ERNW
Wir leben IT-Security

ERNW Newsletter 19 / September 2007

Liebe Partner, liebe Kollegen,

willkommen zur 19ten Ausgabe des ERNW-Newsletters mit dem Thema:

Metrikbasiertes Patchen mit CVSS 2.0
Konzept mit Methode

Version 1.0 vom 12. September 2007

von: Dror-John Röcher, droecher@ernw.de

Dieser Newsletter erläutert eine Methodik zur Bewertung von Schwachstellen, die auf den Metriken des Common Vulnerability Scoring Systems (CVSS) basiert, und stellt dar, wie diese Bewertung in den Patchmanagement Prozess eingebunden werden kann.

INHALTSVERZEICHNIS

1 EINLEITUNG

Effizientes Patchmanagement stellt auch im Sommer 2007 noch eine Herausforderung an die IT-Security Abteilungen dar. Effiziente Patch-Werkzeuge sind zwar weit verbreitet, Prozesse oft formal definiert oder de-facto etabliert, aber eine Frage wird in vielen Umgebungen bis heute stark vernachlässigt, obwohl die Beantwortung dieser Frage die Grundlage der Entscheidung „to patch or not to patch?“ ist: „Wie kritisch ist die Schwachstelle am Tag X in der Umgebung Y?“. Oder anders ausgedrückt: „Wenn Patche für zwei Produkte verfügbar sind – welcher hat die höhere Priorität?“. Es fehlt eine anerkannte Metrik zur Bewertung von Schwachstellen. Diese Lücke soll das „Common Vulnerability Scoring System“, kurz CVSS, schließen. Bisher verlassen sich Organisationen entweder auf die Einstufung der Schwachstelle durch den Hersteller der Software oder durch einen spezialisierten Dienstleister. Ersteres hat den Nachteil, dass verschiedene Hersteller unterschiedliche Kriterien zur Bewertung anlegen, letzteres berücksichtigt in keiner Weise den Zeitpunkt der Bewertung oder die Spezifika der Organisation.

In den folgenden Abschnitten wird nach einer Erläuterung des CVSS diskutiert, wie eine transparente und nachvollziehbare Bewertung von Schwachstellen zu einer Verbesserung des Patchmanagement-Prozesses beitragen kann.

2 DAS COMMON VULNERABILITY SCORING SYSTEM (CVSS)

CVSS[1] wurde als offener und universeller Standard zur Bewertung von Schwachstellen in IT-Produkten entwickelt und erfreut sich zunehmend großer Verbreitung in der InfoSec-Industrie. Im folgenden Abschnitt wird zunächst die Funktionsweise von CVSS erläutert, bevor ein mögliches Konzept des Patchmanagements auf Basis der CVSS Metriken diskutiert wird.

2.1 Grundlagen des CVSS

CVSS in der Version 2.0 definiert Metriken zur Berechnung von drei verschiedenen Punktzahlen (so genannten *Scores*), die jeweils eine eigene Aussage haben.

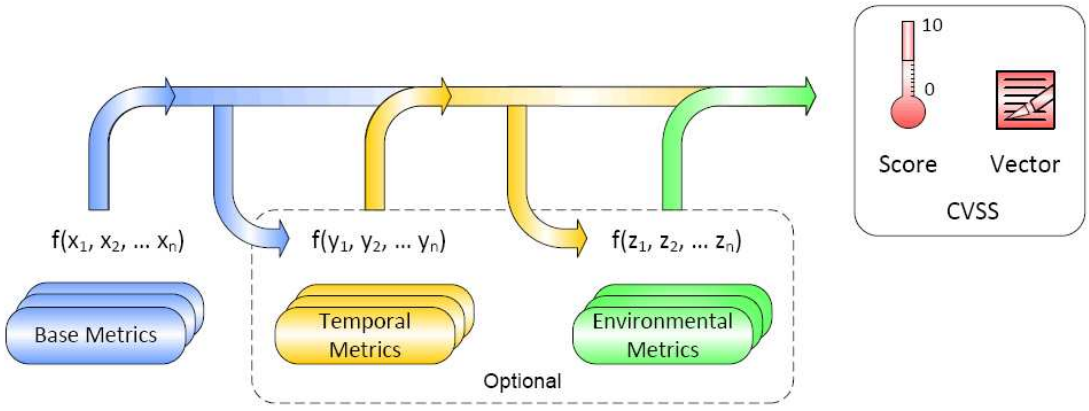

Abbildung 1 Zusammenhang zwischen den verschiedenen Metriken des CVSS

2.1.1 Base Metriken & Base Score

Der Base Score bewertet die Schwachstelle ohne Berücksichtigung weiterer Faktoren und stellt die – wie der Name schon andeutet – Basis für die Grundlage der nachfolgenden Scores dar.

