



Uwierzytelnianie w sieci na potrzeby dostępu do usług

Protokół Kerberos

Krzysztof Boryczko

Kerberos – definicja i cechy

- Kerberos – protokół uwierzytelniania sieciowego.
- Model klient / serwer.
- Klient (np. użytkownik) zostaje w bezpieczny sposób uwierzytelniony na potrzeby dostępu do danej aplikacji.
- Możliwość zastosowania w modelu *Single sign-on (SSO)* czyli jednokrotne uwierzytelnienie daje dostęp do wielu usług.
- Uwierzytelnianie w sposób szyfrowany – wykorzystanie kryptografii symetrycznej.
- Nie ma przesyłania w sieci hasła czy innych tajnych informacji – wykorzystuje się bilety i klucze.
- Niezależny od platformy i środowiska
(choć istnieje kilka implementacji: MIT, Heimdal, Microsoft).

Bezpieczeństwo – model AAA

- Authentication: *uwierzytelnienie*
 - Potwierdzenie tożsamości użytkownika – przykładowe protokoły to: Kerberos, NTLM, NIS , LDAP, HTTPS, itd.
- Authorization: *autoryzacja*
 - Ustalenie uprawnień dostępu do zasobów (danych, usług, aplikacji, urządzeń, itp.) – SMB, NIS, LDAP, itd.
- Accounting: *rozliczanie*
 - Zbieranie informacji o wykorzystaniu przez użytkowników zasobów (systemu, procesora, sieci, aplikacji, itp.) – protokoły i narzędzia związane z dziennikowaniem (logowaniem) informacji.

Uwierzytelnienie

- Proces uwierzytelnienia użytkownika może przebiegać na podstawie:
 - Tego co użytkownik wie – hasło, PIN, itp.
 - Tego co użytkownik posiada – token programowy, karta, telefon, klucz sprzętowy, itp.
 - Tego co jest jego częścią (cechą) użytkownika – podpis odręczny, dane biometryczne – odcisk palca, tęczówka oka, głos, itp.
- Silne uwierzytelnienie musi wykorzystywać przynajmniej dwa czynniki z dwóch różnych grup.

Nauki związane z szyfrowaniem

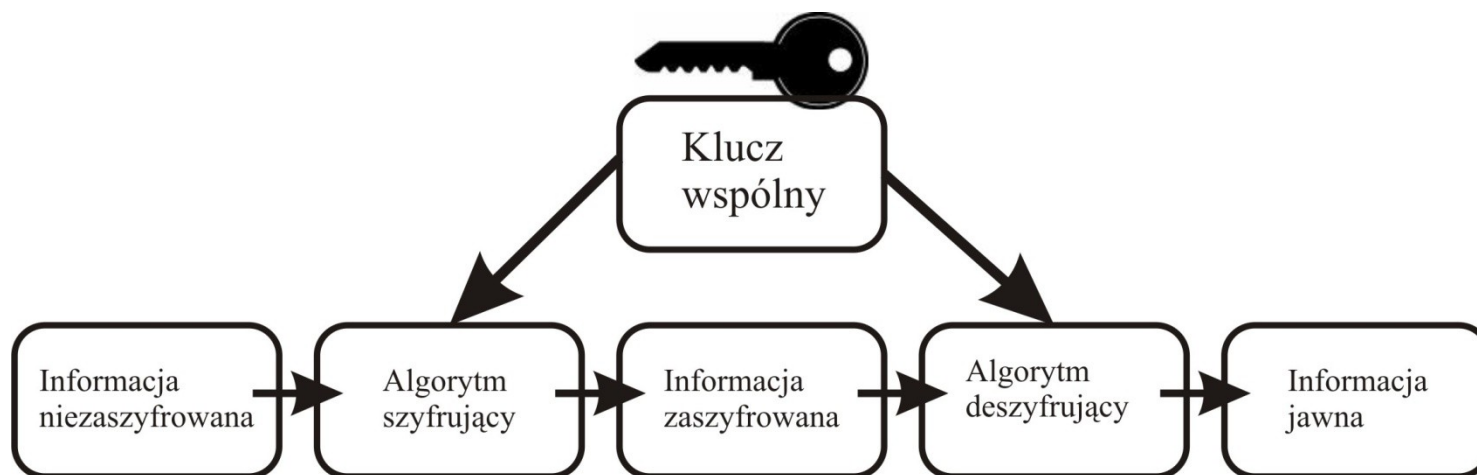
- Dziedzina nauki zajmująca się szyfrowaniem to *kryptologia*.
- *Kryptologia* się na:
 - *Kryptografię* – nauka o tworzeniu tzw. szyfrów systemów kryptograficznych tj. algorytmów szyfrowania i deszyfrowania.
 - *Kryptoanalizę* – nauka o łamaniu szyfrów systemów kryptograficznych.

Pojęcia związane z szyfrowaniem

- *Szyfrowanie* – procedura przekształcania informacji nieszyfrowanej (jawnej) w informację zaszyfrowaną (tajną) za pomocą odpowiedniego klucza.
- *Deszyfrowanie* – procedura przekształcania informacji zaszyfrowanej w jawną z wykorzystaniem odpowiedniego klucza.
- *Klucz* – ciąg znaków o określonej długości, który umożliwia wykonywanie czynności kryptograficznych takich jak szyfrowanie lub deszyfrowanie.
- *Szyfrogram* – zaszyfrowana informacja, która nie jest możliwa do odczytania bez odpowiedniego klucza deszyfrującego.

