuing Root CA for a specific SSL certificate occurs. Note that EMET will only detect the anomaly, but the connection will not be stopped. EMET 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. If a matching certificate is found, EMET verifies that the issuing Root CA of this certificate is one of the Root CA chosen by the user. A set of trusted Root CA can be defined by importing their

certificates only from the Windows [Trusted Root Certification Authorities](#) store. Once imported, pinning rules can be created to associate the SSL certificate subjects to a specific set of Root CA certificates.

Exceptions for each pinning rule can also be added. With these exceptions it's possible to have less restrictive rules, allowing EMET to accept SSL certificates even if the pinning rule doesn't match. Exceptions are related to some properties of the Root CA certificate, such as key size, hashing algorithm, issuer country, and public key component.

1.4   Reporting

EMET 4.0 has reporting capability provided through an additional component called the EMET Agent, which replaces and enhances the EMET Notifier of EMET 3. Once EMET is installed, this new component is set to automatically start with Windows. It will show up in the system tray area of the taskbar with an EMET icon and can be hidden any time or configured to run hidden in a permanent way (via Group Policy).

EMET Agent is a required component to perform 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 levels of logging: Information (ID 0), Warning (ID 1), and Error (ID 2). 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.

- **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 or details about the untrusted SSL certificate on the current HTTPS connection.
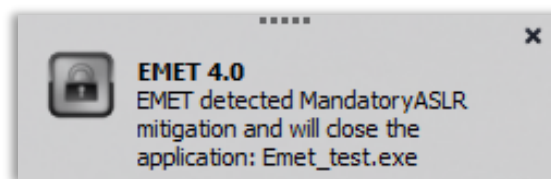
Figure 9: EMET Agent notification

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

- **Send reports for the Early Warning Program:** EMET 4.0 introduces 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 0day 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 will also send 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, and at http://aka.ms/emet40ps, for more information on the type of data that will be sent to Microsoft.

NOTE: EMET's reporting features are only available on Desktop applications. Modern applications cannot leverage this feature.

1.5 Supported Operating Systems and software requirements

1.5.1   Supported Operating Systems and Applications

EMET 4.0 supports the following operating systems and service pack levels:

Client Operating Systems
- Windows XP service pack 3 and above
- Windows Vista service pack 1 and above
- Windows 7 all service packs
- Windows 8

Server Operating Systems
- Windows Server 2003 service pack 1 and above
- Windows Server 2008 all service packs
- Windows Server 2008 R2 all service packs
- Windows Server 2012

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

Table 1: system mitigations compatibility matrix

	Mitigation	XP	Server 2003	Vista	Server 2008	Win7	Server 2008 R2	Win8	Server 2012
System Mitigations	DEP	✔	✔	✔	✔	✔	✔	✔	✔
	SEHOP	✘	✘	✔	✔	✔	✔	✔	✔
	ASLR	✘	✘	✔	✔	✔	✔	✔	✔
Application Mitigations	DEP	✔	✔	✔	✔	✔	✔	✔	✔
	SEHOP	✔	✔	✔	✔	✔	✔	✔	✔
	NULL Page	✔	✔	✔	✔	✔	✔	✔	✔
	Heap Spray	✔	✔	✔	✔	✔	✔	✔	✔

Mandatory ASLR	✕	✕	✔	✔	✔	✔	✔	✔
EAF	✔	✔	✔	✔	✔	✔	✔	✔
Bottom-up	✔	✔	✔	✔	✔	✔	✔	✔
Load library checks	✔	✔	✔	✔	✔	✔	✔	✔
Memory protection checks	✔	✔	✔	✔	✔	✔	✔	✔
Simulate execution flow	✔	✔	✔	✔	✔	✔	✔	✔
Stack pivot	✔	✔	✔	✔	✔	✔	✔	✔

