

Utwardzanie ochrony systemu operacyjnego serwerowych środowisk Linuksowych: RSBAC

1. Wprowadzenie

Utwardzanie ochrony systemu operacyjnego jest szczególnie ważne dla serwerów pracujących bezustannie oraz dla serwerów o zakładanym wysokim poziomie bezpieczeństwa przechowującym tajne dane. System taki powinien być odporny na większość (jeśli nie wszystkie) możliwe próby ataku, włamania. Oczywiście jest to bardzo trudne do osiągnięcia, jeśli w ogóle jest to możliwe. Już w latach 80-tych zaczęto się zastanawiać nad zwiększeniem bezpieczeństwa systemów serwerowych, głównym organizatorem tych rozszerzeń były organizacje wojskowe i rządowe, które musiały dobrze chronić informacje. Zaproponowano wówczas nowy model bezpieczeństwa – **Obowiązkową Kontrolę Dostępu (MAC)**, zamiast aktualnie wykorzystywanej **Uznaniowej Kontroli Dostępu (DAC)**.

Rozwój systemów wykorzystujących politykę MAC jest dosyć powolny, głównie ze względu na trudności administracyjne tych systemów. Jednak jako ich bardzo duży plus jest odporność na większość ataków, włamań – warunkiem jednakże jest poprawne skonfigurowanie polityki, co jest zadaniem bardzo trudnym.

Niniejsze ćwiczenie ma zaznajomić z utwardzaniem ochrony na podstawie systemu wykorzystującego politykę MAC o nazwie RSBAC (Rule-set Based Access Control).

2. Informacje o systemie RSBAC

System RSBAC został stworzony przez Amon Ott w 1996-97 jako praca magisterska. RSBAC jest cały czas rozwijany przez jego twórcę oraz grono jego współpracowników. Projekt ten jest całkowicie otwarty, jednak jego rozwój nie jest bardzo szybki ze względu na wysoki poziom bezpieczeństwa, który ma zapewniać.

RSBAC to zestaw łat na jądro systemu Linux rozszerzających bezpieczeństwo systemu. Wspiera on mechanizmy takie jak:

- ścisła kontrola dostępu (MAC – Mandatory Access Control),
- rozszerzone prawa do plików,
- listy kontroli dostępu (ACL – Access Control Lists),
- role,
- typy,
- piaskownice (rozszerzone polecenie chroot),
- wiele innych.

Dzięki modułowej budowie, RSBAC jest bardzo elastyczny. Każdy moduł daje pewne możliwości zapewniania bezpieczeństwa, ale często dany problem można rozwiązać na kilka sposobów przy użyciu różnych modułów. Wybór odpowiedniego modułu/rozwiązania należy do administratora.

3. Utwarczanie ochrony systemu na podstawie zabezpieczenia serwera Apache przy wykorzystaniu RSBAC

Instalacja systemu RSBAC oraz podstawowe polecenia

Na zainstalowany system Linux powinno się skompilować nowe jądro systemu wykorzystujące system RSBAC oraz zainstalować (wpierw skompilować) narzędzia niezbędne do konfiguracji systemu RSBAC.

Istnieją dwa rodzaje narzędzi, polecenia w trybie tekstowym oraz polecenia z interfejsem użytkownika. Największym narzędziem jest `rsbac_menu`, który umożliwia dostęp do wszystkich możliwych ustawień wszystkich modułów, jest to narzędzie z interfejsem użytkownika.

Każde inne narzędzie wprowadza użytkownika bezpośrednio do dokładniejszych informacji/ustawień.

1. Pierwsze uruchomienie

Pierwsze uruchomienie spowoduje utworzenie domyślnych reguł, które pozwalają na uruchomienie systemu, jednak nie pozwalają na zalogowanie się. Jest to spowodowane zablokowaniem możliwości zmiany użytkownika przez wszystkie programy (między innymi przez `login`), można to ominąć dodając przy starcie opcję `rsbac_auth_enable_login`. Jednak zaraz po zalogowaniu się należy stworzyć użytkownika *security officer*: `secoff` (`uid=400`), który będzie ukrytym zarządcą systemu i do jego zadań należy konfiguracja zabezpieczeń systemu w wykorzystaniem rozszerzeń RSBAC.

