Ausweisvergabe

OpenCA ist eine freie Software zum Aufbau von Public Key Infrastructures (PKI). Sie basiert auf OpenSSL, ist relativ einfach konfigurierbar und bereits in mehreren Organisationen im produktiven Einsatz. Eine besondere Stärke ist der Batchbetrieb zur automatisierten Vergabe von Zertifikaten. Michael Bell, Oliver Welter

Clients, die E-Mails signieren und verschlüsseln können, sind nur die halbe Miete. Damit das gesamte System funktioniert, ist entweder ein hinreichend großes Web of Trust wie bei PGP oder eine zentralisierte Public Key Infrastructure erforderlich. OpenCA stellt die entsprechende Software zur Verfügung. Das wichtigste Element einer PKI ist die Zertifizierungsstelle (CA), mit deren geheimem Schlüssel die ausgestellten Zertifikate beglaubigt sind. Um die Integrität der CA zu schützen, ist sie im Normalfall auf speziell gesicherten Rechnern untergebracht und nicht mit dem öffentlichen Netzwerk verbunden.

Um Informationen zu allen ausgegebenen Zertifikaten sowie die Zertifikats-Sperrlisten für alle Benutzer öffentlich bereitzustellen, sind die relevanten Daten von der CA auf ein im Netz erreichbares System zu transferieren. Die Identifizierung der PKI-Teilnehmer ist üblicherweise an eine Registrierungsstelle (RA) delegiert. Das dortige Personal überprüft und bestätigt die Anträge der Benutzer und leitet sie schließlich an die CA weiter. OpenCA **[1]** bildet diese Anforderungen durch sein flexibles Interface-Konzept ab (Abbildung 1).

Interface, Core, Backend – der interne Aufbau

Für kryptographische Funktionen greift OpenCA auf die Bibliotheken und das Kommandozeilenprogramm des OpenSSL-Projekts zurück **[2]**. OpenCA ist, mit Ausnahme einiger Hilfsprogramme, in Perl implementiert und nutzt die DBI-Schnittstelle, um die Datenbank einzubinden. Prinzipiell lässt sich jede Datenbank einsetzen, für die ein DBI-Modul verfügbar ist. Aktiv unterstützt werden derzeit MySQL, PostgreSQL, Oracle, DB 2 sowie DBM.

Kern der Architektur ist der OpenCA-Daemon, ein eigenständiger, permanent laufender Prozess. Er erfüllt selbstständig keine Aufgabe in der PKI, sondern dient als Bindeglied zwischen Webschnittstelle, Konfiguration, Daten und dem funktionalen Backend. Aus dem Verzeichnis »openca/etc« liest er beim Start die Konfigurationsdateien und initialisiert das Datenbanksystem und die kryptographische Schicht.

In der Standardkonfiguration nehmen Dateien das Schlüsselmaterial auf, der Rechner führt die Verschlüsselungsoperationen selbst aus. Alternativ ist der Einsatz von OpenSC-kompatiblen Smartcards **[3]** oder der Hardwarelösungen von NCipher NShield **[4]** und Chrysalis Luna CA3 **[5]** möglich. Die Kommunikation mit dem Benutzer geschieht über die CGI-Schnittstelle eines externen Webservers, wo ein Hilfsprogramm die Daten aufbereitet und über eine Socketverbindung an den Daemon weitergibt oder die Antwort an den Browser zurückliefert.

Auch wenn der Name etwas anderes suggeriert – von sich aus tut der OpenCA-Daemon gar nichts. Wie es sich für eine Webanwendung gehört, wird nur dann was gearbeitet, wenn jemand zusieht. Ruft der Benutzer eine Webseite des OpenCA-Systems auf, so enthält dieser Aufruf ein Kommando und die zugehörigen Daten. Für jedes Kommando existiert eine Perl-Datei, die der Daemon ausführt. Die Kommandos operieren nicht direkt auf Daten und Ressourcen, sondern greifen über Module und Objekte auf das Backend-System zu, oft benötigte Operationen sind dabei in Funktionsbibliotheken zusammengefasst.

Schnittstellen für verschiedene Aufgaben

Ein Interface ist nichts anderes als eine Ansammlung solcher Kommandos, kombiniert mit einer Menüstruktur, um diese auszuführen. In der Standard-Distribution von OpenCA sind sieben Interfaces definiert (siehe Kasten „Standard-Interfaces“). Für den OpenCA-Daemon

ist die Namensgebung irrelevant, identifiziert und unterschieden werden die Interfaces anhand einer eindeutigen ID. Welche Kommandos auf einem Interface zur Verfügung stehen, legt die Datei »openca/etc/rbac/acl.xml« fest. Listing 1 enthält einen Ausschnitt dieser Datei für die Funktion »csr view«. »<module>« ist die ID des Interface, »<operation>« bezeichnet das Kommando.

Rollenmodelle

Die Tags »<role>« und »<owner>« erlauben weitere Filtereinstellungen, die in der Standardinstallation mit Wildcards belegt sind, also alle Werte erlauben. Detaillierte Hinweise zur Konfiguration sind im OpenCA-Guide **[7]**, Kapitel 4.1, unter dem Stichwort »ACL« zu finden. Nach jeder Änderung der Konfiguration muss übrigens ein »openca/etc/openca_rc restart« folgen.

