Datenreduktion mittels kryptographischer Hashfunktionen in der IT-Forensik: Nur ein Mythos?

Harald Baier[1,2] · Christian Dichtelmüller [1]

Hochschule Darmstadt[1]
Fachbereich Informatik
Haardtring 100, D-64295 Darmstadt

Center for Advanced Security Research Darmstadt - CASED[2]
Mornewegstraße 32, D-64293 Darmstadt
harald.baier@cased.de,
christian.dichtelmueller@stud.h-da.de

Zusammenfassung

Kryptographische Hashfunktionen werden in der Computerforensik unter Anderem dazu eingesetzt, bereits bekannte Dateien effizient an Hand ihres Hashwerts zu erkennen. Enthält diese bekannte Datei keine strafbaren Inhalte, kann diese zunächst für die weitere forensische Untersuchung ausgeblendet werden. Dies führt zu einer Datenreduktion der tatsächlich zu untersuchenden Dateien. Die weltweit verbreitetste Datenbank für diese Methode der Datenreduktion ist das Reference Data Set (RDS) der National Software Reference Library (NSRL) des National Institute of Standards and Technology (NIST). Trotz des hohen Verbreitungsgrades des RDS ist bis heute unklar, wie gut dieser Ansatz zur Datenreduktion tatsächlich funktioniert. Die vorliegende Arbeit hat daher zum Ziel, die Datenreduktion mittels RDS in der Praxis zu bestimmen. Dazu definieren wir unterschiedliche Nutzergruppen, wie sie im Rahmen einer forensischen Untersuchung auftreten werden, und bestimmen jeweils die Datenreduktionsrate. Im Ergebnis führt die Anwendung der RDS zu Datenreduktionsraten, die für praktisch relevante Szenarien in der Größenordnung von 20 Prozent liegen. Aus unserer Sicht erfüllt dieser Ansatz daher in der Praxis nicht die an ihn gestellten Anforderungen.

1 Einleitung

Computer, mobile Endgeräte oder andere IT-Systeme sind in der heutigen Welt weit verbreitet. Ein Großteil der Arbeit in Büros, aber auch privat, wird mit Hilfe dieser Geräte ausgeführt. Ob dies das Schreiben von Briefen, das Versenden von E-Mails oder das Führen von elektronischen Kalendern ist, alle diese Tätigkeiten erzeugen Daten. Im Falle einer IT-forensischen Untersuchung können diese Daten dazu beitragen, Verbrechen aufzuklären oder Beschuldigte zu entlasten.

Oft werden dazu persistente Speicher (magnetische Festplatten, Flashspeicher) untersucht, deren Speicherkapazität sich dem Mooreschen Gesetz entsprechend exponentiell steigert. Damit vergrößert sich auch die Datenmenge exponentiell, die Computerforensiker analysieren müssen.

Wo früher noch ein paar hundert Megabyte an Daten auf einem privaten Computer anfielen, sind dies heute schon mehrere Hundert Gigabyte oder sogar Terabyte. Server können noch deutlich mehr Daten enthalten. Diese erheblichen Mengen an Daten erschweren es Forensikern, relevante Spuren zu finden. Eine Vorsortierung der Daten, eine Datenreduktion also, ist deswegen wünschenswert.

Das Auffinden von strafbaren Inhalten gleicht daher oft der sprichwörtlichen Suche nach der Nadel im Heuhaufen. Um diese Suche zu vereinfachen, stehen dem IT-Forensiker zwei Ansätze zur Verfügung: Whitelisting 'verkleinert den Heuhaufen', während Blacklisting die 'Nadel vergrößert'.

Eine Whitelist ist eine Datenbank nicht-inkriminierter Dateien. Typische Dateien einer Whitelist sind Dateien gängiger Betriebssysteme sowie Applikationen wie Browser, Mailclients oder Office-Suiten. Im Rahmen einer forensischen Untersuchung werden die auf dem beschlagnahmten Datenträger gefundenen Dateien automatisiert gegen die Dateien der Whitelist abgeglichen. Diese Vorgehensweise ist in Standardwerken der Computerforensik wie [Ges11], [Carr05] oder [Case10] beschrieben. Gängige forensische Software wie EnCase oder FTK bieten Importfunktionen für Whitelists an.

Aus Effizienzgründen (insbesondere im Hinblick auf Speicherbedarf der Whitelist) referenziert die Whitelist Dateien mittels kryptographischer Hashwerte. Die Sicherheitseigenschaften solcher Hashfunktionen, insbesondere die Preimage-Resistance, garantieren, dass ein vorgelegter Hashwert der Whitelist tatsächlich zu der Datei auf dem untersuchten Datenträger gehört.

