Automatisierte Erkennung von Sicherheitslücken mittels CVE, CPE und NVD

Konstantin Knorr[1] · Matthias Scherf[2] · Manuel Ifland[3]

[1]Hochschule Trier
knorr@hochschule-trier.de

[2]Universität Trier
scherf@uni-trier.de

[3]Siemens AG
manuel.ifland@siemens.com

Zusammenfassung

Privatpersonen und Unternehmen setzen in ihren IT-Systemen vermehrt Softwareprodukte von Drittanbietern ein, in denen Sicherheitslücken auftreten können. Die Hersteller dieser Software reagieren darauf mit einer Schwachstellenmeldung und häufig mit einem Patch oder einem Update zur Behebung. 2015 wurden um die 6.000 solcher Sicherheitslücken mit einer offiziellen CVE-Nummer versehen und in der NVD erfasst. Insbesondere bei umfangreichen IT-Systemen stellt sich die Frage, wie sich die Erkennung und Bewertung von bekannten Sicherheitslücken automatisieren lässt. CVE wird durch Standards wie CPE, CVSS und CWE ergänzt, um so detailliertere Informationen zu Schwachstellen zu erfassen. Das vorliegende Papier untersucht anhand eines Testsystems und eines dreimonatigen Testlaufs, wie stark sich die Erkennung verwundbarer Software mit Hilfe von Standards automatisieren lässt. Für eine Auswertung der Schwachstellen wurde das „Software Vulnerability Tool“ entwickelt. Die Ergebnisse zeigen, dass die Schwachstellenerkennung größtenteils automatisiert werden kann, dass jedoch in einigen Fällen manuelle Anpassungen vorgenommen werden müssen. Dies ist u. a. auf die Komplexität der Standards, Abstimmungsprobleme zwischen den Standards, auf abweichende Herstellerpraktiken sowie auf Mängel beim Erstellen der Schwachstellenmeldungen zurückzuführen.

1 Einleitung

Privatpersonen und Unternehmen setzen in IT-Systemen in großem Maße Softwareprodukte von Drittanbietern ein. Auf diese Weise lässt sich mit geringem Aufwand nahezu jede Art von benötigter Funktionalität realisieren. Beispiele hierfür sind Betriebssysteme, Internet-Browser, Office Anwendungen oder Media Player. Derartige Softwareprodukte können allerdings von Sicherheitslücken betroffen sein, die den sicheren Betrieb der IT-Systeme sowie der gesamten IT-Infrastruktur beeinträchtigen können. Aus diesem Grund ist ein möglichst hoher Grad an Automatisierung bei der Erkennung von bekannten Sicherheitslücken in Softwareprodukten, für die in der Regel bereits ein Patch zur Behebung existiert, sinnvoll.

P. Schartner (Hrsg.) · D•A•CH Security 2016 · syssec (2016) 328-338.

Ziel dieser Arbeit ist die Beantwortung der Frage, wie gut sich verfügbare Standards und Datenbanken als Informationsgrundlage für die automatisierte Erkennung von bekannten Sicherheitslücken eignen. Eine Auflistung und Beschreibung der für dieses Papier relevanten Standards befindet sich in Kapitel 2.1. Für eine Auswertung der Sicherheitslücken wurde das „Software Vulnerability Tool“ (SVT) entwickelt. Um eine möglichst realistische Beurteilung bezüglich der Qualität und der Quantität zu erhalten, werden die SVT-Ergebnisse mit den Ergebnissen einer bereits auf dem Markt befindlichen Software mit vergleichbarer Funktionalität verglichen. Unser Vorgehen ist nützlich, wenn zusätzlich Informationen wie z.B. die Kritikalität der Schwachstelle vor dem Einspielen des Patches automatisiert geprüft werden sollen.

Das SVT importiert „Common Vulnerabilities and Exposures“ (CVE)-Informationen der öffentlich verfügbaren „National Vulnerability Database“ (NVD) in eine Datenbank und ermöglicht es, diese anhand unterschiedlicher Kriterien nach CVE-Einträgen zu durchsuchen. Außerdem können „Common Platform Enumeration“ (CPE)-Profile erstellt, gespeichert und die Datenbank anhand des Profils nach Sicherheitslücken durchsucht werden. Zusätzlich ist es möglich, diverse Statistiken zu erzeugen.

Das Papier beschreibt alle aufgetretenen Probleme und gesammelten Erfahrungen im Umgang mit SVT und den genannten Standards. Es werden dazu bewusst ausschließlich frei verfügbare Standards und Anwendungen eingesetzt, um uns von der zunehmenden Kommerzialisierung des Schwachstellenmanagements z.B. in SIEM-Lösungen (Security Information and Event Management) abzuheben.

Der restliche Beitrag hat die folgende Struktur: Abschnitt 2 diskutiert die grundlegenden Standards. Die verwendete Methodik ist Gegenstand von Abschnitt 3, während Abschnitt 4 die Ergebnisse beschreibt. Abschnitt 5 diskutiert die Ergebnisse. Den Abschluss bildet Abschnitt 6 mit einem Fazit und Ausblick.

