Kerberos Authentication Service w wersji 5

Michał Grzejszczak

20 stycznia 2003

Spis treści

1 Wstęp

Kerberos Authentication Service został stworzony przez Massachusetts Institute of Technology (MIT) aby zapewnić bezpieczeństwo połączeń sieciowych wykorzystywanych przez system Athena. Pierwsze trzy wersje były używane w projekcie Athena, natomiast czwarta została przyjęła się szeroko poza MIT. Piąta wersja powstała na podstawie kontaktów z użytkownikami i ich doświadczeń oraz oczekiwań względem systemu.

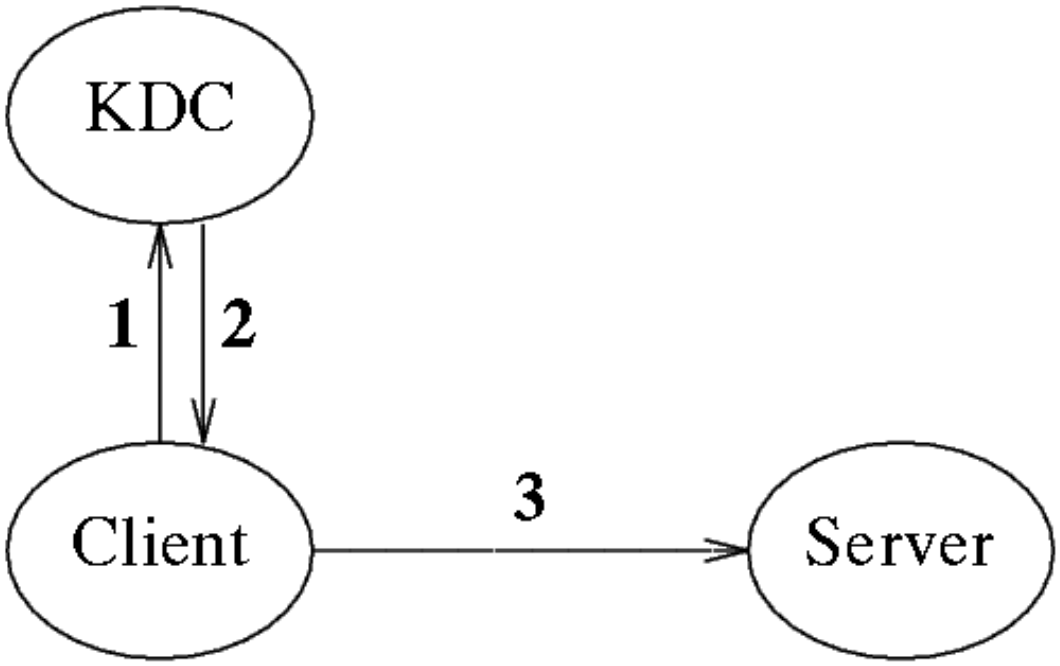
2 Model Kerberos

2.1 Key Distribution Center

KDC jest integralną częścią Kerberos. Zapewnia ona czasowo ważne „bilety", które stanowią swego rodzaju potwierdzenia dla łączących się programów. Klient aby przedstawić się serwerowi wysyła mu swój bilet odebrany od KDC. KDC zapewnia, że bilet o takim numerze nie zostanie pobrany przez innego klienta do czasu upływu ważności pierwszego. KDC jest zatem swego rodzaju „mężem zaufania". Instalacja sytemu Kerberos zawiera obowiązkowo KDC, które wyznacza „realm". Jest to obszar w sieci, w którym aplikacje chcące skorzystać z usługi Kerberos muszą się połączyć z lokalnym KDC i odebrać bilet.

Serwery, z którymi można się połączyć za pomocą Kerberos ustanawiają z KDC tajny klucz, na podstawie którego serwer może potem identyfikować prawdziwość biletów przysyłanych od klientów.

Rysunek 1: Komunikacja wstępna z KDC



2.2 Przebieg autentykacji

K_csdsa

Klient, który chce połączyć się z danym serwerem musi dostać bilet, do tego serwera od KDC. Kroki autentykacji:

1. Klient wysyła do KDC komunikat ze swoim identyfikatorem, jakimś unikalnym identyfikatorem dla komunikatu (np. timestamp) [nonce] i identyfikatorem serwera z jakim chce się później połączyć.

2. Odebrawszy wiadomość KDC ustala klucz dla sesji między klientem a serwerem (zapewnia jednocześnie, że nie wyda tego samego klucza do końca ważności poprzedniego). Tworzy bilet, na który składa się:

   - Identyfikator klienta.
   - Klucz sesji.
   - Daty (timestamp) początku ważności i końca ważności biletu.
   - Inne dane.

