SSL-Authentisierung mit Nutzerzertifikaten

Eine sichere Alternative zu Benutzername/Passwort

Text: **Jürgen Brauckmann, Reimer Karlsen-Masur** (DFN-CERT GmbH)

Foto: © LordRunar - iStock.

Überblick

Die Verwendung von Nutzername/Passwort zur Anmeldung an Diensten hat bekanntermaßen diverse Nachteile. Wenn Nutzer ihr Passwort selbst wählen, setzen sie ein zu schwaches. Wird das Passwort dem Nutzer zugeteilt, schreibt er es auf, so dass es von Dritten gefunden werden kann. Die Wiederverwendung von Passwörtern an mehreren Diensten ist ein Sicherheitsrisiko, und eine diensteübergreifende Identität ist nicht ohne Zusatzaufwand zu erreichen. Zudem sind Passwort-Rate-Angriffe ein Risiko.

In der DFN-PKI werden seit vielen Jahren sowohl für Server wie auch für Nutzer X.509 Zertifikate ausgestellt. Wir möchten hier zeigen, wie sich Server- und Nutzerzertifikate für eine gegenseitige Authentisierung kombinieren lassen, um ein sicheres Login ohne Nutzername/Passwort realisieren zu können.

Authentisierung mit Nutzerzertifikaten vermeidet die Nachteile von Passwörtern: Der Nutzer muss sich nicht für jeden Dienst ein eigenes Passwort merken, sondern hat ein einziges Zertifikat, das vielfältig einsetzbar ist. Zertifikate haben ein gleichmäßiges Sicherheitsniveau, das nicht durch Nutzer mit zu schwachen Passworten am Server zerstört werden kann. Passwort-Rate-Angriffe gegen den Server sind prinzipbedingt nicht möglich. Daher ist eine Authentisierung mit Nutzerzertifikaten deutlich vertrauenswürdiger als Passwörter, was sich auch in der juristischen Bewertung widerspiegeln kann, wenn es z.B. um eine Anmeldung für wichtige Vorgänge wie Prüfungsanmeldungen geht. Voraussetzung für eine Authentisierung mit Nutzerzertifikaten ist zunächst einmal der Einsatz eines Serverzertifikats.

Serverzertifikate

Zertifikate für Server dienen zur Absicherung von Netzwerkverbindungen über das SSL-Protokoll (bzw. dem heute verwendeten Nachfolger TLS, wobei der Sprachgebrauch bei der Bezeichnung SSL blieb). SSL wird sehr häufig zusammen mit HTTP verwendet. Diese Kombination wird dann HTTPS genannt.

Üblicher Einsatzzweck von HTTPS sind alle „wichtigeren“ Dienste im Web, bei denen es nicht nur um einen öffentliche Informationsabruf, sondern um vertrauliche Kommunikation geht, wie z.B. Online-Banking, Web-Shops oder Webmail. Für andere Dienste wie E-Mail, Verzeichnisdienste oder Autorisierungsdienste (RADIUS) gibt es ebenfalls Varianten, die SSL nutzen.

SSL mit Serverzertifikaten bietet zwei Funktionen:

- Der Server authentisiert sich gegenüber dem Client. Vertraut der Client der Zertifizierungsstelle, die das Serverzertifikat ausgestellt hat, so kann er sich sicher sein, mit dem richtigen Server verbunden zu sein und nicht durch eine Man-in-the-Middle-Attacke angegriffen und abgehört zu werden. Damit der Client der Zertifizierungsstelle vertraut, muss in der Regel das zugehörige Wurzelzertifikat in der Software vom Hersteller vorinstalliert sein. In der DFN-PKI wird dieses durch das Wurzelzertifikat „Deutsche Telekom Root CA 2“ erreicht.
- Nach erfolgter Authentisierung wird die Verbindung verschlüsselt und ist anschließend (im Rahmen der Stärke der verwendeten Algorithmen) abhör- und manipulationssicher.

