

# DbProxy

## Zaimplementowana funkcjonalno

- utrzymywanie kilku starych poacie do baz danych
- eliminacja bezpoednich pocze z checków i odcienie baz danych
- proksowanie dowolnych zapyta sql, mechanizm cache przyspiesza wykonywanie zapyta
- obsuga wielu baz danych, dokladnie tych do których zainstalowane s drivery DBI np: Oracle, Pg(postgres), mysql
- wielotkow, obsuga dowolnej ilosci checków w tym samym momencie, ograniczeniem jest liczba pocze z bazami
- opcja komunikacji szyfrowanej z proxy przy pomocy algorytmu HMAC
- hasa baz danych sa zakodowane algorytmem HMAC w pliku konfiguracyjnym kluczem z opcji "cipherpass", nie musimy ich wogóle podawa w checkach

## Informacje

- kod i skrypt wykonywalny znajduje sie na cw w pliku `/usr/local/nagios/avantis/sbin/dbproxy.pl`
- plik konfiguracyjny znajduje si `/usr/local/nagios/avantis/etc/dbproxy.cfg`, opisy parametrów w pliku
- pliku logu znajduje si w `/usr/local/nagios/avantis/var/log/dbproxy.log`
- plik z pidem uruchomionego procesu znajduje si w `/usr/local/nagios/avantis/var/run/dbproxy.pid`
- plik z kluczem szyfrującym komunikacje z proxy znajduje si w `/usr/local/nagios/avantis/etc/ciphernet.txt`
- dbproxy jest uruchamiane automatycznie po restarcie cw w skrypcie `/etc/rc.local`
- dostępne s polecenia (s to skrypty w `/usr/local/nagios/avantis/bin/`):
  - `dbproxystart` - uruchomienie proxy, jeli co pójdzie nie tak zostanie wywietlone na ekranie
  - `dbproxystartverbose` - uruchomienie proxy w trybie verbose, mnóstwo komunikatów na ekran, przydatne przy problemach
  - `dbproxystop` - kulturalne zatrzymanie proxy poprzez wysanie sygnału TERM
  - `dbproxystatus` - robi zwykly "ps v -p pidproxy"
  - proxy mona odpali rcznie komend `/usr/local/nagios/avantis/sbin/dbproxy.pl -c /usr/local/nagios/avantis/etc/dbproxy.cfg` :
    - opcja `-c` specyfikuje uity plik konfiguracyjny
    - opcja `-v` dziaa jak komenda `dbproxystartverbose`
- pliki zwizane z proxy :
  - biblioteka PERL szyfrujco-deszyfrujca `/usr/local/nagios/avantis/lib/cryptmodule.pm`, wane funkcje dla checków :
    - `$cipher = init_cipher($klucz)`, inicjuje mechanizm szyfrujacy z podanego klucza, który musi si zgadza z kluczem uzywanym przez proxy, jeli chcemy t funkcj wykorzysta w checku
    - `$cipher = initfile_cipher($filename)`, jak wyzej ale inicjuje mechanizm szyfrujcy z pliku, zalecam uywanie tej funkcji, po za tym uwagi jak wyzej
  - biblioteka PERL dla checków `/usr/local/nagios/avantis/lib/dbproxy.pm`, wane funkcje dla checków
    - `$gniazdo = dbproxy_connect($proxyhost,$port)`, pobiera nazwe/ip i numer portu jako argumenty i zwraca gniazdo komunikacyjne z `$proxyhost` lub 0 w przypadku bdu, oczywicie jesli nie podamy tych argumentów zostanie uity domylnie localhost i port 27005
    - `dbproxy_disconnect($gniazdo)`, rozacz z `$proxyhost`
    - `$wynik {{ dbproxy_query($cipher,$gniazdo,$dbhost,$query,@arg)`, jesli komunikacja ma nie by szyfrowana to argument `$cipher` musi mie warto undef, jeli szyfrowana musimy go zainicjowa funkcj `init_cipher()` lub `initfile_cipher()`, `$gniazdo` to wynik funkcji `dbproxy_connect()`, `$dbhost` to nazwa sekcji hosta bazy danych z pliku konfiguracyjnego z ktorym proxy ma nawizane poczenie, `$query` to zapytanie sql, `@arg` to lista argumentów dla `$query` jeli uywamy podstawie co przyspiesza wykonywanie zapyta, poniewa s juz one wziesniej przygotowane np. `$wynik {{ dbproxy_query(undef,$gniazdo,"sigma","SELECT ? + ? FROM DUAL",7,11)`. Zazwyczaj wiemy czego oczekujemy w wyniku, jeli wystpi bd zwracany jest komunikat "WARNING\t\tTRESCE BDU" (sowo WARNING, potem dwa znaki tabulacji i tre bdu). Jeli select nie zwróci wierszy zwracany jest tekst "0E0", ten tekst jest rownie zwracany gdy update/insert nie zmodyfikuje adnych wierszy, update/insert zwraca liczb zmodyfikowanych/nowych wierszy. Przykladem jest check `/usr/local/nagios/avantis/bin/check_dbjobs`
  - skrypt szyfrujacy/deszyfrujacy hasa algorytmem HMAC `/usr/local/nagios/avantis/bin/passcrypt.pl` :
    - zaszyfrowanie hasa `./passcrypt.pl -k "tajny klucz" -t "secret password"`
    - odszyfrowanie hasa `./passcrypt.pl -d -k "tajny klucz" -t "zakodowane secret password"`
  - **plik z przykladami** `/usr/local/nagios/avantis/bin/dbproxy_example.pl`
- kade poczenie z baz danych to jeden wtek procesu, wic jeli wszystkie bd zajte to nastpny check dostanie komunikat e nie ma wolnych pocze. Z tego wynika e prawidlowo moe zosta obsuonych naraz tyle checków ile mamy nawizanych pocze z bazami danych. Wic jeli mamy dwa stae poczenia z sigm i przyjd 3 zapytania do dbproxy w tym samym czasie, to dwa zostan obsuone a trzeci dostanie komunikat WARNING o braku wolnych pocze.

