



Zentrum für sichere Informationstechnologie – Austria Secure Information Technology Center – Austria

A-1040 Wien, Weyringergasse 35
Tel.: (+43 1) 503 19 63-0
Fax: (+43 1) 503 19 63-66

A-8010 Graz, Inffeldgasse 16a
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

<http://www.a-sit.at>
E-Mail: office@a-sit.at

SPEZIFIKATIKON PAMID+FOLB

PROXYTUNNEL AUTHENTISIERUNG MITTELS MOA-ID

INCL. FAIL-OVER + LOADBALANCING

AKTUALISIERTE VERSION VOM 30.12.2004

Autor: Christian Rechberger

eMail: Christian.Rechberger@iaik.tugraz.at

Zusammenfassung:

Ziel des Projektes PAMID ist es, die Verwendung von Proxytunnel mittels MOA-ID zu authentisieren und damit die Sicherheit und Handhabbarkeit zu verbessern. Zusätzlich wird die Proxytunnellösung um Möglichkeiten der Lastverteilung und der Ausfallsicherheit erweitert.

Kern der Lösung ist ein *Connection Admission Control Modul*. Weiters sind dabei kleine Erweiterungen von Proxytunnel Client und -Server (inkl. config-files) notwendig. Die Abwärtskompatibilität ist dabei jedoch gewährleistet. Neben einer Schritt-für-Schritt Beschreibung des verwendeten Protokolls werden auch die einzelnen Module und deren Kommunikation untereinander im Detail beschrieben.

Inhaltsverzeichnis

| | |
|--|---|
| Inhaltsverzeichnis | 1 |
| 1. Aufgabenstellung | 2 |
| 2. Architektur der Lösung für MOA-ID Anbindung | 2 |
| 2.1. Protokoll | 2 |
| 2.1.1. Beschreibung | 2 |
| 2.2. Analyse | 3 |
| 2.2.1. Sicherheitstechnische Betrachtung dieser Herangehensweise | 4 |
| 3. Architektur der Lösung für Fehlertoleranz und Lastverteilung | 4 |
| Ad Schritt 1: | 5 |
| Ad Schritt 2 und 3: | 5 |
| Ad Schritt 4: | 5 |
| Ad Schritt 5: | 5 |
| 3.1. Entscheidung über Auswahlstrategie | 5 |
| 3.2. Verhalten im Fehlerfall | 5 |
| 4. Spezifikation der Schnittstellen | 6 |
| 4.1. PT Client , PT Server ↔ CAC-Modul | 6 |
| 4.2. Browser ↔ CAC-Modul | 6 |
| 4.3. CAC-Modul ↔ MOA-ID AUTH | 7 |
| 5. Literatur | 7 |

1. Aufgabenstellung

Bisher wird die Verwendung des Proxytunnel nur über vor-ausgestellte Clientzertifikate authentisiert. Aufgrund einer Anfrage vom BMGF im Dez. 2003 ist es das Ziel des Projektes PAMID, die Verwendung von Proxytunnel mittels MOA-ID zu authentisieren und damit die Sicherheit und Handhabbarkeit zu verbessern. Bei der Erweiterung der bestehenden Proxytunnelkomponenten (Client und Server) ist auf Abwärtskompatibilität zu achten. Außerdem soll die verwendete Architektur möglichst unabhängig vom verwendeten Tunnelungsprotokoll sein.

2. Architektur der Lösung für MOA-ID Anbindung

Die realisierte Lösung löst 2 Knackpunkte, an denen sich etliche Ansätze zu messen hatten:

- Verknüpfung von Proxytunnel Verbindung mit MOA-ID Handshake
- Tunneln des MOA-ID Handshakes

Kern der Lösung ist ein *Connection Admission Control Modul* (nachfolgend CAC-Modul genannt).

2.1. Protokoll

Hier wird MOA-ID in die Authentisierung integriert. Die realisierte Lösung basiert auf folgenden Eckpfeilern:

- Web Browser wird verwendet (für BKU-Auswahl und MOA-ID Protokoll)
- Kleine Erweiterungen von PT-Client und PT-Server (+config-files) sind notwendig, Abwärtskompatibilität jedoch gewährleistet.
- Verknüpfung von Proxytunnel Verbindung mit MOA-ID Handshake wird mittels Geschäftsbereich-Angabe (Target) realisiert
- Tunneln des MOA-Protokolls wird über neue Features von MOA-ID 1.2 realisiert (derzeit noch Developer-Version)

