Der SecurityWatcher

Überblick über das SecurityWatcher-Plugin

by Thorsten Kamann, Peter Roßbach

NOTICE:

Der SecurityWatcher ist für das aktive Management der Sicherheitsregeln (Security Policy) der Centaurus-Platform zuständig. Er führt automatisch eine Änderung der Sicherheitsregeln des Servers, durch ohne diesen zu unterbrechen. Weiterhin kann ein Sicherheitsregelwerk für jede Webanwendung automatisch übernommen und entfernt werden.

1. Idee der dynamischen Sicherheitsüberwachung

Innerhalb eines Web-Container tummeln sich - aus Sicht eines Providers - idealerweise etliche Hosts mit den verschiedensten Anwendungen. Da nicht alle Kunden immer nur Gutes im Schilde führen, sind entsprechende Schutzmaßnahmen dringend notwendig. Es muss verhindert werden, dass jedermann auf die eine oder andere Ressource des Rechners oder der Anwendung eines anderen Kunden zugreift. Die Anwendungen und Anwender, die sich einen Rechner bzw. Server teilen, müssen effektiv voreinander geschützt werden.

Wenn Ihr erster netter Zeitgenosse mal eben mit `System.exit(1);` Ihren Server versenkt, werden Sie spätestens dann über mehr Sicherheit nachdenken wollen.
Genau hier setzt der SecurityWatcher in der Centaurus-Platform an. Die Centaurus-Platform wird immer mit einem SecurityManager der JVM gestartet und verwandelt diese in eine Trutzburg.

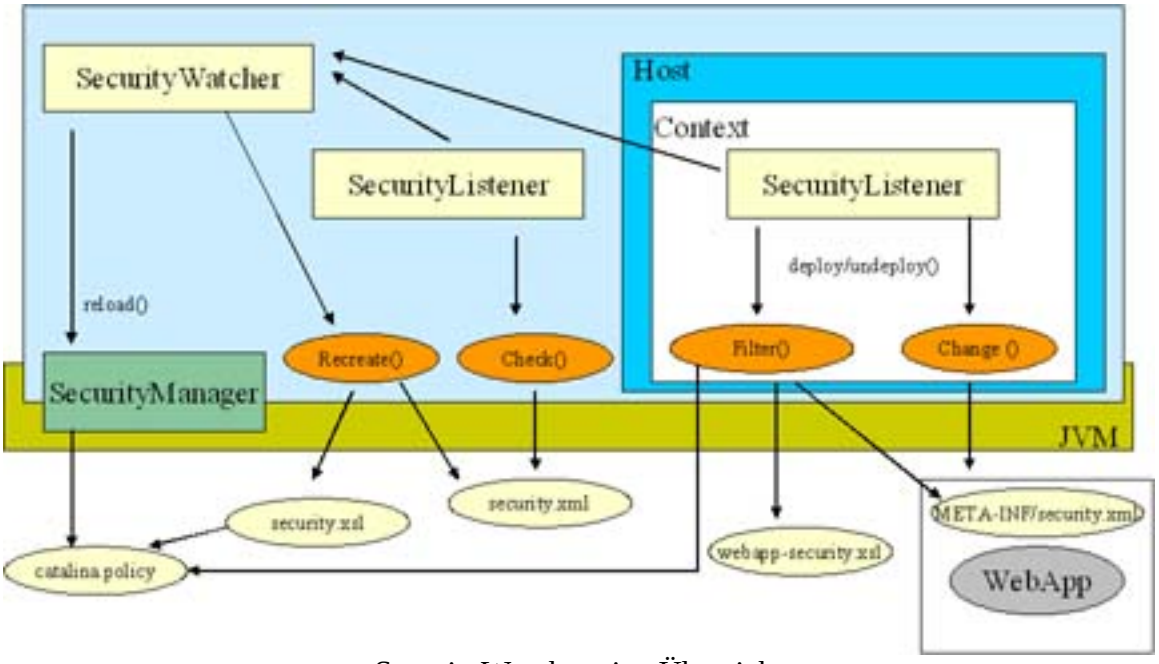
Viele Anbieter von Web Anwendungen nutzen heutzutage die Gelegenheit des so genannten Hostings, bei dem die eigenen Webanwendungen auf einem externen Server zusammen mit einer Vielzahl anderer Anwendungen weiterer Anbieter der Öffentlichkeit kostengünstig zugänglich gemacht werden. Die Gründe für ein wenig mehr Sicherheit der Anwendungen, die innerhalb eines solchen Servers ablaufen, liegen auf der Hand. Die aus Kostengründen geteilten Ressourcen sollen natürlich geteilt bleiben. Wir Java Entwickler haben da mit den

