IT-Sicherheit des Integrationsportals „JobIMPULS“

Risikoanalyse, Maßnahmen und Festlegungen zum Schutz der Webanwendungen

Stand: 09.2021

Version 2.5

Inhalt

1 Risiken organisatorischer Mängel

1.1. Auswahl und Konzeption der Webanwendungen

Das Integrationsportal JobIMPULS ist wie andere Webanwendungen auch ein verteiltes, komplexes System, welches aus unterschiedlichen Komponenten (z. B. Webserver, Applikationsserver, Hintergrundsysteme) und zugehörigen Schnittstellen besteht.

Bei individuell entwickelten Webanwendungen sind im Allgemeinen die Frameworks, Komponenten und Schnittstellen im Rahmen der Konzeption auszuwählen und deren Einbindung und Absicherung zu betrachten. Im Integrationsportal JobIMPULS kommen ausschließlich Open Source Technologien zum Einsatz, also keine kommerziellen Standardprodukte.

Aufgrund der hohen Komplexität der Technologie haben die Auswahl und Definition von Frameworks, Komponenten und Schnittstellen eine hohe Bedeutung.

Hierzu gehören u. a.:

- Einheitlich umgesetzte Sicherheitsfunktionen (Authentisierung, Autorisierung).
- Serverseitig umgesetzte Sicherheitsfunktionen, damit clientseitige Manipulationen des Browsers keinen Zugang zu schutzbedürftigen Daten und Funktionen ermöglichen.
- Trennung zwischen Webservern, Datenbankservern und weiteren Hintergrundservern. Der Zugang zu den Datenbankservern ist nicht über eine public IP, sondern nur über eine private IP möglich.
- Erzeugung sicherer SessionIDs und Cookies.
- Validierung von Eingabedaten in Formularen.

1.2. Entwicklung und Erweiterung der Webanwendungen

Neu- und Weiterentwicklungen erfolgen auf der Grundlage klarer Vorgaben, eines standardisierten Vorgehens und eines entsprechenden Test- und Releasemanagements.

1.2.1 Vorgehensmodell

Alle Entwicklungsphasen werden strukturiert durchlaufen, Sicherheitsaspekte dabei von Beginn an berücksichtigt. Tests inkl. Sicherheitstests werden zunächst im Testsystem und dann im Testkonto des Livesystems durch den Chefprogrammierer (Head of Development/Seniorentwickler) und weitere Tester durchgeführt, bevor eine neue Funktion freigeschaltet wird.

Im Detail wird wie folgt vorgegangen (hier standardisiert dargestellt):

- Anforderung wird angemeldet (durch Kunden, Weiterentwicklungsprozess, Fehlerbehandlung etc.).
- Anforderung wird auf Logik und Sinnhaftigkeit geprüft. Erste Sicherheits-, Datenverarbeitungs- und Datensparsamkeitsaspekte werden identifiziert (Kundenbetreuer, Produktmanager, Projektmanager).
- Anforderung wird in das Ticketsystem aufgenommen und dort den kompletten Lebenszyklus über dokumentiert.

- Das Ticketsystem (Redmine) ist durch Passwort- und Rollenzugriffsregelung geschützt. Veränderungen am Ticket werden unwiderruflich dokumentiert. Check-ins von Programmcodes können nur mit einer entsprechenden Ticketnummer durchgeführt werden. Dadurch ist eine eindeutige Nachvollziehbarkeit von Veränderungen am System gegeben.
- Die Anforderung wird gegen den aktuellen Softwarestand auf Implementationsaufwand sowie gegen weitere Abhängigkeiten geprüft (Head of Entwicklung bzw. durch Senior-Entwickler).
- Eventuelle Korrektur oder Erweiterung der Anforderungen wird am Ticket dokumentiert (Head of Entwicklung bzw. durch Senior-Entwickler).
- Eine Anforderung wird durch den zugewiesenen Entwickler durchgeführt.
- Rückmeldung erfolgt durch Entwickler bei evtl. Fragen und Ergänzungen durch entsprechenden fachlich verantwortlichen im Ticket. Probleme und Änderungen der Anforderungen werden mit Durchführenden, Zeitpunkt und Inhalt durch System dokumentiert (alle Beteiligten).
- Entwickler erstellt auf eigener Entwicklungsumgebung in Verbindung mit Testdatenbanken die Lösung für die Anforderung.
- Entwicklungsumgebungen werden zu regelmäßigen Zeitpunkten mit dem Livesystem synchronisiert (Live zu Test – manuell täglich und automatisch wöchentlich).
- Die Lösungen werden mithilfe von festgelegten und definierten Frameworks, Libraries und Softwares entwickelt.
- Entwickler testet gegen standardisierte Testfälle und gegen sicherheitsrelevante Tests die betreffenden Hauptfunktionalitäten der beeinflussten Umgebungsfunktionen.
- Nach Meldung der Fertigstellung über das Ticketsystem wird der Test auf der Entwicklertestumgebung von einem weiteren Entwickler bzw. von dem Ticketersteller durchgeführt (Vier-Augen-Prinzip).
- Nach evtl. Korrekturen und Retests wird nach Freigabe die Lösung auf das Testsystem commited und erneut getestet (Ticketersteller). Hier folgt ein abschließender Codereview durch den führenden Entwickler bzw. einem Seniorentwickler. Angrenzende Softwarefunktionalitäten werden ebenfalls getestet.
- Nach Freigabe wird die Lösung für das abschließenden Deployment zur Liveumgebung eingestellt.
- Zu regelmäßig festgelegten Zeitpunkten finden Standarddeployments statt. Außerhalb dieses Zeitraumes sind nur Hotfixdeployments erlaubt, die kritischen Zustände auf Livesystemen beheben.
- Nach dem Deployment zum Livesystem wird die Lösung von den Beteiligten sofort getestet, um evtl. Probleme auf dem Livesystem frühzeitig zu erkennen. (Ticket/Anforderungsersteller, Entwickler)

1.2.2 Programmierrichtlinien

Einheitliche Programmierrichtlinien, Programmierstile und Technologien ermöglichen einheitliche Sicherheitsmechanismen.

Die Richtlinien werden durch den führenden Entwickler im Entwicklungsprozess (siehe Vorgehensmodell) regelmäßig durch Codereviews sichergestellt. Diese Richtlinien werden in regelmäßigen Abständen hinterfragt und diskutiert und bei grundlegenden Bedarf (Softwareweiternetwicklung, Prozessverbesserungen, Systemaktualisierungen etc.) angepasst.

1.2.3 Testfälle und Testdaten

Durch sorgfältige Spezifikation von Testfällen und Testdaten sollen alle denkbaren Anwendungsfälle abgedeckt werden, damit vorhandene Fehler schnell entdeckt und behoben werden können.

1.2.4 Barrierefreiheit

Angestrebt wird ein Wert von mindestens 90 Punkten im BITV Test – in jedem Fall in den öffentlich zugänglichen und durch die Bewerber benutzten Bereichen.

1.3 Schutz personenbezogener Daten

Im Integrationsportal JobIMPULS wird das Benutzerverhalten nicht ausgewertet, weder durch User Tracking noch durch User Profiling. SessionIDs und Cookies dienen ausschließlich der Authentifizierung.

1.3.1 Verwendung von Cookies

Informationen zu Seitenaufrufen und Eingaben werden nicht konkreten Benutzern zugeordnet und nicht über einen längeren Zeitraum protokolliert. Personenprofile werden nicht erstellt.

1.3.2 Nachladen fremder Inhalte

Im Prozess des Aufrufens von Stellenanzeigen von fremden Server werden Layout und Styles- Dateien von Servern der jeweiligen Kunden geladen. Dabei werden Bilder, sowie teilweise Styles-Dateien von Servern der jeweiligen Quelle geladen.

