Report 101500651
Externe Sicherheitsüberprüfung beA Webanwendung

Atos IT Solutions and Services GmbH

Durchgeführt von

Unternehmensberatung GmbH

Version:	**1.0**
Autor:	**I. Lorch**
Verantwortlich:	**I. Lorch**
Datum:	**18.12.2015**
Vertraulichkeitsstufe:	**Streng vertraulich**

Inhaltsverzeichnis

1 Management Summary

Im folgenden Kapitel finden Sie die Ergebnisse des Audits, sowie die von SEC Consult empfohlenen Maßnahmen.

1.1 Ergebnisse des Audits

Bei der externen Sicherheitsüberprüfung für das Unternehmen Atos IT Solutions and Services GmbH nach dem Blackbox Ansatz untersuchte SEC Consult die beA Webanwendung sowie dessen Client-Security Komponente und das zugehörige Backend (für die vollständige Auflistung der überprüften Systeme siehe Kap. 2.3).

Bei diesem konkreten Projekt wurde ein Timebox-Ansatz zur Ermittlung des Aufwands herangezogen. Dies bedeutet, dass SEC Consult nur die innerhalb der angegebenen Zeit gefundenen Schwachstellen dokumentieren kann. Alle Angriffe wurden aus der Sicht eines Außenstehenden durchgeführt und fanden mit eingeschränktem Wissen (zur Verfügung gestellte Dokumentationen siehe Kapitel 2.3) über interne Strukturen statt

Ziel dieser Sicherheitsüberprüfung war es, verschiedene Arten von Schwachstellen, sowie übliche Konfigurationsfehler in der getesteten Software zu identifizieren. Im Verlauf des Audits wurde getestet, ob die Anwendungen in der Lage sind, möglichen Angriffen zu widerstehen. Kapitel 1.2 zeigt dabei die empfohlenen Maßnahmen basierend auf den Ergebnissen des Audits auf.

Die Testergebnisse zeigen, dass die beA Webanwendung sowie die Client-Security Komponente und das zugehörige Backend ein hohes Sicherheitsniveau aufweisen. Eine während der Sicherheitsüberprüfung identifizierte Schwachstelle wurde bereits während des Testzeitraums in kürzester Zeit behoben. Kleine Schwachstellen verbleiben u. a. noch im Bereich der Preisgabe von potentiell sensiblen Versionsinformationen der eingesetzten Software der Systeme.

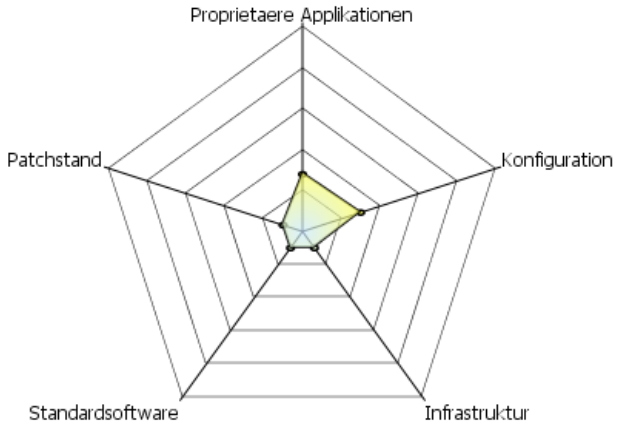
1.1.1 Impact / Worst Case Szenarien

SEC Consult konnte während des Audits einzelne, für Webanwendungen typische Schwachstellen von geringem Risiko identifizieren. Die meisten Schwachstellen ergeben sich dabei aus Konfigurationsmängeln und einer unzureichenden Validierung von Benutzereingaben. Unter Ausnutzung dieser Schwachstellen kann ein Angreifer aus dem Internet:

- Aufgrund von unzureichender Eingabevalidierung eingeschränkt JavaScript Code im Browser anderer Benutzer der Webanwendung ausführen.
- Informationen über die Architektur sowie Versionsnummern der installierten Software sammeln um diese dann ggf. für weiterführende Angriffe zu benutzen.

1.1.2 Technische Risikobewertung

Anhand der Risikoeinschätzung von SEC Consult in fünf Dimensionen ergibt sich folgendes Risikoprofil:



Legende: Bewertung der einzelnen Dimensionen nach Schulnotensystem (1: geringes Risiko; 5: hohes Risiko)

- **Proprietäre Applikationen:** Hier ist ein leichter Ausschlag zu verzeichnen. Teile der gefundenen Schwachstellen befinden sich in der proprietären Webanwendung.
- **Patchstand:** In den Tests konnten keine Schwachstellen gefunden und ausgenutzt werden, die auf nicht eingespielte Sicherheitspatches zurückzuführen sind.
- **Standardsoftware:** In der eingesetzten Standardsoftware wurden keine sicherheitskritischen Fehler gefunden.
- **Konfiguration:** Die Konfiguration der Komponenten ist in den meisten Fällen einwandfrei. Leichte Konfigurationsmängel im verwendeten Applikationsserver wurden festgestellt.
- **Infrastruktur:** Eine Überprüfung der dem System zugrunde liegenden Infrastruktur war nicht Bestandteil des Audits.

Es ergibt sich ein technisches Gesamtrisiko von **7,54 (Gering)**. Dies zeigt, dass die beA Webanwendung effektive Schutzmaßnahmen implementiert hat.

Disclaimer

Dieser Bericht ist streng vertraulich und nur für die interne, vertrauliche Verwendung beim Auftraggeber bestimmt. Der Empfänger verpflichtet sich, für die Geheimhaltung der streng vertraulichen Inhalte im Sinne der Organisation Sorge zu tragen. Der Empfänger übernimmt die Verantwortung für die weitere Verteilung des Dokuments.

Bei diesem konkreten Projekt wurde ein Timebox-Ansatz zur Ermittlung des Aufwands herangezogen. Dies bedeutet, dass SEC Consult lediglich innerhalb des vereinbarten Zeitfensters Schwachstellen identifizieren und dokumentieren kann. Aus diesem Grund kann aus der Überprüfung in diesem Projekt keinerlei Anspruch auf Vollständigkeit der in diesem Bericht dokumentierten Sicherheitslücken abgeleitet werden.

Des Weiteren stellt die Sicherheitsüberprüfung eine Augenblicksbetrachtung zum Zeitpunkt der Überprüfung dar. Eine Bewertung des zukünftigen Sicherheitsniveaus oder möglicher zukünftigen Risikoschwachstellen kann davon nicht abgeleitet werden.

