**Technische Berichte in Digitaler Forensik**

**Herausgegeben vom Lehrstuhl für Informatik 1 der Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) in Kooperation mit dem Masterstudiengang Digitale Forensik (Hochschule Albstadt-Sigmaringen, FAU, Goethe-Universität Frankfurt am Main)**

# Analyse der Spurenmenge der Anwendung VirtualBox Version 5.0.10 r104061 unter Windows 7

Frank Block

31.03.2016

Technischer Bericht Nr. 10

Zusammenfassung:

Die Applikation VirtualBox wird zur Erstellung und Verwaltung von virtuellen Maschinen unter verschiedenen Betriebssystemen benutzt und unterstützt dabei ebenfalls mehrere Gastbetriebssysteme wie Windows, Linux und Mac OS X. Da sich virtuelle Maschinen zum Beispiel gut eignen um Angriffe durchzuführen und anschließend alle lokalen Spuren des Angriffs durch Löschen der VM zu verwischen, ist es aus forensischer Sicht beispielsweise relevant zu wissen, ob eine solche VM auf dem System existiert hat. Diese und andere Spuren werden in dieser Arbeit näher beleuchtet.

Entstanden im Rahmen des Moduls Browser- und Anwendungsforensik des Studiengangs Digitale Forensik im Wintersemester 2015/2016 unter der Anleitung von Felix Freiling, Holger Morgenstern und Michael Gruhn.

# Inhaltsverzeichnis

# Tabellenverzeichnis

# Abbildungsverzeichnis

# 1 Analysevorgehen

Für die Analysen dieses Ergebnisberichts wurde eine Windows 7 SP1 x86 VM verwendet und dabei intensiver Gebrauch der Snapshot Funktion gemacht. In dieser VM wurde VirtualBox (Version 5.0.10 r104061) installiert und mittels der in Kapitel 3 aufgeführten Aktionen auf Veränderungen getestet. Nach jeder Aktion wurde ein Snapshot des Zustands erstellt, zum vorherigen Zustand zurück gesprungen, die VM laufen gelassen ohne irgendwelche Aktionen durchzuführen und davon ebenfalls ein Snapshot erstellt (Noise-Reduction). Die einzelnen Aktionen wurden dabei drei Mal durchgeführt um temporäre Schwankungen auszuschließen und teilweise manuell an einem späteren Zeitpunkt der VM nochmals durchgeführt um die Ergebnisse zu vergleichen.

Generell wurde für die Analysen die Zustandsmethode mit der Ereignismethode kombiniert. Zur Analyse der durchgeführten Veränderungen (hier wurde innerhalb der VM mit Sysinternals Sync v.2.2 gearbeitet) wurden zum einen von jedem Snapshot Diskdumps erstellt, ein Diff mittels „idifference2.py“ von fiwalk[1] (Commit: a64c13dd95793c84356397e8f4b447c7d300f7bb) erstellt und die veränderten Dateien verglichen (Zustandsmethode). Um die relevanten Dateien zu identifizieren wurden diese Vergleiche noch mit den Ergebnissen aus dem Sysinternals Werkzeug Procmon[2] (Version 3.20) verifiziert. Bei Procmon kam dabei ein Filter für Prozesse zum Einsatz, der den „VirtualBox.exe“ Prozess selbst und alle Prozesse die den String „vbox“ beinhalten umfasste. Konkret waren das folgende Prozesse:

- VirtualBox.exe
- VBoxSVC.exe
- VBoxService.exe
- VBoxTestOGL.exe
- VBoxTray.exe

Es wurden ebenfalls Vergleiche der Registry mittels Regshot[3] (Version 1.9.0 x86 Unicode) (Zustandsmethode) und eine Analyse der entsprechenden API Calls für den Registry Zugriff mittels Procmon durchgeführt. Da die Relevanz der Veränderungen für diese Arbeit marginal ist, wird lediglich in Kapitel 3 und 5.2 darauf eingegangen.

[1] https://github.com/simsong/dfxml
[2] https://technet.microsoft.com/de-de/sysinternals/processmonitor.aspx
[3] http://sourceforge.net/projects/regshot/

# 2 Dateiinhalte

Die folgenden Angaben beziehen sich auf eine Beispielhaft erstellte VM mit dem Namen „test“.

| Dateipfad(e) | Dateiinhalt |
|---|---|
| **%VM-Folder%\Logs\VBox.log**<br><br>**%VM-Folder%\Logs\VBox.log.1**<br><br>**%VM-Folder%\Logs\VBox.log.2**<br><br>**%VM-Folder%\Logs\VBox.log.3** | **Dateiformat**: Text/Logeinträge<br><br>**Inhalt**: Logeinträge zum Lauf einer VM<br><br>**Zweck**: Debugging bei Fehlern der laufenden VM.<br><br>**Details**:<br><br>Der grobe Aufbau der Datei ist wie folgt:<br><br>`Timestamp Informationen`<br><br>Der Timestamp hat das Format HH:MM:SS:µs und ist zeitlich relativ zum Start der VM. Die Logeinträge beinhalten vor allem Informationen zu den folgenden Kategorien, in immer dieser Reihenfolge:<br><br>• Hostsystem (die VirtualBox Version, Windows Version, Größe des RAMs,…)<br>• Die VM selbst (Anzahl der CPUs, RAM, UUID, Pfade zur VM zugehörigen Dateien/Ordnern, Geräten,…)<br>• Das Aktivieren und Bereitstellen von Geräten (wie Audio, Netzwerk, …)<br>• CPU spezifische Informationen (z.B. Features)<br>• Statusänderungsinformationen (z.B. „Changing the VM state from 'POWERING_ON' to 'RUNNING'“)<br>• Zustand der VM beim Ausschalten (CPU Register Werte, die GDT, …)<br>• Statistiken zum Lauf (zum Beispiel Anzahl der Bytes die auf die Festplatte geschrieben wurde)<br><br>Diese Kategorien werden zum Teil mit entsprechenden Markern gekennzeichnet, wie zum Beispiel:<br><br>`********** Guest state at power off **********`<br><br>Für jeden Start der VM wird eine neue Logdatei erstellt, die Informationen über den Lauf dort eingetragen und die bereits vorhandenen Logfiles nach hinten geschoben: alte .3 wird mit .2 überschrieben, alte .2 mit .1 usw. Es existieren maximal 4 VBox.log Dateien, was 4 VM Starts entspricht. |

