



LinOTP Handbuch

© LSE Leading Security Experts GmbH

Version 1.3

Inhaltsverzeichnis

1	Einführung	3
1.1	Glossar	3
1.2	Systemübersicht	4
2	Anwendungsfälle	8
2.1	VPN und SSL VPN	8
2.2	Webseiten	8
2.3	Lokale Anmeldung / Domänenanmeldung	9
3	Installation	10
3.1	Ausrollsystem	10
3.2	RADIUS-Server	10
4	Konfiguration	12
4.1	Schlüsselerzeugung	12
4.2	LDAP-Server	12
4.3	Ausrollsystem	15
4.4	RADIUS-Server	15
5	Benutzung	17
5.1	OTP-Benutzer anlegen	17
5.2	OTP-Benutzer resynchronisieren	18
5.3	OTP-Benutzer PIN ändern	19
5.4	OTP-Benutzer sperren und reaktivieren	19
5.5	OTP-Benutzer löschen	19
5.6	Automatisierung	20
A	Upgrade von LinOTP v1.0 auf v1.1	21
B	Fehlerbehandlung / FAQs	23
B.1	TLS-Probleme	23
B.2	Einstellungen in ldap.conf werden von otpadm nicht berücksichtigt	23
C	Lizenztexte	25

1 Einführung

Mit LSE LinOTP haben Sie sich für ein flexibles, zeitgemäßes Authentisierungssystem entschieden. Herzlichen Glückwunsch zu Ihrer Wahl.

LSE LinOTP ist eine Lösung für RADIUS-Authentifizierung mit Einmalpassworten - One Time Passwords (OTP) - am Linux-Backend. Die Laufzeitkomponenten von LinOTP sind quelloffen, so dass Sie selber die Möglichkeit haben, Anpassungen vorzunehmen oder ein Code-Audit durchzuführen. LinOTP basiert auf GNU/Linux und lässt sich dadurch schlank und ressourcenschonend einsetzen und skaliert gleichzeitig für große Installationen.

Dies ist das Handbuch für die LSE LinOTP Enterprise-Edition. Jedoch lassen sich die Kapitel zum FreeRADIUS- (3.2 und 4.4) und OpenLDAP-Server (4.2) auch für die Community-Edition verwenden. Speziell die Kapitel über das Ausrollsystem und dessen Benutzung (Kapitel 5) bezieht sich ausschließlich auf die Enterprise-Edition.

1.1 Glossar

OTP (One Time Password) ist ein Einmalpasswort, das nur einmal zur Authentisierung verwendet werden kann. Es ist nicht möglich, sich mit einem solchen Passwort ein zweites mal zu authentisieren.

HMAC-OTP (HMAC-Based One Time Password Algorithm) ist ein offener Standard (RFC4226) für die Erzeugung von Einmalpasswörtern.

RADIUS (Remote Authentication Dial In User Service) ist ein offener Standard (RFC2865) zur Authentifizierung von Benutzern. LinOTP ermöglicht die Authentifizierung von Benutzern mit Hilfe der Aladdin eToken NG OTP an einem kompletten Linux-Backend.

1.2 Systemübersicht

Komponenten

LinOTP integriert sich in FreeRADIUS und OpenLDAP. Zentrale Komponente von LinOTP ist das FreeRADIUS-Modul (rlm.linotp) zur Authentifizierung mittels HMAC-OTP. Es greift über die Konnektor-Bibliothek (libotpdb) auf ein LDAP-Verzeichnis zu, in dem die notwendigen Attribute pro User gespeichert sind. Das LinOTP Adminwerkzeug (otpadm) dient der Initialisierung von eToken NG OTP und der Verwaltung der entsprechenden Attribute im LDAP.

Hintergründe zu HMAC-OTP

HMAC-OTP ist ein offener Standard, der in RFC 4226 definiert ist. HMAC-OTP beschreibt einen Algorithmus, der mit Hilfe eines geheimen Schlüssels, eines Zählers und Hash-Funktionen ein Einmalpasswort berechnet. Der geheime Schlüssel und der Zähler werden für einen Benutzer sowohl im Backend - im Falle von LinOTP im openldap - als auch auf einem Software- oder Hardware-Token gehalten, den der Benutzer mit sich führt. Bei der Authentisierung wird der Benutzer bspw. durch einen Tastendruck auf den Token den Zähler erhöhen und der Token wird einen neuen OTP-Wert berechnen. Diesen tippt der Benutzer bei der Authentisierung in das Passwort-Feld und der OTP-Wert wird mit dem Benutzernamen an der RADIUS-Server gesendet. Der RADIUS-Server ist nun in der Lage, seinerseits mit den abgespeicherten Daten des Benutzers (Geheimer Schlüssel, Zähler) den OTP-Wert zu berechnen. Wenn hier der gleiche OTP-Wert herauskommt, ist klar, dass der Benutzer, der sich authentisieren will, im Besitz des geheimen Schlüssels ist, da er ohne diesen nicht den OTP-Wert berechnen könnte. Ohne Wissen um den geheimen Schlüssel bzw. den Zähler ist es nicht möglich ein weiteres Einmal-Passwort zu berechnen, d.h. es ist aufgrund der Kombination der Hash-Funktionen Stand heute nicht möglich, von einem OTP-Wert auf den nächsten zu schließen.

Wenn nun der Benutzer aber öfters OTP-Werte mit dem Token erzeugt hat ohne sich zu authentisieren, d.h. den OTP-Wert in das Passwort-Feld eingetragen hat, ist der Zähler in seinem Token schon um mehrere Stellen weitergerückt, ohne dass das Backend davon weiß. Er hat bspw. 25 mal ein OTP-wert erzeugt, ohne sich mit diesem zu authentisieren. Bei der nächsten Authentisierung wird der RADIUS-Server von dem Benutzer nun also einen OTP-Wert erhalten, der aus dem Key und dem Zähler (letzter Zählerstand +25) berechnet wurde. Das Backend wird nun aber als erstes den OTP Wert (letzter Zählerstand + 1) berechnen und feststellen, dass die OTP-Werte nicht zusammen passen. Um diesem Problem zu begegnen lässt sich auf Seiten des Backends ein Fenster definieren,

wie viele OTP-Werte das Backend bei der Überprüfung berechnen soll. Wird dieses Fenster auf 50 gesetzt, so kann der Benutzer also 49 mal ein OTP-Wert erzeugen, ohne sich mit diesem authentisieren zu müssen.