Im Zuge des Audits wurden auf Systemen des Auftraggebers im Scope, falls erforderlich, lokale Dateien erstellt (z.B. temporäre Dateien, Log-Dateien, oder vom Auftragnehmer hochgeladene Programme zur Ausnutzung von etwaigen Schwachstellen). Dies geschieht, falls erforderlich, entweder manuell oder automatisiert durch Schwachstellenscanner. Diese Dateien wurden nach dem Audit soweit dem Auftragnehmer möglich entfernt. Eine vollständige Entfernung ist jedoch bedingt durch das Vorgehen in einem Sicherheitsaudit (z.B. fehlender Systemzugriff oder keine ausreichenden Rechte) nicht immer möglich. Es können daher ausgewählte dieser lokalen Dateien auch nach Beendigung des Auftrags vorhanden sein, die bei Bedarf vom Auftraggeber selbst zu entfernen sind.

1.2 Empfohlene Maßnahmen

Aufgrund der Ergebnisse der Sicherheitsüberprüfung hat SEC Consult folgende Maßnahmen als sinnvoll eingestuft.

1.2.1 Maßnahmen mit unmittelbarem Handlungsbedarf

1. **Behebung der in diesem Report aufgezeigten Sicherheitslücken.** Im Rahmen der Sicherheitsüberprüfung wurden mehrere Schwachstellen gefunden. Diese Schwachstellen sollten ehestmöglich behoben werden. Lösungsansätze für die einzelnen Lücken finden sich am Ende der jeweiligen Detailbeschreibung.

   Die Spezialisten von SEC Consult stehen Ihnen diesbezüglich gerne telefonisch oder vor Ort und „Hands on" im Zuge unserer „Care and Repair" – Services zur Verfügung.

1.2.2 Weiterführende Maßnahmen

1. **Abnahme Tests bei Eigenentwicklungen.** Jedes System sollte vor Produktivsetzung einer Sicherheitsüberprüfung unterzogen werden, die vom Umfang so gewählt werden sollte, dass kritische Applikationen gründlicher überprüft werden als weniger kritische. Da bei selbstentwickelten Applikationen der Source Code zur Verfügung steht, empfehlen sich für kritische Applikationen Glassbox Test bzw. Source Code Reviews. Durch die Durchführung vor Produktivsetzung wird das Risiko drastisch reduziert und eventuelle Down-Times vermieden.
2. **Zyklische externe Sicherheitsüberprüfungen.** Im Laufe der Zeit werden immer neue Schwachstellen und Angriffsvektoren bekannt auf die proprietäre Software nicht überprüft wurde. Eine zyklische Durchführung von externen Sicherheitsüberprüfungen kann auf mittlere Sicht das hohe Sicherheitsniveau einer Anwendung halten.
3. **Überprüfung der in diesem Test nicht berücksichtigten Systeme.** Die Sicherheit des Gesamtsystems ist in vielen Fällen vom schwächsten Glied abhängig. Selbst ein Server, der als nicht kritisch eingestuft wird, kann es einem Angreifer beispielsweise ermöglichen, über die Firewall hinwegzukommen und andere Systeme von intern zu attackieren. Daher ist es sinnvoll, in regelmäßigen Abständen den gesamten IP Range des Unternehmens zu testen.
4. **Source Code Audit von kritischen Applikationen.** Besonders kritische Applikationen, die beispielsweise wichtige Informationen verwalten, können mittels eines Source Code Audits besonders detailliert überprüft werden. So können auch Schwachstellen gefunden werden, die in einem Blackbox-Audit nur schwer zu finden sind.
5. **Stresstest der IT Infrastruktur und Applikationen zur Vorbeugung gegen (D)DoS Attacken.** Denial of Service bzw. Distributed Denial of Service Attacken können massive Schäden für Unternehmen verursachen. Mit Hilfe von Stresstests kann festgestellt werden, ob die Infrastruktur bzw. die Anwendungen solchen Angriffen standhalten können und ob die etablierten Prozesse für DoS Attacken funktionieren.

1.2.3 Notwendige Handlungen im Nachgang der Überprüfung

1. **Entfernen von Testkonten, welche für die Auditoren erstellt wurden**. Um den sicheren Betrieb der Systeme sicherzustellen, empfiehlt SEC Consult nicht mehr benötigte Accounts zu entfernen oder zu deaktivieren. Während der Tests wurden den Auditoren folgende Token zur Verfügung gestellt:

Typ	System	Kommentar
Chipkarte	test.bea-brak.de	beAcard188
Chipkarte	test.bea-brak.de	beA64card
Software-Token	test.bea-brak.de	Anna Mitarbeiter
Software-Token	test.bea-brak.de	Initial SystemverwalterStaging

2. **Entfernen von auditspezifischen Konfigurationsänderungen.** Nach Beendigung des Audits empfiehlt SEC Consult alle Test-spezifischen Änderungen wieder rückgängig zu machen. Folgende spezifische Änderungen wurden vorgenommen:
    - Freischaltung der SEC Consult IP-Range (92.60.14.128/26)

2 Vorgehensweise

Im folgenden Kapitel wird die Vorgehensweise, die SEC Consult bei der Sicherheitsüberprüfung anwendet, erläutert.

2.1 Testmethode

SEC Consult führt Penetrationstests durch, um die Sicherheit eines Gesamtsystems oder einzelner Systemkomponenten zu überprüfen. Die Tools, Methoden und Techniken, die von SEC Consult eingesetzt werden, fallen in die folgenden drei Kategorien:

1 Allgemein bekannt in Computer-Security und Hacker Communities[1]
2 Von SEC Consult entwickelt, um weitergehende Sicherheit erreichen zu können
3 Einsatz von Spezialisten-Wissen. Automatische Werkzeuge können manche Sicherheitslücken nicht entdecken, sehr wohl aber das geschulte Auge von Spezialisten.

Für das Unternehmen Atos IT Solutions and Services GmbH wurde ein Penetrationstest an der externen Webanwendung beA durchgeführt. Dabei überprüfte SEC Consult von außerhalb – also ausschließlich aus dem Internet – ob das System ausreichend geschützt sind, um eventuellen Angriffen widerstehen zu können. Alle Angriffe wurden aus der Sicht eines Außenstehenden durchgeführt und fanden mit eingeschränktem Wissen über interne Strukturen statt (siehe Kapitel 2.3).

2.2 Durchgeführte Testklassen

Details zur Risikobewertung können in Kapitel 2.5 nachgelesen werden. Die beA Webanwendung sowie das zugehörige Backend und die Client-Security Anwendung wurden, sofern die Services es erlaubten, auf folgende Fehlerklassen getestet:

2.2.1 Server-Konfiguration

	Konfigurationsfehler	
	Ausnutzbare Konfigurationsfehler für verschiedene Arten von Server-Software	
Angriffsvektor	Getestet[1]	Ausnutzbar[2]
Enumeration von Server-Inhalten	JA	NEIN
Ausnutzung von Default-Accounts	JA	NEIN
Enumeration von Benutzeraccounts	JA	NEIN
Ausnutzung gefährlicher Protokollfeatures	JA	NEIN
Ausnutzung nicht ausreichend gesetzter Berechtigungen	JA	NEIN
Ausnutzung von ungeschützter Funktionalität	JA	NEIN
Enumeration von Server-internen Informationen	JA	NEIN
Sammeln von Informationen über System- oder Fehlermeldungen	JA	JA
Erraten von Passwörtern	JA	NEIN

[1] Der Ergebnisbericht enthält gegebenenfalls auch Quellcodeauszüge frei erhältlicher Tools und Exploits von Drittherstellern.