Die Zugriffskontrolle auf die Interfaces lässt sich komplett ausschalten, wenn IP-Filter oder Bordmittel des Webservers ausreichen. Dazu ist in allen Dateien in »openca/etc/access_control« der Inhalt des »<login>«-Tag zu ändern:

```
<login>
     <type>none</type>
</login>
```

Dabei geht aber nicht nur die rollenbasierte Filterung verloren, diese Konfiguration ist auch riskant und daher nur bei wirklich einfachen Systemen zu empfehlen. Die Zugriffskontrolle der Interfaces besteht aus den drei Stufen:

- Channel (http/https)
- Login
- Rolle

Sie wird über die Dateien in »openca/etc/access_control« eingerichtet.

Listing 2 zeigt eine typische Konfiguration für ein Zertifikat-basiertes Login mit Rollen-basierter Zugriffskontrolle (RBAC). Der Eintrag bei »<channel>« lässt nur Verbindungen per HTTPS mit einer Schlüssellänge von mindestens 128 Bit zu, kürzere Schlüssel können noch bei älteren Browsern auftreten (zum Beispiel Netscape 4.x) oder bei Windows-Systemen, die unter Exportbeschränkungen ausgeliefert wurden. In solchen Fällen muss der Wert entsprechend angepasst werden.

Sollen unverschlüsselte Verbindungen über HTTP möglich sein, muss »<asymmetric_keylength>« auf »0« und »<protocol>« als Wildcard-Eintrag ».*« gesetzt sein. Beschwert sich OpenCA trotz korrekter Einstellungen über zu kurze Schlüssel, dann exportiert wahrscheinlich der Webserver die Zertifikatsinformationen nicht vollständig. Beim Webserver Apache **[9]** lässt sich das Problem durch den Eintrag »SSLOptions +StdEnvVars +ExportCertData« in der Hostdefinition beheben.

Der Login-Typ »<passwd>« ermöglicht eine Anmeldung mit einer Benutzername-Passwort-Kombination, weist dem Benutzer die angegebene Rolle zu und verwendet diese dann für die RBAC. Um mehrere Benutzerkonten anzulegen, kann der Block »<user>« beliebig oft kopiert werden. Weitere Login-Methoden erlauben eine Authentifizierung gegen externe Datenbanken sowie mit X509-Zertifikaten.

Die folgende Kurzanleitung beschreibt das Setup von zwei getrennten Maschinen für CA/Batch und RA/Pub. Alternativ lassen sich beide Interface-Gruppen in parallelen Verzeichnissen auf einer Maschine oder im selben Verzeichnis mit gemeinsamer Datenbank betreiben.

Listing 1: Zugriff auf das Kommando »csr view«

```
01 <permission>
02     <module>(0|1)</module>
03     <role>.*</role>
04     <operation>csr view</operation>
05     <owner>.*</owner>
06 </permission>
```

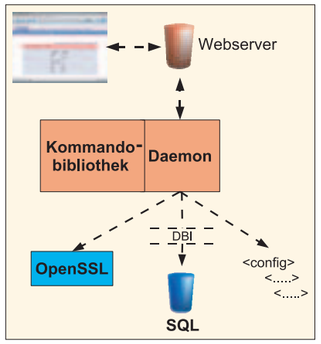

Abbildung 1: Interner Aufbau des OpenCA-Systems: Der Daemonprozess hält Konfiguration, Daten und Funktion zusammen und kommuniziert über einen externen Webserver mit dem Webbrowser.

Installation der CA

Für einige Distributionen gibt es fertige Pakete, aufgrund des zügigen Entwicklungstempos empfiehlt es sich aber, immer die neuesten Quellen von der OpenCA-Webseite bei Sourceforge **[8]** herunterzuladen. Aktuell ist die Version 0.9.2.1. Nach dem Auspacken des Tar-Archivs folgt der übliche Aufruf von

Standard-Interfaces

»pub« gestattet den Zugriff auf alle ausgestellten Benutzerzertifikate sowie Informationen zu den CA-Zertifikaten, es dient als Verteilungspunkt (CDP) der Zertifikatsperrlisten (CRL) und ermöglicht den Benutzern das Beantragen von Zertifikaten. Dabei kann zwischen der Schlüsselerzeugung im Browser (unterstützt werden Mozilla/Firefox und IE) oder auf dem Server gewählt werden. Für Zertifikatsanträge (CSR) im PKCS#10-Format, wie sie zum Beispiel bei Webserver-Zertifikaten benutzt werden, steht ebenfalls ein besonderes Formular zur Verfügung.

»scep« stellt einen Server für Zertifikatsmanagement über das SCEP-Protokoll **[6]** zur Verfügung.

»ldap« ermöglicht die Pflege von Zertifikaten in einem LDAP-Verzeichnis. Im Regelbetrieb ist dies erforderlich, da der Export der Zertifikate in das Verzeichnis automatisch beim Veröffentlichen geschieht.

