Manual for Bouncer

Version 3.0.0 (June 2018)

excubits.com
Einfach. Sicher. Excubits.

Imprint

Copyright:	Excubits
Web:	https://excubits.com
Contact:	info@excubits.com
Version:	3.0.0
Status:	Published

Table of Contents

What is Bouncer?

Excubits Bouncer is a security software for Windows computers. Bouncer blocks unknown executables. For example, computer viruses, worms, cryptolockers etc. It does not matter whether the malicious software comes from network drives, USB sticks, external hard drives, CD/DVD ROMs or e-mail attachments. Even newly discovered security holes, for which there are no updates available yet, Bouncer can significantly mitigate.

Bouncer works according to the exclusion principle: All known executables run as desired. All unknown and potentially dangerous executables are blocked by bouncer. The whitelist of known programs can be extended at any time.

In depth protection in the kernel

Bouncer runs as a so-called driver in the core of the operating system. This allows our software to block unknown or malicious program files much earlier than conventional security tools. Once configured, Bouncer can protect a system from new, unknown malware without further signature updates.

Additional options for even more protection

Parent-based rules allow to specify which applications a program is allowed to start and which not. This feature is very useful, because cyber criminals can secretly launch malicious programs through a web browser or Office applications. Users can now effectively prevent this attack vector with bouncer.

Users can also use Command line checking to specify which command line parameters can be used to start a program and which cannot. This way you can additionally secure interpreters such as Powershell, JScript, Java, Python and others.

Advice

In the following chapters we will describe how to install Bouncer, how it works, and how to configure the system suitably. Please take the time to fully read and understand this manual in order to operate Bouncer correctly. It is very important to understand the functioning of Bouncer to configure the driver properly and to achieve best results.

Read these instructions carefully and follow the recommendations and descriptions exactly to avoid crashes or a blocking system.

System requirements

Bouncer runs and protects the following versions of Microsoft Windows:

Version	32-bit/64-bit
Windows XP	On request, only for corporate customers
Windows Vista	On request, only for corporate customers
Windows 7 (incl. Windows Server)	yes / yes
Windows 8 (incl. 8.1, Server, Core)	yes / yes
Windows 10 (incl. Server, Core)	yes / yes

To run Bouncer, you need at least 8 MB of free hard disk space. Note that Bouncer creates a log file in which the drivers logs security relevant events. Please ensure that there is enough free space available for creating and writing into the log file. From time to time you shall archive or delete older entries in the log file to reduce the size.

Setup

To install and configure Bouncer, you shall have administration access to the computer. After Bouncer was successfully installed and is running, you need no administrative privileges (access) to use a Windows PC protected by Bouncer. Once installed, Bouncer runs transparently in the background and keeps up the protection, no matter who is logged on.

Automatic Installation

The installer can automatically configure and start the driver and tools. Double-click the installation application. Simply follow the dialog boxes of the installation program.

The installation program installs the driver and creates a basic configuration. The following chapter contains a detailed description of the configuration file and how to customize it. The installation program has already taken a few steps off your hands. You will find the `bouncer.ini` file in your system directory (usually `C:\Windows\`), the log file was created and the driver was registered.

Under `C:\Programs\` or `C:\Programs (x86)\` you will find the Excubits folder. Click your way through to the Bouncer folder. There you will now find this manual, control scripts and applications (`./tools`).

Starting the installation manually

You can perform the automatic installation manually by first unpacking the installation application, for example using Win-RAR or 7zip into a folder named Excubits. Now we recommend to copy this folder into the directory `C:\Program Files (x86)\` or `C:\Program Files\`. Now go to the bouncer folder and double-click onto Install.exe. The program will now install the driver and the tools as if you had installed using the installation package.

Uninstallation

You can remove the driver and the tools at any time using the uninstall program. Simply run the program `Uninstall.exe`, which is included in the installation package. This program removes the driver, the tray application, the log and configuration file from the system.

Configuration notes

Bouncer works strictly according to the rules specified by you. From a folder that is not in the whitelist, no program file, nor any library (DLL) or diver (SYS) can be started. In the worst case, this can lead to the operating system not starting correctly or not starting at all. Therefore, it is very important to configure the list with caution. Always check the paths and file names you specify using the Explorer to ensure that the correct spelling has been selected.

