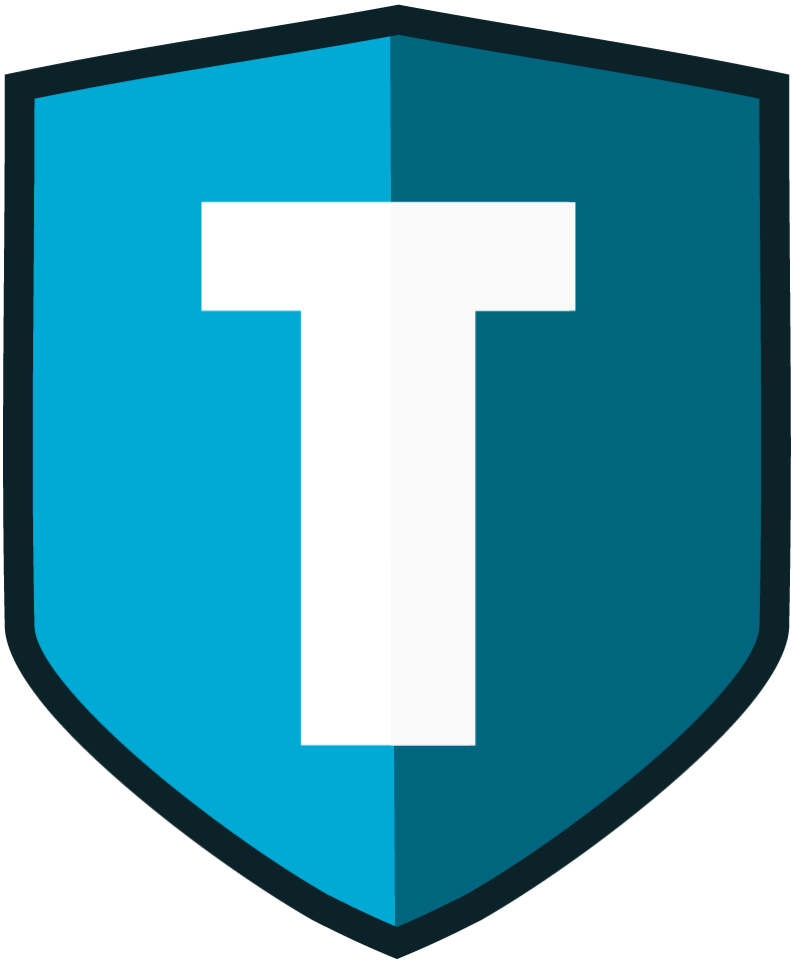
Handbuch für Türsteher

Version 2.4.1 (Oktober 2016)



excubits.com
Excubits UG (haftungsbeschränkt)
Einfach. Sicher. Excubits.

Impressum:

Copyright:	Excubits
Internet:	https://excubits.com
E-Mail:	info@excubits.com
Version:	2.4.1
Status:	Veröffentlicht

Inhaltsverzeichnis

1 Einleitung

Herzlich willkommen zu Türsteher, Ihrer Sicherheitssoftware aus Deutschland. Nach dem Türsteher-Prinzip lässt unsere Sicherheitssoftware nur die Programme auf Ihrem System laufen, die auf der Gästeliste stehen. Dadurch kann Türsteher vor unbekannter, nicht freigegebener und schädlicher Software schützen.

Zahlreiche spektakuläre Angriffe auf Computersysteme zeigen, dass IT-Systeme zu einem lukrativen Ziel für Cyber-Verbrechen werden. Klassische Abwehrstrategien mittels Firewalls und Anti-Viren-Systemen können ausgeklügelten Angriffen heute nicht mehr Stand halten. Veraltete Viren-Definitionen, zu schwache Firewall-Regeln und den Schutzprogrammen noch unbekannte Attacken lassen intelligent durchgeführte Angriffe zu.

Nicht so bei Türsteher, durch sein innovatives Schutzkonzept kann proaktiv verhindert werden, dass unbekannte oder bösartige Programmdateien auf Windows-PCs ausgeführt werden. Einmal konfiguriert kann Türsteher ein System somit ohne weitere Aktualisierungen vor neuen, unbekannten Schadprogrammen schützen.

Der in Türsteher implementierte Kernel-Treiber blockiert beispielsweise das versehentliche Ausführen von verbotenen Programmdateien von USB-Sticks, externen Festplatten, Netzlaufwerken, CD-/DVD-ROMs oder E-Mail-Anhängen. Außerdem kann Türsteher verhindern, dass Schadprogramme durch Sicherheitslücken auf dem Windows-PC gestartet werden könnten, oder dass Anwender heruntergeladene und nicht freigegebene Programme ausführen. Über Elter-basierte Regeln können Sie zudem genau festlegen, ob und welche Anwendung ein Programm starten darf und welche nicht. So ist es beispielsweise möglich Ihrem Web-Browser zu verbieten, Systemprogramme zu starten[1]. Mit dem Kommandozeilen-Scanning können Sie außerdem festlegen, mit welchen Kommandozeilenparametern ein Programm gestartet werden darf. Mit dieser Funktion können Sie Interpreter wie Powershell, JScript, Java, Python und weitere stark absichern.

Wie Türsteher funktioniert und wie das Schutzsystem zu konfigurieren ist, wird in den folgenden Kapiteln dargestellt. Nehmen Sie sich für die Installation und Konfiguration Zeit. Um Türsteher optimal zu nutzen und Ihr System sicher betreiben zu können, ist es sehr wichtig, die Funktionsweise von Türsteher verstanden zu haben.

[1] Eine häufig durchgeführte Angriffsmethode ist, im Browser über eine Sicherheitslücke Code zu starten, der seinerseits Systemprogramme (z. B. `cmd.exe`, `bitsadmin.exe` oder `powershell.exe`) ausführt, um den Rechner dann dauerhaft mit Schadprogrammen zu infizieren.

1.1 Systemanforderungen

Türsteher läuft unter den folgenden Versionen von Windows[2]:

Version	32-bit/64-bit
Windows XP	*Auf Anfrage, nur für Firmenkunden*
Windows Vista	*Auf Anfrage, nur für Firmenkunden*
Windows 7	ja / ja
Windows 8	ja / ja
Windows 8.1	ja / ja
Windows 10	ja / ja
Windows 10 Anniversary Update	ja / ja

Zum Ausführen von Türsteher sind nur die von Microsoft für die fehlerfreie und optimale Nutzung von Windows spezifizierten Systemanforderungen zu erfüllen. Es wird empfohlen, mindestens 8 MB zusätzlich freien Festplattenspeicher für die Installation und den Betrieb von Türsteher frei verfügbar zu haben.

Hinweis: Beachten Sie auch, dass Türsteher eine Protokolldatei anlegt, in der die Dateipfade und Dateinamen blockierter Programmdateien gespeichert werden. Abhängig von der Anzahl der protokollierten Einträge kann diese Protokolldatei auch mehrere Megabytes groß werden. Sorgen Sie dafür, dass für die Protokolldatei ausreichend Speicherplatz auf der Festplatte zur Verfügung steht, bzw. löschen Sie ältere Einträge in der Protokolldatei, wenn Sie diese zwecks Beweissicherung nicht mehr benötigen.

1.2 Installationspaket

Das Installationspaket von Türsteher liegt als komprimierte Installations-Anwendung (`Tuersteher.exe`) vor:

Mit einem Doppelklick auf die ausführbare Datei können Sie das Installationspaket von Türsteher in ein Verzeichnis Ihrer Wahl entpacken. Wenn Sie die EXE-Datei nicht

[2] Auch Microsoft Windows-Sever (auch Core-Edition), sofern die jeweilige Server-Version auf demselben Betriebssystemkern wie eines der oben genannten Betriebssysteme basiert.

ausführen möchten, können Sie diese mit WinRAR oder 7zip auch entpacken[3] und die Installation dann manuell durchführen.

1.2.1 Automatische Installation

Optional kann das Installationsprogramm den Treiber und die mitgelieferten Tools automatisch für das System konfigurieren und starten. Wenn Sie die mitgelieferten Tools und den Treiber von Türsteher automatisch installieren möchten, folgen Sie einfach den Dialogen des Installationsprogramms.

Das Installationsprogramm installiert den Treiber und legt eine Basiskonfiguration an. In Kapitel 2 finden Sie eine detaillierte Beschreibung der Konfigurationsdatei. Das Installationsprogramm hat Ihnen einige Schritte bereits abgenommen. So finden Sie die `türsteher.ini` Datei in Ihrem Systemverzeichnis (meist `C:\Windows\`), zudem wurde die Logdatei angelegt und der Treiber vollautomatisch im System registriert.

Hinweis: Um die Konfiguration von Türsteher kennenzulernen und die Basiskonfiguration Ihren Bedürfnissen anzupassen, lesen Sie bitte Kapitel 2.

1.2.1.1 Automatische Installation manuell anstoßen

Sie können die automatische Installation des Treibers auch später anstoßen. Im entpackten Installationspaket befindet sich das Programm `Install.exe`

Mit diesem Programm können Sie den Treiber und die mitgelieferten Tools nachträglich installieren, ganz so, als hätten Sie die Installation über das Installationspaket durchgeführt. Bitte installieren Sie Türsteher stets in ein leeres Verzeichnis und vermeiden Sie, Türsteher in Verzeichnisse mit betsehenden Dateien und Programmen zu installieren.

Sie können den Treiber und die Tools jederzeit mit dem Deinstallationsprogramm wieder vom jeweiligen Rechner entfernen. Führen Sie dazu einfach das Programm mit dem Namen `Uninstall.exe` aus,

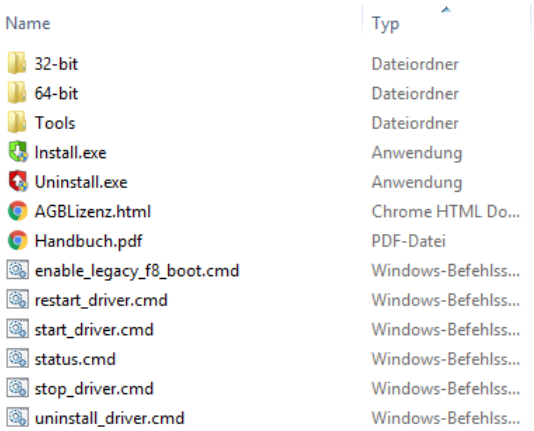
[3] 7-zip kann unter http://www.7-zip.org kostenlos heruntergeladen werden.

welches sich im Installationspaket von Türsteher befindet. Dieses Programm entfernt den Treiber, die Tray-Anwendung, die Log- und Konfigurationsdatei und kann Türsteher auf diese Weise restlos vom System entfernen. Bitte achten Sie darauf, dass im Verzeichnis der `Uninstall.exe` nur Dateien des Türsteher Installationspaketes liegen, um mögliche Datenverluste bei der Deinstallation zu vermeiden.

