Report Penetrationstest

beA-Kanzleisoftware-Schnittstelle

Auftraggeber: [redacted]

Version:	*1.0*
Datum:	*29.04.2016*
Status:	***Final***
Seitenanzahl:	*21*
Klassifizierung:	*Vertraulich*

Autor: [redacted]
E-Mail: [redacted]@atos.net
Atos IT Solutions and Services GmbH

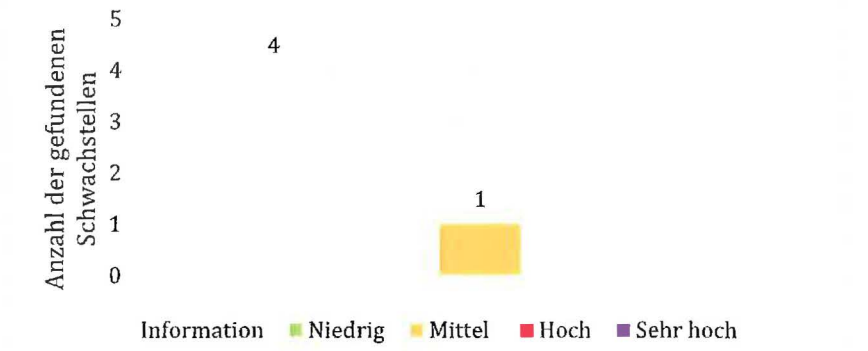
Inhaltsverzeichnis

1 Management Summary

Für die Applikation beA-Kanzleisoftware-Schnittstelle wurde im Zeitraum vom 14.03.2016 bis 22.03.2016 ein Penetrationstest durchgeführt. Ziel des Penetrationstests war die Überprüfung der SOAP Schnittstelle beA-Kanzleisoftware-Schnittstelle auf Schwachstellen der OWASP Top 10 und Schwächen im Rollen- und Rechtemodell.

Die beA-Kanzleisoftware Schnittstelle ermöglicht es Herstellern von Kanzleisoftware Lösungen den Zugriff auf das beA in Ihre Produkte zu integrieren. Die Kanzleisoftware-Schnittstelle besteht aus einer fachlichen Schnittstelle zum Zugriff auf Postfächer und Nachrichten und einem dazugehörigen optionalen Toolkit, welches die Ver- und Entschlüsselung von Nachrichten und von Anhängen durchführt und die Signaturerstellung und Prüfung anbietet.

Grundsätzlich ergaben sich aus dem Penetrationstest 1 Schwachstelle mit mittlerer Kritikalität und 4 Informationen.



Zusammenfassung der gefundenen Schwachstellen

Die Bewertung der gefundenen Schwachstellen erfolgt aus rein technischer Sicht auf Basis des CVSS Base Score.

Maßnahmen sollten sein, dass Schwachstellen mit der Kritikalität „Sehr hoch" und „Hoch" umgehend bereinigt werden müssen, während Schwachstellen mit der Kritikalität „Mittel" bereinigt werden sollten.
Des Weiteren wäre es sinnvoll die als „Niedrig" eingestuften Schwachstellen ebenfalls zu bereinigen.

Die Applikation ist anfällig für Denial-of-Service Angriffe. Durch eine hohe Anzahl automatisch generierter Requests kann die Erreichbarkeit des Servers temporär unterbrochen werden.

Das untrustedCertificate, welches mit dem Webservice Aufruf getConfiguration abgerufen wird, kann ausgetauscht werden. Dadurch ist eine erfolgreicher Man-in-the-Middle möglich.

Verschiedene Fehlerseiten beinhalten Informationen, wie z.B. Server Version oder möglichen Entry Points.

Die TLS Verschlüsselungskonfiguration konnte aus dem Atos-Netz heraus nicht getestet werden. Diese sollte vor der Produktivsetzung auf mögliche Schwachstellen untersucht werden.

2 Auftragsdetails

Untersuchungsgegenstand und überprüfte Komponenten:

- Name der Applikation
  - Applikation: beA-Kanzleisoftware-Schnittstelle
  - URL: https://test.bea-brak.de
  - Webservice: beAPortType
- Ziel des Penetrationstests
  - Penetrationstest OWASP Top 10
  - Überprüfung Rollen- und Rechtemodell
- Out of Scope
  - Denial of Service
- Testzeitraum
  - 14.03.2016– 22.03.2016
- Benutzerkonten, Rollen und Rechte
  - Zertifikat: [redacted]
  - Zertifikat: [redacted]
  - Zertifikat: [redacted]
- Verwendete Quell IP Adresse:
  - IP Adresse: Atos Intranet
- Rahmenbedingungen
  - Der Penetrationstest wurde aus dem Atos Intranet durchgeführt.
  - Der CVSS Temporal und Environmental Score finden keine Anwendung bei der Gewichtung der Schwachstellen.

Auftraggeber:

- [redacted]@atos.net, +49 [redacted]

Auditor:

- [redacted]@atos.net, +49 [redacted]

3 Schwachstellenübersicht

Entsprechend der international anerkannten Organisation „National Vulnerability Database (NVD) of the National Institute of Technology (NIST)" wird die empfohlene Bewertung des Risikos anhand der CVSS v3.0 Base Score Metrik entsprechend der folgenden Tabelle angegeben.

Risikobewertung	CVSS Score
Sehr hoch	9,0-10
Hoch	7,0 - 8,9
Mittel	4,0 - 6,9
Niedrig	0,1 - 3,9
Information	0

3.1 Übersicht nach Schwachstellenkategorien

	CVSS Score				
Schwachstelle	**Info**	**Niedrig**	**Mittel**	**Hoch**	**Sehr Hoch**
Information Gathering					
Server Version wird auf Fehlerseite preisgegeben	0				
Informationen auf Fehlerseite	0				
Configuration and Deployment Management Testing					
Fremdes UntrustedCertificate wird akzeptiert	0				
Denial of Service Angriff möglich			5		
Identity Management Testing					
Authentication Testing					
Authorization Testing					
Session Management Testing					
Input Validation Testing					
Testing for Error Handling					
Testing for weak Cryptography					
Verschlüsselungskonfiguration konnte nicht getestet werden	0				
Business Logic Testing					
Client Side Testing					

3.2 Zusammenfassung der Empfehlungen

In diesem Kapitel werden die empfohlenen Maßnahmen gesammelt aufgelistet bzw. zusammengefasst. Ziel ist es, sich schnell einen Überblick über die nötigen Schutzmaßnahmen verschaffen zu können. Die einzelnen genaueren Beschreibungen der Maßnahmen werden jeweils den betreffenden
Schwachstellen beigefügt.

