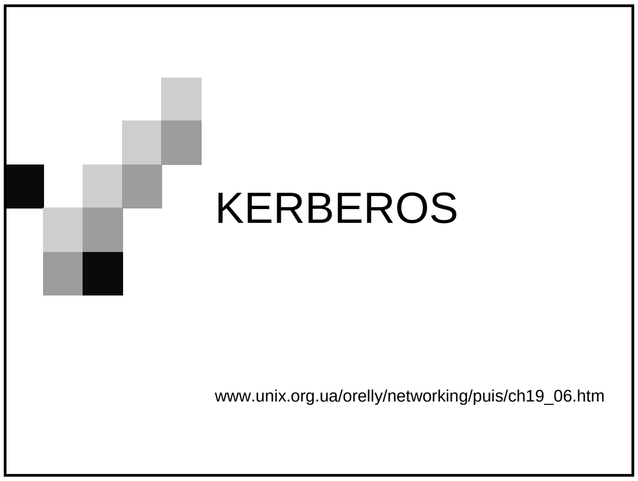
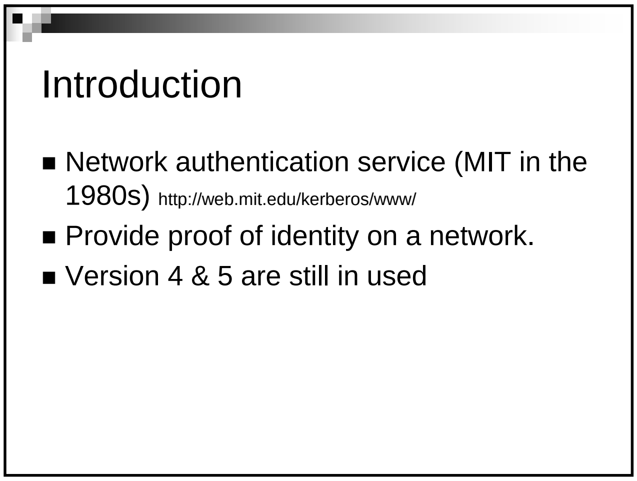


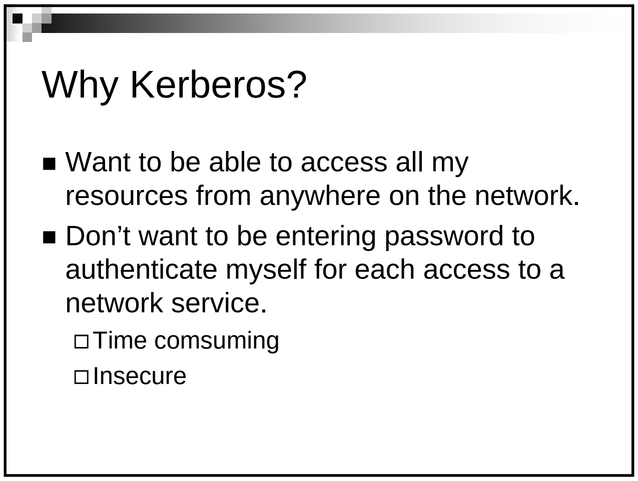
KERBEROS

www.unix.org.ua/orelly/networking/puis/ch19_06.htm



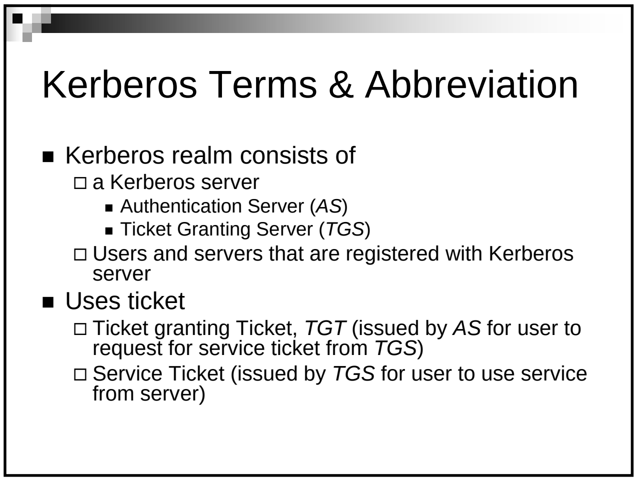
Introduction

- Network authentication service (MIT in the 1980s) http://web.mit.edu/kerberos/www/
- Provide proof of identity on a network.
- Version 4 & 5 are still in used



Why Kerberos?

