Säuberung eines infizierten Windows-Betriebssystems auf Basis von Malware-Analyse-Berichten

Alexander Knaus Johannes Stüttgen Jan Göbel
Markus Engelberth Felix C. Freiling

{alex.knaus|johannes.stuettgen}@googlemail.com
{goebel|engelberth}@uni-mannheim.de

Abstract: Wir stellen einen Ansatz vor, wie man Systeme, die durch Schadsoftware infiziert sind, automatisiert säubern kann. Grundlage für die Säuberung ist ein Malware-Analyse-Bericht, der durch eine dynamischen Analyse des Schadprogramms mit Hilfe eines Sandbox-Systems erstellt werden kann. Aus diesem Bericht werden die Informationen über die durch das Schadprogramm vorgenommenen Systemveränderungen gewonnen. Wir präsentieren die Ergebnisse einer prototypischen Umsetzung des Ansatzes, der für einen Großteil heute existierender Schadprogramme gut funktioniert.

1 Einleitung

Durch die stetig ansteigende Zahl an Schadprogrammen und die zahlreichen Verbreitungsvektoren wie E-Mail oder das World Wide Web steigt auch die Zahl der infizierten Rechner stetig an. Wurde eine solche Infektion erkannt, etwa durch einen Virenscanner, muss der Rechner "desinfiziert" werden. Dies kann einerseits dadurch geschehen, dass man den Rechner in einen früheren, noch nicht-infizierten Zustand zurückversetzt. Andererseits kann man aber auch gezielt nur diejenigen Änderungen am System rückgängig machen, die durch die Schadsoftware verursacht wurden, meist durch ein spezielles Programm von einem Antivirenhersteller. Im ersten Fall besteht die Gefahr, dass zwischenzeitliche Änderungen an Benutzerdateien verloren gehen. Im zweiten Fall ist man darauf angewiesen, dass der Antivirenhersteller tatsächlich ein spezielles Entfernungswerkzeug für die spezifische Schadsoftware zur Verfügung stellt. Außerdem ist bei solchen Werkzeugen oft unklar, in welchem Ausmaß das bereits infizierte System bereinigt werden konnte.

In dieser Arbeit stellen wir einen Ansatz vor, der es auf einfache Weise erlaubt, Systeme von den Folgen einer Infektion zu befreien. Wir gehen dabei davon aus, dass neben dem infizierten Rechner jeweils auch die Schadsoftware (als ausführbare Datei) vorhanden ist, die die Infektion verursachte. Die Idee besteht darin, die Schadsoftware innerhalb einer kontrollierten Umgebung nochmals auszuführen und zur Laufzeit zu protokollieren, welche Änderungen am System vorgenommen werden. Genau diese Änderungen können dann anschließend wieder rückgängig gemacht werden. Da wir uns auf Windows-Systeme konzentrieren, betrachten wir Veränderungen in der Prozessliste (etwa das Starten von neuen Prozessen), Veränderungen in der Registry, und Veränderungen im Dateisystem.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

Offensichtlich hat ein solcher Ansatz Probleme, wenn Schadprogramme *zerstörende* Änderungen vornehmen, also etwa existierende Dateien oder Registry-Schlüssel überschreiben. Für die Wiederherstellung bräuchte man dann eine Kopie des vorherigen Wertes, und gerade das versucht der Ansatz ja zu vermeiden. Interessant ist jedoch, dass die überwiegende Mehrzahl von Schadprogrammen, die wir über die vergangenen Jahre mit Hilfe des InMAS-Systems [EFG+11] gesammelt haben, keine solche Änderungen vornimmt.