»ra« fasst alle Funktionen zusammen, um Anträge von Benutzern zu bearbeiten, bestätigen, erneuern oder zurückzuweisen.

»ca« ist das Herzstück der CA. Es stellt die Zertifikate aus und ruft sie zurück, die Sperrlisten (CRL) entstehen ebenfalls hier.

»batch« ermöglicht es, alle durch einen RA-Operator genehmigten Anträge automatisch zu signieren. Es enthält auch ein System zur vollautomatischen Erstellung von Zertifikaten, das im Artikel näher erläutert wird.

»node« dient für den Datenaustausch zwischen den einzelnen Interfaces, falls diese mehr als eine Datenbank benutzen. Im Regelfall sind »ca« und »batch« sowie »ra«, »pub« und »ldap« auf einer eigenen Maschine installiert und teilen sich eine gemeinsame Datenbank. Auf beiden Maschinen ist dann dann je ein »node«-Interface notwendig. Das später angeführte Installationsbeispiel erzeugt eine solche Konfiguration.

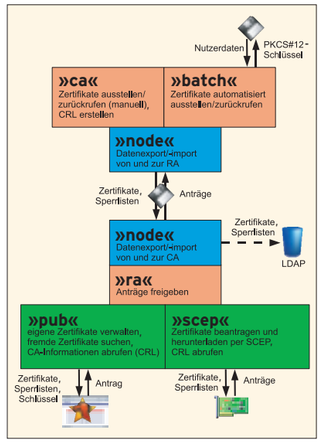

Abbildung 2: Architektur von OpenCA: Die Aufgaben der verschiedenen Interfaces und die Daten-Ein- und -Ausgaben.

»./configure«, das etwa zwei Dutzend Parameter erwartet. Im Unterverzeichnis »configs/« des ausgepackten Archivs befinden sich einige Beispiele, die nach Änderung der persönlichen Daten direkt übernommen werden können. Anschließend installiert ein »make && make install-offline« die Interfaces »ca«, »batch« und »node«.

Im nächsten Schritt sind noch einige Konfigurationsoptionen in der Datei »openca/etc/config.xml« zu setzen. Im allgemeinen Teil führt der Eintrag für den »sendmail«-Befehl oft zu langen Fehlersuchen, weil ein anderes Mailprogramm auf dem Server eingesetzt wird. Zweiter obligatorischer Stolperstein ist der Datenbankteil, bei Verwendung einer SQL-Datenbank muss »dbmodule« auf den Wert »DBI« gesetzt sein, danach sind Art, Zugangsdaten und Port der Datenbank einzutragen. Mit »localhost« wird die Datenbank direkt per Socket angesprochen.

Abschließend ist noch die korrekte Einstellung für den Datenaustausch auszuwählen. Es sind sieben verschiedene Beispiele in der Datei angegeben, standardmäßig ist kein Datenaustausch konfiguriert, was nur Sinn macht, falls sich alle Interfaces eine Datenbank teilen. Sonst ist die erste Konfiguration zu löschen und die zweite zu aktivieren.

Zentrale Konfiguration

Aus historischen Gründen gibt es neben der zentralen »config.xml« noch weitere Konfigurationsdateien. Um die Änderungen trotzdem zentral an einer Stelle durchführen zu können, kommt ein Template-Mechanismus zum Einsatz. Aus allen Dateien mit der Endung ».template« erzeugt der Aufruf von

```
./configure_etc
```

eine eigene Konfigurationsdatei mit den Daten aus »config.xml«. Wer Änderungen an solchen automatisch erstellten Konfigurationsdateien manuell vornimmt, muss immer parallel auch die ».template«-Datei anpassen, andernfalls gehen die Änderungen bei einem erneuten Aufruf des Konfigurationsskripts verloren. Der Template-Mechanismus wird auch für einige Dateien im Webserver-Verzeichnis benutzt.

Initialisierung

Die Basiskonfiguration des Systems ist nun erledigt, alles Weitere erfolgt über das Webinterface. Nach Ausführung von »openca/etc/openca_start« und dem Start des Webservers sollte über die URL »http://localhost/ca« eine Login-Maske zu sehen sein. Nach Anmeldung mit dem Standardbenutzer Root und dem Kennwort »root« erscheint die in Abbildung 3 gezeigte Übersichtsseite. Weiter geht's mit dem Menüpunkt »Initialisierung« im ersten Reiter.

Die erste Phase dient dazu, die Tabellenstruktur in der Datenbank zu erzeugen und und anschließend das Schlüsselpaar der CA zu erstellen und zu zertifizieren. Bei der Wahl der Schlüsselparameter ist darauf zu achten, dass die Anwendungen sie auch unterstützen. Die meisten Cisco-Geräte beispielsweise akzeptieren nur CA-Schlüssel mit maximal 2048 Bit.

Bei der Vergabe des Schlüssel-Passworts ist Kreativität gefragt, aber nicht zu viel: Von der Shell oder Perl interpretierbare Sonderzeichen wie »$%&*'#« sollten nicht vorkommen.

