Software-Whitelisting mit Microsoft AppLocker

Martin Reuter[1] · Daniel Loevenich[1] · Markus Ullmann[1,2]

[1]Bundesamt für Sicherheit in der Informationstechnik
{martin.reuter | daniel.loevenich | markus.ullmann}@bsi.bund.de

[2]Hochschule Bonn-Rhein-Sieg

Zusammenfassung

Moderne Unternehmen gestatten ihren Mitarbeiterinnen und Mitarbeitern zur Erledigung der geschäftlichen Aufgaben die Mitnahme privater IT-Endgeräte. Neben der Verbesserung von Motivation und Produktivität durch Bring-Your-Own-Device (BYOD) ergeben sich zusätzliche Bedrohungen für die IT-Infrastruktur der Unternehmen. Die mitgebrachten Geräte befinden sich in der Regel nicht im Eigentum des Unternehmens. Mit dem Internet wird längst nicht mehr das reine Abrufen von Webseiten über das Hyper Text Transfer Protocol verbunden. Im Zeitalter der Smartphones und Tablets werden kompakte Anwendungen (Apps) verwendet, mit Hilfe derer sich Daten vielseitig verarbeiten lassen und über öffentliche Netze (Internet) mit einem oder mehreren Servern synchronisieren lassen. Im BYOD-Szenario überschneiden sich betriebliche und private Nutzung. Eine Möglichkeit aus Sicht einer IT-Abteilung eines Unternehmens weiterhin Kontrolle über die Endgeräte im BYOD-Szenario zu behalten, ist der Einsatz eines konsequenten Software-Whitelisting auf den Endgeräten. Mit Hilfe eines Software-Whitelisting kann explizit die Ausführung von Software device-individuell auf eine entsprechende Whitelist beschränkt werden und damit eine unternehmensweite Sicherheitspolitik auf den Devices durchgesetzt werden. Wichtig aus Sicht einer IT-Infrastruktur ist allerdings, dass dieser Sicherheitsmechanismus vollständig greift und nicht umgangen werden kann. Microsoft stellt in seinen Windows Enterprise Versionen ab Windows 7 hierzu das Systemwerkzeug AppLocker zur Verfügung. Innerhalb dieses Beitrages werden neben den Grundlagen und Methoden des Software-Whitelistings sowie einer Einführung in das Produkt Microsoft AppLocker die Ergebnisse einer Sicherheitsanalyse veröffentlicht. Es wird aufgezeigt, dass diverse Schwachstellen in AppLocker ausgenutzt werden können, um letztlich beliebigen Code auf einem Device auszuführen und hiermit die Sicherheitspolitik eines Unternehmens vollständig zu unterlaufen. Basierend auf den Ergebnissen der Schwachstellenanalyse werden Handlungsempfehlungen abgeleitet und formuliert, um dennoch einen sicheren Einsatz von AppLocker als Werkzeug für das Software-Whitelisting weitestgehend zu gewährleisten.

1 Ausgangssituation / Fragestellung

Unternehmen und Organisationen überlassen ihren Mitarbeiterinnen und Mitarbeitern zunehmend mehr Freiheiten in der Nutzung unternehmenseigener IT im privaten Umfeld. Ebenfalls kann es ihnen gestattet sein, private (mobile) Endgeräte (Smartphones, Tablets, Notebooks) an die interne IT-Infrastruktur anzubinden. Diese Vorgehensweise hat sich unter dem Begriff BYOD - Bring-Your-Own-Device etabliert. In beiden Ansätzen soll die Produktivität und Zufriedenheit der Mitarbeiter gesteigert werden. Gleichzeitig steigt hierdurch allerdings die Bedrohungslage für ein Unternehmen. Die Endgeräte befinden sich in der Regel außerhalb der

P. Schartner · K. Lemke-Rust · M. Ullmann (Hrsg.) · D•A•CH Security 2015 · syssec (2015) 200-211.

Gewalt und Kontrolle einer IT-Abteilung / eines IT-Verantwortlichen. Mitgebrachte mobile Endgeräte sind nicht Eigentum des Unternehmens. Die Vielzahl unterschiedlicher Betriebssysteme auf den mobilen Endgeräten führt zu einer komplexen heterogenen Infrastruktur.

Mobile Endgeräte bewegen sich in der Regel in offenen Umgebungen (öffentliches WLAN, UMTS, etc.), in denen keine Kontrolle über mögliche Schutzmechanismen besteht. Hier sind Schutzmaßnahmen auf den mobilen Endgeräten jedoch von enormer Bedeutung.

Die unkontrollierte Ausführungsmöglichkeit von (unbekannter) Software und Programmcode stellt ein Risiko für eine gesamte IT-Infrastruktur dar. IT-Verantwortliche und Administratoren von Organisationen sind für die Umsetzung und Einhaltung der IT-Schutzziele Integrität, Vertraulichkeit und Verfügbarkeit verantwortlich. Ihnen muss daher ein verlässliches Tool bereitgestellt werden, um eine unternehmensweite Sicherheitspolitik umzusetzen und die Funktionssicherheit zu gewährleisten.

Unter der Funktionssicherheit eines Systems wird verstanden, dass „[…] nur solche Zustände angenommen werden, die zu keiner unautorisierten Informationsveränderung oder -gewinnung führen“ [Ecke14, S. 6]. Funktionssichere Systeme lassen sich durch sichere Systemmerkmale definieren.

