Maja Górecka-Wolniewicz

Wykorzystanie technologii RadSec w usłudze eduroam

1. Motywacje

Protokół RADIUS, zdefiniowany w RFC 2865, jest podstawą funkcjonowania usługi eduroam. Obecna zasada działania eduroam opiera się na systemie hierarchicznym. Trzon stanowią dwa serwery poziomu root (główny i zastępczy), kolejny poziom to serwery krajowe, dalej serwery instytucji itd. Komunikacja między serwerami wykorzystuje mechanizm proxy. Polega on na tym, że serwer, do którego przychodzi zlecenie, na podstawie analizy nazwy użytkownika umieszczonej jako tożsamość mobilna (outer identity) ustala, kto ma obsłużyć zlecenie. Jeśli domena użytkownika (realm) nie jest obsługiwana na bieżącym serwerze, serwer przekazuje zlecenie do innego serwera, zdefiniowanego w konfiguracji jako właściwy dla danego użytkownika, lub do serwera nadrzędnego. Komunikacja serwer-serwer, w której jeden z serwerów staje się klientem to właśnie funkcjonalność proxy.

Obecnie projekt eduroam korzysta głównie z oprogramowania freeRADIUS i Radiator. Przekazywanie na zasadzie proxy odbywa się poprzez sieć globalną, a pakiety są przesyłane na ogół bez dodatkowych zabezpieczeń, poza mechanizmem kodowania w ramach atrybutu EAP-Message. Jedyną metodą poprawy bezpieczeństwa jest wprowadzenie tunelowania zleceń (iptunnel) lub zastosowanie protokołu IPsec.

Warto podkreślić, że sporo przekazywanych danych jest zapisywanych otwartym tekstem – tak są przesyłane nazwa użytkownika, numer IP, czas logowania.

Największe niedociągnięcia protokołu RADIUS są efektem zawodności wynikającej z oparcia protokołu na protokole UDP. UDP został wybrany do transmisji pakietów RADIUS głównie w celu uproszczenia działania, jednak ta metoda nie sprawdza się w sytuacji stosowania w RADIUS-ie skomplikowanych protokołów EAP, wymagających komunikacji, na którą składa się wymiana wielu pakietów. UDP nie gwarantuje dostarczenia komunikatu, protokół RADIUS wprawdzie dopuszcza określoną liczbę retransmisji, ale nadal nie mamy pewności, czy zlecenie zostało dostarczone do celu – tego typu zawodność drastycznie wzrasta w przypadku przeciążonych sieci. Ponadto RADIUS nie pozwala autorytatywnie, za pomocą techniki PKI, stwierdzić, że klient jest prawdziwy – służą do tego klucze tajne ustanawiane przez komunikujące się strony, ich bezpieczeństwo gwarantuje wiarygodność klienta, ale ta metoda znajduje swoich oponentów.

Wiele ze wskazanych powyżej problemów zostało uwzględnionych w protokole Diameter, definiowanym jako następca RADIUS-a (RFC 3588). Protokół ten będzie zapewne w przyszłości stopniowo wypierał RADIUS-a, jednak obecnie implementacje Diameter nie są jeszcze gotowe. Istnieje natomiast mechanizm zastępczy, posiadający nawet gotową implementację, działającą w oparciu o protokół o nazwie RadSec.

2. Opis protokołu RadSec

Wychodząc naprzeciw problemom związanym z niezawodnością i bezpieczeństwem transmisji pakietów RADIUS stworzono projekt o nazwie RadSec. W dokumencie pt. "RadSec – a secure, reliable Radius Protocol", opracowanym przez grupę Open System Consultants (OSC), działającą od 1991 w Melbourne, w Australii przedstawiono zasady funkcjonowania protokołu RadSec współpracującego z RADIUS-em.

Oprogramowanie Radiator implementuje funkcjonalność RadSec, dając w efekcie możliwość bezpiecznej i niezawodnej komunikacji między klientem a serwerem.

Standard opisujący to podejście jest fazie przygotowania, pod koniec 2006 roku pojawił się Internet-Draft o nazwie draft-winter-radsec-00, dokument nosi tytuł: "RadSec version 2 – A Secure and Reliable Transport for the RADIUS Protocol".