| | |
|---|---|
| **%VM-Folder%\Logs\VBoxHardening.log** | **Dateiformat**: Text/Logeinträge<br><br>**Inhalt**: Logeinträge vermutlich mit Informationen zur Bereitstellung der virtualisierten Umgebung.<br><br>**Zweck:** Vermutlich Debuggen von Problemen auf der Host Seite, beim Bereitstellen der virtualisierten Umgebung.<br><br>**Details:**<br><br>Bei jedem Start der VM wird die bestehende „VBoxHardening.log“ Datei gelöscht und neu erstellt.<br><br>Der grobe Aufbau der Datei ist wie folgt:<br><br>`Unbekannter Identifier  Informationen`<br><br>Der Identifier verändert sich hin und wieder, kann aber keinen bestimmten Arten von Logeinträgen zugeordnet werden. Ein Timestamp ist es absehbar auch nicht, da bestimmte Identifier später nochmals auftauchen. Bei allen Testläufen wurden exakt 12 verschiedene Identifier festgestellt, deren Auftreten von einmal bis 1743 Mal reichen. Der Identifier hat dabei immer dieselbe Form:<br><br>`hhh.hhh`<br><br>Wobei „h“ für ein Hexadezimalzeichen steht. Ebenfalls auffällig war bei den Analysen, dass bei exakt 10 Identifiern die ersten 3 Zeichen identisch sind und nur der hintere Teil verschieden, und die übrigen Zwei komplett unterschiedlich zu allen sind. Es wird vermutet, dass diese bestimmte Prozesse oder Threads (vor allem von VirtualBox selbst) identifizieren.<br><br>Am Anfang der Lodatei wird die VirtualBox Version ausgegeben, direkt im Anschluss folgen Informationen wie Erstellungsdatum und Größe zu den Host System DLLs „ntdll.dll“, „kernel32.dll“, „KernelBase.dll“ und „apisetschema.dll“. Diese Informationen werden insgesamt 3 Mal ausgegeben, jedes Mal mit einem anderen Identifier. Der Großteil der restlichen Logdatei (gemittelt ca. 84%) wird durch Einträge bestimmt, die den String „supR3Hard“ beinhalten. Z.B.:<br><br>`3b4.978: supR3HardenedVmProcessInit: Opening vboxdrv...`<br><br>Dabei beinhalten die meisten Einträge Verweise auf Systemdateien des Hostdateisystems die verwendet/geladen werden. |

| | |
|---|---|
| **%VM-Folder%\test.vbox** | **Dateiformat**: XML<br><br>**Inhalt**: VM Konfigurationseinstellungen sowie Verweise auf Festplatten, sonstigen Medien und Snapshots<br><br>**Zweck:** Speicherung der VM Konfiguration<br><br>**Details:**<br><br>Diese Datei enthält die Einstellungen der VM (wie zum Beispiel Netzwerkadapter und Anzahl der CPU Kerne), einen Verweis auf die zugehörige(n) Festplatte(n) (bei VMDK jeweils nur auf die erste VMDK Datei; siehe Eintrag zu „test.vmdk“) und ebenfalls die Snapshots (sowohl deren Festplattendateien und alle zu einem Snapshot gehörenden Einstellungen). Für jeden Snapshot werden alle Einstellungen separat gespeichert.<br><br>Diese Datei wird bei Veränderungen neu erstellt (erhält neue Inode/File ID) und der alte Inhalt plus Veränderungen eingetragen. Dies geschieht allerdings zunächst in die temporäre Datei test.vbox-tmp. Bevor der Vorgang abgeschlossen ist, wird die aktuelle test.vbox Datei in test.vbox-prev umbenannt (dabei wird eine bestehende test.vbox-prev überschrieben) und abschließend test.vbox-tmp in test.vbox umbenannt. |

| | |
|---|---|
| **%VM-Folder%\test.vbox-prev** | **Dateiformat**: XML<br><br>**Inhalt**: VM Konfigurationseinstellungen vor der letzten Veränderung<br><br>**Zweck:** Möglichkeit um zu alten Einstellungen zurückzukehren<br><br>**Details**:<br><br>Diese Datei beinhaltet dieselbe Art an Informationen wie test.vbox. Konkret beinhaltet sie prinzipiell den Zustand von test.vbox vor der Durchführung einer bestimmten Aktion, die Veränderungen an test.vbox durchführt. Es existieren allerdings Fälle in denen bestimmte Aktionen zu scheinbar mehreren Veränderungen führen, was zum einen dazu führen kann, dass zum Beispiel nach der Veränderung der RAM Größe der einzige Unterschied beider Dateien ein Timestamp ist, oder dass sogar nach dem ersten Setzen einer ISO als CD-ROM beide Dateien komplett identisch sind (in Procmon konnten dabei drei Mal die Dateioperationen beobachtet werden, die bei Veränderungen normalerweise nur einmalig ausgeführt werden; siehe die Beschreibungen bei „test.vbox“). Die Vermutung ist, dass mindestens die letzte Iteration entweder nur die „beiläufige“ oder eben keine Veränderung mehr an „test.vbox“ durchführt, wodurch danach die eigentlichen Veränderungen in beiden Dateien enthalten sind. |

| | |
|---|---|
| **%VM-Folder%\test.vmdk** | **Dateiformat**: Text und/oder der Festplatteninhalt im RAW Format<br><br>**Inhalt:** Metadaten zur Festplatte und der eigentliche Festplatteninhalt<br><br>**Zweck:** Bereitstellen einer Festplatte für die VM<br><br>**Details**:<br><br>Wenn die Option für die Aufsplittung der Festplatte bei 2GB verwendet wird, oder eine „Flat“ Disk erzeugt wurde (kein dynamisches Wachsen sondern die Größe der Festplatte wird von vornherein beanschlagt) beinhaltet die erste VMDK Datei lediglich Text und die eigentlichen Festplatten Daten sind in einer oder mehreren separaten VMDK Dateien.<br><br>Bei der Verwendung einer einzigen VMDK Datei für die gesamte Festplatte mit dynamischem Wachstum befindet sich am Anfang der Datei ebenfalls Text, allerdings befinden sich die RAW Festplatten Daten ebenfalls in der Datei. Diese Metadaten finden sich aber lediglich in der ersten/einzigen VMDK Datei.<br><br>Der Text Part umfasst in allen Fällen die folgenden Informationen:<br><br>• Der „createType“ (definiert um welche Art von Festplatte es sich handelt)<br>• Liste der zu dieser Festplatte zugehörigen VMDK Datei(en)<br>  ○ Falls keine Aufsplittung gewählt wurde, steht hier der Name der Flat Disk VMDK Datei und im Falle einer dynamisch wachsenden Festplatte referenziert sich die VMDK Datei selbst.<br>  ○ Bei Aufsplittung stehen hier alle dazugehörigen VMDK Dateien<br>• Informationen über die Disk- und BIOS Geometrie (z.B. die Anzahl der Sektoren)<br>• UUID der Disk selbst<br>• „Parent UUID“ - Verweis auf „Eltern“ Disk (nur relevant für Snapshots); hier nur aus „0“en bestehend<br>• „Modification UUID“ - diese besteht komplett aus „0“en wenn bisher noch keine Veränderungen an der VMDK Datei durchgeführt wurden, ansonsten eine nicht 0 UUID<br>• „Parent Modification UUID“ - Nur bei Snapshots relevant; besteht hier nur aus „0“en |

