

SoapAnalyzer

IT Security / Betreuer: Prof. Dr. Eric Dubuis
 Experte: Andreas Dürsteler

Vertraulichkeit und Integrität von Daten sind unverzichtbar geworden. Auch im Umfeld von Webdiensten kommen diese Sicherheitsziele zum Tragen, indem Teile von Nachrichten verschlüsselt oder digital signiert werden. Für eine rasche Webdienst-Entwicklung stehen Werkzeuge zur Verfügung bei denen die Sicherheit lediglich konfiguriert werden muss. Dies hat zur Folge, dass der Inhalt der SOAP-Nachrichten unlesbar ist und die Entwicklerin oder der Entwickler das genaue Vorgehen der Verschlüsselung und der Signatur-generierung nicht mehr zu kennen braucht.

Die Applikation SoapAnalyzer hilft den Studierenden die Hintergründe der einzelnen Konfigurations-schritte nachzuvollziehen und zu verstehen, was die Entwicklungs-umgebung im Hintergrund macht. Mit ihr können verschlüsselte Inhalte von SOAP-Nachrichten entschlüsselt und Signaturen überprüft werden. Diese Vorgänge werden Schritt für Schritt zum besseren Verständnis dargestellt.

Ausgangslage

Das SOAP-Protokoll ist ein auf XML basiertes Nachrichtenformat. Durch die vom W3C vorgeschlagenen Security-Erweiterungen können Vertraulichkeit und Integrität bei der Übertragung erreicht werden.

Aktuelle Entwicklungsumgebungen (z. B. NetBeans IDE) unterstützen Entwicklerinnen und Entwickler bei der Implementation der Web Services soweit, dass sie die Sicherheit beim Web Service

lediglich konfigurieren müssen. Der technische Hintergrund und somit das Verständnis dafür, was der Web Server konkret macht (welche Verschlüsselung und/oder Signaturmethode verwendet wird), bleibt weitgehend verborgen.

Ziel der Thesis

In dieser Bachelor-Arbeit galt es eine interaktive Applikation zu entwickeln, welche die von einem Web Service generierten SOAP-Nachricht entschlüsselt und deren Sicherheitsaspekte (Vertraulichkeit und Integrität) sichtbar macht.

Realisierung

Da die potenziellen Benutzer unterschiedliche Betriebssysteme im Einsatz haben, ist es äusserst wichtig, dass die Applikation plattformunabhängig läuft. Daher wurde das Programm in Java 6 geschrieben.

Um die Benutzerfreundlichkeit zu steigern und weitere Darstellungs-

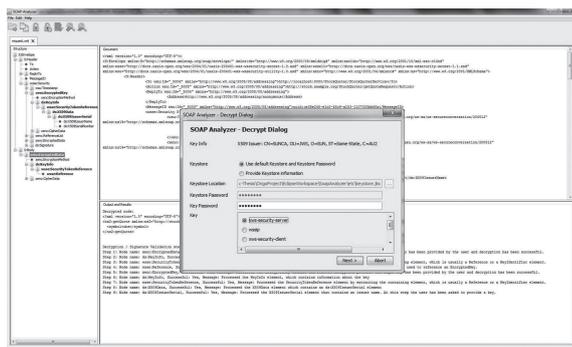
möglichkeiten zu erhalten, wurde neben einem Command Line-Interface auch eine grafische Benutzeroberfläche implementiert. Bedingt durch die Tatsache, dass der Web Service Security-Standard keine abschliessende Aufzählung der Verschlüsselungs- bzw. Signiermöglichkeiten aufweist, muss die Applikation eine einfache Möglichkeit für zukünftige Erweiterung anbieten. Dies wurde mittels eines Plugin-Systems realisiert.



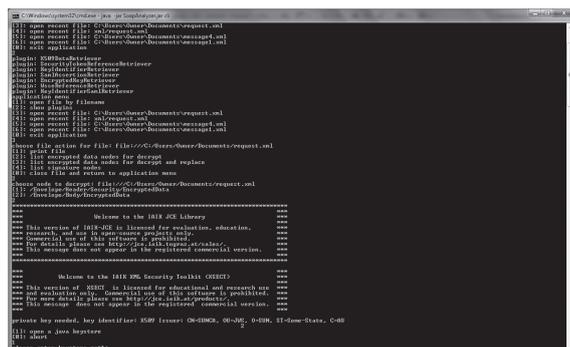
Urs Dix



Thomas Schmid



Grafische Benutzeroberfläche



Command Line Interface