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

Table 2: application mitigations compatibility matrix

Application Mitigations

Mitigation	32 bit Processes	64 bit Processes
DEP	✔	✔ (already mandatory)
SEHOP	✔	✕
NULL Page	✔	✔
Heap Spray	✔	✔
Mandatory ASLR	✔	✔
EAF	✔	✔
Bottom-up	✔	✔
Load library checks	✔	✕
Memory protection checks	✔	✕
Simulate execution flow	✔	✕
Stack pivot	✔	✕

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 available for Internet Explorer only but it can be configured for certain other browsers with an experimental setting. Refer to the following table for details on which versions of Internet Explorer this feature is available for:

Table 3: application compatibility for Certificate Trust

	Desktop	Modern (Windows 8)
Internet Explorer 32-bit	✔	✕
Internet Explorer 32/64-bit	✔	✕
Internet Explorer 64-bit (Enhanced Protected Mode or pure 64-bit)	✔	✕

1.5.2   Software requirements

EMET 4.0 requires .NET Framework 4. Also, 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 must be installed.

2   Configuring EMET

Without configuring EMET, there is no benefit from having it installed. 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 Sections 2.2 and 2.3 of this guide for further instructions on how to use those tools to achieve this.

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

Another option for configuring EMET is using Protection Profiles. The installation process of EMET 4.0 installation process will configure EMET with the Popular Software Profile and the Certificate Trust Profile. Refer to paragraph 2.1 for details on what is contained in these Protection Profiles.

A new method to configure EMET 4.0 is to use the new Configuration Wizard. At the end of the installation, a Configuration Wizard will offer 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 paragraph 2.2.1.

EMET configuration is saved in the registry sub-key *HKLM\SOFTWARE\Microsoft\EMET* and some limited user-specific settings are saved also in *HKCU\SOFTWARE\Microsoft\EMET*.

2.1   EMET Protection Profiles

EMET 4.0 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 3rd 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 supported versions of Microsoft Internet Explorer, WordPad, applications that are part of the Microsoft Office suite, Adobe Acrobat 8-11, Adobe Reader 8-11, and Oracle Java 6 and 7.
- **Popular Software.xml:** Enables mitigations for common applications, including Microsoft Internet Explorer and the Microsoft Office suite.
- **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 4.0 protection profiles have been updated with an optimized configuration that take into consideration the limited compatibility issues of some software. Certificate Trust rules shipped with EMET 4.0 are configured with specific expiration dates that will de-activate each rule one day before the expiration of the protected SSL certificate.

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

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

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

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

With this rule, we enable all mitigations for Windows Media Player, except Mandatory ASLR, EAF, and SEHOP. Another important information is the Path. We have for instance "*\Windows Media Player\wmplayer.exe". The path is what EMET uses to register its mitigations for an application. It has to match the target application's path for the mitigations to be effective.

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 are not valid in the executable image name itself. For instance "wmplayer.exe" or "*\wmplayer.exe" are valid paths, while "*player.exe" or "*wmplayer.exe" are not. This is due to a limitation of the Application Compatibility Framework in Windows that EMET relies on.

The protection files are well commented themselves. Reading them is a great way to learn more about this feature. Protection Profiles can be enabled via the EMET Graphical User Interface, the EMET Command Line Tool or via Group Policy.

2.2 EMET Graphical User Interface

One method of interacting with EMET is through the graphical user interface (GUI). 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.