Wir haben unseren Ansatz prototypisch für Windows-Systeme implementiert und diesen Prototypen SMART genannt. SMART generiert aus einem Schadprogramm *automatisch* ein entsprechendes Desinfektionsprogramm. Voraussetzung ist ein Werkzeug zur verhaltensbasierten Analyse von Windows-Programmen. In unserem Falle kam dabei die CW-Sandbox [WHF07] zum Einsatz, aber SMART ist unabhängig von einer konkreten Sandbox, solange diese einen Verhaltensbericht in XML-Format produziert. Ein Anwender übergibt SMART lediglich den Verhaltensbericht und muss anschliessend nur die daraufhin generierte Säuberungsdatei ausführen. Auf einer repräsentativen Menge von 170 Schadprogrammen konnten in 90 Fällen (52%) *alle* Systemänderungen wiederhergestellt werden und in weiteren 71 Fällen (42%) *alle relevanten* Systemänderungen (als relevant werden Änderungen bezeichnet, die eine Ausführung des Schadprogramms komplett verhindern, etwa das Entfernen eines Autostart-Eintrags in der Registry oder das Löschen des Schadprogramms im Dateisystem). Somit ist in einer Vielzahl von praktischen Fällen die vollkommen automatisierte Erstellung von Säuberungsprogrammen möglich.

Der weitere Aufbau dieser Arbeit ist wie folgt: In Abschnitt 2 werden verwandte Arbeiten, sowie einige Grundlagen vorgestellt, die für das Verständnis von SMART notwendig sind. Die Arbeitsweise von SMART wird in Abschnitt 3 beleuchtet und anschließend wird in Abschnitt 4 eine Evaluation des Tools präsentiert. Bevor diese Arbeit mit einer Zusammenfassung abgeschlossen wird, werden in Abschnitt 5 einige Einschränkungen diskutiert.

2 Grundlagen und Verwandte Arbeiten

Als Basis für die von SMART generierten Säuberungsdateien dienen Verhaltensberichte sogenannter Sandbox-Systeme, beispielsweise der *CWSandbox* [WHF07]. Diese führen eine dynamische Analyse der Schadprogramme durch: In einer kontrollierten Umgebung werden diese ausgeführt und jegliche Aktivitäten werden protokolliert [BMKK06]. Das Resultat einer solchen Analyse ist ein Bericht (oftmals im XML-Format), der das Verhalten eines Schadprogramms während dessen Ausführung widerspiegelt. Das Aufzeichnen des Verhaltens ist meist durch API-Hooking oder Modifikationen des Betriebsystemkerns realisiert. Dadurch registrieren die Sandbox-Systeme etwaige Systemaufrufe des zu untersuchenden Programms und können so auch Einfluss auf dessen Programmfluss nehmen.

Der übliche Weg, den Antivirenprodukte verwenden, um einen Rechner zu desinfizieren, besteht darin, dass verdächtige Dateien gelöscht oder nicht mehr zugreifbar gemacht werden (Quarantäne). Die Wahrscheinlichkeit einer erfolgreichen Säuberung hängt im Wesentlichen von der Güte der vorhandenen Signaturen (auch als *Virendefinition* bekannt) ab [Coh90]. Zum Teil wird heute von Antivirenprodukten bereits mit heuristischen oder

dynamischen Erkennungsmethoden gearbeitet. Für derartige Methoden müssen jedoch weiterhin die Regeln, die bösartiges Verhalten ausmachen, manuell spezifiziert werden.

Ein Ansatz, wie eine derartige Spezifizierung automatisiert ablaufen kann, stellen Christodorescu et al. [CJK07] vor. Die Unterscheidung zwischen einer bös- und einer gutartigen Handlungsweise wird durch den Vergleich des Verhaltens eines bekannten Schadprogramms mit dem einer Reihe von ungefährlichen Programmen erzielt. Dieser Ansatz zielt auf die automatisierte Erkennung von Schadsoftware und nicht auf deren Entfernung.

Für spezielle Schadprogramme bieten viele AntiViren-Unternehmen auf ihren Internetseiten sogenannte *Malware Removal Tools* an. Diese beseitigen neben dem eigentlichen Schadprogramm ebenfalls die durch das Schadprogramm verursachten Systemänderungen. Die Erstellung dieser Tools erfolgt jedoch in der Regel nicht automatisch.

Mit der integrierten *System Restore* Funktionalität [Cor11] bietet Microsoft den Nutzern ihrer Betriebssysteme die Möglichkeit, das System in einen Zustand zu bringen, das es zu einem früheren Zeitpunkt hatte. Diese Systemwiederherstellung wirkt sich dabei auf Windows-Systemdateien, installierte Programme und vorgenommene Veränderungen in der Windows-Registry aus. Dabei werden auch Änderungen an ausführbaren Dateien, wie beispielsweise Skripten und Batchdateien rückgängig gemacht. Persönliche Dateien wie etwa E-Mails, Dokumente und Fotos bleiben davon jedoch unberührt. Im Gegensatz zu SMART erfordert System Restore jedoch, dass man *vor* der Infektion aktiv wird.