Anhand der angeforderten Bilder und Tracker können die Betreiber der anderen Server Abrufstatistiken ihrer Webseiten führen. Nutzerprofile können damit nicht erstellt werden. Es werden lediglich gezielt beispielsweise bestimmte Stellenanzeigen aufgerufen, die Nutzer nicht auf die Seiten Dritter navigiert.

Bei der Umsetzung kundenspezifischer Layouts innerhalb unserer Software werden Styles und Bilder auf eigene Server übernommen und vorhandene Tracker, sowie ausführbare Scripts gelöscht. Alle externen Datenabrufe werden damit unterbunden und bleiben in der Kontrolle von Jobcenter Consulting.

1.3.3 Verwendung von Javascript

In den HTML-Seiten der Webanwendung ist kein Javascript-Code eingebettet, der zur Erstellung von Benutzerprofilen verwendet wird oder werden könnte. Javascript selbst kommt an vielen Stellen der Webanwendung zum Einsatz.

1.3.4 Speicherung personenbezogener Daten

Personenbezogene Daten werden nur rechtmäßig erhoben und angemessen gespeichert. Von Dritten können sie nicht unbefugt ausgelesen werden. Es gibt beispielsweise keine Möglichkeit, Kontaktdatenfelder für Dritte freizuschalten. Die Bewerberprofile sind stets anonymisiert und hochgeladene Dateien vor unbefugtem Zugriff geschützt. Hierzu befinden sich schutzbedürftige Dateien in einer Datenbankstruktur und können nur mit spezieller Authentisierung aufgerufen werden.

1.4 Vergabe von Zugriffsrechten

Über Zugriffsrechte wird geregelt, welche Person im Rahmen ihrer Funktion bevollmächtigt wird, die Anwendungen oder Daten zu nutzen. Die Zugriffsrechte (z. B. lesen, schreiben, ausführen) auf das System, Teilsysteme oder Daten sind von der Funktion abhängig, die die Person wahrnimmt.

Die Rollen sind unterteilt in Administration, Seniorentwickler, Entwickler und Support.

Auf das Livesystem (Server, Datenbanken, Filesystem), haben nur Administration und in Vertretung der Seniorentwickler schreibend und lesend Zugriff. Auf Testsystem haben Administration, Seniorentwickler, Entwickler schreibend und lesend Zugriff, der Support nur lesend.

Kundenspezifisch sind die Zugriffsrechte zum Teil über das Partner-Admintool steuerbar, zum Teil nehmen wir sie nach entsprechendem Auftrag des Auftraggebers über die Datenbank vor. Hierfür sind diverse Rollen definiert, auf die konkrete Zuordnung haben wir aber keinen Einfluss.

Aktuelle Rollen sind insbesondere:

- Masteruser/ Administrator
- Führungskraft
- Fallmanager
- Arbeitgeberservice
- Maßnahmenmanager
- Maßnahmenbetreuer
- beauftragter Dritter
- Vertreter

Einem Vermittlerkonto können mehrere Rollen zugeordnet werden.

2 Risiken technischen Versagens

2.1 Authentisierung

Authentisierungsmechanismen werden in allen geschützten Bereichen der Webanwendung eingesetzt. Hierdurch soll ausgeschlossen werden, dass

- Unbefugte Zugriff auf IT-Systeme oder Daten nehmen können,
- Verursacher von Problemen nicht identifiziert werden können oder
- die Herkunft von Daten nicht bestimmt werden kann.

Systemseitig werden bei der Erstellung von Konten, sichere Passwörter vergeben.

Sicherheitsrisiken für Benutzerkonten können entstehen bei Benutzern, die Passwörter wählen, die einfach zu erraten sind. Systemseitig werden deshalb nur Passwörter zugelassen, die den Vorgaben sicherer Passwörter entsprechen. Hierbei wird sichergestellt, dass die Passwortmindestlänge acht Zeichen beträgt und mindestens ein Großbuchstabe sowie mindestens eine Zahl verwendet wird.

Eine regelmäßige erzwungene Änderung des Passwortes ist noch nicht umgesetzt aber bereits in Planung.

2.2 Validierung von Ein- und Ausgabedaten

2.2.1 Validierung von Eingabedaten

Web-Anwendungen werden im Allgemeinen von generischen Clients (Web-Browsern) verwendet, sodass Benutzer beliebige Eingabedaten an den Server übermitteln können. Werden schadhafte Eingaben eines Angreifers von der Web-Anwendung verarbeitet, können möglicherweise Schutzmechanismen der Web-Anwendung umgangen werden. Vor Freischaltung neuer Programmteile werden diese auf SQL Injection, Path Traversal und Remote File Inclusion getestet.

Durch softwareseitig standardisierte Überprüfungsmechanismen werden Formulareingaben bereits vor dem Absenden an den Server validiert. Hier werden auf Länge der Eingaben Art der Eingabe (numerisch, alphanumerisch, Sonderzeichen) je nach Eingabefall geprüft.

Zur Eingabe von Daten werden softwareseitig auch sogenannte WYSIWYG-Editoren benutzt, die im Standardeinsatz auch die Übermittlung von Scriptanweisungen erlauben würden, die bei der erneuten Ausgabe als ausführbares Script interpretiert werden könnten.

Bereits bei der Eingabe werden HTML-Standardformatierungen in eine eigene Sprache (eigenes Markup) übersetzt und beim Speichern der Daten wird auf wünschenswerte z. B. Farbformatierungen, Linkaufrufe etc. und nicht wünschenswerte Eingaben wie z. B. Scripts hin überprüft und durch anschließenden Ersetzungs-Routinen geändert. Dadurch wird gewährleistet, dass kein Schadcode in der Datenbank gespeichert und später ausgegeben werden kann.

2.2.2 Validierung von Ausgabedaten

Die Daten werden vor der Ausgabe ausreichend validiert und können keinen Schadcode enthalten. Bereits bei der Eingabe der Daten sind diese auf ihren Anwendungsfall hin überprüft. Beispielsweise werden bei Datumsangaben nur numerische Daten und bei E-Mail-Eingaben auf Länge und Bestandteile der E-Mail hin überprüft. Bei der Eingabe der Daten, z.B. bei Textblöcken werden diese durch Ersetzungs-Routinen auf ungewollten Codebestandteile hin überprüft. Entsprechenden Codebestandteile werden dann in nicht-ausführbaren Textzeichen ausgegeben, sodass eine Interpretation als Script bzw. Anwendung nicht möglich ist.

2.2.3 SQL-Injection

SQL Injection ist das Einbetten einer nicht autorisierten Suchanfrage in die Datenbankabfrage. Durch entsprechende Tests wird sichergestellt, dass derartige Befehle von der Datenbank nicht ausgeführt werden.

Wie bereits beschrieben werden Daten auf Plausibilität überprüft und Textblöcke auf Schadcodes gecheckt. Anschließend werden dann datenbankseitig Spezialzeichen durch Maskierungen entwertet. Es wird ausnahmslos jeder an die Datenbank übergebenen Parameter geprüft.

2.2.4 Datei-Upload

Der Datei-Upload ist auf gewisse Dateitypen beschränkt. Diese Beschränkung obliegt der Plausibilität an der entsprechenden Funktion. Beispielsweise werden beim Profildatenupload nur Dateien mit .docx, .doc, .pdf, .tiff, .jpg, und .png Endungen zugelassen.

Zur Bestimmung des Dateityps wird ebenfalls der Inhalt der Datei herangezogen.

2.2.5 Sanitizing

Durch entsprechende Validierungsroutinen werden Nutzereingaben vor der Weiterverarbeitung von eventuellem Schadcode bereinigt. Hier sei verwiesen auf die Verwendung von eigenem Markup bei WYSIWYG-Editoren und Maskierungen von Sonderzeichen, sowie auf die von Python und PHP verwendeten Filter-Routinen, um Eingaben mit Sonderzeichen entsprechend zu Maskieren, damit der Code neutralisiert wird.

