Sicherheit von SCOPELAND-Web-Anwendungen

Das Thema Sicherheit ist eines der Kernthemen der IT-Branche und zählt zu den notwendigen Schlüsselkompetenzen jedes Software-Anbieters, der langfristig auf dem Markt bestehen will.

SCOPELAND® unterstützt die Entwicklung sicherer Web-Anwendungen durch die „Out-of-The-Box“ Verwendung verschiedener Sicherheitsfeatures und das Bereithalten eines geeigneten Toolsets, um die Sicherheit für die jeweilige Anwendung optimal anzupassen. Die Funktionen werden ständig verbessert, um auf aktuelle Entwicklungen vorbereitet zu sein, und um künftige Trends zu antizipieren. Dabei orientiert sich Scopeland Technology an den gängigen Industriestandards und den Empfehlungen anerkannter Experten auf dem Gebiet der Web-Sicherheit, wie denen des Open Web Application Security Project (OWASP), SANS Institut und des Bundesamts für Sicherheit in der Informationstechnik (BSI).

In diesem White Paper sollen die wichtigsten Sicherheitsfeatures, Lösungen sowie deren Vorteile, die SCOPELAND® unterstützt und anbietet, vorgestellt werden.

Inhalt

1. Schutz vertraulicher Daten

Vertrauliche Daten und Funktionen für einen eingeschränkten Benutzerkreis sind Kernbestandteile der meisten Web-Anwendungen. Dies stellt Anbieter von Web-Anwendungen vor die Aufgabe, einen scheinbaren Widerspruch aufzulösen: Eine Web-Anwendung soll per Definition aus dem öffentlichen und für jeden Zugänglichen „Web“ erreichbar sein, auf der anderen Seite vertrauliche Daten und Funktionen vor unberechtigtem Zugriff schützen.

1.1. Authentifizierung von Benutzern

Typischerweise muss ein Benutzer einer Web-Anwendung sich per Benutzername und Passwort einloggen. SCOPELAND® bietet standardmäßig an, Daten gegen eine Datenbank oder per Lightweight Directory Access Protocol (LDAP) / Active Directory (AD) zu prüfen. Passwörter werden dafür verschlüsselt („gehasht“) abgespeichert. Dies erschwert ein illegales Auslesen der Passwörter, falls sich ein Angreifer lesenden Zugriff auf die Benutzerdaten verschafft.

Alle größeren Webserver bieten ausgereifte Standardlösungen für die Authentifizierung von Usern an. SCOPELAND® unterstützt diese Lösungen und empfiehlt die Verwendung des PBKDF2 Algorithmus mit zusätzlichem Salt. Damit werden neben einfachen Brute-Force-Angriffen auch fortschrittlichere Rainbow-Table-Angriffe unwirtschaftlich und quasi ausgeschlossen.

Zudem kann per SCOPELAND® eine Password Policy definiert werden, um einem der größten Risiken von Web-Anwendungen entgegen zu wirken: Durch den Nutzer selbst gewählte, schwache und daher unsichere Passwörter.

1.2. Autorisierung des Benutzers für bestimmte Ressourcen

SCOPELAND® bedient sich des Rollenkonzepts, welches von den meisten Webservern angeboten wird. Der Webserver stellt sicher, dass nur Nutzer, die über gewisse Rollen verfügen, Zugriff auf bestimmte Seiten oder Ordner sowie den darin hinterlegten Daten, Ressourcen und Funktionen haben.

Für viele Anwendungsbereiche ist dies jedoch zu grob. Daher bietet SCOPELAND® weiter die Möglichkeit, innerhalb der Anwendungslogik dynamisch zu entscheiden, ob ein Nutzer Zugriff auf bestimmte Daten, Ressourcen oder Funktionen erhält. SCOPELAND® bietet hierfür entsprechende Einstellungsmöglichkeiten und Application-Programming-Interfaces (APIs) an, um dem Anwendungsentwickler die nötigen Andockpunkte für die zu implementierende Geschäftslogik zu bieten. So kann beispielsweise vor besonders

sicherheitskritischen Tätigkeiten nochmals eine Authentifizierung erzwungen werden, um sicherzustellen, dass tatsächlich noch der berechtigte Nutzer die Anwendung bedient. Weiter lassen sich die Berechtigungen der Nutzer für die Verwendung in der Applet-Logik auslesen und Nutzer können zwangsweise abgemeldet werden.

