

ISA 666

Internet Security Protocols

Final Review

Final Review

Study the following chapters from the book:

- 9-12
- 19.1-19.7
- 23
- For Kerberos v5, study the slides provided in this review instead of reading chapter 14.

Study the following slides from the ISA-666 Website:

- Authentication
- TCP/IP Basics & Firewalls
- SSL, TLS, & SSH

ISA 666

Internet Security Protocols



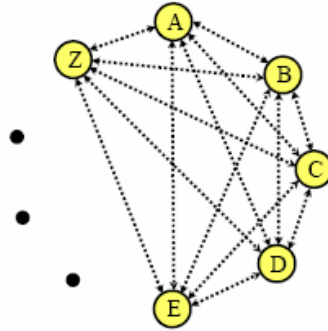
Kerberos

Authentication Problem In Distributed Systems

- Assume an open distributed environment in which users at workstations wish to access services on servers distributed throughout the network.
 - How many symmetric keys would you need?
- Restrict access to authorized users and to be able to authenticate requests for service.
 - How can you do this without modifying every node individually?

How many Symmetric Keys needed?

- N entities – There will be $N(N-1) / 2$ keys total
- Each entity has to store N-1 keys
- $K_{A,B}$ – Symmetric Key for A and B
- Administration Problems:
 - Adding new entities
 - Removing existing entities
 - Changing keys



Trusted Third Parties

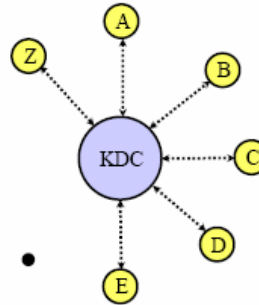
- Trusted Third Party (TTP) – another party (besides the communicating parties A and B) that is responsible for managing the "trust" in the system
- Types of TTP:
 - Key Distribution Centers (in symmetric systems)
 - Certificates Authorities (in asymmetric systems)

KDC (I)

- Key Distribution Center (KDC)
- N entities – $O(N)$ symmetric keys
- K_A – symmetric master key between A and KDC

- Administration Issues:

- Easy to add new entities
- Easy to remove existing entities
- Easy to change entities key



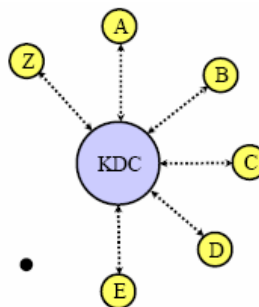
KDC (II)

- Advantages:

- Centralized administration
- One point of high security
- Easy system extendibility
- Easy function extendibility

- Disadvantages:

- KDC must be online
- KDC is point of high security
- Much communication through KDC
- KDC may be performance bottleneck



Threats to Authentication Algorithms In Distributed Systems

- Some threats:
 - **Subject Impersonation:** A mal actor may gain access to a particular workstation and pretend to be another user operating from that workstation.
 - **Address Impersonation:** A mal actor may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.
 - **Password stealing:**
 - **Using replay attacks:** A mal actor may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations.
 - **Installing Trojans on the server:** Login Trojan horse to steal password
 - **Impersonating Servers:** A bogus server may trick client into believing it is the right server
- **One solution:** Kerberos

What is Kerberos? (I)

History:

- MIT developed Kerberos to protect network services provided by Project Athena.
- The protocol was named after the Greek mythological character *Kerberos* (or *Cerberus*), known in Greek Mythology as being the *monstrous three-headed guard dog of Hades*.



Versions:

- Several versions of the protocol exist; versions 1–3 occurred only internally at MIT.
- Kerberos 4 was published late 1980s.
- Kerberos 5 was published in 1993 (RFC 1510)
- In 2005, modifications to Kerberos 5 was made (RFC4120).

What is Kerberos? (II)

Description:

- Kerberos uses as its basis the Needham-Schroeder protocol.
- It makes use of a trusted third party, a (KDC), which consists of
 - an Authentication Server (AS)
 - and a Ticket Granting Server (TGS).
- Kerberos works on the basis of "tickets" which serve to prove the identity of users.
- Kerberos maintains a database of secret keys; each entity on the network — whether a client or a server — shares a secret key known only to itself and to Kerberos.
- For communication between two entities, Kerberos generates a session key which they can use to secure their interactions.

