Konfiguration

by Thorsten Kamann

Die Konfiguration des SecurityWatchers befinden sich in einer XML-Datei. Hier haben sie verschiedene Möglichkeiten die Sicherheitsregeln einzelner Host und Kontexte (Anwendungen) zu bestimmen.

1. Die Konfigurationsdatei

Die kompletten Sicherheitsregeln eines ganzen Profils der Centaurus-Platform befinden sich in einer XML-Datei. Diese finden Sie unter `${centaurus.base}/conf/security.xml`.

Die Konfiguration enthält mehrere Typen von Permissions:

system	Dieser Type Permissions definiert regeln die für alle Elemente und Komponenten innerhalb der gesamten Centaurus-Platform gelten.
engine	Mit diesem Typ bestimmen Sie Regeln, die für alle Engines gelten sollen. Mit dem Attribute `name` können Sie diesen Eintrag für bestimmte Engines filtern.
host	Wenn Sie Regeln bestimmen wollen, die für jeden installierten Host gelten sollen, dann können Sie das mit diesem Typ machen. Sollen die Regeln nur für einen bestimmten Host gelten, dann fügen Sie das Attribute `name` hinzu. Wenn die Regel nur für einen Host in einer bestimmten Regel gelten soll, dann können Sie noch das Attribute `engine` hinzu.
context	Regeln können Sie natürlich auch auf Basis des Kontexts (Anwendungsebene) bestimmt werden. Auch hier können Sie auf bestimmte Kontexte filtern (Attribute `name`). Natürlich kann hier noch feiner gefiltert werden,

	z.B. auf einen Host (`host`) und/oder eine Engine (`engine`).

Innerhalb der Permissions können Sie beliebig viele Regeln definieren (Element `grant`). Sehen Sie dazu auch die Basiskonfiguration.

2. Sicherheitsregeln definieren

Regeln können nur innerhalb eines Permission-Elements definiert werden. Es gibt allerdings keine Einschränkungen in Bezug auf die Anzahl der Regeln innerhalb eines Permission-Elements.
Eine Sicherheitsregel wird mit dem Element `grant` definiert. Die Konfiguration einer solchen Regel wird durch ein oder mehrere Attribute bestimmt. Folgende Atrribute können Sie verwenden:

codebase	Bestimmt die `codebase` für die diese Regel gelten soll. Bedenken Sie dass die `codebase` immer mit dem Prefix `file:` beginnen sollte. Auch Platzhalter werden unterstützt: • * - Die Regel gilt für alle Dateien im Verzeichnis • - - Die Regel gilt für alle Verzeichnis/Dateien rekursiv • / - Es werden automatisch 3 Regeln mit den beiden Platzhaltern * und - sowie dem kompletten Pfad erzeugt • ${appbase} - wird durch das aktuelle Host-Applikationsverzeichnis ersetzt (nur für `type=host` oder `type=context`) • ${workdir} - wird durch das aktuelle Arbeitsverzeichnis ersetzt (nur für `type=host` oder `type=context`) • ${docbase} - wird durch das aktuelle Dokumenten-Root des Kontexts ersetzt (nur für `type=context`)
permission	Bestimmt den Namen der Permission. Mögliche Werte sind hier: • java.io.FilePermission • java.net.SocketPermission • java.util.PropertyPermission Weitere Möglichkeiten und viele weitere Informationen finden Sie im Permission-Guide von Sun.
name	Mit diesem Attribut bestimmen Sie die Regel etwas genauer. Als Beispiel nehmen wir die 3 Permissions:

	• java.io.FilePermission - bestimmt die Dateisystem-Resource auf die zugegriffen werden darf • java.net.SocketPermission - bestimmt die Adresse auf die zugergiffen werden darf • java.util.PropertyPermission - gibt den Namen der Property an, auf die zugegriffen werden darf Die Platzhalter für die `codebase` gelten für die `java.io.FilePermission` genauso.
action	Hiermit können Sie die Aktionen, die erlaubt sein sollen, bestimmen. Nicht alle Regeln benötigen unbedingt dieses Attribute. Wenn Sie dieses Attribute einsetzen müssen Sie auch ebenfalls das Attribute `name` setzen. Mögliche Aktionen für die obigen 3 Permissions aus dem obigen Beispiel sind: • java.io.FilePermission - read,write,delete,execute • java.net.SocketPermission - connect, resolve, accept, listen • java.util.PropertyPermission - read,write