Ein anderer Weg, um ein System von einem Schadprogramm zu säubern, wird mit dem Tool *RETRO* aufgezeigt [KWZK10]. Während des normalen Betriebs eines Rechners wird ein sogenannter *Action History Graph* aufgezeichnet, der die ausgeführten Aktionen eines Systems beinhaltet und zur Säuberung verwendet wird. RETRO erfordert jedoch eine *kontinuierliche* Aufzeichnung der Systemaktivität, SMART hingegen nicht.

3 SMART

Das in Java implementierte Programm SMART besteht aus zwei verschiedenen Teilprogrammen: dem *SMART-Generator* und der *SMART-Executable*. Der Generator erstellt automatisch eine ausführbare Datei zur Säuberung. Diese Säuberungsdatei stellt wiederum das zweite Teilprogramm dar und wird als SMART-Executable bezeichnet.

Abbildung 1 visualisiert die Arbeitsweise von SMART: Nachdem das Schadprogramm in einem Sandbox-System analysiert wurde (Schritt 1) bezieht der SMART-Generator die durchgeführten Veränderungen des Schadprogramms aus dem generierten Verhaltensbericht (Schritt 2). Mittels einer Parser-Klasse, die die Verwendung von unterschiedlichen Sandbox-Systemen ermöglicht, werden alle relevanten Informationen aus einem Bericht extrahiert und in einem Change-Bericht zusammengefasst (Schritt 3). Dieser beinhaltet Informationen über die vorgenommenen Änderungen des Dateisystems, der Windows-Registry und des Prozessmanagements. Für die Rücksetzung der manipulierten Registry Einträge werden, falls vorhanden, Standardwerte aus dem Windows-Betriebssystem extrahiert, unter dem der SMART-Generator aktuell ausgeführt wird.Die ausführbare SMART-Executable erstellt der SMART-Generator durch Zusammenfügen der Klassendateien zur

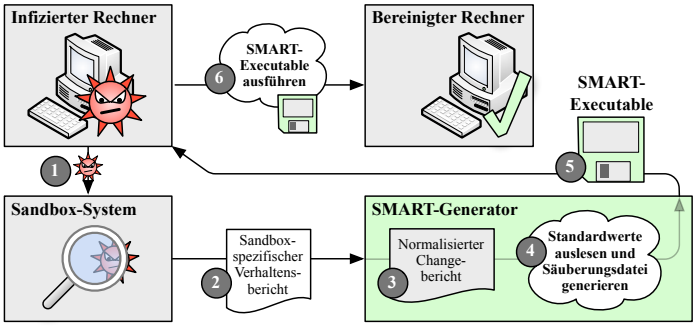

Abbildung 1: Schematische Darstellung der Arbeitsweise von SMART

Säuberung des Betriebssystems und des erzeugten Change-Berichts zu einem ausführbaren Java-Archiv (Schritt 4), das nun auf den infizierten Rechner kopiert werden muss (Schritt 5). Die SMART-Executable bezieht die Informationen der zu annullierenden Veränderungen aus dem übergebenen Change-Bericht. Vor dem Säuberungsvorgang (Schritt 6), wird noch eine Sicherung der zu löschenden Dateien in einem Archiv angelegt, da dies im Vergleich zur Ablage in einem herkömmlichen Ordner, einen weitestgehend sicheren und isolierten Aufbewahrungsort für Schadprogramme darstellt.

Den eigentlichen Säuberungsvorgang führt die SMART-Executable nach einer bestimmten Reihenfolge durch – der *Drei-Stufen-Säuberung*. In der 1. Stufe werden die gestarteten Prozesse terminiert. In der 2. Stufe erfolgt die Überschreibung der veränderten Registry-Einträge mit den Standardwerten. Den Abschluss bildet die 3. Stufe, in der die Dateisystemänderungen annulliert werden. Die Ausführungsreihenfolge der Säuberung spielt dabei eine wichtige Rolle. Falls die Prozesse etwa erst nach der Annullierung der Dateisystemänderungen terminiert werden, kann es passieren, dass das Schadprogramm versucht, die gelöschten Dateien wieder neu zu erstellen. Werden hingegen zuerst die Prozesse terminiert und anschließend die Dateien beseitigt, kann dieses Risiko reduziert werden.

