Malwarebytes

ENDPOINT SECURITY

Malwarebytes Anti-Exploit
Unmanaged Client Administrator Guide

Version 1.12

16 April 2018

Malwarebytes

Notices

This document is provided "as-is." The information contained in this document is subject to change without notice and is not warranted to be error-free. If you find any errors, we would appreciate your comments; please report them to us in writing.

Third Party Project Usage

Malwarebytes software is made possible thanks in part to many open source and third party projects. A requirement of many of these projects is that credit is given where credit is due. Information and licenses for each project are available at:

https://www.malwarebytes.com/support/thirdpartynotices/

Sample Code in Documentation

The sample code described herein is provided on an "as is" basis, without warranty of any kind, to the fullest extent permitted by law. Malwarebytes does not warrant or guarantee the individual success developers may have in implementing the sample code on their development platforms. You are solely responsible for testing and maintaining all scripts.

Malwarebytes does not warrant, guarantee or make any representations regarding the use, results of use, accuracy, timeliness or completeness of any data or information relating to the sample code. Malwarebytes disclaims all warranties, express or implied, and in particular, disclaims all warranties of merchantability, fitness for a particular purpose, and warranties related to the code, or any service or software related there to.

The Malwarebytes Protection Strategy

Malwarebytes' products incorporate several prevention features which utilize a layered defense strategy to protect you against malware threats which you face daily. Each layer is designed to disrupt the attack chain at a different stage. While all Malwarebytes products are highly effective in dealing with attacks that are becoming all too commonplace, we can only assure your protection when you take advantage of the full product suite, allowing each prevention layer to do the job they are best suited for.

It's your data. Protect it wisely!

CWB06-1120a

Table of Contents

Table of Contents (continued)

Introduction

Every week, new financial, state-sponsored and cyber-espionage targeted attacks are discovered. These sophisticated advanced persistent threats use arsenals of vulnerability exploits which have been weaponized to steal confidential information and trade secrets. Organizations remain infected while security companies rush to develop signature updates for an outdated security model.

As software vulnerabilities are discovered and disclosed, traditional approaches to securing of companies and users are based mostly on blacklisting. This applies to binaries (antivirus), spam, network attacks (IDS/IPS) and web filtering (IP/URL blacklisting). When it comes to software vulnerabilities, most vendors focus on detection on a *per attack* or *per vulnerability* basis, as it is easier to create signatures for something that is known and that can be studied in a lab.

Unfortunately, this approach is reactive in nature and does not provide enough protection as proven by the fact that new breaches are discovered on a daily basis. Existing security solutions are slow to react, since they need to be patched by receiving up-to-date malware or network attack signatures in order to provide an effective defense. While the reactive signature approach provides good and specific identification of existing attacks, it is extremely ineffective in protection against new and unknown attacks.

Malwarebytes Anti-Exploit protects against targeted attacks and corporate cyber-espionage. It protects where traditional security measures fail. It consists of an innovative patent-pending application shielding technology which prevents malicious exploits from compromising endpoints through software vulnerabilities.

Malwarebytes Anti-Exploit **is vulnerability-agnostic**. Unlike intrusion detection and prevention products, once an application is protected by *Malwarebytes Anti-Exploit*, the shielded application cannot be exploited through any of its present or future zero-day vulnerabilities. Unlike other vulnerability and intrusion detection products, *Malwarebytes Anti-Exploit* does not require a patient-zero infection.

Malwarebytes Anti-Exploit **is malware-agnostic**. Unlike antivirus and security suites, *Malwarebytes Anti-Exploit* does not care if the malicious payload (trojan, rootkit, rogue antivirus, virus, bot, etc.) is known and detected by antivirus signatures, heuristics or by any other means. *Malwarebytes Anti-Exploit* prevents malicious shellcode and payload from executing even if antivirus products cannot detect it. *Malwarebytes Anti-Exploit* detects what antivirus products normally miss, making it the perfect companion to traditional antivirus and security suites.

Malwarebytes Anti-Exploit **is the most complete anti-exploit (or exploit mitigation) tool in the market.** Unlike other similar tools, *Malwarebytes Anti-Exploit* incorporates multiple protection layers, which in turn are made up of multiple techniques which work in harmony to block exploit attempts at different stages of the vulnerability attack. All techniques are 100% generic and do not depend on any type of blacklisting signature updates, white-listing or sandboxing, making it extremely reliable and resilient to known as well as unknown zero-day vulnerability exploit attacks. In fact, *Malwarebytes Anti-Exploit* has been proven to stop hundreds of zero-day attacks without any previous knowledge of the vulnerability or the exploit.

- **Layer0: Application Hardening –** This is the first and foremost protection against exploits. It consists of different proven and well-known techniques which generically harden applications to be less susceptible to vulnerability exploit attacks. Mitigation techniques in this layer include mandatory Data Execution Prevention (DEP) Enforcement, Bottom-Up ASLR Enforcement and dynamic Anti-Heap Spraying.
- **Layer1: Advanced Memory Protection –** This layer consists of multiple advanced memory protection techniques which detect exploit attempts which try to bypass built-in operating system protections such as Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR). Examples of these techniques are attempts to bypass operating system protections using Return Oriented Programming (ROP) techniques and other such exploit techniques.
- **Layer2: Memory Caller Protection –** This protection layer incorporates multiple memory techniques to prevent exploit code from executing from memory in both 32 and 64-bit operating systems, such as from specific or special memory areas.
- **Layer3: Application Behavior Protection –** This protection layer is the last defense against exploit attempts. In case an exploit is able to bypass all memory protections and/or uses sandbox escape techniques such as those typically used in Acrobat Reader and Java exploits, this layer prevents applications protected by *Malwarebytes Anti-Exploit* from dropping and executing the malicious payloads. This layer also protects against macro exploits executing shell commands in MS Office abusing Windows Management Instrumentation (WMI) and Visual Basic for Applications (VBA7), and guards against insecure Java operations issuing system commands.

What’s New in Malwarebytes Anti-Exploit

Several new features have been added to *Malwarebytes Anti-Exploit* in this version. Following is a selected list of what you can expect in our newest version:

Protection:

- Protection against exploit-driven Process Hollowing attack

Fixes:

- Fixed a conflict with Norton Security
- Fixed opening/closing/hang issues with MS Office apps
- Fixed issues with opening of applications on Windows XP
- Fixed ROP block with a Chinese Bank plugin
- Fixed user notification issues on Windows XP
- Fixed crashes in Firefox browser

Usability:

- Hypervisor Code Integrity Compliant
- Compatible with Windows Device Guard

System Requirements

Following are minimum requirements for an endpoint on which *Malwarebytes Anti-Exploit* may be installed. Please note that these requirements do not include any other functionality that the endpoint is responsible for.

- **Operating System:**
    - Windows 10
    - Windows 8.1
    - Windows 8
    - Windows 7
    - Windows Vista
    - Windows XP (Service Pack 3)
    - Windows Small Business Server 2011
    - Windows Server 2003/2003 R2 (32/64-bit)
    - Windows Server 2008/2008 R2 (32/64-bit)
    - Windows Server 2012/2012 R2 (64-bit)
- **CPU:** 800 MHz or faster
- **RAM:** 2048 MB (server and 64-bit operating system), 1024 MB (32-bit operating system), 256 MB (512 MB or more recommended for Windows XP)
- **Free Disk Space:** 10 MB (initial, log retention may influence this number)
- **Screen Resolution:** 800x600 or higher

Program Installation

Malwarebytes Anti-Exploit may be installed locally via the GUI-based setup program, or remotely using a command line-based installer. For both methods, the process begins by procuring a copy of *Malwarebytes Anti-Exploit*. If you do not have a copy of the program in your possession, it can be downloaded from the Malwarebytes website (https://www.malwarebytes.com/antiexploit/) or by contacting your Malwarebytes representative.

GUI-based Installation

Malwarebytes Anti-Exploit is packaged in a single setup file, which contains all necessary components required to install, configure and operate the program. The setup file name is:

```
mbae-setup-<version>.exe
```

<*version*> represents the current version of the file. Double click on the setup file to begin installation. The installation process uses standard installation steps, which are itemized here:

- Select language to be used for installation, then click **OK** to continue.
- Installation greeting message. Click **Next** to continue.
- Read and accept the License Agreement, then click **Next** to continue.
- Read the Information panel showing new features and changes, then click **Next** to continue.
- Confirm/select alternate installation location, then click **Next** to continue.
- Select location for program shortcuts, then click **Next** to continue.
- Choose whether a desktop icon will be created, then click **Next** to continue.
- Confirm installation options, then click **Install** to continue.
- Click **Finish** to acknowledge that installation has completed.

Command Line Installation

Malwarebytes Anti-Exploit may also be installed remotely from the command line. Using this method, command line parameters can be used to tailor the install to your individual needs. As with the GUI-based installer, the process revolves around use of the setup file. Command line parameters are not case-sensitive. You may specify their usage with upper or lower case characters. They are shown here in upper case for clarity.

Installer Command Line Switches

The following command line parameters can be used as part of the installation process.

/DIR=path Set program installation path. The default path is:
32-bit OS: `%ProgramFiles%\Malwarebytes Anti-Exploit`
64-bit OS: `%ProgramFiles(x86)%\Malwarebytes Anti-Exploit`

/LOG Creates log file "Setup Log YYYY-MM-DD #001.txt" in `%TEMP%` directory

/SP- Disables the "This will install... Do you wish to continue?" prompt at the beginning of Setup.

/VERYSILENT Suppress installation windows and perform default install (unless **/DIR** is specified).

/SUPPRESSMSGBOXES Suppress message boxes. Effective only when combined with **/VERYSILENT**.

/NORESTART Instructs Setup not to reboot even if necessary.

/NOICONS Instructs Setup to not create a folder or shortcuts under the Start Menu.

Examples of Command Line Installation

As mentioned previously, this installation model allows a system administrator to install *Malwarebytes Anti-Exploit* on one or more remote endpoints, and enables the usage of command line parameters to tailor the installation to his corporate needs. Three examples are shown here to illustrate how this method may be used. Please note that these examples include the current version of the *Malwarebytes Anti-Exploit* setup file.

```
psexec \\targetcomputer -u DOMAIN\administrator -p mypassword -d
       \\FILESERVER\Installers\mbae-setup-1.08.2.1043.exe /log /SP- /VERYSILENT
       /SUPPRESSMSGBOXES
```

In the above example, psexec is used to install *Malwarebytes Anti-Exploit* on *targetcomputer*. Authentication is required and is provided by inclusion of the –u and –p parameters. Background operation is specified with the –d switch. A *Malwarebytes Anti-Exploit* installer file stored on a fileserver is being used here.

```
psexec \\* -u DOMAIN\administrator -p mypassword -d
       \\FILESERVER\Installers\mbae-setup-1.08.2.1043.exe /log /SP- /VERYSILENT
       /SUPPRESSMSGBOXES /NOICONS
```

In the above example, *psexec* is being used to install *Malwarebytes Anti-Exploit* on all endpoints in the domain. **Please note** the target endpoint's specification here has been replaced with a wildcard. This allows *Malwarebytes Anti-Exploit* to be installed on all endpoints in the domain through a single execution of the installer. All other components of this command are identical to the previous example.

```
[network_agent] mbae-setup-1.08.2.1043.exe /log /SP- /VERYSILENT
       /SUPPRESSMSGBOXES /NOICONS
```

In the above example, *Malwarebytes Anti-Exploit* is being installed to a remote endpoints using an existing network management agent. Following installation, a setup log will be found in the **%TEMP%** directory of each endpoint, in the format:

```
Setup Log yyyy-mm-dd #001.LOG
```

yyyy-mm-dd corresponds to the installation date, supplemented by a sequence number (if there was more than one installation on the same endpoint. You can verify that *Malwarebytes Anti-Exploit* is installed and running following successful installation by the presence of its icon in the Windows system tray.

Uninstaller Command Line Parameters

The uninstaller (unins000.exe) can also be used with command line parameters to tailor behavior to your individual needs. Following installation, it can be found in the program installation path. The uninstaller must be executed from this location.

/VERYSILENT When specified, the uninstaller operates in the background. No visible indication of the process will be displayed before, during or after the uninstallation. Shared files which are no longer used are deleted automatically without prompting. Any critical error messages will still be shown on the screen. If a restart is required and the **/NORESTART** command is not used (see below), the uninstaller will reboot without asking.

/SUPPRESSMSGBOXES Instructs the uninstaller to suppress message boxes. Only has an effect when combined with **/VERYSILENT**.

/LOG Causes a log file to be created in the user's **%TEMP%** directory detailing file uninstallation and [UninstallRun] actions taken during the uninstallation process. This can be a helpful debugging aid. The log file is created with a unique name based on the current date. (It will not overwrite or append to existing files.) Information contained in the log file is technical in nature and is not intended to be understandable by end users. It is also not designed to be machine-parseable. The file format is subject to change without notice.

/NORESTART Instructs the uninstaller not to reboot even if necessary.

As with the command line installation model, uninstallation may be performed using *psexec* or a network agent of your choice. Individual endpoints are specified by the names assigned to them. Uninstallation may be performed on all endpoints by using a "*" wildcard in place of the endpoint's name.