   Po tym wysyła do klienta bilet zaszyfrowany tajnym kluczem serwera, do którego klient chce się dostać (ten klucz jest ustalony wcześniej pomiędzy KDC i serwerem) oraz klucz

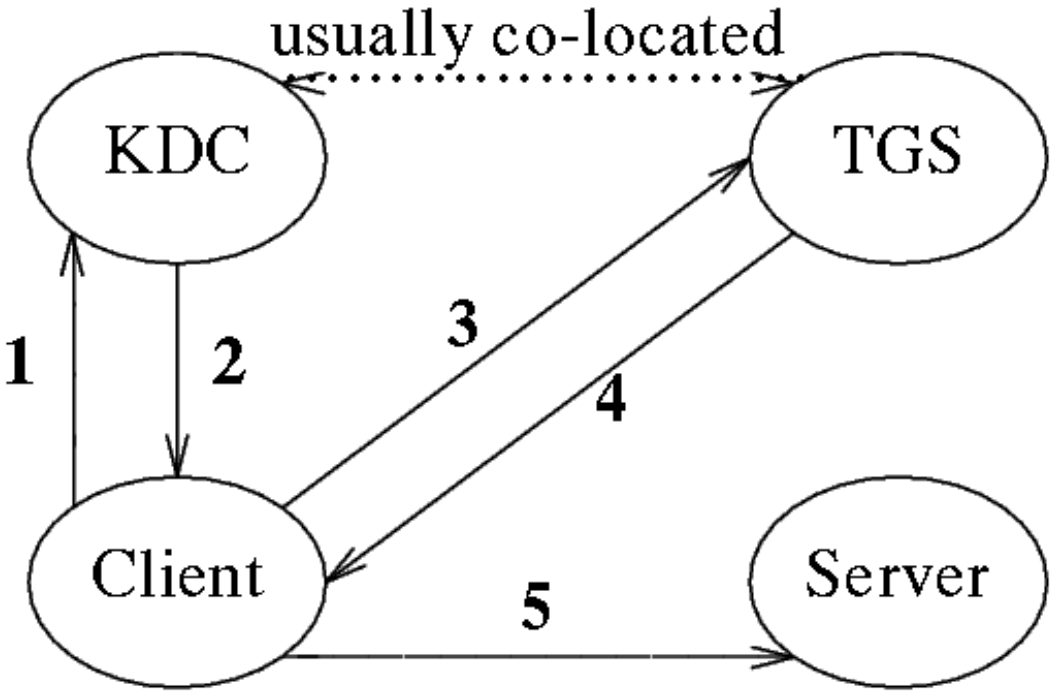
sesji i nonce zaszyfrowane kluczem tajnym klienta (otrzymanym z hasła). [Potencjalna słabość]

3. Klient odebrawszy wiadomość odszyfrowuje klucz sesji i nonce (nonce żeby się upewnić czy to odpowiedź na jego zapytanie). Potem tworzy „authenticator", czyli też jakiś timestamp i wysyła „authenticator" zaszyfrowany kluczem sesji (już go zna) i bilet (z nim nic nie mógł zrobić).
4. Serwer, po odebraniu komunikatu odszyfrowuje bilet, sprawdza id klienta, sprawdza daty ważności biletu i ewentualnie odsyła komunikat zaszyfrowany kluczem sesji, który właśnie odszyfrował jeśli klient wymaga od niego autentykacji.

Po tych krokach serwer i klient posiadają obaj klucz sesji, unikalny w skali realmu i mogą go wykorzystać do szyfrowania komunikatów między sobą.

2.2.1 Dodatkowa wymiana komunikatów

Rysunek 2: Dodatkowa komunikacja



Aby zmniejszyć skalę niebezpieczeństwa związanego używaniem klucza tajnego klienta, używa się dodatkowej wymiany komunikatów z serwerem Ticket Granting Serwer (TGS). Powyższe

kroki służą zazwyczaj do uzyskania biletu i szyfru sesji do serwera TGS, który jest logicznie odrębny od KDC ale działa na tych samych danych i jest zwykle zlokalizowany w tym samym węźle. Klient odbiera od tego serwera bilet do docelowego serwera stąd utrzymywanie klucza klienta przez klienta po odebraniu komunikatu od TGS nie jest konieczne.

3 Cechy Kerberos w wersji 5

3.1 Metody szyfrowania

Kerberos został zaprojektowany z myślą o wykorzystaniu technologii klucza symetrycznego. Jednakże zwiększająca się infrastruktura wspierająca technologię klucza publicznego skłania do umożliwienia wykorzystywania w Kerberos również tej techniki. Obecnie wersja 5 Kerberos obsługuje moduły szyfrujące co umożliwia łatwy wybór metody szyfrowania jakiej chcemy używać (nie było tego w wersji 4). Implikuje to dodatkowe znaczniki w przesyłanych komunikatach informujące odbierającego jaki szyfr zastosowano podczas szyfrowania wiadomości.

Dostępność różnych szyfrów jest dość duża, jednak ważne jest aby szyfr zapewniał integralność, czyli wykrywał zmiany w zaszyfrowanej wiadomości powstałe podczas transportu. Większość rozwiązań można bogacić o tą cechę dodając w jakiś sposób sumę CRC szyfrowanej wiadomości itp.

3.2 Obsługiwane protokoły

Kerberos obsługuje większość protokołów sieciowych (nie było tak w wersji 4) co umożliwia zastosowanie go w wielu środowiskach.

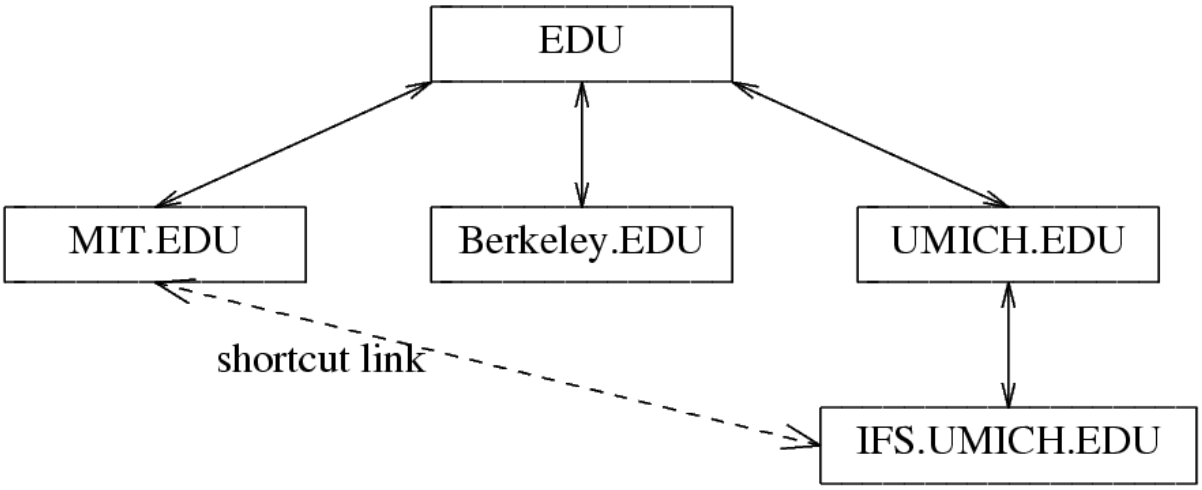
3.3 Hierarchia realmów

Realmy Kerberos zhierarchizowane są w drzewo. Jest to związane z tym jak wygląda łączenie się aplikacji z różnych realmów Kerberos. Aplikacja klienta niejako „chodzi” po drzewie otrzymując od każdego napotkanego KDC bilet do KDC o 1 węzeł bliższego docelowemu. Istnieje możliwość zakładania w drzewie linków aby zmniejszyć ruch między uczęszczanymi realmami. Bilet uzyskany w efekcie takiego przejścia nosi sygnatury wszystkich realmów, przez które przeszedł. Umożliwia to nie zaakceptowanie przez aplikację biletu, który przeszedł przez realm uważany za „podejrzany”.
Taka realizacja połączeń zmniejsza ilość kluczy jakie należy wymieniać między realmami aby utrzymać system w pełnej sprawności. Połączeń jest z grubsza *O(n)*, a nie jak w wersji 4 *O(n*n)*.

4 Literatura

Dostępne materiały w sieci:

Rysunek 3: Drzewo połączeń międzyrealmowych



1. http://www.mit.edu/kerberos/www/ - oficjana strona oryginalnej implementacji.
2. http://www.sun.com/software/security/kerberos/ - strona z informacjami o implementacji SUNa.
3. http://www.microsoft.com/technet/prodtechnol/windows2000serv/deploy/kerberos.asp - strona z informacjami o implementacji w Windows 2000. Implementacje dla innych windowsów można znaleźć w dziale technet.