Die Erzeugung einer Whitelist ist aufwändig, da für jede Datei entschieden werden muss, ob diese potenziell strafbare Inhalte enthält oder nicht. Ein koordiniertes Vorgehen ist daher wünschenswert. Die weltweit verbreitetste Datenbank für diese Methode der Datenreduktion ist das Reference Data Set (RDS) der National Software Reference Library (NSRL) des US-amerikanischen National Institute of Standards and Technology (NIST) [NIST12].

Trotz des hohen Verbreitungsgrades des RDS ist bis heute unklar, wie gut dieser Ansatz zur Datenreduktion tatsächlich funktioniert. Die einzige uns bekannte Arbeit, die sich mit der Datenreduktionsrate beschäftigt, ist ein Vortrag des NIST-Mitarbeiters Douglas White [Whit08]. Darin behauptet er, dass nach Anwendung der RDS nur noch 30% des Datenträgers manuell untersucht werden müssen, die Datenreduktionsrate also 70% beträgt.

Die vorliegende Arbeit hat daher zum Ziel, die Datenreduktion mittels RDS in der Praxis zu bestimmen, da wir die von White behaupteten Raten für viel zu optimistisch halten. Für unsere Untersuchung simulieren wir verschiedene Computeranwender und legen für diese ein Datenträgerimage an. Für jedes der Images bestimmen wir die Datenreduktionsrate. Im Ergebnis führt die Anwendung der RDS zu Datenreduktionsraten, die für praktisch relevante Szenarien in der Größenordnung von 20 Prozent liegen, d.h. dass 80 Prozent der Dateien nicht ausgefiltert werden und daher manuell untersucht werden müssen. Aus unserer Sicht erfüllt dieser Ansatz daher in der Praxis nicht die an ihn gestellten Anforderungen.

Der weitere Aufbau dieser Arbeit ist wie folgt: In Abschnitt 2 erläutern wir die Grundlagen von kryptographischen Hashfunktionen sowie ihren Einsatz im Whitelisting der Computerforensik. Danach stellen wir in Abschnitt 3 typische Nutzungsformen von Computern vor, auf deren Grundlage wir die von uns simulierten Anwenderprofile erstellen. Anschließend geben wir in Abschnitt 4 die Datenreduktionsraten für die von uns simulierten Computernutzer vor. Wir schließen diese Arbeit mit einer Zusammenfassung und einem Ausblick in Abschnitt 5.

2 Hashfunktionen und Whitelisting

In diesem Kapitel erklären wir zunächst in Abschnitt 2.1, was eine Hashfunktion ist und welche Anforderungen wir an diese für den Einsatz in kryptographischen Anwendungen stellen. Danach erklären wir in Abschnitt 2.2, warum kryptographische Hashfunktionen den Zweck für Whitelisting erfüllen und erläutern die RDS.

2.1 Kryptographische Hashfunktionen

Es bezeichne $\{0,1\}^*$ die Menge der Bitstrings nicht-limitierter Länge, und es sei $bs \in \{0,1\}^*$. Wie üblich bezeichne h eine Hashfunktion. Ist n eine positive ganze Zahl, dann ist gemäß [MeOV97] h eine Funktion mit zwei Eigenschaften:

- *Komprimierung*: $h : \{0,1\}^* \longrightarrow \{0,1\}^n$.
- *Effiziente Berechnung*: Die Berechnung des Hashwerts $h(bs)$ ist 'schnell' in der Praxis.

Zur Anwendung in der Kryptographie muss h weitere Bedingungen erfüllen:

1. *Urbild-Resistenz (Preimage Resistance)*: Es sei ein Hashwert $H \in \{0,1\}^n$ gegeben. Dann ist es **in der Praxis** nicht möglich, eine Eingabe (d.h. einen Bitstring bs) mit $H = h(bs)$ zu finden.
2. *Zweite-Urbild-Resistenz (Second Preimage Resistance)*: Es sei ein Bitstring $bs_1 \in \{0,1\}^*$ gegeben. Dann ist es **in der Praxis** nicht möglich, eine weitere Eingabe (d.h. einen zweiten Bitstring) $bs_2 \in \{0,1\}^*$ zu finden, der $bs_1 \neq bs_2$ und $h(bs_1) = h(bs_2)$ erfüllt.
3. *Kollisionsresistenz (Collision Resistance)*: Es ist nicht möglich **in der Praxis**, irgendwelche zwei Bitstrings $bs_1, bs_2 \in \{0,1\}^*$ mit $bs_1 \neq bs_2$ und $h(bs_1) = h(bs_2)$ zu finden.

