System Kerberos

Patryk Czarnik

Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski

Bezpieczeństwo sieci komputerowych – MSUI 2010/11

Użytkownicy i usługi

Usługa

- Funkcjonalność, z której można (chcieć) korzystać, np.:
  - system plików
  - drukarka
  - aplikacja
- Dostępna lokalnie lub poprzez sieć

Użytkownik

- Osoba fizyczna (korzystająca z terminala)
- Inna aplikacja

Użytkownicy i usługi

Usługa

- Funkcjonalność, z której można (chcieć) korzystać, np.:
  - system plików
  - drukarka
  - aplikacja
- Dostępna lokalnie lub poprzez sieć

Użytkownik

- Osoba fizyczna (korzystająca z terminala)
- Inna aplikacja

Dostęp do usług w sieci

Wymagania bezpieczeństwa

- identyfikacja klienta
- uwierzytelnienie klienta
- kontrola dostępu
- uwierzytelnienie serwera

Architektura bezpieczeństwa

1. Dwie strony:
   - wzajemne uwierzytelnianie
   - kryptografia asymetryczna lub wcześniejsza wymiana kluczy
2. Trzecia strona:
   - zdejmuje z serwera usługi obsługę bezpieczeństwa
   - ujednolicona obsługa dla wszystkich usług

Dostęp do usług w sieci

Wymagania bezpieczeństwa

- identyfikacja klienta
- uwierzytelnienie klienta
- kontrola dostępu
- uwierzytelnienie serwera

Architektura bezpieczeństwa

1. Dwie strony:
   - wzajemne uwierzytelnianie
   - kryptografia asymetryczna lub wcześniejsza wymiana kluczy
2. Trzecia strona:
   - zdejmuje z serwera usługi obsługę bezpieczeństwa
   - ujednolicona obsługa dla wszystkich usług

Idea „*Single Sign On*”

Kiedy

- Wiele różnych usług
- Ale o powiązanych politykach dostępu
- Ten sam użytkownik ma uprawnienia do wielu usług

Rozwiązanie

- Niech użytkownik loguje się tylko jeden raz
- Możliwe realizacje:
  - zapamiętanie hasła/klucza przez aplikację klienta
  - systemy oparte o serwer uwierzytelnienia i przepustki, np. Kerberos

Zalety

- Wygoda
- Ujednolicona architektura i polityka bezpieczeństwa

Idea „*Single Sign On*”

Kiedy

- Wiele różnych usług
- Ale o powiązanych politykach dostępu
- Ten sam użytkownik ma uprawnienia do wielu usług

Rozwiązanie

- Niech użytkownik loguje się tylko jeden raz
- Możliwe realizacje:
  - zapamiętanie hasła/klucza przez aplikację klienta
  - systemy oparte o serwer uwierzytelnienia i przepustki, np. Kerberos

Zalety

- Wygoda
- Ujednolicona architektura i polityka bezpieczeństwa

Idea „*Single Sign On*”

Kiedy

- Wiele różnych usług
- Ale o powiązanych politykach dostępu
- Ten sam użytkownik ma uprawnienia do wielu usług

Rozwiązanie

- Niech użytkownik loguje się tylko jeden raz
- Możliwe realizacje:
  - zapamiętanie hasła/klucza przez aplikację klienta
  - systemy oparte o serwer uwierzytelnienia i przepustki, np. Kerberos

Zalety

- Wygoda
- Ujednolicona architektura i polityka bezpieczeństwa

Kerberos – historia

- Narodziny w MIT w ramach projektu Athena, lata 80-te XX wieku
- Wersje robocze do nr 3
- Wersja 4, koniec lat 80-tych – stosowana do dziś
- Wersja 5 (1993, RFC 1510) – rozszerzenie funkcjonalności i „uszczelnienie” bezpieczeństwa (szczegóły dalej)
- W 2005 roku RFC 4120 – głównie poprawki redakcyjne
- Cerber – mityczny trzygłowy pies strzegący Hadesu