## Moliwe zauwaone problemy

- Utrata poczenia z baz - nastpi automatycznie ponowne poczenia po upywie czasu okrelonego parametrem "timeout", jeli s jeszcze wolne poczenia do bazy danych to inne checki zostan przez nie obsuone, jeli brak wolnych pocze to checki dostan odpowiedni komunikat/warning o ich braku, te checki które trafiy na martwe poczenie wisz do momentu ponownego poczenia z baz i po poczeniu zostan obsuone.
- Bład otwarcia portu serwer: Adres jest ju w uyciu - jeli zdarzy si przy uruchamianiu oznacza ze kto chca podwójnie uruchomi proxy, nalezy

- za pomocy komenda netstat wykry proces i wysła do niego sygna TERM, wtedy adres się zwolni.
- Zmienna \$ORACLE\_HOME jest ustawiana wg pliku konfiguracyjnego globalnie dla wszystkich pocze a nie wg zmiennej z powłoki. Parametr host dla serwerów oracle powinien odpowiada wpisowi z tnsnames.ora
- Generalnie trzeba zajrzeć w plik logu tam jest wszystko co się dzieje z demonem.

## Inne uwagi

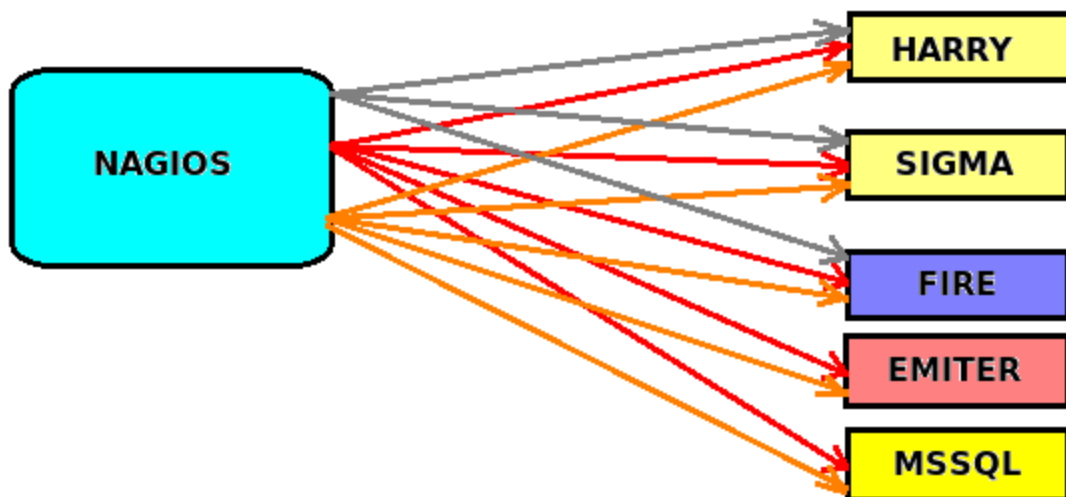
- W przypadku pocze lokalnych do dbproxy tzn. do localhosta używanie poczenia szyfrowanego jest niepotrzebne, stosujemy je tylko do pocze nawiązywanych z innych maszyn (innych niż ta na którym nasłuchuje demon dbproxy);