Ist der Schlüssel erzeugt, folgt der Zertifizierungsantrag. Die Angaben hier sind sorgfältig zu prüfen, denn sie erscheinen

Listing 2: Zugriffskontrolle eines Interface

```
01 <openca>
02     <access_control>
03         <channel>
04             <type>mod_ssl</type>
05             <protocol>ssl</protocol>
06             <source>.*</source>
07             <asymmetric_cipher>.*</asymmetric_cipher>
08             <asymmetric_keylength>0</asymmetric_keylength>
09             <symmetric_cipher>.*</symmetric_cipher>
10             <symmetric_keylength>128</symmetric_keylength>
11         </channel>
12         <login>
13             <type>passwd</type>
14             <database>internal</database>
15                 <passwd>
16                     <user>
17                         <name>root</name>
18                         <algorithm>sha1</algorithm>
19                         <digest>3Hbp8MAAbo+RngxRXGbbujmC94U</digest>
20                         <role>CA Operator</role>
21                     </user>
22                 </passwd>
23         </login>
24         <acl_config>
25             <acl>yes</acl>
26             <list>/usr/local/openca/openca/etc/rbac/acl.xml</list>
27 
   <command_dir>/usr/local/openca/openca/etc/rbac/cmds</command_dir>
28             <module_id>@ra_module_id@</module_id>
29 
   <ca_cert>/usr/local/openca/openca/var/crypto/cacerts/cacert.pem</ca
30 _cert>
31             <map_role>yes</map_role>
32             <map_operation>yes</map_operation>
33         </acl_config>
34     </access_control>
35     <token_config_file>/usr/local/openca/openca/etc/token.xml
   </token_config_fil
36 e>
37 </openca>
```

nachher in jedem ausgestellten Zertifikat. Sieht die Hierarchie keine übergeordnete CA vor, erlaubt es das Webinterface, ein selbst signiertes Zertifikat zu erzeugen. Alternativ signiert eine übergeordnete CA den Antrag. Dazu ist er per Diskette zu exportieren. Dann muss nach dem Import des CA-Zertifikats über das Webinterface das Root-Zertifikat manuell in das Verzeichnis »var/crypto/chain« eingepflegt werden.

Die Phasen 2 und 3 erstellen zwei Zertifikate für den ersten Operator und den Webserver des Online-Systems. Um die vorgeschlagenen eindeutigen Namen (DNs) der Zertifikate zu ändern, muss die Datei »openca/etc/servers/ca.conf« editiert werden. Beide Schritte sind auch später über das PUB/RA-Interface möglich, jedoch ergibt sich hier ein klassisches Henne-Ei-Problem, da die RA zum Betrieb ein SSL-Zertifikat braucht. Hier kann während der Initialisierung auch ein mit OpenSSL erstelltes und selbst signiertes Zertifikat einspringen.

Die Installation des Online-Systems der RA unterscheidet sich nur unwesentlich von der des CA-Systems. Lediglich der Aufruf von »make install-offline« wird zu »make install-online« und in der Datei »config.xml« ist für den Datenaustausch das Beispiel »5.« auszuwählen. Auch hier ist nun ein Aufruf von »./configure_etc« erforderlich, um die Konfigurationsdateien zu erstellen, danach ist der OpenCA-Daemon startbereit. Die Initialisierung der Datenbank des Online-Interface geschieht über den entsprechenden Menüpunkt des »node«-Interface »https://myca/node/«.

Online-System einrichten

Die Einrichtung des Online-Systems gestaltet sich etwas aufwändiger, da die aus den Konfigurationsoptionen abgeleiteten Werte für die Zertifikatsnamen (DN) in der Regel nicht dem gewünschten Format entsprechen. Erster Anlaufpunkt hierfür ist die Datei »openca/etc/servers/pub.conf«. Neben den gültigen Schlüssellängen legt sie fest, welche Informationen in das Zertifikat kommen. Zur Abfrage der Informationen lassen sich normale Freitext-Felder oder Auswahlmenüs definieren. Die Textfelder können eine Syntax-Prüfung erhalten,

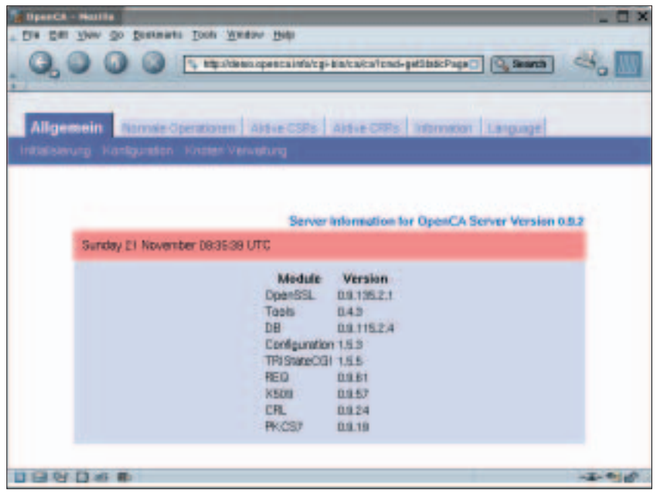

Abbildung 3: Startseite des CA-Moduls nach erfolgreichem Login.

um zum Beispiel die Eingabe einer E-Mail-Adresse zu erzwingen oder Sonderzeichen auszufiltern.