	Konfigurationsfehler		
	Ausnutzbare Konfigurationsfehler für verschiedene Arten von Server-Software		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Mitlesen unverschlüsselter, sensitiver Daten		JA	NEIN
Weitere Angriffsvektoren			

2.2.2 Patch Level

	Server Patch-Level		
	Diese Schwachstellenklasse bezieht sich auf bekannte Lücken, die mit automatischen Tools identifiziert werden können.		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Ausnutzung öffentlich bekannter Sicherheitslücken		JA	NEIN
Weitere Angriffsvektoren			

2.2.3 Standard-Software und proprietäre Applikationen

	Authentisierungsfehler		
	Fehler in der Authentisierung von Benutzern der Anwendung		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Umgehen der Authentisierung		JA	NEIN
Weitere Angriffsvektoren			

	Autorisierungsfehler		
	Ein unautorisierter oder unberechtigter Benutzer kann auf geschützte Objekte zugreifen		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Zugriff auf geschützte Funktionalität		JA	NEIN
Zugriff auf geschützte Ressourcen		JA	NEIN
Weitere Angriffsvektoren			

	Ausgabe von Informationen		
	Der Angreifer kann interne Informationen über die Anwendung oder die Serverumgebung sammeln		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Ausnutzen von Dateiendungsverarbeitungen		JA	NEIN
Sammeln von Informationen über Entwickler-Kommentare		JA	NEIN
Sammeln von Informationen über System- oder Fehlermeldungen		JA	NEIN
Lesen von Sampledateien oder alten, unreferenzierten Files		JA	NEIN
Weitere Angriffsvektoren			

	Interpreter Injection / Eingabe-Validierung		
	Die Anwendung übergibt nicht validierte Parameter an einen Interpreter, gefährliche Library-Funktionen oder OS-APIs		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Zugriff auf das Dateisystem		JA	NEIN
Code Injection		JA	NEIN
Command Injection		JA	NEIN
Format String Injection		JA	NEIN
IMAP/SMTP Injection		JA	NEIN
LDAP Injection		JA	NEIN
ORM Injection		JA	NEIN
Overflowing Character Buffers		JA	NEIN
Path Traversal		JA	NEIN
SQL Injection		JA	NEIN
SSI Injection		JA	NEIN
XML Injection		JA	NEIN
XPath Injection		JA	NEIN
Weitere Angriffsvektoren			

	State -/ Session-Management-Fehler		
	State- oder Session Status wird von der Anwendung nicht richtig gehandhabt.		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Ermittlung von Session-Identifikatoren		JA	NEIN
Ausnutzung von Problemen im State-Management		JA	NEIN
Weitere Angriffsvektoren			

	Unsichere Funktionalität (Minimalprinzip)		
	Die Applikation bietet Funktionalität, die zu Sicherheitsproblemen führt.		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Ausnutzung von Sample-Anwendungen		JA	NEIN
Upload beliebiger Files		JA	NEIN
Weitere Angriffsvektoren			

	Unsicheres Management vertrauenswürdiger Daten		
	Vertrauenswürdige oder interne Daten können vom Angreifer verändert werden.		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Manipulation applikationsinterner Daten am Client		JA	NEIN

Lesen applikationsinternen oder vertraulicher Daten am Client		JA	NEIN
Weitere Angriffsvektoren			

	Unsichere Algorithmen		
	Der Einsatz unsicherer Algorithmen erlaubt die Kompromittierung sensitiver Informationen		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Ausnutzung schwacher Verschlüsselungsalgorithmen		JA	NEIN
Ausnutzung schwacher Zufallszahlengeneratoren		JA	NEIN
Weitere Angriffsvektoren			

	Verwundbarkeit durch Denial-of-Service		
	Das Service kann durch den Angreifer unbenutzbar gemacht werden.		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Aufbrauchen limitiert verfügbarer Ressourcen		NEIN	NEIN
Aussperren von Benutzeraccounts		JA	NEIN
Weitere Angriffsvektoren			

	Verwundbarkeit gegenüber client-seitigen Attacken (Web Browser)		
	Diese Schwachstellenklasse bezieht sich auf Webapplikationen. Benutzer der Anwendung werden zum Ziel dieser Angriffe.		
Angriffsvektor		Getestet[1]	Ausnutzbar[2]
Cross-Site Request Forgery (XSRF)		JA	NEIN
HTML Injection / Cross-Site Scripting (XSS)		JA	NEIN[3]
HTTP Response Splitting / header injection		JA	NEIN
Frame Spoofing		JA	NEIN
Session Fixation		JA	NEIN
Weitere Angriffsvektoren			

[1]Getestet: Der Angriffsvektor wurde in dieser Überprüfung von SEC Consult getestet.

[2]Ausnutzbar: Der Angriffsvektor wurde in dieser Überprüfung als ausnutzbare Schwachstelle identifiziert.

[3]Hier existiert ein theoretischer Angriffsvektor, welcher jedoch aufgrund von Beschränkungen aktuell nicht ausnutzbar ist (siehe Kapitel 5.1.5).

2.3 Umfang und Zeitplan

Von 07.12.2015 bis 18.12.2015 fand eine externe Sicherheitsüberprüfung statt, bei der es galt Sicherheitslücken in einer Teststellung der beA Webanwendung des Unternehmens Atos IT Solutions and Services GmbH zu finden. SEC Consult testete dabei das System mit der folgenden IP Adresse:

- 185.62.147.189 (test.bea-brak.de)

Ebenfalls Teil der Sicherheitsüberprüfung waren das Backend der beA Webanwendung sowie die zugehörige Client-Security Anwendung.

Für die Sicherheitsüberprüfung wurden folgende Dokumentationen und Hilfsmittel zur Verfügung gestellt:

- Umsetzungsfeinkonzept beA-System (Version 3.1, 14.01.2015)
- beA Client-Security – Kryptographische Funktionen für elektronische Nachrichten
- Zwei Chipkarten (beAcard188, beA64card) + kompatibles Chipkartenlesegerät
- Zwei Software-Token (Anna Mitarbeiter, Initial SystemverwalterStaging)

Der Registrationsprozess eines neuen Benutzers konnte im Laufe der Sicherheitsüberprüfung nicht untersucht werden, da die zur Verfügung gestellten Chipkarten bereits registriert waren.

Es wurden keine Tests durchgeführt, bei denen die Verfügbarkeit der Services willentlich gefährdet wurde.

2.4 Disclaimer

Dieser Bericht ist streng vertraulich und nur für die interne, vertrauliche Verwendung beim Auftraggeber bestimmt. Der Empfänger verpflichtet sich, für die Geheimhaltung der streng vertraulichen Inhalte im Sinne der Organisation Sorge zu tragen. Der Empfänger übernimmt die Verantwortung für die weitere Verteilung des Dokuments.