Standalone Installation using Active Directory GPO

In addition to the standard EXE installer, we also provide a standalone Microsoft Installer (MSI) package. The MSI can be deployed via Active Directory GPO by using the following example command to perform a silent install:

```
        msiexec /i mbae-setup-<version>.msi /quiet
```

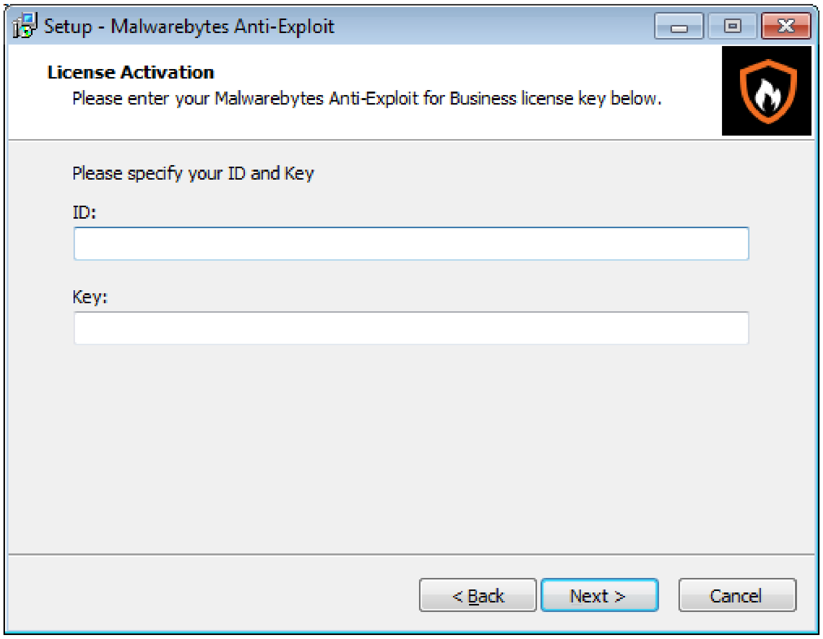
<version> represents the current version of the file.

License Key Activation

The standalone version of *Malwarebytes Anti-Exploit* requires a license ID and KEY to be entered for it to work correctly. There are two ways of doing this depending on the type of installation – manual versus command-line.

Manual Standalone Activation

When executing the standalone installer manually (i.e. double-clicking on mbae-setup-<version>.exe) the installation process will prompt for a valid ID and KEY.



Setup - Malwarebytes Anti-Exploit

License Activation

Please enter your Malwarebytes Anti-Exploit for Business license key below.

Please specify your ID and Key

ID:

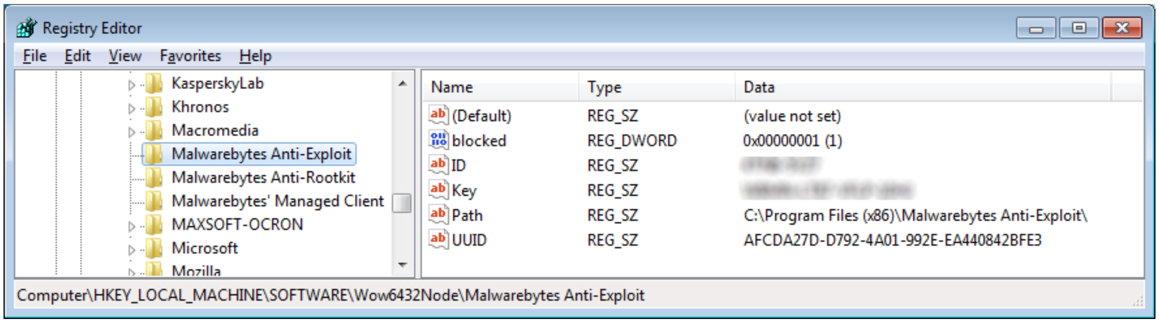
Key:

Command-Line Standalone Activation

Requires a valid license ID and KEY to be entered in the registry. If no valid ID:KEY pair is entered into the registry, the Anti-Exploit Windows Service starts but protection is stopped. The Anti-Exploit Service accepts requests and configuration changes from the command line, but they will not take effect until a valid ID:KEY combination is entered. Appropriate ID:KEY values need to be added to the registry by the network administrator as follows:

- 32-bit operating systems:
  - HKLM\SOFTWARE\Malwarebytes Anti-Exploit\ID
  - HKLM\SOFTWARE\Malwarebytes Anti-Exploit\Key
- 64-bit operating systems:
  - HKLM\SOFTWARE\Wow6432Node\Malwarebytes Anti-Exploit\ID
  - HKLM\SOFTWARE\Wow6432Node\Malwarebytes Anti-Exploit\Key

A screenshot of this registry modification is shown below.


Registry Editor
File Edit View Favorites Help
KasperskyLab
Khronos
Macromedia
Malwarebytes Anti-Exploit
Malwarebytes Anti-Rootkit
Malwarebytes' Managed Client
MAXSOFT-OCRON
Microsoft
Mozilla

Name	Type	Data
(Default)	REG_SZ	(value not set)
blocked	REG_DWORD	0x00000001 (1)
ID	REG_SZ	[redacted]
Key	REG_SZ	[redacted]
Path	REG_SZ	C:\Program Files (x86)\Malwarebytes Anti-Exploit\
UUID	REG_SZ	AFCDA27D-D792-4A01-992E-EA440842BFE3

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Malwarebytes Anti-Exploit

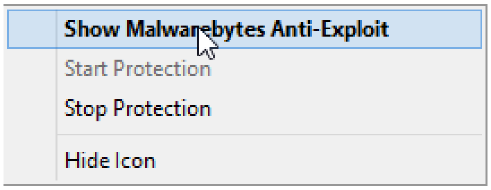
Please note that this registry modification is necessary only when deploying *Malwarebytes Anti-Exploit* in standalone mode. When deployed through the *Malwarebytes Management Console*, the license key needs to be entered into the console and the console will manage the licensing for all endpoints.

User Interface

There are two methods by which users may communicate with *Malwarebytes Anti-Exploit*. The first method is the graphical user interface (GUI). The GUI is available to (a) users who have installed *Malwarebytes Anti-Exploit* themselves, or (b) on endpoints where *Malwarebytes Anti-Exploit* has been installed by system administrators and have been granted GUI privileges by the system administrator. If the system administrator has disallowed access to the user interface (thus preventing interaction with the program), the user interface will simply be unavailable. Please see *Configuring Anti-Exploit for Maximum Protection* (page 18) for further information in this area. Modification of operational settings via the GUI is only available to local users with administrator privileges.

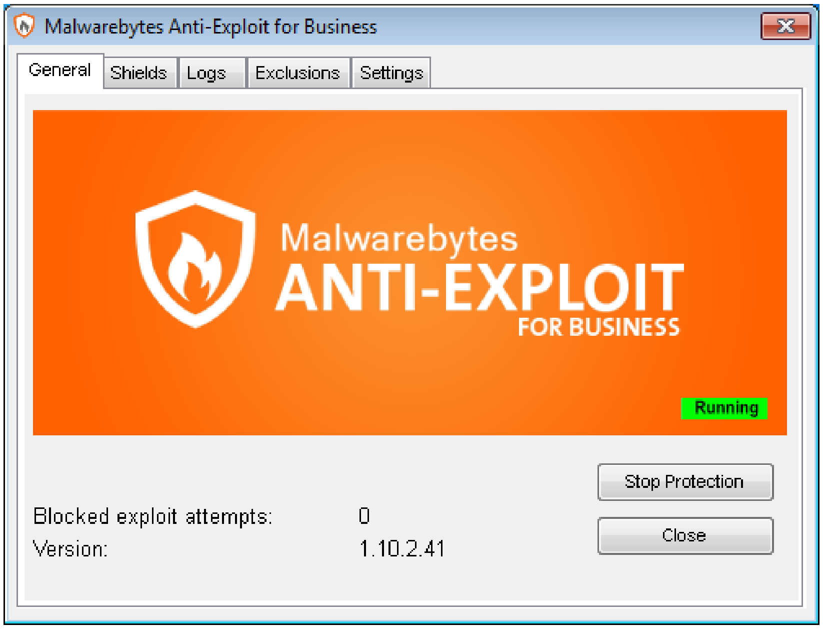
System Tray

Clicking the *Malwarebytes Anti-Exploit* icon in the system tray will display the menu shown below.



General Tab

The screenshot below shows the *Malwarebytes Anti-Exploit* main screen. It is visible initially after installing *Malwarebytes Anti-Exploit* (local installation only), or if launched from the system tray (as shown above). As mentioned previously, an installation performed by a system administrator may deny GUI privileges to a local user, so access to this screen may be denied based on that criteria.

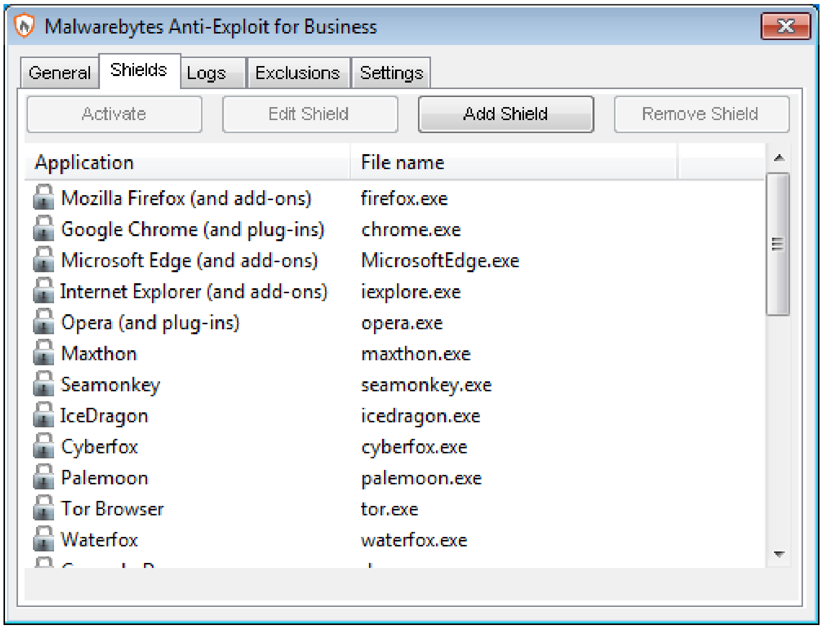


Blocked exploit attempts: 0
Version: 1.10.2.41

Information presented in the GUI is spread across five tabs. The General tab provides general status, allows users to close the interface, and allows local admins to start or stop protection.

Shields Tab

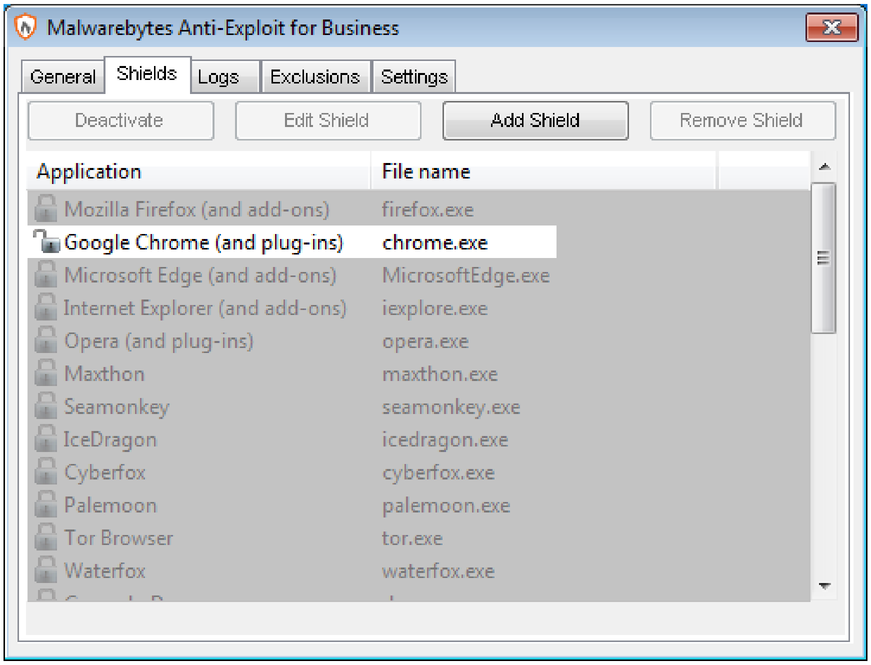
The Shields tab provides a visible indication of the installed applications which can be protected by *Malwarebytes Anti-Exploit*. The appearance of the padlock next to the application's name is an indicator of what is being protected. Let's demonstrate that.



