
Protokoll

GK9.1 "Security Concepts"

Systemtechnik Lernbuero
5BHIT 2016/17

Lukas Zuba

Note:
Betreuer: T.Michaeler

Version 1.0
Begonnen am 12. Oktober 2017
Beendet am 13. Oktober 2017

Inhaltsverzeichnis

1	Einführung	1
1.1	Ziele	1
1.2	Voraussetzungen	1
1.3	Aufgabenstellung	1
2	Ergebnis	2
2.1	Überblick erhalten	2
2.2	Installieren von CAS Server	2
2.2.1	Installieren mittels source als zip	2
2.2.2	Installieren mittels vorkonfiguriertem Projekt	2
2.2.3	Installieren mittels template	3
2.3	Ausführen des CAS Server	3
2.3.1	Ausführen über externe Konsole	3
2.3.2	Ausführen über integrierte Server von Eclipse	3
2.4	Https konfigurieren	4

1 Einführung

Diese Uebung soll helfen die Funktionsweise eines Central Authentication Service (CAS). Dabei sollen erste Erfahrung mit den Technologien Single Sign On (SSO) und Ticket Granting Ticket Service (TGT) gemacht werden.

1.1 Ziele

Das Ziel dieser Uebung ist die Installation eines CAS-Servers und das Testen einer Anbindung mittels einem CAS-Clients. Die CAS Implementierung basiert auf einem CAS Protokoll, das mit Hilfe von 3 Teilen umgesetzt wird: der CAS-Client (meist ein Webbrowser) stellt eine Anfrage an ein Webapplikation stellt. Die Nutzung der Webapplikation erfordert einen Authentifikation, aus diesem Grund wird der Client an den CAS-Server weitergeleitet. Wenn dieser Schritt erfolgreich war, dann wird vom CAS-Server ein Service-Ticket an den Client ausgestellt. Der Client wiederrum leitet das Ticket an die Webapplikation weiter, wo eine sichere Verbindung zwischen Webapplikation und CAS-Server aufgebaut wird, um das Ticket zu prüfen. Wenn diese Validierung erfolgreich war, kann das Resultat der Webapplikation an den Client zurückgegeben werden.

1.2 Voraussetzungen

- Grundlagen zu Central Authentication Service, Single Sign On- und Ticket Granting Ticket Service
- Java Programmierkenntnisse
- Installation Java JDK 1.8, Apache Tomcat
- Installation und Verwendung von Repository Management git
- Verwendung von Build Tools maven oder gradle

1.3 Aufgabenstellung

Installieren Sie eine CAS-Server und probieren Sie damit eine Webapplikation Ihrer Wahl einzubinden.

- Download: Apereo CAS 5.0.9
- Installation einer Demo Anwendung
- Entwicklung einer eigenen Webapplikation, die in der CAS Infrastruktur eingebunden wird
- Entwerfen Sie ein Konzept zum Testen der SSO- und TGT-Funktionalität

2 Ergebnis

Folgende Punkte wurden bearbeitet:

- Überblick erhalten
- Installieren von CAS Server
- Ausführen des CAS Servers
- Https konfigurieren

2.1 Überblick erhalten

Zu beginn der Übung habe ich noch immer nicht ganz verstanden, wie der CAS Server ausgeführt wird, ob das eine eigenständige Anwendung ist, ob das nur eine Erweiterung für ein Projekt ist, ob ich CAS an einen Dienst anhängen muss.

Zusätzlich habe ich dann ewig im Internet nach einem Quick-Start und einem erweiterten Tutorial gesucht. Ein Tutorial habe ich zwar gefunden, habe allerdings erst etwas später bemerkt, dass dieses Tutorial seit über einem Jahr veraltet ist und nicht mehr verwendet werden sollte und hat mich auf die offizielle Webseite weitergeleitet. Etwas ähnliches wie einen Quick-Start habe ich nicht gefunden, daher hatte/habe ich noch immer eine klare Vorstellung, wie die Struktur auf dem Server und auf dem Client aussehen soll und über Technologie ich einsetzen soll, außer Tomcat und CAS.

Zuletzt habe ich dann begriffen, dass der Server zumindest über eine war-Datei auf einem Tomcat Server deployed werden kann und damit den ersten Schritt legt.

2.2 Installieren von CAS Server

Als ersten Schritt habe ich den CAS Server heruntergeladen/installiert und dabei einige Möglichkeiten versucht.

2.2.1 Installieren mittels source als zip

Hierbei habe ich über den Link, welcher in der Aufgabebeschreibung bereitgestellt war, den cas server als Sourcecode für Java heruntergeladen. Daraufhin habe ich mir gedacht, dass die Source-dateien in einem Java Projekt verwendet werden müssen. Allerdings fand ich keine Anleitung, wie ich die Source-dateien in ein Java Projekt implementiere und dann deployen kann. Daher habe ich diesen Weg wieder verworfen.

2.2.2 Installieren mittels vorkonfiguriertem Projekt

Als zweite Option habe ich auf der offiziellen Webseite von CAS einen Link zu einer Webseite gefunden, die Eclipse Maven/Gradle Projekt modular zusammenbauen kann. Dort habe ich ein

Maven Projekt ausgewählt und 'Tomcat' als Rolle hinzugefügt und dieses Projekt dann heruntergeladen. Dieses Projekt habe ich dann in Eclipse eingefügt und dann hat er begonnen Abhängigkeiten herunterzuladen. Das hat allerdings kein Ende nach 10 Minuten genommen, also hab ich dieses Idee verworfen.

2.2.3 Installieren mittels template

Zuletzt habe ich auf der offiziellen Webseite einen Link zu einem Github Repo gefunden, in dem ein Beispielrepo bereitgestellt wurde und einen Anleitungen, wie CAS gebildet und deployed werden kann. Dieses Repo habe ich dann runtergeladen und in Eclipse als Maven Projekt eingebunden. Das hat dann keine Probleme gemacht und konnte dann mit der Aufgabe weiter machen.

2.3 Ausführen des CAS Server

Nachdem ich ein Projekt mit einem CAS Server hatte, habe ich versucht diesen CAS Server auf einem Tomcat Server zu deployen.

2.3.1 Ausführen über externe Konsole

Als erstes habe ich, wie im Github Repo erklärt, die cas.war Datei über eine externe Konsole zu starten. Diese Methode gab mir ,neben einem Fehler, dass kein JAVA_HOME Variable gefunden/richtig ausgeführt werden konnte, einen Fehler, dass der Prozess zu wenige Rechte hat, um den Server zu starten.

2.3.2 Ausführen über integrierte Server von Eclipse

Dann ist mir eingefallen, dass Eclipse eine Möglichkeit bittet, Tomcat Server über Eclipse zu starten.

Davor musste ich aber noch alle Maven Probleme beseitigen, die mit dem Importieren entstanden sind. Also habe ich eine 'Maven Update' und eine 'Maven Build' mit Goal 'clean install' durchgeführt. Daraufhin habe ich eine neue Config für einen 'Apache Tomcat 9' Server erstellt und diesem das entsprechende Projekt zugewiesen.

Als ich den Server gestartet habe, bekam ich 2 unterschiedliche Fehler: 'Access Denied' und 'Failed to start child'.

Access Denied habe ich nur lösen können, indem ich die Serverkonfig/den Server in Eclipse komplett gelöscht und neu angelegt habe.

Failed to start child war meist einfach zu lösen, indem ich 'Maven Update' und 'Maven Build' mit Goal 'clean install' nochmals durchgeführt habe. Selten hat das nicht geholfen, da half auch nur noch gesamtes Projekt lösen und neu importieren.

Manchmal hat dann der Server mit CAS gestartet ohne Problem, abgesehen von dem Problem, dass Eclipse Server eigenständig terminiert, wenn diese nicht innerhalb von 45 Sekunden starten. Diese Einstellung habe ich bei jeder Serverkonfiguration jedesmal ändern müssen.

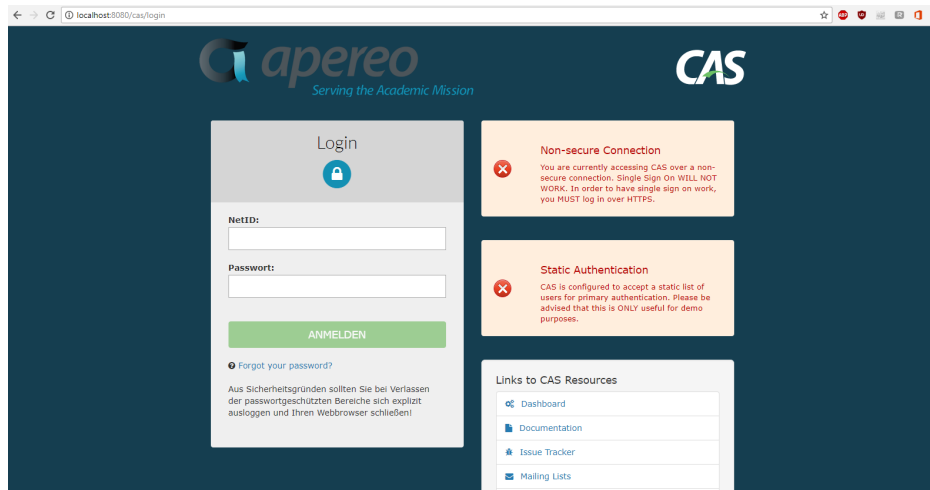


Abbildung 1: Gestarteter CAS Server

2.4 Htpps konfigurieren

Als ich einen funktionierenden Server hatte, habe ich begonnen Htpps zu ermöglichen, da nur unter Htpps Single Log on funktioniert.

Als erstes habe ich mit dem Java-Tool keytool ein Zertifikat erstellt und dieses im Projektordner gespeichert.

Dann habe ich die Konfiguration des Tomcat Server gesucht und habe dort den SSL Connector eingeschaltet und angepasst.

Als ich dann versucht habe, den Server zu starten und dann startet der Server nicht mehr mit dem Fehler Access Denied. Dieses Problem habe ich bis jetzt noch immer nicht gelöst.

Literatur

Tabellenverzeichnis

Listings

Abbildungsverzeichnis

1	Gestarteter CAS Server	4
---	----------------------------------	---