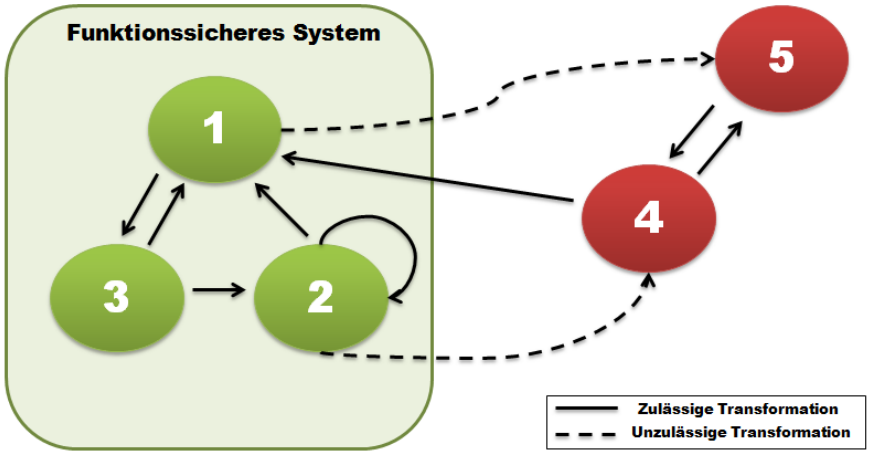

Abb. 1: Zustandsübergangsdiagramm funktionssicheres System

In Abbildung 1 wird ein funktionssicheres System mit den sicheren Systemzuständen 1, 2 und 3 abgebildet. Ein Übergang zu den unsicheren und nicht zulässigen Systemzuständen 4 und 5 außerhalb des funktionssicheren Systems darf nicht erfolgen. Die Transformationen werden in gängigen (Windows) Betriebssystemen durch Ausführung von Software und Code manuell und automatisiert bewirkt. Eine Software setzt sich in der Regel aus einer oder mehreren ausführbaren Dateien zusammen. Unter Windows gelten Dateien als ausführbar, wenn sie im Portable Executeable (PE) Dateiformat vorliegen. Ausführbare Dateien und Programmcode beeinflussen durch Transformationen den Gesamtzustand eines Systems. Unautorisierte Informationsveränderungen und -abflüsse werden u.a. durch Schadsoftware verursacht und sind zu unterbinden.

2 Grundlagen und Methoden

Die Ausführung von Software und Programmcode auf mobilen IT-Endgeräten innerhalb eines BYOD-Szenarios muss einheitlich geregelt sein und deren Umsetzung überwacht werden. A10.4.2 (Schutz vor mobiler Software) der ISO 27001 beschreibt die Notwendigkeit einer

Richtlinie für die Nutzung von mobiler Software. Die Richtlinie legt fest, dass „[..] nur vorab freigegebene und genehmigte Software genutzt werden darf“ [KeKl12, S. 47]. Neben organisatorischen Maßnahmen sollen technische Hilfsmittel und Werkzeuge zum Schutz beitragen.

2.1 Grundlegendes zum Software-Whitelisting

Positivlisten (engl. Whitelists) folgen dem Prinzip, ausschließlich zuvor explizit Erlaubtes zuzulassen. Im Umkehrschluss verweigern Negativlisten (Blacklists) alles, was zuvor explizit ausgeschlossen wurde [Bild13, S. 20-22]. In der Informationstechnik werden diese Prinzipien in unterschiedlichen Bereichen eingesetzt. Das Regelwerk einer Firewall nach dem Whitelisting-Verfahren legt fest, dass bekannte Kommunikationsteilnehmer ausschließlich über benötigte und erwünschte Kommunikationskanäle miteinander in Verbindung treten können. Im Blacklisting-Verfahren werden nicht erwünschte Kommunikationsteilnehmer und Kanäle ausgeschlossen.

Software-Whitelisting lässt sich als Methode dazu einsetzen, das Risiko durch unbekannte ausführbare Software- und Programmcodes zu minimieren. Regelwerke bestimmen, welche Anwendungen durch den Benutzer und das System ausgeführt werden dürfen. Ausschließlich bekannte und vorab definierte Software und Programmcodes lassen sich auf dem Endgerät ausführen. Neben dem Whitelisting-Verfahren existiert auch ein Backlisting-Verfahren. Bei diesem wird explizit festgelegt, welche Software und Programmcode nicht auf einem System ausgeführt werden dürfen. Die Menge der schadhaften Software ist zwar endlich, jedoch ist es unmöglich, die gesamte Menge in Erfahrung zu bringen; insbesondere bliebe die Angriffsfläche für neue, noch nicht bekannte, Schadsoftware bestehen.

2.2 Prüf- und Kontrollverfahren

Ein Werkzeug, das Software-Whitelisting umsetzt, muss an Hand von festgelegten Kriterien durch Prüf- und Kontrollverfahren zuverlässig erkennen können, welche Software ausgeführt werden darf. Ausführbare Dateien müssen zunächst als solche durch das Betriebssystem erkannt werden, damit der darin enthaltene Programmcode ausgeführt werden kann. Unter Windows liegen derartige Binärdateien im Portable Executeable Format (PE-Datei) vor. Produkte, die das Software-Whitelisting umsetzen, müssen ausführbare Dateien entsprechend der PE/COFF-Spezifikation [Micr13a] ebenfalls als solche identifizieren können. Softwareentwickler können Maßnahmen ergreifen, um Authentizität und Integrität einer kompilierten Software zu gewährleisten. Durch die beiden Schutzziele wird sichergestellt, dass die Software tatsächlich vom Entwickler stammt und nicht durch unberechtigte Dritte verändert wurde. Digital signierte PE-Dateien liegen unter Windows im sogenannten Windows Authenticode Portable Executeable Signature Format vor. Authenticode basiert auf Public Key Cryptography Standards (PKCS) #7 sowie X.509 v3 Zertifikaten und sichert Authentizität und Integrität zu [Micr08].

