Module 4

① Explain Remote user authentication principles.

→ 1) User authentication is a fundamental process of verifying, the identity of a system entity claiming a specific identity.

2) User authentication is the basis for more type of access control and for user accountability.

3) RFC 2828 dyines user authentication as the process of verifying an identity claimed by or for a system entity.

4) An authentication process consists of two types:

a) Identification step: Presenting an identifier to the security system.

b) Verification step: Presenting or generating authentication information that corroborates the binding between the entity and the identifier.

5) There are four general means of authenticating a user's identity, which can be used alone or in combination.

1) Something individual knows: Examples include a password, PIN, or answers to a prearranged set of questions.

2) Something individual possesses: Example include electronic keycards, smart cards, and physical keys.

3) Something the individual is (static biometric): Examples include by recognition by fingerprint, retina and face.

4) Something the individual does (dynamic biometric): Examples include recognition by voice patterns, handwriting characteristics and typing rhythm.

② Explain Remote user authentication using asymetric encryption. ②

→ Mutual Authentication:

1) Mutual Authentication or two way authentication refers to two parties authenticating each other at same time in an authentication protocol.

2) This protocol assumes that each of the two parties is in possesion of the current public key of other.

3) This protocol uses timestamps.

4) Timestamps: Party A accepts a message as fresh only if the message contains a timestamp that in A's judgment, is close enough to A's knowledge of current time.

Working

1) $A \rightarrow AS: ID_A \| ID_B$

2) $AS \rightarrow A: E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T])$

3) $A \rightarrow B: E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T]) \| E(PU_b, E(PR_a, [K_s \| T]))$

1) A protocal using timestamps is provided in that uses a central system, referred to as an authentication server (AS), because it is not actually responsible for secret key distribution. Rather the AS provides public key certificate.

2) The session key is chosen and encrypted by A, hence there is no risk of exposure by the AS.

3) The timestamps protect against replays of compromised keys. This protocol is compact but as before requires synchronization of clocks.

One way authentication:

They focus on two main areas, when communication on non-secure network

1) confidentiality
2) Authentication.

1) If confidentiality is the primary concern, then the following may be more efficient. (3)

$A \rightarrow B: E(PU_b, K_s) \,||\, E(K_s, M)$

* The message is encrypted with a one time secret key.
* A also encrypts this one-time key with B's public key.
* Only B will be able to use the corresponding private keys to recover the one time key and then use that key to decrypt the message.

2) If authentication is the primary concern, then a digital signature may suffice.

$A \rightarrow B: M \,||\, E(PR_a, H(M))$.

To counter such scheme both the message and signature can be encrypted with recipients public key.

$A \rightarrow B: E(PU_b, [M \,||\, E(PR_a, H(M))])$

↓

$A \rightarrow B: M \,||\, E(PR_a, H(M)) \,||\, E(PR_{as}, [T \,||\, ID_A \,||\, PU_a])$

↓

$A \rightarrow B: E(PU_b, (M \,||\, E(PR_a, H(M)) \,||\, E(PR_{as}, [T \,||\, ID_A \,||\, PU_a]))$

(3) What is kerberos? With a neat diagram explain an overview of Kerberos.

→ Kerberos is an authentication service developed as part of project Athena at MIT, and is one of the best known and most widely implemented trusted third party key distribution systems.

Kerberos provide a centralized authentication server whose function is to authenticate users to servers and servers to users.

1) User log on to workstation and requests service on host.

