Single-Sign-On mit Java Web-Applikationen

Kongress 2006

Dipl.-Wirtsch.-Inf. (FH)

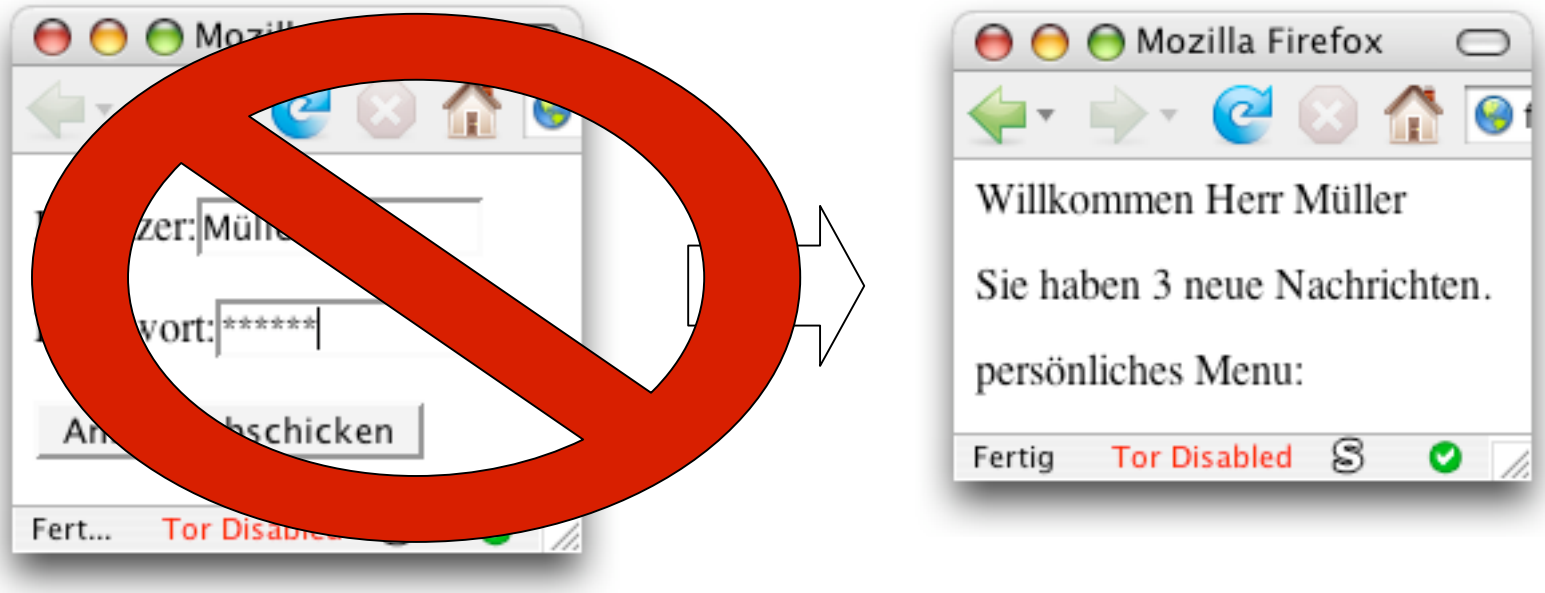
Michael Behrendt

<mb@michael-behrendt.net>

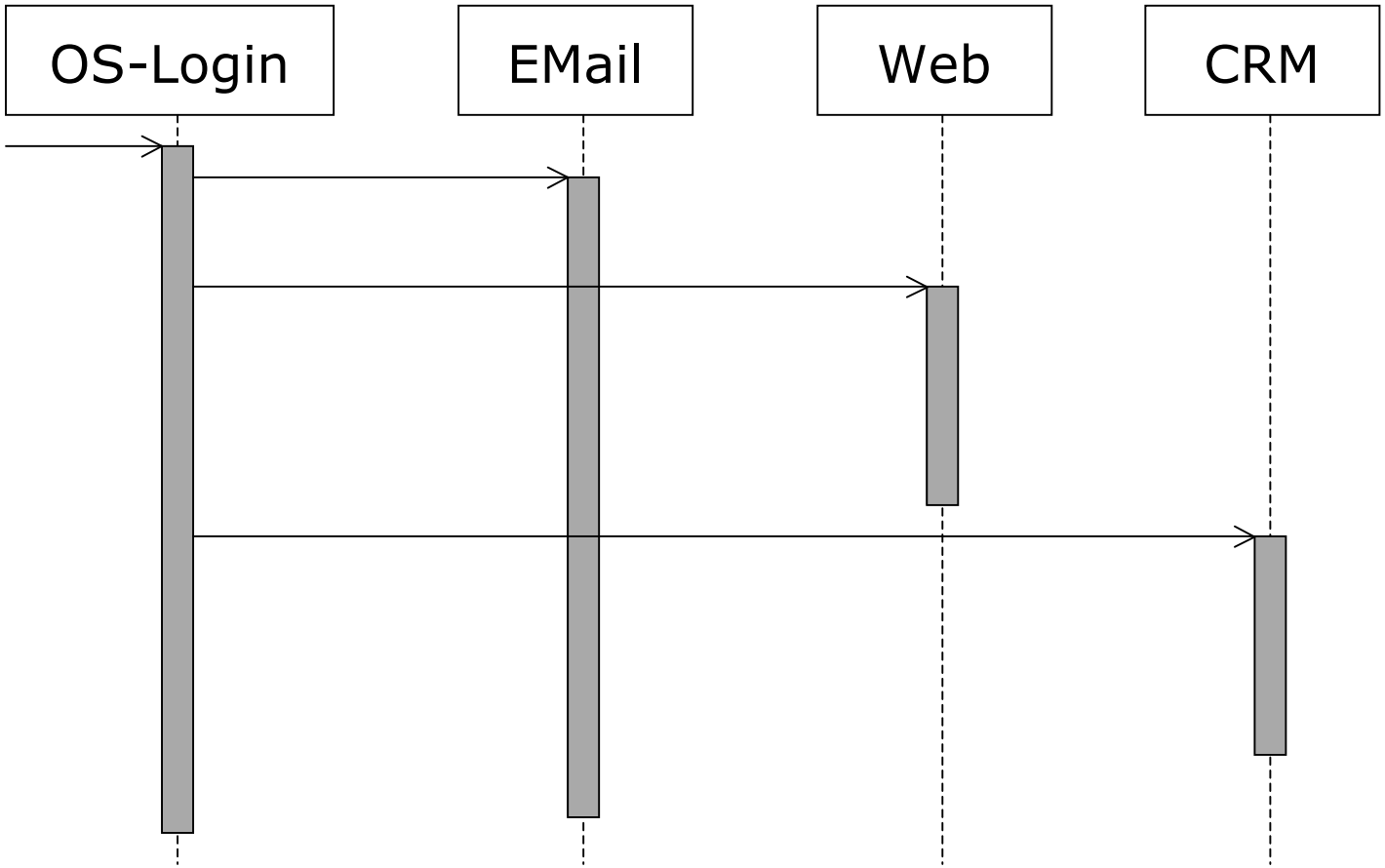
Inhalt