3. Sicherheitsregeln von Webanwendungen

Ein häufiger Anwendungsfall ist es, dass Webanwendungen Dateien schreiben müssen. Das ist standardmässig nicht erlaubt. Damit der Administrator nicht bei jeder Webanwendung aktiv eingreifen muss, gibt es die Möglichkeit eigene Sichereitsregeln im WAR-Archiv oder DocBase abzulegen. Der Name einer solchen Datei ist `security.xml` und muss sich im Verzeichnis `/META-INF` der Webanwendung befinden.

Diese Regeln werden - bevor Sie zu dem SecurityWatcher hinzugefügt werden - gefiltert, damit nur bestimmte Regeln erlaubt sind. Der Filter befindet sich in `${centaurus.base}/conf/xsl/webapps_security.xsl`.
Standardmässig sind das:

- java.io.FilePermission - nur innerhalb des AppBase des Hosts
- java.net.SocketPermission - nur die Aktionen `connect` und `resolve` sind erlaubt
- java.util.PropertyPermission - Es sit nur lesender Zugriff erlaubt

Die Struktur weicht etwas von der Konfigurationsdatei des SecurityWatchers ab:

```
<?xml version="1.0" encoding="UTF-8"?>
<security>
        <permissions>
                <grant
                        permission="java.io.FilePermission"
```

```
                        name="temp/"
                        action="read,write,delete"/>
        </permissions>
</security>
```

Eine `codebase` ist nicht zwingend vorgeschrieben. Wenn diese fehlt wird das DocBase des Kontexts als `codebase` verwendet. Wenn Sie die Regel nur einem bestimmten Teil der Anwendung zukommen lassen wollen können Sie dies so tun:
`codebase="/WEB-INF/lib/special.jar"`
Dateiberechtigungen können Sie nur innerhalb des eigenen DocBase vornehmen. Werden mehr Berechtigung gebraucht, dann muss dies in der Konfigurationsdatei vorgenommen werden.

4. Aktivierung des SecurityWatchers

Wenn Sie für die Erstellung neuer Hosts das HostCreator-Plugin nutzen, dann baruchen Sie keine weiteren Anpassungen vornehmen.

Wenn Sie manuell die `${centaurus.base}/conf/server.xml` verändern, kontrollieren Sie die folgenden Elemente:

- Engine
- Host

Alle diese Einträge müssen einen Listener haben:

```
<Listener className="de.centaurus.platform.plugins.security.SecurityWatcherLifecycleLis
```

5. Die DTD der Konfigurationsdatei

Leider noch nicht fertig

Dieser Bereich der Dokumentation ist für dieses Release noch nicht fertig geworden.
Beim nächsten Release wird dieser Bereich sicher verfügbar sein.