1.2.2 Manuelle Installation

Um Türsteher manuell zu installieren, entpacken Sie den gesamten Inhalt des Türsteher-Archivs auf den Zielrechner. Folgen Sie den Anweisungen des Installationsprogramms und entpacken die Dateien in ein leeres Verzeichnis Ihrer Wahl.

Sie sollten nun folgende Struktur vorfinden:



In den Ordnern mit den Namen `32-bit` und `64-bit` befinden sich die Treiberdateien für die unterschiedlichen Betriebssystemversionen von Windows (32-bit Version bzw. 64-bit Version) sowie jeweils eine Konfigurationsdatei und eine leere Protokolldatei.

Im Hauptverzeichnis (im Bild oben zu sehen) befindet sich diese Anleitung, Steuerungsskripte und Anwendungen (`./Tools`), die Ihnen bei der Installation helfen und Sie beim späteren Betrieb mit Türsteher unterstützen können.

1.3 Administratoren-Rechte für die Installation

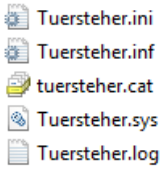
Um Türsteher installieren zu können, benötigen Sie administrativen Zugriff auf den Rechner. Für den späteren Betrieb sind keine Administratoren-Rechte nötig. Türsteher läuft nach erfolgreicher Installation und Start unabhängig vom angemeldeten Anwender vollständig transparent im Hintergrund und hält den Schutz damit immer aufrecht, egal, wer am Rechner letztlich angemeldet ist.

Zum An- bzw. Abschalten und für die Deinstallation des Treibers werden ebenfalls Administratoren-Rechte benötigt. Ein Anwender mit normalen Benutzer-Rechten kann Türsteher also nicht deaktivieren und auch nicht deinstallieren.

2 Installation vorbereiten

Um Türsteher zu installieren, wechseln Sie zuerst in das für Ihr Betriebssystem zutreffende Unterverzeichnis des Türsteher-Paketes, sprich das Verzeichnis, in welches Sie das Archiv von Türsteher entpackt haben. Alle in den folgenden Abschnitten beschriebenen Schritte beziehen sich stets auf das für Ihr Betriebssystem gewählte Verzeichnis aus dem Türsteher-Paket. Um herauszufinden, welches Betriebssystem Sie verwenden, liegt dem Paket von Türsteher ein Hilfsprogramm bei, das Ihnen anzeigt, welches Windows in welcher Version auf Ihrem PC läuft. Starten Sie hierzu das Programm `WindowsArchitecture.exe` (zu finden im Verzeichnis `./Tools`). Dieses zeigt Ihnen die verwendete Architektur und die Version Ihres Windows-Systems auf dem Bildschirm an.

Wenn Sie eine 32-bit Version von Windows nutzen, gehen Sie in das Verzeichnis: `\32-bit\`. Wenn Sie eine 64-bit Version von Windows 8.1 verwenden, gehen Sie in das Verzeichnis: `\64-bit\`:



Bevor Sie den Treiber installieren, müssen Sie die Konfigurationsdatei für Ihren Rechner anpassen. Die Konfiguration von Türsteher befindet sich ausschließlich in der Datei `tuersteher.ini`. Es handelt sich um eine Datei im Unicode-Format, d. h., alle Angaben können auch Buchstaben aus Alphabeten verwenden, die nicht ASCII sind. So sind beispielsweise auch Zeichen aus dem kyrillischen, asiatischen und arabischen Raum möglich und können in der Konfiguration von Türsteher angegeben werden, also Zeichen wie beispielsweise diese:

- `חלו, مَرْحَبَاً, галдёж,`

Türsteher unterscheidet wie für Windows üblich, **nich**t zwischen Groß- und Kleinschreibung. Es ist also egal, ob Regeln für Pfade vollständig in Großbuchstaben, gemischt, komplett in Kleinbuchstaben oder gemischt definiert werden. Für Türsteher sind dies stets dieselben Zeichenketten. Daneben unterstützt Türsteher sog. Platzhaltersymbole (engl. wildcards). Mit diesen können Sie Regeln verallgemeinern. So können Sie beispielsweise definieren, dass sämtliche Dateien mit der Endung `.scr` blo-

ckiert werden sollen oder Dateien mit der Endung `.dll` in einem bestimmten Verzeichnis freigegeben sind. Als Symbole erkennt Türsteher den Stern ***** für eine beliebige Anzahl von Zeichnen und das Fragezeichen **?** für exakt ein beliebiges Zeichen.

Türsteher implementiert ein **striktes Regelsystem**, bei dem stets präzise angegeben werden muss, welche Dateien oder Pfade erlaubt bzw. explizit nicht erlaubt sind. Dateien und Pfade, die nicht der Konfiguration angegeben wurden, werden vom Treiber im späteren Betrieb ohne Rückfrage blockiert. Das kann im schlimmsten Fall dazu führen, dass das Betriebssystem nicht mehr korrekt oder überhaupt nicht mehr startet, daher ist es sehr wichtig, die Liste mit Vorsicht zu konfigurieren. Prüfen Sie die von Ihnen angegebenen Pfade und Dateinamen am besten stets über den Explorer, um sicherzustellen, dass die korrekte Schreibweise gewählt wurde.

Achtung !!!

Lesen Sie die folgenden Schritte aufmerksam durch und halten Sie sich exakt an die Empfehlungen und Beschreibungen, um Abstürze oder ein blockierendes System zu vermeiden.

Öffnen Sie zunächst die Datei `tuersteher.ini` durch einen Doppelklick auf den Dateinamen. Nun sollte sich der Texteditor (normalerweise Notepad) öffnen:

```
[#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:\Fi
rmenskripte\*
*>*
[CMDBLACKLIST]
*explorer.exe>*wscript.exe*
[EOF]
```

Hinweis: In der Demoversion/Vollversion darf die ini-Datei höchstens 5KB/3MB groß sein. Größere Dateien werden vom Treiber vollständig ignoriert. Der Treiber startet und schützt den Computer dann nicht!

In den folgenden Abschnitten werden die verschiedenen Bereiche genauer erläutert, sie sind zur Orientierung in denselben Farben dargestellt. In Ihrer Textverarbeitung wird die Konfiguration stets ohne Farben angezeigt.

Die Konfigurationsdatei gliedert sich in sechs Abschnitte:

1) hier als blauer Bereich: allgemeine Konfiguration,

2) hier als grüner Bereich: die Whitelist,

3) hier als roter Bereich: die Blacklist,

4) hier als violetter Bereich: die Parent-Whitelist,

5) hier als dunkelroter Bereich: die Parent-Blacklist,

6) hier als gelber Bereich: die Kommandozeilen-Whitelist,

7) hier als braun-roter Bereich: die Kommandozeilen-Blacklist und

8) hier als grauer Bereich: das Ende der Konfiguration.

Die folgenden Abschnitte beschreiben, wie Sie ihre `tuersteher.ini` entsprechend Ihrer individuellen Installation konfigurieren können.

2.1.1 Türsteher aktivieren und deaktivieren

Aus der `.ini` relevant:

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

Wenn im allgemeinen Konfigurationsbereich

- `[LETHAL]`

definiert wurde, blockiert Türsteher nicht freigegebene Dateien, d. h., Türsteher ist **scharf geschaltet**. Wenn `[#LETHAL]` angegeben wurde, werden die Dateien nicht blockiert, Türsteher läuft, ist aber wie eine gesicherte Waffe **nicht scharf**.

Bei Erstinstallation sollte für die ersten Starts von Türsteher immer die Option

- `[#LETHAL]`

genutzt werden, um sich mit der Funktionsweise und der Protokolldatei vertraut zu machen. Sobald Sie sicher sind, dass alles nach Ihren Vorstellungen funktioniert, können Sie die Option `[LETHAL]` setzen und Türsteher damit endgültig aktivieren, also scharf schalten. Erst dann blockiert Türsteher nicht zuvor freigegebene ausführbare Programmdateien.

2.1.2 Protokolldatei aktivieren und deaktivieren

Sie können in Türsteher das aktive Protokollieren mit der Zeile

- `[LOGGING]`

aktivieren und mit

- `[#LOGGING]`

deaktivieren. Prinzipiell sollte die Protokollierung stets aktiviert sein, damit Sie nachvollziehen können, welche potenzielle Angriffe Türsteher entdecken konnte. Wenn die Protokollierung aktiviert ist, schreibt Türsteher in das Windows-Systemverzeichnis (üblicherweise `C:\Windows\`) eine Datei mit dem Namen `tuersteher.log`.

Diese Datei liegt im Unicodeformat vor und kann von jedem gängigen Texteditor geöffnet werden (z. B. mit Notepad, Notepad++).

2.1.3 Hashing aktivieren

Wenn neben pfadbasierten Regeln auch Hashwerte von Dateien als Referenz verwendet werden sollen, muss die Zeile

- `[SHA256]`

angegeben werden, andernfalls ist

- `[#SHA256]`

anzugeben.

Sind Hashwerte aktiviert, prüft Türsteher neben dem Pfad auch auf SHA-256 basierte Hashwerte. D. h. anstatt einer Pfadregel kann auch schlicht der Hashwert der freizugebenden (whitelist) oder zu blockierenden (blacklist) Datei angegeben werden. Türsteher erkennt dann die aktuell zur Ausführung anstehende Programmdatei an ihrem individuellen Hashwert.

Bitte beachten Sie, dass die Pflege solcher Hashlisten mitunter zeitaufwendig sein kann und beispielsweise bei Aktualisierungen Ihrer Software (z. B. Updates, Patches) stets neue Listen bzw. aktualisierte Listen zu erstellen sind[4].

2.1.4 Parentchecking aktivieren

Soll Türsteher zusätzlich die Elter-basierte Regelprüfung anwenden, muss die Zeile

- `[PARENTCHECK]`

angegeben werden, möchten Sie die Elter-basierte Regelprüfung nicht nutzen, ist

- `[#PARENTCHECK]`

anzugeben.