2.2.6 Kodierungsschema

Es wird einheitlich UTF-8 verwendet, Layoutvorlagen von Kunden werden bei Bedarf umkodiert. Damit ist sichergestellt, dass Nutzereingaben bei der Filterung einheitlich erkannt werden.

2.2.7 Kommentar- und Texteingabefunktion

Die Kommentar- und Textfunktion einer Webanwendung erlaubt eine Formatierung der Texte durch HTML. Diese Eingaben werden eingeschränkt. Die Beschränkung wird auf nur gewünschte HTML-Auszeichnungen zugelassen (sogenanntes Whitelistung) und anschließen in eigene Markup umgewandelt bzw. wo diese nicht notwendig sind, ganz deaktiviert.

2.3 Fehlerbehandlung

Alle Fehlermeldungen werden automatisch in Error Logs gespeichert und jede Stunde an einen vordefinierten Personenkreis zur sofortigen Überprüfung und Behebung gesandt. Dadurch soll ausgeschlossen werden, dass Fehler unerkannt bleiben.

Aufgezeichnet werden Fehler, die bei der serverseitigen Verarbeitung von Daten oder Aufrufen auftreten. Eingabefehler die bereits durch Überprüfungen im Browser auftreten werden nicht geloggt. Beispielsweise werden Timeouts, Exceptions, Zugriffe auf nicht vorhandene Dateien geloggt. Dabei werden u. a. Zeit, Modul, Fehlermeldung, Zieldatei und Eingabe-URL, SourceIP aufgezeichnet, um die Fehlerquelle genau zu lokalisieren.

Zum Personenkreis derer, die eine Fehlermeldungen erhalten, gehören jeweils zwei Mitglieder des Entwicklungsteams: Teamleiter Entwicklung und Seniorentwickler, sowie zwei Mitglieder des Supportteams. Diese werden gleichzeitig benachrichtigt, wenn Fehlermeldungen auftreten und prüfen dann unabhängig die Fehlermeldungen. So wird durch ein Mehraugenprinzip gewährleistet, dass keine Fehlermeldung übersehen bzw. unterschätzt wird. Die Urlaubsregelung ist dementsprechend festgelegt, dass jeweils zwei Personen die Möglichkeiten haben, die Fehlermeldungen zu überprüfen.

Der Prozess der Fehlerbehebung wird analog zum Entwicklungsprozess durchgeführt. Hierbei wird ein fünf-stufiges Priorisierungslevel eingesetzt, wobei die oberste Stufe die sofortige Behebung des Problems anstößt und nach Lösungsbestätigung einen Hotfix bewirkt.

Errorlogs werden zusätzlich serverseitig abgelegt und können nur durch den Administrator über eine sichere Verbindung eingesehen werden. Durch das Berechtigungssystem ist es ausgeschlossen, dass Errorlogs über den Browser aufgerufen werden können.

Errorlogs werden spätestens nach vier Wochen automatisch gelöscht.

2.3.1 Ressourcenfreigabe im Fehlerfall

Durch eine Vielzahl von Maßnahmen, u. a. Echtzeitdatenreplikation mit DRBD, automatische Kill- und Restartskripte u. ä. soll sichergestellt werden, dass bei Fehlern betroffene Ressourcen nicht blockieren. Mehrere derartige Skripte sind u. a. im Servermonitorsystem Nagios hinterlegt.

2.3.2 Sichere Fehlerbehandlung

Treten während der Ausführung der Sicherheitskomponenten (z. B. Autorisierung, Authentisierung) Fehler auf, kann die aufgerufene Aktion nicht ausgeführt werden. In keinem Fall können Fehler dazu führen, dass nicht autorisierte Funktionen zugelassen werden und unbefugt auf Ressourcen zugegriffen werden kann.

2.3.3 Keine detaillierten Fehlermeldungen für Nutzer

Weder in den PHP- noch in den Python-Dateien werden Fehlermeldungen an den Nutzer ausgegeben, die detaillierte Hinweise zur Fehlerursache beinhalten. Bei Fehlern passiert aus Nutzersicht entweder gar nichts (Anwendung friert ein), oder es wird eine speziell programmierte allgemeine Fehlermeldung ausgegeben.

2.3.4 Keine clientseitige Fehlerbehandlung

Fehlerbehandlungen werden nicht auf dem Client durchgeführt, sondern nur auf dem jeweiligen Server.

2.4 Nachvollziehbarkeit sicherheitsrelevanter Ereignisse

Sicherheitsrelevante Ereignisse, z. B. erfolglose Loginversuche, werden protokolliert und stündlich an den vordefinierten Personenkreis (siehe Fehlerbehandlung) versandt, damit sie bei Bedarf schnell nachvollzogen werden können. Somit können Ursachen rasch ermittelt werden und es bleiben keine Fehler unbemerkt.

Ereignisse auf der System- und Netzebene werden entsprechend protokolliert.

2.4.1 Abdeckung zu protokollierender Ereignisse

Sicherheitsrelevante Ereignisse der Webanwendung werden angemessen protokolliert. Konfigurationsänderungen würden schnell entdeckt werden.

Ereignisse können wie folgt definiert werden:

- Anmeldungen per SSH
- Mehrfach Anmeldeversuche
- wenn Dateien geändert werden
- wenn auf Dateisysteme von außen und nicht innerhalb von Funktionsaufrufen zugegriffen wird
- Erfolglose Zugriffe auf nicht vorhandenen Dateien
- Unbekannte Parameterübergaben und noch nicht gefilterte ungültige Eingaben

2.4.2 Präzise Protokollierung von Ereignissen

Alle notwendigen Eigenschaften eines Ereignisses werden protokolliert, z. B. nutzerseitig mit Datum, Uhrzeit, IP-Adresse und der Datei, von der aus auf die fehlerhafte Datei zugegriffen wurde (siehe auch Fehlerbehandlung).

2.4.3 Zugriffsschutz

Der Schutz der Protokolldaten ist gewährleistet. Siehe oben.

2.4.4 Datenschutzaspekte bei der Protokollierung

2.4.4.1 Systemgenerierung und Modifikation von Systemparametern

Derartige Änderungen werden in geeigneter knapper Weise manuell im Projektmanagementsystem dokumentiert.

2.4.4.2 Einrichten von Benutzern

Änderungen werden nicht gesondert dokumentiert, da die benötigten Informationen aus Backups bei Bedarf schnell gewonnen werden können und nur Masteruser/Administratoren derartige Änderungen vornehmen.

2.4.4.3 Erstellung von Rechteprofilen

Siehe die vorherigen Ausführungen. Da Rollen durch den Auftraggeber zugewiesen und über das Partner-Admintool selbst gesteuert werden können, erfolgt keine parallele Dokumentation durch uns als IT-Dienstleister.

2.4.4.4 Einspielen und Änderung von Anwendungssoftware

Siehe die vorherigen Ausführungen. Das erfolgt automatisch in der Kombination aus Redmine und Git.

2.4.4.5 Änderungen an der Dateiorganisation

Auch die Dateiorganisation wird über Git gemanagt und dokumentiert.

2.4.4.6 Durchführung von Datensicherungsmaßnahmen

Es existiert ein eigener Backupserver mit einer Vielzahl regelmäßiger Datensicherungen gemäß Backupkonzept. Dabei gibt es inkrementelle Backups und Vollbackups. Die entsprechenden Protokolldaten können bei Bedarf direkt vom Backupserver gewonnen werden und werden nicht zusätzlich extern dokumentiert.