Achtung: In die Berechnung des OTP-Wertes fließt nur der geheime Schlüssel und der Zähler ein. D.h. es ist unerheblich wann der OTP-Wert benutzt wird. So kann bspw. der Benutzer den nächsten OTP-Wert erzeugen, sich einen Kaffee holen und erst eine Stunde später mit diesem OTP-Wert authentisieren. Somit ist es also durchaus möglich, sich eine TAN-Liste zu erzeugen.

Bei einer erfolgreichen Authentisierung am Backend wird im Backend der Zähler aber auf den Zählerstand des gerade benutzten OTP-Wertes gesetzt. D.h. ältere OTP-Werte (die zu früheren Zählerständen gehören) sind somit nicht mehr verwendbar.

Beispiel:

Zähler	OTP-Wert
35	839208
36	908472
37	239450
38	839012

Nun authentisiert sich der Benutzer mit dem OTP-Wert 839012. Im Backend wird ebenfalls der Zähler auf 38 gesetzt. die OTP-Werte 35, 36 und 37 werden - wenn man sich mit diesen zu authentisieren versucht - zu einer Ablehnung führen.

Funktionsweise

Der Benutzer authentifiziert sich mit seinem Benutzernamen, einem Passwort und dem OTP-Wert an bspw. einer Webseite oder einer VPN- Verbindung. Dieser RADIUS-Client leitet diese Authentifizierungsdaten an den RADIUS-Server weiter. Der RADIUS-Server, d.h. das RADIUS-Modul von LinOTP, erfragt aus dem LDAP-Verzeichnis den HMAC-OTP-Key des entsprechenden Benutzers und berechnet daraus den nächsten OTP-Wert. Der RADIUS-Server beantwortet die Authentifizierungsanfrage des RADIUS-Clients entsprechend.

Hintergründe zu LinOTP

OTP PIN

LinOTP verwendet zur Authentisierung zusätzlich zum OTP-Wert eine sog. OTP-PIN, die ein statisches Passwort ist. Dieses ist bei der Authentisierung direkt in Passwortfeld vor den OTP-Wert zu schreiben. LinOTP wird im Hintergrund die letzten Acht Stellen des

Feldinhalte als OTP-Wert abschneiden und vergleichen, die vorderen Stellen wird es als festen Passwort-Anteil OTP-PIN überprüfen. So lassen sich die zwei Faktoren

- Wissen (OTP-PIN)
- Besitz (OTP-Wert)

darstellen.

LDAP

LinOTP speichert die für die OTP-Berechnung notwendigen Werte (Geheimer Schlüssel und Zähler) verschlüsselt in einer OpenLDAP Datenbank zu dem zugehörigen Benutzer. Da bei erfolgreicher Authentisierung der Zähler hochgesetzt wird, benötigt das FreeRADIUS Modul von LinOTP Schreibzugriff auf das ZählerAttribut.

Synchronisationsfenster

In LinOTP wird in der Datei

```
1 /etc/freeradius/linotp.conf
```

mit dem Parameter `resync-window-size` das Synchronisationsfenster für alle Benutzer auf einen festen Wert eingestellt.

Lizenzen

Die zur Laufzeit benötigten Komponenten von LinOTP (FreeRADIUS-Plugin und Konnektor-Bibliothek) stehen unter der GNU General Public License (GPL) Version 2. Das Administrationswerkzeug zur Initialisierung der eToken NG OTP, Dokumentation und Skripte sind bei der LSE zu lizenzieren.

Erweiterbarkeit

Auf Basis der GPLv2 haben Sie die Möglichkeit die Funktionsweise des FreeRADIUS-Moduls zur Authentifizierung um neue, eigene Aspekte zu erweitern oder die Software eigenen Audits und Code-Reviews zu unterziehen.

Sicherheitsparameter

Die Sicherheit eines HMAC-OTP-basierten Setups wird bestimmt durch verschiedene (teils konfigurierbare) Parameter.

In der Standardkonfiguration werden folgende Parameter verwendet:

- OTP ist eine 6-stellige Zahl. Dies entspricht der empfohlenen Mindestlänge für HMAC-OTPs (RFC4226) und ist in Verbindung mit Aladdin eToken NG OTP die einzig mögliche Konfiguration.
- OTP PIN ist ein alphanumerisches Passwort mit einer Mindestlänge von 8 Zeichen. Die Mindestlänge ist im Adminwerkzeug (otpadm) mit Hilfe des `-minlen` Parameters konfigurierbar. Wir empfehlen, keine PINs kürzer als 8 Zeichen zu erlauben.
- Maximale Zahl von Fehlversuchen. Dieser Wert steht per Default auf 10 maximalen Fehlversuchen, nach denen der Account im LDAP als gesperrt vermerkt wird. Die Zahl der Fehlversuche ist im Adminwerkzeug (otpadm) mit Hilfe des `-maxfail` Parameters für jeden User separat konfigurierbar.
- HMAC-OTP Synchronisationsfenster. Dieser Wert bestimmt, welche Zählerabweichungen vom FreeRADIUS-Plugin akzeptiert werden dürfen. Der Standardwert beträgt 50. Damit ist eine erfolgreiche Authentifizierung möglich, wenn der eToken NG OTP maximal 50 Schritte von dem im Verzeichnis gespeicherten Stand entfernt ist.

Weiterführende Empfehlungen zur Abwägung Benutzbarkeit/Security und den relevanten HMAC-OTP-Parametern finden Sie in RFC4226.

Geschützte Attributspeicherung

LinOTP schützt die vertraulichen Attribute durch die Verwendung von symmetrischer Verschlüsselung (OTP-Key) und kryptographischen Hashfunktionen (OTP-Pin).