With EMET 4.0 we have redesigned the GUI in order to facilitate and streamline the configuration process. When EMET GUI is launched the following window is presented to the user.
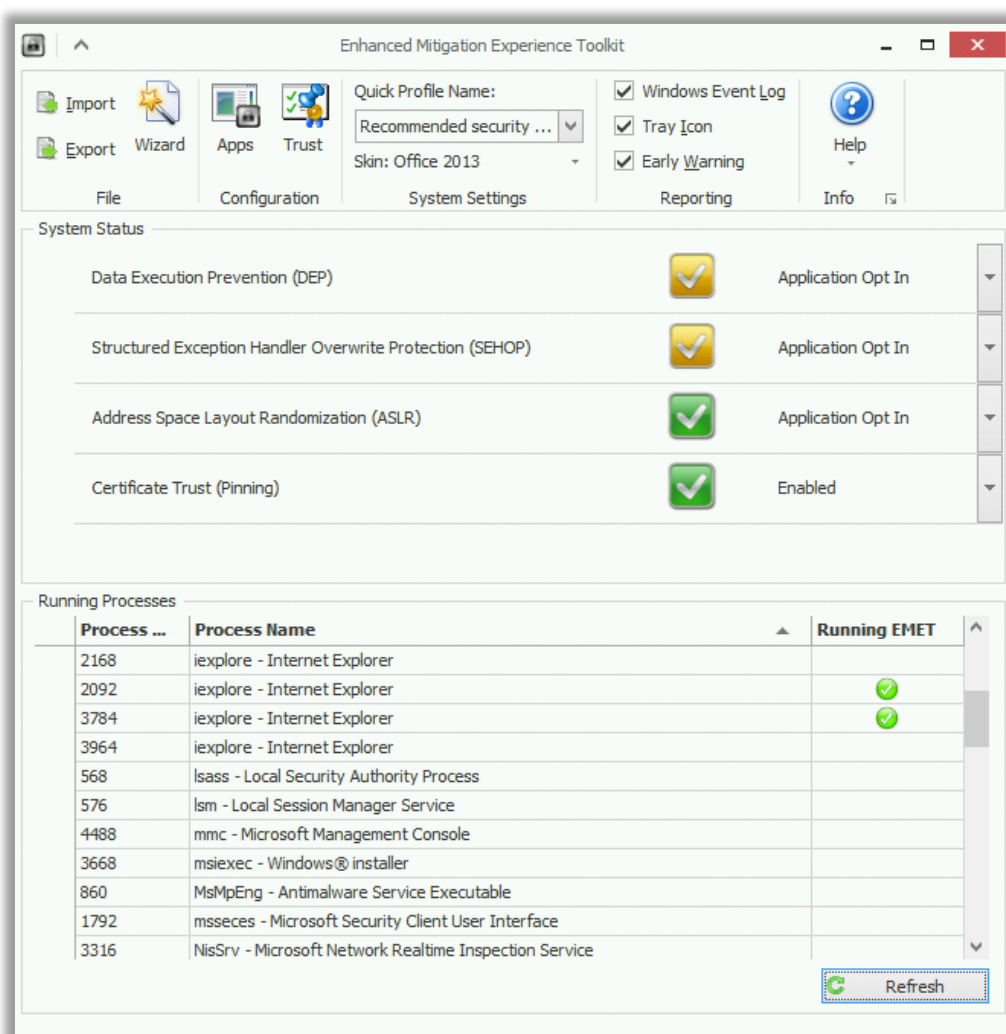

Figure 10: EMET GUI main window

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 2.2.1 - Configuration Wizard for additional details.
  - **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 2.2.3 - Configuring Mitigations for Applications and 2.2.4 - 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 2.2.2 - Configuring System-wide Settings and 2.2.6 - Configuring Appearance for additional details.
  - **Reporting:** This group allows to toggle the Reporting options. See 2.2.5 - 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.

2.2.1   Configuration Wizard

In EMET 4.0 we introduced a Configuration Wizard that facilitates the initial EMET configuration. The Configuration Wizard will be displayed at the end of the EMET installation, and in case of a clean EMET installation it will allow to either apply the recommended settings or to configure EMET manually. In case of an upgrade from a previous version of EMET, the option to keep the existing settings will be displayed.

The Configuration Wizard will automatically detect whether the system already has some EMET settings, and will offer different options accordingly.

We strongly suggest to always apply the Recommended Settings and toggle EMET's configuration afterwards as needed.

2.2.1.1   New installation (no previous settings)

Use Recommended Settings: This option will delete any existing setting and apply the recommended settings:
- **Application Configuration:** Add protections for Internet Explorer, WordPad, Microsoft Office, Adobe Acrobat and Reader, and Oracle Java.
- **Certificate Trust:** Add rules for Microsoft and other 3rd party popular online services.