Ustawienie uprawnień do zmiany uprawnień dla programu `/bin/login`:

```
attr set fd FILE auth may setuid 1 /bin/login
```

2. Przygotowanie systemu do bezpiecznego używania serwera apache2

Na początku należy ustalić w jaki sposób będziemy zabezpieczać serwer. Po pierwsze należy określić, które katalogi będą podlegały restrykcją:

- konfiguracja serwera apache2 – `/etc/apache2/*`
- katalog z logami – `/var/log/apache2/*`
- plik blokady – `/var/run/apache2.pid`
- katalog ze stronami www – `/srv/www/htdocs`, `/usr/share/apache2/error`, `/usr/share/apache2/icons`
- katalog ze skryptami – `/srv/www/cgi-bin`
- katalog plików tymczasowych – `/tmp`
- katalog z globalną konfiguracją – `/etc`

Rozdzielenie katalogów ze skryptami i stronami www pozwala nam zwiększyć bezpieczeństwo.

1. Dodanie możliwości zmiany uprawnień (użytkownika) na uid=30 (wwwrun) oraz gid=9 (www) przez aplikację /usr/sbin/apache2-prefork

```
auth_set_cap FILE add /usr/sbin/httpd2-prefork 30
auth_set_cap -g FILE add /usr/sbin/httpd2-prefork 8
```

2. Ustawienie typów RC (Role Compatibility)

Stworzenie typów

Korzystając z aplikacji rsbac_rc_type_menu tworzymy typy FD. Należy wybrać "New Type", zaakceptować numer, a następnie zmienić nazwę na:

- WWW_Config
- WWW_Logfile
- WWW_LockFiles
- WWW_Files
- WWW_Cgi-bin
- Configfiles
- Tempfiles

Następnie należy skopiować uprawnienia z "General FD" do wszystkich powyższych nowo utworzonych typów, poprzez "Copy Rights to Type". Należy jeszcze stworzyć typ NETOBJ o nazwie HTTP_NETOBJ (oraz UNIX_NETOBJ) i skopiować uprawnienia z "General_NETOBJ"

Przypisanie typów do zasobów

Wykorzystać program rsbac_fd_menu do przyporządkowania typów do odpowiednich katalogów:

- typ Configfiles do katalogu /etc
- typ Tempfiles do katalogu /tmp
- typ WWW_Config do katalogu /etc/apache2
- typ WWW_Logfile do katalogu /var/log/apache2
- typ WWW_LockFiles do katalogu /var/run/apache2 (należy go wcześniej utworzyć)

Nie można przypisać typu do pliku (PidFile), gdyż plik ten jest usuwany przy wyłączeniu serwera, a uprawnienia są przypisywane do konkretnych i-węzłów, a nie do nazw.

Dodatkowo trzeba zmodyfikować konfigurację serwera, aby wykorzystywał ten katalog, w pliku /etc/apache2/httpd.conf należy dopisać linię: PidFile /var/run/apache2/httpd2.pid oraz w pliku /etc/init.d/apache2 zmodyfikować linię pidfile=... tak aby odpowiadała temu samemu plikowi.

- typ WWW_Files do katalogu /srv/www/htdocs
- typ WWW_Cgi-bin do katalogu /srv/www/cgi-bin

Przed restartowaniem systemu należy sprawdzić, czy wszystkie pliki i katalogi, które miały zmieniane uprawnienia, posiadają prawa dostępu dla użytkownika root. Dopiero tak ustawiony system można restartować.

3. Ustawienie ról RC

Stworzenie ról

Wykorzystać program `rsbac_rc_role_menu`, aby utworzyć dwie nowe role, tworzymy je poprzez kopiowanie z `General User`, a następnie należy zmienić nazwy na:

- `WWW_Server` – potrzebny do startu serwera
- `WWW_User` – potrzebny do dalszego działania serwera

Przypisanie ról

- `WWW_Server` należy przyporządkować jako RC Initial Role do pliku `/usr/sbin/httpd2-prefork` programem `rsbac_fd_menu`
- `WWW_User` należy przyporządkować jako RC Default Role użytkownikowi `wwwrun` (30) przy użyciu programu `rsbac_user_menu`

To przyporządkowanie ról zapewni samoczynną zmianę roli wykorzystywanej przez serwer, gdyż na początku uruchamia się z prawami użytkownika `root` oraz rolą `WWW_Server`, następnie zmienia uprawnienia na uprawnienia użytkownika `wwwrun`, któremu została przyporządkowana rola `WWW_User` i tym samym rola zostanie zmieniona.