1.3. Schutz der Kommunikation zwischen Benutzer und Server

Eine Verschlüsselung der Kommunikation zwischen dem Browser des Benutzers und des Servers ist notwendig, um Vertraulichkeit zu erreichen. SCOPELAND® empfiehlt und unterstützt hier den Industriestandard **Transport Layer Security** (**TLS)**, der besser bekannt ist unter seiner Vorgängerbezeichnung **Secure Sockets Layer** / **SSL**.

Um **Man-In-The-Middle-Angriffe** zu unterbinden, werden SCOPELAND®-Anwendungen in der Regel so konfiguriert, dass sie strikt auf eine korrekte Signierung der Zertifikate durch eine **Certificate Authority** achten.

1.4. Verschlüsselung von sensiblen Daten

Sensible Daten sollten auch in der Datenbank nicht im Klartext lesbar sein. Neben dem bereits angesprochenen asymmetrischen Hashing-Verfahren, welches für Passwörter verwendet wird, benötigen andere Daten eine symmetrische Verschlüsselung. Die großen Datenbanksysteme bieten dafür Funktionen an, um Teile oder auch die ganze Datenbank zu verschlüsseln. Scopeland Technology empfiehlt, diese Funktionen zu nutzen, wenn mit besonders sensiblen Daten umgegangen wird. Hierbei sollte beachtet werden, dass die Ent-/Verschlüsselung zu Lasten der Performance geht und daher nur überlegt eingesetzt werden sollte.

2. Sicherheit bei Ein-/Ausgabe von Daten in Web-Anwendungen

Die Möglichkeit für Benutzer, Daten an die Anwendung zur weiteren Verarbeitung einzugeben ist essentiell für moderne Web-Anwendungen. Hiermit bietet sich aber auch ein mögliches Einfallstor für Angriffe, wenn die Anwendung solche Eingaben nicht mit der notwendigen Sorgfalt prüft und als Gefahrenquelle behandelt. So könnte im schlimmsten Fall Programmcode eines Angreifers zur Ausführung gebracht werden.

2.1. Eingabevalidierung

Der erste Schritt umfasst eine konsequente **Validierung** der Eingaben. SCOPELAND® unterstützt die Standard-Funktionen der jeweiligen Plattform (JSF, .Net), durch die bereits eine zufriedenstellende Qualität der Validierung erreicht wird. Standard-Fälle wären zum Beispiel, dass in Zahlen-Feldern nur Zahlen eingegeben werden können, Texte einer bestimmten Form und Länge entsprechen, oder Werte nach einem Whitelist-Verfahren geprüft werden. Darüber hinaus bietet SCOPELAND® die Möglichkeit, **Regular Expression als Validierungs-Regeln** zu definieren. Damit erhält der Anwendungsentwickler ein mächtiges Werkzeug, um die Validierung auch per Muster und nicht nur per Whitelist-Verfahren implementieren zu können.

2.2. Validierung von Uploads

Für die Validierung von Datei-Uploads steht die Möglichkeit zur Verfügung, Uploads auf bestimmte Dateitypen zu beschränken. Dabei verlässt sich SCOPELAND® nicht allein auf die Datei-Endung, sondern prüft auch den Datei-Inhalt. Damit kann zuverlässig verhindert werden, dass Schadcode über Skripte oder ähnliches hochgeladen und danach (in der Regel über Ausnutzung irgendeiner Sicherheitslücke) ausgeführt werden kann.

2.3. Spezialbehandlung für Parameter

Auch nach der Validierung müssen Benutzereingaben immer als potentiell gefährlich angesehen werden. In SCOPELAND®-Anwendungen werden Benutzereingaben als **SQL-Parameter** mit SQL verknüpft, wodurch wirksam **SQL-Injections** verhindert werden.

2.4. Escaping von Ausgabe-Daten

Wenn Daten zum Aufbau einer Webseite benutzt werden, muss immer beachtet werden, dass solche Daten manipuliert sein könnten – mit dem Ziel, der Webseite in unerlaubter Weise zu schaden. Damit dies nicht passiert, unterstützt SCOPELAND® das **Escaping von Ausgabe-Daten**: Etwaige Steuerungs- oder Sonderzeichen, welche die Darstellung ungewollt beeinflussen könnten, werden in eine harmlose Variante umgewandelt. Dies verhindert unter anderem das weit verbreitete **Cross-Site-Scripting**.