Diese Sicherheitsanforderungen sind nicht unabhängig voneinander. Zum Beispiel folgt aus der Kollisionsresistenz auch die Urbild-Resistenz (d.h. die erstere Forderung ist stärker). Details finden sich in [MeOV97]. Die Sicherheitsanforderungen implizieren eine wichtige Eigenschaft kryptographischer Hashwerte: Den *Lawineneffekt (avalanche effect)*. Dazu sei ein Bitstring bs mit zugehörigem Hashwert $h(bs)$ gegeben. Ersetzen wir bs durch einen anderen Bitstring bs', dann verhält sich $h(bs')$ pseudo-zufällig, wir haben also keine Kontrolle über die Ausgabe einer Hashfunktion, wenn wir die Eingabe ändern. Unterscheiden sich zum Beispiel bs und bs' in genau einem Bit, sehen die beiden zugehörigen Ausgaben $h(bs)$ und $h(bs')$ 'sehr' verschieden aus. Genauer formuliert ändert sich jedes Bit in $h(bs')$ mit einer Wahrscheinlichkeit von 50%, unabhängig von der Anzahl geänderter Bits in bs'. Typische kryptographische Hashfunktionen finden sich in Tabelle 1.

Name	MD5	SHA-1	SHA-256	SHA-512	RIPEMD-160
n	128	160	256	512	160

Tab. 1: Typische kryptographische Hashfunktionen

2.2 White- und Blacklisting in der Computerforensik

In diesem Abschnitt gehen wir auf zwei prominente Anwendungsfälle kryptographischer Hashfunktionen in der Computerforensik ein: White- und Blacklisting. Wir diskutieren insbesondere Whitelisting und die weltweit verbreitetste Whitelist, das Reference Data Set (RDS) des

US-amerikanischen NIST. Wir erläutern auch, warum deutsche Strafverfolgungsbehörden und Sachverständige auf die US-amerikanische RDS zurückgreifen.

White- und Blacklists zielen darauf ab, bereits bekannte Dateien auf einem zu untersuchenden Datenträger wiederzuerkennen. Aus Effizienzgründen (insbesondere im Hinblick auf Speicherbedarf der Liste) referenziert die White- oder Blacklist Dateien mittels kryptographischer Hashwerte. Die zugrundeliegende Idee ist sehr einfach: Da kryptographische Hashfunktionen pseudozufällige Ausgaben bei Änderung der Eingabe produzieren, können die Hashwerte einer Eingabe als eindeutige und sehr kurze Identifikatoren eines beliebig langen Eingabestrings angesehen werden. In der Computerforensik berechnet man typischerweise die Hashwerte über den Payload einer Datei (d.h. die Hashfunktionen werden auf Dateiebenen angewendet).

Die in Abschnitt 2.1 vorgestellten Sicherheitseigenschaften solcher Hashfunktionen, insbesondere die Preimage-Resistance, garantieren, dass ein vorgelegter Hashwert der White- oder Blacklist tatsächlich zu der Datei auf dem untersuchten Datenträger gehört. Aus heutiger Sicht ist es zeitlich zu aufwändig, zu einem gegebenen Hashwert der Liste eine andere Eingabe zu finden, insbesondere in der Kombination mehrerer kryptographischer Hashfunktionen, wie es unten dargestellt in der RDS realisiert wird.

Eine Whitelist ist eine Datenbank nicht-inkriminierter Dateien. Typische Dateien einer Whitelist sind Dateien gängiger Betriebssysteme sowie Applikationen wie Browser, Mailclients oder Office-Suiten. Im Rahmen einer forensischen Untersuchung werden die auf dem beschlagnahmten Datenträger gefundenen Dateien automatisiert gegen die Dateien der Whitelist abgeglichen. Diese Vorgehensweise ist in Standardwerken der Computerforensik wie [Ges11], [Carr05] oder [Case10] beschrieben. Gängige forensische Software wie EnCase oder FTK bieten Importfunktionen für Whitelists an. Die Dateien der Whitelist werden zunächst für die weitere Untersuchung ausgeblendet. Daher hat der Ermittler manuell weniger Dateien zu sichten.

Eine Blacklist hingegen enthält bekanntermaßen inkriminierte Dateien. Findet der Computerforensiker durch automatisierten Abgleich der Hashwerte eines Datenträgers einen Treffer in der Blacklist, sieht er sich diese Datei 'per Hand' an, um über einen möglichen strafbaren Inhalt direkt zu entscheiden.