Im Folgenden finden Sie eine Beschreibung der verwendeten Algorithmen.

$$\text{LinOtpPinEnc} = \text{SHA256}(\text{pin} || \text{LinOtpPinSeed})$$

$$\text{LinOtpKeyEnc} = \text{AES256 - CBC}(\text{key}, \text{LinOtpKeyIV}, \text{otpkey})$$

`LinOtpKeyIV` und `LinOtpPinSeed` haben beide eine Größe von 256 bits und werden aus der Zufallszahlenquelle `/dev/random` erzeugt.

2 Anwendungsfälle

LSE LinOTP kann als Backend für viele verschiedene Authentisierungs-Anwendungsfälle verwendet werden. Somit können Sie mit einer Installation Ihres LinOTP-Backends die Sicherheit an mehreren Stellen in Ihrem Netzwerk erhöhen.

Das System, an dem Sie sich mit einem Einmalpasswort anmelden wollen, muss in diesem Fall als RADIUS-Client gegenüber LinOTP auftreten.

2.1 VPN und SSL VPN

Die verbreiteten VPN- oder SSL-VPN-Systeme sind in der Lage, die Benutzer, die eine VPN-Verbindung aufbauen wollen, gegen einen RADIUS-Server zu authentifizieren.

Dies können Produkte wie OpenVPN™, SSLEplorer/Barracuda SSL VPN, phion SSL VPN, SonicWALL® SSL-VPN oder einfach ein Microsoft™ RAS Server sein.

Das SSL-Gateway tritt gegenüber LinOTP als RADIUS-Client auf. Das Gateway sendet zur Authentifizierung nicht mehr Benutzername und Passwort sondern das vom Benutzer eingegebene Einmal-Passwort zum RADIUS-Server.

2.2 Webseiten

LinOTP kann auch zur Authentifizierung von Benutzern an Webseiten herangezogen werden. Dies kann prinzipiell auf zwei Arten geschehen, entweder durch den Webserver oder durch die Web-Applikation.

Im Falle des Apache Webserver kann bspw. mit Hilfe des Moduls `mod_auth_radius` der Zugriff auf bestimmte Bereiche einer Webseite auf OTP-Benutzer beschränkt werden. Der Apache Webserver tritt dann als RADIUS-Client auf. Ähnliches gilt für den Microsoft™ Internet Information Server.

Meistens wird man jedoch granularere, feinere Berechtigungsstrukturen abbildern wollen, so dass die Authentifizierung der Benutzer durch die Web-Applikation zielführender

ist. In einem solchen Fall muss das Authentisierungsmodul der entsprechenden Applikation um die Fähigkeit als RADIUS-Client auftreten zu können erweitert werden. Die verschiedenen Web-Programmiersprachen liefern hierzu entsprechende RADIUS-Bibliotheken.

2.3 Lokale Anmeldung / Domänenanmeldung

Auch die Authentisierung an einem lokalen Arbeitsplatz kann mit LinOTP abgebildet werden. Dies kann auf viele verschiedene Arten und Weisen erfolgen, deren auch nur ansatzweise Betrachtung den Rahmen dieses Handbuchs sprengen würde.

Falls zur Authentifizierung kein Kerberosserver herangezogen wird, so tritt der lokale Client als RADIUS-Client gegenüber LinOTP auf. Im Falle von Unix-ähnlichen Betriebssystemen wird hier `pam_radius_auth` verwendet. Die Pluggable Authentication Modules (PAM) erlauben ein flexibles Vorgehen beim Authentifizieren und Rechtezuweisen der Benutzer. Hierdurch kann ein Benutzer sich an einer Konsole, an einem graphischen Interface oder per SSH an einem Client authentisieren.

Für Windows XP oder 2003 wird ein GINA-Ersatz - für Windows Vista, 2008 oder Windows 7 ein Credential Provider benötigt, der die OTP-Benutzereingaben entgegennimmt und gegenüber LinOTP als RADIUS-Client auftritt. Mit der [LSE RadiusGina](#) hat die LSE ein Produkt im Portfolio, das auch die OTP-basierte starke Authentisierung an Windows-Systemen ermöglicht.

3 Installation

3.1 Ausrollsystem

Das Ausrollsystem dient der Initialisierung der eToken NG OTP und der weiteren Verwaltung der OTP-Attribute im LDAP-Verzeichnis. Dazu steht das LSE LinOTP Adminwerkzeug (otpadm) zur Verfügung.

Für die Installation benötigen Sie die Aladdin Laufzeitumgebung (RTE) sowie die libusb-Bibliothek. Informationen zur Installation der Aladdin Laufzeitumgebung können Sie deren Dokumentation entnehmen. Die libusb-Bibliothek ist in den meisten Distributionen bereits als Paket enthalten und kann einfach installiert werden.

Erfordert: Aladdin RTE 4.55 oder 5.00¹ und libusb0.1 (getestet: 0.1.5, 0.1.8, 0.1.12)

Installieren Sie das mitgelieferte Binary „otpadm“ wahlweise in `/usr/local/bin` oder ein anderes Verzeichnis in Ihrem Suchpfad.

Lizenzdatei

Das Adminwerkzeug (otpadm) erfordert zur Laufzeit eine gültige LSE-Lizenzdatei. Sie erhalten diese beim Erwerb einer Lizenz für LSE LinOTP. Die Lizenzdatei ist personalisiert und nicht übertragbar. Bitte kopieren Sie die Lizenzdatei nach `$HOME/.otpadm-license` oder `/etc/otpadm/license` (für systemweite Nutzung).

3.2 RADIUS-Server

Der RADIUS-Server dient mit Hilfe des LinOTP FreeRADIUS-Plugins (rlm_linotp) der Authentifizierung mittels OTP.

Das FreeRADIUS-Plugin erhalten Sie im Quellcode. Zur Installation benötigen Sie gcc, make, libtool sowie die libusb-, libldap- und libssl-Bibliotheken und deren Header-Dateien.

¹ Für eToken NG-OTP Java 72k wird Aladdin RTE 5.00 benötigt.

