

Overview

Motivation: "single signon" in TCP/IP LAN environment
 Developed at MIT as part of project Athena, from mid 1980's
 Shared key authentication and key distribution protocol, based on
 Needham Schroeder

Kerberos

Kerberos vision: Single Signon

Logon once, terminal acts as proxy for authentication to other servers
 Kerberizing applications = modifying the application to use Kerberos
 routines for authentication

Examples:

1. telnet
2. rlogin
3. ftp
4. NFS (Network File system)

A typical day at the office

1. logon to Unix desktop (enter uname, password)
2. connect laptop to network
3. ftp from desktop to laptop to transfer files (enter uname password)
4. connect to windows server to read Word attachment in email (enter uname password)
5. connect to fast server to run computation intensive simulation (enter uname password)
6. etc ...

Kerberos version 4

Baseline assumptions

Server S = Key distribution center = KDC has database of (User, $K_{User,S}$)

$K_{User,S}$ is a hash of User's password

K_S is server's key, known to server only

Naive approach:

user logs on to terminal with password

terminal remembers password

terminal uses password to authenticate user to other services

Problem:

User may run insecure software that transfers password

Solution:

Minimize the time that terminal remembers password

authenticate to other services using an authentication "ticket" that expires (e.g. after 1 day), in place of password.

On inclusion of network addresses in tickets

$n(T)$ included so that

1. replay attacks from different machines can be detected (forcing attackers to spoof network addresses)
2. Alice cannot delegate her ticket to other users machines (removed in V5)

Kerberos Version 5

Phase 1: obtain a Ticket Granting Ticket

Motivation: minimize amount KDC needs to remember, allow KDC to be distributed

(Alice enters password into terminal T)

1. $T \rightarrow S$: Alice requests $TGT, n(T)$
2. $S \rightarrow T$: $\{K_A, TGT\}_{K_{A,S}}$

$TGT = \{A, n(T), v, K_A\}_{K_S}$ used to fetch tickets for Alice's use

$n(T)$ = network address of T

K_A is session key for use by A

v = validity period of ticket

Phase 2: Logging into a remote machine Bob

(Alice enters "login Bob")

3. $T \rightarrow S$: $A, B, TGT, \{A, n(T), timestamp\}_{K_A}$ (where $TGT = \{A, n(T), v, K_A\}_{K_S}$, as above)
4. $S \rightarrow T$: $\{B, K_{A,B}, ticket_{A,B}\}_{K_A}$ (Where $ticket_{A,B} = \{A, n(T), v', K_{A,B}\}_{K_{B,S}}$)
5. $T \rightarrow B$: $ticket_{A,B}, \{timestamp\}_{K_{A,B}}$
1. (Bob verifies timestamp is current)
6. $B \rightarrow T$: $\{timestamp + 1\}_{K_{A,B}}$

Delegation

Examples:

Alice starts a batch job to run overnight, delegates to it the right to access Alice's files on other machines

Alice logs on to mozart, then wants to log on to wagner from there

Approach: Allow Alice to request ticket granting tickets with

different network addresses, for specific services

Alice permitted to give a ticket with Bob's network address to Bob

Authorization-data field encodes restrictions on Bob, application

specific interpretation

Boolean Fields in TGT's:

1. *Forwardable:* Alice may use this TGT to request a TGT with

Bob's network address

2. *Proxiable:* Alice may use this TGT to request tickets that she

can give to Bob to access services on behalf of Alice

Other Differences in Protocol Message Structure

V4: step 2. $S \rightarrow T: \{K_A, TGT\}_{K_{A,S}}$

V5: step 2. $S \rightarrow T: \{K_A\}_{K_{A,S}}, TGT$

V4 step 4. $S \rightarrow T: \{B, K_{A,B}, ticket_{A,B}\}_{K_A}$

V5 step 4. $S \rightarrow T: \{B, K_{A,B}\}_{K_A}, ticket_{A,B}$

reason: not necessary to double encrypt tickets

Crypto in Kerberos implementations:

DES, 3-DES in use

work on inclusion of AES in progress

Potential Vulnerabilities

(see Bellare and Merritt)

can replay old tickets: timestamps are supposed to prevent this, but replays can be done during lifetime of ticket

ticket lifetimes tend to be long. e.g. 8 hours.

network clock protocols tend to be insecure, so clock attacks are a

danger

attacker can mount password guessing attack after collecting enough tickets

attacker with root access on client machine can modify Kerberos

software on client to record passwords