Application	File name
Mozilla Firefox (and add-ons)	firefox.exe
Google Chrome (and plug-ins)	chrome.exe
Microsoft Edge (and add-ons)	MicrosoftEdge.exe
Internet Explorer (and add-ons)	iexplore.exe
Opera (and plug-ins)	opera.exe
Maxthon	maxthon.exe
Seamonkey	seamonkey.exe
IceDragon	icedragon.exe
Cyberfox	cyberfox.exe
Palemoon	palemoon.exe
Tor Browser	tor.exe
Waterfox	waterfox.exe

Unshielded Applications

If an application has been unshielded (i.e. protection has been turned off), a subtle change is visible on this tab. Here, protection has been removed from the Google Chrome browser application via the command line or GUI using the Activate or Deactivate buttons. For activating and deactivating application shields remotely via the command-line program (MBAE-CLI.EXE), please see *Configuring Anti-Exploit for Maximum Protection* (page 18). Please note the highlighted area in this screenshot.

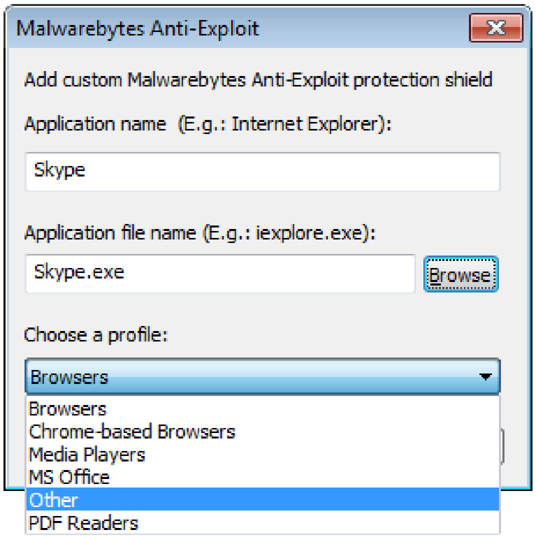


Application	File name
Mozilla Firefox (and add-ons)	firefox.exe
Google Chrome (and plug-ins)	chrome.exe
Microsoft Edge (and add-ons)	MicrosoftEdge.exe
Internet Explorer (and add-ons)	iexplore.exe
Opera (and plug-ins)	opera.exe
Maxthon	maxthon.exe
Seamonkey	seamonkey.exe
IceDragon	icedragon.exe
Cyberfox	cyberfox.exe
Palemoon	palemoon.exe
Tor Browser	tor.exe
Waterfox	waterfox.exe

The padlock icon to the left of Google Chrome is now unlocked. This is a direct result of unshielding the application, and serves as a visible indicator to the user that the application is unprotected.

Adding Custom Shields

Network administrators and local endpoint administrator users can add anti-exploit protection for third-party or in-house custom applications. This can be performed either from a centralized manner via the command-line program (MBAE-CLI.EXE) or locally on each endpoint from the user interface *Shields* tab by clicking the **Add Shield** button.



Malwarebytes Anti-Exploit

Add custom Malwarebytes Anti-Exploit protection shield

Application name (E.g.: Internet Explorer): Skype

Application file name (E.g.: iexplore.exe): Skype.exe

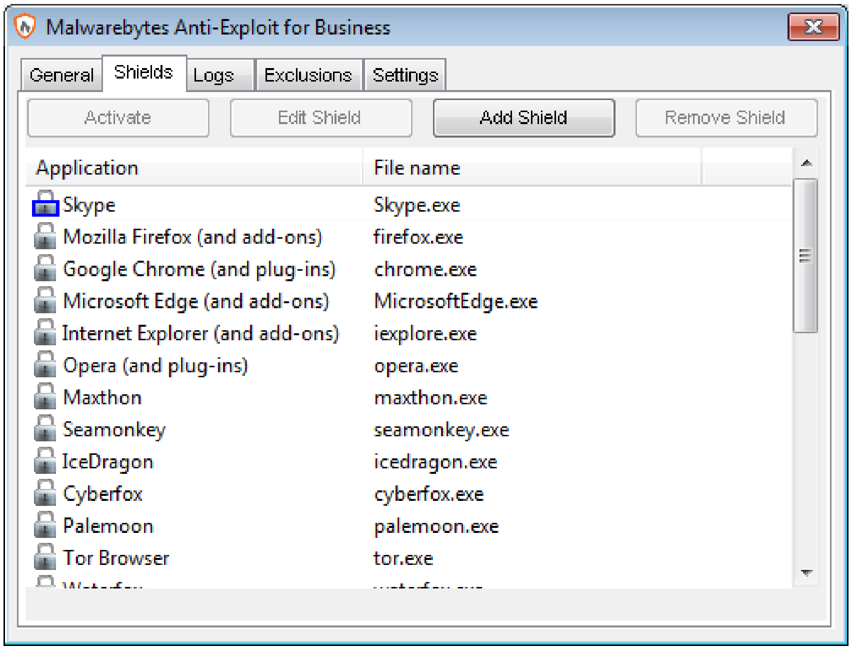
Browse

Choose a profile: Browsers

The following specifications must be provided when adding new shields via the GUI:

- **Application name:** Display name that will appear in the Shields tab
- **Application file name:** Name of the application's executable file. You may use the *Browse* button to find the file to be shielded using Windows Explorer.
- **Protection profile**: *Malwarebytes Anti-Exploit* protection profile to apply to the application. It is important to choose a profile that matches the application type being protected, so that the best anti-exploit techniques are applied. Various protection profiles are available (Browsers, Chrome-based Browsers, PDF Readers, MS Office, Media Players and Other). This provides the most effective protection with the least possibility of conflicts and incompatibilities.

Please note the colored contour around the Skype custom shield in the following screenshot. The blue color indicates the custom shield was added locally from the GUI, whereas a green colored shield indicates the shield was added remotely by the administrator via the command-line program MBAE-CLI.EXE.



Malwarebytes Anti-Exploit for Business

Application	File name
Skype	Skype.exe
Mozilla Firefox (and add-ons)	firefox.exe
Google Chrome (and plug-ins)	chrome.exe
Microsoft Edge (and add-ons)	MicrosoftEdge.exe
Internet Explorer (and add-ons)	iexplore.exe
Opera (and plug-ins)	opera.exe
Maxthon	maxthon.exe
Seamonkey	seamonkey.exe
IceDragon	icedragon.exe
Cyberfox	cyberfox.exe
Palemoon	palemoon.exe
Tor Browser	tor.exe

You may wish to consider adding custom shields for third-party Internet-facing applications, as they are at an increased risk to vulnerability exploits. Some examples might be Skype and other VoIP applications, Instant Messaging applications, RSS readers that use a browser rendering engine, etc.

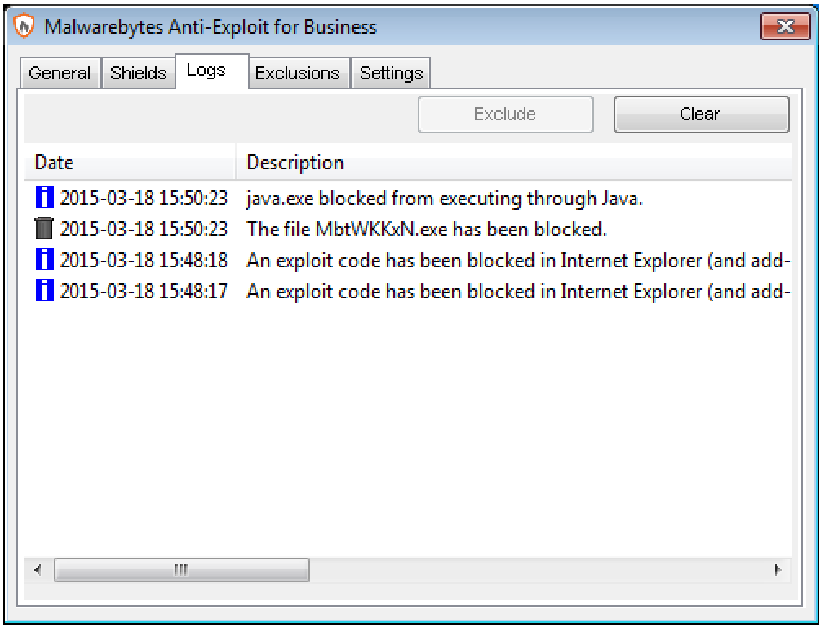
Deleting Custom Shields

Shields can be deleted from the Shields tab of the interface by clicking the **Remove Shield** button, or remotely via the command-line program MBAE-CLI.EXE. Please be aware that only previously added shields can be deleted. Pre-defined shields shipped by default with *Malwarebytes Anti-Exploit* cannot be deleted, only deactivated. Also note that shields added by the administrator remotely via MBAE-CLI.EXE (i.e. green shields) cannot be deleted or deactivated locally from the GUI by the user. Only the administrator (via MBAE-CLI.EXE) can delete or deactivate these shields. Additionally the administrator may delete or deactivate any custom shields added locally by the user via the GUI (i.e. blue shields).

Logs Tab

The Logs tab provides a list of events related to operation of *Malwarebytes Anti-Exploit*. All events are displayed in reverse chronological order. There is no provision for changing the order of displayed events.

All information presented here is also available in system logs (discussed on page 31). If you are a local admin, you may clear this display of events shown here by clicking the **Clear** button. **Please note** that this only clears the display in the user interface. System logs remain intact.



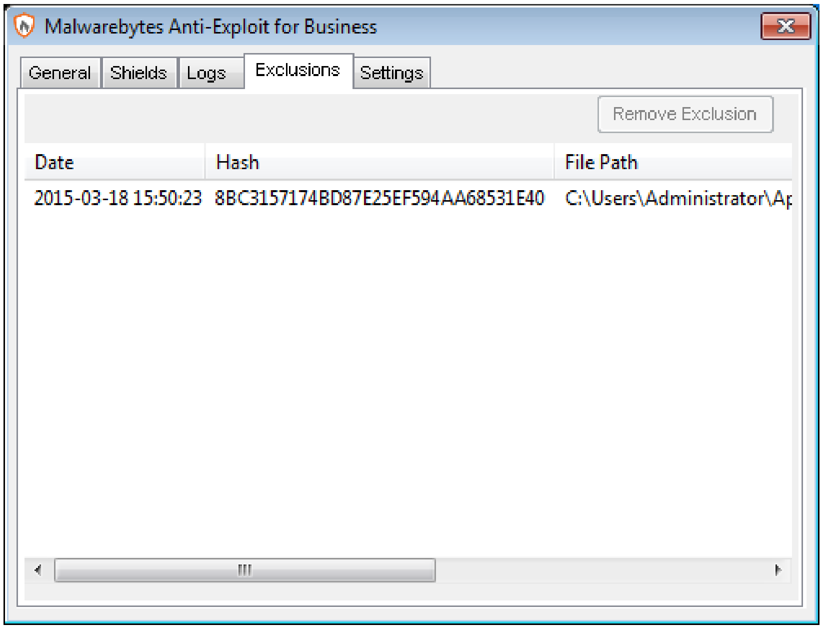
	Date	Description
ℹ	2015-03-18 15:50:23	java.exe blocked from executing through Java.
🗑	2015-03-18 15:50:23	The file MbtWKKxN.exe has been blocked.
ℹ	2015-03-18 15:48:18	An exploit code has been blocked in Internet Explorer (and add-
ℹ	2015-03-18 15:48:17	An exploit code has been blocked in Internet Explorer (and add-

Two different icons are displayed to the left of the system date. These indicate the category of information being displayed, and are provided as a quick method of focusing your attention. The information categories are:

ICON	MEANING
ℹ	*Malwarebytes Anti-Exploit* has blocked an exploit attempt. Full details of the techniques used are available in file *mbae-alert.log* (see page 31 for information on system logs).
🗑	*Malwarebytes Anti-Exploit* has prevented the specified program from delivering a malicious payload, according to Layer 3 mitigation techniques. For entries of this type, the program's file path and MD5 hash are also displayed for verification purposes. As a local admin, if you are familiar with the program and trust it, you can highlight the log entry and click the **Exclude** button to prevent it from being blocked in the future. **Please note** that an exclusion is based specifically on the MD5 hash of the program, and not the name of the program. If a new version of a trusted program is released using the same file name, the MD5 hash of the new version will be different than the old (excluded) version, and the previous exclusion will no longer be honored.

Exclusions Tab

The Exclusions tab is a list of all programs which are being excluded from anti-exploit testing. Entries here may be a direct result of what *Malwarebytes Anti-Exploit* had considered to be an exploit attempt in the past. Exclusions are typically added by a local admin, because a flagged program is trusted and considered safe.

