Vysoké učení technické v Brně
Fakulta informačních technologií

Síťové aplikace a správa sítí – Projekt

LDAP server

November 19, 2017 Róbert Kolcún, xkolcu00

Contents

1 Úvod **2**
1.1 Cieľ projektu 2

2 Prehľad problematiky **3**
2.1 ASN kódovanie 3
2.1.1 Typ správy 3
2.1.2 Dĺžka správy 3
2.2 LDAP správy 3

3 Implementácia **5**
3.1 Rozšírenia 5
3.2 Server 5
3.3 Rozpoznanie správ a spracovanie 5
3.4 Filtre 5
3.5 Pomocné funkcie 5

4 Testovanie **7**

5 Spustenie a použitie programu **8**
5.1 Kompilácia 8
5.2 Spustenie 8

1 Úvod

1.1 Cieľ projektu

Cieľom tohto projektu je implementácia jednoduchého paralelného LDAP servera (LDAPv2), ktorý ma podporovať správy typu Bind Request, Bind Resposnse, Search Request, Search Response Entry, Search Response Done a Unbind Request. Server ďalej podporuje filtre typu And, Or, Not, Equality Match a Substring
Server je implementovaný v programovacom jazyku C++ ako konzolová aplikácia, ktorá podporuje vstupné argumenty port a súbor so vstupnou databázou.

2 Prehľad problematiky

Lightweight Directory Access Protocol (LDAP), je štandardizovaný aplikačný protokol navrhnutý na dotazovanie a modifikáciu adresárových služieb. Je vhodný na udržiavanie adresárov a prácu s informáciami o používateľoch, npr. na vyhľadávanie používateľov v príslušných adresároch. Protokol LDAP je založený na odporúčaní X.500. [2]
Komunikácia medzi serverom a klientom je kódovana pomocou ASN[1] ktorý pozostáva z hexadecimálnych hodnôt.

2.1 ASN kódovanie

2.1.1 Typ správy

Jeden z hlavných dôvodov úspechu ASN.1[1], je že táto notácia je spojená s niekoľkými štandardizovanými kódovacimi pravidlami, ako BER(Basic Encoding Rules) alebo PER(Packed Encoding Rules) ktoré sa ukázali ako užitočné pre aplikácie z hľadiska obmedzenia šírky pásma. [4] Hodnoty sú posielané v hexadecimálnom formáte, pri komunikácií sa najprv posiela typ hodnoty, npr. 0x02 - reprezentuje ze nasledujúci blok dát pozostáva z číselných hodnôt. Základne typy:

- **0x01** - obsah booleovej hodnoty
- **0x02** - obsah číselných hodnôt
- **0x30** - obsah je sekvencia - kolekcia hodnôt rôznych dátovych typov
- **0x0a** - obsah je sekvencia - kolekcia hodnôt rovnakého dátoveho typu

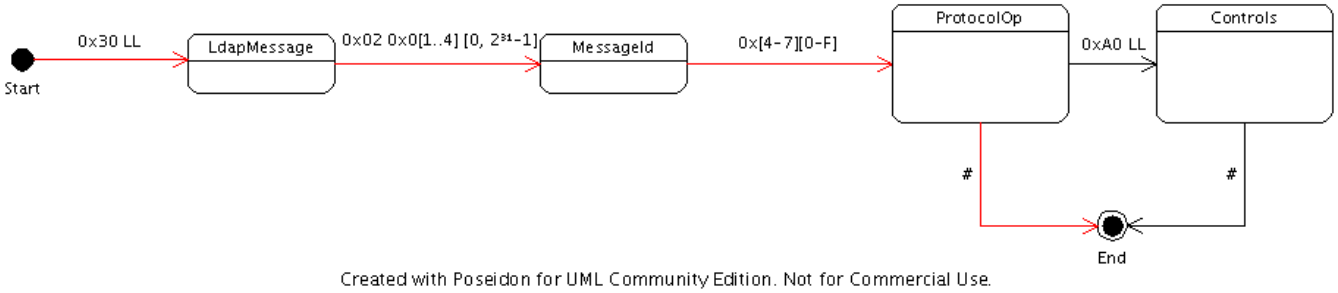
2.1.2 Dĺžka správy

Po určeni typu správy nasleduje jej dĺžka. Číselná dĺžka je reprezentovaná dvoma spôsobmi:

