

Bezpieczeństwo systemów komputerowych

Kerberos

Marcin Peczarski

Instytut Informatyki Uniwersytetu Warszawskiego

19 grudnia 2010

Co to jest Kerberos?

- ▶ System uwierzytelniania z zaufaną trzecią stroną
- ▶ Zaufany arbiter, uwierzytelniający użytkowników i udostępniający im dostęp do różnych usług sieciowych
- ▶ Jedno uwierzytelnienie umożliwia komunikację z wieloma serwerami
- ▶ Oparty na szyfrowaniu symetrycznym
- ▶ Oparty na protokole Needhama–Schroedera
- ▶ Opracowany w MIT w ramach projektu Athena
- ▶ Kerberos (Cerber) w mitologii greckiej trójgłowy pies strzegący wejścia do Hadesu

Problem

- ▶ Otwarte rozproszone środowisko sieciowe
- ▶ Wiele stacji roboczych
- ▶ Wiele serwerów usług
- ▶ Wielu użytkowników, korzystających z różnych stacji roboczych i różnych usług

Niebezpieczeństwa

- ▶ Użytkownik może uzyskać dostęp do stacji roboczej i podszyć się pod innego użytkownika pracującego na tej stacji.
- ▶ Użytkownik może zmienić adres sieciowy stacji roboczej i udawać, że korzysta z innej stacji roboczej.
- ▶ Użytkownik może podsłuchiwać wymianę danych w sieci.
- ▶ Nie można polegać na kontroli tożsamości użytkownika przeprowadzanej przez stację roboczą.
- ▶ Nie można polegać na kontroli tożsamości użytkownika przeprowadzanej przez klienta usługi.
- ▶ Uwierzytelnianie się użytkownika wobec każdego serwera jest niepraktyczne, każdy serwer musiałby przechowywać bazę tajnych danych identyfikacyjnych dla wszystkich użytkowników.

Model

- ▶ Każde dwie strony posiadają wspólny, tajny, znany tylko im, klucz.
- ▶ W przypadku użytkownika rolę klucza pełni kryptograficzny skrót hasła.
- ▶ Centralny serwer utrzymuje bazę klientów i ich kluczy.
- ▶ Centralny serwer zna tajne klucze wszystkich serwerów usługowych.
- ▶ Każdy serwer usługowy zna tylko swój tajny klucz.
- ▶ Centralny serwer wydaje użytkownikom bilety, umożliwiające dostęp do serwerów usługowych.

Model interakcji

- ▶ Raz na sesję użytkownika:
 - ▶ użytkownik U rozpoczyna pracę na stacji roboczej, używając klienta C;
 - ▶ C wysyła do serwera uwierzytelniającego AS (Authentication Server) żądanie biletu do usługi przyznawania biletów;
 - ▶ AS weryfikuje prawa dostępu U w bazie danych;
 - ▶ AS wysyła do C bilet do usługi przyznawania biletów TGT (Ticket Granting Ticket);
 - ▶ C prosi U o hasło, w celu weryfikacji.
- ▶ Raz na typ usługi:
 - ▶ C wysyła do serwera przyznającego bilety TGS (Ticket Granting Server) bilet TGT i żądanie przyznania biletu do serwera usługowego S;
 - ▶ TGS weryfikuje żądanie i tworzy bilet B;
 - ▶ TGS wysyła do C bilet B.
- ▶ Raz na sesję usługi:
 - ▶ C wysyła do S bilet B;
 - ▶ S weryfikuje poprawność B;
 - ▶ opcjonalnie S wysyła dane uwierzytelniające go wobec C.

Bilet

- ▶ Służy do bezpiecznej identyfikacji klienta wobec serwera.
- ▶ Jest ważny tylko dla jednego klienta i jednego serwera.
- ▶ Klient może używać go wielokrotnie, aż do wygaśnięcia jego ważności.
- ▶ Klient ani strona trzecia nie mogą zdeszyfrować zawartości biletu.

$$T_{C,S} = K_S(C \parallel S \parallel TS_1 \parallel TS_2 \parallel K_{C,S}),$$

gdzie:

K_S – szyfrowanie tajnym kluczem serwera,

C – identyfikator klienta (adres sieciowy, identyfikator użytkownika),

S – identyfikator serwera,

TS_1 – czas początku ważności biletu,

TS_2 – czas końca ważności biletu,

$K_{C,S}$ – klucz sesji.

