

**TUGAS KULIAH MAKALAH  
KEAMANAN SISTEM INFORMASI (EC-5010)**

***KERBEROS***

**NAMA : Christian Henry Wijaya**

**NIM : 132 03 041**



**Program Studi Teknik Elektro  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
2006**

## **ABSTRAK**

Kerberos merupakan suatu protocol autentikasi jaringan. Kerberos dirancang untuk memberikan autentikasi yang kuat untuk aplikasi client/server dengan menggunakan *secret-key cryptography*.

Internet merupakan tempat yang tidak aman. Banyak protokol yang menggunakan Internet tidak memberikan sistem autentikasi yang aman. Tools untuk mencuri password sering digunakan oleh para hacker. Jadi, aplikasi yang mengirimkan password yang tidak dienkripsi melalui jaringan sangatlah rentan dan berbahaya.

Kerberos diciptakan oleh MIT (Massachusetts Institute of Technology) sebagai solusi untuk masalah keamanan ini. Dalam tulisan ini akan dibahas mengenai Kerberos: Apa itu Kerberos, mengapa Kerberos, bagaimana cara kerja Kerberos, serta kelemahannya.

## DAFTAR ISI

Abstrak	2
Daftar Isi	3
Bab 1 Pendahuluan	4
1.1 Latar Belakang	4
1.2 Batasan Masalah	5
1.3 Tujuan Penulisan	5
1.4 Metodologi Penelitian	5
Bab 2 Dasar Kerberos	6
2.1 Authentication, Integrity, Confidentiality, dan Authorization	6
2.2 Apa itu Kerberos	7
2.3 Dasar-Dasar Kerberos	8
Bab 3 Operasi Kerberos	12
3.1 Authentication Server Request (AS_REQ)	14
3.2 Authentication Server Reply (AS_REP)	14
3.3 Ticket Granting Server Request (TGS_REQ)	15
3.4 Ticket Granting Server Reply (TGS_REP)	15
3.5 Application Request (AP_REQ)	17
Bab 4 Kesimpulan	19
Daftar Pustaka	20

# **BAB 1**

## **PENDAHULUAN**

### 1.1 Latar Belakang

Internet merupakan tempat yang tidak aman. Banyak protokol yang menggunakan Internet tidak memberikan sistem autentikasi yang aman. Tools untuk mencuri password sering digunakan oleh para hacker. Jadi, aplikasi yang mengirimkan password yang tidak dienkripsi melalui jaringan sangatlah rentan dan berbahaya.

Beberapa situs menggunakan firewall untuk mengatasi masalah keamanan jaringan mereka. Tetapi sayangnya, firewall mengasumsikan bahwa ancaman bahaya berasal dari luar, padahal seringkali pada kenyataannya tidaklah demikian. Ancaman bahaya justru sering datang dari dalam. Kerusakan-kerusakan dan kejahatan-kejahatan jaringan komputer sering berasal dari pihak dalam. Firewall juga memiliki kelemahan bahwa mereka membatasi bagaimana pengguna dapat menggunakan Internet. Firewall dapat diasosiasikan dengan pernyataan yang mengatakan bahwa tidak ada komputer yang lebih aman daripada komputer yang tidak terhubung ke dalam jaringan dan komputer tersebut dimatikan. Kita tahu, bahwa hal tersebut tidak realistis dan tidak dapat diterima.

Oleh karena hal-hal tersebut di atas, maka diperlukan suatu protokol autentikasi jaringan yang dapat diandalkan. Salah satunya adalah Kerberos. Kerberos dirancang untuk memberikan autentikasi yang kuat untuk aplikasi client/server dengan menggunakan *secret-key cryptography*. Kerberos diciptakan oleh MIT (Massachusetts Institute of Technology) di Amerika Serikat.

## 1.2 Batasan Masalah

Dalam tulisan ini akan dibahas tentang apa itu Kerberos, cara kerja Kerberos, dan keterbatasannya. Namun sebelumnya akan dibahas tentang pengertian secara umum mengenai autentikasi (authentication), integritas (integrity), kerahasiaan (confidentiality), dan autorisasi (authorization).