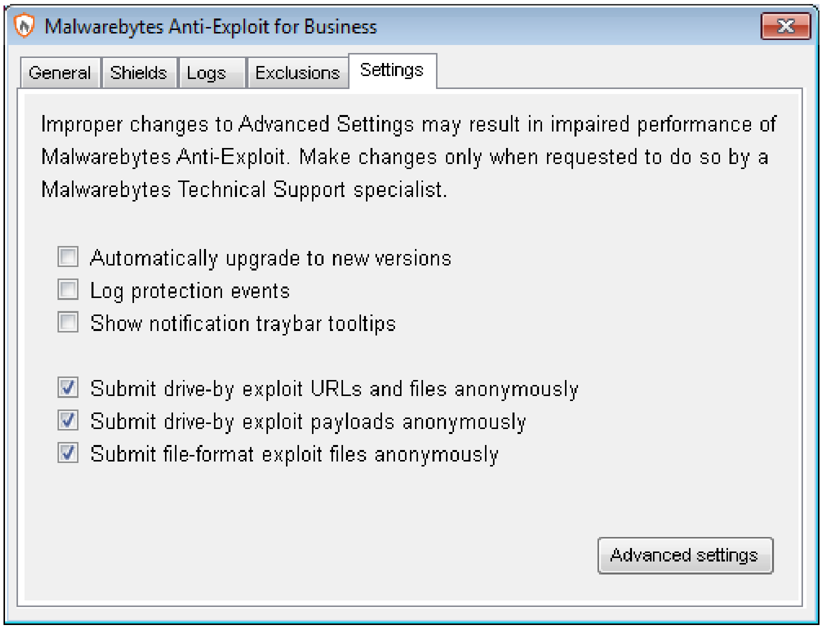


Date	Hash	File Path
2015-03-18 15:50:23	8BC3157174BD87E25EF594AA68531E40	C:\Users\Administrator\Ap

Referring to the Logs tab (on the previous page), this specific program was evaluated as a Layer 3 exploit attempt. Because it is known to be safe, it was excluded. As a result of that exclusion, an entry for the program was immediately created here on the Exclusions tab.

Settings Tab

The Settings tab performs three purposes. First, it allows you to define preferences. You may choose whether your installation of *Malwarebytes Anti-Exploit* will be automatically upgraded when a new program version is available. You may choose whether protection events are lincluded in the **Logs** display. You may also choose whether notifications are displayed in your taskbar when an application is launched that causes anti-exploit shields to go into action for the newly-launched application. A screenshot is shown below.



Improper changes to Advanced Settings may result in impaired performance of Malwarebytes Anti-Exploit. Make changes only when requested to do so by a Malwarebytes Technical Support specialist.

□ Automatically upgrade to new versions
□ Log protection events
□ Show notification traybar tooltips

☑ Submit drive-by exploit URLs and files anonymously
☑ Submit drive-by exploit payloads anonymously
☑ Submit file-format exploit files anonymously

A second purpose of this tab is to allow you to select whether detected exploits are sent to the Malwarebytes Research team by your installation of *Malwarebytes Anti-Exploit*. Being able to inspect exploits allows us to do our job of protecting you even better, and allows us to constantly upgrade our knowledge base of malware being used. **Please note** that any information which you elect to provide to us pertains only to exploits. No information about you or your company is ever sent to us.

The third purpose is to provide access to Advanced Settings via a button at the bottom of the screen. When this button is pressed, Advanced Settings launches in a new program window. Before digging deeper, a brief introduction and a warning are appropriate.

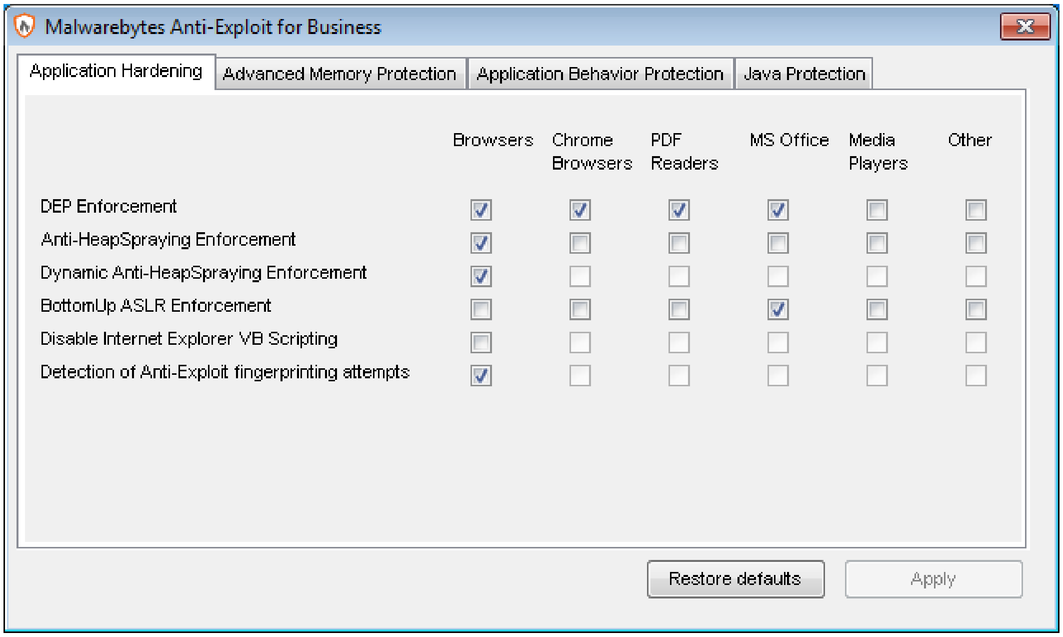
Advanced Settings allows users to configure or fine-tune some of the exploit mitigations included in *Malwarebytes Anti-Exploit*. It is important to note that not all exploit mitigations included in *Malwarebytes Anti-Exploit* can be modified using Advanced Settings. *Malwarebytes Anti-Exploit* has pre-defined defaults which already strike the best possible balance between seamless performance and protection. Those exploit mitigations that are made available for configuration have been deemed relevant to be tuned by users in scenarios where certain non-standard or heavily customized computing environments result in unexpected behavior by *Malwarebytes Anti-Exploit* (e.g. false positives).

WARNING: Improper changes to these configurations may result in improper performance and protection offered by *Malwarebytes Anti-Exploit*. Make changes only when required to do so by a Malwarebytes Customer Success specialist.

Now, let's look at each of the Advanced Settings tabs.

Application Hardening

This section refers to exploit mitigation techniques whose objective is to make protected applications more resilient against vulnerability exploit attacks, even if those applications have not been patched to the latest available fixes by their respective vendors. A screenshot shows the organization of the tab.



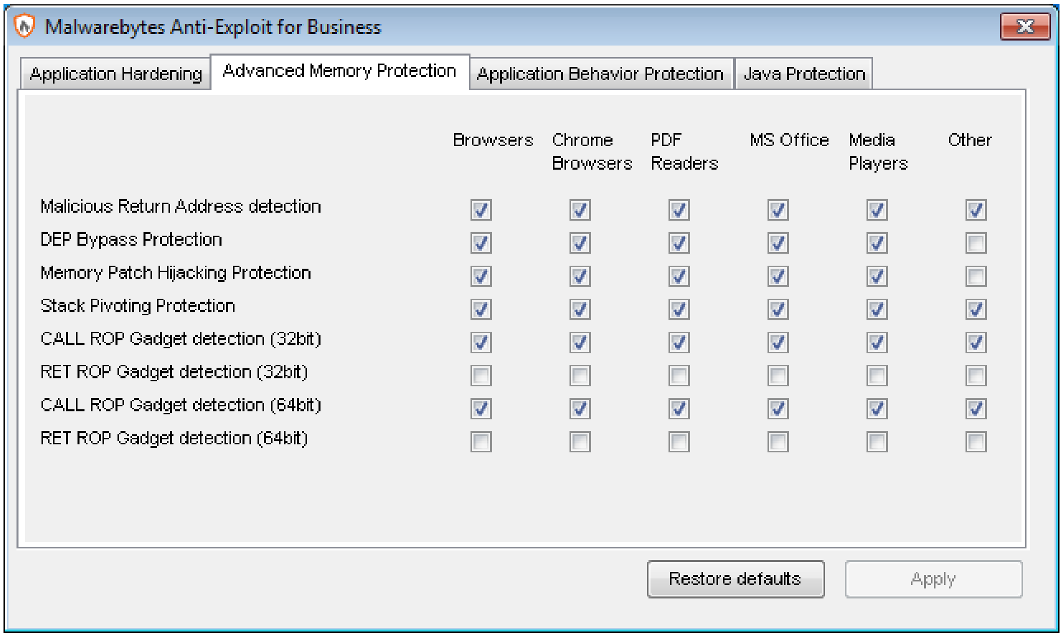
Malwarebytes Anti-Exploit for Business
Application Hardening
Advanced Memory Protection
Application Behavior Protection
Java Protection
Restore defaults
Apply

	Browsers	Chrome Browsers	PDF Readers	MS Office	Media Players	Other
DEP Enforcement	☑	☑	☑	☑	□	□
Anti-HeapSpraying Enforcement	☑	□	□	□	□	□
Dynamic Anti-HeapSpraying Enforcement	☑	□	□	□	□	□
BottomUp ASLR Enforcement	□	□	□	☑	□	□
Disable Internet Explorer VB Scripting	□	□	□	□	□	□
Detection of Anti-Exploit fingerprinting attempts	☑	□	□	□	□	□

- **DEP Enforcement** is tasked with activation of permanent Data Execution Prevention (DEP) in those applications that do not do this by default.
- **Anti-HeapSpraying Enforcement** is designed to reserve certain memory ranges, to prevent them from being abused by Heap-Spraying attack techniques.
- **Dynamic Anti-HeapSpraying Enforcement** analyzes the memory heap of a protected process in order to find evidence of malicious shellcode on the heap using heap spraying techniques.
- **Bottom-Up ASLR Enforcement** is tasked with addition of some randomization to the memory heap when the process starts up.
- **Disable Internet Explorer VB Scripting** is tasked with preventing the deprecated Visual Basic scripting engine from loading. The scripting engine is frequently abused by exploits. This setting applies only to the browser family.
- **Detection of Anti-Exploit fingerprinting attempts** is a technique which detects attempts by popular exploit kits (e.g. Angler) of fingerprinting the victim machine to determine if it should be attacked by its exploit arsenal.

Advanced Memory Protection

This tab allows tuning of exploit mitigation techniques whose objective is to prevent exploit shellcode from bypassing built-in operating system security mechanisms (e.g. DEP) which allow the exploit to execute. A screenshot shows the organization of the tab.

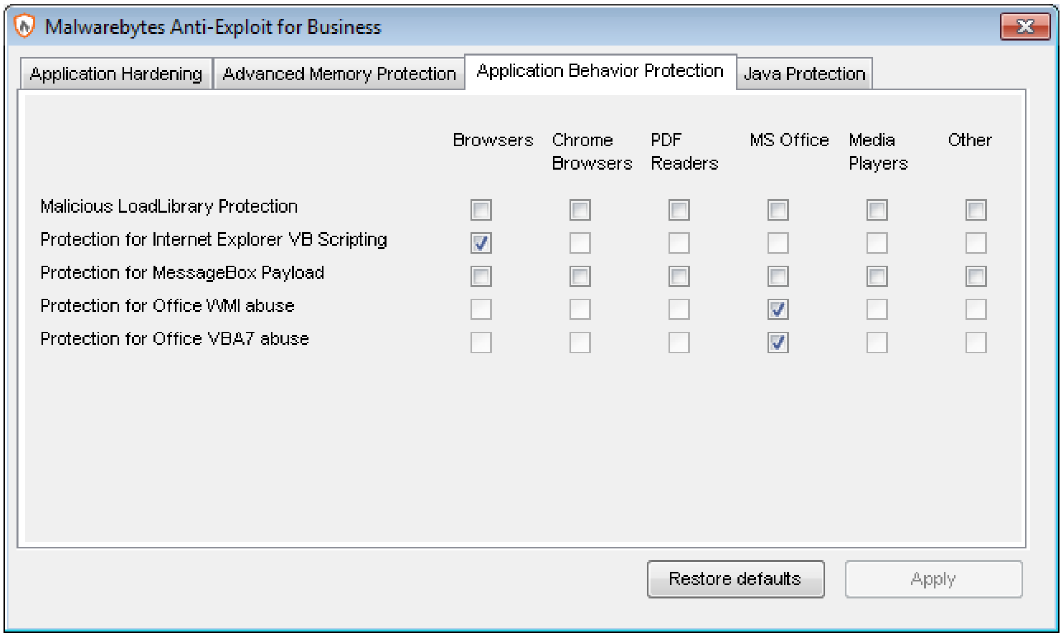

Malwarebytes Anti-Exploit for Business
Application Hardening
Advanced Memory Protection
Application Behavior Protection
Java Protection
Restore defaults
Apply

	Browsers	Chrome Browsers	PDF Readers	MS Office	Media Players	Other
Malicious Return Address detection	☑	☑	☑	☑	☑	☑
DEP Bypass Protection	☑	☑	☑	☑	☑	□
Memory Patch Hijacking Protection	☑	☑	☑	☑	☑	□
Stack Pivoting Protection	☑	☑	☑	☑	☑	☑
CALL ROP Gadget detection (32bit)	☑	☑	☑	☑	☑	☑
RET ROP Gadget detection (32bit)	□	□	□	□	□	□
CALL ROP Gadget detection (64bit)	☑	☑	☑	☑	☑	☑
RET ROP Gadget detection (64bit)	□	□	□	□	□	□

- **Malicious Return Address detection** detects if shellcode is executed outside of any loaded module, preventing payload from executing in memory.
- **DEP Bypass Protection** is tasked with detecting attempts to turn off Data Execution Prevention (DEP).
- **Memory Patch Hijacking Protection** is designed to detect and prevent against attempts to use WriteProcessMemory to bypass Data Execution Prevention (DEP).
- **Stack Pivoting Protection** is used to detect and prevent exploit code from creating and utilizing a fake memory stack.
- **ROP Gadget detection** is tasked with detection and prevention of Return Oriented Programming (ROP) gadgets when a Windows API is called. Provisions are made for individualized protection of call and return instructions, as well as protection settings specific to 32 and 64-bit environments.

