Produktivdatenbank in ein Testsystem übernehmen (Anmeldung als sadmin)

Diese Seite beschreibt, was zu tun ist, wenn die Datenbank eines Produktivsystems in ein bestehendes Test- oder Integrationssystem (im Folgenden „Testsystem“) übernommen wird und man sich danach wieder als Systemadministrator sadmin anmelden möchte. Sie gilt ab Webdesk 4.56 und richtet sich an Systemadministratoren und Berater. Grundlagen zur Konfigurationsdatei und zum Schlüssel finden Sie in Konfigurationsdatei webdesk.properties und Umgebungsvariablen[1].

Warum die Anmeldung nach der Übernahme scheitert

Webdesk verschlüsselt Passwörter, die in der Datenbank gespeichert sind, mit einem Schlüssel (AES-256). Dazu gehört auch das Passwort von sadmin: Es ist kein gewöhnlicher Benutzer, sondern ein verschlüsselt gespeicherter Systemparameter. Bei jeder Anmeldung entschlüsselt Webdesk den gespeicherten Wert und vergleicht ihn mit dem eingegebenen Passwort.

Der Schlüssel liegt aber nicht in der Datenbank, sondern bei der Installation: im Eintrag webdesk.encryption.key der Datei webdesk.properties oder in der Datei enc.key. Das Testsystem hat seinen eigenen Schlüssel. Nach der Übernahme enthält die Datenbank Werte, die mit dem Schlüssel des Produktivsystems verschlüsselt wurden, das Testsystem entschlüsselt sie aber weiter mit seinem eigenen Schlüssel:

```
Produktivsystem: Datenbank (verschlüsselt mit Schlüssel P) + webdesk.properties mit
 Schlüssel P
Testsystem:      Datenbank übernommen (Schlüssel P)         + webdesk.properties mit
 Schlüssel T
                 # Entschlüsseln mit T schlägt fehl
```

Folgen:

- Die Anmeldung als sadmin scheitert, auch mit dem richtigen (produktiven) Passwort. Im Log steht eine Meldung mit javax.crypto.AEADBadTagException: Tag mismatch.
- Alle anderen verschlüsselt gespeicherten Passwörter aus der Produktivdatenbank sind im Testsystem ebenfalls unlesbar, z. B. von Konnektoren, Repositories der Dokumentenverwaltung oder Groupware-Zugriffen.

Die Lösung ist, dem Testsystem den Schlüssel des Produktivsystems zu geben.

Vor dem ersten Start: Produktivkonfiguration beachten

Achtung: Mit der Produktivdatenbank übernimmt das Testsystem auch die Konfiguration des Produktivsystems, z. B. Zeiterfassungs-Anbindungen, Konnektoren, E-Mail-Versand und geplante Jobs. Sobald die Passwörter wieder entschlüsselt werden können, kann das Testsystem auf produktive Umsysteme zugreifen, Daten schreiben oder E-Mails versenden. Prüfen und deaktivieren bzw. umstellen Sie diese Einstellungen, bevor Sie das Testsystem mit dem Produktivschlüssel in Betrieb nehmen.

Hinweis: Mit dem Produktivschlüssel kann das Testsystem alle verschlüsselten Produktivdaten lesen. Schützen Sie die Konfiguration des Testsystems danach genauso wie die des Produktivsystems.

Anmeldung über Reverse Proxy (SSO per HTTP-Header)

Viele Installationen melden die Benutzer über einen Reverse Proxy an, der den Benutzernamen in einem HTTP-Header an Webdesk übergibt (Systemparameter AuthenticationOptions.ssoHeaderNameRemoteUser, Standard X-Remote-User). Webdesk übernimmt den Header aber nur für Anfragen, deren URL mit dem Systemparameter AuthenticationOptions.ssoHeaderAuthenticationValidationUrl beginnt, z. B. https://<prod-host>/webdesk/.

