Malwarebytes Malware Remediation User Guide

Version 2.5.2A

3 February 2016

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.

Contents

1.0 Introduction

Malwarebytes Malware Remediation is designed to allow business users to detect and remove malware from computers. It is built upon the power of our flagship anti-malware product, *Malwarebytes Anti-Malware*, which allows *Malwarebytes Malware Remediation* to succeed where other anti-malware applications often fail.

Malwarebytes Anti-Malware is considered to be the next step in the detection and removal of malware. We have compiled a number of new technologies that are designed to quickly detect, destroy, and prevent malware. *Malwarebytes Anti-Malware* can detect and remove malware that even the most well-known antivirus and antimalware applications on the market today cannot.

Implementation in a portable form provides increased flexibility for IT staff to quickly and easily deploy the product, use it to remediate threats, gather logs, and continue with their daily tasks – all without a large investment in time or resources.

1.1 What's New

The following changes have been made in version 2.5.2 of *Malwarebytes Malware Remediation*.

- Updated program to fetch signature updates over a secure HTTPS channel.

The following changes have been made in version 2.5.1 of *Malwarebytes Malware Remediation*.

- Addressed a memory exception error during a scan which affected systems with multiple users.

The following changes have been made in version 2.5 of *Malwarebytes Malware Remediation*.

- Ability to selectively restore files using two different methods
- Added command to specify program environmental settings
- Added setting to enable/disable color display, which caused issues with some deployment utilities
- Added capability to track endpoint name and IP address in scan log
- Added capability to exclude several types of objects from scanning
- Added CEF-compatible event logging for all program features
- Added program mode to allow scanning of endpoints without remediation or quarantining functionality

1.2 Key Features

Malwarebytes Malware Remediation offers the following key features:

- Selective remediation capability
- Remediation of earlier scan results without requiring a second scan
- Four different types of scans to analyze your computer for malware threats, regardless of whether they are based in memory, file system or registry
- Ability to perform full scans for all local drives
- Ability to utilize Malwarebytes signature updates, assuring that even the newest threats can be detected
- Intelligent heuristics to analyze potential threats when they are designed to evade signatures
- Ability to quarantine detected threats, and to restore if needed
- Ability to deploy product to computers using your preferred methods
- Ability to exclude several object types from scanning
- Command line capabilities allow IT staff to modify certain program configuration settings, execute scans, and gather logs through integration with customer-supplied scripts, batch files, and group policy updates
- Product leaves no lasting footprint on computer
- CEF-compatible event logging
- Ability to use the program in *scan only* or *scan and remediate* mode.

1.3 System Requirements

Following are minimum requirements for a computer on which *Malwarebytes Malware Remediation* may be installed. Please note that these requirements do not include other functionality that the computer is responsible for.

- **Operating System:** Windows 10 (32/64-bit), Windows 8.1 (32/64-bit), Windows 8 (32/64-bit), Windows 7 (32/64-bit), Windows Vista (32/64-bit), Windows XP (Service Pack 2 or later, 32-bit only)
- **CPU:** 800 MHz or faster
- **RAM:** 256 MB (512 MB or more recommended)
- **Free Disk Space:** 20 MB
- **Screen Resolution:** 800x600 or higher
- **Active Internet Connection**, for license validation and threat signature updates
- **USB 2.0 Port** (optional, depending on deployment method)

1.4 External Access Requirements

If your company's Internet access is controlled by a firewall or other access-limiting device, you must grant access for *Malwarebytes Malware Remediation* to reach Malwarebytes services. These are:

https://data.service.malwarebytes.org	Port 443	outbound
https://data-cdn.mbamupdates.com	Port 443	outbound
https://*.mwbsys.com	Port 443	outbound

NOTE: If your firewall rule set cannot accommodate wildcard specifications, please replace the last rule with:

https://keystone.mwbsys.com	Port 443	outbound

2.0 Using Malwarebytes Malware Remediation

Malwarebytes Malware Remediation is designed specifically for use by IT staff. It may be deployed to a computer by local insertion of a USB drive which contains the program, or by pushing the program out to the computer using psexec, Powershell, or any other deployment method which you may currently use.

Section 3 of this guide details all command line functionality which can be used with *Malwarebytes Malware Remediation*.

2.1 License Key Status

Malwarebytes Malware Remediation uses a license key, which was provided to you upon your purchase of the product. Once registered, the license key is considered active for 48 hours – unless a different time interval was specified at time of purchase. Each time the product is used on a computer, license status is checked.

If your license deactivates (times out), you cannot perform critical operations that the product is intended for. If this occurs, you must re-register the product (see section 3.3.1 of this guide for further details). This is to prevent unauthorized use of the product. There is no additional cost to re-register the product.

2.2 Getting Started

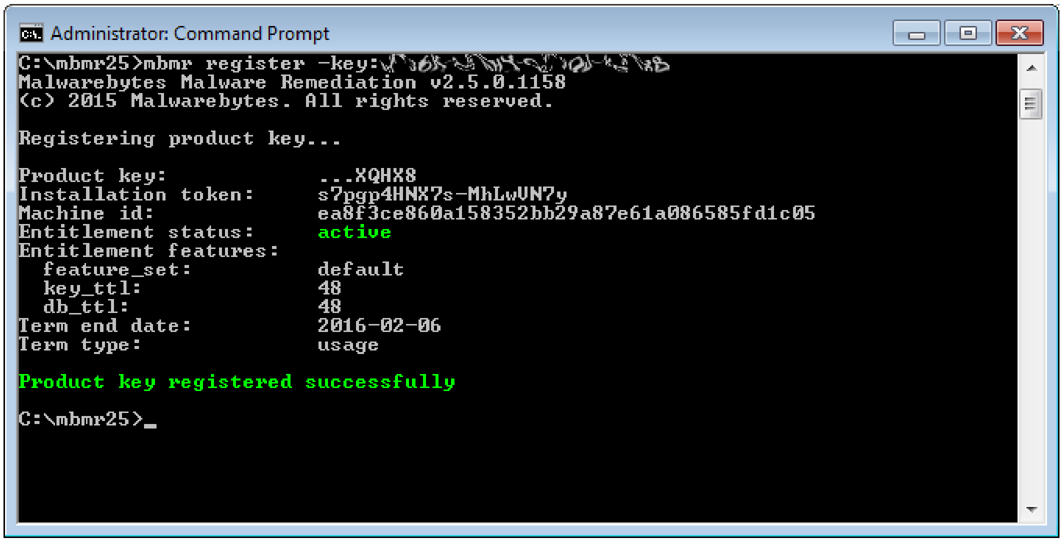
Getting started with *Malwarebytes Malware Remediation* is very simple. Using a computer with a live Internet connection, access a Windows command line prompt and issue the following commands:

```
<path>mbmr register –key:<prodkey>
<path>mbmr update
```

`<path>` indicates the drive and folder where *Malwarebytes Malware Remediation* is located (primary drive, secondary drive, or USB drive). You may also navigate to that location from the command line if you wish..

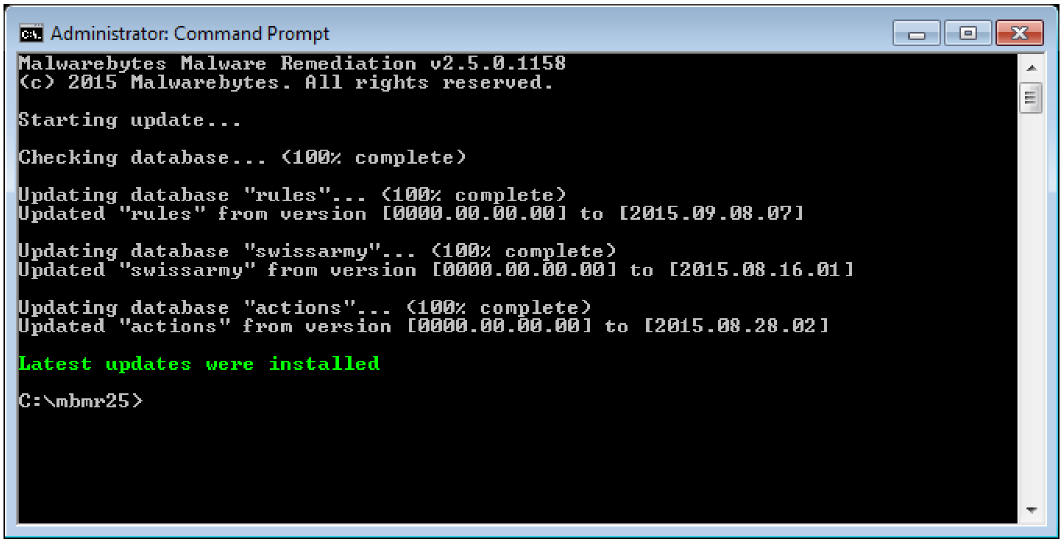
Please note: You must substitute your license key for ***<prodkey>*** in the above example. To demonstrate, *Malwarebytes Malware Remediation* was saved to folder *mbmr25* on drive C, then activated using a test license key.

Please note that some deployment utilities (e.g. *psexec*) do not support color display as is shown below. When using a utility that does not support color, program messages are displayed in standard monochrome video. Please refer to Section 3.3.7 for explicit settings with regard to color output.



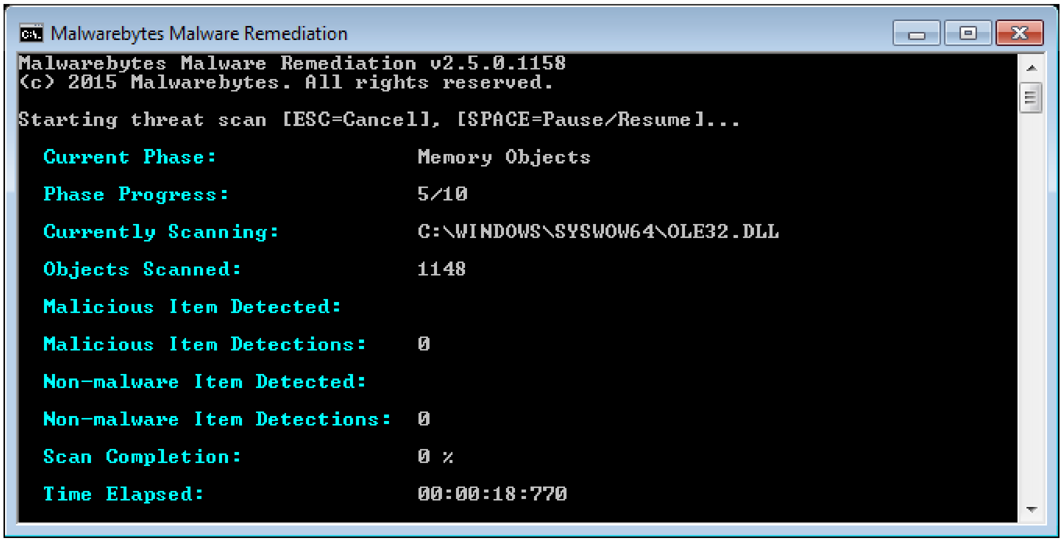
```
C:\mbmr25>mbmr register -key:[redacted]
Malwarebytes Malware Remediation v2.5.0.1158
(c) 2015 Malwarebytes. All rights reserved.

Registering product key...

Product key:            ...XQHX8
Installation token:     s7pgp4HNX7s-MhLwUN7y
Machine id:             ea8f3ce860a158352bb29a87e61a086585fd1c05
Entitlement status:     active
Entitlement features:
  feature_set:          default
  key_ttl:              48
  db_ttl:               48
Term end date:          2016-02-06
Term type:              usage

Product key registered successfully

C:\mbmr25>_
```