Configuration in Unicode format

Before you can install and run the driver, you shall modify the configuration file regarding your individual installation of Windows and your installed applications. The configuration of Bouncer is specified in the `C:\Windows\bouncer.ini`. It is a file in Unicode format that can be opened with any common text editor (e.g. Notepad, Notepad++). You can also use characters from other languages, such as:

- Галдёж
- مْرَحَبًا
- הלו

Upper or lower case

Bouncer's rules are not case-sensitive. This means that you can specify the rules for file names and file paths completely in uppercase, mixed lowercase and uppercase, or only lowercase letters.

Wildcards

In addition, Bouncer also support wildcards. You can use these to generalize rules. For example, you can use `*.scr` to define that all files with the file `extension.scr` should be blocked. As symbols, Bouncer recognizes the **star *

for any number of characters and the **question mark ?** for exactly one character.

Configuration file `bouncer.ini`

To configure and activate Bouncer, go to the configuration file `bouncer.ini`. You can do this via the corresponding directory or via the tray application at the bottom right:

```
[#LETHAL]
[LOGGING]
[SHA256]
[#CMDCHECK]
[WHITELIST]
C:\Windows\*
C:\Program Files\*
C:\Program Files (x86)\*
C:\ProgramData\Microsoft\*
7FBFAB17FE55578159F482A3C9741F02EF5C15C939F4BF1C7B164FAA0AB6DDA3
[BLACKLIST]
C:\Windows\System32\msiexec.exe
C:\Program Files\Internet Explorer\iexplore.exe
C:\Program Files (x86)\Internet Explorer\iexplore.exe
[CMDWHITE-LIST]
!*explorer.exe>*wscript.exe*C:\Firmenskripte\*
*>*
[CMDBLACKLIST]
*explorer.exe>*wscript.exe*
[EOF]
```

Restart after each change of configuration

Every change in the configuration file requires a restart of Bouncer. This is the only way changes can be accepted.

Activating and deactivating Bouncer

The character **#** (also called hashtag) means switched off, without the hashtag the corresponding component is switched on:

`[#LETHAL]` = off
`[LOGGING]` = on
`[SHA256]` = on
`[#CMDCHECK]` = off

During the initial installation, Bouncer should always be inactive, i.e. `[#LETHAL]`. This allows you to test the settings without causing problems with incorrect configurations. Once you have completed and tested the configuration, you can activate Bouncer by setting `[LETHAL]`. Now unknown and dangerous programs are blocked.

Activating and deactivating the Log File

The character # means switched off, without the hashtag the corresponding component is switched on:

`[LOGGING]` = on
`[#LOGGING]` = off

We recommend that you always activate logging. Bouncer then writes each event to the log file (`C:\Windows\bouncer.log`). For example, if a program tries to start, that was not specified in the whitelist. You can open the log file easily by using the tray application. To do this, select the "open log file" option.

Turn hashing on and off

If you want to use hash values of files as a reference, enable `[SHA256]`. If you do not want to use the hash function of Bouncer, disable this function by setting `[#SHA256]`.

Read more about handling hash values in the chapter "Use of Hash Values".

Turn on Command line checking

If you'd like to use Bouncer's Command line checking engine, enable it by setting `[CMDCHECK]`. If you do not want to use Command line checking, specify `[#CMDCHECK]`.

Read more about Command line checking in the chapter "Command line checking".

Configure Whitelist

```
[WHITELIST]
C:\Windows\*
C:\Program Files\*
C:\Program Files (x86)\*
C:\ProgramData\*
```

Below the entry `[WHITELIST]` you define all paths from which executables are allowed to be started. Here you should define at least the file paths that are absolutely necessary for the operation of Windows and the programs you have installed, i.e. in particular all paths (or files) required by the Windows operating system. The most important paths are pre-specified by the installer.

As of Windows 7 - 10, these are usually the following paths:

```
C:\Windows\*
C:\Program Files\*
C:\ProgramData\Microsoft\*
```

If you are using a 64-bit version of Windows, you will also find the path for installed 32-bit programs:

```
C:\Program Files (x86)\*
```

Make sure that you end **each rule** with the `*` symbol. The star symbol serves as wildcard and allows all files and subdirectories in these folders. Alternatively, you can also list each file individually. But this is time-consuming and only makes sense in high-security areas.