Bei der Elter-basierten Regelprüfung gleicht Türsteher vor Ausführung eines neuen Prozesses stets ab, welcher sog. Vater-Prozess den neuen ausführbaren Prozess starten möchte. Ist der jeweilige Vater-Prozess in der whitelist, darf auch der sog. Kind-Prozess starten, andernfalls wird er blockiert.

[4] Wir bieten für die optimale Konfiguration entsprechende Schulungen an, die Sie bei der Ersteinrichtung und optimalen Pflege der Listen unterstützt. Für weitere Details sprechen Sie uns einfach an.

Beachten Sie, dass die Elter-basierte Regelprüfung nicht nur das Starten von Anwendungen (meist `.exe`-Dateien) erkennt, sondern auch das Laden und Starten von Laufzeitbibliotheken sowie Treiberdateien (`.dll`, `.sys`, `.ocx`, `.drv`, `.cpl`). Sie sind daher in der Lage genau zu spezifizieren, welche Laufzeitbibliotheken oder Treiber ein Vater-Prozess laden (und starten), darf und welche nicht.

Das Parentchecking bietet zusätzlichen Schutz und ermöglicht es für Angriffe anfällige Anwendungen wie Web-Browser, Browser-Plugins, Office-Anwendungen und PDF-Anzeigeprogramme stärker abzusichern, als dies mit Bordmittel von Windows möglich wäre. So kann beispielsweise verhindert werden, dass ein Web-Browser eine Kommandozeilen-Shell (`cmd.exe`) oder Shellskripte (`wscript.exe`, `powershell.exe`) ausführen kann. Diese werden von Angreifern häufig nach Ausnutzen einer Sicherheitslücke gestartet, um Schadprogramme aus dem Internet nachzuladen und diese dann dauerhaft auf dem angegriffenen Rechner zu installieren.

Man kann mithilfe des Parentchecking jedoch auch gezielt dafür sorgen, dass beispielsweise bestimmte Laufzeitbibliotheken von Anwendungen nicht geladen werden können. Dies ist insbesondere dann sehr wichtig, wenn in einer Laufzeitbibliothek Sicherheitslücken bestehen, die noch nicht geschlossen sind. Sofern der Vater-Prozess auf die Bibliothek nicht zwingend angewiesen ist, kann diese blockiert werden[5].

2.1.5 Kommandozeilen-Prüfung aktivieren

2.2 Soll Türsteher zusätzlich die Kommandozeilen-Regelprüfung anwenden, muss die Zeile

- `[CMDCHECK]`

angegeben werden, möchten Sie die Kommandozeilen-Regelprüfung nicht nutzen, ist

- `[# CMDCHECK]`

anzugeben.

Mit der Kommandozeilen-Regelprüfung können Sie angeben, mit welcher Kommandozeile bestimmte Anwendungen gestartet werden dürfen und mit welchen nicht. Dies ist besonders zum Blockieren oder Freischalten von sog. Skript-Interpretern hilfreich, denn die vom jeweiligen Interpreter geladenen Skripte werden über Kommandozeilenparameter übergeben. Ruft man beispielsweise eine JS-Datei mit dem Explorer auf, wir der Pfad und Dateiname an den Skript-Interpreter (`wscript.exe`) weitergereicht. Mit der Kommandozeilen-Regelprüfung können Sie bestimmen, welche

[5] Für Weitere Details hierzu können Sie sich gerne mit uns in Verbindung setzen.

Pfade und Dateien an den Skript-Interpreter weitergereicht werden und welche nicht. So ist es häufig sinnvoll, nur die eigenen Skripte zu erlauben und alle anderen strikt zu blockieren. Dies sichert Ihre IT damit stärker ab und hilft insb. VB, JS und JAVA-Anwendungen sicherer auszuführen als bisher.

2.3 Die Whitelist konfigurieren

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

Unterhalb des Eintrags `[WHITELIST]` definieren Sie alle Pfade, von denen aus Programmcode gestartet werden darf. Hier sollten Sie mindestens jene Dateipfade hinterlegen, die für den Betrieb von Windows und den von Ihnen installierten Programmen zwingend notwendig sind, d. h. insbesondere alle vom Betriebssystem Windows benötigten Pfade (oder Dateien).

Ab Windows 7 sind dies in der Regel die folgenden Pfade:

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

Wenn Sie eine 64-bit Version von Windows nutzen, finden Sie zusätzlich noch den Pfad für installierte 32-bit Programme unter:

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

Bitte achten Sie darauf, **jede Pfadangabe** mit dem Symbol `*` zu beenden. Das Sternsymbol dient hier als Platzhalter (whildcard) und gibt sämtliche Dateien und Unterverzeichnisse in diesen Ordnern frei. Alternativ können Sie auch jede Datei einzeln aufführen, dies ist jedoch aufwendig und nur im Hochsicherheitsbereich sinnvoll.

Ggf. hat der Hersteller Ihres Computers noch weitere Verzeichnisse für Treiber angelegt, die sich häufig unterhalb des Hauptlaufwerks befinden (meist unter `C:\`). Bei Rechnern der Hersteller DELL, ACER und ASUS heißen diese Verzeichnisse oft:

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

- `C:\Intel\* oder C:\AMD\*`
- `C:\OEM\*`

Wenn Sie zusätzliche Programme in anderen Verzeichnissen installiert haben, müssen Sie diese ebenfalls in der Whitelist freigeben. Die Einträge könnten dann beispielsweise wie folgt lauten:

- `C:\Mein Programmordner\ProgrammA\*`
- `F:\Share\SuperTool\Tool\*`

Wie eingangs erwähnt, können die Pfadangaben auch Unicode-Zeichen enthalten. D. h., auch Zeichen aus Sprachen, die nicht in ASCII vorhanden sind, beispielsweise:

- `C:\Users\مَـرْحَبَـا\галдёж\*`

Neben Pfadangaben können Sie auch einzelne Programmdateien in die Whitelist eintragen. Hierzu schreiben Sie einfach den kompletten Pfad mit Angabe des Dateinamens und seiner Erweiterung in eine Zeile. So können Sie beispielsweise bestimmte Programmdateien zulassen, ohne den gesamten Pfad und dessen Inhalte zu erlauben. Befinden sich beispielsweise im Ordner unter `F:\Sandbox\` mehrere DLLs und EXE-Dateien, Sie möchten allerdings nur eine bestimmte Anwendung mit dem Namen `TestA.exe` zulassen, so fügen Sie in die Whitelist folgende Regel ein:

- `F:\Sandbox\TestA.exe`

Auf diese Weise lässt Türsteher die Anwendung `TestA.exe` zu, alle anderen Programmdateien im Verzeichnis `F:\Sandbox\` werden weiterhin blockiert.

Wie bereits oben erwähnt, unterstützt Türsteher sog. Wildcards (Platzhaltersymbole). Mit diesen können Sie verallgemeinerte Regeln definieren, wie beispielsweise sämtliche `.exe` Dateien in einem Verzeichnis oder Dateien, die mit `A` beginnen und `.exe` abschließen. Oder Programmdateien, von einem beliebigen Laufwerk, die mit `hallo` beginnen und mit `.exe` abschließen:

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

Das Sternsymbol steht dabei für ein oder mehrere beliebige Zeichen, das Fragezeichen steht für genau ein Zeichen.

2.3.1 Prioritätsregeln

Neben Wildcards kann Türsteher auch sog. Prioritätsregeln verarbeiten. Mit dem Ausrufezeichen **!** können Sie die Priorität einer Regelzeile erhöhen. So ist es z. B. möglich, in der `[WHITELIST]` eine Regel über die `[BLACKLIST]` zu heben. Nehmen wir beispielsweise an, dass folgende Blacklist-Regel gesetzt wurde:

```
...
[BLACKLIST]
C:\Windows\Temp\*
...
```

Aus Sicherheitsgründen sollte man Zugriffe auf temporäre Ordner einschränken, dennoch kann es sein, dass bestimmte Updateprogramme von Microsoft (oder anderen Herstellern), ihre Programmaktualisierungen genau in diesen Ordnern ausführen möchten. Was kann man tun? Nun, mit den Prioritätsregeln kann man diese Anwendungen dann mittels einer Prioritätsregel in der `[WHITELIST]` freigeben. Eine Prioritätsregel überstimmt die Regel aus der `[BLACKLIST]`. Nehmen wir an, dass die Anwendung `AVUpdater.exe` in `C:\Windows\Temp\` ausgeführt werden muss, dann können wir dies nun wie folgt realisieren:

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

Mit dieser Regel überstimmt die Priorität in der `[WHITELIST]` für `C:\Windows\Temp\AVUpdater.exe` dann die Regel `C:\Windows\Temp\*` in der `[BLACKLIST]`.

Prioritätsregeln funktionieren in allen Regel-Bereichen für Türsteher, also in der normalen White- und Blacklist von Türsteher, aber auch für die Elter-basierten sowie für die Kommandozeilen-Regeln.

Bitte beachten: Prioritätsregeln müssen vor Nicht-Prioritätsregeln stehen. Die Auswertung der Regeln arbeitet nach dem Schema, dass sobald die erst Regel greift diese

als dann auch herangezogen wird. Es wird dann nicht weiter nach ggf. ebenfalls gültigen Regeln gesucht. Wenn in einer Whitelist-Sektion die Regel

>

als erste Regel steht, besitzt diese Regel keine Priorität und wird ggf. durch eine Regel aus der Blacklist verboten, selbst dann, wenn es eine Prioritätsregel gibt, diese aber unterhalb der *>* Regel steht. Prioritätsregeln müssen daher immer vor allgemeineren Regeln stehen, damit diese nicht „ausgestochen“ werden.

2.3.2 Verwendung von Hashwerten

Statt Verzeichnissen und Dateinamen kann Türsteher auch SHA-256 Hashwerte von auszuführenden Dateien gegen die whitelist abgleichen. Hierzu geben Sie einfach den jeweiligen SHA256 Hashwert der zu erlaubenden Datei an. Die Angabe des Dateipfades oder Dateinamens ist hier nicht notwendig[6].

Bitte beachten Sie, dass sich nach einem Update der entsprechenden Datei auch deren Hashwert ändert. D. h. Sie müssen dann auch den Hashwert in der Whitelist aktualisieren. Wenn Sie Hashwerte von vielen Dateien verwenden, ist dies keine einfache Aufgabe und sollte stets sorgfältig geplant und durchgeführt werden.