## 1.3 Tujuan Penulisan

Tulisan ini bertujuan untuk memahami apa itu Kerberos dan bagaimana cara kerjanya dalam mengatasi masalah keamanan autentikasi jaringan.

## 1.4 Metodologi Penelitian

Metodologi penelitian yang dilakukan adalah dengan melakukan studi literatur-literatur yang terkait dengan tema.

## **BAB 2**

### **DASAR KERBEROS**

#### 2.1 Authentication, Integrity, Confidentiality, dan Authorization

*Authentication* atau autentikasi adalah proses verifikasi identitas dari seorang anggota yang memberikan suatu data, dan integritas dari data tersebut. *Principal* adalah anggota yang identitasnya telah diverifikasi. *Verifier* adalah anggota yang meminta jaminan identitas dari principal. Integritas data adalah jaminan bahwa data yang diterima adalah data yang sama dengan data yang dikirimkan. Mekanisme autentikasi berbeda dalam jaminan yang mereka berikan: beberapa menyatakan bahwa data dibuat oleh principal pada satu titik di masa lampau, beberapa menyatakan bahwa data dibuat oleh principal di masa sekarang (present), dan beberapa yang lainnya lagi menyatakan bahwa data baru saja dibuat oleh principal ketika dikirimkan. Mekanisme juga berbeda dalam hal jumlah verifier: beberapa ada yang menggunakan verifier tunggal untuk setiap message, ada yang menggunakan multiple verifiers. Perbedaan ketiga adalah dalam hal apakah mekanisme tersebut mendukung non-repudiation, yaitu kemampuan verifier untuk membuktikan kepada pihak ketiga bahwa pesan tersebut benar-benar dibuat oleh principal.

Karena perbedaan-perbedaan ini mempengaruhi performa, maka kita perlu melihat kebutuhan-kebutuhan yang dibutuhkan oleh masing-masing aplikasi sebelum kita menentukan metoda mana yang akan digunakan. Contohnya, autentikasi untuk *electronic mail (email)* membutuhkan multiple verifiers dan non-repudiation, tetapi toleran terhadap latensi waktu.

Layanan keamanan lainnya adalah kerahasiaan (confidentiality) dan otorisasi (authorization). *Confidentiality* adalah perlindungan informasi terhadap mereka yang tidak seharusnya menerima informasi tersebut. Kebanyakan metode autentikasi yang baik memberikan pilihan untuk confidentiality ini. *Authorization* adalah proses dimana seseorang menentukan apakah principal dapat melakukan suatu operasi. Otorisasi biasanya dilakukan setelah principal melalui proses autentikasi.

## 2.2 Apa itu Kerberos

Kerberos merupakan layanan autentikasi yang dikembangkan oleh MIT (Massachusetts Institute of Technology) Amerika Serikat, dengan bantuan dari Proyek Athena. Tujuannya adalah untuk memungkinkan pengguna (user) dan layanan (service) untuk saling mengautentikasi satu dengan yang lainnya. Dengan kata lain, saling menunjukkan identitasnya.

Tentu saja ada banyak cara untuk menunjukkan identitas pengguna kepada layanan tersebut. Salah satu cara yang paling umum adalah dengan penggunaan password. Seseorang “log in” ke dalam server dengan mengetikkan username dan password, yang idealnya hanya diketahui oleh pengguna dan server tersebut. Server tersebut kemudian diyakinkan bahwa orang yang sedang berusaha mengaksesnya adalah benar-benar pengguna.

Namun, penggunaan password ini memiliki banyak kelemahan. Misalnya seseorang memilih password yang mudah ditebak oleh orang lain dalam beberapa kali usaha percobaan. Dalam hal ini dikatakan bahwa password tersebut lemah. Masalah lainnya timbul ketika password ini akan dikirimkan melalui jaringan: Password tersebut harus melalui jaringan tanpa dienkripsi. Dengan kata lain, jika ada orang lain yang

“mendengarkan” jaringan tersebut dapat mencegat password itu dan mendapatkannya kemudian menggunakannya untuk masuk sebagai pengguna.