In Spezialfällen kann Escaping hinderlich sein, daher ist es möglich das Escaping feingranular zu steuern. Der Anwendungsentwickler kann an nötigen Stellen die Ausgabe auf diese Weise sehr flexibel dynamisch umbauen. Dies muss vom Anwendungsentwickler bewusst aktiviert und mit besonderer Sorgfalt behandelt werden, da ein Abschalten des Escapings ansonsten Risiken birgt.

2.5. URL-Parameter

Nicht nur Eingaben per Web-Formular sind eine Gefahrenquelle. Auch das verbreitete Übergeben von Parametern per URL ist ein möglicher Angriffsvektor. Hierfür besitzt SCOPELAND® ein besonderes Feature: Die **Verschlüsselung der URL-Parameter**. Durch eine symmetrische Verschlüsselung werden die URL-Parameter für einen potentiellen Angreifer nicht mehr lesbar, was eine unerlaubte Manipulation der URL-Parameter erheblich erschwert.

3. Cross-Site-Angriffe

Es existieren verschiedene sogenannte „Cross-Site“-Angriffsmethoden, die darauf basieren, eine gültige Session eines legal angemeldeten Benutzers auszunutzen – durch das Unterschieben von illegalen Requests. Dazu gehören **Cross-Site Request Forgery (XSRF)**, **Session Riding, Clickjacking** und das bereits erwähnte **Cross-Site-Scripting (XSS)**.

SCOPELAND® setzt mehrere Techniken ein, um solche Angriffe zu verhindern. So wird in SCOPELAND®-Anwendungen die vom *World Wide Web Consortium* (W3C) empfohlene **Content Security Police (CSP)** realisiert. Der Zugriff auf Webseiten außerhalb der eigenen Anwendung ist damit standardmäßig nicht erlaubt. Auch per **X-Frame-Options** wird die **Same-Origin-Policy** verwendet, was Cross-Site-Angriffe oder das Laden von Ressourcen anderer Seiten stark erschwert. Vom Anwendungsentwickler kann der Zugriff in eigener Verantwortung pauschal oder gezielt für bestimmte Webseiten frei gegeben werden, falls nötig. Auch die Verwendung eines **Anti-XSRF-Tokens** kann aktiviert werden. Damit wird für jede User-Session ein kryptografisch sicherer Token erzeugt, welcher als Sicherheitsmerkmal in alle Seiten eingefügt und in allen Requests des Users erwartet wird. Illegale, aus anderer Quelle untergeschobene Requests können so erkannt werden.

4. Session-ID-Sicherheit

Um zu verhindern, dass die im Cookie gesetzte Session-ID ausgelesen wird, verwenden SCOPELAND®-Web-Anwendungen standardmäßig das **HttpOnly-Flag**. Die Session-ID wird nach dem ersten Anmelden neu vergeben und nicht über die URL sichtbar gemacht. Die Session-ID wird mit kryptografisch sicheren Algorithmen generiert, was **Brute-Force-Attacken** gegen die Session-ID praktisch unmöglich macht und **Session Hijacking** damit stark erschwert.

Natürlich unterstützten SCOPELAND®-Web-Anwendungen einen **Session-Timeout**. Es wird empfohlen, die Timeout-Zeit minimal zu wählen – im Kontext der Anforderungen.

5. Ausfallsicherheit

Ob durch Hacking-Angriffe oder durch schlichtes technisches Versagen: Server und Hardware können ausfallen. Damit in diesem Fall nicht die ganze Anwendung bis zu einer Reparatur nicht verfügbar ist, empfiehlt Scopeland Technology, die Anwendungen immer auf mehrere Server zu verteilen. SCOPELAND®-Web-Anwendungen sind Cluster-fähig und können daher auf mehrere Rechner verteilt werden. Zudem sind SCOPELAND®-Anwendungen Cloud-fähig und können folglich von geeigneten Cloud-Anbietern gehostet werden.

6. Angriffe durch Überlastung: Distributed-Denial-of-Service-Angriffe (DDoS)

Distributed-Denial-of-Service-Angriffe (DDoS, auch verkürzt **Denial of Service**, **DoS**) sind selbst für große Firmen ein Problem. Da hier die Infrastruktur angegriffen wird, und nicht die Anwendung selbst, kann Scopeland Technology hierfür keine direkte Lösung erarbeiten, bietet jedoch eine umfassende Beratung an, welche Maßnahmen ergriffen werden sollten, um auf DoS-Angriffe vorbereitet zu sein.