- **Reporting:** Enable all Reporting mechanisms (Windows Event Log, Tray Icon, and Early Warning Program).

Configure Manually Later: This option will not configure EMET.

2.2.1.2   Upgrade from previous version of EMET or re-configuration

Use Recommended Settings: This option will delete any existing setting and will apply the recommended settings:

- **Application Configuration:** Add protections for Internet Explorer, WordPad, Microsoft Office, Adobe Acrobat and Reader, and Oracle Java.
- **Certificate Trust:** Add rules for Microsoft and other 3rd party popular online services.
- **Reporting:** Enable all Reporting mechanisms (Windows Event Log, Tray Icon, and Early Warning Program).

Keep Existing Settings: This option will keep the existing EMET 3.0 configuration. Two optional settings, related to EMET's new features, can be automatically configured:

- **Certificate Trust:** Add rules for Microsoft and other 3rd party popular online services
- **Reporting:** Enable Early Warning Program.

2.2.2   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 operating system. EMET's GUI provides notification of this when it happens.

The list of available system mitigations varies between different versions of Windows. This reason is that some system mitigations are not available on some operating system. Section 1.5 - Supported Operating Systems and software requirements contains more information about mitigation 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" window.

2.2.3   Configuring Mitigations for Applications

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

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

It is possible to Add (*Ctrl+Add*) and Remove (*Ctrl+Substract*) applications from the list by clicking the corresponding buttons. When adding an application, a user will get prompted with the regular open file dialog and once having selected one it will get added to this list. Then, it is possible to configure it. The "Add Wildcard" (*Ctrl+Multiply*) 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.

EMET will only be in place with the selected configuration after confirming the configuration by clicking on the Ok button and after restarting the newly added/configured application(s).

2.2.3.1   *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 will report the exploitation attempt and terminate the process.
- **Audit only:** EMET will report the exploitation attempt and will 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 the Audit mode are:
    - EAF
    - ROP mitigations: LoadLib, MemProt, Caller, StackPivot, SimExecFlow
    - SEHOP only on Windows XP and Vista

2.2.4   Configuring Certificate Trust (pinning rules)

This feature is only available for Internet Explorer when run in Desktop mode. It is not available in the Modern Internet Explorer app on Windows 8. To enable this feature the "Certificate Trust (Pinning)" must be enabled as described in Section 2.2.2, and the *iexplore.exe* process must be added in the list of protected applications, as described in Section 2.2.3. 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+Add*) in the "Add / Remove" ribbon, type in the Fully Qualified Domain Name of the website as showed in its SSL certificate (NOTE: wildcards or other symbols are not accepted and the name must be unique).
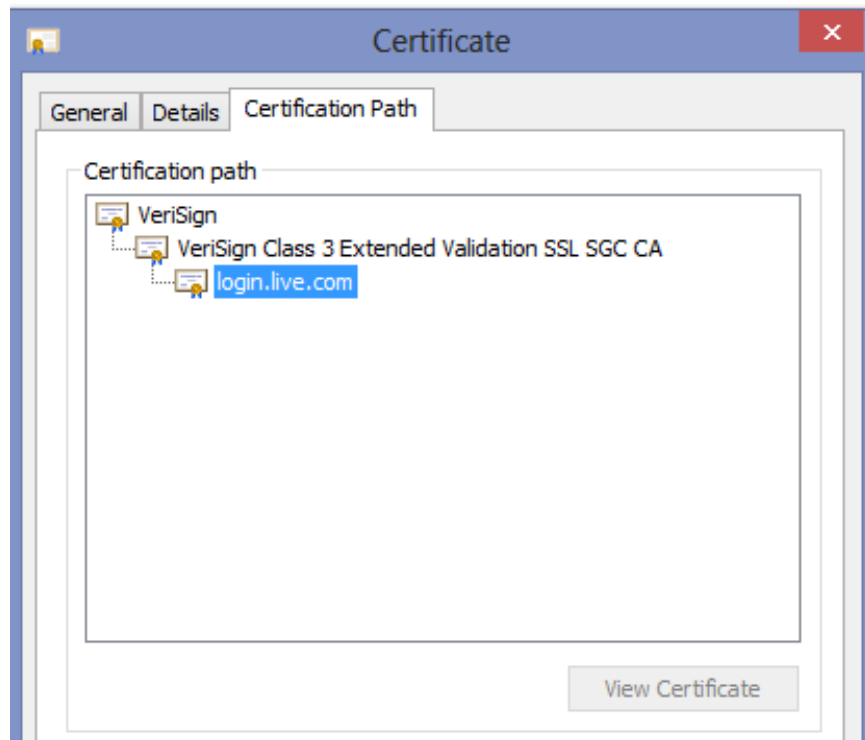
Figure 11: certificate trust chain for login.live.com

