Excubits Bouncer - Manual

Version 2.5.0 (June 2017)

excubits.com
Excubits UG (haftungsbeschränkt)
Simple. Secure. Excubits.

Imprint:

Copyright:	Excubits
Web:	https://excubits.com
Contact:	info@excubits.com
Version:	2.5.0
Status:	Released

Contents

1 Introduction

Thanks for choosing Excubits Bouncer, a powerful path-, hash- and parent-based whitelisting driver that assists you in monitoring, tracking and blocking malicious executables on Microsoft Windows systems.

Excubits Bouncer can lock down Microsoft Windows to prevent infections by typical malware (e. g. ransomware, unwanted downloads, unknown and untrusted software, and many more). When used properly, Excubits Bouncer will block most browser and e-mail based malware attacks. But Bouncer can also avoid accidentally starting malicious executables, dynamic link libraries and drivers from external USB drives, CD/DVD-ROMs, network drives, e-mail attachments, the web browser's cache and even through exploits. With Bouncer's parent checking feature you can also fully control what executables can be started with other executables. For example you are able to stop your web browser from starting system management tools like `reg.exe`, `bitsadmin.exe`, `powershell.exe` or other scripting hosts[1]. Hence you can add an additional security layer to your system, especially for vulnerable applications which are often abused to initially start first or second stage malware.

Excubits Bouncer was entirely written by our Development Team, no third parties commissioned, and is truly Made in Germany (with love). Excubits is proud to provide you software without pesky registration and without telemetry back channels. Our software is free of ads and hidden (spyware) functionality.

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 security results[2].

[1] System tools (shells, updaters, interpreters, etc.) are often misused to infect a system through an exploited application which starts a dropper or malware-loader that finally infects the system.

[2] Please note: Excubits Bouncer is not an Anti Virus (AV), thus it cannot remove malware.

1.1 System Requirements

Bouncer runs and protects the following versions of Microsoft Windows[3]:

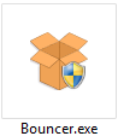
Version	32-bit/64-bit
Windows XP	*Business customers only, please contact us*
Windows Vista	*Business customers only, please contact us*
Windows 7	yes / yes
Windows 8 (includes 8.1)	yes / yes
Windows 10 (includes AU, CU)	yes / yes

To install and run Bouncer ensure, that the requirements to install and run your version of Microsoft Windows are met. At least you need enough free hard disk space to extract the installation archive completely. It is also recommended that at least 8 MBs of free disk space is available for installation and operation of Bouncer. You shall also install all recent patches and service packs before installing and using Bouncer[4].

Please note that Bouncer creates a log file where paths and file names of blocked executable files will be stored. Depending on the amount of log entries, this log file can require several megabytes on your disk. 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.

1.2 Installation Package

The installation package of Bouncer comes as a compressed RAR-archive self extracting executable:



[3] Excubits Bouncer also supports Microsoft Windows Server Editions, if the Server Edition's Kernel version bases on one of the Windows versions listed above.

[4] Ensure you have patched Windows Vista and 7 regarding the known code signature bugs which can cause failures while installing and starting kernel drivers using state of the art code signing certificates. See Windows-Patch KB3033929. Additionally you may also install patches KB2813430, KB3123479 and KB3097966.

By double clicking on the executable file, you can extract the archive and automatically run the installer. You can also use WinRAR or 7-zip to manually open and extract the files out of the executables archive[5].

1.2.1 Automatic installation

The self extracting executable can run an installer that setups the driver and the provided tools automatically. If you want to install the driver and tools of Bouncer automatically, just follow the dialogs of the installation program.

The installation program will create an initial configuration file, sets a tray application, installs the driver and starts it. In Chapter 2 you will find a detailed description of the configuration file and how to adjust the file regarding your individual system.

1.2.1.1 Perform Automatic-Installation manually

You can also initiate the automatic installation later, e.g. if you have extracted the self extracting archive manually. Just go to your installation folder and execute the application `Install.exe`:

This application will setup the driver and the provided tools automatically. Please ensure to install Bouncer on an empty folder, and avoid to install it on a folder that already contains other files and applications. Also ensure that the folder contains all files from the original archive, otherwise the installation may fail.

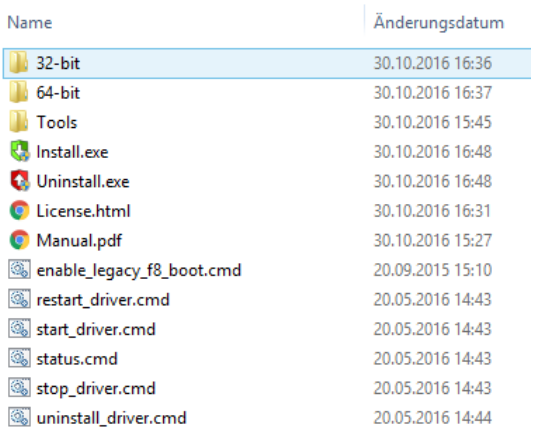
You can also remove and delete the driver, tools and all files (including the log and configuration) from the system by executing `Unistall.exe`:

This application will remove all Bouncer related files and Registry entries from your computer. To avoid loss of data in the `Unistall.exe` folder and its sub folders, please ensure that the folder containing `Unistall.exe` does not contain other files than the files shipped with Excubits Bouncer.

[5] 7-zip is free software, you can download it from http://www.7-zip.org.

1.2.2 Manual installation

In order to fully install Bouncer, extract the entire contents of the archive on your target computer that will run Bouncer. After unpacking the archive you shall see the following structure:



The folders `64-bit` and `32-bit` contain driver files for different architectures of Microsoft Windows, one configuration file and an empty log file. In the main directory (see the picture above) you shall also see the manual, general purpose control scripts (`.cmd` files) and a path to additional applications (`./Tools`) that help you to install and operate Bouncer more comfy.

1.3 Administration rights needed for installation

To install and configure Bouncer, you shall have administration access to the computer. After Bouncer was successfully installed and is running, you need no administration 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. Bouncer works independently in the Windows kernel and does not distinguish between different users[6].

For switching on and off, removing, or restarting the driver, administration access is required again. You can use an administration command line shell (`cmd.exe`) to do most of the operations using the provided control scripts or by directly calling the `net.exe` or `sc.exe` commands[7]. Please note that any user with normal privileges cannot disable nor uninstall Bouncer.

---

[6] We highly encourage you to not use your Windows PC with so called admin privileges for daily work and operations. For daily use we recommend to use the PC and applications with standard user privileges (or guest account privileges).