2.4.4.7 Aufruf von Administrationstools

Es wird das Login als solches protokolliert, aber keine weiteren identifizierenden Daten etwa z. B. IP-Adresse, Logindauer o. ä. Dies kann grundsätzlich ausgebaut werden.

2.4.4.8 Versuche unbefugten Einloggens und Überschreitung von Befugnissen

Unter anderem werden fehlgeschlagene Loginversuche protokolliert, z. B. als „password mismatch".

2.4.4.9 Eingabekontrolle

An zahlreichen Stellen wird mit sekundengenauem Zeitstempel unter Angabe des Benutzers protokolliert, wer welche Eingaben gemacht hat. Eine Aufzählung würde den Rahmen eines IT-Sicherheitskonzepts sprengen, beispielhaft seien alle Gesprächsdokumentationen, Fallmanagementeinträge, Wiedervorlagen,

Vermittlungsvorschläge, Stellenspeicherungen, Bewerbungs- und Vermittlungsaktivitäten sowie Nachrichten genannt.

2.4.4.10 Datenübermittlungen

Datenübermittlungen an Dritte finden nicht statt, auch nicht an das SGB II Fachverfahren. Datenimporte aus dem SGB II Fachverfahren werden mit Zeitstempel, Datenart, Dateinamen, Dateigröße und Bemerkungen protokolliert. Ein Datenimport kann stets erst nach dem Upload durch das Jobcenter erfolgen.

2.4.4.11 Abrufverfahren

Die, im Rahmen der Zugangsrechte durch den Mitarbeiter des Jobcenters oder eines beauftragten Dritten, erfolgende generelle Arbeit mit dem System wird nicht protokolliert. Das wäre eine unzulässige Verhaltenskontrolle. Das System ordnungsgemäß zu nutzen, liegt in der Verantwortung des Nutzers, nicht des IT-Dienstleisters.

2.4.4.12 Löschung von Daten

Nicht jede Veränderung von Feldern, die relevante personenbezogene Daten beinhaltet, wird protokolliert. Das Löschen von Bewerberkonten und Stellenangeboten wird mit einem Zeitstempel protokolliert, ebenso automatisierte Löschungen, z. B. durch den so genannten “garbage collector”.

2.4.4.13 Aufruf von Programmen

Besonders „sensible” Programme, die nur ausnahmsweise zu bestimmten Anlässen durch einzelne Benutzer aufgerufen würden, sind derzeit nicht erkennbar.

Protokolldaten werden mit den Backups maximal ein Jahr aufbewahrt und dann automatisch gelöscht.

2.5 Keine Offenlegung von Informationen

Webseiten und Daten enthalten keine vertraulichen Informationen, die nicht für die Nutzung der Webanwendung erforderlich sind.

2.5.1 Keine Informationen zu Sicherheitsmechanismen und -attributen

Es werden keine detaillierten Informationen über Sicherheitsmechanismen oder -attribute ausgegeben. Passwörter haben beispielsweise weder ein striktes vorgegebenes Format, (Mindesteigenschaften müssen eingehalten werden) noch eine bestimmte Länge; es wird lediglich die Mindestlänge vorgegeben.

2.5.2 Keine vertraulichen Informationen in Kommentaren

Kommentare (z. B. im HTML-Quelltext) enthalten keine Informationen zu bekannten Fehlern, Funktionsweisen, eingesetzten Techniken oder der angebundenen Infrastruktur. Dies wird vor dem Release durch den Chefprogrammierer überprüft.

2.5.3 Unbekannte Dateiendungen

Es werden keine Dateien mit unbekannter Dateiendung (z. B. temporäre Dateien mit .tmp oder Backup-Dateien mit .bak von Skripten der Webanwendung) im Quelltext ausgeliefert.

3 Vorsätzliche Handlungen

3.1 SQL Injection

Alle neuen Programmteile werden vor dem Release stets auf SQL Injection überprüft. Dadurch wird ausgeschlossen, dass die Datenbank andere als die vorgesehenen Befehle ausführt. Ein unberechtigter Zugriff, die eine Manipulation von Daten, das Ausführen von Betriebssystembefehlen, die Kontrolle über die Datenbank oder der Zugriff auf weitere Server, ist ausgeschlossen.

Die Eingabedaten werden entsprechend validiert und gefiltert (siehe bereits ausgeführte Detailinformationen weiter oben).

3.2 Nur berechtigter Zugriff auf Daten

Die ausgelieferten Dateien enthalten no-cache und no-proxy Informationen. Auch werden Browsereinstellungen des Nutzers, Kennwörter zu speichern und automatisch einzugeben, übersteuert.

3.2.1 Zugriff auf den Browser-Cache

Zugangsdaten werden nicht im Browsercache gespeichert, sondern lediglich verschlüsselt in Cookies, die u. a. nach 30 Minuten Inaktivität verfallen. Damit ist ein Systemzugang durch einen Unbefugten auch in dem Fall nicht möglich, wenn dieser Zugang zum Rechner eines Nutzers erhält und welcher sich nicht zuvor ausgeloggt hat.

3.2.2 Verwendung von GETParametern

Schützenswerte Daten werden nicht in GET-Parametern übermittelt und können nicht über die URL ausgelesen werden. Beispielsweise werden keine SessionID's, keine personenbezogenen Daten, keine Eingabedaten, die über Formulare vorgenommen werden, in der URL übermittelt. Es werden lediglich Portal-ID, User-ID und bei verschiedenen Modulen Attribute zur Tabellensortierung übergeben.

3.2.3 Verwendung von Cookies

Sitzungsdaten sind in Cookies gespeichert. Die Sitzung gilt nur für eine bestimmte IP-Adresse zum Zeitpunkt des Logins. Ändert sich die IP-Adresse, „verfällt" das Cookie wie bei 30-minütiger Inaktivität. Infolgedessen besteht kein Risiko durch einen potentiellen Angreifer, den Inhalt von Cookies unbefugt auszulesen und zu versenden.

3.3 Schutz gegen automatisierte Nutzung

Typische automatisierte Nutzungen, wie z. B. sehr schnell wiederholte erfolglose Login-Versuche („brute force") oder Denial-of-Service-Angriffe, führen zum automatischen Blockieren der entsprechenden IP-Adressen.

3.3.1 Automatisiertes Sammeln (Enumeration) von Benutzerdaten

Informationen über registrierte Benutzer können nicht über eine URL abgerufen werden. In Bewerberprofilen werden keine E-Mail-Adressen oder sonstige Kontaktdaten veröffentlicht.

3.3.2 Kein Provozieren von Kontensperrungen

Benutzerkonten werden nicht nach wenigen fehlgeschlagenen Anmeldeversuchen gesperrt, um Brute-Force-Angriffe zu erschweren. Bei Brute-Force-Angriffen werden vielmehr die beteiligten IP-Adressen der Angreifer blockiert. Die Benutzerkonten können von den berechtigten Nutzern also auch dann genutzt werden, wenn Angreifer zu ihrem Benutzernamen mit Brute-Force Passwörter „ausprobieren".

3.4 Webanwendungslogik

Ein Umgehen der Sicherheitsmechanismen, z. B. Authentisierung durch bestimmte logische Abläufe ist ausgeschlossen. Jedes Modul verwendet den zentralen Authentisierungsmechanismus.

3.4.1 Feldlängen und Filtern

Bei der Validierung von Eingaben werden zunächst vorgegebene Feldlängen berücksichtigt, bevor die Daten weiter gefiltert werden (siehe Angaben weiter oben).

3.5 Keine rein clientseitig umgesetzten Sicherheitsfunktionen

Es gibt keine ausschließlich clientseitig umgesetzten Schutzmaßnahmen. Die entscheidenden Schutzmaßnahmen sind immer serverseitig.