| | |
|---|---|
| **%VM-Folder%\Snapshots\{UUID}.vmdk** | **Dateiformat**: Text und Festplatteninhalt im RAW Format<br><br>**Inhalt:** Metadaten zum Snapshot und alle Veränderungen an der Festplatte seit Erstellung des Snapshots<br><br>**Zweck:** Die Möglichkeit jederzeit an einen bestimmten Zustand der VM zurückspringen zu können (um zum Beispiel die Auswirkungen von Malware rückgängig zu machen).<br><br>**Details**:<br><br>Für jeden Snapshot wird für jede Festplatte eine separate VMDK Datei mit entsprechender UUID erstellt. Diese neue Datei enthält allerdings nicht den Zustand der Festplatte zu diesem Zeitpunkt sondern alle Veränderungen an der Festplatte seit Erstellung des Snapshots. Wird ein nachgelagerter Snapshot erstellt, wird diese Snapshot VMDK quasi „eingefroren“ und eine weitere neue VMDK mit neuer UUID erstellt, in der fortan Veränderungen eingetragen werden. Der Aufbau der Datei entspricht im Groben den Ausführungen für „test.vmdk“ mit den folgenden Ausnahmen:<br><br>• „Parent UUID“ - Zeigt auf die UUID der vorherigen durch einen Snapshot erstellten VMDK Datei oder beim ersten Snapshot auf die UUID der eigentlichen Festplatte<br><br>• „Parent Modification UUID“ - Zeigt auf die „Modification UUID“ des Eltern Snapshots oder eigentlichen Festplatte, wenn diese modifiziert wurde (eine nicht 0 „Modification UUID“ hat), ansonsten besteht sie nur aus „0“en |

| | |
|---|---|
| **%USERPROFILE%\.VirtualBox\VBoxSVC.log**<br>**%USERPROFILE%\.VirtualBox\VBoxSVC.log.1**<br>**%USERPROFILE%\.VirtualBox\VBoxSVC.log.2**<br>**%USERPROFILE%\.VirtualBox\VBoxSVC.log.3**<br>**%USERPROFILE%\.VirtualBox\VBoxSVC.log.4**<br>**%USERPROFILE%\.VirtualBox\VBoxSVC.log.5**<br>**%USERPROFILE%\.VirtualBox\VBoxSVC.log.6**<br>**%USERPROFILE%\.VirtualBox\VBoxSVC.log.7**<br>**%USERPROFILE%\.VirtualBox\VBoxSVC.log.8**<br>**%USERPROFILE%\.VirtualBox\VBoxSVC.log.9**<br>**%USERPROFILE%\.VirtualBox\VBoxSVC.log.10** | **Dateiformat**: Text/Logeinträge<br><br>**Inhalt**: Logeinträge über Hostdateisystemzugriffe, Netzwerkinformationen und Laden von Treibern<br><br>**Zweck**: Debuggen von Treiber/Datei/Netzwerkproblemen<br><br>**Details**:<br><br>Der grobe Aufbau der Datei ist wie folgt (vgl. VBox.log):<br><br>`Timestamp Informationen`<br><br>Der Timestamp hat das Format HH:MM:SS:µs und ist zeitlich relativ zum Start von VirtualBox. Am Anfang der Logdatei befinden sich Informationen zum Hostsystem (die VirtualBox Version, Windows Version, Größe des RAMs,…). Der Anfang ist fast identisch mit dem Inhalt von „selectorwindow.log“. Bei existierenden VMs befinden sich anschließend Einträge über das Laden der jeweiligen vbox Datei inkl. dem Dateipfad. Darüber hinaus enthält diese Logdatei:<br><br>• Informationen zum Laden von Treibern bzw. des Extension Packs<br>• Fehlermeldungen über<br>  ○ fehlende Medien wie zum Beispiel VMDK Dateien (inkl. Dateipfad)<br>  ○ fehlende VMs (inkl. VM Name)<br>  ○ Dateisystem Probleme wie zum Beispiel volllaufende Festplatte oder der Zugriff auf eine VMDK die von einem anderen Prozess verwendet wird<br>• Netzwerkinformationen wie DNS Server und das Starten und Stoppen von Netzwerken (wie zum Beispiel dem NAT Network)<br><br>Bei jedem Start von VirtualBox wird eine neue „VBoxSVC.log“ angelegt und zuvor alle früheren Logdateien um eins nach hinten verschoben: „VBoxSVC.log.9“ wird in „VBoxSVC.log.10“ umbenannt und dabei überschrieben, „VBoxSVC.log.8“ in „VBoxSVC.log.9“ usw. Insgesamt existieren maximal 11 Instanzen dieser Logdatei. |

| | |
|---|---|
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml** | **Dateiformat**: XML<br><br>**Inhalt**: VirtualBox Konfigurationseinstellungen wie registrierte VMs und Netzwerkadapter Einstellungen<br><br>**Zweck:** Zentrale Speicherung globaler VirtualBox Einstellungen<br><br>**Details**:<br><br>Enthält eine Liste der registrierten VMs (die UUID und der Pfad zur entsprechenden vbox Datei), der verwendeten Netzwerkadapter und deren Konfiguration und jeweils eine Liste der Dateipfade von hinzugefügten Medien. Dies passiert zum Beispiel beim Setzen einer ISO Datei als CD-ROM Laufwerk für eine VM oder beim Hinzufügen einer bestehenden VMDK Datei als Festplatte (siehe auch Kapitel 4.17).<br><br>Diese Datei wird bei Veränderungen neu erstellt (erhält neue Inode/File ID) und der alte Inhalt plus Veränderungen eingetragen. Dies geschieht allerdings zunächst in die temporäre Datei VirtualBox.xml-tmp. Bevor der Vorgang abgeschlossen ist, wird die aktuelle VirtualBox.xml Datei in VirtualBox.xml-prev umbenannt (dabei wird eine bestehende VirtualBox.xml-prev überschrieben) und abschließend VirtualBox.xml-tmp in VirtualBox.xml umbenannt. |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | **Dateiformat**: XML<br><br>**Inhalt**: VirtualBox Konfigurationseinstellungen vor der letzten Veränderung<br><br>**Zweck:** Möglichkeit um zu alten Einstellungen zurückzukehren.<br><br>**Details**:<br><br>Diese Datei beinhaltet dieselbe Art an Informationen wie VirtualBox.xml. Konkret beinhaltet sie prinzipiell den Zustand von VirtualBox.xml vor der Durchführung einer bestimmten Aktion, die Veränderungen an VirtualBox.xml durchführt (oder durchführen könnte). In den Fällen bei denen bestimmte Aktionen nicht zu konkreten Veränderungen an VirtualBox.xml geführt haben (war zum Beispiel beim Wiederherstellen eines Snapshots der Fall; siehe Kapitel 4.11), ist der Inhalt von VirtualBox.xml und VirtualBox.xml-prev anschließend identisch. |