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+Add*) 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.

Optionally, it is possible to define four additional checks to allow validation exceptions when a pinning rule for the assigned website does not match (a not-pinned Root CA will be accepted as valid if the defined optional checks are satisfied):

- **Minimum Key Size:** if the Root CA certificate has a key size that is equal or bigger than the selected value, the certificate will be considered valid even if the Root CA is different than the one(s) defined.

- **Allowed Country:** if the issuer country of the Root CA certificate is the same as the one specified in this field, the certificate will be considered valid even if the Root CA is different than the one(s) defined.
- **Blocked Hashes:** if the hashing algorithm of the Root CA certificate is not one selected from this field, the certificate will be considered valid even if the Root CA is different than the one(s) defined.
- **PublicKey Match:** when this option is selected, EMET will verify just the Public Key component of the Root CAs present in a pinning rule without matching subject name and serial number.

NOTE: These optional checks are designed to prevent false reporting and to enable some automatic exceptions for selected Root CA that meet certain criteria; the most restrictive configuration for a pinning rule is the one that have all these optional checks disabled (N/A or unchecked). When a non-pinned Root CA is validated through one of the first three exceptions described above, no visible warning will be displayed by EMET Agent but an event will be written to track this exceptional validation.

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+Substract*) 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).

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. Configuration via Group Policy is not supported for this feature.

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

2.2.5   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 Agent will perform an action. This action can be defined to either write to the Windows Events Log, display a warning, or both. The Early Warning Program is only available for exploits detection, and is not available on Windows XP.

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.

- If "**Windows Event Log**" is selected, EMET will write to the Windows Events Log.
- If "**Tray Icon**" is selected, the EMET Agent will display a pop-up that will warn the user, and will contain the details of the attack.
- If "**Early Warning**" is selected, EMET will generate 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 will send this information to Microsoft through the standard Microsoft Error Reporting channel. When Early Warning is enabled, users will have the opportunity to review the information sent to Microsoft in advance before transmitting it.

NOTE: please refer to section 4 - Advanced Options for the advanced configuration of custom "Tray Icon" messages.

2.2.6   Configuring Appearance

In EMET 4.0 it is possible 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.

2.2.7   Accessibility

With EMET 4.0 we have tried to make EMET GUI closer 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

2.3   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.

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

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

Add an application to EMET

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

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

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

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

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

Example usage:

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

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

List which applications EMET has been enabled for

EMET_Conf --list

Display all the application mitigation settings for EMET, 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 Section 4 - Advanced Options. By default unsafe options are not visible through either the command-line utility or the UI.

Import/Export application settings from an xml file

EMET_Conf --import <xml file>

Imports previously exported 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"

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

3   Deploying EMET

With EMET 4.0, 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.

3.1 System Center Configuration Manager
EMET 4.0 is easily integrated into System Center Configuration Manager for deployment and configuration purposes.

3.1.1   Creating the Application to Deploy EMET to Clients
The first step in deploying EMET is to download the EMET 4.0 MSI. 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 (*).
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.

3.1.2   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.