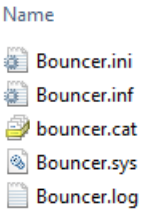
[7] For example use: `net stop bouncer`, `net start bouncer`, `sc query bouncer`, to uninstall the driver you can enter `net delete bouncer`

2 Prepare installation

To install Bouncer, first go into the appropriate subdirectory representing your architecture (32-bit or 64-bit) of Microsoft Windows. All steps described in the following sections always refer to the individually selected directory representing your architecture of Windows. If you do not know the exact architecture of your version of Microsoft Windows, you can just use the tool `WindowsArchitecture.exe` (see `./Tools`). It will show you a message box stating the architecture of your version of Windows.

For example, if you use a 32-bit version of Windows, use the driver located in the path named `\32-bit\`. If you use a 64-bit version of Windows, navigate to the driver located in `\64-bit\`.

Each directory contains the following files:



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 is located in the file `bouncer.ini`. The file is in Unicode format, i.e. all the entries may use letters of alphabets which are non-ASCII, for example Characters from Cyrillic, Asian and Arab alphabets can be used:

- مَرْحَبـاً
- הלו
- галдёж
-

Please note, that Bouncer internally does not distinguishes between uppercase and lowercase, hence Bouncer is case **insensitive**.

Bouncer does also support wildcards. A wildcard character is a symbol such as an asterisk (*) or a question mark (?) that is used to represent one (?) or more (*) characters. Hence wildcards can be used in place of one or more characters and can help to specify more complex rules, like for example all files in all sub folders in `C:\` ending

with .exe: `C:\*.exe`. Or all executables named `demo.exe` on all drives: `?:\demo.exe`.

Bouncer implements a very strict rules engine, meaning you shall **exactly** specify which files or paths are explicitly allowed and which are not. Files and paths that are not listed in the whitelist will be blocked by default, no matter where they are located or if they are necessary for the core system. Files and paths that are listed in the blacklist will always be blocked, even if there was a rule in the whitelist.

First, open the `bouncer.ini` file by double clicking on the file name. Now the text editor (in most cases Notepad) should open up and display the following configuration (the colors of the individual sections shown here are just for illustration). The configuration file is divided into six sections (marked as blue, green, red, purple, dark-red, and gray):

ⓘ Demo/Full version limits the size of the ini file to 5KB/3MB. If you exceed the size, the driver discards the whole configuration and will not start up, hence will not protect.

```
[#LETHAL]
[LOGGING]
[SHA256]
[#PARENTCHECK]
[#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
[PARENTWHITELIST]
C:\Windows\*>*
C:\Program Files\*>*
C:\Program Files (x86)\*>*
C:\ProgramData\Microsoft\*>*
[PARENTBLACKLIST]
C:\Program Files*\Google\*>*cmd.exe
C:\Program Files*\Google\*>*script.exe
C:\Program Files*\Google\*>*powershell*
[CMDWHITELIST]
!*explorer.exe>*wscript.exe*C:\Firmenskripte\*
*>*
[CMDBLACKLIST]
*explorer.exe>*wscript.exe*
[EOF]
```

The following subsections will describe how to configure the different parts of the `.ini` file. For your convenience and orientation the heading is kept in the same color as in the demo `.ini` shown above; please note that when you open the `.ini` file with a text editor there is no coloring, it is used within this documentation just for your convenience.

2.1.1 Activate and Deactivate Bouncer

We are now looking at the blue part:

```
[#LETHAL]
[LOGGING]
[SHA256]
[#PARENTCHECK]
[#CMDCHECK]
```

The blue section specifies whether Bouncer shall block detected files or not. When

- `[LETHAL]`

is defined in the configuration area of the `.ini` file. Bouncer will block any attempt to start an executable file from an untrusted path. We call Bouncer to be in lethal mode, e.a. the driver is armed and ready.

If you specify

- `[#LETHAL]`

in the configuration area of the `.ini` file, detected files shall be logged (if logging was enabled) but Bouncer will not block such executable files. In this mode, Bouncer is like a secured weapon, it is ready to enforce, but will not.

When you install and run Bouncer for the very first time we recommend to make use of the `[#LETHAL]` option to become familiar with the operation of Bouncer and to watch out what the system is doing by checking Bouncer's log file. Once you are sure that everything is working well, you can set the `[LETHAL]` option and turn Bouncer into lethal mode, and to protect your system.

2.1.2 Enable and disable logging

You can enable active logging by using the line

- `[LOGGING]`

or disable logging by using

- `[#LOGGING]`

in the configuration area of the `.ini` file. In principle, we suggest to always enable logging, so you can see which potential attacks Bouncer has detected.

If logging was enabled, Bouncer creates a logfile into your Windows's installation path (usually `C:\Windows\`) named `bouncer.log`. This file is in Unicode format and can be opened by any standard text editor such as Notepad (`notepad.exe`).

The log file can be accessed with just read access by any application, so you can write your own watch dog scripts, tools, or applications that can react on any change in the log file. For example your watch dog application can send SMS/e-mail text messages to your company's head of IT security, in case of an attack. You can also send snmp messages to your central management server etc[8].

2.1.3 Enable and disable hashing

Besides path rules for white- and blacklisting, Bouncer also supports high secure SHA256 hash values in its `[WHITELIST]` and `[BLACKLIST]` definitions.

If you enable hash-based checking you can alternatively white- or blacklist a file by its SHA256 hash value. To enable hash-based checking, set

- `[SHA256]`

to disable it, just set

- `[#SHA256]`

in the configuration area of the `.ini` file. With hashing mode you are able to protect from drives and paths where you do not have any access control enabled. This should help users to enhance security, but consider that the configuration is a bit messy. Please note, that you have to update the hash values on each and every update on the hash listed files. By definition of a cryptographic secure hashing function, any change of a file will cause to change the file's hash value, so you must update[9].

2.1.4 Enable and disable parent checking

Parent checking is another very powerful feature to control what executables a specific application is allowed to load and start, or what it is not allowed to start. This includes files like typical executable files, but also drivers, libraries, and plug-ins.

---

[8] If you need more information, additional consultation or a sample watch dog that you can modify regarding your needs, please do not hesitate and contact us. Our team will help you to get the best out of Bouncer.

[9] If you need more information or additional consultation our team is happy to help our customers with hashing mode.

To enable parent checking just set

- `[PARENTCHECK]`

to disable it just specify

- `[#PARENTCHECK]`

in the configuration area of the `.ini` file.

2.1.5 Enable and disable parent command line scanner

Bouncer also supports command line scanning, hence you can white- and blacklist command line parameters with Bouncer. You are able to white- and blacklist executables by their command line options. This feature can be very beneficial to lock down interpreters and virtual machine (e.g. .NET or Java) executables which are often misused by intruders and malware's first and second stage infection mechanisms[10].