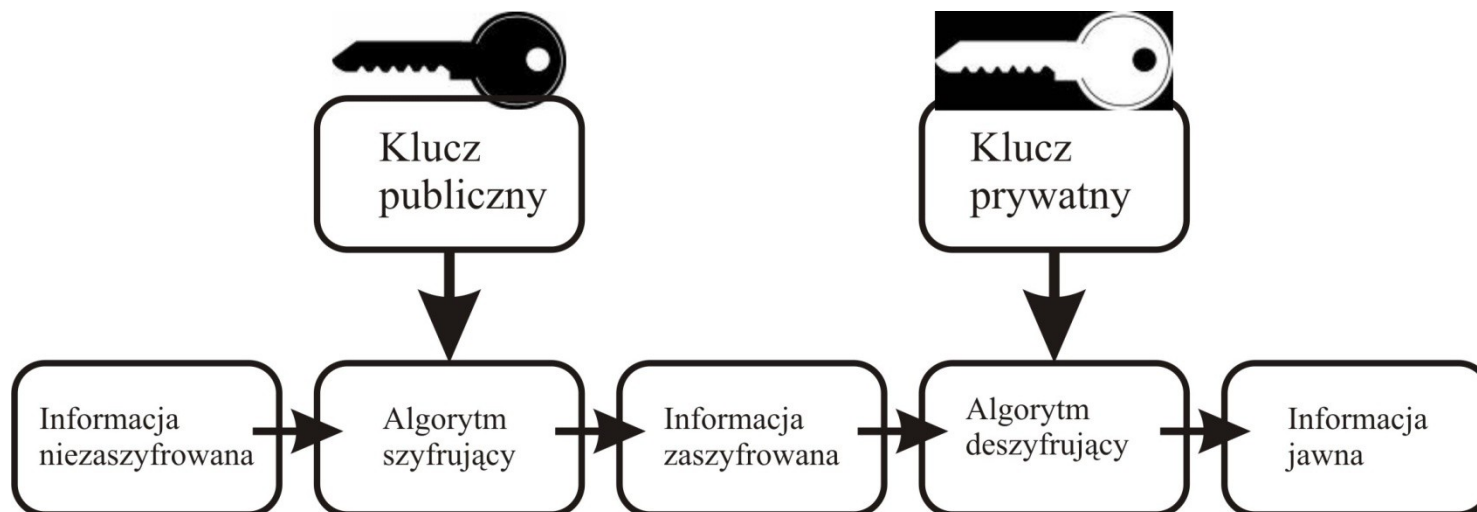
Szyfrowanie symetryczne

- Ten sam klucz jest wykorzystywany do szyfrowania i deszyfrowania informacji.
- Klucz jest zazwyczaj przypisywany do danego kanału informacyjnego, a nie do posiadacza.
- Przykłady: DES, 3DES, AES, RC4, IDEA, Blowfish



Szyfrowanie asymetryczne

- Klucz publiczny (jawny) dostępny w usłudze dla wszystkich, upubliczniony.
- Klucz prywatny (tajny) unikatowy, znany jedynie właścicielowi, chroniony.



Funkcja skrótu (hash)

- Podstawowe cechy funkcji skrótu:
 - Skrót daje się utworzyć łatwo, natomiast odwrócenie operacji ma być niemożliwe (funkcja jednokierunkowa).
 - Wynik generowany jest na podstawie całego wejścia.
 - Zmiana jednego bajtu (znaku) – całkiem inny wynik.
 - Niezależnie od długości wejścia wynik ma tę samą długość dla danego algorytmu.
- Przykłady algorytmów:
DES, MD2, MD4, MD5, SHA1, SHA256, SHA512, itp.

Funkcja skrótu – zastosowanie

- Przykładowe zastosowania funkcji skrótu:
 - Przechowywanie haseł w różnych systemach.
Przykładowo: Unix/Linux – DES, MD5, SHA256, SHA512.
 - Wyliczanie sum kontrolnych dla plików czy wiadomości (zapewnienie integralności danych).
 - Tworzenie skrótów z wiadomości dla potrzeb podpisu elektronicznego.
- Wykorzystanie „solenia”
 - Ustala się dane (klucz czy dane losowe) i miesza się je z danymi, z których będzie skrót.
 - Generowany jest skrót z „posolonych” danych.

Kerberos - historia

- Historia protokołu *kerberos*
 - Stworzony w MIT w trakcie realizacji projektu Athena, którego realizacja rozpoczęła się w 1983 r.
 - Udostępniony publicznie od wersji IV – 1989 r.
 - Wersja V – ukazała się w 1993 r. i obowiązuje do dzisiaj.
 - W 1997 r. Microsoft ogłasza wykorzystanie protokołu *kerberos* do uwierzytelniania, co wprowadza od wersji Windows 2000 (wstępnie alternatywnie do protokołu NTLM, a później jako „jedyne”).

Kerberos - dokumenty

- Dokumenty techniczne dotyczące *kerberosa*
 - RFC 1510 – opis protokołu w wersji V (przestarzały).
 - RFC 4120 – poprawiony i aktualny opis protokołu w wersji V.
 - RFC 3244 – zmiana i ustawianie haseł – rozszerzenie Microsoftu.
 - RFC 4757 – wykorzystanie algorytmu *RC4*,
 - RFC 3962 – wykorzystanie algorytmu *AES*,
 - RFC 3961, RFC 4121, ...

Kerberos – idea działania

- Uwierzytelnianie oparte jest o mechanizm przydzielania biletów.
- Bilet ma określony czas życia.
- W procesie uwierzytelniania użytkownika dla dostępu do usługi uczestniczy podmiot pośredniczący *Trusted Third Party (TTP)*.
- Użytkownik nie uwierzytelnia się bezpośrednio w serwerze usługi lecz za pomocą *TTP*.
- Użytkownik nie przesyła żadnych poufnych informacji jak np. hasło – wykorzystywany jest mechanizm kluczy symetrycznych.

Kerberos – rola haseł

- Hasła (ich skróty) służą jako klucze szyfrujące przesyłane wiadomości.
- Klucze użytkowników (hasła) przechowywane są w bazie *kerberosa*.
- Serwer usługi, tak jak użytkownik – ma swój klucz w bazie *kerberosa*.
- Szyfrowanie jest symetryczne – *kerberos* ma klucze w swej bazie, a użytkownik czy usługa zna ten sam klucz.
- Klucze sesji mają znacznik czasowy.

Kerberos – elementy systemu

- CL – Client

Klient który chce otrzymać dostęp do usługi.

- SS – Service Server

Serwer świadczący żadaną usługę.