Spezielle Filter-Software für Firewalls, Router und Proxys kann eingesetzt werden, um DoS-Attacken zu erkennen und deren Auswirkungen zumindest stark zu mindern. Hierfür kann der Traffic auch über externe Filter-Services darauf spezialisierter Firmen geleitet werden.

Gegen viele DoS-Angriffe reicht aber bereits eine Lastverteilung mit ausreichend dimensionierten Reserven. Wie bereits erwähnt, sind SCOPELAND®-Web-Anwendungen Cluster- und Cloud-fähig und können daher auf mehrere Rechner verteilt werden. Da gerade Cloud-Anbieter über sehr große Reserven an Rechenleistung verfügen, welche dynamisch hinzugeschaltet werden kann, ist dies ein guter Weg, gegen DoS-Attacken gewappnet zu sein.

7. Unerlaubte automatisierte Nutzung von Web-Anwendungen

Bei der Eingabe von ungültigen Anmeldedaten werden keine Informationen zurück geliefert, die darauf schließen lassen, ob nur das Passwort falsch war, oder ob auch das Benutzerkonto nicht existiert. Dadurch wird es automatisierten Brute-Force-Angriffen schwergemacht, Zugangsdaten zu ermitteln – bei ausreichend sicheren Passwörtern ist dies quasi unmöglich. Weiter gibt es, je nach Anmeldetyp SCOPELAND®-intern oder bereitgestellt durch das verwendete Authentifizierungsverfahren (wie beispielsweise Windows-Authentifizierung), die Möglichkeit, bei zu vielen Fehlversuchen den Zugang zu sperren oder die Anmeldeversuche zu verzögern.

Bei anderen Eingaben (außer der Anmeldung) muss der Anwendungsentwickler sicherstellen, dass bei Fehleingaben keine Antworten gegeben werden, die sicherheitskritische Informationen enthalten.

Für die Abwehr von automatisierter Nutzung kommen häufig auch CAPTCHAs (Completely Automated Public Turing Test To Tell Computers and Humans Apart) zum Einsatz. Da auf SCOPELAND®-Web-Anwendungen meist nur authentifizierte Nutzer Zugriff haben und daher nach erfolgter Anmeldung eine unerlaubte automatisierte Nutzung sehr unwahrscheinlich ist, gab es bisher keinen Bedarf für den Einsatz von CAPTCHAs. Mechanismen für die Einbindung von CAPTCHAs könnten jedoch bei Bedarf kurzfristig in das Produkt integriert werden.

Weiter sind Infrastrukturmaßnahmen wie Teergruben denkbar, um eine Anwendung vor automatisierter Nutzung zu schützen – beispielsweise auf der Basis von IP-Adressen. Scopeland Technology bietet hier keine eigene Lösung an, sondern eine umfassende Beratung, welche Maßnahmen ergriffen werden können.

8. Vom Anwendungsentwickler durchzuführende Maßnahmen

Einem SCOPELAND®-Anwendungsentwickler wird viel Arbeit für sicherheitsrelevante Konfiguration und Prüfung abgenommen. Diverse Abläufe funktionieren automatisch, so dass der Entwickler sich möglichst auf die Geschäftslogik konzentrieren kann. Vorgänge, wie das Setzen von Security-Headern, Ausgabe-Escaping, sichere Sessions-IDs erzeugen, Erzeugen von sicheren SQL-Abfragen und anderes, funktionieren ganz einfach im Hintergrund.