Źródło obrazka: Wikipedia

Projekt Athena

- MIT, DEC, IBM, lata 1983–1991
- Rozproszone środowisko różniących się maszyn, umożliwiające swoboną pracę z dowolnego terminala
- Wynalazki:
  - Kerberos
  - duży udział w rozwoju X Windows
  - pierwszy komunikator internetowy Zephyr
  - system usług katalogowych Hesiod
- Różne odmiany stosowane do dziś w sieciach uczelnianych

Kerberos – założenia

- Kryptografia symetryczna
- Hasło nigdy nie przesyłane otwartym tekstem
- Hasło (nawet zaszyfrowane) nie przechowywane u klienta dłużej niż na potrzeby uwierzytenienia
- Hasło nigdzie nie przechowywane niezaszyfrowane
- Hasło wystarczy podać raz na sesję, nawet na potrzeby wielu usług (*Single Sign On*)
- Wyznaczony serwer uwierzytelnienia
- Serwery usług bez informacji związanych z uwierzytelnianiem
- Uwierzytelnienie w obie strony
- Wsparcie dla ustalania klucza sesji

Jak działa Kerberos – ogólnie

- W systemie mamy do czynienia ze:
  - stronami (użytkownicy, usługi itp.),
  - przepustkami,
  - centrum dystrybucji kluczy (**KDC**).
- Strony posiadając ważne przepustki są w stanie:
  - identyfikować się,
  - bezpiecznie się komunikować.
- Uwierzytelnianie na podstawie przepustek można dołożyć do dowolnej aplikacji.

Gdzie się używa Kerberosa?

- Protokoły sieciowe – różne poziomy abstrakcji
  - najczęściej przy korzystaniu z protokołów aplikacji, np. w TELNET, FTP (warstwa 7. ISO/OSI);
  - czasami dodaje się ją do zapewnienia bezpieczeństwa w warstwie sesji, np. w SSH, RPC, CORBA (warstwa 5. ISO/OSI);
  - można tego użyć też na niższych poziomach, np. w IP, UDP, TCP (warstwy 4. i 3. ISO/OSI).
- Systemy Windows od Windows 2000 – domyślny system uwierzytelnienia w sieci.
- Mac OS X, Linux i BSD – implementacje dostępne.

Wybrane implementacje Kerberosa

- MIT Kerberos – najpowszechniejsza niekomercyjna wersja.
- Heimdal – otwarta implementacja oparta o eBones, napisaną w Szwecji w czasie gdy USA zabraniały eksportu MIT-Kerberosa.
- ShiShi – inna otwarta implementacja.
- Windows 2000 Kerberos, kontynuowana w kolejnych Windowsach.
- Implementacja Suna dla Javy (w ramach Java Generic Security Service API i Java Authentication and Authorization Service).

Schemat działania Kerberosa – komunikacja

- W protokole biorą udział 3 węzły komunikacyjne:
  - A – klient jakiejś usługi,
  - B – serwer tejże usługi,
  - S – centrum dystrybucji kluczy.
- Protokół oparty na schemacie Needhama-Schrödera.

Schemat Needhama-Schrödera

	Komunikacja	Komunikat
1.	$A \to S$	A, B, N_A
2.	$S \to A$	$\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_B}\}_{K_A}$
3.	$A \to B$	$\{K_{AB}, A\}_{K_B}$
4.	$B \to A$	$\{N_B\}_{K_{AB}}$
5.	$A \to B$	$\{N_B - 1\}_{K_{AB}}$

Oznaczenia:

- A – nazwa klienta,
- B – nazwa serwera,
- K_A – tajny klucz klienta,
- K_B – tajny klucz serwera,
- K_{AB} – tajny klucz sesji między A i B,
- N_A – identyfikator jednorazowy A,
- $\{M\}_K$ – komunikat M zaszyfrowany kluczem K.