Radiator implementuje **protokół RadSec wersji 1** (zgodnie z dokumentem OSC), jego funkcjonowanie opiera się na poniższych zasadach:

1. RadSec służy do przenoszenia ruchu między dwoma współpracującymi serwerami RADIUS lub między klientem RADIUS a serwerem RADIUS. W obu sytuacjach jeden koniec połączenia działa jako klient, drugi jako serwer.
2. Serwer RadSec może funkcjonować jako klient innego serwera RadSec.

3. Serwer RadSec słucha na nadchodzące połączenia na porcie TCP, domyślnym numerem portu jest 2083 (oficjalny nr portu RadSec wg IANA).
4. Klient RadSec ustanawia połączenie z serwerem RadSec wg znanego mu adresu IP i portu serwera. Strumień połączenia może wspierać funkcjonalność TCP/IP keepalive.
5. Po ustanowieniu połączenia rozpoczyna, się faza zwana powitaniem TLS (TLS handshaking). Stosowany jest protokół TLS v.1. Serwer RadSec musi zaprezentować certyfikat PKI, w którym pole CN właściciela certyfikatu jest identyczne z nazwą DNS odpowiadającą adresowi IP, na którym serwer nasłuchuje. Powitanie kończy się niepowodzeniem, jeśli nie zostanie pokazany właściwy certyfikat, dający się z sukcesem zweryfikować.
6. Jeśli z jakiegoś powodu na tym etapie połączenie musi zostać zerwane, klient będzie próbował ponownie połączyć się po określonym w konfiguracji interwale czasu (domyślnie 5 sekund).
7. Nie jest wspierane wznowienie sesji TLS.
8. Po ustanowieniu połączenia TLS zlecenia i odpowiedzi protokołu RADIUS są przekazywane wzdłuż ustalonego strumienia. Pakiety RADIUS są kodowane w tym samym formacie jak w przypadku stosowania protokołu UDP. Długość pakietu RADIUS określa granice rekordu – nie są stosowane separatory rekordów.
9. Na wzór konwencjonalnego protokołu RADIUS używanego przy proxowaniu, każde połączenie RadSec korzysta z dedykowanego klucza wspólnego, znanego wyłącznie klientowi i serwerowi.
10. Klient RadSec może używać albo typowego pola identyfikacji w pakiecie RADIUS, albo atrybutu Proxy-State do dopasowywania odpowiedzi do zlecenia. Zaleca się stosowanie atrybutu Proxy-State w celu uniknięcia ograniczeń nazwy wynikających z 8-bitowego pola identyfikatora.
11. Odpowiedzi RADIUS mogą być przekazywane do serwera w innej kolejności niż przyjęte zlecenia. Nie ma gwarancji, że odpowiedź nadejdzie na konkretne zlecenie – np. jeśli serwer przekazuje zlecenie dalej, to taki serwer nie wysyła odpowiedzi.
12. Klient może używać atrybutów User-Name, Realm lub innej kombinacji atrybutów do określenia, do którego serwera przesłać zlecenia (proxy oparte na nazwie domenowej).
13. Zaleca się używanie szyfrowania TLS oraz wymuszenie dwustronnego uwierzytelnienia między klientem a serwerem, w szczególności, gdy protokół jest przekazywany w niezabezpieczonej sieci globalnej.

RadSec wersji 2 został szczegółowo opisany w dokumencie Internet-Draft. Jego funkcjonalność jest zgodna z wersją 1. Dodatkowe własności standardu wersji 2 obejmują:

- definicję sposobu szybkiej detekcji problemów i wprowadzenie metod raportowania o problemie,
- wycofanie adresów IP i haseł wspólnych do wzajemnej identyfikacji stron komunikacji, zezwolenie na dynamiczne ustawianie połączeń, które nie były wcześniej skonfigurowane (do tego celu można np. wykorzystać rekordy NAPTR w DNS-ie).