Alternativ zum Signaturverfahren mit Authenticode lassen sich über ausführbare Dateien einer Software Dateihashwerte berechnen, welche anschließend im Regelwerk hinterlegt werden. Bei einer Ausführung werden die Dateihashwerte abgeglichen.

Möglich sind auch Datei- und Verzeichnispfadangaben zu ausführbaren Dateien.

2.3 Code- und Software-Signierung

Damit Authentizität und Integrität einer Software verifiziert werden können, ist es erforderlich, dass vorab alle ausführbaren Dateien durch den Softwareherausgeber digital signiert wurden. Herstellern und Softwareentwicklern stehen hierfür verschiedene Methoden und Verfahren in Microsoft Betriebssystemen zur Verfügung [Micr07]. Voraussetzung für das Signieren von Software und Programmcode ist ein gültiges Softwareherausgeberzertifikat (SPC). Über die Daten einer ausführbaren Datei wird mit Hilfe einer Hashfunktion ein Hashwert errechnet. Durch die Signaturfunktion wird mittels privaten Schlüssels des Herausgebers aus dem Hashwert eine Signatur erzeugt. Diese wird in PE-Dateien üblicherweise eingebettet oder in Katalogdateien vorgehalten.

In der Praxis sind die Werkzeuge (z.B. SignTool) zur Signierung bereits in der Entwicklungsumgebung (hier: Visual Studio) enthalten oder über das Windows Software Development Kit (SDK) beziehbar.

2.4 Vorgehensweise

Software-Whitelisting lässt sich als ein kontinuierlicher Prozess abbilden, der sich in fünf Phasen unterteilt:

Abb. 2: Umsetzungsprozess von Software-Whitelisting

Innerhalb eines Audits wird festgehalten, welche ausführbaren Inhalte einer Software benötigt werden. In Windows-Betriebssystemen zählen hierzu beispielsweise die assoziierten Dateiformate EXE und DLL. Das Ergebnis des Audits gibt Aufschluss darüber, welche ausführbaren Inhalte tatsächlich zur täglichen Arbeit der Mitarbeiterinnen und Mitarbeiter eines Unternehmens benötigt werden und für einen stabilen Betrieb der Software erforderlich sind. Aus den Ergebnissen des Audits lassen sich im nächsten Schritt einzelne Regeln ableiten und zusammenfassen. Mit Aktivierung des Software-Whitelistings werden die festgelegten Regeln auf den Systemen durchgesetzt. Alle Entscheidungen über die Ausführung oder das Blockieren von ausführbaren Dateien werden im Monitoring zusammengeführt. IT-Verantwortliche und Administratoren können das Monitoring dazu einsetzen, um die Kompatibilität zugelassener Software zu verbessern oder Sicherheitsvorfälle zu erkennen.

2.5 Annahmen und Grenzen

Durch Einsatz von Software-Whitelisting ist es möglich, die annehmbaren Zustände eines Systems zu definieren. Hierbei ist jedoch die Erfüllung und Einhaltung bestimmter Anforderungen und Maßnahmen erforderlich. Software-Whitelisting unterliegt zudem bestimmten Grenzen. Ohne ein Verfahren zum Software-Whitelisting kann die Funktionssicherheit einer IT-Infrastruktur nicht gewährleistet werden.

Ausführbare Dateien einer Software müssen zuverlässig erkannt werden. Hierzu greift ein Werkzeug für Software-Whitelisting auf bestimmte Prüf- und Kontrollverfahren zurück, mit denen sich ausführbare Dateien als solche identifizieren lassen. Darüber hinaus sollte die Integrität und Authentizität der ausführbaren Dateien festgestellt werden.

Schwachstellen innerhalb von Public-Key-Infrastrukturen oder eines Hashwertverfahrens können für Angriffe missbraucht werden, die das Ziel verfolgen, nicht zugelassene ausführbare Dateien auf einem geschützten Zielsystem auszuführen. Beispielsweise können gestohlene Zertifikate namhafter Softwareherausgeber dazu eingesetzt werden, um Schadsoftware zu signieren.

Die Sicherheit und Funktion einer Software, die durch ein Whitelisting-Verfahren auf Systemen zugelassen wird, muss vorab geprüft sein, um über sämtliche Zustandsübergänge Kenntnis zu erlangen. Sicherheits- und Funktionstests können beispielsweise mit Backbox-Tests, Fuzzing oder Schwachstellenscans durchgeführt werden. Sofern der Quelltext einer Software zur Verfügung steht, bieten sich auch statische Code-Analysen an.

Die Regelwerke des Whitelisting-Werkzeuges müssen vollständig sein und dürfen keiner nicht zugelassenen Software ermöglichen, auf den Systemen ausgeführt zu werden.