Nutzerzertifikate

So wie sich ein Dienst mit seinem Serverzertifikat gegenüber dem Webbrowser (und damit dem Nutzer) authentisiert, kann sich auch der Nutzer mit seinem eigenen, individuellen Nutzerzertifikat gegenüber dem Dienst authentisieren. Damit Nutzerzertifikate zur Authentisierung durch den Webserver verwendet werden können, muss der Server, genau wie der Client, der Zertifizierungsstelle vertrauen, die das Nutzerzertifikat ausgestellt hat. Die Nutzerzertifikate enthalten Angaben über den Inhaber, wie z.B. seinen Namen, die zugehörige Organisation und eventuell seine E-Mail-Adresse. Wie bei den Serverzertifikaten sind die Daten deshalb vertrauenswürdig, weil sie bei der Ausstellung des Zertifikats mit den in der Policy der Zertifizierungsstelle festgelegten Regeln verifiziert wurden.

Daher darf der Betreiber eines Webservers nicht einfach allen in seinem Betriebssystem vorinstallierten Zertifizierungsstellen vertrauen, sondern muss genau definieren, von welchen Zertifizierungsstellen die Zertifikate seiner Nutzer kommen dürfen.

Beschränkte Nutzerzertifikate nur für die Authentisierung

Nutzerzertifikate können grundsätzlich neben der Client-Authentisierung auch zur Signatur und Verschlüsselung von E-Mails oder anderen Daten eingesetzt werden. Das ist im Prinzip sehr praktisch, erfordert aber ein wenig Vorsicht: Verliert der Nutzer seinen privaten Schlüssel oder die Passphrase, sind die verschlüsselten Daten verloren. Daher empfiehlt sich grundsätzlich ein Backup von Zertifikaten und ihren privaten Schlüsseln. Idealerweise wird dieses Backup nur von den Nutzern selbst durchgeführt, um die Vertrauenswürdigkeit der Zertifikate nicht zu gefährden. Allerdings wird man Nutzer oft überfordern, wenn man von ihnen ein sicheres Backup verlangt.

Werden die Zertifikate dienstlich eingesetzt und werden potentiell wichtige Daten damit verschlüsselt, so hat die Einrichtung der Nutzer offensichtlich ein Interesse, das Backup selbst in die Hand zu nehmen. Sichere zentrale Backup-Prozesse sind aber aufwändig.

Deshalb kann es sinnvoll sein, Nutzerzertifikate prinzipiell ohne E-Mail-Verschlüsselungsoption auszustellen. In diesem Fall kann es nämlich keine verschlüsselte Daten oder E-Mails geben, für die man im Zweifel ein Backup des Zertifikats bräuchte.

Foto © maheyfotolia.com

Speicherung von Zertifikaten

Zu Server- und Nutzerzertifikaten gehört immer ein Schlüsselpaar, bestehend aus öffentlichem und privatem (geheim) Schlüssel. Der Zugriff auf diesen privaten Schlüssel ermöglicht die Funktionen, die nur der Inhaber des Zertifikats durchführen können soll: Authentisierung gegenüber einer anderen Partei und Entschlüsselung von Daten.

Daher muss der private Schlüssel vor einem Zugriff durch andere geschützt werden.

Im Dateisystem
Wird der private Schlüssel im Dateisystem abgespeichert, so muss er immer durch eine Passphrase geschützt sein. Beispiele für die Speicherung im Dateisystem sind private Schlüssel in der Zertifikatverwaltung von Mozilla-Produkten („Mozilla Network Security Services Keystore"), in der Registry unter Windows, als sogenannte PEM- oder PKCS#12-Dateien oder im Keystore von Java. Bei Serverzertifikaten kann der Schutz des Schlüssels durch eine Passphrase mit einem Verfügbarkeitsziel kollidieren: Ein unbeaufsichtigter, automatischer Neustart des Servers ist beim Einsatz von Passphrases für den Schlüssel nicht mehr möglich. Trotz Passphrase ist auch immer ein Schutz des privaten Schlüssels durch Dateisystemrechte erforderlich, damit andere Nutzer auf demselben System diesen nicht kopieren können.