- AS – Authentication Server

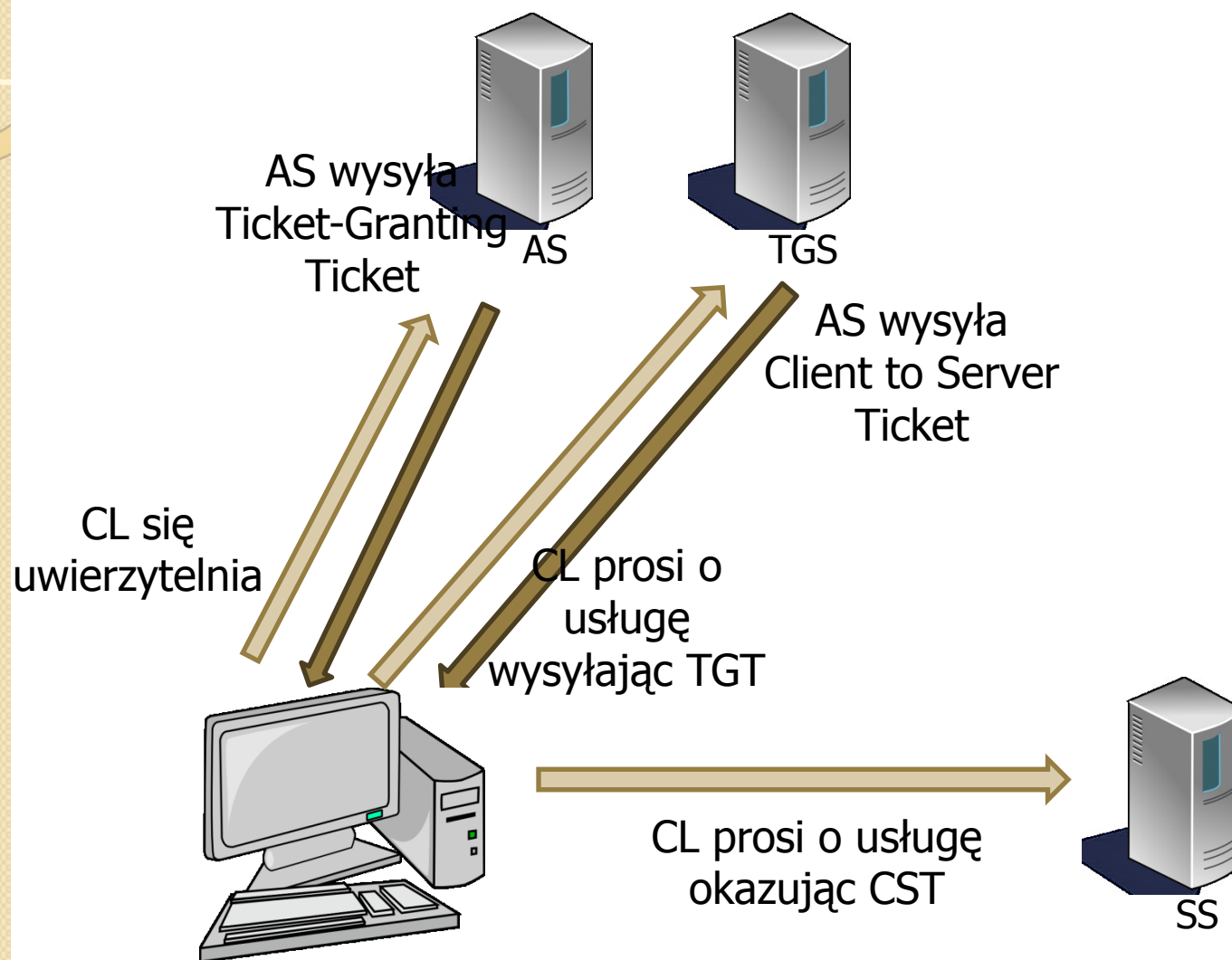
Serwer dokonujący uwierzytelnienia klienta.

- TGS – Ticket-Granting Server

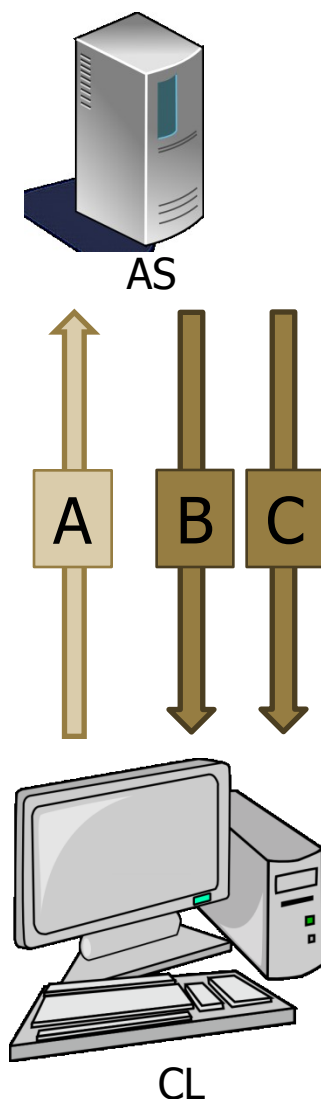
Serwer zajmujący się dystrybucją kluczy

Trusted Third Party
(w praktyce jeden system – KDC)