Once the program has been activated, it is necessary to load threat signatures from Malwarebytes servers into the product. This enables *Malwarebytes Malware Remediation* to detect threats using the most current reference material available. Malware threat signatures are updated several times daily, and rootkit signatures are updated as required.



```
Malwarebytes Malware Remediation v2.5.0.1158
(c) 2015 Malwarebytes. All rights reserved.

Starting update...

Checking database... (100% complete)

Updating database "rules"... (100% complete)
Updated "rules" from version [0000.00.00.00] to [2015.09.08.07]

Updating database "swissarmy"... (100% complete)
Updated "swissarmy" from version [0000.00.00.00] to [2015.08.16.01]

Updating database "actions"... (100% complete)
Updated "actions" from version [0000.00.00.00] to [2015.08.28.02]

Latest updates were installed

C:\mbmr25>
```

Once threat signatures have been updated in your local installation, you can use *Malwarebytes Malware Remediation* to detect and remove malware from your computer. Following is a screenshot of a scan in process.



```
Malwarebytes Malware Remediation v2.5.0.1158
(c) 2015 Malwarebytes. All rights reserved.

Starting threat scan [ESC=Cancel], [SPACE=Pause/Resume]...

  Current Phase:                Memory Objects

  Phase Progress:               5/10

  Currently Scanning:           C:\WINDOWS\SYSWOW64\OLE32.DLL

  Objects Scanned:              1148

  Malicious Item Detected:

  Malicious Item Detections:    0

  Non-malware Item Detected:

  Non-malware Item Detections:  0

  Scan Completion:              0 %

  Time Elapsed:                 00:00:18:770
```

While a scan is in process, this screen is constantly updated. Please note the line titled **Phase Progress**. There are ten (10) phases of a scan, which are:

1 Pre-scan Operations
2 System Drivers
3 Master Boot Record
4 Physical Sectors
5 Memory Objects
6 Startup Objects
7 Registry Objects
8 Heuristic Analysis
9 Filesystem Objects
10 Scan Complete

This line will reflect each as they are in process. Each phase of the scan requires a different amount of time to complete, so this cannot be used as a method of estimating how long a scan will take to complete.

2.2.1 Interactions with Anti-Rootkit Scanning

When called upon to perform anti-rootkit scanning, *Malwarebytes Malware Remediation* uses a special driver which may be incompatible with driver versions used by *Malwarebytes Anti-Malware* and/or *Malwarebytes Anti-Rootkit*. If this occurs, *Malwarebytes Malware Remediation* must unload the incompatible driver so that it may load its own version. The only way the driver can be gracefully unloaded is by terminating the Malwarebytes program which loaded the driver.

This will only occur during active scans by *Malwarebytes Anti-Rootkit* or by *Malwarebytes Anti-Malware* version 2.0 and above (free, trial or premium), or *Malwarebytes Anti-Malware* version 2.0 and above using real-time protection (trial or premium).

If *Malwarebytes Anti-Malware* was terminated to allow *Malwarebytes Malware Remediation* to run **and** a reboot was required to remove threats detected by *Malwarebytes Malware Remediation*, protection will return to its normal state after the reboot. If a reboot is not required, you must manually restart *Malwarebytes Anti-Malware* to regain the real-time protection that was turned off temporarily.

2.3 Remediation Now or Later?

Malwarebytes Malware Remediation offers four types of scans which may be executed. It also offers the capability to automatically decide which threats should be removed, or to allow the user to override default remediation methods selected by the program. This may be valuable in many circumstances, including:

- General assessment of a computer's health with regard to malware
- Ability to collect and analyze evidence of infections
- Exclusion of known false positives

Scans may be executed for the purpose of remediation, or for diagnostic discovery. A remediation scan combines a scan with a remediation method, so that detected threats may be immediately cleaned from the computer. A diagnostic scan omits the remediation method, so that a scan is executed and results are reported. The user may then determine how to proceed. This may be valuable if you wish to assess the general health of a computer, or if you wish to collect data about one or more computers without eliminating evidence that you may wish to retain.

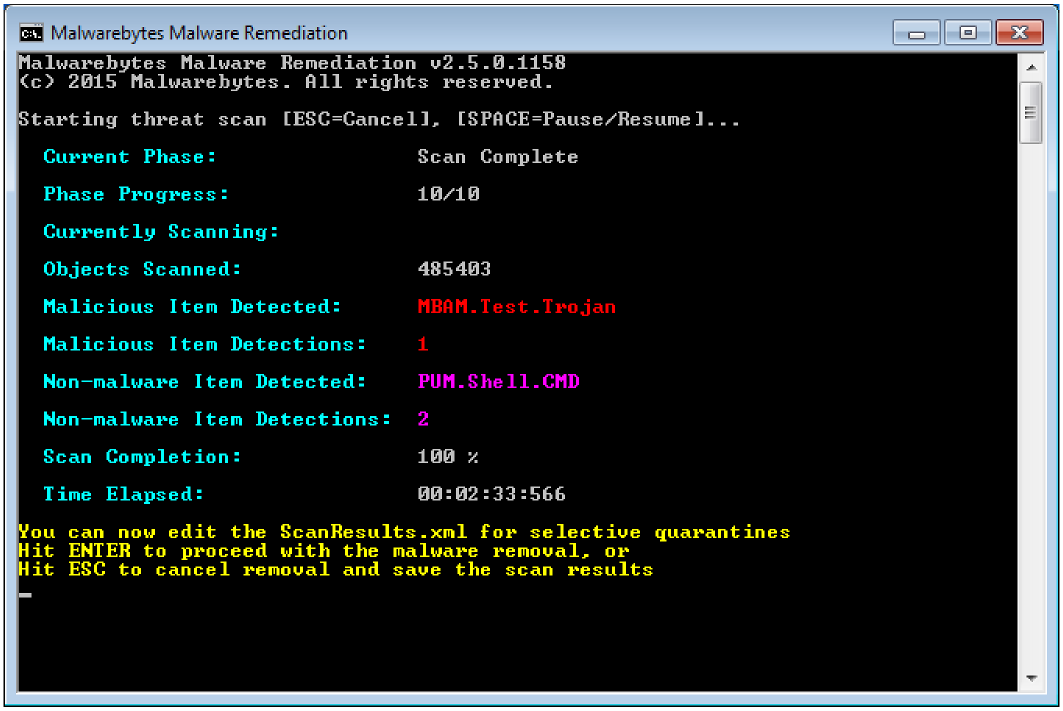
These capabilities are listed below.

2.3.1 Remediation Scan

A *remediation scan* combines a scan with an automatic remediation method, so that detected threats may be immediately cleaned from the computer. No user intervention is required once the scan begins.

2.3.2 Selective Remediation Scan

A *selective remediation scan* combines a scan with a remediation method, but also includes the **–confirmremove** parameter. This additional specification causes the scan process to pause before any remediation method has taken place. The scan has saved "working data" pertaining to any detected threats in an intermediate file named **ScanResults.xml**, which is saved in the directory that *Malwarebytes Malware Remediation* was executed from. The user may then open and inspect this file (using an XML processor or editor of their choice) to determine if any detected threats should not be remediated. The following screenshot show a selective remediation scan, paused at the point where user inspection and intervention is possible, followed by a sample of the ScanResults.xml file which this scan generated. The XML file presentation has been modified to improve readability.



```
Malwarebytes Malware Remediation v2.5.0.1158
(c) 2015 Malwarebytes. All rights reserved.

Starting threat scan [ESC=Cancel], [SPACE=Pause/Resume]...

  Current Phase:                Scan Complete

  Phase Progress:               10/10

  Currently Scanning:

  Objects Scanned:              485403

  Malicious Item Detected:      MBAM.Test.Trojan

  Malicious Item Detections:    1

  Non-malware Item Detected:    PUM.Shell.CMD

  Non-malware Item Detections:  2

  Scan Completion:              100 %

  Time Elapsed:                 00:02:33:566

You can now edit the ScanResults.xml for selective quarantines
Hit ENTER to proceed with the malware removal, or
Hit ESC to cancel removal and save the scan results
_
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<ScanResults>
    <Detections>
        <Detection>
            <Info>
                <Name>MBAM.Test.Trojan</Name>
                <Path>C:\Users\mbuser.NOAH_WIN7X64\Desktop\test-trojan.exe</Path>
                <Hash>a3d4ef3efe8da19549315db36b9719e7</Hash>
                <Class>1</Class>
            </Info>
            <Action>remove</Action>
        </Detection>
        <Detection>
            <Info>
                <Name>PUP.Optional.Dotpitch</Name>
                <Path>C:\Users\mbuser.NOAH_WIN7X64\Desktop\Test_PUP.exe</Path>
                <Hash>2a4db17cd6b5d75f8226399dc3417090</Hash>
                <Class>1</Class>
            </Info>
            <Action>remove</Action>
        </Detection>
        <Detection>
            <Info>
                <Name>PUM.Shell.CMD</Name>
                <Path>HKCU\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON|Shell</Path>
                <Hash>e88fea4324677db940b82c1a58ab9b65</Hash>
                <Class>8</Class>
            </Info>
            <Action>remove</Action>
        </Detection>
    </Detections>
</ScanResults>
```

Please note that each detected threat is documented in the same manner, which allows for easy identification of data within the file. An XML tag titled **<Action>** contains a remediation method, and is preset to the value **remove**. You may not wish to remove the detected threat. Here are the three choices which are available to you.

- **remove:** Delete the malware in its present form, render it harmless, and move it to Quarantine
- **delete:** Delete the malware from the computer without copying it to Quarantine
- **ignore:** Leave the file intact in its current location.

Once you are done inspecting/editing this file, you may resume or cancel the selective remediation scan. If you cancel the remediation phase of the scan, the intermediate file will be retained. If you continue the remediation phase, the intermediate file will be used to control remediation, and will be deleted once remediation is complete.

2.3.3 Diagnostic Scan

When executing a diagnostic scan, do not provide any specifications for remediation of threats detected during the scan. This causes scan results to be saved to intermediate file **ScanResults.xml**, which is saved into the directory that *mbmr* was executed from. Please refer to the previous section for more information on this topic. If you determine that removing threats detected from the last executed scan should be performed, modify this file to suite your needs, and run another scan using the **–removelastscan** parameter to remove detections.