Enable Command Line Scanning by setting

- `[CMDCHECK]`

Disable it, by setting

- `[#CMDCHECK]`

in the init part of Bouncer's .ini file. Also specify

- `[CMDWHITELIST]`

and

- `[CMDBLACKLIST]`

for your command line white- and blacklist.

Please start Bouncer in `[#LETHAL]` mode to play with this feature the very first time. Also check the `bouncer.log` to get comfortable with this feature and to learn the basic rules you should allow on a clean (golden image) system.

2.1.5.1 What is parent checking good for?

If you look at current attack mechanisms on office applications and browsers for example, most of the underlying and exploited file containers (document, media or web-content files) include some kind of malicious macro, script, or language direc-

[10] Information Security Analyst Casey Smith showed how to bypass ordinary application whitelisting solutions on Windows. To avoid such attacks you can combine classical whitelisting, parent checking and command line scanning which makes our solution splendid.

tive that initiates the final attack. Some other malicious document files make use of exploits, but at the end the result is the same: The malicious code ensures that the intended malware finds its way to your machine, gets persistent and started.

To start the malware, the attackers often use Windows API calls to e.g. Run, Exec, LoadLibrary or CreateProcess etc. These calls can easily be avoided with classic Bouncer if they target a blacklisted path (`%temp%`) as the source to start the malicious executable. But attackers do not just use API functions and executables. We also see a lot of exploits and malicious scripts/macros that start some scripting host first, to download, save, and make the final malware executable persistent on a target's machine.

For example they open up a command line shell (`cmd.exe`), start a batch file, run *Powershell* or *Windows Scripting Host* scripts etc. They also often call scripts or system tools like `reg.exe`, `bitsadmin.exe` to download a file, and make it autorun in registry, etc. With bouncer's parent feature you are able to limit the initial steps of such threats. You can, for example, specify that your office suite or web browser is not allowed to start up certain (often exploited) system tools like scripting hosts or the command shell to avoid exploration.

Take Microsoft Word as an example: For most all day, every day work, Word's executable `winword.exe` shall never ever run `cmd.exe`, `bitsadmin.exe` or `*script.exe`, nor `powershell.exe`. Hence you can set a parent rule saying that `winword.exe` is not allowed to run these executables. Just enforcing such rules helps a lot to mitigate against a whole bunch of threats that are currently in the wild. The same is true for browsers. Just think about `cmd.exe` again: Why should any web browser start up `cmd.exe`, `powershell.exe`, `*script.exe`, `bitsadmin.exe` etc. Hence from a in-depth security perspective it truly makes sense to have parent rules to mitigate against common threat vectors[11].

[11] If you need more information our development team is happy to help and assist.

2.2 How to configure the whitelist

We are now looking at the green part:

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

Define your whitelist below the entry `[WHITELIST]`. Here you can specify all paths of which you are going to start program code from. At least you shall specify all paths which are necessary for booting and operating Windows and your installed applications correctly.

These paths are typically:

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

If you are using a 64-bit version of Windows, you shall also define the path to your 32-bit applications. They are typically located at:

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

As you can see, wildcards are used in place of one or more characters and can help specifying more advanced rules. Here we have specified general rules for typical Windows paths and all paths beyond them. For example: Due to the fact that `C:\Windows\*` was whitelisted with the asterisk, we have implicitly whitelisted `C:\Windows\System32\` and all the other sub folders located under `C:\Windows`.

Please note: 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:\`).

For example, computers from the manufacturers DELL, ACER and ASUS, have often one of the following folders in `C:\`:

- `C:\ACER\*`
- `C:\DELL\*`
- `C:\ASUS\*`
- `C:\OEM\*`

- `C:\Intel\*`
- `C:\AMD\*`
- `C:\DRIVERS\*`

If you have installed applications on other drives, specify them, too. For example:

- `C:\My Personal Folder\Program A\*`
- `F:\Share\Apps\MySuperTool\*`

As already mentioned, all rules may also contain Unicode characters, for example something like:

- `C:\Users\مَـرْحَبَـا\ғалдёж\*`

Besides paths you can also specify individual program files in the whitelist. To do so, simply specify the full path, including the filename itself and the file's extension. For example, the folder `F:\Sandbox\` contains several applications and libraries, but you only want to allow (whitelist) the application `TestA.exe`. All the other DLLs and EXE files in the folder shall be blocked. Well, just add the following rule

- `F:\Sandbox\TestA.exe`

With this line, Bouncer only allows running application `TestA.exe`, all other program files in the directory `F:\Sandbox\` are still blocked.

If you just want to whitelist the drive letter using wildcards, you can use the question mark instead of the `F` character symbol. For example like `?:\Sandbox\TestA.exe`. This rule will allow `TestA.exe` in a folder named `.\Sandbox\` from all drives (`A` to `Z`), this will also include external devices like USB-Sticks, CD-ROMs etc.

2.2.1 Using 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 amount of specific rules for example by just blacklisting a whole directory and whitelisting specific executables you would like to allow. This can be very helpful for the typical temporary folders which, for security reasons, shall be blacklisted, but for some specific application should be allowed.

A priority rule can be set by adding the exclamation symbol **!** at the beginning of a rule's line, e.g.:

```
...
[WHITELIST]
!C:\Windows\Temp\AVUpdaterXy0001.exe
C:\Windows\*
C:\Program Files\*
C:\ProgramData\Microsoft\*
...
[BLACKLIST]
C:\Windows\Temp\*
...
```

In the example from above we declared `C:\Windows\Temp\*` to be on the blacklist. For good reasons you shall limit access to this folder, but it often happens that legit applications need to write and execute from `C:\Windows\Temp\`, hence you cannot block the folder without having issues afterwards. With priority rules you can define rules that will overwrite other rules, so in our example the whitelist rule

```
!C:\Windows\Temp\AVUpdaterXy0001.exe
```

will overwrite the blacklist rule

```
C:\Windows\Temp\*
```

Now the `AVUpdater.exe` can execute from `C:\Windows\Temp\` but other applications started from `C:\Windows\Temp\` will still be blocked.

Please note: If you have set priority rules in both sections `[WHITELIST]` and `[BLACKLIST]`, then the priority rule from `[BLACKLIST]` will always overwrite the priority rules from the `[WHITELIST]`.

You should also consider the order of the rules you wish to set. A priority rule shall be declared before any (general) rule that is not a priority rule and would also match. In the example from above we declared the priority

```
!C:\Windows\Temp\AVUpdaterXy0001.exe
```

before

```
C:\Windows\*
```

because the rule `C:\Windows\*` from the whitelist section also matches, if the updater `AVUpdateXy0001.exe` is called. But the rule `C:\Windows\*` is not a priority rule and is going to be overwritten (marked as blacklisted) by the rule

```
C:\Windows\Temp\*
```

from the blacklist section. So take care when defining priority rules. Also try different ways to overcome a rule you have defined to ensure you did not leave a gap that could cause serious security hollows.