| | |
|---|---|
| **%USERPROFILE%\.VirtualBox\selectorwindow.log**<br><br>**%USERPROFILE%\.VirtualBox\selectorwindow.log.1** | **Dateiformat**: Text/Logeinträge<br><br>**Inhalt**: Logeinträge zu bestimmten VM Management Aktionen von VirtualBox.<br><br>**Zweck**: Ggf. Nachvollziehbarkeit bestimmter durchgeführter Aktionen.<br><br>**Details**:<br><br>Der grobe Aufbau der Datei ist wie folgt (vgl. VBox.log):<br><br>`Timestamp Informationen`<br><br>Der Timestamp hat das Format HH:MM:SS:µs und ist zeitlich relativ zum Start von VirtualBox. Am Anfang der Logdatei befinden sich Informationen zum Hostsystem (die VirtualBox Version, Windows Version, Größe des RAMs,…). Der Anfang ist fast identisch mit dem Inhalt von „VBoxSVC.log“. Ansonsten befinden sich in dieser Datei fast ausschließlich Verweise auf bestimmte UUIDs (von VMs, VMDK Dateien, Snapshots,…) die bei bestimmten Aktionen (siehe Kapitel 3) verändert/hinzugefügt/gelöscht wurden, aber keinerlei Namen oder Pfade. Zum Beispiel werden bei dem Hinzufügen einer VM unter anderem diese beiden Einträge hinzugefügt:<br><br>`00:11:54.801612 GUI: UIMediumEnumerator: **Medium** with key={**113c4c06-e279-41da-9067-135d089f5ba9**} **created**`<br><br>`00:11:55.312347 GUI: UIMediumEnumerator: **Machine registration** event received, ID = **090ceeb8-c730-4300-a3b9-a4cc04b621c6**`<br><br>Der erste Eintrag enthält die UUID der erstellten VMDK Datei und der zweite die UUID der VM selbst.<br><br>In dieser Datei konnten keine Hinweise auf VM Namen oder Pfade entdeckt werden.<br><br>Bei jedem erneuten Start von VirtualBox wird die aktuelle „selectorwindow.log“ Datei in „selectorwindow.log.1“ umbenannt und diese dabei überschrieben. Für die neuen Logeinträge wird dann eine neue „selectorwindow.log“ Datei erstellt. Insgesamt gibt es also maximal zwei Versionen dieser Logdatei, eine für die aktuelle und eine für die letzte VirtualBox Instanz. |

**Tabelle 1: Dateiinhalte**

# 3 Anmerkungen

Nach den Analysen von Kapitel 4.1 bis 4.14 wurden für Vergleiche der globalen Konfigurationsdatei „VirtualBox.xml“ bzgl. des Löschens einer VM noch die VirtualBox Versionen 4.3.34 r104062, 4.3.18 r96516 und 4.2.18 r88781 installiert (siehe Kapitel 4.15, 4.16 und 4.17). Grund dafür waren die XML Attribut Werte „GUI/LastItemSelected“, „GUI/GroupDefinitions/“, „GUI/RecentListHD“ und „GUI/RecentFolderHD“, die sich in der „VirtualBox.xml“ der VirtualBox Installation des Autors befanden, bis Dato aber nicht in der zu analysierenden VirtualBox Installation. Erst nach der Installation dieser Versionen wurde der eigentliche Grund für diese Werte entdeckt, der nichts mit den Versionen zu tun hatte sondern mit den Aktionen die in den letzten 3 Unterkapiteln von Kapitel 3 beschrieben werden. Der Vollständigkeit halber wurden die Tests der letzten 3 Kapitel jedoch trotzdem mit allen 4 Versionen durchgeführt. Für diese Analysen wurden dabei jedoch nicht die in Kapitel 1 erwähnten Methoden verwendet, sondern lediglich die Datei „VirtualBox.xml“ verglichen.

Durch die zusätzliche Verwendung der Ereignismethode konnte festgestellt werden, dass sowohl für Veränderungen an „.vbox“ Dateien als auch an „VirtualBox.xml“ jeweils immer eine neue temporäre Datei erstellt wird, die dann am Ende in den endgültigen Dateinamen umbenannt wird (siehe Kapitel 2, die Zeilen für „test.vbox“ und „VirtualBox.xml“). Der temporäre Dateiname taucht folglich nie im Diskdump auf.

Bezüglich der VM Festplatten Analysen wurde sich lediglich auf das VMDK Dateiformat konzentriert.

Neben den Informationen aus Kapitel 5.2 konnten noch relevante Veränderungen durch VirtualBox in den folgenden Registrywerten vorgefunden werden:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU*\4
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU\iso\0

Die angeführten Nummern (0 und 4) können natürlich variieren.

Beide Registrywerte enthielten den Name einer zu einer VM als CD-Rom Laufwerk hinzugefügten ISO Datei.

Für weitere Analysen bzgl. Spuren zu gelöschten VMs (siehe Kapitel 5) können in zukünftigen Arbeiten ebenfalls die Event Logs mit einbezogen werden. Diese wurden für diese Arbeit außer Acht gelassen.