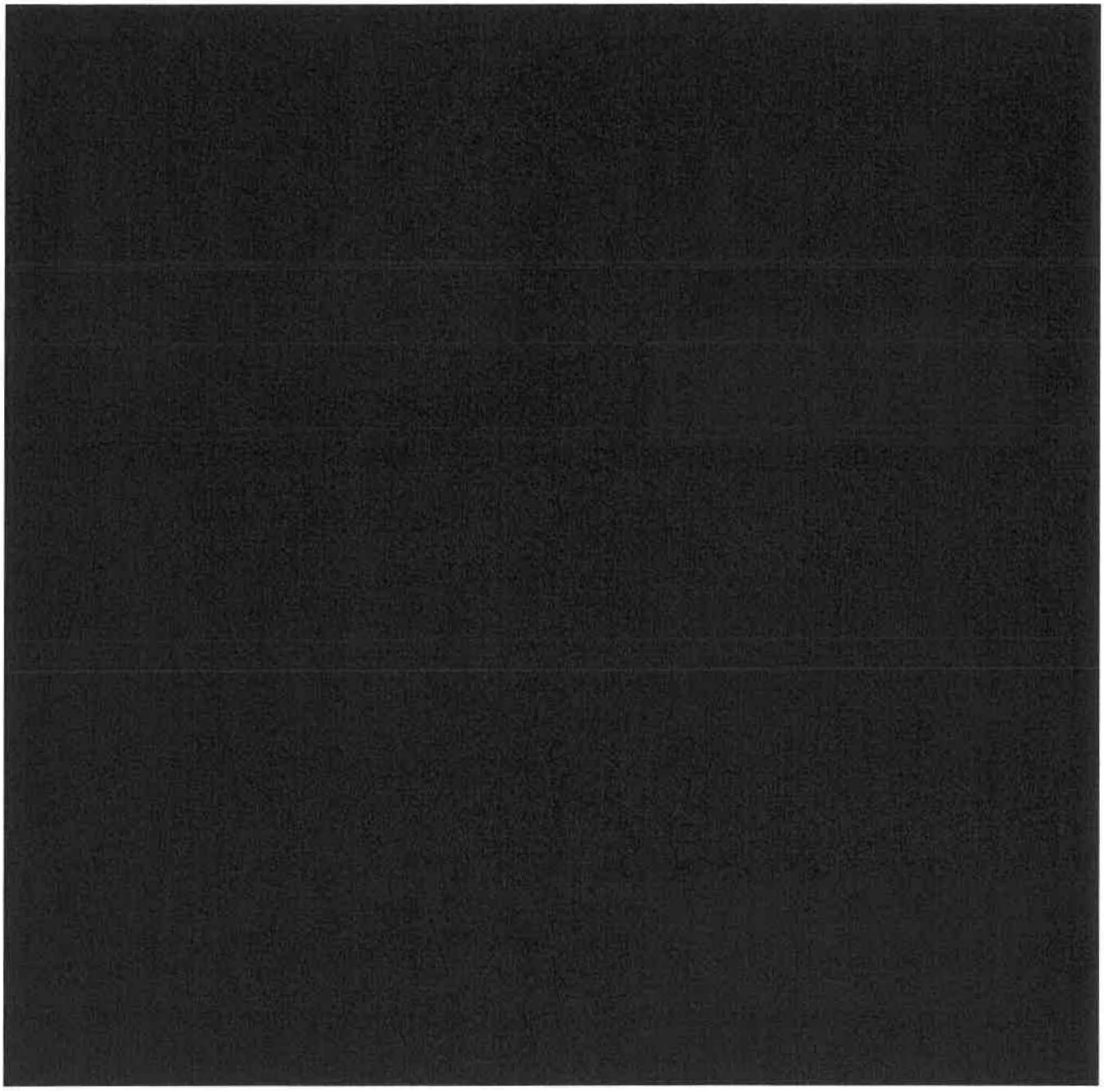
Folgende Maßnahmen werden vorgeschlagen:

1. Gegen DoS-Angriffe sollte ein entsprechender Schutzmechanismus realisiert werden, z.B. Requestlimitierung auf Benutzer.
2. Falsche Zertifikate sollten auf [redacted] erkannt und geblockt werden.
3. Fehlerseiten sollten bereinigt werden und unnötige Informationspreisgaben entfernt werden
4. Die Verschlüsselungskonfiguration sollte vor Produktivsetzung auf mögliche Schwachstellen überprüft werden.

Für die Priorisierung von Maßnahmen zur Reduktion von potentiellen Schwachstellen ist nicht nur die bloße Anzahl von Schwachstellen entscheidend. Viel wichtiger ist die Kritikalität der Schwachstellengruppen – also das Schadenspotential das entsteht, wenn eine Schwachstelle ausgenutzt werden würde. Z.B. sind eine fehlende Authentifizierung oder ein Standardpasswort kritischer zu bewerten als die Schwachstelle eines veralteten Verschlüsselungsalgorithmus.

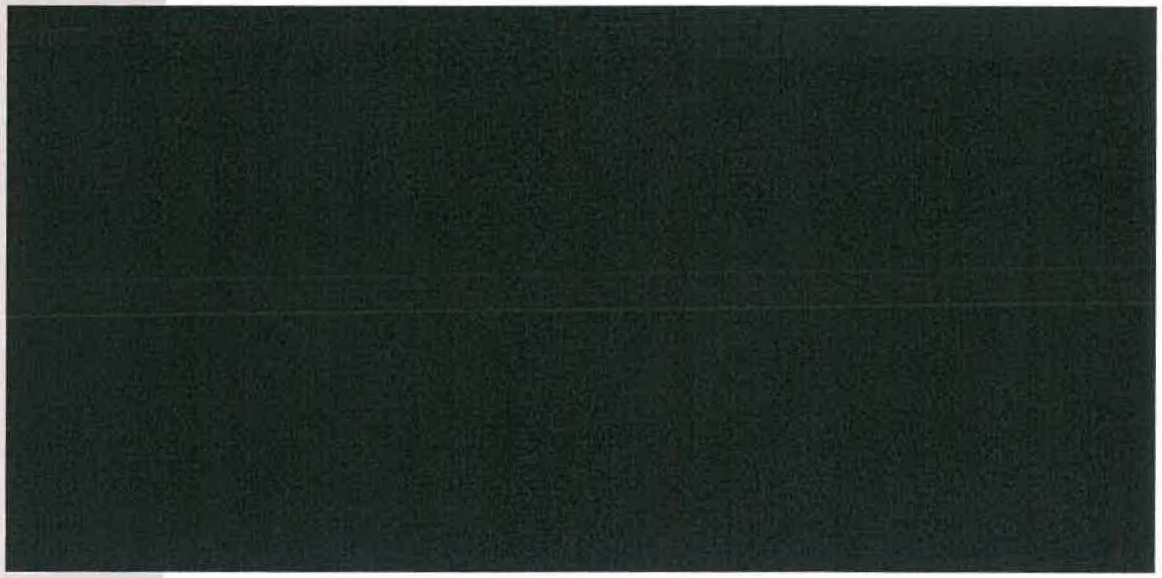
4 Durchgeführte Tests und Ergebnisse

In den folgenden Unterkapiteln werden die Schwachstellen und Funde angelehnt an den OWASP Testing Guide gruppiert und detailliert dokumentiert.