- Was ist Single-Sign-On?
- Protokolle
- Implementierung
  - Client
  - Active Directory Setup
  - Server (Java)

Ziel: Automatische Authentifizierung



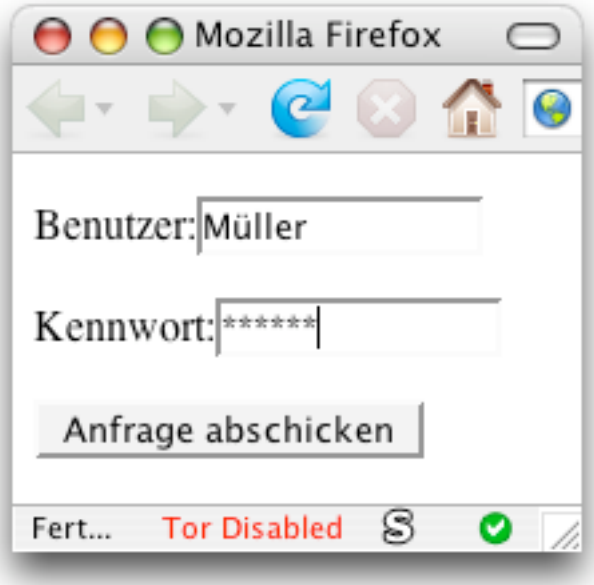
Idee: Weiterverwenden der Authentifizierung