Directories of third parties

Your computer's manufacturer may have added special paths for drivers and system applications. Ensure that you also include these paths into your whitelist. They are often located directly below the main drive (usually `C:\`):

- `C:\DELL\*`
- `C:\ASUS\*`
- `C:\DRIVERS\*`
- `C:\Intel\*`
- `C:\AMD\*`
- `C:\OEM\*`

Add other trusted programs to the whitelist

If you have installed additional programs like Gimp, Veracrypt or Notepad++ in other directories, you shall also add them to the whitelist. For example:

```
D:\PortableApps\VeraCrypt\*
D:\PortableApps\Gimp\*
```

In addition to path specifications, you can also enter individual program files in the whitelist. To do this, write the complete path with the file name and its extension in one line.

For example, you can allow certain program files without allowing the entire path and its contents. If there are several DLLs and EXE files in the folder `F:\Sandbox\`, but you only want to allow a specific application named `TestA.exe`, add the following rule to the whitelist:

```
F:\Sandbox\TestA.exe
```

Using Wildcards correctly

Bouncer supports wildcards. You can use these to define individual rules. For example: You want to allow all `.exe` files in the `F:\Sandbox` directory. Or you want to start files that start with A and ends with `.exe`. Or you want to use program files from any drive starting with hello and ending with `.exe`:

```
F:\Sandbox\*.exe
A*.exe
?:\hallo*.exe
```

The star symbol stands for one or more arbitrary characters, the question mark stands for exactly one character.

Priority Rules

Priority rules are rules, that can overwrite any other rules, whether they are on the white- or blacklist. Although Bouncer supports a very powerful rules mechanism, priority rules provide more flexibility. Priority rules can help to reduce the number of specific rules for example by just blacklisting a whole directory and whitelisting specific executables you would like to allow.

For example, we recommend blacklisting the path `C:\Windows\Temp\*`. All programs that are in this directory can no longer be started. However, it can happen that certain update programs and installers want to execute their processes in this folder. A priority rule can solve this problem. To do this, we give the desired rule a higher priority in the [`WHITELIST`] with an exclamation mark. The rule in the whitelist now overrides the rule in the blacklist.

Let's assume that the desired update is `AVUpdater.exe`. Then the rules are as follows:

```
[WHITELIST]
!C:\Windows\Temp\AVUpdater.exe
[BLACKLIST]
C:\Windows\Temp\*
```

Priority rules work in all rule areas: in the white- and blacklist and in the Command line check (`CMDCHECK`).

Advice

Rules with a higher priority must be the first in order. Example:

```
[WHITELIST]
C:\Windows\*
!C:\Windows\Update.exe
[BLACKLIST]
*Update.exe
```

wrong

```
[WHITELIST]
!C:\Windows\Update.exe
C:\Windows\*
[BLACKLIST]
*Update.exe
```

right

Use of Hash Values

Instead of directories and file names, Excubits Bouncer also supports SHA256 hash values of files. You can specify any SHA256 hash value of a file in the whitelist that shall be allowed. Specifying the file's path or name is not necessary here, just the hash value, but you can mix up hash value rules and path/filename rules together.

If you have set up a proper user management on NTFS drives, path rules are secure enough. If you want deeper security or have special external locations where executables shall be started from, SHA256 hash values are a great way to provide security on files, that might be subject for manipulation.

In general, we recommend that you only use hash values for directories that are potentially vulnerable to change. For example, network drives that do not have special authorization protection and can be changed by users. Here, application programs could be infected with a virus or replaced by malware, for example. On such drives hash values should then be used.

Configure Blacklist

```
[BLACKLIST]
C:\Windows\System32\msiexec.exe
C:\Program Files\Internet Explorer\iexplore.exe
C:\Program Files (x86)\Internet Explorer\iexplore.exe
```

Below the entry [BLACKLIST] you define all paths from which **no program code** is allowed to be started. Programs on this list are automatically blocked.