3 Microsoft AppLocker

Im Enterprise-Bereich der Microsoft Windows Betriebssysteme (seit Windows 7) ist ein integriertes Software-Whitelisting- und Blacklisting-Verfahren integriert. Mit Microsoft AppLocker, dem Nachfolger der Softwareeinschränkungsrichtlinien, lässt sich ein betriebssystemweites Software-Whitelisting für ausführbare Dateien und Programmcode umsetzen. Das Produkt bietet Regelwerke für ausführbare Dateien (EXE, COM), Windows-Installer (MSI), Skripte (Stapelverarbeitung, PowerShell), Laufzeitbibliotheken (DLL) und App-Pakete (AppX).

Microsoft AppLocker ist ein fester Bestandteil der Microsoft-Betriebssysteme Windows 7/8, Server 2008/2012. Zu beachten ist, dass AppLocker in den Client-Betriebssystemen Windows 7 und 8 jeweils in den Ausführungen Ultimate (Windows 7) und Enterprise aktiv eingesetzt werden kann.

Beworben wird AppLocker durch Microsoft mit den folgenden Anwendungsszenarien [Micr09a]:

- Anwendungsbestand
- Schutz vor unerwünschter Software
- Lizenzierungskonformität
- Softwarestandardisierung
- Verbesserte Verwaltbarkeit

Architektur und Funktionsweise von AppLocker sind in der öffentlichen Herstellerdokumentation nicht detailliert beschrieben. Sie beschränkt sich auf Konfiguration und Betrieb.

3.1 Aufbau

AppLocker lässt sich in fünf Komponenten einteilen: Einer Kernkomponente, zwei administrativen Komponenten, einer Konfigurationsspeicherkomponente und einer Überwachungskomponente für das Monitoring.

Die Kernkomponente ist in Form eines Kernel-Mode Treibers (AppID) und eines Windows-Dienstes (AppIDSvc) realisiert und trägt dafür Sorge, dass die Regeln kontinuierlich durchgesetzt werden. Für die Administration von AppLocker wird eine grafische Administrationsoberfläche über die Microsoft Management Console (MMC) und die Gruppenrichtlinienverwaltung bereitgestellt. Alternativ kann auf AppLocker CmdLets der Windows-PowerShell zurückgegriffen werden. Die Konfiguration sowie das Regelwerk von AppLocker werden innerhalb der lokalen Windows-Registrierungsdatenbank unter der Security Descriptor Definition Language (SDDL) sowie als binäre Access Control Entries (ACE) hinterlegt. Das Monitoring innerhalb der Überwachungskomponente erfolgt durch die Windows-Ereignisanzeige und ist unterhalb der Anwendungs- und Dienstprotokolle eingegliedert.

3.2 Regelwerke

AppLocker verfügt über fünf Regelwerke, die an ein assoziiertes Dateiformat geknüpft sind: Ausführbare Regeln (EXE, COM), Windows-Installer-Regeln (MSI, MSP), Skriptregeln (PS1, BAT, CMD, VBS, JS), DLL-Regeln (DLL) und App-Paketregeln (APPX). Jedes der Regelwerke lässt sich unabhängig von den anderen Regelwerken aktivieren, deaktivieren oder in dem Modus „Regelüberwachung“ konfigurieren. Im Gegensatz zum Modus „Regelerzwingung“ werden im Modus „Regelüberwachung“ ausschließlich die nach dem Regelwerk bestimmten zugelassenen oder blockierten Dateien protokolliert. Die blockierten Dateien werden weiterhin ausgeführt.

Die technische Referenz zu AppLocker [Micr12] lässt irrtümlicherweise darauf schließen, dass AppLocker lediglich die dort spezifizierten Dateiformate unterstützt. In den Dialog-Fenstern der grafischen Administrationskomponente werden die einzulesenden Dateien nach den assoziierten Endungen gefiltert. Trotz des Filters lassen sich jedoch alle ausführbaren Dateien auswählen.

3.3 Regelarten

Je nach Regelwerk werden für die Regelerstellung bis zu drei verschiedene Regelarten angeboten:

- **Herausgeberregeln:** Ausführbare Dateien werden anhand der digitalen Signatur (Authenticode-Signatur) eines Softwareherausgebers identifiziert. Die Authenticode-Signatur kann dabei in die ausführbare Datei eingebettet oder über eine Katalogdatei sichergestellt sein. Der Vorteil von Herausgeberregeln besteht darin, dass die gesamte Software und ihrer ausführbaren Bestandteile eines bestimmten Softwareherausgebers innerhalb eines Regeleintrages zusammengefasst werden können. Dies hält den administrativen Aufwand gering und sorgt für eine bessere Übersichtlichkeit. Darüber hinaus ist es möglich, Herausgeberregeln zu generalisieren, indem der Regel weitere Bedingungen (Sekundärbedingungen) über Produkt-, Datei- oder Versionsnummern hinzugefügt werden.
- **Dateihashregeln:** Bei der Regelerstellung werden Hashwerte über ausführbare Dateien berechnet und im Regelwerk hinterlegt. Vor Ausführung einer Datei wird ihr Dateihash durch AppLocker erneut berechnet und mit den Einträgen in der Datenbank abgeglichen.
- **Pfadregeln:** In einer Pfadregel kann ein Verzeichnis angegeben werden, in dem alle ausführbaren Dateien uneingeschränkt ausgeführt werden können. Die Pfade können auch auf einzelne Dateien verweisen.