4.1.2 Informationen auf Fehlerseite

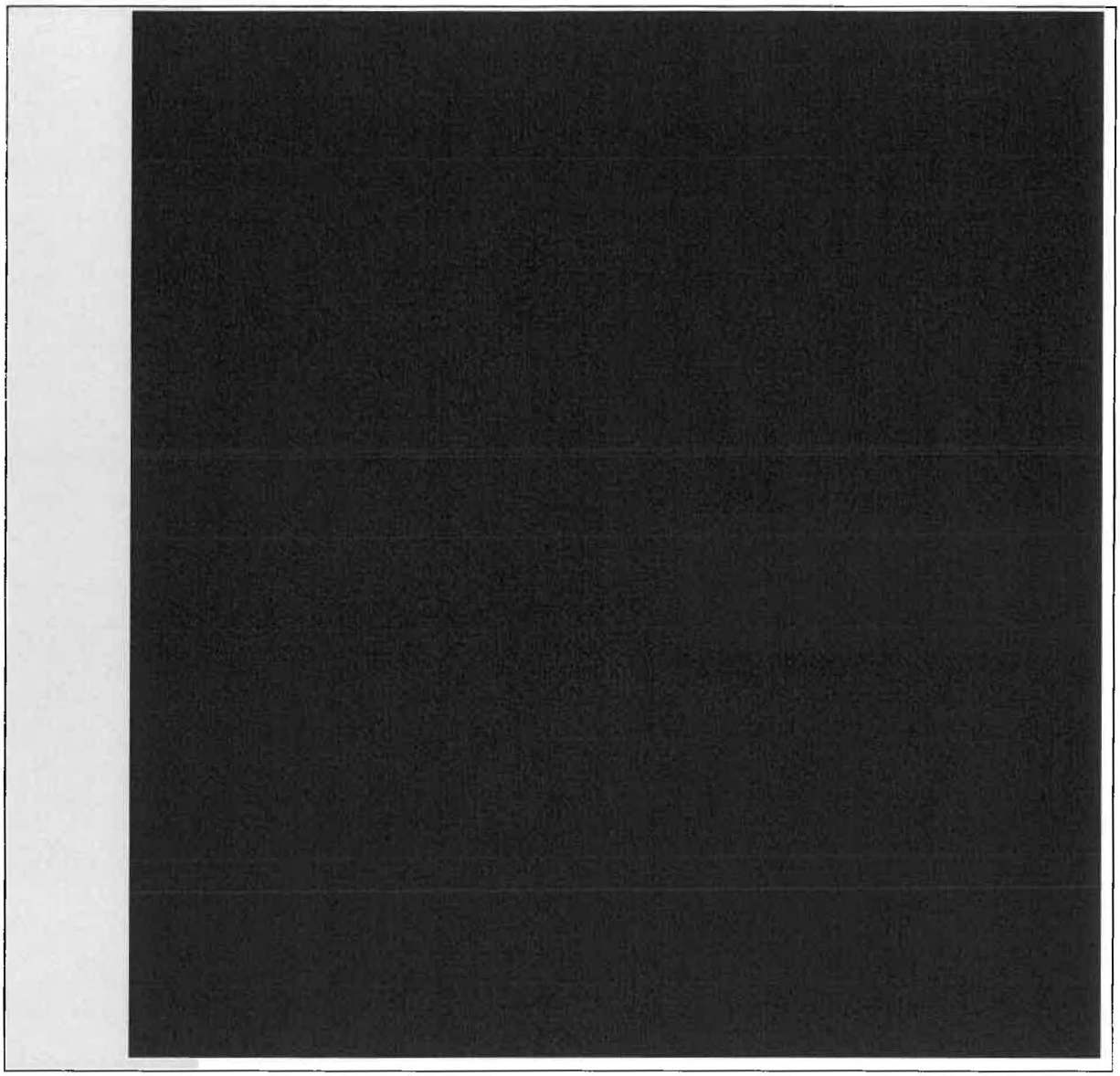
Kritikalität:	**Information**
CVSS Base Score:	0
Base Vektor:	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:N/A:N
Schwachstelle:	Die Applikation soll nur über ein vorgeschaltetes Proxy System (oder Load Balancer) erreichbar sein. Über den Aufruf von mindestens einer URL antwortet das System mit einem Verweis auf das eigentlich verwendete Back-End- oder ein anderes System. Ein Angreifer kann mit dem Wissen um die zugrunde liegenden Back-End Systeme sein Angriffe zielgerichteter durchführen.
Maßnahme:	Wenn Systeme hinter Load Balancern oder Proxies betrieben werden, sollten sämtliche Verweise von IP Adressen oder FQDN auf die vorgeschalteten LoadBalancer/Proxies zeigen.
URL:	https://test.bea-brak.de/bea/BeAPortType/forgot_password.%7Bpb
Nachweis:	Die o.g. URL verursacht bei Eingabe eine Fehlerseite, welche einige Informationen, u.a. eine IP-Adresse enthält.



4.2 Configuration and Deployment Management Testing

4.2.1 Fremdes UntrustedCertificate wird akzeptiert

Kritikalität:	**Information**
CVSS Base Score:	0
Base Vektor:	CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N
Schwachstelle:	Das mit dem Webservice getConfiguration angefragte untrustedCertificate kann durch eine Man-in-the-Middle Angriff ausgetauscht werden. Während des Tests konnte in der Server Response zu diesem Webservice das ursprüngliche untrustedCertificate gegen das Zertifikat der Man-in-the-Middle Software ausgetauscht werden, welches auch dem Java-Keystore hinzugefügt wurde. Governikus lässt die Anmeldung zu und der Traffic kann durch die Software mitgelesen werden. Hinweis: Da es sich bei diesem Finding um einen Angriff auf die Clientinstanz handelt und der SOAP Webservice selbst keine Verwundbarkeit aufweist, wird dieser Befund lediglich als „Information" bewertet.
Maßnahme:	Auf Governikus-Seite sollte ein entsprechender Whitelisting-Ansatz realisiert werden, der das Einfügen eines solchen Zertifikates verhindert.
Webservice:	getConfiguration
Nachweis:	[redacted]