Blacklist Example 1: You want to block a certain directory

In the whitelist you have shared the Windows directory via C:\Windows*. The whitelist rule now allows all programs in this directory to be started, even in all subdirectories. If you want to block a certain directory, because you think it is unsafe you can put it onto the blacklist. For example, the directory C:\Windows\Fonts*. To block this directory and all programs in it:

```
[WHITELIST]
C:\Windows\*
[BLACKLIST]
C:\Windows\Fonts\*
```

You can block individual applications or entire directories. It is also possible to specify the hash value of a file to be blocked.

Blacklist Example 2: A security issue

Suppose a security hole has been discovered in Microsoft Browser Internet Explorer and there is no update for this vulnerability yet. You can now use the blacklist to prevent criminals from exploiting this gap. You can now simply define the following rule:

```
[WHITELIST]
C:\Windows\*
[BLACKLIST]
C:\Windows\Program Files\Internet Explorer\*
```

If you are using a 64-bit version of Microsoft Windows, add an additional rule:

```
[WHITELIST]
C:\Windows\*
[BLACKLIST]
C:\Windows\Program Files\Internet Explorer\*
C:\Windows\Program Files (x86)\Internet Explorer\*
```

With just some simple rules you can avoid running untrusted or exploitable applications or libraries. Once the vulnerability has been patched you can simply remove the rules, use the application again.

Advice

Instead of an entire directory, it may also be appropriate to disable certain files, such as a vulnerable DLL to a plug-in, for example, if they are at risk due to a security breach. It is often the case that certain libraries or plug-ins are vulnerable to attacks. Cyber criminals use exploits to trigger the security breach in such libraries/plug-ins to infect your computer. If you block the vulnerable library or plug-in using Bouncer's blacklist, they can no longer be exploited. After the libraries or plug-ins have been updated, you can remove the rule from the blacklist and use them again.

Caution with using blacklist rules

Please note that disabling programs (drivers, libraries or plug-ins) sometimes result in stopping the application or system from working properly. Hence, before disabling any executable you shall always test the behaviors and be careful with what you disable. We heavily encourage you to do some testing on demo or test machines, before deploying any updated Bouncer blacklist rules to a production line computer system.

Recommendations for the [BLACKLIST]

We recommend to blacklist the following system folders and applications because

they are often used as one step to infect computers with malware. The list is not complete, we will try to update if we have gained more information and knowledge about new threats and attacking techniques. You can download the latest version of the blacklist from here:
https://excubits.com/content/files/blacklist.txt

Silent Rules

Silent Rules allow you to block events which you do not want showing up in the logs. So, with Silent Rules you are able to calm down annoying alerts you cannot get rid of, because e.g. the operating system's core automatically triggers them without any chance to block them

For example: If you would like to blacklist a Windows' core library or driver that cannot be removed via the system's configuration, and thus causing "harmless"

alerts each and every time the operating systems tries to launch it. There is no way to avoid such attempts, but with Silent Rules you are able to calm them down. Just specify the $ character before a blacklist rule and it will not show up in the logs.

A simple silent rule is shown here:

```
[BLACKLIST]
$*notepad.exe
```

This example rule defines that `notepad.exe` should be blocked and that no log entry should be written to the log file. So, if Notepad starts, it will be blocked by Bouncer without any event logged.

Advice

Silent rules can only be specified in the blacklist areas [BLACKLIST] and [CMDBLACKLIST].

Parentchecking: Conditional rules for parent processes

Bouncer also supports conditional rules for parent processes in the [WHITELIST] and [BLACKLIST]. There are programs that start subprograms as required after starting the main program. We call the main program the parent process, the subprograms child processes. The fact that parent processes start subprograms, i.e. child processes, is necessary and useful for programs such as the Explorer. But hackers use this technique to execute their evil executables through media players, browsers or office applications. For example, a Word document contains a macro which forces Word to download and execute a cryptolocker. With parentchecking Bouncer can block such attacks.

During parentchecking, bouncer checks which parent processes wants to start some executable before executing the child process. If the corresponding parent process is on the whitelist, the child process is allowed to start, otherwise it will be blocked. For example, you can define that Word or a PDF reader may not execute processes, shell codes, runtime libraries or drivers (`.dll, .sys, .ocx, .drv, .cpl`).