2 Grundlagen

2.1 Relevante Standards und Projekte

Zur automatisierten Verarbeitung von bekannten Sicherheitslücken in Softwareprodukten bieten sich eine Reihe von Standards an, die es ermöglichen, Softwareprodukte sowie bekannte Sicherheitslücken zu identifizieren und zu verwalten [Chri10]. Eine Übersicht der für dieses Papier relevanten Standards findet sich in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

Geht es um die eindeutige Identifikation von bekannten Sicherheitslücken in Softwareprodukten sollte auf den De-facto-Standard CVE gesetzt werden. Die CVE wird von nahezu allen Herstellern und Anbietern von Softwareprodukten weltweit eingesetzt, um öffentlich bekannte Sicherheitslücken und -Risiken eindeutig zu identifizieren und zu nummerieren. CVE-Nummern werden in der Regel zentral von der MITRE Corporation auf Anfrage zugewiesen. Größere Hersteller, Organisationen oder Sicherheitsforscher können sich als „CVE Numbering Authority“ anerkennen lassen und unabhängig von der MITRE selbst CVE-Nummern vergeben.

Geschuldet der Tatsache dass CVE-Nummern primär zentral von der MITRE Corporation vergeben und verwaltet werden und zudem die Anzahl der angemeldeten Sicherheitslücken zugenommen hat, kam es in der Vergangenheit zu Engpässen und Verzögerungen bei der Vergabe

neuer CVE-Nummern (http://seclists.org/oss-sec/2016/q1/560). Mit dem „Distributed Weakness Filing“ [DWF]-Projekt gibt es mittlerweile eine Alternative, die ähnlich vorgeht, aber DWF-Nummern schneller vergeben kann.

Tab. 1: Zusammenfassung der für dieses Papier relevanten Standards

Standard	Abkürzung	Verwaltet von	Aktuelle Version	Beschreibung
Common Configuration Enumeration	CCE	NIST	5	Einheitliche Benennung der Konfiguration von Software
Common Platform Enumeration	CPE	NIST	2.3	Standard zur Benennung von IT-Produkten und Plattformen
Common Vulnerabilities and Exposures	CVE	MITRE Corp.	-	Eindeutige Identifikation von öffentlich bekannten Sicherheitslücken und –Risiken
Common Vulnerability Scoring System	CVSS	FIRST	3.0	Framework zur Charakterisierung und Einstufung des Schweregrads von Sicherheitslücken
Common Weakness Enumeration	CWE	MITRE Corp.	2.9	Liste mit Softwareschwachstellen
Distributed Weakness Filing	DWF	DWF Project	-	Eindeutige Identifikation von Sicherheitslücken und -Risiken analog zu CVE

Zur Identifikation von Softwareprodukten auf einem IT-System eignet sich insbesondere die „Common Platform Enumeration“ (CPE), vgl. [ChWS11]. Bei der CPE handelt es sich um einen Standard zur eindeutigen Benennung von IT-Produkten und IT-Plattformen, bei der ein maschinenlesbares Format verwendet wird. Damit bietet CPE gute Möglichkeiten zum automatischen Vergleich von Namen. Die NVD verwaltet ein CPE-Dictionary, welches bereits eine große Anzahl an Einträgen zu Produkten und Plattformen enthält. Am 06.05.2016 waren insgesamt 112.137 Einträge im CPE-Dictionary enthalten. CPE spezifiziert die Namensgebung, den Aufbau des Wörterbuchs, Regeln zum Matching von CPE-Ausdrücken sowie eine Sprache zur Gruppierung von CPE-Namen zur Beschreibung von IT-Plattformen.

Im Zuge der Identifikation von Sicherheitslücken sollte ebenfalls die Charakterisierung und Einstufung des Schweregrads berücksichtigt werden, wofür sich insbesondere das „Common Vulnerability Scoring System“ (CVSS) [FIRST, MeSR07] eignet. CVSS stellt ein Framework dar, um auf der einen Seite die Ausnutzbarkeit und auf der anderen Seite die Auswirkung (Impact) einer Sicherheitslücke standardisiert bewerten zu können. Aus den Metriken (1) Base Score, (2) Temporal Score und (3) Environmental Score wird durch eine Formel ein CVSS-Gesamtscore im Intervall [0, 10] berechnet, wobei 10 der maximalen Kritikalität entspricht. Basierend auf diesen Informationen lassen sich Entscheidungen zum weiteren Umgang mit einer Sicherheitslücke treffen, z.B. wie zeitnah ein entsprechender Patch eingespielt werden sollte.

Bei der „Common Weakness Enumeration“ [CWE] handelt es sich um ein Projekt und einen Katalog zur Erfassung von Ursachen für Softwareschwachstellen. Ziel der CWE ist es, Sicherheitslücken und deren Ursachen in Softwareprodukten besser verstehen und klassifizieren zu können.