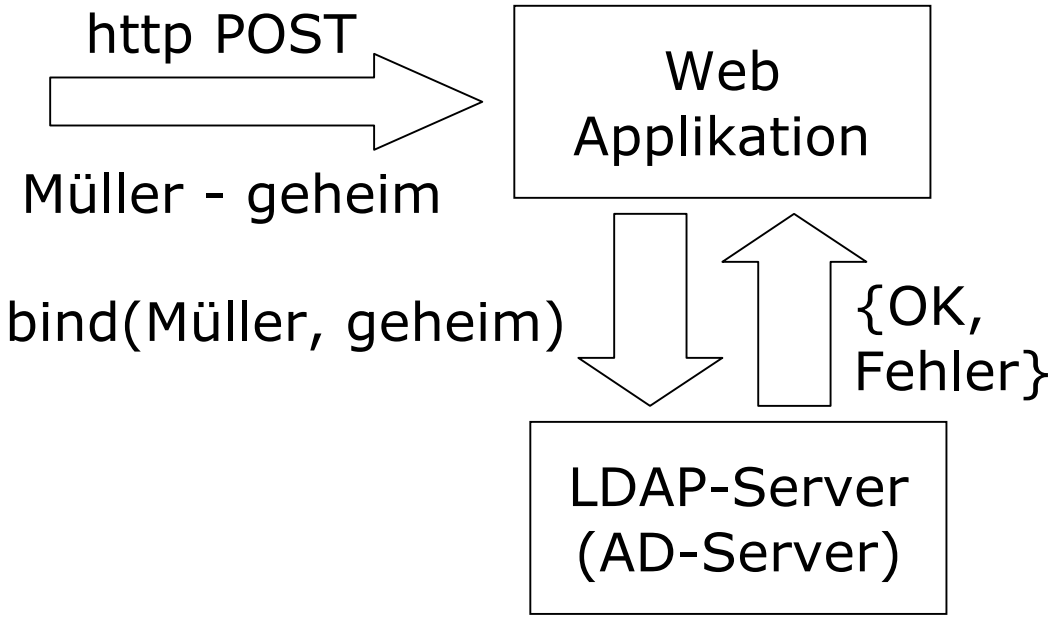


Benötigte Infrastruktur

- Netzwerk
  - Authentifizierung per Kerberos Server (z.B. Microsoft Active Directory)
  - Uhrzeitsynchronisation
- Arbeitsplatz
  - Rechner ist Mitglied in Kerberos Realm (= Windows Domäne)
  - Anwender verwendet Domänen-Login
- Web-Anwendung
  - Microsoft IIS/.NET (nativ)
  - Apache, Tomcat/Java (nachrüstbar)

Einheitliches Passwort ≠ Single-Sign-On





- Zentrale Passwortverwaltung
- Geheimnis wird mehrmals offen gelegt
- Sicherheitsprobleme nicht gelöst
- Lösung: Verzicht auf Passwort Authentifizierung

Protokolle

- Kerberos
  - Grundlage: Needham-Schroeder, 1978
  - nur symmetrische Kryptographie in Kerberos
  - Authentication Server kennt:
    - Encryption Keys der Benutzer
    - Encryption Keys der Anwendungen / Dienste
- SPNego
  - „Kerberos über http“
  - http Erweiterung (RFC4559)
  - Token Format (RFC4178)