Bei diesem konkreten Projekt wurde ein Timebox-Ansatz zur Ermittlung des Aufwands herangezogen. Dies bedeutet, dass SEC Consult lediglich innerhalb des vereinbarten Zeitfensters Schwachstellen identifizieren und dokumentieren kann. Aus diesem Grund kann aus der Überprüfung in diesem Projekt keinerlei Anspruch auf Vollständigkeit der in diesem Bericht dokumentierten Sicherheitslücken abgeleitet werden.

Des Weiteren stellt die Sicherheitsüberprüfung eine Augenblicksbetrachtung zum Zeitpunkt der Überprüfung dar. Eine Bewertung des zukünftigen Sicherheitsniveaus oder möglicher zukünftigen Risikoschwachstellen kann davon nicht abgeleitet werden.

Im Zuge des Audits wurden auf Systemen des Auftraggebers im Scope, falls erforderlich, lokale Dateien erstellt (z.B. temporäre Dateien, Log-Dateien, oder vom Auftragnehmer hochgeladene Programme zur Ausnutzung von etwaigen Schwachstellen). Dies geschieht, falls erforderlich, entweder manuell oder automatisiert durch Schwachstellenscanner. Diese Dateien wurden nach dem Audit soweit dem Auftragnehmer möglich entfernt. Eine vollständige Entfernung ist jedoch bedingt durch das Vorgehen in einem Sicherheitsaudit (z.B. fehlender Systemzugriff oder keine ausreichenden Rechte) nicht immer möglich. Es können daher ausgewählte dieser lokalen Dateien auch nach Beendigung des Auftrags vorhanden sein, die bei Bedarf vom Auftraggeber selbst zu entfernen sind.

2.5 Einzelrisikobewertung

Alle gefundenen Sicherheitsrisiken wurden mit einem Risk-Score bewertet. Dieser Risk-Score wird in einer Risiko-Matrix ermittelt, die sich aus Wahrscheinlichkeit und Schwere zusammensetzt. Die Wahrscheinlichkeit bezeichnet dabei die Wahrscheinlichkeit mit der ein Angreifer den Fehler findet und ausnutzen kann. Die Schwere bezieht sich auf den Schweregrad der Lücke und ihre Auswirkungen. Da der Schweregrad das Risiko wesentlich stärker beeinflusst als die Wahrscheinlichkeit, fließt er quadriert in die Gleichung ein.

Durch die Multiplikation von Wahrscheinlichkeit und Quadrat der Schwere ergibt sich der Risk-Score, der eine sehr differenzierte Einschätzung des Risikos, das durch eine Sicherheitslücke entsteht, erlaubt.

Schwere / Wahrscheinlichkeit	1	4	9	16	25
1	1	4	9	16	25
2	2	8	18	32	50
3	3	12	27	48	75
4	4	16	36	64	100
5	5	20	45	80	125

Um eine einfache textuelle Beschreibung des Risikos zu ermöglichen, wurden die Risk-Scores in vier Kategorien unterteilt:

Risk-Score	Bewertung
1 – 10	gering
11 – 24	mittel
25 – 60	groß
61 – 125	kritisch

2.6 Gesamtrisikobewertung

Um ein Gesamtrisiko für ein System, ein Netzwerk oder ein ganzes Unternehmens angeben zu können, müssen die Einzelrisiken aufsummiert werden. Eine einfache Addition ist jedoch nicht möglich, da dies nicht dem wirklichen Verhalten einzelner Schwachstellen zueinander entspricht. Zwei Schwachstellen, von denen das gleiche Risiko ausgeht, bedeuten gemeinsam nicht das doppelte Risiko.

Daher wird zum Summieren der Einzelrisiken die energetische Summenformel angewandt:

$$10\lg(10^{R_1/10}+10^{R_2/10}+\ldots+10^{R_n/10}) = R_{gesamt}$$

R ... Einzelrisiko
R_{gesamt} ... Gesamtrisiko

3 Schwachstellenübersicht

Aus dem Securityaudit ergibt sich für das Unternehmen Atos IT Solutions and Services GmbH folgende Auflistung an Schwachstellenklassen:

Risikobewertung	Anzahl der Schwachstellenklassen
Gering	2
Mittel	0
Groß	0
Kritisch	0
Gesamt	**2**

3.1 Gesamtrisiko pro System

Die folgende Tabelle enthält eine Risikowertung für jedes Einzelsystem, auf dem Sicherheitsrisiken identifiziert wurden.

System	Einsatzbereich	Risiko
test.bea-brak.de (185.62.147.189)	Web	Gering (7,54)
Gesamt	**-**	**Gering (7,54)**

3.2 Einzelrisiken

Die folgende Liste enthält eine Aufzählung aller gefundenen Sicherheitslücken.

Sicherheitslücke	System	Risiko	Seite
Limitiertes nicht-permanentes Cross-Site Scripting	test.bea-brak.de (185.62.147.189)	Gering (4,00)	16
Information Disclosure	test.bea-brak.de (185.62.147.189)	Gering (5,00)	19
Autorisierungsfehler	test.bea-brak.de (185.62.147.189)	Behoben	23
Gesamt	**-**	**Gering (7,54)**	**-**

4 Gefundene Schwachstellenklassen

Im folgenden Kapitel werden Schwachstellenklassen, die beim Unternehmen Atos IT Solutions and Services GmbH im Zuge des Security Audits gefunden wurden, erläutert.

4.1 Information Disclosure Schwachstellen

Fehlermeldungen, Kommentare und andere Informationen in statischen oder dynamisch generierten Webinhalten, aber auch Default- und individuelle Komponenten eines heterogenen Systems, enthalten oft Informationen, die für den Endbenutzer nicht sichtbar sein sollten. Unter diesen Bereich fallen folgende Klassen:

- Allgemeine Information Disclosure: Verwendete Softwareprodukte und deren Version. Benutzerinformationen und Login-Daten. Namen verwendeter Systemkomponenten. Defaultkomponenten verwendeter Systeme sind aktiviert und liefern sensible Informationen.

4.2 Cross-Site Scripting / -Tracing Schwachstellen

Mittels Cross-Site Scripting (XSS) Attacken werden clientseitige Scripte (JavaScript, VB-Script, etc.) mit Hilfe von Fehlern in Webapplikationen in den Webbrowser potentieller Opfer geschleust. XSS Angriffe werden sehr oft dazu benutzt, um Authentifizierungstoken (Session IDs) zu stehlen, können aber je nach Funktionalität der Applikation auch dazu benutzt werden, Inhalte von Webauftritten zu verändern oder den Webbrowser anderer Benutzer fernzusteuern.