Application Behavior Protection

This section refers to exploit mitigation techniques designed to prevent the exploit payload from executing and infecting the system. These protections are the last line of defense if memory corruption exploit mitigations from previous layers are bypassed. This layer is also tasked with detecting exploits that do not rely on memory corruption (e.g. Java sandbox escapes, application design abuse exploits, etc.) and blocking their malicious actions.

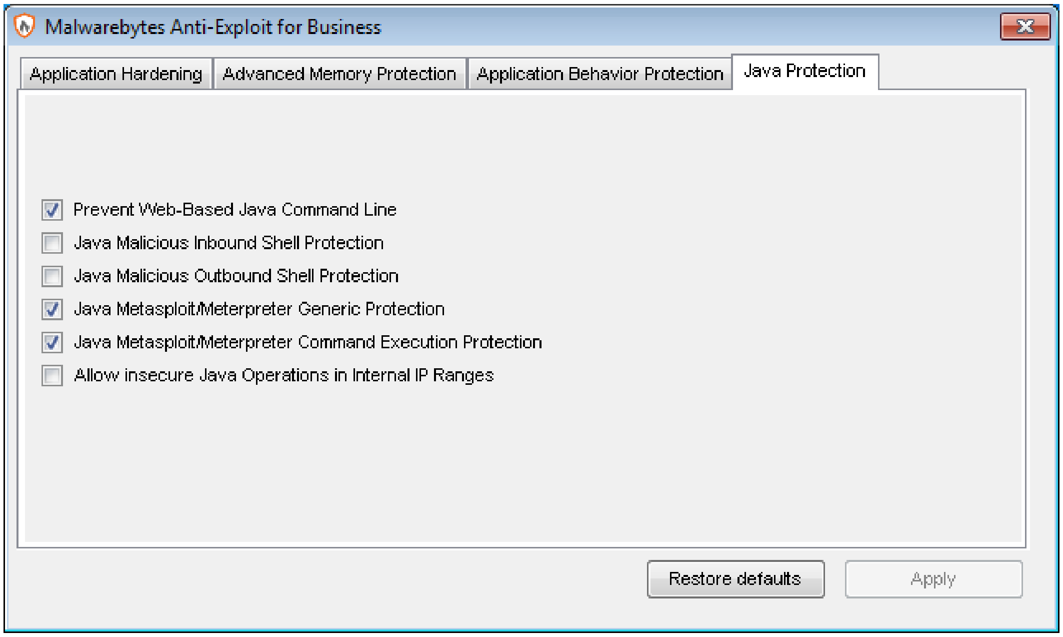

Malwarebytes Anti-Exploit for Business
Application Hardening
Advanced Memory Protection
Application Behavior Protection
Java Protection
Restore defaults
Apply

	Browsers	Chrome Browsers	PDF Readers	MS Office	Media Players	Other
Malicious LoadLibrary Protection	□	□	□	□	□	□
Protection for Internet Explorer VB Scripting	☑	□	□	□	□	□
Protection for MessageBox Payload	□	□	□	□	□	□
Protection for Office WMI abuse	□	□	□	☑	□	□
Protection for Office VBA7 abuse	□	□	□	☑	□	□

- **Malicious LoadLibrary Protection** is tasked with preventing an exploit from delivery a payload library from a UNC network path.
- **Protection for Internet Explorer VB Scripting** is designed to detect and prevent exploits related to an application design vulnerability known as CVE-2014-6332. For further information on this exploit, please refer to https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-6332.
- **Protection for MessageBox Payload** is tasked with preventing exploits from delivering a messagebox as its payload. It is turned off by default as these payloads are normally only used in proof of concepts and do not cause any harm.
- **Protection for Office WMI abuse** protects against macro exploits in Microsoft Office using Windows Management Instrumentation (WMI).
- **Protection for Office VBA7** abuse protects against macro exploits in Microsoft Office using Visual Basic for Applications (VBA7).

Java Protection

This section refers to mitigation techniques which are unique to exploits commonly used in Java programs.



Malwarebytes Anti-Exploit for Business
Application Hardening
Advanced Memory Protection
Application Behavior Protection
Java Protection
Restore defaults
Apply

☑ Prevent Web-Based Java Command Line
□ Java Malicious Inbound Shell Protection
□ Java Malicious Outbound Shell Protection
☑ Java Metasploit/Meterpreter Generic Protection
☑ Java Metasploit/Meterpreter Command Execution Protection
□ Allow insecure Java Operations in Internal IP Ranges

- **Prevent Web-Based Java Command Line** protects against web-based Java programs issuing system commands.
- **Java Malicious Inbound Shell Protection** is designed to detect and prevent remote shell exploits whose payloads rely on inbound sockets.
- **Java Malicious Outbound Shell Protection** is tasked with detection and prevention of remote shell exploits whose payloads rely on outbound sockets.
- **Java Metasploit/Meterpreter Generic Protection** is designed to generically detect and prevent attempts to use the Metasploit Java/Meterpreter payload.
- **Java Metasploit/Meterpreter Command Execution Protection** is tasked with detecting and blocking commands in an established Java/Meterpreter session.
- **Allow Insecure Java Operations in Internal IP Ranges** is primarily used to allow insecure internal tools and applications used within a corporate network without compromising on protection from external Java threats.

Configuring Anti-Exploit for Maximum Protection

Malwarebytes Anti-Exploit comes pre-configured to provide the best balance between security and compatibility with corporate environments. However, *Malwarebytes Anti-Exploit* can be configured to provide even more protection than is shipped by default. The following *Advanced Configuration* options are recommended techniques for businesses wanting to maximize the protection offered by *Malwarebytes Anti-Exploit*. All of these options can be configured under *Advanced Settings* from both the endpoints' *Malwarebytes Anti-Exploit* user interface, or centrally from the *Malwarebytes Management Console's* Policy module. Please test these configurations on a few machines first, to verify compatibility with the corporate environment and with installed applications before deploying changes to all your endpoints.

Disable Internet Explorer VB Scripting

In 2015, attackers began abusing a zero-day exploit in the deprecated VB Scripting Engine (vbscript.dll), and this has become a favorite exploit method. Enable the *Disable Internet Explorer VB Scripting* technique (found in the *Application Hardening* tab) to prevent the VB Scripting Engine from being loaded in Internet Explorer. Even if this technique is disabled, *Malwarebytes Anti-Exploit* still provides protection against vbscript.dll attacks from its *Application Behavior Protection* (Layer3) in order to prevent certain process-creation APIs from being called by vbscript.dll. However, companies wanting to lock-down even further may choose to enable this feature under the Application Hardening Layer. Please note that this technique might break functionality of Intranet applications which rely on the deprecated VB Scripting engine.

Java Malicious Inbound Shell Protection

By default, *Malwarebytes Anti-Exploit* provides protection against Java exploit payloads and Java-based reverse shells like Metasploit's Meterpreter. For increased protection, businesses may elect to enable *Java Malicious Inbound Shell Protection* (found in the *Application Behavior Protection* tab) as an added layer of protection from Java-based reverse-shell exploit payloads. Please note that this may trigger alerts with Intranet Java applications that make heavy use of inbound listening sockets.

Other Advanced Configuration Options

Businesses may choose to enable other advanced settings of *Malwarebytes Anti-Exploit* in order to provide even further protection. The following options should be tested more thoroughly for compatibility with the corporate environment and applications before deployment to all endpoints, as they have been observed to cause conflicts or false positives under certain configurations. The available options for further lock-down are the following:

- *Application Hardening ► Anti-HeapSpraying Protection*
  Enable for PDFReaders and/or Office
- *Application Hardening ► BottomUp ASLR Enforcement*
  Enable for Browser and/or PDFReaders
- *Application Behavior Protection ► Malicious LoadLibrary Protection*
  Enable for each application family

Command Line Interface

Earlier in this guide, it was shown how system administrators could use *psexec* (or other network management tools) as a means of installing *Malwarebytes Anti-Exploit* onto remote endpoints with a command line-based setup program. The same capability exists for day-to-day operation and management of remote endpoints, using a program named *mbae-cli*. This program is installed as part of the *Malwarebytes Anti-Exploit* package. When used to control *Malwarebytes Anti-Exploit*, it needs to execute on the target endpoint with SYSTEM privileges. It is designed to manage program operations, and provides additional functionality not available in the GUI interface. Additionally, it has the capability to override any settings made to the remote endpoint *locally* using the GUI interface. Let's look at *mbae-cli* with a focus on the options which it provides.

Parameters for Maintenance Tasks

A single parameter may be used with each execution of an *mbae-cli* command, and a return code will be returned in response to the command. The following is a list of all parameters that can be used as part of an *mbae-cli* command to perform maintenance tasks on *Malwarebytes Anti-Exploit*.

Start Protection

Purpose:

Initiate protection by *Malwarebytes Anti-Exploit*. Please note that *Malwarebytes Anti-Exploit* starts by default, so there is no need to issue the **/start** command unless a previous **/stop** command has been issued.

Usage:

```
mbae-cli /start
```

Arguments:

none

Stop Protection

Purpose:

Stop *Malwarebytes Anti-Exploit* protection. This command simply pauses the protection. Both the service and the GUI remain running after issuing the **/stop** command.

Usage:

```
mbae-cli /stop
```

Arguments:

none

Check Protection Status

Purpose:

Check operational status of *Malwarebytes Anti-Exploit*.

Usage:

```
mbae-cli /status
```

Arguments:

none

Show Alerts upon Detection

Purpose:

Configure *Malwarebytes Anti-Exploit* to show alert popups if an exploit attempt has been detected. This command also toggles status if a **/noalert** command had been previously issued. This is a default setting.

Usage:

```
mbae-cli /alert
```

Arguments:

none

Do Not Show Alerts upon Detection

Purpose:

Configure *Malwarebytes Anti-Exploit* to prevent display of alert popups upon detection of an exploit attempt.

Usage:

```
mbae-cli /noalert
```

Arguments:

none

Show User Interface

Purpose:

Configure *Malwarebytes Anti-Exploit* to display both the user interface and system tray icon. This command also toggles status if a **/nogui** command had been previously issued. This is a default setting.

Usage:

```
mbae-cli /gui
```

Arguments:

none

Do Not Show User Interface

Purpose:

Configure *Malwarebytes Anti-Exploit* to prevent display of the user interface as well as the system tray icon. If the user attempts to manually execute *Malwarebytes Anti-Exploit* after a **/nogui** command has been issued, it will not launch.

Usage:

```
mbae-cli /nogui
```

Arguments:

none

Shield a Pre-Defined Application

Purpose:

Configure *Malwarebytes Anti-Exploit* to activate protection for a pre-defined application. The application to be shielded must be specified as an argument. You may also protect other applications through use of the **/shield-add** command.

Usage:

```
mbae-cli /shield <appname>
```

Arguments:

appname Name of the application to be shielded. Please refer to the following list for examples of *appname*, and the corresponding display name for each program.

appname	Display Name
firefox	Mozilla Firefox
chrome	Google Chrome
edge	Microsoft Edge
iexplore	Internet Explorer
opera	Opera
maxthon	Maxton
seamonkey	Seamonkey
icedragon	IceDragon
cyberfox	Cyberfox
palemoon	Palemoon
tor	Tor Browser Bundle
waterfox	Waterfox
comododragon	Comodo Dragon
java	Java
acrobat	Adobe Acrobat
acrord32	Adobe Reader
foxitphantom	Foxit Phantom
foxitreader	Foxit Reader
libreoffice	Libre Office and OpenOffice
winword	Microsoft Office Word
excel	Microsoft Office Excel
powerpnt	Microsoft Office PowerPoint
wmplayer	Windows Media Player (wmplayer)
mplayer2	Windows Media Player (mplayer2)
vlc	VLC Player
winamp	Winamp Player
quicktimeplayer	QuickTime Player

Example:

```
mbae-cli /shield iexplore
```

Unshield a Pre-Defined Application

Purpose:

Deactivate protection for a pre-defined application. The application must be specified as an argument. Use **/shield-del** to remove shields from other applications.

Usage:

```
mbae-cli /unshield <appname>
```

Arguments:

appname Name of the application to be unshielded. Refer to ***/shield*** for a list of applications.

Example:

```
mbae-cli /unshield iexplore
```

Add a Custom Shield

Purpose:

Add a custom shield to protect a new application. Four arguments provide specifications for the custom shield. Please note that improper specification of a protection profile will negatively impact performance and compatibility of the shield being applied.

Usage:

```
mbae-cli /shield-add -e "<exe>" -n <app> -d "<dname>" -p <type>
```

Arguments:

-e <*exe*> Exact file name of the executable application to be shielded.

-n <*app*> Application Nickname. Used also by */shield-del* command. Should be a one-word name (with no spaces) to reference the application. Names associated with pre-defined applications (refer to */shield* for a list) cannot be used here.