RadSec wersji 2, tak jak jego poprzednik funkcjonuje w oparciu o protokół TCP, gwarantując w ten sposób niezawodny transport. Używa domyślnie portu 2083, zarejestrowanego przez IANA dla potrzeb implementacji RadSeca w Radiatorze, ale obecnie trwają dyskusję nad zmianą portu na 1812/TCP.
Internet-Draft dokładnie precyzuje jakiego typu zlecenia i odpowiedzi mogą być przesyłane między dwoma węzłami RadSec. Poza pakietami Access-Request, Access-Chalange, Access-Reject, Access-Accept, Accounting-Request i Accounting-Response, są akceptowane zapytania Status-Server oraz pakiety rozszerzenia RADIUS-a: Disconnect-Request, CoA-Request oraz odpowiedzi na te pakiety. Pakiety innych typów powinny być odrzucane. Węzeł RadSec działa jednocześnie jako klient i serwer.
Ponieważ pomiędzy węzłami są na ogół tworzone tunele TLS do transmisji danych, a ustanawianie połączenia TLS jest operacją kosztowną (z punktu widzenia wykorzystania zasobów komputerowych), zaleca się utrzymywania otwartych połączeń przez pewien czas. Utrzymywanie połączeń (connection keepalive) może być realizowane na dwa sposoby:

- poprzez wykorzystanie opcji gniazda TCP – jeśli ta własność jest aktywna w danym systemie operacyjnym,
- poprzez transfer pakietów Status-Server – węzeł RadSec, który zainicjował połączenie powinien wysyłać do partnera w ustalonych w konfiguracji odstępach czasu (zaleca się stosowanie interwałów czasowych zawartych w przedziale 1 min. - 60 min.) pakiet Status-Server i odbierać w odpowiedzi Access-Accept, węzły RadSec powinny odpowiadać na te zlecenia regularnie, jeśli interwał pomiędzy zleceniami jest nie mniejszy niż 1 min.

Internet-Draft zaleca stosowanie do utrzymywania połączeń TCP opcji gniazda TCP, jeśli jest ona zaimplementowana w systemie.
Istotnym problemem jest wykrycie nieaktywnych węzłów – jest bardzo ważnie, by odbyło się to możliwie szybko (typowo timeouty związane z brakiem połączenia są długie). W przypadku, gdy jest stosowane regularne wysyłanie komunikatu Status-Server, brak odpowiedzi w zadanym czasie można traktować jako wskazanie, że serwer przestał być dostępny.
W wersji 2 RadSeca zaostrzono wymagania związane z ustanawianiem połączenia TLS. Po ustaleniu połączenia TCP następuje faza powitania TLS w celu ustanowienia sesji TLS. Obie strony wzajemnie muszą zaprezentować certyfikaty używane w komunikacji. Ponieważ niezawodna identyfikacja serwera jest możliwa wyłącznie wówczas, gdy jest stosowany DNSSec pola CN lub dNSName w certyfikacie mogą służyć do weryfikacji certyfikatu partnera. Jeżeli polegamy na tradycyjnej usłudze DNS, to ustanowienie połączenia nie może odbyć się w oparciu o te pola. Operatorzy infrastruktury RADIUS powinni zdefiniować własny model zaufania, np. polegający na stosowaniu konkretnego rozszerzenia certyfikatu do przenoszenia informacji gwarantującej rozpoznanie partnera. W projekcie eduGAIN stosowanym dla usługi eduroam nowej generacji, autoryzacja klienta odbywa się na podstawie SubjectAltName, który musi zawierać URN z puli nazewnictwa eduroam.
Po ustanowieniu połączenia TLS pakiety są przekazywane przez szyfrowany tunel. Granice pakietu są wyliczane na podstawie pola długości pakietu RADIUS. Internet-Draft dokładnie precyzuje jakie metody szyfrowania powinny być implementowane przez węzeł RadSec (p. 3.2 dokumentu).
Rozdział dotyczący zastosowania wspólnego klucza (shared secret) w protokole RadSec wersji 2 jest w trakcie dyskusji – postuluje się zastosowanie rozwiązania zalecanego dla RADIUS-a działającego w ramach IPSec.

