Testowanie serwera Radius

Tomasz Wolniewicz UCI UMK (twoln@umk.pl)
dokument przygotowany w ramach projektu B-R eduroam-PIONIER
Wersja 2.3-a

Spis treści

Wstęp

Uruchomienie serwera Radius jest zadaniem stosunkowo skomplikowanym. Niezbędne jest sprawdzenie wielu sytuacji, w co zaangażowane jest wiele składowych – oprogramowanie karty bezprzewodowej, oprogramowanie klienta 802.1x na komputerze przenośnym, access-point, przełącznik z obsługą VLAN-ów itd. W sytuacji, kiedy coś nie działa niezbędne są narzędzia testowania, które pozwalają na zmniejszenie liczby składników. Dobrze jest również dysponować narzędziami, które można uruchamiać automatycznie i w ten sposób monitorować poprawność działania serwera lub serwerw.

Poniżej opisujemy dwa programy: **radtest** i **eapol_test**. **radtest** jest elementem oprogramowania FreeRadius i jest instalowany razem z serwerem, nadaje się do prowadzenia bardzo prostych testów i dlatego może być przydatny tylko we wczesnej fazie instalowania serwera. **eapol_test** wymaga skompilowania pakietu wpa_supplicant, ale za to jest narzędziem pozwalającym na przetestowanie właściwie wszystkich aspektów komunikacji z serwerem Radius, z tego powodu uważamy go za niezbędne wyposażenie administratora eduroam.

Komputer na którym działa program testujący spełnia rolę access-pointa i podłączonego do niego laptopa, zatem jego adres musi być zarejestrowany jako klient testowanego serwera Radius (w przypadku serwera FreeRadius adres komputera testującego musi być wpisany do pliku `clients.conf`). Niezbędne jest przy tym przydzielenie odpowiedniego klucza (sekret) pozwalającego na wymianę informacji między klientem i serwerem.

radtest

Pakiet FreeRADIUS zawiera dwa programy klienckie: **radclient** i **radeapclient** oraz skrypt **radtest**. Ten ostatni jest najwygodniejszy do wczesnych testów.

W odpowiedzi na próbę uwierzytelnienia wysłaną przez radtest otrzymujemy odpowiedź serwera w formie Access-Accept lub Access-Reject, widzimy wszystkie przesłane atrybuty, na przykład ustawienia VLAN-u albo wiadomość przesłaną w ramach pakietu Access-Reject. Zasadniczą wadą radtest jest to, że korzystamy ze zwykłego hasła, które przesyłamy przez sieć zaszyfrowane w dość prymitywny sposób. Nie możemy też przetestować uwierzytelniania stosującego EAP, a więc takiego jakie stosuje się w ***eduroam***.

radtest wywołujemy:

```
radtest uzytkownik@realm haslo_uzytkownika adres_serwera 0 sekret
```

Gdyby Radius pracował na innym porcie niż 1812, to trzeba by jeszcze i to uwzględnić w postaci: `adres_serwera:port`.

Jeżeli wszystko jest skonfigurowane prawidłowo to powinniśmy otrzymać Access-Accept. Jeżeli nie będzie żadnych odpowiedzi, to albo nie działa Radius, albo coś blokuje dostęp do portu.

eapol_test

eapol_test jest programem testowym zawartym w dystrybucji **wpa_supplicant**. Pozwala na przetestowanie serwera Radius z zastosowaniem wielu metod EAP i dlatego jest szczególnie przydatny jako narzędzie w ***eduroam***. Korzystając z **eapol_test** możemy zobaczyć całą wymianę informacji między klientem i serwerem. Taki program jest bardzo wygodny w czasie instalowania nowej wersji serwera Radius. Nie musimy konfigurować radiowego punktu dostępowego a następnie uruchamiać procedury logowania z laptopa, wystarczy zapuścić test i od razu widać jaki przychodzi rezultat.

Jeżeli **eapol_test** uruchomimy na serwerze, na którym działa nasz główny serwer Radius, to, korzystając z testowego konta na polskim centralnym serwerze Radius, możemy sprawdzić poprawnośc komunikacji z serwerem krajowym, nie korzystając z pośrednictwa naszego własnego serwera. Z tego testu można korzystać w sytuacji gdy zauważany błąd ale nie mamy pewności, która strona go powoduje.

eapol_test korzysta z pliku konfiguracyjnego zgodnego z **wpa_supplicant**. Zalecamy, aby tworzyć odrębny plik dla każdego typu EAP.

Poniżej zamieszczamy przykłady plików konfiguracyjnych dla najczęściej używanych metod EAP.

Używanie eapol_test

Podstawowe opcje programu eapol_test to:

`-c nazwa_pliku` – wskazanie pliku konfiguracyjnego

`-a adres_ip` – adres serwera Radius (w postaci numerycznej)

`-s sekret` – wspólny klucz używany między serwerem radius i klientem

Pozostałe opcje można zobaczyć wywołując eapol_test z opcją -h.