Auf Hardware
Außer im Dateisystem können private Schlüssel auch auf speziellen Hardware-Komponenten wie Smartcards und USB-Krypto-Token gespeichert werden.

Der private Schlüssel kann in Hardware-Komponenten besser geschützt werden als in einem Dateisystem. Das Kopieren des privaten Schlüssels kann verhindert werden, und der Schutz durch eine Passphrase kann durch Fehleingabezähler deutlich verbessert werden.

Hardware-Komponenten müssen in die verwendete Software extra integriert werden. Dies geschieht mit Treibersoftware. Hier gibt es zwei grundlegende Varianten:

- Microsoft Crypto API ist ein API-Standard und Treibersystem für Microsoft-Produkte.
- PKCS#11 ist ein API-Standard, der auf allen Nicht-Microsoft-Systemen verbreitet ist.

Vor- und Nachteile der Authentisierung mit Nutzerzertifikaten

Die Nutzung von Nutzerzertifikaten zur Authentisierung bringt eine Reihe von Vorteilen gegenüber anderen Authentisierungsverfahren:

- Die Authentisierung des Nutzers ist eine Zwei-Faktor-Authentisierung:
  1. *Der Besitz des privaten Schlüssels (entweder als Datei oder als Hardware-Krypto-Token).*
  2. *Die Kenntnis des Passworts zur Aktivierung des privaten Schlüssels im Token.*
- Es wird keine Nutzername/Passwort-Kombination über eine vielleicht sogar unverschlüsselte Netzwerkverbindung transportiert.
- Ein Nutzerzertifikat realisiert automatisch eine einheitliche Identität über mehrere Dienste hinweg. Dienstespezifische Kennungen können dadurch abgelöst werden.
- Auch bei einem Wechsel des Zertifikats bleibt die Identität (der Name im Zertifikat) gleich, und die Authentisierung funktioniert unverändert auch mit dem neuen Zertifikat.
- Ein Nutzerzertifikat kann zur Anmeldung an mehreren Diensten benutzt werden, ohne auf die Problematik von wiederverwendeten Passwörtern achten zu müssen.

Beispiele für Systeme, bei denen eine Anmeldung mit Nutzerzertifikaten als einheitlicher Identität „out of the box" mög-

lich und erprobt ist: Wiki, Content-Management-System, Trouble-Ticket-System, Anmeldung am WLAN (eduroam), Kalenderanwendung, Versionskontrollsysteme, Anmeldung am IMAP- und SMTP-Server usw.

Das Interessante dabei: Diese Anwendungen müssen nicht erst mühsam in eine Single-Sign-On-Lösung eingezwängt oder mit Identity Providern zusammengeschlossen werden, sondern die einheitliche Identität kommt ausschließlich durch das Zertifikat zustande.

Trotzdem haben gerade im Dateisystem gespeicherte Nutzerzertifikate auch Nachteile: Für den normalen Anwender ist die Speicherung intransparent. Deshalb wird beispielsweise öfters vergessen, bei einer Neuinstallation eines Systems das Zertifikat zu kopieren. Ein weiterer Nachteil ist die Tatsache, dass moderne Malware inzwischen auch versucht, Nutzerzertifikate anzugreifen und nicht 100%ig ausgeschlossen werden kann, dass ein im Dateisystem gespeichertes Nutzerzertifikat gestohlen wird. Diese Nachteile können durch den Einsatz von Hardware-Krypto-Token zur Speicherung der Nutzerzertifikate ausgeglichen werden:

- Nutzer haben ihren Schlüssel „direkt in der Hand“. Die Analogie mit physikalischen Türschlüsseln kann helfen, die Aufmerksamkeit der Nutzer zur richtigen Verwendung der Zertifikate zu steigern.
- Es ist kein unbemerktes Entwenden oder Kopieren des privaten Schlüssels durch Angreifer möglich.
- Nutzer werden besser vor einer selbst verschuldeten Fehlbedienung geschützt und können ihren privaten Schlüssel und das Zertifikat nicht aus Versehen löschen, überschreiben oder unbemerkt verlieren.