4.2.2 Denial of Service Angriff möglich

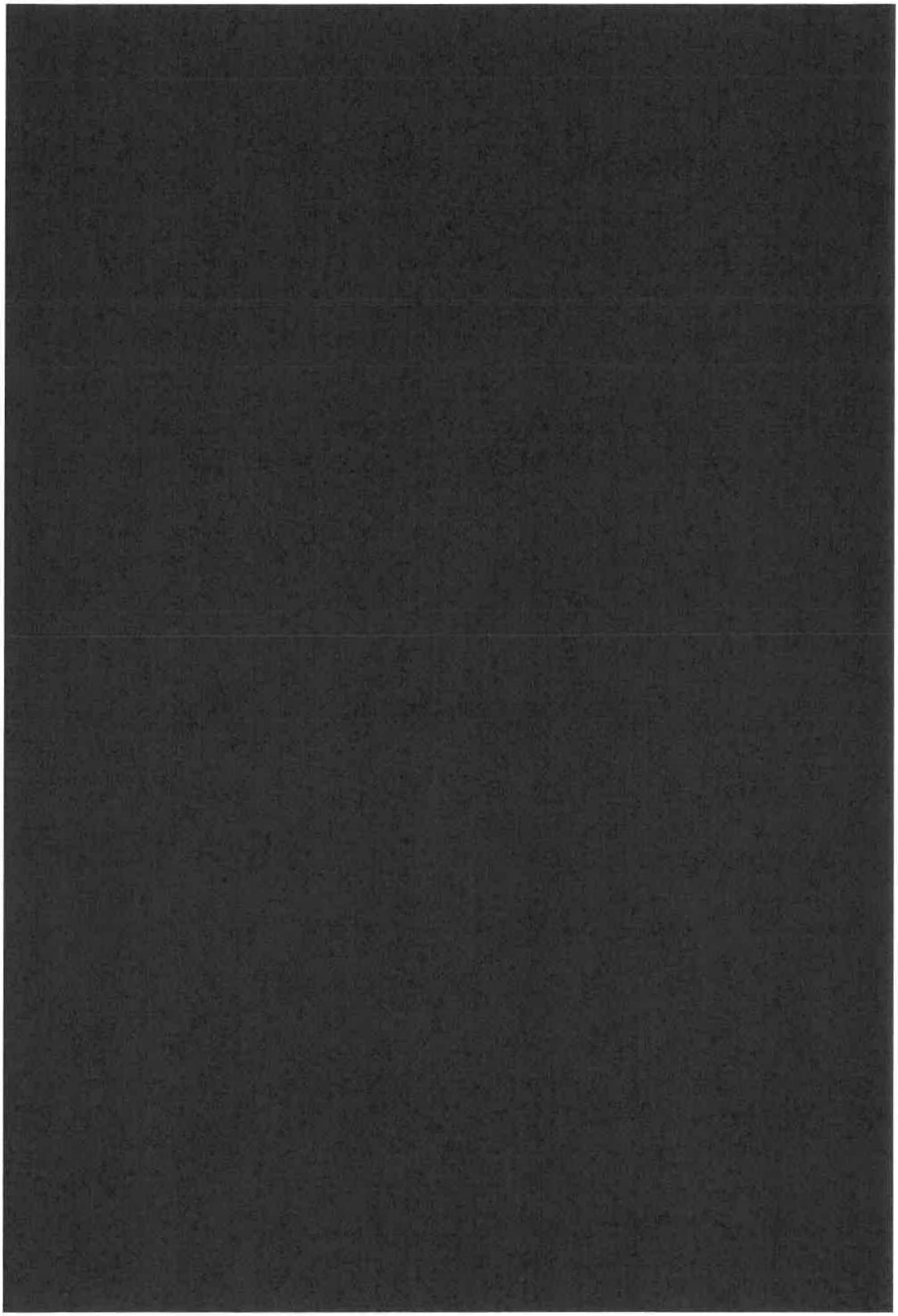
Kritikalität:	**Mittel**
CVSS Base Score:	5
Base Vektor:	CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:N/I:N/A:L
Schwachstelle:	Das Backend-System kann durch Denial of Service Angriffe temporär außer Betrieb gesetzt werden. Während des Tests ist das aufgefallen, als ein automatischer Scanner die Applikationsstruktur auf mögliche Schwachstellen untersuchen sollte und zu diesem Zweck eine große Anzahl an Requests an das System gesendet hat. Nach kurzer Zeit hat das System nicht mehr reagiert und konnte nicht mehr erreicht werden. Angreifer haben dadurch die Möglichkeit, für möglicherweise schwerwiegende Systemausfälle zu sorgen.
Maßnahme:	Das System sollte einen entsprechenden Schutz gegen solche Denial of Service Angriffe aufweisen. Beispielsweise könnten Benutzer, von denen eine ungewöhnlich hohe Anzahl an Requests kommt, anhand der IP gesperrt werden.
URL:	https://test.bea-brak.de
Nachweis:	Nach kurzer Verwendung eines automatischen Scanners war das Backendsystem temporär nicht mehr erreichbar.