3.4 Installation und Einrichtung

Bei einer Aktivierung von AppLocker ist es erforderlich, dass der Anwendungsidentitätsdienst (AppIDSvc) auf allen Systemen, die durch AppLocker geschützt werden sollen, automatisch gestartet wird. In einer zentral administrierten Windows-Domäne sollte der Dienst über die Gruppenrichtlinien permanent (erzwungen) gestartet werden.

Wird die Regelerzwingung bzw. -überwachung aktiviert, erfolgt eine Restriktion bzw. Überwachung durch AppLocker erst, sobald sich mindestens ein Eintrag im jeweiligen Regelwerk befindet.

Für alle Regelwerke lassen sich Standardregeln erzeugen. Diese Regeln sollen als Vorlage dienen und sind primär zu Testzwecken vorgesehen.

Im Microsoft TechNet befindet sich ein Handbuch und Schritt-für-Schritt Anleitungen zu AppLocker [Micr09b].

3.5 Sicherheitseinschränkungen und Grenzen

Microsoft AppLocker verfügt über Sicherheitseinschränkungen, auf die der Hersteller in seiner Produktspezifikation hinweist [Micr11]:

AppLocker überwacht ausschließlich ausführbare Dateien, deren Ausführung innerhalb des Win32-Subsystems erfolgt. Subsysteme wie POSIX werden nicht in die Überwachung einbezogen. Emulierte 16-bit Anwendungen, innerhalb der Windows NT Virtual DOS Machine, werden ebenfalls nicht überwacht. Das POSIX Subsystem sowie die Virtual DOS Machine können als Einfallstor für Schadsoftware ausgenutzt werden. AppLocker kann die Ausführung von Skripten in Skriptsprachen von Drittanbietern nicht kontrollieren. Ist eine Skriptumgebung eines Drittanbieters durch die Regelwerke freigegeben, lassen sich innerhalb dieser Umgebung sämtliche Skripte ohne eine weitere Kontrolle ausführen. Regelwerke in AppLocker können gezielt umgangen werden, wenn entsprechende Flags im Funktionsaufruf von Programmen gesetzt werden.

AppLocker kann durch Ausnutzen der sogenannten „Time-of-Check Time-of-Use“ - Schwachstelle umgangen werden [Whit13]. Zwischen Prüfzeitpunkt und Verwendung einer ausführbaren Datei können ihre Daten abgeändert werden, ohne dass dies von AppLocker registriert wird. Zugelassene Dateien können innerhalb eines Man-in-the-Middle Szenarios manipuliert und anschließend ausgeführt werden.

4 AppLocker Schwachstellenanalyse

Die Ergebnisse einer durchgeführten Schachstellenanalyse zeigen auf, dass AppLocker über weitere, nicht durch den Hersteller angegebene, Sicherheitseinschränkungen verfügt [Reut14]. Im Folgenden sollen die Ergebnisse dargestellt und die Auswirkungen der Schwachstellen bewertet werden.

4.1 Fehlerhafte Verifikation von PE-Dateien

In der Vergangenheit hat eine Schwachstelle der WinVerifyTrust-Funktion in Windows Betriebssystemen [Nist13] dazu geführt, dass Manipulationen an gültig signierten PE-Dateien durch das Betriebssystem nicht erkannt werden konnten. Zur Hashwertberechnung einer PE-Datei wird die eingebettete Signatur nicht miteinberechnet. Durch die Schwachstelle innerhalb

der WinVerifyTrust-Funktion werden die Signaturbestandteile einer PE-Datei falsch überprüft. Ein veröffentlichtes Windows-Update schließt die Schwachstelle in der WinVerifyTrust-Funktion im Betriebssystem. AppLocker verwendet jedoch nach wie vor eine fehlerhafte Verifikationsfunktion, sodass manipulierte PE-Dateien weiterhin ausgeführt werden können [Reut14, S. 39-41].

Durch das Ausnutzen der Schwachstelle innerhalb der Verifikationsfunktion von AppLocker ist es möglich, beliebige Daten in eine signierte PE-Datei einzuschleusen, ohne dabei die Gültigkeit der Signatur zu beeinträchtigen. Die einzuschleusenden Daten folgen unmittelbar der eingebetteten Signatur und werden nicht mit in die Dateihashwertbildung einbezogen.

Potentieller Schadcode kann hierdurch an den AppLocker Regelwerken vorbei zur Ausführung gebracht werden. Zur Veranschaulichung können mit Hilfe der disitools von Didier Stevens [Stev14] digital signierte PE-Dateien manipuliert (siehe Abbildung 3) und anschließend auf einem durch AppLocker geschützten System ausgeführt werden.

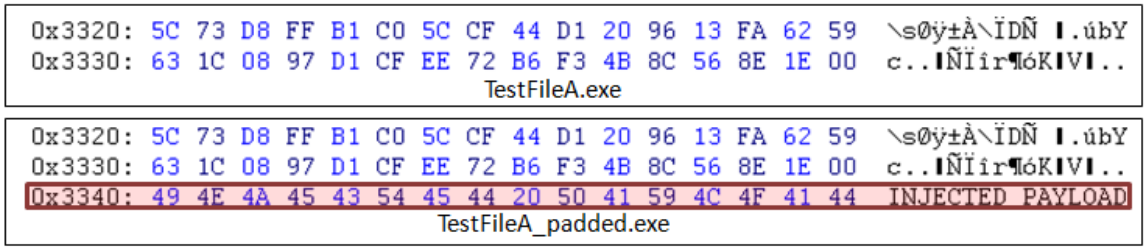

Abb. 3: Auszug einer manipulierten PE-Datei

Einige Softwarehersteller haben den Fehler innerhalb der WinVerifyTrust-Funktion bewusst genutzt, um Konfigurationsdaten in einen Windows-Installer einzubinden [Stev13].