2.4 Excluding Items from Scanning

It is not uncommon to have legitimate items stored on your endpoints which may be identified as threats by anti-virus or anti-malware software. *Malwarebytes Malware Remediation* recognizes that, and offers two methods to exclude those items from scanning. Those methods are:

- **Exclude by specification –** This method allows interactive or scripted exclusion of files, folders, and wildcards.
- **Exclude List –** In addition to the previous method, this method allows exclusion of file extensions, registry keys, registry values, and vendor (the name which Malwarebytes uses to identify threats). Items to be excluded are enclosed in one or more XML files. A sample exclusion list is shown here. Please note that indentation has been added here to aid in understanding.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ScanExclusions>
   <Exclusions>
      <Exclusion>
         <Type>folder</Type>
         <Path>c:\virus\a</Path>
      </Exclusion>
      <Exclusion>
         <Type>wildcard</Type>
         <Path>c:\virus\*trojan*</Path>
      </Exclusion>
      <Exclusion>
         <Type>file</Type>
         <Path>c:\virus\test.exe</Path>
      </Exclusion>
      <Exclusion>
         <Type>regkey</Type>
         <Path>HKLM\SYSTEM\CURRENTCONTROLSET\SERVICES\1394843d</Path>
      </Exclusion>
      <Exclusion>
         <Type>regval</Type>
         <Path>HKCU\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN|DESKBAR</Path>
         <Path>...</Path>
      </Exclusion>
      <Exclusion>
         <Type>vendor</Type>
         <Path>MBAM.Test.Trojan</Path>
      </Exclusion>
      <Exclusion>
         <Type>ext</Type>
         <Path>mp3;flac</Path>
      </Exclusion>
   </Exclusions>
</ScanExclusions>
```

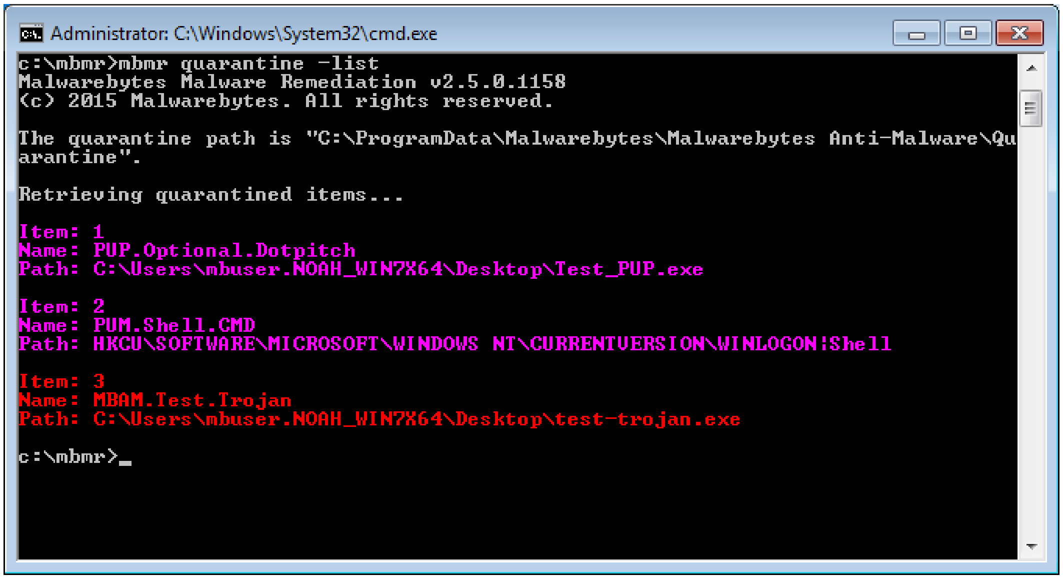
Please see Section 3.3.4 for complete details.

2.5 Restoring Items from Quarantine

Malwarebytes Malware Remediation offers several different methods of restoring items from Quarantine. You may choose any of the following methods:

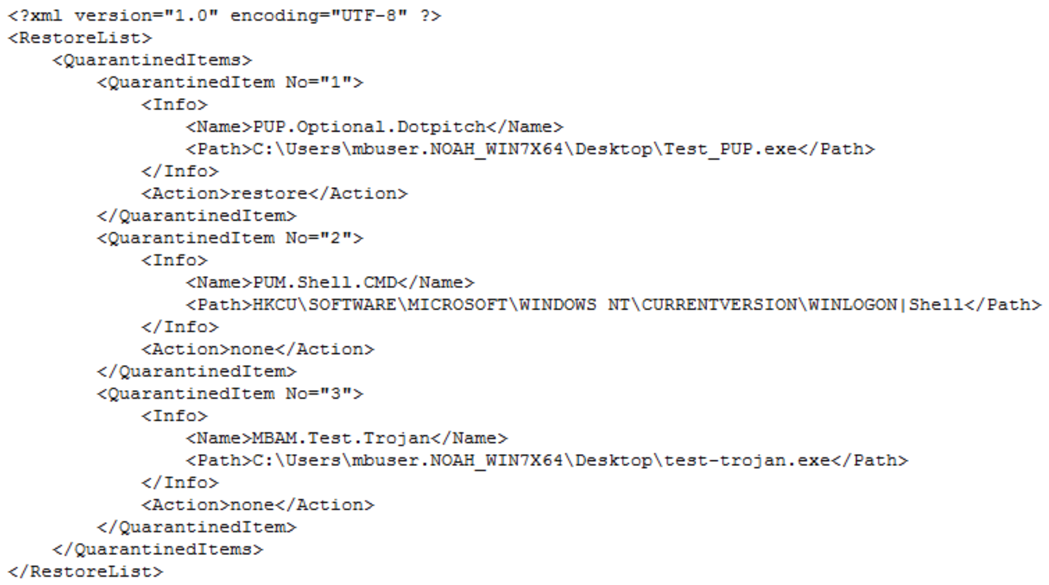
- **Restore all –** Restores all items currently stored in Quarantine to their original locations
- **Restore by index number –** This method utilizes a screen/file-based list to selectively restore items to their original locations. This is typically a manual operation, though it may also be performed using a script. The list of items changes after each restore operation, so the list must be recreated when multiple restore operations are required.
- **Automated restore –** The XML list of items stored in Quarantine is interactively modified to specify which items will be selectively restored. The modified list is then used programmatically to perform the restore operation.

In the screenshot below, command `mbmr quarantine -list` was executed to provide a listing of the contents of Quarantine. This list shows the index number that can be used for restoration, the name of the threat that was placed into Quarantine, and its original location before it was placed in Quarantine.



```
c:\mbmr>mbmr quarantine -list
Malwarebytes Malware Remediation v2.5.0.1158
(c) 2015 Malwarebytes. All rights reserved.

The quarantine path is "C:\ProgramData\Malwarebytes\Malwarebytes Anti-Malware\Qu
arantine".

Retrieving quarantined items...

Item: 1
Name: PUP.Optional.Dotpitch
Path: C:\Users\mbuser.NOAH_WIN7X64\Desktop\Test_PUP.exe

Item: 2
Name: PUM.Shell.CMD
Path: HKCU\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON|Shell

Item: 3
Name: MBAM.Test.Trojan
Path: C:\Users\mbuser.NOAH_WIN7X64\Desktop\test-trojan.exe

c:\mbmr>_
```

The next screenshot shows the contents of file **RestoreList.xml**, which is the file generated at the same time that the screen-based list is created. For demonstration purposes, the file has been edited so that only the first quarantined item will be restored using the XML file as input to command `mbmr quarantine -restorelist`.



```xml
<?xml version="1.0" encoding="UTF-8" ?>
<RestoreList>
    <QuarantinedItems>
        <QuarantinedItem No="1">
            <Info>
                <Name>PUP.Optional.Dotpitch</Name>
                <Path>C:\Users\mbuser.NOAH_WIN7X64\Desktop\Test_PUP.exe</Path>
            </Info>
            <Action>restore</Action>
        </QuarantinedItem>
        <QuarantinedItem No="2">
            <Info>
                <Name>PUM.Shell.CMD</Name>
                <Path>HKCU\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON|Shell</Path>
            </Info>
            <Action>none</Action>
        </QuarantinedItem>
        <QuarantinedItem No="3">
            <Info>
                <Name>MBAM.Test.Trojan</Name>
                <Path>C:\Users\mbuser.NOAH_WIN7X64\Desktop\test-trojan.exe</Path>
            </Info>
            <Action>none</Action>
        </QuarantinedItem>
    </QuarantinedItems>
</RestoreList>
```

The above is merely an introduction to the methods. Please consult Section 3.3.6 of this guide for specific usage guidance.

3.0 Command Line Parameters

Malwarebytes Malware Remediation supports a variety of command line parameters, which can be used from a command prompt, batch file or script. When used from a script, additional commands may be required to support the scripting model being used.

3.1 Conventions

The command line structure uses modifiers. These are shown as hyphens (-) immediately preceding parameters. Multiple modifiers may be combined with a parameter. When multiple parameters are used, they must be separated by spaces. In addition, the following conventions are used:

- **text without brackets or braces**
  Items you must type as shown
- **<text inside angle brackets>**
  Required information for which you must supply a value
  Example: **mbmr <*parameter_1*>**
- **[text inside square brackets]**
  Optional items
  Example: **mbmr [*parameter_1*]**
- **Grouping of dots (…)**
  A set of specifications
  Example: **mbmr <*parameter_1*> [*parameter_2*] … [*parameter_n*]**
- **{text inside braces}**
  A set of required items; choose one from the list provided
  Example: **mbmr {0 | 1 | 2 | 3}**
- **vertical bar (|)**
  Separator between mutually exclusive items; choose one
  Example: **mbmr <0 | 1 | 2 | 3>**

3.2 Command Line Overview

Malwarebytes Malware Remediation commands are specified in the following format:

mbmr { register | update | version | scan | errorout | quarantine | settings } [*options*]

Following is a list of high-level commands which may be executed. Each command is detailed in Section 3.3.

register Using your license key, this unlocks the features of *Malwarebytes Malware Remediation*. This will also show license status.

update Updates the product's threat signature databases.

version Displays the program version number.

scan Scans the computer for malware and optionally removes malware found during the scan.

errorout Specifies where error output is directed to.

quarantine Controls program actions related to threat quarantine activities.

settings Used to specify universal program settings. These settings are persistent, and are used for color and proxy functionality.

In addition, you may type **mbmr** without any additional specifications to see a list of valid commands. This list will span multiple windows if the command line is launched to its default size, so you will achieve best results by stretching the window to show more command line dialog at one time.

3.3 Command Line Reference

Commands listed here are listed individually. Each command performs tasks according to parameters. These are primarily used by a system administrator via script, batch file, GPO update, or remote desktop. The admin may configure *Malwarebytes Malware Remediation* to operate as a remote task, invisible to the computer user.

3.3.1 register

Usage:

mbmr register [–key:<*prodkey*>]

Purpose:

Specifies the unique license key assigned to the partner or customer. This will be passed to the licensing server for validation to ensure it is active (non-expired). If the key is valid and the license is active, it will also display status about the license, such as expiration date, volume purchased, volume used, etc.

If the key is active, the local installation will operate with this status for 48 hours (or the time interval specified in your Malwarebytes license agreement). This "Last Known Good" status is persisted on the USB or wherever the binaries are stored. This allows the USB installation to work as if it were fully registered on offline computers or without needing the key. **A live Internet connection is required.**

If **–key** is not specified, license status and the expiration date/time will be displayed. **Please note** that if the key is not active, the user may not update threat signature databases, scan for malware, list contents of quarantine, or restore files from quarantine.

Parameters:

–key:<*prodkey*>

Specification of <*prodkey*>, the license key assigned to the user.

3.3.2 update

Usage:

mbmr update

Purpose:

Updates local threat signature databases. This command will result in an error condition if (a) the license is not active, or (b) if no active Internet connection is available. If threat signature databases have expired (timed out), this command must precede execution of a scan. If a proxy server is needed to access the Internet, you must run the **proxy** command before attempting to perform the update.

Parameters:

none

3.3.3 version

Usage:

mbmr version

Purpose:

Displays the version number of the *Malwarebytes Malware Remediation* installation.

Parameters:

none

3.3.4 scan

Usage:

