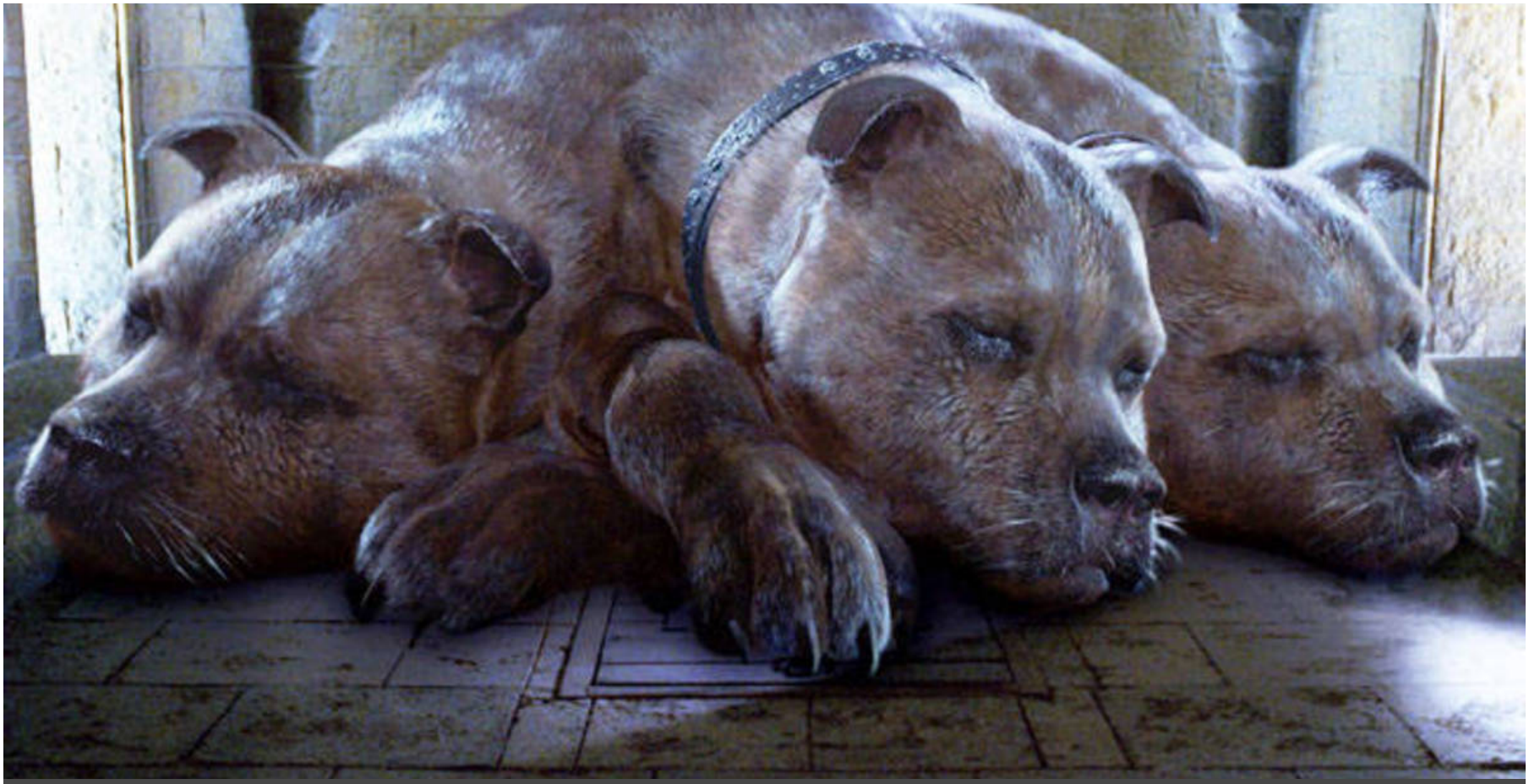


Kerberos

Cerberus

- Cerberus (Greek: Κέρβερος *Kerberos*) is a multi-headed dog that guards the gates of the Underworld to prevent the dead from leaving
- Kerberos is named after a three-headed dog because authentication is based on interaction between three systems
  - Requesting system (Principal)
  - Endpoint destination system
  - Kerberos server

Kerberos

- Network authentication protocol for client/server applications using symmetric (or public/private key) cryptography
  - Authentication
  - Access control
- **Single Sign-On (SSO)**
- Assumption: Network is insecure – Eve is watching!
- Developed in late 1980's at MIT as part of *Project Athena*
  - MIT / DEC / IBM project for distributed campus-wide computing environment
- Last updated in 2005 by IETR – Added AES support in v5

Kerberos

- Cross platform
  - Windows, Linux, *BSD, OS X
- Widespread application support*
  - Windows domains
  - SSH (OpenSSH)
  - IMAP, SMTP (Cyrus, sendmail, postfix)
  - CIFS/SMB (Samba, Windows, Netapp)
  - NFS
  - Database (SQL Server, Postgres)
  - HTTP (Apache, nginx, …)
  - DNS (Windows, bind)
  - ** support may be through GSSAPI or SASL layers*

Kerberos Terminology

- *All good systems have completely unique terminology!*
- **Principal** = Identity
  - **User Principal Names (UPN)** = Users
  - **Service Principal Names (SPN)** = Systems
- **Realm** = Authentication / Administration domain
  - All principals are assigned to realms
- **Key Distribution Center (KDC)**
  - Kerberos Database – What principals exist in this realm?
  - Authentication Service (AS)
  - Ticket-Granting Service (TGS)

Kerberos Terminology

- Examples of principals – these are NOT emails!
  - `alice@EXAMPLE.COM`
    - UPN for user `alice` in realm `EXAMPLE.COM`
  - `login/node1.example.com@EXAMPLE.COM`
    - SPN for service `login` on host `node1.example.com` in realm `EXAMPLE.COM`
- Case sensitive!
  - Convention is lowercase principals, uppercase realms

Alice (`alice@EXAMPLE.COM`) wants to use myservice.

- TGT: Ticket Granting Ticket
- TGS: Ticket Granting Service

Kerberos Workflow

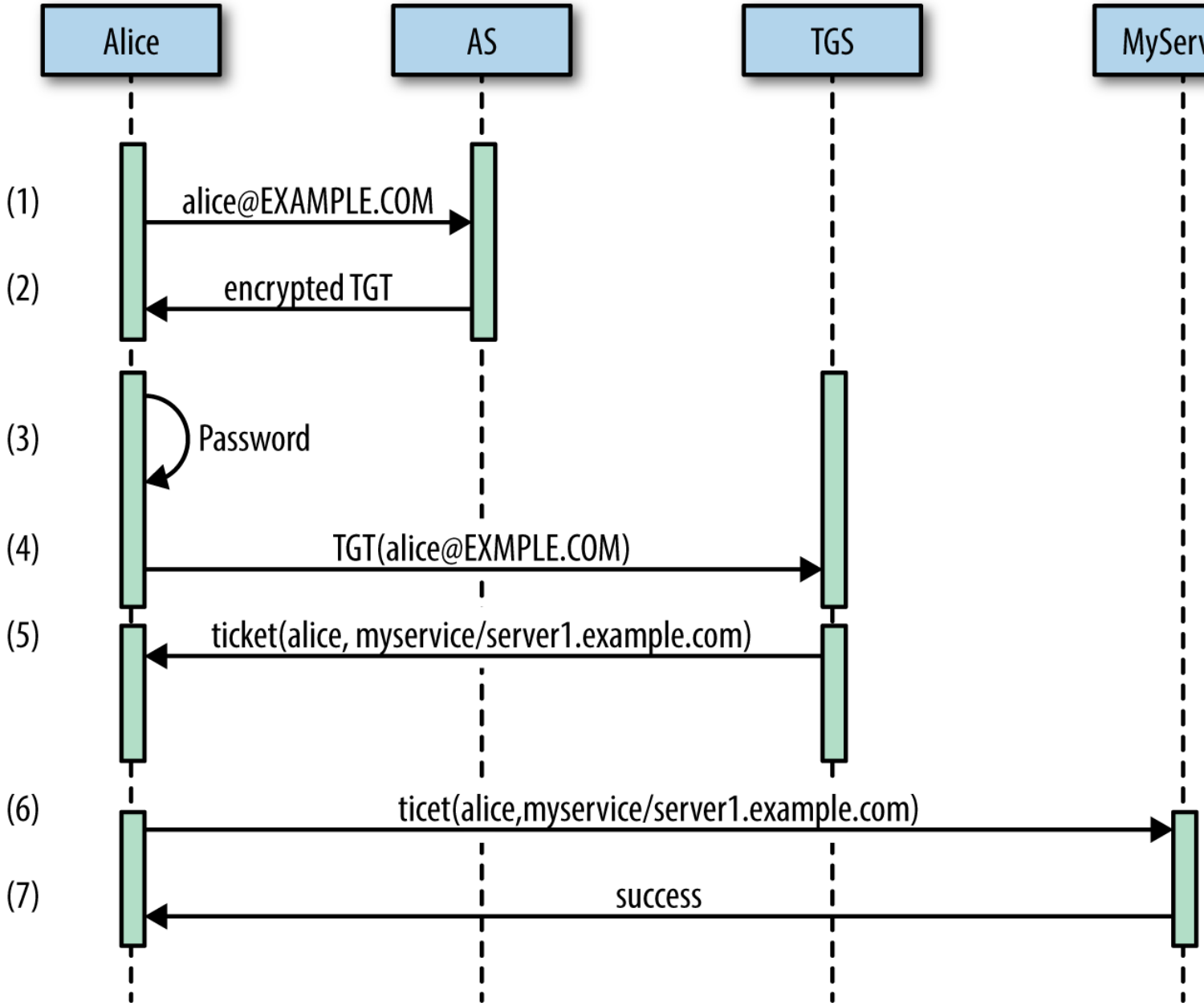
1. Alice needs to obtain a TGT
    1. Send request to AS identifying herself as the principal `alice@EXAMPLE.COM`.
2. AS responds by providing a TGT that is encrypted using Alice's key (password)
3. Alice enters her password to decrypt the TGT message
    1. This is *authentication….*
4. Alice requests a service ticket for `myservice` from the TGS, including the TGT with the request

Alice (`alice@EXAMPLE.COM`) wants to use myservice.

- TGT: Ticket Granting Ticket
- TGS: Ticket Granting Service

Kerberos Workflow

5. The TGS validates the TGT and provides Alice a service ticket (ST), encrypted with `myservice` principal's key
    5. This is *authorization…*
6. Alice presents the service ticket to `myservice`, which can then decrypt it using its key and validate
7. The service `myservice` permits Alice to use the service



https://www.safaribooksonline.com/library/view/hadoop-security/9781491900970/ch04.html

Kerberos Limitations

- Single point of failure (KDC server)
- Time synchronization required – tickets valid for only 5 minutes
- Compromise of authentication infrastructure allows attacker to impersonate any user (for symmetric cryptography implementation)
- All principals (users, systems) must have a trust relationship with KDC (same realm or trusted realm)
  - Does not work with unknown/untrusted clients