Trotzdem gibt es Dinge, die SCOPELAND® dem Anwendungsentwickler nicht abnehmen kann, und die er daher selbst sicherzustellen hat. Daraus ergeben sich folgende Vorgaben für die Anwendungsentwicklung:

Konfiguration

- Session-Timeout immer minimal im Kontext der Anforderungen wählen.
- Darauf achten, vor einem Produktivbetrieb alle Test- und Standardkonfigurationen zu entfernen, inklusive Test-Nutzer.
- Eine Passwortrichtlinie vergeben. Als Minimalanforderung sollte ein Passwort acht Zeichen lang sein, sowie Buchstaben und Zahlen enthalten.
- Sicherheitseinstellungen des Servers überprüfen. Wird durchgängig TLS/SSL verwendet? Sind nur benötigte Ports ansteuerbar? Korrekte Log-Einstellungen? Korrekte Fehler-Anzeigen? Sind die Response-Header des Servers so eingestellt, dass möglichst wenig über die Umgebung preisgegeben wird (Webserver Fingerprinting)?
- Ist Management-Zugriff ggfs. nur aus Intranet möglich?
- Fehlerseiten für Produktiveinsatz konfigurieren. Je nach Anforderungen Anti-XSRF-Token aktivieren und konfigurieren.
- Neuere SCOPELAND®-Versionen benutzen auch neuere Software-Komponenten von Drittanbietern. Daher empfehlen wir, immer mit neuen SCOPELAND®-Versionen zu arbeiten.

Rechte-Management

- Seiten und andere Ressourcen (Dokumente, Vorlagen, XML-Dateien, etc.) nach Rollen-Zugriffsrechten in Ordnern sammeln und diesen Ordnern dann die Rechte für die Rollen geben.
- Soweit möglich sollte es vermieden werden Rollen- und Rechteprüfungen selbst zu implementieren. Empfohlenes Vorgehen ist, die Seiten so zu erstellen, dass diese klar den Rollen zugeordnet werden können und dem Server den Zugriff über das Rollenmodell absichern zu lassen.

- Wird Rechtemanagement selbst implementiert, sind Rechte und Rollen niemals per URL-Parameter zu steuern. Prüfungen sollten immer wieder neu erfolgen. Wenn Daten zu Rechten und Rollen gespeichert werden, dann nur als Session-Variablen oder in der DB.
- Falls User innerhalb einer Seite unterschiedliche Rechte haben können (Herr Meier darf den Datensatz bearbeiten, Herr Müller nicht; Herr Meier darf Datensatz X sehen, Herr Müller nicht), sollte immer vor Ausführung der Aktion/Anzeige des Datensatzes geprüft werden, ob der Nutzer das entsprechende Recht hat.
- Wenn Daten in einem temporären Verzeichnis gespeichert werden, auf Zugangsbeschränkungen auf dieses Verzeichnis achten, damit keine unerwünschten Downloads von sensiblen Daten möglich sind. Beim IIS muss hierfür ein RequestFiltering auf diesem Verzeichnis eingerichtet werden; für den Tomcat sind keine zusätzlichen Einstellungen nötig, wenn der Standard-Ordner für temporäre Dateien benutzt wird.

Geschäftslogik

- Validierungsregeln für Eingabefelder sollten möglichst genau gesetzt werden.
- Wenn möglich, Benutzereingaben per Whitelist-Verfahren prüfen: „Ist diese Eingabe für dieses Feld überhaupt zulässig?“
- Dateiuploads mit fester Dateierweiterung benutzen, um das Hochladen gefährlicher Dateitypen zu verhindern.
- Keine Daten aus der Datenbank direkt in Formulare/Links schreiben und für die Navigation oder Logik-Steuerung benutzen. Beispiel: Nicht in Links die Primary-Keys von Datensätzen des Ziels schreiben. Besser: Temporäre IDs vergeben und diese mit den PK verknüpfen. So können keine Ziele angesteuert werden, die nicht erlaubt sind, da diese in den temporären IDs nicht enthalten sind.
- SCOPELAND® bietet für Experten die Möglichkeit SQL-Macros zu nutzen. Dies ist ein mächtiges Werkzeug, für dessen Einsatz entsprechende Fachkenntnis vorausgesetzt wird. Besonders zu beachten ist, dass für Parameter die an das SQL-Macro übergeben werden KEINE besonderen Sicherheitsmaßnahmen durch die SCOPELAND®-Runtime vorgenommen werden. Es dürfen also KEINE ungeprüften User-Eingaben in SQL-Macros benutzt werden, sonst kann dies SQL-Injections ermöglichen.