Um Cross-Site Scripting zu verhindern, müssen spezielle Zeichen wie ", ', < oder > in der Benutzerausgabe durch ihre HTML Äquivalente (`&quot;`, `&#39;`, `&lt;`, `&gt;`) ersetzt werden. Besser jedoch ist es, einen Whitelist-Ansatz zu wählen, bei welchem im Benutzerinput nur jene Zeichen erlaubt werden, die unbedingt notwendig sind. Beispielsweise sollte ein Eingabeformularfeld nur Ziffern erlauben, wenn eine Postleitzahl einzugeben ist. Dabei ist darauf zu achten, dass diese Überprüfung serverseitig durchgeführt wird und sich nicht nur auf Felder in Formularen beschränkt.

Die in diesem Audit identifizierte Cross-Site Scripting Schwachstelle ist von geringem Risiko, da die Anzahl der für Scripte einsetzbare Zeichen stark begrenzt ist.

5 Detailanalyse

In diesem Kapitel werden die einzelnen Angriffe und die gefundenen Sicherheitslücken detailliert erläutert. Zudem werden Best Practices für eine sichere Einsatzumgebung der beA Webanwendung empfohlen.

5.1 test.bea-brak.de (185.62.147.189)

5.1.1 Allgemein

Auf dem Host `test.bea-brak.de` läuft die beA Webanwendung in einer Testumgebung. Im Zuge der Sicherheitsüberprüfung wurde diese, sowie dessen Client-Security Komponente und das zugehörige Backend überprüft. Dabei wurden unter anderem die Anwendungen decompiliert, analysiert und modifiziert sowie die Kommunikation zwischen diesen analysiert und manipuliert um Schwachstellen zu identifizieren.

Die eingesetzte TLS Transportverschlüsselung weist ein weit überdurchschnittliches Sicherheitsniveau auf. Aktuell befinden sich noch drei nicht Perfect Forward Secrecy unterstützende Chiffren im Einsatz, welche zur weiteren Steigerung des Sicherheitsniveau deaktiviert werden können.

5.1.2 Whois Information

Die Whois Information wird für jede IP Adresse aus der Permission to Attack überprüft, um sicherzustellen, dass diese auch dem Unternehmen oder dessen Vertragspartner gehört, welche auditiert werden. Die nachfolgende Tabelle stellt die öffentlich in Datenbanken verfügbare Whois Information dar.

```
inetnum:        185.62.147.0 - 185.62.147.255
netname:        BRAK-BEA
descr:          Bundesrechtsanwaltskammer Service BEA
country:        DE
admin-c:        ON936-RIPE
tech-c:         MS11367-RIPE
status:         ASSIGNED PA
mnt-by:         MOSAIC-MNT
mnt-by:         SBS-MNT
source:         RIPE
```

5.1.3 Portscan Ergebnisse

Portnummer	Protokoll	Service	Version
443	TCP	HTTP/SSL	-

Bitte beachten Sie, dass die unter Service bzw. Version beschriebenen Werte **nicht** dem realen Service entsprechen müssen.

5.1.4 Best Practices: Schutz der Einsatzumgebung der beA Webanwendung

Um die Sicherheit des Gesamtsystems zu gewährleisten ist es notwendig, dass der Schutz der Einsatzumgebung der beA Webanwendung am lokalen Computer sichergestellt wird. Dazu werden dem Benutzer der beA Webanwendung folgende ‚Best Practices' empfohlen:

Anti-Virus Software

Ein Virenscanner stellt sicher, dass sich auf einem Rechner keine bösartige Software wie Viren, Malware und Trojanische Pferde befindet. Dabei sollte der eingesetzte Computer regelmäßig vollständig von der Anti-Virus Software untersucht werden. Bei der Übertragung von Daten auf den lokalen Computer (z.B. Downloads oder USB-Sticks) muss mit Hilfe der Anti-Viren Software sichergestellt werden, dass diese keine bösartige Software enthalten.

Firewall

Die Netzwerkverbindung sollte vor Angriffen von außen geschützt werden. Dazu sollten geeignete Schutzmaßnahmen wie eine Firewall installiert sein. Handelsübliche Router bieten diese zumeist bereits an. Ist dies nicht der Fall, sollte lokal am Computer auf welchem die beA Webanwendung eingesetzt wird eine Softwarefirewall installiert werden.

Betriebssystem- / Software Updates

Um die Sicherheit des eingesetzten Betriebssystems und der Software zu gewährleisten, sollten diese regelmäßigen Updates unterzogen werden. Wo/Wenn immer möglich sollten automatische Updates aktiviert sein: Dabei ist darauf zu achten dass die automatischen Updates vom System auch wirklich automatisch (d.h. ohne Nachfrage/Zutun des Nutzers) installiert werden.

Schutz vor unbefugtem Zugriff

Es muss sichergestellt werden, dass keine Unbefugten Zugriff auf den eingesetzten Computer erhalten. Beim Verlassen des Computers sollte dieser Heruntergefahren oder mit einem Passwort gesperrt werden. Bei Abwesenheit müssen zusätzliche Schutzmaßnahmen wie das Verschließen des Raumes getroffen werden, um einen unbefugten Zugriff sowie Manipulation der Hardware zu verhindern.

Passwortsicherheit

Es sind hinreichend komplexe Passwörter einzusetzen. Passwörter sollten mindestens aus 8 Zeichen und 3 der 4 möglichen Zeichenklassen (Groß-, Klein, Ziffern und Sonderzeichen) bestehen und regelmäßig geändert werden. Des Weiteren muss sichergestellt werden, dass die eingesetzten Passwörter geheim bleiben und keinen Unbefugten bekannt werden.

Browsersicherheit

Zum Schutz des eingesetzten Computers sollte der Browser immer in seiner aktuellsten Version eingesetzt und mit Updates versorgt werden. Des Weiteren dürfen keine Plugins oder Addons aus unbekannten Quellen ausgeführt oder installiert werden.

5.1.5 Limitiertes nicht-permanentes Cross-Site Scripting

Ein Skript der Webapplikation gibt ungefilterten User-Input aus. Dies führt zu einer Cross-Site Scripting-Schwachstelle. Beim nicht-permanenten Cross-Site Scripting kann HTML- oder JavaScript-Code über einen speziellen Link auf der Webseite eingeschleust werden. Das JavaScript des Angreifers wird im Browser des Opfers im Kontext der angegriffenen Webseite ausgeführt, wenn das Opfer über den speziellen Link (z.B. aus Phishing-Emails) auf die Webseite gelangt. Die Schwachstelle kann vom Angreifer ausgenutzt werden, um Eingaben von Benutzern der Webseite mitzulesen (Keylogger-Attacken), Inhalte der Webseite zu verändern oder auf andere Seiten umzuleiten.

In diesem speziellen Fall ist die Möglichkeit die Schwachstelle auszunutzen jedoch, auf Grund der begrenzten Zeichenanzahl (max. 10 Zeichen) welche im verwundbaren Parameter zulässig sind, stark begrenzt. Trotzdem sollte die Schwachstelle behoben werden, so dass eventuelle Konfigurationsänderungen, z.B. das Erhöhen der zulässigen Zeichenanzahl, keine Auswirkungen auf die Sicherheit des Systems haben.