Antragstypen

Verschiedene Antragstypen verfügen jeweils über einen eigenen Konfigurationsblock. Beispielsweise kann entweder der Browser des Antragstellers ein Schlüsselpaar für das Zertifikat generieren oder der Server. Für Browser der Mozilla-Familie (Typ »SPKAC«) und den Microsoft Internet Explorer (Typ »IE«) gibt es je einen eigenen Block, beide sind im Normalfall aber praktisch identisch zu konfigurieren. Soll der Server die Schlüssel generieren, ist der Typ »BASIC« richtig. Soll die Registrierungsstelle auch Zertifizierungsanträge im PKCS#10-Austauschformat akzeptieren, legen die Angaben im Block »PKCS10« einige Bedingungen fest, die von einem solchen Antrag erfüllt werden müssen.

Der Template-Mechanismus erstellt wiederum die Datei »pub.conf«, die endgültigen Änderungen sind daher in der Datei »pub.conf.template« durchzuführen und anschließend in »pub.conf« zu übersetzen. Um nicht jedes Mal alle Konfigurationsdateien neu zu erzeugen, empfiehlt sich dafür die Verwendung von »bin/openca-configure«.

Änderungen an den Dateien in »openca/etc/servers« erfordern keinen Neustart des OpenCA-Daemon, da sie bei jedem Aufruf einer Webseite neu eingelesen werden. Bevor das System an das öffentliche Netz darf, sind unbedingt die Standardkennwörter für die Zugriffskontrolle zu ändern. Zur Erzeugung der Passwort-Hashes dient das Programm »bin/openca-digest«.

Die PKI an sich ist nun einsatzbereit, falls ein LDAP-Server die Zertifikate bereitstellen soll, ist es nun an der Zeit, ihn zu installieren. Das OpenLDAP-Paket der aktuellen Distributionen ist für den Zweck bestens geeignet, eine Konfigurationsdatei und Initialisierungsdateien liegen in »contrib/openldap/« des Quellcode-Verzeichnisses.

Die Zugangsdaten für den LDAP-Server sind in den entsprechenden Abschnitt in »config.xml« einzutragen. Ist dort »update_ldap_automatic« auf »yes« gesetzt, exportiert OpenCA die Zertifikate bei Ausstellung automatisch in das LDAP-Verzeichnis.

OpenCA im Betrieb

Im operativen Betrieb des Systems muss das PKI-Personal drei verschiedene Aufgaben bewältigen:

- Ausstellen von Zertifikaten
- Zurückziehen von Zertifikaten
- Erstellen einer Sperrliste

Das Ausstellen eines Zertifikats, also der Regelfall im Alltagsbetrieb, wird durch

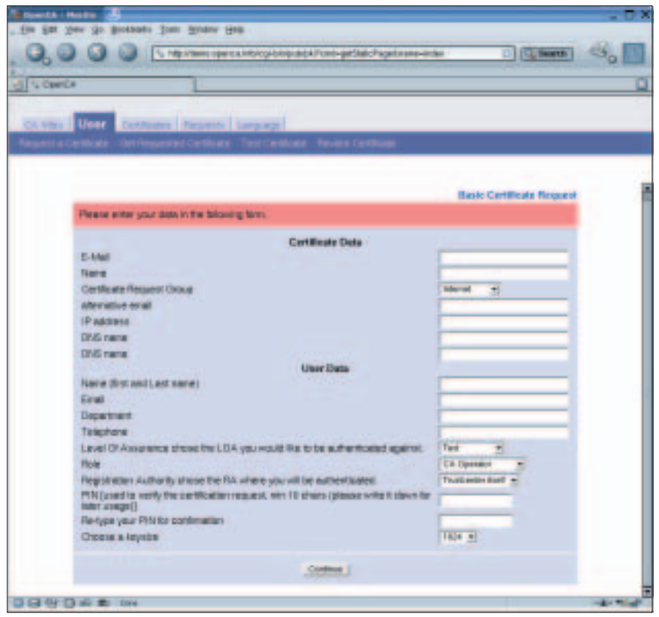

Abbildung 4: Formular zur Beantragung eines Zertifikats. Die abgefragten Elemente gehen in den DN beziehungsweise in die Erweiterungen des Zertifikats ein und sind frei konfigurierbar.

den Benutzer eingeleitet, sobald er über das öffentliche Interface einen Zertifizierungsantrag stellt. Dazu muss der Benutzer nichts weiter tun, als die URL »https://*myca*/pub/« aufrufen und im Menü »Nutzer | Beantragen eines Zertifikates« auswählen.

Zertifikate per Web

Verwendet der Benutzer einen unterstützten Browser, dann kann er das Schlüsselpaar innerhalb des Browsers selber erzeugen und schickt mit Hilfe des Webformulars in Abbildung 4 einen Zertifizierungsantrag an die RA. Kann er das nicht, übernimmt die RA diese Aufgabe. Der Eintrag im Feld »Role« wird zum einen für die Zugriffskontrolle herangezogen, bestimmt aber auch den Aufbau der so genannten Zertifikatserweiterungen.