Poświadczenie

- ▶ Służy do uzyskania dostępu przez klienta do usługi serwera.
- ▶ Może być użyte tylko raz.
- ▶ Klient może wygenerować tyle poświadczeń, ile potrzebuje.

$$A_{C,S} = K_{C,S}(C \parallel TS),$$

gdzie:

$K_{C,S}$ – szyfrowanie kluczem sesji,

C – identyfikator klienta,

TS – znacznik czasu.

Wymiana z serwerem uwierzytelniającym

- Komunikat 1. – klient C prosi serwer uwierzytelniający AS o bilet do usługi przyznawania biletów

$$C \parallel TGS \parallel N_1,$$

gdzie:

C – identyfikator klienta,

TGS – identyfikator serwera przyznającego bilety,

N_1 – identyfikator jednorazowy.

- Komunikat 2. – AS zwraca bilet do usługi przyznawania biletów

$$K_C(K_{C,TGS} \parallel TGS \parallel N_1) \parallel T_{C,TGS}.$$

Otrzymanie biletu do serwera usługi

- Komunikat 3. – klient C prosi serwer TGS o bilet do serwera S

$$A_{C,TGS} \parallel T_{C,TGS} \parallel S \parallel N_2,$$

gdzie:

S – identyfikator serwera,

N_2 – identyfikator jednorazowy.

- Komunikat 4. – TGS, po poprawnej weryfikacji, zwraca żądany bilet

$$K_{C,TGS}(K_{C,S} \parallel S \parallel N_2) \parallel T_{C,S}.$$

Żądanie dostępu do usługi

- Komunikat 5. – klient C uwierzytelnia się wobec serwera S

$$A_{C,S} \parallel T_{C,S}.$$

- Komunikat 6. – serwer opcjonalnie uwierzytelnia się wobec klienta

$$K_{C,S}(TS + 1),$$

gdzie

TS – znacznik czasowy z poświadczenia wysłanego przez klienta.

Uwagi

- ▶ Powyższy opis jest uproszczony.
- ▶ Komunikaty zawierają dodatkowe pola z opcjami.
- ▶ Istnieją dwie wersje: 4 i 5, różniące się formatami komunikatów.
- ▶ Dla poprawnego działania konieczna jest synchronizacja zegarów z dokładnością do kilku minut.

Królestwo

- ▶ Pełne środowisko składające się z serwera uwierzytelniającego i wielu klientów.
- ▶ Serwer uwierzytelniający posiada identyfikatory i zaszyfrowane hasła wszystkich użytkowników.
- ▶ Wszyscy użytkownicy są zarejestrowani w serwerze uwierzytelniającym.
- ▶ Serwer uwierzytelniający ma wspólny tajny klucz z każdym serwerem.
- ▶ Wszystkie serwery są zarejestrowane w serwerze uwierzytelniającym.

Uwierzytelnianie pomiędzy królestwami

- ▶ Serwery uwierzytelniające w każdym królestwie muszą dzielić tajne klucze z serwerami uwierzytelniającymi w innych królestwach.
- ▶ Serwery uwierzytelniające muszą być nawzajem zarejestrowane.
- ▶ Użytkownicy jednego królestwa mogą uzyskiwać dostęp do serwerów z innego królestwa.
- ▶ Klient, uzyskawszy bilet do lokalnej usługi przyznawania biletów, prosi o przyznanie biletu do usługi przyznawania biletów w innym królestwie.

Różnice między wersjami

- ▶ Wersja 4 stosuje DES. W wersji 5 szyfrogram jest oznaczany identyfikatorem algorytmu szyfrowania.
- ▶ W wersji 5 klucze szyfrujące mają oznaczenie typu i pole długość.
- ▶ Wersja 4 wymaga stosowania adresów IP. W wersji 5 adresy sieciowe mają oznaczenie typu i pole długość, co pozwala na stosowanie dowolnych adresów sieciowych.
- ▶ W wersji 4 nadawca komunikatu decyduje o kolejności bajtów. W wersji 5 wszystkie komunikaty zdefiniowano za pomocą ASN.1 i BER.
- ▶ W wersji 4 czas ważności biletu jest kodowany na 8 bitach z kwantem 5 minut, co ogranicza maksymalny czas ważności do 1280 minut. W wersji 5 koduje się dokładny czas początku i końca ważności biletu.
- ▶ W wersji 5 możliwe jest przekazywanie uprawnień przyznanych dla jednego klienta innemu klientowi z innego komputera.

Różnice między wersjami, cd.

- ▶ W wersji 4 bilety są szyfrowane podwójnie. W wersji 5 zrezygnowano z tego.
- ▶ Wersja 4 korzysta z niestandardowego trybu PCBC pracy algorytmu DES. Tryb PCBC miał zapewniać kontrolę nienaruszalności danych. Wersja 5 stosuje standardowy tryb CBC i osobny mechanizm kontroli nienaruszalności.
- ▶ W wersji 4 klucz sesji z biletu może być używany wielokrotnie w kolejnych połączeniach z serwerem, co może narazić komunikację na atak powtórzeniowy. W wersji 5 klient i serwer mogą negocjować klucz podsesji, który będzie używany tylko w jednym połączeniu.
- ▶ W wersji 5 przy uwierzytelnianiu serwera wobec klienta znacznik czasu nie jest zwiększany – stosowany jest bardziej skomplikowany format komunikatu, uniemożliwiający jego podrobienie.

Znaczniki biletów (1)

- ▶ W wersji 5 wprowadzono znaczniki w biletach, znacznie rozszerzające funkcjonalność protokołu.
- ▶ INITIAL
 - ▶ Bilet został wydany bezpośrednio przez AS, a nie przez TGS na podstawie biletu na przyznanie biletu.
- ▶ PRE-AUTHENT
 - ▶ Zastosowano wstępne uwierzytelnianie, która ma utrudniać ataki na hasło.
- ▶ HW-AUTHENT
 - ▶ Wstępne uwierzytelnianie wymagało zastosowania sprzętu, który miał posiadać wyłącznie dany użytkownik.

Znaczniki biletów (2)

► RENEWABLE

- Bilet odnawialny – może zostać użyty do uzyskania zastępczego biletu, którego okres kończy się później.
- Długie okresy ważności biletów powodują większe ryzyko przy ich kradzieży.
- Krótkie okresy ważności biletów zwiększają obciążenie systemu i powodują częstsze prośby do użytkownika o podanie hasła.
- Kompromisem są bilety odnawialne z dwoma czasami ważności: jeden dla tego biletu, drugi będący najpóźniejszym czasem zakończenia ważności.
- Klient może odnowić bilet.
- TGS może odmówić odnowienia biletu.

► MAY-POSTDATE, POSTDATED, INVALID

- Klient może uzyskać wiele biletów na sesję, z odpowiednio rozłożonymi wartościami czasu.
- Wszystkie bilety oprócz pierwszego są od początku nieważne.
- Gdy jest wymagany nowy bilet, odpowiedni bilet może zostać uaktualniony, bez konieczności przedstawiania biletu na przyznanie biletu.

Znaczniki biletów (3)

- ▶ PROXIABLE, PROXY

- ▶ Umożliwia serwerowi działanie jako pełnomocnik w imieniu klienta.

- ▶ FORWARDABLE, FORWARDED

- ▶ Umożliwia przekazywanie biletów między królestwami.
- ▶ W wersji 4 współpraca n królestw wymaga wzajemnego uwierzytelniania każdego z każdym, czyli $n(n - 1)/2$ tajnych kluczy.
- ▶ W wersji 5 królestwa mogą tworzyć strukturę drzewiastą.
- ▶ Klient może poruszać się w górę drzewa do wspólnego wierzchołka, a następnie w dół do królestwa docelowego.

Problemy

- ▶ Serwer uwierzytelniający stanowi pojedynczy punkt awarii (ang. single point of failure). Można zastosować rozwiązanie z serwerem awaryjnym.
- ▶ Atak na centralny serwer kompromituje cały system.
- ▶ Domyślna konfiguracji Kerberos wymaga synchronizacji zegarów z dokładnością 5 minut. W praktyce wystarczy zastosowanie protokołu synchronizacji czasu (ang. Network Time Protocol).
- ▶ Administracja serwerem nie jest ustandaryzowana.

Rozszerzenia

- ▶ Użycie kryptografii z kluczem publicznym