Smartcard-Authentifizierung mit Oracle-Forms

Dr. Peter Koch
Nordrheinische Ärzteversorgung, Düsseldorf

Dr. Peter Alteheld
MT AG, Ratingen

Schlüsselworte

Smartcard, X509-Zertifikate, RSA, HTTPS, MS-GINA, SSH, SAP-SNC, Wallet, Oracle Internet Directory, Single Sign-On

Zusammenfassung

Um den steigenden Sicherheitsansprüchen der Nordrheinischen Ärzteversorgung gerecht zu werden, wurde das Passwort-basierte Anmeldeverfahren ersetzt durch ein Authentifizierungsverfahren, das die Nutzung aller vorhandenen Anwendungen nur bei Vorhandensein einer durch PIN-Eingabe freigeschalteten Smartcard im Kartenleser des Anwenders erlaubt. Erforderlich waren neben der Anmeldung an Windows-2000 Arbeitplätzen, SAP-Anmeldung, Browser-Nutzung, Email-Abruf und SSH Verbindungsaufbau auch die Anmeldung mit Oracle Forms-Applikationen. Der Vortrag enthält einen kurzen Überblick über die Grundlagen von 2-Faktor-Authentifizierungsverfahren und beschreibt die letztendlich zur Lösung eingesetzte Hard- und Software. Es wird kurz darauf eingegangen, warum die Smartcard-basierte Authentifizierung unter den vorhandenen 2-Faktor-Authentifizierungsverfahren ausgewählt wurde. Im Anschluss an den theoretischen Teil des Vortrages wird die Lösung vorgeführt.

Authentifizierungsverfahren

Authentifizierungsverfahren dienen dazu die Identität eines Benutzers festzustellen. Hierbei befindet sich in der Regel die Person P, die sich gegenüber einem System S authentifizieren möchte, nicht am gleichen Standort wie das System S. Üblicherweise erfolgt der "Beweis" der Identität dadurch, dass Person P ein Passwort eingibt, das an das System S übermittelt und dort auf Korrektheit überprüft wird.

Aus der Tatsache, dass beim System S ein korrektes Passwort ankommt, kann das System S jedoch nicht wirklich sicher schließen, dass dieses Passwort von der korrekten Person eingegeben wurde. Prinzipiell kann es von jeder anderen Person stammen, die durch Ausspähen oder Erraten des Passwortes in dessen Kenntnis gelangt ist. Auch ist nicht einmal sicher, ob das Passwort überhaupt zum Zeitpunkt der Authentifizierung von einer Person eingegeben wurde. Es kann sich auch um einen Wert handeln, der (lange) vorher erfasst wurde und nun (maschinell) übermittelt wurde, während sich die Person P längst irgendwo anders befindet.

Diese Probleme sollen durch sogenannte 2-Faktor-Authentifizierung vermieden werden. Bei dieser Authentifizierung erfolgt die Identifizierung von Person P dadurch, dass P einen einmaligen Gegenstand besitzt. Das zusätzlich zu verwendende Passwort dient im wesentlichen dazu, einen Missbrauch des Gegenstandes im Verlustfall zu vermeiden. Damit lassen sich die meisten Probleme der Passwort Authentifizierung vermeiden. Voraussetzung hierbei ist aller-

dings, dass es sich bei dem verwendeten Gegenstand wirklich um ein Unikat handelt, d. h. es lässt sich mit vertretbarem Aufwand nicht duplizieren, und der Besitz dieses Gegenstandes kann von einer Person P gegenüber einem entfernten System S verlässlich auf elektronischem Wege bewiesen werden. Leider sind genau diese Voraussetzungen bei erschreckend vielen Systemen nicht gegeben, wie die folgenden Beispiele zeigen:

2-Faktor-Authentifizierung mittels biometrischer Merkmale

Verwendet man als einmaligen Gegenstand ein einmaliges Körperteil (Fingerabdruck, Augenhintergrund, Gesichtszüge, etc.), so hat man den zusätzlichen Vorteil, dass dieser Gegenstand nicht verloren und auch nicht an andere Personen weitergegeben werden kann. Andererseits gibt es derzeit keine Verfahren, mit denen man das Vorhandensein eines Körperteils verlässlich auf elektronischem Wege beweisen könnte. Falls sich das zu identifizierende Körperteil zusätzlich an einem entfernten Ort befindet und die Kommunikation zwischen Server und Standort der Körperteils nur ungesichert erfolgen kann, so verkompliziert sich die Situation zusätzlich.