Erfordert: libssl (getestet: OpenSSL 0.9.7e, 0.9.8e), libldap2 (getestet: 2.1.30, 2.3.32), FreeRADIUS (getestet: 1.1.0, 1.1.3, 1.1.6, 1.1.7).

Auf Debian-basierten Systemen können Sie die erforderlichen Pakete installieren wie folgt:

```
1 $ apt-get install gcc make libtool libusb-dev libldap2-dev libssl-dev libz-dev
```

Das FreeRADIUS-Plugin nutzt für die LDAP-Kommunikation eine Konnektor-Bibliothek (libotpdb), die ebenfalls im Quellcode ausgeliefert wird und vor dem Plugin installiert werden muss.

Wechseln Sie ins Verzeichnis libotpdb und installieren Sie die Bibliothek wie folgt:

```
1 $ make
2 $ sudo make install
```

Im Standardfall wird die Bibliothek nach `/usr/local` installiert. Auf manchen Systemen ist `/usr/local/lib` nicht im Suchpfad des Linkers eingetragen. In diesem Fall fügen Sie einen Eintrag `/usr/local/bin` in die Datei `/etc/ld.so.conf` ein und rufen `/sbin/ldconfig` auf:

```
1 $ echo /usr/local/lib >> /etc/ld.so.conf
2 $ ldconfig
```

Entpacken Sie die Quellen des FreeRADIUS (`freeradius-1.1.7.tar.gz`) und wechseln Sie in das Verzeichnis:

```
1 $ tar -xvzf freeradius-1.1.7.tar.gz
2 $ cd freeradius-1.1.7
```

Kopieren Sie nun das `rlm_linotp`-Verzeichnis in die Quellen des FreeRADIUS:

```
1 $ cp -r /path/to/rlm_linotp src/modules/rlm_linotp
2 $ echo rlm_linotp >> src/modules/stable
```

Bauen und installieren Sie nun FreeRADIUS wie in der Dokumentation des FreeRADIUS beschrieben.

4 Konfiguration

4.1 Schlüsselerzeugung

LinOTP legt die vertraulichen OTP-Parameter symmetrisch verschlüsselt in der Datenbank ab. Dazu wird ein Schlüssel verwendet, den Sie mit Hilfe von `otpadm` erzeugen können:

```
1 $ otpadm genkey
2 01:00:028bdbc58cf2cbd5f5fa0e19cc0f906a431df37ab7fdca809fb461c96b3768de
```

Bitte notieren Sie diesen Schlüssel oder speichern Sie ihn in einer geschützten Datei. Er wird für die nachfolgenden Konfigurationsschritte benötigt.

4.2 LDAP-Server

Zur Nutzung des `LinOtpAccount`-Schemas sind einige Konfigurationsschritte erforderlich. Es wird angenommen, dass eine Grundinstallation des OpenLDAP `slapd` bereits vorhanden und betriebsbereit ist.

Diese Einstellungen werden in den folgenden Beispielen angenommen; bitte ersetzen Sie diese Werte entsprechend Ihren Anforderungen:

```
1 BASEDN          dc=linotp,dc=com
2 LDAP_HOST       10.0.0.123
3 LDAP_ADMIN      cn=admin,dc=linotp,dc=com
4 LDAP_OTPBASE    ou=users,dc=linotp,dc=com
```

Richten Sie für das RADIUS-Plugin einen separaten Benutzer mit eingeschränkten Rechten ein. Im folgenden Beispiel richten wir den Benutzer `cn=radius-otp,dc=linotp,dc=com` ein. Sie können diesen Benutzer auch mit anderem Namen anlegen.

```
1 $ ldapadd -x -W -h $LDAP_HOST -D $LDAP_ADMIN << END
2 dn: cn=radius-otp,$BASEDN
3 objectClass: simpleSecurityObject
4 objectClass: organizationalRole
5 cn: radius-otp
6 description: RADIUS OTP
7 userPassword: ersetzmichXXX
8 END
```

Wahlweise können Sie für die OTP-Verwaltung mit `otpadm` einen separaten LDAP-Benutzer anlegen, oder einen bereits vorhandenen Admin-Benutzer verwenden. Dieser User muss die notwendigen Rechte besitzen, um die `objectClass` der OTP-Benutzer zu ergänzen und die entsprechenden OTP-Attribute zu schreiben.

Wenn Sie wünschen können Sie an dieser Stelle einen Testuser anlegen, mit dem Sie die Beispiele in dieser Dokumentation nachvollziehen können:

```
1 $ ldapadd -x -W -h $LDAP_HOST -D $LDAP_ADMIN << END
2 dn: ou=users,dc=linotp,dc=com
3 ou: users
4 objectClass: organizationalUnit
5 dn: uid=testuser,ou=users,dc=linotp,dc=com
6 objectClass: posixAccount
7 objectClass: organizationalRole
8 cn: testuser
9 uid: testuser
10 uidNumber: 502
11 gidNumber: 502
12 description: RADIUS OTP testuser
13 homeDirectory: /home/testuser
14 userPassword: lmr!
15 END
```

Kopieren Sie nun die mitgelieferte Schema-Datei `linotp.schema` in das Schema-Verzeichnis Ihrer OpenLDAP-Installation. Das Verzeichnis heisst in der Regel `/etc/ldap/schema` oder `/etc/openldap/schema`.

```
1 $ cp linotp.schema /etc/ldap/schema
```

Bearbeiten Sie jetzt die Konfigurationsdatei des `slapd`-Daemons. Sie findet sich in der Regel in `/etc/ldap/slapd.conf`.

```
1 $ vim /etc/ldap/slapd.conf
```

Suchen Sie in der Datei den Abschnitt, in dem mittels `include` die Schema-Dateien eingebunden werden. Fügen Sie hinter den vorhandenen Zeilen die folgende Zeile ein:

```
1 include /etc/ldap/schema/linotp.schema
```

Stellen Sie sicher, dass das `posixAccountSchema` in Ihrer `slapd`-Konfiguration bekannt und konfiguriert ist.

```
1 # Schema and objectClass definitions
2 include /etc/ldap/schema/core.schema
3 include /etc/ldap/schema/cosine.schema
4 include /etc/ldap/schema/inetorgperson.schema
5 include /etc/ldap/schema/nis.schema
6 include /etc/ldap/schema/linotp.schema
```

