Zur Person

Michael Ströder

- Freiberuflicher Berater
- Schwerpunkte Verzeichnisdienste & IT-Sicherheit
  - LDAP / X.500
  - Benutzerverwaltung (Identity Management / Provisioning)
  - PKI / X.509, Verschlüsselung, Digitale Signatur, VPN
  - Single Password, Single Sign-On
- OSS-Projekte im LDAP-Umfeld
  - python-ldap
  - web2ldap

Übersicht

I. Anwendungsbeispiele

II. Ausgewählte Client-Software

III. LDAP-Bind-Methoden

IV. Use-Cases mit verschiedenen Clients durchführen

V. Diskussion

Sicherheit: Simple Bind

- Übertragung von Distinguished Name (Bind-DN) und Credential (Passwort)
- Keinerlei Schutz im Protokoll gegen Abhören => Einsatz von SSL/TLS!
- Weit verbreiteter Einsatz, da oft nur Simple Bind unterstützt von Drittanbietern

Sicherheit: SASL Bind

- Generische Schicht für beliebige Mechanismen (Simple Authentication and Security Layer)
- Passwortbasierte Mechanismen
  - PLAIN
  - CRAM-MD5, DIGEST-MD5 (Challenge-Response)
- andere Mechanismen:
  - GSSAPI (für Kerberos V)
  - EXTERNAL (für SSL-Client-Zertifikate oder Unix Domain Socket)
- SASL Mech. gemäß Sicherheitsanforderungen wählen!
- Oft nur schlechte Unterstützung durch Dritthersteller

Sicherheit: Passwortspeicherung

- In Server-Datenbank (DIB):
  - nur als Hash-Wert (vorzugsweise *salted SHA-1* oder stärker), geeignet nur für Simple Bind
  - im Klartext, erforderlich für Challenge-Response-Authentifizierung (z.B. SASL Bind mit *DIGEST-MD5*)
- Client-seitige Konfiguration enthält Passwort meist im Klartext oder sehr selten client-seitig reversibel verschlüsselt

Sicherheit: Autorisierung

- Server-seitige Zugriffsrechte bevorzugt!
- Kein herstellerübergreifender Standard für ACLs
- => Rechteabfrage durch Client nur sehr rudimentär und proprietär
- => Schwierigkeiten für benutzerfreundliche, herstellerunabhängige LDAP-Clients, da gesperrte Eingabefelder nicht „ausgegraut“ werden können

Schema: Subschema Subentry

- Spezieller Eintrag enthält Schemainformationen
- Korrekte Abfrage via LDAP:
  - Abfrage des DN im Attribut *subschemaSubentry*
  - Explizite Auslesen der Schemaelementattribute: attributeTypes, dITStructureRules, ditContentRules, matchingRuleUse, matchingRules, nameForms, objectClasses
- Ein generischer LDAP-Browser kann sich aber in der Praxis nicht auf die Vollständigkeit des Schema verlassen. :-(

Datenformate: LDIF

- Textformat zur Repräsentation von LDAP-Daten und -Operationen
- RFC 2849
- Verschiedene Trenner für Attributwert-Encoding: ':' , '::', ':<'
- sieht erst mal simpel aus, aber nur in Sonderfällen mit einfachen Mitteln vollständig korrekt zu manipulieren

LDAP-Clients: Shell-Tools (1)

- OpenLDAP liefert Kommandozeilen-Tools (siehe /opt/openldap-RE24/bin/ in VMWare-Image)
  - ldapsearch: Suche nach Einträgen
  - ldapadd: Einträge neu anlegen (simples LDIF)
    - ldapmodify: Modifizieren (LDIF mit Change Records)
    - ldapdelete: Löschen von Einträgen
  - ldapwhoami: Abfrage der Identität nach Authentifizierung (Tests für SASL Bind inkl. Mapping!)
  - ldappasswd: LDAP Modify Password ext. op.