Insgesamt gilt in allen diesen Fällen die simple Weisheit, dass eine Kette nur so stark ist wie das schwächste Kettenglied. Übertragen auf biometrische Authentifizierungssysteme bedeutet dies z.B., dass ein noch so teures System zur Iris-Erkennung letztendlich wertlos ist, wenn es möglich ist, das Video-Signal der benutzten Kamera aufzuzeichnen und später vom Band abzuspielen. Ähnliches gilt für Fingerabdruckleser, die auf Tesafilmstreifen mit Fingerabdrücken genau so reagieren, wie auf echten Finger.

Da in den allermeisten Fällen die Erkennung des biometrischen Merkmals lokal erfolgt und das Ergebnis an den Server übermittelt wird, muss der lokale Rechner (incl. Netzwerkanschluss) manipulationssicher sein, da der Server seiner Antwort (biometrisches Merkmal OK/nicht OK) vertraut. Bei Windows-Arbeitsplatzrechnern ist dies allgemein nicht der Fall.

2-Faktor-Authentifizierung mittels SecurID-Token

Die Idee bei SecurID-Token ist die, dass es sich erstens um einmalige Gegenstände handelt, da jeder Token eine andere Zahlenfolge anzeigt, andererseits der Beweis des Besitzes leicht erfolgen kann, indem man die angezeigte Nummer übermittelt. Die Nummer selber kann unverschlüsselt übertragen werden, da sie sich alle 60 Sekunden „unvorhersehbar“ ändert.

Leider ist die Schlussfolgerung: „Jemand kennt die Nummer, die ein Token anzeigt, also hat er das Token innerhalb der letzten 60 Sekunden in der Hand gehabt“ falsch. So kannte der Autor bereits zum Zeitpunkt der Herausgabe dieses Tagungsbandes die Nummern, die sein Token am 11.11. 11:11 Uhr 2004 anzeigen wird, nämlich 219729 (11:15-11:25 Uhr: 072558, 675801,144883, 246214, 569635, 331624, 505000, 294821, 303538, 260949, 597544). Und das liegt nicht am Karneval, sondern am Mathematik-Studium!

Authentifizierung mittels asymmetrischer Verschlüsselung

Im Gegensatz zur symmetrischen Verschlüsselung, bei der zum Entschlüsseln genau der

Schlüssel verwendet werden muss, der auch zur Verschlüsselung verwendet wurde, erfolgt die asymmetrischen Ver-/Entschlüsselung mit einem Schlüssel-Paar. Was mit dem einen der beiden Schlüssel verschlüsselt wurde, kann nur mit dem anderen entschlüsselt werden und umgekehrt.

Dieses Verfahren ermöglicht nicht nur die Übertragung von verschlüsselten Nachrichten über unsichere Kanäle ohne vorherige Übertragung des Schlüssels über einen sicheren Kanal, sondern auch ein verlässliches Authentifizierungsverfahren. Dabei erhält jede Person, die sich authentifizieren soll, ein Schlüsselpaar, bestehend jeweils aus einem öffentlichen und einem dazugehörigen privaten Schlüssel. Falls eine Person P sich am Server S authentifizieren möchte, übermittelt sie ihren öffentlichen Schlüssel an S. Der Server generiert daraufhin eine Zufallszahl und schickt diese verschlüsselt mit dem öffentlichen Schlüssel von P zurück an P. Ist P in der Lage die verschlüsselte Nachricht zu entschlüsseln und die Zufallszahl zurück an S zu schicken, ist damit 100%ig sicher bewiesen, dass P im Besitz des zugehörigen privaten Schlüssel ist.

Dieses Verfahren ist aus Sicht des Autors derzeit die einzige Möglichkeit, auf elektronischem Wege von einem entfernten Standort über einen unsicheren Kommunikationskanal verlässlich zu beweisen, dass man im Besitz einer bestimmten Information ist (hier der private Schlüssel), ohne dass die Information selber übertragen werden muss.

2-Faktor-Authentifizierung mittels Smartcard

Smartcard Authentifizierungsverfahren benutzen (wenn sie korrekt eingesetzt werden) obiges Verfahren. Der private Schlüssel befindet sich dabei innerhalb der Smartcard, wo er zwar benutzt, aus der er aber nicht ausgelesen werden kann. Dass ein Auslesen des privaten Schlüssels auch mit Sonderrechten oder speziellen (nur dem Hersteller der Karte bekannten) Befehlen nicht möglich ist, wird in Deutschland durch das Signaturgesetz sichergestellt, dass strikte Auflagen an die verwendeten Chips, das eingesetzte Smartcard Betriebssystem und das Initialisierungsverfahren vorgibt.