Bearbeiten Sie nun die Zugriffsrechte auf die Attribute der Objektklasse `LinOtpAccount`. Dieser Schritt ist sehr wichtig; Bei falscher Rechteverteilung können Unbefugte an sicherheitskritische OTP-Parameter gelangen. Testen Sie deshalb bitte direkt im Anschluss an diesen Schritt die erwartungsgemässe Funktion der Zugriffsbeschränkung. Fügen Sie diesen Block zu den vorhandenen `access to`-Einträgen hinzu:

```
1 access to attrs=@LinOtpAccount
2 by dn="cn=admin,dc=linotp,dc=com" write
3 by dn="cn=radius-otp,dc=linotp,dc=com" write
4 by * none
```

Aus Sicherheitsgründen erfolgt die Kommunikation des Adminwerkzeugs und des RADIUS-Plugins mit dem LDAP-Server nur über einen TLS- geschützten Transportkanal. Damit dieser ordnungsgemäss funktioniert, muss der `slapd`-Daemon entsprechend konfiguriert werden.

Bitte stellen Sie sicher, dass sie mit den Einstellungen `TLSCertificateFile`, `TLSCertificateKeyFile` und `TLSCACertificateFile` in der Datei `slapd.conf` den TLS-Betrieb aktiviert haben. Sie können wahlweise selbstsignierte oder CA-signierte Zertifikate konfigurieren. Stellen Sie nun sicher, dass sie mit `ldapsearch` per TLS auf den LDAP-Server zugreifen können:

```
1 $ ldapsearch -ZZ -h $LDAP_HOST -b $BASEDN "(objectClass=*)" "
```

Sollten Sie bei diesem Schritt auf Schwierigkeiten stossen, lesen Sie bitte auch den Abschnitt 5.1 zu TLS-Problemen.

4.3 Ausrollsystem

Das Adminwerkzeug `otpadm` kann mit Hilfe einer Konfigurationsdatei und mit Kommandozeilenparametern konfiguriert werden.

`otpadm` liest die Konfigurationsdateien `/etc/otpadm/otpadmrc` und `$HOME/.otpadmrc`. Dort können wie im folgenden Beispiel Parameter eingetragen werden. Eine minimale Konfiguration kann so aussehen:

```
1 $ cat $HOME/.otpadmrc
2 --host 10.0.0.123
3 --binddn cn=admin,dc=linotp,dc=com
4 --filter uid=%s,ou=users,dc=linotp,dc=com
5 --enckey 01:00:028bdbbc58cf2cbd5f5fa0e19cc...
```

Die verfügbaren Parameter können Sie sich mit Hilfe der `-help` Funktion von `otpadm` anzeigen lassen. Alle Kommandozeilen-Parameter sind auch in der Konfigurationsdatei gültig.

Bitte beachten: Wenn Sie LDAP- oder TLS-spezifische Einstellungen vornehmen möchten, beachten Sie bitte die veränderten Pfade zu den entsprechenden Konfigurationsdateien:

OpenLDAP-Konfiguration für otpadm:
`/etc/otpadm/openldap/ldap.conf`

OpenSSL-Konfiguration für otpadm:
`/etc/otpadm/ssl`
`/etc/otpadm/ssl/certs`
`/etc/otpadm/ssl/cert.pem`
`/etc/otpadm/ssl/lib/engines`
`/etc/otpadm/ssl/private`

4.4 RADIUS-Server

Konfigurieren Sie nun abschliessend das FreeRADIUS-Plugin (`rlm_linotp`). Machen Sie dazu bitte das Konfigurationsverzeichnis Ihrer FreeRADIUS- Installation ausfindig. Im Normalfall lautet das Verzeichnis `/etc/raddb`, `/etc/freeradius` oder `/usr/local/etc/raddb`.

Kopieren Sie die mitgelieferte Datei `linotp.conf` in das Verzeichnis:

```
1 $ cp linotp.conf /etc/freeradius/
```

Stellen Sie sicher, dass die Datei restriktive Zugriffsrechte hat (0600 oder `-rw-----`). Bearbeiten Sie `linotp.conf` und tragen Sie dort Ihre Daten ein. Erforderlich sind die Parameter `ldaphost`, `binddn`, `bindpw`, `searchfilter` und `enckey`. So kann eine minimale Konfiguration aussehen:

```
1 linotp {
2     ldaphost = 10.0.0.123
3     ldapport = 389
4     binddn = "cn=radius-otp,dc=linotp,dc=com"
5     bindpw = "ersetzmichXXX"
6     searchfilter = "uid=%s,ou=users,dc=linotp,dc=com"
7     resync-window-size = 10
8     enckey = "01:00:028bdb58cf2cbd5f5fa0e19cc..."
9 }
```

Aktivieren Sie nun das `linotp`-Plugin durch Bearbeiten der Datei `radiusd.conf`. Fügen Sie bitte die im folgenden Beispiel angegebenen Einträge in die Abschnitte `modules` und `authenticate` ein:

```
1 modules {
2     ...
3     $INCLUDE ${confdir}/linotp.conf
4 }
```

```
1 authenticate {
2     ...
3     linotp
4     ...
5 }
```

Sie können nun das `linotp`-Plugin als Standard-Method zur Authentifizierung konfigurieren. Selbstverständlich sind auch andere Konfigurationen möglich. Bearbeiten Sie dazu die Datei `users` im FreeRADIUS-Konfigurationsverzeichnis und setzen dort:

```
1 DEFAULT Auth-Type = LinOTP
```

Die Konfiguration ist nun abgeschlossen. Sie können `freeradius` nun starten und beispielsweise mit Hilfe des `radclient`-Programms testen. Selbstverständlich wird dies erst funktionieren, wenn Sie Benutzer mit OTP-Attributen im LDAP angelegt haben:

```
1 $ echo "User-Name = testuser@LOCAL, User-Password = pin926920" | \
2     radclient -s -x 10.0.0.123 auth testing123
```

5 Benutzung

Im normalen Regelbetrieb sind keine Administrativen Eingriffe notwendig. Administrative Aktionen werden nötig wenn,

1. ein Benutzer einen neuen OTP Token bekommen soll (vgl. OTP-Benutzer anlegen),
2. ein Benutzer seinen OTP-Token verloren hat (vgl. OTP-Benutzer sperren),
3. ein Benutzer seine OTP-PIN (Passwort) vergessen hat (vgl. OTP-PIN setzen) oder
4. ein Benutzer zu oft einen OTP-Wert erzeugt hat, ohne ihn zu benutzen (vgl. OTP resynchronisieren).