3.1.3   Create the EMET Configuration Target Collection
1. From Assets and Compliance | Device Collections choose to Create Device Collection.
2. Name the Device Collection (Clients with EMET Installed), and choose the limiting collection.
3. On the membership rules page, click Add Rule, and choose a Query Rule.
4. Name the query, and choose Edit Query Statement.
5. In the criteria tab, click the yellow star.
6. In Criterion Properties, keep the type as Simple value, and choose select.
7. Choose Installed Applications as the attribute class.
8. Choose Display Name as the Attribute.
9. After clicking OK, click the Value button.
10. Choose EMET from the list of values. NOTE: At least one system will have to have reported its hardware inventory up post-EMET client install for this value to be populated. If it's not in the list, simply type the value in.
11. After completing the query rule, choose how often 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.

3.1.4   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. NOTE: 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)

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". NOTE: 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.

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

3.2   Group Policy

EMET 4.0 comes with group policy support. When EMET is installed, EMET.admx and EMET.adml files are also installed to the "Deployment\Group Policy Files" folder. These files must then be copied onto \Windows\PolicyDefinitions and \Windows\PolicyDefinitions\en-US folders respectively. Once this is done, EMET system and application mitigation settings can be configured via Group Policy.

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

1. **System Mitigations:** Named 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.

2. **Default Protection:** There are three: Internet Explorer, Recommended Software, and Popular Software. Protection Profiles are pre-configured EMET settings that cover common home and enterprise software. Apply these policies to enable them.

3. **Application Configuration:** This leads to a freeform editor where additional applications not part of the default protection profiles can be configured. The syntax is application executable name followed by an optional list of mitigations that does not need to be enabled. If no mitigation is specified, all EMET application mitigations will be enabled.

4. **Default Action and Mitigation Settings:** These settings are related to the advanced settings for the ROP mitigations, described in section 1.2.9, and for the default action when an exploit is detected (Audit only or Stop).

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

6. **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.

7. **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*. To make them effective in EMET, the following command must be executed:

> *EMET_Conf --refresh*

Please note that when Group Policy is applied, there is often a short delay before Group Policy are written into the registry.

This command can be run separately, for instance at system startup, at logon time, or with a scheduled task.

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.

3.3 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 Section 2.1.

4   Advanced Options

4.1.1.1   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".

4.1.1.2   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 4.0 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.

4.1.1.3   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;thirdpartybrowser.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.

5 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.

5.1 System Settings

DEP

1. Configuring the system setting for DEP changes a boot option for Windows. For systems using BitLocker, this will cause BitLocker to detect that "system boot information has changed" and 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.

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

SEHOP

1. 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

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

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

For more information on how to turn on the unsafe ASLR setting, refer to Section 4 - Advanced Options.

5.2 Application Specific Settings

DEP

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

SEHOP

1. 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

Null Page

None

Heap Spray

None

Bottom-up randomization

None

EAF

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

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

3. 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

1. EMET's mitigations only become active after the address space for the core process and the static dependencies has been set up. Mandatory ASLR does not force address space randomization on any

of these. The main focus of Mandatory ASLR is to protect dynamically linked modules, such as plug-ins.

2. Windows XP and Windows Server 2003 do not support randomization. Since Mandatory ASLR does not protect the core process or static imports (see #1 above), they will always be at predictable addresses. Consequently, Mandatory ASLR is unable to provide any meaningful protection against attacks on these platforms and is therefore disabled. For more information on which platforms support which mitigations, see Section 1.5 - Supported Operating Systems and software requirements.

Load library checks
None

Memory protection checks
None

Caller checks
None

Simulate execution flow
None

Stack pivot
None

6   Frequently Asked Questions

6.1   EMET 3.0 Questions

- **Is my configuration from EMET 3.0 compatible with EMET 4.0?**

  Yes, EMET 3.0 settings and exported setting files are compatible with EMET 4.0.