Beide Listen tragen zur Datenreduktion bei. Liefert eine Blacklist aber direkt Indizien für strafbare Inhalte, reduziert eine Whitelist lediglich die händisch zu untersuchenden Dateien. Wir diskutieren daher im Folgenden nur noch Whitelists.

Die Erzeugung einer Whitelist ist aufwändig, da für jede Datei entschieden werden muss, ob diese potenziell strafbare Inhalte enthält oder nicht. Ein koordiniertes Vorgehen ist daher wünschenswert. Die weltweit verbreitetste Datenbank für diese Methode der Datenreduktion ist das Reference Data Set (RDS) der National Software Reference Library (NSRL) des US-amerikanischen National Institute of Standards and Technology (NIST) [NIST12].

```
"SHA-1","MD5","CRC32","FileName","FileSize","ProductCode","OpSystemCode","SpecialCode"

"AC91EF00F33F12DD491CC91EF00F33F12DD491CA","DC2311FFDC0015FCCC12130FF145DE78",\\
"14CCE9061FFDC001","WORD.EXE",1217654,103,"T4WKS",""
```

Abb. 1: Beispieleintrag in der NIST Reference Data Set (RDS)

Jeder Eintrag der RDS[1] besteht aus dem SHA-1 Hashwert (Secure Hash Algorithm 1, [SHS95]),

[1] `http://www.nsrl.nist.gov`, besucht am 30.04.2012

dem MD5 Hashwert (Message Digest Algorithm 5, [Rive92]), der CRC-32 Prüfsumme, dem Dateinamen und -länge der hinterlegten Datei. Ein Beispieleintrag der RDS für ein verbreitetes Textverarbeitungsprogramm der Größe ca. 1,2 MiB findet sich in Abbildung 1.

Da die RDS keine illegalen Inhalte enthält, wird sie weltweit zur Datenreduktion eingesetzt, auch von deutschen Strafverfolgern, Sachverständigen oder sonstigen Computerforensikern. Damit steht eine weltweit akzeptierte Datenbank für Whitelisting zur Verfügung.

3 Anwenderprofile und Testumgebungen

Unsere Aussagen über die Datenreduktion sollen praxisnah sein, d.h. wir wollen reale Benutzerprofile für unsere Tests simulieren. Daher beschreiben wir zunächst in Abschnitt 3.1 typische Anwendungsfälle für die Computernutzung, da daraus die im Rahmen einer Post-Mortem-Analyse relevanten Dateien entstehen. Im Anschluss beschreiben wir in Abschnitt 3.2 die Auswahl von Hard- und Software sowie die Anwenderprofile für unsere Tests zur Datenreduktion.

3.1 Anwendungsfälle der Computernutzung

Die Verwendungsmöglichkeiten eines Computers oder eines mobilen Endgerätes sind sehr vielseitig. Im Jahr 2010 haben laut einer Studie für den IT-Verband BITKOM[2] 83% der Bundesbürger einen Computer genutzt. Eine neuere Studie [BITK11b] vom 08.04.2011 besagt sogar, dass 79% der Bundesbürger einen Computer täglich nutzen. Eine Studie über das Nutzungsverhalten von Computernutzern ist leider nicht vorhanden, aber es existieren mehrere Studien über einzelne Anwendungsmöglichkeiten für den Computer. Im Folgenden sind einige der Anwendungsmöglichkeiten aufgeführt:

1. *Internet*: Eine Studie [BIT11d] für den BITKOM gibt an, dass 74% der 16- bis 74-jährigen Privatpersonen 2010 das Internet benutzt haben.
2. *Musik*: Musik wird nicht mehr ausschließlich über das Radio oder eine Stereoanlage konsumiert. 30% der Bundesbürger nutzen auch den Computer oder ein mobiles Endgerät zum Musik hören. Dies ergab eine Umfrage, welche ebenfalls der BITKOM in Auftrag gegeben hat [BITK11c].
3. *Spielen*: Nach einer durch die Universität Hohenheim veröffentlichten Studie [Quan11] spielen etwa 25% der Bundesbürger ab 14 Jahren regelmäßig. Sie nutzen dazu einen Computer oder eine Konsole.
4. *Speichern von Fotografien*: Laut einer Studie [BITK09] des Meinungsforschungsinstitutes Forsa für den BITKOM aus dem Jahr 2009 fotografieren 60% der Deutschen digital und ein Großteil speichert seine Fotografien auf einem Computer. Die Studie gibt weiterhin an, dass jeder zweite seine Fotos auch am Rechner nachbearbeitet.
5. *Anschauen von Filmen*: Eine ebenfalls durch den BITKOM in Auftrag gegebene Studie [BITK11a] aus dem Jahr 2011 hat ermittelt, dass etwa 25% der Internetnutzer Filme aus dem Internet herunterlädt oder im Internet DVDs bestellt.
6. *Arbeiten*. Der Hauptverwendungszweck für den Computer ist aber immer noch das Arbeiten. Der Hightech-Verband BITKOM [BITK10] ermittelte, dass in Deutschland 61% der Beschäftigten einen Computer regelmäßig während der Arbeit verwenden.