4. Utworzenie szablonu gniazdka

Wykorzystać program `rsbac_nettemp_def_menu`. Stworzyć nowy szablon gniazdka [New Template] (z numerem mniejszym niż domyślne szablony, gdyż `rsbac` sprawdza od najniższego numeru, aż do pierwszego znalezionej). Szablon powinien być skonfigurowany następująco:

```
Template: 20000
Name: HTTP
Address Family: INET
Socket Type: STREAM
Address: 0.0.0.0 (można podać adres IP konkretnego
interfejsu)
Valid Length: 32
Protocol: TCP
Network Device:
Min Port: 80
Max Port: 80
```

wybrać NetTemp Attributes:

```
Template number: 20000/HTTP
RC Type: HTTP_NETOBJ (jeden RC który jeszcze nie był
przyporządkowany)
RC Type NT: 0/General NETTEMP
```

Stworzyć drugi szablon:

```
Template: 10000
Name: UNIX
Address Family: UNIX
Socket Type: STREAM
Address:
Valid Length:
Protocol:
```

```
Network Device:
Min Port:
Max Port:
```

wybrać NetTemp Attributes:

```
Template number: 10000/UNIX
RC Type: UNIX_NETOBJ (jedyne RC który jeszcze nie był
przyporządkowany)
RC Type NT: 0/General NETTEMP
```

5. Ustawienie uprawnień

Wykorzystać program `rsbac_rc_role_menu` do ustawienia uprawnień dwóm stworzonym rodom:

dla roli `WWW_Server`:

```
Type Comp FD → General_FD → R + Exec + Map_exec
Type Comp FD → WWW_Config → R
Type Comp FD → WWW_Logfiles → R + Append_open + Write (opcjonalnie)
Type Comp FD → WWW_LockFiles → R + Create + Delete + Write_Open + Write +
Truncate
Type Comp FD → WWW_Files → R
Type Comp FD → WWW_Cgi-bin → R
Type Comp FD → Tempfiles → R + W + Create
Type Comp FD → Configfiles → R
Type Comp NETOBJ → General_NETOBJ → Create + Modify_system_data
Type Comp NETOBJ → HTTP_NETOBJ → Bind + Listen + Close
Type Comp NETOBJ → UNIX_NETOBJ → Create + Close + Connect
```

dla roli `WWW_User`:

```
Type Comp FD → General_FD → R + Execute + Map_exec
Type Comp FD → WWW_Config → R
Type Comp FD → WWW_Logfiles → R + Append_open
Type Comp FD → WWW_LockFiles → R
Type Comp FD → WWW_Files → R
Type Comp FD → WWW_Cgi-bin → R + Execute
Type Comp FD → Tempfiles → R + W + Create
Type Comp FD → Configfiles → Search (Opcjonalnie)
Type Comp NETOBJ → General_NETOBJ → Accept + Send + Receive +
Net_shutdown + Write + Read + Get_status_data
Type Comp NETOBJ → HTTP_NETOBJ → -
Type Comp NETOBJ → UNIX_NETOBJ → -
```

3. Uruchomienie serwera apache2

Po uprzednim skonfigurowaniu serwera oraz przypisaniu mu uprawnień należy uruchomić serwer poprzez skrypt `/etc/rc.d/apache2 start`.

Dla celów testowych można w trakcie ustawiania próbować uruchamiać serwer `apache2` jednocześnie obserwując plik `/var/log/messages`, gdzie są umieszczane wszystkie komunikaty o błędach.