-d <*dname*> Display name. Friendly name to be displayed in the GUI and used in reporting. Values containing spaces or special characters must be enclosed by quotes (i.e. "my value").

-p <*type*> Protection profile to apply to the customer shields. Must be one of the following:

- browser
- chromebrowser
- pdfreader
- office
- mediaplayer
- other

Example:

```
mbae-cli /shield-add -e "my app.exe" -n myapp -d "My Browser" -p browser
```

Delete a Custom Shield

Purpose:

Remove a custom shield from an application. This command may not be used to remove a pre-defined application shield.

Usage:

```
mbae-cli /shield-del <app>
```

Arguments:

<*app*> Application Nickname. Used also by */shield-add* command (**-n** argument).

Example:

```
mbae-cli /shield-del myapp
```

List Custom Shields to Text File

Purpose:

Generate a list of custom shields, sending output to a text file. Information displayed includes (a) executable file, (b) application nickname, (c) display name and (d) protection profile applied.

Usage:

```
mbae-cli /shield-list <file>
```

Arguments:

<*file*> Output file to be generated.

Example:

```
mbae-cli /shield-list d:\protection\mbae-shields.txt
```

Add File to Exclusion List

Purpose:

Add a file to a list of files which will be excluded from exploit detection by *Malwarebytes Anti-Exploit* application behavior protection layer.

Usage:

```
mbae-cli /excl-add <MD5>
```

Arguments:

<MD5> The MD5 hash signature of a file to be excluded. This 128-bit signature represents both data and the order of data in a file. Virtually any modification made to a given file will result in a different hash. *Malwarebytes Anti-Exploit* uses MD5 hashes as a way of representing a file to be excluded or included, because the hash is much more specific than simply a file name and/or file path. An internet search will yield many free programs that can generate reliable MD5 hashes for your files.

Example:

```
mbae-cli /excl-add F6C75620A1A77241C4E810C2409BADC9
```

Remove File from Exclusion List

Purpose:

Remove a file from the exclusion list. Once removed from the list, it may potentially result in detections during exploit blocking by *Malwarebytes Anti-Exploit.*

Usage:

```
mbae-cli /excl-del <MD5>
```

Arguments:

<MD5> The MD5 hash signature of a file to be excluded.

Example:

```
mbae-cli /excl-del F6C75620A1A77241C4E810C2409BADC9
```

List Command Line Exclusions

Purpose:

Generate a list of exclusions which have been added via the command line interface. This does not include local exclusions created by the user via the *Malwarebytes Anti-Exploit* user interface.

Usage:

```
mbae-cli /excl-list <file>
```

Arguments:

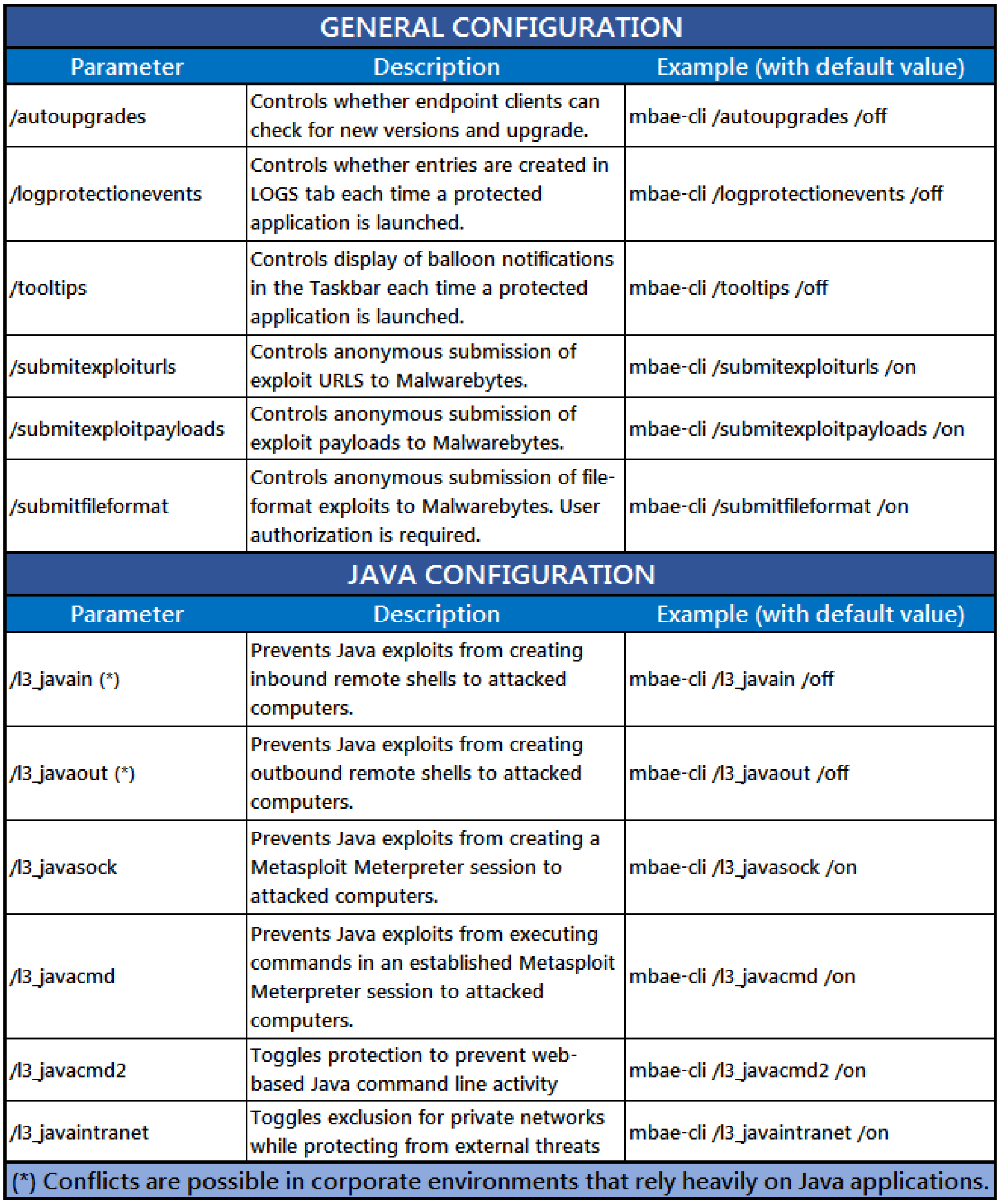
<file> Output file to be generated.

Example:

```
mbae-cli /excl-list d:\protection\mbae-exclude.txt
```

Parameters for General Configuration

A single parameter may be used with each execution of an *mbae-cli* command, and a return code will be returned in response to the command. The following is a list of all parameters that can be used as part of an *mbae-cli* command to perform general configuration of *Malwarebytes Anti-Exploit*.



GENERAL CONFIGURATION

Parameter	Description	Example (with default value)
/autoupgrades	Controls whether endpoint clients can check for new versions and upgrade.	mbae-cli /autoupgrades /off
/logprotectionevents	Controls whether entries are created in LOGS tab each time a protected application is launched.	mbae-cli /logprotectionevents /off
/tooltips	Controls display of balloon notifications in the Taskbar each time a protected application is launched.	mbae-cli /tooltips /off
/submitexploiturls	Controls anonymous submission of exploit URLS to Malwarebytes.	mbae-cli /submitexploiturls /on
/submitexploitpayloads	Controls anonymous submission of exploit payloads to Malwarebytes.	mbae-cli /submitexploitpayloads /on
/submitfileformat	Controls anonymous submission of file-format exploits to Malwarebytes. User authorization is required.	mbae-cli /submitfileformat /on

JAVA CONFIGURATION

Parameter	Description	Example (with default value)
/l3_javain (*)	Prevents Java exploits from creating inbound remote shells to attacked computers.	mbae-cli /l3_javain /off
/l3_javaout (*)	Prevents Java exploits from creating outbound remote shells to attacked computers.	mbae-cli /l3_javaout /off
/l3_javasock	Prevents Java exploits from creating a Metasploit Meterpreter session to attacked computers.	mbae-cli /l3_javasock /on
/l3_javacmd	Prevents Java exploits from executing commands in an established Metasploit Meterpreter session to attacked computers.	mbae-cli /l3_javacmd /on
/l3_javacmd2	Toggles protection to prevent web-based Java command line activity	mbae-cli /l3_javacmd2 /on
/l3_javaintranet	Toggles exclusion for private networks while protecting from external threats	mbae-cli /l3_javaintranet /on

(*) Conflicts are possible in corporate environments that rely heavily on Java applications.

Parameters for Advanced Configuration

The following are all parameters that can be used as part of an *mbae-cli* command to perform advanced configuration of exploit mitigation techniques. A return code is returned in response to each command. Commands use the following structure:

```
mbae-cli /technique [/on|/off] /family
```

You may turn each mitigation technique on or off, and you may specify the **/family** which the technique should be applied to, using the following values.

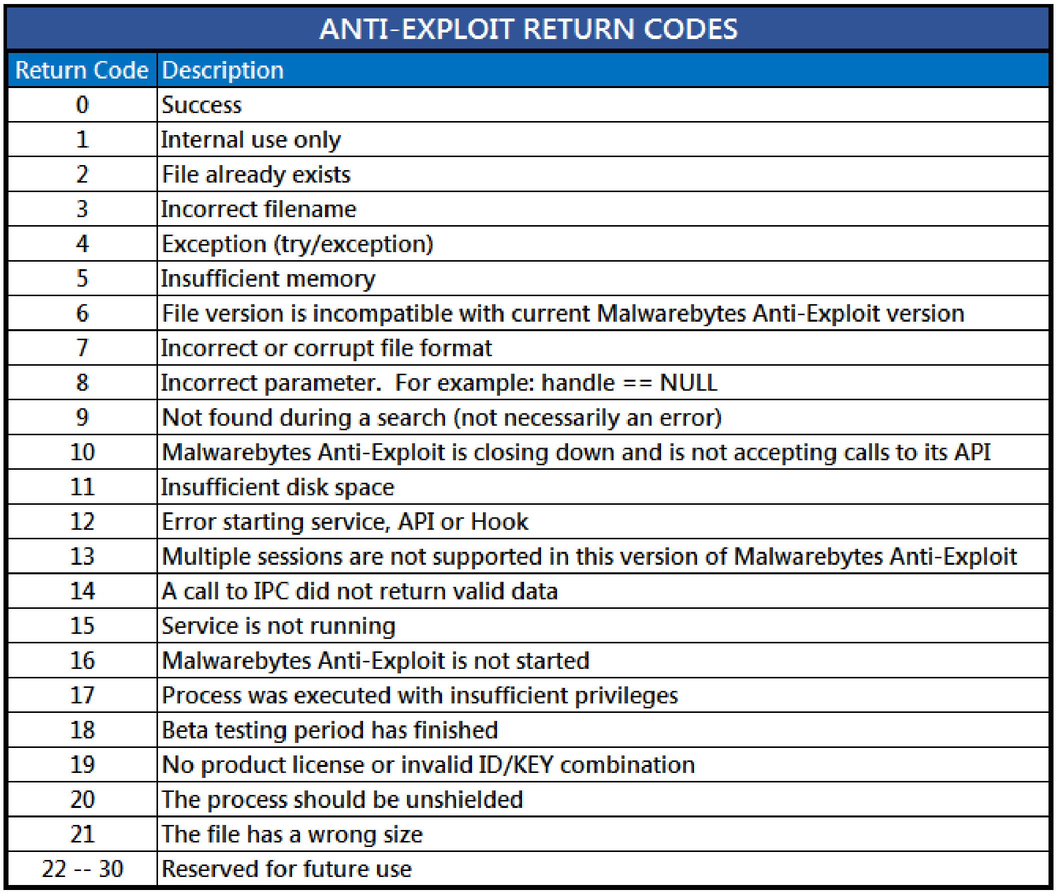
- /browser
- /chromebrowser
- /pdfreader
- /office
- /mediaplayer
- /java
- /other

If the same mitigation technique is to be applied to a second application family, the command should be issued again, specifying the additional family to be protected by this technique.

ADVANCED CONFIGURATION		
Technique	Description	Example
/l0_dep	DEP Enforcement	mbae-cli /l0_dep /on /other
/l0_ah	Anti-Heap Spraying Enforcement	mbae-cli /l0_ah /on /mediaplayer
/l0_dah	Dynamic Anti-Heap Spraying Enforcement	mbae-cli /l0_dah /off /browser
/l0_baslr	Bottom-Up ASLR Enforcement	mbae-cli /l0_baslr /off /office
/l0_vb_disable	Disable Internet Explorer VB Scripting	mbae-cli /l0_vb_disable /on /browser
/l0_xmlhttp	Anti-Exploit Fingerprinting Detection	mbae-cli /l0_xmlhttp /off /browser
/l1_dep	DEP Bypass Protection	mbae-cli /l1_dep /on /other
/l1_wpm	Memory Patch Hijacking Protection	mbae-cli /l1_wpm /on /other
/l1_piv	Stack Pivoting Protection	mbae-cli /l1_piv /off /other
/l1_ropc32	CALL ROP Gadget Detection (32-bit)	mbae-cli /l1_ropc32 /off /pdfreader
/l1_ropc64	CALL ROP Gadget Detection (64-bit)	mbae-cli /l1_ropc64 /off /pdfreader
/l1_ropr32	RET ROP Gadget Detection (32-bit)	mbae-cli /l1_ropr32 /off /pdfreader
/l1_ropr64	RET ROP Gadget Detection (64-bit)	mbae-cli /l1_ropr64 /off /pdfreader
/l2_caller	Malicious Memory Caller Protection	mbae-cli /l2_caller /off /other
/l3_dll	Malicious LoadLibrary Protection	mbae-cli /l3_dll /on /browser
/l3_vb_exec	Protection for Internet Explorer VB Scripting	mbae-cli /l3_vb_exec /off /browser
/l3_msgbox	Protection for MessageBox Payload	mbae-cli /l3_msgbox /on /browser
/l3_office_wmi	Protection against Office WMI abuse	mbae-cli /l3_office_wmi /off /office
/l3_office_vba7	Protection against Office VBA7 abuse	mbae-cli /l3_office_vba7 /off /office

Return Codes

Following is a list of all return codes that *Malwarebytes Anti-Exploit* will return in response to commands which have been executed. Return codes may be evaluated based on testing of environmental variable **%ERRORLEVEL%**.



ANTI-EXPLOIT RETURN CODES	
Return Code	Description
0	Success
1	Internal use only
2	File already exists
3	Incorrect filename
4	Exception (try/exception)
5	Insufficient memory
6	File version is incompatible with current Malwarebytes Anti-Exploit version
7	Incorrect or corrupt file format
8	Incorrect parameter. For example: handle == NULL
9	Not found during a search (not necessarily an error)
10	Malwarebytes Anti-Exploit is closing down and is not accepting calls to its API
11	Insufficient disk space
12	Error starting service, API or Hook
13	Multiple sessions are not supported in this version of Malwarebytes Anti-Exploit
14	A call to IPC did not return valid data
15	Service is not running
16	Malwarebytes Anti-Exploit is not started
17	Process was executed with insufficient privileges
18	Beta testing period has finished
19	No product license or invalid ID/KEY combination
20	The process should be unshielded
21	The file has a wrong size
22 -- 30	Reserved for future use

As the above table shows, the first return code (return code 0) means the operation was successful. In most cases, it will be the desired response to an mbae-cli command. Return codes 22 through 30 have been allocated but are not currently in use. Any return code value higher than 30 is a Microsoft operating system error, and has been shifted higher by 30 (i.e. return code 47 is actually Microsoft error code 17).

Examples

Perhaps the best way to illustrate usage of mbae-cli is to provide examples of how it can be used. These examples are simplistic in nature, and have been chosen to illustrate successful and unsuccessful results. In both cases, mbae-cli is being executed from the Windows command line interface (cmd.exe), invoked with Administrator privileges. The system prompt will be displayed in abbreviated form for clarity.

7.5.1 Start Malwarebytes Anti-Exploit

In the first example, we will start *Malwarebytes Anti-Exploit* protection, verify success/failure, and repeat the process.

```
1:      C:\>mbae-cli /start
2:      C:\>echo %errorlevel%
        0
3:      C:\>mbae-cli /start
4:      C:\>echo %errorlevel%
        0
```

In line 1, protection was started. The return code (line 2) was returned as 0, indicating success. Another start command was issued in line 3, and the return code again indicated success. While protection had been successfully started already, the second attempt did not consider the original state – only that the intent was to start protection and that the result was as intended.

Locally Exclude a File from Malwarebytes Anti-Exploit Protection

In this example, a file will be added to the local exclusion list. In order to do this, a MD5 hash of the file was required to properly identify the file, and this was calculated prior to execution of this example. Once the file has been added to the local exclusion list, a second attempt will be made to perform the same operation.

```
1:      C:\>mbae-cli /excl-add 02cc452c1972995048eac6f3ae4477f6
2:      C:\>echo %errorlevel%
        0
3:      C:\>mbae-cli /excl-add 02cc452c1972995048eac6f3ae4477f6
4:      C:\>echo %errorlevel%
        2
```

Line 1 shows the command to exclude a file, with the file's MD5 hash used for identification purposes. Interrogating the return code shows a successful exclusion. Line 3 is a second attempt to exclude the file. Line 4 returns return code 2. Referring to the above table, return code 2 corresponds to the error "File already exists."

Exclusions

Usage of mbae-cli.exe is intended to be performed by a system administrator, whether it be from the command line, or via scripts executed from a network agent. Commands executed in this manner have higher privileges than those executed from within the *Malwarebytes Anti-Exploit* graphical user interface. These primarily relate to the treatment of exclusions. Following are important distinctions to keep in mind.

1. Not all detections can be excluded. Only Layer 3 detections with file name, path and MD5 can be excluded.
2. Exclusions added by administrator via mbae-cli.exe only include MD5 and will show up in the GUI EXCLUSIONS tab as "Managed by IT Admin". These exclusions cannot be deleted by endpoint users. They can only be deleted by mbae-cli.exe.
3. Local exclusions may have different file names/paths even though they share the same MD5 (i.e. same file).
4. In case (3) the first added exclusion file name and path will be shown (only for local exclusions).
5. If a user or administrator tries to add an exclusion with the same MD5 as a previously-added exclusion, an error message will be returned ("file is already excluded") or return code (in the case of mbae-cli.exe).
6. An administrator may delete via mbae-cli.exe a local exclusion previously added by the user.
7. A local user may not delete an exclusion added by administrator via mbae-cli.exe.

Verifying Program Functionality

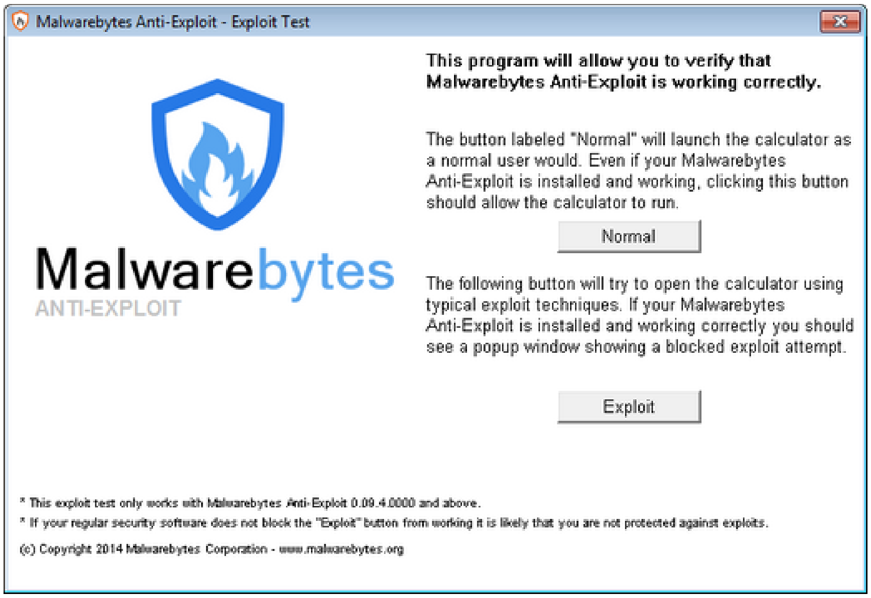
After installing *Malwarebytes Anti-Exploit*, you may wish to run a few tests so that you can see for yourself that it is doing its part to protect you and your endpoint. There are two methods you can use to confirm its functionality.

Exploit Tester

Malwarebytes has made a simple exploit tester available on our public forums for download. The forum post provides a download link and an explanation of the tester's behavior. The link to read about (and download) the tester is:

https://forums.malwarebytes.com/index.php?showtopic=139368

After downloading and extracting the exploit tester to a directory of your choice, double click it to launch the tester. You will see the exploit tester as shown below.



This program will allow you to verify that Malwarebytes Anti-Exploit is working correctly.

The button labeled "Normal" will launch the calculator as a normal user would. Even if your Malwarebytes Anti-Exploit is installed and working, clicking this button should allow the calculator to run.

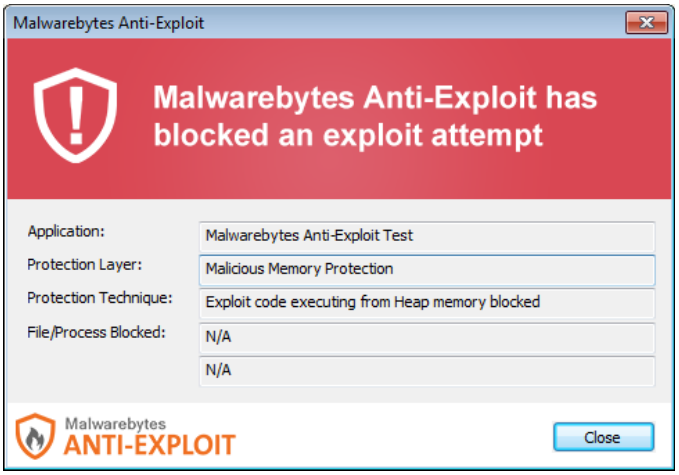
The following button will try to open the calculator using typical exploit techniques. If your Malwarebytes Anti-Exploit is installed and working correctly you should see a popup window showing a blocked exploit attempt.

* This exploit test only works with Malwarebytes Anti-Exploit 0.09.4.0000 and above.

* If your regular security software does not block the "Exploit" button from working it is likely that you are not protected against exploits.

(c) Copyright 2014 Malwarebytes Corporation - www.malwarebytes.org

The screenshot provides instructions which allow you to perform the test. After clicking the **Exploit** button, you will see results of the test, as shown below.



Malwarebytes Anti-Exploit has blocked an exploit attempt

Application: Malwarebytes Anti-Exploit Test

Protection Layer: Malicious Memory Protection

Protection Technique: Exploit code executing from Heap memory blocked

File/Process Blocked: N/A

N/A

This is the same message that you would see during normal program operation if an exploit attempt has been detected and blocked.

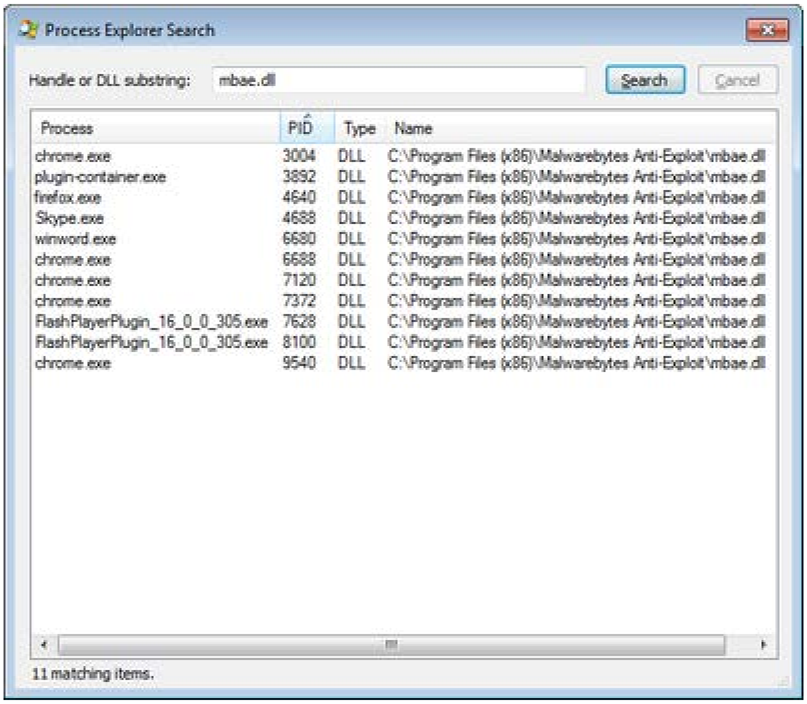
DLL Injection Verification

This method – which is also described in the forum post (referred to in the previous section) – provides a more technical approach to verifying *Malwarebytes Anti-Exploit* functionality. When protecting an application, *Malwarebytes Anti-Exploit* uses a method referred to as DLL Injection. This method allows *Malwarebytes Anti-Exploit* to monitor input/output functionality of protected programs to guard against potentially malicious activity.

You can use a task management utility such as *Process Explorer* to view running processes and tasks. It is downloadable at the following link:

http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx

The utility's Find function will allow you to investigate DLL usage in running processes. The screenshot below shows an example of this.