[2] www.bitkom.org

3.2 Testumgebungen

Um die Testumgebungen flexibel und einfach handhabbar zu machen, sind die Testumgebungen nur virtuell erstellt worden. Für die Erstellung der virtuellen Systeme ist der kostenlose VMWare Player[3] verwendet worden. Mit Hilfe dieser Software lassen sich virtuelle Umgebungen erstellen, in denen unterschiedliche Betriebssysteme lauffähig sind. Für diese Arbeit wurde der VMWare-Player in der Version 3.1.4 verwendet.

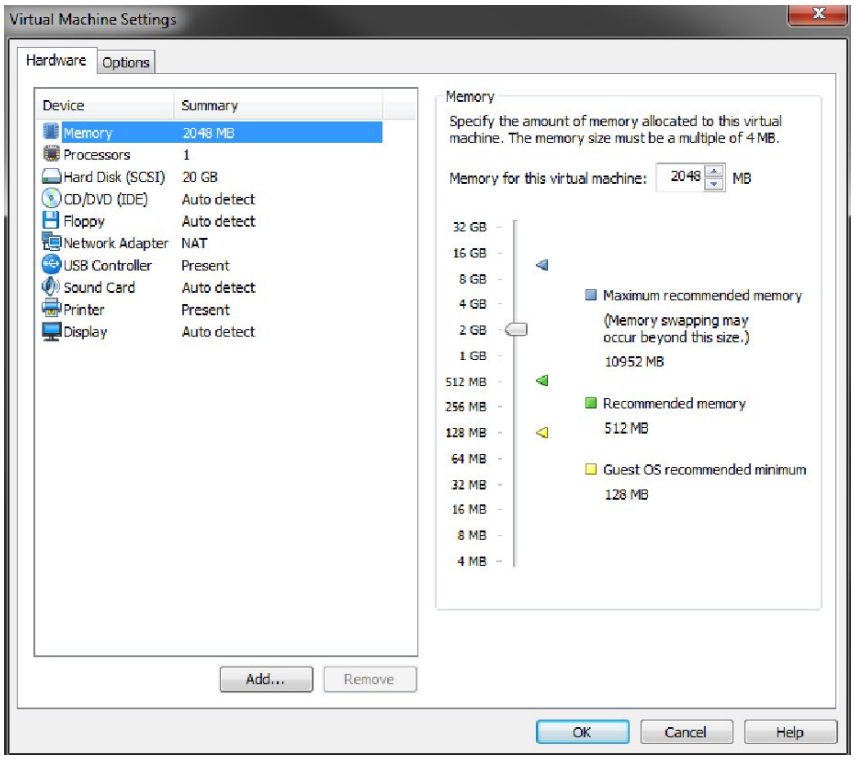

Abb. 2: Beispiel einer simulierten Hardwarekonfiguration

Die simulierte Hardware wurde je nach Anforderung ausgelegt. Besonders die Größe des simulierten Festplattenspeichers variierte zwischen den einzelnen Testumgebungen und ihren dargestellten Verwendungszwecken. Abbildung 2 zeigt eine beispielhafte Grundkonfiguration für die simulierte Hardware. Der Grundkonfiguration wurde ein Prozessor, ein 2048 Megabyte Arbeitsspeicher und 20 Gigabyte Festplattenspeicher zugeordnet. Die restliche Hardware wurde durch den VMWare-Player automatisch zugewiesen.

Im Hinblick auf die Auswahl der Betriebssysteme sind wir wie folgt vorgegangen: Um einen Vergleich mit den von Douglas White erhobenen Daten in Bezug auf die Datenreduktion zu erhalten, ist als ein Betriebssystem Windows XP ausgewählt worden. Das ebenfalls von Douglas White verwendete Windows 2000 eignet sich nicht mehr, da dieses Betriebssystem zu alt ist und deswegen eine sehr geringe Verbreitung hat. Da wir auch ein neueres Betriebssystem untersuchen wollen, haben wir uns neben Windows XP auch für Windows 7 entschieden. Windows Vista eignet sich auf Grund seiner Unbeliebtheit und der damit verbundenen Seltenheit nicht. Zusätzlich zu den beiden wohl am häufigsten verwendeten Betriebssystemen wollten wir auch eine weniger verbreitete Alternative testen. Hier fiel die Wahl auf Linux, weil wir dies