# 4 Veränderungen bei bestimmten Aktionen

## 4.1 Erstellen einer VM

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Datei existiert noch nicht |
| **%VM-Folder%\Logs\VBoxHardening.log** | Datei existiert noch nicht |
| **%VM-Folder%\test.vbox** | Datei wird erstellt |
| **%VM-Folder%\test.vbox-prev** | Datei wird erstellt; Beim interaktiven Erstellen werden die neu gesetzten Einstellungen immer wieder in die vbox Datei geschrieben, wodurch auch eine 2. vbox-prev Datei existieren kann, die zwar überschrieben wird aber aus einem Disk Dump wieder hergestellt werden kann. |
| **%VM-Folder%\test.vmdk** | Datei wird erstellt |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Wird mit erster VM erstellt; Falls Datei bereits existiert wird ein weiterer Eintrag für die VM mit dem Verweis auf die zugehörige vbox Datei hinzugefügt. |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Wird bei erster VM noch nicht erstellt; Ansonsten wird VirtualBox.xml-prev mit aktueller VirtualBox.xml überschrieben. |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Logeinträge werden hinzugefügt |

Tabelle 2: Erstellen einer VM

## 4.2 Verändern von VM Einstellungen

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Keine |
| **%VM-Folder%\Logs\VBoxHardening.log** | Keine |
| **%VM-Folder%\test.vbox** | Veränderungen gemäß Einstellungen inkl. potentiell automatisch hinzugefügter Veränderungen (wie zum Beispiel der Aktualisierung eines Timestamps) |
| **%VM-Folder%\test.vbox-prev** | Hier existieren mehrere Möglichkeiten (siehe entsprechender Eintrag in Kapitel 2):<br>• Vorheriger Zustand<br>• Neuer Inhalt entspricht den Veränderungen aus der test.vbox, nur die automatisch eingefügten Veränderungen unterscheiden sich<br>• Inhalt komplett identisch mit test.vbox |
| **%VM-Folder%\test.vmdk** | Keine |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Keine (Ausnahme: Einstellung hat mit Registrierung von Medien zu tun: zum Beispiel neue ISO Datei wird als optisches Laufwerk verwendet) |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Keine, mit derselben Ausnahme wie für VirtualBox.xml. Bei Veränderungen wird auch hier wieder VirtualBox.xml-prev mit aktueller VirtualBox.xml überschrieben. |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Logeinträge werden hinzugefügt |

Tabelle 3: Verändern von VM Einstellungen

## 4.3 Starten der VM

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Wird erstellt; Logeinträge werden hinzugefügt |
| **%VM-Folder%\Logs\VBoxHardening.log** | Wird erstellt; Logeinträge werden hinzugefügt |
| **%VM-Folder%\test.vbox** | Keine |
| **%VM-Folder%\test.vbox-prev** | Keine |
| **%VM-Folder%\test.vmdk** | Veränderungen gemäß Aktionen in der VM; Bei keinerlei Veränderungen durch die VM werden dennoch beim ersten Start Veränderungen an der VMDK Datei durchgeführt (es werden Informationen über die Disk Geometrie (z.B. die Anzahl der Sektoren) hinzugefügt). |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Keine |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Keine |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Keine |

Tabelle 4: Starten der VM

## 4.4 Stoppen einer laufenden VM

Hier wurde die VM „hart“ über VirtualBox und nicht über ein ACPI Event oder innerhalb der VM ausgeschaltet.

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Keine |
| **%VM-Folder%\Logs\VBoxHardening.log** | Keine |
| **%VM-Folder%\test.vbox** | Folgende Aktualisierungen/Eintragungen finden statt:<br>• Zeitstempel „lastStateChange“ über letzte VM Status-Änderung<br>• Die Position des Fensters der VM zu diesem Zeitpunkt<br>• Aktion die zum Beenden der VM geführt hat<br>• Nano Sekunden Unix Timestamp der „/VirtualBox/HostInfo/GUI/LanguageID“ Property wird aktualisiert. |
| **%VM-Folder%\test.vbox-prev** | Da die Veränderungen an test.vbox nicht atomar in einem Schritt durchgeführt werden, ist der Inhalt dieser Datei nicht der Zustand von test.vbox vor dem Ausführen dieser Aktion sondern beinhaltete bei den Tests den neuen Zustand von test.vbox bis auf einen anderen Timestamp und den fehlenden Eintrag zur Position des Fensters |
| **%VM-Folder%\test.vmdk** | Nach dem ersten Start werden beim Stoppen ebenfalls Veränderungen an der VMDK Datei durchgeführt (ebenfalls Geometrie Informationen, aber diesmal bzgl. der BIOS Geometrie). |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Keine |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Keine |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Nach dem ersten Starten/Stoppen einer VM überhaupt werden Logeinträge hinzugefügt, danach nicht mehr. Für den Start weiterer VMs werden ebenfalls nur beim ersten Stoppen Logeinträge erzeugt. |

Tabelle 5: Stoppen einer laufenden VM

## 4.5 Erstellen eines Snapshots

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Keine |
| **%VM-Folder%\Logs\VBoxHardening.log** | Keine |
| **%VM-Folder%\test.vbox** | Neue VMDK wird eingetragen und alle Einstellungen werden dupliziert und dem neuen Snapshot zugeordnet. Darüber hinaus werden dem „Machine“ Tag die folgenden Attribute hinzugefügt:<br>• currentSnapshot - Wert enthält UUID des aktuellen Snapshots<br>• currentStateModified="false"<br>• lastStateChange wird aktualisiert |
| **%VM-Folder%\test.vbox-prev** | Hier haben scheinbar mehrere interne Aktionen dazu geführt, dass der Inhalt dem aktuellen Zustand von test.vbox gleicht (siehe auch Kapitel 2). |
| **%VM-Folder%\test.vmdk** | Keine |
| **%VM-Folder%\Snapshots\{UUID}.vmdk** | Wird erstellt |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Keine |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Keine |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Logeinträge werden hinzugefügt |

Tabelle 6: Erstellen eines Snapshots

## 4.6 Starten und Stoppen der VM nach Erstellung des Snapshots

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Logeinträge werden hinzugefügt |
| **%VM-Folder%\Logs\VBoxHardening.log** | Logeinträge werden hinzugefügt |
| **%VM-Folder%\test.vbox** | Das Attribut „currentStateModified="false"“ des „Machine“ Tags wird bereits beim Starten gelöscht und ansonsten Veränderungen analog zu Kapitel 4.4. |
| **%VM-Folder%\test.vbox-prev** | Da wieder mehrfach Veränderungen an test.vbox durchgeführt werden, enthält diese Datei nicht mehr das Attribut „currentStateModified="false"“. Ansonsten analog zu Kapitel 4.4. |
| **%VM-Folder%\test.vmdk** | Keine |
| **%VM-Folder%\Snapshots\{UUID}.vmdk** | Veränderungen innerhalb der VM werden auf diese Datei angewendet. |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Keine |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Keine |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Keine |

Tabelle 7: Starten und Stoppen der VM nach Erstellung des Snapshots

## 4.7 Exportieren einer VM

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Keine |
| **%VM-Folder%\Logs\VBoxHardening.log** | Keine |
| **%VM-Folder%\test.vbox** | Keine |
| **%VM-Folder%\test.vbox-prev** | Keine |
| **%VM-Folder%\test.vmdk** | Keine |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Keine |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Keine |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Logeinträge werden hinzugefügt |