Kerberos – ogólny schemat

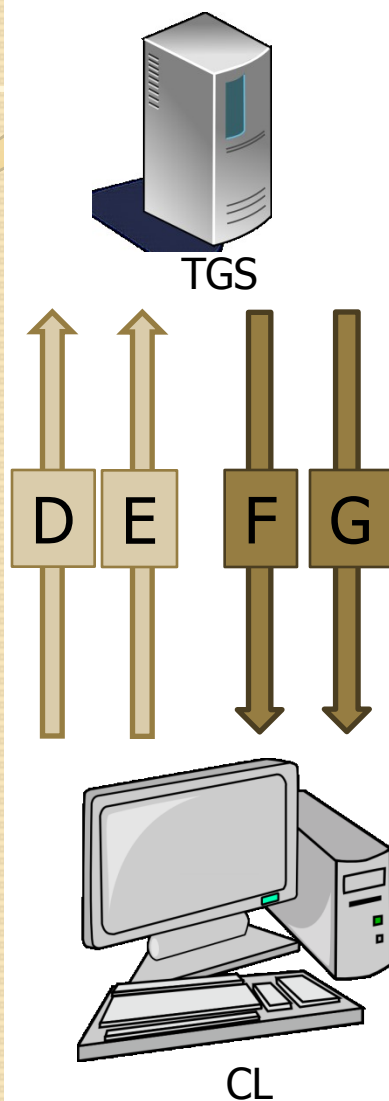


Uwierzytelnienie klienta



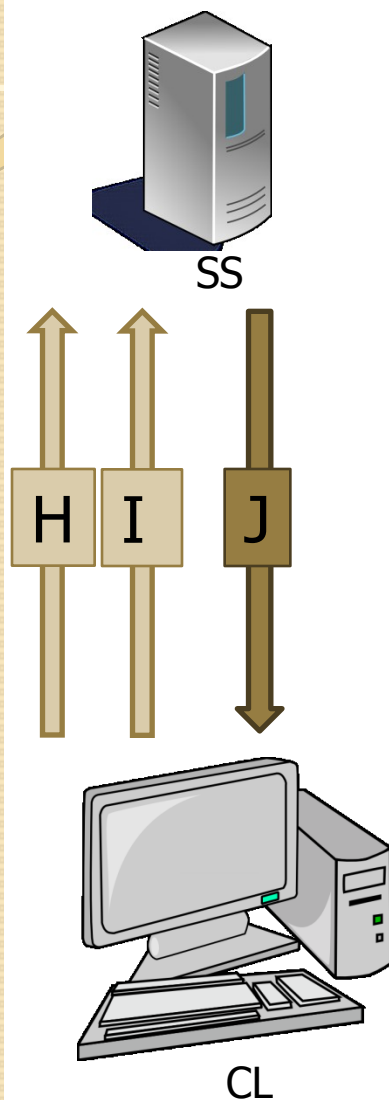
1. Użytkownik się przedstawia (podaje login).
2. CL wysyła otwartym tekstem wiadomość „A” z nazwą użytkownika i prośbą o dostęp do usługi.
3. AS sprawdza czy użytkownik jest w bazie i wysyła wiad. „B” z kluczem sesji CL/TGS, zaszyfrowaną kluczem (hasłem) użytkownika.
4. AS wysyła wiad. „C” z TGT, w którym jest ID klienta, jego adres, okres ważności i klucz sesji CL/TGS zaszyfrowany kluczem TGS.
5. Użytkownik rozszyfrowuje wiad. „B” za pomocą swego hasła, które wprowadza przy logowaniu.
6. Użytkownik ma klucz sesji CL/TGS który wykorzystuje do dalszej komunikacji – SSO.
7. Użytkownik **nie** może odszyfrować wiad. „C”, ponieważ zaszyfrowana jest kluczem TGS.
8. Klient może potwierdzić swą tożsamość dla TGS.

Nadanie dostępu do usługi



1. CL wysyła wiad. „D” z TGT otrzymanym w wiad. „C” (zaszyfrowanym kluczem TGS) i ID usługi
2. CL wysyła wiad. „E” ze swoim identyfikatorem i znacznikiem czasowym zaszyfrowaną kluczem sesji CL/TGS.
3. TGS rozszyfrowuje wiad. „D” i otrzymuje TGT, w którym zawarty jest klucz sesji CL/TGS.
4. TGS mając klucz sesji CL/TGS rozszyfrowuje wiadomość „E”.
5. TGS wysyła wiad. „F” z kluczem sesji CL/SS zaszyfrowaną kluczem sesji CL/TGS.
6. TGS wysyła wiad. „G” z CST, w którym jest ID klienta, jego adres, ważność klucza i klucz sesji CL/SS zaszyfrowany kluczem SS.
7. Użytkownik **nie** może odszyfrować wiad. „G”, ponieważ zaszyfrowana jest kluczem SS.
8. Klient może potwierdzić swą tożsamość dla SS.

Żądanie usługi od serwera



1. CL wysyła wiad. „H” z CST otrzymanym w wiad. „G” (zaszyfrowanym kluczem SS) i ID usługi
2. CL posiadając klucz sesji CL/TGS rozszyfrowuje wiadomość „F” i otrzymuje klucz sesji CL/SS.
3. CL wysyła wiad. „I”, w której jest identyfikator klienta i znacznik czasowy zaszyfrowaną kluczem sesji CL/SS.
4. SS używając własnego klucza rozszyfrowuje wiadomość „H” i otrzymuje klucz sesji CL/SS.
5. SS mając klucz sesji CL/SS rozszyfrowuje wiadomość „I” otrzymując ID klienta.
6. SS pozwala na dostęp do swych usług wysyłając do klienta wiadomość „J” ze zwiększonym o 1 znacznikiem czasowym zaszyfrowaną CL/SS.
7. Klient mając klucz CL/SS odszyfrowuje wiadomość „J” i sprawdza znacznik czasowy.
8. Rozpoczyna korzystanie z SS.

Struktura nazw

- Użytkowników w bazie *KDC* określa się z języka angielskiego jako *principals*.
- W *kerberosie* nazwa użytkownika zdefiniowana jest jako *login@realm*
- *Relam* (królestwo) jest czymś w rodzaju nazwy domeny w usługach katalogowych.
- Pełna nazwa użytkownika pełni analogiczną rolę jak nazwa wyróżniająca w *LDAP*.
- Nie ma struktury drzewiastej.
- Nawa królestwa – dowolny ciąg znaków ASCII – zwyczajowo dużymi literami.
- Przyjęło się, że odpowiada ona domenie *DNS*.

Wymagania instalacyjne

- W praktyce *AS* oraz *TGS* to jeden system *Key Distribution Center (KDC)*
- Komputer powinien mieć statyczną konfigurację protokołu IP i nazwę FQDN.
- Konieczna jest synchronizacja czasu pomiędzy elementami systemu – *NTP*.
- Nazwa królestwa powinna być zgodna z nazwą domeny DNS oraz ze strukturą nazw *LDAP* przy wykorzystaniu go do autoryzacji.

Instalacja KDC – RH

- Instalacja pakietu *krb5-server* i związanych z nim bibliotek.
- Instalacja pakietu *krb5-workstation* zawierającego narzędzia klienckie.
- Pakiet *krb5-server* zawiera dwie usługi:
 - *krb5kdc* – Key Distribution Center,
 - *kadmind* – demon do zarządzanie zawartością bazy *kerberos*.
- Główny katalog z plikami to */var/kerberos/krb5kdc* znajdują się w nim:
 - *kdc.conf* – główny plik konfiguracyjny *KDC*,
 - *kadm5.acl* – plik zawierający definicję list kontroli dostępu
 - Pliki bazy z użytkownikami.