Nach Abschluss des Säuberungsvorgangs überprüft SMART den Erfolg der Säuberung. Alle zu beseitigenden Änderungen aus dem Change-Report werden nochmals einer separaten Kontrolle unterzogen. Dabei soll festgestellt werden, ob die zuvor terminierten Prozesse immer noch beendet sind, die Registry-Einträge den Standardwert aufweisen und die gelöschten Dateien nicht wiederhergestellt wurden. Eine Übersicht des Säuberungserfolgs wird in einem Bericht im HTML-Format tabellarisch dargestellt.

4 Evaluation

Um bei jeder Ausführung eines Schadprogramms die gleichen Rahmenbedingungen zu schaffen, haben wir eine virtuelle Maschine als Basis-System eingesetzt. Als Virtualisierungssoftware haben wir *Parallels Desktop 5* verwendet und als Gastbetriebssystem wurde *Windows 7* mit allen aktuell verfügbaren Sicherheitsupdates installiert. Für die Evaluation

haben wir einen bekannten Referenzdatensatz von Schadsoftware verwendet [RTWH09], der eine grosse Bandbreite aktueller Schadprogramme abdeckt. Jedes Schadprogramm dieses Referenzdatensatzes ist mit einem Bezeichner von sechs unabhängigen AntiViren-Produkten versehen. Aus diesem Datensatz haben wir für 170 unterschiedliche Schadprogramme CWSandbox Reports erzeugt. Dies waren Schadprogramme aus den Familien *Adult-Browser*, *Allapple*, *Casino*, *Dorfdo*, *Flystudio*, *LdPinch*, *Magic Casino*, *Swizzor*, *Woikoiner*, *Bancos*, *Ejik*, *Podnuha*, *Rotator*, *Vapsup*, *Viking.DLL*, *Viking.DZ* und *Zhelatin*.

Zunächst befand sich unser Evaluations-System in einem sauberen Ursprungszustand. Dann wurde das System infiziert, indem das entsprechende Schadprogramm über ein extern eingebundenes Laufwerk eingeschleust und ausgeführt wurde. Nach einer Ausführungszeit von 5 Minuten wurde ebenfalls über das externe Laufwerk die SMART-Datei auf das System gebracht und gestartet, um den Säuberungsprozess durchzuführen. Nachdem SMART seinen Säuberungsbericht erzeugt hatte, was zugleich auch ein Indiz für den erfolgreichen Programmablauf darstellte, wurde das Gastsystem abschließend mit den i*Microsoft Security Essentials* gescannt, um zu überprüfen ob noch Reste des Schadprogramms zu finden waren. Das komplette Vorgehen wurde protokolliert und abschließend zusammen mit dem SMART-Report vom Gastsystem kopiert, bevor dieses wieder in den Ursprungszustand zurückgesetzt wurde. Ab hier begann der Kreislauf von vorne.

Einen Überblick über die Ergebnisse der 170 Säuberungsdurchläufe zeigt Tabelle 1. Dabei enthält die erste Spalte Informationen über den Erfolg der Säuberung und die zweite Spalte gibt an, wie viele Durchläufe diesem Ergebnis zuzurechnen sind. Die letzte Spalte enthält die Namen der entsprechenden Schadprogramm-Familien.

Tabelle 1: Säuberungsergebnisse basierend auf 170 unterschiedlichen Schadprogrammen

Säuberungserfolg	**Anzahl**	**Schadprogramm-Familien**
Erfolgreich	90	Adult-Browser, Allapple, Casino, Dorfdo, Flystudio, LdPinch, Magic Casino, Swizzor, Woikoiner
Erfolgreich, jedoch nicht alle Registry-Werte wiederhergestellt	71	Bancos, Ejik, Podnuha, Rotatot, Vapsup, Viking.DLL, Viking.DZ, Zhelatin
Nicht erfolgreich	9	Rotator

Die Änderungen von 90 Schadprogrammen konnten komplett rückgängig gemacht werden, d.h. es wurden alle durch das Schadprogramm erstellten Prozesse beendet, die erstellten Dateien wurden entfernt und alle Registryänderungen konnten annulliert werden. Die anschließende Virenscanner-Überprüfung fand auch keine schadhaften Fragmente mehr.