5.1 OTP-Benutzer anlegen

Soll ein Benutzer in Zukunft einen OTP-Token verwenden können, so müssen für ihn im LDAP die notwendigen OTP-Parameter angelegt werden. Abhängig von der Art des Tokens geschieht dies auf verschiedene Weisen. Der entscheidende Punkt ist, dass im LDAP der gleiche geheime Key angelegt wird, wie dem Benutzer auf einem Token zur Verfügung gestellt wird.

Im Falle von Software-Token wie LSE Mobile OTP oder auch im Falle des Aladdin eTokenNG OTP wird dieser geheime Schlüssel von Ihnen selber erzeugt und ins LDAP geschrieben. Dies stellt außerdem einen entscheidenden Vertrauens- und Sicherheitsaspekt dar.

Andere hardwarebasieren HMAC-OTP Token wie der Aladdin eToken PASS oder der Safeword Alpine sind bereits werkseitig mit einem geheimen Schlüssel bestückt und so angeliefert worden. Der geheime Schlüssel, der ins LDAP geschrieben werden muss, wird mit diesen Token in einer Datei (XML-Datei) mitgeliefert.

Zum Anlegen der OTP-Benutzer wird die LinOTP-Komponente `otpadm` verwendet. Wir gehen davon aus, dass Sie bereits eine Administrationskomponente für Ihre Benutzer im LDAP im Einsatz haben. Daher legt `otpadm` keinen LDAP-Benutzer komplett neu an, sondern weist einem bereits bestehenden LDAP-Benutzer lediglich die notwendigen Attribute zur OTP-Nutzung zu.

Verfügt ein LDAP-Benutzer nicht über die notwendigen LinOTP-Attribute, so wird der RADIUS-Server eine OTP-Authentisierungsanfrage für einen solchen Benutzer ablehnen.

Dieses Beispiel setzt einen vorhandenen LDAP-Benutzer mit Objektklasse `posixAccount` voraus.

Ein eToken NG OTP wird wie folgt initialisiert und mit einem geheimen, zufälligen OTP Key bestückt:

```
1 $ otpadm init testuser
2 Enter PIN or <enter> to autogenerate:
3 Generated PIN '90592109'
4 uid=testuser,ou=users,dc=linotp,dc=com initialized successfully
```

Alternativ kann auch ein schon vorhandener Key importiert werden. Unterstützt wird bisher das Aladdin eToken PASS Format (XML):

```
1 $ otpadm --import-key etpass:/path/to/key.xml:000300000468 --no-init-token init testuser
2 Imported key 'etpass:/path/to/key.xml:000300000468' successfully.
3 Enter OTP PIN or <enter> to autogenerate:
4 Generated PIN '90592109'
5 uid=testuser,ou=users,dc=linotp,dc=com initialized successfully
```

5.2 OTP-Benutzer resynchronisieren

Hat ein Benutzer zu viele OTP-Werte erzeugt, ohne sich mit diesen zu authentisieren, weil er bspw. seinen Bekannten den neuen OTP-Token demonstriert hat und immer wieder den Knopf drückte, oder weil ein Kind das neue Gerät sehr spannend fand, so ist der Zähler des Tokens wahrscheinlich aus dem Synchronisationsfenster herausgelaufen. D.h. die nächste Authentisierung wird fehlschlagen, weil das Backend nicht den gleichen OTP-Wert berechnen kann.

Das LinOTP-Backend muss mit dem OTP-Token resynchronisiert werden. Dies kann mit der LinOTP-Komponente `otpadm` durchgeführt werden.

Hierzu muss der Benutzer lediglich den aktuellen OTP-Wert mitteilen. Zur Resynchronisierung ist kein Zugriff auf den Token erforderlich. LinOTP wird mit Hilfe dieses OTP-Wertes solange den nächsten Wert berechnen, bis es den passenden findet und dann den Zähler im LDAP wieder auf den richtigen Wert setzen.

Rufen Sie dazu `otpadm` auf wie folgt:

```
1 $ otpadm resync testuser <limit> <otp>
2 $ otpadm resync testuser 100000 123123
```

Limit gibt die Zahl der Iterationen an, in denen versucht wird, den Counter in Erfahrung zu bringen.

Sofern Zugriff auf den Token gegeben ist, können Sie auch eine harte Resynchronisierung durchführen. Rufen Sie dazu `otpadm init` auf wie bei der ursprünglichen Initialisierung; Schlüssel, Counter, Failcount und Maxfail werden dabei neu geschrieben.

5.3 OTP-Benutzer PIN ändern

Hat der Benutzer seine OTP-PIN vergessen, so ermöglicht `otpadm` die OTP-PIN des Benutzers zurückzusetzen.

Mit dem „setpin“-Befehl können Sie das statische Passwort eines Benutzers wie folgt ändern:

```
1 $ otpadm setpin testuser
2 Enter PIN or <enter> to autogenerate:
3 Generated PIN '17382716'
4 Setting PIN for 'uid=testuser,ou=users,dc=linotp,dc=com'
```

5.4 OTP-Benutzer sperren und reaktivieren

Hat ein Benutzer seinen OTP-Token verloren oder glaubt er, dass OTP-Passworte, die er noch nicht benutzt hat, kompromittiert sind, so ist es nötig die OTP-Funktionalität dieses Benutzers zu sperren. Hierbei wird nicht der Benutzer komplett gesperrt oder deaktiviert sondern lediglich die OTP-Funktionalitäten. D.h. die Authentisierung via LinOTP ist für diesen Benutzer nicht mehr möglich. Andere LDAP-basierte Authentisierungen (Benutzername und Passwort gegen das LDAP) sind hiervon nicht betroffen.