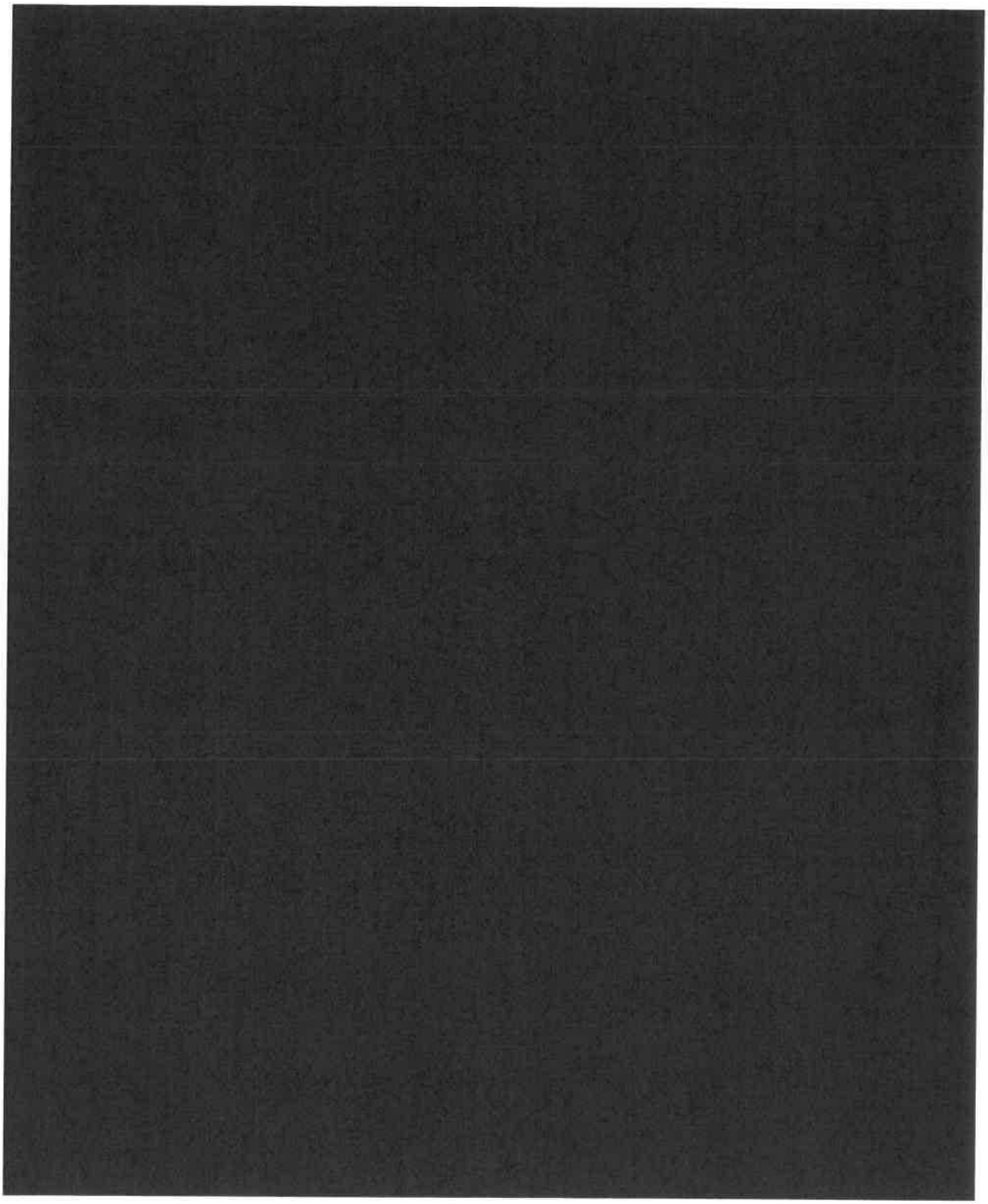
4.3 Testing for weak Cryptography

4.3.1 Verschlüsselungskonfiguration konnte nicht getestet werden

Kritikalität:	**Information**
CVSS Base Score:	0
Base Vektor:	CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:U/C:N/I:N/A:N
Schwachstelle:	Zum Aufbau abgesicherter Verbindungen und zum Verschlüsseln der Datenübertragung kann Secure Sockets Layer (SSL) bzw. Transport Layer Security (TLS) eingesetzt werden. Bei der sicheren Kommunikation zwischen einem Client und Server, können mittels SSL/TLS unterschiedliche Technologien, Algorithmen und Zertifikatstypen verwendet werden. Einige der Algorithmen, die zur Verschlüsselung und Signierung eingesetzt werden können, sind jedoch veraltet bzw. kompromittiert. Ein Angreifer kann Schwachstellen in veralteten Algorithmen dazu verwenden, um die Vertraulichkeit und Integrität der Kommunikation zu verletzen. Hinweis: Es war nicht möglich, die eingestellte Serverkonfiguration auf unsichere Verschlüsselungsalgorithmen zu testen.
Maßnahme:	Vor Produktivsetzung sollte der entsprechende Server erneut geprüft und die Konfiguration ggf. angepasst werden.
URL:	https://test.bea-brak.de

4.4 Durchgeführte Testszenarien

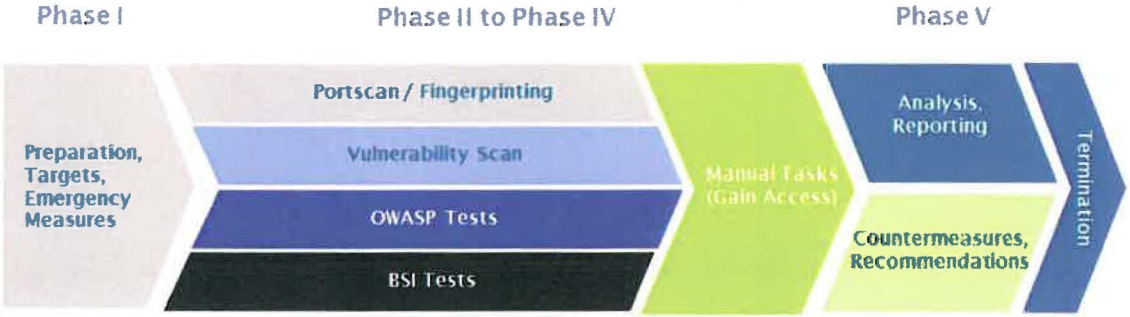




5 Allgemeine Informationen

5.1 Test Durchführung

Grundsätzlich sind die Penetrationstests für Web-Applikationen an den OWASP1 Testing Guide angelehnt, welcher ein großes und effizientes Spektrum für die Überprüfung dieser Services bietet.



Test Methodology (Phasenmodell)

Phase 1:
Bevor die eigentlichen Tests und Scans beginnen, müssen zunächst in Phase 1 die entsprechenden rechtlichen Bedingungen und weitere Anforderungen sowie der genaue Scope des Tests spezifiziert werden. Sind sämtliche organisatorischen Rahmenbedingungen ausgehandelt kann mit dem Start von Phase 2 begonnen werden.

Phase 2:
In diesem Abschnitt des Penetrationstests wird das Testobjekt durch erste Analysen wie Port-Scans und Foot-printing analysiert. Des Weiteren werden in dieser Phase automatisierte Scan-Tools zum Einsatz gebracht, um so viele Informationen wie möglich über das Testobjekt zu sammeln und um erste automatisierte Angriffe auf die Systeme durchzuführen. Auf Basis der Ergebnisse der eingesetzten Tools können erste potentielle Schwachstellen eruiert werden, welche dann von den Tools bereits aktiv ausgenutzt werden. Auf Basis dieser Informationen und der Analyse aus Phase 3 werden dann weitere manuelle Attacken in Phase 4 ausgeführt.

Phase 3:
Es werden die entsprechenden Scan-Ergebnisse ausgewertet und analysiert. Dabei werden zum einen die bereits durch die Tools ausgeführten Angriffe ersichtlich und zum anderen können die gesammelten Informationen der Scans für die Ausführung von manuellen Angriffen in Phase 4 genutzt werden.

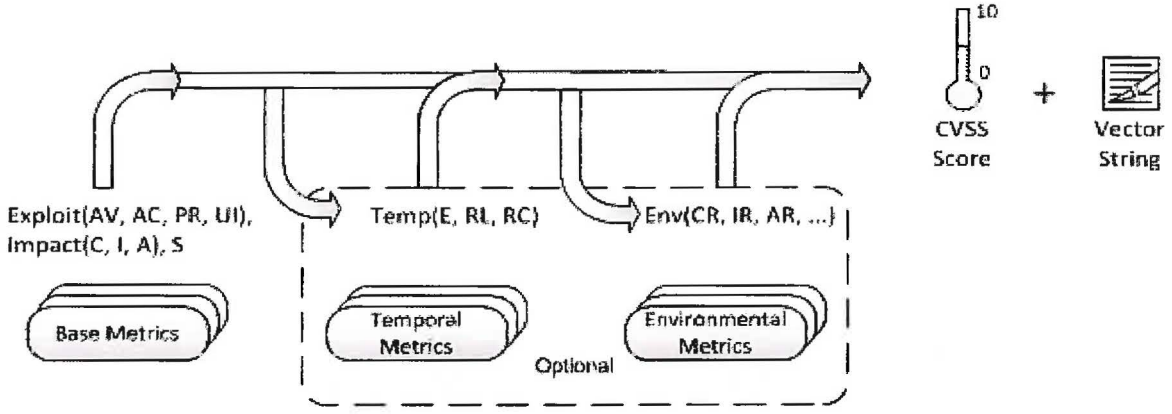
Phase 4:
Basierend auf den Scan-Ergebnissen aus Phase 2 sowie der genaueren Analyse in Phase 3 werden dann weitere manuelle Attacken auf das Testobjekt durchgeführt. In dieser Phase wird versucht Schwachstellen auszunützen um beispielsweise Angriffe für die Erweiterung von Rechten (Privilege Escalation) oder zur generellen Überprüfung von Rechten (Access Control) durchzuführen. Zusätzlich werden weitere Angriffe wie XSS oder SQL-Injection durchgeführt.

Phase 5
Abschließend werden die gefundenen Schwachstellen und Funde basierend auf dem CVSSv3 bewertet und dokumentiert. Dabei werden die Wahrscheinlichkeit der Ausnutzung sowie schwere der Schwachstelle und weitere Parameter einbezogen, um einen entsprechenden Risk Score zu generieren.