Mit der Seriennummer seines Antrags begibt sich der Benutzer dann zu der zuständigen Registrierungsstelle, wo er sich durch Vorlage geeigneter Unterlagen identifiziert. Die Aufgabe des RA-Operators ist es, die Daten des Antrags mit der Legitimation des Benutzers zu vergleichen und bei Übereinstimmung dem Antrag stattzugeben. Der Aufruf der RA-Site »https://myca/ra/« und die Auswahl von »Information | Zertifizierungsanträge | Neu« zeigt eine Liste aller ungeprüften Anträge, ein Klick auf die Seriennummer öffnet die in Abbildung 5 gezeigte Detailansicht des Antrags.

Am Fuß der Seite sind einige Schaltflächen zu sehen: »Bearbeiten des Antrages« erlaubt dem Operator die Korrektur der Daten, um zum Beispiel Tippfehler auszubessern. Ist der Antrag in Ordnung, signiert ihn der Operator mittels »Antrag genehmigen« und bringt ihn damit in den In-Bearbeitung-Status. Verfügt der Operator über kein eigenes Zertifikat, kann er den »Antrag genehmigen ohne ihn digital zu signieren«.

Datenaustausch

Die bestätigten Anträge müssen nun per Diskette zur CA gelangen. Im Menü »Administration | Datenaustausch« auf »https://*myca*/node/« sind verschiedene Makrofunktionen für den Austausch von Daten mit höheren und niedrigeren Ebenen verfügbar. Die CA ist aus Sicht der RA eine höhere Ebene. Der Operator wählt also im letzten Menü »Exportieren von Daten zu einer höheren Ebene der Hierarchie« der Punkt »Anträge« aus. Falls auch genehmigte Rückrufanträge vorliegen, kann er mit der Option »Alle« auch beide Antragsarten gemeinsam exportieren.

Es folgen die Aufforderung, das Exportmedium einzulegen, und anschließend das Protokoll des Exportvorgangs. Das häufigste Problem an dieser Stelle beschreibt die Fehlermeldung:

```
/bin/tar: /dev/fd0: Cannot open: ?
Permission denied
```

Der OpenCA-Prozess hat keinen Zugriff auf das Diskettenlaufwerk. Dann gilt es, mit »ps axu | grep openca« den Linux-Benutzer des Prozesses herauszufinden und ihm die notwendigen Rechte zu geben. Läuft der Export korrekt, werden die Seriennummern der exportierten Anträge einzeln angezeigt:

```
Exporting approved REQUEST ...
Exporting all necessary objects.
20512.pkcs#10_with_pkcs#7_signature
```

Mit dieser Diskette im CA-Rechner ruft der Operator dann das gleiche Menü auf, wählt aber diesmal »Importieren von Daten von einer niedrigeren Ebene der Hierarchie | Anträge«. Wie schon beim Export werden die Seriennummern der importierten Anträge angezeigt:

```
Importing approved REQUEST ...
Cleaning up the collected import logs ...
20512.pkcs#10_with_pkcs#7_signature inserted
```

Vergabestelle

Zur Bearbeitung einzelner Zertifikate geht der CA-Operator genauso vor wie der RA-Operator bei der Genehmigung. Im CA-Interface »http://localhost/ca/« im Menü »Information | Zertifizierungsanträge | Genehmigt« ist eine Liste aller freigegebenen Anträge zu finden, in der ersten Spalte der Liste steht die Seriennummer des Operatorzertifikats, mit dem der Antrag signiert wurde. Bei ohne Signatur bestätigten Anträgen ist hier ein »n/a« eingetragen.

Ein Klick auf die Seriennummer öffnet den Antrag (siehe Abbildung 6), durch ein Icon rechts oben wird die Integrität der Operator-Signatur bestätigt. Nach ei-

nem Klick auf »Zertifikat ausstellen« wird erst der Operator nach der PIN des CA-Schlüssels gefragt, dann das Zertifikat ausgestellt.
Bei einer großen Anzahl von Zertifikaten ist dieses Vorgehen umständlich. Eine Bestätigung der Anträge im Einzelnen ist auch nicht notwendig, sofern sie von den Operatoren signiert sind. Um das Ausstellen zu vereinfachen, gibt es im Batch-Interface »http://localhost/batch« die Möglichkeit, alle Zertifikate für eine bestimmte Benutzergruppe (Rolle) mit einem Aufruf auszustellen. Die gleiche Funktion ist für die Bearbeitung von Rückrufanträgen vorhanden.