Allerdings gibt es auch für Hardware-Krypto-Token Nachteile. Zunächst sind dabei die Kosten für die Beschaffung zu nennen. Des Weiteren kann die Integration problematisch sein: Nicht jede Client-Anwendung unterstützt die Verwendung von Hardware-Krypto-Token.

Ein weiteres Problem liegt in der Verfügbarkeit der Treiber: Es gibt nicht für jedes Hardware-Krypto-Token Unterstützung für jedes Betriebssystem – insbesondere ist die Nutzung auf mobilen Endgeräten mit iOS, BlackBerry OS oder Android praktisch nicht möglich (abgesehen von teuren Spezial-Lösungen wie dem „Merkel-Phone“).

Deshalb hat die Einführung von Hardware-Krypto-Token immer Projekt-Charakter.

Konfiguration der Client-Authentisierung

Die Konfiguration für die Client-Authentisierung ist nicht viel komplexer als die Konfiguration von SSL selbst. Allerdings muss sorgfältig gearbeitet werden, da Kleinigkeiten dafür sorgen können, dass entweder zu viele (unerwünschte) Nutzer Zugang erhalten oder aber der Webserver gar nicht mehr zugänglich ist. Das folgende Beispiel ist für einen Apache httpd.

Erster Schritt: Vertraute Wurzelzertifizierungsstellen

Um das vom Webbrowser geschickte Nutzerzertifikat zu validieren, benötigt der Webserver die Zertifikate der Wurzelzertifizierungsstelle der Nutzerzertifikate in einem eigenen Verzeichnis, das mit der Einstellung SSLCACertificatePath konfiguriert wird.

```
# Verzeichnis mit den vertrauten Wurzel-CA-Zertifikaten
#
SSLCACertificatePath /etc/apache2/ssl/trust-store-for-client-auth/ca-certs
```

Die Wurzelzertifikate müssen in dem konfigurierten Verzeichnis im PEM-Format vorliegen und einen speziellen Dateinamen haben, damit der Apache sie korrekt nutzt. Unter Linux muss zum Erzeugen der speziellen Dateinamen das Werkzeug c_rehash genutzt werden.

Im Falle der DFN-PKI Global Hierarchie ist nur das Zertifikat „Deutsche Telekom Root CA 2” nötig.

Zweiter Schritt: OCSP oder Sperrlisten konfigurieren

Nutzerzertifikate können aus den verschiedensten Gründen gesperrt werden. Damit der Apache gesperrten Zertifikaten den Zugriff verweigert, muss die Abfrage von Sperrinformationen, z.B. über OCSP, konfiguriert werden.

```
# OCSP konfigurieren
#
SSLOCSPEnable on

# Anzusprechender OCSP-Responder, wenn das Zertifikat
# keine eigene OCSP-Responder-URL enthält
#
SSLOCSPDefaultResponder http://ocsp.pca.dfn.de/OCSP-Server/OCSP

# OCSP-Responder aus dem Zertifikat verwenden, wenn
# vorhanden
#
SSLOCSPOverrideResponder off
```

OCSP ist ab Apache 2.3 nutzbar. Alternativ können auch Sperrlisten (CRLs) verwendet werden.

Dritter Schritt: Client-Authentisierung anschalten
Nach der Konfiguration der Wurzelzertifikate und der Sperrinformationen kann nun die Authentisierung innerhalb eines Virtual-Hosts oder auch eines Directory-Kontextes angeschaltet werden:

```
<VirtualHost www.example.org:443>
 # SSL anschalten
 #
 SSLEngine on

 # Verifizierungstiefe. Muss für die DFN-PKI auf 3 oder größer
# gesetzt werden
#
 SSLVerifyDepth 3

 # Client-Authentisierung anschalten. Kein Zugriff ohne
 # Client-Zertifikat
 #
 SSLVerifyClient require
 SSLOptions +StrictRequire
...
</VirtualHost>
```

Mit diesen Parametern wird der Zugriff auf all die Nutzer beschränkt, die sich mit einem Zertifikat aus der PKI unterhalb der konfigurierten Wurzelzertifikate ausgewiesen haben. Im Fall der DFN-PKI können das dann Nutzer aus allen Sub-CAs unterhalb der Deutsche Telekom Root CA 2 sein, also von deutlich mehr als 300 Einrichtungen. Es müssen daher weitere Einschränkungen konfiguriert werden, um nur die Nutzer von gewünschten Einrichtungen und nur bestimmte Nutzergruppen zu erlauben.