[3] `www.vmware.com/de/products/desktop_virtualization/player/overview.html`

in unserem akademischen Umfeld selbst verwenden. Aus der großen Anzahl an Distributionen wählten wir die im Moment beliebteste Linux-Distribution Ubuntu zum Testen der Datenreduktion aus. Die Marktanteile von Windows XP und Windows 7 rechtfertigen wie in Tabelle 2 dargestellt unsere Wahl im Hinblick auf die Verbreitung der beiden Betriebssysteme, Linux ist dabei als zusätzlicher Testfall zu sehen. Die Zahlen in Tabelle 2 stammen für Net Applications aus einem Artikel der PCWelt[4], für StatCounter von deren Webseite[5].

Anbieter \ Betriebssystem	Windows XP	Windows 7	Linux
Net Applications	56,72%	20,87%	0,96%
StatCounter	50,59%	25,86%	0,75%

Tab. 2: Marktanteile ausgewählter Betriebssysteme 12/2010 (Quelle: Siehe Fließtext)

Auf Basis der in Abschnitt 3.1 genannten Nutzungsformen von Computern und mit der Hilfe der Testumgebungen simulieren wir in dieser Arbeit folgende Datenträger bzw. Anwenderprofile:

1. Windows XP Professional mit Service Pack 3 (nur Betriebssysteminstallation)
2. Windows XP mit Standardsoftware
3. Windows XP Spieler
4. Windows 7 Professional 32bit (nur Betriebssysteminstallation)
5. Windows 7 mit Standardsoftware
6. Windows 7 Universal (d.h. alle Nutzungsformen)
7. Ubuntu 11.04

Die 'nackten' Betriebssysteminstallationen dienen als grundlegender Test, da eine Whitelist alle diese Dateien enthalten und die Datenreduktionsrate bei ca. 100% liegen sollte. Gleiches gilt für die Installation samt Standardsoftware. Diese Softwareauswahl haben wir für alle Anwenderprofile unterstellt; sie befindet sich daher außer bei den reinen Betriebssysteminstallationen auf allen Testumgebungen. Die installierte Software ist Adobe Reader X (Version 10.1), Avira Antivir Personal (Free Antivirus 10.0.0.650), Firefox 4.0, Internet Explorer 8.0, Skype, ICQ 7.5, OpenOffice 3.3.0, Microsoft Office 2007 Home and Student, Irfan View 4.30 sowie Winamp 5.62. Eine Begründung für diese Softwareauswahl haben wir in [Dich10] dargestellt.

Beim Spielerprofil wurde der Festplattenspeicher auf 250 Gigabyte und das RAM auf 4 Gigabyte erhöht. Wir installierten zahlreiche Spiele und Zusatzsoftware für Spieler, z.B. Total War Shogun 2, Fallout 3 – New Vegas, World of Warcraft, Starcraft 2, Dragon Age Origins, Civilisation 5, Civilisation 4 mit den beiden Addons Warlords und Beyond the Sword.

Beim Benutzerprofil *Universal* haben wir weitere Nutzungsformen aus Abschnitt 3.1 hinzugefügt und durch private Dokumente ergänzt. Diese wurden zufällig nach Dateityp (z.B. jpg, doc, avi) aus dem Internet heruntergeladen.

Die Ergebnisse unserer Tests stellen wir in Abschnitt 4 vor. Weitere Anwenderprofile samt zugehöriger Datenreduktionsraten sind in [Dich10] dargestellt.

[4] `www.pcwelt.de/news/Januar-2011-Aktuelle-Marktanteile-Browser-OS-Google-1457221.html`

[5] `gs.statcounter.com/#os-ww-monthly-201012-201012-bar`

4 Datenreduktionsraten

In diesem Abschnitt stellen wir für die in Abschnitt 3 diskutierten Profile die Datenreduktionsraten bei Anwendung der weltweit bekanntesten Whitelist vor, der Reference Data Set (RDS) 2.36 des US-amerikanischen NIST.

Es bezeichne M_G die Gesamtanzahl der Dateien auf dem zu untersuchenden Datenträger und M_{RDS} die Anzahl der Dateien auf dem Datenträger, die in der RDS indiziert sind. Wir schreiben R für die *Datenreduktionsrate* der RDS angewandt auf diesen Datenträger. Dann setzen wir

$$R = \frac{M_{RDS}}{M_G} \, . \tag{1}$$

Offensichtlich gilt $0 \leq R \leq 1$ und wir können R als Prozentsatz der bekanntermaßen nichtinkriminierten Dateien auf dem Datenträger ansehen. Es ist klar, dass ein größeres R eine größere Menge an automatisch ausgeblendeten Dateien bedeutet. Daher sollte R möglichst groß sein.