The rules for parent checking have the following general format:

```
Path/Filename Father>Path/Filename Child
```

Please note that the path/file name is separated by the **>** symbol. No spaces are allowed in between. Bouncer also supports wildcards, including all Unicode characters, such as:

```
C:\مْرَحَبا\галдёж\x.exe>C:\Windows\*.dll
```

Advice

If you want to use parent rules and normal rules in Bouncer at the same time, you have to follow the correct order: The first rule fitting is the rule that is taken. You shall always put the most important rule in the first position.

Examples for Parent-Rules in the [WHITELIST]

Two sample parent rules for the whitelist are given here:

```
!*MicrosoftEdge.exe>*MicrosoftEdge.exe
!*microsoftedgecp.exe>*microsoftedgecp.exe
```

These two rules specify, that the Microsoft Edge Browser may only start itself and the so-called Microsoft Edge Content Process. You should then use a blacklist rule to restrict Edge itself from starting any other processes.

The following three rules allow Microsoft Word to start processes only from system folders:

```
*\Office1*\WINWORD*.EXE>?:\Windows\*
*\Office1*\WINWORD*.EXE>?:\Program Files\*
*\Office1*\WINWORD*.EXE>?:\Program Files (86)\*
```

With these two rules the user specific program Thonny for the user folder `C:\Users\Excubits\...` is allowed:

```
!C:\Users\Excubits\AppData\Local\Programs\Thonny\*>C:\Users\Excubits
\AppData\Local\Programs\Thonny\*
!C:\Users\Excubits\AppData\Local\Programs\Thonny\*>C:\Users\Excubits
\.thonny\*
```

Examples for Parent-Rules in the [BLACKLIST]

Some simple parent rules for the blacklist are given here:

```
[BLACKLIST]
*iexplore.exe>*cmd.exe
*iexplore.exe>*powershell.exe
*chrome.exe>*bitsadmin.exe
*firefox.exe>cmd.exe
*flash*>cmd.exe
*flash*>powershell.exe
*flash*>*script*.exe
*flash*>*bitsadmin.exe
*flash*>C:\Users\*
```

The examples just show rules for common applications that are often exploited. The first example defines that Internet Explorer is not allowed to start up a command line prompt (`cmd.exe`). The second example defines that Internet Explorer is not allowed to start the Powershell interpreter. The third rule disallows the Chrome Browser to start `bitsadmin.exe`. The fourth example interrupts Firefox to start a command line shell. The last rules ensure that Adobe

Flash (often exploited) cannot start critical and often misused system shells and malware droppers.

Nearly all Windows applications load many dynamic link libraries. Some attackers try to exploit applications or a system by letting an application load an infected (evil) library from a hooked place, instead the intended and original one, by changing the load order of libraries. Using parent blacklists can help to avoid such attacks, too. For example, you can block any library from user paths:

```
C:\Windows\*.exe>C:\Users\*.dll
```

Command line checking

With Command line checking you can specify which command lines certain applications may execute or not. This is especially useful for blocking or enabling so-called script interpreters, because the scripts loaded by the interpreter are passed via command line parameters to the application. For example, if you run a JS-file using Explorer, the path and file name are passed to the script interpreter (`wscript.exe`). With Command line checking you can determine which paths and files are allowed or not. We recommend that you only allow your own scripts and strictly block all others. This secures your IT and helps to run VB, JS and JAVA applications more securely than before.

CMD-Check-Whitelist configuration

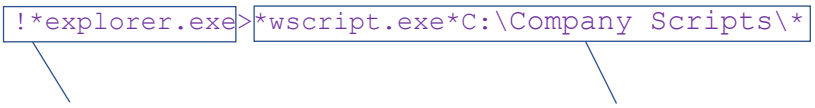
In the command line whitelist, specify the command lines you want to start. If you want to activate the Command line check, you must activate it with [`CMDCHECK`] and configure the [`CMDWHITELIST`].

Command line whitelist example

```
[CMDWHITELIST]
!*explorer.exe>*wscript.exe*C:\Company Scripts\*
*>*

[CMDBLACKLIST]
*>*wscript.exe*
```

The exclamation point at the beginning of the command line whitelist defines a priority rule. It is necessary because a strict rule in the command line blacklist prohibits all scripts of the `wscript.exe` interpreter. The `wscript.exe` program is usually only allowed to open scripts from the `C:\Company Scripts\*` directory. And only via Windows Explorer. Microsoft Word or Internet Explorer cannot run `wscript.exe`.