- **I have EMET 3.0 installed. Should I uninstall it before installing the new version?**

  You don't need to uninstall EMET 3.0 before installing EMET 4.0. EMET 4.0 will provide an upgrade experience to safely migrate current settings from EMET 3.0.

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

  As of EMET 4.0, we no longer support EMET 1.x or EMET 2.x. We will continue to support EMET 3.0 for 12 months from the release of EMET 4.0.

6.2 General Mitigation Questions

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

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

6.3 Troubleshooting Problems with Mitigations

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

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

  To prevent BitLocker from continually asking for your recovery key, you will need to 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:

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

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

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

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

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

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

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

6.4 General Questions

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

  The GUI requires that .NET 4.0 is installed on the system. If you get this error after copying the binaries form another machine, try running the installer on the local machine. It will direct you to a location where you can download the .NET 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 Section 1.5 - Supported Operating Systems and software requirements for more details.

- **I have an old version of EMET installed. How do I upgrade to EMET 4.0?**

  It is recommended that you first uninstall the old version of EMET via Windows Control Panel, and then manually delete the HKLM\Software\Microsoft\EMET and HKLM\Software\Policies\Microsoft\EMET keys prior to running the EMET 4.0 installer.

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

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

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

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

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

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

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

7   Support

EMET 4.0 is currently supported through the TechNet Forums at http://go.microsoft.com/fwlink/?LinkID=213962.

Customers with a Premier or Professional support contract can leverage these channels to receive support for EMET 4.0.

Users can send email to 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.

A. Appendix: EMET Compatibility

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

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

In EMET 4.0, application compatibility testing was done on all the Microsoft and 3rd party applications that are part of the EMET protection profiles on all the supported platforms. A list of these applications and identified compatibility issues can be found in the table below.

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

Y: Compatible / N: Not Compatible

Table 4: common software compatibility matrix

Application	Mandatory ASLR	DEP	SEHOP	EAF	Heap Spray	Bottom Up	Null Page	Load library checks	Memory protection checks	Simulate execution flow	Stack pivot
Internet Explorer 6-10	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Windows Media Player	✘	✔	✔	✘	✔	✔	✔	✔	✔	✔	✔
Skype	✔	✔	✔	✘	✔	✔	✔	✔	✔	✔	✔
Lync Communicator	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Lync 2013	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Live Writer	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Visio Viewer 2003-2007-2010-2013	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
PowerPoint Viewer 2003-2007-2010-2013	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Visio 2003-2007-2010-2013	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Access 2003-2007-2010-2013	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Excel 2003-2007-2010-2013	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Outlook 2003-2007-2010-2013	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
PowerPoint 2003-2007-2010-2013	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Word 2003-2007-2010	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

Publisher 2003-2007-2010-2013	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
InfoPath 2003-2007-2010-2013	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Google Chrome	✔	✔	✔ (1)	✔	✔	✔	✔	✔	✔	✔	✔
Google Talk	✔	✘	✔	✔	✔	✔	✔	✔	✔	✔	✔
Mozilla Firefox	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Mozilla Firefox Plugin Container	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Thunderbird	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Thunderbird Plugin Container	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Adobe Photoshop CS2/CS3/CS4/CS5/CS5.1/CS6	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Adobe Acrobat Reader 8-9-10-11	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Adobe Acrobat 8-9-10-11	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Winamp	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Opera	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
WinRAR (winrar.exe, unrar.exe, rar.exe)	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
WinZip	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

VLC Player	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
RealPlayer (realplay.exe, realconverter.exe)	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
mIRC	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
7-Zip (7z.exe, 7zG.exe, 7zFM.exe)	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Safari	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
QuickTime	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
iTunes	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Pidgin	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Java 6-7 (java.exe, jawaw.exe, jawaws.exe)	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

(1) Only Windows 7/Server 2008R2 and later versions. The SEHOP implementation for Windows XP/Server 2003 and Windows Vista/Server 2008 is not compatible with this software.

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

Please feel free to contact us via the information in Section 7 - Support to let us know of any incompatibilities that have been encountered.