Generell empfehlen wir, die Verwendung von Hashwerten nur für die Verzeichnisse zu nutzen, die potenziell durch Veränderung gefährdet sind. Üblicherweise lassen sich NTFS-Dateisysteme sehr gut vor nicht-intendierter Veränderung und Manipulation durch Benutzerrechte absichern. Wenn die Rechner nicht mit Admin-Rechten genutzt werden, besteht beispielsweise für das Windows-Verzeichnis (`C:\Windows\`) nur eine geringe Gefahr, dass hier Schadprogramme eingefügt oder bestehende Dateien manipuliert werden können. Anders sieht es auf Netzlaufwerken aus, die teilweise über keinen speziellen Berechtigungsschutz verfügen. Hier können Anwendungsprogramme dann potenziell manipuliert (z. B. mit einem Virus infiziert) oder durch Schadprogramme ersetzt werden. Auf solchen Laufwerken sollten dann Hashwerte genutzt werden. Gerne beraten wir unsere Kunden und zeigen hier optimale Prozesse zur Nutzung und Absicherung auf.

[6] Bitte achten Sie darauf die Hash-basierte Prüfung über Setzen der Option `[SHA256]` im Kopfbereich der `.ini` zu aktivieren.

2.4 Die Blacklist konfigurieren

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

Unterhalb des Eintrags `[BLACKLIST]` definieren Sie alle Pfade, von denen aus **kein Programmcode** gestartet werden darf. Standardmäßig blockiert Türsteher alle Pfade, die nicht in der Whitelist angegeben werden. Sie können die Blacklist ggf. auch leer lassen und müssen am Anfang noch nichts eintragen. Sie haben über die Blacklist aber die Möglichkeit, bestimmt Pfade oder Dateien explizit zu blockieren. Diese Option eignet sich hervorragend dafür, bestimmte Programme aus hierarchisch tiefer liegenden Pfaden zu blockieren.

Türsteher unterstützt für die Blacklist die bekannten Wildcards (Platzhaltersymbole). Mit diesen können Sie verallgemeinerte Regeln definieren, wie beispielsweise sämtliche `.exe` Dateien in einem Verzeichnis sollen blockiert werden, oder Dateien, die mit `A` beginnen und `.exe` abschließen sollen blockiert werden:

- `C:\Testverzeichnis\*.exe`
- `C:\Users\*`
- `A*.exe`

Das Sternsymbol steht dabei für ein oder beliebig viele Zeichen, das Fragezeichen steht für genau ein Zeichen.

2.4.1 Beispiel 1

Sie möchten den Windows Taschenrechner (liegt üblicherweise in `C:\Windows\System32\calc.exe`) blockieren, haben in der Whitelist aber den Pfad zu Windows über

- `C:\Windows\*`

freigegeben. Die Whitelist-Regel erlaubt, dass der Taschenrechner in `C:\Windows\System32\` gestartet werden kann. Um dies wirksam zu verhindern, können Sie die Programmdatei zum Taschenrechner explizit in der Blacklist verbieten:

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

Statt einer einzelnen Anwendung können auch ganze Verzeichnisse blockiert werden oder den Hashwert einer zu blockierenden Datei angeben (wenn sie `[SHA256]` aktiviert haben).

2.4.2 Beispiel 2

Angenommen, im Microsoft Browser Internet Explorer wurde ein Sicherheitslücke entdeckt, diese wurde jedoch noch nicht geschlossen und Sie möchten verhindern, dass der Internet Explorer und seine zahlreichen Bibliotheken unbeabsichtigt gestartet werden können und die Lücke ggf. für einen Angriff ausgenutzt werden kann. Sie können nun einfach folgende Regel definieren:

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

Wenn Sie eine 64-bit Version von Microsoft Windows verwenden, fügen Sie noch zusätzlich die folgende Regel ein:

- `C:\Windows\Program Files (x86)\Internet Explorer\*`

Auf diese Weise wird der Internet Explorer und all seine Komponenten durch Türsteher effektiv blockiert. Sobald die Sicherheitslücke geschlossen wurde und Sie den Browser wieder freigeben möchten, können Sie die Regel einfach wieder entfernen.

Statt eines ganzen Verzeichnisses kann es in manchen Fällen auch zweckmäßig sein, nur einzelne Programmdateien (wie z. B. DLLs zu Plug-ins) zu deaktivieren. Insbesondere dann, wenn diese aufgrund einer Sicherheitslücke gefährdet sind.

Häufig ist es so, dass bestimmte Bibliotheken oder Plug-ins für Angriffe verwundbar sind- Cyber-Kriminelle nutzen die Lücken dann aus, um Ihren Rechner mit weiterem Schadcode zu infizieren. Wenn Sie die verwundbaren Plug-ins/Bibliotheken über die Blacklist blockieren, können diese nicht mehr genutzt werden und Angreifer können sie auch nicht für einen Angriff ausnutzen. Nachdem eine Programmaktualisierung für die verwundbaren Komponenten vorliegt, kann diese eingespielt werden und die Regel aus der Blacklist entfernt werden.

2.4.3 Vorsicht bei Verwendung Blacklist-Regeln

Bitte beachten Sie, dass das Deaktivieren von Programmdateien manchmal dazu führt, dass bestimmte Programme nicht mehr korrekt funktionieren. Das Ausschalten bestimmter Dateien (insbesondere von DLLs und Treibern) sollte stets von Experten und mit größter Vorsicht erfolgen. Wir empfehlen ausdrücklich die Stabilität der Systeme über Referenz- und Testsysteme vorab zu verifizieren, bevor Regeln der Blacklist mittels rollout auf die Systeme gespielt werden.

2.4.4 Verwendung von Hashwerten

Statt Verzeichnissen und Dateinamen kann Türsteher auch SHA-256 Hashwerte von auszuführenden Dateien gegen die Blacklist abgleichen. Hierzu geben Sie einfach

den jeweiligen Hashwert der zu blockierenden Datei an. Die Angabe des Dateipfades oder Dateinamens ist hier dann nicht notwendig.

Bitte beachten Sie, dass sich nach einem Update der entsprechenden Datei auch deren Hashwert ändert bzw. für neue Dateien neue Hashwerte eingefügt werden müssen. D. h. Sie müssen die Hashwerte in der Blacklist aktualisieren. Wenn Sie Hashwerte von vielen Dateien verwenden, ist dies keine einfache Aufgabe und muss stets mit größter Sorgfalt durchgeführt werden.

2.4.5 Leise Regeln (Silent Rules)

Türsteher unterstützt sog. leise Regeln. Diese Regeln führen dazu, dass eine Anwendung über die Blacklist blockiert werden, jedoch kein Eintrag im Log angelegt wird. Dies ist dann sinnvoll, wenn Sie Systemkomponenten des Betriebssystems oder installierter Anwendungen blockieren möchten, dies aber systembedingt nicht entsprechend konfigurieren können, sprich das Betriebssystem oder die Anwendung versuchen die blockierte Anwendung oder Bibliothek zu starten und generieren dadurch unerwünschte Einträge in der Logdatei.

Leise Regeln werden stets durch das Dollarzeichen **$** am Anfang der Regelzeile definiert, also beispielsweise

$*notepad.exe

Diese Regel besagt, dass `notepad.exe` blockiert werden soll und gleichzeitig kein Eintrag in der Logdatei generiert wird, wenn der Fall eintritt, sprich notepad.exe tatsächlich von Türsteher blockiert wurde.

Leise Regeln können nur in den Blacklist-Bereichen `[BLACKLIST]`, `[PARENTBLACKLIST]` und `[CMDBLACKLIST]` angewendet werden.

2.5 Die Parent-Whitelist konfigurieren

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

Wurde die Elter-basierte Regelprüfung über die Konfiguration mittels der Zeile

- `[PARENTCHECK]`

aktiviert, **muss** die `[PARENTWHITELIST]` konfiguriert werden.

Unterhalb des Eintrags `[PARENTWHITELIST]` definieren Sie alle Pfade, von denen aus Programmcode gestartet werden darf. Hier wird bestimmt, welcher Vater-Pro-

zess welche Kind-Prozesse starten darf. Also beispielsweise, dass `explorer.exe` andere Prozesse starten darf, wie z. B. `notepad.exe` etc. Die Regeln für die Elter-basierte Prüfung haben dabei folgendes Format:

Pfad/Dateiname Vater`>`*Pfad/Dateiname Kind*

Bitte beachten, dass der Pfad/Dateiname durch das Symbol **>** getrennt ist und zwischen dem Symbol **kein** Leerzeichen stehen darf! Türsteher unterstützt wie schon für die Blacklist und Whitelist Wildcards (Platzhaltersymbole * und ?).

Erlaubte Regeln sehen beispielsweise wie folgt aus:

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

Die erste Regel gibt an, dass alle ausführbaren Programme im Verzeichnis `C:\Windows\` mit all seinen Unterverzeichnissen, Vater-Prozess für einen beliebigen Kind-Prozess sein kann. Die zweite Regel definiert, dass `C:\Windows\explorer.exe` die Kommandozeile starten darf. Die dritte Regel definiert, dass `C:\Windows\explorer.exe` eine definierte Anwendung, nämlich `TestAnwendung.exe`, von einem beliebigen Laufwerk starten darf.

Üblicherweise sollten Prozesse aus den Verzeichnissen

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

andere Prozesse (dies gilt auch für DLLs und SYS etc.) ausführen dürfen. Man kann die Regeln hier allerdings auch granularer gestalten und das System entsprechend weitreichend absichern. Neben Pfadangaben können Sie auch einzelne Programmdateien in die Parentwhitelist eintragen. Hierzu schreiben Sie einfach den kompletten Pfad mit Angabe des Dateinamens und seiner Erweiterung.