Tabelle 8: Exportieren einer VM

## 4.8 Klonen einer VM

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Keine |
| **%VM-Folder%\Logs\VBoxHardening.log** | Keine |
| **%VM-Folder%\test.vbox** | Es konnten keine beobachtet werden; Allerdings wird gemäß den Erklärungen in Kapitel 2 die Datei „erneuert“, aber keine Veränderung vorgenommen. |
| **%VM-Folder%\test.vbox-prev** | Gemäß den Erklärungen in Kapitel 2 wird diese Datei mit der „alten“ test.vbox überschrieben, wodurch der Inhalt beider Dateien identisch ist (vorausgesetzt es gab keine Veränderungen an „test.vbox“). |
| **%VM-Folder%\test.vmdk** | Keine |
| **%VM-Folder%\Snapshots\{UUID}.vmdk** | Keine |

Tabelle 9: Klonen einer VM – Veränderungen an Quell-VM

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Klon erhält keine Kopie und wird vorerst nicht erstellt |
| **%VM-Folder%\Logs\VBoxHardening.log** | Klon erhält keine Kopie und wird vorerst nicht erstellt |
| **%VM-Folder%\cloned.vbox** | Inhalt identisch zu Quell-VBOX Datei mit Ausnahme der UUIDs für die Maschine selbst, Snapshots, Festplatten und anderer Medien. |
| **%VM-Folder%\cloned.vbox-prev** | Klon erhält keine Kopie und wird vorerst nicht erstellt |
| **%VM-Folder%\cloned.vmdk** | Wird erstellt – Kopie der Quell VM mit leichten Veränderungen:<br>• Name der VMDK Datei<br>• CID wird angepasst<br>• UUID wird angepasst |
| **%VM-Folder%\Snapshots\{UUID}.vmdk** | Wird erstellt – Kopie der Quell VM mit leichten Veränderungen:<br>• Name der VMDK Datei<br>• CID wird angepasst<br>• UUID für Snapshot VMDK und der entsprechenden Eltern-VMDK wird angepasst. |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Ein weiterer Eintrag für die VM mit dem Verweis auf die zugehörige vbox Datei wird hinzugefügt. |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Die aktuelle VirtualBox.xml-prev wird mit aktueller VirtualBox.xml überschrieben. |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Logeinträge werden hinzugefügt |

Tabelle 10: Klonen einer VM

## 4.9 Hinzufügen einer bestehenden VM

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Wird von bestehender VM übernommen; Falls nicht vorhanden auch vorerst nicht erstellt |
| **%VM-Folder%\Logs\VBoxHardening.log** | Wird von bestehender VM übernommen; Falls nicht vorhanden auch vorerst nicht erstellt |
| **%VM-Folder%\added.vbox** | Keine (auch bei ungültigen Einstellungen für die neue Umgebung wie zum Beispiel nicht vorhandene ISO Datei) |
| **%VM-Folder%\added.vbox-prev** | Keine |
| **%VM-Folder%\added.vmdk** | Keine |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Wird mit erster VM erstellt; Falls Datei bereits existiert wird ein weiterer Eintrag für die VM mit dem Verweis auf die zugehörige vbox Datei hinzugefügt. |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Wird bei erster VM noch nicht erstellt; Ansonsten wird VirtualBox.xml-prev mit aktueller VirtualBox.xml überschrieben. |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Logeinträge werden hinzugefügt |

Tabelle 11: Hinzufügen einer bestehenden VM

## 4.10 Importieren einer VM

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Neue VM besitzt diese Datei noch nicht |
| **%VM-Folder%\Logs\VBoxHardening.log** | Neue VM besitzt diese Datei noch nicht |
| **%VM-Folder%\imported.vbox** | Wird erstellt; Inhalt zum Großteil identisch mit Quelle; Mindestens folgende Anpassungen (weitere Anpassungen hängen vor allem vom lokalen Support für die entsprechenden Features ab, bzw. ob zum Beispiel ISO Dateien referenziert werden die in der Zielumgebung nicht existieren/registriert sind):<br>• UUID der Maschine selbst und der VMDK Datei(en)<br>• Dateiname der VMDK<br>Der Zeitstempel „lastStateChange“ wird hierbei nicht angepasst. |
| **%VM-Folder%\imported.vbox-prev** | Neue VM besitzt diese Datei noch nicht |
| **%VM-Folder%\imported.vmdk** | VMDK wird erstellt; Inhalt bis auf kleine Veränderungen identisch mit Quelle; Die folgenden Anpassungen werden durchgeführt:<br>• Name der VMDK Datei<br>• CID wird angepasst<br>• UUID wird angepasst |
| **%VM-Folder%\Snapshots\{UUID}.vmdk** | Wird erstellt – Inhalt bis auf kleine Veränderungen identisch mit Quelle; Die folgenden Anpassungen werden durchgeführt:<br>• Name der VMDK Datei<br>• CID wird angepasst<br>• UUID für Snapshot VMDK und der entsprechenden Eltern-VMDK wird angepasst. |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Wird mit erster VM erstellt; Falls Datei bereits existiert wird ein weiterer Eintrag für die VM mit dem Verweis auf die zugehörige vbox Datei hinzugefügt. |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Wird bei erster VM noch nicht erstellt; Ansonsten wird VirtualBox.xml-prev mit aktueller VirtualBox.xml überschrieben. |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Logeinträge werden hinzugefügt |

Tabelle 12: Importieren einer VM

## 4.11 Wiederherstellen eines Snapshots

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Keine |
| **%VM-Folder%\Logs\VBoxHardening.log** | Keine |
| **%VM-Folder%\test.vbox** | Die UUIDs für die VMDK Dateien des aktuellen Snapshots werden gemäß der der neuen Snapshot VMDK angepasst, das Attribut „currentStateModified="false"“ wieder eingefügt (falls gefehlt) und das Attribut „lastStateChange“ aktualisiert. |
| **%VM-Folder%\test.vbox-prev** | Der Inhalt war bei den Tests identisch mit test.vbox bis auf die Tatsache dass der Eintrag für die alte Snapshot VMDK mit der alten UUID noch zusätzlich zur neuen existierte. |
| **%VM-Folder%\test.vmdk** | Keine |
| **%VM-Folder%\Snapshots\{UUID}.vmdk** | Alte Datei wird gelöscht und eine neue mit neuer UUID wird erstellt. |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Gemäß den Ausführungen aus Kapitel 2 wurde bei den Tests die Datei zwar neu erstellt aber keine Veränderungen vorgenommen. |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Durch den Anstoß der Neuerstellung von „VirtualBox.xml“ wird die „VirtualBox.xml-prev“ durch die alte „VirtualBox.xml“ überschrieben, wodurch nun beide Dateien denselben Inhalt haben. |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Logeinträge werden hinzugefügt |

Tabelle 13: Wiederherstellen eines Snapshots

## 4.12 Löschen einer VM

In diesem Fall wurde die Option „Nur löschen“ gewählt, welche die VM nur aus VirtualBox löscht aber die Dateien der VM erhält.