Wady scentralizowanego rozwiązania

Słabości

- Klient za każdym razem musi prosić o nową przepustkę (w praktyce wiąże się to z wpisywaniem hasła).
- Z kolei przepustki wielorazowe mogą stać się ofiarą ataku powtórzeniowego.
- Przepustka z czasem ważności — problem z dobraniem bezpiecznego ale wygodnego czasu.

Rozwiązanie

- Dodatkowo rozdzielamy funkcje uwierzytelnienia i wydawania przepustek – mamy 2 serwery: *serwer uwierzytelnienia* (KDC) i *serwer przepustek* (TGS).

Wady scentralizowanego rozwiązania

Słabości

- Klient za każdym razem musi prosić o nową przepustkę (w praktyce wiąże się to z wpisywaniem hasła).
- Z kolei przepustki wielorazowe mogą stać się ofiarą ataku powtórzeniowego.
- Przepustka z czasem ważności — problem z dobraniem bezpiecznego ale wygodnego czasu.

Rozwiązanie

- Dodatkowo rozdzielamy funkcje uwierzytelnienia i wydawania przepustek – mamy 2 serwery: *serwer uwierzytelnienia* (KDC) i *serwer przepustek* (TGS).

Zmodyfikowany schemat Needhama-Schrödera

Kerberos v4

	Komunikacja	**Komunikat**
1.	$A \rightarrow KDC$	A, TGS, N
2.	$KDC \rightarrow A$	$\{K_{A,tgs}, N, \{T(A, TGS)\}_{K_{tgs}}\}_{K_A}$
3.	$A \rightarrow TGS$	$\{V(A)\}_{K_{A,tgs}}, \{T(A, TGS)\}_{K_{tgs}}, B, N$
4.	$TGS \rightarrow A$	$\{K_{A,B}, N, \{T(A, B)\}_{K_B}\}_{K_{A,tgs}}$
5.	$A \rightarrow B$	$\{V(A)\}_{K_{A,B}}, \{T(A, B)\}_{K_B}$, zamówienie, N
6.	$B \rightarrow A$	$\{N\}_{K_{A,B}}$

Poświadczenie: $V(X) = X, t$

Przepustka: $T(X, Y) = X, Y, t_1, t_2, K_{XY}$

Zmodyfikowany schemat Needhama-Schrödera

Kerberos v5

	Komunikacja	**Komunikat**
1.	$A \rightarrow KDC$	A, TGS, N
2.	$KDC \rightarrow A$	$\{K_{A,tgs}, N\}_{K_A}, \{T(A, TGS)\}_{K_{tgs}}$
3.	$A \rightarrow TGS$	$\{V(A)\}_{K_{A,tgs}}, \{T(A, TGS)\}_{K_{tgs}}, B, N$
4.	$TGS \rightarrow A$	$\{K_{A,B}, N\}_{K_{A,tgs}}, \{T(A, B)\}_{K_B}$
5.	$A \rightarrow B$	$\{V(A)\}_{K_{A,B}}, \{T(A, B)\}_{K_B}$, zamówienie, N
6.	$B \rightarrow A$	$\{N\}_{K_{A,B}}$

Poświadczenie: $V(X) = X, t$

Przepustka: $T(X, Y) = X, Y, t_1, t_2, K_{XY}$

Niektóre różnice między wersjami 4 a 5

- Dokładniejsze określenie czasów (w wersji 4 z dokładnością do 5 minut, max 21 godzin).
- Przepustki krótko ważne, ale z możliwością przedłużenia ważności po sprawdzeniu w TGS.
- Opcje (flagi) w przepustkach.
- Wybór algorytmu szyfrowania (w v4 pojedynczy DES) i protokołu sieci.
- Zmiana niebezpiecznego trybu DES PCBC na CBC.
- Kolejność bajtów zgodnie ze standardami (w v4 konfigurowalna).
- Domeny i przekazywanie uwierzytelnienia.
- Optymalizacja obliczeniowa (w v4 podwójne szyfrowanie niektórych pól).
- Zalecenie by klucz sesji na potrzeby realizacji usługi był renegocjowany nawet gdy korzystamy z tego samego biletu (ryzyko ataku potwórzeniowego).