2.1.1. Beschreibung

Der Authentisierungsvorgang ist in folgende Teilschritte aufgeteilt

1. SSL – Handshake zwischen PT-Client und PT-Server, wobei unter anderem das Zertifikat des PT-Clients zum PT-Server übertragen wird.
2. PT-Client startet Browser mit CAC-Modul URL (localhost, da über vordefiniertes Port) falls Verbindung noch nicht authentisiert.
3. Dabei wird als HTTP GET Parameter die ID (Zertifikats-Seriennummer oder SSL Session-ID) und weitere Angaben zu verwendeten Client- und Serversockets (je nachdem ob der gesamte Tunnel oder einzelne Verbindungen authentisiert werden sollen) übertragen.
4. CAC-Modul antwortet dem Browser mit StartAuthentication-Link auf MOA-ID AUTH (localhost, da getunnelt über vordefiniertes Port), wobei hier als *OA* die localhost CAC-Modul URL angegeben wird und als *Target* ein Datum, welches aus Clientzertifikat sowie Client- und Serversocketinformationen abgeleitet wird.
5. Browser sendet diese Informationen über den Tunnel zu MOA-ID AUTH (eventuell ohne Benutzer-Interaktion mittels http-Redirect)
6. MOA-ID AUTH lenkt Browser auf BKU-Auswahl-Seite

nicht gegeben ist kann als Alternative die SSL Session-ID verwendet werden. Diese ist für jeden Tunnel eindeutig und sowohl dem Client als auch dem Server bekannt. Damit nicht jeder einzelne Verbindungsaufbau innerhalb eines SSL-Tunnel neu authentisiert werden muss, kann SSL Session-Resuming verwendet werden. Die Bedingung, das SSL-Protokoll für diesen Zweck nicht zu modifizieren, ist weiterhin erfüllt. Außerdem sind Clients hinter NAT-Boxen weiterhin problemlos.

2.2.1. Sicherheitstechnische Betrachtung dieser Herangehensweise

Um einen Tunnel zuzulassen prüft der Proxytunnel Server mit Hilfe des CAC-Modules, ob zur aktuellen Session-ID eine gültige Authentisierung vorliegt. Die Authentizität dieser Information wird durch MOA-ID AUTH gewährleistet.

Die SSL Session-ID wird während des Handshakes unverschlüsselt übertragen. Dies nützt einem potentiellen Angreifer jedoch nichts, da er für ein SSL Session-Resume auch das während des ersten Handshakes aufgeteilte Mastersecret benötigt, welches jedoch nur der authentifizierte Client besitzt.

3. Architektur der Lösung für Fehlertoleranz und Lastverteilung

Nachfolgend eine Darstellung der Lösung.