```
mbmr scan [–full | –threat | –hyper | –path:<path>]
              [–exclude:<paths>]
              [–excludelist:<exclists>]
              [–noarchive]
              [–ignorepu]
              [–stopondetect [–malware:<cnt>] [–pu:<cnt>]]
              [–tag:<tagname>]
              [–stdlog:<filepath>]
              [–noscanresults]
              [–pfi:<secs>]
              [–ark]
              [–remove [–confirmremove] | [–noreboot]
              [–removelastscan [–noreboot]
              [–stdout:{off | detail | summary}]
```

Purpose:

Executes a scan based on parameters specified. If the program license is inactive, attempts to perform a scan will result in an error. Current threat signature databases are also required. If this command is executed without a directive to quarantine detected threats, scan results are saved to file **ScanResults.xml**. This file is found in the same directory that *Malwarebytes Malware Remediation* was executed from.

Parameters:

–full

A full scan includes all scanning capability which *Malwarebytes Malware Remediation* has to offer.

–threat

This will examine "hotspots" on a computer for malware without analyzing the entire computer. If **–remove** is specified, any threats found during the scan will be quarantined.

–hyper

Initiates a scan focusing only on Memory Objects and Heuristics to determine if malware is actively running on the computer. If **–remove** is specified, any threats found during the scan will be quarantined.

–path:<*path*>

Semicolon-delimited list of folder paths to scan on the computer's file system. Use double-quotes for paths that contain spaces. Paths specified are recursive, so subfolders will also be included automatically. The presence of the paths specified will be verified, and an error will result if the paths do not exist. Similarly, an error will be generated if the paths are encrypted, as the program is not capable of decrypting the path. If **–remove** is specified, any threats found during the scan will be quarantined. **Please note** that this method scans only the folder path(s) which have been specified. This scan type **does not** include memory, startup modules or heuristic analysis.

–exclude:<*paths*>

Excludes specific files, folders, and wildcard specifications from scanning. **Please note** that the following rules apply:

- Multiple exclusion items must be separated by semicolons.
- Files specified without a directory path (e.g. file.exe) must be in the directory where *Malwarebytes Malware Remediation* is installed.

- Files and/or folders which contain embedded spaces must be enclosed in double quotes. As an example, references to **`C:\Program Files\file.exe`** should instead be specified as **`“C:\Program Files\file.exe”`**.

–excludelist:<*exclist*>

Provides for exclusions itemized within one or more XML files specified as parameters for this command. Files have no naming convention (filename or extension), but they must be in XML format. The following items may be excluded using exclusion lists:

- Files
- Extensions
- Threat Vendor
- Folders
- Registry Keys
- Wildcards
- Registry Values

Please note that the following rules apply:

- Multiple items (within the **`<Path>`** element) must be separated by semicolons.
- Files specified without a directory path (e.g. file.exe) must be in the directory where *Malwarebytes Malware Remediation* is installed.
- When excluding a registry value (*regval*), it must be preceded by the registry key (*regkey*), delimited by the pipe “|” character.
- Extensions pertain to the entire file system that is subject to scanning.
- Threat vendor is specific to Malwarebytes definitions.
- Multiple exclusion list files must be separated by semicolons.

A sample Exclusion List is shown in Section 2.4 of this guide.

–noarchive

By default, the contents of archives (zip, rar, etc.) are scanned. Use this option to disable archive scanning. *Malwarebytes Malware Remediation* will stop scanning an archive if it finds a single infected file, and will quarantine the entire archive file.

–ignorepu

Instructs the scanner to ignore all Potentially Unwanted Programs (PUPs) and Potentially Unwanted Modifications (PUMs) that may be installed on the target computer.

–stopondetect [–malware:<*cnt*>] [–pu:<*cnt*>]

Instructs the scanner to terminate the scan when a certain number of either malware or potentially unwanted items are found. This allows a quick determination of whether the computer is infected without requiring a full scan to be performed. If <*cnt*> is not reached, the scan will run to completion.

Specify counts for either malware threats, PUP/PUM or both. The scanner will terminate when either criterion is met.

If **–stopondetect** is specified, at least one of the **–malware** or **–pu** options must also be specified.

–tag:<*tagname*>

This text string will be sent along with all usage data to the Malwarebytes billing system. If the string includes embedded spaces, it must be surrounded by double quotes (“). It will help you to associated billing events with your billing system. Typical usage would be for you to add the Job ID or Store ID or Employee ID or all of these, so that you can see these on your invoice.

–stdlog:<*filepath*>

Specifies the log location for normal output. If not specified, normal output will be written to **.\logs\MBMR-STDOUT.XML**. Use double-quotes (“) for paths that contain spaces. **Please note** that use of the default specification for <*filepath*> will result in loss of scan data for any scans previously executed, as a new scan overwrites existing results.

–noscanresults

Instructs the scanner to disregard creation of intermediate file **ScanResults.xml**. This parameter disables the ability to selectively remediate items during a scan, and is only valid for diagnostic scans.

–pfi:[<*secs*>]

Controls the frequency at which log file **ScanProgress.xml** is updated. This log file is stored in the folder specified by **–stdlog**. File creation frequency is in the range of 1-3600 seconds. This file is only created when **–stdout** is set to summary. If this option is not specified, no Scan Progress file will be created.

–ark

Enables Anti-rootkit scanner functionality to be used during the scan. Any rootkits found will be removed if **–remove** is specified.

–remove

Instructs the scanner to quarantine any malware, PUPs and PUMs found during the scan. This parameter is not allowed if **–stopondetect** is specified. If **–remove** and **–noreboot** are both specified in a scan command and the scan detected threats during its execution, a warning message will be displayed after the scan has completed to notify the user that a reboot is required to remove the threat(s) from the computer.

–confirmremove

This parameter is associated with a *selective remediation scan*. It instructs the scanner to pause after scanning, so that the user may inspect/edit intermediate file **ScanResults.xml** and determine if remediation should proceed, and which threats (if any) should be ignored. This parameter is used in conjunction with **–remove**. If this parameter is used and **–remove** is not specified, an error will result.

–noreboot

Some malware executes in a manner that requires a reboot to complete the removal process. If this occurs, the scanner will automatically reboot the system (after a 30-second warning dialog). If an immediate reboot is not desired, use this option. Please note that certain malware may not be fully removed if this option is used. If **–remove** and **–noreboot** are both specified in a scan command and the scan detected threats during its execution, a warning message will be displayed after the scan has completed to notify the user that a reboot is required to remove the threat(s) from the computer.

–removelastscan

This parameter is associated with a *diagnostic scan*. Instead of executing a new scan, it instructs the scanner to use intermediate file **ScanResults.xml** to remediate threats detected during the last scan that was executed.

–stdout:{off | detail | summary}

Controls the level of output to the console. Defaults to **summary** if not specified.

3.3.5 errorout

Usage:

mbmr errorout [[–console:{on | off }] [–delete] [–errlog:< *file*>] | [–reset]]

Purpose:

Specifies where error output will be directed to. Values set using this command will persist until they are cleared or modified. Issuing this command without arguments will display current settings.

Parameters:

–console:{off | on}

Specifies if error output is displayed on the console. Default value is ON.

–delete

Deletes the output file, if it exists. This command uses the default error log location, unless the error log location has been changed using the **–errlog** switch.

–errorlog:<*file*>

Specifies the log location for error output. This will overwrite any previously-specified location. If <*file*> contains any embedded spaces, please enclose <*file*> in double quotes ("). The default location is **.\logs\MBMR-ERROUT.TXT**.

–reset

Reverts settings associated with this command back to default values.

3.3.6 quarantine

Usage:

```
mbmr quarantine [–list]
                [–path:<path>]
                [–resetpath]
                [–restoreall]
                [–restore:<itemNos>]
                [–restorelist]
```

Purpose:

Set/reset location of quarantine, list quarantine contents, and restore files from quarantine. Use this command without any additional arguments to display the current quarantine location.

Parameters:

–path:<*path*>

Specifies the location to be used for quarantined content after this command has been executed. This replaces any previously-specified location. If <*path*> contains any embedded spaces, please enclose <*path*> in double quotes ("). **Please note** that quarantined content existing prior to execution of this command will not be moved to the new location.

–resetpath

Causes the quarantine file folder to revert to the default folder specification. Any files stored in quarantine prior to execution of this command will not be moved to the default folder. The default quarantine folder is:

- **Windows XP:** `C:\Documents and Settings\All Users\Application Data\Malwarebytes\Malwarebytes Anti-Malware\Quarantine`
- **Other OS versions:** `C:\ProgramData\Malwarebytes\Malwarebytes Anti-Malware\Quarantine`

–list

Shows the current quarantine location, lists contents of the quarantine to screen output, and generates file **RestoreList.xml** for use by the **–restorelist** option. In addition, an index number is associated with each quarantined file. The index number is used primarily in conjunction with the **–restore:<*itemNumbers*>** option, to simplify manual restore operations.

–restoreall

Restores all quarantined items to their original locations.

–restore:<*item1*> [,*item2*] … [,*itemn*]

Restores one or more items from the list of quarantined items shown on the screen (or in file **RestoreList.xml**). Items are specified by their index number. When multiple items are to be restored via a single execution of this command, their index numbers should be separated by commas without delimiting spaces. **Please note** that execution of this command will delete file **RestoreList.xml**, and that any pre-existing index numbers still shown on the screen are no longer valid. You must exercise the **–list** option again prior to any subsequent execution of this option.

–restorelist

Using file **RestoreList.xml** as a guide, this command restores specified files to their original location from Quarantine. For each file to be restored, modify the **`<Action>`** element associated with the file to be restored, changing the value to **restore**. Unless changed by the user (or by a third-party application), the default value of the **`<Action>`** element is **none**. A sample **RestoreList.xml** file is shown in Section 2.5 of this document.

3.3.7 settings

Usage:

```
mbmr settings [–color:off|on]
              [–proxy.clear]
              [–proxy.enabled:true | false]
              [–proxy.server:<host>]
              [–proxy.port:<port>]
              [–proxy.user:<user>]
              [–proxy.password:<password>]
              [–log.enabled:true|false]
              [–log.server:<host>]
              [–log.port:<port>]
              [–log.events:<eventname>:on|off]
              [–log.test]
              [–customdb.enabled:true | false]
              [–customdb.add:<customHashRule>]
              [–customdb.load:<openIOCFiles>]
              [–customdb.clear]
              [–customdb.list]
```

Purpose:

Used to define several program settings that will be universally used by *Malwarebytes Malware Remediation*.

Parameters:

–color:off|on

Specifies whether program output can utilize color, or if display will be limited to monochrome. Applications which attempt to capture standard output and error output of *Malwarebytes Malware Remediation* may encounter issues. Turning color off solves this problem.

–proxy.clear

Clears all settings associated with proxy servers.

–proxy.enabled:true | false

Enables or disables use of a proxy server for external Internet access. Internet access is required for program updates as well as threat signature updates. If this variable is set to true, the proxy **host** and **port** must also be specified.

PLEASE NOTE: If your network requires use of a proxy server for Internet access, proxy settings must be defined and enabled before a **register** or **update** command may be successfully executed.

–proxy.server:<*host*>

Hostname and/or IP address of proxy server providing external Internet access.

–proxy.port:<*port*>

Port number on proxy <*host*> which is used for external Internet access.

–proxy.user:<*user*>

Username for proxy usage, if authentication is required.

–proxy.password:<*password*>

Password for username <*user*>, if authentication is required to use proxy.

–log.enabled:true|false

Specifies whether program execution will be logged to a syslog server. All data utilizes a CEF (Common Event Format) standard. If this parameter is set to *true*, the syslog *host* IP/FQDN and *port* number must also be specified before event logging can take place.

–log.server:<*host*>

IP address or Fully-Qualified Domain Name (FQDN) of a syslog server which will receive event logs generated by *Malwarebytes Malware Remediation*. A valid *port* number must also be specified before logging can take place.

–log.port:<*port*>

Valid port number for the syslog server which will receive event logs generated by *Malwarebytes Malware Remediation*. A valid syslog *host* specification must also be specified before logging can take place.

–log.events:<*eventname*>:on|off

Specifies whether logging is enabled/disabled for each potential event which may be logged by *Malwarebytes Malware Remediation*. These events are discussed at length in Section 5 of this document, and are itemized here:

- ScanStartEvent
- DetectionEvent
- RestoreEvent
- CustomDbUpdateEvent
- ScanEndEvent
- RemovalEvent
- RestoreStartEvent
- DbUpdateEvent
- RemoveLastScanEvent
- RestoreEndEvent

–log.test

Attempts to make contact with the syslog server using the *host* and *port* specifications which have been provided. Results of the connection attempt are shown on-screen. This command does not generate an event log entry. No additional parameters are required.

–customdb.enabled:true | false

Specifies whether custom database rules utilizing OpenIOC are enabled (true) or disabled (false). Default value is false.

–customdb.add:<customHashRule>

Allows a single MD5 hash to be added to the Rules database without requiring use of the OpenIOC XML file as an input medium. The MD5 hash specified here is incrementally added to existing rules.

–customdb.load:<openIOCFiles>

Loads one or more OpenIOC files into the Rules database. When multiple files are specified, they must be separated by semicolons. If embedded spaces are present in the file and/or path specification, the full path and file must be enclosed by double quotes. OpenIOC files must be saved with the extension XML to be used with this command. Files created by Mandiant's IOC Editor may be renamed from extension .IOC to .XML directly before use with this command. **Please note** that this command will overwrite existing rules whose **`<IndicatorItem>`** elements match.

–customdb.clear

Deletes all existing custom rules.

–customdb.list

Lists all existing custom rules.

4.0 Scan Log

Finalized results of scans executed by *Malwarebytes Malware Remediation* are saved in file MBMR-STDOUT.XML, a scan log which may be imported by several document formats as well as for integration by internet-based applications. The root XML element in the log file is **<mbam-log>**. All subsequent data is grouped by section. Those sections – and data related to those sections – are described here.

4.1 <header> Section

This section provides high-level information about the scan that was performed.

4.1.1 <date>

Local time at which the scan began, as well as the time zone in which the computer is located. Time zones are referenced to Greenwich Mean Time (GMT).

FORMAT: *yyyy/mm/dd hh:mm:ss zone*, where:

yyyy	=	Year
mm	=	Month
dd	=	Date
hh	=	Hours
mm	=	Minutes
ss	=	Seconds
zone	=	Difference (in hours) between local time and GMT

4.1.2 <logfile>

XML log filename, which is hard-coded as **MBMR-STDOUT.XML**.

4.1.3 <isadmin>

Flag denoting whether the logged-in user had administrator access. Valid values are **yes** or **no**.

4.2 <engine> Section

This section provides information about *Malwarebytes Malware Remediation*, as well as version numbers of the program as well as associated databases which are used during the scan.

4.2.1 <version>

Version of *Malwarebytes Malware Remediation* being used for the scan.

4.2.2 <malware-database>

Malware threat signature database version being used for the scan.

FORMAT: v*yyyy.mm.dd.nn*, where:

yyyy	=	year
mm	=	month
dd	=	date
nn	=	daily version number

4.2.3 <rootkit-database>

Anti-Rootkit database version being used for the scan. The Anti-Rootkit component of *Malwarebytes Malware Remediation* uses its own signature database.

FORMAT: v*yyyy.mm.dd.nn*, where:

yyyy	= year
mm	= month
dd	= date
nn	= daily version number

4.2.4 <license>

Specifies the license type in use by the product. Valid value for the-remediation product is **premium**.

4.2.5 <file-protection>

This feature is not used in the remediation product, and can be ignored.

4.2.6 <web-protection>

This feature is not used in the remediation product, and can be ignored.

4.2.7 <self-protection>

This feature is not used in the remediation product, and can be ignored.

4.3 <system> Section

This section provides basic information about the system being scanned.

4.3.1 <hostname>

The name assigned to the computer being scanned.

4.3.2 <ip>

The IP address associated with the computer being scanned. If dynamic addressing (DHCP) is used, this is the IP address as of the time that the computer was scanned. It may not represent the same computer at a later date.

4.3.3 <osversion>

Operating System version in use on the computer being scanned. This field will also include Service Packs in use (if applicable).

4.3.4 <arch>

CPU architecture of the system. Valid values are **x86** (32-bit) or **x64** (64-bit).

4.3.5 <username>

Windows user name associated with execution of the scan.

4.3.6 <filesys>

The file system of the computer's primary disk drive (meaning the drive on which the operating system is loaded). Valid values are **NTFS**, **FAT** or **FAT32**.

4.4 <summary> Section

This section provides summary information on the scan that was performed. Referring to the list below, elements beginning with **<processes>** are directly related to elements in the **<options>** section which enable or disable corresponding functionality. The controlling elements will be referenced in descriptions here.

4.4.1 <type>

Type of scan which was executed. Valid values are **custom**, **threat**, or **hyper**.

4.4.2 <result>

Final result of scan. Valid values are **cancelled**, **completed** or **failed**. If a scan was executed with **–stopondetect** and terminated as a result of this specification being set, **<result>** will be set to **completed**.

4.4.3 <objects>

Number of objects scanned

4.4.4 <time>

Elapsed time of scan from start to finish (in seconds).

4.4.5 <processes>

Number of threats detected in running processes.

4.4.6 <modules>

Number of threats detected in memory modules.

4.4.7 <keys>

Number of threats detected in registry keys.

4.4.8 <values>

Number of threats detected in registry values.

4.4.9 <datas>

Number of threats detected in registry data.

4.4.10 <folders>

Number of threats detected in folders. Controlled by **<filesystem>**.

4.4.11 <files>

Number of threats detected in files. Controlled by **<filesystem>**.

4.4.12 <sectors>

Number of threats detected in disk sectors. Controlled by **<rootkits>**.

4.5 <options> Section

This section provides information on various categories that were employed during the scan. Many of these categories are directly responsible for results shown in the **<summary>** section. Settings from this section are referenced in **<summary>** items that are directly related.

4.5.1 <memory>

Denotes whether scanning of running memory processes is active. Set to **enabled** when **–threat** or **–hyper** scan types are specified, or **disabled** when **–path** scan type is specified.

4.5.2 <startup>

Denotes whether scanning of startup-related processes and modules is active. Set to **enabled** when **–threat** or **–hyper** scan types are specified, or **disabled** when **–path** scan type is specified.

4.5.3 <filesystem>

Denotes whether scanning of the file system is active. Set to **enabled** when **–threat** or **–path** scan types are specified, or **disabled** when **–hyper** scan type is specified.

4.5.4 <archives>

Denotes whether scanning of archive files is active. This includes ZIP, RAR, ARJ, CAB and 7Z files. Valid values are **enabled** (default value) or **disabled** (when **–noarchive** is specified as part of **scan** command). When enabled, scanning of archiving is limited to three levels deep. When disabled, the archive is scanned as a single file. Encrypted (password-protected) archives cannot be effectively scanned.

4.5.5 <rootkits>

Denotes whether anti-rootkit scanning is active. Valid values are **enabled** or **disabled** (default value). Value is determined based on setting of scan parameter **–ark**.

4.5.6 <deeprootkit>

Denotes whether deep rootkit scanning is active. Set to **enabled** when *Malwarebytes Malware Remediation* scan engine has determined this method of scanning is required. The default value is **disabled**.

4.5.7 <heuristics>

Denotes whether heuristics are employed during scanning. Heuristics enables enhanced detection of threats which may avoid detection by signatures only. Set to **enabled** when **–threat** or **–hyper** scan types are specified, or **disabled** when **–path** scan type is specified.

4.5.8 <pup>

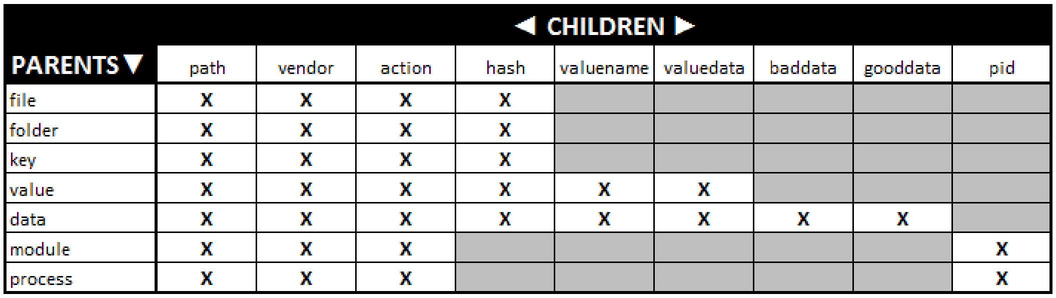
Denotes whether scanning of Potentially Unwanted Programs (PUPs) is active. Set to **enabled** by default in the remediation product. Reverts to **disabled** if **–ignorepu** parameter is specified in scan settings.

4.5.9 <pum>

Denotes whether scanning of Potentially Unwanted Modifications (PUMs) is active. Set to **enabled** by default in the remediation product. Reverts to **disabled** if **–ignorepu** parameter is specified in scan settings.

4.6 <items> Section

There are seven different major categories of threats which may be detected during a scan. Each major category has up to eight fields which describe the threat in more detail. The chart below shows the relationships between the major categories (parents) and the detail fields (children).



	◀ CHILDREN ▶								
PARENTS ▼	path	vendor	action	hash	valuename	valuedata	baddata	gooddata	pid
file	X	X	X	X					
folder	X	X	X	X					
key	X	X	X	X					
value	X	X	X	X	X	X			
data	X	X	X	X	X	X	X	X	
module	X	X	X						X
process	X	X	X						X

Descriptions of all detail fields are as follows. This section provides information on threats which were detected during the scan. In this section, **<file>** is the parent entry for each file system-based threat and **<key>** is the parent entry for each registry-based threat. Several subentries exist for each **<file>** or **<key>**, describing the threat. These subentries are described below. Please note that this section is shown with the parent entry if no threats were detected.

4.6.1 <path>

Where the threat was detected. If the threat was found on the Windows filesystem, this contains the full drive/directory/filename. If the threat was found in the Windows registry, this entry contains the registry key/value name/value data corresponding to the threat.

4.6.2 <vendor>

Name of the threat (or threat family), as categorized by the Malwarebytes Research Team. Please note that the same threat may be identified with different names by the various antivirus/anti-malware products.

4.6.3 <action>

This describes the action taken to remediate the detected threat.

4.6.4 <hash>

A 32-byte identifier which Malwarebytes uses to identify a specific threat. While threat names may be extremely similar (and easy to confuse), the MD5 value supplied here is highly unique.

4.6.5 <valuename>

The name of a registry entry that was detected as a threat and removed. This is present only when a threat of this type has been detected, and will be accompanied by <**valuedata**> when present.

4.6.6 <valuedata>

The value of a registry entry that was determined to be a threat and was removed as a result. This is present only when a registry entry (represented by <**valuename**>) has been removed.

4.6.7 <baddata>

Contains registry data associated with a detected threat. This data will be replaced or deleted according to specifications contained in Malwarebytes threat signatures. This element is present only when required, and will be accompanied by **<gooddata>** when it is present.

4.6.8 <gooddata>

Contains registry data used to replace **<baddata>**. This element may contain data or a null value, and is present only when required. When present, it will always be accompanied by **<baddata>**.

4.6.9 <pid>

Process identifier associated with a threat. This field applies only to modules and processes detected during memory scan phases.

4.7 Sample Log File

The following is a sample log resulting from a scan. It is provided solely to illustrate how results appear in a real-world scenario. **Please note** that indentation has been added to this example for readability purposes.