Inovasi utama dalam Kerberos adalah gagasan bahwa password tersebut dapat dilihat sebagai suatu *shared secret*, sesuatu rahasia yang hanya pengguna dan server yang mengetahuinya. Menunjukkan identitas dilakukan tanpa pengguna harus membuka rahasia tersebut. Ada suatu cara untuk membuktikan bahwa kita mengetahui rahasia tersebut tanpa mengirimnya ke jaringan.

### 2.3 Dasar-Dasar Kerberos

Pendekatan dasar dari Kerberos adalah menciptakan suatu layanan yang tujuan satu-satunya adalah untuk autentikasi. Alasannya adalah untuk membebaskan layanan tersebut dari keharusan untuk mengurus record akun pengguna. Dalam pendekatan ini, pengguna dan layanan harus memercayai Kerberos authentication server (AS). AS ini berperan sebagai pengenalan kepada mereka. Untuk melakukan hal ini, pengguna dan layanan harus mempunyai *shared secret key* yang telah terdaftar di AS. Key tersebut dinamakan *long-term keys*, karena memang digunakan dalam jangka waktu yang cukup lama, yaitu berminggu-minggu atau berbulan-bulan.

Ada tiga langkah dasar dalam proses autentikasi pengguna kepada layanan. Pertama, pengguna mengirimkan request kepada AS, meminta untuk mengautentikasi dirinya terhadap layanan. Dalam langkah kedua, AS bersiap untuk memperkenalkan pengguna dan layanan satu sama lainnya. Hal ini dilakukan dengan cara menciptakan suatu secret key yang baru dan random yang akan dibagikan hanya kepada pengguna dan layanan. AS

mengirimkan pesan kepada pengguna yang terdiri atas dua bagian. Satu bagian mengandung random key bersama nama layanan, yang dienkripsi dengan long-term key milik pengguna. Bagian lainnya mengandung random key yang sama bersama nama pengguna, yang dienkripsi dengan long-term key milik layanan. Dalam bahasa Kerberos, pesan yang pertama sering disebut *credentials*, sedangkan pesan yang kedua disebut *ticket*, dan random key tersebut disebut dengan *session key*.

Pada tahap ini, hanya pengguna yang mengetahui session key. Pengguna membuat suatu pesan, misalnya timestamp, kemudian dienkripsi menggunakan session key. Pesan ini disebut *authenticator*. Pesan authenticator ini dikirimkan bersama dengan ticket kepada layanan. Kemudian layanan mendekripsikan ticket dengan long-term key-nya, mendapatkan session key, yang pada gilirannya digunakan untuk mendekripsikan authenticator. Layanan tersebut memercayai AS, sehingga ia dapat yakin bahwa hanya pengguna yang terdaftar yang dapat membuat authenticator semacam itu. Hal ini mengakhiri proses autentikasi pengguna terhadap layanan. Bagaimana Kerberos beroperasi atau bekerja akan dijelaskan dengan lebih detail pada bab selanjutnya.

## 2.4 Ticket Granting Server

Salah satu ketidaknyamanan dalam penggunaan password adalah setiap kali kita mengakses layanan, maka kita harus mengetikkan password tersebut. Hal tersebut dapat menjadi menyulitkan jika kita memiliki banyak akun dan kita harus memasukkan password tersebut satu per satu. Mungkin kita akan mencoba membuat password yang mudah untuk diketik, dan sebagai akibatnya maka password kita tersebut akan menjadi mudah untuk ditebak. Dan jika kita menggunakan password yang

sama untuk setiap akun kita, maka jika password kita tersebut terbongkar, seluruh akun kita menjadi rentan ditembus.