Konfiguracja KDC

- W pliku `/var/kerberos/krb5kdc/kdc.conf` w sekcji `[realms]` znajdują się definicje obsługiwanych królestw.
- Definicje są w postaci atrybutów i ich wartości poprzedzonych znakiem „=”
- Najważniejsze opcje wchodzące w skład definicji królestwa:
 - *nazwa królestwa* – zgoda z nazwą domeny i pisana dużymi literami,
 - *master_key_type* – rodzaj funkcji skrótu, którą zaszyfrowany jest główny klucz do bazy,
 - *supported_enctypes* – rodzaje algorytmów szyfrowania (skrót), które obsługuje KDC. Nimi mogą być szyfrowane klucze.
 - *acl_file* – ścieżka na plik z listami kontroli dostępu (`kadm5.acl`),
 - *admin_keytab* – ścieżka do pliku z kluczami do usługi *kadmin*,

Konfiguracja KDC c.d.

- Przykładowy plik konfiguracyjny *KDC* (*kdc.conf*):

```
[realms]
STUDENCI.AGH.EDU.PL = {
    #master_key_type = aes256-cts
    acl_file = /var/kerberos/krb5kdc/kadm5.acl
    dict_file = /usr/share/dict/words
    admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
    supported_enctypes = aes256-cts:normal aes128-cts:normal des3-
hmac-sha1:normal arcfour-hmac:normal des-hmac-sha1:normal des-
cbc-md5:normal des-cbc-crc:normal des-cbc-crc:v4 des-cbc-crc:afs3
}
```


Konfiguracja uprawnień

- Uprawnienia użytkowników do zarządzania atrybutami użytkowników w bazie są w postaci list kontroli dostępu.
- Zawarte są w pliku `/var/kerberos/krb5kdc/kadm5.acl`.
- Przykładowa postać pliku:

```
*/admin@ STUDENCI.AGH.EDU.PL      *  
pwch@ STUDENCI.AGH.EDU.PL c
```

- Każda linia zawiera:
 - nazwę użytkownika (lub wyrażenie regularne ją opisujące),
 - uprawnienia użytkownika – takie jak: zmiana haseł, dodawanie użytkowników, usuwanie użytkowników, itp. lub wszystkie uprawnienia, co oznaczone jest „*”.
- Nadane uprawnienia w królestwie:
 - Użytkownicy o nazwie kończącej się na „*admin*” – wszystkie,
 - Użytkownik *pwch* zmiana haseł użytkowników.

Baza użytkowników

- Wszyscy użytkownicy (*principals*) znajdują się w bazie.
- Użytkownicy to zarówno osoby jak i usługi.
- Baza zawiera tylko nazwy użytkowników i skróty z ich haseł, będące kluczami szyfrującymi podczas komunikacji.
- Baza zaszyfrowana jest hasłem, z którego skrót zwany jest *master key*.
 - Algorytm jakim jest wykonane skrót określony jest w pliku konf. KDC
 - Klucz jest przechowywany w pliku zwanym *stash file* znajdującym się w głównym katalogu *kerberos*
 - *Stash file* ma taką nazwę jak królestwo poprzedzoną kropką
- Bazę dla KDC dla królestwa STUDENCI.AGH.EDU.PL można utworzyć za pomocą *kdb5_util*

```
[root@dns1 ~]#kdb5_util create -r STUDENCI.AGH.EDU.PL -s
```

Zarządzanie użytkownikami

- Zarządzanie bazą użytkowników za pomocą programów:
 - *kadmin.local* – program uruchamiany lokalnie na *KDC* z uprawnieniami użytkownika *root*,
 - *kadmin* – umożliwia połączenie się do *KDC* z innego komputera użytkownikowi posiadającemu odpowiednie uprawnienia.
- Programy te działają albo w trybie interaktywnym albo wsadowym.
- Przykład wywołania programu *kadmin.local*:

```
[root@dns1 ~]#kadmin.local -r STUDENCI.AGH.EDU.PL
```

- Proste dodanie użytkownika w konsoli programu *kadmin*

```
kadmin.local: addprinc -pw haslo_użytkownika nazwa_użytkownika
```

Atrybuty użytkownika

- Przykład opisu użytkownika z programu *kadmin* – część opisująca podstawowe atrybuty użytkownika

```
kadmin.local: getprinc franio
Principal: franio@STUDENCI.AGH.EDU.PL
Expiration date: [never]
Last password change: Wed Mar 17 17:11:59 CET 2010
Password expiration date: Fri Apr 16 18:11:59 CEST 2010
Maximum ticket life: 1 day 00:00:00
Maximum renewable life: 0 days 00:00:00
Last modified: Wed Mar 17 18:44:25 CET 2010
(root/admin@STUDENCI.AGH.EDU.PL)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
```

Kodowanie hasła użytkownika

- Ciąg dalszy opisu użytkownika z programu *kadmin* – część zawierająca klucze (skrót z hasła):

Number of keys: 8

Key: vno 1, AES-256 CTS mode with 96-bit SHA-1 HMAC, no salt

Key: vno 1, AES-128 CTS mode with 96-bit SHA-1 HMAC, no salt

Key: vno 1, Triple DES cbc mode with HMAC/sha1, no salt

Key: vno 1, ArcFour with HMAC/md5, no salt

Key: vno 1, DES with HMAC/sha1, no salt

Key: vno 1, DES cbc mode with RSA-MD5, no salt

Key: vno 1, DES cbc mode with CRC-32, Version 4

Key: vno 1, DES cbc mode with CRC-32, AFS version 3

Attributes:

Policy: polityka

Polityki związane z hasłem

- W bazie *KDC* możliwe są do ustawienia zasady związane z bezpieczeństwem hasła (polityka).
- Nową politykę tworzy się w programie *kadmin* za pomocą polecenia *add_policy*.
- Możliwe do ustawienia parametry polityki haseł to:
 - *minlife* – minimalny wiek życia hasła (klucza),
 - *maxlife* – maksymalny wiek życia hasła,
 - *minlength* – minimalna długość hasła,
 - *minclasses* – minimalna liczba klas znaków w haśle,
 - *history* – liczba pamiętanych haseł.
- Nazwa polityki jest jednym z atrybutów użytkownika.
- Ustawienie polityki użytkownikowi w czasie jego tworzenia (*addprinc*) lub modyfikacji (*modprinc*) – atrybut *policy*.