Wir weisen darauf hin, dass unsere Maßeinheit der Datenreduktionsrate die Anzahl an Dateien ist im Unterschied zu Douglas White [Whit08], der die Datenreduktion in Bytes misst. Da wir allerdings auf Dateiebene die Datenreduktion durchführen und die Anzahl bekannter bzw. unbekannter Dateien für den Erfolg bzw. Misserfolg relevant ist, halten wir unsere Maßeinheit für aussagekräftiger. Für eine Umrechnung muss man nur die durchschnittliche Dateigröße verwenden, die für große Datenträger nicht allzu unterschiedlich sein dürfte.

Profil	**Dateien gesamt**: M_G	**In RDS hinterlegt**: M_{RDS}	**Datenreduktionsrate**: R
Windows XP (nur OS)	10.467	5.490	52,45%
Windows XP Standardsoftware	22.801	9.689	42,49%
Windows XP Spieler	126.684	18.213	14,38%
Windows 7 (nur OS)	56.233	18.703	33,26%
Windows 7 Standardsoftware	77.601	23.414	30,17%
Windows 7 Universal	322.128	42.296	13,13%
Ubuntu	172.789	26.664	15,43%

Tab. 3: Datenreduktionsraten

Unsere Testergebnisse für die 7 in dieser Arbeit dargestellten Anwenderprofile sind in Tabelle 3 zusammengefasst. Die Spalte *Dateien gesamt* bezeichnet die Anzahl der Dateien, die sich für das jeweilige Profil auf dem Datenträger befunden haben. Wie oben definiert verwenden wir dafür die Variable M_G. Die Spalte *In RDS hinterlegt* beschreibt die Anzahl der Dateien, die in der RDS referenziert sind und daher automatisiert ausgeblendet werden. Die zugehörige Variable ist M_{RDS}. Die Spalte *Datenreduktionsrate* ist wie oben erklärt der Quotient aus den beiden vorhergehenden Spalten, d.h. der Anteil der Dateien, die in der RDS referenziert werden bezogen auf die auf dem Datenträger gespeicherten Dateien.

Es fällt auf, dass bereits die 'nackte' Installation eines Windows XP Betriebssystems nur eine Datenreduktionsrate von knapp über 50% erreicht und damit weit unter der behaupteten Rate

von 70% von Douglas White [Whit08] liegt (wir weisen erneut auf die unterschiedlichen Maßeinheiten von White und uns hin). Für Windwos 7 liegt die Datenreduktion sogar nur bei ca. 33%. Hintergrund ist, dass in der RDS folgende Dateiklassen nicht indiziert sind:

1. *Lokalisierte Dateien*: Da die RDS vom US-amerikanischen NIST gepflegt wird, finden lokalisierte Dateien für nicht-englische Betriebssysteminstallationen oder Anwendungen kaum Einzug in die RDS. Daher werden diese lokalisierten Dateien nicht erkannt. Beispieldateien sind in Abbildung 3 dargestellt.
2. *Installationsdateien der VMware*: Wir verwenden eine virtuelle Maschine als Host für unsere Testumgebungen. Dadurch liegen eine Reihe von VMware spezifischen Dateien vor, die nicht durch die RDS erkannt werden. Als Anwendungsfall denke man etwa an eine Ermittlung im Web-Hosting- oder Cloud-Bereich, wo oft virtuelle Maschinen gesichert werden.
3. *Veränderbare benutzer- oder hostspezifische Dateien*: Nicht alle Dateien eines Betriebssystems sind starr und invariant bei bzw. nach der Installation. Einige Dateien unterliegen einem ständigen Wandel ihrer Inhalte oder haben von sich aus einen individuellen Inhalt. Beispiele für solche Dateien sind Log-Files oder auch Verknüpfungen, temporäre Installations- oder Wiederherstellungsdateien. Auch solche Dateien, die speziell für die Benutzer eines Betriebssystems angelegt werden, sind individuell und daher nicht in der RDS indiziert. Während der Installation des Betriebssystems wird z.B. ein Benutzername verlangt, wodurch sich einige dieser Dateien immer unterscheiden.

```
Lokalisierte Beispieldateien:
Dokumente und Einstellungen/Administrator/Eigene Dateien/Eigene Bilder/Beispielbilder.lnk
Dokumente und Einstellungen/Administrator/Eigene Dateien/Eigene Musik/Desktop.ini

VMware-spezifische Beispieldateien:
Program Files/VMware/VMware Tools/intl.dll
Documents and Settings/All Users/Application Data/VMware/VMware Tools/Unity
Filters/adobeflashcs3.txt

Benutzerspezifische Beispieldateien:
Documents and Settings/Administrator/Favorites/Desktop.ini} oder
Documents and Settings/Administrator/Local Settings/History/History.IE5/index.dat
```