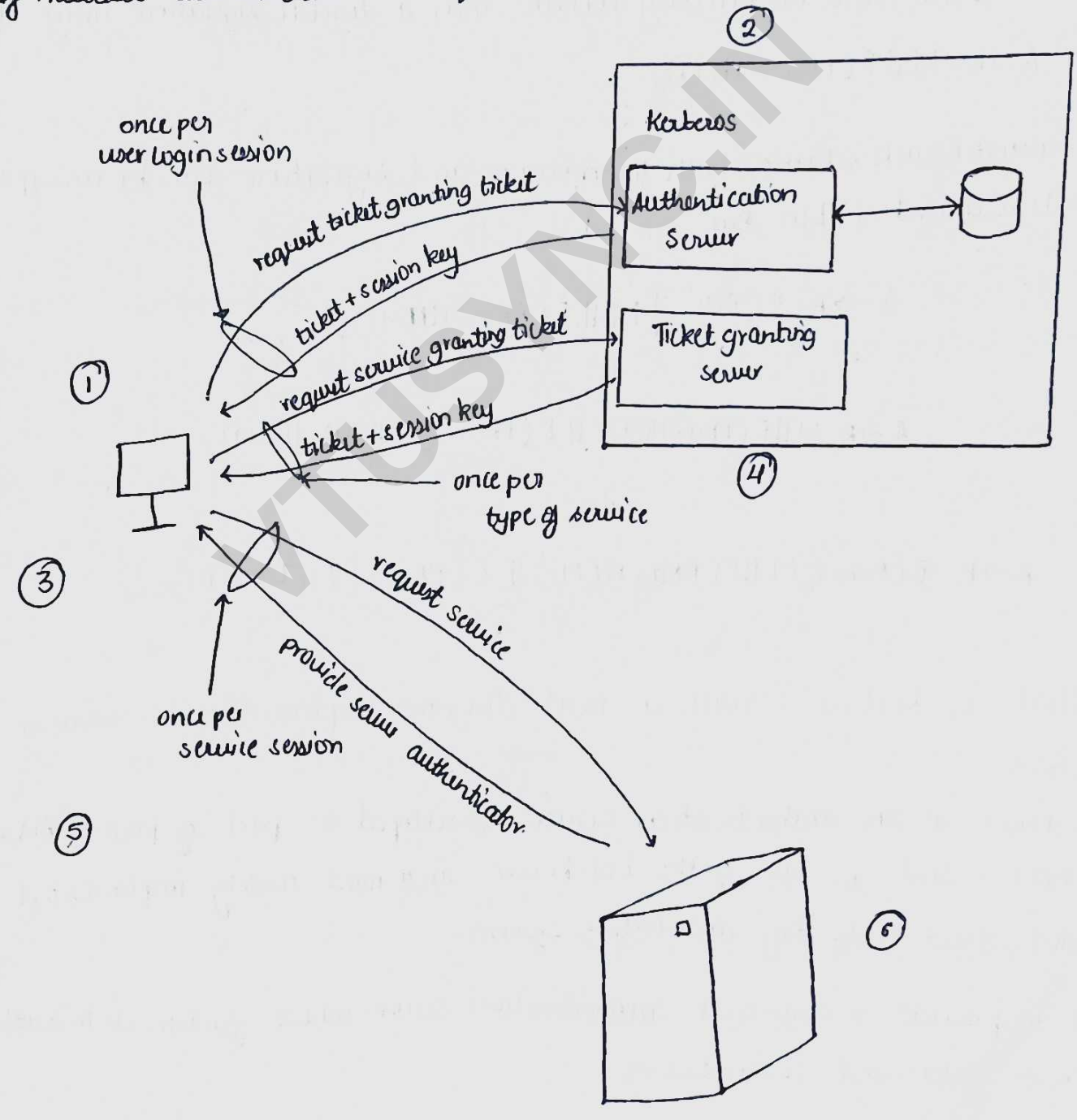
2) AS verifies user's access right in database, creates a ticket granting ticket and session key. Results are encrypted using key derived from user's password.

3) Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user name, network address and time to TGS.

4) TGS decrypt ticket and authenticator, verifies request, then creates ticket for requested server.

5) Workstation sends ticket and authenticator to server.

6) Server verifies that ticket and authenticator match, then grant access to service. If mutual authentication is required, server returns an authenticator.



(4) Explain Kerberos authentication service with version 4 dialouge.

→

1) Kerberos V4 is a basic third party authentication scheme.

2) The core of Kerberos is the Authentication server (AS) and Ticket granting servers (TGS). these are trusted by all users and servers and must be securely administred.

3) The protocol include a sequence of interactions between the client AS, TGT and desired server. Version 4 of Kerberos makes use of DES, in a rather elaborate protocol, to provide authentication service.

4) The heart of the first problem is the lifetime associated with the ticket granting ticket.

5) If the lifetime is very short, then the user will be repeatedly asked for a password. If the lifetime is long then the opponent has a greater opportunity for replay. Similarly, if an opponent captures a service granting ticket and uses it before it expires, the opponent has acess to corresponding service.

6) The second problem is that there may be a requirement for servers to authenticate themselves to users.

7) First consider the problem of captured ticket-granting tickets and need to determine that the ticket presenter is the same as the client for whom the ticket was issued.

5) Explain the exchange of message involved in Kerberos version 4.