Wenn Sie zusätzliche Programme in anderen Verzeichnissen installiert haben, müssen Sie diese ebenfalls in der Parentwhitelist freigeben. Die Einträge könnten dann beispielsweise wie folgt lauten:

- `C:\Mein Programmordner\ProgrammA\*>C:\Windows\*.dll`
- `F:\Share\SuperTool\Tool\*>C:\Windows\*`
- `F:\Share\Tool\*>F:\Share\Tool\*`

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

Beachten Sie, dass Programme vielfach auch Laufzeitbibliotheken aus Windows-Verzeichnissen benötigen, um fehlerfrei zu starten. Daher sollten Sie für jede freigegebene Anwendung mindestens auch eine Vater-Kind-Freigabe auf

- `Pfad/Name_Ihrer_Anwendung>C:\Windows\*.dll`

setzen. Die Parentwhitelist ermöglicht es aber auch genau zu definieren, welche Laufzeitbibliotheken eine Anwendung überhaupt nachladen darf. Auf diese Weise können Sie für bestimmte Anwendungen eine abgeschlossene Menge an DLLs definieren, die durch die Anwendung geladen und gestartet werden darf. So könnten beispielsweise angreifbare Plugin-Bibliotheken ausgeschlossen werden und eine Anwendung bis zum Bereitstehen eines Updates gesichert werden. Sie stellen damit aber auch sicher, dass eine Anwendung nur die Ihnen bekannten Laufzeitbibliotheken laden und nutzen darf, und können so dafür sorgen, dass die Anwendung ausschließlich auf vertrauenswürdige Bibliotheken zurückgreifen kann[7].

Wir empfehlen dringen, bei den ersten Konfigurationsversuchen Türsteher im nicht-lethalen, also `[#LETHAL]`, Modus zu starten und das Verhalten im Log zu verifizieren, bevor das Parentchecking im Echtbetrieb genutzt wird.

Wie bereits an anderer Stelle erwähnt, können die Pfad- und Dateiangaben auch Unicode-Zeichen enthalten, z. B. beispielsweise:

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

2.5.1 Vorsicht bei Verwendung von Parentwhitelist-Regeln

Wenn Sie Parentchecking aktiviert haben und im Bereich `[PARENTWHITELIST]` keine Regeln angeben, besteht die Gefahr, dass Ihr System nicht mehr startet, Programme nicht gestartet werden können, oder abstürzen. Es ist daher sehr wichtig, die Regeln sorgfältig zu definieren. So sollten am Anfang immer die Systemordner über allgemeine Wildcard-Regeln mit dem ***** Symbol freigegeben werden, also mindestens:

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

[7] Es gibt bestimmte Angriffsszenarien, in denen Angreifer dafür sorgen, dass eigentlich vertrauenswürdige Anwendungen eine schädliche Laufzeitbibliothek nachladen und dadurch dann angegriffen werden bzw. weitere Angriffe gestartet werden können. Für weitere Informationen und Beratung können Sie sich gerne mit uns in Verbindung setzen.

Auf einer 64-bit Version von Windows zusätzlich noch

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

Sofern Sie Treiber und spezielle Anwendungen Ihres PC-Herstellers in weiteren Unterordnern von `C:\` hinterlegt haben, sollten diese ebenfalls freigegeben werden. Üblich sind hier beispielsweise:

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

Bitte sorgen Sie aus Sicherheitsgründen auch dafür, dass diese Ordner mittels Zugriffsrechten vor Schreibzugriffen geschützt sind.

Es wird empfohlen bei den ersten Konfigurationsversuchen Türsteher stets im nicht-lethalen, also `[#LETHAL]`-Modus zu starten und das Verhalten im Log zu verifizieren.

2.6 Die Parent-Blacklist konfigurieren

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

Wurde die Elter-basierte Regelprüfung über die Konfiguration mittels der Zeile

- `[PARENTCHECK]`

aktiviert, können Sie auch Regeln für die Parentblacklist spezifizieren. Wenn Sie keine Regeln für verbotene Vater-Anwendungen hinterlegen möchten, können Sie den Bereich nach `[PARENTBLACKLIST]` auch leer lassen und definieren direkt `[CMDWHITELIST]`.

Unterhalb des Eintrags `[PARENTBLACKLIST]` definieren Sie alle Programme, die bestimmte Anwendungen nicht starten dürfen. Sie können beispielsweise festlegen, dass der Internet Explorer keine `cmd.exe` Shell starten darf, oder dass Chrome keine Skriptinterpreter wie `powershell.exe` und `wscript.exe` starten kann.

Die Regeln für die Elter-basierte Prüfung haben dabei folgendes Format:

Pfad/Dateiname Vater>*Pfad/Dateiname Kind*

Bitte beachten, dass der Pfad/Dateiname durch das Symbol **>** getrennt ist und zwischen dem Symbol **kein** Leerzeichen stehen darf! Türsteher unterstützt wie schon für die Blacklist und Whitelist die Platzhaltersymbole: * und ?

Regeln für die Parentblacklist könnten wie folgt aussehen:

- `*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\*`

Die erste Regel gibt an, dass der Internet Explorer keine `cmd.exe`-Shell starten darf. Die zweite Regel gibt an, dass der Internet Explorer den Interpreter `powershell.exe` nicht ausführen darf. Die dritte Regel gibt an, dass der Chrome-Browser die Anwendung `bitsadmin.exe` nicht starten darf. Die vierte Regel hindert Firefox daran, den Kommandozeileninterpreter (cmd.exe) zu starten. Die letzten Regeln verhindern, dass das sehr häufig für Angriffe ausgenutzte Adobe Flash Plugin sicherheitskritische Systemtools starten kann[8].

Beachten Sie, dass Programme vielfach auch diverse Laufzeitbibliotheken laden. Teilweise können Angreifer durch ausgefeilte Tricks das Nachladen bestimmter DLLs ausnutzen, um schädlichen Code auf dem Rechner zu starten. Über die die Parentblacklist kann beispielsweise verhindert werden, dass Anwendungen DLLs aus Benutzerverzeichnissen laden können:

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

Wie bereits weiter oben erwähnt, können die Pfad- und Dateiangaben auch Unicode-Zeichen enthalten:

- `C:\مَرْحَبَا\галдёж\x.exe>C:\مَرْحَبَا\*.dll`
- `C:\مَرْحَبَا\галдёж\x.exe>C:\مَرْحَبَا\*.dll`

Wie Sie sehen, sind damit auch umfangreiche Regeln mit Unicode-Zeichen kein Problem und leicht umsetzbar.

[8] Die Beispiele zeigen Anwendungen, die in der Praxis sehr häufig bei Angriffen auf Browser ausgenutzt werden. Neben diesen gibt es weitere, für weitere Informationen können Sie sich gerne mit unseren Experten in Verbindung setzen.

2.7 Die Kommandozeilen-Whitelist konfigurieren

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

Wurde die Kommandozeilen-Regelprüfung über die Konfiguration mittels der Zeile

- `[CMDCHECK]`

aktiviert, **muss** die `[CMDWHITELIST]` konfiguriert werden.

Im Bereich der Kommandozeilen-Whitelist konfigurieren Sie die Kommandozeilen, die Sie freigeben möchten. Im gezeigten Beispiel wurde eine Prioritätsregel definiert, die `wscript.exe` nur Skripte aus dem Verzeichnis `C:\Firmenscripte\*` öffnen lässt und auch nur über den Windows Explorer. So darf beispielsweise Microsoft Word oder der Internet Explorer wscript.exe nicht ausführen, auch nicht, um Skripte aus dem Verzeichnis `C:\Firmenscripte\*` zu starten. Der Eintrag vor >, also

```
!*explorer.exe
```

definiert den Eltern-Prozess. Der Eintrag nach >, also

```
*wscript.exe*C:\Firmenskripte\*
```

definiert den freigegebenen Kommandozeilen-Parameter.

Die Regelzeile besagt demnach dass der Explorer den Windows Skriptinghost mit dem definierten Parameter ausführen darf. Die Prioritätsregel wird in diesem Beispiel benötigt, da in der Kommandozeilen-Blacklist eine strenge Regel zum Verbieten aller Skripte des `wscript.exe`-Interpreters gesetzt wurde (siehe unten)[9].

2.8 Die Kommandozeilen-Blacklist konfigurieren

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

Wurde die Kommandozeilen-Regelprüfung über die Konfiguration mittels der Zeile

- `[CMDCHECK]`

aktiviert, **muss** die `[CMDBLACKLIST]` konfiguriert werden.

Im Bereich der Kommandozeilen-Blacklist konfigurieren Sie die Kommandozeilen, die Sie **nicht** freigeben (also blockieren) möchten. Im gezeigten Beispiel wurde eine Regel definiert, die `wscript.exe` für sämtliche Aufrufe blockiert.

[9] Kommandozeilen-Regeln können teilweise sehr komplex werden. Wir unterstützen und beraten unsere Kunden gerne bei der Erstellung solcher Regeln.

Der Eintrag vor >, also

```
*
```

definiert jeden möglichen Prozess. Der Eintrag nach >, also

```
*wscript.exe*
```

definiert den Kommandozeilen-Parameter, hier alle Kombinationen von `wscript.exe` und Kommandos an diesen Prozess. Mit dieser Regel kann sichergestellt werden, dass kein Elternprozess `wscript.exe` mit irgend einem Kommandozeilenparameter starten kann.

Das gezeigte Beispiel kann insbesondere vor Kryptolockern schützen, die in letzter Zeit vermehrt als JS-Skripte per E-Mail versandt werden. Da im Bereich KMUs häufig Skripte ausgeführt werden müssen, kann der Skriptinghost nicht einfach auf die Blacklist gesetzt werden. Hier können Kommandozeilen-White- und Blacklists für mehr Sicherheit sorgen, indem genau definiert wird, welche Skriptdateien überhaupt ausgeführt werden können. So sollten insbesondere Skripte aus temporären Ordern oder von externen Datenträgern standardmäßig auf die Blacklist gesetzt werden, da dies ein Einfallstor für Schadcode darstellt.

2.9 Ende der Konfiguration

Die Konfigurationsdatei muss stets mit folgender Zeile beendet werden:

```
[EOF]
```

Hinweis: Bitte beachten Sie, dass Türsteher die Konfigurationsdatei nicht akzeptiert und den Treiber **nicht lädt**, wenn diese nicht mit `[EOF]` abgeschlossen wird.

2.10 Türsteher für den ersten Start vorbereiten

Wenn Sie die oben angegebenen Schritte durchgeführt haben und die Regeln Ihrer Konfiguration entsprechend angepasst haben, können Sie Türsteher für den ersten Start vorbereiten. Um Ihre Konfiguration zu verifizieren, sollten Sie Türsteher am Anfang **nicht** im scharfen Modus starten. Setzen Sie vor dem allerersten Start von Türsteher die Option wie folgt:

- `[#LETHAL]`

Auf diese Weise startet Türsteher nicht im scharfen Modus und Sie können über die Protokolldatei `tuersteher.log` prüfen, ob Ihr Regelwerk wie gewünscht funktioniert. Wenn alles korrekt konfiguriert wurde, sollte Windows starten, ohne dass Meldungen in die Logdatei geschrieben werden. Gleiches gilt für den Start der von Ihnen freigegebenen Anwendungen.