[1] *http://www.first.org/cvss*

Einmal festgelegt, ändert sich der Base Score im Gegensatz zu den beiden anderen Scores nicht mehr. Die sechs Metriken, die in die Berechnung des Base Scores eingehen, sind:

- **Access-Vector (AV):** Der Access-Vector gibt wieder, wie eine Schwachstelle ausgenutzt werden kann. Die möglichen Werte sind:
  - Local (L): Um die Schwachstelle auszunutzen, ist physischer Zugriff auf das System oder die Ausführung durch einen lokalen Benutzer notwendig. Ein typisches Beispiel ist ein lokaler „Privilege Escalation" Angriff, bei dem ein Benutzer sich unautorisiert zusätzliche Rechte verschafft.
  - Adjacent Network (A): Die Schwachstelle kann ausgenutzt werden, falls der Angreifer Zugriff auf das lokale Netzwerksegment, bzw. die Ethernet-Kollisionsdomäne, des Ziels hat. Ein typisches Beispiel sind Angriffe gegen Link-local Protokolle wie etwa LLDP.
  - Network (N): Die Schwachstelle kann aus dem Netzwerk heraus ausgenutzt werden, ohne das Zugriff auf das lokale System oder das lokale Netzwerksegment notwendig ist. Ein typisches Beispiel ist ein Buffer-Overflow in einem Serverdienst, der mittels TCP/IP adressiert wird.
- **Access-Complexity (AC):** Diese Metrik bewertet die Komplexität eines erfolgreichen Angriffs auf eine Schwachstelle. Die möglichen Werte sind:
  - High (H): Um die Schwachstelle auszunutzen, muss eine Vielzahl von spezifischen Faktoren zusammenspielen. Zum Beispiel muss ein Angreifer über erhöhte Privilegien verfügen, oder ein Angreifer muss ein Opfer verleiten, eine präparierte Website zu besuchen.
  - Medium (M): Die Schwachstelle kann ausgenutzt werden, sobald bestimmte Umstände zusammentreffen, bzw. wenn die Ziele bestimmte Voraussetzungen erfüllen. Zum Beispiel kann eine nicht-default Konfiguration Voraussetzung sein.
  - Low (L): Eine Schwachstelle kann ohne das Vorhandensein limitierender Faktoren ausgenutzt werden, oder das Zielsystem kann von einer Vielzahl an Systemen und Benutzern aus erreicht werden, zum Beispiel anonym aus dem Internet.
- **Authentication (Au):** Diese Metrik bewertet, wie oft eine erfolgreiche Authentifizierung stattfinden muss, bevor die Schwachstelle ausgenutzt werden kann.
  - Multiple (M): Um die Schwachstelle auszunutzen muss sich ein Angreifer mindestens zwei Mal, ggf. mit den gleichen Credentials authentifizieren.
  - Single (S): Ein Angreifer muss sich genau einmal authentifizieren um die Schwachstelle ausnutzen zu können.
  - None (N): Keine Authentifizierung notwendig.
- **Confidentiality Impact (C):** In welchem Maße die Vertraulichkeit von Informationen durch eine erfolgreich ausgenutzte Schwachstelle verletzt wird, wird durch die Metrik *Confidentiality Impact* bewertet. Mögliche Werte sind:
  - None (N): Keine Auswirkungen auf die Vertraulichkeit der Informationen auf dem angegriffenen System.
  - Partial (P): Einige Informationen auf dem System werden dem Angreifer zugänglich, allerdings hat der Angreifer keinen Einfluss darauf, welche Informationen er erhält. Ein Beispiel ist ein Angriff auf eine Datenbank, durch den der Angreifer Zugriff auf den Inhalt bestimmter Tabellen der Datenbank erhält.
  - Complete (C): Der Angreifer erhält uneingeschränkten lesenden Zugriff auf alle Informationen des Systems.
- **Integrity Impact (I):** In welchem Maße die Integrität von Informationen bzw. des Ziels durch eine erfolgreich ausgenutzte Schwachstelle verletzt wird, wird durch den Faktor *Integrity Impact* bewertet. Mögliche Werte sind:
  - None (N): Keine Auswirkungen auf die Integrität auf dem angegriffenen System.

  - Partial (P): Einige Daten auf dem System können durch den Angreifer modifiziert werden, allerdings hat der Angreifer keinen Einfluss darauf, welche Daten er modifizieren kann, bzw. die Modifikationen sind inhaltlich beschränkt.
  - Complete (C): Der Angreifer kann uneingeschränkt Daten auf dem Ziel modifizieren.
- **Availability Impact (A):** In welchem Maße die Verfügbarkeit von Informationsressourcen durch eine erfolgreich ausgenutzte Schwachstelle verletzt wird, wird durch die Metrik *Availability Impact* bewertet. Mögliche Werte sind:
  - None (N): Keine Auswirkungen auf die Verfügbarkeit des angegriffenen Systems.
  - Partial (P): Die Verfügbarkeit der Informationsressourcen wird beeinträchtigt (etwa durch kurzzeitigen Ausfall oder schlechtere Performance).
  - Complete (C): Durch einen erfolgreichen Angriff wird die Verfügbarkeit soweit beeinträchtigt, dass die Informationsressourcen nicht mehr für legitime Benutzer zur Verfügung stehen.