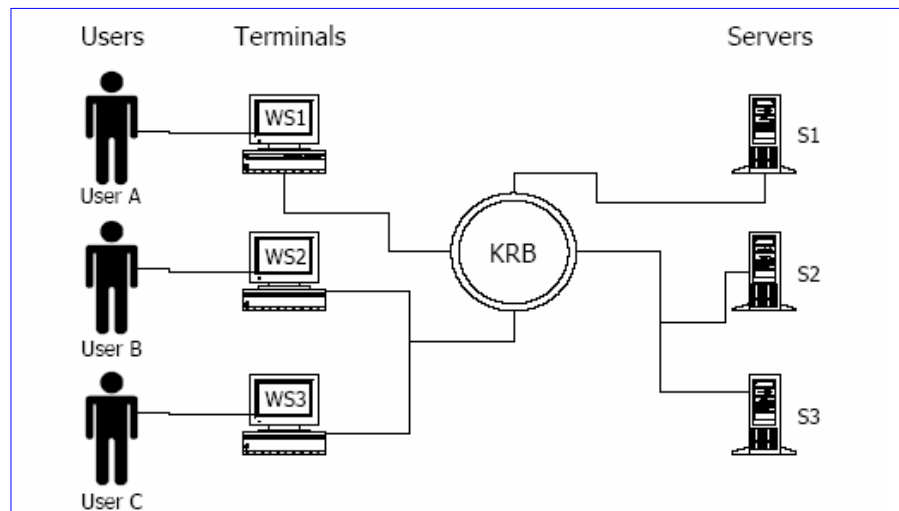
What Kerberos Provides

- A centralized authentication service
- Authenticate users to services
- Authenticate services to users
- Servers are relieved of the burden of maintaining authentication information
- Supports inter-server authentication

Kerberos Design Requirements

- **Secure**
 - A network eavesdropper should not be able to obtain the necessary to impersonate a user.
- **Reliable**
 - Kerberos should be highly available and should employ a distributed server architecture.
- **Transparent**
 - The user shouldn't be aware that authentication is taking place.
- **Scalable**
 - The system should be capable of supporting large numbers of clients and servers.

Kerberos Environment (I)



Kerberos Environment

- KRB consists of:
 - AS – Authentication Server
 - TGS – Ticket Granting Server
 - DB – Data Base of entity keys
- Separation between two actions:
 - Authentication – logging into the “network”
 - Communication – holding a session between two parties

Kerberos V5 Architecture (I)

- V5 improves over V4:
 - Standard message byte ordering
 - Multiple network address types
 - Multiple encryption algorithms
 - Arbitrary ticket lifetimes
 - Improved protocols
 - Authentication forwarding, proxying, postdating
 - Inter-realm authentication

Kerberos V5 Architecture (II)

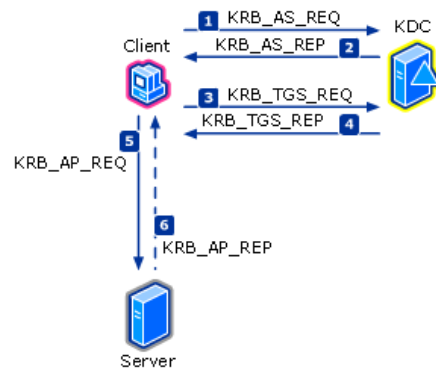
- Standard message byte ordering:
 - V4 message structure is determined by the sender
 - V5 messages are defined with Abstract Syntax Notation (ASN.1)
- Multiple network address types:
 - V4 uses IP addresses
 - V5 allows other types of addresses
- Multiple encryption algorithms:
 - V4 uses DES for encryption
 - V5 allows specifying different kinds of encryption algorithms
- Arbitrary ticket lifetimes:
 - V4 ticket lifetimes is limited by 1280 minutes
 - V5 ticket lifetimes can be arbitrarily long

Kerberos V5 Architecture (III)

