Piotr Kowalski
Wydział Matematyki, Informatyki i Mechaniki

Szyfrowanie połączeń

między serwerem Oracle i klientem

Warszawa, 24.11.2004

1. Wstęp

Przy współczesnym powszechnym i ciągłym przepływie informacji w Internecie trudno obejść się bez skutecznych zabezpieczeń. W tej pracy chciałbym przedstawić kilka sposobów zabezpieczania transmisji danych od serwera Oracla do jego klienta.

Z powodu wysokich kosztów i braku możliwości instalacji Advanced Security Option, skorzystałem z opcji szyfrowania przez stunnel, dostępny na zasadzie licencji GNU (General Public License). Kod można dowolnie modyfikować. Kolejną zaletą jest wyjątkowo prosta konfiguracja dla kogoś, kto kiedykolwiek dokonywał szyfrowania transmisji pomiędzy dwoma punktami. Wadą jest niestety to, że na każdej końcówce klienckiej musimy wprowadzić niestandardową konfigurację.

2. Instalacja

Na początku sprawdzamy, czy mamy zainstalowaną bibliotekę SSL – OpenSSL lub SSLeay. Prawie we wszystkich obecnie używanych systemach taka biblioteka istnieje i nie musimy się o nią martwić. W niektórych przypadkach – jeśli biblioteka nie znajduje się w */usr/local/ssl* (spotkałem się z tym w BSD) – musimy podać ścieżkę do tej biblioteki. Robimy to poprzez dodanie do `./configure` opcji `--with-ssl=DIR`, gdzie *DIR* zastępujemy naszą lokalizacją bibliotek. Następnie ściągamy pakiet stunnel lub korzystamy z wbudowanego w niektóre dystrybucje (np. w RedHat 9.0 mamy od razu wersję 4.04). Należy pamiętać, że pakiet do wersji 4.02 zawierał pewne błędy, co ze względów bezpieczeństwa dyskwalifikuje wcześniejsze wersje. Z drugiej strony zawsze dobrze jest mieć najnowszą wersję – pozwoli to nam uniknąć wielu niepotrzebnych problemów.

Pakiet stunnel możemy ściągnąć ze strony producenta dla wersji pod Microsoft Windows: http://www.stunnel.org/download/binaries.html. Dla niektórych dystrybucji np. PLD mamy już przygotowane gotowe pakiety, więc warto sprawdzić czy taki istnieje. Pod pozostałe systemy ściągamy źródła ze strony: http://www.stunnel.org/download/source.html i kompilujemy. Jest to opisane w pliku INSTALL, ale w większości przypadków ogranicza się do wykonania poleceń:

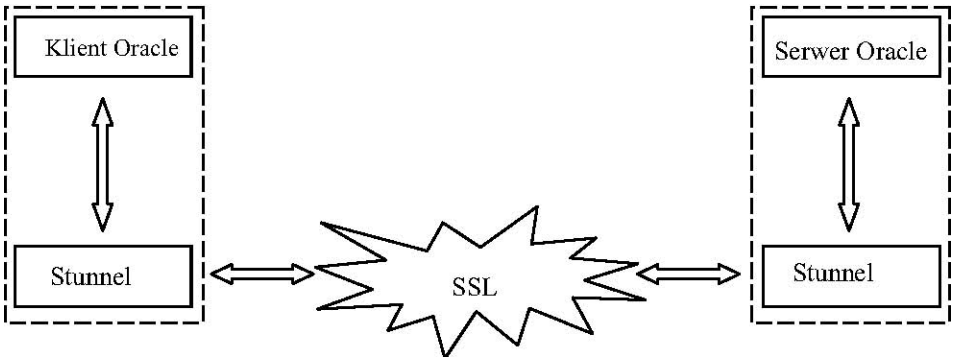
```
./configure lub ./configure –with-ssl=”ścieżka do bibliotek ssl”
make
make install
```

Należy pamiętać o certyfikacie dla stunnel. Jest to chyba najczęstsza przyczyna braku działania stunnel. Jeśli ktoś nie potrafi tworzyć certyfikatów, to może skorzystać z instrukcji w języku polskim – znajduje się ona w źródłach w katalogu *doc/pl/* plik *tworzenie_certyfikatow.html*. Warto przeczytać też w tym katalogu *faq.stunnel-2.html*, który opisuje najczęstsze problemy i często wystarcza jeśli jakieś się pojawią. Dla osób, które pragną tworzyć bardziej skomplikowane certyfikaty lub np. selektywny dostęp, dobrą pozycją jest [9].

Następnie tworzymy plik konfiguracyjny. Jeśli planujemy stworzyć tunel tylko do Oracla, to wystarczy wykorzystać podany tu opis. Jeśli jest to za mało, to zapraszam na stronę http://www.stunnel.org/

Opis konfiguracji stunnela do połączenia dla bazy Oraclowej zaczerpnąłem z pracy [1]. Dołączyłem też pewne uzupełnienia ze strony http://www.stunnel.org i grup dyskusyjnych.

Aby wdrożyć to rozwiązanie, należy odpowiednio skonfigurować pakiet Stunnel, wygenerować odpowiednie certyfikaty, a następnie zmienić konfigurację sieci Net8. Konfiguracja ta ma zapewniać łączenie się klienta po interfejsie lokalnym z programem stunnel, a po stronie serwera nasłuch procesu listenera na interfejsie lokalnym. Ilustruje to rysunek 1.



Rysunek 1 Komunikacja po interfejsie lokalnym bez szyfrowania

Przykładowo:

Plik TNSNAMES.ORA na stacji klienckiej

```
SECUR.KATOWICE.ALTKOM =
(DESCRIPTION =
```

```
(ADDRESS_LIST =

      (ADDRESS = (PROTOCOL = TCP)(HOST = 127.0.0.1)(PORT = 20000)) )
(CONNECT_DATA =
(SERVICE_NAME = secur)
)
)
```

Oprócz tego należy uruchomić program stunnel po stronie klienta z następującymi parametrami:

```
stunnel -c -d 20000 -r serwer.oracle:port_stunnel
```

Natomiast po stronie serwera konfiguracja listenera wygląda następująco:

```
LISTENER =
(DESCRIPTION_LIST =
(DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
)
)
SID_LIST_LISTENER =
(SID_LIST =

(SID_DESC =
(GLOBAL_DBNAME = secur)
      (ORACLE_HOME = /vol/oracle)
(SID_NAME = secur)

)
)
```

Oprócz tego należy uruchomić program stunnel z następującymi parametrami:

```
stunnel -d port_stunnel -r localhost:1521
```

Taka konfiguracja zapewni szyfrowanie połączeń pomiędzy serwerem a klientem Oracle. Dzięki możliwościom programu Stunnel, oprócz szyfrowania połączeń, można również dokonywać weryfikacji stacji klienckich na podstawie ich certyfikatów.

Polecam też przeczytanie dyskusji na temat konfiguracji na stronie http://www.stunnel.org/examples/oracle.html, która dość dokładnie pokazuje różne problemy mogące się pojawić w takiej konfiguracji.

3. Testy

Do testów użyłem przede wszystkim tabeli z osobami (`select * from dz_osoby;`) – była dość duża oraz nie obciążała obliczeniowo serwera. Te dwa warunki były dla mnie najważniejsze, gdyż dzięki temu otrzymałem wyniki, na które nie wpłynęły obliczenia serwera. Pierwotnie chciałem użyć zapytań wykorzystywanych do normalnej pracy. Niestety przez to, że są bardzo złożone obliczeniowo, wydaje mi się, że mogą wprowadzać niepotrzebne zafałszowania, które są skutkiem obliczania wyników przez serwer. Z drugiej strony są one dużo bardziej zbliżone do rzeczywistości niż prosty select, który przekazuje wszystkie dane z tabeli osoby, więc z nimi też wykonałem kilka testów.

Testy były wykonywane z włączonym i wyłączonym tunelowaniem przez stunnel, po godzinach pracy uczelni. Każdy z serii testów poprzedzałem jednym takim samym testem, aby serwer mógł wrzucić sobie wszystko do pamięci podręcznej i jego obliczenia nie wpływały na obciążenie. Parametry serwera: 2 procesory Pentium III (Cascades) 700 MHz, 2 MB cache, 512 MB pamięci, serwer w „stanie spoczynku” zużywał 1,9% mocy procesora. Do testów była użyta silna stacja robocza z zainstalowanym Microsoft Windows 2003, aby braki jej mocy nie powodowały, że serwer musi czekać na odebranie danych.

Dla dużej tabeli (z osobami) średnia różnica obciążeń z 5 prób (5 dla włączonego i 5 dla wyłączonego tunelowanie) wyniosła 2,1% obciążenia serwera. Jednak czas przesyłania całości zmniejszył się z 2:57 (bez szyfrowania) do 2:53 (z włączonym tunelowaniem).

Nie potrafię stwierdzić, czy było to wynikiem mniejszej ilości danych wysyłanych przez stunnel, czy sieć była wtedy mniej obciążona. Dla wielu zapytań maksymalna różnica wyniosła 4,7%. Wynikało to prawdopodobnie z tego, że na początku ustalają się parametry transmisji, a samej "treści" jest proporcjonalnie mało. Tutaj czasu właściwie nie dawało się mierzyć. W testach minimalna zmierzona różnica wyniosła 1,9%. Wydaje mi się, że nie jest to dużo i wyniki są zadowalające.

Wszystkich, którzy są zainteresowani dokładnymi danymi dla różnego rodzaju szyfrowań zapraszam na stronę http://stunnel.mirt.net/perf.html, gdzie jest tabela porównawcza.

4. Podsumowanie

Zalecam włączenie szyfrowania danych pomiędzy serwerem a klientem USOS. Narzut na dodatkową pracę serwera nie jest duży, a zyskujemy dużo większe bezpieczeństwo transmisji.

Niestety za bezpieczeństwo trochę płacimy. W przypadku awarii połączenia mamy już dwie możliwości wystąpienia błędu – Oracle lub stunnel. Nie możemy w nieskończoność zostawiać otwartych nieaktywnych sesji (domyślnie 12 godzin) – po tym czasie połączenie jest zrywane.

Kolejnym małym (ale miłym) plusem stunnel jest to, że jego korzenie są polskie. Jest to chyba jeden z najpopularniejszych „polskich” pakietów dodawanych do prawie wszystkich dystrybucji Linuksa.

5. Bibliografia

1. Advanced Security Option i inne metody szyfrowania połączeń w Oracle 9i Marcin Przepiórowski (Altkom Akademia S.A.) (http://www.ploug.org.pl/seminarium/seminarium_VIII/pliki/aso.pdf)
2. Nieoficjalna strona domowa *stunnel* (http://www.stunnel.org)
3. Strona domowa programu *stunnel* (http://stunnel.mirt.net/)
4. Opis składni polecenia *stunnel* po polsku (http://stunnel.mirt.net/static/stunnel.pl.html)
5. Archiwum grupy pl.comp.security (http://niusy.onet.pl/niusy.html?t=archiwum&group=pl.comp.security)
6. Archiwum grupy pl.comp.bazy-danych (http://niusy.onet.pl/niusy.html?t=archiwum&group=pl.comp.bazy-danych)
7. Opis instalacji szyfrowania połączenia Oracle (http://www.stunnel.org/examples/oracle.html)
8. Strona domowa firmy Oracle (http://www.oracle.com)
9. Tworzenie certyfikatów (http://thom.artcom.pl/tworzenie_certyfikatow.html)