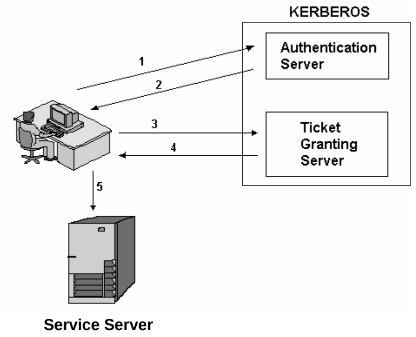
- Want to be able to access all my resources from anywhere on the network.
- Don't want to be entering password to authenticate myself for each access to a network service.
  - Time comsuming
  - Insecure



Kerberos Terms & Abbreviation

- Kerberos realm consists of
  - a Kerberos server
    - Authentication Server (*AS*)
    - Ticket Granting Server (*TGS*)
  - Users and servers that are registered with Kerberos server
- Uses ticket
  - Ticket granting Ticket, *TGT* (issued by *AS* for user to request for service ticket from *TGS*)
  - Service Ticket (issued by *TGS* for user to use service from server)

How Does it Work?

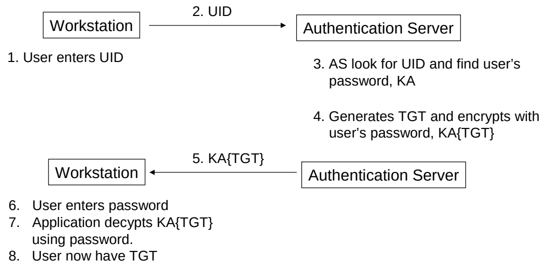


Initial Authentication
1. Authenticate
2. Receive TGT

Using TGT
3. Request Service Ticket
4. Receive Service Ticket
5. Get Service

Initial Kerberos Authentication

Kerberos version 4


Workstation → 2. UID → Authentication Server

1. User enters UID

3. AS look for UID and find user’s password, KA

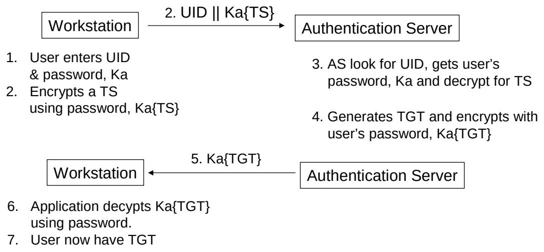
4. Generates TGT and encrypts with user’s password, KA{TGT}

Workstation ← 5. KA{TGT} ← Authentication Server

6. User enters password
7. Application decypts KA{TGT} using password.
8. User now have TGT

Initial Kerberos Authentication

Kerberos version 5


Workstation → 2. UID || Ka{TS} → Authentication Server

1. User enters UID & password, Ka
2. Encrypts a TS using password, Ka{TS}

3. AS look for UID, gets user’s password, Ka and decrypt for TS

4. Generates TGT and encrypts with user’s password, Ka{TGT}

Workstation ← 5. Ka{TGT} ← Authentication Server

6. Application decypts Ka{TGT} using password.
7. User now have TGT

Note

- Authentication is by password
- User’s password is never transmitted
- User’s knows their own password & Kerberos Server has a copy stored in it’s database in encrypted form
- Password is used to encrypt the Ticket Granting Ticket to secure from eavesdropper.

Why the Change?

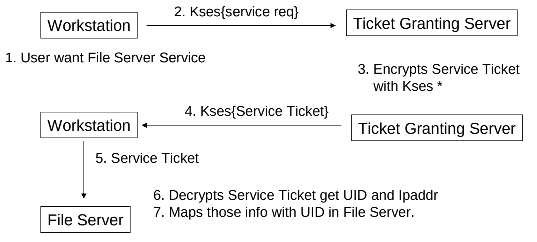
- Kerberos 4 was designed to minimize the amount of time the user’s password is stored on the workstation. Kerberos server doesn’t check if user is who he says he is.
- Attacker can intercept the encrypted *TGT* and mount a dictionary attack to guess the password.
- Kerberos 5 is more secure. Kerberos server makes sure that user’s password is valid before sending the *TGT* back to the user.

What is Ticket Granting Ticket

- A block of data that contains:
  - Session key : Kses
  - Ticket for *TGS* which is encrypted with both the session key and the Ticket Granting Server’s Key

    Ktgs{Kses{Ttgs}}
- User’s workstation can now contact the Kerberos *TGS* to obtain tickets for any services within the Kerberos realm.