The entry before > defines the parent process. The entry after > defines the command line parameter.

Rules for Command line checking can become very complex. If you need more information and additional consultation, please do not hesitate and contact us. We offer additional trainings and support.

Configuring the command line blacklist

In the section for the command line blacklist you specify parents and command line commands you would like to block. Ensure you have enabled Command line checking by setting [CMDCHECK]. Then **you must** also configure the [CMDBLACKLIST].

Example of a CMDCHECK-rule:

```
[CMDBLACKLIST]
*>*wscript.exe*
```

In this example calling wscript.exe is forbidden for all parent processes. Here the first * defines any process. *wscript.exe* defines the command line parameter. All combinations of wscript.exe and commands to this process are now blocked. You can use this rule to ensure that no parent process can start wscript.exe with any command line parameter.

Advice

Since scripts are often used in SOHO environments blocking scripting hosts is not an option. Here Command line checking can help to mitigate. To get familiar with this option we recommend that you enable Command line checking in [#LETHAL] mode first, then leave the [CMDWHITELIST] empty and specify *>* in the [CMDBLACKLIST]. Then e.g. start applications, try to open scripts, open JAVA applications etc. Check the log file to see what Bouncer logs, then try to specify granular rules for your cmdwhitelist and cmdblacklist.

End of configuration

The configuration file shall always end with the following line:

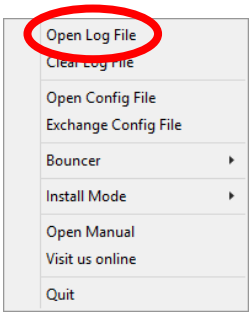
[EOF]

Advice

Please note that Bouncer does not accept the configuration file and does not load the driver if it is not completed with [EOF].

Prepare Bouncer for First Start / Simulation Mode

When you have completed all the steps from above, you can prepare Bouncer for the first start. To verify your configuration, you shall not start Bouncer in lethal mode, instead start it into simulation mode:



`[#LETHAL]` = off
`[LOGGING]` = on

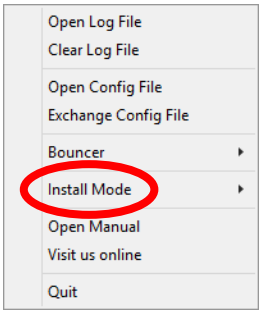
If you have activated the simulation mode, you can restart the computer. After restart, Bouncer now writes detected events into the log file, however in simulation mode **without blocking it**. If you have configured everything correctly, no messages should appear in the log file (`bouncer.log`). Check the behavior of Bouncer and your applications extensively to ensure you have not missed something. Remember to restart after each change of your configuration to apply the changes. You should now carry out these steps until Bouncer does not write any entries to the log file.

Once you are fine with your rules, you can switch Bouncer into lethal mode (=switching simulation mode off). To do so, set the following line in the configuration file:

`[LETHAL]` = on

Now you only have to restart Bouncer and the driver is now fully active. Unknown executables will blocked by Bouncer.

Installation Mode



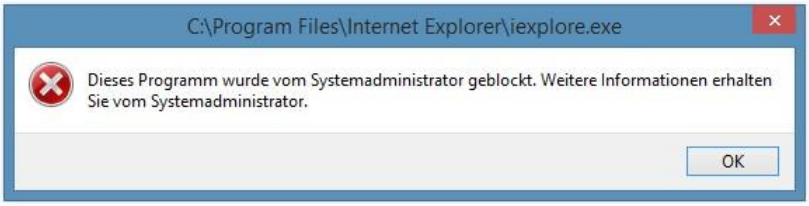
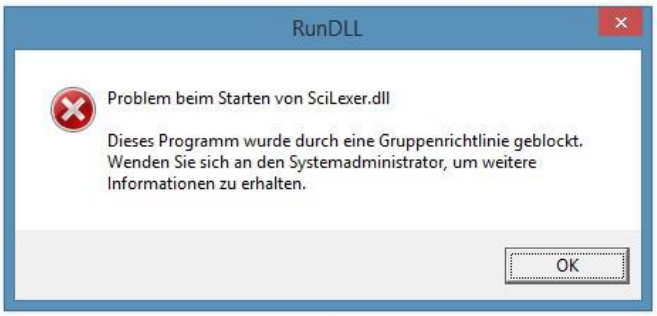
You want to install a Windows updates or a new program on your computer make use of installation mode. Windows can then use all directories for the update and can also make changes to the system while the computer is booting up.