## Selekty wieloliniowe i wielokolumnowe

- Kolumny rozdzielane są znakami tabulacji "\t" (Zwróć uwagę! mechanizm wczytania usuwa z wartości kolumny tabulatory jeśli występują i zamienia na spacje),
- Linie rozdzielane są znakami nowej linii "\n",

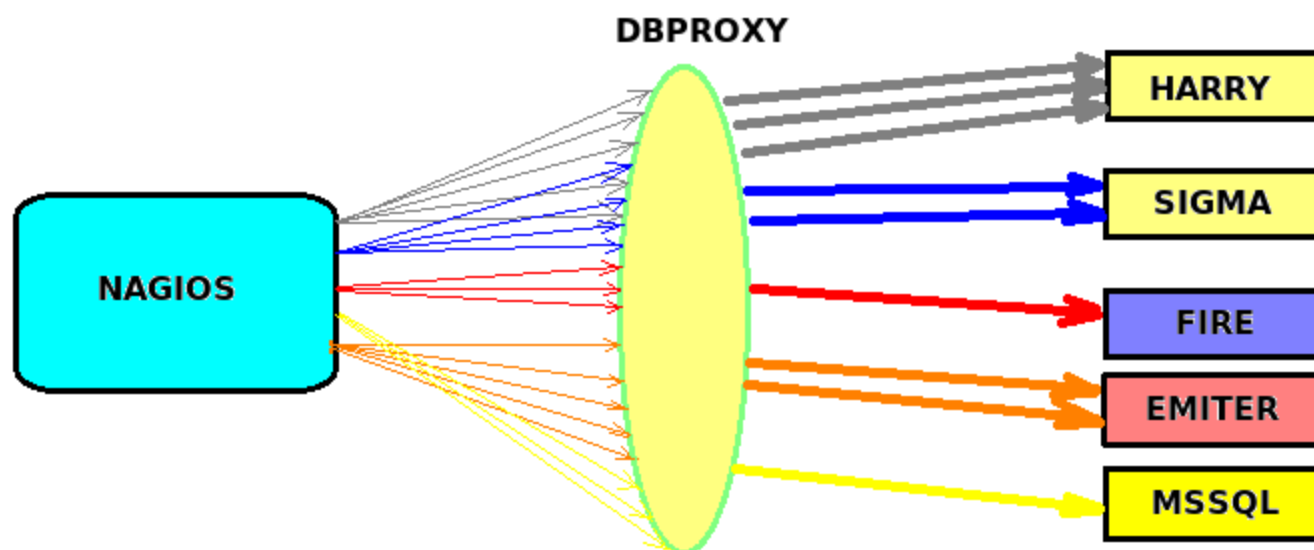
## RYSUNEK PRZEDSTAWIA ZALETY DBPROXY:

## BEZ DBPROXY: WIELE BEZPOŚREDNICH, CHWILOWYCH POŁĄCZEŃ Z BAZA



- z punktu widzenia bazy, wiele chwilowych połączeń z nagiosa, większa liczba procesów na ich obsługę
- dłuższe i bardziej skomplikowane checki

## SYTUACJA PO ZASTOSOWANIU MECHANIZMU DBPROXY



- kilka stałych połączeń pomiędzy dbproxy a bazami danych, wiele chwilowych połączeń z nagiosa do dbproxy, => obciążenie baz danych

- w checkach uniwersalne funkcje niezależne od typu bazy danych  
`dbproxy_connect()`, `dbproxy_query()` => krótsze checki
- mechanizm cache => szybsze checki

w razie Awarii

Jeeli pojawio si wiele checków polecam zajrze na serwer nagios do `/usr/local/nagios/avantis/SVN/DNIU/dbproxy`

-- Main.WitoldOlszewski - 26 Apr 2007