Kerberos introduction

Kerberos

What is Kerberos?

Kerberos is a computer network authentication protocol which 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.

[Wikipedia]

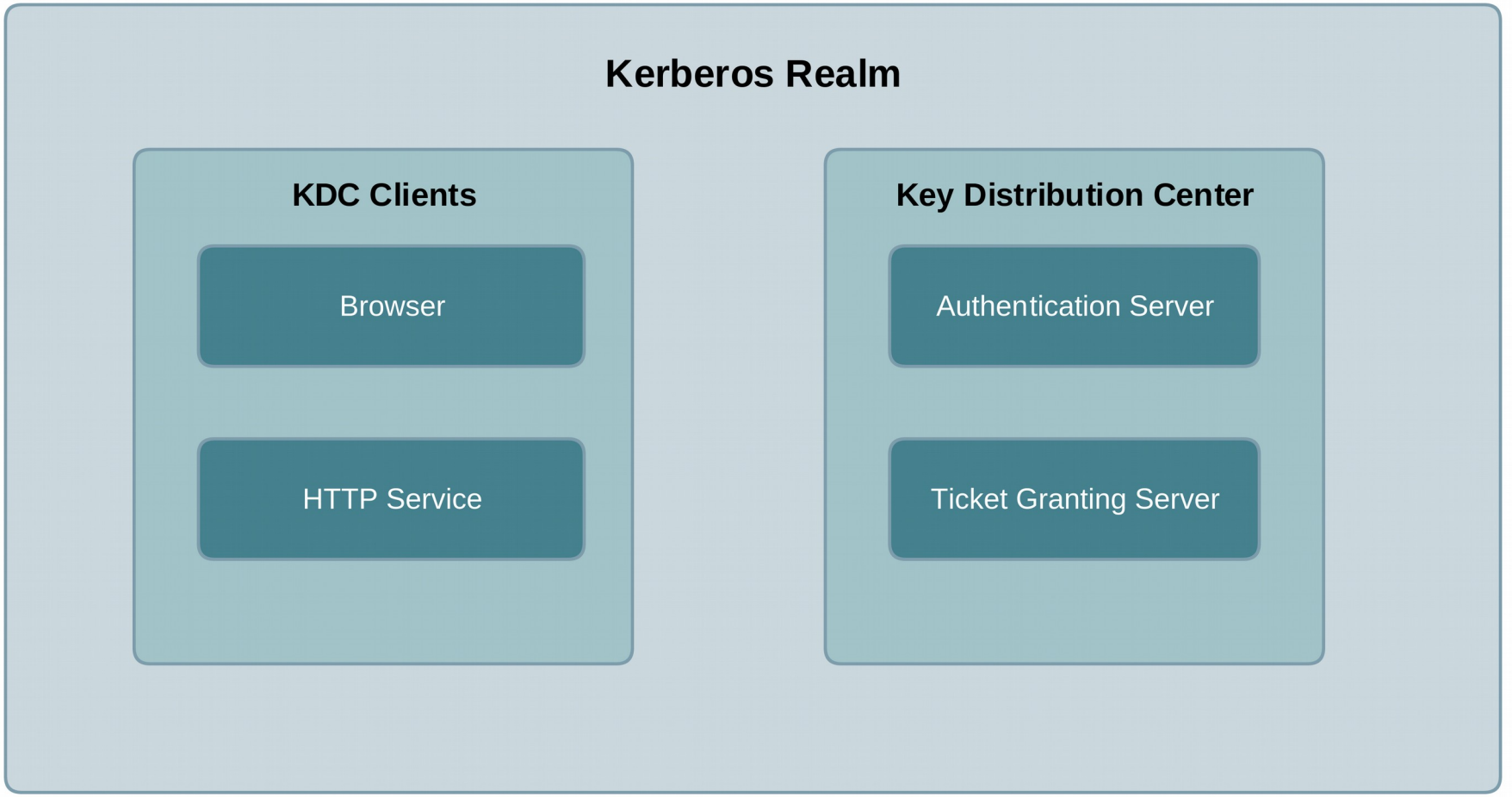
What is Kerberos?

- Developed on Massachusetts Institute of Technology (MIT)
- Windows 2000 and later use Kerberos as a default authentication mechanism (although uses its own implementation)
- Kerberos is also built into Active Directory solution
- UNIX-based operating systems have tools for authenticating over Kerberos

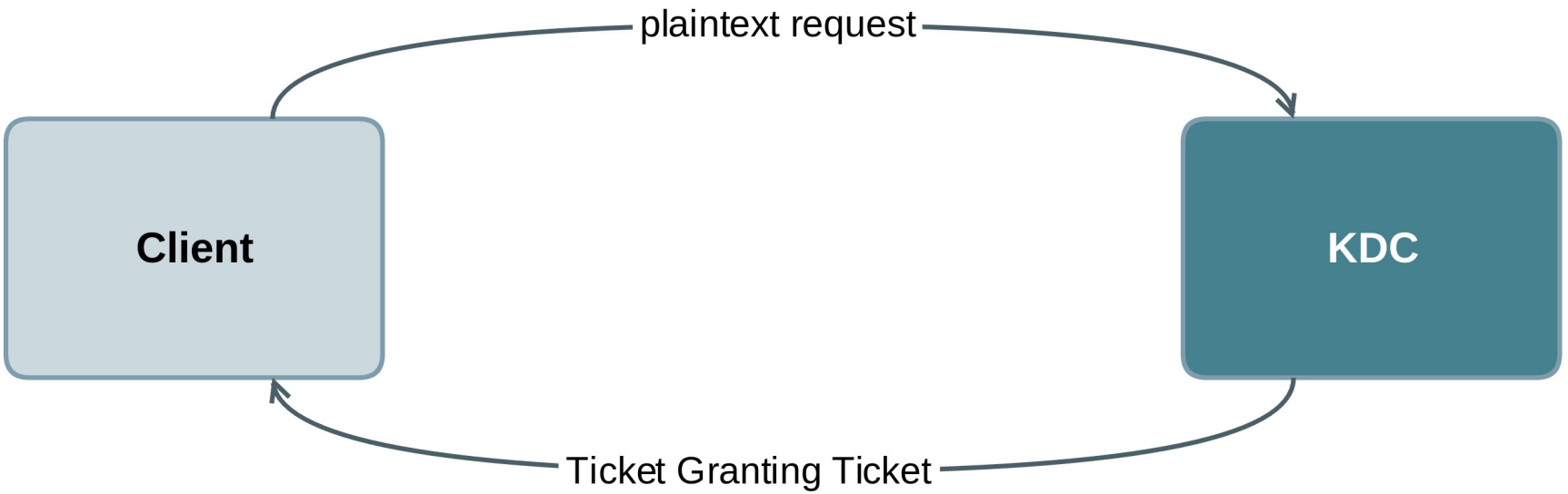
Kerberos use cases

- Authentication in Windows systems
- Providing SSO service in untrusted networks

Kerberos architecture



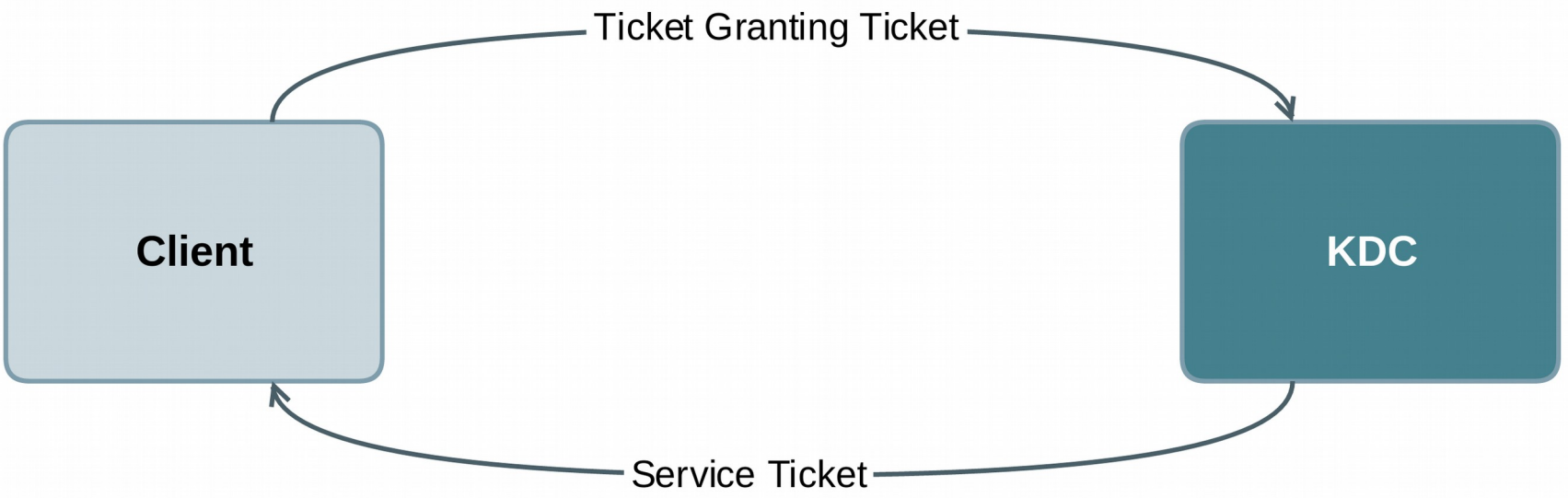
Kerberos negotiation process – part I



Kerberos negotiation process – part I

- Client sends a message with both encrypted and unencrypted parts
- Interestingly, unencrypted part caries all information about user
- Encrypted part is part of the protocol itself
- KDC lookups the user in its database
- If lookup is successful, KDC issues Ticket Granting Ticket signed by KDC's private key

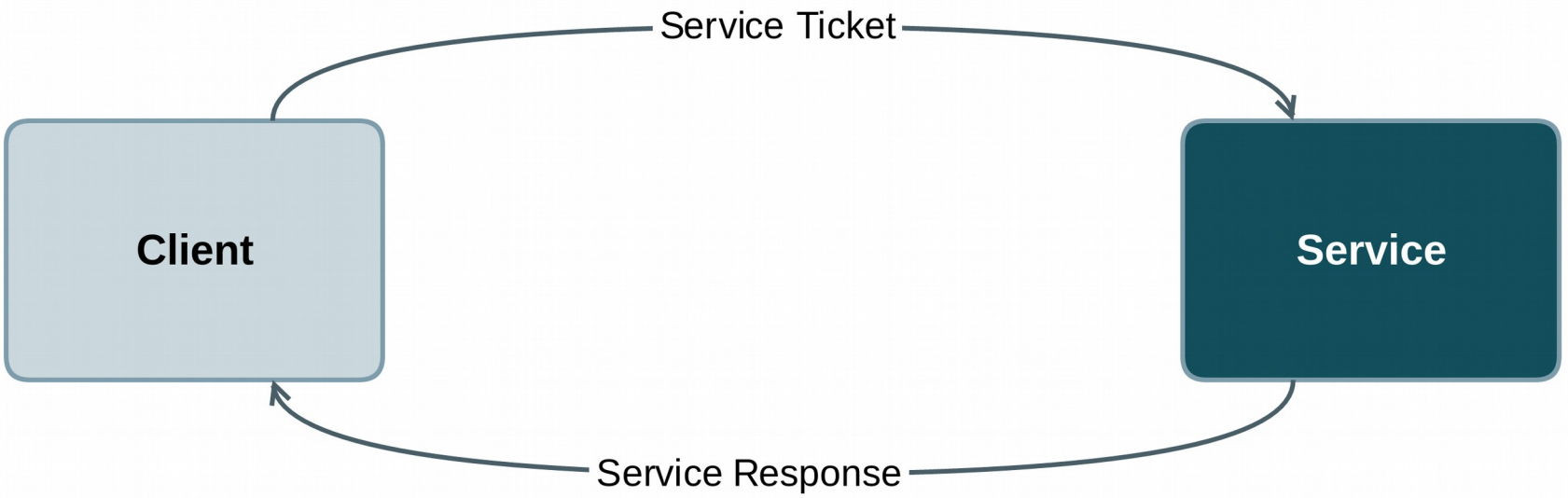
Kerberos negotiation process – part II



Kerberos negotiation process – part II

- To be able to authenticate with a service, Client sends the TGT with Service ID to KDC
- KDC checks Service ID in its database
- KDC decrypts TGT and if it's successful, Client is authorized to by granted access to Service
- KDC issues a Service Ticket signed by service's own public key
- This is the moment when a trust between Client and Service is established

Kerberos negotiation process – part III



Kerberos negotiation process – part III

- Client sends the encrypted Service Ticket to Service
- Service decrypts Service Ticket, if successful then Client is authorized to make a request to Service

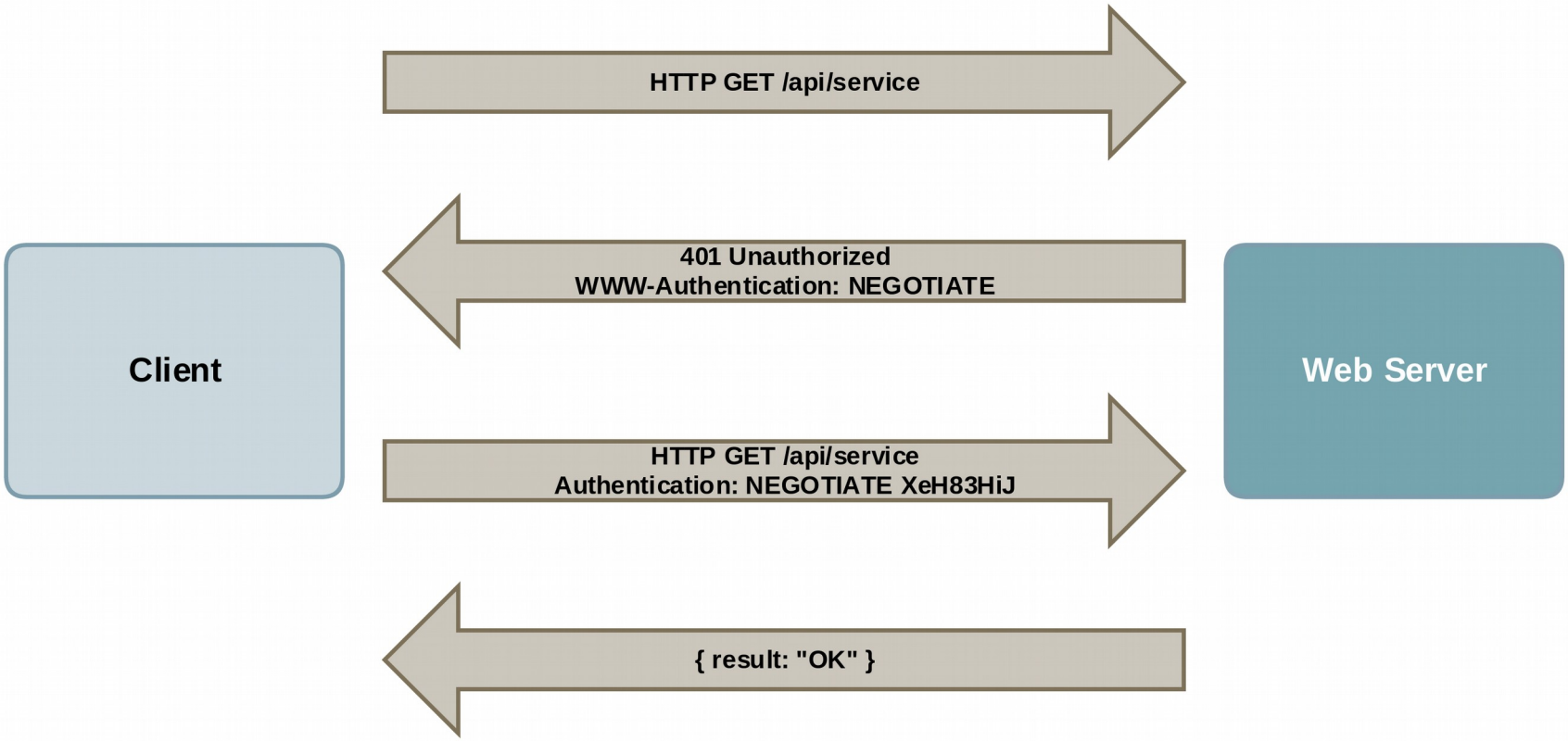
Kerberos negotiation process – important notes

- Target service does not have to talk to KDC during authorization process
- Authentication in Kerberos is ensured implicitly – by decrypting messages
- Passwords are never stored in clear text and are not sent over the network

Kerberos in HTTP - SPNEGO

- Kerberos is designed for client/server environments
- It's difficult to use pure Kerberos in web and thin client environments
- SPNEGO was created to overcome this issue – a wrapper protocol over Kerberos

SPNEGO


HTTP GET /api/service
401 Unauthorized
WWW-Authentication: NEGOTIATE
Client
Web Server
HTTP GET /api/service
Authentication: NEGOTIATE XeH83HiJ
{ result: "OK" }

SPNEGO

- When a Client wants to access protected Service without proper authentication data, it gets rejected with 401 Unauthorized status code
- Client then has to request a ticket from KDC
- Next, Client has to wrap the ticket in a SPNEGO envelope and send it to a web server requesting the same Service in Authorization header
- Web server can unwrap SPNEGO envelope and use the ticket inside as user credentials

Kerberos limitations

- KDC is a single point of failure, though it can be mitigated when using multiple servers
- Kerberos has strict time limitations on difference between clocks on hosts taking part in the process
- Kerberos cannot be used in scenarios where clients come from outside of the domain (typical Internet use case)
- Problematic creating staged environment: single-domain vs separate domain for each environment

Summary

- Authentication protocol well suited for intranets in companies
- Integrated in Active Directory on Windows servers
- Complicated usage in HTTP world, need to use wrapper protocol like SPNEGO

Kerberos introduction

Kerberos

What is Kerberos?

Kerberos is a computer network authentication protocol which 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.
[Wikipedia]

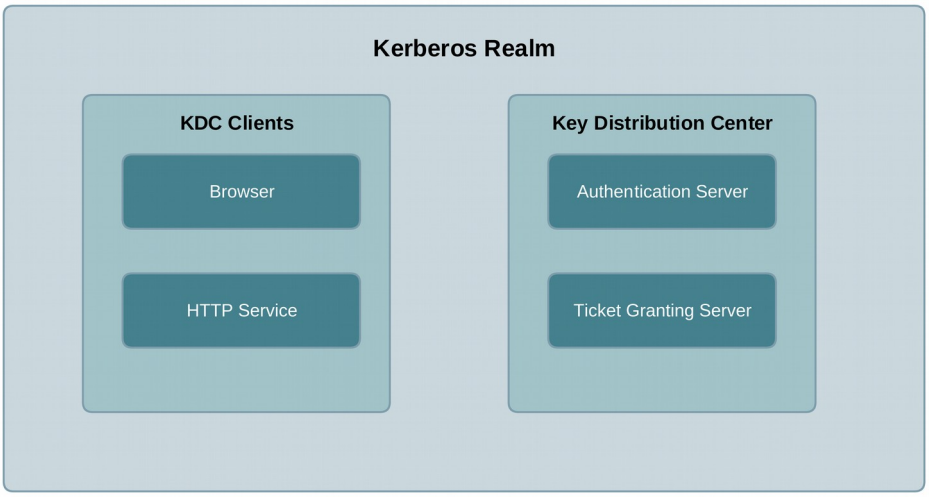
What is Kerberos?

- Developed on Massachusetts Institute of Technology (MIT)
- Windows 2000 and later use Kerberos as a default authentication mechanism (although uses its own implementation)
- Kerberos is also built into Active Directory solution
- UNIX-based operating systems have tools for authenticating over Kerberos

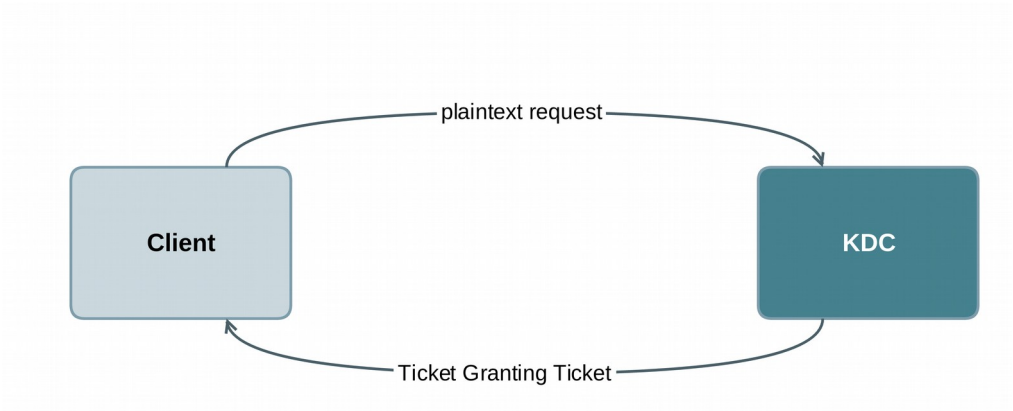
Kerberos use cases

- Authentication in Windows systems
- Providing SSO service in untrusted networks

Kerberos architecture



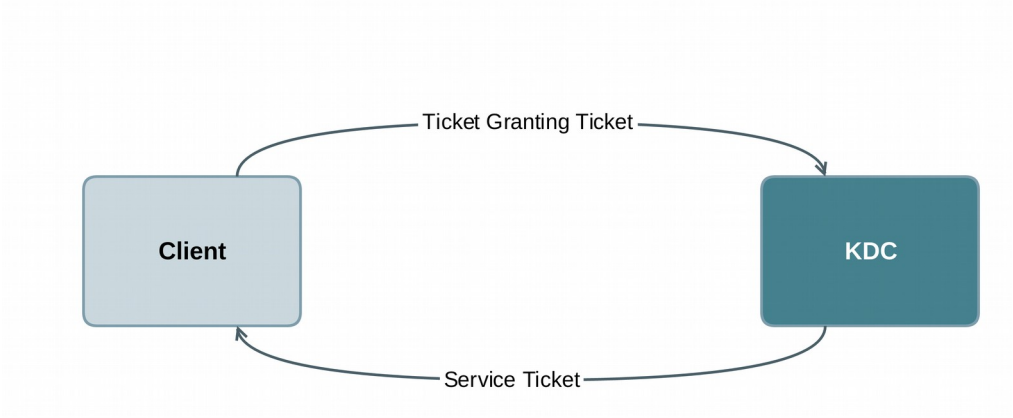
Kerberos negotiation process – part I



Kerberos negotiation process – part I

- Client sends a message with both encrypted and unencrypted parts
- Interestingly, unencrypted part caries all information about user
- Encrypted part is part of the protocol itself
- KDC lookups the user in its database
- If lookup is successful, KDC issues Ticket Granting Ticket signed by KDC's private key

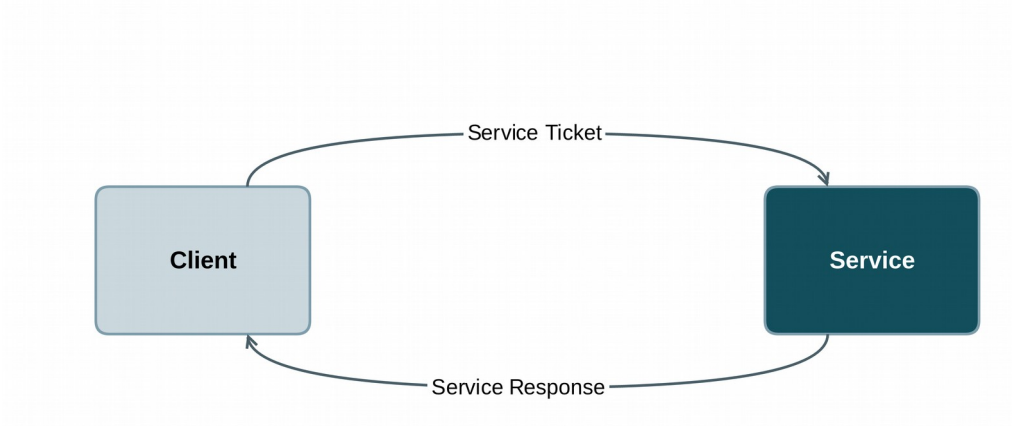
Kerberos negotiation process – part II



Kerberos negotiation process – part II

- To be able to authenticate with a service, Client sends the TGT with Service ID to KDC
- KDC checks Service ID in its database
- KDC decrypts TGT and if it's successful, Client is authorized to by granted access to Service
- KDC issues a Service Ticket signed by service's own public key
- This is the moment when a trust between Client and Service is established

Kerberos negotiation process – part III



Kerberos negotiation process – part III

- Client sends the encrypted Service Ticket to Service
- Service decrypts Service Ticket, if successful then Client is authorized to make a request to Service

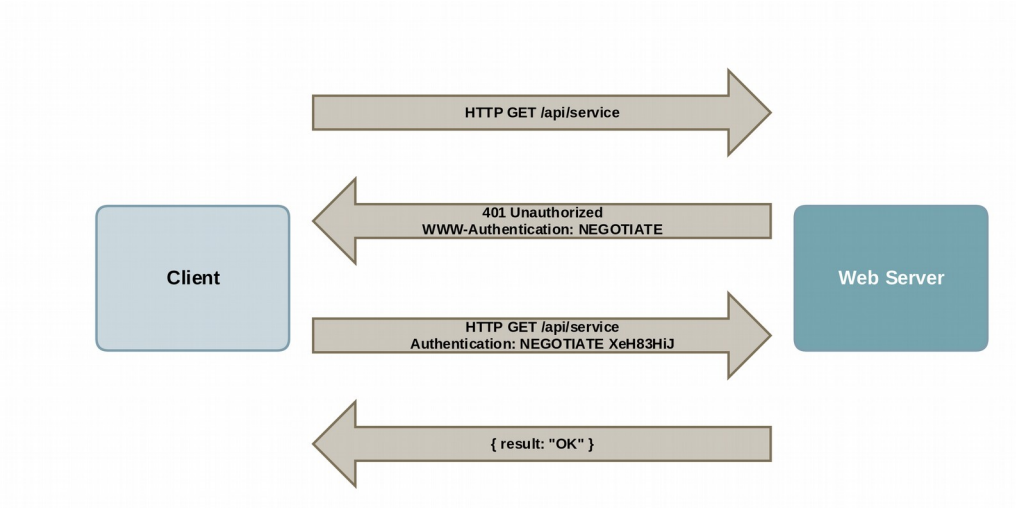
Kerberos negotiation process – important notes

- Target service does not have to talk to KDC during authorization process
- Authentication in Kerberos is ensured implicitly – by decrypting messages
- Passwords are never stored in clear text and are not sent over the network

Kerberos in HTTP - SPNEGO

- Kerberos is designed for client/server environments
- It's difficult to use pure Kerberos in web and thin client environments
- SPNEGO was created to overcome this issue – a wrapper protocol over Kerberos

SPNEGO


HTTP GET /api/service
401 Unauthorized
WWW-Authentication: NEGOTIATE
Client
Web Server
HTTP GET /api/service
Authentication: NEGOTIATE XeH83HiJ
{ result: "OK" }

SPNEGO

- When a Client wants to access protected Service without proper authentication data, it gets rejected with 401 Unauthorized status code
- Client then has to request a ticket from KDC
- Next, Client has to wrap the ticket in a SPNEGO envelope and send it to a web server requesting the same Service in Authorization header
- Web server can unwrap SPNEGO envelope and use the ticket inside as user credentials

Kerberos limitations

- KDC is a single point of failure, though it can be mitigated when using multiple servers
- Kerberos has strict time limitations on difference between clocks on hosts taking part in the process
- Kerberos cannot be used in scenarios where clients come from outside of the domain (typical Internet use case)
- Problematic creating staged environment: single-domain vs separate domain for each environment

Summary

- Authentication protocol well suited for intranets in companies
- Integrated in Active Directory on Windows servers
- Complicated usage in HTTP world, need to use wrapper protocol like SPNEGO