Do not forget to deactivate the installation mode after installation.

Advice

In installation mode, Bouncer does not protect you from malware. You shall not surf the web or test untrusted software while in installation mode.

System warnings

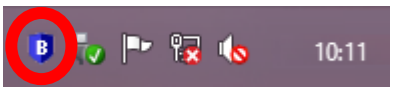
Message	Meaning
	This message is displayed if, for example, Internet Explorer is on bouncer's blacklist and has been blocked by Bouncer.
	If an application tries to run a DLL from a folder that is not shared, you might see this message.

Utilities (Tools)

Bouncer can always be used without any user-mode application and integrates transparently into your system, so there is no sign that Bouncer is present. The tools presented here are just an extra. They are not needed for use with Bouncer.

Application for the tray area

Bouncer Tray (`BouncerTray.exe`) can be found after installation in the tray area of the Windows task bar on the bottom right. You can recognize Bouncer by a B-symbol in different colors:



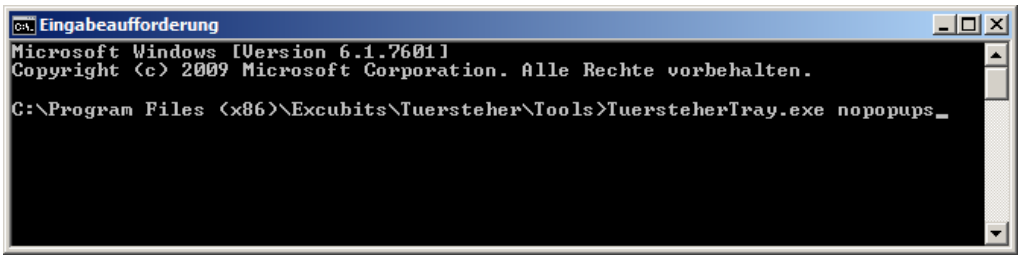
The active tray application checks whether the Bouncer log file changes.

Different colors for different modes

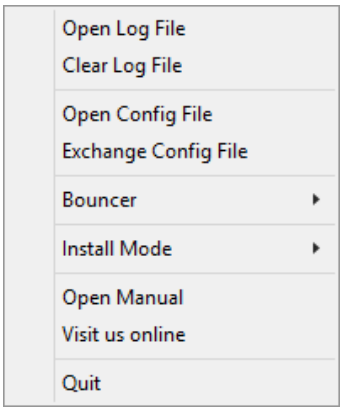
Tray Icon	Meaning
	If Bouncer is active and no events have been detected, the B-symbol is green.
	If Bouncer blocks an executable file, the icon turns red. In addition, the application displays in a balloon which file(s) have been blocked and writes this information to the Windows event log.
	If Bouncer has been switched to installation mode, the B-symbol turns yellow. You are not protected.
	If Bouncer is not active, the B-symbol is grey. You are not protected.
	If Bouncer is in simulation mode, the icon is blue. In simulation mode you are not protected.

Disable tray application alerts

If you do not want the tray application to display warning messages, you can start the application with the command line option `nopopups`, all tooltips are suppressed. To do this, start the tray application as follows:
`TuersteherTray.exe nopopups`.



Control during operation



If you click on the tray Application's icon (left-click), a context menu shall open up. There you can open up the log file, open up the `.ini` files, exchange ini file with another one, start and stop the driver, and to quit the tray Application. You can also turn Bouncer into install mode if you would like to install software and do not want to block the installer while doing so.

Advice

Administrator rights are always required to start or stop the driver, and to change the configuration.