2.1.2 Temporal Metriken & Temporal Score

Der Temporal Score berücksichtigt die Besonderheiten der Schwachstelle zum Zeitpunkt der Untersuchung. Dadurch wird dem „Vulnerability Lifecycle“[2] Rechnung getragen. Der Temporal Score wird aus dem Base Score und den Temporal Metriken berechnet. Die Metriken und ihre möglichen Werte für den Temporal Score sind:

- **Exploitability (E)**: Diese Metrik bewertet die Verfügbarkeit von Exploit-Code zum Zeitpunkt der Bewertung. Öffentlich verfügbarer oder gar automatisierter Exploit-Code erhöht die Anzahl potentieller Angreifer. Innerhalb des „Exploit Lifecycles“ ist für eine Schwachstelle initial kein Exploit-Code verfügbar, je länger die Schwachstelle bekannt ist, umso ausgereifter und zuverlässiger wird oft der Exploit-Code.
  - Unproven (U): Kein Exploit-Code verfügbar, nur theoretisch ausnutzbar.
  - Proof-of-Concept (POC): Proof-of-Concept Code oder Exploit-Code, der nur unter bestimmten, rein theoretischen Voraussetzungen funktioniert, ist verfügbar.
  - Functional (F): Funktionierender Exploit-Code ist verfügbar, der die Schwachstelle in den Meisten Fällen erfolgreich ausnutzt.
  - High (H): Entweder ist der Exploit-Code in automatischem, mobilem Code (als Malware) verfügbar, oder es ist kein Exploit-Code notwendig um die Schwachstelle auszunutzen.
  - Not Defined (ND): Durch setzen des Wertes „Not Defined“ wird die Metrik in der Berechnung des Temporal Scores ignoriert.
- **Remediation Level (RL):** Oft werden Schwachstellen in Software publiziert, bevor es einen offiziellen Fix vom Hersteller gibt. In der Zwischenzeit kann es publizierte Workarounds oder Fixes von Drittanbietern geben. Diesem Aspekt des „Vulnerability Lifecycles“ wird durch die Metrik Remediation Level Rechnung getragen. Sie bewertet die Verfügbarkeit von Maßnahmen zum Zeitpunkt der Untersuchung.
  - Official Fix (OF): Eine Lösung durch den Hersteller ist entweder in Form eines Patches oder eines Upgrades verfügbar.
  - Temporary Fix (TF): Eine temporäre Lösung ist durch den Hersteller zur Verfügung gestellt worden, z.B. durch einen temporären Hotfix, ein Tool zum Beheben der Schwachstelle oder durch einen offiziellen Workaround.
  - Workaround (W): Ein inoffizieller, nicht vom Hersteller publizierter Workaround oder Patch ist verfügbar.

[2] *http://www.microsoft.com/germany/technet/sicherheit/newsletter/vulnerability.mspx#EMB*

  - Unavailable (U): Entweder existiert keine Maßnahme um die Schwachstelle zu beheben, oder die Maßnahme ist nicht anwendbar.
  - Not Defined (ND): Durch setzen des Wertes „Not Defined“ wird die Metrik in der Berechnung des Temporal Scores ignoriert.
- **Report Confidence (RC):** Diese Metrik bewertet die Vertrauenswürdigkeit der Information über die Existenz der Schwachstelle. Oft wird zunächst nur die Existenz einer Schwachstelle in einem Produkt publiziert, bevor weitere Quellen diese Existenz bekräftigen oder der Hersteller dies bestätigt.
  - Unconfirmed (UC): Eine einzelne Quelle oder mehrere sich widersprechende Quellen behaupten die Existenz der Schwachstelle. Dies kann z.B. die Form eines Gerüchts in einem Hacker-Forum annehmen.
  - Uncorroborated (UR): Mehrere unabhängige Quellen publizieren die Existenz der Schwachstelle z.B. in bekannten Mailinglisten, ggf. mit widersprüchlichen technischen Details.
  - Confirmed (C): Der Hersteller hat die Existenz der Schwachstelle offiziell bestätigt oder funktionierender Exploit-Code in Form eine Proof-of-Concept ist verfügbar, der die Existenz der Schwachstelle bestätigt.
  - Not Defined (ND): Durch setzen des Wertes „Not Defined“ wird die Metrik in der Berechnung des Temporal Scores ignoriert.

2.1.3 Environmental Metriken und Environmental Score

Jedes Unternehmen hat eine eigene Risikokultur und eigene Anforderungen an die Sicherheit; gleichzeitig kann ein spezifisches System für ein Unternehmen hoch-kritisch sein und das gleiche Produkt spielt in einem anderen Unternehmen nur eine Nebenrolle. Diesen umgebungsspezifischen Faktoren tragen die Metriken des Environmental Score Rechnung. Die Metriken und ihre möglichen Werte für den Environmental Score sind:

- **Collateral Damage Potential (CDP):** Diese Metrik bewertet den möglichen Schaden durch eine erfolgreiche Ausnutzung einer Schwachstelle in Bezug auf die untersuchte Umgebung. Der Schaden kann materieller Natur sein (z.B. Diebstahl eines Systems) oder auch als Verlust von Produktivität oder Umsatz bewertet werden.
  - None (N): Es gibt keinen potentiellen Schaden.
  - Low (L): Ein geringer Schaden durch die Ausnutzung der Schwachstelle kann entstehen.
  - Low-Medium (LM): Erfolgreiche Ausnutzung der Schwachstelle kann zu mäßigen Schaden führen.
  - Medium-High (MH): Beträchtlicher Schaden durch eine erfolgreiche Ausnutzung der Schwachstelle kann entstehen.
  - High (H): Der potentielle Schaden durch eine erfolgreiche Ausnutzung der Schwachstelle ist katastrophal.
  - Not Defined (ND): Durch setzen des Wertes „Not Defined“ wird die Metrik in der Berechnung des Temporal Scores ignoriert.
- **Target Distribution (TD):** Die relative Verbreitung von Systemen mit der spezifischen Schwachstelle innerhalb der untersuchten Umgebung wird durch die Target Distribution näherungsweise bestimmt. Dadurch wird die Anzahl betroffener Systeme als Faktor in der Berechnung des Temporal Score mit aufgenommen.
  - None (N): Es existieren keine betroffenen Systeme in der Umgebung oder nur in stark vereinzelter, hoch-spezialisierter Form in Labor-Umgebungen.
  - Low (L): Die betroffenen Systeme machen 1% – 25% der Gesamtumgebung aus.
  - Medium (M): Die betroffenen Systeme machen 26% – 75% der Gesamtumgebung aus.