- Buttons bieten beim Navigieren zwischen Seiten in punkto Sicherheit einige Vorteile gegenüber Links. Sollen Daten von einer Seite zur nächsten übergeben werden, bieten Links nur die Möglichkeit dies per URL-Parameter zu realisieren. Wird per Button navigiert, können statt URL-Parameter auch Session-Variablen gesetzt werden. Dies ist sicher gegenüber Manipulationen des Users und sollte daher in Erwägung gezogen werden.
- Eine Navigation auf andere Seiten nicht direkt von Nutzereingaben abhängig machen. Beispiel: Keinen „OpenURL"-Aufruf durchführen mit einem Wert, der direkt vom Nutzer kommt. Hier immer prüfen/filtern, dass nur zulässige Werte aufgerufen werden.
- Bei sensiblen Formularfeldern das AutoComplete anwendungsseitig deaktivieren.
- Antworten auf Fehleingaben dürfen keine sicherheitskritischen Informationen enthalten.
- Bei besonders kritischen Operationen den Nutzer nochmal zur Eingabe des Passworts auffordern. Beispiele sind die Änderung des Passworts oder das Löschen von besonders wichtigen Daten.
- Bei Nutzung von XHTML-Fragmenten müssen Nutzer-Eingaben besonders sorgfältig geprüft werden, da die automatische Validierung durch JSF/ ASP hier nicht greift. Die Option *„ContentType=text/html"* darf nur genutzt werden, wenn die enthaltenden Daten nicht vom Nutzer geändert werden können. *„ContentType=text/html"* deaktiviert das Escaping des Inhaltes und würde sonst einen möglichen Cross-Site-Scripting Angriffsvektor öffnen.
- Werden HTML-Emails versendet und werden darin Daten verwendet, die vom User stammen, müssen diese User-Daten vom Anwendungsentwickler selbst escaped werden. Dazu stellt SCOPELAND® entsprechende Funktionen bereit.
- Wird eigenständig geschriebener JavaScript Code in SCOPELAND Anwendungen eingebaut, muss der JavaScript Code den gängigen Richtlinien für sicheren JS-Code entsprechen.
- Bei Nutzung des XHTML-Editor-Controls wird die *Content Security Policy* deaktiviert. Bei hohen Sicherheitsanforderungen sollte am besten auf das Control verzichten werden.
- Bei Nutzung eines AutoRefresh Controls greift der Timeout nicht, solange noch eine Seite geöffnet ist. Dies ist beim Einsatz zu bedenken und abzuwägen.
- Bei SQL-Statements keinen IN-Operator verwenden. Diese werden ohne SQL-Parameter verwendet und eröffnen daher zumindest theoretisch die Möglichkeit von SQL-Injection.

9. Anhang

9.1. Vergleich mit gebräuchlichen Sicherheitsstandards

Es gibt verschiedene Sicherheitsstandards, an denen die Sicherheit von Web-Anwendungen gemessen werden kann. Um eine bessere Einordnung der Schutzmaßnahmen in SCOPELAND® zu ermöglichen, vergleichen wir diese im Folgenden mit den Standards OWASP TOP 10 und CWE/SANS TOP.

9.2. OWASP TOP 10

Das Open Web Application Security Project (OWASP) ist die vermutlich bekannteste Institution in Fragen der Sicherheit in Web-Anwendungen. In regelmäßigen Abständen veröffentlich das OWASP eine Top 10 der ihrer Einschätzung nach gefährlichsten Angriffsmethoden auf Web-Anwendungen. Im Folgenden finden Sie die OWASP TOP 10 und Schutzmaßnahmen durch SCOPELAND®.

	OWASP Top 10 (2017)	Schutzmaßnahme SCOPELAND®
1	Injection	Eingabevalidierung, SQL-Parameter
2	Broken Authentication	Starke Passwort-Hashing Verfahren, kryptografisch sichere Session IDs, nach dem Einloggen neue Session-ID, Session-ID nicht in URL, Session-Timeout, HttpOnly-Flag
3	Sensitive Data Exposure	SSL/TLS, Verschlüsselung sensibler Daten, Passwörter stark hashen
4	XML External Entities (XXE)	Nachladen externer Dateien (wie DTD) nur wo nötig, Verwendung aktueller Komponenten, Validierung der Eingabedaten
5	Broken Access Control	Benutzung von Standardverfahren des Webservers zur Autorisierung, spezielle Fälle durch Anwendungsentwickler sichergestellt
6	Security Misconfiguration	Toolgestützte Konfiguration (*Scopeland Configuration Manager*) Konfiguration
7	Cross-Site Scripting (XSS)	Eingabevalidierung, Ausgabe-Escaping, Content Security Police, X-Frame-Options mit Same-Origin-Policy
8	Insecure Deserialization	Scopeland® verwendet direkt keine Objektserialisierung; Absicherung indirekter Serialisierung von Komponenten durch Verwendung aktueller Versionen ohne bekannte Schwachstellen
9	Using Components with Known Vulnerabilities	Verwendung aktueller Komponenten; vor Release automatische Prüfung ob verwendete Komponenten eine als *Common Vulnerabilities and Exposures* (CVE) gemeldete Schwachstellen enthalten
10	Insufficient Logging & Monitoring	Vielfältige, konfigurierbare Möglichkeiten des Loggings und Monitorings