2.2.2 Using 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.

Please note that for hash rules any legit update on a SHA256 hash value specified file results in an update to the hash value in the `.ini` file. The process of updating and maintaining is complicated, so you shall take extra care[12].

We also recommend that you limit access permissions onto such folders, e.g. open up a PowerShell console and type:

```
$acl = Get-Acl "C:\Program Files"
Set-Acl "D:\Portable Apps" $acl
```

In the example we first obtain the access permissions from a system folder, here it is `C:\Program Files\`, and copy the permissions to an external folder named `D:\Portable Apps\`.

2.3 How to configure the blacklist

We are now looking at the red part:

```
[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 shall define all paths from which no program code shall be started or which application you want to explicitly block on the machine. This option is ideal for blocking certain programs from paths that are part of a whitelisted path (an example will follow up below).

By default Bouncer will block all paths that are not specified in the whitelist. You could leave the blacklist blank (e.a. no entries below `[BLACKLIST]`), if there is nothing you want Bouncer to block.

[12] If you run high risk systems like payment servers, POS, or ATMs we recommend hash values for all files. Please contact us for more details on a comfy initialization and maintenance.

2.3.1 Example 1

Suppose you would like to block the Windows calculator (located in `C:\Windows\system32\calc.exe`) but have already whitelisted the path to Windows via

- `C:\Windows\*`

The given rule will also allow executables located in `C:\Windows\System32\`. Thus the calculator will also be allowed. To block the calculator (`calc.exe`) you shall explicitly deny the calculator with the following blacklist rule:

- `C:\Windows\System32\calc.exe`

Instead of a single application you can also block an entire directory. Just use the wild card symbol * after the path's name, for example `C:\Users\TestUser\*`.

2.3.2 Example 2

Suppose that Microsoft Internet Explorer (IE) is hit by a serious security vulnerability. If this vulnerability is not patched and you would like to prevent IE and its libraries getting exploited there is little you can do under normal circumstances, except of stopping to use your PC and disconnecting it from the Internet.

Well, with Bouncer it is quite easy to mitigate. Just define the following rule:

- `C:\Windows\Program Files\Internet Explorer\*`

If you are using a 64-bit version of Microsoft Windows, you shall additionally use the following rule to avoid running IE and its components:

- `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 closed you can simply remove the rules, use the application again, and everything is fine.

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.

2.3.3 Additional note on blacklist rules: be careful!

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 an production line computer system.

2.3.4 Using hash values

Instead of directories and file names Excubits Bouncer also supports SHA256 hash values of files for the blacklist. You can specify any SHA-256 hash value of a file in the blacklist that shall be blocked. Specifying the file's path or name is not necessary then. Please note that for hash rules any legit update on a SHA256 hash value specified file results in an update to the hash value in the `.ini` file!

2.3.5 Using 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 configuartion, 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 given here:

```
$*notepad.exe
```

With this Silent Rule the application `notepad.exe` will be blocked, but there will be no log entry if this rule was triggered. Please note that the **$** symbol shall be specified as the first character for the rule and before the priority identifier **!** and all the rest of the specific rule.

Silent Rules can be used in all blacklist parts of Excubits Bouncer, thus can be specified in `[BLACKLIST]`, `[PARENTBLACKLIST]` and `[CMDBLACKLIST]`.

2.4 How to configure the parent whitelist

```
[PARENTWHITELIST]
C:\Windows\*>*
```

```
C:\Program Files\*>*
C:\Program Files (x86)\*>*
C:\ProgramData\Microsoft\*>*
```

If you enabled parentchecking by setting the basic configuration you **must** configure rules in the `[PARENTWHITELIST]` part. This feature enables you to specify which parent process is allowed to start another executable. This includes all executables like `.exe`, `.dll`, `.ocx`, `.sys`, etc. Please note, even if you do not make use of the parent checking feature, you shall define at least the line `[PARENTWHITELIST]`.

Specify the parent (by path and exe or with wildcards), followed by the character **>** and then specify the child by its path and exe or by using wildcards:

*parent_exe_path***>***child_exe_path*

where `parent_exe_path` is the full path (incl. the file's name) to the parent's executable and `child_exe_path` is the full path to the child's executable. Please note and always ensure that between the character **>** there are no spaces allowed.

Wildcard symbols can be used in place of one or more characters and can help specifying more advanced rules like in this example:

- `C:\Windows\*>*`
- `C:\Windows\explorer.exe>*cmd.exe`
- `C:\Windows\explorer.exe>?:\TestAnwendung.exe`

The first rule specifies that all executables from `C:\Windows\` and its sub folders are allowed to initiate and start other processes (this also includes libraries and drivers). The second rule specifies that `C:\Windows\explorer.exe` is allowed to start the command line prompt. The third rule specifies that `C:\Windows\explorer.exe` is allowed to start the application `TestAnwendung.exe` from any drive.

Normally you shall specify the basic Windows Operating System folders to your parent checking whitelist and allow them to execute any other executable:

- `C:\Windows\*>*`
- `C:\Program Files\*>*`
- `C:\Program Files (x86)\*>*`
- `C:\ProgramData\Microsoft\*>*`

Additionally you can specify more specific and more granular rules, if you are a more advanced and practiced user. For example you could limit access to pre-defined (and known to be misused) executables, libraries and drivers only, limiting overall attack surface of applications.

If you have installed applications into other locations you shall specify dedicated rules for each and every application, too. For example:

- `C:\MyFolder\ProgramA\*>C:\Windows\*.dll`
- `C:\MyFolder\ProgramA\*>C:\MyFolder\ProgramA\*`
- `F:\Share\SuperTool\Tool\*>C:\Windows\*.dll`
- `F:\Share\Tool\*>F:\Share\Tool\*`
- `F:\Share\Tool\*>C:\Windows\*.dll`

As seen in the example, most applications need to access libraries from the system folders to run properly. Thus you shall at least specify the needed libraries precisely by name and path, or by just specifying a more generic rule like in the example above: `F:\Share\SuperTool\Tool\*>C:\Windows\*`

In general and for initial starting with parent checking rules we always recommend to specify

- `path_of_application\*>path_of_application\*`
- `path_of_application\*>C:\Windows\*.dll`

for each application that does not fall into the generic configuration.

More advanced rules may explicitly specify the libraries and executables by their path and filename to restrict what an application is allowed to start, especially what libraries it is allowed to be run with. To enumerate the executable libraries needed, you can start up Bouncer in `[#LETHAL]` mode without the dedicated rules for the application, start it up and check on the executables logged by Bouncer to specify more granular rules[13].

As noted above, Bouncer fully supports Unicode. As such you can also specify parent checking rules containing Unicode characters like:

- `C:\مَرْحَبًا\ғалдёж\x.exe>C:\Windows\*.dll`
- `C:\مَرْحَبًا\ғалдёж\x.exe>C:\مَرْحَبًا\ғалдёж\*.dll`

2.4.1 Be careful with specifying and using rules for the parentwhitelist

If you have parent checking enabled, but do not specify rules in the field dedicated `[PARENTWHITELIST]` area, there is high risk to crash or brick your system or applications. Therefore it is again very important to carefully define the rules for parent

[13] For our customers: If you have any questions or need help, please feel free and contact us on how to set up parent rules. We also offer in-depth training and additional consulting.

checking. At the beginning you shall specify at least system folders with general wildcard rules like in the example that was already mentioned above:

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

On 64-bit machines running a 64-bit version of Microsoft Windows you shall specify

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

If you have installed third party drivers and additional applications outside the Windows Operating system folders, you shall specify these folders too. For example:

- `C:\DELL\*>*, C:\HP\*>*, C:\ACER\*>*, etc.`
- `C:\Intel\*>*, C:\AMD\*>*`
- `C:\OEM\*>*`

Please ensure to protect these folders against unauthorized attempts to alter their content. On NTFS this can easily be achieved with access protection, also ensure that you do not use the system with administration permissions all the time then.

We also suggest that you start Bouncer in `[#LETHAL]` mode for the very first time using parent checking features to get comfortable with its configuration and to avoid crashing the system. If you feel comfortable with your configuration and there is nothing logged by Bouncer, you can switch to `[LETHAL]` mode.

2.5 How to configure the parent blacklist