Automatisches Ausstellen von Zertifikaten

Die ausgestellten Zertifikate exportiert man wieder über das »node«-Interface per Diskette. Beim Import der Daten auf der Online-Seite gelangen die Zertifikate automatisch in das LDAP-Verzeichnis, die Benutzer erhalten eine E-Mail-Benachrichtigung. Die Textvorlagen dazu liegen im Verzeichnis »openca/lib/mails/de_DE/« der CA-Maschine.
Wenn die Benutzer ihre Schlüssel selber erzeugt haben, ist es wichtig, dass sie das Zertifikat mit dem gleichen Browser abholen, mit dem sie auch das Schlüsselpaar erstellt haben. Erst dadurch wird der private Schlüssel dem Zertifikat zugeordnet und im Browser benutzbar. Für Server-seitig erzeugte Schlüssel gibt es verschiedene Wege, um das Zertifikat zuzustellen. Hinweise gibt's im OpenCA-Guide-Kapitel 6.4 **[7]**.
Das vorzeitige Zurückziehen eines Zertifikates kann erforderlich sein, wenn der private Schlüssel verloren oder gestohlen wurde oder wenn ein Benutzer die Voraussetzungen für eine Zertifizierung nicht mehr erfüllt, zum Beispiel das Unternehmen verlässt. Bei jedem zurückgezogenen Zertifikat muss sofort eine neue Sperrliste erscheinen. Sperrlisten haben aber nur eine begrenzte Gültigkeitsdauer und sind auch dann regelmäßig zu erneuern, wenn keine neuen Sperrungen hinzukommen.
Der Operator der CA erstellt die Sperrliste über das Interface und überträgt sie per Datenaustausch an das Online-System (RA). Das Verhalten der Client-Anwendungen, falls keine oder keine gültige Sperrliste verfügbar ist, variiert – korrekterweise sollten sie dann das Zertifikat als ungültig ansehen.

Batchprozessor

Das Batchsystem macht die automatische Ausführung komplexer Aufgaben ohne den Eingriff eines Operators möglich. Das OpenCA-Standardpaket enthält unter anderem eine Beispiel-Implementierung, die eine vollautomatische Erzeugung von Zertifikaten aus importierten Nutzerdaten erlaubt.
Initialisiert wird ein Auftrag durch Import eines Datensatzes mittels einer Datei, welche die Datensätze als strukturierten Ascii-Text enthält. Der Aufbau der Datei ist sehr einfach, sodass sie problemlos durch externe Benutzerverwaltungssysteme erzeugt werden kann. Während des Importvorgangs entsteht für jeden Datensatz ein Verzeichnis in »openca/var/bp/users« für die Ablage der Datenelemente, dann wird die Prozessnummer in die Liste der aktiven Prozesse eingetragen. Anzahl und Größe der importierten Datensätze und -elemente sind frei wählbar.
Der eigentliche Batchprozessor ist ein Zustandsautomat, der über das Webinterface »batch« startet. Einmal angestoßen führt er eine vorher definierte Anzahl von Schritten für jeden aktiven Prozess aus. Die Zustände, die ein Prozess während der Bearbeitung durchläuft, entsprechen den Zuständen des Automaten. Jedem möglichen Status ist ein Kommando zugeordnet, ist der Prozess an der Reihe, wird abhängig vom aktuellen Zustand das entsprechende Kommando aufgerufen.

Zustandsautomat

Ein Kommando ist als Perl-Datei implementiert und hat Zugriff auf das Datenverzeichnis des Prozesses und auf die Objekte und Module des OpenCA-Systems. Die aus dem Prozess resultierenden Antrags- und Zertifikatsobjekte landen in der Datenbank des Hauptsystems, sodass sie später gegebenenfalls über das normale Verwaltungssystem zugänglich sind. Durch die Aufteilung des Gesamtprozesses in mehrere Einzelschritte lassen sich Fehler besser lokalisieren und die Teilfunktionen für verschiedenen Abläufe wiederverwenden.

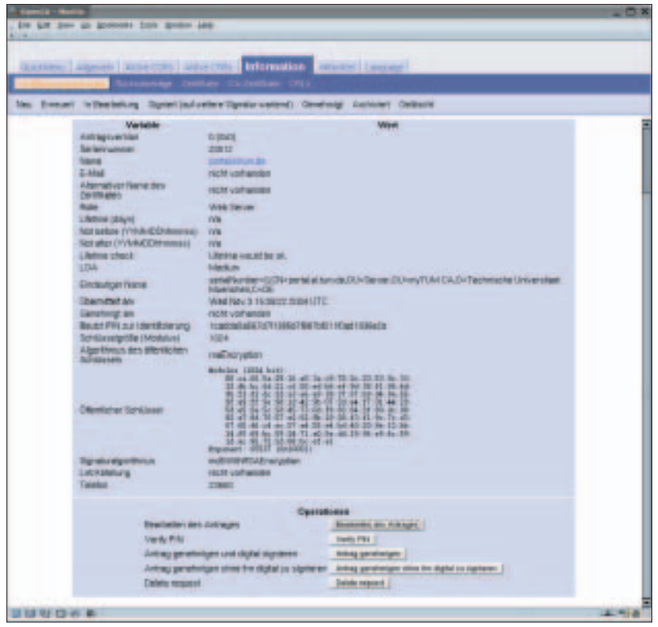

Abbildung 5: Detailansicht eines Antrags für den RA-Operator.