Needham-Schroeder Protokoll

Client

Dienstname
Zufallszahl
Username

Authentication Server

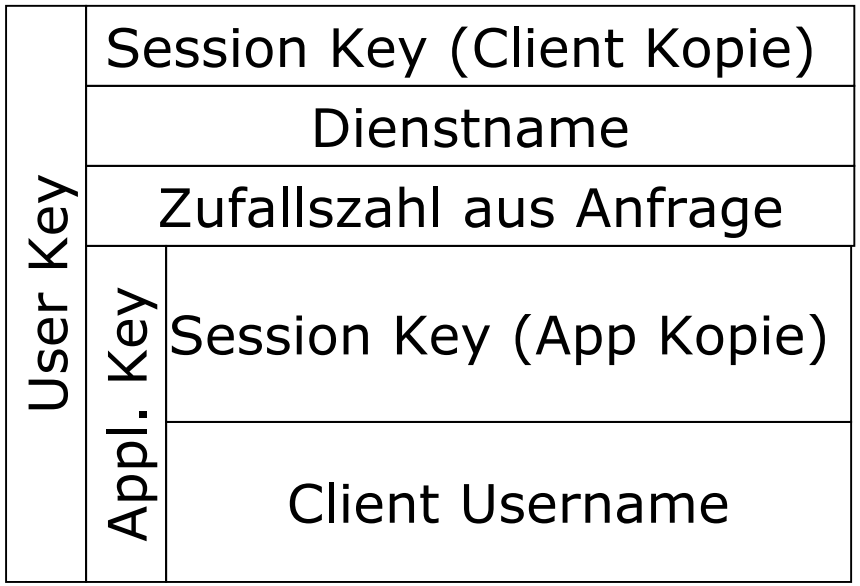
Application Server

Needham-Schroeder Protokoll

Client

Authentication Server

Application Server

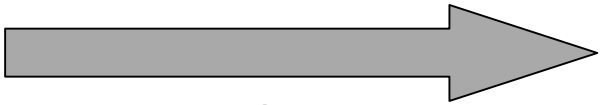


Needham-Schroeder Protokoll

Client

Authentication Server

Appl. Key	Session Key (App Kopie)
	Client Username



+ Anfrage

Application Server

http mit SPNego (RFC4559)

C: GET dir/index.html

S: HTTP/1.1 401 Unauthorized

S: WWW-Authenticate: Negotiate

C: GET dir/index.html

C: Authorization: Negotiate a87421000492aa874209af8bc028

S: HTTP/1.1 401 Unauthorized

S: WWW-Authenticate: Negotiate 749efa7b23409c20b92356

C: GET dir/index.html

C: Authorization: Negotiate 89a8742aa8729a8b028

S: HTTP/1.1 200 Success

S: WWW-Authenticate: Negotiate ade023456a42f8bc02889eca

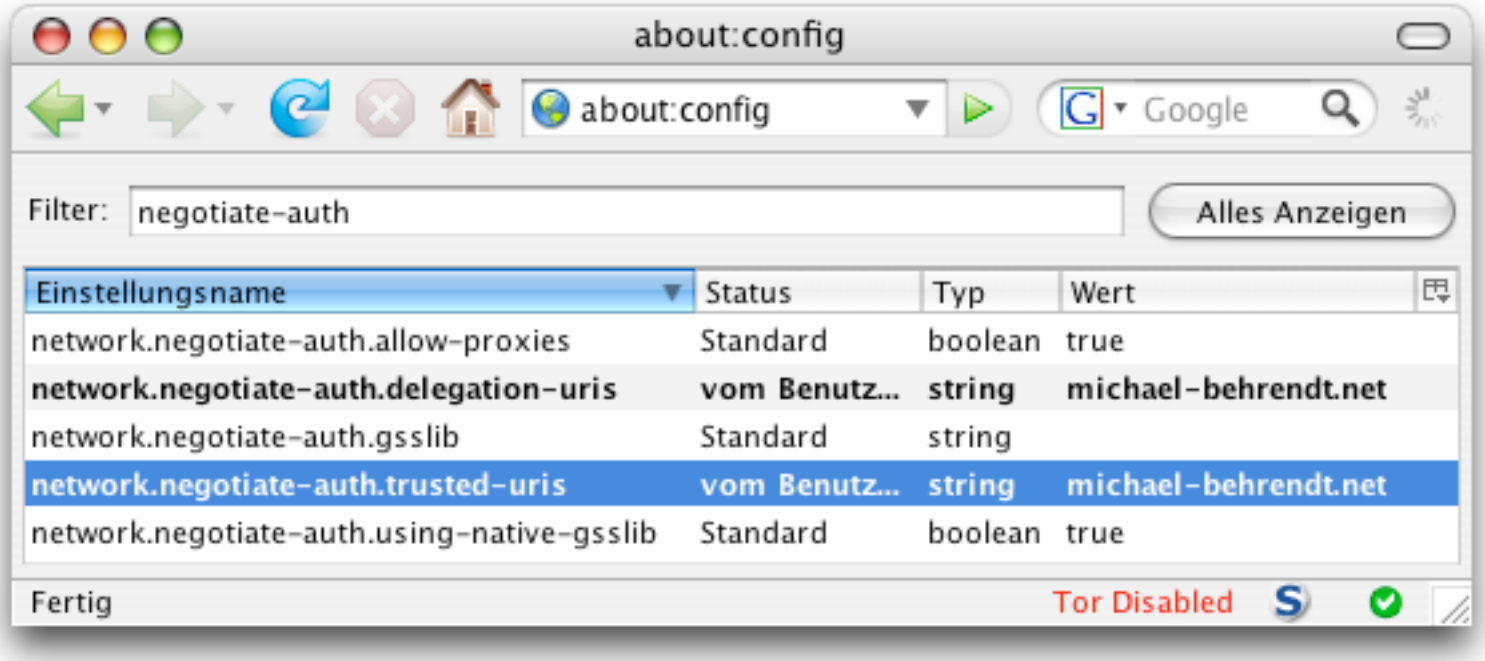
SPNego Token (RFC4178)