  - High (H): Die betroffenen Systeme machen 76% – 100% der Gesamtumgebung aus.
  - Not Defined (ND): Durch setzen des Wertes „Not Defined“ wird die Metrik in der Berechnung des Temporal Scores ignoriert.
- **Confidentiality Requirements (CR):** Diese Metrik ermöglicht die Gewichtung von Sicherheitsanforderungen der betroffenen Systeme einer Umgebung in Bezug auf die Vertraulichkeit der von den Systemen zur Verfügung gestellten Informationsressourcen.
  - Low (L): Verletzung der Vertraulichkeit der Informationen auf den betroffenen Systemen hat nur geringe Auswirkungen auf die Organisation oder ihre Mitarbeiter.
  - Medium (M): Verletzung der Vertraulichkeit der Informationen auf den betroffenen Systemen hat deutliche spürbare Auswirkungen auf die Organisation oder ihre Mitarbeiter.
  - High (H): Verletzung der Vertraulichkeit der Informationen auf den betroffenen Systemen hat katastrophale Auswirkungen auf die Organisation oder ihre Mitarbeiter.
  - Not Defined (ND): Durch setzen des Wertes „Not Defined“ wird die Metrik in der Berechnung des Temporal Scores ignoriert.
- **Integrity Requirements (IR):** Diese Metrik ermöglicht die Gewichtung von Sicherheitsanforderungen der betroffenen Systeme einer Umgebung in bezug auf die Integrität der von den Systemen zur Verfügung gestellten Informationsressourcen.
  - Low (L): Verletzung der Integrität der Informationen auf den betroffenen Systemen hat nur geringe Auswirkungen auf die Organisation oder ihre Mitarbeiter.
  - Medium (M): Verletzung der Integrität der Informationen auf den betroffenen Systemen hat deutliche spürbare Auswirkungen auf die Organisation oder ihre Mitarbeiter.
  - High (H): Verletzung der Integrität der Informationen auf den betroffenen Systemen hat katastrophale Auswirkungen auf die Organisation oder ihre Mitarbeiter.
  - Not Defined (ND): Durch setzen des Wertes „Not Defined“ wird die Metrik in der Berechnung des Temporal Scores ignoriert.
- **Availability Requirements (AR):** Diese Metrik ermöglicht die Gewichtung von Sicherheitsanforderungen der betroffenen Systeme einer Umgebung in Bezug auf die Verfügbarkeit der von den Systemen zur Verfügung gestellten Informationsressourcen.
  - Low (L): Beeinträchtigung der Verfügbarkeit der Informationen auf den betroffenen Systemen hat nur geringe Auswirkungen auf die Organisation oder ihre Mitarbeiter.
  - Medium (M): Beeinträchtigung der Verfügbarkeit der Informationen auf den betroffenen Systemen hat deutliche spürbare Auswirkungen auf die Organisation oder ihre Mitarbeiter.
  - High (H): Beeinträchtigung der Verfügbarkeit der Informationen auf den betroffenen Systemen hat katastrophale Auswirkungen auf die Organisation oder ihre Mitarbeiter.
  - Not Defined (ND): Durch setzen des Wertes „Not Defined“ wird die Metrik in der Berechnung des Temporal Scores ignoriert.

2.2 Eine exemplarische Schwachstellen-Bewertung

Zur Verdeutlichung der Methodik wird eine exemplarische Bewertung einer Schwachstelle in zwei fiktiven Webservern einer fiktiven Organisation durchgeführt.

Geprüft wurde eine aus dem Internet erreichbare DMZ-Umgebung in der insgesamt 10 Server mit verschiedenen Funktionalitäten stehen, unter anderem zwei Linux-basierte Web-Server mit einem Apache in der Version 2.0.36. Diese Version des Apache Servers weist eine Schwachstelle auf (CVE-2002-0392[3]), die dazu benutzt werden kann um entweder Code auf dem Server auszuführen oder einen Denial-of-Service Angriff gegen den Apache Server durchzuführen. Einer der beiden Webserver (Server1) ist die „offizielle" Webpräsenz des Unternehmens und dient ausschließlich der Darstellung des Unternehmens im Internet wohingegen der zweite Server (Server2) als Kundenportal mit angeschlossenem Bestellwesen dient. Über dieses System wird der größte Teil der Kundenbestellungen abgewickelt.

Base Score: Die Schwachstelle kann über ein Netzwerk direkt gegen den Apache Server durchgeführt werden und erfordert keine Authentifizierung am System. Da entweder Code ausgeführt werden kann oder ein Denial-of-Service Angriff gegen den Apache Server durchgeführt werden kann, müssen zwei Bewertungen durchgeführt werden Der höhere der beiden Base Scores dient als Grundlage für die weiteren Berechnungen:

1. Ausführung von Code: AV:N/AC:L/Au:N/C:P/I:P/A:N, Base Score: 6,4
2. Denial-of Service: AV:N/AC:L/Au:N/C:N/I:N/A:C, Base Score 7,8

Temporal Score: Exploit Code ist zum Zeitpunkt der Untersuchung verfügbar, aber noch nicht in automatisierter Form, es existiert ein offizieller Patch des Herstellers (Apache Foundation), der die Existenz der Schwachstelle auch offiziell bestätigt hat.

1. E:F/RL:OF/RC:C, Temporal Score: 6,4

Environmental Score: Da zwei verschiedene Systeme mit unterschiedlichen Aufgaben von der Schwachstelle betroffen sind, müssen diese Systeme getrennt betrachtet werden.

1. Server1: Server1 dient exklusiv der Darstellung der Organisation im Internet. Es liegen keine vertraulichen Informationen auf dem System, allerdings kommt der Verfügbarkeit des Systems eine hohe Bedeutung zu.
   CDP:L/TD:L/CR:L/IR:M/AR:M
   Environmental Score: 1,7
2. Server2: Server2 ist ein Kundenportal mit angeschlossenem Bestellwesen. Auf dem System werden vertrauliche Kundendaten verarbeitet und der Verfügbarkeit des Systems kommt eine sehr hohe Bedeutung zu.
   CDP:H/TD:L/CR:H/IR:H/AR:H
   Environmental Score: 2,3

3 PATCHEN MIT CVSS

Ein häufig anzutreffendes Patch-Schema unterscheidet zwischen drei Dringlichkeitsstufen:

1. Emergency Patch: Die Schwachstelle ist derart kritisch für die Umgebung, dass schnellstmöglich alle betroffenen Systeme gepatcht werden müssen; bei Servern wird in der Regel ein „Notwartungsfenster" eröffnet um das Einspielen des Patches nicht bis zum nächsten regulär geplanten Wartungsfenster aufzuschieben. Die Testphase des Patches wird auf ein absolut notwendiges Minimum reduziert.
2. Regular Patch: Patche zu Schwachstellen mittlerer Kritikalität werden über den regulären Patchmanagement-Zyklus eingespielt; bei Servern in der Regel im nächsten

[3] *http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0392*

planmäßigen Wartungsintervall. Die Patche werden vor dem Roll-Out intensiv geprüft und formal freigegeben.

3. Lifecycle Patch: Patche die Schwachstellen niedriger oder sehr niedriger Kritikalität beheben werden häufig nicht über das Patchmanagement-System ausgerollt, sondern im Zuge des Lifecycle-Managements in das nächste Update der Installationsbasis aufgenommen. Dadurch wird die Zahl der zu verwaltenden Patche im regulären Patchmanagement überschaubar gehalten.

Basierend auf der vorhergehenden Darstellung stellt sich die Frage, welcher Patch in welche Kategorie fällt; dazu muss die Schwachstelle bewertet und kategorisiert werden. Die Entscheidungen zur Kategorisierung der Schwachstellen sind oft geprägt von subjektiven Befindlichkeiten, sind nicht transparent, nicht nachvollziehbar und erst recht nicht dokumentiert. Im Rahmen einer Revision, gestiegener Compliance-Anforderungen oder auch einer Konformität zu ISO 27001 kann dieser Missstand gravierende Konsequenzen nach sich ziehen. CVSS bewertet Schwachstellen auf einer Skala von 0 – 10 und die Kategorisierung kann anhand von selbstgewählten Schwellenwerten[4], optimaler weise des Environmental Scores, durchgeführt werden. Zum Einen wird dadurch die Bewertung transparent und nachvollziehbar, zum Anderen beinhaltet diese Herangehensweise ein hohes Automatisierungspotential, inklusive automatisierter Dokumentation, wodurch der gesamte Prozess effizienter durchlaufen werden kann. Ein exemplarischer Prozess mit einem sehr hohen Automatisierungsgrad ist in Abbildung 2 abgebildet und soll an dieser Stelle exemplarisch erörtert werden.

3.1 Ein exemplarischer Patch-Prozess

Die Informationen über Schwachstellen in Softwareprodukten inklusive CVSS Base und Temporal Metriken wird per XML-Feed[5] in einen Parser importiert, der zunächst überprüft, ob die Information eine neue Schwachstelle betrifft oder ob es sich um ein Update einer bestehenden Schwachstelle (z.B. eine Änderung des Temporal Score aufgrund eines neu verfügbaren Exploits) handelt. Entsprechend wird entweder ein neues Ticket eröffnet oder ein bestehendes Ticket modifiziert. Der nächste Schritt besteht in der automatischen Berechnung des Environmental Scores. Dazu muss die Verbreitung der betroffenen Software und das Collateral Damage Potential der Software ermittelt werden. Im Idealfall ist diese Information inklusive eines Verantwortlichens im Zuge eines Business-Continuity Prozess in einer Asset Datenbank verfügbar, andernfalls kann ggf. auf eine Lizenzdatenbank oder ein zentrales Inventar zurückgegriffen werden. Ist keine dieser Quellen verfügbar und soll trotzdem vordringlich ein Patchprozess eingeführt oder optimiert werden, dann bleibt nur der suboptimale Weg über die Erstellung einer Datenbank, die die eingesetzten Produkte inklusive der Verbreitung und des Collateral Damage Potential erstellt werden. Basierend auf den definierten Schwellenwerten wird das Ticket kategorisiert:

- Score = 0: Schwachstelle betrifft das Unternehmen nicht, dokumentieren, Ticket schliessen.
- Score < Schwellenwert 1: Lifecycle Patch, Ticket zur Bearbeitung an das Lifecyclemanagement Team weiterleiten um den Patch in den nächsten Lifecycle Release aufzunehmen. Kategorisierung in der Dokumentationsdatenbank festhalten.

---

[4] Die Definition der Schwellenwerte muss an die Risikokultur des Unternehmens angepasst werden. Eine aggressive Patchpolitik würde mit niedrigeren Schwellenwerten (z.B. 0,1 – 1,5: Lifecycle Patch, 1,6 – 6,0 Regular Patch, 6,1 – 10, Emergency Patch) arbeiten als eine eher konservativ angelegt Patchpolitik (z.B. 0,1 – 3,0 Lifecycle Patch, 3,1 – 8,9 Regular Patch, 9,0 – 10,0 Emergency Patch) arbeiten.

[5] Das NIST stellt die National Vulnerability Database als kostenlosen XML-Feed inklusive CVSS-Base-Metriken unter http://nvd.nist.gov/ zur Verfügung. Dieser Feed beinhaltet allerdings keine Temporals Metriken, die im beschriebenen Prozess benötigt werden. Eine Übersicht, welche kommerziellen Security Provider (Secunia, Cisco, etc.) XML-Feeds inklusive Temporal Metrike zur Verfügung stellen, ist dem Autor nicht bekannt.

- Score < Schwellenwert 2: Regular Patch, Ticket zur Bearbeitung an die verantwortliche Fachabteilung zwecks Test des Patches und ggf. Freigabe weiterleiten. Kategorisierung in der Dokumentationsdatenbank festhalten.
- Score > Schwellenwert 2: Emergency Patch, high-priority Ticket zur Bearbeitung an die verantwortliche Fachabteilung zwecks Emergency Test des Patches und ggf. Freigabe weiterleiten. Kategorisierung in der Dokumentationsdatenbank festhalten.

Der Test und die ggf. daraus folgende Freigabe des Patches sind die einzigen Schritte im Prozess, die eine manuelle Bearbeitung erfordern. Insbesondere ist die Bewertung vollständig automatisiert, wodurch subjektive Faktoren ausgeschlossen werden. Durch die Dokumentation der Bewertung kann jederzeit überprüft werden, warum ein bestimmter Patch zu einem gewissen Zeitpunkt nicht eingespielt wurde. Ein weiterer Vorteil dieses Ansatzes ergibt sich in der Handhabung von upgedateten Informationen. Häufig stehen zum initialen Release einer Schwachstelle noch nicht alle Informationen zur Verfügung, oder aber es ist zum Beispiel kein Exploit verfügbar. Update-Informationen im XML-Feed können gegen die Dokumentations-Datenbank abgeglichen werden. Dadurch kann sich die Bewertung ändern und das entsprechende Ticket kann entweder wieder eröffnet werden oder ein noch offenes Ticket kann anders kategorisiert werden.

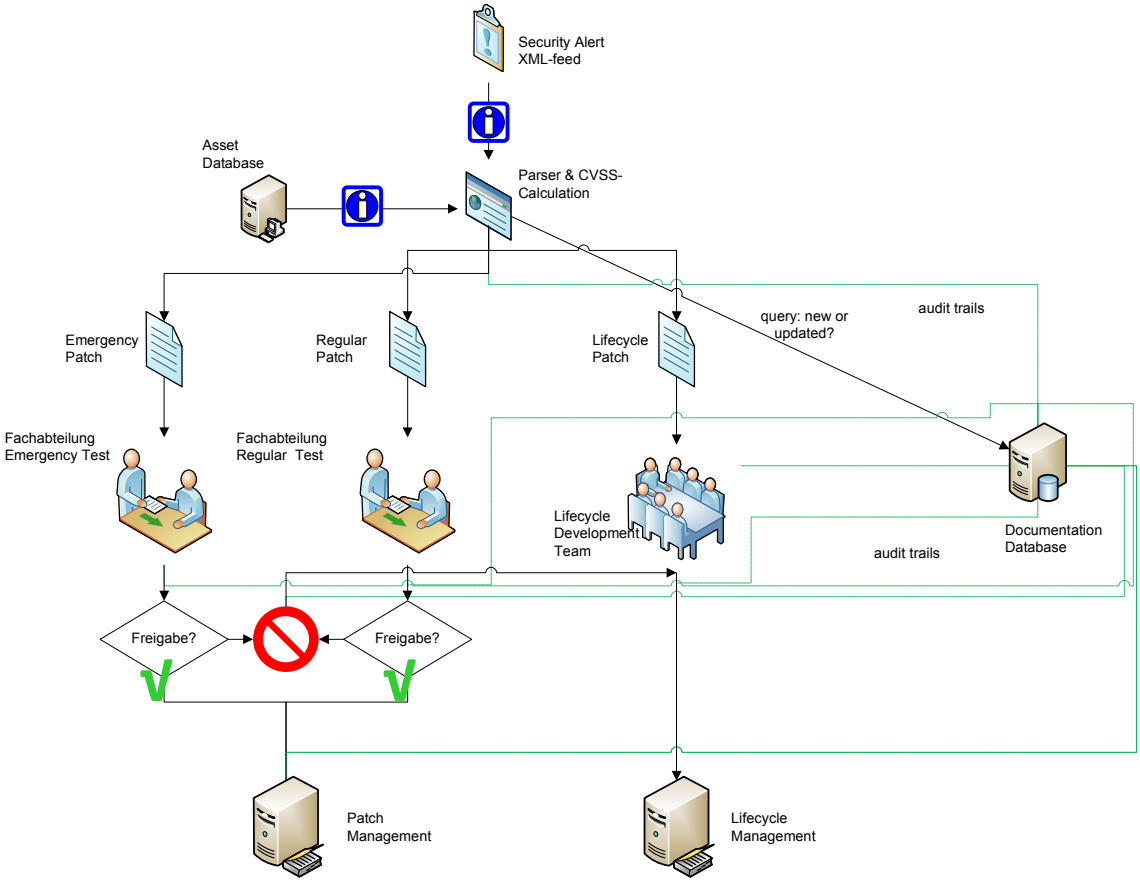

Abbildung 2 Patch-Management Prozeß mit CVSS

3.2 Kritische Faktoren

Die kritischen Punkte im Prozess sind nach Ansicht des Autors insbesondere die Asset Datenbank, die die notwendigen Informationen enthält um den Environmental Score zu berechnen und die

Definition der Schwellenwerte. Die Asset Datenbank sollte im Zuge eines proaktiven Risikomanagements bzw. Business Continuity Managements erstellt sein und regelmäßig aktualisiert werden. Ohne die dort enthaltenen Informationen läuft der Prozess ins Leere. Die Definition der Schwellenwerte kann nur sinnvoll unter Berücksichtigung der Risikokultur, der fallbasierten Modellierung der CVSS-Werte und der Analyse historischer Daten geschehen. Die fallbasierte Modellierung hat das Ziel CVSS-Scores zu definierten Szenarien zu berechnen und die Szenarien den Patch-Kategorien zuzuordnen. Ein Beispiel für eine fallbasierte Definition ist in Abbildung 3 abgebildet[6]. Die Schwachstelle hat einen sehr hohen Base-Score, einen hohen Temporal-Score, aufgrund der geringen Target-Distribution (ggf. wird die Software in der Umgebung nur auf einem einzigen System eingesetzt) allerdings nur einen niedrigen Environmental-Score. Obwohl das System potentiell extrem wichtig für das Unternehmen ist, ist der Environmental-Score (ggf. unangemessen) niedrig. Dieses Extrembeispiel verdeutlicht die Problematik der Definition sinnvoller Schwellenwerte.

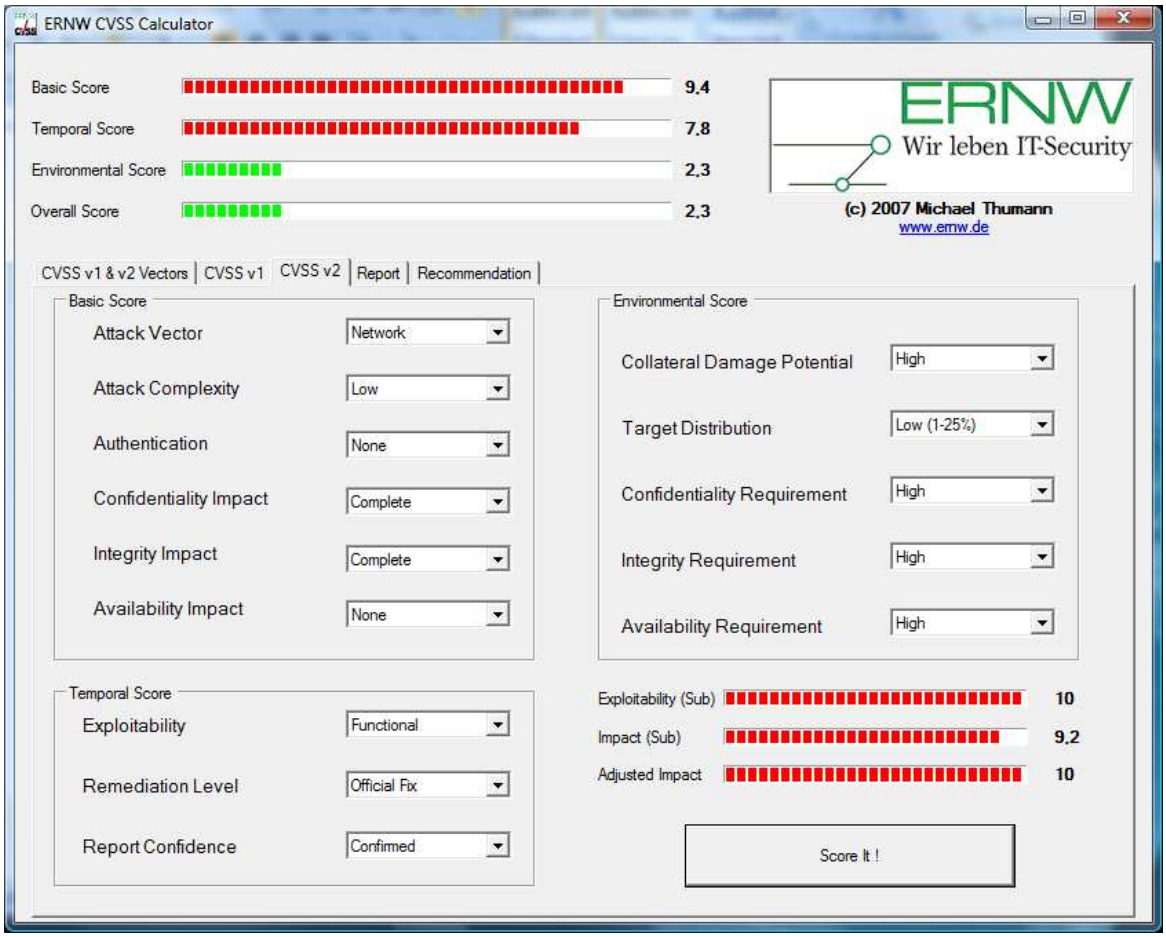

Abbildung 3 Fallbasierte Analyse zur Bestimmung der Schwellenwerte

Die historische Analyse zeigt, dass nicht alle theoretisch möglichen CVSS-Scores in der Praxis auch vorkommen, sondern das charakteristische Häufungen um diskrete Scores herum auftreten (siehe

[6] *Die Analyse wurde mit Hilfe des ERNW-CVSS-Calculators erstellt, der sowohl CVSS 1.1 als auch CVSS 2.0 Scores berechnen kann, das Speichern und Laden von Bewertungs-Dateien ermöglicht und über die ERNW-Website frei verfügbar ist: http://www.ernw.de*

Abbildung 4).

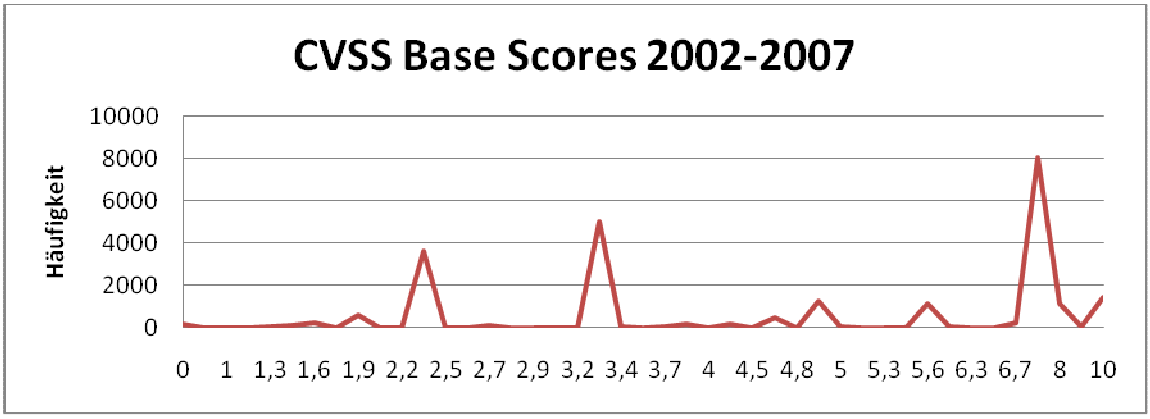

Abbildung 4 Base-Scores der Schwachstellen 2002-2007

Die Entwickler von CVSS 2.0 haben auch eine „offizielle“ Empfehlung der Schwellenwerte veröffentlicht:

Score 0 – 3: „Wait for next Service Pack“
Score 4 – 5: „Next Patch Cycle“
Score 6 – 7: „Patch within 6-7 days“
Score > 7: “Patch as soon as possible”

Die Definition von sinnvollen Schwellenwerten zur Kategorisierung ist immer ein Kompromiss – trotzdem sollte die Definition möglichst sorgfältig durchgeführt werden und die Auswirkungen der gewählten Schwellenwerte sollten bedacht, dokumentiert und den relevanten Parteien kommuniziert werden.

4 ZUSAMMENFASSUNG

CVSS kann ein wertvolles Werkzeug zur Bewertung von Schwachstellen sein und kann auch im Zuge einer Automatisierung innerhalb des Patchprozesses sinnvoll eingesetzt werden um Patche zu kategorisieren und zu priorisieren. Die Bewertung anhand von definierten Kriterien behebt die Problematik der subjektiven oder herstellerabhängigen Bewertung und führt zu einer einheitlichen Methodik mit reproduzierbaren Ergebnissen.

Die fallbasierte Analyse zeigt allerdings auch recht schnell die logischen Beschränkungen des Systems auf – im Randbereich „kritische Schwachstellen in wenig-verbreiteter Software“ kommt es systembedingt zu vergleichsweise niedrigen Bewertungen, die ggf. dazu führen, dass kritische Systeme nicht, oder erst verspätet, gepatcht werden. Diesem Nachteil steht allerdings der große Vorteil einer transparenten, nachvollziehbaren und einheitlichen Bewertung gegenüber. Michael Thumann, Senior Security Consultant der ERNW GmbH, hat einen CVSS-Kalkulator für die CVSS Versionen 1.1 und 2.0 programmiert, der auf Wunsch kostenlos per Email (an Roland Fiege, rfiege@ernw.de) erhältlich ist.

Mit freundlichen Grüßen,

[Dror-John Röcher].

ERNW GmbH
Dror-John Röcher
Senior Security Consultant

ERNW Enno Rey Netzwerke GmbH
Breslauer Str. 28
69124 Heidelberg
Tel. +49 6221 480390
Fax +49 6221 419008
Mobil +49 173 6745905
www.ernw.de