5.1.5.1 Proof-of-Concept

Beim Aufruf der folgenden URL wird der eingefügte JavaScript Code ausgeführt. Der Payload `';open()//` öffnet dabei beispielhaft einen neuen Tab/Pop-Up:

```
https://test.bea-brak.de/bea/index.xhtml?dswid=';open()//&jfwid=9999
```

Die Abbildung zeigt, dass beim Aufruf der URL der eingeschleuste JavaScript Payload ausgeführt und ein neuer Browsertab geöffnet wurde:

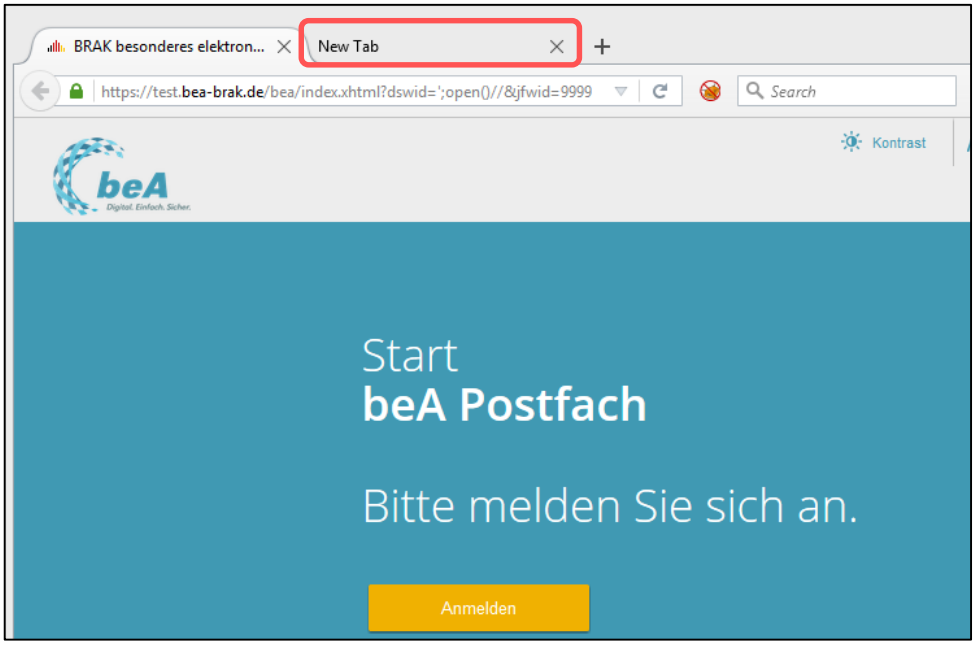


Abbildung 1: Eingeschleuster JavaScript Payload wird ausgeführt und öffnet einen neuen Tab.

5.1.5.2 Lösung

Alle Benutzer-Ein- und Ausgaben müssen einer strikten Ein- und Ausgabeüberprüfung unterzogen werden. Bei derartigen Überprüfungen sind Whitelist-Methoden den Blacklist-Methoden unbedingt vorzuziehen. Es wird empfohlen, beispielsweise in Parametern oder Eingabefeldern, welche nur numerischen Input aufweisen können (z.B. IDs, Postleitzahlen, usw.), nur die Eingabe von Zahlen zu erlauben. Dabei ist es wichtig, diese Überprüfung serverseitig durchzuführen, da clientseitige Maßnahmen umgangen werden können.

Um Cross-Site Scripting zu verhindern, müssen zusätzlich spezielle Zeichen wie z.B. [;()"´`,<>/\'=] in der Benutzerausgabe durch ihre HTML Äquivalente (", ', <, >, ...) ersetzt werden. Auf keinen Fall dürfen nur bestimmte HTML-Tags wie z.B. <script> gefiltert werden, da es zahlreiche andere Methoden gibt, Cross-Site Scripting auszunutzen bzw. derartige Filter zu umgehen.

5.1.5.3 Risiko Matrix

Schwere / Wahrscheinlichkeit	**1**	**4**	**9**	**16**	**25**
1	1	4	9	16	25
2	2	8	18	32	50
3	3	12	27	48	75
4	**4**	16	36	64	100
5	5	20	45	80	125

Schwere: Bezeichnet den Schweregrad des Fehlers (1...leicht - 25...sehr schwer).
Wahrscheinlichkeit: Bezeichnet die Wahrscheinlichkeit, mit der die Lücke von einem Angreifer ausgenutzt wird (1...unwahrscheinlich - 5...sehr wahrscheinlich).

5.1.5.4 Risiko Klassifizierung

Wahrscheinlichkeit	Die Schwachstelle ist leicht zu entdecken. Für die erfolgreiche Ausnutzung dieser ist jedoch Benutzer-Interaktion (z.B. das Besuchen einer präparierten Webseite oder das Klicken auf einen Link) erforderlich.
Schwere	Auf Grund der begrenzten Zeichenanzahl (max. 10 Zeichen) welche im verwundbaren Parameter zulässig sind, ist die Schwere als gering einzuschätzen.
Risiko	Gering (4)
ÖNORM A 7700	Kapitel 7: Behandlung von Benutzereingaben Kapitel 8: Behandlung von Datenausgaben

5.1.6 Information Disclosure

Über Information-Disclosure Schwachstellen kann ein Angreifer an Daten und Informationen über ein System gelangen, die bei weiteren Angriffen wesentliche Hilfen darstellen. Dazu gehört beispielsweise die Ausgabe von Versionsnummern. In vielen Fällen können bestimmte Schwachstellen nicht ohne weiteres Wissen über das System ausgenutzt werden. Information Disclosures erleichtern die Ausnutzung solcher Schwachstellen.

5.1.6.1 Proof-of-Concept

Der Aufruf der folgenden URL hat einen Fehler zur Folge, welcher die eingesetzte JBoss Version preisgibt:

```
https://test.bea-brak.de/bea/settings/RES_NOT_FOUND
```