9.3. CWE/SANS TOP 25

Das US-amerikanische SANS Institut ist spezialisiert auf Forschung, Entwicklung, Training und Informationsvermittlung auf dem Feld der IT-Sicherheit. Zusammen mit der Common Weakness Enumeration (CWE) Community gibt SANS eine Liste von den ihrer Einschätzung nach 25 „Most Dangerous Software Errors“ heraus.

Im Folgenden sind die 25 Punkte und die Schutzmaßnahmen durch SCOPELAND® dokumentiert.

CWE ID	Name	Schutzmaßnahme SCOPELAND®
CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	Eingabevalidierung, Ausgabe-Escaping, Content Security Police, X-Frame-Options mit Same-Origin-Policy
CWE-787	Out-of-bounds Write	Nicht betroffen, durch Java/.Net
CWE-20	Improper Input Validation	Weitreichende Eingabevalidierung
CWE-125	Out-of-bounds Read	Nicht betroffen, durch Java/.Net
CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	Nicht betroffen, durch Java/.Net
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	Eingabevalidierung, Prepared SQL Statements
CWE-200	Exposure of Sensitive Information to an Unauthorized Actor	Produkt unterstützt ein rollenbasiertes Rechtesystem, um Usern nur notwendige Daten zugänglich zu machen
CWE-416	Use After Free	Nicht betroffen, durch Java/.Net
CWE-352	Cross-Site Request Forgery (CSRF)	Anti-XSRF-Tokens auf Session Ebene
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	OS-Aufrufe sind nicht Bestandteil des Produkts; falls benötigt sind diese vom Anwendungsentwickler abzusichern
CWE-190	Integer Overflow or Wraparound	Sicherstellung durch Tests mit Grenzwerten
CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	Rollen/Rechte Einstellungen, Aktivierung von Sicherheitsfunktionen welche ein freies Aufrufen von URLs verbieten
CWE-476	NULL Pointer Dereference	Fehlertolerante Programmierung und Execption-Handling
CWE-287	Improper Authentication	Benutzung von Standardverfahren des Webservers zur Autorisierung
CWE-434	Unrestricted Upload of File with Dangerous Type	Whitelist Filterung möglich, durch Anwendungsentwickler zu konfigurieren
CWE-732	Incorrect Permission Assignment for Critical Resource	Wird durch Anwendungsentwickler sichergestellt

CWE-94	Improper Control of Generation of Code ('Code Injection')	Eingabevalidierung, Ausgabe-Escaping, im Normalfall keine Verwendung von Usereingaben in Ausführbaren Kontext
CWE-522	Insufficiently Protected Credentials	Wird durch Anwendungsentwickler sichergestellt
CWE-611	Improper Restriction of XML External Entity Reference	Nachladen externer Dateien (wie DTD) nur bei Notwendigkeit, Verwendung aktueller Komponenten, Validierung der Eingabedaten
CWE-798	Use of Hard-coded Credentials	Keine Verwendung von Hard-coded Credentials
CWE-502	Deserialization of Untrusted Data	SCOPELAND® verwendet direkt keine Objektserialisierung, Absicherung indirekter Serialisierung von Komponenten durch Verwendung aktueller Versionen ohne bekannte Schwachstellen
CWE-269	Improper Privilege Management	Produkt unterstützt ein rollenbasiertes Rechtesystem, um Usern nur notwendige Privilegien zugänglich zu machen
CWE-400	Uncontrolled Resource Consumption	Basisfunktionen wie Limitierung der Datenmenge können von Anwendungsentwickler sichergestellt werden; für weitergehende Anforderungen wird die Verwendung einer *Web Application Firewall* empfohlen.
CWE-306	Missing Authentication for Critical Function	Produkt unterstützt ein rollenbasiertes Rechtesystem, um Usern nur notwendige Privilegien zugänglich zu machen
CWE-862	Missing Authorization	Benutzung von Standardverfahren des Webservers zur Autorisierung