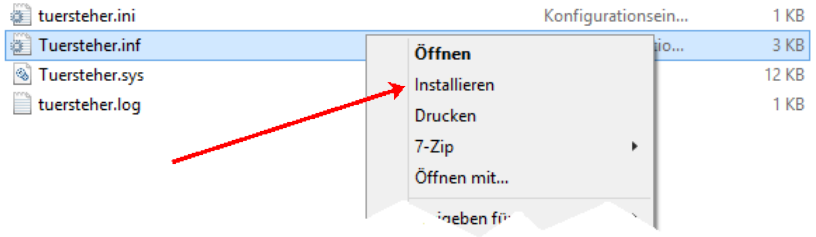
Kopieren Sie die von Ihnen angepasste Regeldatei `tuersteher.ini` in Ihren Widows Sytemordner (meist `C:\Windows\`). Prüfen Sie nochmals, dass in der Datei alle Regeln entsprechend Ihrer vorliegenden Systemkonfiguration stimmen und, dass die Option `[#LETHAL]` gesetzt ist. Wenn nicht bereits geschehen, kopieren Sie die Datei `tuersteher.log` ebenfalls in Ihren Winows Systemordner (meist `C:\Windows\`).

Prüfen Sie nun nochmals, dass im Windows Systemordner (meist `C:\Windows\`) die beiden Dateien

- `tuersteher.ini`
- `tuersteher.log`

vorhanden sind, und in `tuersteher.ini` Ihre individuellen Regeln stehen!

Sie können den Treiber nun durch Rechts-Klick und Auswahl der Option „Installieren...“ auf die Datei `tuersteher.inf` installieren:



Wechseln Sie nun in das Hauptverzeichnis von Türsteher und führen das Skript `start_driver.cmd` zum Starten von Türsteher als Administrator aus (rechts Klick auf die Datei und „Ausführen als Administrator“ auswählen). Nun sollte Türsteher

im Hintergrund aktiv sein. Sie können den Rechner nun einige Male hochfahren, Anwendungen starten und deren Funktionalität testen.

Um Türsteher zu deaktivieren, führen Sie das Skript `stop_driver` als Administrator aus.

Nach Aktualisieren der Regeln in `tuersteher.ini`, müssen Sie diese wieder in das Windows Systemverzeichnis (meist `C:\Windows\`) kopieren **und** den Treiber neu starten. Verwenden Sie dazu das Skript `restart_driver.cmd` und führen es als Administrator aus.

2.11 Funktionalität testen

Öffnen Sie die Protokolldatei `tuersteher.log` und prüfen, ob nicht Anwendungen protokolliert werden, die Sie eigentlich freigegeben haben. Da üblicherweise das Verzeichnis `C:\Windows\` freigegeben wurde, sollte sich der Datei-Explorer, der Taschenrechner, Notepad oder MS-Paint starten lassen, ohne dass sich dabei Einträge zu diesen Programmen in der Protokolldatei finden lassen. Falls doch, prüfen Sie nochmals Ihre Konfiguration in `tuersteher.ini`.

Zu Testzwecken können Sie eine Programmdatei wie `notepad.exe` auf die Blacklist setzen, den Treiber über das Skript `restart_driver.cmd` neu starten und prüfen, ob nach Start von Notepad ein entsprechender Eintrag in der Protokolldatei zu finden ist. Denken Sie daran, nach dem Aktualisieren der Regeln in `tuersteher.ini` müssen Sie diese wieder in das Windows Systemverzeichnis (meist `C:\Windows\`) kopieren **und** den Treiber neu starten. Verwenden Sie dazu das Skript `restart_driver.cmd` und führen es als Administrator aus.

Als weiteren Test können Sie bei aktiviertem Parentchecking beispielsweise dem Explorer (`explorer.exe`) verbieten, dass dieser den Taschenrechner (`calc.exe`) starten kann. Hierzu fügen Sie unter `[PARENTBLACKLIST]` die folgende Zeile ein:

- `*explorer.exe>*calc.exe`

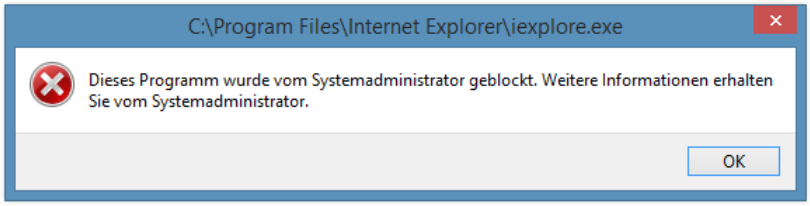
Wenn Sie den Treiber nun neu starten und über den Explorer versuchen, den Taschenrechner zu starten, sollte Türsteher diesen Versuch erkennen und melden. Versuchen Sie nun eine `cmd.exe`-Shell zu öffnen und von dort aus `calc.exe` zu starten. Dies sollte funktionieren und der Taschenrechner gestartet werden. Als Übung können Sie nun versuchen, das Starten von `calc.exe` auch über eine `cmd.exe`-Shell zu blockieren. Die Lösung finden Sie in der folgenden Fußnote[10].

[10] Die Lösung lautet: `*cmd.exe>*calc.exe`

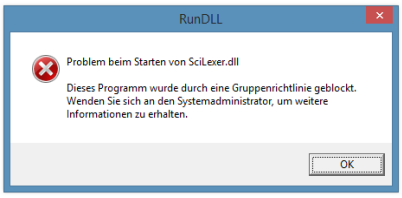
Türsteher arbeitet strikt nach den von Ihnen vorgegebenen Regeln. Aus einem Verzeichnis, das nicht in der Whitelist steht, kann auch keine Programmdatei gestartet werden. Gleiches gilt für Treiberdateien und sog. Dynamic-Link-Libraries (DLLs).

Sobald Sie mit den gesetzten Regeln zufrieden sind, können Sie Türsteher scharf schalten. Ändern Sie dazu [`#LETHAL`] in [`LETHAL`]. Nach Ändern der entsprechenden Zeile in `tuersteher.ini` müssen Sie diese wieder in das Windows Systemverzeichnis (meist `C:\Windows\`) kopieren **und** den Treiber neu starten. Verwenden Sie dazu das Skript `restart_driver.cmd` und führen es als Administrator aus. Nun sollte Türsteher im scharfen Modus aktiv sein, Programmdateien außerhalb der zugelassenen Pfade können nicht mehr gestartet werden, da sie von Türsteher blockiert werden.

Wenn Sie beispielsweise die oben beispielhaft angegebene Blacklist-Regel für den Internet Explorer zu Testzwecken aktivieren, sollte beim versuchten Start des Internet Explorers die folgende Meldung angezeigt werden:



Wenn eine Anwendung versucht, eine DLL aus einem nicht freigegebenen Ordner auszuführen, sehen Sie z. B. die folgende Meldung:



2.12 Wichtiger Hinweis für neue und aktualisierte Regeln

Bitte beachten Sie, dass der Treiber von Türsteher nach Änderung der `.ini`-Datei stets neu gestartet werden muss, damit die Änderungen aktiv werden!

3 Hilfsprogramme (Tools)

Türsteher läuft vollständig unabhängig im Kern des Betriebssystems und benötigt keine Anwendung zur Steuerung. Dennoch werden zu Demonstrationszwecken und für die einfachere Konfiguration und für den Betrieb von Türsteher zwei Hilfsprogramme geliefert, welche die Arbeit erleichtern können. Diese werden in den folgenden Abschnitten kurz vorgestellt. Die Anwendungen sind optional und müssen weder installiert noch benutzt werden, Türsteher funktioniert auch ohne diese Tools.

3.1 Anwendung für den Tray-Bereich

Mit Türsteher Tray (`TuersteherTray.exe`) wird ein Programm geliefert, das im Tray-Bereich der Windows Taskleiste (links neben der Uhr) ein T-Symbol in unterschiedlichen Farben anzeigt:

Wurde die Tray-Anwendung gestartet, prüft diese, ob sich die vom Türsteher angelegte Logdatei verändert hat. Ist Türsteher aktiv und es wurden keine Gefahren abgewehrt, ist das T-Symbol stets grün:

Hat Türsteher eine ausführbare Datei blockiert, färbt sich das T-Symbol rot:

Zudem zeigt die Anwendung in einer Sprechblase an, welche Datei(en) von Türsteher aktuell blockiert, wurde(n) und schreibt diese Information auch in das Windows Event-Log.

Wurde Türsteher in den Installations-Modus geschaltet, färbt sich das T-Symbol gelb:

Bitte beachten Sie: Im Installations-Modus schützt Türsteher nicht, es können sämtliche ausführbaren Programme ausgeführt werden. Die Tray-Anwendung zeigt dann jede halbe Stunde eine Warnmeldung an. Diese weist Sie darauf hin, dass Türsteher noch im Installationsmodus ist.

Ist Türsteher nicht aktiv, so ist das T-Symbol grau:

Die Tray-Anwendung zeigt dann jede halbe Stunde eine Warnmeldung an. Diese weist Sie darauf hin, dass Türsteher nicht aktiv ist und kein Schutz besteht.

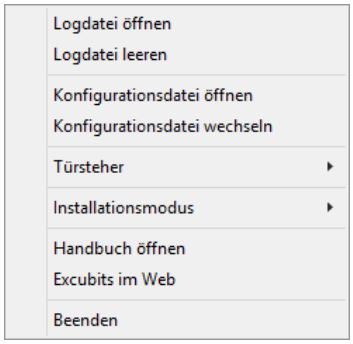
Diese Anwendung demonstriert, wie flexibel unser Produkt ist, und dass keine Verbindung zum Kern des Betriebssystems notwendig ist. Sie können Türsteher mit oder ohne diese Anwendung verwenden.

Wenn Sie keine Warnmeldungen der Tray-Anwendung angezeigt bekommen möchten, können Sie die Anwendung mit der Kommandozeilenoption `nopopups` starten, dann werden alle sog. *Tool-tips* unterdrückt.