Die Sperrfunktion „disable“ führt dazu, dass der Benutzer im Verzeichnis als gesperrt markiert und seine Anmeldeversuche an `rlm_linotp` mit dem Status „gesperrt“ abgelehnt werden.

```
1 $ otpadm disable testuser
2 Disabling 'uid=testuser,ou=users,dc=linotp,dc=com'
3 $ otpadm enable testuser
4 Enabling 'uid=testuser,ou=users,dc=linotp,dc=com'
```

5.5 OTP-Benutzer löschen

Mit „remove“ können Sie die OTP-bezogenen Attribute eines Benutzers im LDAP löschen:

```
1 $ otpadm remove testuser
2 Removing 'uid=testuser,ou=users,dc=linotp,dc=com'
```

5.6 Automatisierung

Das Adminwerkzeug `otpadm` unterstützt auch einen nicht-interaktiven Betrieb. Alle Werte, die normalerweise interaktiv abgefragt werden, können auch über einen Filedescriptor eingelesen werden.

Die entsprechenden Parameter sind

```
1 --key-fd <file-descriptor>
2 --otp-pin-fd <file-descriptor>
3 --token-pin-fd <file-descriptor>
```

```
1 $ otpadm --otp-pin-fd 3 --token-pin-fd 4 --key-fd 5 init testuser \
2   3< /path/to/otp-pin-file \
3   4< /path/to/token-pin-file \
4   5< /path/to/key-file
```

Eine Ausnahme bildet das LDAP bind Passwort (`bindpw`). Dieses kann wahlweise als Kommandozeilenparameter (`-bindpw pass`) oder in der Konfigurationsdatei `otpadmrc` übergeben werden.

A Upgrade von LinOTP v1.0 auf v1.1

Die entscheidenden Erweiterungen von Version 1.0 auf Version 1.1 von LinOTP sind:

- Erweitertes LDAP-Schema
- Neue Attribute für OTP-PIN und OTP-Key

Wir stellen Ihnen mit `linotp-ldap-upgrade` ein Werkzeug zur Verfügung, mit dem Sie die Umstellung auf das neue Schema und die neuen Attribute erledigen können.

Folgen Sie dazu bitte diesen Schritten:

1. Neue Version der Schemadatei installieren

Stoppen Sie dazu den LDAP-Server, kopieren Sie die Datei `linotp.schema` aus dem LinOTP v1.1 Installationsarchiv in den Schema-Pfad und starten Sie den LDAP-Server neu.

```
1 $ cp linotp-1.1/linotp.schema /etc/ldap/schema/linotp.schema
```

2. Verschlüsselungsschlüssel erzeugen

Rufen Sie `otpadm` auf, um einen Schlüssel zu erzeugen. Bitte notieren Sie sich die Ausgabe oder lenken Sie sie in eine geschützte Datei um, da der Schlüssel in den nachfolgenden Schritten benötigt wird.

```
1 $ otpadm genkey
2 01:00:028bdb58cf2cbd5f5fa0e19cc0f906a431df37ab7fdca809fb461c96b3768de
```

3. LDAP-Einträge konvertieren

Rufen Sie dazu wie im folgenden Beispiel das mitgelieferte Perl-Script `linotp-ldap-upgrade`¹ auf.

Mit dem Aufruf dieses Scripts werden die Attribute `LinOtpKey` und `LinOtpPin` jedes OTP-Benutzers im LDAP-Verzeichnis ausgelesen und als neue Attribute `LinOtpKeyEnc` bzw. `LinOtpPinHash` geschützt neben den alten Attributen abgelegt.

Ein Beispielaufruf des Migrationsscripts:

¹Unter Umständen müssen fehlende Perl-Module nachinstalliert werden, bevor das Script ausgeführt werden kann. Sämtliche benötigten Module finden Sie auf [CPAN](#).

```
1 $ linotp-ldap-upgrade --ldapuri ldap://ldapserver/ \  
2   --ldapbase dc=linotp,dc=com \  
3   --binddn cn=admin,dc=linotp,dc=com upgrade
```

Dadurch können - mit gewissen Einschränkungen - neue und alte Versionen des FreeRADIUS-Plugin parallel betrieben werden.

Im Parallelbetrieb müssen folgende Einschränkungen beachtet werden:

- Wenn ein User mit otpadm-1.1 ausgerollt wird, werden ihm nur die neuen Attribute zugeordnet. Ein Benutzer, der mit otpadm-1.1 ausgerollt wurde, kann sich also nicht an einem FreeRADIUS authentifizieren, der noch LinOTP 1.0 verwendet.
- Wenn eine Pin mit otpadm-1.1 geändert wird, wirkt sich das auf alte und neue Attribute aus.
- Achtung: Wenn eine Pin mit otpadm-1.0 geändert wird, betrifft das nur alte Attribute. Der Administrator sollte nur noch das otpadm v1.1 verwenden.
- Attribute liegen in diesem Zustand einmal geschützt und einmal ungeschützt im LDAP.

4. Upgrade der FreeRADIUS-Plugins

Installieren Sie auf dem RADIUS-Server die neue v1.1 Version des FreeRADIUS-Plugins.

Bearbeiten Sie die Datei `linotp.conf` und fügen Sie den `enckey`-Parameter mit dem Schlüssel aus dem ersten Schritt der Datei hinzu.

5. LDAP-Einträge finalisieren

Diesen Schritt sollten Sie erst ausführen, wenn alle FreeRADIUS-Server auf die LinOTP Version 1.1 upgegradet sind.

Führen Sie nun erneut das mitgelieferte Perl-Script `linotp-ldap-upgrade` aus, diesmal mit dem Kommando *finalize*.

Dabei werden die alten, ungeschützten Attribute aus dem LDAP gelöscht. **Achtung:** Nach diesem Schritt werden alte Versionen des FreeRADIUS-Plugins (v1.0) nicht mehr funktionieren!