→

1) Authentication Service exchange to obtain ticket granting ticket

a) $C \rightarrow AS$: $ID_c \| ID_{tgs} \| TS_1$

b) $AS \rightarrow C$: $E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_c \| AD_c \| ID_{tgs} \| TS_2 \| Lifetime_2])$

2) Ticket granting service exchange to obtain service granting ticket.

c) $C \rightarrow TGS$: $ID_v \| Ticket_{tgs} \| Authenticator_c$

d) $TGS \rightarrow C$: $E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_4])$

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_c \| AD_c \| ID_{tgs} \| TS_2 \| Lifetime_2])$

$Ticket_v = E(K_v, [K_{c,v} \| ID_c \| AD_c \| ID_v \| TS_4 \| Lifetime_4])$

$Authenticator_c = E(K_{c,tgs}, [ID_c \| AD_c \| TS_5])$

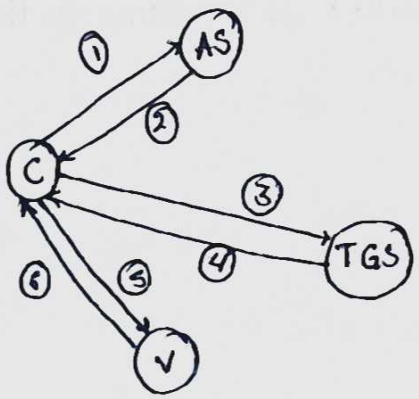
3) Client / server authentication exchange to obtain service.

e) $C \rightarrow V$ $Ticket_v \| Authenticator_c$

f) $V \rightarrow C$ $E(K_{c,v}, [TS_5 + 1])$

$Ticket_v$: $E(K_v, [K_{c,v} \| ID_c \| AD_c \| ID_v \| TS_4 \| Lifetime_4])$

$Authenticator_c = E(K_{c,v}, [ID_c \| AD_c \| TS_5])$



(6) What are the principal difference between version 4 and 5 of Kerberos

→

Version 4	Version 5
1) Use DES exclusively, which has security limitation	Support multiple encryption algorithm making more secure and flexible
2) Fixed format ticket with limited fields making extension difficult	Uses a more flexible ticket format with ASN.1 encoding
3) Limited to IPv4 addresses	Supports both IPv4 and IPv6 addresses
4) Limited support for cross-realm authentication	Stronger cross realm authentication.
5) Fixed ticket lifetimes, requiring user to log in again after expiration	Allows renewable tickets, reducing the need for frequent re-authentication.

(7) Web Security consideration

→ The world wide web operates on a client server model over the Internet and TCP/IP intranets making it vulnerable to security threats.

Key consideration

1) Complex software : web application and servers have hidden vulnerabilities that can be exploited.
2) Server exploitation : A compromised web server can be a gateway to attack an organizations internal system.
3) User awareness : Many user are unaware of security risk, making them susceptible to attacks.

2) Web Security Threats:

1) Passive Attacks: Eaves dropping on network traffic, unauthorized access to restricted information

2) Active attacks: Impersonation, message alteration

3) Threat Location: webserver, web browser and network traffic between them.

3) ~~User~~ Web security Approaches:

1) Encryption: HTTPS to secure communication.

2) Authentication: Multifactor authentication, strong password.

3) firewall and IDS: Monitoring and filtering traffic for threats

4) Content security Policies: Preventing XSS and other web based attacks.

(8) Email threats and comprehensive email security.

→ Email is a widely used communication tool but is vulnerable to various security threats. These threats can be classified into four categories

1) Authenticity-related threats: Unauthorized access to an email system, leading to identity theft or phishing attack.

2) Integrity related threats: Unauthorized modification of email content, such as email tampering, man in middle attacks.

3) Confidentially related threats: Exposure of sensitive information due to email interception or data leaks.

4) Availability related threats: Disruption preventing users from sending or receiving emails such as denial of service (DOS) attack.

Mitigation Techniques for email security.

1) STARTTLS: Encrypt SMTP communication using TLS to ensure secure email transmission
2) S/MIME: Provides encryption and digital signatures to secure email content.
3) DNSSEC: Ensure authentication and integrity of DNS records.
4) DANE: Strengthens authentication by verifying public through DNSSEC
5) SRF : Prevent email spoofing by verifying the senders domain through DNS record.
6) DKIM : Uses cryptographic signatures to verify email authenticity and integrity.
7) DMARC: Helps domain owners enforce SPF and DKIM policies while providing visibility into unauthorized email usage.