Kerberos mengatasi masalah ini dengan memperkenalkan suatu layanan baru, yaitu Ticket Granting Server (TGS). TGS secara logika berbeda dari AS, meskipun mungkin mereka berada dalam satu mesin yang sama (keduanya sering disebut sebagai KDC – Key Distribution Center) Tujuan dari TGS adalah untuk menambahkan layer tambahan sehingga pengguna hanya perlu untuk memasukkan password sebanyak satu kali saja. Ticket dan session key yang didapat dari password tersebut digunakan untuk ticket selanjutnya.

Jadi, sebelum mengakses suatu layanan regular, pengguna meminta ticket dari AS untuk “berbicara” dengan TGS. Ticket ini disebut *ticket granting ticket (TGT)* , sering juga disebut sebagai *initial ticket*. Session key untuk TGT dienkripsi menggunakan long-term key milik pengguna, sehingga password dibutuhkan untuk mendekripsikannya kembali dari respon AS kepada pengguna.

Setelah menerima TGT, kapanpun pengguna ingin mengakses layanan, dia meminta ticket bukan dari AS tetapi dari TGS. Lebih jauh lagi, jawabannya tidak dienkripsi menggunakan secret key milik pengguna, tetapi menggunakan session key yang datang bersama TGT, sehingga password pengguna tidak diperlukan untuk mendapatkan session key yang baru.

Hal tersebut dapat diilustrasikan sebagai berikut. Bayangkan Anda adalah seorang karyawan di sebuah gedung. Anda menunjukkan kartu ID regular untuk mendapatkan kartu ID guest. Sekarang, ketika Anda ingin memasuki berbagai ruangan di gedung tersebut, Anda tidak menunjukkan kartu ID regular lagi dan lagi, yang mungkin dapat mudah sekali hilang atau dicuri orang, tetapi Anda tinggal menunjukkan kartu ID guest yang

hanya berlaku untuk suatu waktu yang singkat saja. Jika kartu ID guest tersebut hilang atau dicuri, Anda dapat dengan segera memblokirnya dan membuat kartu yang baru, suatu hal yang tidak dapat dilakukan terhadap kartu ID regular.

Keuntungan dari hal ini adalah jika password biasanya valid untuk jangka waktu yang panjang, misalnya berminggu-minggu atau berbulan-bulan, maka TGT hanya valid untuk jangka waktu yang pendek, misalnya hanya delapan hingga sepuluh jam saja. Setelah itu, TGT tidak dapat digunakan lagi oleh siapapun. TGT ini disimpan dalam *credentials cache*.