4. Zadania

- Uruchomić system Linux wykorzystując nowe jądro (RSBAC), w trybie `rsbac_auth_enable_login` – wybrać przy starcie komputera z menu bootloadera.
- Uruchomić serwer `apache2` i sprawdzić czy poprawnie działa, jeśli nie należy sprawdzić dlaczego (komunikaty błędów systemu) i umożliwić zmianę uprawnień.
- Napisać prosty skrypt `php`, który odczytuje plik `/etc/passwd` i przesyła poprzez `www`.
- Sprawdzić działanie oraz sprawdzić komunikaty błędów systemu (konsola 10'ta).
- Zalogować się jako użytkownik `secoff` i uruchomić skrypt `“apache_rsbac.sh”`.
- Powtórnie sprawdzić poprawność działania skryptu oraz komunikaty błędów.
- Napisać `“rc_set_item ROLE 5 type_comp_fd 9 R”`, a następnie sprawdzić działanie skryptu i komunikaty błędów.
- Następnie należy zapoznać się ze skryptem `“apache_rsbac.sh”`.

Następnie należy stworzyć katalog `/srv/www/htdocs/tajne/` oraz plik `index.html` w tym katalogu i zablokować dostęp do tego katalogu z poziomu zabezpieczeń RSBAC, korzystając z już utworzonej polityki, jedynie dopisując lub modyfikując ją. Rozwiązanie należy przedstawić prowadzącemu zajęcia.

5. Problemy do dyskusji

- Wady, zalety systemu RSBAC.
- Polityka MAC, a polityka DAC.
- Czy potrzebne są systemy wspierające politykę MAC?
- Co rozumiemy przez błąd dnia zerowego?

6. Bibliografia

[WWW] Projekt RSBAC – <http://www.rsbac.org>

7. Dodatek A

Wszystkie polecenia można wykonywać bez pośrednictwa menu, poniżej zostaną zaprezentowane wszystkie polecenia wykorzystywane podczas ćwiczenia.

Umożliwienie zmiany uprawnień serwerowi Apache z praw administratora `“root”` (oraz grupy `root`) na użytkownika `wwwrun(30)` i grupę `www(8)`:

```
(zmiana grupy) auth_set_cap -g FILE add /usr/sbin/httpd2-
```

```
prefork 8  
(zmiana użytkownika)auth_set_cap FILE add /usr/sbin/httpd2-  
prefork 30
```

Utworzenie nowych (o nowych numerach) typów poprzez skopiowanie domyślnego typu (General FD - 0 lub General NETOBJ - 0), wykorzystujemy dwa rodzaje typów FD i NETOBJ:

```
rc_copy_type FD 0 4  
rc_copy_type FD 0 5  
rc_copy_type NETOBJ 0 4
```

Zmiana nazwy nowo utworzonym typom:

```
rc_set_item TYPE 4 type_fd_name WWW_Config  
rc_set_item TYPE 4 type_netobj_name HTTP_NETOBJ
```

Przypisanie plików i katalogów do konkretnych typów:

```
(Katalog) attr_set_file_dir RC DIR /etc/apache2 rc_type_fd  
4  
(Plik) attr_set_file_dir RC FILE /etc/localtime rc_type_fd  
4
```

Utworzenie nowych ról poprzez skopiowanie ich z domyślnej roli (General User):

```
rc_copy_role 0 4  
rc_copy_role 0 5
```

Zmiana nazwy nowo utworzonym rolam:

```
rc_set_item ROLE 4 name WWW_Server  
rc_set_item ROLE 5 name WWW_User
```

Przypisanie użytkownikowi wwwrun domyślnej roli WWW_User (5):

```
attr_set_user RC wwwrun rc_def_role 5
```

Przypisanie programowi (plikowi) roli WWW_Server (4), jaką otrzyma w przypadku uruchomienia:

```
attr_set_file_dir RC FILE /usr/sbin/httpd2-prefork  
rc_initial_role 4
```

Utworzenie nowego szablonu gniazdka o numerze 20000 i nazwie HTTP:

```
net_temp new_template 20000 HTTP
```

Przypisanie typu adresu na Internetowy:

```
net_temp set_address_family 20000 INET
```