3.5.1 Validierung von Daten

Die Eingabevalidierung ist serverseitig und z. B. hinsichtlich der Vollständigkeit bei Pflichtfeldern clientseitig mit JavaScript umgesetzt. Ist die JavaScript-Unterstützung auf dem Client deaktiviert, wird das betreffende Formular nicht abgesandt und die Eingaben werden nicht verarbeitet.

3.5.2 Keine clientseitig gesetzten Parameter zur Authentisierung

Die Berechtigungen sind in den entsprechenden Benutzerkonten gespeichert, nicht aber etwa durch clientseitig gesetzte Parameter.

3.6 Session-Management

Unberechtigte Dritte können SessionIDs nicht ermitteln und sie wegen der Kombination mit der IP-Adresse und dem automatischen Verfall nach 30 Minuten Inaktivität auch nicht verwenden. Die Funktionalität der Webanwendung kann somit von Dritten nicht mit den Rechten des legitimen Benutzers genutzt werden.

3.6.1 Session-Fixation

Bei einem Session-Fixation-Angriff lasst sich ein Angreifer zunächst eine SessionID von der Web-Anwendung zuweisen und übermittelt diese dem Opfer (z. B. über einen Link in einer E-Mail). Folgt das Opfer diesem Link und authentisiert sich anschließend gegenüber der Web-Anwendung, mit der vom Angreifer übermittelten SessionID, so kann der Angreifer die Anwendung anschließend mit der ihm bekannten SessionID verwenden.

In der JobIMPULStechnologie werden SessionIDs nicht per URL übergeben und sind nicht außerhalb des Rechners des Nutzers verwendbar.

3.6.2 Erraten von gültigen SessionIDs

Das Erraten von SessionIDs, z. B. durch simples Hochaddieren, ist nicht möglich. Die SessionIDs werden durch auf den Applicationserver enthaltenden Session-Generator über ein standardisiertes Verfahren (PHP, Python) vergeben.

3.6.3 Session Timeout

Sitzungen inaktiver Benutzer werden nach zwei Stunden automatisch beendet. Die Sessions werden invalidiert.

3.7 Cross-Site Scripting (XSS)

Beim Cross-Site Scripting (XSS-Angriffe) versucht ein Angreifer indirekt Schadcode an den Browser des Benutzers zu senden. Da die Ein- und Ausgaben validiert werden, können potenzielle Angreifer schadhafte Codes nicht in die Webanwendung einschleusen und verteilen.

3.8 Cross-Site Request Forgery (CSRF, XSRF)

Funktionen der Webanwendung können nicht ohne weitere Überprüfung der Authentizität des HTTP-Requests genutzt werden. Es wird bei jedem Aufruf überprüft ob dieser von JCC-Servern stammt. Ausnahmen sind Stellenanzeigen von externen Quellen.

Die Anmeldung der Benutzer erfolgt immer manuell.

3.9 Keine Umgehung der Autorisierung

Bei Angriffen gegen die Autorisierungskomponente einer Webanwendung wird versucht, auf Funktionen oder Daten zuzugreifen, die nur einer eingeschränkten Benutzergruppe zur Verfügung stehen.

3.9.1 Null-Byte

Das Null-Byte wird von der Filterkomponente der Webanwendung und den Hintergrundsystemen einheitlich interpretiert. Dieses sowie nachfolgenden Zeichen werden nicht ignoriert.

3.9.2 Path Traversal

Bei der Eingabe von Pfadangaben für den Zugriff über die Webanwendung werden nicht vorgesehene Ressourcen nicht abgerufen. Ordner sind gegen Direktaufrufe geschützt, offenbaren also nicht ihren Inhalt.

3.9.3 Objekt-Referenzen

Die Webanwendung verwendet an verschiedenen Stellen Objekt-Referenzen zur Adressierung von Datenbankeinträgen. Diese werden von der Autorisierungskomponente immer berücksichtigt, sofern es um schutzwürdige Daten geht und nicht z. B. bloß um Hilfetexte des Content Management Systems.

3.10 Keine Einbindung von fremden Daten und Schadcode

Durch Validierung der Ein- und Ausgabedaten ist gewährleistet, dass keine Inhalte, wie z. B. Schadcodes, eingebunden werden können.

3.10.1 Unbefugte Weiterleitung

Es gibt keine Weiterleitungsfunktionen, die beliebige Werte als Zieladresse akzeptieren würden.

3.10.2 Inhalte von Partnern

Für die eingestellten Inhalte sind die Lizenzkunden selbst verantwortlich. Schadsoftware können Nutzer der Lizenzkunden durch die bereits erwähnten Überprüfungen nicht einbringen.

3.10.3 Kein Missbrauch von Upload-Funktionen

Die Upload-Funktionen sind so realisiert, dass nicht einfach beliebige Dateien in eine Verzeichnisstruktur auf dem Server gespeichert werden können. Mit Ausnahme einfacher Logo-Dateien und Bewerbungsfotos (je eins pro Konto im vordefinierten Format mit Größenbeschränkung), werden hochgeladene Dateien nicht in Verzeichnisstrukturen, sondern in Datenbanken gespeichert und durchlaufen zuvor entsprechende Prüfungen.

Die Uploadfunktionen ist nur im angemeldetem Zustand nutzbar. Die Dateibeschränkungen entsprechen den dazu verwendeten Kontexten und Funktionen und werden überprüft. Zur Überprüfung wird nicht nur die Dateiendung, sondern ebenfalls der Inhalt verwendet. Die Dateiverzeichnisstrukturen sind nicht einsehbar und werden durch automatische Scripte erzeugt und in der Datenbank hinterlegt.

3.11 Injection-Angriffe

Bei einem Injection-Angriff versucht ein Angreifer, Befehle in die Web-Anwendung zu injizieren und auszuführen (vgl. hierzu die bereits oben dargestellten Schutzmaßnahmen).

3.12 Clickjacking

Bei einem Clickjacking-Angriff werden Teile einer Webseite bei der Darstellung überdeckt, sodass für den Benutzer nicht sichtbare, transparente Ebenen die angezeigten Inhalte der Webseite überlagern.

In diesen transparenten Ebenen können beliebige Inhalte oder Bedienelemente eingebunden werden, ohne dass sie für den Benutzer sichtbar sind. Klickt der Benutzer auf die vermeintlichen Inhalte der Webseite, so wird der Klick nicht an den sichtbaren Inhalt, sondern an die überlagerten Ebenen gesendet und somit entführt.

Für Clickjacking geeignete Anwendungen sind für das Integrationsportal JobIMPULS nicht ersichtlich.

4 Planung und Konzeption

4.1 Dokumentation der Architektur

Alle Abhängigkeiten und Schnittstellen sind dokumentiert. Für den Betrieb notwendige Komponenten, die nicht Bestandteil der Webanwendung sind, werden als solche gekennzeichnet, vgl. hierzu auch die graphische Darstellung der Serverarchitektur. Aus der Dokumentation geht hervor, welche Komponenten Sicherheitsmechanismen umsetzen.

Dies betrifft die folgenden Bereiche:

- Benutzermanagement,
- Authentisierung,

- Autorisierung,
- Session-Management,
- Protokollierung und
- Transportsicherheit.

Detaillierte Angaben hierzu betreffen unmittelbar schutzbedürftige Geschäftsgeheimnisse. Selbstverständlich existieren für interne Zwecke geeignete Dokumentationen, u. a. mit einem Wiki für das Entwicklerteam. Diese Informationen sind jedoch nur für den internen Gebrauch bestimmt.

4.2 Entwicklung und Erweiterung von Anwendungen

Für die Entwicklung sind Regeln festgelegt. Es wird nach einem festgelegten Vorgehensmodell entwickelt, getestet und freigeschaltet.

4.2.1 Historie von Änderungen am Quelltext

Die Historie der Änderungen wird in einem CVS festgehalten.