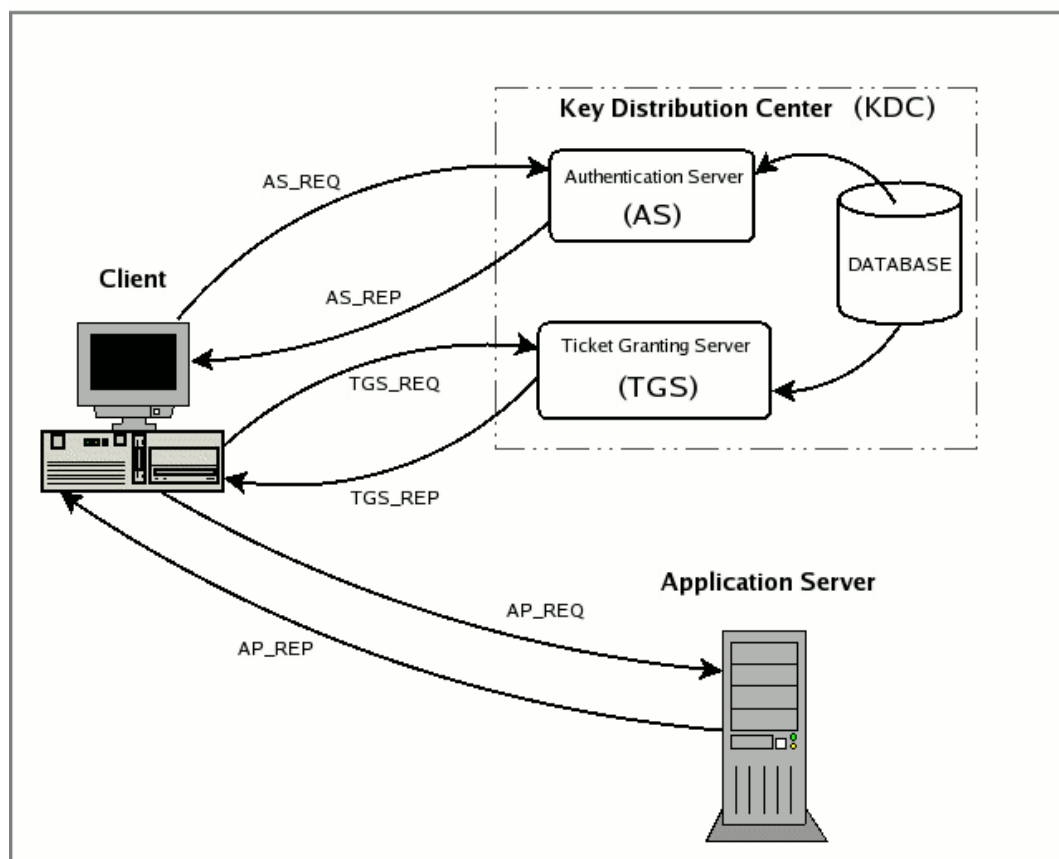
## **BAB 3**

### **OPERASI KERBEROS**

Pada bab ini akan dijelaskan operasi atau cara kerja Kerberos secara lebih detail. Sebelum menjelaskan bagaimana kerberos beroperasi atau bekerja, kita harus mengetahui paket apa saja yang dikirim di antara pengguna dan KDC (AS dan TGS), dan antara pengguna dengan layanan selama proses autentikasi. Perlu diingat bahwa layanan tidak pernah berkomunikasi secara langsung dengan KDC (Key Distribution Center). Berikut adalah daftar paket-paket:

- **AS\_REQ** adalah request autentikasi pengguna awal. Pesan ini ditujukan kepada komponen KDC, yaitu AS.
- **AS\_REP** adalah jawaban dari AS terhadap pesan sebelumnya. Pada dasarnya pesan ini mengandung TGT (dienkripsi menggunakan TGS secret key) dan session key (dienkripsi menggunakan secret key dari pengguna).
- **TGS\_REQ** adalah request dari pengguna kepada Ticket Granting Server (TGS) untuk mendapatkan service ticket. Paket ini mengandung TGT yang didapat dari pesan sebelumnya dan authenticator yang dibuat oleh pengguna dan dienkripsi dengan session key.
- **TGS\_REP** adalah jawaban dari Ticket Granting Server terhadap pesan sebelumnya. Dalam paket ini terdapat service ticket yang diminta (dienkripsi dengan secret key dari layanan) dan session key milik layanan yang dibuat oleh TGS dan dienkripsi dengan session key sebelumnya yang dibuat oleh AS.

- **AP\_REQ** adalah request yang dikirimkan oleh pengguna kepada layanan/aplikasi agar dapat mengakses layanannya. Komponennya adalah service ticket yang didapat dari TGS dengan jawaban sebelumnya dan authenticator yang dibuat oleh pengguna, tetapi kali ini dienkripsi menggunakan session key milik layanan (dibuat oleh TGS).
- **AP\_REP** adalah jawaban yang diberikan oleh layanan kepada pengguna untuk membuktikan bahwa layanan tersebut adalah benar merupakan layanan yang ingin diakses oleh pengguna. Paket ini tidak selalu diminta.