3. Implementacje

3.1. Radiator

Pierwszą działającą implementacją RadSeca był Radiator 3.13, wydany w marcu 2005. Radiator jest oprogramowaniem komercyjnym. Korzystanie z RadSeca w Radiatorze jest bardzo proste. Po pierwsze należy w pliku konfiguracyjnym zamieścić blok <ServerRADSEC> ... </ServerRADSEC> o postaci:

```
<ServerRADSEC>
 Secret mysecret
 UseTLS
 TLS_CAFile ścieżka_do_pliku_z_certyfikatem_nadrzędnym
 TLS_CertificateFile ścieżka_do_pliku_z_certyfikatem_serwera
 TLS_PrivateKeyFile ścieżka_do_pliku_z_kluczem_prywatnym_serwera
 TLS_PrivateKeyPassword hasło_do_klucza_prywatnego
 TLS_CertificateType PEM
</ServerRADSEC>
```

Ta deklaracja sprawia, że serwer rozpoczyna nasłuch na domyślnym porcie RadSec (2083, jeśli chcemy zastosować inny port, należy w bloku umieścić dyrektywę Port nr_portu).
Aby nasz serwer realizował połączenia z wybranym serwerem zgodnie z protokołem RadSec należy w deklaracji obsługi domeny użyć bloku <AuthBy RADSEC> zamiast <AuthBy RADIUS>, np.

```
<Handler Realm=/.+/,Client-Identifier=/^(?!ROOTRADIUS$)/>
 <AuthBy RADSEC>
  <Host etlr1.radius.terena.nl>
  Port 2083
```

```
  Protocol tcp
  NoreplyTimeout 5
  UseTLS
  Secret mysecret
 </Host>
</Handler>
```

Implementacja Radiatora pozwala używać albo protokołu TCP, albo SCPT (Stream Control Transmission Protocol, http://www.sctp.org).

3.2. RadSec-Proxy

Jest to oprogramowanie aktualnie przygotowywane. Jego autorem jest Stig Venaas z UNINETT. Pomysł napisania tzw. generic RadSec-Proxy wynikł z faktu, że przeciągały się rozmowy na temat włączenia protokołu RadSec do ogólnie dostępnego oprogramowania FreeRADIUS. RadSec-Proxy będzie oprogramowaniem bezpłatnym, którego funkcjonalność wystarczy do uruchomienia serwera proxy. Podczas spotkania grupy roboczej TF-Mobility, 12.01.2007, Stig Venaas prezentując projekt określił stan prac następująco: "w połowie gotowy". Działa już prototyp, w oparciu o który były przeprowadzane pierwsze testy. Brakuje kilku ważnych funkcji, jak sprawdzanie certyfikatów, przekazywanie komunikatów accountingowych, ponowne szyfrowanie niektórych atrybutów, samodzielna retransmisja pakietów (obecnie serwer tylko przekazuje zapytania retransmisji).
Ukazanie się oprogramowania RadSec-Proxy autorstwa Stiga Venaasa zapewne będzie stanowiło przełom w dalszych losach usługi eduroam.