Bei 71 der 170 Schadprogramme konnten lediglich ein paar Registryänderungen nicht rückgängig gemacht werden – entweder weil das Schadprogramm nicht standardmäßig vorhandene Schlüssel löschte oder deren Werte modifizierte. Neun Schadprogramme der Familie *Rotator* brachten die SMART-Executable zum Absturz. Dementsprechend wurde diese nicht vollständig und ordnungsgemäß beendet, so dass diese Säuberungsversuche als erfolglos zu betrachten sind. Insgesamt konnten somit 94,7 % (161 von 170) der Infektionen komplett bzw. größtenteils durch die Benutzung von SMART beseitigt werden.

ISBN 978-3-88579-286-4

5 Einschränkungen

Falls ein Schadprogramm gelöscht wurde und kein Verhaltensbericht vorliegt, ist es nicht möglich, die durch dieses Schadprogramm vorgenommenen Änderungen mittels SMART zu annullieren. Dies liegt daran, dass kein Change-Bericht erstellt werden kann.

SMART erbt zudem das Problem der dynamischen Analyse, dass nur *ein* Ausführungspfad betrachtet wird. Dementsprechend können auch nur die während der Analyse beobachteten Änderungen rückgängig gemacht werden. Jedoch müssen diese nicht zwangsläufig den Änderungen auf dem ursprünglich infizierten Rechner entsprechen. Vor allem bei Schadprogrammen mit Analysesystem-Erkennung ist dies problematisch. Auch technologisch sehr fortschrittliche Schadprogramme lassen sich kaum in automatisierter Weise beseitigen, etwa Schadprogramme im Master Boot Record oder Rootkits. Der Prozentsatz solcher Schadprogramme am Gesamtaufkommen ist jedoch relativ gering [EFG+11].

Danksagung: Wir bedanken uns bei den anonymen Gutachtern für ihre hilfreichen Hinweise und Kommentare zu dieser Arbeit.

Literatur

[BMKK06] Ulrich Bayer, Andreas Moser, Christopher Kruegel und Engin Kirda. Dynamic Analysis of Malicious Code. *Journal in Computer Virology*, 2:67–77, 2006.

[CJK07] Mihai Christodorescu, Somesh Jha und Christopher Kruegel. Mining specifications of malicious behavior. In *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, ESEC-FSE '07, Seiten 5–14, New York, NY, USA, 2007. ACM.

[Coh90] F.B. Cohen. *A short course on computer viruses*. ASP Press, 1990.

[Cor11] Microsoft Corporation. What is System Restore? *http://windows.microsoft.com/en-US/windows7/What-is-System-Restore*, 2011. Zugriffsdatum: 02.04.2001.

[EFG+11] Markus Engelberth, Felix Freiling, Jan Göbel, Christian Gorecki, Thorsten Holz, Ralf Hund, Philipp Trinius und Carsten Willems. Das Internet-Malware-Analyse-System (InMAS). *Datenschutz und Datensicherheit - DuD*, 35:247–252, 2011. 10.1007/s11623-011-0061-7.

[KWZK10] Taesoo Kim, Xi Wang, Nickolai Zeldovich und M. Frans Kaashoek. Intrusion recovery using selective re-execution. In *Proceedings of the 9th USENIX conference on Operating systems design and implementation*, OSDI'10, Seiten 1–9, Berkeley, CA, USA, 2010. USENIX Association.

[RTWH09] Konrad Rieck, Philipp Trinius, Carsten Willems und Thorsten Holz. Automatic Analysis of Malware Behavior using Machine Learning. Bericht 18–2009, Berlin Institute of Technology, December 2009.

[WHF07] Carsten Willems, Thorsten Holz und Felix Freiling. Toward Automated Dynamic Malware Analysis Using CWSandbox. *IEEE Security and Privacy*, 5:32–39, March 2007.