Vierter Schritt: Beschränkung des Zugriffs
Generell gibt es verschiedene Möglichkeiten, wie eine Zugriffsbeschränkung durchgeführt werden kann. Hierbei muss besonders darauf geachtet werden, dass nicht aus Versehen Nutzer aus den falschen Sub-CAs Zugriff erlangen. Wird beispielsweise nur nach einem Namen „Max Mustermann“ geprüft, so könnte dieser Name ja in mehreren Sub-CAs innerhalb der PKI vergeben werden. Daher müssen immer auch noch weitere Informationen über das Sub-CA-Zertifikat kontrolliert werden.

Beschränkung mit SSLRequire
Innerhalb eines Directory- oder .htaccess-Kontext kann mit der Direktive SSLRequire direkt spezifiziert werden, welche Zertifikate Zugriff haben. Mit folgendem Beispiel wird der Zugriff auf den Nutzer mit CN „Max Mustermann“ in der CA „Global Services CA“ beschränkt.

```
<directory /srv/www/authtest>
SSLRequire ( %{SSL_CLIENT_I_DN} eq „CN=DFN-Verein CA
Services,O=DFN-PKI,O=DFN-Verein,C=DE“ \
and %{SSL_CLIENT_CERT_CHAIN_1} =~ m|/9jJI/
MGLnPRv5COrZxbcwL7WSqp\n-----END | \
and %{SSL_CLIENT_S_DN_CN} eq „Max Mustermann“)

</directory>
```

Die Auswertungssprache ist sehr umfangreich. Für Details muss die Dokumentation von Apache herangezogen werden (für Version 2.4: http://httpd.apache.org/docs/2.4/mod/mod_ssl.html).

Beschränkung mit FakeBasicAuth
Alternativ kann der FakeBasicAuth-Mechanismus verwendet werden. Hierbei wird der Apache ähnlich konfiguriert wie für Nutzername/Passwort-Authentisierung. Anstelle von Nutzernamen wird der DN des Zertifikats in einer leicht zu wartenden Textdatei konfiguriert.

Beschränkung durch die Anwendung
Wenn im Apache eine komplexere Anwendung mit eigener, X.509-Zertifkate unterstützenden Nutzerverwaltung läuft, so sollte natürlich dieser Mechanismus verwendet werden. Aber auch hierbei muss auf die Beschränkung auf die richtigen Zertifizierungsstellen geachtet werden.

Fazit

Eine Authentisierung mit Nutzerzertifikaten kann eine interessante Alternative zu Nutzername/Passwort-Mechanismen sein, und gewährleistet ein gleichmäßigeres Sicherheitsniveau. Allerdings erfordert die Ausgabe der Zertifikate einen gewissen Aufwand, und die Akzeptanz bei nicht technischen Nutzern ist nicht automatisch gegeben. Hardware-Krypto-Token können das Verständnis der Nutzer für ihr Zertifikat deutlich erhöhen, auch wenn man sich dabei vorab intensiv mit Treiber- und Kompatibilitätsfragen befassen muss.

Die Einrichtung im Webserver ist einfach und unproblematisch. Die Sicherheit der Systeme wird erhöht, da die Angreifbarkeit über schwache oder aufgeschriebene Passworte abgewehrt wird und Passwort-Rate-Angriffe gegen den Server prinzipiell nicht möglich sind.

Insbesondere die einheitliche Identität über mehrere Systeme hinweg, die man bei der Verwendung von Zertifikaten automatisch ohne weiteren Aufwand erhält, ist sowohl für Administratoren als auch für Nutzer eine große Erleichterung. ◆