Using the Ticket Granting Ticket



Workstation

2. Kses{service req}

Ticket Granting Server

1. User want File Server Service

3. Encrypts Service Ticket with Kses *

Workstation

4. Kses{Service Ticket}

Ticket Granting Server

5. Service Ticket

File Server

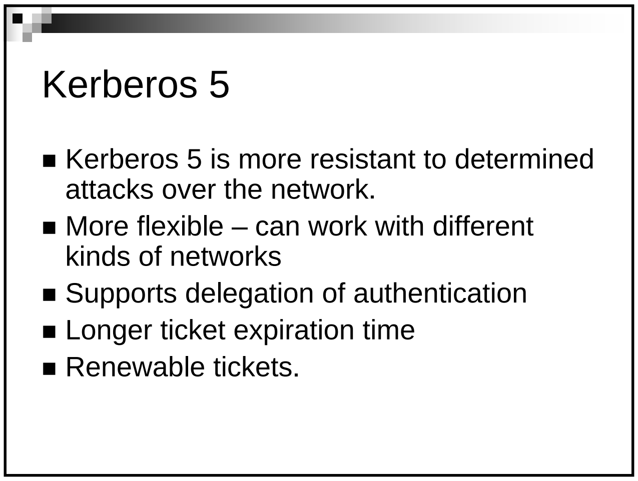
6. Decrypts Service Ticket get UID and Ipaddr

7. Maps those info with UID in File Server.

•Service Ticket is another ticket, Tx encrypted by File Server’s Key, Kfs{Tx}
•Tx contains UID, IPaddr, expiration time.

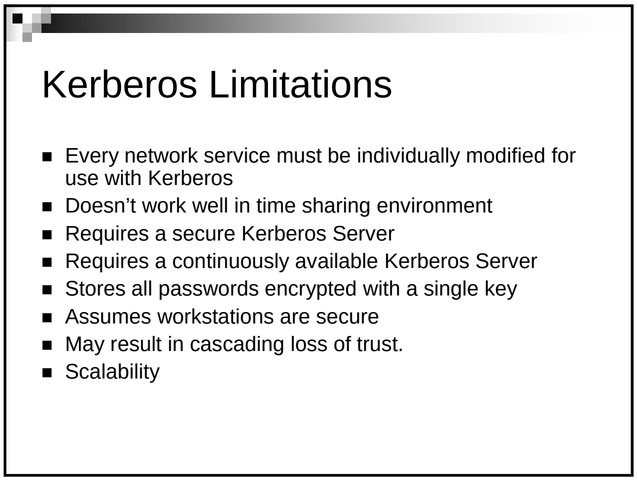
How is identity established

- *TGS* can establish user identity because the request is encrypted using the session key (available only if user can decrypt the *TGT* from the *AS*)
- File Server Service can establish user’s identity because the ticket (encrypted with File Server Service’s key) contains the user’s info – put in there by *TGS*.



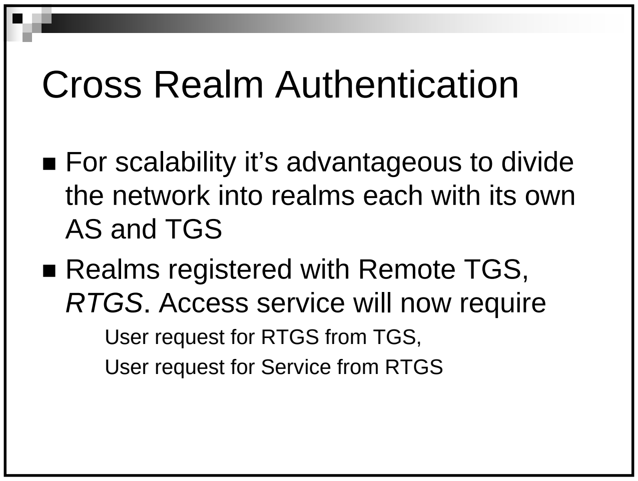
Kerberos 5

- Kerberos 5 is more resistant to determined attacks over the network.
- More flexible – can work with different kinds of networks
- Supports delegation of authentication
- Longer ticket expiration time
- Renewable tickets.



Kerberos Limitations

- Every network service must be individually modified for use with Kerberos
- Doesn’t work well in time sharing environment
- Requires a secure Kerberos Server
- Requires a continuously available Kerberos Server
- Stores all passwords encrypted with a single key
- Assumes workstations are secure
- May result in cascading loss of trust.
- Scalability



Cross Realm Authentication

- For scalability it’s advantageous to divide the network into realms each with its own AS and TGS
- Realms registered with Remote TGS, *RTGS*. Access service will now require

  User request for RTGS from TGS,
  User request for Service from RTGS