Technical background

There are a lot of security applications on the market featuring heavy weight endpoint security solutions to defend against malware attacks. Such systems often install quite lot applications, dynamic libraries, they run several services and drivers that might slow down system performance and at the end the user does not know what such a product really does. Most of the solutions also need periodical updates to their engines and internal databases to function properly and to protect the system regarding the newest malware out there. Most of them also send back additional forensics (or telemetry) information, so the companies can measure out information of new (unknown) threats to build updates for. This can be critical in some scenarios regarding data protection and thus might be disabled; on the other hand, without additional forensic information classic AVs cannot build signatures detecting and mitigating against new attacks.

Add an additional layer of security

Problems arise if the security solution cannot update the databases or if it is subject to a new threat that is unknown to the AV. Attackers constantly create thousands of malware executables a day that are invisible to the majority of AVs on the market. For example, receiving an e-mail containing an infected attachment, one accidentally click installs the malware. Your AV can do little against such an unknown threat until it gets updated and is able to detect it – but it may then be too late, the system is already owned by malware and the AV might have been deactivated. Thus, no chance to clean it without huge effort and costs afterwards. Having another protective barrier up one's sleeve can prevent worse and will help to mitigate against. Excubits Bouncer is such an additional barrier. Having it set up correctly and running all the time, one can avoid a lot of those standard situations where malware usually gets executed or installed just by accident.

In depth protection

Bouncer does not scan for file extensions, it scans for memory initializations. The driver gets notified if any process tries to load executable code into memory, and this can be any type of file, including all extensions one can think of. Bouncer checks if the target memory was marked as executable, if this was the case, Bouncer's rules engine filters out the corresponding file. For this reason, someone can also load an image file, a MP3 or a text file with the executable flag and at the end this will also result in an alert. But normally only real applications are getting loaded with executable flag enabled, hence Bouncer's filter mechanism works very well and cannot be tricked by fake calls to ShellExecute or LoadLibrary using filenames like evil.exe.jpg or evil.dll.mp3 to load and execute a Windows executable with any name or extension.

In general, it is difficult to provide a full list of file extensions that will be blocked by Bouncer. Normally it should be .exe, .scr, .ocx, .com, .dll, .cpl, .lib, .so, .bin, .sys and .drv because these extensions are often used for executables.

Protection without constant queries

Bouncer works autonomously and is not dependent on decisions of the user. Bouncer never asks the user for a safety-relevant decision. Unlike other security products, Bouncer does not bother or even force the user to make any decision.

Protection at the earliest possible time

Bouncer fires up directly after kernel init, this means, the bootloader loads the kernel and performs its initializing process, then hands over to the kernel and performs the kernel init. This is where Bouncer is fired up, thus everything happens at an early stage and this is why Bouncer is able to protect your PC even on boot-up.

General recommendations

Bouncer can protect you from any executable files. Normally, you would not need any additional protection systems. Unfortunately, not all Windows users use Bouncer. It is therefore in all of our interest to use an antivirus program and a firewall in addition to Bouncer. This way, you can prevent dangerous programs that cannot be started by Bouncer from being (accidentally) distributed to other people.

Keep system and programs always up to date

You should also always import all updates and service packs for your operating system and all installed programs. This applies in particular to all applications used with the Internet such as browsers, plug-ins (e.g. Flash, PDF plug-ins, Java, .NET or Silverlight) etc.

Do not use administrator rights permanently

The computer should not be used permanently with administrator user rights. For everyday operation it is advisable to create a user account with standard user rights or limited user rights and to work with this user account. If possible, you should use a browser with sandbox technology. This reduces the risk of vulnerabilities being exploited. Google Chrome and the latest version of Internet Explorer have integrated sandbox technology.

Explanation of symbols / special characters

Symbol	Keyword	Meaning
#	Deactivate	off
?	Wildcard	Sign for exactly one arbitrary character.
*	Wildcard	Sign for any number of characters.
!	Priority rules	Rules with exclamation marks **(!)** take precedence over other rules.
$	Silent Rules	Rules with dollar signs **($)** at the beginning do not create log entries, although Bouncer block these files.
>	Parentchecking	Separator between parent and child process during parent checking. No space between them: `C:\مُرَحَبا\галдёж\x.exe>C:\Windows\*.dll`

Index of Keywords