Identyfikacja stron w Kerberosie

- Identyfikator składa się z trzech elementów:
  - podstawa (ang. *primary*),
  - instancja (ang. *instance*),
  - domena (ang. *realm*).

np. `czarnik/praca@mimuw.edu.pl`

Trochę o domenach Kerberosa

- W dużej sieci komputery KDC i TGS mogą stać się wąskim gardłem.
- Dlatego wprowadzono możliwość rozbicia jednego serwisu Kerberosa na wiele (domen) – każda domena ma własne KDC i TGS.
- Konieczny jest jednak mechanizm delegacji odpowiedzialności – jedna domena (TGS) rejestruje się w drugiej (odległej) domenie (RTGS).
- Zwiększa się liczba wymian: $A \rightarrow TGS \rightarrow RTGS \rightarrow B$.
- Czasami konieczne jest wprowadzenie wielu pośredników.
- Nazwa każdego z nich pojawia się w przepustce.
- Konieczne jest wprowadzenie hierarchii.

Opcje (flagi) przepustek

- **Przepustki początkowe** – wydane bezpośrednio przez KDC serwera, do którego się odnosimy.
- **Przepustki wstępnie uwierzytelnione** – wydane na podstawie informacji z jakiegoś wstępnego mechanizmu uwierzytelnienia, np. za pomocą karty chipowej. Ta własność jest dziedziczona między różnymi TGS.
- **Przepustki nieważne** – stosowane do wykonywania operacji wsadowych; można zażyczyć sobie wydanie przepustki ważnej w przyszłości, centrum usunie bit nieważności w odpowiednim momencie.
- **Przepustki przesunięte w czasie** – j.w.

Opcje przepustek c.d.

- **Przepustki odnawialne** – bezpieczniej jest mieć wiele przepustek na krótkie okresy niż jedną na długi okres czasu. Takie przepustki mają 2 czasy: kiedy przepustka się unieważnia i o ile można ją przedłużyć poza ten czas (przykład).
- **Przepustki pośrednikowe** – czasami chcemy, aby jakiś serwis w naszym imieniu wykonywał jakieś operacje. Możemy scedować nasze prawa do takiego serwisu za pomocą przepustki powyższego rodzaju. W przepustkach tego typu podana jest jawna lista serwisów, które mogą z nich korzystać. Przepustki takie nie mogą służyć do wydawania przepustek.
- **Przepustki do przekazywania** – j.w., ale mogą służyć do wydawania przepustek.

Baza danych Kerberosa

- **name** – identyfikator strony;
- **key** – klucz tajny strony;
- **p_kvno** – numer wersji klucza strony;
- **max_life** – maksymalny czas życia przepustki;
- **max_renewable_life** – maksymalny całkowity czas życia dla przepustek odnawialnych.

Dodatkowe pola bazy (Athena)

- **K_kvno** – numer wersji klucza Kerberosa,
- **expiration** – moment unieważnienia pozycji,
- **attributes** – bitowy wektor atrybutów,
- **mod_date** – moment ostatniej modyfikacji,
- **mod_name** – identyfikator strony, która ostatnio modyfikowała.

Stałe w Kerberosie

- Minimalny czas życia przepustki – powinien uwzględniać spotykane czasy podróży w dwie strony, czasy szyfrowania i deszyfrowania oraz czas przetwarzania komunikatu.
- Maksymalny dopuszczalny czas pojedynczego odnowienia przepustki.
- Maksymalny dopuszczalny czas życia przepustki.
- Informację, czy dopuszczalne jest wydawanie przepustek z pustym polem adresu.
- Czy mogą być wydawane przepustki pośrednikowe, do przekazywania, odnawialne oraz na inne daty.