Die Response des Servers mit detaillierten Informationen über die eingesetzt JBoss Version sieht wie folgt aus:

```
HTTP/1.1 404 Not Found
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=utf-8
Date: Tue, 08 Dec 2015 09:32:27 GMT
Strict-Transport-Security: max-age=15724800
X-Expires-Orig: None
Cache-Control: max-age=0, must-revalidate, private
Content-Length: 1114

<html><head><title>JBoss Web/7.5.9.Final-redhat-1 - JBWEB000064: Error
report</title><style><!--H1 {font-family:Tahoma,Arial,sans-
serif;color:white;background-color:#525D76;font-size:22px;} H2 {font-
family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-
size:16px;} H3 {font-family:Tahoma,Arial,sans-serif;color:white;background-
color:#525D76;font-size:14px;} BODY {font-family:Tahoma,Arial,sans-
serif;color:black;background-color:white;} B {font-family:Tahoma,Arial,sans-
serif;color:white;background-color:#525D76;} P {font-
family:Tahoma,Arial,sans-serif;background:white;color:black;font-size:12px;}A
{color : black;}A.name {color : black;}HR {color : #525D76;}--></style>
</head><body><h1>JBWEB000065: HTTP Status 404 -
/bea/settings/RES_NOT_FOUND</h1><HR size="1"
noshade="noshade"><p><b>JBWEB000309: type</b> JBWEB000067: Status
report</p><p><b>JBWEB000068: message</b>
<u>/bea/settings/RES_NOT_FOUND</u></p><p><b>JBWEB000069: description</b>
<u>JBWEB000124: The requested resource is not available.</u></p><HR size="1"
noshade="noshade"><h3>JBoss Web/7.5.9.Final-redhat-1</h3></body></html>
```

Der Aufruf der folgenden URL mit den Sonderzeichen `">` im Parameter `text` hat einen Fehler zur Folge,

welcher die eingesetzte EdaWeb Version sowie einen Stack Trace preisgibt:

```
https://test.bea-brak.de/xwiki/bin/view/Main/Search?text=">&f_type=DOCUMENT&f_locale=de&f_locale=&r=1
```

Die Response des Servers mit detaillierten Versionsinformationen über EdaWeb sowie Stack Traces sieht wie folgt aus:

```
HTTP/1.1 500 Internal Server Error
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=utf-8
Content-Language: en
Date: Tue, 15 Dec 2015 10:50:18 GMT
Connection: close
Strict-Transport-Security: max-age=15724800
Set-Cookie: citrix_ns_id=YrfpVilY4ZkQNslbt9KF6zzzWXQ0006; Domain=.bea-
brak.de; Path=/; HttpOnly
X-Expires-Orig: None
Cache-Control: max-age=0, must-revalidate, private
Content-Length: 2939

<!DOCTYPE html><html><head><title>EdaWeb/3.0.24 - Error report</title><style
type="text/css">H1 {font-family:Tahoma,Arial,sans-
serif;color:white;background-color:#525D76;font-size:22px;} H2 {font-
family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-
size:16px;} H3 {font-family:Tahoma,Arial,sans-serif;color:white;background-
color:#525D76;font-size:14px;} BODY {font-family:Tahoma,Arial,sans-
serif;color:black;background-color:white;} B {font-family:Tahoma,Arial,sans-
serif;color:white;background-color:#525D76;} P {font-
family:Tahoma,Arial,sans-serif;background:white;color:black;font-size:12px;}A
{color : black;}A.name {color : black;}.line {height: 1px; background-color:
#525D76; border: none;}</style> </head><body><h1>HTTP Status 500 - Invalid
URL [http://test.bea-
brak.de/xwiki/bin/view/Main/Search?text=&amp;quot;&amp;gt;&amp;amp;f_type=DOCUMENT&amp;amp;f_
locale=de&amp;amp;f_locale=&amp;amp;r=1]</h1><div class="line"></div><p><b>type</b>
Exception report</p><p><b>message</b> <u>Invalid URL [http://test.bea-
brak.de/xwiki/bin/view/Main/Search?text=&amp;quot;&amp;gt;&amp;amp;f_type=DOCUMENT&amp;amp;f_
locale=de&amp;amp;f_locale=&amp;amp;r=1]</u></p><p><b>description</b> <u>The server
encountered an internal error that prevented it from fulfilling this
request.</u></p><p><b>exception</b></p><pre>javax.servlet.ServletException:
Invalid URL [http://test.bea-
brak.de/xwiki/bin/view/Main/Search?text=&amp;quot;&amp;gt;&amp;amp;f_type=DOCUMENT&amp;amp;f_
locale=de&amp;amp;f_locale=&amp;amp;r=1]
```

```
      org.xwiki.resource.servlet.RoutingFilter.constructExtendedURL(RoutingFil
ter.java:193)
      org.xwiki.resource.servlet.RoutingFilter.doFilter(RoutingFilter.java:100
)
</pre><p><b>root
cause</b></p><pre>org.xwiki.resource.CreateResourceReferenceException:
Invalid URL [http://test.bea-
brak.de/xwiki/bin/view/Main/Search?text=&quot;&gt;&amp;f_type=DOCUMENT&amp;f_
locale=de&amp;f_locale=&amp;r=1]
      org.xwiki.url.ExtendedURL.&lt;init&gt;(ExtendedURL.java:124)
      org.xwiki.resource.servlet.RoutingFilter.constructExtendedURL(RoutingFil
ter.java:191)
      org.xwiki.resource.servlet.RoutingFilter.doFilter(RoutingFilter.java:100
)
</pre><p><b>root cause</b></p><pre>java.net.URISyntaxException: Illegal
character in query at index 56: http://test.bea-
brak.de/xwiki/bin/view/Main/Search?text=&quot;&gt;&amp;f_type=DOCUMENT&amp;f_
locale=de&amp;f_locale=&amp;r=1
      java.net.URI$Parser.fail(URI.java:2848)
      java.net.URI$Parser.checkChars(URI.java:3021)
      java.net.URI$Parser.parseHierarchical(URI.java:3111)
      java.net.URI$Parser.parse(URI.java:3053)
      java.net.URI.&lt;init&gt;(URI.java:588)
      java.net.URL.toURI(URL.java:939)
      org.xwiki.url.ExtendedURL.&lt;init&gt;(ExtendedURL.java:122)
      org.xwiki.resource.servlet.RoutingFilter.constructExtendedURL(RoutingFil
ter.java:191)
      org.xwiki.resource.servlet.RoutingFilter.doFilter(RoutingFilter.java:100
)
</pre><p><b>note</b> <u>The full stack trace of the root cause is available
in the EdaWeb/3.0.24 logs.</u></p><hr
class="line"><h3>EdaWeb/3.0.24</h3></body></html>
```

5.1.6.2 Lösung

Der Server sollte so konfiguriert sein, dass dem Benutzer keine technischen Fehlermeldungen/Informationen angezeigt werden. Debugging Output oder detaillierte Fehlermeldungen sollten in einer Produktivumgebung deaktiviert sein. Es wird empfohlen, nur einen Fehlercode ohne zusätzliche Systeminformationen anzuzeigen.

5.1.6.3 Risiko Matrix

Schwere / Wahrscheinlichkeit	**1**	**4**	**9**	**16**	**25**
1	1	4	9	16	25
2	2	8	18	32	50
3	3	12	27	48	75
4	4	16	36	64	100
5	**5**	20	45	80	125

Schwere: Bezeichnet den Schweregrad des Fehlers (1...leicht - 25...sehr schwer).
Wahrscheinlichkeit: Bezeichnet die Wahrscheinlichkeit, mit der die Lücke von einem Angreifer ausgenutzt wird (1...unwahrscheinlich - 5...sehr wahrscheinlich).