Wenn also Server S eine Zufallszahl mit dem öffentlichen Schlüssel eines Smartcard Keys verschlüsselt und Person P ist in der Lage die Zufallszahl zurückzuschicken, ist damit der Beweis erfolgt, dass Person P zum Zeitpunkt der Authentifizierung im Besitz der Smartcard war. Denn nur innerhalb der Smartcard können Daten entschlüsselt werden, die mit dem entsprechenden öffentlichen Schlüssel verschlüsselt wurden.

Damit eine Smartcard im Verlustfall nicht missbraucht werden kann, können Entschlüsselungsoperationen mit dem auf der Karte enthaltenen privaten Schlüssel nur nach vorheriger Eingabe einer PIN durchgeführt werden. Mehrfache Falscheingabe der PIN führt zur Sperrung der Karte. Gesperrte Karten können mit einer sogenannten PUK wieder entsperrt werden.

Zertifikate

Beim obigen Authentifizierungsverfahren gibt es noch ein (lösbares) Problem. Wenn Person P

ihren öffentlichen Schlüssel an S übermittelt und behauptet, den passenden privaten Schlüssel zu besitzen, wie kann Server S dann sicher sein, dass der übermittelte öffentliche Schlüssel in Wirklichkeit nicht von einer anderen Person stammt. Im einfachsten Fall dadurch, dass Server S alle öffentlichen Schlüssel aller Personen kennt. Dann muss Person P nicht einmal ihren öffentlichen Schlüssel zu Beginn der Authentifizierung übermitteln. Dies hat aber den Nachteil, dass alle Schüssel von einen zentralen Stelle direkt an die Endbenutzer verteilt werden müssen. Will oder kann man das nicht organisatorisch sicherstellen, so kann man die Endbenutzer bitten, sich ihren öffentlichen Schlüssel in Form eines Zertifikates von einer vertrauenswürdigen Stelle (sogenannte Zertifizierungsstelle) zu besorgen.

Zertifizierungsstellen bieten folgende Dienstleistung an: Komm zu mir, beweise wer du bist, nenne mir deinen öffentlichen Schlüssel, beweise, dass du den passenden privaten Schlüssel besitzt, zahle Geld und ich signiere als Gegenleistung deinen öffentlichen Schlüssel zusammen mit deinem Namen und weiteren Informationen (insbesondere Gültigkeitsdaten).

Übermittelt nun eine Person ihren öffentlichen Schlüssel in Form eines Zertifikates, kann der Server S die elektronische Unterschrift des Zertifikates überprüfen. Ist sie korrekt, kann er davon ausgehen, dass der im Zertifikat enthaltene Name zur Person gehört, die authentifiziert werden möchte.

Ein weiterer Vorteil von Zertifikaten besteht darin, dass sie befristet für ein bestimmtes Zeitintervall ausgegeben werden können und dass sie mittels sogenannter Sperrlisten gezielt gesperrt werden können.

Projektanforderungen

Bei der Nordrheinischen Ärzteversorgung wurde die Ablösung aller Passwörter gegen die Authentifizierung mittels Smartcard und PIN angestrebt. Alle Passwörter bedeutete hier: Passwörter für Windows Logon (an Samba PDC), Oracle-Anmeldung, SAP-Anmeldung, E-mail-Abruf, Internet-Zugang von innerhalb der Firma, Intranet-Zugriff von außerhalb der Firma, Telnet-Zugriff, VPN-Aufbau von außen.

Die Mitarbeiterkarten sollten neben der Authentifizierung auch für die Zutrittskontrolle im Gebäude, die Zeiterfassung und die bargeldlose Bezahlung des Mittagessens verwendet werden.

Die Mitarbeiterkarten und die dazugehörigen PINs sollten so verteilt werden, dass nur der Mitarbeiter selber die PIN kennt. Das Verfahren musste so gestaltet werden, dass es auch von Systemadministratoren oder anderen Personen mit Sonderrechten nicht umgangen werden kann.