4.2.2 Testen von Sicherheitsfunktionalität

Testfälle berücksichtigen immer auch die Sicherheitsfunktionalität. Dazu zählen Autorisierung-, Authentisierungs- und Filterkomponenten.

Es werden toolgestützte und manuelle Penetrationstests durchgeführt. Dabei werden z. B. automatisierte Webseitenaufrufe durch eine Software durchgeführt und Formulareingaben mit beliebigen Zeichenfolgen und absichtlich nicht korrekten Eingaben erzeugt.

Die Tests werden regelmäßig und zusätzlich einmal im Jahr vom Head of Development durchgeführt. Das Ergebnis wird zusammen mit der Geschäftsführung ausgewertet und bei Bedarf werden Gegenmaßnahmen ergriffen.

4.2.3 Vier-Augen-Prinzip bei Tests der Anwendungslogik

Vor der Inbetriebnahme werden Tests durch den Chefprogrammierer, der die jeweilige Anwendung nicht selbst programmiert hat, und einen weiteren Tester durchgeführt. Zunächst im Testsystem und anschließend im Testkonto des Livesystems.

4.2.4 Konformität zum HTML Standard

Bei Neuentwicklungen werden die Webseiten auf Konformität zu dem verwendeten Standard getestet.

4.2.5 Geschützter Zugriff auf Hintergrundsysteme

Der Zugriff auf die Hintergrundsysteme ist lediglich über die definierten Schnittstellen der Anwendung möglich.

4.2.6 Schutz vor unbefugter Manipulation der Daten

Die Daten können bei der Verteilung der Anwendung nicht durch Dritte manipuliert werden.

4.2.7 Sicherheitsprüfung der Anwendung

Es werden regelmäßige Sicherheitsprüfungen für bestehende Komponenten durchgeführt und die zugrunde liegenden Komponenten, etwa Betriebssystem, Datenbanksystem etc., zeitnah aktualisiert.

Erforderliche Aktualisierungen, Patches, Softwareupdates werden regelmäßig, wie vom Hersteller ausgeliefert, durchgeführt. Sicherheitskritische Updates werden sofort nach Bekanntwerden und Bereitstellung einer neuen Version durchgeführt. Informationen werden direkt beim Hersteller der Software abgefragt (Newsletter, Blog)

4.2.8 Tests nach Änderungen an der Anwendung

Tests erfolgen nicht nur isoliert für eine neue Funktion, sondern im Kontext ihrer Anwendung.

4.2.9 Getrennte Produktiv-, Test- und Entwicklungsumgebung

Die Produktiv-, Test- und Entwicklungsumgebungen werden auf getrennten Systemen betrieben.

4.3 Web-Tracking

Es erfolgt kein Web-Tracking.

4.4 Sicherer Entwurf der Logik von Web-Anwendungen

Die Anforderungen an die abgebildete Logik werden exakt erfasst und korrekt umgesetzt, sodass ausschließlich beabsichtigte und vorgesehene Aktionen durchgeführt werden können.

4.4.1 Dokumentation von Anwendungsfällen

Bei der Konzeption werden alle Funktionen anhand von Anwendungsfällen dokumentiert.

4.4.2 Aktive Inhalte

Die Webanwendung verwendet an vielen Stellen Javascript, häufig zur Navigation und für Ajax-Elemente. Dies entspricht den Erwartungen der Nutzer an eine moderne Webanwendung. Mit Standardeinstellungen ist die Nutzung der Webanwendung für die jeweils letzten beiden Browserversionen des Internet Explorer und Mozilla Firefox garantiert.

4.5 Sichere Anbindung von Hintergrundsystemen

Die Hintergrundsysteme, etwa Datenbanken, sind sicher an die Webanwendung angebunden und nur über private IP-Adressen zugänglich.

4.5.1 Keine direkten Zugriffe auf Hintergrundsysteme

Die Benutzer der Webanwendung können nicht direkt auf die Hintergrundsysteme zugreifen. Der Zugriff ist ausschließlich über vordefinierte Schnittstellen und Funktionen der Webanwendung möglich.

4.5.2 Geschützte Verbindung

Die Datenübertragung erfolgt verschlüsselt. Die Verbindung erfolgt über ein https-Verbindung, die das OpenSSL-Protokoll verwendet. Hier liegt eine 2048bit-SSL-Verschlüsselung zugrunde.

4.5.3 Zugriff über einen technischen Benutzer

Es werden mehrere Dienstkonten mit unterschiedlichen Zugriffsrechten für die jeweiligen Hintergrundsysteme verwendet. Dienstkonten werden nicht von Administratoren verwendet.

4.5.4 Zugriff über den authentisierten Webanwendungs-Benutzer

Nutzer können nicht direkt auf Hintergrundsysteme zugreifen, sondern nur indirekt im Rahmen ihrer jeweiligen Rechte über die Webanwendung.

4.6 Systemarchitektur

Es wird eine mehrschichtige Netzwerkarchitektur (Multi-Tier) verwendet: Webschicht, Anwendungsschicht, Datenschicht, Hintergrundschicht. Die Datenschicht enthält auch dedizierte Server mit den bewerberbezogenen Daten der Kunden.

4.7 Dokumentation an Veränderungen des bestehenden Systems

Alle Systemänderungen werden intern automatisch in der Kombination aus dem Projektmanagementsystem Redmine und dem Versionsmanagementsystem Git dokumentiert. Git ermöglicht zum Beispiel auch einfache und schnelle „roll backs" von Veränderungen zurück zu einer vorherigen stabilen Version einer Datei, falls es doch einmal ein Fehler durch den Test-Workflow geschafft hat.

Auf das Projektmanagementsystem und Git gibt es redundante Zugriffsberechtigungen.

5 Umsetzung

5.1 Authentisierung

Alle Dateien der Webanwendung in den Benutzerkonten sind nur nach entsprechender Authentisierung zugänglich.

5.1.1 Zentrale Authentisierungs-Komponente

Die Authentisierungslogik ist nur an einer Stelle und nicht mehrfach im Programmcode realisiert.

Treten während der Authentisierung Fehler auf, wird die angeforderte Aktion abgebrochen und die Anfrage zurückgewiesen.

5.1.2 Erzwingen sicherer Passwörter

Bei systemseitiger Generierung von Benutzerkonten werden sichere Passwörter vergeben. Bei Erstellung von Passwörtern durch die Nutzer werden, wie bereits zuvor dokumentiert, Regeln zugrunde gelegt.

5.1.3 Anzeige der Stärke des gewählten Passworts

Die Stärke des gewählten Passworts wird beim Registrierungs- und Änderungsprozess angezeigt.

5.1.4 Keine automatische Authentisierung

Die im Browser mögliche automatische Authentisierung, wird systemseitig übersteuert.

5.1.5 Automatisiertes Ausfüllen von Formularfeldern

Die Option „autocomplete=off" ist für alle Formularfelder mit vertraulichen Daten gesetzt. Dadurch werden die Browser angewiesen, die Daten der entsprechenden Formularfelder nicht zu speichern.

5.1.6 Erneute Authentisierung bei sicherheitskritischen Funktionen

Bei sicherheitskritischen Aktionen (z. B. Änderung des Passworts) muss eine erneute Authentisierung des Benutzers (durch Eingabe des alten Passworts bei einem Passwortwechsel) erfolgen.

5.1.7 Einheitliches Sicherheitsniveau

Es gibt kein automatisiertes Zurücksetzen von Passwörtern. Besonders berechtigte Nutzer, insbesondere Masteruser, können jedoch neue Passwörter vergeben.

5.2 Umfassende Ein- und Ausgabevalidierung

Alle an die Webanwendung übergebenen Daten werden entsprechend gefiltert (vgl. dazu die Ausführungen weiter oben).