LDAP-Clients: Shell-Tools (2)

- Alle Tools haben gleiche Optionen für Authentifizierung
- Geeignet für...
  - Initialisierung mit wenigen Daten
  - simple Dokumentation in Betriebskonzept ohne weitere Abhängigkeiten
  - Grundlegende Funktionstests
- Nicht geeignet für robuste Skripte (z.B. kontinuerlicher Datenabgleich)
- Verwendung einer Skriptsprache für robustere Daten- und Fehlerbehandlung (Perl, Python, Ruby etc.)

LDAP-Clients: Generische Clients (1)

- Clients mit Schema-Support (auf VMWare-Image installiert):
  - JXPlorer http://www.jxplorer.org
    - Apache Directory Studio http://directory.apache.org
  - web2ldap http://www.web2ldap.de
    (ich bin voreingenommen ;-)
- Weitere: phpldapadmin, gawor LDAPBrowser (buggy, kein Schema-Support), gq (buggy)

LDAP-Clients: Generische Clients (2)

- Generische Clients können für Start in LDAP-Administration nützlich sein
- Besser angepasste Tools für tägliche Administration
- Problemfälle:
  - Select-Listen zur Vermeidung von Fehleingaben
  - Zusammengesetzte Attributwerte
  - Ähnliche Daten
  - Wiederkehrende Abläufe
  - Alias-NAMEn für Attributtypen (*uid* und *userid*)

LDAP-Clients: Angepasste Clients

- Customizing in JXplorer und Apache Directory Studio
- Als Beispiel hier Anpassungen in web2ldap (siehe auch siehe Präsentation ODD 2006)
  - LDIF- und HTML-Templates
  - Plugin-Klassen
- GOsa² (siehe http://www.gosa-project.org): sehr mächtig mit Nebenaktionen (erfordert Konfigurationsaufwand und Schema-Anpassungen im Server)
- Selbst implementierte Clients für eigenes Schema und Use-Cases. Kann wider Erwarten zeitsparendste Lösung sein!

LDAP-Clients: Adressbücher

- Spezifisch zur Abfrage von Adressdaten
- Mozilla Messenger / Seamonkey / Thunderbird für Abfrage der E-Mail-Adresse, Einbindung ins Adressbuch
- OpenOffice
- Problemfälle:
  - kein Schreibzugriff zur Pflege von Adressdaten
  - mehrwertige Attribute (z.B. mehrere E-Mail-Adressen)

Use-Case: Benutzer neu anlegen

- Auswahl strukturelle Objektklasse und ggf. Hilfsobjektklassen
- Festlegung des RDN
- Befüllen der Attribute

Use-Case: Benutzer modifizieren

- Auswahl weiterer Hilfsobjektklassen
- Befüllen/Ändern der Attribute

Use-Case: Passwort setzen (1)

- In OpenLDAP setzen des Attributs *userPassword*
- LDAP Modify Request
  - direktes Ändern des Attributs *userPassword*
  - ggf. client-seitiges Hashing
- Modify Password Extended Operation (server-seitig)
  - Aufruf einer server-seitigen Routine
  - Server-seitiges Hashing (slapd.conf: *password-hash*)
- Ggf. weitere Passwort-relevante Attribute für Passwortrichtlinie, Samba und/oder Kerberos

Use-Case: Passwort setzen (2)

Unterschiedliche Fälle:

- Benutzer ändert Kennwort selbst
  (altes Passwort meist bekannt)
- Admin ändert Kennwort für Benutzer
  (altes Passwort nicht bekannt)

Use-Case: Pflege Gruppen

- Unterschiedliche Gruppeneinträge:
  - *groupOfNames*:
    DN des Mitglieds in Attribut *member*
  - *posixGroup*:
    uid des Mitglieds in Attribut *memberUid*
  - andere proprietäre Schemata
- Problemfälle:
  - Übersichtliche Darstellung vieler Gruppen
  - Handhabung grossen Gruppen
  - Konkurrente Schreibzugriffe auf Gruppeneinträge

Programmierung: APIs

LDAP-Module für viele Programmiersprachen:

- LDAP C SDK von OpenLDAP, Microsoft, Sun/Mozilla
- Java (JNDI, Java LDAP SDK)
- Python (python-ldap.sf.net)
- Perl
- php-ldap (Autoren erweitern nicht mehr)