Polityki związane z hasłem

- Przykład definicji polityki o nazwie „*polityka*”

```
kadmin.local: addpol -minlife "3 days" -maxlife "180 days"  
-minlength 8 -minclasses 3 -history 7 polityka
```

- Wypisanie atrybutów zdefiniowanej polityki – „*polityka*”

```
kadmin.local: getpol polityka  
Policy: polityka  
Maximum password life: 2592000  
Minimum password life: 259200  
Minimum password length: 8  
Minimum number of password character classes: 3  
Number of old keys kept: 7  
Reference count: 2
```

Konfiguracja stacji klienckiej

- Na komputerze klienckim instalacja *krb5-workstation*.
- Plik konfiguracyjny klienta to */etc/krb5.conf*
- Plik ten składa się z następujących sekcji:
 - *[logging]* – ścieżki na pliki zawierające odpowiednie dzienniki (logi),
 - *[libdefaults]* – ustawienia domyślne dla biblioteki *kerberosa*,
 - *[realms]* – definicje królestw i ich parametrów,
 - *[domain_realms]* – określenie zależności pomiędzy nazwami domen *DNS*, a odpowiadającymi im nazwami królestw,
 - *[appdefaults]* – ustawienia domyślne dla aplikacji wykorzystujących *kerberosa*.
- Konfiguracja uwierzytelnienia użytkownika w czasie podłączenia do systemu z wykorzystaniem protokołu *kerberos* za pomocą biblioteki *PAM*.

Konfiguracja stacji klienckiej c.d.

- Przykładowa zawartość sekcji *[realms]* pliku */etc/krb5.conf*

```
[realms]
STUDENCI.AGH.EDU.PL = {
    kdc = krberv.studenci.agh.edu.pl:88
    admin_server = krberv.studenci.agh.edu.pl:749
    default_domanin = studenci.agh.edu.pl
}
```

- Występuje tu definicja jednego królestwa *STUDENCI.AGH.EDU.PL*, w którym:
 - *[kdc]* – adres (wraz z portem) serwera usługi *kerbeos*,
 - *[admin_server]* – adres serwera umożliwiającego administrację bazą *KDC* – zwykle ten sam komputer co *KDC*
 - *[default_domain]* – nazwa domeny DNS, w której są hosty z definiowanego królestwa

Testowanie działania *KDC*

- Musi być skonfigurowany klient – plik */etc/krb5.conf*.
- W celu wykonania uwierzytelnienia użytkownika można użyć polecenia *kinit pirincipal@KROLESTWO*
- Po pomyślnym uwierzytelnieniu użytkownik otrzyma bilet *TGT*, który zostanie zapisany w cache'u.
- Wyświetlenie zawartości cache'u użytkownika wykonuje się poleceniem *klist*.
- Standardowo bilety trzymane są w katalogu */tmp* i mają do nich dostęp tylko ich właściciele.
- Nazwy plików są według schematu *krb5cc_numer*.
- Wyczyszczenie cache'a użytkownika odbywa się za pomocą polecenia *kdestroy*.
- Zmiana hasła przez użytkownika – polecenie *kpasswd*.

Testowanie działania *KDC* c.d.

- Przykład wyświetlenia zawartości cache'a użytkownika:

Ticket cache: FILE:/tmp/krb5cc_0

Default principal: franio@STUDENCI.AGH.EDU.PL

Valid starting	Expires	Service principal
----------------	---------	-------------------

03/18/15 15:31:23	03/19/15 15:31:23	
-------------------	-------------------	--

krbtgt/STUDENCI.AGH.EDU.PL@STUDENCI.AGH.EDU.PL		
--	--	--

Kerberos 4 ticket cache: /tmp/tkt0

- Bilet który ma użytkownik:
 - znajduje się w pliku */tmp/krb5cc_0*
 - ważny jest przez dobę,
 - jest to bilet na bilety – *TGT*, a więc może dać dostęp do innych usług (użytkownik może otrzymać inne bilety dzięki niemu).

Uwierzytelnianie – użycie PAM

- *PAM (Pluggable Authentication Modules)* to modułarny system uwierzytelniania użytkownika.
- Umożliwia zastosowanie wielu rodzajów uwierzytelniania Takich jak m. in. hasło, odcisk palca, system *kerberos* czy *LDAP* , itp.
- Pozwala definiować różne sposoby uwierzytelniania do różnych usług.
- Dzięki temu użytkownik nie musi istnieć w systemie by mieć dostęp do usługi
Np. konfiguracja serwera pocztowego z użytkownikami w *kerberosie*.
- Możliwe jest wykorzystywanie wielu różnych modułów
Np. moduł *pam_cracklib* służy do sprawdzania złożoności hasła.
- *PAM* ujednolica proces uwierzytelnienia użytkownika
Aplikacja prosi o uwierzytelnienie, a *PAM* wykonuje resztę.

Konfiguracja PAM – RH

- Katalog z plikami konfiguracyjnymi *PAM* to */etc/pam.d*
- Konfiguracja uwierzytelniania w systemie – *system-auth*
- Linia składa się z: *typ kontrola moduł argumenty_modułu*.
- *Typ* określa grupę do której odnosi się moduł; możliwe są:
 - *auth* – uwierzytelnia użytkownika, a następnie może nadać odpowiednie przywileje użytkownikowi, jak np. członkostwo w grupie,
 - *account* – zarządza kontem użytkownika na podstawie określonych zasad przydzielania zasobów; np. liczba maksymalnych logowań, dzień tygodnia, maksymalna liczba zalogowanych itp.,
 - *password* – moduł odpowiedzialny za zmianę hasła; tu może być zrobione sprawdzanie złożoności hasła,
 - *session* – wykonanie czynności po uwierzytelnieniu użytkownika; np. montowanie systemów plików, zbieranie informacji, itp.

Konfiguracja PAM c.d.