5.3 Session-Management

Zur Authentifizierung werden SessionIDs eingesetzt.

5.3.1 Eigenschaften von SessionIDs

Die Gültigkeitsdauer einer SessionID beträgt 30 Minuten nach Inaktivität. Die SessionID wird über einem innerhalb des Applicationserver integrierten Session-Generator erstellt. Es fließen keine extern bekannten oder erratbaren Daten in die Berechnung der SessionID ein.

5.3.2 Übertragungsmethoden für SessionIDs

Die SessionID wird in Cookies übertragen.

5.3.3 Verschlüsselte Übertragung von SessionIDs

Die SessionID wird wie alle Daten in Bewerberkonten verschlüsselt über https übertragen.

5.3.4 Manuelles Beenden von Sitzungen (Abmelden)

Auf allen Seiten der Webanwendung besteht in der "Topnavigation" eine deutlich sichtbare Abmeldemöglichkeit. Nach erfolgter Abmeldung ist die Sitzung vollständig beendet und verliert die SessionID ihre Gültigkeit.

5.3.5 Automatisiertes Beenden von Sitzungen (Session Timeout)

Das Session Timeout beträgt 30 Minuten nach Inaktivität.

5.3.6 Parallele Sitzungen eines Benutzers

Parallele Sitzungen unter dem gleichen Benutzerkonto sind grundsätzlich möglich.

5.3.7 Wechsel der SessionID bei der Authentisierung

Zum Schutz vor Session-Fixation-Angriffen wird nach erfolgter Anmeldung eine bereits bestehende SessionID durch eine neue ersetzt.

Ebenso wird nach einem Wechsel von einem ungesicherten Kommunikationskanal (HTTP) auf einen gesicherten Kommunikationskanal (HTTPS) eine neue SessionID vergeben.

5.3.8 Verwendung von Zusatzmerkmalen

Neben der SessionID werden weitere Merkmale zur Zuordnung zwischen Benutzer und Sitzung verwendet, namentlich die IP-Adresse. Die Session ID hat eine definierte Länge, die aus Sicherheitsgründen zusätzlich abgefragt wird.

5.3.9 IP-Adresse als Zusatzmerkmal

Die IP-Adresse wird serverseitig gespeichert und geprüft. Wechselt die IP-Adresse im Laufe einer Sitzung, führt dies zur automatischen Abmeldung.

5.4 Schutz vor automatisierter Nutzung

Vgl. hierzu die Ausführungen weiter oben.

5.5 Sichere Konfiguration

Nicht benötigte HTTP-Methoden sind deaktiviert. Die Komprimierung der ausgegebenen Webseiten wird z. B. nicht verwendet und ist ausgeschaltet.

Schutzbedürftige Daten werden immer stets einen verschlüsselten Transportkanal übertragen. Die Webanwendung gibt in den Header-Feldern der HTTP-Response das Zeichenkodierungsschema UTF-8 mit an.

Die Webanwendung wird über ein von der Anwendung entkoppeltes System administriert (Partner-Admintool). Das System ist alleine für diesen Zweck bestimmt und hat keine direkte Verbindung zu der Webanwendung.

5.6 Kontrolliertes Einbinden von Daten und Inhalten

Die Webanwendung stellt sicher, dass ausschließlich vorgesehene Daten und Inhalte eingebunden und an den Benutzer ausgeliefert werden.

5.6.1 Filterung der Pfadangaben

Die Benutzereingaben werden auf unerwünschte Zeichen zur Manipulation des Pfades (z. B. „/.." und „\..") gefiltert.

5.6.2 Indizes statt Dateinamen

Bei der Auswahl der Quell-Dateien werden zunehmend Indizes anstelle von Dateinamen verwendet, denen serverseitig hinterlegte Dateinamen zugeordnet werden.

5.6.3 Remote File Inclusion

Bei der Einbindung externer Styles und Javascript-Dateien (Remote File Inclusion) ist die vertrauenswürdige Herkunft dieser Dateien sichergestellt.

5.6.4 Prüfung hochgeladener Dateien

Hochgeladene Dateien werden wie weiter oben bereits dargestellt geprüft, u. a. durch Auswertung des Dateiheaders.

5.6.5 Speicherung hochgeladener Dateien

Hochgeladene Dateien werden in einem Verzeichnis gespeichert, welches nicht über die Web-Schnittstelle erreichbar ist, bzw. in entsprechenden Datenbanken.

5.7 Schutz vertraulicher Daten

Alle vertraulichen Daten werden verschlüsselt übertragen.

5.7.1 Sichere kryptographische Algorithmen

Es werden übliche kryptographische Algorithmen serverseitig umgesetzt.

- Übertragungskanal mit Https-Verschlüsselung
- Passwortverschlüsselung durch SHA2+ zusätzlichen zufällig generierten Schlüssel
- Serverzugriffe per SSH-Verschlüsselung (RSA)

5.7.2 Schutz des Transportkanals (SSL/TLS)

Es werden sichere Transportkanäle eingesetzt (SSL, https, sftp, SSH). Die Verschlüsselung erfolgt über 2048-bit SSL durch OpenSSL.

5.7.3 Schutz der Daten im Browsercache

Das clientseitige Zwischenspeichern (Cachen) von vertraulichen Daten der Web-Anwendung wird durch folgende Direktiven in den HTTP-Headern der Webanwendung unterbunden:

- Cache-Control: no-store
- Pragma: no-cache

5.7.4 Cookie-Flags

Es wird die Domain als Cookie-Flag gesetzt.

5.7.5 Konfiguration der Formularfelder zur Authentisierung

Wenn der Benutzer sein Passwort in das Passwortfeld eintippt, wird dieses nicht im Klartext wiedergegeben, sondern durch sogenannte Wildcards ersetzt (Sterne).

Darüber hinaus wird der Web-Browser angewiesen, vertrauliche Formulardaten (z. B. den Benutzernamen und das Passwort) nicht zwischenzuspeichern und beim nächsten Aufruf des Formulars als Auswahl vorschlagen. Die Option „autocomplete="Off" ist hierfür bei der Definition des Formulars im Formularkopf gesetzt.

5.8 Zugriffskontrolle

Die Autorisierungskomponente stellt sicher, dass Benutzer nur solche Aktionen durchführen können, für die sie über ausreichende Berechtigungen verfügen. Die Zuweisung von Rechten erfolgt dabei auf der Grundlage von Benutzer-Rollen.

5.8.1 Abdeckung aller verwalteten Ressourcen

Die Autorisierungskomponente berücksichtigt alle verwalteten Ressourcen der Webanwendung.

5.8.2 Zugriffskontrolle auf allen System-Ebenen

Die Zugriffskontrolle ist auf allen Ebenen der Webanwendung realisiert.

5.8.3 Durchführung der Zugriffskontrolle bei jedem Zugriff

Jeder Zugriff auf geschützte Inhalte und Funktionen wird kontrolliert, bevor er ausgeführt wird.

5.8.4 Zugriffskontrolle bei URL-Aufrufen

Wo Ressourcen der Webanwendung genutzt werden, ist auch der direkte Aufruf der Ressource über die URL geschützt, Beispiel: deaktivierte Bewerberprofile.

5.8.5 Zugriffskontrolle bei Objekt-Referenzen

Vgl. hierzu die Ausführungen weiter oben.

5.9 Verhinderung von Cross-Site Request Forgery (CSRF, XSRF)

Vgl. hierzu die Ausführungen weiter oben.

5.10 Verhinderung der Blockade von Ressourcen (DoS)

5.10.1 Schutzmechanismen wie gegen Enumeration und Brute-Force

Bei Verdacht auf einen Angriff wird die aufrufende IP-Adresse temporär gesperrt.