Handle or DLL substring: mbae.dll

Process	PID	Type	Name
chrome.exe	3004	DLL	C:\Program Files (x86)\Malwarebytes Anti-Exploit\mbae.dll
plugin-container.exe	3892	DLL	C:\Program Files (x86)\Malwarebytes Anti-Exploit\mbae.dll
firefox.exe	4640	DLL	C:\Program Files (x86)\Malwarebytes Anti-Exploit\mbae.dll
Skype.exe	4688	DLL	C:\Program Files (x86)\Malwarebytes Anti-Exploit\mbae.dll
winword.exe	6680	DLL	C:\Program Files (x86)\Malwarebytes Anti-Exploit\mbae.dll
chrome.exe	6688	DLL	C:\Program Files (x86)\Malwarebytes Anti-Exploit\mbae.dll
chrome.exe	7120	DLL	C:\Program Files (x86)\Malwarebytes Anti-Exploit\mbae.dll
chrome.exe	7372	DLL	C:\Program Files (x86)\Malwarebytes Anti-Exploit\mbae.dll
FlashPlayerPlugin_16_0_0_305.exe	7628	DLL	C:\Program Files (x86)\Malwarebytes Anti-Exploit\mbae.dll
FlashPlayerPlugin_16_0_0_305.exe	8100	DLL	C:\Program Files (x86)\Malwarebytes Anti-Exploit\mbae.dll
chrome.exe	9540	DLL	C:\Program Files (x86)\Malwarebytes Anti-Exploit\mbae.dll

11 matching items.

Here, we have run a search for mbae.dll. This is the dynamic link library (DLL) that *Malwarebytes Anti-Exploit* uses to inject itself into applications which have been designated for protection. In this instance, there are ten protected processes as shown in search results.

Logs

Malwarebytes Anti-Exploit maintains operational information in a number of program logs. A few of these logs are pertinent here, as they maintain dynamic information on the application. In addition, you may be called upon to provide information from these logs to Malwarebytes Customer Success if you require technical support. These logs are stored in **%AllUsersProfile%\Malwarebytes Anti-Exploit**. That translates to:

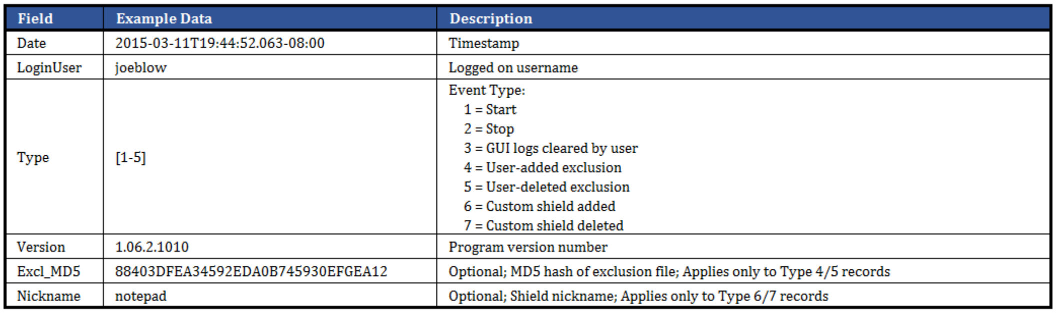
Windows XP:	**C:\Documents and Settings\All Users\Application Data\Malwarebytes Anti-Exploit**
Windows Vista/7/8:	**C:\ProgramData\Malwarebytes Anti-Exploit**

Please note that **%AllUsersProfile%** may map to a different directory on a endpoint that is configured for a language other than English. Following is a complete listing of all log files related to *Malwarebytes Anti-Exploit*.

Log File	Purpose
applications.dat	List of protected applications
exclusions.dat	Whitelist. Files excluded by the user or administrator
mbae-alert.log	Alert details; Can be imported into centralized reporting platform
mbae-default.log	Internal troubleshooting information
mbae-service.log	System events and information; Can be imported into centralized reporting platform
mbae-config.dat	Global Settings; Protection enabled, GUI enabled, Alerts enabled, etc.
mbae-report.dat	Report displayed by GUI application
mbae-svc.dat	Report pending to be notified to GUI application
mbae-protector.xpe	For use by Malwarebytes Tech Support for troubleshooting
dyn-config.dat	Dynamic configuration file (for troubleshooting use by Malwarebytes Customer Success)

mbae-service.log

This file contains detailed information pertaining to administration of *Malwarebytes Anti-Exploit*. Contents of this file are of value if you are troubleshooting a *Malwarebytes Anti-Exploit* control issue.



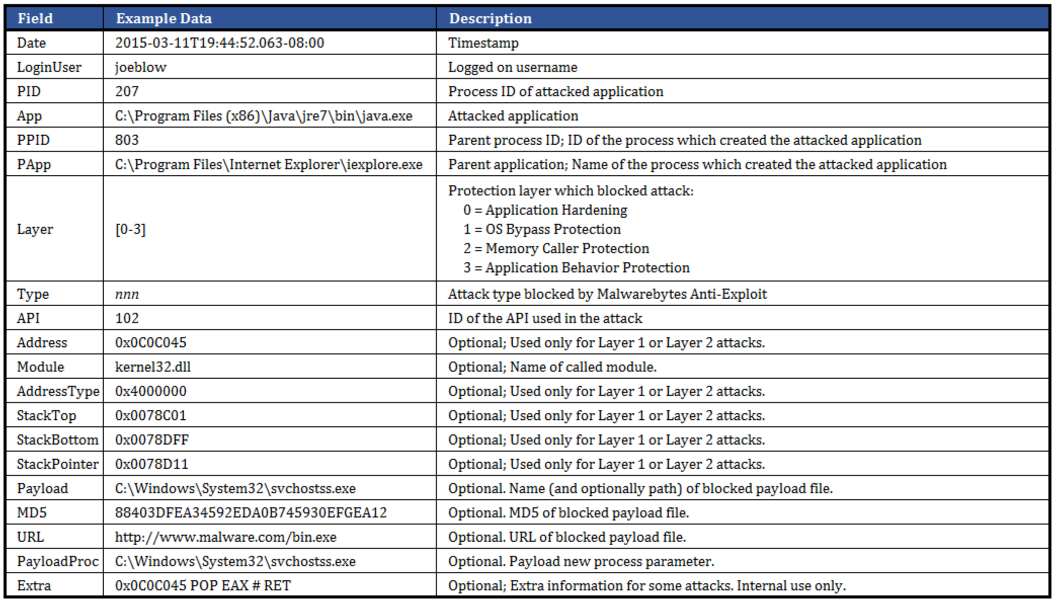
Field	Example Data	Description
Date	2015-03-11T19:44:52.063-08:00	Timestamp
LoginUser	joeblow	Logged on username
Type	[1-5]	Event Type: 1 = Start 2 = Stop 3 = GUI logs cleared by user 4 = User-added exclusion 5 = User-deleted exclusion 6 = Custom shield added 7 = Custom shield deleted
Version	1.06.2.1010	Program version number
Excl_MD5	88403DFEA34592EDA0B745930EFGEA12	Optional; MD5 hash of exclusion file; Applies only to Type 4/5 records
Nickname	notepad	Optional; Shield nickname; Applies only to Type 6/7 records

Seven sample records are shown here, illustrating each of the event types.

```
"2015-03-11T19:44:52.863-08:00";"joeblow";"1";"1.06.2.1010";"";””

"2015-03-11T21:10:45.123-08:00";"joeblow";"2";"1.06.2.1010";"";””

"2015-03-12T08:08:08.008-08:00";"joeblow";"3";"1.06.2.1010";"";””

"2015-03-12T21:15:35.321-
08:00";"joeblow";"4";"1.06.2.1010";"88403DFEA34592EDA0B745930EFGEA12";””

"2015-03-12T23:22:06.566-
08:00";"joeblow";"5";"1.06.2.1010";"88403DFEA34592EDA0B745930EFGEA12";””

"2015-03-13T06:54:43.127+08:00";"joeblow";"6";"1.06.2.1010";"";"notepad"

"2015-03-13T06:59:31.763+08:00";"joeblow";"7";"1.06.2.1010";"";"notepad"
```

mbae-alert.log

This file contains detailed information pertaining to each exploit blocking event from *Malwarebytes Anti-Exploit*.



Field	Example Data	Description
Date	2015-03-11T19:44:52.063-08:00	Timestamp
LoginUser	joeblow	Logged on username
PID	207	Process ID of attacked application
App	C:\Program Files (x86)\Java\jre7\bin\java.exe	Attacked application
PPID	803	Parent process ID; ID of the process which created the attacked application
PApp	C:\Program Files\Internet Explorer\iexplore.exe	Parent application; Name of the process which created the attacked application
Layer	[0-3]	Protection layer which blocked attack: 0 = Application Hardening 1 = OS Bypass Protection 2 = Memory Caller Protection 3 = Application Behavior Protection
Type	*nnn*	Attack type blocked by Malwarebytes Anti-Exploit
API	102	ID of the API used in the attack
Address	0x0C0C045	Optional; Used only for Layer 1 or Layer 2 attacks.
Module	kernel32.dll	Optional; Name of called module.
AddressType	0x4000000	Optional; Used only for Layer 1 or Layer 2 attacks.
StackTop	0x0078C01	Optional; Used only for Layer 1 or Layer 2 attacks.
StackBottom	0x0078DFF	Optional; Used only for Layer 1 or Layer 2 attacks.
StackPointer	0x0078D11	Optional; Used only for Layer 1 or Layer 2 attacks.
Payload	C:\Windows\System32\svchostss.exe	Optional. Name (and optionally path) of blocked payload file.
MD5	88403DFEA34592EDA0B745930EFGEA12	Optional. MD5 of blocked payload file.
URL	http://www.malware.com/bin.exe	Optional. URL of blocked payload file.
PayloadProc	C:\Windows\System32\svchostss.exe	Optional. Payload new process parameter.
Extra	0x0C0C045 POP EAX # RET	Optional; Extra information for some attacks. Internal use only.

Following is a sample of how log data appears in its native format. Please note that all data is surrounded by quotation marks for consistent handling by other programs. Each log record is a single line of data. It is shown in multiple lines due to line wrapping.

```
"2015-03-1T19:44:52.863-08:00";"joeblow";"207";"C:\Program Files
(x86)\Java\jre7\bin\java.exe";"803";"C:\Program Files\Internet
Explorer\iexplore.exe";"3";"701";"102";"0x0C0C045";"kernel32.dll";"0x4000000";"
0x0078C01";"0x0078DFF";"0x0078D11";"C:\Windows\System32\svchostss.exe";"88403DF
EA34592EDA0B745930EFGEA12";"http://www.malware.com/bin.exe";"C:\Windows\System3
2\svchostss.exe";"0x0C0C045 POP EAX # RET"
```

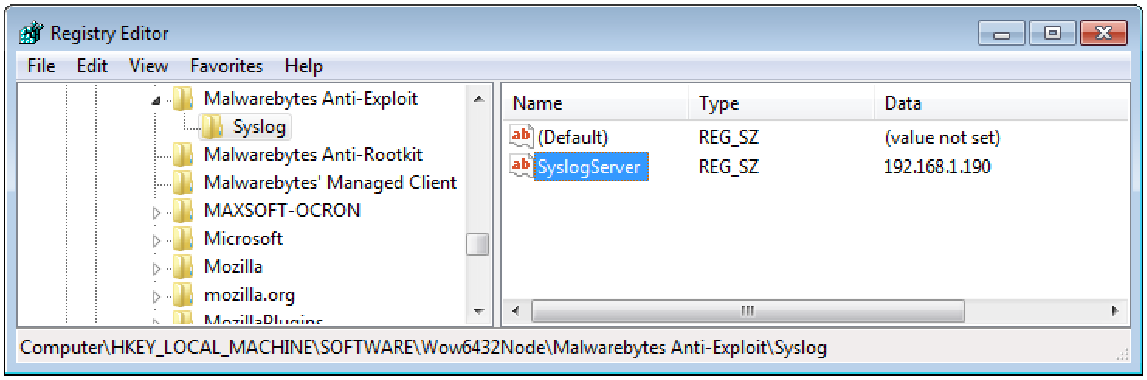
Submitting Events to a Syslog Server

The *Malwarebytes Anti-Exploit* standalone client can be configured to send alert and service events to a Syslog server. In order to activate this feature, the following registry key needs to be added:

- 32-bit Operating Systems:
  - HKLM\SOFTWARE\Malwarebytes Anti-Exploit\Syslog\SyslogServer
- 64-bit Operating Systems:
  - HKLM\SOFTWARE\Wow6432Node\Malwarebytes Anti-Exploit\Syslog\SyslogServer

PLEASE NOTE: When used in conjunction with *Malwarebytes Management Console*, the console's syslog feature offers greater versatility and should be used instead.

The string value (type REG_SZ) for registry entry *SyslogServer* must contain a valid IP or hostname of the Syslog server. If an invalid or no entry is found, Syslog functionality will not activate itself. A screenshot of this registry setting is shown below.



Name	Type	Data
(Default)	REG_SZ	(value not set)
SyslogServer	REG_SZ	192.168.1.190

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Malwarebytes Anti-Exploit\Syslog

Other parameters may also be added under the *Syslog*\ registry key to further customize Syslog functionality, shown as follows:

REGISTRY KEY	TYPE	DATA
SyslogServer	REG_SZ	If invalid or no entry, Syslog is disabled
Port	REG_DWORD	514 (default)
Protocol	REG_DWORD	0=TCP / 1=UDP (default UDP)
Facility	REG_DWORD	1 (value based on RFC 5424)
SeverityForAlerts	REG_DWORD	1 (value based on RFC 5424)
SeverityForNotifications	REG_DWORD	6 (value based on RFC 5424)
Application	REG_SZ	MBAE