```
<?xml version="1.0" encoding="UTF-16" ?>
<mbam-log>
    <header>
        <date>2015/01/08 16:28:18 -0800</date>
        <logfile>MP-STDOUT.XML</logfile>
        <isadmin>no</isadmin>
    </header>
    <engine>
        <version>2.00.0.1030</version>
        <malware-database>v2015.01.07.14</malware-database>
        <rootkit-database>v2015.01.07.01</rootkit-database>
        <license>premium</license>
        <file-protection>disabled</file-protection>
        <web-protection>disabled</web-protection>
        <self-protection>disabled</self-protection>
    </engine>
    <system>
        <osversion>Windows 8.1</osversion>
        <arch>x64</arch>
        <username>administrator</username>
        <filesys>NTFS</filesys>
    </system>
    <summary>
        <type>custom</type>
        <result>completed</result>
        <objects>231856</objects>
        <time>30</time>
        <processes>0</processes>
        <modules>0</modules>
        <keys>1</keys>
        <values>0</values>
        <datas>0</datas>
        <folders>0</folders>
        <files>2</files>
        <sectors>0</sectors>
    </summary>
    <options>
        <memory>enabled</memory>
        <startup>enabled</startup>
        <filesystem>enabled</filesystem>
        <archives>enabled</archives>
        <rootkits>disabled</rootkits>
        <deeprootkit>disabled</deeprootkit>
        <heuristics>enabled</heuristics>
        <pup>enabled</pup>
        <pum>enabled</pum>
    </options>
    <items>
        <key>
            <path>HKLM\SOFTWARE\Google\Chrome\Extensions\blmchfpimpbbdmgpcieclabeafkljbhm</path>
            <vendor>PUP.Optional.Groovorio.A</vendor>
            <action></action>
            <hash>8dfc9b5b4841ff3703a7196e3ec5ab55</hash>
        </key>
        <file>
            <path>c:\temp2\test-trojan.exe</path>
            <vendor>MBAM.Test.Trojan</vendor>
            <action></action>
            <hash>e2b839bb6a1fda5c4bdadd73ac56cb35</hash>
        </file>
        <file>
            <path>c:\temp2\Test_PUP.exe</path>
            <vendor>PUP.Optional.Dotpitch</vendor>
            <action></action>
            <hash>d1c952a2cbbe3006b11f51aeba4ad32d</hash>
        </file>
    </items>
</mbam-log>
```

4.8 Sample Scan Progress File

The following is a sample Scan Progress log created during a scan. When requested, these are generated at regular intervals for integration with third-party applications. It is provided here solely to illustrate how results appear in a real-world scenario. **Please note** that indentation has been added to this example for readability purposes.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ScanProgress>
    <CurrentScanPhase>8</CurrentScanPhase>
    <CurrentScanPhaseName>Filesystem Objects</CurrentScanPhaseName>
    <LastScanPhase>10</LastScanPhase>
    <ItemsScanned>7595</ItemsScanned>
    <PUCount>0</PUCount>
    <VirusCount>1</VirusCount>
    <ScanCompletion>90</ScanCompletion>
    <CurrentlyScannedFile>c:\Windows\System32\drivers\acpi.sys</CurrentlyScannedFile>
    <CurrentVirus>MBAM.Test.Trojan</CurrentVirus>
    <CurrentPU></CurrentPU>
    <ElapsedTime>00:07:14:707</ElapsedTime>
</ScanProgress>
```

5.0 Event Logging to syslog

Malwarebytes Malware Remediation integrates very easily into a corporate network, providing highly effective results in the detection and remediation of threats on endpoints. That integration has been extended further through the addition of event logging using industry-standard methods. Based on user requests, we have implemented logging using CEF (Common Event Format), and more specifically, output is tailored to the ArcSight Security Intelligence platform and others which support the CEF format.

This section of the guide is devoted to detailed descriptions of how we have implemented event logging, so that you may easily understand log results and customize reporting in your specific environment.

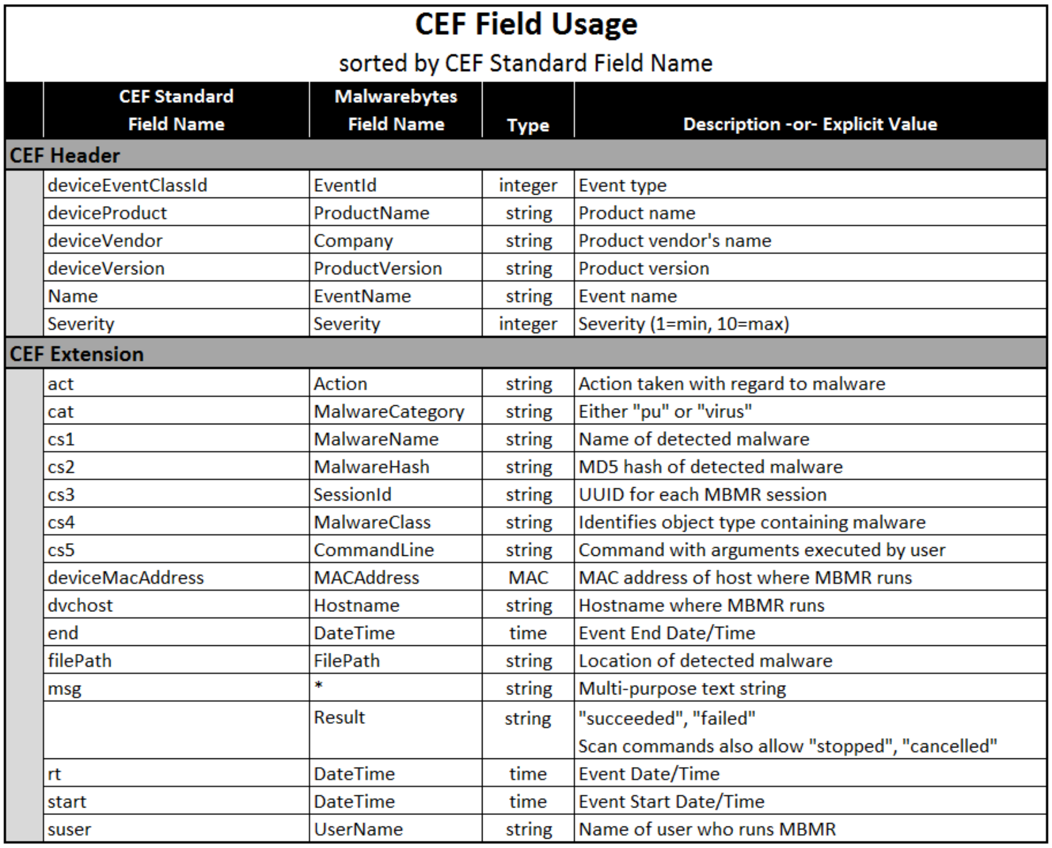
5.1 Construction of a Log Entry

All event logs use a standardized format, which consist of a syslog prefix, a header and an extension. They are described as follows:

- **syslog prefix:** A mandatory entry that is applied for compliance with syslog standards. It includes:
  - **Event date**, including month, day and year, in the format (e.g. **Jul 15 2015**)
  - **Event time**, using 24-hour clock, in the format **hh:mm:ss** (e.g. **12:25:40**)
  - **Hostname** which logs pertain to (e.g. **SFO-VM1234.internal.contoso.com**)
- **Header:** Mandatory fields which identify the product generating log entries. Vendors may use non-standard field names for these fields, but their usage must correspond to fields and their order within the log record.
  - **CEF Version**, in the format "CEF:<*version*>". <*version*> is a single-digit, and is used for compliance with the CEF standard as well as to specify how remaining data should be interpreted.
  - **Device Vendor** identifies the vendor of the product which is generating log entries. As it pertains here, this will be "Malwarebytes"
  - **Device Product** identifies the product which is generating log entries. As it pertains here, this will be "Malwarebytes Malware Remediation"
  - **Device Version** identifies the product version. Malwarebytes Malware Remediation is identified not only by the version of the executable, but also by each major components used in conjunction with the program. All components which follow the executable program version are bounded by square brackets. Those components are:
    - Engine (MBAM Core DLL)
    - Rules Database
    - Actions Database
    - Anti-Rootkit Database (Swiss Army Knife)
  - **Signature ID** is a unique numeric identifier which Malwarebytes has assigned to each event type. A full list of all Signature IDs can be found later in this section.
  - **Name** is a simple text description for each event that corresponds to a specific Signature ID
  - **Severity** is the relative importance of any event, with 1 being the least important and 10 representing a critical event.
- **Extension:** This is a series of fields which are not mandatory by CEF standards. These fields represent values that each vendor select for inclusion in their event logs. Because these fields are not mandatory, vendors will commonly use non-standard field names, and may also include labels for the non-standard fields. If a customer elects to use these labels, this would improve readability of log information, though the same label used by multiple vendors may also create confusion for the user.