- Możliwe jest wystąpienie kilku linii określających ten sam *typ*, a więc *auth*, *account*, *password* czy *session*.
- Kolejne linie tego samego *typu* definiują stos.
- Proces uwierzytelniania przebiega przez kolejne elementy stosu, a więc wykonywane są odpowiednie moduły.
- Końcowy wynik uwierzytelnienia zależy od ostatecznego wyniku przejścia przez stos – ustawiana jest flaga typu uwierzytelnienie się „powiodło” lub się „nie powiodło”.
- *Kontrola* definiuje sposób zachowania *PAM-API*, a więc określa co ma nastąpić w wypadku powodzenia czy porażki w procesie uwierzytelnienia przez dany moduł – jaką wartość flagi ustawić i czy przerwać wykonanie stosu.
- Daje to możliwość różnych sposobów uwierzytelnienia.

Konfiguracja PAM c.d.

- Możliwe wartości *kontroli* to:
 - *required* – wykonanie modułów powoduje ustawienie końcowego rezultatu sukces lub porażkę i przejście dalej w stosie,
 - *requisite* – podobnie jak *required* lecz przerwane jest wykonywanie stosu – w wypadku porażki, taka jest zwrócona i zakończony zostaje proces uwierzytelniania,
 - *sufficient* – w wypadku porażki przechodzi się do kolejnych elementów stosu i wynik końcowy nie jest ustawiany, a w wypadku sukcesu proces uwierzytelnienia kończy się pozytywnie,
 - *optional* – wykonanie modułu nie ustawia końcowego rezultatu i przechodzi się do kolejnych elementów stosu; ma głównie znaczenie gdy jest jedynym elementem stosu.
- W dalszej części linii znajduje nazwa modułu do wykonania wraz z listą argumentów.

Konfiguracja PAM – kerberos

- Fragment pliku *system-auth* dotyczący uwierzytelnienia; widoczne jest wprawdzie uwierzytelnianie typowe dla systemu Linux, a następnie z wykorzystaniem *kerberos*:

auth	required	pam_env.so
auth	sufficient	pam_fprintd.so
auth	sufficient	pam_unix.so nullok try_first_pass
auth	requisite	pam_succeed_if.so uid >= 500 quiet
auth	sufficient	pam_krb5.so use_first_pass
auth	required	pam_deny.so

- Podobnie, część dotycząca hasła:

password	requisite	pam_cracklib.so try_first_pass retry=3
password	sufficient	pam_unix.so sha512 shadow nullok
		try_first_pass use_authtok
password	sufficient	pam_krb5.so use_authtok
password	required	pam_deny.so

Kerberos w systemie Windows

- Firma Microsoft w używa protokołu *kerberos* jako natywnego do uwierzytelniania w usługach *Active Directory* (od początku istnienia tych usług).
- Wprowadzone zostało rozszerzenie – do biletu dodano pole zwane *PAC* – *Privilege Account Certificate*.
- Rozszerzenie to umożliwia nie tylko uwierzytelnienie klienta, ponieważ zawiera dane związane z autoryzacją użytkownika.
- Pole to zawiera wiele danych, głównie identyfikatory związane z bezpieczeństwem; między innymi takie jak:
 - ID grupa podstawowej do której należy użytkownik,
 - tablica ID grup do których należy użytkownik,
 - SID domeny do której należy użytkownik.

Kerberos – protokoły sieciowe

- *Kerberos* wykorzystuje architekturę klient / serwer.
- Usługa *KDC* musi nasłuchiwać i oczekiwać na połączenia; standardowo wykorzystuje protokół UDP – port 88.
- Usługa *kadmin* nie jest konieczna, lecz z reguły wykorzystywana – używa protokołu TCP – port 749.
- W serwerze *DNS* powinny pojawić się rekordy typu *SRV* określające lokalizację tych usług w sieci; przykładowo:

krbserve	IN	192.168.13.2			
_kerberos._udp	SRV	0	0	88	krbserve
_kerberos-adm._tcp	SRV	0	0	749	krbserve



Usługa synchronizacji czasu

Network Time Protocol (NTP)