### 3.1 Authentication Server Request (AS\_REQ)

Pada tahap ini, pengguna meminta AS untuk mendapatkan Ticket Granting Ticket. Request ini tidak dienkripsi dan tampak seperti ini:

$$\text{AS\_REQ} = ( \text{Principal}_{\text{Client}} , \text{Principal}_{\text{Service}} , \text{IP\_list} , \text{Lifetime} )$$

### 3.2 Authentication Server Reply (AS\_REP)

Ketika pesan sebelumnya masuk, AS memeriksa apakah  $\text{Principal}_{\text{Client}}$  dan  $\text{Principal}_{\text{Service}}$  berada di database KDC. Jika salah satu saja dari antaranya tidak ada, maka pesan error dikirimkan kepada pengguna. Jika keduanya ada, maka AS akan memroses sebagai berikut:

- Secara random, AS akan membuat session key yang akan menjadi rahasia antara pengguna dan TGS, sebutlah  $\text{SK}_{\text{TGS}}$ .
- AS membuat Ticket Granting Ticket. Di dalamnya terdapat request dari principal pengguna, principal layanan, daftar IP address, tanggal dan waktu, lifetime, dan  $\text{SK}_{\text{TGS}}$ .

$$\text{TGT} = ( \text{Principal}_{\text{Client}} , \text{krbtgt/REALM@REALM} , \text{IP\_list} , \text{Timestamp} , \text{Lifetime} , \text{SK}_{\text{TGS}} )$$

- AS membuat dan mengirimkan balasan yang berisi: ticket yang dibuat sebelumnya yang dienkripsi menggunakan secret key untuk layanan ( $\text{K}_{\text{TGS}}$ ), principal layanan, timestamp, lifetime, dan session key semuanya dienkripsi menggunakan secret key untuk pengguna melakukan request terhadap layanan ( $\text{K}_{\text{USER}}$ )

$$AS\_REP = \{ \text{Principal}_{\text{Service}} , \text{Timestamp} , \text{Lifetime} , \text{SK}_{\text{TGS}} \} K_{\text{User}} \{ \text{TGT} \} K_{\text{TGS}}$$

### 3.3 Ticket Granting Server Request (TGS\_REQ)

Pada tahap ini, pengguna yang sudah terautentikasi (artinya dalam credential cache nya terdapat TGT dan session key  $\text{SK}_{\text{TGS}}$ ) dan ingin mengakses layanan tetapi belum mempunyai tiket yang sesuai, mengirimkan request (TGS\_REQ) kepada Ticket Granting Service dengan konstruksi sebagai berikut:

- Membuat authenticator dengan principal pengguna, timestamp dari mesin pengguna dan mengenkripsi semuanya dengan session key yang dibagi bersama dengan TGS.
- Membuat paket request yang berisi: principal layanan yang mana dibutuhkan tiket dan lifetime yang tidak dienkripsi, Ticket Granting Ticket yang sudah dienkripsi menggunakan key dari TGS, dan authenticator yang baru saja dibuat

$$TGS\_REQ = ( \text{Principal}_{\text{Service}} , \text{Lifetime} , \text{Authenticator} ) \{ \text{TGT} \} K_{\text{TGS}}$$

### 3.4 Ticket Granting Server Reply (TGS\_REP)

Ketika pesan sebelumnya tiba, pertama-tama TGS memverifikasi bahwa principal dari layanan yang diminta ada di dalam database KDC. Jika ada, TGS membuka TGT dan mengekstrak session key  $\text{SK}_{\text{TGS}}$  yang digunakan untuk mendekripsikan authenticator. Agar service ticket dapat dibuat, TGS memeriksa apakah kondisi-kondisi ini tercapai:

- TGT belum kadaluwarsa
- $Principal_{CLIENT}$  yang terdapat pada authenticator cocok dengan yang ada pada TGT
- Authenticator tidak terdapat pada replay cache dan belum kadaluwarsa
- Jika  $IP\_List$  tidak kosong, maka TGS memeriksa apakah alamat IP sumber dari paket request terdapat pada list tersebut

Kondisi-kondisi tersebut di atas membuktikan bahwa TGT benar-benar kepunyaan pengguna yang melakukan request dan kemudian TGS memulai proses sebagai berikut:

- Secara random, TGS membuat session key yang akan menjadi rahasia antara pengguna dengan layanan, sebutlah  $SK_{SERVICE}$ .
- TGS membuat service ticket yang di dalamnya berisi principal pengguna, principal layanan, daftar alamat IP, tanggal dan waktu, lifetime, dan  $SK_{SERVICE}$ . Ticket ini disebut  $T_{SERVICE}$ .