4. Testy RadSeca w ramach eduroam

4.1. Założenia

W ramach inicjatywy GÉANT JRA5 są prowadzone prace nad poprawą infrastruktury uwierzytelniania i autoryzacji w roamingu międzydomenowym. W ramach zdobywania doświadczenia praktycznego w tym zakresie opracowano rozbudowany plan testów, których celem jest sprawdzanie funkcjonowania nowych technologii w infrastrukturze eduroam. W testach używano protokołów RADIUS i RadSec. Wyszukiwanie partnerów komunikacji odbywało się albo statycznie, albo poprzez wykorzystanie usługi DNS.
Korzystanie z RadSeca wymaga ustanowienia infrastruktury PKI.
W testach wykorzystywana jest właściwość Radiatora zw. DNSROAM – możliwość przekazania zlecenia serwerowi, który nie jest wskazany statycznie, lecz zostanie wyszukany na bieżąco za pomocą procedur określonych w bloku <Resolver> (blok <Resolver> jest niezbędny, jeśli konfigurujemy w Radiatorze korzystanie z AuthBy DNSROAM). Podstawowym celem stosowania AuthBy DNSROAM jest możliwość ustanowienia bezpiecznych, niezawodnych połączeń w ramach federacji. Pod pojęciem federacji RADIUS (zwanej również sieć RADIUS – mesh) kryje się zbiór serwerów, działających w ramach niezależnych, ale współpracujących ze sobą instytucji. Celem jest umożliwienie członkom jednej instytucji korzystania z zasobów kontrolowanych przez RADIUS-a w innej, współpracującej instytucji. Zastosowanie bloku AuthBy DNSROAM razem z usługą DNS do przechowywania informacji o serwerze docelowym dla danej domeny (realm), pozwala na wygodne i dobrze skalowalne zarządzanie federacją RADIUS. Działanie AuthBy DNSROAM oparte jest na nazwie domeny (realm) wyodrębnionej z nazwy użytkownika (User-Name). Najpierw jest poszukiwany blok <Route> odpowiadający danej domenie. Jeśli nastąpi dopasowanie, a docelowy serwer jest serwerem RadSec, jest tworzony tunel TLS, jeśli serwer działa zgodnie z protokołem RADIUS zlecenie jest transportowane przy użyciu protokołu UDP. Jeśli nie zostanie znaleziony żaden blok <Route> dla danej domeny, to serwer docelowy jest poszukiwany za pomocą usługi DNS. Funkcjonowanie tego typu operacji opiera się na wykorzystaniu rekordów NAPTR (RFC 3403), ewentualnie rekordów A lub AAAA.

Testy związane z modernizacją infrastruktury eduroam bazowały na założeniu, że przejście na nowy styl działania nie nastąpi w określonym momencie, lecz będzie obowiązywała tzw. faza przejściowa. Oznacza to,

że wybrane technologie muszą uwzględniać koegzystencję kilku modeli.

4.2. Modele stosowane w testach

Wybrane do testów modele mogą zostać sklasyfikowane w dwóch grupach: hierarchiczne oraz w postaci sieci (typu peer-to-peer).

Model w pełni hierarchiczny
Ten model odpowiada obecnej sytuacji, w której istnieje serwer nadrzędny (root), poniżej zlokalizowane są serwery krajowe, dalej serwery instytucji. Każdy węzeł RADIUS miałby w tym układzie zostać zastąpiony węzłem RadSec. Każdy węzeł ma skonfigurowane statycznie dane dotyczące potencjalnych partnerów komunikacji. Z punktu widzenia środowiska PKI obsługującego takie rozwiązanie, każdy węzeł musi ufać 'przyległym' węzłom w swojej hierarchii.

Model statycznej sieci dla poziomu górnego i hierarchiczne poziomy niższe
Jeśli węzły poziomu górnego miałyby statyczną konfigurację dotyczącą każdego węzła poziomu głównego (krajowego), istnienie serwera nadrzędnego (root) staje się zbędne. Takie rozwiązanie zostało jednak odrzucone jako niepraktyczne.

Model dynamicznej sieci dla poziomu głównego i hierarchiczne poziomy niższe
Zasada jest taka sama jak w poprzednim modelu, z tym, że węzły poziomu głównego realizują dynamicznie nawiązywanie połączenia, korzystając z DNSROAM. W tym rozwiązaniu niekorzystnym aspektem jest niejednorodność struktury.

Model dynamicznej sieci
Takie rozwiązanie jest obecnie tylko teoretycznie, w praktyce, wobec dużej liczby węzłów uczestniczących w usłudze eduroam, możliwość niezawodnego dynamicznego wykrywania partnerów jest mało prawdopodobna. W tej "idealnej" sytuacji nie byłyby potrzebne serwery poziomu krajowego. Udział sieci krajowych w przedsięwzięciu wiązałby się raczej z utrzymywaniem infrastruktury PKI wspierającej ten model.

Mieszany model poziomów niższych, model sieci na poziomie głównym
W tym modelu również zbędny jest serwer nadrzędny. Serwery poziomu krajowego porozumiewają się między sobą. Poniżej mamy do czynienia z strukturą mieszaną: istnieją kraje nadal stosujące model hierarchiczne, gdzie indziej jest stosowany model dynamicznej sieci.