- Authentication forwarding, proxying, and postdating:
 - V5 allows forwarding credentials from one client to another
 - An example where it might be used is when a user logs in to a remote system and wants authentication to work from that system as if the login were local.
 - V5 allows proxying of credentials
 - At times it may be necessary for a principal to allow a service to perform an operation on its behalf.
 - V5 allows issuing postdated tickets
 - Applications may occasionally need to obtain tickets for use much later, e.g., a batch submission system would need tickets to be valid at the time the batch job is serviced.
- Inter-realm authentication:
 - V5 includes “realm” in the authentication protocols
- Improved protocols:
 - V5 introduces a pre-authentication step between client and AS
 - V5 eliminates double encryption in messages
 - V5 changes the encryption mode (from PCBC to CBC)

Kerberos V5 Protocol Simplified (I)

- Kerberos Exchange



The RFC standard Kerberos version 5 authentication protocol communication sequences consist of six (five required and one optional) messages.

Kerberos V5 Protocol Simplified (IIa)

The Authentication Service Exchange

1. Kerberos authentication service request (KRB_AS_REQ)

The client contacts the Key Distribution Center's authentication service for a short-lived ticket (a message containing the **client's identity**, and **Workstation's identity** — in Windows its the SID) called a **ticket-granting ticket (TGT)**. This happens at logon. **Encryption using the hash of the password and a timestamp is sent if preauthentication is used**

2. Kerberos authentication service response (KRB_AS_REP)

The authentication service (AS) constructs the TGT and creates a session key the client can use to encrypt communication with the **ticket-granting service (TGS)**. The TGT has a limited lifetime. At the point that the client has received the TGT, the client has not been granted access to any resources, even to resources on the local computer.

Why use a TGT? Couldn't the AS simply issue a ticket for the target server?

Kerberos V5 Protocol Simplified (IIb)

- Yes, but if the AS issued tickets directly, the user would have to enter a password for every new server/service connection.
- Issuing a TGT with a short lifespan (typically 10 hours) gives the user a valid ticket for the ticket-granting service, which in turn issues target-server tickets.
- The TGT's main benefit is that the user only has to enter a password once, at logon.

Kerberos V5 Protocol Simplified (III)

The Ticket-Granting Service Exchange

3. Kerberos ticket-granting service request (KRB_TGS_REQ)

The client wants access to local and network resources. To gain access, the client sends a request to the TGS for a ticket for the local computer or some network server or service. This ticket is referred to as the service ticket or service ticket. To get the ticket, the client presents the TGT, an authenticator, and the name of the target server (the Server Principal Name or SPN).

4. Kerberos ticket-granting service response (KRB_TGS_REP)

The TGS examines the TGT and the authenticator. If these are acceptable, the TGS creates a service ticket. The client's identity is taken from the TGT and copied to the service ticket. Then the ticket is sent to the client.

Note:

The TGS cannot determine if the user will be able to get access to the target server. It simply returns a valid ticket. Authentication does not imply authorization.

Kerberos V5 Protocol Simplified (IV)

The Client/Server Exchange

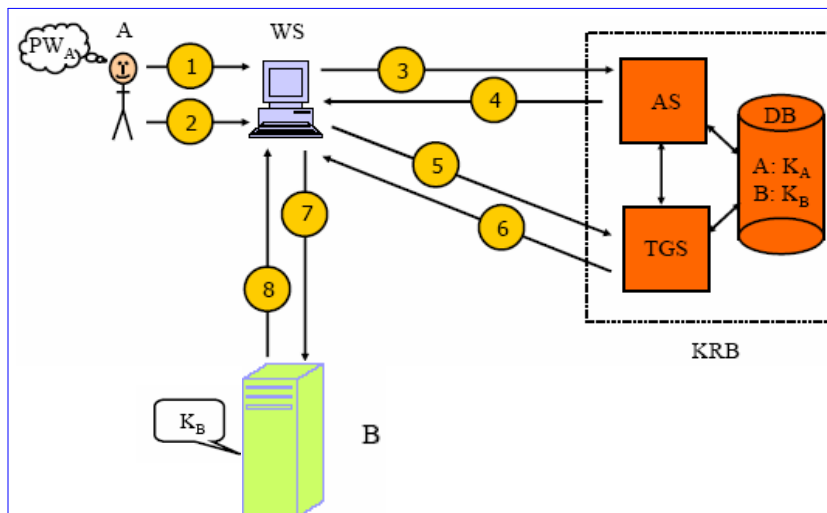
5. Kerberos application server request (KRB_AP_REQ)

After the client has the **service ticket**, the client sends the ticket and a new **authenticator** to the target server, requesting access. The server will decrypt the ticket, validate the authenticator, and for Windows services, create an access token for the user based on the SIDs in the ticket.