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Keine |
| **%VM-Folder%\Logs\VBoxHardening.log** | Keine |
| **%VM-Folder%\test.vbox** | Keine |
| **%VM-Folder%\test.vbox-prev** | Keine |
| **%VM-Folder%\test.vmdk** | Keine |
| **%VM-Folder%\Snapshots\{UUID}.vmdk** | Keine |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Der entsprechende Eintrag für die VM wird gelöscht. |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Die aktuelle VirtualBox.xml-prev wird mit aktueller VirtualBox.xml überschrieben. |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Logeinträge werden hinzugefügt |

Tabelle 14: Löschen einer VM

## 4.13 Löschen einer VM (inklusive Dateien)

In diesem Fall wurde die Option „Alle Dateien löschen“ gewählt, welche sowohl die VM aus VirtualBox als auch deren Dateien löscht.

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Datei wird gelöscht |
| **%VM-Folder%\Logs\VBoxHardening.log** | Datei wird gelöscht |
| **%VM-Folder%\test.vbox** | Datei wird gelöscht |
| **%VM-Folder%\test.vbox-prev** | Datei wird gelöscht |
| **%VM-Folder%\test.vmdk** | Datei wird gelöscht |
| **%VM-Folder%\Snapshots\{UUID}.vmdk** | Datei wird gelöscht |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Der entsprechende Eintrag für die VM wird gelöscht. |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Die aktuelle VirtualBox.xml-prev wird mit aktueller VirtualBox.xml überschrieben. |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Logeinträge werden hinzugefügt |

Tabelle 15: Löschen einer VM (inklusive Dateien)

## 4.14 Löschen eines Snapshots

| Dateipfad | Veränderungen |
|---|---|
| **%VM-Folder%\Logs\VBox.log** | Keine |
| **%VM-Folder%\Logs\VBoxHardening.log** | Keine |
| **%VM-Folder%\test.vbox** | Erstellte VMDK und Snapshot Einstellungen werden wieder gelöscht. |
| **%VM-Folder%\test.vbox-prev** | Entspricht (bis auf den aktualisierten „lastStateChange“ Wert) vollständig dem vorherigen Zustand von test.vbox. |
| **%VM-Folder%\test.vmdk** | Keine |
| **%VM-Folder%\Snapshots\{UUID}.vmdk** | Wird gelöscht |
| **% USERPROFILE %\.VirtualBox\VBoxSVC.log** | Logeinträge werden hinzugefügt |
| **% USERPROFILE %\.VirtualBox\VirtualBox.xml** | Keine |
| **%USERPROFILE%\.VirtualBox\VirtualBox.xml-prev** | Keine |
| **%USERPROFILE%\.VirtualBox\selectorwindow.log** | Logeinträge werden hinzugefügt |

Tabelle 16: Löschen eines Snapshots

## 4.15 Schließen von VirtualBox nach dem Löschen einer VM

Die folgenden Ausführungen sind mindestens für diese Windows VirtualBox x86 Versionen gültig:

- 5.0.10 r104061
- 4.3.34 r104062
- 4.3.18 r96516
- 4.2.18 r88781

Für dieses Szenario wurde zuvor VirtualBox, mit der zu löschenden VM selektiert, geschlossen und erneut gestartet. Durch den Schließ-Vorgang wurden unter anderem die folgenden beiden Einträge in die Datei „VirtualBox.xml“ hinzugefügt:

```
<ExtraDataItem name="GUI/GroupDefinitions/" value="m=090ceeb8-c730-4300-a3b9-
a4cc04b621c6,m=e456250b-9abc-40dd-8fde-dad980a0fc1d,m=687234a6-ba3f-49db-8a62-
dd711a8e8673,m=cec052a9-90f0-40b2-9283-f1bd7f9f1936"/>

<ExtraDataItem name="GUI/LastItemSelected" value="m=test"/>
```

„GUI/GroupDefinitions/“ beinhaltet eine Liste von UUIDs aller registrierten VMs und „GUI/LastItemSelected“ den Namen der VM die beim Schließen von VirtualBox selektiert war. Wird die VM „test“ gelöscht, bleiben beide Einträge bis zum Schließen von VirtualBox unberührt. Erst beim Schließen wird die UUID herausgenommen und „GUI/LastItemSelected“ aktualisiert. Das heißt bis zum Schließen von VirtualBox kann der Hinweis für eine gelöschte VM in der Datei „VirtualBox.xml“ vorgefunden werden.

## 4.16 Löschen einer VM die eine existierende Disk hinzugefügt bekommen hat

Die folgenden Ausführungen sind mindestens für diese Windows VirtualBox x86 Versionen gültig:

- 5.0.10 r104061
- 4.3.34 r104062
- 4.3.18 r96516
- 4.2.18 r88781

Für dieses Szenario wurde zuvor VirtualBox, mit der zu löschenden VM („test“) selektiert, geschlossen und erneut gestartet. Anschließend wurde die VMDK Datei einer anderen nicht registrierten VM („test3“), in dessen Verzeichnis kopiert und der zu löschenden VM („test“) hinzugefügt, was zu folgenden zusätzlichen Einträgen in der „VirtualBox.xml“ Datei geführt hat:

```
<ExtraDataItem name="GUI/RecentFolderHD" value="C:/Users/troopers/VirtualBox VMs/test"/>

<ExtraDataItem name="GUI/RecentListHD" value="C:\Users\troopers\VirtualBox
VMs\test\existent.vmdk"/>
```

Im letzten Schritt wurde die VM „test“ mit der Option „Alle Dateien löschen“ gelöscht und VirtualBox geschlossen. Dies führte auch zum Löschen der Datei „C:\Users\troopers\VirtualBox VMs\test\test3-disk1.vmdk“. Aber auch nach einem erneuten Start blieben diese beiden Einträge unverändert und damit der Hinweis auf diese gelöschte VM.