Auf einer Testinstallation wurden bereits mehrere zehntausend Zertifikate mit dem Batchsystem ausgestellt, ohne dass ein manueller Eingriff erforderlich war. In den Systemen der Autoren ist das Batchsystem in den Alltagsbetrieb integriert und funktioniert ohne Probleme.

Flexibler durch neue Datenbankstruktur

Auch die Zukunft von OpenCA zeigt sich spannend. Die wichtigsten Änderungen sind der Wegfall von DBM als Datenspeicher und eine Konzentration auf SQL-Datenbanken. Somit sind künftig auch komplexe Suchfunktionen in den Datenbeständen möglich. Durch die Migration aller dynamischen Daten aus dem Dateisystem in die Datenbank erwarten die Entwickler eine deutliche Steigerung von Ausführungsgeschwindigkeit und unterstützten Nutzerzahlen, vor allem im Batchbetrieb.
Die Umstrukturierung der Datenbanken erlaubt auch eine Unterstützung des Vier-Augen-Prinzips für die Freigabe von Anträgen sowie die Durchführung von Keybackup. Das Batchsystem wird zu einem generellen Hilfssystem für automatisierbare Prozesse ausgebaut und hat künftig auch Zugriff auf Daten aus den anderen Schnittstellen. Die Einführung eines eigenen Audit-Moduls, das jede Nutzung der privaten Schlüssel der CA protokolliert, erhöht die Sicherheit des Systems und erlaubt die Bereitstellung der Schlüssel für eine vollautomatische Bearbeitung. Für periodisch auftretende Aufgaben wie die monatliche Erstellung einer CRL ist ein eigenes Scheduler-System oder ein Kommandozeilen-Interface in der Diskussion.
Als vollständig neue Funktion wird die Unterstützung für CA-Rollover hinzukommen und den gleitenden Austausch des CA-Schlüssels am Ende der Gültigkeitsdauer ermöglichen.

Fazit

OpenCA ist eine voll funktionsfähige Open-Source-Lösung zum Aufbau von PKI-Strukturen mit S/MIME gemäß PKIX-Standard. Sind alle organisatorischen Maßnahmen und Regeln definiert, lässt sich das Gesamtsystem in ein bis zwei Tagen konfigurieren und in Betrieb nehmen. Für den Einsatz als PKI für VPN-Lösungen bringt OpenCA ein eigenes Interface für das Simple Certificate Enrollment Protocol (SCEP) mit – ein Protokoll von Cisco, das Infrastruktur-Hardware wie Routern oder Switches automatisch Zertifikate zuteilen kann. Die Bestätigung der über SCEP eingebrachten Anträge muss derzeit noch manuell geschehen, in der Entwicklungsversion ist eine Automatisierung mit Pre-shared Secrets bereits in der Testphase. Zusammen mit der geplanten Erweiterung des Batchsystems ist damit eine vollständig automatisch arbeitende PKI möglich.
Bei Problemen hilft die Entwicklergemeinschaft in der Regel schnell weiter, vier der sechs derzeit aktiven Haupt-Entwickler stammen aus Deutschland und bieten teilweise auch kommerzielle Support-Leistungen an. Das Produkt ist in einigen Hochschulen, Verwaltungen und einem großen Finanzinstitut im produktiven Einsatz. Mit einer kontinuierlichen Weiterentwicklung und Pflege der Software in den nächsten Jahren ist zu rechnen. Unter der URL [http://demo.openca.info] lädt eine Online-Demo zum Ausprobieren ein. *(uwo)* ■

Abbildung 6: Detailansicht des genehmigten Antrags zur Ausstellung des Zertifikats. Das Schloss-Symbol oben rechts bestätigt die Integrität der Operator-Signatur.

Infos

[1] OpenCA-Webseite: [www.openca.org]
[2] OpenSSL-Projekt: [http://www.openssl.org]
[3] OpenSC-Projekt für Smartcards: [http://www.opensc.org]
[4] NCipher NShield und HSM: [http://www.ncipher.com/nshield/index.html]
[5] Chrysalis Luna CA3 HSM: [http://www.safenet-inc.com/products/luna/luna_ca3.asp]
[6] SCEP: [http://www.ietf.org/internet-drafts/draft-nourse-scep-10.txt]
[7] OpenCA-Guide: [http://www.openca.info/docs/]
[8] Download aktueller OpenCA-Versionen: [http://sourceforge.net/projects/openca]
[9] Apache Webserver: [http://httpd.apache.org]

Die Autoren

Oliver Welter ist seit 2002 am Lehrstuhl für Datenverarbeitung der TU München tätig und hat dort ein Pilotprojekt zur sicheren E-Mail-Kommunikation mit OpenCA gestartet. Wenn er keine Erstsemester mit „Grundlagen der Informatik" ärgert oder an einer Linux-Kiste schraubt, ist er meist auf, im oder unter Wasser zu finden.
Michael Bell ist seit 1998 an der Humboldt-Universität zu Berlin und beschäftigt sich dort im Computer- und Medienservice (ehemals Rechenzentrum) seit 1999 mit dem Thema Sicherheit. Seit 2001 ist OpenCA zur Liste seiner Hobbys hinzugekommen.