6 Betrieb

6.1 Protokollierung sicherheitsrelevanter Ereignisse

Siehe die Ausführungen weiter oben.

6.2 Restriktive Herausgabe sicherheitsrelevanter Informationen

6.2.1 Neutrale Fehlermeldungen

Es werden neutrale Fehlermeldungen eingesetzt.

6.2.2 Filterung von Kommentaren im HTML-Quelltext

Der Quelltext enthält keine sicherheitsrelevanten Kommentare.

6.2.3 Keine sicherheitskritischen Ressourcenangaben in der Datei robots.txt

Die Datei enthält keine sicherheitsrelevanten URLs.

6.3 Zeitnahes Einspielen sicherheitsrelevanter Patches und Updates

Die Programmierer und Systemadministratoren halten sich permanent über entsprechende Sicherheitsupdates auf dem Laufenden: sowohl beim Betriebssystem als auch in allen weiteren eingesetzten Komponenten und Tools. Es werden nur vertrauenswürdige und bewährte Open Source Technologien eingesetzt.

Wie jedes Update durchlaufen auch Sicherheitsupdates den regulären Testzyklus. Die Dokumentation erfolgt in der Projektdatenbank Redmine und dem Versionsmanagementsystem GIT. Über GIT kann notfalls auch schnell wieder auf eine frühere Version zurückgeschaltet werden.

6.4 Dokumentation der zugelassenen Benutzer und Rechteprofile

Die Benutzer- und Rechtedokumentation wird in den jeweiligen Administrationstabellen durchgeführt. Für den Masteruser ist sie auch über das so genannte Partner-Admintool zugänglich und veränderbar. Neben der generellen Rechteverwaltung insbesondere der Vermittlungsfachkräfte, gibt es einzeln zuweisbare Rechte für bestimmte Abfragen, die über das Administrationstool zum Data Warehouse gesteuert werden.

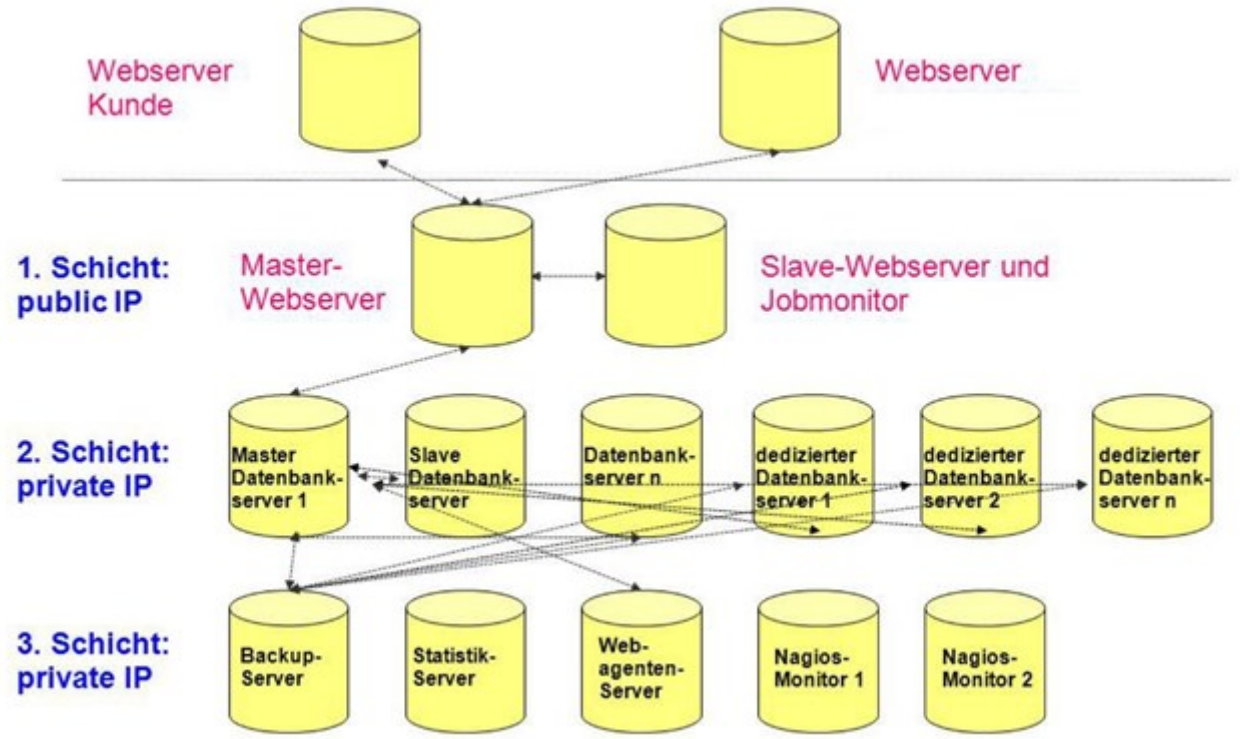
Bestimmte Rechte können auch für das ganze Portal festgelegt werden, z. B. für Firmen das Recht, Stellenanzeigen selbst freizuschalten.

Die Rechteverwaltung ist mit Ausnahme des Rechts von Bewerbern, ihr Profil zu bearbeiten, was durch die zuständige Vermittlungsfachkraft administriert wird, dem Zugriff der einzelnen Nutzer entzogen und nur durch die Masteruser zugänglich. Diese können sie jederzeit auf Aktualität überprüfen und bei Bedarf entsprechende Aktualisierungen vornehmen.

Portalseitige Rechte können nur durch uns als IT-Dienstleister administriert werden.

Die Rechtedokumentation ist in die regelmäßigen Backups einbezogen, womit bei Bedarf auch eine Verlaufsdokumentation erstellbar ist.

7 Übersicht Netzplan



8 Geltung, Evaluierung & Anpassung dieses IT- Sicherheitskonzepts

Das Sicherheitskonzept ist bei jeder Änderung der aktuellen örtlichen und personellen Gegebenheiten und aus sonstigen Anlässen, die Auswirkungen auf das Sicherheitskonzept haben, fortzuschreiben und spätestens nach einem Jahr zu überprüfen. Die letzte Überprüfung fand im Mai 2018 statt.

9 Name und Anschrift des Verantwortlichen

Jobnet.AG
Sitz der Gesellschaft:
Luisenstraße 41
10117 Berlin

Internet: www.jobnet.ag
zentrale E-Mail: info@jobnet.ag

Aufsichtsrat:
Prof. Dr. Franz Egle (Vorsitzender)
Prof. Dr. Joachim Thomas (Stellvertreter)
Prof. Dr. Klaus Zierer

Vorstand:
Dr. Christoph Wesselmann (Vorsitzender), cw@jobnet.ag
Ralf Bultschnieder, Dipl.-Kfm. Dipl.-Volksw., rb@jobnet.ag

Handelsregister:
Berlin Charlottenburg
HRB 163211 B

10. Name und Anschrift des Datenschutzbeauftragten

Vladyslav Prykhodko
Jobnet.AG
Büro Leipzig: Harkortstraße 5
04107 Leipzig
Deutschland
Tel.: +49 341 217 17 94
E-Mail: vp@jobnet.ag
Website: www.jobnet.ag

Jede betroffene Person kann sich jederzeit bei allen Fragen und Anregungen zum Datenschutz direkt an unseren Datenschutzbeauftragten wenden.

REPUBLIK ÖSTERREICH Parlamentsdirektion SIEGEL	**Unterzeichner**	Parlamentsdirektion
	Datum/Zeit-UTC	2021-12-06T11:09:54+01:00
	Prüfinformation	Informationen zur Prüfung des elektronischen Siegels und des Ausdrucks finden Sie unter: https://www.parlament.gv.at/siegel
Hinweis	Dieses Dokument wurde elektronisch besiegelt.	