[1] Das Open Web Application Security Project (OWASP) ist eine offene Community mit dem Ziel, Unternehmen und Organisationen zu unterstützen, sichere Anwendung zu entwickeln, zu kaufen und zu warten. https://www.owasp.org/images/5/52/OWASP_Testing_Guide_v4.pdf

5.2 Common Vulnerability Scoring System (CVSS)

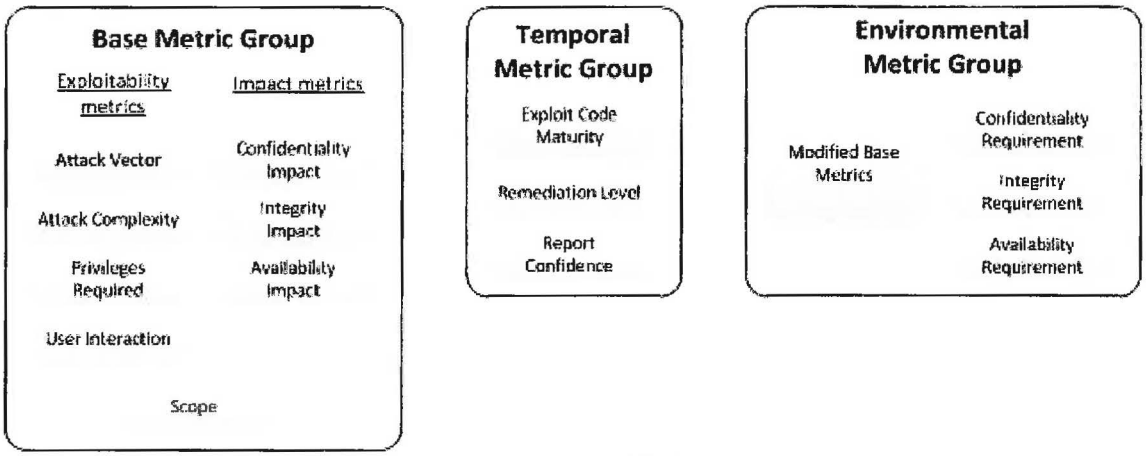
Software, Hardware und Firmware Schwachstellen stellen ein kritisches Risiko für jede Organisation dar die ein Computer Netzwerk betreibt, dabei sind diese schwer zu kategorisieren und abzuschätzen. CVSS2 bietet einen Weg die grundlegenden Charakteristiken von Schwachstellen zu erfassen und deren Schweregrad durch einen numerischen Score darzustellen. Weiterhin wird der Score auch textuell dargestellt. Der Score kann als qualitative Repräsentation (Information, niedrig, mittel, hoch, sehr hoch) übersetzt werden um Unternehmen zu helfen ihre Schwachstellen-Management-Prozesse richtig zu priorisieren und zu bewerten.



CVSS Metriken und Formeln

Die Bewertung einer Schwachstelle erfolgt anhand von drei Kategorien:

- Base Metric - Eigenschaften, die sich im Lauf der Zeit nicht ändern
- Temporal Metric - Eigenschaften, die sich im Lauf der Zeit ändern können
- Environmental Metric - Eigenschaften, die sich auf das Umfeld eines Systems beziehen



CVSS Metrik Gruppen

Atos verwendet für die Bewertung einer Schwachstelle die Base Metric Group. Die Temporal Metric Group und Environmental Metric Group finden keine Anwendung bei der Gewichtung der gefundenen Schwachstellen.

Die folgenden Kapitel beschreiben für das allgemeine Verständnis den Aufbau und Anwendung der Bases Metric Group.

[2] CVSS - Common Vulnerability Scoring System
http://www.first.org/cvss

5.2.1 Base Metric Group

Die Base Metric Group repräsentiert die wesentlichen Charakteristiken einer Schwachstelle die konstant über eine bestimmte Zeit und über die Benutzerumgebung hinaus vorhanden sind. Sie ist aus der Exploitability Metric, der Impact Metric und dem Scope zusammengesetzt.

5.2.1.1 Exploitability Metrics

Die Exploitability Metrics zeigt die Charakteristik der angreifbaren Komponenten der Schwachstelle. Daher sollte jede der unten gelisteten Exploitability Metrics relativ zu den angreifbaren Komponenten gewertet werden und die Eigenschaften der Schwachstelle wiederspiegeln die zu einem erfolgreichen Angriff führen.

Attack Vector (AV)

Diese Metrik beschreibt die Umstände unter denen eine Schwachstelle ausnutzbar ist. Der Wert dieser Metrik wird größer je weiter entfernt (logisch und physisch) der Angreifer sein kann um die schwache Komponente auszunutzen. Es besteht die Annahme, dass die Anzahl von potentiellen Angreifern einer Schwachstelle die vom Internet ausnutzbar ist größer ist, als die Anzahl potentiellen Angreifer die die Schwachstelle nur durch physischen Zugang zu einem Gerät ausnutzen können, wodurch ein höherer Score entsteht. Die Liste von möglichen Werten ist in den folgenden Punkten aufgeführt.

- Network (N)

  Eine Schwachstelle die über einen Netzwerkzugang ausnutzbar ist bedeutet, dass die angreifbare Komponente an den Network Stack gebunden und der Zugriff über OSI Layer 3 (Netzwerk-Schicht) stattfindet. Eine solche Schwachstelle wird oft „remotely exploitable" genannt. Ein Beispiel für einen Netzwerk Angriff ist ein Angriff der einen denial of service (DoS) durch senden eines speziell erstellten TCP Pakets über das öffentliche Internet (z.B. CVE 2004 0230) auslöst.

- Adjacent (A)

  Eine Schwachstelle die mit benachbartem Netzwerkzugang ausnutzbar ist meint, dass die angreifbare Komponente an den Network Stack gebunden ist, der Angriff aber auf dasselbe geteilte physikalische (z.B. Bluetooth, IEEE 802.11) oder logische (z.B. Lokales IP Subnet) Netzwerk limitiert ist und nicht über OSI Layer 3 hinweg (z.B. Router) durchgeführt werden kann. Ein Beispiel für einen Adjacent Angriff wäre eine ARP (IPv4) oder neighbor discovery (IPv6) Flooding, die zu einem denial of service auf dem lokalen LAN Segment führt. Siehe auch CVE 2013 6014.

- Local (L)

  Eine Schwachstelle die durch lokalen Zugang ausnutzbar ist bedeutet, dass die angreifbare Komponente nicht an den Network Stack gebunden ist, und der Angreifer durch read/write/execute Fähigkeiten agiert. In manchen Fällen muss der Angreifer auch lokal eingeloggt sein um die Schwachstelle auszunutzen. Andernfalls ist die Schwachstelle womöglich auf Interaktion mit dem User angewiesen um eine bösartige Datei auszuführen.

- Physical (P)

  Bei einer Schwachstelle die durch physikalischen Zugang ausnutzbar ist, muss der Angreifer die angreifbare Komponente physikalisch manipulieren. Physikalische Interaktion kann kurz (z.B. evil maid attack) oder dauerhaft sein. Ein Beispiel einer solchen Attacke ist eine Kaltstartattacke die einem Angreifer Zugang zu Schlüsseln der Festplattenverschlüsselung gewährt, nachdem er physikalischen Zugang zum System hat. Ein weiteres Beispiel sind periphere Angriffe wie z.B. Firewire/USB Direct Memory Access Angriffe.