Praktikable Regelungen für folgende Fälle mussten festgelegt werden: „Hab meine Karte zuhause vergessen“, „Hab meine Karte verloren“, „Meine Karte ist kaputt“, „Hab drei mal eine falsche PIN eingegeben“, „Hab meine PIN vergessen“.

Eingesetzte Hardware

Die Nordrheinischen Ärzteversorgung hat alle Tastaturen gegen solche mit eingebautem Kartenleser ausgetauscht. Die eingebauten Kartenleser mussten die sichere PIN-Eingabe ermöglichen, was zusammen mit anderen Restriktionen die Auswahl auf folgende zwei Typen reduziert hat: Cherry G83-6700 (seriell) und Dell-Smartboard (USB). In Ausnahmefällen (z.B. Mitarbeiter benötigt aus gesundheitlichen Gründen ergonomische Tastatur) wurden externe Kartenleser mit Keypad (Omnikey 3610) eingesetzt. Die Zertifizierungsstelle setzt externe Leser mit Keypad und Display (Kobil Kaan Professional) ein.

Bei den verwendeten Smartcards handelt es sich um NetKey E4-Karten der Firma TeleSec in einer Spezialausführung mit Legic-Sender für die berührungslose Zutrittskontrolle.

Was gibt es am Markt?

Das Ergebnis einer ausführlichen Marktbetrachtung, zweier Cebit Besuche (2002 und 2003) und einer Vielzahl von Besprechungen mit Vertretern (fast) aller Anbieter solcher Produkte lässt sich wie folgt zusammenfassen: Aus Sicht der Vertreter ist das alles kein Problem, Details lassen sich sicherlich im Rahmen eines Projektes lösen. Bis heute konnte aber niemand auch nur eine teilweise funktionierende Lösung vorführen oder gar zu einem Festpreis anbieten.

Windows Logon mit Smartcard ist einfach lösbar, wenn man ein Active-Directory einsetzt und die Zertifikate von einer Microsoft CA ausstellen lässt. SAP ist eine besonders harte Nuss und Lösungen, die alle unsere Anforderungen erfüllen, gibt es nicht.

Zum Teil haben die vorhandenen Produkte sogar eklatante Sicherheitslücken. So findet man die eingegebenen PINs in Logfiles. Oder Produkte speichern die PIN, damit der Benutzer nicht immer wieder neu aufgefordert werden muss, diese einzugeben. Oder Smartcards werden nur als (angeblich sicherer) Speicher für die bisherigen Passwörter benutzt. Eine Anmeldung ohne Karte ist dann problemlos möglich, vorausgesetzt man hat vorher die Passwörter aus der Karte ausgelesen (z.B. mit dem frei verfügbaren Programmen aus dem OpenSmartCard Projekt).

Andererseits sind Zertifikats-basierte Lösungen auf dem Vormarsch. Dies betrifft insbesondere Oracle. So kann Oracle Application Server 10g (OAS 10g) mittlerweile Zertifikate nicht nur benutzen, sondern enthält erstmalig auch Software zur Erstellung von Zertifikaten (Oracle10g Certificate Authority). Während der Projektphase war OAS 10g allerdings noch nicht verfügbar.

Eingesetzte Software

Aufgrund der obigen Erfahrungen blieb keine andere Möglichkeit, als eine Lösung selber zu programmieren. Das hat die Projektlaufzeit um ca. ein Jahr verlängert, aber wegen der damit gleichzeitig verbundenen Kosteneinsparung war dies leicht zu verschmerzen.

Programmiert wurden mehrere Windows-DLLs und ein Unix-Dämon. Darüber hinaus war es

notwendig detailliertes Know-How über das verwendete TCOS 2.0 Betriebssystem der verwendeten Karte aufzubauen.

Die einzelnen Software-Komponenten sind im folgenden beschrieben.

Zertifizierungs-Software (Client-Server-Modus)

Zertifikate lassen sich leicht (und in perfekter Qualität) von der frei verfügbaren OpenSSL Software erzeugen. Diese besteht aus den OpenSSL Bibliotheken sowie Programmen mit Kommandozeilen-Interface, die nichts anderes machen, als die Bibliotheksroutinen aufzurufen.

Die OpenSSL Routinen können von Forms-Anwendungen (genauso wie Routinen in anderen Windows-DLLs) im Client-Server-Modus über das Foreign-Function-Interface aufgerufen werden. Allerdings ist dies nicht auf direktem Wege möglich, da Forms als Aufrufparameter nur Zeichenketten und Zahlen liefern kann und nicht die von OpenSSL benötigten Datentypen, insbesondere keine DER-codierten Zertifikate. Aus diesem Grund musste hier eine Windows DLL programmiert werden, deren Aufgabe im wesentlichen darin besteht, Zeichenketten in das interne Format von OpenSSL umzuwandeln und umgekehrt.

In ähnlicher Weise wurde der Zugriff von Forms auf das in Windows-2000 vorhandene Smartcard-API ermöglicht. Damit war das Lesen und Beschreiben von Smartcards aus einer Forms Anwendung heraus möglich.

Damit war es insgesamt möglich, eine Forms Anwendung zu erstellen, die Zertifikate erstellen, verwalten und auf die Mitarbeiterkarten schreiben kann. Die Anschaffung separater Zertifizierungssoftware, von der man ohnehin nur einen Bruchteil der Funktionalität benötigt hätte, konnte damit entfallen.

Windows-Anmeldung

Die Windows-Anmeldung erfolgt über ein von Microsoft standardisiertes API, das sogenannte Graphical Identification API (GINA). Wie eine eigene GINA-Bibliothek erstellt werden kann, die statt der vorhandenen MS-GINA die Authentifizierung eines Windows-Anwenders übernimmt, ist seitens Microsoft ausführlich dokumentiert und entsprechende Beispielprogramme sind im Netz verfügbar.

Genau ein solche GINA wurde programmiert und das oben beschriebene Authentifizierungsverfahren implementiert.

Da eine GINA nicht nur den Anmeldevorgang steuert, sondern auch die Bildschirm-Sperrung kontrolliert, konnte die eigenprogrammierte GINA so erstellt werden, dass das Entfernen der Karte oder das Anspringen des Bildschirmschoners zum sofortigen Sperren des Bildschirmes führt. Zur Entsperrung wird dann das gleiche Authentifizierungsverfahren benutzt wie zur eigentlichen Anmeldung.

Unix-Anmeldung

Die SecureShell (SSH) ist ein sicheres Äquivalent zum Telnet-Protokoll, das unter anderem auch die Schlüssel-basierte Authentifizierung vorsieht. Die Nordrheinische Ärzteversorgung setzt hier das freie Produkt PuTTY ein, das aus Sicht des Autors zu den besten am Markt verfügbaren SSH Implementierungen gehört. Es unterstützt zwar in der verfügbaren Version nur die Verwendung von Schlüsseln, die in Dateien gespeichert werden, da aber auch der PuTTY Sourcecode frei verfügbar ist, konnte die Smartcard Unterstützung leicht im Sourcecode ergänzt werden.

SAP-Anmeldung

Die SAP-Anmeldung ist hier (auf der DOAG-Tagung) nur der Vollständigkeit halber erwähnt. SAP selber hat eigene Schnittstellen für die sichere Authentifizierung (SAP-SNC = SecureNetworkComputing). Diese sind allerdings außerhalb des SAP-Umfeldes so gut wie nicht bekannt und SAP selber zertifiziert lediglich die Produkte von Fremdanbietern.

Client-Authentifizierung mit dem Mozilla-Browser

Zur Client-Authentifizierung in einer HTTPS-Verbindung übermittelt der Client sein Zertifikat an den Server. Der überprüft die Gültigkeit der enthaltenen Unterschrift und fordert dann den Client auf den Besitz des zum Zertifikat passenden privaten Schlüssels zu beweisen. An dieser Stelle muss der Browser auf die Smartcard zugreifen. Dieser Zugriff erfolgt bei Microsoft-Produkten mittels eines Cryptographic Service Providers (CSP) und bei (fast) allen anderen Produkten über die PKCS#11-Schnittstelle.

Da die Nordrheinische Ärzteversorgung Mozilla als Browser einsetzt, musste eine PKCS#11-Bibliothek programmiert werden. Fertige Bibliotheken werden zwar auch für die eingesetzte Karte/Leser-Kombination angeboten, allerdings stellte sich heraus, dass diese einen exklusiven Zugriff auf die Karte und/oder Leser erfordern. Da aber bereits die Anmelde GINA eine Verbindung zum Kartenleser öffnet und wegen der automatischen Sperrung bei Kartenentfernung auch nicht schließen kann, konnten diese Produkte nicht eingesetzt werden.

Der in Mozilla enthaltenen Email-Client benutzt die gleiche Zertifikatsverwaltung wie der Browser. Somit konnte auch die bei der Email-Abholung notwendige Authentifizierung auf Smartcards umgestellt werden (POP3 via SSL).

Anmeldung mit Oracle-Forms (Client-Server-Modus)

Parallel zur Einführung der Mitarbeiterkarten hat die Nordrheinische Ärzteversorgung mit der Umstellung ihrer Forms-Anwendungen vom Client-Server-Modus auf den Browser-basierten Zugriff (Forms im Web) begonnen. Daher wurde die Smartcard-basierte Authentifizierung mit einem Forms Runtime-Client nicht weiter verfolgt. Sie ist aber möglich, wie erste Tests gezeigt haben.

Im wesentlichen muss hierzu der On-Logon Trigger so programmiert werden, dass er anstelle

einer Passwort-basierten Anmeldung das oben beschriebene Verfahren durchführt.

Anmeldung mit Oracle-Forms (Forms im Web)

Die Anforderung der Nordrheinischen Ärzteversorgung bestand darin, dass nur Anwender mit einer solchen Smartcard auf eine Forms-Anwendung zugreifen können, deren Zertifikat mit dem unternehmenseigenen Root-Zertifikat signiert wurde und noch Gültigkeit besitzt.

Forms-Anwendungen laufen seit Forms9i nur noch im Web, d.h. die eigentliche Forms-Anwendung, der ausführbare Code sprich die FMX-Dateien liegen auf einem Web-Server, dem `Oracle Application Server` – kurz Application Server oder OAS. Die Kommunikation zwischen dem Forms Client, einem Java Applet - das in der Regel auf der Oracle-eigenen Java Virtual Machine (dem `Oracle JInitiator`) im Browser ausgeführt wird, und der Forms-Anwendung ist so voreingestellt, dass sie über HTTP erfolgt. Um die Anmeldung an eine Forms-Anwendung nur bei Vorliegen einer Smartcard mit passendem Zertifikat zuzulassen, muß die Verbindung über HTTPS erfolgen. Dazu ist der Application Server umzukonfigurieren, Zertifikate sind zu erstellen und abzulegen sowie die Vertrauenswürdigkeit von Root-Zertifikaten festzulegen.

Damit nach dem Forms-URL-Aufruf keine Eingabe der Datenbank-Logon-Daten erforderlich ist, müssen diese über einen Dienst aus einer Ablage gelesen werden. Der Dienst wird von der OAS-Komponente `Oracle Single Sign-On Server` – kurz: SSO-Server - gestellt. Die Ablage bildet das `Oracle Internet Directory (OID)`.

Zertifizierungs-Software (Forms im Web)

Die bereits vorgestellte Forms-Anwendung zum Erstellen, Verwalten und Abspeichern von Mitarbeiter-Zertifikaten auf Smartcards deckt einen großen Teil der Erfordernisse bezüglich des Zertifikate-Managements ab. Diese Anwendung arbeitet bisher im Client-Server-Modus mit Aufrufen an das Foreign Function Interface (ORA_FFI), um OpenSSL-Funktionalitäten auszuführen. Künftig soll auch diese Anwendung im Web laufen. Die ORA_FFI-Aufrufe würden dort versuchen, auf dem Application Server OpenSSL-Funktionen zu initiieren. Damit dies auf dem Client erfolgt, sind die ORA_FFI-Aufrufe durch entsprechende Webutil_C_API-Aufrufe zu ersetzen.

Erst vor wenigen Monaten hat Oracle den Security-Software-Hersteller Phaos Technology übernommen, dessen Java APIs zum Zertifikate-Management in den OAS integriert werden sollen – siehe Oracle Technology Network (OTN). Damit liesse sich diese Anwendung auch auf Nicht-Windows-Clients erweitern.

Der Application Server greift nicht direkt auf Zertifikate zu, sondern auf Wallets, in denen neben dem Zertifikat des Besitzers der private Schlüssel und die vertrauenswürdigen Root-Zertifikate gespeichert werden. Für die Verwaltung von Wallets hat Oracle die Java-Anwendung `Oracle Wallet Manager (OWM)` bereitgestellt. Damit kann z.B. das Server-Zertifikat in ein Server-Wallet importiert werden, das für die Ermöglichung von SSL-Verbindungen erforderlich ist. Um anmeldefenster-freies Logon zu unterstützen, können

Client-Wallets per OWM in das Oracle Internet Directory hochgeladen werden.

SSL-Konfiguration des Application Servers

Der Application Server kann so konfiguriert werden, dass SSL mit Client-Authentifizierung verlangt wird. Das erfolgt in der Konfigurationsdatei des Oracle HTTP Servers `httpd.conf`. Forms-Konfigurationsdateien bleiben dabei unangetastet.

Parameter	**Bedeutung**
`SSLEngine [on | off]`	SSL möglich? Ja | Nein
`SSLWallet File`	URL oder Pfad des Server Wallets
`SSLVerifyClient`	`none`: nur Server-Zertifikat erforderlich `optional`: Client-Zertifikat kann angegeben werden `required`: Client-Zertifikat muss angegeben werden
`SSLRequireSSL`	Unterbindung des Zugriffs ohne SSL

Abb. 1: Parameter zur Konfiguration von SSL-Verbindungen zum Application Server.

Mit einer ganzen Palette weiterer SSL-Direktiven läßt sich das OAS-Modul mod_ossl weiter konfigurieren. Insbesondere kann ein Pfad angeben werden, in dem alle Root-Zertifikate liegen, die als nicht vertrauenswürdig eingestuft werden – Verbindungsanfragen von Benutzern mit entsprechend signierten Zertifikaten werden abgewiesen.

Automatische Datenbank-Anmeldung

Befindet sich im Kartenleser eine gültige Smartcard und hat der Benutzer die passende Forms-URL mit HTTPS angefordert, so erscheint das Anmeldefenster, das nach den Datenbank-Logon-Daten (Datenbank-Benutzer, Paßwort für diese Datenbank-Benutzerkonto und Datenbankname) fragt. In der Regel soll der Anwender diese Daten jedoch nicht kennen. Anhand des per Smartcard identifizierten Benutzers soll die Anwendung automatisch mit den Datenbank-Logon-Daten versorgt werden. Der Forms Server läßt sich dazu so konfigurieren, daß nach dem Aufruf der Forms-URL der Forms Server den Single Sign-On Server nach diesen Daten fragt, der diese dann aus dem Oracle Internet Directory (OID) holt. Dazu ist in der Forms-Konfigurationsdatei `formsweb.cfg` im anwendungsspezifischen Kontext (named configuration) `ssoMode=true` zu setzen. Erfolgt dies in dem ‚Default'-Kontext, so gilt dies für alle Forms-Anwendungen, für die es nicht explizit ausgeschlossen wird.

Weiter müssen die Datenbank-Logon-Daten für den SSO-Benutzer vom SSO-Administrator im OID unter dem Punkt ‚Resource Access Information' erfasst werden. Hierbei kann unter Verwendung eines ‚Default Resource Access Descriptors' die Anmeldung unter dem gleichen Datenbank-Benutzer für alle SSO-Benutzer ermöglicht werden.

Single Sign-On ist nur in der Application Server Enterprise Edition verfügbar und nicht in der

neuen ‚Forms and Reports only'-Installationsoption.

Resümee

Nur die Authentifizierung per Smartcard und PIN entspricht den Sicherheitsanforderungen der Nordrheinischen Ärzteversorgung. Mithilfe einer selbsterstellten Forms-Anwendung wurde eine unternehmenseigene Zertifizierungsstelle implementiert, mit der die Mitarbeiter-Smartcards gelesen, beschrieben und verwaltet werden. Zur Smartcard-Authentifizierung an Forms-Anwendungen im Web sind Browser und Application Server zu konfigurieren. Um eine unternehmensweite, alle Anwendungen übergreifende Sicherheitslösung per Smartcard-Authentifizierung einzuführen, bedarf es vertiefter Kenntnisse über das Betriebssystem-Anmeldeverfahren, das Zertifikate-Management, dem Smartcard-Betriebssystem und den darauf zugreifenden APIs.

Kontaktadressen:

Dr. Peter Koch
Nordrheinische Ärzteversorgung
Tersteegenstr. 9
D-40474 Düsseldorf

Telefon: +49(0)211-4302-1294
Fax: +49(0)211-4302-1426
E-Mail doag.pkoch@dfgh.net

Dr. Peter Alteheld
MT AG
Stadionring 16
D-40878 Ratingen

Telefon: +49(0)2102-30961-0
Fax: +49(0)2102-30961-20
E-Mail dr.peter.alteheld@mt-ag.com
Internet: www.mt-ag.com