Nach dem erfolgreichen Ausführen des Migrationsscripts ist das Upgrade auf Version v1.1 abgeschlossen.

B Fehlerbehandlung / FAQs

B.1 TLS-Probleme

Sowohl das FreeRADIUS-Plugin als auch das Adminwerkzeug (otpadm) nutzen die TLS-Funktionen der libldap.

Standardmässig erzwingt libldap die Verifizierung des Zertifikats der Gegenstelle, in unserem Fall also des LDAP-Servers. Da in der Regel noch kein CA-Zertifikat installiert ist, kann es deshalb passieren, dass die Verbindung zum LDAP-Server nicht zustande kommt und stattdessen eine Fehlermeldung ähnlich der folgenden angezeigt wird:

```
1 ldap_start_tls_s failed: Connect error (-11)
2 additional info: error:14090086:SSL routines: \
3   SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
```

Sofern das SSL-Zertifikat Ihres LDAP-Servers von einer CA signiert ist, können Sie einfach das CA-Zertifikat der libldap zugänglich machen. Dazu bearbeiten Sie die Datei ldap.conf und fügen einen Eintrag hinzu:

```
1 TLS_CACERT /path/to/cacert.pem
```

Im Fall des FreeRADIUS-Plugins wird die System-installation der libldap genutzt. Die korrekte Konfigurationsdatei ist damit \$HOME/.ldaprc und /usr/local/etc/openldap/ldap.conf bzw. auf manchen Systemen /etc/ldap/ldap.conf oder /etc/openldap/ldap.conf.

Bitte beachten: Im Fall des Adminwerkzeugs (otpadm) wird eine andere Konfigurationsdatei genutzt. Bitte machen Sie die Änderungen entsprechend in der Datei /etc/otpadm/openldap/ldap.conf.

B.2 Einstellungen in ldap.conf werden von otpadm nicht berücksichtigt

Das Adminwerkzeug (otpadm) enthält eine eigene Version der LDAP- und OpenSSL-Bibliotheken. Um Probleme durch mögliche Versionsunterschiede zu verhindern, sind diese Bibliotheken so konfiguriert, dass sie ihre Konfigurationsdateien zur Laufzeit an anderer Stelle suchen.

Statt der üblichen systemweiten Pfade für OpenSSL und OpenLDAP werden stattdessen die Verzeichnisse `/etc/otpadm/ssl` und `/etc/otpadm/openldap` durchsucht. Bitte sehen Sie den Abschnitt zur Konfiguration von `otpadm` für weitere Informationen.

```
/etc/otpadm/ssl/  
/etc/otpadm/openldap/ldap.conf
```

C Lizenztexte

Dieses Produkt schließt auch Software ein, die mit der nachfolgend reproduzierten Lizenz erstellt wurde:

Copyright 1998–2007 The OpenLDAP Foundation
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted only as authorized by the OpenLDAP Public License.

A copy of this license is available in the file LICENSE in the top-level directory of the distribution or, alternatively, at <http://www.OpenLDAP.org/license.html>.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Individual files and/or contributed packages may be copyright by other parties and/or subject to additional restrictions.

This work is derived from the University of Michigan LDAP v3.3 distribution. Information concerning this software is available at <http://www.umich.edu/~dirsvcs/ldap/ldap.html>.

This work also contains materials derived from public sources.

Additional information about OpenLDAP can be obtained at <http://www.openldap.org/>.

Portions Copyright 1998–2006 Kurt D. Zeilenga.
Portions Copyright 1998–2006 Net Boolean Incorporated.
Portions Copyright 2001–2006 IBM Corporation.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted only as authorized by the OpenLDAP Public License.

Portions Copyright 1999–2005 Howard Y.H. Chu.
Portions Copyright 1999–2005 Symas Corporation.
Portions Copyright 1998–2003 Hallvard B. Furuseth.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that this notice is preserved. The names of the copyright holders may not be used to endorse or promote products derived from this software without their specific prior written permission. This software is provided ‘‘as is’’ without express or implied warranty.

Portions Copyright (c) 1992–1996 Regents of the University of Michigan
. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that this notice is preserved and that due credit is given to the University of Michigan at Ann Arbor. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. This software is provided ‘‘as is’’ without express or implied warranty
.

The OpenLDAP Public License
Version 2.8, 17 August 2003

Redistribution and use of this software and associated documentation (‘‘Software’’), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source form must retain copyright statements and notices,
2. Redistributions in binary form must reproduce applicable copyright statements and notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution, and
3. Redistributions must contain a verbatim copy of this document.

The OpenLDAP Foundation may revise this license from time to time. Each revision is distinguished by a version number. You may use this Software under terms of this license revision or under the

terms of any subsequent revision of the license.

THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S) OF THE SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The names of the authors and copyright holders must not be used in advertising or otherwise to promote the sale, use or other dealing in this Software without specific, written prior permission. Title to copyright in this Software shall at all times remain with copyright holders.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Copyright 1999–2003 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.

Copyright (c) 1998–2007 The OpenSSL Project.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the

OpenSSL Toolkit. (<http://www.openssl.org/>)”

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)”

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ‘‘AS IS’’ AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Copyright (C) 1995–1998 Eric Young (eay@cryptsoft.com)
All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the

same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)". The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-)
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement:
"This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG 'AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

SOFTWARE-LIZENZVERTRAG II

Die Module "LinOTP Konnektorbibliothek"/ "libotpdb" und "LinOTP FreeRADIUS-Plugin"/ "rlm_linotp" erhalten Sie unter den Bedingungen der nachfolgend reproduzierten Lizenz (GNU General Public License)

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights , we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights . These restrictions translate to certain responsibilities for you if you distribute copies of the software , or if you modify it .

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software , and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software .

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original , so that any problems introduced by others will not reflect on the original authors' reputations .

Finally , any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses , in effect making the program proprietary. To prevent this , we have made it clear that any patent must be licensed for everyone's free use or not licensed at all .

The precise terms and conditions for copying, distribution and modification follow .

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it , either verbatim or with modifications and/or translated into another

language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for

making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not

excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any

later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS