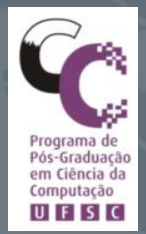
Advanced Security Protocols - Kerberos

Design and Verification of Security Protocols and Security Ceremonies

Programa de Pós-Graduacão em Ciências da Computacão
Dr. Jean Everson Martina

August-November 2016



UNIVERSIDADE FEDERAL
DE SANTA CATARINA

What is Kerberos?

- Kerberos is an authentication protocol that works on the basis of 'tickets' to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner;

What is Kerberos?

- Kerberos is an authentication protocol that works on the basis of 'tickets' to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner;
- Kerberus (Cerberus) ferocious three-headed guard dog of Hades (hellhound).

What is Kerberos?

- Kerberos is an authentication protocol that works on the basis of 'tickets' to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner;
- Kerberus (Cerberus) ferocious three-headed guard dog of Hades (hellhound).

Where you know Kerberos from?

- Some will know it from operating systems authentication point of view;

Where you know Kerberos from?

- Some will know it from operating systems authentication point of view;
- Others will know it as Fluffy.

Where you know Kerberos from?

- Some will know it from operating systems authentication point of view;
- Others will know it as Fluffy.

Kerberos - History

- Massachusetts Institute of Technology (MIT) developed Kerberos to protect network services provided by Project Athena;

Kerberos - History

- Massachusetts Institute of Technology (MIT) developed Kerberos to protect network services provided by Project Athena;
- The protocol is based on the earlier Needham–Schroeder symmetric key protocol;

Kerberos - History

- Massachusetts Institute of Technology (MIT) developed Kerberos to protect network services provided by Project Athena;
- The protocol is based on the earlier Needham–Schroeder symmetric key protocol;
- Several versions of the protocol exist; versions 1–3 occurred only internally at MIT.

Kerberos - History

- Kerberos version 4 primarily was designed by Steve Miller and Clifford Neuman;

Kerberos - History

- Kerberos version 4 primarily was designed by Steve Miller and Clifford Neuman;
- Published in the late 1980s;

Kerberos - History

- Kerberos version 4 primarily was designed by Steve Miller and Clifford Neuman;
- Published in the late 1980s;
- Neuman and Kohl published version 5 in 1993 with the intention of overcoming existing limitations and security problems;

Kerberos - History

- Kerberos version 4 primarily was designed by Steve Miller and Clifford Neuman;
- Published in the late 1980s;
- Neuman and Kohl published version 5 in 1993 with the intention of overcoming existing limitations and security problems;
- Version 5 appeared as RFC 1510, and was made obsolete by RFC 4120 in 2005.

Kerberos - Provisions

- Users wish to access services on servers.;

Kerberos - Provisions

- Users wish to access services on servers.;
- Provides a centralized authentication server to authenticate users to servers and servers to users;

Kerberos - Provisions

- Users wish to access services on servers.;
- Provides a centralized authentication server to authenticate users to servers and servers to users;
- Relies on conventional encryption, making no use of public-key encryption.

Kerberos - Description

- The client authenticates itself to the Authentication Server (AS) which forwards the username to a key distribution center (KDC);

Kerberos - Description

- The client authenticates itself to the Authentication Server (AS) which forwards the username to a key distribution center (KDC);
- The KDC issues a ticket-granting ticket (TGT), which is time stamped, encrypts it using the user's password and returns the encrypted result to the user's workstation;

Kerberos - Description

- The client authenticates itself to the Authentication Server (AS) which forwards the username to a key distribution center (KDC);
- The KDC issues a ticket-granting ticket (TGT), which is time stamped, encrypts it using the user's password and returns the encrypted result to the user's workstation;
- This is done infrequently, typically at user logon;

Kerberos - Description

- The client authenticates itself to the Authentication Server (AS) which forwards the username to a key distribution center (KDC);
- The KDC issues a ticket-granting ticket (TGT), which is time stamped, encrypts it using the user's password and returns the encrypted result to the user's workstation;
- This is done infrequently, typically at user logon;
- The TGT expires at some point, though may be transparently renewed by the user's session manager while they are logged in.

Kerberos - Description

- When the client needs to communicate with another principal the client sends the TGT to the ticket-granting service (TGS), which usually shares the same host as the KDC;

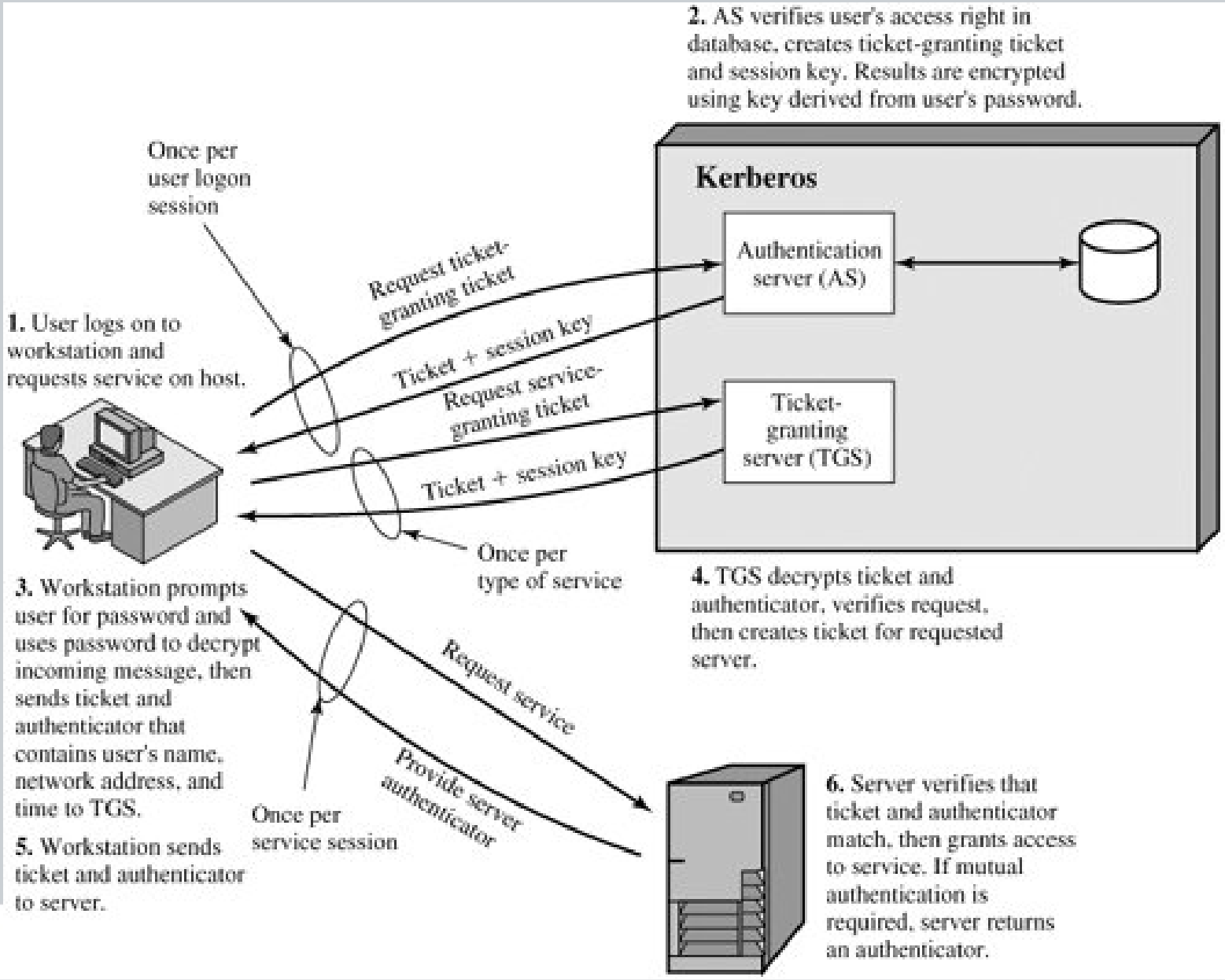
Kerberos - Description

- When the client needs to communicate with another principal the client sends the TGT to the ticket-granting service (TGS), which usually shares the same host as the KDC;
- After verifying the TGT is valid and the user is permitted to access the requested service, the TGS issues a ticket and session keys, which are returned to the client;

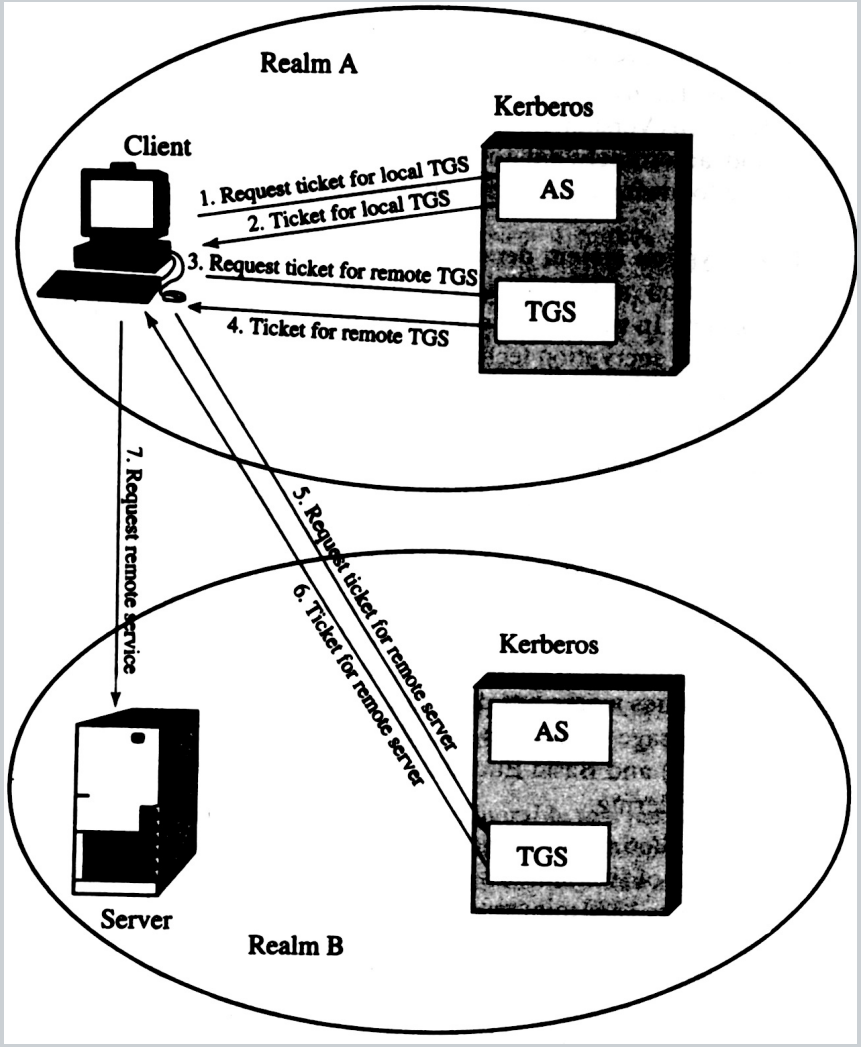
Kerberos - Description

- When the client needs to communicate with another principal the client sends the TGT to the ticket-granting service (TGS), which usually shares the same host as the KDC;
- After verifying the TGT is valid and the user is permitted to access the requested service, the TGS issues a ticket and session keys, which are returned to the client;
- The client then sends the ticket to the service server (SS) along with its service request.

Kerberos - Description



Kerberos - Inter-Realms



Kerberos - Client Authentication

1. C $\rightarrow$ AS: C, TGS, TS_1

Kerberos - Client Authentication

1. C $\rightarrow$ AS: C, TGS, TS_1
2. AS $\rightarrow$ C: $E(K_C, (K_{C,TGS}, TS_2, TTL, Ticket_{TGS}))$

Kerberos - Client Authentication

1. C $\rightarrow$ AS: C, TGS, TS_1
2. AS $\rightarrow$ C: $E(K_C, (K_{C,TGS}, TS_2, TTL, Ticket_{TGS}))$

$$Ticket_{TGS} = E(K_{TGS}, (K_{C,TGS}, C, AD_C, TGS, TS_2, TTL))$$

- Everything is garanteed by K_C and K_{TGS};

Kerberos - Client Authentication

1. C $\rightarrow$ AS: C, TGS, TS_1
2. AS $\rightarrow$ C: $E(K_C, (K_{C,TGS}, TS_2, TTL, Ticket_{TGS}))$

$$Ticket_{TGS} = E(K_{TGS}, (K_{C,TGS}, C, AD_C, TGS, TS_2, TTL))$$

- Everything is garanteed by K_C and K_{TGS};
- Everything is timestamped;

Kerberos - Client Authentication

1. C $\rightarrow$ AS: C, TGS, TS_1
2. AS $\rightarrow$ C: $E(K_C, (K_{C,TGS}, TS_2, TTL, Ticket_{TGS}))$

$$Ticket_{TGS} = E(K_{TGS}, (K_{C,TGS}, C, AD_C, TGS, TS_2, TTL))$$

- Everything is garanteed by K_C and K_{TGS};
- Everything is timestamped;
- It is built to reduce load on AS.

Kerberos - Authorisation Service

1. C $\rightarrow$ TGS: V, $Ticket_{TGS}$, $Authenticator_C$

Kerberos - Authorisation Service

1. C $\rightarrow$ TGS: V, $Ticket_{TGS}$, $Authenticator_C$
2. TGS $\rightarrow$ C: $E(K_{C,V}, (V, TS_4, TTL_4, Ticket_V)$

Kerberos - Authorisation Service

1. C $\rightarrow$ TGS: $V, Ticket_{TGS}, Authenticator_C$
2. TGS $\rightarrow$ C: $E(K_{C,V}, (V, TS_4, TTL_4, Ticket_V)$

$$Ticket_V = E(K_V, (K_C, V, C, AD_C, V, TS_4, TTL_4))$$

Kerberos - Authorisation Service

1. C $\rightarrow$ TGS: $V, Ticket_{TGS}, Authenticator_C$
2. TGS $\rightarrow$ C: $E(K_{C,V}, (V, TS_4, TTL_4, Ticket_V)$

$$Ticket_V = E(K_V, (K_C, V, C, AD_C, V, TS_4, TTL_4))$$
$$Authenticator_C = E(K_{C,TGS}(C, AD_C, TS_3))$$

- K_V is shared between V e TGS
- $TS_4 + TTL_4 < TS + TTL$

Kerberos - Service Request

1. C $\rightarrow$ V: $Ticket_V, Authenticator_C$

Kerberos - Service Request

1. C $\rightarrow$ V: $Ticket_V, Authenticator_C$
2. V $\rightarrow$ C: $E(K_{C,V}, TS_5 + 1)$

Kerberos - Service Request

1. C $\rightarrow$ V: $Ticket_V, Authenticator_C$
2. V $\rightarrow$ C: $E(K_{C,V}, TS_5 + 1)$

$$AuthenticatorC = E(K_{C,V}, (C, AD_C, TS_5))$$

- $TS_5 + 1$ is for mutual authentication.

Kerberos – Problems and Limitations

- Single Point of Failure;

Kerberos – Problems and Limitations

- Single Point of Failure;
- Clock-sync is required due to timestamps;

Kerberos – Problems and Limitations

- Single Point of Failure;
- Clock-sync is required due to timestamps;
- Difficult inter-realm authentication;

Kerberos – Problems and Limitations

- Single Point of Failure;
- Clock-sync is required due to timestamps;
- Difficult inter-realm authentication;
- Interoperability problems due to loads of different ciphers.

Kerberos – Problems and Limitations

- Single Point of Failure;

Kerberos – Problems and Limitations

- Single Point of Failure;
- Clock-sync is required due to timestamps;

Kerberos – Problems and Limitations

- Single Point of Failure;
- Clock-sync is required due to timestamps;
- Difficult inter-realm authentication;

Kerberos – Problems and Limitations

- Single Point of Failure;
- Clock-sync is required due to timestamps;
- Difficult inter-realm authentication;
- Interoperability problems due to loads of different ciphers.

Kerberos – Verification

- Done mostly by Bella and Paulson;

Kerberos – Verification

- Done mostly by Bella and Paulson;
- It was crucial to the understanding of some concepts like goal availability;

Kerberos – Verification

- Done mostly by Bella and Paulson;
- It was crucial to the understanding of some concepts like goal availability;
- Proofs took around a year to be build the first time;

Kerberos – Verification

- Done mostly by Bella and Paulson;
- It was crucial to the understanding of some concepts like goal availability;
- Proofs took around a year to be build the first time;
- It was part of a big EPSRC project;

Kerberos – Verification

- Done mostly by Bella and Paulson;
- It was crucial to the understanding of some concepts like goal availability;
- Proofs took around a year to be build the first time;
- It was part of a big EPSRC project;
- The verification demonstrates that some methods have a lot of difficulty on dealing with time-stamps and counting over channels.

Discussion

- Have you ever used Kerberos?

Discussion

- Have you ever used Kerberos?
- Do you know the genealogy of Kerberos?

Discussion

- Have you ever used Kerberos?
- Do you know the genealogy of Kerberos?
- How can we overcome the shortcomings of Kerberos?

Questions????

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

UNIVERSIDADE FEDERAL
DE SANTA CATARINA