5.2 Mapping Malwarebytes Fields to CEF Format

As mentioned previously, log entries are comprised of three separate sections. The *syslog prefix* and *Header* are mandatory, and must conform to rigid standards. The *extension* provides flexibility that vendors require to capture important details related to their products, while still conforming to CEF standards. Malwarebytes is no exception.



CEF Field Usage

sorted by CEF Standard Field Name

	CEF Standard Field Name	Malwarebytes Field Name	Type	Description -or- Explicit Value
CEF Header				
	deviceEventClassId	EventId	integer	Event type
	deviceProduct	ProductName	string	Product name
	deviceVendor	Company	string	Product vendor's name
	deviceVersion	ProductVersion	string	Product version
	Name	EventName	string	Event name
	Severity	Severity	integer	Severity (1=min, 10=max)
CEF Extension				
	act	Action	string	Action taken with regard to malware
	cat	MalwareCategory	string	Either "pu" or "virus"
	cs1	MalwareName	string	Name of detected malware
	cs2	MalwareHash	string	MD5 hash of detected malware
	cs3	SessionId	string	UUID for each MBMR session
	cs4	MalwareClass	string	Identifies object type containing malware
	cs5	CommandLine	string	Command with arguments executed by user
	deviceMacAddress	MACAddress	MAC	MAC address of host where MBMR runs
	dvchost	Hostname	string	Hostname where MBMR runs
	end	DateTime	time	Event End Date/Time
	filePath	FilePath	string	Location of detected malware
	msg	*	string	Multi-purpose text string
		Result	string	"succeeded", "failed" Scan commands also allow "stopped", "cancelled"
	rt	DateTime	time	Event Date/Time
	start	DateTime	time	Event Start Date/Time
	suser	UserName	string	Name of user who runs MBMR

As mentioned previously, *msg* is a multi-purpose field. The CEF format provides six fields which may contain custom data, that which the vendor has determined to be important with relation to their product, but does not conform to a standard CEF field. Malwarebytes utilizes five of these six fields, and also utilizes *msg* to provide more robust log content. Its usage in each log message will be detailed in the next section.

5.3 Malwarebytes Log Events

Malwarebytes Malware Remediation currently generates log entries for nine different event categories. This section of the guide describes each of those categories in detail. The following table lists fields which are common to all log entries created by *Malwarebytes Malware Remediation*.

	CEF Standard Field Name	Malwarebytes Field Name	Description -or- Explicit Value
CEF Header			
	deviceVendor	Company	"Malwarebytes"
	deviceProduct	ProductName	"Malwarebytes Malware Remediation"
	deviceVersion	ProductVersion	Version of program, engine and databases
CEF Extension			
	cs3	SessionId	UUID for each MBMR session
	dvchost	Hostname	Hostname where MBMR runs
	deviceMacAddress	MACAddress	MAC address of host where MBMR runs
	cs5	CommandLine	Command with arguments executed by user
	outcome	Result	"succeeded", "failed" Scan commands also allow "stopped", "cancelled"
	suser	UserName	Name of user who runs MBMR

In addition to these common fields, the following events utilize several fields specific to the event being logged. The remainder of this section is devoted to descriptions of each of these events.

5.3.1 1000 – ScanStartEvent

1000	**ScanStartEvent**	**Generated whan a scan command is initiated**		
	EventName	**Severity**	**Fields**	**Mapping**
	ScanStarted	1	ScanType (string)	msg **Format:** "msg=ScanType:%s" **Value(s):** threat hyper path full
			Time	start

5.3.2 1001 – DetectionEvent

1001	**DetectionEvent**	**Generated when malware is detected during a scan**		
	EventName	**Severity**	**Fields**	**Mapping**
	Malware Detected	9 (PUM) 10 (virus)	Action	act **Value(s):** none
			MalwareCategory	cat **Value(s):** pu (PUM) virus (virus malware)
			MalwareName	cs1
			MalwareHash	cs2
			MalwareClass	cs4 **Value(s):** 0x01 (File) 0x02 (Folder) 0x04 (RegKey) 0x08 (RegVal) 0x10 (Process) 0x20 (Module) 0x40 (Ads) 0x80 (Physical Sector)
			Time	real-time

5.3.3 1002 – RemovalEvent

1002	RemovalEvent	Generated when malware is removed during "scan -remove" or "scan -removelastscan" commands		
	EventName	**Severity**	**Fields**	**Mapping**
	Malware Detected	9 (PUM) 10 (virus)	Action	act **Value(s):** quarantined
			MalwareCategory	cat **Value(s):** pu (PUM) virus (virus malware)
			MalwareName	cs1
			MalwareHash	cs2
			MalwareClass	cs4 **Value(s):** 0x01 (File) 0x02 (Folder) 0x04 (RegKey) 0x08 (RegVal) 0x10 (Process) 0x20 (Module) 0x40 (Ads) 0x80 (Physical Sector)
			Time	real-time

5.3.4 1003 – ScanEndEvent

1003	ScanEndEvent	Generated when a scan command ends		
	EventName	**Severity**	**Fields**	**Mapping**
	ScanEnded	1	DetectionCount(int)	msg
			RemovalCount(int)	msg **Format:** "msg=DetectionCount:%d Removal Count:%d" **Note:** Fields are combined as part of the *msg* field
			Time	end

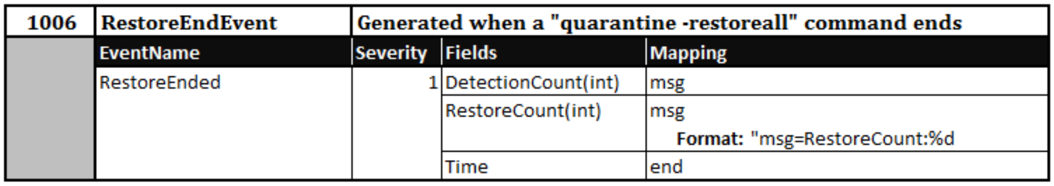
5.3.5 1004 – RestoreStartEvent

1004	RestoreStartEvent	Generated when a "quarantine -restoreall" command is initiated		
	EventName	**Severity**	**Fields**	**Mapping**
	Restore Started	1	Time	Start

5.3.6 1005 – RestoreEvent

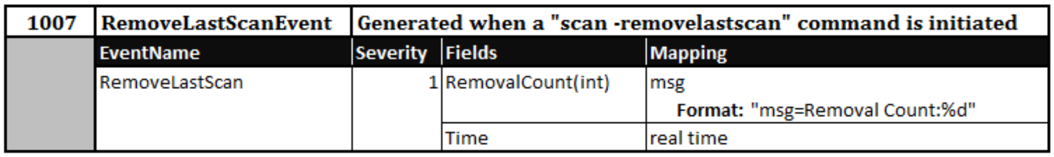
1005	RestoreEvent	Generated when an item is restored during a "quarantine -restoreall" command		
	EventName	**Severity**	**Fields**	**Mapping**
	Item Restored	5	Action	act **Value(s):** restored
			FilePath	filePath **Value(s):** Name and complete path of a file being restored
			Time	real-time

5.3.7 1006 – RestoreEndEvent



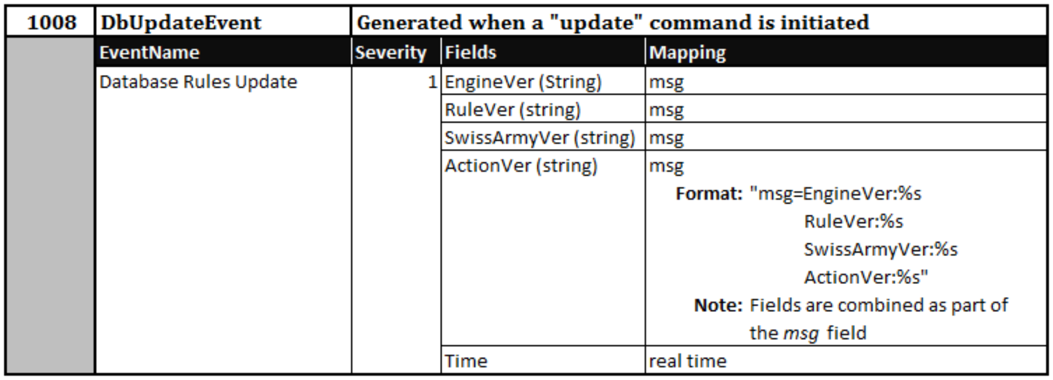
1006	RestoreEndEvent	Generated when a "quarantine -restoreall" command ends		
	EventName	Severity	Fields	Mapping
	RestoreEnded	1	DetectionCount(int)	msg
			RestoreCount(int)	msg **Format:** "msg=RestoreCount:%d
			Time	end