- ASN.1 Container
- NegotiationToken ::= CHOICE {
  negTokenInit [0] NegTokenInit,
  negTokenResp [1] NegTokenResp }
- NegTokenInit ::= SEQUENCE {
  mechTypes [0] MechTypeList,
  reqFlags [1] ContextFlags OPTIONAL,
  mechToken [2] OCTET STRING OPTIONAL,
  mechListMIC [3] OCTET STRING OPTIONAL,
  ... }
- NegTokenResp ::= SEQUENCE {
  ...
  responseToken [2] OCTET STRING OPTIONAL,
  ... }

Client Implementierung

- Internet Explorer
  - Zone „lokales Intranet“ > Erweitert > „*.michael-behrendt.net“ hinzufügen
  - Sicherheit > „Integrierte Windows-Authentifizierung aktivieren“ einschalten
- Firefox
  - network.negotiate-auth.trusted-uris
  - network.negotiate-auth.delegation-uris
  - jeweils „michael-behrendt.net“ hinzufügen

Client Implementierung - Firefox



Active-Directory Setup

- Benutzerkonto anlegen
  (Username = Hostname = z.B. www)
- Schlüssel exportieren

```
ktpass -princ HTTP/www.michael-
behrendt.net@AD.MICHAEL-BEHRENDT.NET
-mapuser www
-pass geheim
-out c:\temp\www.keytab
-crypto rc4-hmac-nt
```

- Keytab Datei sicher (!!!) übertragen

Kerberos Principal - User Mapping

- Ktpass erzeugt mapping Principal - User
- Auflisten der Mappings

```
setspn -l www
```

- Löschen eines Mappings

```
setspn -d HTTP/www.michael-
behrendt.net www
```

- Ktpass und setspn sind Bestandteil der Windows Support Tools (siehe suptools.msi auf Windows Installation Medien)

Server Implementierung

- Java Virtual Machine
  - Java Authentification and Authorization Service (JAAS)
  - Java Generic Security Services (JGSS)
  - SPNego Tokenverarbeitung
- SPNego Protokoll http-Erweiterung
  - Tomcat Valve
  - Servlet Filter
  - Anwendung

Java Authentification and Authorization Service (JAAS)

- JVM Parameter
  -Djava.security.auth.login.config=jaas.conf
- Beispiel

```
server {
com.sun.security.auth.module.Krb5LoginModule
required
useKeyTab=true
keyTab=www.keytab
storeKey=true
realm=AD.MICHAEL-BEHRENDT.NET
principal=HTTP/www.michael-behrendt.net@AD.MICHAEL-
BEHRENDT.NET
}
```

- „server" ist Lookup Key für Java-Code

Java Generic Security Services (JGSS)

- JVM Parameter
  -Djava.security.krb5.conf=krb5.conf
- Beispiel

```
[libdefaults]
default_realm=AD.MICHAEL-BEHRENDT.NET
default_tkt_enctypes= rc4-hmac
default_tgs_enctypes= rc4-hmac
[realms]
AD.MICHAEL-BEHRENDT.NET = {
    kdc = AD.MICHAEL-BEHRENDT.NET:88 }
[domain_realm]
.MICHAEL-BEHRENDT.NET = AD.MICHAEL-BEHRENDT.NET
MICHAEL-BEHRENDT.NET = AD.MICHAEL-BEHRENDT.NET
```

SPNego Tokenverarbeitung

- Bis Java 5
  - SPNego Token nicht unterstützt
  - Fremdanbieter
    - www.quest.com
    - www.appliedcrypto.com
    - evtl. www.bouncycastle.org (Open Source)
- Ab Java 6
  - Support für SPNego Tokens per Oid(„1.3.6.1.5.5.2“) anstelle Kerberos Oid(„1.2.840.113554.1.2.2“)
  - org.ietf.jgss.* unverändert

Literatur

- Jason Garman, Kerberos - The Definitive Guide, O'Reilly Verlag, 2003
- Advanced JGSS Security Programming, http://download.java.net/jdk6/docs/technotes/guides/security/jgss/lab/part5.html