```
[PARENTBLACKLIST]
C:\Program Files*\Google\*>*cmd.exe
C:\Program Files*\Google\*>*script.exe
C:\Program Files*\Google\*>*powershell*
```

If you do not want to block any parent process from starting a child process, just leave this part of the configuration empty. Please note, even if you do not make use of the parent checking feature, and even if you do not want to blacklist any parent you shall define at least the line `[PARENTBLACKLIST]`.

If you want to use the feature you can specify any parent (by path and exe or by wildcards), followed by the character > and then specify the child by its path and exe or by wildcards:

parent_exe_path>*child_exe_path*

Where `parent_exe_path` is the full path to the parent's executable and `child_exe_path` the child's full path to the executable that is not allowed to be started with the parent. Please note and always ensure that between the character **>** there are no spaces allowed.

Wildcard symbols can be used in place of one or more characters and can help specifying more advanced rules like in this example:

- `*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 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`[14]. 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 to avoid such attacks, too. For example you can block any library from user or external paths:

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

As noted above, Bouncer also fully supports Unicode for parent blacklist rules. As such you can specify parent checking rules containing Unicode characters like:

[14] The examples just show some applications that are often exploited by attackers via the browsers Internet Explorer and Google Chrome. These are just examples, there are many more applications that can be exploited through an exploit. If you need more information and additional consultation, please do not hesitate and contact us. We offer additional trainings and support.

- C:\مَـرْحَبَـا\галдёж\x.exe>C:\Windows*.dll
- C:\مَـرْحَبَـا\галдёж\x.exe>C:\مَـرْحَبَـا\галдёж*.dll

2.6 Configure the Command Line Whitelist

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

If you do not want to block any parent process from starting a child process, just leave this part of the configuration empty. Please note, even if you do not make use of the parent checking feature, and even if you do not want to blacklist any parent you shall define at least the line `[CMDWHITELIST]`.

If you want to use the feature you can specify any parent (by path and exe or by wildcards), followed by the character **>** and then specify the command line by its path and exe or by wildcards:

parent_exe_path>*command_line*

Where `parent_exe_path` is the full path to the parent's executable and `command_line` is the full command line parameter used by the parent. Please note and always ensure that between the character **>** there are no spaces allowed. Wildcard symbols ***** and **?**, and the priority rule symbol **!** can be used in place of one or more characters and can help specifying more advanced rules.

To get familiar with this option we recommend that you enable command line scanning in `[#LETHAL]` mode, then to 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 to whitelist and blacklist several applications and their command line options to get more practice. At the end of section 2.7 we show an example for the Windows Scripting Host interpreter, only allowing this interpreter to start scripts from entrusted locations.

2.7 Configure the Command Line Blacklist

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

If you do not want to block any parent process from starting a child process, just leave this part of the configuration empty. Please note, even if you do not make use of the parent checking feature, and even if you do not want to blacklist any parent you shall define at least the line `[CMDBLACKLIST]`.

If you want to use the feature you can specify any parent (by path and exe or by wildcards), followed by the character **>** and then specify the command line by its path and exe or by wildcards:

*parent_exe_path***>***command_line*

Where `parent_exe_path` is the full path to the parent's executable and `command_line` is the full command line parameter used by the parent. Please note and always ensure that between the character **>** there are no spaces allowed. Wildcard symbols ***** and **?**, and the priority rule symbol **!** can be used in place of one or more characters and can help specifying more advanced rules.

For example you can limit access to `.js` scripts to only start from trusted locations. To do so, you first blacklist scripts in the `[CMDBLACKLIST]` and then allow them from the trusted location. As you can see, the example makes use of advanced techniques like wildcards and priority rules:

```
...
[CMDWHITELIST]
...
!*explorer.exe>*wscript.exe*C:\CompanyScripts\*.js
[CMDBLACKLIST]
...
*>*wscript*
```

In the example we blacklist any command line operation on the *Windows Scripting Host* wscript and just allow it on `C:\CompanyScripts\*` for `.js` files. So professional users that need to execute scripts can still do so, but work in a more secure environment by just allowing scripts only from trusted (and secured) locations. You can adapt this example to other scripting hosts, too. It also works for the JAVA runtime environment. You could blacklist `*>*java*` in the `[CMDBLACKLIST]` and define an appropriate rule in the `[CMDWHITELIST]`, e.g. some similar to `!*cmd.exe>javaw <your opt> <path>`.

2.8 Finalizing the configuration

We are now looking at the gray part:

```
[EOF]
```

The configuration must always be finalized with a line containing `[EOF]`. If this line is missing, the driver will stop initializing itself and **will not protect** you.

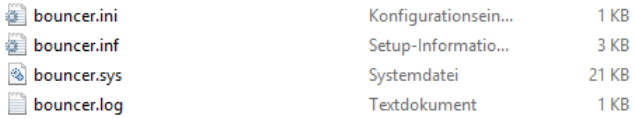
2.9 Prepare Bouncer for the first start

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 in non-lethal as follows

- `[#LETHAL]`

This way Bouncer starts in non-lethal mode and you can check the log file (`bouncer.log`) to see if your rules function as expected.

To install Bouncer, go to the directory regarding your Version of Windows and where you have already adjusted and modified the configuration file `bouncer.ini`:



Copy this `bouncer.ini` to your Windows system folder (in most cases `C:\Windows\`). Cross check again, that the copied file in the system folder contains all the rules you have specified, and that the option `[#LETHAL]` is set.

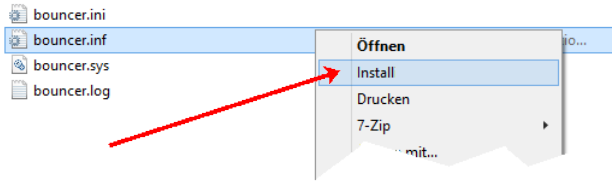
If not already done, copy the file `bouncer.log` to your Windows system folder (in most cases `C:\Windows\`), too.

Cross check again, that following files are located in your Windows system folder:

- `bouncer.ini`
- `bouncer.log`

and that `bouncer.ini` contains all your individually specified rules.

You can now install the driver by right-clicking on `bouncer.inf` and selecting the option „Install...“:



Now change to the root path of the Bouncer archive package and run the script `start_driver.cmd` as administrator (right click on the script file and select „Run as Admin“). Now Bouncer shall be running. You can restart your computer, if you

like or start testing right away. To stop the driver run the script `stop_driver` as an administrator.

Please note: You shall restart Bouncer every time you have changed the rules in `bouncer.ini`. You can use the script `restart_driver.cmd` and start it as an administrator. If you change `bouncer.ini` outside of your Windows system path (`C:\Windows\`), ensure that you copy the new version of `bouncer.ini` into your Windows' system path again, before restarting the driver.

Also ensure that all rules end up with a line feed and that they do not contain any space or tab characters. Bouncer will not trim the lines, so any space or other white space character is treated like a valid path and file name.

2.10 Function Testing

Open the log file `bouncer.log` and check whether there are applications logged. Since you allowed `C:\Windows\*`, the applications Explorer, Calculator, Notepad and Paint shall be started without any notice in Bouncer's log file.

For testing you may add `notepad.exe` or `calc.exe` to the blacklist, restart the driver and try to start `notepad.exe` or `calc.exe` again. It shall be blocked and listed in the log file.

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. Just try and plug an USB stick into your PC that contains some executables. Try to execute them from the USB stick. If there was no rule to allow executables from the USB stick's drive, they shall be blocked by Bouncer.

If you want to use parent checking you can also try:

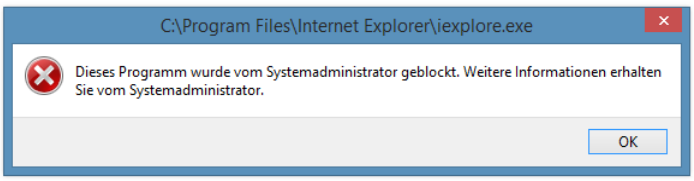
```
[PARENTWHITELIST]

*>*

[PARENTBLACKLIST]

*cmd.exe>*calc.exe
```

Now try to start the calculator through a command line shell. Since there is a dedicated blacklist rule, it shall be reported.

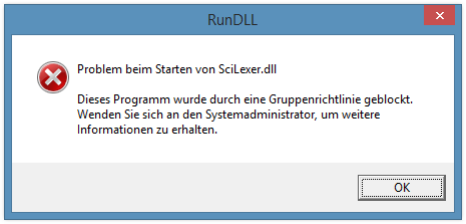
Once you are fine with your rules, you can switch Bouncer into lethal mode. Change your `bouncer.ini` from `[#LETHAL]` to `[LETHAL]`. After changing the corresponding row in `bouncer`.ini you shall copy it to your Windows system folder (usually `C:\Windows\`). Then restart the driver. Now Bouncer should be active and in the le-

thal mode. Program files outside the whitelisted paths cannot be started and will be blocked.

If you enable, for example, the sample rule from above regarding Internet Explorer, the following message should be displayed when attempting to start IE:



When an application attempts to run a DLL from a folder that was not whitelisted, you shall for example see the following message box:



2.11 Important note on new rules and change of existing rules

Please note that you shall restart (or stop and start) the driver each and every time you have changed your rules in `bouncer.ini`. Since the driver is independent from any user mode application and process it does not know of changes, so you have to restart.

If you have specified any SHA256 hash value rules always consider that any change on the specified file will cause that the file's hash value changes, too. If you still want to white- or blacklist a legit changed file, you must update the hash value in the `.ini` file, too. Special care must be taken if you install a patch or update to your system. Also ensure that you only update hash values of trusted applications. Excubits provides additional trainings and consultation for hash based rules and can assist you on setting up such rules for your dedicated ATMs and POS. If you have any questions, please contact us.

3 Tools

The installation package of Bouncer features two additional tools for your convenience: A *tray application* monitoring the log file and helping you to define rules and do some basic configuration of the driver. These tools are not needed to run the driver. 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 are just an extra, but not needed for use with Bouncer.

3.1 The Tray Application

The tray application is a simple Application that creates a tray icon in the Windows task bar indicating the current status of the driver.

If everything is okay, and the driver is running you shall see a green B-icon:

If Bouncer has detected an attempt to start an executable from an untrusted source, the tray icon shall turn into an red B-icon:

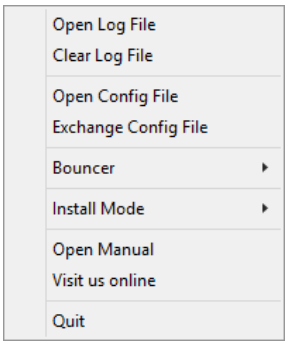
If Bouncer is in install-mode, the B-icon shall be yellow:

If Bouncer's driver was stopped (deactivated), the B-icon shall be gray:

If the driver is not running for more than 30 minutes, the Application will remind you to start the driver again.

The Tray Application by default does show balloon tool tips, if you do not like to be notified you can disabling the tool tips by calling the Tray Application with the command line option `nopopups`.

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. Additionally you can also open this Manual and visit us online at https://excubits.com.

Please note: For security vivid operations like starting and stopping the driver, administration rights are necessary, default (non-admin) users cannot stop, restart or set the driver into install mode. Additionally we also recommend to use notepad++[15] to edit the configuration file comfy.

[15] You can download notepad++ here: https://notepad-plus-plus.org/
It is open source and free of charge.

4 Some additional notes

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 a 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.

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.

Of course Bouncer is not the Silver Bullet and there are still attack vectors to pass by, there are other solutions with greater mitigation and security impact, but they often come with higher complexity and are not easy to maintain. We belief that Bouncer features the right balance between usability and additional security with regards to the ordinary Windows installation and everyday business. It can avoid classic attack vectors through exploits that target applications, where just one accidental click infects a system, even if it is protected by an AV. On the other hand, using Bouncer is not too complicated, integrates transparently into existing systems, so users will not notice that Bouncer is running and thus are not bothered and this shall not be underestimated. A security tool that bugged out its users will more likely be disabled or worked around, making the overall system even more prone for attacks.

Please note: Excubits Bouncer is not an Anti Virus, thus cannot clean any malware infected files or computers. Bouncer ideally enhances system security in combination with a firewall and AV installed, hence can mitigate against attack vectors that cannot be overcome by the ordinary AV or combined Desktop Firewall, because of the problematic nature regarding timely AV definition updates.

Together with a sandboxed web browser (e.g. Google's Chrome Browser), not surfing with administration access rights highly increases security on daily work. If additionally used together with Microsoft's Enhanced Mitigation Experience Toolkit[16] (EMET), overall security is close to a Silver Bullet. EMET is a great set of tools designed to protect your Windows-based systems before new security threats are addressed by security updates through the vendor itself or security products like malware scanners.

4.1 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

[16] For more details see http://www.microsoft.com/emet.

```
# Türsteher/Bouncer Anti-EXE Blacklist
#
*\AppData\Local\Temp\*.bat
*\AppData\Local\Temp\*.cmd
*\AppData\Local\Temp\*.com
*\AppData\Local\Temp\*.exe
*\AppData\Local\Temp\*.scr
*\AppData\Local\Temp\*.sys
*\AppData\Roaming\*.bat
*\AppData\Roaming\*.cmd
*\AppData\Roaming\*.com
*\AppData\Roaming\*.exe
*\AppData\Roaming\*.scr
*\AppData\Roaming\*.sys
*\at.exe
*\Temp\*.zip*\*.exe
*\Temp\*7z*\*.exe
*\Temp\*rar*\*.exe
*\Temp\*sfx\*.exe
*\Temp\*wz*\*.exe
*aspnet_compiler.exe
*attrib.exe
*auditpol.exe
*bash.exe
*bcdboot.exe
*bcdedit.exe
*bitsadmin*
*bootcfg.exe
*bootim.exe
*bootsect.exe
*ByteCodeGenerator.exe
*cacls.exe
*cdb.exe
*csc.exe
*csi.exe
*debug.exe
*DFsvc.exe
*diskpart.exe
*dnx.exe
*eventvwr.exe
*fsi.exe
*hh.exe
*IEExec.exe
*iexplore.exe
*iexpress.exe
*ilasm.exe
*InstallUtil*
*InstallUtil.exe
*journal.exe
```

```
*jsc.exe
*kd.exe
*lxssmanager.dll
*mmc.exe
*mrsa.exe
*MSBuild.exe
*mshta.exe
*mstsc.exe
*netsh.exe
*netstat.exe
*ntsd.exe
*odbcconf.exe
*powershell.exe
*powershell_ise.exe
*PresentationHost.exe
*quser.exe
*rcsi.exe
*reg.exe
*RegAsm*
*regini.exe
*Regsvcs*
*regsvr32.exe
*RunLegacyCPLElevated.exe
*runonce.exe
*scrcons.exe
*script.exe
*sdbinst.exe
*set.exe
*setx.exe
*Stash*
*syskey.exe
*systemreset.exe
*takeown.exe
*taskkill.exe
*UserAccountControlSettings.exe
*utilman.exe
*vbc.exe
*vssadmin.exe
*windbg.exe
*wmic.exe
*xcacls.exe
?:\$Recycle.Bin\*
C:\ProgramData\Microsoft\Windows Defender\Scans\FilesStash\*
C:\Users\Public\*
C:\Windows\ADFS\*
C:\Windows\debug\WIA\*
C:\Windows\Fonts\*
C:\Windows\PLA\Reports\*
C:\Windows\PLA\Reports\de-DE\*
```

```
C:\Windows\PLA\Rules\*
C:\Windows\PLA\Rules\de-DE\*
C:\Windows\PLA\Templates\*
C:\Windows\Registration\CRMLog\*
C:\Windows\servicing\Packages\*
C:\Windows\servicing\Sessions\*
C:\Windows\System32\Com\dmp\*
C:\Windows\System32\FxsTmp\*
C:\Windows\System32\LogFiles\WMI\*
C:\Windows\System32\Microsoft\Crypto\RSA\MachineKeys\*
C:\Windows\System32\spool\drivers\color\*
C:\Windows\System32\spool\PRINTERS\*
C:\Windows\System32\spool\SERVERS\*
C:\Windows\System32\Tasks\*
C:\Windows\System32\Tasks_Migrated\*
C:\Windows\SysWOW64\Com\dmp\*
C:\Windows\SysWOW64\FxsTmp\*
C:\Windows\SysWOW64\Tasks\*
C:\Windows\Tasks\*
C:\Windows\Temp\*
C:\Windows\tracing\*
```

5 FAQ

5.1 Bouncer is a strange name. What does it mean?

You know these guys at the front and back door of your favorite night club that check the guest list, your dress code etc. We thought that the word bouncer perfectly describes what our software does: it checks the entry against a guest list (the whitelist) and lets executables in or not.

5.2 What kind of malware will Bouncer protect against?

Bouncer is a path-based whitelisting driver that can block malicious executables like EXEs, DLLs and system drivers on Windows. Bouncer ensures that you cannot start and init an executable image (exe, dll, sys, ocx, scr) by accident, for example if you click on an attachment of an e-mail or download you did not initiate.

Bouncer can lock down your Windows OS to prevent infection by typical malware and ransomware, especially the well known crypto locker malware. Bouncer can also expeditiously avoid starting malicious executables, dynamic link libraries and drivers accidentally from external USB drives, e-mail attachments, network shares, the browser's cache and even through a nasty exploit for example.

5.3 Is there a list of executable extensions Bouncer blocks?

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 is 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 files 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, .dll, .lib, .so, .bin, .sys and .drv because these extensions are often used for executables. But it can be any other extension, too.

5.4 Are path rules secure?

Well, it depends on how you define secure and especially the conditions you use Bouncer. During development we ran through dozens of proof of concepts, we tried different options and configurations etc. Finally we came to the conclusion that we must balance between comfort and effective protection.

Bouncer features the right balance between usability and additional security with regards to the ordinary Windows installation and everyday business. It can avoid classic attack vectors through exploits that target applications, where just one accidental click infects a system even if it is protected by an AV. On the other hand, using Bouncer is not too complicated, so users will not notice that Bouncer is running and thus are not bothered. The latter should not be underestimated, because a security tool that bugged out its users will more likely be disabled or worked around, making the overall system even more prone for attacks.

Of course Bouncer is not the Silver Bullet and there are still attack vectors to pass by, there are other solutions with greater mitigation and security impact, but they often come with higher complexity and are not easy to maintain.

Bouncer ideally enhances system security in combination with a firewall and AV installed, hence can mitigate against attack vectors that cannot be overcome by the ordinary AV due to the update difficulty mentioned above. Together with a sandboxed web-browser, not surfing with root/admin permissions highly increases security on daily work. If additionally used together with Microsoft's Enhanced Mitigation Experience Toolkit (EMET), overall security is close to a Silver Bullet. EMET is a great set of tools designed to protect your Windows-based systems before new security threats are addressed by security updates through the vendor itself or security products like malware scanners.

5.5 What is the benefit of hash rules?

With hashing mode you are able to protect from drives and paths where you do not have any access control enabled. This should help some users to enhance security, but consider that the configuration is a bit messy. Please note, that you have to update the hash values on each and every update on the hash listed files, since by definition any change of a file will cause to change the file's hash value.

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. For example: if you need to use TrueCrypt on an external drive and if you have installed its portable version onto this

drive to easily access the encrypted container on every PC you plug in the drive, the application is potentially prone to be attacked (infected, changed, etc.). If you calculate TrueCrypt's (and its drivers) hash value and put these into the `.ini` file you can ensure that you always execute a legit version of TrueCrypt.

Some users in the field of production lines, Service PCs, POS, and ATM want to ensure that their machines only run entrusted code, independently from access permissions on the drive or pre-installed software by the vendor. They often do not want to whitelist applications through software updates without additional checks. These users can hash the whole system (or just the executables they need and entrust) to ensure that the system can only execute code that was whitelisted through a dedicated hash list. This is a solution for high security facilities and for systems in the area of banking, critical infrastructures, law enforcement, military, intelligence agencies, etc.

5.6 Is Bouncer a Kernel Mode Driver (KMD)? How does it work?

Yes for sure, Bouncer is a real kernel mode driver (KMD), it fully runs in the Windows kernel. There are no tricks and no gimmicks. The driver is absolutely independent from user-mode, it does not communicate with any servicing process in user mode, this is what makes Bouncer very special in its own sense. Most of the other security applications need at least one user-mode service (or other auto-start application) that manage how their engine works. Technically, Bouncer implements a WDF minifilter KMD that filters out binary executable code. For more details on kernel based monitoring, please read our technically supported whitepaper KernelBased-Monitoring written by Florian Rienhardt (Founder and Chief Development Officer at Excubits).

5.7 What means pausing the driver?

Pausing the driver is technically equal to stop the driver, doing whatever you want to do and then start the driver again. It is just a convenience feature.

5.8 When does the driver start up?

In a very early stage on boot up. The driver 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 in an very, very early stage and this is why Bouncer is able to lock down your PC even on boot-up, because the driver is able to block system drivers and system applications and libraries needed to fully boot and start up Windows. We think that this is something very special in contrast to other solutions that often use drivers on a higher level or just ordinary Windows Services that get started later on.

You can proof it by yourself by just setting no whitelist rules to the `.ini` and by starting bouncer in `[#LETHAL]`, `[LOGGING]` mode. You will see what drivers will have been blocked on start up, here you can see that Bouncer starts at an very early stage. Bouncer would be able to block system critical drivers, too. Not that you shall block them, but you can see how early Bouncer is ready to protect.

5.9 Some software isn't working properly now, what can I do?

Enable the so called non-lethal mode of Bouncer by setting `[#LETHAL]` in the `Bouncer.ini`. Then restart the driver and try to install or start the software that was not working properly. Open the log file (`Bouncer.log`) and check if there are any files logged that were not caught by your current rules set. If so, try to add the files or paths to your rules set and start your application again. If Bouncer does not block and log any executables then, you can enable lethal mode again and your software should now run properly. If not, contact me with detailed information about your OS, installed software etc.

5.10 Is Bouncer bullet proof, 100% secure?

Well, answering bullet proof or silver bullet question is a bit difficult. There is no protection software out there that will guarantee 100%. Bouncer can avoid classic attack vectors through exploits that target applications, where just one accidental click infects a system even if it is protected by an AV. On the other hand, using Bouncer is not too complicated, so users will not notice that Bouncer is running and thus are not bothered. The latter should not be underestimated, because a security tool that bugged out its users will more likely be disabled or worked around, making the overall system even more prone for attacks.

Bouncer ideally enhances system security in combination with a firewall and AV installed, hence can mitigate against attack vectors that cannot be overcome by the ordinary AV due to the update difficulty mentioned above. Together with a sandboxed web-browser, not surfing with root/admin permissions highly increases security on daily work. If additionally used together with Microsoft's Enhanced Mitigation Experience Toolkit (EMET), overall security is close to a Silver Bullet. EMET is a great set of tools designed to protect your Windows-based systems before new security threats are addressed by security updates through the vendor itself or security products like malware scanners.

5.11 Does Bouncer support Windows Server?

Yes it supports Windows Server, see introduction of this manual. Bouncer can dramatically helps to secure Cloud Infrastructures based on Microsoft Windows Servers.

Bouncer also supports Windows Server Core Editions, yet another thing that makes our solution very special and outstanding!

If you need more information, please do not hesitate and contact us.

5.12 Does Bouncer supports Virtual Machines?

Yes of course, Bouncer perfectly runs on VM-based Windows installations. It can not only protect your virtualized Windows guest system it can also help to dramatically enhance security of your Windows-based VM-host. If you have any questions on virtualization, please feel free and contact us.

5.13 Can I use on and build a VM-based Secure Desktop?

Sure, you can use Bouncer as a base for virtualized (VM-based) Secure Desktops in modern BYOD scenarios as well as on highly secured virtual business desktop environments. Excubits Bouncer runs on many Virtual Machine Platforms (e.g. VM-Ware, Virtual-Box, SINA VW) and you can use it to build up secure Virtual Desktops or the underlying Hosts to enhance overall security. If you need more information, please do not hesitate and contact us.

5.14 Does Bouncer support the Windows Event Log?

Yes it does. Additionally you can also set up your own reporting tools for central reporting (e.g. sending e-mail warnings to administrators, SMS or short messages to mobile and smart phones, snmp traps, etc.). Bouncer is open and we support our customers to integrate the driver into their existing infrastructure on request.

5.15 Some (automatic) software updates cannot be installed, why?

Some software updating processes internally work the same way as malware. They create temporary folders, copy executable files into such and start their update tools from there. As such, they will also be blocked by Bouncer. In such cases you SHALL disable Bouncer while installing updates of such kind. Do not forget to enable the driver after having installed the update.

5.16 Does Bouncer support Windows 10 Anniversary Update and Windows 10 Creators Update?

Bouncer supports Windows 7, 8, 8.1, 10, 10 Anniversary Update and Creators Update: 32-bit and 64-bit editions.