5.3.8 1007 – RemoveLastScanEvent



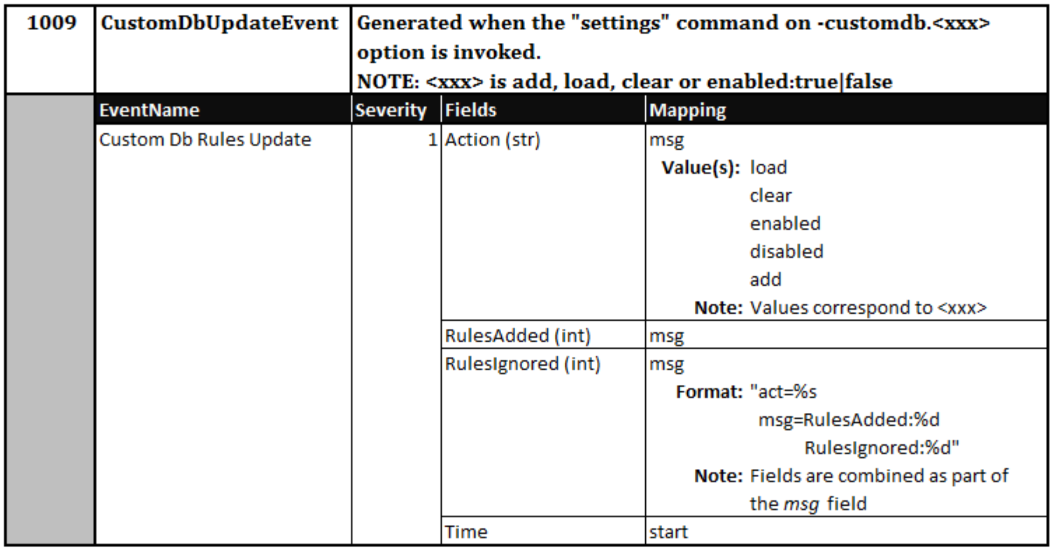
1007	RemoveLastScanEvent	Generated when a "scan -removelastscan" command is initiated		
	EventName	Severity	Fields	Mapping
	RemoveLastScan	1	RemovalCount(int)	msg **Format:** "msg=Removal Count:%d"
			Time	real time

5.3.9 1008 – DbUpdateEvent



1008	DbUpdateEvent	Generated when a "update" command is initiated		
	EventName	Severity	Fields	Mapping
	Database Rules Update	1	EngineVer (String)	msg
			RuleVer (string)	msg
			SwissArmyVer (string)	msg
			ActionVer (string)	msg **Format:** "msg=EngineVer:%s RuleVer:%s SwissArmyVer:%s ActionVer:%s" **Note:** Fields are combined as part of the *msg* field
			Time	real time

5.3.10 1009 – CustomDbUpdateEvent



1009	CustomDbUpdateEvent	Generated when the "settings" command on -customdb.<xxx> option is invoked. NOTE: <xxx> is add, load, clear or enabled:true|false		
	EventName	Severity	Fields	Mapping
	Custom Db Rules Update	1	Action (str)	msg **Value(s):** load clear enabled disabled add **Note:** Values correspond to <xxx>
			RulesAdded (int)	msg
			RulesIgnored (int)	msg **Format:** "act=%s msg=RulesAdded:%d RulesIgnored:%d" **Note:** Fields are combined as part of the *msg* field
			Time	start

5.4 Further reading

Malwarebytes recommends that you obtain a copy of the following document from ArcSight. It is a detailed guide pertaining to the CEF logging format, as well as their recommendations targeted to users and developers.

> ***Implementing ArcSight CEF*** *(Revision 20, dated 05 June 2013)*
> https://protect724.hp.com/docs/DOC-1072

6.0 Customizing the Rules Database with OpenIOC

Before discussing how Malwarebytes uses OpenIOC to supplement our robust threat database, it is best to provide a very brief introduction to OpenIOC technology. The following text is quoted directly from the OpenIOC web site (http://www.openioc.org):

> *In the current threat environment, rapid communication of pertinent threat information is the key to quickly detecting, responding and containing targeted attacks. OpenIOC is designed to fill a void that currently exists for organizations that want to share threat information both internally and externally in a machine-digestible format. OpenIOC is an extensible XML schema that enables you to describe the technical characteristics that identify a known threat, an attacker's methodology, or other evidence of compromise.*
>
> *OpenIOC was originally designed to enable MANDIANT's products to codify intelligence in order to rapidly search for potential security breaches. Now, in response to requests from across the user community, MANDIANT has standardized and open sourced the OpenIOC schema and is releasing tools and utilities to allow communication of threat information at machine speed.*

6.1 Implementing IOC Definitions

Many companies are now using this technology to identify and act upon threats on their computer networks. In conjunction with the threat database which powers *Malwarebytes Malware Remediation* (and our other anti-malware programs), we are now providing the ability to extend our database further with OpenIOC format. Please note that Mandiant refers to output from their IOC editor as Indicators of Compromise (IOC's), while Malwarebytes uses the term rules. In this section of the guide, either term may be used, though every effort has been made to not create ambiguous meanings.

6.1.1 Mandiant IOC Editor

The Mandiant IOC Editor is the most commonly used editor for creating custom Indicators of Compromise (IOC's). It is made available as a free open-source tool, and is downloadable at:

https://www.fireeye.com/services/freeware/ioc-editor.html

This tool allows the user to select a specific type of IOC, guides the user's entry of appropriate data to satisfy criteria associated with the IOC, and constructs the logical structure which the IOC requires. It does not allow browsing of the file system or registry, and does not offer auto-complete functionality as part of data entry. Please note that IOC's created by the Mandiant IOC Editor use the .IOC file extension, and the extension must be renamed to .XML for use by *Malwarebytes Malware Remediation*. Other applications can create the IOC file directly in XML format.

6.1.2 Restrictions Which Apply to Custom Rules

There are a few restrictions which apply to the creation of custom rules. These are as follows:

- Mandiant's IOC Editor uses UUID for rule names which it creates.
- Malwarebytes Malware Remediation prepends each custom rule with the text "**CustomRule.**"
- Rule names may not exceed 128 characters in length.
- Valid characters in a rule name include letters A-Z, numbers 0-9, and special characters period (.). hyphen (-), underscore (_), and the pair of curly brackets ({ }). A period may not be used as the first character of a rule name.

6.1.3 Top-Level OpenIOC XML file

The XML file used in conjunction with *Malwarebytes Malware Remediation* must conform to the construction shown in the example below. The second line of this example wraps due to its length. It is likely that you will create this file using the Mandiant IOC Editor (or another open-source editor) which constructs the file for you.

```
<?xml version="1.0" encoding="us-ascii"?>
<ioc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="739767f0-27ba-49ca-9510-b8c56343dc48"
last-modified="2015-09-14T20:55:27" xmlns="http://schemas.mandiant.com/2010/ioc">
   <short_description>*New Unsaved Indicator*</short_description>
   <authored_date>2015-09-14T20:39:44</authored_date>
   <links />
   <definition>
      <Indicator operator="OR" id="966d7c5c-b0dc-41ff-a6e8-ba4714bf6937">

         <!– custom rule entries are defined here -->

      </Indicator>
   </definition>
</ioc>
```

Please note that all UUID entries shown in files which you create will be different than those shown here. They are unique to the system where file creation takes place.

6.2 Creating Custom Rules

While Mandiant's IOC Editor provides the capability to create a wide range of rules for identifying malware, five specific rules apply to *Malwarebytes Malware Remediation*. For each of these rule types, an example is provided to show the construction of the rule, content examples, applicable criteria, and when the rule applies (relative to the scope of *Malwarebytes Malware Remediation*).

6.2.1 Custom Hash Rule

The Customer Hash rule provides for identification of a threat using its 32-bit MD5 hash value.

Syntax:

```
<IndicatorItem id="6de269b9-69aa-4701-aaa8-2f40a7a14a6a" condition="is">
    <Context document="FileItem" search="FileItem/Md5sum" type="mir" />
    <Content type="md5">b5891462c9ca5bddfe63d3bae3c14e0b</Content>
</IndicatorItem>
```

Criteria:

Condition value = "is"

When Used:

Full scan, Path scan

6.2.2 Custom File Rule

The Custom File rule identifies a file (by name).

Syntax:

```
<IndicatorItem id="cfdcc16b-d1ac-4bcd-96fc-f7e72957b6f1" condition="is">
    <Context type="mir" search="FileItem/FileName" document="FileItem"/>
    <Content type="string">trojans.exe</Content>
</IndicatorItem>

<IndicatorItem id="cfdcc16b-d1ac-4bcd-96fc-f7e72957b6f1" condition="contains">
    <Context type="mir" search="FileItem/FileName" document="FileItem"/>
    <Content type="string">trojan</Content>
</IndicatorItem>
```

Criteria:

Condition value = "is" or "contains". When "is" condition is used, the filename does not include the directory path. The filename is case-insensitive.

When Used:

Full scan, Path scan

6.2.3 Custom Folder Rule

The Custom Folder rule identifies a folder/path.

Syntax:

```
<IndicatorItem id="5372a89e-a0b3-4a39-91ad-57dfca92a105" condition="is">
   <Context type="mir" search="FileItem/FilePath" document="FileItem"/>
   <Content type="string">c:\virus\a</Content>
</IndicatorItem>

<IndicatorItem id="5372a89e-a0b3-4a39-91ad-57dfca92a105" condition="contains">
   <Context type="mir" search="FileItem/FilePath" document="FileItem"/>
   <Content type="string">foo\a</Content>
</IndicatorItem>
```

Criteria:

Condition value = "is" or "contains". When "is" condition is used, the folder path is the absolute path of the folder. The folder path is case-insensitive.

When Used:

Full scan, Path scan

6.2.4 Custom Registry Key Rule

The Custom Registry Key rule identifies a specific registry key.

Syntax:

```
<IndicatorItem id="4bbc6c65-062c-47e5-8a93-1d589a7584da" condition="is">
   <Context type="mir" search="RegistryItem/KeyPath" document="RegistryItem"/>
   <Content type="string">HKLM\Software\Test</Content>
</IndicatorItem>
```

Criteria:

Condition value = "is". The registry key is case-insensitive.

When Used:

Full scan, Hyper scan, Threat scan

6.2.5 Custom Registry Value Rule

The Custom Registry Value rule identifies a specific registry value. It requires three (3) child indicator items to properly identify the registry value.

Syntax:

```
<Indicator id="99146874-2b11-48aa-aaa5-256cc80d0bc8" operator="AND">
   <IndicatorItem id="8971b887-580c-4619-973f-fbbbfe6aa7b6" condition="is">
      <Context type="mir" search="RegistryItem/KeyPath" document="RegistryItem"/>
      <Content type="string">HKLM\Software\Test2</Content>
   </IndicatorItem>
   <IndicatorItem id="a71e44d5-d022-4f84-aaee-81fa7141e580" condition="is">
      <Context type="mir" search="RegistryItem/ValueName" document="RegistryItem"/>
      <Content type="string">Foo</Content>
   </IndicatorItem>
   <IndicatorItem id="60eaa413-a6f9-4b66-8df5-7a4fa5a1ae4f" condition="is">
      <Context type="mir" search="RegistryItem/Value" document="RegistryItem"/>
      <Content type="string">12345</Content>
   </IndicatorItem>
</Indicator>
```

Criteria:

All indicators should use "is" condition. All indicator values are case-insensitive.

When Used:

Full scan, Hyper scan, Threat scan

6.3 Further Reading

- Mandiant ICO Editor User Guide (version 2.2.0.0)
  https://www.fireeye.com/content/dam/fireeye-www/services/freeware/ug-ioc-editor.pdf
- OpenIOC web site
  http://www.openioc.org/

7.0 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. The *Malwarebytes Third Party License Supplement* is a downloadable reference which specifies each of these projects, and where they are used. It can be downloaded from:

https://www.malwarebytes.org/pdf/guides/ThirdPartyLicenseSupplement.pdf