6. Kerberos application server response (optional) (KRB_AP_REP)

Optionally, the client might request that the target server verify its own identity. This is called mutual authentication. If mutual authentication is requested, the target server will take the **client computer's timestamp** from the authenticator, encrypt it with the **session key** the TGS provided for client-target server messages, and send it to the client.

Kerberos V5 Protocol In Depth (I)



Kerberos V5 Protocol In Depth (II)

Acquiring Network Credentials:

- ① User A sits at workstation WS and enters his name “A”.
At this point the new entity “A+WS” is created.
- ② User A enters his password PW_A .
Workstation WS computes key $K_A = h(PW_A)$ and erases password PW_A from its memory.
Workstation can optionally compute some pre-authentication data from the user’s password. (MS implementation is disclosed by Brezak).

Kerberos V5 Protocol In Depth (III)

Acquiring Network Credentials (continued):

- ③ Workstation WS contacts the Authentication Server (AS) on behalf of user A and requests “Network Credentials” to the entity A+WS.
Workstation WS sends to AS the following data in the clear:
 $\langle A, WS, Realm_A, TGS, Times_1, Nonce_1 \rangle$, where $Times_1$ gives the time validity interval, and $Nonce_1$ is random value.
There is also an option of sending the pre-authentication data.

Authentication Server AS checks whether user A is permitted to log in to the network from workstation WS, and if so it replies to A+WS with the following two items:
 - (1) $\langle Realm_A, A, WS, TKT_{TGS} \rangle$
 - (2) $E_{K_A} (\langle K_{A,TGS}, Times_2, Nonce_1, Realm_{TGS}, TGS \rangle)$
where $TKT_{TGS} = E_{K_{KRB}} (\langle K_{A,TGS}, Realm_A, A, WS, Times_2 \rangle)$.
Workstation WS decrypts the encrypted item (2) with key K_A .

Kerberos V5 Protocol In Depth (IV)

- Establishing Connection with Server:

- 5 When user A wishes to get service from server B, workstation WS contacts the Ticket Granting Server (TGS) and requests a ticket for server B. The request consists of:

$\langle B, \text{Times}_3, \text{Nonce}_3, \text{TKT}_{\text{TGS}}, \text{Auth}_3 \rangle$, where

$\text{Auth}_3 = E_{K_{A,\text{TGS}}} (\langle A, \text{WS}, \text{Realm}_A, \text{Timestamp}_3 \rangle)$.

- 6 Ticket Granting Server TGS checks whether user A is permitted to get service to server B from workstation WS, and if so it replies to A+WS with the following two items:

(1) $\langle \text{Realm}_A, A, \text{WS}, \text{TKT}_B \rangle$

(2) $E_{K_{A,\text{TGS}}} (\langle K_{A,B}, \text{Times}_4, \text{Nonce}_3, \text{Realm}_B, B \rangle)$

where $\text{TKT}_B = E_{K_B} (\langle K_{A,B}, \text{Realm}_A, A, \text{WS}, \text{Times}_4 \rangle)$.

Kerberos V5 Protocol In Depth (V)

- Establishing Connection with Server (continued):

- 7 Workstation WS decrypts the encrypted item (2) sent by the TGS with key $K_{A,\text{TGS}}$.

Workstation WS contacts server B and requests to hold a session by sending the following:

$\langle \text{TKT}_B, \text{Auth}_5 \rangle$, where

$\text{Auth}_5 = E_{K_{A,B}} (\langle A, \text{WS}, \text{Realm}_A, \text{Timestamp}_5, \text{Subkey}, \text{Seq\#} \rangle)$.

The fields Subkey and Seq# are optional.

- 8 Server B decrypts the encrypted ticket TKT_B with the key K_B . Server B decides whether it wishes to provide service to A+WS, and if so it replies to A+WS with the following authentication data:

$\langle \text{Auth}_6 \rangle = E_{K_{A,B}} (\langle \text{Timestamp}_5, \text{Subkey}, \text{Seq\#} \rangle)$.