5.1.6.4 Risiko Klassifizierung

Wahrscheinlichkeit	Die Wahrscheinlichkeit, dass die Schwachstellen von einem Angreifer entdeckt werden, wird als hoch eingeschätzt.
Schwere	Es handelt sich um Information Disclosure Schwachstellen, die nicht zur direkten Kompromittierung des Systems führen. Die erhaltenen Informationen können jedoch vom Angreifer bei späteren Attacken verwertet werden.
Risiko	Gering (5)
ÖNORM A 7700	Kapitel 10: System- und Fehlermeldungen

5.1.7 Autorisierungsfehler

Autorisierungsfehler erlauben es einem Angreifer unbefugt auf Ressourcen oder Funktionen zuzugreifen, für welche er nicht berechtigt ist. Durch Ausnutzen einer solchen Schwachstelle könnte ein Angreifer auf sensible Daten anderer Benutzer zugreifen. Über den Parameter `msgid` wird im Nachrichteneingang und den Entwürfen die zu betrachtende/bearbeitende Nachricht identifiziert. Einem Angreifer war es möglich diesen Parameter zu manipulieren und somit beschränkten Zugriff auf Informationen über Nachrichten anderer Benutzer zu erhalten.

Folgende Informationen über fremde Nachrichten konnten eingesehen werden:

- Absender
- Empfänger
- Eigenes Aktenzeichen
- Aktenzeichen der Justiz
- Nachrichtentyp
- Gesendet-, Empfangen-, Zugegangen-Datum
- Dateiname, Name und Größe angehängter Dateien

Die hier beschriebene Schwachstelle wurde am Fr. 11.12.2015 an das Unternehmen Atos IT Solutions and Services GmbH gemeldet und von den Entwicklern behoben. Ein Recheck am Mo. 14.12.2015 zeigte, dass die **Schwachstelle behoben ist**.

5.1.7.1 Proof-of-Concept

Die folgenden beiden URLs sind Beispiele für den Aufruf von Nachrichten sowie Entwürfen. Durch das Abändern des Parameters `msgid` war es möglich auf Informationen über die Nachrichten anderer Benutzer zuzugreifen:

```
https://test.bea-brak.de/bea/messages/view/view.xhtml?post-
boxid=100757&msgid=106877&dswid=1870
https://test.bea-brak.de/bea/messages/create/createMes-
sage.xhtml?msgid=107628&dswid=3500
```

Der folgende Screenshot zeigt, welche Informationen ein authentifizierter Benutzer über die Nachricht mit der `msgid` 106892 eines anderen Benutzers einsehen konnte. Die Client-Security Anwendung verhinderte einen Zugriff auf den Inhalt der Nachricht, jedoch waren große Teile der Metadaten der Nachricht für einen unberechtigten Benutzer immer noch ersichtlich:

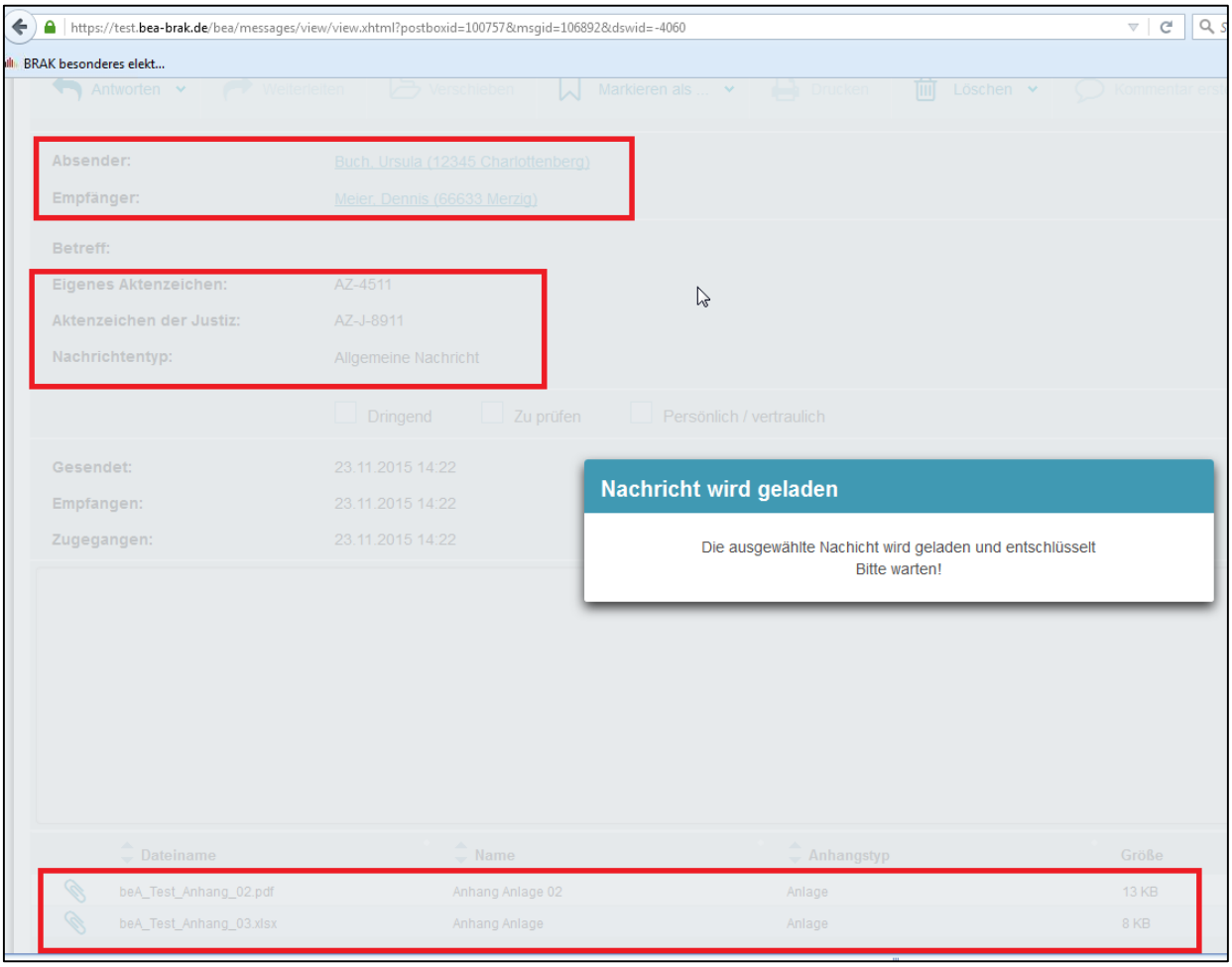


Abbildung 2: Meta-Informationen, welche ein nicht autorisierter Benutzer über Nachrichten anderer Benutzer einsehen konnte.

6 Version History

Version	Datum	Status/Änderungen	Erstellt von	Verantwortlich
1.0	18.12.2015	Finale Version	I. Lorch	I. Lorch