Abb. 3: Beispieldateien für nicht-indizierte Dateien eines 'nackten' Windows-Betriebssystems

Offensichtlich gelten vergleichbare Argumente auch für Installationen mit Standardsoftware (diese ist am Ende von Abschnitt 3.2 genannt). Beim Spieler, dessen Profil ebenfalls am Ende von Abschnitt 3.2 erläutert wird, beträgt die Datenreduktionsrate gar nur knapp 15%.

Es ist klar, dass durch Hinzunahme indvidueller Dateien, wie wir es für den allgemeinen Nutzer *Universal* angelegt haben, die Datenreduktionsrate nur sinken kann. Für den praktisch vermutlich am realistischsten Nutzer *Universal* liegt die Datenreduktionsrate für Windows 7 nur bei 13,13%.

Dies führt uns zu dem Schluss, dass Datenreduktion mittels Whitelist zwar ein Standardverfahren der Computerforensik ist, dieses aber bei weitem nicht das hält, was man sich davon verspricht.

5 Zusammenfassung und Ausblick

Wir haben die Datenreduktionsraten für praktisch relevante Nutzergruppen bestimmt und gezeigt, dass die Anwendung der RDS zu Datenreduktionsraten in der Größenordnung von 20 Prozent führt. Aus unserer Sicht erfüllt dieser Ansatz daher in der Praxis nicht die an ihn gestellten Anforderungen.

Offensichtliche Verbesserungen sind die Aufnahme lokalisierter Dateien in eine landesspezifische Whitelist. Dafür müsste es dann pro Land eine verantwortliche Stelle und wenn möglich eine international standardisierte Austauschschnittstelle geben.

In einem nächsten Schritt wäre zu untersuchen, inwiefern andere Ansätze für das Whitelisting (z.B. auf Basis von Fuzzy Hashing Ansätzen) erfolgversprechender sind. Die RDS kann auf Grund der Eigenschaften jedenfalls nicht für die Erkennung ähnlicher Inhalte oder von Fragmenten eingesetzt werden.

Literatur

[BITK09] BITKOM: Digitalfotos auf Papier bleiben beliebt. http://www.bitkom.org/de/markt_statistik/62013_61435.aspx (2009).

[BITK10] BITKOM: 61 Prozent aller Berufstätigen arbeiten mit dem Computer. http://www.bitkom.org/de/presse/66442_64770.aspx (2010).

[BITK11a] BITKOM: 12 Millionen Deutsche kaufen Filme im Internet. http://www.bitkom.org/de/markt_statistik/64038_67900.aspx (2011).

[BITK11b] BITKOM: Computernutzung nimmt weiter zu. http://www.bitkom.org/de/markt_statistik/64050_67616.aspx (2011).

[BITK11c] BITKOM: Fast jeder Dritte hört Musik mit dem PC, jeder Vierte mit dem Handy. http://www.bitkom.org/de/markt_statistik/64050_68464.aspx (2011).

[Carr05] B. Carrier: File System Forensic Analysis. Addison-Wesley (2005).

[Case10] E. Casey: Handbook of Digital Forensics and Investigation. Elsevier Academic Press (2010).

[Dich10] C. Dichtelmüller: Zur Bedeutung von Hashfunktionen fÃ¼r die Datenreduktion in der computerforensischen Ermittlung. Diplomarbeit, Hochschule Darmstadt.

[Ges11] Computer Forensik : Systemeinbrüche erkennen. dpunkt Verlag (2011).

[MeOV97] A. Menezes, P. Oorschot, S. Vanstone: Handbook of Applied Cryptography. CRC Press (1997).

[NIST12] NIST: NSRL download – description of the RDS contents. http://www.nsrl.nist.gov/RDS/rds_2.36/READ_ME.txt (2012).

[Quan11] T. Quandt: Deutsche Computerspieler: viele Gelegenheitszocker, wenige Extremgamer. https://www.uni-hohenheim.de/thema.html?&tx_ttnews (2011).

[Rive92] R. Rivest: The MD5 Message-Digest Algorithm. In: (1992).

[SHS95] SHS: Secure Hash Standard. In: (1995).

[Whit08] D. White: Hashing of File Blocks: When Exact Matches Are Not Useful. http://www.nsrl.nist.gov/Presentations.htm (2008).