## 4.17 Hinzufügen einer bestehenden Disk – Anknüpfung an Kapitel 4.16

Die folgenden Ausführungen sind mindestens für diese Windows VirtualBox x86 Versionen gültig:

- 5.0.10 r104061
- 4.3.34 r104062
- 4.3.18 r96516
- 4.2.18 r88781

Die folgenden Schritte wurden im Anschluss an die Veränderungen aus Kapitel 4.16 durchgeführt. Da die restlichen Veränderungen analog zu Kapitel 4.2 erfolgen, wird hier lediglich auf die Besonderheiten in Bezug auf die Datei „VirtualBox.xml“ und die bereits gelöschte VM eingegangen.

Wenn an diesem Punkt zu einer anderen VM eine bereits existierende VMDK Datei als Festplatte hinzugefügt wird, führt das zu den folgenden beiden Veränderungen (vergleiche Kapitel 4.16):

```
<ExtraDataItem name="GUI/RecentFolderHD" value="C:/Users/troopers/VirtualBox VMs/anotherVM"/>

<ExtraDataItem name="GUI/RecentListHD" value="C:\Users\troopers\VirtualBox
VMs\anotherVM\anotherVM.vmdk;C:\Users\troopers\VirtualBox VMs\test\existent.vmdk;"/>
```

Der Pfad zu der VM verschwindet durch diese Aktion zwar aus „GUI/RecentFolderHD“, in „GUI/RecentListHD“ bleibt der Pfad zu der ursprünglichen VMDK Datei jedoch bestehen und damit auch der Hinweis auf die gelöschte VM.

# 5 Ergebnisse bezüglich der Detektierung gelöschter VMs

Das relevante Szenario an dieser Stelle ist eine VM die inkl. all ihrer Dateien gelöscht wurde.

## 5.1 Indizien auf dem Dateisystem

Wie in Kapitel den Kapiteln 4.15, 4.16 und 4.17 beschrieben, können vor allem in der „VirtualBox.xml“ bzw. der „VirtualBox.xml-prev“ Hinweise auf gelöschte VMs vorgefunden werden. Darüber hinaus können in der Logdatei VBoxSVC.log Hinweise auf zu einer VM gehörigen Medien mit Pfaden und potentiell auch UUIDs vorgefunden werden, die ebenfalls gute Indikatoren für gelöschte VMs liefern. Allerdings bleiben diese Informationen höchstens solange erhalten bis die Logdateien vollgelaufen sind oder VirtualBox mindestens 11 Mal neugestartet wurde. Auch die Logdatei „selectorwindow.log“ kann potentiell Hinweise liefern, allerdings liefert sie normalerweise nur UUIDs. Mit Glück können diese möglicherweise mit Informationen aus „VBoxSVC.log“ zu konkreten VMs korreliert werden, allerdings wurden im Rahmen dieser Tests keine UUIDs aus „selectorwindow.log“ in „VBoxSVC.log“ gefunden.

## 5.2 Indizien in der Registry

Die folgenden Registerwerte scheinen vor allem durch das Auswählen von Dateien des Dateisystems über VirtualBox beeinflusst zu werden (wenn ein kleines Explorer Fenster zur Auswahl geöffnet wird). Solange bis zum Löschen einer VM und einer Analyse keine Aktionen durchgeführt wurden die diese Registerwerte beeinflussen bleibt der Wert bestehen und kann einen Hinweis auf diese VM geben auch wenn Sie bereits gelöscht wurde.

- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\FirstFolder\0
    - Dieser Registerwert wurde vor allem durch die Aktion aus Kapitel 4.2 beim Auswählen einer ISO Datei beeinflusst. Dabei wurde der Ordner der „test“ VM mit vollem Pfad in diesen Register Wert geschrieben. Allerdings wurde zum Beispiel beim Exportieren einer VM der Pfad zur VM wieder gelöscht, da hier ein Zielordner ausgewählt wird.
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU\vbox\0
    - Beim Hinzufügen einer bestehenden VM wird in diesen Registerwert vor allem der Name der ausgewählten vbox Datei geschrieben. Beim Hinzufügen einer weiteren VM wurde bei Tests dieser Wert nicht überschrieben sondern ein weiterer Registerwert mit der nächst höheren Nummer angelegt, weshalb diese Informationen nicht so flüchtig zu sein scheinen wie der „FirstFolder“ Wert.
- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU\ova\0
    - Beim Importieren einer VM wird in diesen Registerwert vor allem der Name der ausgewählten ova Datei geschrieben. Beim Importieren einer weiteren VM wurde bei Tests dieser Wert nicht überschrieben sondern ein weiterer Registerwert mit der nächst höheren Nummer angelegt, weshalb diese Informationen nicht so flüchtig zu sein scheinen wie der „FirstFolder“ Wert.

Die angeführten Nummern (in diesem Fall jedes Mal 0) können natürlich variieren.

Aufgrund von einiger weiterer Aktivität in den „Shell Bag“ Registry Bereichen, wurden noch die folgenden Werkzeuge verwendet um Hinweise auf Ordner im Bezug zu den VMs zu identifizieren:

- https://github.com/williballenthin/shellbags (Commit: 1224b8d745f574938b8a9ac350ffa6a398adaaa1)
- Mitec Windows Registry Recovery 1.5.3.0
- RegRipper 2.8
- ShellBagsView 1.16

Es konnten zwar keine Original Pfade zu irgendwelchen der erstellten VMs gefunden werden, allerdings wird auch hier scheinbar jedes Mal bei einem Dateisystem Zugriff (siehe weiter oben) ein Eintrag der folgenden Form generiert:

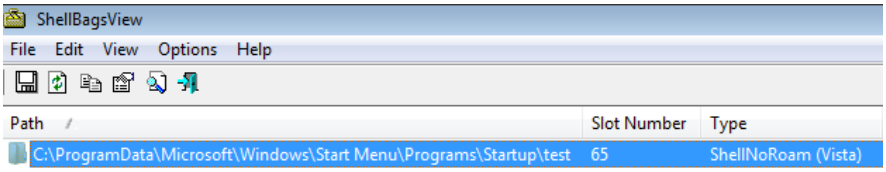


**Abbildung 1: ShellBagsView**

Der Wert im „Startup“ Ordner entspricht dabei nicht dem VM Namen sondern dem Ordnernamen in dem sie sich die VM befindet. Erzeugt werden diese Einträge vor allem bei den folgenden Aktionen:

- Exportieren einer VM
- Importieren einer VM
- Hinzufügen einer bestehenden VM
- Hinzufügen von bestehenden Medien zu einer VM

In dem Ordner „Startup“ konnten zwar weder Dateien noch Ordner mit diesen Namen vorgefunden werden, die Informationen aus den „Shell Bags“ bleiben aber auch nach dem Löschen der VM bestehen und können einen Hinweis auf die ursprüngliche Existenz geben.