$T_{Service} = ( Principal_{Client} , Principal_{Service} , IP\_list , Timestamp , Lifetime , SK_{Service} )$

- TGS mengirim pesan balasan yang berisi: ticket yang baru saja dibuat yang dienkripsi menggunakan secret key milik layanan ( $K_{SERVICE}$ ), principal layanan, timestamp, lifetime, dan session key yang baru, semuanya dienkripsi menggunakan session key yang diekstrak dari TGT.

$$\text{TGS\_REP} = \{ \text{Principal}_{\text{Service}}, \text{Timestamp}, \text{Lifetime}, \text{SK}_{\text{Service}} \} \text{SK}_{\text{TGS}} \{ \text{T}_{\text{Service}} \} \text{K}_{\text{Service}}$$

### 3.5 Application Request (AP\_REQ)

Pengguna, yang mempunyai credential untuk mengakses layanan, dapat meminta server layanan agar ia dapat mengakses sumber-sumber melalui pesan AP\_REQ. Tidak seperti pesan-pesan sebelumnya dimana KDC terlibat, AP\_REQ bukanlah standar tetapi bervariasi bergantung aplikasi yang digunakan. Oleh karena itu, programmer aplikasi harus membuat strategi dimana pengguna dapat menggunakan credentialnya untuk membuktikan identitasnya kepada server. Contoh strategi yang dapat digunakan adalah sebagai berikut:

- Pengguna membuat authenticator yang berisi principal pengguna dan timestamp, dan mengenkripsi semuanya dengan menggunakan session key  $\text{SK}_{\text{SERVICE}}$

$$\text{Authenticator} = \{ \text{Principal}_{\text{Client}}, \text{Timestamp} \} \text{SK}_{\text{Service}}$$

- Membuat paket request yang berisi ticket layanan  $\text{T}_{\text{SERVICE}}$  yang dienkripsi menggunakan secret key nya dan authenticator yang baru saja dibuat.

$$\text{AP\_REQ} = \text{Authenticator} \{ \text{T}_{\text{Service}} \} \text{K}_{\text{Service}}$$

Ketika request sebelumnya tiba, server layanan membuka ticket menggunakan secret key untuk layanan yang diminta dan mengekstrak session key  $\text{SK}_{\text{SERVICE}}$  yang digunakan untuk

mendekripsikan authenticator. Untuk mengautentikasi pengguna, maka server memeriksa kondisi-kondisi sebagai berikut:

- Ticket belum kadaluwarsa
- Principal<sub>CLIENT</sub> yang ada dalam authenticator cocok dengan yang ada di ticket
- Authenticator tidak terdapat dalam replay cache dan belum kadaluwarsa
- Jika IP\_List (diekstrak dari ticket) tidak kosong, maka diperiksa apakah alamat IP dari AP\_REQ berada dalam daftar tersebut..

## **BAB 4**

### **KESIMPULAN**

Kesimpulan yang didapat dari tulisan ini adalah proses autentikasi merupakan suatu hal yang sangat kritical dalam keamanan sistem komputer. Tanpa mengetahui identitas dari pengguna yang melakukan request terhadap suatu layanan, sulit menentukan apakah operasi tersebut akan diizinkan atau tidak. Metode autentikasi tradisional sudah tidak cocok lagi digunakan saat ini karena para hacker selalu mengawasi jaringan untuk menyadap password. Penggunaan metoda autentikasi yang kuat yang tidak menunjukkan password menjadi suatu keharusan. Untuk memenuhi kebutuhan tersebut, Kerberos merupakan sistem autentikasi yang cocok digunakan.

## DAFTAR PUSTAKA

- <http://www.wikipedia.org>
- <http://www.cmf.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html>
- Brian Tung. *The Moron's Guide to Kerberos, Version 2.0*.
- B. Clifford Neuman and Theodore Ts'o, *Kerberos: An Authentication Service for Computer Networks*, IEEE Communications, 32(9) pp33–38. September 1994.