9.4. BSI IT-Grundschutz-Kompendium

Im Folgenden werden Punkte des BSI IT-Grundschutz-Kompendiums aufgeführt, die auf die Entwicklung und Betrieb von Webanwendungen (APP.3.1) zutreffen. Die aufgeführten Punkte beziehen sich auf die technischen Anforderungen. Nicht technische Anforderungen (wie Anforderungsmanagement, Projektdokumentation oder Vorgehensmodelle) werden hier nicht behandelt und sind in der Liste nicht enthalten. Es werden Basis- und Standardanforderungen nach BSI beschrieben. Um eine höhere Stufe (erhöhter Schutzbedarf) zu erreichen, sind individuelle Analysen für die jeweilige Anwendung nötig.

BSI-ID	Geforderte Schutzmaßnahme	Schutzmaßnahme SCOPELAND®
APP.3.1.A1	Authentisierung bei Webanwendungen	Benutzung von Standardverfahren des Webserverr; Passwort-Richtlinie
APP.3.1.A2	Zugriffskontrolle bei Webanwendungen	Rollen/Rechte System zur Authentifizierung; Spezielle Fälle durch Anwendungsentwickler sichergestellt
APP.3.1.A3	Sicheres Session-Management	Kryptografisch sichere Session IDs, nach dem Einloggen neue Session-ID. Session-ID nicht in URL, Session-Timeout, HttpOnly-Flag

APP.3.1.A4	Kontrolliertes Einbinden von Daten und Inhalten bei Webanwendungen	Möglichkeit für Whitelist Filterung, Uploads nur in spezielle Ordner, Ausgabe-Escaping
APP.3.1.A5	Protokollierung sicherheitsrelevanter Ereignisse von Webanwendungen	Umfangreiches Logging im System
APP.3.1.A7	Schutz vor unerlaubter automatisierter Nutzung von Webanwendungen	Je nach Anwendungsfall von Anwendungsentwickler sicherzustellen, unter Umständen kann auch die Verwendung einer *Web Application Firewall* empfehlenswert sein
APP.3.1.A14	Schutz vertraulicher Daten	Datenübermittlung via http-POST, SSL/TLS, Starke Passwort-Hashing Verfahren
APP.3.1.A16	Umfassende Eingabevalidierung und Ausgabekodierung	Eine umfassende Eingabevalidierung und Ausgabekodierung findet standardmäßig statt
APP.3.1.A19	Schutz vor SQL-Injection	Eingabevalidierung, Prepared SQL Statements
APP.3.1.A11	Sichere Anbindung von Hintergrundsystemen	Einrichtung eines Datenbankservers nach üblichen Sicherheitsstandards empfohlen, bei weiteren Hintergrundsystemen Anwendungsspezifisch festzulegen
APP.3.1.A12	Sichere Konfiguration von Webanwendungen	Unterstützt von *Scopeland Configuration Manager*, nicht benötigte http-Methoden deaktiviert, Zugriffsversuche können beschränkt werden
APP.3.1.A13	Restriktive Herausgabe sicherheitsrelevanter Informationen	Restriktive Fehlerseiten, möglichst wenig Informationen über das System nach Außen
APP.3.1.A15	Verifikation essenzieller Änderungen	Wird durch Anwendungsentwickler sichergestellt
APP.3.1.A17	Fehlerbehandlung	Protokollierung von Fehlern, sorgfältige Fehlerbehandlung
APP.3.1.A21	Sichere HTTP-Konfiguration bei Webanwendungen	Content-Security-Policy (CSP), X-FRAME-OPTIONS, Strict-Transport-Security, httponly Cookie
APP.3.1.A23	Verhinderung von Cross-Site-Request-Forgery	Anti-XSRF-Tokens

Scopeland Technology GmbH
Düsterhauptstraße 39 - 40
D - 13469 Berlin
Tel. +49 (30) 209 670 - 0
Fax +49 (30) 209 670 - 111
info@scopeland.de
www.scopeland.de

Geschäftsführer: Karsten Noack
Amtsgericht Charlottenburg HRB 176787 B
USt.-Nr.: DE 183446349