Somit lassen sich manipulierte PE-Dateien weiterhin durch AppLocker ausführen, obwohl ihre Signatur ungültig ist.

4.2 Windows-Dienste

Windows-Dienste sind langlebige Anwendungen ohne Benutzeroberfläche, welche innerhalb einer eigenen Sitzung im Hintergrund des Systems ausgeführt werden. Über Schnittstellen kann eine vom Benutzer ausgeführte Anwendung mit Windows-Diensten interagieren. Zur Installation von Windows-Diensten sind administrative Rechte erforderlich.

Windows-Dienste lassen sich durch AppLocker nicht überwachen und einschränken. Hierdurch lässt sich unerwünschte Software und Schadcode an AppLocker vorbei ausführen [Reut14, S. 41-43]. Eine Überprüfung von digitalen Signaturen und Dateihashwerten von ausführbaren Dateien der Windows-Dienste erfolgt nicht. Durch manipulierte Windows-Dienste lässt sich Schadsoftware ausführen. Des Weiteren lassen sich neue Prozesse erstellen, deren Ausführung ebenfalls nicht der Kontrolle durch AppLocker unterliegen.

Bereits installierte und somit bestehende Windows-Dienste lassen sich manipulieren. Eine Manipulation an den ausführbaren Dateien von Diensten lässt sich durch AppLocker nicht erfassen. Angreifer können diese verändern oder austauschen.

4.3 Systemkontext: NT-Autorität\System

Die AppLocker Schutzfunktion erfasst keine Anwendungen, die unter dem Sicherheitskontext „NT-Autorität\SYSTEM“ (höchst-privilegierter Sicherheitskontext in Windows-Betriebssystemen) ausgeführt werden. Eine Anwendungssteuerung erfolgt daher ausschließlich im Benutzerkontext [Reut14, S. 46f].

Durch unautorisierte Privilegienerweiterung (Privilege Escalation) können Schadsoftware und -code beliebig ausgeführt werden.

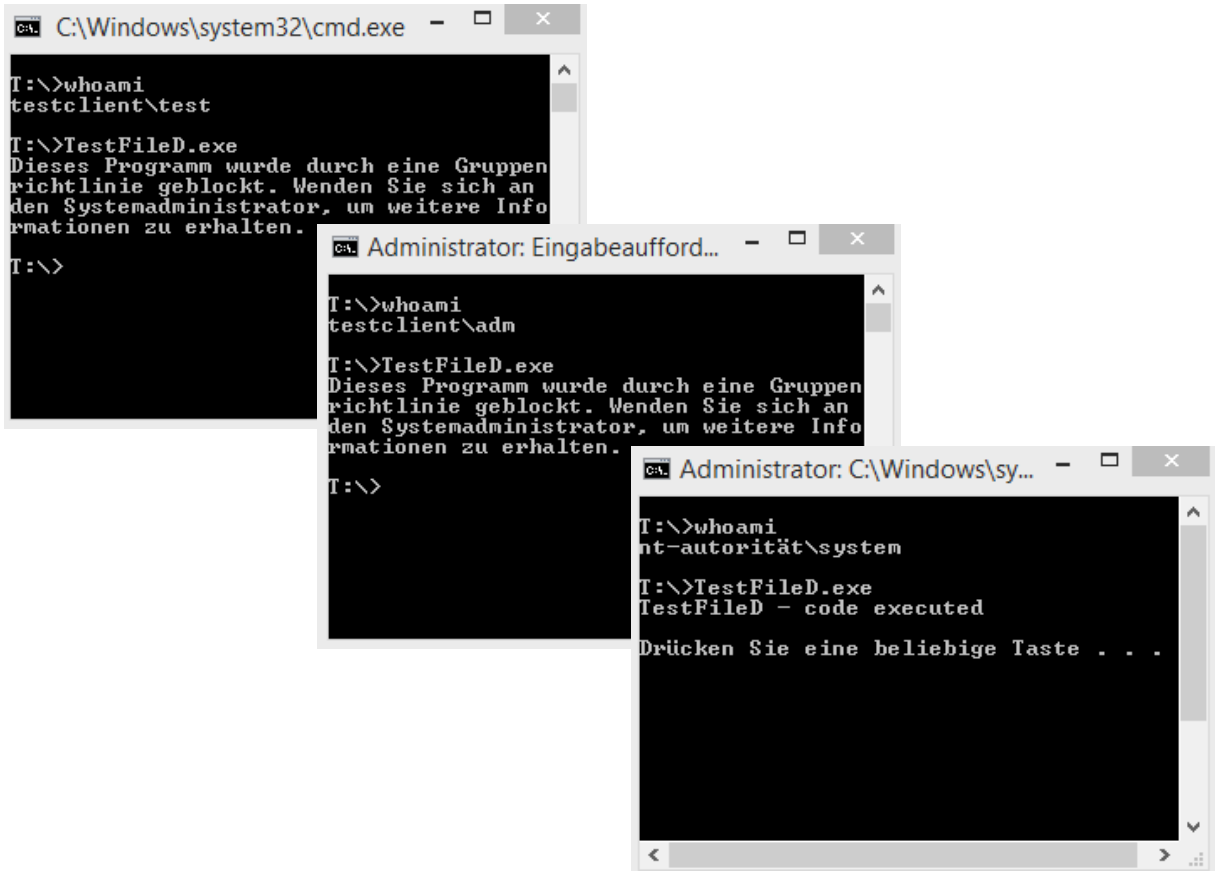

Abb. 4: Ausführung einer Testdatei innerhalb verschiedener Sicherheitskontexte

Abbildung 4 zeigt drei Kommandozeilen, die mit unterschiedlichen Sicherheitskontexten ausgeführt werden (links: Benutzer, Mitte: Administrator, rechts: lokales System). Für die Testdatei (TestFileD.exe) existiert kein Eintrag im Regelwerk in AppLocker, sodass diese nicht ausgeführt werden darf. Ein expliziter Blacklist-Eintrag im Regelwerk von AppLocker, der die Ausführung der Testdatei für den „Benutzer“ NT-Autorität\System verbietet, führt ebenfalls zu dem in Abbildung 4 dargestellten Verhalten.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) hat die ermittelten Schwachstellen begutachtet und eine Cyber-Security Empfehlung (BSI-CS 117) mit Handlungsempfehlungen für IT-Verantwortliche und Administratoren veröffentlicht.

5 Schlussfolgerungen und weitere Dokumente

Grundsätzlich ist die Funktionssicherheit in einer IT-Infrastruktur im BYOD-Szenario ohne ein Software-Whitelisting nicht gegeben bzw. nur mit enormem Aufwand realisierbar. Die Überprüfung der zuzulassenden Software durch ein Whitelisting-Verfahren stellt eine wichtige Voraussetzung für die Funktionssicherheit einer IT-Infrastruktur dar. Alle Funktionen und Schnittstellen einer durch das Whitelisting zuzulassenden Software müssen bekannt und entsprechend als unschädlich deklariert sein.

Bei Verwendung von Microsoft AppLocker sind technische Einschränkungen zu beachten und entsprechende Maßnahmen zu treffen. Die technischen Einschränkungen ergeben sich aus den genannten Sicherheitseinschränkungen der Produktspezifikation sowie den Ergebnissen aus der Schwachstellenanalyse. Ihnen kann entgegen gewirkt werden, wenn bestimmte Voraussetzungen sowie Maßnahmen im organisatorischen und technischen Umfeld eingehalten werden.

- **Organisatorische Voraussetzungen:** Innerhalb von Organisationen ist sicherzustellen, dass eine vollständige Administrationsgewalt über die IT-Systeme besteht. Die Clients sollten zudem administriert sein, um Redundanzen und Fehler in der Konfiguration zu vermeiden. Ausführbare Dateien von Windows-Diensten sollten vor Manipulation durch Berechtigungen im Dateisystem besonders geschützt sein.
- **Technische Voraussetzungen:** Damit ein unberechtigtes Abschalten von AppLocker verhindert wird, müssen Administrationskonten strikt von Benutzerkonten getrennt sein. Auf die in AppLocker vordefinierten Standardeinstellungen und -regeln sollte in Produktivumgebungen verzichtet werden, da diese Pfadregeln enthalten. Ausführbare Dateien, die innerhalb eines Verzeichnispfads angegeben sind, können uneingeschränkt ohne Integritäts- und Authentizitätsprüfung ausgeführt werden. Die Systempartition sollte vor Manipulationen durch eine Festplattenverschlüsselung geschützt sein. Eine Verschlüsselung des Netzwerkverkehrs verhindert Man-in-the-Middle Angriffe [Whit13]. Innerhalb des abgesicherten Modus ist AppLocker nicht aktiv. Der Zugang zum abgesicherten Modus sollte daher ausschließlich für IT-Verantwortliche und Administratoren ermöglicht werden.
- **Organisatorische Maßnahmen:** Die Planung des Regelwerks ist von wesentlicher Bedeutung für einen sicheren Einsatz von Software-Whitelisting mit AppLocker. Anhand der Regelwerke wird entschieden, welche Software durch das Betriebssystem ausgeführt werden darf. Bei der Planung sollte nach Möglichkeit geprüft werden, ob signierte Software innerhalb von Herausgeberregeln zugelassen werden kann. Alternativ eignen sich Dateihashwertregeln. Auf Pfadregeln sollte verzichtet werden. Referenzcomputer bilden einen Produktivclient identisch ab. Auf diesem können Regelwerke erstellt und getestet werden. Nach Softwareupdates ergeben sich in der Regel Änderungen im Regelwerk. Ist eine Softwareverteilung vorhanden, sollte die Regelerstellung in den Prozess der Software-Paketierung integriert werden. So kann sichergestellt werden, dass die zu verteilende Software funktionsfähig ist und alle erforderlichen Komponenten aufgeführt werden dürfen.
- **Technische Maßnahmen:** Zur Regelerstellung können die Windows PowerShell AppLocker CmdLets behilflich sein. Diese lassen eine automatisierte Regelerstellung zu. Vom Hersteller bereitgestellte Hotfixes und Updates für AppLocker sollten auf durch AppLocker geschützten Systemen installiert sein. Strategische Kombinationen von Regeln (z.B. Dateihashwertregel / Pfadregel) können die Sicherheit verbessern. Software,

die herstellerseitig nicht signiert ist, kann nachträglich mit eigenem Zertifikat digital signiert werden. Sollten Komponenten wie das POSIX-Subsystem oder die Unterstützung von 16-Bit Anwendungen nicht benötigt werden, sind diese zu deaktivieren. Bei Skriptumgebungen und -sprachen von Drittanbietern sind andere Sicherheitsmaßnahmen zu treffen (z.B. Virtualisierung, Sandboxing, etc.).

Neben dem Einsatz von Software-Whitelisting sind weitere Maßnahmen auf den Systemen (mobile Endgeräte) und der IT-Infrastruktur erforderlich. Hierzu zählen beispielsweise: Antiviren-Produkte, Firewall-Lösungen, Systemupdates, Intrusion Detection Systeme (IDS) und eine Überwachung der Systemschnittstellen (USB, FireWire, etc.).

Trotz Erfüllung und Umsetzung der Voraussetzungen und Maßnahmen besteht beim Einsatz von AppLocker ein Restrisiko, dass Endgeräte in undefinierte Zustände gelangen. Die Schwachstelle in der Verifikationsfunktion von AppLocker besteht weiterhin, sodass nach wie vor manipulierte PE-Dateien mit gültiger Signatur ausgeführt werden können. Dies kann insbesondere dann eintreten, wenn Inhalt und Funktion einer Software nicht vollständig analysiert und in Erfahrung gebracht werden konnte.

Eine Marktbetrachtung zeigt, dass zum aktuellen Zeitpunkt ein Software-Whitelisting nicht einheitlich zum Standardumfang gängiger Betriebssysteme zählt. Insbesondere im Umfeld von Smartphones und Tablets stellen die Hersteller (Google, Apple, Microsoft) werksseitig keine Werkzeuge für ein App-Whitelisting bereit. In Windows-Umgebungen lässt sich AppLocker ausschließlich auf Windows 7 Ultimate, Enterprise sowie Windows 8 Enterprise aktivieren. Aus diesem Grund können Smartphones mit dem Windows-Phone Betriebssystem und Tablets mit Windows RT nicht mit überwacht werden. Um AppLocker auch auf Tablets einsetzen zu können, ist demnach eine Installation von Windows 8 Enterprise erforderlich.

Microsoft stellt für AppLocker einen Design Guide zur Verfügung, welcher IT-Verantwortliche und Administratoren bei der Planung und Konfiguration unterstützt [Micr13b].

Für den Einsatz von AppLocker stellt das BSI im IT-Grundschutz-Katalog die Maßnahme M 4.419 „Anwendungssteuerung ab Windows 7 mit AppLocker“ und die Cyber-Security Empfehlung BSI-CS 117 „Sicherer Einsatz von AppLocker“ bereit.

Literatur

[Bild13] C. Bildsten: Application Whitelisting: Smartphones in High Security Environments, Master-Thesis an der Universität Linköping (2013)

[CaCa09] H. Carvey, E. Casey, Windows Forensic Analysis, Syngres Publishing, Burlington (2009)

[Ecke14] C. Eckert: IT-Sicherheit, Konzepte - Verfahren - Protokolle, Oldenburg Verlag, München, 9. überarbeitete Auflage (2014)

[KeKl12] H. Kersten, G. Klett: Mobile Device Management, Hüthig Jehle Rehm GmbH, Heidelberg (2012)

[Micr07] Code-Signing Best Practices (2007)
https://msdn.microsoft.com/en-us/library/windows/hardware/dn653556%28v=vs.85%29.aspx

[Micr08] Windows Authenticode Portable Executable Signature Format (2008)
https://msdn.microsoft.com/en-us/gg463180.aspx

[Micr09a] Verwendungsszenarien für AppLocker-Richtlinien (2009)
https://technet.microsoft.com/de-de/library/ee424357%28v=ws.10%29.aspx

[Micr09b] AppLocker Step-by-Step Guide (2009)
https://technet.microsoft.com/de-de/library/dd723686%28v=ws.10%29.aspx

[Micr11] Security Considerations for AppLocker, Microsoft Corporation (2011)
http://technet.microsoft.com/de-de/library/ee844118%28v=ws.10%29.aspx

[Micr12] AppLocker Overview (2012)
https://technet.microsoft.com/en-us/library/hh831440.aspx

[Micr13a] Microsoft PE and COFF Specification (2013)
https://msdn.microsoft.com/en-us/library/gg463119.aspx

[Micr13b] AppLocker Design Guide, Microsoft Corporation (2013)
http://www.microsoft.com/en-us/download/details.aspx?id=40330

[Nist13] Vulnerability Summary for CVE-2013-3900
https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-3900

[Reut14] M. Reuter: Analyse und Bewertung von Software-Whitelisting in Microsoft Betriebssystemen für mobile Clients, Bachelor-Thesis an der Hochschule Bonn-Rhein-Sieg, Sankt Augustin (2014)

[RuSI12] M. Russinovich, D. Solomon, A. Ionescu: Windows Internals Part 1, Microsoft-Press, Washington (2012)

[Stev13] D. Stevens: Digital graphology: It's all in the signature. In: (IN)SECURE Issue 39, S. 60-63 (2013)

[Stev14] D. Stevens: Disitool
http://blog.didierstevens.com/programs/disitool/

[Whit13] O. Whitehouse (NCC Group): Bypassing Windows AppLocker using a Time of Check Time of Use Vulnerability
https://www.nccgroup.trust/media/481134/2013-12-04_-_ncc_-_technical_paper_-_bypassing_windows_applocker-2.pdf