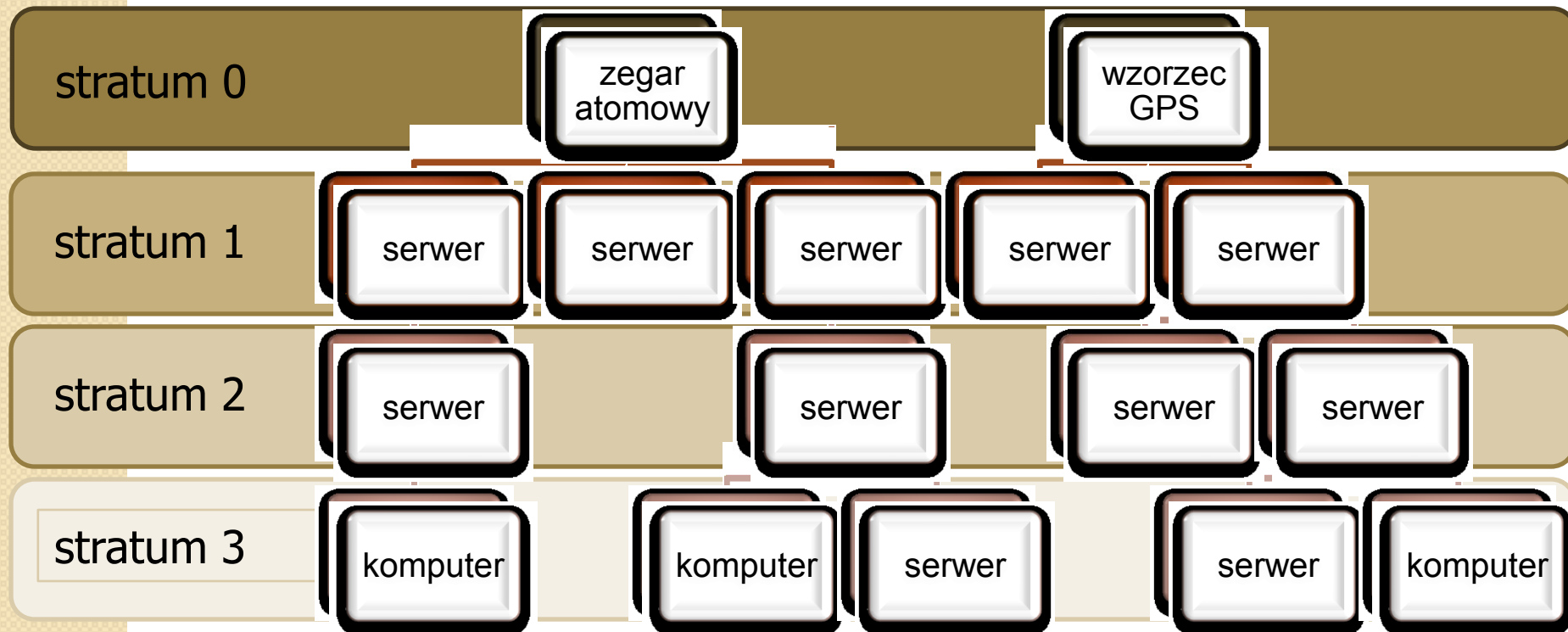
Krzysztof Boryczko

NTP definicja i dokumenty

- *Network Time Protocol (NTP)* – jest to protokół umożliwiający synchronizację czasu pomiędzy systemem klienckim (komputer, telefon, itp.), a serwerem czasu.
- Wykorzystuje *Coordinated Universal Time (UTC)*.
- Dokładność synchronizacji sięga 10 ms, a w sieciach lokalnych może nawet osiągnąć 200 μ s.
- Jeden z najstarszych protokołów w sieci – początki 1985 r.
- Dokumenty opisujące protokół *NTP*:
 - RFC 1305 – specyfikacja wersji trzeciej protokołu *NTP* – 1992 r.
 - RFC 778, RFC 891 I RFC 956 – starsze dokumenty mające związek z technicznymi aspektami protokołem *NTP*.
- Aktualnie obowiązuje wersja 4 jednak brak dla niej RFC.
- Uproszczona wersja *NTP*, to *SNTP v.4* – RFC 2030.

Synchronizacja czasu – NTP

- *NTP* ma korzenie w systemie *UNIX* jednak aktualnie używany praktycznie przez wszystkie systemy (pełna implementacja w systemach Microsoft od Windows 2003 Server)
- Serwery czasu tworzą drzewo synchronizacji.



Poziomy synchronizacji NTP

- Kolejne poziomy w drzewie synchronizacji:
 - *stratum 0* – początek drzewa. Znajdują się tu wzorcowe zegary czasu UTC, takie jak: zegary atomowe, wzorce laserowe, wzorce radiowe oparte o GPS, GLONASS, itp.,
 - *stratum 1* – serwery czasu podłączone bezpośrednio czy np. GPS, ale nie przez sieć, do zegarów poziomu *stratum 0*,
 - *stratum 2* – serwery czasu podłączone przez sieć do serwerów poziomu wyższego. Ta i powyższa warstwa powinny składać się z sprzętowych serwerów czasu, serwerów korporacyjnych itp,
 - *stratum 3* – warstwa ta przeznaczona jest dla lokalnych serwerów czasu, z którymi komunikują się już stacje klienckie.

Konfiguracja NTP – RH

- Konfiguracja stacji klienckiej jak i serwera czasu oparta jest o te same pliki konfiguracyjne. Różni się jedynie ich zawartością.
- Instalacja systemu NTP – instalacja pakietu *ntp*.
- Główny plik konfiguracyjny */etc/ntp.conf* zawiera między innymi listę serwerów czasu użytych do synchronizacji:

```
server 0.fedora.pool.ntp.org  
server 1.fedora.pool.ntp.org  
server 2.fedora.pool.ntp.org
```

- W Polsce znajdują się dwa główne serwery czasu zlokalizowane w Laboratorium Czasu i Częstotliwości Głównego Urzędu Miar uruchomione od 14.05.2008 r.:
 - *tempus1.gum.gov.pl* – 212.244.36.227
 - *tempus2.gum.gov.pl* – 212.244.36.228

Konfiguracja klient/serwer NTP

- Różnica pomiędzy konfiguracją klienta a serwera czasu wynika z definicji dotyczących zezwolenia na synchronizację czasu innym hostom z systemem.
- Ustawienie ograniczeń synchronizacji – parametr *restrict*.
- Wartość *default* parametru *restrict* – domyślne ustawienie obowiązujące wszystkich klientów. Standardowe ustawienie w dystrybucji *Fedora* to:

```
restrict default kod nomodify notrap nopeer noquery
```

- Oznacza to zabronienie wszystkim klientom na synchronizację – z lokalnym systemem lecz pozwala synchronizować czas ze źródłem serwera.

Konfiguracja klient/serwer c.d.

- Zabronienie innym komunikacji z naszym hostem (dotyczy również serwerów czasu – dla nich inna wartość *restrict*):

```
restrict default ignore
```

- Konfigurując serwer czasu należy zdefiniować uprawnienia dostępu dla klientów z obsługiwanych sieci:

```
restrict 192.168.13.0 mask 255.255.255.0 nomodify notrap
```

- Ustawienia te pozwalają hostom z sieci 192.168.13.0/24 na synchronizację lecz zabraniają modyfikacji serwera.
- Po uruchomieniu usługi synchronizacja czasu może zająć kilka minut; przyspieszenie – dodanie opcji *iburst* w linii definiującej serwer:

```
server tempus1.gum.gov.pl iburst  
server tempus2.gum.gov.pl iburst
```

Testowanie klienta/serwera

- Wykonanie konfiguracji i uruchomieniu usługi *ntpd*.
- Po zsynchronizowaniu czasu pojawią się w dziennikach systemu */var/log/messages* komunikaty tego typu:

```
Mar 20 00:18:09 dns1 ntpd[968]: synchronized to 212.244.36.227,  
stratum 1
```

```
Mar 20 00:18:09 dns1 ntpd[968]: kernel time sync status change  
2001
```

- Widoczne jest, że czas został zsynchronizowany z serwerem 212.244.36.227 poziomu stratum 1
- Do sprawdzenia stanu klienta *NTP* służy polecenie *ntpstat*

```
[root@dns1 ~]# ntpstat  
synchronised to NTP server (212.244.36.228) at stratum 2  
time correct to within 21 ms  
polling server every 64 s
```

NTP – protokoły sieciowe

- *NTP* wykorzystuje architekturę klient / serwer.
- Serwer musi nasłuchiwać i oczekiwać na połączenia dla zdefiniowanych w konfiguracji sieci.
- *NTP* do komunikacji używa protokołu UDP i portu 123.