2.2 NVD

Die NVD [NVD] ist eine der größten kostenfreien Datenbanken für Sicherheitslücken. Sie wird von der Computer Security Abteilung des „National Institute of Standards and Technology“ (NIST) betrieben und von der US-Regierung finanziert. Jede öffentlich gemachte Schwachstellenmeldung bekommt als Primärschlüssel eine eindeutige CVE-Nummer zugewiesen, welche mit zusätzlichen Informationen wie dem CVSS-Score, CPE-Namen der verwundbaren Software sowie CWE-Details zur Schwachstelle angereichert wird. Zum Beispiel beschreibt der NVD-Eintrag zur CVE-2015-8440 (https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-8440) eine Sicherheitslücke im Adobe Flash Player mit einem CVSSv2-Base-Score von 10.0.

Ein enthaltener CPE-Eintrag ist z.B. `cpe:/a:adobe:flash_player:19.0.0.245`; der dazugehörige CWE-Eintrag lautet „Permissions, Privileges, and Access Control (CWE-264)“. Diese Daten werden über ein XML-Schema zum Download zur Verfügung gestellt.

3 Methodik

3.1 Grundkonzept

Vor der Entwicklung des SVT ist ein allgemeines Konzept zur automatisierten Erkennung von bekannten Sicherheitslücken erstellt worden. Dieses spiegelt eine mögliche Vorgehensweise zur automatisierten Identifizierung von bekannten Sicherheitslücken wider.

Zunächst müssen Informationen zu den IT-Systemen eines Unternehmens oder dem eines Privatanwenders erfasst werden. Diese Angaben sollten eine möglichst detaillierte Auskunft, wie Name, Hersteller und Version, über die installierten Softwareanwendungen und das Betriebssystem bieten. Die über die Computersysteme gesammelten Informationen müssen anschließend mit einer Informationsquelle abgeglichen werden, in welcher bekannte Sicherheitslücken aufgeführt sind. Dadurch kann identifiziert werden, ob beispielsweise für ein bestimmtes Softwareprodukt eine Sicherheitslücke bekannt ist und wie kritisch beziehungsweise schwerwiegend diese ist.

Als Datenbasis für die Informationen zu den bekannten Sicherheitslücken kann das von der NVD bereitgestellte, frei verfügbare CVE-Verzeichnis genutzt werden. Dieses Verzeichnis beinhaltet neben den CVE-Angaben zusätzliche, durch die Standards CPE, CVSS und CWE bereitgestellte Informationen zu betroffenen IT-Produkten beziehungsweise –Plattformen, sowie zur Kritikalität und zur Ursache einer Sicherheitslücke.

Um die Überprüfung zu automatisieren, müssen zwei Probleme gelöst werden:

1. Die Erfassung der Informationen über die Computersysteme muss automatisiert werden.
2. Die Überprüfung, ob Sicherheitslücken zu diesen Anwendungen bekannt sind, muss implementiert werden.

3.2 Vorgehen

Die Untersuchung verwendet als Betriebssystem Microsoft Windows 7 und 26 frei verfügbare Softwareprodukte, u.a. Mozilla Firefox, WinRAR, Skype, .NET-Framework, VLC Player, Apache Open Office und PuTTY, die aus fünf Top-Downloadlisten ausgewählt wurden. Scherf [Sche15] gibt alle Details dazu. Der Untersuchungszeitraum war vom 23.08.2015 bis zum

20.11.2015. Die Anwendungen des Softwareprofils und das definierte Vergleichsprodukt Secunia PSI (vgl. Abschnitt 5.2) wurden alle auf einem zur Verfügung stehenden IT-System installiert. Ab dem Beginn der Testphase wurde in regelmäßigen Abständen von bis zu maximal zwei Tagen eine Überprüfung der Softwareanwendungen mit Secunia PSI durchgeführt.

Nach der Testphase wurde nach bekannten CVE-Einträgen für das durch CPE spezifizierte Softwareprofil im frei verfügbaren CVE-Verzeichnis der NVD gesucht und mit den anderen Ergebnissen verglichen. Dazu wurde das SVT entwickelt, vgl. Abschnitt 3.3.

Sobald Secunia PSI eine Sicherheitswarnung zu einer im Softwareprofil definierten Anwendung meldete, wurde diese dokumentiert und die betroffene Anwendung anschließend auf eine neuere Version aktualisiert, um die entsprechende Sicherheitslücke zu schließen. Das Softwareprofil ist dadurch nicht statisch, sondern änderte sich im Hinblick auf die installierte Version der Anwendungen dynamisch über den Testzeitraum.

