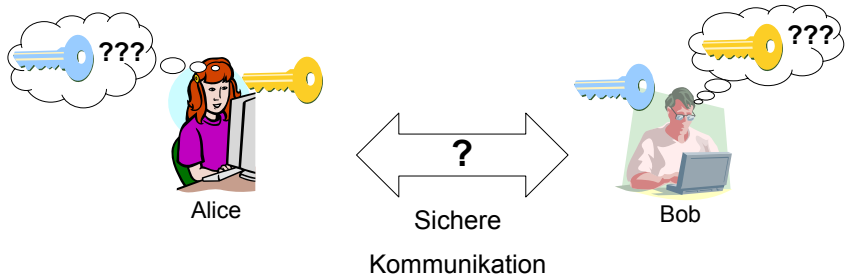
Netzsicherheit: Architekturen und Protokolle
Kerberos

TeleMatics

1. Einführung
2. Kerberos Version 4
3. Kerberos Version 5

Schlüsselspiel

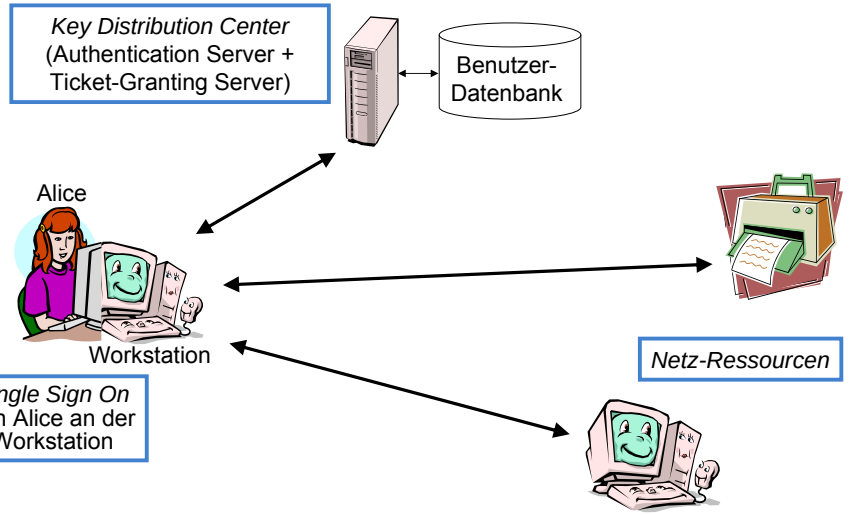


zentrale Frage: Woher kommt das Schlüsselmaterial?

Einführung in Kerberos

- Kerberos: Protokoll, das den Zugriff auf Ressourcen schützt
- Ziele von Kerberos
  - **Authentifizierung**
    - Anmeldung mittels Benutzernamen und Passwort
  - **Autorisierung**
    - Zugriff auf Ressourcen durch Rechtesystem beschränkt
    - Rechte werden von einem Administrator vergeben
  - **Accounting**
    - Protokollierung von Ressourcen-Nutzung
- Single-Sign-On für Netzwerke

Kerberos – Überblick



Kerberos Komponenten

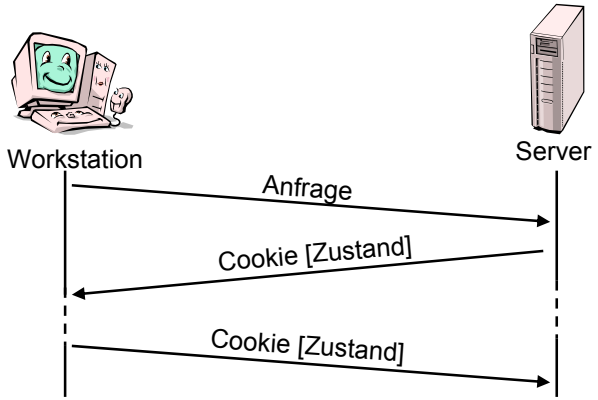
- *Authentication Server* (AS)
  - Authentifizierung der Benutzer
  - Ausstellen eines Authentifizierungs-Tokens
    - *Ticket-Granting-Ticket* (TGT)
- *Ticket-Granting Server* (TGS)
  - Ressourcen-Zugangs-Server
  - Autorisierung des Ressourcen-Zugriffs bei Vorlage eines gültigen TGTs
  - Ausstellen von Zugangsberechtigungen (Tickets)
- Benutzer-Datenbank
  - Speichert Client Master Secrets aller Benutzer und Ressourcen

Kerberos allgemein

- Kerberos Versionen
  - Version 1 bis 3 heute nicht mehr im Einsatz
  - Version 4 und Version 5 konzeptionell ähnlich
  - dennoch erhebliche Unterschiede: Version 4 ist
    - einfacher
    - leistungsfähiger
    - arbeitet allerdings nur in IPv4 Netzen
- Anwendungen, die Kerberos unterstützen
  - Telnet
  - BSD rtools
  - NFS
  - SSL, SSH
  - OSF/DCE
  - Windows 2000, 2003, XP, Vista

Ausflug: Tokens/Cookies

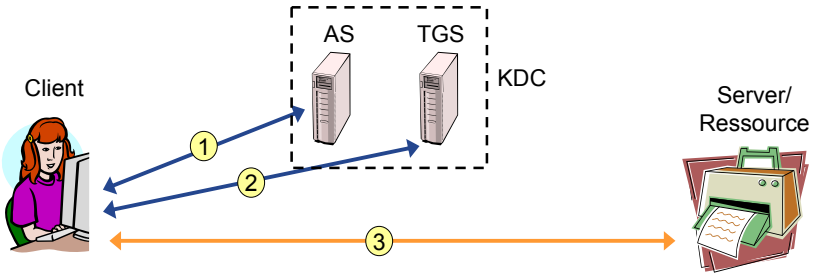
- Server speichert Zustand pro Anfrage
  - Problem: Überlastung des Servers
  - Lösung: Token/Cookies



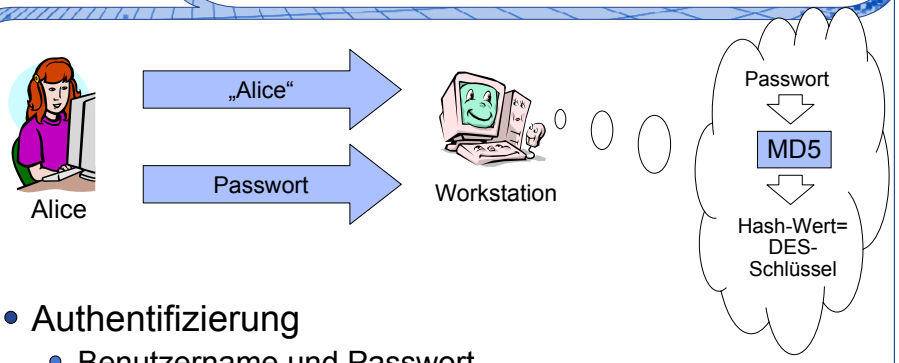
Netzsicherheit
Architekturen und Protokolle
Kerberos

Kerberos – The Big Picture

1. Anmeldung
   - Client erhält ***Ticket-Granting-Ticket*** vom ***Authentication-Server***
2. Ressourcenanforderung
   - Vorlage des Ticket-Granting-Tickets beim ***Ticket-Granting-Server***
   - Client erhält ***Ticket*** für die Ressource
3. Kommunikation mit der Ressource



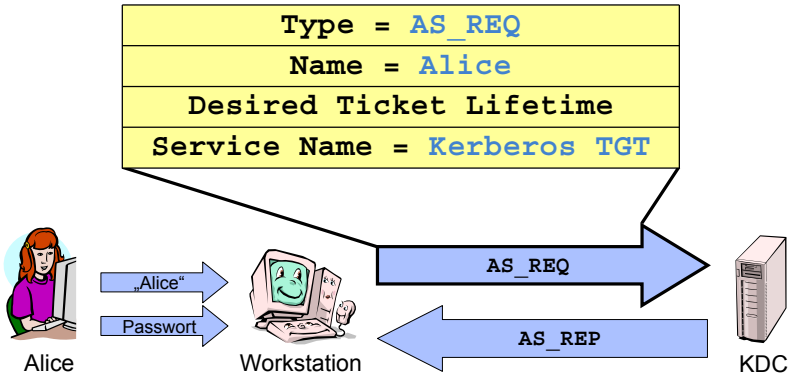
Anmeldung an der Workstation



- Authentifizierung
  - Benutzername und Passwort
  - Umwandeln des Passworts in einen *DES-Schlüssel*
    - *Client-Master-Secret*
    - Umwandlung durch Hash-Funktion (MD5), siehe RFC4120

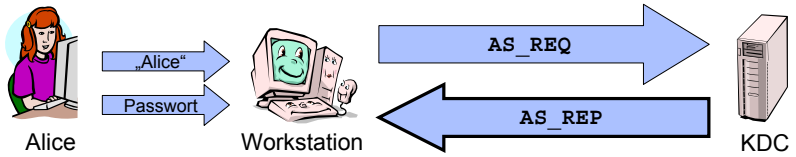
Anmelden am Netz

- *Authentication Server Request* (`AS_REQ`)
  - Übertragen des Benutzernamens im Klartext zum KDC



Anmelden am Netz

- Authentication Server Reply (`AS_REP`)
  - verschlüsselt mit dem Master Secret des Clients
  - *Session Key*: geheimer Sitzungsschlüssel
  - *Ticket-Granting Ticket* (TGT) enthält
    - Benutzernamen, Adresse
    - Sitzungsschlüssel, Gültigkeitsdauer
    - …



Vereinfachtes Paketformat *AS_REP*

Type = *AS_REP*	Session Key (Alice↔KDC)	Ticket Granting Ticket (verschlüsselt mit Master Secret des KDC)	KDC`s Timestamp
Name = *Alice*	Name = *KDC*		
Alice`s Timestamp	Ticket Lifetime		
Ticket Expiration Time	KDC`s Key Version Number		
Credentials Length	TGT Length		

Verschlüsselt mit Master Secret des Clients

Ticket Granting Ticket (vereinfacht)

Client Name	`Alice`
Network Layer Address	`192.168.178.55`
Session Key	`(Alice ↔ KDC)`
Ticket Lifetime	`60`
KDC`s Timestamp	`9:20am`
Server Name	`KDC`

Verschlüsselt mit dem Master Secret des KDC

Diskussion

- Warum tauscht Kerberos einen *Sitzungsschlüssel* aus und verwendet nicht das *Master Secret* des Client für die Kommunikation?
- Warum erfolgt der *Zugriff auf Ressourcen* über den Umweg über das TGT?

Offline-Password-Guessing Angriff

Ziel: *Erlangen des Benutzer-Passworts*

- **`AS_REQ`** und **`AS_REP`** abhören und speichern
  - eindeutig einem Benutzer zuzuordnen
  - Client Name im Klartext enthalten
- Wörterbuch-Angriff
  - pro Wort aus dem Wörterbuch
    - Wort mittels MD5 in DES-Schlüssel umwandeln
    - **`AS_REP`** entschlüsseln
    - Testen der entschlüsselten Nachricht auf Plausibilität
      - z.B. über Zeitstempel
    - Schlüsselkandidaten an weiteren Nachrichten testen
  - ist Sitzungsschlüssel bekannt, können alle weiteren Nachrichten entschlüsselt werden

Diskussion: Offline-Password-Guessing-Angriff

Ist dieser Angriff auch als *aktiver Angriff* möglich? Wenn ja, warum, wenn nein, warum nicht?

Kerberos Credentials

- *Ticket-Granting-Ticket*
  - ausgestellt vom Authentication-Server
  - ermöglicht Nutzung des Ticket-Granting-Servers
- *Tickets*
  - ausgestellt von Ticket-Granting-Server
  - ermöglichen Nutzung von Ressourcen
- *Authenticator*
  - Einschränkung von Replay Attacken (Angriff durch Wiedereinspielen)

Aufbau eines Tickets

- Ticket für
  - Kommunikation von Alice mit Bob
  - angefordert von Alice

Client Name	`Alice`
Network Layer Address	`192.168.178.55`
Session Key	`(Alice ↔ Bob)`
Ticket Lifetime	`60`
KDC's Timestamp	`9:20am`
Server Name	`Bob`

Verschlüsselt mit Client Master Secret von Bob

Kerberos Credentials (2)

- Authenticators
  - erzeugt von Alice (=Client)
  - nur *einmal* einsetzbar
  - Verhinderung von Replay-Angriffen
  - Bedingung: Synchronisation der Systemuhren
    - nur in Zeitfenster gültig

Name = `Alice`
Zeitstempel

verschlüsselt mit dem jeweils verwendeten Schlüssel
(z.B. Session Key Alice↔KDC oder Session Key Alice↔Bob)

Adressen in Tickets

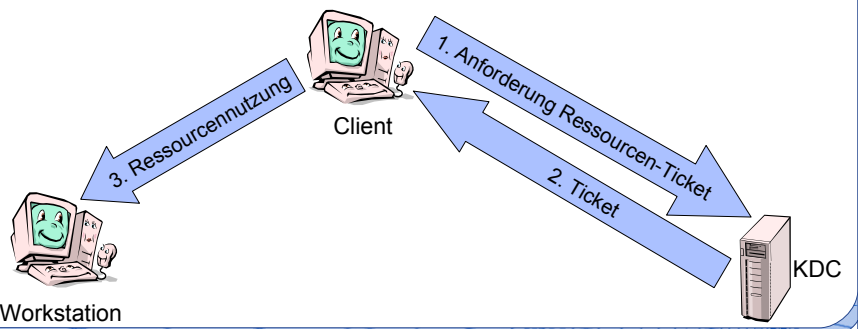
- Netzwerk-Adresse des Clients in jedem Ticket
  - Vergleich der Absender-Adresse mit der enthaltenen Adresse bei Empfang eines Tickets
    - keine Weitergabe von Tickets möglich
    - Schutz vor *Ticket-Diebstahl*
    - Verhinderung der Nutzung eines abgefangenen Tickets
- Problem
  - Fälschung der Absender-Adresse einfach
    - kein wirksamer Sicherheitsmechanismus
  - Rechteübertragung in Kerberos v4 nicht möglich
    - gewünscht, z.B. Batch-Prozess, der auf eigene Daten zugreift

Zusammenfassung Anmeldevorgang

- KDC authentifiziert Alice anhand
  - Kenntnis des Client Master Secrets von Alice
    - aus Passwort abgeleitet
    - in Benutzer-Datenbank des KDC
- Ticket-Granting-Ticket
  - KDC kann damit vorherige Authentifizierung überprüfen
  - Auslagerung des Server-Zustands
- Alice und KDC verfügen nach Anmeldevorgang über einen Sitzungsschlüssel
  - Client Master Secret muss nicht mehr verwendet werden
  - langlebiges Geheimnis geschützt
  - Sitzungsschlüssel in TGT

Ressourcenanforderung

- Ressourcen-Nutzung nach Wiedervorlage d. TGT beim KDC
  - Ticket-Granting-Server (TGS) gibt Tickets aus
  - Zugangskontrolle durch jede Ressource
  - Erweiterung: Zugriffsbeschränkung durch KDC
    - Ausstellung eines Tickets anhand zusätzlicher Informationen



Anfordern eines Ressourcen-Tickets

→ Alice möchte ein Ticket für Bob

- *Ticket-Granting Server Request* (**`TGS_REQ`**)
  - enthält TGT
  - Ressourcen-Name
  - Authenticator verschlüsselt mit Sitzungsschlüssel
- Überprüfung durch Ticket-Granting-Server
  - Absenderadresse
  - Name
  - Zeitstempel

Type = **`TGS_REQ`**
KDC`s Key Version Number
TGT
Authenticator
Alice`s Timestamp
Desired Ticket
Server Name = Bob

Anfordern eines Ressouren-Tickets (2)

Ticket-Granting Server Reply (`TGS_REP`)

- Session Key Alice ↔ Bob
- Ticket verschlüsselt mit Master Secret der Ressource

Type = TGS_REP	Session Key Alice↔Bob	Ticket für den Ressourcen-Zugriff auf Bob (verschlüsselt mit Bobs Master Secret)	KDC`s Timestamp
Name = Alice	Name = Bob		
Alices Timestamp	Ticket Lifetime		
Ticket Expiration Time	Bob`s Key Version Number		
Credentials Length	Ticket Length		

Verschlüsselt mit dem Session Key Alice ↔ KDC

Zusammenfassung Ressourcen-Anforderung

- Wiedervorlage
  - Ticket Granting Ticket beim Ticket Granting Server
  - Ticket Granting Server muss *keinen Zustand halten*
  - Wiedergewinnung des Sitzungsschlüssels Alice ↔ KDC
- Tickets zum Zugriff auf Ressource
  - Ausgestellt durch Ticket Granting Server
  - Enthält vom Ticket Granting Server erzeugten Sitzungsschlüssel Alice↔Bob

Kommunikation mit der Ressource

- *Application Request* (`AP_REQ`) enthält
  - Ticket und Authenticator
  - Überprüfung durch Ressource analog zu Überprüfung eines TGT durch TGS

Type = AP_REQ	Ticket (verschlüsselt mit Master Secret von Bob)	Authenticator (verschlüsselt mit Session Key Alice↔Bob)
Bob`s key version number		

Kommunikation mit der Ressource

- *Application Reply* (`AP_REP`)
  - enthält Authenticator
  - danach Austausch der Anwendungsdaten
    - Ungeschützt, Integritätsschutz oder Verschlüsselung mit Integritätsschutz, … → Aufgabe der Anwendungsprotokolle
- `AP_REP` eigentlich in Dokumentation nicht erwähnt, aber von vielen Anwendungen so verwendet

Diskussion: Kerberos in großen Netze

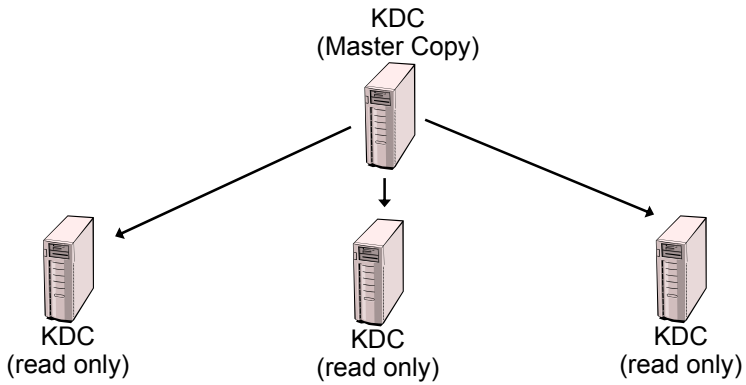
Welche Probleme können Sie sich vorstellen, wenn Kerberos in einem großen Netz eingesetzt wird?

Schlüssel-Server für große Netze

- Einzelner KDC ist *Single-Point-of-Failure*
  - Replizierung des Schlüssel-Servers
- Zentraler Punkt: Wissen alles Master-Secrets
  - Gliederung des Netzes in Domänen
    → so genannte *Realms*

Replizierte Schlüssel-Server

- Alle KDCs besitzen gleiches KDC Master-Secret
  - ein Master Copy der Benutzerdatenbank
  - ein oder mehrere *read-only*-Slaves

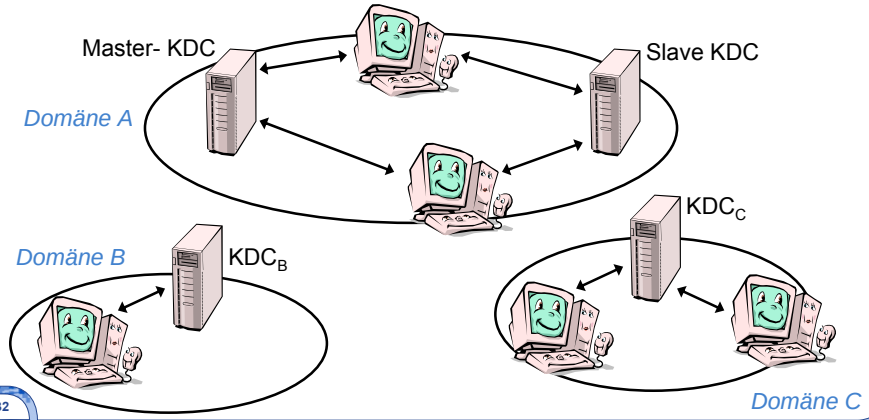


Replizierte Schlüssel-Server

- Master-Copy der Benutzerdatenbank
  - alle Änderungen auf der Master-Copy
  - Authentifizierung über Master-Copy KDC und read-only KDC
  - Ausfall des Masters
    - keine Update-Operationen möglich
    - Netz bei Ausfall des Masters aber weiter nutzbar
- Synchronisierung der read-only Slaves
  - periodisch oder per Administrations-Kommando
  - „Klartext-Übertragung“ mit anschließendem kryptographischen Hash
    - Client Master Secrets verschlüsselt mit KDC Master-Secret
    - Hash zum Schutz vor Manipulation, Vertauschen, Anfügen von Daten

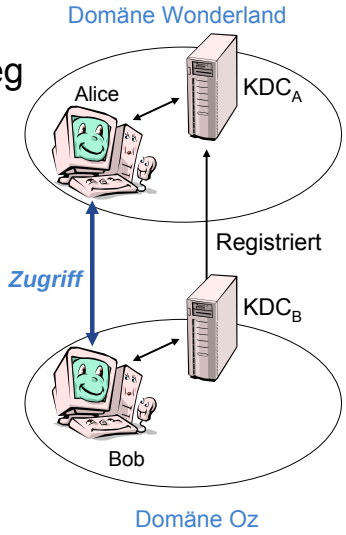
Domänen

- Lösung für viele administrative Bereiche: *Domänen*
  - eigene Benutzer-Datenbank für jede Domäne (Realm)
  - innerhalb der Domäne Replizierung möglich
  - KDCs einer Domäne besitzen gleiches KDC Master-Secret



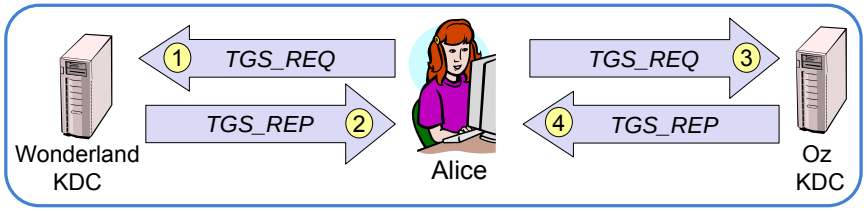
Inter-Domänen-Authentifizierung

- **Problem:** Authentifizierung über Domänengrenzen hinweg
  - Nutzung von Ressourcen in anderer Domäne
  - Autorisierung durch KDC der anderen Domäne
- **Lösung:** KDC kann als Client eines anderen KDC registriert sein



Inter-Domänen-Authentifizierung (2)

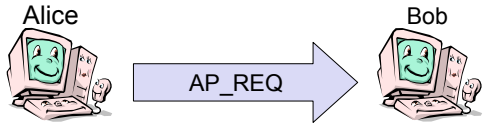
- Alice@Wonderland möchte mit Bob@Oz kommunizieren
  1. Alice fordert Ticket für KDC der Domäne Oz an
  2. Wonderland-KDC erstellt Ticket für Oz-KDC
  3. Alice fordert Ticket für Bob von Oz-KDC an
     - Ticket von Wonderland-KDC als TGT
  4. Oz-KDC erstellt Ticket, mit dem Alice auf Bob zugreift



- Verkettung von Inter-Domänen Tickets in Kerberos v4 nicht möglich
  - Domänen-Feld des Ticket und Absender-Domäne müssen übereinstimmen

Passwort-Änderung

- Passwort-Änderung
  - Benutzer eines OS kann jederzeit Passwort ändern
    → Änderung beeinflusst nur ihn
  - Änderung des Client Master-Secrets genauso einfach?
  - **Problem**: ausgestellte Tickets sind mit Master-Secret verschlüsselt, das aus dem alten Passwort generiert wurde!



Ticket für Bob ist mit dessen Master-Secret verschlüsselt

Änderung des Passworts
→ neues Master-Secret

- **Frage**: werden alle bereits ausgestellten Tickets ungültig?

Passwort-Änderung (2)

- Lösung: *Versionsnummer der Schlüssel*
  - Speicherung mehrerer Schlüsselversionen
    - Gültigkeitsdauer eines Tickets auf 21,25 Stunden beschränkt
  - Jedes Ticket, jede Nachricht enthält Versionsnummer
    - ID des verwendeten Schlüssels
- Problem: *Replizierung des KDCs*
  - Verteilung des neuen Master-Secrets auf Slave-KDCs
  - einloggen mit neuem Passwort unter Umständen nicht sofort möglich
  - altes Passwort weiterhin gültig

  → Verwirrung des Benutzers

Zusammenfassung Kerberos v4

- Single-Sign-On-Network
  - Authentifizierung mit Benutzernamen und Passwort
  - Anmeldung beim Authentication-Server
  - Anforderung der Ressourcenutzung beim Ticket-Granting-Server
  - Autorisierung durch den Ticket-Granting-Server
  - Schutzmechanismen der Anwendung nicht festgelegt

Zusammenfassung Kerberos v4 (2)

- KDC als Single-Point-of-Failure
  - Replizierung des KDC
    - 1x Master-Copy User-Datenbank, beliebige Read-Only-Slaves
    - Problem: Update des Passworts
  - Domänen
    - Interdomänen-Authentifizierung
    - keine Verkettung
- Kerberos Netzwerktrace
  - siehe Vorlesungsmaterialien *ns-xx-KerberosTrace*

Verwenden Sie zur Prüfungsvorbereitung bitte unbedingt die Folien mit Anmerkungen!

Literatur

Sichere Netzwerkkommunikation,
Bless et al., Springer, 2005.

Network Security – Private Communication in a Public World,
Kaufmann, Perlman, Speciner, Prentice Hall, 2002.

Telnet Authentication: Kerberos Version 4,
D. Borman, RFC 1411, 1993