6. Die Basiskonfiguration

Nach einer Neuinstallation der Centaurus-Platform sieht die `${centaurus.base}/conf/security.xml` so aus:

```
<security>

        <!-- Here are the permissions for the System and Server Code-->
        <permissions type="system">
                <grant codebase="file:${java.home}/lib/-"
```

```
                      permission="java.security.AllPermission"/>
                <grant codebase="file:${java.home}/jre/lib/ext/-"
                     permission="java.security.AllPermission"/>
                <grant codebase="file:${java.home}/../lib/-"
                     permission="java.security.AllPermission"/>
                <grant codebase="file:${java.home}/lib/ext/-"
                     permission="java.security.AllPermission"/>
                <grant codebase="file:${catalina.home}/common/-"
                     permission="java.security.AllPermission"/>
                <grant codebase="file:${catalina.home}/shared/-"
                     permission="java.security.AllPermission"/>
                <grant codebase="file:${catalina.home}/server/-"
                     permission="java.security.AllPermission"/>
                <grant codebase="file:${catalina.base}/lib/-"
                     permission="java.security.AllPermission"/>
                <grant codebase="file:${catalina.base}/plugins/-"
                     permission="java.security.AllPermission"/>
                <grant codebase="file:${catalina.base}/hosts/localhost/webapps/-"
                     permission="java.security.AllPermission"/>
                <!-- Here are all java.util.PropertyPermissions -->
                <grant permission="java.util.PropertyPermission"
                     name="java.home" action="read"/>
                <grant permission="java.util.PropertyPermission"
                     name="java.naming.*" action="read"/>
                <grant permission="java.util.PropertyPermission"
                     name="javax.sql" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="javax.sql.DataSource.Factory" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="os.name" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="os.version" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="os.arch" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="file.separator" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="path.separator" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="line.separator" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.version" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.vendor" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.vendor.url" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.class.version" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.specification.version" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.specification.vendor" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.specification.name" action="read"/>
```

```
                <grant permission="java.util.PropertyPermission"
                    name="java.vm.specification.version" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.vm.specification.vendor" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.vm.specification.name" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.vm.version" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.vm.vendor" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.vm.name" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="file.encoding" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="jaxp.debug" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="user.home" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="user.name" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="user.dir" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.library.path" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.io.tmpdir" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.ext.dirs" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.compiler" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="java.class.path" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="catalina.home" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="catalina.base" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="OJB.*" action="read"/>
                <grant permission="java.util.PropertyPermission"
                    name="centaurus.*" action="read"/>

                <!-- Here are all java.lang.RuntimePermissions -->
                <grant permission="java.lang.RuntimePermission"
                     name="getAttribute"/>
                <grant permission="java.lang.RuntimePermission"
                     name="accessDeclaredMembers"/>
                <grant permission="java.lang.RuntimePermission"
                    name="setContextClassLoader"/>
                <grant permission="java.lang.RuntimePermission"
                     name="accessClassInPackage.org.apache.jasper.runtime"/>
                <grant permission="java.lang.RuntimePermission"
                     name="accessClassInPackage.org.apache.jasper.runtime.*"/>
                <grant permission="java.lang.RuntimePermission"
                    name="accessClassInPackage.org.apache.catalina"/>
```

```
                <grant permission="java.lang.RuntimePermission"
                    name="accessClassInPackage.org.apache.catalina.*"/>
                <grant permission="java.lang.RuntimePermission"
                    name="accessClassInPackage.org.apache.tomcat.util.http"/>
                <grant permission="java.lang.RuntimePermission"
                   name="accessClassInPackage.org.apache.tomcat.util.http.*"/>
                <grant permission="java.lang.RuntimePermission"
                   name="accessClassInPackage.org.apache.tomcat.util.net"/>
                <grant permission="java.lang.RuntimePermission"
                   name="accessClassInPackage.org.apache.tomcat.util.net.*"/>

                <!-- Here are the java.net.SocketPermissions -->
                <grant permission="java.net.SocketPermission"
                    name="localhost:*" action="listen, connect, resolve, accept"/>
                <grant permission="java.net.SocketPermission"
                    name="127.0.0.1:32000" action="connect, resolve"/>
                <grant permission="java.net.SocketPermission"
                    name="*:21" action="listen, connect, resolve, accept"/>
                <grant permission="java.net.SocketPermission"
                    name="*:25" action="listen, connect, resolve, accept"/>
                <grant permission="java.net.SocketPermission"
                    name="*:80" action="listen, connect, resolve, accept"/>
                <grant permission="java.net.SocketPermission"
                    name="*:443" action="listen, connect, resolve, accept"/>

                <!-- Here are the other Permissions -->
                <grant permission="java.lang.reflect.ReflectPermission"
                     name="suppressAccessChecks"/>
        </permissions>
</security>
```