Ustalenie wielkości adresu na 32 bity:

```
net_temp set_valid_len 20000 32
```

Ustalenie typu gniazdka na strumieniowe:

```
net_temp set_type 20000 STREAM
```

Ustalenie protokołu na TCP:

```
net_temp set_protocol 20000 TCP
```

Ustalenie minimalnego portu, który będzie miał prawo wykorzystać:

```
net_temp set_min_port 20000 80
```

Ustalenie maksymalnego portu, który będzie miał prawo wykorzystać:

```
net_temp set_max_port 20000 80
```

Przypisanie szablonu gniazdka do typu NETOBJ o nazwie HTTP_NETOBJ (4):

```
attr_set_net RC NETTEMP rc_type 4 20000
```

Ustalanie uprawnień dla poszczególnych typów i ról:

ustalenie dla roli WWW_Server (4), typu General FD (0) praw odczytu, wykonywanie:

```
rc_set_item ROLE 4 type_comp_fd 0 R EXECUTE MAP_EXEC
```

ustalenie dla typu WWW_Config (4) przy korzystaniu z roli WWW_Server (4) praw odczytu:

```
rc_set_item ROLE 4 type_comp_fd 4 R
```

ustalenie dla typu General NETOBJ (0) przy korzystaniu z roli WWW_Server (4) praw tworzenia i modyfikacji danych systemowych:

```
rc_set_item ROLE 4 type_comp_netobj 0 CREATE  
MODIFY_SYSTEM_DATA
```

8. Dodatek B

Skrypt ustanawiający politykę bezpieczeństwa określoną w powyższym ćwiczeniu.

rsbac_apache2:

```
export LANG=C  
auth_set_cap -g FILE add /usr/sbin/httpd2-prefork 8  
auth_set_cap FILE add /usr/sbin/httpd2-prefork 30  
  
rc_copy_type FD 0 4  
rc_copy_type FD 0 5  
rc_copy_type FD 0 6
```



```

rc_copy_type FD 0 7
rc_copy_type FD 0 8
rc_copy_type FD 0 9
rc_copy_type FD 0 10
rc_copy_type NETOBJ 0 4
rc_set_item TYPE 4 type_fd_name WWW_Config
rc_set_item TYPE 5 type_fd_name WWW_Logfiles
rc_set_item TYPE 6 type_fd_name WWW_Lockfiles
rc_set_item TYPE 7 type_fd_name WWW_Files
rc_set_item TYPE 8 type_fd_name WWW_Cgi-bin
rc_set_item TYPE 9 type_fd_name ConfigFiles
rc_set_item TYPE 10 type_fd_name TempFiles
rc_set_item TYPE 4 type_netobj_name HTTP_NETOBJ
rc_set_item TYPE 5 type_netobj_name UNIX_NETOBJ

attr_set_file_dir RC DIR /etc/apache2 rc_type_fd 4
attr_set_file_dir RC FILE /etc/localtime rc_type_fd 4
attr_set_file_dir RC DIR /var/log/apache2 rc_type_fd 5
attr_set_file_dir RC DIR /var/run/apache2 rc_type_fd 6
attr_set_file_dir RC DIR /srv/www/htdocs rc_type_fd 7
attr_set_file_dir RC DIR /srv/www/cgi-bin rc_type_fd 8
attr_set_file_dir RC DIR /etc rc_type_fd 9
attr_set_file_dir RC DIR /tmp rc_type_fd 10

rc_copy_role 0 4
rc_copy_role 0 5
rc_set_item ROLE 4 name WWW_Server
rc_set_item ROLE 5 name WWW_User
attr_set_user RC wwwrun rc_def_role 5
attr_set_file_dir RC FILE /usr/sbin/httpd2-prefork
rc_initial_role 4

net_temp new_template 20000 HTTP
net_temp set_address_family 20000 INET
net_temp set_valid_len 20000 32
net_temp set_type 20000 STREAM
net_temp set_protocol 20000 TCP
net_temp set_min_port 20000 80
net_temp set_max_port 20000 80
attr_set_net RC NETTEMP rc_type 4 20000

net_temp new_template 10000 UNIX
net_temp set_address_family 10000 UNIX
net_temp set_type 10000 STREAM
attr_set_net RC NETTEMP rc_type 5 10000

rc_set_item ROLE 4 type_comp_fd 0 R EXECUTE MAP_EXEC
rc_set_item ROLE 4 type_comp_fd 4 R
rc_set_item ROLE 4 type_comp_fd 5 R APPEND_OPEN WRITE
rc_set_item ROLE 4 type_comp_fd 6 R CREATE DELETE \
WRITE OPEN WRITE TRUNCATE

```

```

rc_set_item ROLE 4 type_comp_fd 7 R
rc_set_item ROLE 4 type_comp_fd 8 R
rc_set_item ROLE 4 type_comp_fd 9 R
rc_set_item ROLE 4 type_comp_fd 10 R W CREATE

rc_set_item ROLE 4 type_comp_netobj 0 CREATE \
  MODIFY_SYSTEM_DATA
rc_set_item ROLE 4 type_comp_netobj 4 BIND LISTEN CLOSE
rc_set_item ROLE 4 type_comp_netobj 5 CREATE CLOSE CONNECT

rc_set_item ROLE 5 type_comp_fd 0 R EXECUTE MAP_EXEC
rc_set_item ROLE 5 type_comp_fd 4 R
rc_set_item ROLE 5 type_comp_fd 5 R APPEND_OPEN
rc_set_item ROLE 5 type_comp_fd 6 R
rc_set_item ROLE 5 type_comp_fd 7 R
rc_set_item ROLE 5 type_comp_fd 8 R EXECUTE
rc_set_item ROLE 5 type_comp_fd 9 SEARCH
rc_set_item ROLE 5 type_comp_fd 10 R W CREATE

rc_set_item ROLE 5 type_comp_netobj 0 ACCEPT SEND RECEIVE \
  NET_SHUTDOWN READ WRITE GET_STATUS_DATA
rc_set_item ROLE 5 type_comp_netobj 4

```

rsbac_apache2_full:

```

export LANG=C
auth_set_cap -g FILE add /usr/sbin/httpd2-prefork 8
auth_set_cap FILE add /usr/sbin/httpd2-prefork 30

rc_copy_type FD 0 4
rc_copy_type FD 0 5
rc_copy_type FD 0 6
rc_copy_type FD 0 7
rc_copy_type FD 0 8
rc_copy_type FD 0 9
rc_copy_type FD 0 10
rc_copy_type FD 0 11
rc_copy_type NETOBJ 0 4
rc_set_item TYPE 4 type_fd_name WWW_Config
rc_set_item TYPE 5 type_fd_name WWW_Logfiles
rc_set_item TYPE 6 type_fd_name WWW_Lockfiles
rc_set_item TYPE 7 type_fd_name WWW_Files
rc_set_item TYPE 8 type_fd_name WWW_Cgi-bin
rc_set_item TYPE 9 type_fd_name ConfigFiles
rc_set_item TYPE 10 type_fd_name TempFiles
rc_set_item TYPE 11 type_fd_name Library
rc_set_item TYPE 4 type_netobj_name HTTP_NETOBJ
rc_set_item TYPE 5 type_netobj_name UNIX_NETOBJ

attr_set_file_dir RC DIR /etc/apache2 rc_type_fd 4
attr_set_file_dir RC FILE /etc/localtime rc_type_fd 4

```

```

attr_set_file_dir RC DIR /var/log/apache2 rc_type_fd 5
attr_set_file_dir RC DIR /var/run/apache2 rc_type_fd 6
attr_set_file_dir RC DIR /srv/www/htdocs rc_type_fd 7
attr_set_file_dir RC DIR /srv/www/cgi-bin rc_type_fd 8
attr_set_file_dir RC DIR /etc rc_type_fd 9
attr_set_file_dir RC DIR /tmp rc_type_fd 10
attr_set_file_dir RC DIR /lib rc_type_fd 11
#attr_set_file_dir RC DIR /usr/lib rc_type_fd 11
#
# biblioteki potrzebne dla servera apache
attr_set_file_dir RC DIR /usr/lib/apache2 rc_type_fd 11
attr_set_file_dir RC DIR /usr/lib/php/extensions \
rc_type_fd 11
attr_set_file_dir RC FILE /usr/lib/libaprutil-0.so.0.9.6 \
rc_type_fd 11
attr_set_file_dir RC FILE /usr/lib/libgdbm.so.3.0.0 \
rc_type_fd 11
attr_set_file_dir RC FILE /usr/lib/libdb-4.3.so \
rc_type_fd 11
attr_set_file_dir RC FILE /usr/lib/libexpat.so.0.5.0 \
rc_type_fd 11
attr_set_file_dir RC FILE /usr/lib/libapr-0.so.0.9.6 \
rc_type_fd 11
attr_set_file_dir RC FILE \
/usr/lib/apache2-prefork/mod_cgi.so rc_type_fd 11
attr_set_file_dir RC FILE \
/usr/lib/apache2-prefork/mod_ssl.so rc_type_fd 11
attr_set_file_dir RC FILE /usr/lib/libssl.so.0.9.7 \
rc_type_fd 11
attr_set_file_dir RC FILE /usr/lib/libcrypto.so.0.9.7 \
rc_type_fd 11
attr_set_file_dir RC FILE \
/usr/lib/apache2-prefork/libphp4.so rc_type_fd 11

rc_copy_role 0 4
rc_copy_role 0 5
rc_set_item ROLE 4 name WWW_Server
rc_set_item ROLE 5 name WWW_User
attr_set_user RC wwwrun rc_def_role 5
attr_set_file_dir RC FILE \
/usr/sbin/httpd2-prefork rc_initial_role 4

net_temp new_template 20000 HTTP
net_temp set_address_family 20000 INET
net_temp set_valid_len 20000 32
net_temp set_type 20000 STREAM
net_temp set_protocol 20000 TCP
net_temp set_min_port 20000 80
net_temp set_max_port 20000 80
attr_set_net RC NETTEMP rc_type 4 20000

```

```

net_temp new_template 10000 UNIX
net_temp set_address_family 10000 UNIX
net_temp set_type 10000 STREAM
attr_set_net RC NETTEMP rc_type 5 10000

rc_set_item ROLE 4 type_comp_fd 0 SEARCH GET_STATUS_DATA
rc_set_item ROLE 4 type_comp_fd 4 R
rc_set_item ROLE 4 type_comp_fd 5 R APPEND_OPEN WRITE
rc_set_item ROLE 4 type_comp_fd 6 R CREATE DELETE \
WRITE_OPEN WRITE TRUNCATE
rc_set_item ROLE 4 type_comp_fd 7 R
rc_set_item ROLE 4 type_comp_fd 8 R
rc_set_item ROLE 4 type_comp_fd 9 R
rc_set_item ROLE 4 type_comp_fd 10 R W CREATE
rc_set_item ROLE 4 type_comp_fd 11 R MAP_EXEC

rc_set_item ROLE 4 type_comp_netobj 0 CREATE \
MODIFY_SYSTEM_DATA
rc_set_item ROLE 4 type_comp_netobj 4 BIND LISTEN CLOSE
rc_set_item ROLE 4 type_comp_netobj 5 CREATE CLOSE CONNECT

rc_set_item ROLE 5 type_comp_fd 0 SEARCH GET_STATUS_DATA
rc_set_item ROLE 5 type_comp_fd 4
rc_set_item ROLE 5 type_comp_fd 5 APPEND_OPEN WRITE \
CHDIR CLOSE GET_PERMISSIONS_DATA GET_STATUS_DATA SEARCH
rc_set_item ROLE 5 type_comp_fd 6
rc_set_item ROLE 5 type_comp_fd 7 R
rc_set_item ROLE 5 type_comp_fd 8 R EXECUTE
rc_set_item ROLE 5 type_comp_fd 9 SEARCH
rc_set_item ROLE 5 type_comp_fd 10 R W CREATE
rc_set_item ROLE 5 type_comp_fd 11

rc_set_item ROLE 5 type_comp_netobj 0 ACCEPT SEND \
RECEIVE NET_SHUTDOWN READ WRITE GET_STATUS_DATA
rc_set_item ROLE 5 type_comp_netobj 4

```