Daneben besteht die Möglichkeit eigene Anwendungen für Türsteher zu entwickeln (oder von uns entwickeln zu lassen), die Ihren individuellen Bedürfnissen gerecht werden. So kann zur Alarmierung auch eine E-Mail an den Administrator oder IT-Sicherheitsbeauftragten gesendet werden, SMS-Meldungen sind ebenfalls machbar. Neben dem Windows-eigenen EventLog können auch andere zentrale Logging-Dienste wie z. B. SNMP Traps versandt werden[11].

3.1.1 Steuerung von Türsteher

Durch einen Klick auf das T-Symbol kann ein Menü geöffnet werden:



Dieses bietet die Möglichkeit, die Logdatei zu öffnen oder zu leeren, die Datei mit den Regeln zu öffnen oder zu tauschen, den Treiber zu starten oder zu stoppen sowie die Tray-Anwendung zu beenden. Zudem können Sie auch jederzeit dieses Handbuch öffnen und unsere Web-Seite mit Ihrem Browser öffnen.

Hinweis: Um den Treiber zu starten oder zu stoppen, werden stets Administrator-Rechte benötigt. Ein Anwender mit normalen Rechten kann diese Funktionen nicht ausführen.

[11] Unsere Experten helfen Ihnen gerne weiter und beraten Sie. Auf Wunsch stellen wir Ihnen für Ihre Eigenentwicklung auch gerne Informationen zum Systemdesign von Türsteher bereit.

4 Technische Details

Türsteher implementiert technisch gesehen einen sog. Minifilter (Treiber), der vollständig autark im Kern des Betriebssystems läuft. Auf diese Weise arbeitet Türsteher an zentraler Stelle, sozusagen im Herzen des Betriebssystems. Über definierte Regeln bestimmen Sie als Anwender, wann Türsteher aktiv wird und auszuführenden Programmcode blockiert. Da Türsteher im Kern des Betriebssystems arbeitet, kann er sehr früh eingreifen und von typischen Schadprogrammen nicht ohne Weiteres umgangen werden. Ein Schadprogramm müsste erst selbst in den Kern von Windows gelangen oder Türsteher mittels System- oder Administrator-Rechten deaktivieren. Beides ist unter normalen Bedingungen nicht ohne weiteres möglich und erfordert vom Angreifer sehr viel Fachwissen und/oder schwere Sicherheitslücken im Betriebssystem.

Türsteher erkennt sämtliche Programme, die mit dem Betriebssystemlader in den Arbeitsspeicher geladen werden sollen, dies sind die bekannten `EXE`-Dateien, aber auch Bildschirmschoner (`SRC`), Laufzeitbibliotheken (`DLLs`), Treiber (`SYS`, `DRV`) oder Plug-ins (`OCX`, `DLL`). Dabei spielt die Dateiendung aber letztlich keine Rolle, selbst wenn sich eine ausführbare Programmdatei beispielsweise mit der Dateiendung `.jpg` als Bild tarnt, erkennt Türsteher die ausführbare Programmdatei „dahinter“ und kann diese blockieren. Türsteher verhindert so, dass versehentliche angeklickte bösartige Programmdateien von USB-Sticks, externen Festplatten, Netzlaufwerken, CD-/DVD-ROMs oder E-Mail-Anhängen ausgeführt werden können. Schadprogramme, die durch Sicherheitslücken auf Ihren PC gelangen können, werden von Türsteher ebenso blockiert, wie unbeabsichtigt heruntergeladene Programme aus dem Internet. Das von Türsteher implementierte Whitelisting-Schutzsystem schützt Ihren PC damit proaktiv vor Viren, Würmern, Ransomware, Crypto-Lockern u. v. w.

Türsteher arbeitet autark und ist nicht von Entscheidungen des Anwenders abhängig. Türsteher fragt den Anwender niemals nach einer Entscheidung, sondern entscheidet selbst. Anders als bei vielen anderen Sicherheitsprodukten (insbesondere Anti-Viren-Scannern, Desktop-Firewalls) soll unserer Meinung nach der Anwender nicht über die Sicherheit des Computers entscheiden, sondern Türsteher. In der Folge entscheidet Türsteher immer richtig und blockiert Programmdateien.

Türsteher benötigt zudem keine laufenden Updates oder neueste Viren-Definitionen, um den Schutz aufrechtzuerhalten, da er einfach alle unbekannte, versehentlich oder geheim installierte Software noch vor dem Start blockiert.

Der Treiber von Türsteher startet außerdem zu einem sehr frühen Zeitpunkt des Bootvorgangs und ist damit in der Lage Ihr System bereits während des Hochfahrens des Betriebssystems zu schützen. Mit Türsteher kann man theoretisch sogar die ers-

ten Treiber, die das Betriebssystem lädt, blockieren[12]. Sie können auf diese Weise auch genau bestimmen, mit welchen Treibern Ihr Windows starten darf und auf diese Weise noch mehr Vertrauen in den Start des Betriebssystems legen.

4.1 Ihre Vorteile im Überblick

☑ Schutz vor unbekannter Software.

☑ Schutz vor noch nicht geschlossenen Sicherheitslücken.

☑ Schutz vor Viren, Würmern, Spyware, Ransomware, uvw.

☑ Absicherung von für Angriffe anfälliger Anwendungen.

☑ Ressourcenschonend bei gleichzeitig hoher Performance.

☑ Läuft auch auf virtuellen Maschinen.

☑ Kann Windows-basierte Cloud-Server sichern und härten.

☑ Offline voll funktionsfähig.

☑ Keine Updates notwendig.

☑ Läuft unter allen gängigen Windows-Versionen (inkl. Windows-Server).

☑ 100% Software aus Deutschland.

☑ Keine Werbung und keine Spyware.

☑ Kein lästiger Registrierungszwang.

☑ Kein Informationsrückfluss an den Hersteller Excubits.

[12] Nicht, dass man dies tun sollte, weil das Betriebssystem sonst abstürzt, aber es ist möglich und zeigt, wie früh und tief unser System im Kern von Windows seinen Dienst verrichtet.

5 Generelle Empfehlungen

Türsteher ist kein Antivirusprogramm und kann Dateien auch nicht von schädlichen Inhalten befreien, sondern nur davor schützen, diese zu starten. Wir empfehlen daher, Türsteher stets in Kombination mit einem Antivirusprogramm und einer Firewall zu nutzen.

Sie sollten zudem stets sämtliche Aktualisierungen (sowie Service Packs) für Ihr Betriebssystem und alle installierten Programme einspielen. Dies gilt insbesondere für alle mit dem Internet genutzten Anwendungen wie Browser, Plug-ins (z. B. Flash, PDF-Plug-ins, Java, .NET oder Silverlight) etc.

Wir empfehlen zudem, den Rechner nicht dauerhaft mit Administrator-Benutzerrechten zu nutzen, sondern für den alltäglichen Betrieb ein Benutzerkonto mit Standard-Benutzerrechten bzw. eingeschränkten Benutzerrechten anzulegen und mit diesem Benutzerkonto zu arbeiten. Nach Möglichkeit sollten Sie einen Browser verwenden, der über Sandbox-Technologie (z. B. Google Chrome, aktuelle Version des Internet Explorers) verfügt und die Gefahr ausgenutzter Sicherheitslücken damit weiter reduziert.

Für einen maximalen Schutz in Kombination mit Türsteher, Antivirus und Firewall empfehlen wir zudem Microsofts Enhanced Mitigation Experience Toolkit[13] (EMET) zu verwenden. EMET ist ein großartiges Tool, mit dem Sie zusammen mit Türsteher ein extrem hohes Sicherheitsniveau erreichen können.

Wenn Sie mehr über Angriffsmethoden, Begrifflichkeiten und Definitionen im Bereich IT-Sicherheit wissen möchten, empfehlen wir unsere Online-Glossar unter

- https://excubits.com/content/de/glossar.html

zu besuchen.

5.1 Empfehlungen für die [BLACKLIST]

Wir empfehlen die folgenden Systemanwendungen auf die Blacklist zu setzen, wenn diese nicht zwingend für den täglichen Betrieb verwendet werden müssen. Die hier aufgelisteten Anwendungen und Pfade werden häufig dazu genutzt, Schadcode auf Rechnern zu installieren und stellen damit ein potenzielles Risiko dar. Für die Liste besteht kein Anspruch auf Vollständigkeit, Nutzung auf eigene Gefahr. Bei Rückfragen stehen wir unseren Kunden gerne zur Verfügung.

[13] Weitere Informationen zu finden unter http://www.microsoft.com/emet.

```
?:\$Recycle.Bin\*
C:\Windows\ADFS\*
C:\Windows\Fonts\*
C:\Windows\Minidump\*
C:\Windows\Offline Web Pages\*
C:\Windows\tracing\*
C:\Windows\Tasks\*
C:\Users\Public\*
*\AppData\Local\Temp\*.scr
*\AppData\Local\Temp\*.com
*\AppData\Local\Temp\*.bat
*\AppData\Local\Temp\*.sys
*\AppData\Roaming\*.exe
*\AppData\Roaming\*.scr
*\AppData\Roaming\*.com
*\AppData\Roaming\*.bat
*\AppData\Roaming\*.sys
*regsvr32.exe
*InstallUtil*
*Regsvcs*
*RegAsm*
*InstallUtil.exe
*IEExec.exe
*DFsvc.exe
*PresentationHost.exe
*reg.exe
*vssadmin.exe
*aspnet_compiler.exe
*csc.exe
*ilasm.exe
*jsc.exe
*MSBuild.exe
*vbc.exe
*script.exe
*iexplore.exe
*journal.exe
*bitsadmin*
*iexpress.exe
*mshta.exe
*systemreset.exe
*bcdedit.exe
*mstsc.exe
*powershell.exe
*powershell_ise.exe
*hh.exe
*set.exe
*setx.exe
*\at.exe
*mrsa.exe
```

```
*bcdedit.exe
*bcdboot.exe
*bootcfg.exe
*bootim.exe
*bootsect.exe
*ByteCodeGenerator.exe
*debug.exe
*diskpart.exe
*regini.exe
*regsvr32.exe
*RunLegacyCPLElevated.exe
*UserAccountControlSettings.exe
```

6 FAQ

6.1 Vor Ausführung welcher (schädlichen) Programme schützt Türsteher?

Türsteher arbeitet in der aktuellen Fassung Verzeichnispfad-, Hash- und Elternprozess-basiert, d.h. der Anwender gibt in einer Whitelist an, welche bzw. 'über welche' Programmdateien (`.exe`, `.dll`, `.sys`, `.ocx`, `.scr`, `*.cpl`) gestartet werden dürfen. Zusätzlich kann der Anwender auch explizit Programmdateien verbieten. Das bedeutet, dass alle Dateien, die unbekannt oder unsicher sind, nicht gestartet werden können. Durch das Blacklist-Verfahren können unsichere oder unerwünschte Programme blockiert werden. Ist zum Beispiel der Windows-Browser Internet Explorer erlaubt, jedoch durch eine Sicherheitslücke in einem Plug-in oder seiner Bibliotheken angreifbar, kann dieses einzelne Plug-in oder die Bibliothek blockiert werden, sodass die Sicherheitslücke dann nicht ausgenutzt werden kann.

6.2 Wieso tauchen im Log für freigegebene Pfade in 8.3-Schreibweise angegebene Pfade auf?

Leider verwenden manche Anwendungen die alte 8.3-Schreibweise für Pfade und Dateien (z.B. manche Versionen von MS Office). Da Türsteher komplett im Kernel läuft, haben wir nicht die Möglichkeit, 8.3-Pfade mittels geeigneter APIs in ihre ausgeschriebene Form zu bringen. In solchen Fällen muss man in Türsteher dann auch die 8.3-Schreibweise in die Black- oder Whitelist setzen.

6.3 Unter Windows 7 kann der Treiber wegen einer fehlerhaften Signatur nicht installiert werden. Wieso?

In älteren Versionen von Windows (u.a. Vista und 7) wurde die Signaturprüfung von Treibern von Microsoft fehlerhaft implementiert. Sie sollten sämtliche Updates und Service Packs installieren, um diesen Fehler zu beheben, dann kann auch die Signatur korrekt verifiziert werden. Wenn dies nicht möglich ist, sollten Sie den Microsoft Windows-Patch KB3033929 installieren, widrigenfalls kann Türsteher auf diesen Systemen nicht genutzt werden.

6.4 Welche Dateiendungen erkennt und prüft Türsteher?

Keine! Türsteher prüft ausführbare Dateien nicht nach ihrer Dateiendung, sondern danach, ob eine Datei ausführbar in den Arbeitsspeicher geladen werden soll. D. h. Türsteher erkennt damit auch bewusst über den Dateinamen und Endung verschleierte Aufrufe von ausführbarem Programmcode. So kann ein Angreifer beispielsweise die win32-API-Funktion LoadLibrary() auch mit dem Dateinamen `Bild.jpg` aufrufen. Für den arglosen Anwender sieht die Datei wie ein Bild aus, in Wirklichkeit ist

es allerdings eine Programmdatei, die dann auch tatsächlich auf dem Rechner gestartet wird. Für Türsteher spielt das aber keine Rolle, er erkennt und blockiert auch solche Aufrufe und schützt Ihren Rechner!

6.5 Sind Verzeichnispfad-basierte Regeln nicht unsicher?

Unsere Tests zeigen, dass Verzeichnispfad-basierte Regeln unter NTFS einen guten Kompromiss zwischen Komfort und effektivem Schutz darstellen. Wenn Sie Ihre IT-Systeme mittels Zugriffsschutz absichern, stellen Verzeichnispfad-basierte Regeln sogar eine komfortable Möglichkeit dar, Regeln für die Whitelist zu definieren.

6.6 Wieso sollte man hash-Regeln verwenden?

Wir empfehlen Hashwerten nur für die Verzeichnisse zu nutzen, die leicht manipuliert werden können. Üblicherweise lassen sich NTFS-Dateisysteme sehr gut vor unerwünschter Veränderung und Manipulation durch Benutzerrechte absichern. Wenn die Rechner nicht mit Admin-Rechten genutzt werden, besteht beispielsweise für das Windows-Verzeichnis (`C:\Windows\`) nur eine geringe Gefahr, dass hier Schadprogramme eingefügt oder bestehende Dateien manipuliert werden können. Anders sieht es auf Netzlaufwerken aus, die teilweise über keinen speziellen Berechtigungsschutz verfügen. Hier können Anwendungsprogramme dann potenziell manipuliert (z. B. mit einem Virus infiziert) oder durch Schadprogramme ersetzt werden. Auf solchen Laufwerken sollten dann Hashwerte genutzt werden. Gerne beraten wir unsere Kunden und zeigen hier optimale Prozesse zur Nutzung und Absicherung auf.

6.7 Kann Türsteher vor Angriffen über Rechteausweitung schützen?

Das kommt auf die jeweilige Sicherheitslücke zur Rechteausweitung an. Im Fall der von Google im Januar 2015 veröffentlichten Sicherheitslücke in Windows 8.1 konnte Türsteher diese Lücke erfolgreich durch Blockieren der Treiberdatei `ahcache.sys` schließen, bis ein entsprechendes Update zur Verfügung stand. Pauschal kann dies bei Sicherheitslücken dieser Art nicht gesagt werden. Setzen Sie sich im Einzelfall bitte mit uns in Verbindung. Wir unterstützen unsere Kunden in solchen Fällen gerne.

6.8 Schützt Türsteher 100%? Kann ich mein Anti-Virus-Programm und die Firewall deaktivieren?

Nein, kein Schutzsystem der Welt kann einen Windows-PC zu 100 Prozent vor Attacken schützen. Türsteher erhöht den Schutz erheblich und mindert die Gefahr für erfolgreich auf dem Rechner durchgeführte Attacken, sog. *attack surface mitigation*. Türsteher sollte nicht als einzige Sicherheitslösung, sondern im Sinne eines Schichtenmodells (oder Zwiebelprinzip) zusammen mit einem Anti-Virus-Programm und einer

Firewall genutzt werden. Nur so spielen alle Schutzsysteme Ihre Stärken aus und werden durch das jeweilig andere System optimal ergänzt.

6.9 Läuft Türsteher auch auf Windows Server Betriebssystemen?

Ja, Türsteher läuft auch unter Windows Server (auch Core Edition) und kann so zu einem höheren Schutz der Serverinfrastruktur beitragen.

6.10 Läuft Türsteher auch unter virtuellen Maschinen (VMs)?

Ja, Türsteher lässt sich auch auf gängigen Virtualisierungslösungen von VM-Ware oder Virtual-Box nutzen. Türsteher kann dabei nicht nur die virtualisierten Windows-Maschinen schützen, sondern insbesondere auch Windows-basierte VM-Hosts. Daher eignet sich Türsteher auch hervorragend zur Absicherung von virtualisierten Windows-basierten (Cloud-)Infrastrukturen.

Unsere Experten beraten Sie gerne zu Cloud-basierter Absicherung mit Türsteher und unterstützen Sie im Rahmen eines Beratungsauftrags auch gerne bei deren Umsetzung.

6.11 Können mit Türsteher gesicherte Ausführungsumgebungen geschaffen werden?

Ja, zusammen mit VM-basierter Technologie können Sie mit Türsteher hochgradig gesicherte Desktops erzeugen, auf denen Ihre Kunden und Mitarbeiter nur die Programme ausführen können, die Sie freigegeben haben. Sichern Sie VM-Hosts mit Türsteher-Technologie zusätzlich ab und sorgen Sie dafür, dass Ihre VM-Hosts gehärtet sind.

6.12 Kann man mit Türsteher auch Geräte sperren?

Sie können Treiber unerwünschter Geräte blockieren, damit können diese Geräte an den Rechnern auch nicht mehr genutzt werden (Plug & Play schlägt fehl). Mit Türsteher können Sie z. B. USB-Schnittstellen auf Kernel-Ebene dauerhaft und vollständig deaktivieren. Daneben sind Sie auch in der Lage, USB-Sticks und externe USB-Festplatten durch Blockieren der Treiber auf Endgeräten auf Kernel-Ebene zu blockieren, damit sind diese Geräte keine Gefahr mehr für Ihre IT.

6.13 Unterstützt Türsteher zentrales Logging?

Ja, Türsteher schreibt Ereignisse in das Windows-basierte Event-Log. Zusätzlich kann Türsteher jederzeit um kundenspezifische Module für weiteres Ereignisreporting erweitert werden, z. B. automatischer E-Mail-Versand, SMS-Versand, SNMP Traps, etc.

6.14 Wann startet der Kerneltreiber von Türsteher?

Der Treiber startet direkt nach dem sog. *kernel-init* von Windows. D. h. der sog. Bootloader lädt den Windows-Kernel, führt einige Initialisierungsschritte für den Start des Betriebssystems durch und startet direkt danach den Treiber von Türsteher. Türsteher wird daher, anders als viele andere Sicherheitsprogramme, zu einem sehr frühen Zeitpunkt gestartet und ist in der Lage, ausführbare Programme (wie Treiber und Systembibliotheken) bereits beim Hochfahren von Windows effektiv zu blockieren und den Rechner frühzeitig zu schützen.

6.15 Was bedeutet es, den Treiber zu pausieren?

Technisch gesehen wird der Treiber während des Pausierens gestoppt und nach Ablauf der Pausenzeit wieder gestartet.

6.16 Seit ich Türsteher benutze, kann ich bestimmte Software nicht mehr automatisch aktualisieren. Woran liegt das?

Es gibt Anwendungen, die während des Aktualisierungsprozesses (Update) Programmdateien in Ordner schreiben, die nicht von Türsteher freigegeben sind und von dort aus starten. In solchen Fällen blockiert Türsteher völlig korrekt die Ausführung der Programmdateien aus nicht freigegebenen Ordnern. Es bietet sich in diesen Fällen an, Türsteher für das Einspielen des Updates temporär zu deaktivieren, und nach der Aktualisierung der Software wieder zu aktivieren. Bitte beachten Sie, dass nach Updates oder Installation von Software insbesondere Hash-basierte Regeln aktualisiert werden müssen.

6.17 Was macht Türsteher für den normalen Anwender besonders nutzerfreundlich?

Der Treiber ist in der Lage, das Starten von unbekannten Programmen zu verhindern. Türsteher benötigt, anders als viele andere Sicherheitsprogramme, keine Dialoge, die den meist unkundigen Anwender nach Aktionen fragt. Dadurch reagiert Türsteher bei digitalen Angriffen oder unbeabsichtigten Klicks in der Folge automatisch richtig. Türsteher läuft zudem vollständig transparent im Hintergrund und ist im normalen Betrieb nicht wahrzunehmen. Keine Meldungen, keine Fragen, keine Popups. Ihre Sie und Ihre Mitarbeiter können sich auf die Arbeit konzentrieren.