Diese Systemparameter stehen in der Datenbank und werden bei der Übernahme mitkopiert. Im Testsystem passt die Prüf-URL des Produktivsystems dann nicht zur Adresse des Testsystems. Folge: Bei jeder Anfrage mit dem Header scheitert die automatische Anmeldung mit einer java.lang.NullPointerException in AuthenticationUtils.getNormalizedUser, jedes Mal mit vollständigem Stacktrace im Log und in catalina.out. Die Log-Dateien wachsen dadurch schnell, bis das Laufwerk voll ist und der Server ausfällt. Ein niedrigeres Log-Level verhindert das nicht.

Ändern Sie die Prüf-URL daher vor dem ersten Start des Testsystems auf dessen Adresse. Da eine Anmeldung zu diesem Zeitpunkt noch nicht möglich ist, geschieht das direkt in der Datenbank des Testsystems, bei gestopptem Webdesk und nach einer Sicherung der Tabellen PoBeanProperty und PoBeanPropertyValue:

```
SELECT p.propertyName, v.property
FROM PoBeanProperty p JOIN PoBeanPropertyValue v ON v.BEAN_UID = p.BEAN_UID
WHERE p.beanName = 'AuthenticationOptions'
  AND p.propertyName IN ('authenticationMode', 'ssoHeaderNameRemoteUser',
 'ssoHeaderAuthenticationValidationUrl');

UPDATE PoBeanPropertyValue SET property = 'https://<test-host>/webdesk/'
WHERE BEAN_UID = (SELECT BEAN_UID FROM PoBeanProperty
                  WHERE beanName = 'AuthenticationOptions'
                    AND propertyName = 'ssoHeaderAuthenticationValidationUrl');
```

- Geben Sie die Adresse so an, wie Webdesk die Anfragen sieht, in derselben Form wie im Produktivsystem (Protokoll, Rechnername, Pfad).
- Soll das Testsystem kein SSO verwenden, setzen Sie stattdessen authenticationMode auf DB.
- Ein Eintrag in webdesk.properties reicht nicht: Beim Start haben die Werte aus der Datenbank Vorrang.
- Sobald die Anmeldung wieder funktioniert, kann der Wert in der Maske „Systemparameter“ gepflegt werden.

Ist das Testsystem bereits mit der falschen Prüf-URL gestartet: Webdesk stoppen, den Reverse Proxy vorübergehend keinen Header senden lassen (oder Webdesk direkt ohne Proxy aufrufen), Log-Dateien entfernen, um wieder Platz zu schaffen, und dann die Prüf-URL wie oben ändern.

Vorgehen

1. Webdesk im Testsystem stoppen (bei mehreren Knoten alle Knoten).
2. Sicherung der Datei webdesk.properties des Testsystems anlegen (und von enc.key, falls vorhanden).
3. **Schlüssel des Produktivsystems ermitteln.** In der webdesk.properties des Produktivsystems steht unter webdesk.encryption.key.source, wo der Schlüssel liegt:
   - webdesk_properties: Der Schlüssel ist der Wert von webdesk.encryption.key in derselben Datei.
   - file: Der Schlüssel steht in der Datei enc.key neben webdesk.properties.

   Übertragen Sie den Schlüssel auf einem sicheren Weg, nicht per E-Mail oder Ticket.
4. **Schlüssel im Testsystem eintragen**, auf eine der beiden Arten:

   ```
   webdesk.encryption.key.source = webdesk_properties
   webdesk.encryption.key = <Schlüssel des Produktivsystems>
   ```

   oder enc.key durch die Datei des Produktivsystems ersetzen und webdesk.encryption.key.source = file setzen (der Eintrag webdesk.encryption.key entfällt dann). Bei mehreren Knoten auf jedem Knoten.
5. **Zugangsdaten der Datenbank neu verschlüsseln.** Benutzername und Passwort der Testdatenbank in webdesk.properties sind mit dem bisherigen Schlüssel des Testsystems verschlüsselt und passen nach dem Schlüsselwechsel nicht mehr. Ohne diesen Schritt kann sich das Testsystem nicht mehr mit seiner Datenbank verbinden. Den Schritt brauchen Sie nur, wenn webdesk.credentials.encrypted = true gesetzt ist; bei false (Klartext) oder beim alten Eintrag webdesk.connection.encrypted hängen die Werte nicht vom Schlüssel ab. Zwei Möglichkeiten:

- **Webdesk verschlüsselt beim Start (empfohlen):** Benutzernamen und Passwörter der Webdesk- und der Shark-Datenbank im Klartext eintragen und die Zeilen webdesk.credentials.encrypted und shark.credentials.encrypted **löschen** (nicht auf false setzen):

  ```
  webdesk.connection.username = <Datenbank-Benutzer>
  webdesk.connection.password = <Datenbank-Passwort>
  shark.connection.username = <Datenbank-Benutzer>
  shark.connection.password = <Datenbank-Passwort>
  ```

  Beim nächsten Start verschlüsselt Webdesk die Werte mit dem neuen Schlüssel, trägt die Einträge ...credentials.encrypted = true wieder ein und speichert die Datei. Der Benutzer, unter dem Webdesk läuft, braucht dafür Schreibrechte auf die Datei. Bei false würde Webdesk die Werte als gewollten Klartext stehen lassen.
- **Mit dem Werkzeug:** Auf Windows-Installationen manageWebdeskDBProps.bat im Verzeichnis bin von Tomcat starten, „Basic“-Modus wählen und im Menü die Option zur Eingabe neuer Werte verwenden. Das Werkzeug verschlüsselt die eingegebenen Werte mit dem Schlüssel aus der Datei. Verwenden Sie dafür nicht die Parameter webdeskLogin= / webdeskPassword= des „Expert“-Modus: Bei bereits verschlüsselten Dateien werden sie unverschlüsselt gespeichert. Das Werkzeug unterstützt nur webdesk.encryption.key.source = webdesk_properties.

6. **Bei SSO über einen Reverse Proxy:** Prüf-URL ssoHeaderAuthenticationValidationUrl in der Datenbank auf die Adresse des Testsystems ändern (siehe Abschnitt „Anmeldung über Reverse Proxy“).
7. Webdesk starten. Prüfen, dass das System startet, und dass in webdesk.properties die Zugangsdaten wieder verschlüsselt (als Base64-Text) stehen.
8. Als sadmin mit dem Passwort des Produktivsystems anmelden.
9. Die geänderte webdesk.properties (bzw. enc.key) des Testsystems neu sichern.

Fehlersuche

- **Das Log füllt sich mit java.lang.NullPointerException in AuthenticationUtils.getNormalizedUser, bis das Laufwerk voll ist:** Die Prüf-URL für die Anmeldung per Reverse-Proxy-Header ist noch die des Produktivsystems (siehe Abschnitt „Anmeldung über Reverse Proxy“).
- **Testsystem startet nach dem Schlüsselwechsel nicht, Datenbankverbindung schlägt fehl:** Die Zugangsdaten der Datenbank sind noch mit dem alten Schlüssel des Testsystems verschlüsselt (Schritt 5).
- **Anmeldung als sadmin scheitert weiterhin, im Log steht AEADBadTagException:** Das Testsystem verwendet nicht den Schlüssel des Produktivsystems. Prüfen Sie Schlüsselquelle und Schlüssel bzw. enc.key auf allen Knoten.
- **Die Zugangsdaten stehen nach dem Start weiterhin im Klartext in der Datei:** Webdesk konnte die Datei nicht schreiben (Schreibrechte), oder die Zeilen ...credentials.encrypted stehen noch mit false in der Datei.
- **Der Schlüssel des Produktivsystems ist nicht mehr auffindbar:** Die verschlüsselten Werte aus der Produktivdatenbank, einschließlich des sadmin-Passworts, lassen sich dann nicht wiederherstellen. Wenden Sie sich an den Webdesk-Support.

1. /daisy/personalwolke-admin/11598-dsy.html