Model "spuścizny"
Jest to najbardziej realna sytuacja: część węzłów używa aktualnej konfiguracji eduroam, inne węzły stosują RadSeca. Węzły RadSec stosują dynamiczne wyszukiwanie partnerów (DNSROAM) lub przekazują zlecenia poziom wyżej, jeśli wyszukanie dynamiczne nie daje wyników. Węzły poziomu krajowego muszą mieć umiejętność translacji pakietów RadSec – RADIUS.

4.3. Przebieg testów

W czasie testów realizowanych w 2005/2006 roku uwzględniono modele: w pełni hierarchiczny, mieszany: hierarchiczny poziom niższy i model sieci na poziomie najwyższym oraz konfiguracje sieci serwerów RADIUS (peer-to-peer). W testach wzięli udział przedstawiciele SURFnetu, CESNET-u, ISTF, ARNES, UNINETT, RESTENA, ACAD.
Wyniki zostały omówione szczegółowo w dokumencie opracowanym przez Telematica Instituut pt. "Radiate". Ich podsumowanie pokazuje, że wszystkie trzy wybrane scenariusze działania eduroam mogą być uwzględnione w docelowym rozwiązaniu. Najprostszym we wdrożeniu rozwiązaniem był model

hierarchiczny. Testy pokazały, że jest już możliwe zastąpienie istniejącej hierarchii RADIUS innym rozwiązaniem, bez utraty stabilności usługi. Dodatkowych testów wymagają kwestie dynamicznego wyszukiwania partnerów.

4.4. Kontynuacja prac testowych w nowym środowisku PKI

W ramach prac grupy związanej z projektem JRA5 powstał projekt infrastruktury PKI dedykowanej środowisku eduGAIN (GÉANT Authentication and Authorization Infrastructure). Obecnie działa już testowe PKI dla potrzeb eduGAIN. Dalsze testy RadSeca będą bazować na tej infrastrukturze. Został przygotowany interfejs do wystawiania certyfikatów.
Polska zgłosiła swoje uczestnictwo w drugiej fazie testów. Konfiguracja serwera głównego PL (działającego obecnie na oprogramowaniu Radiator) jest gotowa. Serwer korzysta z certyfikatu wystawionego przez eduGAINSCA – urząd certyfikacji dedykowany do wystawiania certyfikatów usług eduGAIN.
Testy są planowane na najbliższe miesiące.

4.5. Testy oprogramowania RadSec-Proxy

Pierwsza wersja oprogramowania RadSec-Proxy autorstwa Stiga Venaasa ukazała się na początku lutego. Polska bierze udział w testach tego oprogramowania. Rozwiązanie polegające na stosowaniu RadSec-Proxy może być docelowo interesującym rozwiązaniem, nawet jeśli pojawi się FreeRADIUS-a implementacja RadSeca. Z pewnością w polskim środowisku eduroam zaczną się coraz liczniej pojawiać ośrodki korzystające z komercyjnych produktów RADIUS, w których protokół RadSec jako nie obowiązujący jeszcze standard nie jest zaimplementowane. W tej sytuacji zastosowanie dedykowanego rozwiązania RadSec-Proxy będzie bardzo wygodne.
W ramach dotychczasowych alfa-testów oprogramowania RadSec-Proxy zostoły przetestowane konfiguracje:

- radsecproxy współpracujący po bezpiecznym połączeniu z RadSeciem oprogramowania Radiator i dalej wg protokołu UDP z freeradiusem,
- radsecproxy współpracujący bezpośrednio z freeradiusem.

W tych testach stosowane certyfikaty są poświadczone przez urząd uniwersytecki. Docelowo serwery będą działały w infrastrukturze PKI eduGAIN (serwer radius1.eduroam.pl ma już wystawiony certyfikat w tej infrastrukturze).

Bibliografia

1. RFC 2865 - Remote Authentication Dial In User Service (RADIUS)
2. RFC 3588 - Diameter Base Protocol
3. Dokumentacja programu Radiator
4. “RadSec version 2 – A secure and reliable Transport for the RADIUS Protocol”, Internet-Draft, draft-winter-radsec-00