Attack Complexity (AC)

Diese Metrik beschreibt die Bedingungen die existieren müssen um die Schwachstelle auszunutzen, auf die der Angreifer aber keinen Einfluss hat. Wie unten beschrieben, können solche Umstände mehr Informationen über das Ziel, das Vorhandensein von bestimmten Systemkonfigurationen oder Berechnungsfehler benötigen. Wichtiger Weise schließt die Bewertung dieser Metrik zur Ausnutzung der Schwachstelle jeglichen Bedarf an Interaktionen des Users aus. Der Wert dieser Metrik ist höher je einfacher der Angriff ist. Die Liste von möglichen Werten ist in den folgenden Punkten aufgeführt.

- Low (L)

  Spezielle Bedingungen oder Voraussetzungen für den Zugang existieren nicht. Ein Angreifer kann einen reproduzierbaren Erfolg gegen die angreifbare Komponente erwarten.

- High (H)

  Eine erfolgreiche Attacke hängt von den Bedingungen über die Kontrolle des Angreifers hinaus ab. So kann ein erfolgreicher Angriff nicht beliebig gelingen, sondern erfordert Aufwand des Angreifers bei der Vorbereitung oder muss erfolgen noch bevor ein erfolgreicher Angriff gegen die angreifbare Komponente erwartet wird.

Privileges Required (PR)

Diese Metrik beschreibt eine Ebene der Privilegien die ein Angreifer haben muss, bevor er erfolgreich eine Schwachstelle ausnutzen kann. Der Wert dieser Metrik ist am höchsten wenn keine Privilegien benötigt werden. Die Liste von möglichen Werten ist in den folgenden Punkten aufgeführt.

- None (N)

  Der Angreifer ist vor dem Angriff unautorisiert und braucht daher keinen Zugang zu Einstellungen oder Dateien um seinen Angriff durchzuführen.

- Low (L)

  Der Angreifer ist mit Privilegien ausgestattet die ihm grundlegende Nutzerrechte verleihen und normalerweise nur Einstellungen und Dateien betreffen die dem Nutzer gehören. Alternativ richtet ein Angreifer mit niedrigen Privilegien nur Schaden an nicht sensiblen Ressourcen an.

- High (H)

  Der Angreifer ist mit Privilegien ausgestattet die ihm signifikante (z.B. administrative) Kontrolle über die angreifbare Komponente geben und Einstellungen oder Dateien in der gesamten Komponente betreffen.

User Interaction (UI)

Diese Metrik beschreibt die Bedingungen für den Benutzer und nicht die des Angreifers, die nötig sind um erfolgreich eine Komponente angreifen zu können. Sie legt fest ob eine Schwachstelle allein durch den Angreifer oder nur mit Hilfe eines anderen Benutzers (oder eines vom Benutzer initierten Prozesses) ausnutzbar ist. Der Wert dieser Metrik ist am größten wenn keine Interaktion eines anderen Benutzers nötig ist. Die Liste von möglichen Werten ist in der untenstehenden Tabelle aufgeführt.

- None (N)

  Die Schwachstelle kann ausgenutzt werden ohne notwendige Interaktion mit einem anderen Benutzer.

- Required (R)

  Die erfolgreiche Ausnutzung der Schwachstelle benötigt Aktionen eines Benutzers. Ein Beispiel wäre ein Exploit das nur während einer Installation einer Applikation als Systemadministrator erfolgreich wäre.

5.2.1.2 Scope

Eine wichtige Eigenschaft die von CVSS v3.0 erfasst wird ist die Fähigkeit einer Schwachstelle in einer Softwarekomponente Ressourcen zu beeinflussen die über ihre Privilegien und Mittel hinausgehen.

Formal bezieht sich Scope auf die Sammlung von Privilegien, die von einer Gruppe definiert werden (z.B. einer Applikation, einem Betriebssystem oder einer Sandbox Umgebung), denen Zugang zu einer Ressource (z.B. Dateien, CPU, Speicher, etc.) gewährt wird. In manchen Fällen wird basierend auf vordefinierten Regeln oder Standards die Autorisierung einfach oder nur ungenügend kontrolliert. Zum Beispiel im Fall von Netzwerkverkehr der zu einem Netzwerk Switch gesendet wird. Der Switch akzeptiert den Netzwerkverkehr der an einem Ports ankommt und ist die Autorität die den Verkehrsfluss zu anderen Ports kontrolliert.

Wenn es durch eine Schwachstelle einer Softwarekomponente möglich ist, Ressourcen zu beeinflussen die von einer anderen Komponente verwaltet werden, hat ein Scope „change" stattgefunden.

Man kann sich einen Scope „change" wie das Ausbrechen aus einer Sandbox vorstellen. Ein Beispiel wäre eine Schwachstelle in einer Virtuellen Maschine, die es dem Angreifer ermöglicht Dateien auf dem Host OS zu löschen. In diesem Beispiel gibt es zwei separate Privilegien Gruppen. Eine Gruppe die Privilegien für die Virtuelle Maschine mit Benutzern durchsetzt und eine andere die Privilegien für das Hostsystem, in dem die Virtuelle Maschine läuft, definiert und durchsetzt.

Ein Scope „change" würde nicht auftreten, wenn z.B. eine Schwachstelle in Microsoft Word es einem Angreifer erlauben würde alle Systemdateien des Host OS zu kompromittieren, da dieselbe Autorität (Host OS) die Privilegien der Word-Instanz des Benutzers als auch den Zugriff auf Systemdateien durchsetzt.

Der Base Score ist am größten wenn ein Scope „change" stattgefunden hat. Die Liste von möglichen Werten ist in den folgenden Punkten aufgeführt.

- Unchanged (U)

  Eine ausgenutzte Schwachstelle kann nur Ressourcen beeinflussen, die von derselben Autorität verwaltet werden. Dabei sind die angreifbare Komponente und die beeinflusste Komponente dieselbe.

- Changed (C)

  Eine ausgenutzte Schwachstelle kann Ressourcen beeinflussen, die über die vorgesehenen Privilegien hinausgehen. In diesem Fall sind die angreifbare Komponente und die beeinflusste Komponente verschieden.

5.2.1.3 Impact Metrics

Die Impact Metrik bezieht sich auf die Eigenschaften der betroffenen Komponente. Egal ob eine erfolgreich ausgenutzte Schwachstelle, eine oder mehrere Komponenten betrifft, die Impact Metrik wird der Komponente entsprechend gewertet die den schwersten Schaden erlitten hat und die am direktesten mit dem Angriff in Verbindung gebracht werden kann. Deswegen sollten die Auswirkungen die ein Angreifer verursachen kann auf ein angemessenes Ergebnis beschränkt werden.