- **1.** Pri hodnotách dĺžky do 127(vrátane) je dĺžka uložená len na jednom bajte
- **2.** Pri hodnotách dlžky viac ako 127. Hodnota prvotného bajtu mínus 127 určuje počet bajtov na koľkých je daná dĺžka kódovaná. A následne sekvencia bajtov na ktorých sa nachádza potrebná dĺžka správy.

2.2 LDAP správy

Poradie príjimania správ nie je fixné, je možné poslať niekoľko SearchRequest požiadavkov, na každý požiadavok server zašle zodpovedajúcu odpoveď taktiež nieje potrebné aby prvá správa bola Bind Request. Každá LDAP komunikácia je zabalená do LDAP správy, vid. obrázok[5]



- **Bind Request** - Táto správa inicializuje komunikáciu medzi klientom a serverom. Na túto správu server odpovedá správou Bind Response

- **Bind Response** - Správa reprezentujúca odpoveď od servera pre potvrdenie naviazanej komunikácie. Server je pripravený komunikovať s klientom.
- **Search Request** - Správa od klienta obsahujúca filter, podľa ktoré ma server nájsť údaje vo svojej databáze. Odpoveďou s výsledkami je správa typu Search Result Entry. Pri zadávaní filtra a vyhľadávani záznamov v databáze sa nerozlišuje veľkosť písmen(case insensitive). Záznam v databáze obsahuje trojicu prvkov - Meno(cn, CommonName), Login (uid, UserID) a Email(mail).
- **Search Result Entry** - Správa od severa obsahujúca čiastkový výsledok vyhľadávania podľa požiadavkov klienta. Správa je zaslaná pre každy nájdeny záznam samostatne. Server podporuje SizeLimit, pri jeho zadaní sa pošle maximálne stanovený počet týchto správ. Správa obsahuje cn a email záznamu, objectName obsahuje uid záznamu.
- **Search Result Done** - Správa pre klienta indikujúca zaslanie všetkých nájdených záznamom a ukončenie vyhľadávania. Odosiela sa aj pokiaľ nebol nájdeny žiaden záznam v databáze.
- **Unbind Request** - Správa od klienta ukončujúca spojenie so serverom. Server na túto správu neodpovedá a uzatvára spojenie.

3 Implementácia

Program je implementovaný v programovacom jazyku C++ so štandartom c++11, ako konzolová aplikácia. Program príjima 2 argumenty, port ktorý je volitelný, pri neuvedení program pracuje na porte 389 a súbor s databázou záznamov. Server bol implementovaný pomocou RFC4511[3] a Apache stránky[5]

3.1 Rozšírenia

Program podporuje kódovanie UTF-8.

3.2 Server

Každý klient je spracovaný paralelne, funkcia *process_client*() je spustená v novom vlákne a zodpovedá za celé spracovanie klientovych požiadaviek. Server podporuje len protokol IPv4.

3.3 Rozpoznanie správ a spracovanie

Správy od klienta sa rozpoznávaju v cykle *while*, pokiaľ nenastane chyba alebo nepríde od klienta Unbind Request. Hlavička správy sa spracuje funkciou *parse_message*(), ktorá následne zavolá funkciu podľa požiadavky klienta:

- **process_bind_request()** - funkcia spracuje Bind Request a následne odošle Bind Response
- **process_search_request()** - spracovanie Search Request požiadavku, načítanie filtrov pomocou *process_filters*(), vyhľadanie záznamov v databáze *get_database_vector*() a následne zaslanie Search Result Entry správ a Search Result Done správy
- **process_unbind_request()** - spracovanie Unbind Request, vratí false a ukončí sa spracovanie klientovych požiadaviek

3.4 Filtre

Implementácia podporuje filtre And, Or, Not, Equality match a Substring. Trieda *Filter* obsahuje dlžku filtra, typ filtra, vektor filtrov pre stromovú štruktúru, boolean hodnotu pre nastavenie chyby v spracovani filtra, map-u atribútov pre Substring a Equality match:

- **process_filters()** - spracuje filtre v Search Request-e a uloží ich do stromovej štruktúry filtrov
- **get_database_vector()** - vratí vector záznamov ktoré vyhovujú vstupnému filtru

3.5 Pomocné funkcie

- **print_stderr_message()** - vypíše chybovú hlášku na stderr
- **read_char()** - prečitá jeden znak zo socketu
- **write_hex_message()** - vypíše reťazec v hexadecimálnom formáte
- **read_ll()** - prečíta dĺžku z LDAP správy
- **read_message_id()** - prečíta message_id z LDAP správy
- **read_message()** - prečíta správu podľa vstupnej dĺžky
- **hex_to_string_char()** - konvertuje vstupný znak do reťazca

- **create_message_id()** - vytvorí message_id podľa vstupnej hodnoty
- **create_ll()** - zo vstupnej hodnoty vytvorí reťazec obsahujúci dĺžku

4 Testovanie

Program bol testovaný priebežne počas vývoja, pomocou nástrojov Wireshark - kontrola formátu správy a ldapsearch - testovanie filtrov a správnosť vrátených výsledkov. Ako testovacia databáza bola použitá databáza poskytnutá pri zadaní projektu vo WIS-e s prekonvertovaným kódovaním do UTF-8. Program bol otestovaný na školskom servery Merlin, pomocou požiadavok zo servera Eva a nástroja ldapsearch, taktiež bol otestovaný a vyvíjaný na stroji s OS Ubuntu 17.10.

5 Spustenie a použitie programu

5.1 Kompilácia

Pomocou príkazu make je možné daný program preložiť.
Makefile ma niekoľko parametrov:

- **myldap** - Preloženie programu
- **debug** - Preloženie programu s DEBUG výpismi
- **clean** - Vymazanie preloženého programu
- **tar** - Vytvorenie xkolcu00.tar archivu pre odovzdanie

5.2 Spustenie

```
$ ./myldap {-p <port>} -f <súbor>
```

- **-p** <**port**> - Umožnuje špecifikovať konkrétny port, na ktorom má server poslúchať požiadavky klientov. Argument je volitelný a predvolená hodnota portu je 389.
- **-f** <**súbor**> - Cesta k vstupnému textovému súboru vo formáte CSV. Tento argument je povinný

Príklad spustenia

```
$ ./myldap -p 12345 -f isa2017-ldap.csv
```

References

[1] *Introduction to ASN.1* [online]. Posledná zmena: -. Dostupné na: <http://www.itu.int/en/ITU-T/asn1/Pages/introduction.aspx>.

[2] *LDAP* [online]. Posledná zmena: 8.11.2017. Dostupné na: <https://cs.wikipedia.org/wiki/LDAP>.

[3] J. Sermersheim, E. *RFC 4511* [online]. Posledná zmena: Jún 2006. Dostupné na: <https://www.ietf.org/rfc/rfc4511.txt>.

[4] Jr., B. S. K. *A Layman's Guide to a Subset of ASN.1, BER, and DER* [online]. Posledná zmena: 1.11.1993. Dostupné na: <http://luca.ntop.org/Teaching/Appunti/asn1.html>.

[5] Lécharny, E. *Ldap ASN.1 Codec* [online]. Posledná zmena: 5.10.2006. Dostupné na: <https://cwiki.apache.org/confluence/display/DIRxSRVx10/Ldap+ASN.1+Codec>.