W ramach prac projektu PIONIER przygotowane zostały rozszerzenia programu eapol_test. Rozszerzenia zostały wprowadzone do dystrybucji wpa_supplicant począwszy od wersji 0.6.4 i pozwalały m.in. na obsługę atrybutu Chargeable-User-Identifier. Począwszy od wersji 0.6.7 rozszerzenie dedykowane obsłudze atrybutu Chargeable-User-Identifier zostało zastąpione przez nową metodę pozwalającą na dodanie dowolnych atrybutów RADIUS (w tym Chargeable-User-Identifier).

Dodane opcje programu eapol_test to:

`-A adres_ip` – adres klienta (w postaci numerycznej), którego ma użyć eapol_test (musi być zgodny z adresem jednego z interfejsów komputera, na którym program jest uruchamiany).

`-N specyfikacja atrybutu` – powoduje wysłanie dodatkowego atrybutu RADIUS wyspecyfikowanego w postaci `attr_id:syntaks:wartość`, gdzie attr_id to numer identyfikatora atrybutu; syntaks to s, d lub x oznaczajace łańcuch, liczbę całkowitą, ciąg szesnastkowy; wartość – wartość atrybutu; jeżeli podano tylko attr_id, to zostanie wysłany watrybut o wartości NULL. Opcja -N może być użyta wielokrotnie w celu dodania rożnych atrybutów.

Przykłady użycia opcji -N

`-N 89`	wyślij atrybut Chargeable-User-Identity o wartości NULL (odpowiednik użycia dawnej opcji `-i`)
`-N 89:s:XXXX`	wyślij atrybut Chargeable-User-Identity o wartości XXXX (odpowiednik użycia dawnej opcji `-I XXXX`)
`-N "32:s:UMK:UCI"`	wyślij atrybut NAS-Identifier o wartości UMK:UCI
`-N "64:d:13"`	wyślij atrybut Tunnel-Type o wartości 13 (VLAN)

-N "64:x;0x0000000d"	identycznie jak wyżej
-N "64:x;0x0200000d"	identycznie jak wyżej ale użyj tagu o wartości 1 dla atrybutu Tunnel-Type

Przykładowe pliki konfiguracyjne

EAP-TTLS/PAP

```
network={
        ssid="eduroam"
        key_mgmt=WPA-EAP
        eap=TTLS
        ca_cert="certyfikat_ca.cer"
        anonymous_identity="@domena.pl"
        identity="uzytkownik@domena.pl"
        password="haslo_uzytkownika"
        phase2=”auth=PAP"
}
```

linia zawierająca deklarację anonymous_identity jest opcjonalna. W powyższym przykładzie zastosowana jest konwencja ustawiania pustej nazwy użytkownika (zgodnie z zaleceniem RFC 4282), wg innej konwencji stosuje się nazwę „anonymous”.

PEAP/MSCHAPv2

```
network={
        ssid="eduroam"
        key_mgmt=WPA-EAP
        eap=PEAP
        ca_cert="certyfikat_ca.cer"
        anonymous_identity="@domena.pl"
        identity="uzytkownik@domena.pl"
        password="haslo_uzytkownika"
        phase2="auth=MSCHAPV2"
}
```

linia zawierająca deklarację anonymous_identity jest opcjonalna, stosują się te same uwagi, co w przypadku EAP-TTLS.

EAP-TLS

```
network={
        ssid="eduroam"
        key_mgmt=WPA-EAP
        eap=TLS
        ca_cert="certyfikat_ca.cer"
        identity="uzytkownik@domena.pl"
        private_key="nazwa_pliku.p12"
        private_key_passwd="haslo_do_plku_p12"
}
```

powyższy przykład stosuje certyfikat i klucz prywatny użytkownika spakowane w jednym pliku typu PKCS #12.

Kompilacja eapol_test

Ze strony http://hostap.epitest.fi/wpa_supplicant/ należy pobrać najnowszą wersję rozwojową oprogramowania wpa_supplicant.

Paczka rozpakowuje się do katalogu `wpa_supplicant-0.x.y` (gdzie x.y oznaczają numer wersji, np. 6.7). Kompilację przeprowadza się w podkatalogu `wpa_supplicant`. Przed uruchomieniem kompilacji, w tym katalogu należy stworzyć plik konfiguracyjny `.config`. Gotowy plik znajduje się pod adresem: http://www.eduroam.pl/Dokumentacja/eapol_test-config.gz, wystarczy go zachować, a następnie po rozpakowaniu zmienić nazwę na `.config` (kropka na początku nazwy jest istotna!) i wgrać do katalogu `wpa_supplicant-0.x.y/wpa_supplicant`.

Następnym krokiem jest wywołanie polecenia

```
make eapol_test
```

Jeżeli wszystko jest w porządku, tzn. są zainstalowane kompilatory i biblioteki SSL, to powinniśmy uzyskać skompilowany program **eapol_test**, który można skopiować do dowolnej lokalizacji.