Wenn kein Scope „change" aufgetreten ist, sollte die Impact Metrik die Auswirkungen auf confidentiality, integrity und availability(CIA) der gefährdeten Komponente reflektieren. Wenn aber ein Scope „change" aufgetreten ist, sollte die Impact Metrik die Auswirkungen auf CIA der gefährdeten Komponente oder der betroffenen Komponente reflektieren, abhängig davon welche Komponente den größten Schaden erlitten hat.

Confidentiality Impact (C)

Diese Metrik misst die Auswirkungen auf die Vertraulichkeit der Informationen die von einer Softwarekomponente verwaltet werden. Vertraulichkeit bezieht sich auf den eingeschränkten Zugang zu Informationen und für ausschließlich autorisierte Benutzer, sowie die Unterbindung des Zugangs für unautorisierte Benutzer. Der Wert dieser Metrik erhöht sich mit steigendem Verlust an Vertraulichkeit bei der betroffenen Komponente. Die Liste von möglichen Werten ist in den folgenden Punkten aufgeführt.

- High (H)

  Es besteht ein totaler Verlust der Vertraulichkeit der dazu führt, dass alle Ressourcen in der betroffenen Komponente dem Angreifer offenbart werden. Es ist auch möglich, dass nur der Zugang zu einigen Informationen durch den Angriff erfolgt ist, diese jedoch kritisch sind und dadurch direkte und ernste Auswirkungen haben können. Dies wäre der Fall, wenn ein Angreifer beispielsweise das Passwort des Administrators oder private Verschlüsselungsschlüssel eines Web-Servers erbeuten würde.

- Low (L)

  Es besteht leichter Verlust der Vertraulichkeit. Der Angreifer erhält Zugang zu einigen Informationen, hat jedoch keinen Einfluss darauf welche oder wie viele Informationen er erbeuten kann oder die Anzahl bzw. die Art des Verlusts ist beschränkt. Die Offenlegung der Informationen durch den Angreifer bedeutet keinen ernsten und direkten Verlust an Vertraulichkeit für die betroffene Komponente.

- None (N)

  Es besteht kein Verlust der Vertraulichkeit bei der betroffenen Komponente.

Integrity Impact (I)

Diese Metrik misst die Auswirkungen auf die Integrität einer erfolgreich ausgenutzten Schwachstelle. Integrität bezieht sich auf die Vertrauenswürdigkeit und Nachvollziehbarkeit von Informationen. Der Wert dieser Metrik erhöht sich mit den Konsequenzen für die betroffene Komponente. Die Liste von möglichen Werten ist in den folgenden Punkten aufgeführt.

- High (H)

  Es besteht ein totaler Verlust der Integrität oder des Schutzes. Dies wäre der Fall, wenn der Angreifer alle Dateien die von der betroffenen Komponente geschützt werden verändern kann. Alternativ wäre dies der Fall wenn nur manche Dateien verändert werden können, dies aber zu direkten und ernsten Konsequenzen für die betroffene Komponente führen würden.

- Low (L)

  Veränderung von Daten ist möglich aber der Angreifer hat keine Kontrolle über die Konsequenzen einer Veränderung oder die Anzahl der Veränderungen ist beschränkt. Die Datenveränderung hat keinen direkten und ernsten Einfluss auf die betroffene Komponente.

- None (N)

  Es besteht kein Verlust der Integrität bei der betroffenen Komponente.

Availability Impact (A)

Diese Metrik misst den Einfluss auf die Verfügbarkeit der betroffenen Komponente, die aus einer erfolgreichen Ausnutzung einer Schwachstelle resultiert. Während die Confidentiality und Integrity Impact Metriken den Verlust von Vertraulichkeit und Integrität von Daten (z.B. Informationen, Dateien) die von der betroffenen Komponente benutzt werden betreffen, bezieht sich diese Metrik auf den Verlust der Verfügbarkeit der betroffenen Komponente selbst, wie etwa ein Netzwerkservice (z.B. Web, Datenbank, Email). Da Availability die Verfügbarkeit von Informationen und Ressourcen meint, schränken Angriffe die Bandbreite, Rechenleistung oder Speicherplatz verbrauchen die Verfügbarkeit der betroffenen Komponente ein. Der Wert dieser Metrik erhöht sich mit den Konsequenzen für die betroffene Komponente. Die Liste von möglichen Werten ist in den folgenden Punkten aufgeführt.

- High (H)

  Es besteht kompletter Verlust der Verfügbarkeit der dazu führt, dass es dem Angreifer möglich ist den Zugang zu Ressourcen der betroffenen Komponente komplett zu sperren. Dieser Zustand ist entweder während des Angriffes anhaltend oder dauerhaft auch nach dem Angriff vorhanden. Alternativ kann der Angreifer auch nur kurzfristig die Fähigkeit haben die Verfügbarkeit zu blockieren, was aber direkte und ernste Konsequenzen für die beeinflusste Komponente hat (z.B. kann der Angreifer keine bestehenden Verbindungen trennen aber neue Verbindungen verhindern oder er kann eine Schwachstelle die bei jedem erfolgreichen Angriff nur etwas Speicher reserviert wiederholt ausnutzen. Nach wiederholter Ausnutzung führt dies jedoch zum Ausfall des Service.).

- Low (L)

  Es besteht reduzierte Performance oder Unterbrechungen in der Verfügbarkeit von Ressourcen. Selbst wenn eine wiederholte Ausnutzung der Schwachstelle möglich ist, hat der Angreifer nicht die Möglichkeit einen Service für legitime Benutzer vollständig zu blockieren. Die Ressourcen der betroffenen Komponente sind entweder ständig, teilweise oder sporadisch vollständig verfügbar. Es bestehen jedoch keine direkten und ernsten Konsequenzen für die betroffene Komponente.

- None (N)

  Es besteht kein Einfluss auf die Verfügbarkeit der betroffenen Komponente.

5.2.2 Base Vectors

Der v3.0 Vektor String beginnt mit dem Label „CVSS" und einer numerischen Repräsentation der aktuellen Version „3.0". Es folgen Informationen über die Metriken in Form einer Reihe von abgekürzten Metrik Namen, einem „:" und dem entsprechenden Wert der Metrik in abgekürzter Form. Die Kurzformen wurden oben in dieser Spezifikation definiert (in Klammern nach jedem Metrik Namen und Metrik Wert). Die Metriken werden durch einen Schrägstrich „/" voneinander getrennt. In der folgenden Tabelle sind die abgekürzten Metrik Namen und die möglichen Werte zusammengefasst.

Metrik Name	**Mögliche Werte**	**Metrik Name**	**Möglich Werte**
Attack Vector, AV	[N,A,L,P]	Scope, S	[U,C]
Attack Complexity, AC	[L,H]	Confidentiality, C	[H,L,N]
Privileges Required, PR	[N,L,H]	Integrity, I	[H,L,N]
User Interaction, UI	[N,R]	Availability, A	[H,L,N]

Beispiel: Eine Schwachstelle mit folgenden Base Metrik Werten: „Attack Vector: Network, Attack Complexity: Low, Privileges Required: High, User Interaction: None, Scope: Unchanged, Confidentiality: Low, Integrity: Low, Availability. None" ohne spezifizierte Temporal oder Environmental Metriken würde folgenden Vektor produzieren:

CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:L/A:N