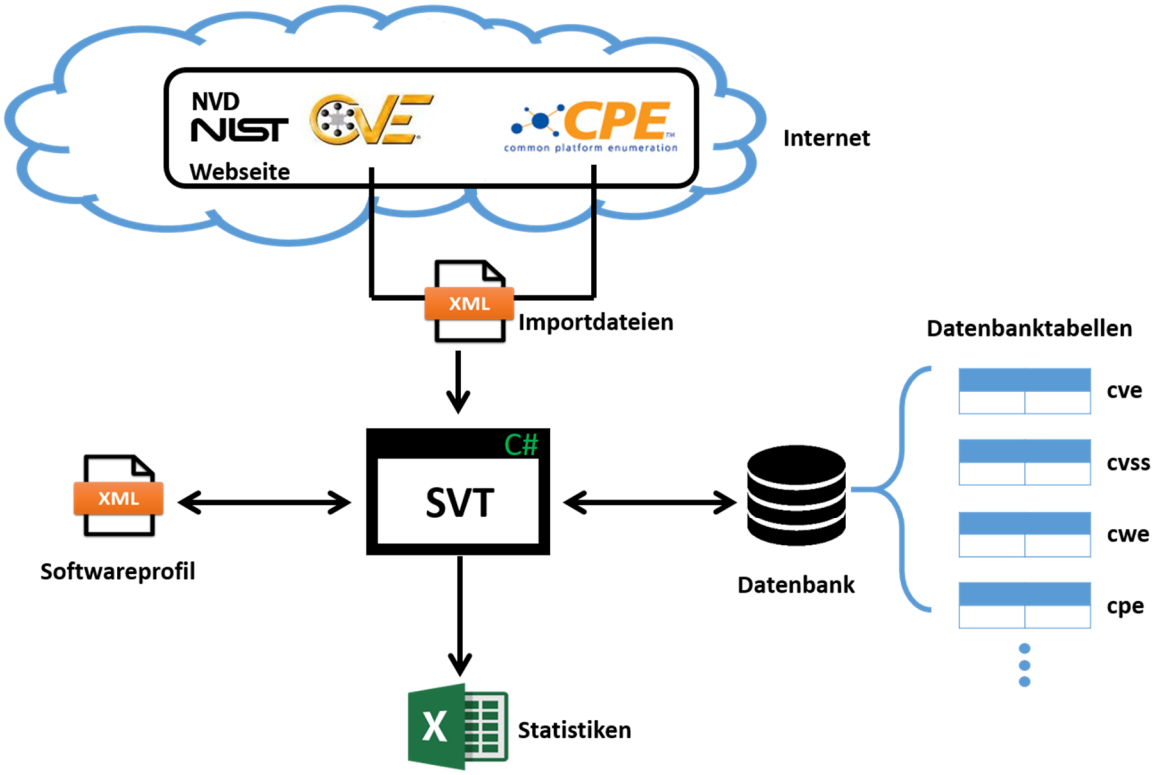

Abb. 1: SVT – Darstellung der ein- und ausgehenden Daten

3.3 Software Vulnerability Tool (SVT)

Alle benötigten Standards müssen vom SVT unterstützt werden und bestehende NVD-Daten über das XML-Schema importierbar sein. Das Softwareprofil wird per CPE hinterlegt und gegen die vorhandenen Informationen verglichen. Das Ergebnis ist eine Statistik mit den relevanten CVE-Nummern, vgl. Abbildung 1.

SVT wurde in der Programmiersprache C# mit Visual Studio 2013 und dem .NET-Framework entwickelt. Die Daten werden in einer SQLite-Datenbank gespeichert. Das Tool kann unter [SVT] bezogen werden. Scherf [Sche15] gibt weitere Informationen.

4 Ergebnisse

4.1 Globale Ergebnisse mittels SVT

Zum Zeitpunkt der Überprüfung befanden sich insgesamt 72.903 CVE-Einträge in der SQLite-Datenbank des SVT. Von diesen besaßen 72.846 eine Bewertung anhand eines CVSS-Base-Scores und 41.239 eine Verknüpfung mit einer CWE-ID. Insgesamt waren 199.138 unterschiedliche CPEs in der Datenbank vorhanden. Es existierten 1.921.344 CVE-CPE-Verknüpfungen, die festlegen, welches Softwareprodukt welche Sicherheitslücke enthält. 87% der in der Datenbank vorhandenen CPEs entfielen auf Anwendungen, 8% auf Betriebssysteme und 5% auf Hardware. Seit 2005 liegt die Anzahl der pro Jahr veröffentlichten Schwachstellen bei 5.000 oder mehr, mit einem Durchschnittswert von 6.400. Die meisten Schwachstellen haben einen CVSS-Base-Score im Intervall [7, 8]. 5.293 Schwachstellen sind mit einem CVSS-Base-Score von 10.0 bewertet.

Von den mehreren Tausend möglichen CWE-Einträgen werden nur 29 genutzt. Die Spitzenreiter mit über 4.000 Verwendungen sind CWE-79 (Cross-Site Scripting), CWE-119 (Improper Restriction of Operations within the Bounds of a Memory Buffer), CWE-264 (Permissions, Privileges, and Access Controls) und CWE-89 (SQL Injection).

4.2 Ergebnisse für das Softwareprofil

Zehn der 30 Produkte im Softwareprofil waren während des Untersuchungszeitraums von Sicherheitslücken betroffen. In der Summe wurden 286 CVE-Nummern identifiziert. Die meisten Schwachstellen entfielen auf Apple iTunes (75), Adobe Acrobat Reader (58), Adobe Flash Player (50), Mozilla Firefox (33) und Microsoft Windows 7 (30). Während für Windows 7 als Gegenmaßnahme Patches von Microsoft zur Verfügung gestellt wurden, behoben die anderen Softwareprodukte die Sicherheitslücken mit einer neuen Version. Der CVSS-Base-Score schwankt je nach Softwareanwendung zwischen 6,8 (VLC Media Player) und 9,2 (Adobe Flash Player). Innerhalb unseres Softwareprofils gab es nur zehn CWE-IDs, die meisten für CWE-119. 81 CVEs hatten keine CWE-ID zugewiesen.

4.3 Vergleich mit Secunia PSI

Tabelle 1 zeigt den Vergleich der Ergebnisse zwischen SVT und Secunia PSI. Bei 18 von 27 Softwareprodukten sind die Ergebnisse identisch. Zu den Unterschieden:

SVT findet beim .NET Framework und beim Internet Explorer automatisiert keine CVEs, da die CPE-Beschreibung in der NVD zu ungenau ist, vgl. Abschnitt 5.1. Secunia PSI zeigt, wenn ein großer Versionssprung vorliegt, die Kritikalität „end of life“ an, obwohl Sicherheitslücken zu der installierten Version vorhanden sind. Während des Testlaufes passierte dies z.B. mit Google Chrome. Zusätzlich gab die Secunia-Webseite in vier Fällen (Apple iTunes, Google Chrome, Java Runtime Environment und PuTTY) fragwürdige CVE-IDs an. Im Fall des VLC Players in der Version 2.2.1 erkannte Secunia PSI eine bereits bekannte Sicherheitslücke nicht (CVE-2015-5949). Secunia PSI gibt offenbar nur Sicherheitswarnungen an, wenn für das Softwareprodukt eine Lösung in Form eines Updates zur Verfügung steht. Martin [Mart14] bespricht Secunia Schwachstellenmeldung detailliert.

Tab. 2: Vergleich der Ergebnisse von Secunia PSI und SVT

	Sicherheitslücken gefunden von	
Softwareanwendung	Secunia PSI	Software Vulnerability Tool
.NET Framework	Ja	zu ungenaue Versionsangabe in CPE
7-Zip	Nein	Nein
Adobe Flash Player	Problem durch Update auf neue Version	Ja
Adobe Reader	Ja	Ja
AdwCleaner	Nein	Nein
Apache OpenOffice	Ja	Ja
Apple iTunes	Angabe fragwürdiger CVEs	Ja
Ccleaner	Nein	Nein
CDBurnerXP	Nein	Nein
FileZilla	Nein	Nein
Gimp	Nein	Nein
Google Chrome	Problem durch Update auf neuere Version Angabe fragwürdiger CVEs	Ja
IrfanView	Nein	Nein
Java Runtime Environment	Angabe fragwürdiger CVEs	Ja
KMPlayer	Nein	Nein
Microsoft Internet Explorer	Ja	zu ungenaue Versionsangabe in CPE
Microsoft Silverlight	Nein	Nein
Mozilla Firefox	Problem durch Update auf neuere Version	Ja
Notepad++	Nein	Nein
PDFCreator	Nein	Nein
Putty	Angabe fragwürdiger CVE	Nein
Recuva	Nein	Nein
Skype	Nein	Nein
TrueCrypt	Nein	Nein
VLC Player	Nein	Ja
Windows 7	Ja	Ja
WinRAR	Nein	Nein

5 Diskussion

5.1 Ergebnisse

Das Zusammenspiel zwischen CVSS und CVE ist sehr gut. So existiert in 99,92% (Stand 24.11.2015) aller CVEs eine Bewertung durch CVSS. Zu beachten ist allerdings, dass nur der CVSS-Base-Score angegeben wird, Temporal- sowie Environmental-Score werden nicht zur Verfügung gestellt. Im Fall des Environmental-Score macht dies auch Sinn, da dieser von der Einsatzumgebung abhängt. Der Temporal-Score könnte definiert werden, müsste aber über die Zeit aktualisiert werden.

Das CVE-Verzeichnis nutzt CWE nur in Form der Angabe der CWE-ID. Der relativ geringe Anteil (56,57%) der CVEs, in denen eine CWE-ID angegeben ist, und die geringe Zahl der überhaupt verwendeten CWE-IDs (29 Stück) zeigt die Problematik des Zusammenspiels. Die

Ursache dafür liegt in der Komplexität von CWE sowie darin, dass eine Zuweisung einer CWE zu einer Sicherheitslücke manuell erfolgt und nur schwer automatisiert möglich ist.

Die schwerwiegendsten Probleme wurden bei CPE beobachtet:

CVEs ohne CPE-Eintrag: Es kommt vor (161 Mal während unserer Untersuchung), dass die von der MITRE Corporation veröffentlichten CVEs noch nicht vom NVD-Team analysiert wurden, um die betroffenen CPEs zuzuordnen. Das beschriebene Problem einer CVE ohne CPE-Eintrag konnte insbesondere häufig in erst kürzlich veröffentlichten CVEs beobachtet werden. Dadurch ist es nicht möglich, neue, erst vor kurzem bekannt gewordene Sicherheitslücken automatisiert einer Software zuzuordnen.

Fehlende bzw. ungenaue CPE-Zuordnungen: Häufig wird in den CVE-Einträgen der NVD nur die aktuellste Softwareversion mit CPE angegeben, obwohl auch vorherige Versionen von der Schwachstelle betroffen sind. In solchen Fällen wird der CPE-Angabe der Zusatz „and previous versions“ hinzugefügt. Das ist im CPE-Standard so nicht vorgesehen und erschwert die automatisierte Auswertung ungemein. Weiterhin ist der Zusatz nur auf der Webseite sichtbar und wird nicht im XML-Schema (v2.0) des CVE-Verzeichnisses von NVD definiert, wodurch es in den zum Download verfügbaren XML-Dateien des CVE-Verzeichnisses zu einem Informationsverlust kommt. Wir konnten dies für das Jahr 2015 insgesamt 5.615 Mal beobachten.

CPE nicht akkurat genug: Viele Produkte im Softwareprofil sind nicht exakt genug in der verwendeten Version im CPE-Dictionary enthalten. SVT lieferte automatisiert z.B. keine Einträge über das .NET-Framework und den Microsoft Internet Explorer, obwohl zu diesen Sicherheitslücken gefunden wurden. Der Grund dafür ist, dass das SVT bei beiden Anwendungen aufgrund der zu ungenauen Versionsangabe in CPE-Einträgen keine Überprüfung durchführen kann, da Sicherheitslücken für mehrere Versionen unter einer CPE-ID zusammengefasst werden. Bsp.: Eingesetzte Version 4.5.50709 vs. CPE-Version 4.5 für das .NET-Framework in der Schwachstellenbeschreibung.

Problem der automatisierten CPE-Generierung: Insbesondere für neue oder weniger stark verbreitete Softwareprodukte bzw. Softwareproduktversionen gibt es in der CPE-Datenbank oft keinen entsprechenden Eintrag. Dieser muss manuell erstellt werden, wobei es bei der detaillierten Namensgebung trotz des vorhandenen CPE-Standards oft Probleme gibt. Die für SVT in Windows 7 erreichbaren Quellen (C#-Bibliothek oder die Auflistung der Programme und Funktionen in der Systemsteuerung) für die Software-Informationen lassen sich leider nicht automatisiert in einen CPE-Eintrag überführen, da je nach Softwareprodukt individuelle Anpassungen vorgenommen werden müssen. Tabelle 3 zeigt zwei Beispiele. Die von Microsoft und CPE verwendeten Benennungen von Softwareprodukten sind inkompatibel.

Tab. 3: Beispiele der notwendigen individuellen Anpassung bei der automatisierten CPE-Generierung

	cpe:/{part}	:{vendor}	:{product}	:{version}
generierte CPE	cpe:/a	:microsoft_corporation	:microsoft_silverlight	:5.1.40728.0
existierende CPE	cpe:/a	:microsoft	:silverlight	:5.1.40416.0
generierte CPE	cpe:/a	:apache_software_foundation	:openoffice_4.1.2	:4.12.9782
existierende CPE	cpe:/a	:apache	:openoffice	:4.1.1

Unterschiedliche Versionen der Standards: Eine weitere Herausforderung stellen unterschiedliche Versionen der Standards dar. So führt die Herausgabe aktualisierter Standards wie z.B. die CVSS-Version 3.0 oder CPE Version 2.3 dazu, dass alle bestehenden CVE-Meldungen bei NVD geändert werden müssen. Geschieht dies nicht umfassend, ergeben sich Inkonsistenzen. Auch für das von NVD verwendete XML-Schema gibt es mittlerweile zwei verschiedene Versionen (v2.0 und v1.2.1).

5.2 Verwandte Arbeiten

Die vorliegende Arbeit beschränkt sich ausschließlich auf kostenfrei erhältliche Tools. Kommerzielle Produkte wie Cisco Security IntelliShield Alert Manager Service, IBM Internet Scanner oder AlienVault Unified Security Management wurden nicht untersucht.

Neben dem Secunia PSI (heute Flexera PSI) können Tools wie SUMo[1] und FileHippo[2] zum Erfassen des Patch-Stands eines Systems verwendet werden. Sie unterscheiden sich allerdings stark darin, in welchem Umfang die für uns relevanten Standards unterstützt werden. Am ähnlichsten zum SVT ist Secunia PSI, der CVE, aber weder CPE noch CVSS unterstützt. Scherf [Sche15] gibt eine genauere Abgrenzung der Tools. Welberg [Welb15] liefert mehr Details zu 30 Produkten zum Vulnerability Management.

Die Software vfeed [vfeed] bietet ebenfalls eine SQLite Datenbank mit allen CVE-Nummern. Leider wird die Datenbank aber nur in unregelmäßigen Abständen von den Autoren aktualisiert. Für Privatanwender bietet sich, da für den nicht kommerziellen Einsatz kostenlos, der Nessus Vulnerability Scanner[3] der Firma Tenable Network Security an. Mit Nessus können lokal und im Netzwerk verfügbare Dienste auf Schwachstellen und fehlende Patches gescannt werden. Nessus liefert neben der Beschreibung der Schwachstellen Lösungen, wie die Schwachstellen beseitigt werden können. Voraussetzung für einen zuverlässigen Einsatz von Nessus ist die Aktualität der Schwachstellendatenbank. Allerdings unterstützt Nessus keine CPE-Namen.

6 Fazit und Ausblick

6.1 Fazit

Das Zusammenspiel zwischen CVE und CVSS ist bereits sehr gut, wohingegen beim Zusammenspiel zwischen CPE und CVE Schwierigkeiten bestehen. Dies liegt zum größten Teil an Problemen, die den Standard CPE selbst betreffen, jedoch durch Verknüpfung mit CVE eben auch diesen beeinflussen. Ein Beispiel hierfür ist die Zusammenfassung mehrerer Versionen eines IT-Produktes zu einem CPE-Eintrag. Das Zusammenspiel zwischen CVE und CWE ist zwar in den NVD-Meldungen vorhanden, aber nur sehr rudimentär umgesetzt.

Eine Herausforderung bei dem beschriebenen Ansatz stellt zudem die Tatsache dar, dass Anwender und Betreiber von IT-Systemen in erster Linie daran interessiert sind, dass ihre IT-Systeme mit allen verfügbaren Sicherheits-Patches betrieben werden. Hierfür eignet sich der Weg über die CVE-Datenbank allerdings nur bedingt, weil diese unzureichend Informationen darüber liefert, welche Sicherheits-Patches fehlen bzw. welche Version konkret eingespielt

[1] http://www.kcsoftwares.com/?sumo

[2] http://www.filehippo.com/de/download_app_manager/

[3] http://www.tenable.com/products/nessus-vulnerability-scanner/

werden muss, um alle bekannten Sicherheitslücken zu schließen. Unser Ansatz erlaubt statt eines „blinden Patchens“ eine sehr viel detailliertere Unterscheidung, ob eine Schwachstelle und der daraus resultierende Patch tatsächlich eingespielt werden muss. Insbesondere der CVSS-Score ist hier hilfreich. Diese dezidierte Vorgehensweise ist z.B. in kritischen Infrastrukturen wichtig, da wegen des hohen Testaufwands und einer evtl. notwendigen Rezertifizierung häufig nur Schwachstellen mit hoher Kritikalität gepatcht werden können.

Der CPE-Standard sieht eigentlich vor, dass Hersteller freiwillig neue Softwareversionen an das CPE-Dictionary melden, was aber z.B. bei Microsoft offenbar nicht geschieht. Ein weiteres Zeichen für die schleppende Verwendung der Standards ist, dass nur wenige Hersteller wie z.B. Oracle einen CVSS-Score in ihren Schwachstellenmeldungen angeben. Diese Versäumnisse müssen dann von der NVD nachgeholt werden.

Trotz dieser Probleme ist es sinnvoll, das CVE-Verzeichnis als eine mögliche Informationsgrundlage für Sicherheitsanwendungen in Betracht zu ziehen, da der CVE-Standard von vielen Herstellern genutzt wird, um Sicherheitslücken eindeutig zu benennen und zu identifizieren. Das von der NVD bereitgestellte CVE-Verzeichnis bietet zurzeit eine der besten Möglichkeiten, kostenlos auf zuverlässige und gut strukturierte Informationen zu Sicherheitslücken zuzugreifen und damit der notwendigen Automatisierung der Behandlung der großen Zahl von Schwachstellen zu begegnen.

6.2 SVT Weiterentwicklung

Die Untersuchung könnte auf einen längeren Zeitraum und ein größeres Softwareprofil ausgedehnt und unter Einbeziehung zusätzlicher Standards wie z.B. CCE und der Temporal- und Environmental-CVSS-Metriken wiederholt werden. SVT könnte über eine zentrale Datenbank und eine Aufteilung in einen Client (Agent zur Erfassung und Überwachung der installierten Software) und Server erweitert werden und in größeren IT-Landschaften eingesetzt werden. Des Weiteren sollte aus Sicht des Patch-Managements und des Patch-orientierten Ansatzes darüber nachgedacht werden, eine Datenbank anzubinden, die nicht nur über Sicherheitslücken informiert, sondern insbesondere auch über die zu installierenden Sicherheits-Patches und -Updates für die eingesetzte Software.

6.3 Zukünftiger Umgang mit Sicherheitslücken

Die NIST-Datenbank adressiert mit CVE ausschließlich öffentlich gemeldete Schwachstelleninformationen und auch nur solche, bei denen der Hersteller einer Software sich dazu entschieden hat, CVE zum Tracking von Sicherheitslücken zu nutzen. Daneben existieren allerdings Sicherheitslücken, für die zwar ein Sicherheits-Patch verfügbar ist, aber keine CVE-Nummer beantragt wurde. Derartige Informationen lassen sich nicht von der CVE-Datenbank beziehen, sondern werden den Nutzern der Software entweder nur direkt in Form sog. Security Bulletins zur Verfügung gestellt oder durch kommerzielle Vulnerability Provider erfasst, so dass ausschließlich zahlende Kunden darauf Zugriff haben. Da der Betrieb einer Datenbank für Schwachstellen Geld kostet und zudem damit Geld verdient werden kann, ist zu vermuten, dass die Anzahl frei verfügbarer und kostenloser Vulnerability Provider wie z.B. die kürzlich eingestellte Open Sourced Vulnerability Database (OSVDB) in Zukunft abnehmen wird[4].

[4] https://blog.osvdb.org/2016/04/05/osvdb-fin/

Des Weiteren lässt sich der Trend beobachten, dass in den letzten Jahren Informationen zu Sicherheitslücken häufig nicht mehr veröffentlicht, sondern geheim gehalten und u.U. sogar verkauft werden. Dies kann auf offiziellem Weg über sog. „Bug Bounty Programme“ der Hersteller oder über Exploit Vendors mit beliebigen zahlungswilligen Kunden geschehen. Insgesamt zeichnet sich damit ein Trend vom klassischen „offenen“ Umgang hin zum kommerziellen „geschlossenen“ Umgang mit Sicherheitsschwachstellen ab.

Literatur

[Chri10] S. Christey: CVE and CVSS. Hg. v. The MITRE Corporation. http://scap.nist.gov/events/2010/itsac/presentations/day1/SCAP_101-CVE_and_CVSS.pdf (zuletzt geprüft am 11.12.2015)

[ChWS11] B.A. Cheikes, D. Waltermire, K. Scarfone: Common Platform Enumeration: Naming Specification Version 2.3. NIST Interagency Report 7695, Aug. 2011

[CWE] Common Weakness Enumeration (CWE). https://cwe.mitre.org (zuletzt geprüft am 01.06.2016)

[DWF] Distributed Weakness Filing (DWF) Project. https://github.com/distributed-weaknessfiling/DWF-Documentation (zuletzt geprüft am 06.05.2016)

[FIRST] FIRST (Hrsg.): Common Vulnerability Scoring System v3.0. Specification Document. https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf (zuletzt geprüft am 06.05.2016)

[Mart14] B. Martin: Reviewing the Secunia 2013 Vulnerability Review. http://blog.osvdb.org/2014/03/18/reviewing-the-secunia-2013-vulnerability-review (zuletzt aktualisiert am 18.03.2014, zuletzt geprüft am 11.12.2015)

[MeSR07] P. Mell, K. Scarfone, S. Romanosky: A Complete Guide to the Common Vulnerability Scoring System Version 2.0. Hg. v. FIRST. 2007, https://www.first.org/cvss/cvss-v2-guide.pdf (zuletzt geprüft am 06.05.2016)

[NVD] National Vulnerability Datenbank. https://nvd.nist.gov (zuletzt geprüft am 16.03.2016)

[PSI] Flexera Personal Software Inspector. http://www.flexerasoftware.com/enterprise/products/software-vulnerability-management/personal-software-inspector (zuletzt geprüft am 22.03.2016)

[Sche15] M. Scherf: Software Vulnerability Tool – Entwicklung einer Anwendung zur Analyse des CVE-Verzeichnisses – Vergleichende Analyse der Standards CVE, CPE, CVSS und CWE. Bachelorarbeit, Hochschule Trier, Dezember 2015.

[SVT] Software Vulnerability Tool, Version 1.0, Download unter: http://public.hochschule-trier.de/~knorr/DACH2016

[vfeed] vFeed – The Correlated Vulnerability and Threat Database. https://github.com/toolswatch/vFeed (zuletzt geprüft am 16.3.2016)

[Welb15] S.M. Welberg.: Vulnerability management tools for COTS software – A comparison. Hg. v. University of Twente, http://doc.utwente.nl/64654/1/Vulnerability_management_tools_for_COTS_software_-_a_comparison_v2.1.pdf (zuletzt geprüft am 11.12.2015)