SESAME

Authentifikationsportal für Web-Systeme

3.3.2006

Procondo GmbH
Ahrstr. 1
53757 Sankt Augustin

Zusammenfassung

Web-Anwendungen sind in vielen Unternehmen integraler Bestandteil von Geschäftsprozess-Implementationen und müssen daher bestimmte Sicherheitsanforderungen erfüllen. Als Mindestanforderung muss die Authentizität und Autorisierung der Anwendungsbenutzer geprüft werden - Handelt es sich um den vorgeblichen Benutzer und darf er die verlangte Aktion auslösen?

Traditionell setzt jede Anwendung ihr eigenes Authentifizierungssystem ein. Dies ist nur für eine kleine Zahl von Anwendungen handhabbar. Darüber hinaus verwaltet jede Anwendung eine eigene Datenquelle für Benutzerinformationen - getrennt von anderen Datenquellen über Firmenmitarbeiter. Die mehrfache Speicherung von Benutzerdaten führt schnell zu Abweichungen, zusätzlichem administrativen Aufwand und Sicherheitsproblemen. Die wiederholte Lösung des Authentifikations- / Autorisierungsproblems erhöht unnötig die Anwendungskomplexität und macht die Anwendungen größer, fehleranfälliger, schlechter testbar und nicht zuletzt unsicherer. Darüber hinaus bedeutet die Verwendung eigener Authentifikationsmechanismen, dass Benutzer gezwungen werden, sich mehrere Kennungen zu merken und sich beim Wechsel zwischen Anwendungen immer wieder erneut anzumelden.

Die geeignete Strategie zur Vermeidung der genannten Probleme besteht darin, die Benutzerverwaltung und Sicherheitsprüfung aus den Anwendungen herauszunehmen und an *zentraler Stelle* zu bündeln. Dieser Ansatz wird mit SESAME verfolgt. Daraus ergeben sich folgende entscheidende Vorteile:

- *Höhere Sicherheit:* Benutzer können authentifiziert werden, bevor sie überhaupt Zugang zum Zielsystem erlangen. Anwendungen, bei denen keinerlei Authentifizierung- / Autorisierung vorgesehen wurde, können nachträglich abgesichert werden.
- *Einheitliche Sicherheitsinfrastruktur:* Alle Anwendungen verwenden dasselbe System für die Authentifikation und Autorisierung. Zusätzliche Sicherheitsmechanismen können an zentraler Stelle eingefügt werden und erhöhen die Sicherheit aller Anwendungen gleichzeitig.
- *Zentralisierte Steuerung der Zugriffsrechte:* Der Zugriff auf Applikationen wird zentral an einer Stelle durchgeführt.
- *Konsistenz der Benutzerdaten:* Es wird nur eine Datenquelle verwendet und *zentral* verwaltet. Damit wird ebenso eine zentrale Vergabe von Zugriffsrechten ermöglicht.
- *Einmalige Anmeldung:* Hat sich der Benutzer authentifiziert, so kann seine Identität von Anwendung zu Anwendung weitergereicht werden, ohne dass eine erneute Anmeldung erforderlich wäre.
- *Anwendungsvereinfachung:* Die Autorisierungsmechanismen können aus der Anwendung entfernt werden, die Anwendung wird schlanker. Sicherheitstest der Anwendung und des Applikationsserver vereinfachen sich. Die Kosten der Anwendungsentwicklung verringern sich.

Für eine Web-Anwendung

- die eine große Nutzerzahl zufriedenstellen soll
- die hochverfügbar sein soll
- die große Rechenleistung erfordert

bietet es sich an, die Anwendung auf mehr als einer physikalischen Maschine laufen zu lassen, d.h. einen Maschinencluster einzusetzen. Damit lassen sich prinzipiell die Datenzugriffe der Benutzer auf die Maschinen verteilen und parallel abarbeiten.

Allerdings entsteht dadurch auch ein neues Problem: Wenn ein Benutzer zwei sequentielle Zugriffe auslöst, so können diese auf zwei unterschiedliche Maschinen verteilt werden. Wenn die Anwendung auf den Maschinen eigene Datenhaltung betreibt, müssten die Sitzungsdaten des Benutzers von der einen auf die andere Maschine transportieren werden, um die Zugriffe korrekt abarbeiten zu können.

Problemlösung: Für eine Anwendung, die einen derartigen Datentransport nicht unterstützen, bietet SESAME eine Sitzungsverteilung an. Diese garantiert, das Zugriffe von derselben Benutzersitzung stets derselben Maschine zugeteilt werden, so dass die Notwendigkeit des Datentransports entfällt. Verschiedene Sitzungen werden auf den Cluster verteilt, so dass sich insgesamt immer noch eine Verteilung der Last und eine Erhöhung der Verfügbarkeit ergibt.

Die Nutzung der Sitzungsverteilung kann auch für clusterfähige Anwendungen vorteilhaft sein, weil damit der Aufwand des Datentransports vermieden wird. Dies sollte allerdings durch eine Performanzmessung der speziellen Applikation ermittelt werden.

Als zentralisiertes Web-Sicherungssystem bringt der Einsatz von SESAME noch eine Reihe weiterer Möglichkeiten mit sich:

- Integration verschiedener Applikationsserver in einer gemeinsamen Domain durch Anpassung der URI
- Mandanten spezifische Form-Based-Authentication zur Anpassung an Regeln der Corporate Identity
- Logging an zentraler Stelle
- Nahtlose Integration in bestehende Umgebungen
- Erweiterbarkeit durch Verwendung offener Standards
- Weitergabe von Sitzungskennungen an die Applikationsserver
- Weitergabe der Benutzer-Credentials mit Hilfe von Basic-Authentication
- SSL-Terminierung
- Automatische Ermittlung der im Cluster verfügbaren Applikationsserver

Das vorliegende Dokument beschreibt die Funktionsweise von SESAME.

Inhaltsverzeichnis

1 Das Problem

1.1 Überblick

Sollen Web-Anwendungen in Geschäftsprozesse eingebunden werden, müssen Mechanismen für die Authentifikation und Autorisierung von Benutzern zur Verfügung gestellt werden. Das Protokoll http ist jedoch zustandslos. Das bedeutet, dass explizite Mechanismen für die Zuordnung zwischen Benutzer und Verbindungen zu einem Server (Session) geschaffen werden müssen. Auf der Basis des Session-Handling können entsprechende Verfahren für die Autorisierung von Benutzern umgesetzt werden.

1.2 Zu geringes Sicherheitsniveau

Befinden sich die Sicherungsmechanismen innerhalb der Anwendungen, müssen Verbindungen zu einem Server/einer Anwendung zugelassen werden, um überhaupt eine Authentifizierung der Benutzer vornehmen zu können. Wünschenswert ist jedoch, Benutzer vor Erreichen des Zielsystems abzuweisen, wenn diese nicht authentifiziert werden kann.

1.3 Hohe Komplexität der Sicherheitsinfrastruktur

Werden Authentifikation und Autorisierung von Benutzern von den Anwendungen vorgenommen, bedeutet dies, dass entweder sicherheitskritische Daten übertragen oder auf dem System gespeichert werden. In beiden Fällen müssen spezielle Sicherungsvorkehrungen getroffen werden.

1.4 Hohe Komplexität und Fehleranfälligkeit der Anwendungen

Meist verwenden insbesondere Web-Anwendungen eigene Mechanismen für die Authentifizierung und Autorisierung von Benutzern. Dies führt dazu, dass Anwendungen komplexer und fehleranfälliger werden.

1.5 Inkonsistenzen der Benutzerdaten

Ein weiteres Problem bei Verwendung von Anwendungslogik für die Authentifikation und Autorisierung ist, dass unter Umständen mehrere verschiedene Datenquellen für die Speicherung der Benutzerdaten verwendet werden. Inkonsistenzen in den Benutzerdaten sowie höhere Kosten bei der Administration der Benutzerdaten sind die Folge.

1.6 Hohe Betriebskosten durch hohe Belastung der Hotline mit Problemen bei der Benutzerverwaltung

Ein sehr hoher Anteil der eröffneten Calls in einer Hotline sind nach Aussagen vieler Firmen Probleme mit vielen verschiedenen Passwörtern, insbesondere bei Passwortänderungen oder vergessenen

Passwörtern. Das Call-Aufkommen und die damit verbundenen Kosten können erheblich gesenkt werden, wenn der Zugang zu den Systemen zentralisiert erfolgt und der Zugriff der Benutzer auf die Anwendungen in Abhängigkeit von deren Rollenzugehörigkeit kontrolliert wird. Innerhalb des Unternehmens verwendet ein Benutzer nur ein Passwort und läuft nicht Gefahr, das Passwort eines seltener benutzten Systems vergessen zu haben oder rechtzeitig zu ändern und belastet so die Hotline nicht mit Problemen aus dem Passwort-Umfeld.

Laut Studien verschiedener Firmen beträgt der Anteil der Calls aus dem Passwort-Umfeld 2-25 Prozent ([gartner]).

1.7 Überforderung des Benutzers

Benutzer sollten effektiv mit den Applikationen arbeiten können. Mit einem applikationsspezifischen Authentifikationsmechanismus wird ein Benutzer jedoch gezwungen, sich immer wieder erneut anzumelden.

Dies kann passieren, weil zwischen mehreren Applikationen während der Arbeit gewechselt wird, oder weil unterschiedliche Benutzerdaten verwendet werden müssen. Dadurch wird der Benutzer von seiner Arbeit abgelenkt und schließlich die Anzahl der eingesetzten Applikationen reduzieren. Bei einer zu hohen Zahl von Benutzerkennungen wird der Benutzer anfangen, Passwörter niederzuschreiben. Dadurch wird jede Sicherheitsstrategie beim Applikationszugriff ad absurdum geführt.

Nach einer Studie von IBM hat die einmalige Anmeldung in Kombination mit nur einer einzigen Benutzerkennung ein Einsparpotential von bis zu 670 Dollar pro Benutzer und Jahr ([NETC]).

2 Mögliche Lösungen

2.1 Abgleichen der Benutzerdaten

Für die eingesetzten Benutzerdaten-Quellen könnten Abgleichmechanismen entwickelt werden. Dies würde aber der vorhandenen Infrastruktur eine weitere sicherheitskritische Komplexitätskomponente hinzufügen. Daher ist es viel günstiger, nur eine zentrale Quelle für Benutzerdaten einzusetzen.

2.2 Verwendung eines applikationsübergreifend standardisierten Authentifizierungsmechanismus

Im Web-Umfeld ist Basic-Authentication ([RFC2617]) ein auf Client- und Serverseite weit verbreiteter Mechanismus zur Authentifizierung von Benutzern. Damit allein wird das Problem nicht gelöst, da eine Autorisierungskomponente fehlt. Deshalb müsste zusätzlich jeder Applikationsserver dieselbe zentrale Datenquelle zur Autorisierung anbinden. Dabei würden aber nicht autorisierte Benutzer bis zum jeweiligen Applikationsserver gelangen. Somit müssten mehrere Wege gesichert werden:

- die Verbindung des Benutzers zum jeweiligen Applikationsserver
- die Verbindung des Applikationsserver zur zentralen Datenquelle

2.3 Umsetzung einer J2EE-Infrastruktur

Genügen alle Anwendungen der J2EE-Infrastruktur, kann auf Standard-Mechanismen der Umgebung bzw. des jeweiligen Applikationsservers für die Authentifikation und Autorisierung zurückgegriffen werden. Anwendungen, die nicht dieser Architektur genügen, können nicht integriert oder müssen mit eigenen Mittel für die Authentifikation von Benutzern versehen werden.

3 Der Ansatz von SESAME

3.1 Erhöhung der Sicherheit durch Authentifikation *außerhalb* der zu überwachenden Zone

Benutzern soll nur dann der Zugang zu Systemen gewährt werden, wenn diese erfolgreich authentifiziert und autorisiert werden konnten. Wurde ein Benutzer jedoch nicht erfolgreich authentifiziert werden, muss dieser außerhalb der zu überwachenden Zone “abgefangen“ werden. Auf diese Art und Weise ergeben sich weniger Angriffsszenarien für die zu schützenden Anwendungen.

3.2 Reduktion der Komplexität der Sicherheitsinfrastruktur

Da die Speicherung der Benutzerdaten und der Zugang zu Systemen nur an wenigen Stellen erfolgt, müssen weniger Komponenten abgesichert werden.

3.3 Reduktion der Komplexität und Fehleranfälligkeit der Anwendungen

Um sowohl Daten- als auch Betriebssicherheit gewährleisten zu können, muss Robustheit der einzelnen Komponenten gewährleistet werden. Dies bedeutet, dass Komponenten, die von allen Anwendungen verwendet werden, nur ein einziges Mal implementiert aber mehrfach verwendet werden. Dieser Grundgedanke der Wiederverwendung ist eines der wesentlichen Paradigmen moderner objektorien tierter Softwareentwicklung. Die Wiederverwendung der Authentifikations- und Autorisierungsmechanismen von mehreren Anwendungen ist jedoch nur möglich, wenn die Verfahren für die Authentifikation und Autorisierung vereinheitlicht werden können.

3.4 Gewährleistung der Datenkonsistenz

Sesam ist ein den Applikationen vorgelagertes System, das für alle Applikationen einheitliche Sicherheitsmechanismen bietet. Es ist daher auf den Applikationsservern keine getrennte Datenquelle für Benutzerdaten erforderlich. Zusätzlich lässt sich Sesam via LDAP an ein zentrales System zur Benutzerverwaltung anbinden, wodurch nie mehr als eine Datenquelle für alle Benutzerdaten erforderlich ist. Die Konsistenz der Benutzerdaten ist daher gewährleistet.

3.5 Kostenreduktion der Benutzerverwaltung

Durch den vereinheitlichten Mechanismus müssen sich Benutzer nicht mehr mehrere Kennungen merken, was die Problematik des Aufschreibens/Vergessens von Kennwörtern entschärft. Einheitliche Anmeldeformulare werden vom Benutzer schneller gehandhabt. Da die Anmeldung für einen festsetzbaren Zeitraum unabhängig von der Anzahl der Applikationen, die ein Benutzer verwendet, nur einmal erfolgen muss, wird die Effizienz der Benutzer erhöht.

3.6 Vorteile von SESAME

- Höhere Sicherheit durch Authentifikation außerhalb der zu Überwachenden Zone
- Zentrale Steuerung von Zugriffsrechten mit Hilfe eines LDAP-Servers
- Integration verschiedener Web-Anwendungen in einer Domain durch Auswertung und Änderung der URIs
- Verwendung von Standard-Mechanismen für die Authentifikation
- Umsetzung verschiedener Corporate Identities für Login- und Fehlerseiten möglich (Mandantenfähigkeit)
- *Zentrales* Logging in Standard-Formaten
- Terminierung von SSL-Verbindungen
- Erweiterbarkeit durch Verwendung offener Standards

4 Skalierung von Web-Anwendungen

4.1 Einsatz paralleler Server (Cluster)

Wenn eine Web-Anwendung zu lange Antwortzeiten hat, ist ein naheliegender Gedanke, die Anwendung parallel auf mehreren Servern zu installieren, um so die Rechenleistung zu erhöhen und die Antwortzeiten zu verbessern. Lange Antwortzeiten können ihre Ursache in hohen Anforderungen an die Rechenleistung durch komplexe Operationen oder durch eine hohe Anzahl von Nutzern haben. Zunächst ist aufgrund der Zustandslosigkeit des http-Protokolls eine Verteilung auf mehrere Server gewährleistet. Dabei muss aber darauf geachtet werden, dass kein Engpass in der parallelen Bearbeitung besteht, z.B. weil auf eine gemeinsame nicht-skalierbare Datenbank zugegriffen wird.

Ein weiterer Anwendungsfalls für die Verwendung paralleler Server ergibt sich wenn redundante Server eine hohe Verfügbarkeit des Gesamtsystems gewährleisten sollen.

4.2 Problem: Datenhaltung einer Benutzersitzung

Eine natürliche Einschränkung der Parallelisierbarkeit ergibt sich aus den Datenabhängigkeiten der einzelnen Benutzeroperationen innerhalb einer Anwendung. Wenn zwei datenabhängige Operationen auf zwei unterschiedliche Maschinen verteilt werden sollen, muss die gemeinsame Datenbasis auf beiden Maschinen zugreifbar sein. Dadurch findet dann - zumindest teilweise - eine Sequentialisierung statt, die im Ergebnis den Geschwindigkeitsgewinn durch Parallelisierung wieder vernichten kann.

4.3 Lösung: Feste Bindung einer Benutzersitzung

Aus Anwendungssicht ist eine einfache Methode dies zu vermeiden, nicht die einzelne Benutzersitzung zu parallelisieren, sondern die Menge der Sitzungen. Dies erschein immer dann vorteilhaft, wenn der einzelne Benutzer üblicherweise zusammenhängende Geschäftsprozesse anstößt, während die Menge der Benutzer eher unabhängig voneinander an verschiedenen Geschäftsfällen gleichzeitig arbeiten.

Herkömmliche Lastverteiler bieten eine Zuordnung auf TCP/IP Ebene, nicht jedoch für das http-Protokoll, bei dem jeder Anwendungszugriff auch über eine neue TCP-Verbindung zulässig ist. SESAME hingegen unterstützt und hält eine derartige, am Sitzungsbeginn festgelegte Zuordnung zwischen Anwendung und Server.

5 Einfachere Benutzerverwaltung: Einmal-Anmeldung

5.1 Merkmale der Einmal-Anmeldung (Single Sign-On)

Ein wichtiges Merkmal von SESAME ist die Vereinfachung der Benutzerverwaltung durch einmalige Anmeldung. Im folgenden werden dazu einige Merkmale und Konsequenzen angeführt.

Die einmalige Anmeldung dient einmal der Bequemlichkeit des Nutzers. Zum Anderen dient sie aber auch der Sicherheit, indem sie vermeidet, dass der Benutzer zu einfache Passwörter verwendet oder das der Benutzer, da er sich nicht alles merken kann, die Passwörter auf unsicheren Medien niederschreibt.

Die Frage ist, wie weit eine einmalig Anmeldung reichen kann und soll ? Die Antwort darauf lässt sich nicht verallgemeinern, sondern hängt von den Sicherheitsrichtlinien des jeweiligen Unternehmenskreises ab, in dem Einmal-Anmeldung zum Einsatz kommt.

Die Anmeldung wird über eine Kette des Vertrauens unter den Anwendungen weitergereicht. Die Sicherheitsrichtlinien begrenzen diese Vertrauenskette an festgelegten Rändern.

Ein Beispiel für eine Vertrauenskette ist, wenn die Anmeldung am Betriebssystem des Benutzerrechners über den Browser weitergeleitet wird bis zu einer Web-Anwendung und deren Server.

Mechanismen zur Einmal-Anmeldung können sich dann in folgenden Merkmalen unterscheiden:

- Die Anmeldung kann von einer Web-Anwendung an eine andere weitergegeben werden. Dies ist die rudimentärste Form einer Einmal-Anmeldung. Sie lässt sich weiter dahingehend unterteilen, dass
  1. beide Anwendungen den gleichen Standard zur Einmal-Anmeldung verwenden.
  2. die beteiligten Anwendungen unterschiedliche Standards verwenden, so dass eine Umsetzung der Benutzer-Authentifikationen notwendig wird.
- Anmeldung über Domain-Grenzen hinweg. Eine Anmeldung bei einer Web-Anwendung für eine Internet Domain, kann auf eine andere Web-Anwendung in einer anderen Internet Domain übernommen werden.
- Anmeldung über Organisationsgrenzen hinweg. Eine Anmeldung kann von einer Organisation auf eine andere übernommen werden. Zur Unterscheidung gegenüber der Anmeldung über Domain-Grenzen gibt es hier zwei Fälle:
  1. Jede Organisation betreibt eine eigenes System zur Einmal-Anmeldung.
  2. Die beteiligten Organisationen betreiben mehrere eigene heterogene Systeme zur Einmal-Anmeldung.

5.2 Zentrales Benutzerverzeichnis (Directory)

Als Kernelement der Einmal-Anmeldung dient die zentrale Benutzerverwaltung. Analog zur Frage nach der Reichweite einer Einmal-Anmeldung, stellt sich hier die Frage der Reichweite beziehungsweise Zentralität der Benutzerdaten.

Einsatz mehrerer Verzeichnisse Hat eine Organisation schon Benutzerverzeichnisse im Einsatz, so ist zu überlegen, ob und mit welchen Mitteln daraus eine einheitliche Sicht für ein Verzeichnis aller Benutzer dargestellt werden kann. Vorgehensweisen sind: Zusammenführung vorhandener Verzeichnisse, Datenaustausch z.B. Replikation zwischen Verzeichnissen.

Eine ergänzende oder auch alternative Möglichkeit ist, der Einmal-Anmeldung den Zugriff auf verschiedene Verzeichnisse aufzuerlegen.

5.3 Systemarchitektur

Ein System zur Benutzerverwaltung besitzt als zentrale Infrastrukturkomponente einen enorm hohen Stellenwert. Fällt es aus, lässt sich schlimmstenfalls keine einzige Anwendung mehr nutzen. Ist es fehlerhaft, wird Angreifern ein Schlupfloch für Manipulationen von Geschäftsdaten geboten.

Im Folgenden sind daher einige Anforderungen an ein solches System zusammengestellt.

- *Benutzeranzahl.* Das System sollte die vorhandene Benutzeranzahl handhaben können und genügend Reserven für Erweiterungen bieten. Systemkomponenten sollten auf die Benutzerzahlen skalierbar ausgelegt sein.
- *Verfügbarkeit.* Das System sollte gegen Datenausfälle und Komponentenausfälle gesichert sein.
- *Sicherheit.* Verbindungen zwischen Systemkomponenten und Vertrauenspfade sollten durch Verschlüsselung abgesichert werden können.
- *Passwortverwaltung.* Das System sollte das Einstellen von Passwortrichtlinien gestatten, den Ablauf und das Sperren von Benutzerzugängen unterstützen.
- *Protokollierung.* Das System sollte ein Protokoll aller wichtigen Ereignisse im Zusammenhang mit Benutzeranmeldungen protokollieren: Anmeldung, Fehlversuche, Sperrung, Ablauf.

6 Technische Beschreibung

6.1 Überblick

In Abschnitt 6.2 wird die Implementation und Funktionsweise von SESAME beschrieben. Geplante Erweiterungen sind in Abschnitt 6.3 beschrieben.

6.2 Implementation

6.2.1 Arbeitsweise von SESAME

SESAME umfasst einen modifizierten Apache Webserver, mit zusätzlichen Modulen, sowie eine Benutzeroberfläche zur Konfiguration. Der Webserver wird als Reverse-Proxy betrieben.

Die Grundfunktion eines Reverse-Proxy besteht darin, dass URI auf verschiedenen Zielsysteme abgebildet werden können. Für den Benutzer erscheint der Reverse-Proxy als Quelle der URI, die Zielsysteme bleiben verborgen. Dadurch können Web-Applikationen, die auf unterschiedlichen Zielsystemen ablaufen, dennoch einen gemeinsamen Namensraum nutzen.

Als Beispiel kann ein Reverse-Proxy die URI

```
gate.mydomain.de/intranet/abc
```

auf

```
intranet.mydomain.de/abc
```

und

```
gate.mydomain.de/webserver/xyz
```

auf

```
webserver.mydomain.de/xyz.
```

umlenken.

Dadurch werden über das nach außen sichtbare System “gate.mydomain.de“ die unterschiedlichen Applikationen auf den Systemen “intranet“ und “webserver“ angesprochen, was aber nach außen hin verborgen bleibt.

Die Anfrage eines Clients (Browser) an SESAME wird vom Reverse-Proxy entgegengenommen und ausgewertet. Falls noch keine Session vorhanden ist, wird mit Hilfe eines Secure-Cookie zwischen dem Client und SESAME eine Session aufgebaut. Dann wird entschieden, ob ein Benutzer für das angegebene Ziel authentifiziert werden muss. Ist dies der Fall, wird der Benutzer auf ein Formular umgeleitet, in dem er seine Kennung und ein Passwort eingeben kann. Die Kennung sowie das zugehörige Passwort werden gegen einen LDAP-Server geprüft, in dem alle Benutzerinformationen gespeichert

sind. Die Benutzerdaten werden innerhalb der Session gespeichert. Bei vorhandener Autorisierung des Benutzers reicht der Reverse-Proxy die Anfrage mit Authentifizierungs- und Sessioninformation an das Zielsystem weiter, mit entsprechend modifizierter URI. Die Antwort des Zielsystems wird vom Reverse-Proxy entgegengenommen und an den Client zurückgereicht.

Der Aufbau einer Client Session erfolgt unabhängig von der Authentifizierung. So ist sichergestellt, dass eine Beobachtung (Tracking) des Benutzers ab der ersten Anfrage möglich ist. Sobald eine Authentifizierung erfolgt ist, werden die Sessiondaten um die Benutzerkennung erweitert. Ein Wechsel der Kennung während dieser Session ist nachverfolgbar.

Session-Handling zwischen Client und SESAME ist mit Hilfe eines Secure-Cookies implementiert ([RFC2109], [RFC2965]). Die Cookie-Informationen werden durch einen MD5-Hashwert gegen Manipulation und über eine AES-Verschlüsselung gegen Ausspähung gesichert.

Die Session zwischen Reverse-Proxy und Ziel-Server wird ebenfalls mit Hilfe von Cookies gesteuert. Dieser Cookie verbleibt auf dem Reverse-Proxy und wird *nicht* an den Client weitergeleitet. Die Benutzerinformationen (“Kennung“ und “Passwort“) werden mit Hilfe von Basic-Authentification-Mechanismen ([RFC2617]) an die Web-Anwendung weitergegeben.

Das Formular zur Eingabe von Benutzerkennung und Passwort kann je nach Mandant unterschiedlich gestaltet sein. Zur Absicherung der übertragenen Benutzerdaten wird zweckmäßig eine https-Verbindung verwendet.

Öffnet ein Benutzer ein weiteres Browser-Fenster und baut eine Verbindung zu SESAME auf, so erkennt SESAME dies und baut keine neue Session auf. Das heißt, Benutzerkennung und Passwort müssen nur einmal eingegeben werden, die Session wird solange aufrecht erhalten, bis der Benutzer den Browser schließt und eine in der Konfiguration von SESAME festgelegte Zeitspanne verstreicht. Ist zum Beispiel als Verfallszeit für den Cookie zehn Minuten eingestellt, muss ein Benutzer sich nicht neu anmelden, wenn er innerhalb von zehn Minuten den Browser wieder öffnet und auf die durch SESAME kontrollierten Seiten zugreift.

Die Abbildung 1 skizziert die Arbeitsweise von SESAME.

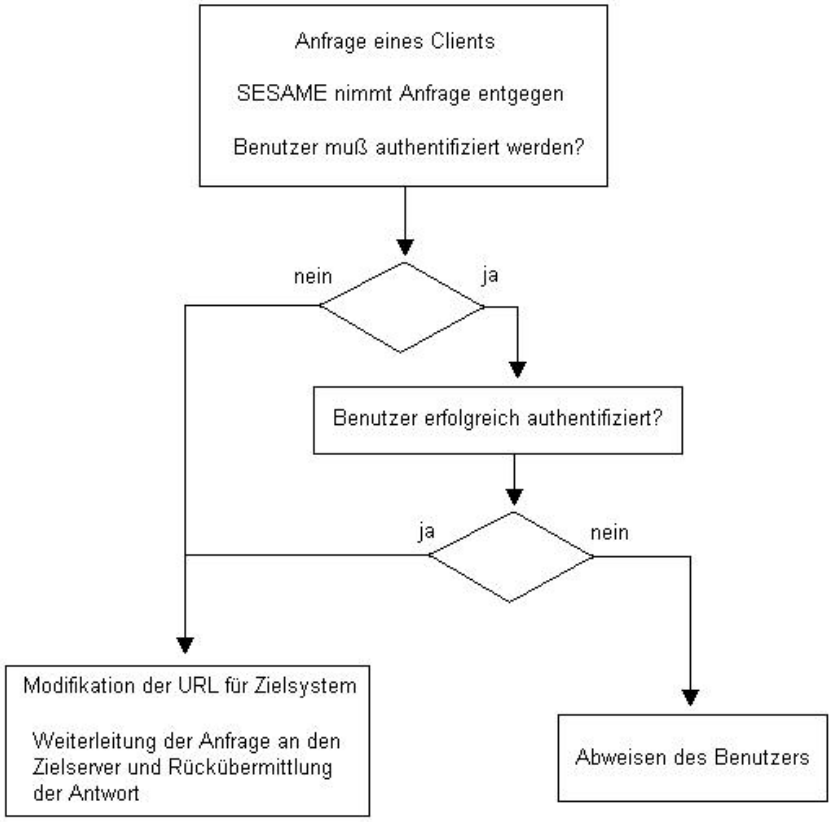

Abbildung 1: Ablauf der Kommunikation bei der Authentifizierung

6.2.2 Log-Funktionalität

SESAME protokolliert keine Dateninhalte, sondern nur Informationen über den Auf- und Abbau von Verbindungen sowie Ausnahme- und Fehlersituation. Das Log-Format von SESAME orientiert sich am Common-Log-Format. Standardmäßig werden Verbindungsinformationen und Fehlermeldungen jeweils in zwei getrennten Dateien gespeichert. Der Name und der Pfad dieser Dateien lässt sich in der Konfigurationsdatei von SESAME festlegen. Es ist auch möglich, die Informationen in einer einzelnen Datei zu protokollieren, auf mehr als zwei Dateien zu verteilen oder von einem Analyseprogramm nachbereiten zu lassen.

Folgende Nachrichtentypen werden in den Protokolldaten unterschieden:

- Warnung
- Fehlermeldung
- Kritischer Fehler
- Information
- Protokoll

- Debug-Meldung

Vordefinierte Protokoll-Level definieren die Menge an Informationen, die protokolliert werden soll. Der Information-Level zum Beispiel berücksichtigt alles bis auf die Protokoll- und Debug-Meldungen. Im Debug-Level werden alle Meldungen protokolliert.

6.3 Geplante Erweiterungen

- Unterstützung weiterer Plattformen (AIX, Windows 2000, HP-UX)
- Implementation für Apache 2.0
- Integration eines Web-Frontends für die Verwaltung von Benutzern, Benutzergruppen sowie deren Zugriffsrechten
- Unterstützung für Selbstanmeldung neuer Benutzer
- Integration weiterer Anwendungen und Infrastrukturen:
  - Anmeldung an Windows-Domäne (NTLM)
  - Anmeldung an Lotus Notes
  - Anmeldung via Chipkarte

7 Systemvoraussetzungen

- Betriebssystem
  - Linux ab Kernel-Version 2.2
  - Solaris ab Version 8
- Hardware für Linux-System oder Windows-System
  - Pentium III, 700 MHz
  - 256 MB RAM
  - 3 GB Festplatte
- Hardware für Solaris-System
  - Sun Ultrasparc 5
  - 256 MB RAM
  - 3 GB Festplatte

8 Funktionalität von SESAME in Stichworten

- *Zentralisierte Authentifikation* am Rande einer DMZ
- *Zentrale* Steuerung der Zugriffsrechte mit Hilfe eines LDAP-Servers
- *Zentrales* Logging
- Nachträgliche Absicherung ungesicherter Applikationen
- Integration verschiedener Server-Systeme in einer Domain durch URI Anpassung (Reverse-Proxy)
- Authentifizierung über Basic-Authentication oder mandantenspezifische Form-Based-Mechanismen
- Session Tracking
- Weitergabe von Benutzer-Credentials und Session an die Zielserver
- SSL-Terminierung

Literatur

[apache] Apache Web-Server http://www.apache.org

[gartner] Attacking Recurring Calls: How To, Password, Service GartnerGroup RAS Services, DF-06-0587, 12 November 1998

[NETC] The New Face of Single Sign-On, Philip Carden Network Computing, March 22, 1999

[openssl] OpenSSL http://www.openssl.org

[RFC2109] HTTP State Management Mechanism, D. Kristol et al., Network Working Group, February 1997

[RFC2617] HTTP Authentication: Basic and Digest Access Authentication, J. Franks et al., Network Working Group, June 1999

[RFC2965] HTTP State Management Mechanism, D. Kristol et al., Network Working Group, October 2000