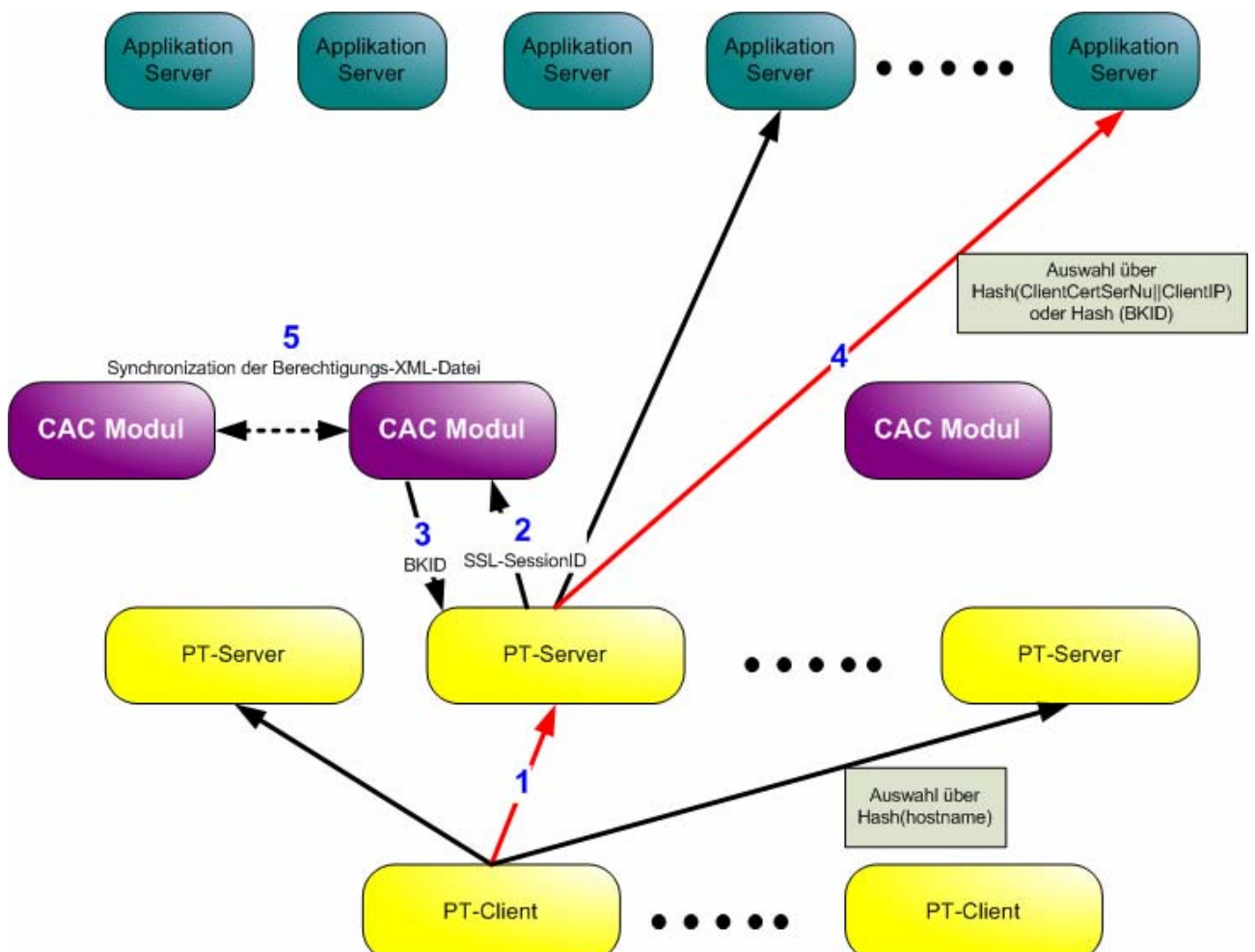


Figure 1 Darstellung eines Verbindungsaufbaues

Ad Schritt 1: Unter der Annahme, dass der hostname des Client-Rechners eindeutig ist wird die Funktion $\text{hash}(\text{hostname}) \bmod \#\text{PTServer}$ verwendet um einen PT-Server auszuwählen.

Ad Schritt 2 und 3: Für jeden PT-Server ist ein dazugehöriges CAC-Modul konfiguriert (vorzugsweise am gleichen Rechner). Aus Last-gründen würde auch ein einziges CAC-Modul für alle PT-Server reichen, Ausfallsicherheit wäre dadurch jedoch nicht gewährleistet (single point of failure). Um nun eine dem Client zugeordnete Zahl als Auswahlkriterium für die Applikationsserver zu erhalten wird zwischen PT-Server und CAC-Modul ein einfaches Protokoll abgearbeitet. Die SSL-SessionID, welche als sichere Verknüpfung zwischen SSL-Handshake des Tunnels und MOA-ID Authentisierung verwendet wird, wird im CAC-Modul als Schlüssel für eine Tabelle verwendet. Darin wird nun eine ID abgelegt, welche uah. von der SessionID für jeden Client(bzw. jede Bürgerkarte) gleich ist.

Ad Schritt 4: Die Auswahl einer der zur Verfügung stehenden Applikationsserver kann nun mit der Formel $\text{hash}(\text{BKID}) \bmod \#\text{AppServer}$ erfolgen. Wird zur Clientauthentisierung nicht MOA-ID verwendet kann ersatzweise die Formel $\text{hash}(\text{ClientCertSerNo}||\text{ClientIPAddress}) \bmod \#\text{AppServer}$ verwendet werden um ebenfalls eine Clientabhängigkeit einzubringen, die entweder von einem eindeutigen Clientzertifikat (falls vorhanden) oder von der Client-IP-Adresse herrührt.

Ad Schritt 5: Für den Fall einer Konfigurationsänderung des CAC-Modules (Änderung der Berechtigungen) muss für eine Synchronisation mit den anderen CAC-Modulen gesorgt werden. Hierbei geht es um das kopieren einer XML-Datei und ev. dem Neustart des Servlets. Je nach Betriebssystem können hierzu scheduled tasks, cron jobs oä. herangezogen werden.

3.1. Entscheidung über Auswahlstrategie

Da es sich bei PT-Client und PT-Server grundsätzlich um das gleiche Produkt handelt, wird das Verhalten nur über Konfigurationsdateien gesteuert. So kann nun das Verhalten bei der Auswahl über diese Konfigurationsdatei gesteuert werden. Die Details dieser Konfiguration werden in [3] näher diskutiert.

3.2. Verhalten im Fehlerfall

Kann zu einem Rechner (PTServer bzw. AppServer) keine Verbindung aufgebaut werden, wird nach einem round-robin Verfahren der nächste ausgewählt und erneut ein Verbindungsaufbau versucht. Falls ein Fehler während einer Verbindung passiert, muss diese erneut aufgebaut werden. Alle weiteren connect-Versuche beginnen mit dem zuletzt erfolgreich verwendeten Host. Nur ein Neustart (PTClient oder PTServer) setzt dieses Verhalten zurück.

Falls jedoch der PTServer verschiedene AppServer kontaktieren muss um eine Verbindung aufzubauen, muss der PTClient keine neue Verbindung aufbauen, es ergibt sich nur eine längere Wartezeit bis zum Verbindungsaufbau.

4. Spezifikation der Schnittstellen

4.1. PT Client , PT Server ↔ CAC-Modul

PT Client (Schritt 2):

Um beim Initiieren eines Tunnels vorab zu Prüfen, ob dieser schon vorher authentisiert wurde, wird der Authentisierungszustand beim CAC-Modul abgefragt. Die Abfrage wird durch einen unauthentisierten Tunnel über den PT Server durchgeführt. Dadurch wird die serverseitige Überprüfung jedoch nicht redundant.

PT Server (Schritt 15):

Nachdem das MOA-ID Protokoll abgearbeitet wurde liegen die Authentisierungsdaten im CAC-Modul vor. Der Proxytunnel Server wartet in der Zwischenzeit (bis zu einem Timeout) und fragt in periodischen Abständen das CAC-Modul ab.

| Request | |
|---------------|--|
| Typ: | HTTP GET über SSL (Proxytunnel) bei PT Client HTTP GET bei PT Server (HTTPS optional) |
| Parametername | Erläuterung |
| version | Versionsnummer "1" |
| Id | Seriennummer des Proxytunnel Clientzertifikats oder SSL Session-ID |
| sourceIP | IP des PT Clients |
| destIP | IP des PT Servers |

Beispiel http://localhost:18081/PAMID_CAC/PamidStatus?certNo=250449923341139&tunnelName=Mailtunnel&sourceIP=129.27.152.94&destIP=129.27.152.40

| Response | |
|--------------------------------|---|
| Typ: | HTTP Response Body |
| Inhalt | Erläuterung |
| | <BKID> beschreibt einen numerischen Wert. Inhalt: Die vom PAMID_CAC Modul errechnete Kennzahl der Personenbindung. |
| BKID <BKID> Status <Status> | Bei „Status OK“ im Body wurde Authentisierung positiv bestätigt. Bei „Status Timeout“ war der Tunnel schon authentisiert, die Timeout ist jedoch abgelaufen und der Tunnel muss daher neu authentisiert werden. Bei jedem anderen Response wird auch neu authentisiert. |
| Version <nr> | Versionsnummer nr=1 |

4.2. Browser ↔ CAC-Modul

Ablauf der Kommunikation zwischen Browser und CAC-Modul um den Tunnel zu authentisieren. (entspricht Schritt 3 und 4)

| Request | |
|---------------|--|
| Typ: | HTTP GET über SSL (Proxytunnel) |
| Parametername | Erläuterung |
| id | Seriennummer des Proxytunnel Clientzertifikats |

| | |
|------------|--|
| | oder SSL Session-ID |
| tunnelName | Bezeichner des zu initierenden Tunnels |
| sourceIP | IP des PT Clients |
| destIP | IP des PT Servers |

Beispiel http://localhost:18081/PAMID_CAC/PamidCac2?id=250449923341139&tunnelName=Mailtunnel&sourceIP=129.27.152.94&destIP=129.27.152.40

| Response | |
|----------|---|
| Typ: | HTTP Redirect über SSL (Proxytunnel) |
| Inhalt | Erläuterung |
| | Redirect auf SelectBKU von MOA-ID Als Target werden die identifizierenden Parameter (ID) angegeben |

Beispiel http://localhost:18080/moa-id-auth/SelectBKU?Target=id:%208B:F8:A1:AC:2B:78:43:14:A9:C0:61:60:4E:2C:8F:B6&OA=http://localhost:18081/PAMID_CAC/PamidCac2_auth&Template=http://129.27.152.40:48080/templates/template_pamid.html&BKUSelectionTemplate=http://129.27.152.40:48080/templates/bkutemplate_pamid.html

4.3. CAC-Modul ↔ MOA-ID AUTH

Diese Schnittstelle wird gemäß [2] Punkte 4.3 – 4.5 umgesetzt.

5. Literatur

- [1] **Installing and Using the IAIK Proxytunnel**, 1999. Available from docs\userdocs.pdf
- [2] R. Schamberger, G. Karlinger, L. Moser: **Spezifikation MOA-ID 1.2**. 2004
- [3] C. Rechberger: **Projektdokumentation PAMID+FOLB 1.2**. 2004