Möglichkeiten unserer Programmierumgebung das große Los gezogen. Die Java virtuelle Maschine (JVM) bietet ausgereifte Sicherheitsmechanismen, die uns als Betreiber von Tomcat Servern gerade zu auffordert eine gesicherte Festung hinter unserer Firewall zu bauen. Auf einem Server laufen meist verschiedene Services und die sollten in verantwortlicher und kontrollierter Weise miteinander umgehen. Der Administrator hat die Aufgabe sowohl die Centaurus-Platform abzusichern, die notwendigen Services freizugeben, als auch die einzelnen Web-Anwendungen entsprechend sinnvoll einzuschränken.

2. Die Lösung ist eine aktive Sicherheitsüberwachung

Die Centaurus-Plattform nutzt die Möglichkeiten des SecurityManager von Java vollständig aus. Grundsätzlich wird das Sicherheitsregelwerk in der Datei */temp/security.policy* statisch gehalten. Der SecurityManager kann dieses Regelwerk (Policy) aber jederzeit aktualisieren. Das gewählte Format der Policy- Dateien sagt uns für eine vernünftige Verarbeitung wenig zu und deshalb haben wir lieber auf das vertraute XML zurückgegriffen. Unsere Policy hat eine DTD, kann somit sicher mit jedem XML Editor erstellt werden und eignet sich prima zur Verarbeitung und Filterung mit XSLT.

Ein globaler Security-Listener erzeugt aus unserem XML-Format das für den SecurityManager von Java lesbare Policy-Format und veranlaßt diesen zu dem entsprechenden Neuladen der Policy. Dies geschieht für die laufenden Prozesse vollständig transparent. Jede Web-Anwendung bekommt über einen *DefaultContext* der Engine einen Listener für das Management der Sicherheitsregeln. Auch dieser SecurityWatcher kann via einstellbaren Zeiträumen Änderungen überwachen. Beim Start oder Stopp wird die entsprechende Rücknahme der Regeln im SecurityManager veranlasst.



SecurityWatcher eine Übersicht

Klicken Sie auf das Bild für die Vollansicht

3. Filterung der Sicherheitsregeln einer Webanwendung

Jede Webanwendung kann mit der Datei *META-INF/security.xml* seine Sicherheitsregeln direkt mitbringen. Natürlich sind diese nicht beliebig, sondern wir erlauben nur den Zugriff auf bestimmte Properties und Zugriffe auf das Dateisystem. Der Filter für die zulässigen Regeln ist mit einem XSLT Stylesheet `conf/xsl/webapps-security.xsl` realisiert.

4. Funktionsweise

Der SecurityWatcher ist als Server-Listener und Context-Listener realisiert, d.h. er wird beim Start der Centaurus-Platform mitgestartet und beim Beenden mit heruntergefahren. Darüberhinaus sind er und der RuleEditor MBeans und können über eine JMX-Konsole (MX4J und MC4J) kontrolliert werden.

Der SecurityWatcher hat bisher nur einen Parameter, den Sie an Ihre Bedürfnisse anpassen können. Dies passiert in der `${centaurus.base}/conf/server.xml`.
Suchen Sie dort den entsprechenden Listenereintrag:

```
<Listener
className="de.planetes.centaurus.plugins.security.SecurityWatcherLifecycl
```

```
... />
```

Dort können Sie folgende Parameter konfigurieren:

checkInterval	Das Intervall in dem der SecurityWatcher die Datei *conf/security.xml* überprüfen soll. Die Angabe erfolgt in Millisekunden (**Standard: 5000**). Ein Wert kleiner 1000 Msec ist nicht erlaubt.

Table 1: