GATE Manual

Thimo Langbehn

4. April 2009

Inhaltsverzeichnis

1 Kerberos

Um in den vollen Genuss der Möglichkeiten eines Kerberos-Systems (Single Sign On) zu kommen muss man eine Kerberos-Implementierung installieren. Die Implementierung besteht im wesentlichen aus einer Kerberos Bibliothek und einigen Programmen zur Ticket(cache) Verwaltung.

1.1 Voraussetzungen

- Die Namensauflösung für die Domäne **„g4t3.de“** muss funktionieren.
- Der Host **„auth.g4t3.de“** muss erreichbar sein (**UDP Port 88**).

1.2 Programme

Linux Je nach Linux-Distribution haben die MIT Kerberospakete verschiedene Namen:

Debian krb5-user

Gentoo mit-krb5

1.3 Konfiguration

Die Bibliothek muss über eine Konfigurationsdatei eine Liste von sogenannten Realm Spezifikationen erhalten, um einer DNS Adresse einen Realm, und diesem einen KDC-Server zuzuordnen. Sie muss wie folgt abgeändert werden:
`http://www.g4t3.de/download/krb5.ini`

```
[libdefaults]
  default_realm = G4T3.DE
  dns_lookup_kdc = true

[realms]
  G4T3.DE = {
    admin_server = auth.g4t3.de
```

```
    kdc = auth.g4t3.de
  }

[domain_realm]
  .g4t3.de = G4T3.DE
  g4t3.de = G4T3.DE

[logging]
```

Die restlichen Realm-Beschreibungen (in der Sektion **[Realms]**) können bestehen bleiben, werden aber für uns nicht benötigt.

Diese Datei ist insofern nicht sicherheitskritisch, als dass sie problemlos gelesen werden darf. Schreibzugriff sollte allerdings nur einem Administrator gestattet sein. Ansonsten könnte durch Manipulation der **kdc** und **admin_server** Adresse eine Spoofing-Attake eingeleitet werden.

1.3.1 Linux

Unter Linux liegt die **krb5.conf** gnaz normal unter /**etc**/.

1.4 Verwendung

Voraussetzungen für die Verwendung von Kerberos sind die oben beschriebenen Einstellungen, eine bestehende Netzwerkverbindung (Internet) und Verfübarkeit des korrekten DNS-Servers. Ein **ping auth.g4t3.de** sollte möglich sein. Weiterhin muss die Uhrzeit des Rechners korrekt eingestellt sein.

1.4.1 Linux

Zeitsynchronisation Unter Linux wird die Zeitsynchronisation am besten per ntp durchgeführt (**ntpdate pool.ntp.org; hwclock –systohc**).

Einloggen Ein Ticket kann mittels **kinit USER** ein Ticket gezogen werden. Sollte der lokale Login-Name mit dem gewünschten G4T3-Usernamen übereinstimmen, so kann der Name auch weggelassen werden. Nützliche Optionen für **kinit** sind **-f** (forwardable) und -l 1d (ein Tag Gültigkeit). Wenn die Anforderung erfolgreich war, kann das nun vorhandene Ticket mit **klist** angezeigt werden.

Passwort ändern Mit dem Programm **$> kpasswd** kan das Kerberos Passwort geändert werden. Es wird genauso benutzt wie das passwd Programm.

1.5 Fehlerbehebung

1. Synchronisiere die Zeit mit einem Internet-Zeitserver. Sollte dies mit einer Fehlermeldung abbrechen, oder die Zeit nicht korrekt synchronisiert werden, so kann eine Ursache dafür in einer dazwischenliegenden Firewall liegen. In diesem Fall kann die Zeit auch manuell gestellt werden. Sie muss nur innerhalb einer Toleranz von 5 Minuten korrekt sein.

2. führe ein **$> ping 193.34.68.209** aus. Wenn dieser Ping nicht möglich ist, so erlaubt eine zwischen dem Client (dir) und dem Server liegende Firewall diese Operation nicht. Hier kann entweder ein weniger restriktives Netz um testen aufgesucht werden, oder die Einstellungen der Firewall angepasst werden.

3. führe ein **$> ping auth.g4t3.de** aus, dabei sollte der Name in die korrekte IP (**193.34.68.209**) aufgelößt werden.

   Funktioniert dieser Test nicht, der obrige aber schon, dann arbeitet die DNS-Auflöung nicht richtig. Dass kann an einem fehlerhaften Nameserver liegen, oder auch an überchreibenden Einstellungen des Clients bzw. alten Werten in dessen Puffer.

   Eine andere IP (nicht 193.34.68.209) wird angezeigt Irgendein Nameserver oder Cache liefert die falschen Werte zurück. Überprüfe die hosts Datei. Diese findet sich unter Windows in **„C:\Windows\System32\drivers\etc\hosts“**, und unter Linux in **/etc/hosts**. Suche nach Einträgen der Domain **„g4t3.de“**. Diese sollten entfernt werden, sie sind nur nötig wenn der Nameserver keine korrekten Ergebnisse liefert.

   Es wird gar keine IP angezeigt Die Domäne wird von dem lokalen Nameserver nicht richtig aufgelößt. Trage die folgenden Einträge in die oben gennante hosts Datei ein:

   ```
   193.34.68.209 auth.g4t3.de
   193.34.68.210 info.g4t3.de
   193.34.68.211 mail.g4t3.de
   193.34.68.212 data.g4t3.de
   193.34.68.213 www.g4t3.de
   ```

4. Überprüfe deine Firewall, ob sie ausgehende Verbindungen zum **Port 88** von **193.34.68.209** zulässt. Ein Test in einer weniger restriktiven Umgebung kann Unklarheiten bezüglich anderer, dazwischen liegender Umgebungen beseitigen).

2 SSH Client

Als SSH Client wird OpenSSH verwendet (Kerberos-Support muss einkompiliert sein).

2.1 Voraussetzungen

- OpenSSH mit Kerberos Support
- Kerberos muss eingerichtet sein (Siehe section1).
- Die Namensauflösung für die Domäne **„g4t3.de“** muss funktionieren.
- Der Host **dev.g4t3.de** (oder ein anderer kerberisierter Server) muss erreichbar sein.

2.2 Konfiguration

In der ssh Konfigurationsdatei **/etc/ssh/ssh_config** (nicht /etc/ssh/sshd_config) muss der Wert von **GSSAPIAuthentication** auf **yes** gesetzt sein.

Der Wert von **GSSAPIDelegateCredentials yes** sollte bei bestimmten Anwendungen ebenfalls gesetzt werden. Mit dieser Option gibt man dem Zielrechner die Möglichkeit, die eigenen Berechtigungen einzufordern, also zum Beispiel auch entsprechende weiterführende ssh Verbindungen zu autorisieren. Generell heißt das, dass ein kompromittierter Zielhost nach einem erfolgreichen login für die Lebensdauer des TGT wie man selbst agieren könnte, insbesondere also auch Dateien löschen, Mails senden und löschen usw. Dieser Zugriff ist aber auf die Lebensdauer des TGTs beschränkt.

Auf der andern Seite ermöglicht dieser Mechanismus einigen Servern nahezu rechtelos zu laufen, da sie die für ihre Dienste erforderlichen Rechte direkt aus diesem übermittelten Ticket ziehen können. Zum einen ist es damit unmöglich, Rechte wahrzunehmen die man selbst nicht hat, und zum anderen kann ein kompromittierter Service so nur sehr begrenzten Schaden anrichten.

2.3 Test

Eine ssh Verbindung zu **dev.g4t3.de** aufbauen. Nach eingabe des Benutzernamens sollte der Loginvorgang automatisch abgeschlossen werden (keine weitere Eingabe eines Passwortes).

In der damit gestarteten Shell **$> ssh www** eingeben. Dies sollte mit einem **Permission denied** vom Server verweigert werden.

Ein TGT mit aktivierter forwarding-Option ziehen (**kinit -f**) und den Vorgang (Einwählen nach **dev.g4t3.de**) wiederholen.

In der damit gestarteten Shell erneut **$> ssh www** eingeben. Dies sollte jetzt erfolgreich sein.

2.4 Fehlerbeherbung

3 SCP

Das mit OpenSSH mitgelieferte scp kann problemlos verwendet werden.

3.1 Voraussetzungen

- OpenSSH muss eingerichtet sein (Siehe section2).
- Kerberos muss eingerichtet sein (Siehe section1).
- Die Hosts **auth.g4t3.de** und **data.g4t3.de** müssen erreichbar sein.

3.2 Konfiguration

Es ist keine weitere Konfiguration erforderlich

3.3 Test

Eine belibige Datei per scp auf dev.g4t3.de kopieren.

3.4 Fehlerbeherbung

3.4.1 Authentifizierung schlägt fehl

Die Authentifizierung kann nicht erfolgreich durchgeführt werden.

1. Lösche eventuell vorhandene Tickets (kdestroy) Und fordere neue an.
2. Log dich per ssh auf **dev.g4t3.de** ein. Wenn irgendwelche Fehlermeldungen auftreten (ein Skript konnte nicht ausgeführt werden, es konnte nicht ins Heimatverzeichniss gewechselt werden usw.), versuche den Grund dafür zu beseitigen. Sobald das erfolgreicht durchgeführt wurde, versuche erneut einen SCP transfer.

4 Maildienste

Unter g4t3.de wird ein kerberosgesicherter Mailserver betrieben. Dieser bietet den g4t3.de usern unter anderem folgende Features:

- Sehr großes Postfach (momentan im zweistelligen GB Bereich)
- (Nahezu) belibige Mail Aliasnamen einstellbar
- Automatische Viren- und Spamkennzeichnung (-bewertung)
- Automatischer Mail-Sammeldienst von anderen Konten.
- Standard- und eigene Mailverteiler.
- Shared-Mailboxes nach Bedarf

4.1 Mailclient Voraussetzungen

- Ein g4t3.de Mailaccount (im Full-Account enthalten)
- Eine Kerberos-Implementierung
- Die Namensauflösung für die Domäne **„g4t3.de“** muss funktionieren.

4.2 Mailclient Programme

Linux Für Linuxnutzer stehen folgende E-Mail Clients zur Verfügung:

- Mozilla Thunderbird
- Evolution

4.3 Mailclient Konfiguration

Der Mailserver läuft auf **mail.g4t3.de** und ist per **IMAP** und **SMTP** erreichbar. Um auf die jeweligen Dienste zuzugreifen, wird ein entsprechendes Service-Ticket benötigt. Das heißt in der Regel, dass die Clientanwendung kerberisiert sein muss.

Um auf den IMAP Server zuzugreifen kann normales IMAP oder IMAP über SSL oder TLS verwendet werden (TLS empfohlen)

Der SMTP Server kann über mehrere Ports angesprochen werden, und erfordert eine Authentifizierung um Mails mit einer Zieladresse außerhalb der g4t3.de Domäne zu versenden (Relay). Dies ist nicht nur sinvoll, sondern auch von der DENIC erwünscht.

4.3.1 Direktes SMTP

Hier gibt es derzeit drei verfügbare Ports.

Port 25 Der Standart SMTP Port, mit optionalem TLS

Port 465 Der IANA URD Port, verbreitet in der Verwendung als SMTPS Port. Hier ist TLS erforderlich.

Port 443 Der HTTPS Port, als einer der Ports die in der Regel nicht geblockt werden, stellt eine Möglichkeit dar aus restriktiven Umgebungen Mails zu versenden.

Sollte keiner der drei Ports aufgrund von Firewalleinstellungen der Umgebung erreichbar sein, kann noch ein SSH Tunnel verwendet werden.

4.3.2 Mozilla Thunderbird für Windows XP

in den Konfigurationswerten (Erreichbar über Einstellungen -> Erweitert -> Konfiguration bearbeiten) müssen einige kerberosrelevante Werte angepasst werden:

network.negotiate-auth.allow-proxies true

network.negotiate-auth.delegation-uris – Defaultwert (leer) –

network.negotiate-auth.gsslib – Defaultwert (leer) –

network.negotiate-auth.trusted-uris – Defaultwert (leer) –

network.negotiate-auth.using-native-gsslib true

Es sollte ein Konto mit folgenden Daten angelegt werden:

Name Frei wählbar

E-Mail Adresse user@g4t3.de (Hier kann auch ein beliebiger eingerichteter Alias eingetragen werden).

Typ imap

Imap-Server mail.g4t3.de

Port TLS auf Port 143, oder SSL auf Port 993

Authentifizierung Sichere Authentifizierung

Für den Benutzernamen wird der eigene Accountname angegeben.

Weiterhin wird noch der dazugehörige SMTP-Server angelegt. Die Konfiguration erfolgt im Thunderbird unter dem Knoten **Ausgehende Mailserver**. Dort wird eine neuer Eintrag hinzugefügt (der Name kann frei gewählt werden), mit den folgenden Einstellungen:

SMTP-Server mail.g4t3.de

Port siehe oben, 443 wird empfohlen

Authentifizierung Verwende Username und Passwort aktivieren.

Username dieses Feld ist leer zu lassen

Direkt in der Hauptkonfiguration z.B. des g4t3 Accounts sollte dann diese SMTP Verbindung für ausgehende Nachrichten gewählt werden. (Rechtsklick auf einen Konteneintrag -> Eigenschaften)

4.3.3 Mozilla Thunderbird für Windows 7 64Bit oder Vista 64Bit

Es sollte ein Konto mit folgenden Daten angelegt werden:

Name Frei wählbar

E-Mail Adresse user@g4t3.de (Hier kann auch ein beliebiger eingerichteter Alias eingetragen werden).

Typ imap

Imap-Server mail.g4t3.de

Port TLS auf Port 143, oder SSL auf Port 993

Authentifizierung Sichere Authentifizierung verwenden

Für den Benutzernamen wird der eigene Accountname angegeben.

Weiterhin wird noch der dazugehörige SMTP-Server angelegt. Die Konfiguration erfolgt im Thunderbird unter dem Knoten **Ausgehende Mailserver**. Dort wird eine neuer Eintrag hinzugefügt (der Name kann frei gewählt werden), mit den folgenden Einstellungen:

SMTP-Server mail.g4t3.de

Port siehe oben, 443 wird empfohlen

Authentifizierung Verwende Username und Passwort aktivieren.

Username dieses Feld ist leer zu lassen

Authentifizierung Sicher Authentifizierung verwenden

Direkt in der Hauptkonfiguration z.B. des g4t3 Accounts sollte dann diese SMTP Verbindung für ausgehende Nachrichten gewählt werden. (Rechtsklick auf einen Konteneintrag -> Eigenschaften)

In den Programm-Konfigurationswerten (Erreichbar über Einstellungen -> Erweitert -> Konfiguration bearbeiten) müssen die folgenden Werte gesetzt werden (dabei wird angenommen das der g4t3 SMTP Server der erste oder einzige ist):

network.auth.use-sspi false

network.negotiate-auth.allow-proxies true

network.negotiate-auth.delegation-uris – Defaultwert (leer) –

network.negotiate-auth.gsslib Pfad zur gssapi64.dll Diese liegt im bin Verzeichnis der Kerberos Installation. Beispielsweise: "C:\Program Files\MIT\Kerberos\bin\gssapi64.dll"

network.negotiate-auth.trusted-uris – Defaultwert (leer) –

network.negotiate-auth.using-native-gsslib false

mail.smtpserver.default.useSecauth true

mail.smtpserver.smtp1.auth_method 1

mail.smtpserver.smtp1.trySecAuth false

mail.smtpserver.smtp1.try_ssl 2

mail.smtpserver.smtp1.useSecAuth true

4.3.4 Mozilla Thunderbird für Linux

in den Konfigurationswerten (Erreichbar über Einstellungen -> Erweitert -> Konfiguration bearbeiten) müssen einige kerberosrelevante Werte angepasst werden:

network.negotiate-auth.allow-proxies true

network.negotiate-auth.delegation-uris – Defaultwert (leer) –

network.negotiate-auth.gsslib – Defaultwert (leer) –

network.negotiate-auth.trusted-uris – Defaultwert (leer) –

network.negotiate-auth.using-native-gsslib true

Es sollte ein Konto mit folgenden Daten angelegt werden:

Name Frei wählbar

E-Mail Adresse user@g4t3.de (Hier kann auch ein beliebiger eingerichteter Alias eingetragen werden).

Typ imap

Imap-Server mail.g4t3.de

Port TLS auf Port 143, oder SSL auf Port 993

Authentifizierung Sichere Authentifizierung

Für den Benutzernamen wird der eigene Accountname angegeben.

Weiterhin wird noch der dazugehörige SMTP-Server angelegt. Die Konfiguration erfolgt im Thunderbird unter dem Knoten **Ausgehende Mailserver**. Dort wird eine neuer Eintrag hinzugefügt (der Name kann frei gewählt werden), mit den folgenden Einstellungen:

SMTP-Server mail.g4t3.de

Port siehe oben, 443 wird empfohlen

Authentifizierung Verwende Username und Passwort aktivieren.

Username dieses Feld ist leer zu lassen

Direkt in der Hauptkonfiguration z.B. des g4t3 Accounts sollte dann diese SMTP Verbindung für ausgehende Nachrichten gewählt werden. (Rechtsklick auf einen Konteneintrag -> Eigenschaften)

4.3.5 Evolution auf Linux

Zuerst muss ein neues Konto angelegt werden.

Voller Name Frei wählbar

E-Mail-Adresse user@g4t3.de (Hier kann auch ein beliebiger Alias eingetragen werden).

Im Fenster Äbrufen von E-Mails"wird dann der IMAP-Server definiert.

Server-Art IMAP

Server mail.g4t3.de

Benutzername der G4T3-Nutzername

Sichere Verbindung SSL-Verschlüsselung

Legitimationsart GSSAPI

Im Fenster "Verschicken von E-Mails"wird der SMTP-Server definiert.

Server-Art SMTP

Server mail.g4t3.de

Erfordert Legitimation Haken setzen

Sichere Verbindung TLS-Verschlüsselung

Legitimationstyp GSSAPI

Benutzername der G4T3-Nutzername

Am Ende kann noch der Name des neuen Kontos eingegeben werden.

4.3.6 Mutt auf Linux

Das textbasierte mailprogramm ist auf dev.g4t3.de bereits installiert. Bei einer eigenen Installation muss darauf geachtet werden, das SASL und IMAP aktiviert sind.

Zur Konfiguration muss lediglich folgender Inhalt in die .muttrc (im Heimatverzeichniss) eingetragen werden. Dabei muss USERNAME durch den eigenen Accountnamen ersetzt werden:

```
# Automatically log in to this mailbox at startup
set spoolfile=''imaps://USERNAME@mail.g4t3.de/INBOX''

# Define the = shortcut, and the entry point for the folder browser (c?)
set folder=''imaps://USERNAME@mail.g4t3.de/INBOX''
set record=''=Sent''
set postponed=''=Drafts''
```

Sollte das G4T3 Master Zertifikat auf dem system nicht installiert sein, fragt Mutt beim erstmaligen Verbindungsaufbau nach der Korrektheit. Nach der manuellen Prüfung kann das Zertifikat permanent (mit a) akzeptiert werden.

4.3.7 Getunneltes SMTP für Windows

Soll statt dem direkten SMTP Zugang ein Tunnel verwendet werden, wird der Mail-Ausgangsserver wie folgt konfiguriert:

Smtp-Server localhost

Port 10025

Authentifizierung keine Authentifizierung

Der Port kann hier natürlich auch anders gewählt werden, muss aber mit dem weitergeleiteten Port des Tunnels übereinstimmen.

Vor dem Versenden von Mails muss dann ein Tunnel zu dem Mailserver aufgebaut werden. Dazu wird in Windows mittels PuTTY eine spezielle Verbindung eingerichtet:

- Der Zielhost ist dev.g4t3.de
- Der Port unverändert 22
- Die Kerberos Authentifizierungsoption muss gesetzt werden (die restlichen Authentifizierungsoptionen sollten nicht gesetzt werden (Forwarding auch nicht).
- Ein Tunnel mit **Source Port: 10025** und **Destination: mail.g4t3.local:25** muss hinzugefügt werden.
- das Ganze wird unter dem sessionnamen **thunderbird** abgespeichert.

Diese Verbindung kann manuell gestartet werden, oder aber mit einer Verknüfung auf **putty.exe -load thunderbird**.

4.3.8 Getunneltes SMTP für Linux

Soll statt dem direkten SMTP Zugang ein Tunnel verwendet werden, wird der Mail-Ausgangsserver wie folgt konfiguriert:

Smtp-Server localhost

Port 10025

Authentifizierung keine Authentifizierung

Der Port kann hier natürlich auch anders gewählt werden, muss aber mit dem weitergeleiteten Port des Tunnels übereinstimmen.

Vor dem Versenden von Mails muss dann ein Tunnel zu dem Mailserver aufgebaut werden. Dazu kann das Kommando **ssh -L 10025:mail.g4t3.de:25 -n dev.g4t3.de &** verwendet werden.

4.4 Mailclient Fehlerbehebung

4.4.1 Mozilla Thunderbird

Wenn die Mails nach dem Versand nicht in das Gesendet (oder Sent) Verzeichnis kopiert werden, so kann man versuchen in den entsprechenden Konten-Einstellungen unter Kopien und Ordner die Einstellung "Anderer Ordner" auswählen und dann über die eigene IMAP-Inbox auf Gesendet navigieren.

4.4.2 Getunneltes SMTP

Wenn keiner der SMTP Ports angesprochen werden kann, oder der Mailclient keine Kerberos-Authentifizierung unterstützt, kann auch ein ssh Tunnel auf einen der g4t3-server verwendet werden um eine ausreichende Autorisierung zum Mailversand zu gewähleisten.

Bei diesem Zugang wird die Authentifizierung über den vorhandenen Tunnel hergestellt, genauer genommen durch die Tatsache dass der Mailursprung aus einem privaten Server-Adressbereich stammt. Das heißt, bevor Mails versendet werden können muss stets ein spezielle Verbindung mit dem Server hergestellt werden. Als Tunnelendpunkt wird hier dev.g4t3.de empfohlen, wobei der lokale Port auf **mail.g4t3.de:25** gemappt werden muss. Wenn einer der anderen Ports verwendet wird, wird ein guter Client aufgrund der unpassenden Zertifikate zumindest eine Warnung ausgeben, evtl. sogar die Verbindung verweigern.

Ein Mailabruf ist damit jedoch nicht direkt möglich, da hier nicht nur das Vorhandensein eines Accounts, sondern auch der spezifische Account geprüft werden müssen (Jeder soll ja nur auf seine eigenen Mails zugreifen).

4.4.3 Tunnelprobleme unter Windows

Wenn der Server nicht erreichbar sein sollte (aus dem Mailclient heraus), überprüfe ob der Tunnel korrekt eingerichtet ist. Ein **netstat -abnp tcp** sollte folgende Zeile enthalten:

```
TCP    127.0.0.1:10025    0.0.0.0:0    ABHÖREN    720    [putty.exe]
```

4.4.4 Tunnelprobleme unter Linux

Wenn der Server nicht erreichbar sein sollte (aus dem Mailclient heraus), überprüfe ob der Tunnel korrekt eingerichtet ist. Ein **netstat -anp tcp –inet** sollte folgende Zeile enthalten:

```
TCP  0  127.0.0.1:10025    0.0.0.0:0    ABHÖREN  xxxx/ssh
```

4.5 Mail Sammeldienst

Auf dem Mailserver läuft ein Subsystem, welches für jeden Mailuser eine beliebige Anzahl von Mailkonten auf anderen Servern abrufen und in ein beliebiges Mailkonto übertragen kann. Die abzurufenden Mailkonten werden über LDAP Verzeichniseinträge eingerichtet, somit kann jeder Nutzer seine eigenen Einträge verwalten. Alle eingerichteten Konten werden alle 5 Minuten auf neue Mails überprüft (polling). Demzufolge ist diese Methode der Mail-Aggregation etwas schlechter als eine vom Quellserver vorgenommene direkte Weiterleitung. Solch eine Weiterleitungs-Option wird allerdings nicht von jedem Mailserver angeboten, und bei jenen die es nicht tun, bietet sich der Sammeldienst an.

Da der Sammeldienst das Passwort des abzurufenden Mailkontos benötigt, muss dieses ebenfalls in dem entsprechenden LDAP Eintrag hinterlegt werden (im Klartext). Die Einträge sind zwar geschützt, denn nur der User und der Sammeldienst haben vollen (lesenden) Zugriff auf die Einträge, LDAP Administratoren haben Zugriff auf die Einträge, nicht aber die Passwort-Attribute. Aber, ein Programm mit root Rechten auf dem LDAP Server oder dem Root-Server kann natürlich ebenfalls Zugriff auf die Daten erlangen. Insbesondere kann der Server-Administrator also auf diese Einträge zugreifen, auch auf die Passwörter.

Demzufolge sollte, so ein Konto mit diesem Mechanismus automatisch abgerufen werden soll, das Zugangspasswort vorher auf ein neues, langes (es muss ja nicht mehr manuell eingegeben werden) und möglichst zufällig generiertes, gesetzt werden. Zum einen erhöht das die Wiederstandskraft gegen Brute-Force Angriffe, und zum anderen kann mit diesem Passwort dann weder in ein anderes System eingebrochen werden, noch auf Teile anderer Passwörter geschlossen werden. Das Programm **pwgen** ist ein tool, mit dem man solche Passwörter erzeuchen kann. Es findet sich z.B. auf **dev.g4t3.de**.

Weiterhin sollte, so möglich, eine SSL gesicherte Verbindung verwendet werden. Andernfalls wäre es für einen Angreifer aufgrund der starken regelmäßigkeit und damit Vorhersagbarkeit der Zugriffe recht leicht, ein im Klartext übertragenes Passwort mitzulesen (so er irgendwie an die Route dazwischen kommt).

Die LDAP Einträge sind in dem Pfad **ou=fetch,ou=mail,dc=g4t3,dc=de** untergebracht. Die Klasse ist **fetchmailAccount** und erlaubt die folgenden Attribute:

mailAddress Der Schlüsselname, die vollständige Mailadresse, welche abgerufen werden soll.

mailDrop Eien beliebige Mailadresse, an die die abgerufenen Mails weitergeleitet werden sollen.

mailServerProtocol (IMAP | POP) Dies spezifiziert das Zugriffsprotokoll, welches verwendet werden soll.

mailServerName Der voll-Qualifizierte DNS Name des Servers, auf welchem der entsprechende Zugansserver für das Mailkonto gehostet wird.

mailServerLoginName Der username mit welchem sich authentifiziert werden soll.

mailServerLoginPasswd Das Authentifizierungspasswort. Dies sollte frisch generiert sein. LDAP Administratoren haben keinen Zugriff auf dieses Attribut.

owner Der user, welcher administrative Rechte für diesen Eintrag erhalten soll (das ist in der Regel der anlegende user).

enabled (TRUE | FALSE) Wenn dieses Attribut auf FALSE gesetzt ist, wird das Mailkonto nicht durch den Sammeldienst abgefragt. Dies ermöglicht es, ein Konto zu deaktivieren ohne es löschen und später wieder neu hinzufügen zu müssen.

useSSL (TRUE | FALSE) Spezifiziert, ob die SSL Version des Zugangsprotokolls genutzt werden soll oder nicht.

fetchReadMail (TRUE | FALSE) Wenn dies auf TRUE gesetzt ist, werden auch bereits als gelesen markierte Mails von dem Mailserver abgerufen. Dies ist nützlich falls man separat noch per Mailclient auf das Konto zugreift.

keepMailOnServer (TRUE | FALSE) Hier verhindert TRUE, dass abgerufene Mails gelöscht werden. In Kombination mit der **fetchReadMail** Option wird dies zu einem höheren Download Aufwand führen.

mailServerPort Sollte der Mailserver auf einem nichtstandart-Port hören (oder muss z.B. ein Tunnel verwendet werden), so kann der entsprechende Port hier eingetragen werden. Wenn dieses Attribute nicht spezifiziert ist, wird der passende (je nach SSL oder nicht) Standard-Port verwendet.

4.6 Mail-Aliasaddressen

Jeder Mailuser kann beliebige Aliasnamen auf sein Mailkonto einrichten, d.h. belibige Adressen der Form **name@g4t3.de**. Auch diese Mail-Aliasnamen werden über LDAP Einträge spezifiziert. Sie befinden sich in **ou=alias,ou=mail,dc=g4t3,dc=de** und sollten die Klasse **mailTranslationTable** enthalten. Diese Klasse enthält folgende Attribute:

owner Derjenige user, welche administrative Rechte an diesem Alias hat.

enabled (TRUE | FALSE) Mit FALSE kann dieser Alias (vorübergehend) deaktiviert werden.

localMailAlias Der aliasname, welcher umgesetzt werden soll (ist schlüsselattribut).

mailDrop Die Mailadresse auf welche die unter dem alias eingehenden Mails weitergeleitet werden.

description Eine optionale kurze Beschreibung des Aliasnamens.

4.7 Mailverteiler

Alle vorhanden Gruppen sind automatisch als Mailverteiler verwendbar. Um neue statische Verteiler einzurichten wird eine neue Gruppe benötigt. Derzeit können solche Gruppen nur von einem Administrator angelgt werden, dies sollte allergind nur ein kleines Hindernis darstellen.

4.8 Mailweiterleitung

Sollte jemand seinen Mail-Account auf g4t3.de nicht direkt abrufen ist es ratsam, die dort ankommenden Mails an eine andere Adresse weiterzuleiten.

Dazu reicht es aus, die eigene Mail-Adresse (Attribut **mail** im LDAP) auf die gewünschte Zieladresse zu setzen. Wenn die Mails zugleich noch an das Mailkonto auf g4t3.de gleitet werden sollen, kann auch ein doppelter **mail** Eintrag verwendet werden.

5 HTTP / HTTPS

Seiten, welche auf dem Server liegen können mit einem Zugriffsschutz versehen werden der gegen Kerberos authentifiziert und somit keine weitere Anmeldung erfordert. Um diese Möglichkeit zu nutzen muss ein Browser verwendet werden welcher mit Kerberos zusammenarbeiten kann. Ein Beispiel dafür ist **Mozilla Firefox**

5.1 Voraussetzungen

- Kerberos muss eingerichtet sein
- www.g4t3.de muss erreichbar sein.
- Das G4T3 Masterzertifikat
  `http://www.g4t3.de/download/g4t3-ca-master.crt`

5.2 Programme

Linux user können die regulären Mozilla Firefox Pakete verwenden.

5.3 Konfiguration

5.3.1 Allgemein

Nach der Installation des Firefox sollte das Zertifikat eingestellt werden. Das kann im Firefox einfach durch den Klick auf den entsprechenden Link im Download-Bereich `http://g4t3.de/download/g4t3-ca-master.crt` geschehen. Das Zertifikat sollte für alle drei möglichen Anwendungszwecke abgesegnet werden.

5.3.2 Linux

Mit dem Filter **nego** sind diese Einstellungen zu setzen:

network.negotiate-auth.allow-proxies true

network.negotiate-auth.delegation-uris https://

network.negotiate-auth.gsslib –LEER–

network.negotiate-auth.trusted-uris https://

network.negotiate-auth.using-native-gsslib true

5.4 Test

Firefox starten und in die Adresszeile `https://g4t3.de` eingeben. Die Seite sollte, eventuell nach einer Anmeldeaufforderung des installierten Kerberos-Frontends, geladen werden könne. Dabei sollten keine Fehlermeldung (weder Zertifikat-Frage, noch Autorisierung) auftreten.

5.5 Fehlerbehebung

5.5.1 Access denied

Wenn die Webseite wegen eines **Access denied** nicht geladen werden konnte, überprüfe folgendes:

1. Ist ein Ticket vorhanden (je nach Kerbero Frontend)?
2. Sind die oben angegebenen about:config Einstellungen richtig gesetzt?
   Bei Abweichenden Einstellungen sind diese zu korrigieren.
3. Überprüfe die Rechner Uhr. Am besten gleiche sie direkt mit einem Internet Zeitserver ab (pool.ntp.org)
4. Lösche eventuell vorhandene Tickets. Wähle nun die oben angegebene Webseite nochmals an.
   Dies sollte fehlschlagen oder eine Anmeldeaufforderung des Kerberos-Frontends nach sich ziehen.

5.5.2 Zertifikat-Warnung

Wenn der Browser die Gültigkeit des Zertifikates von `https://www.g4t3.de` abfragt, stelle sicher dass das G4T3-Master-Zertifikat korrekt in den Browser importiert wurde.

6 FTP

Auch der Datentransfer per FTP ist über Kerberos abgesichert und erfrodert (nach der initialen Anmeldung per Kerberos) keine Passworteingabe.

6.1 Voraussetzungen

- FTP Zugangsberechtigung (im full-account enthalten)
- Kerberos muss eingerichtet sein.
- **data.g4t3.de** muss erreichbar sein.

6.2 Programme

- yafc (mit GSSAPI Unterstüzung kompiliert)
  Achtung! Die Standard-Pakete von Debian und Ubuntu haben keine GSSAPI Unterstüzung.
  Hinweis Das Gentoo yafc ebuild hat einen Bug (223693). Um das Gentoo yafc ebuild mit mit-kerberos kompilieren zu können muss temporär /usr/include/gssapi.h entfernt (verschoben) werden.

6.3 Konfiguration

Der FTP Server läuft auf **data.g4t3.de** und horcht dort auf dem Standard Port **21**. Um eine Verbindung herzustellen, muss eine Client das passende Service-Session Ticket besitzen (das heißt der Client muss Kerberos unterstützen, und es muss ein TGT vorhanden sein). In der Regel wird der passive Modus für den Datentransfer verwendet werden.

6.3.1 yafc

Es ist keine weeitere Konfiguration vornöten.

7 SVN

Auf dem Server können eine beliebige Anzahl von SVN Repositories betrieben werden, sei es für den ausschließlich eigenen Gebrauch, oder für eine Gruppe.

Zugriff auf die SVN Repositories erfolgt direkt über das svn Protokoll:

7.1 Voraussetzungen

- Kerberos muss eingerichtet sein.
- **data.g4t3.de** muss erreichbar sein.

7.2 Programme

Windows 7 64Bit oder Vista 64Bit Hierfür existiert zwar bereits eine TortoiseSVN 64Bit Version:
`http://downloads.sourceforge.net/tortoisesvn/TortoiseSVN-1.5.4.14259-x64-svn-1.5.3.msi?download` (64-Bit)
Allerdings fehlt derzeit (März 2009) noch das die benötigte saslGSSAPI.dll in einer 64Bit Version. Somit kann SVN als 64Bit Lösung derzeit nicht unter Windows verwendet werden. Als Workaround könnten Repositories über dev.g4t3.de in das AFS Home ausgecheckt (und später eingecheckt) werden um anschließend von Windows aus über AFS editiert zu werden.

Linux Allgemein Um unter Linux mittels GSSAPI (Kerberos) und SVN die Repositories anzusteuern wird ein subversion client der Version 1.5 oder höher mit SASL Support benötigt. Das SASL GSSAPI Plugin muss ebenfalls installiert sein.

Debian Etch Das **subversion** Paket ist zu alt, bis jetzt ist keine Alternative bekannt.

Debian Lenny Das **subversion** Paket ist ok, es wird zusätzlich noch das **libsasl2-modules-gssapi-mit** Paket benötigt.

Gentoo **dev-util/subversion** mit einer Version über 1.5.0 mit dem **sasl** USE-Flag funktioniert. Weiterhin muss **dev-libs/cyrus-sasl** mit dem USE-Flag **kerberos** kompiliert sein.

7.3 Konfiguration

7.3.1 Subversion

Für den einfachen (Client) Zugang bedarf es unter Linux keiner weiteren Konfiguration.

7.4 Verwendung

Vor dem Zugriff auf Kerberos-basierte SVN Repositories muss ein Ticket angefordert werden. Je Nach Anmeldeclient wird, sollte dieses zum Zugriffszeitpunkt nicht vorhanden sein, die Eingabe der User-Credentials gefordert. Dies ist jeduch nur einmalsig im Gültigkeitszeitraum des Tickets notwendig.

Wenn ein Ticket vorhanden ist, können vorhandenen Repositories unter der URL **svn://data.g4t3.de/REPOSITORY** angesprochen werden. **REPOSITORY** muss hierbei natürlich durch einen gültigen namen ersetzt werden.

7.5 Test

Als Beispiel kann das Repository dieser Dokumentation augecheckt werden, die dazugehörige URL lautet **svn://data.g4t3.de/g4t3-doc/trunk**.

Dazu bedarf es folgender Schritte:

7.5.1 Linux

Öffne eine Konsole und gebe **svn co svn://data.g4t3.de/g4t3-doc/trunk g4t3-doc** ein.

Der checkout sollte erfolgreich durchgeführt werden können, in dem g4t3-doc erzeichniss sollten nun die LaTeX Quellen zi diesem Dokument aufgeführt sein.

7.6 Migration von Alten Checkouts (Working Copy)

7.6.1 Linux

Hier wechselt man in das Wurzelverzeichnis ders checkout-Verzeichnisses.

Dann kann mit **svn info | grep URL** die ursprüngliche, alte URL des Repositories ermittelt werden. Zum Beispiel *svn+ssh://dev.g4t3.de/var/prj/g4t3-doc/trunk*.

Die neue URL kann nun durch Ersetzen des ersten Teils (bis zum Repository-Namen mit **svn://data.g4t3.de/** gebildet werden. Dies ergibt in diesem Beispiel also *svn://g4t3-doc/trunk*.

Immer noch im Wurzelverzeichnis des Checkouts kann mit dem dem Kommando **svn switch –relocate URL-ALT URL-NEU** der Checkout an die neue Position auf dem Server angepasst werden. Das gibt in dem Beispielfalle also:

svn switch –relocate \
svn+ssh://dev.g4t3.de/var/prj/g4t3-doc/trunk \
svn://data.g4t3.de/g4t3-doc/trunk

8 Synchronisierung

Hier geht es um ein Tool, um den Inhalt von lokalen Verzeichnissen mit dem Inhalt von Verzeichnissen auf dem Server abzugleichen. Unison wurde an der Universität von Pensilvania entwickelt, und stellt eine Synchronisierungslösung dar, welche sowohl unter Windows als auch Linux verfügbar ist.

Mittlerweile wurde die Weiterentwicklung offiziell zwar eingestellt, das Programm liegt aber in einer stabilen Version vor. Desweiteren ist eine kleine Community dabei immer wieder neue Versionen zu entwickeln. Unison basiert auf dem unter Unix üblichen rsync Protokoll.

8.1 Voraussetzungen

- Unison 2.13.16

  Debian unison oder unison-gtk

  Gentoo unison

- SSH mit GSSAPI muss eingerichtet sein.
- **dev.g4t3.de** muss per ssh erreichbar sein (mit Kerberos)

8.2 Konfiguration

Es bedarf keiner weiteren Konfiguration, allerdings muss sehr genau auf gleiche Versionsnummern geachtet werden (es wird Version 2.13.16-2 verwand). Es muss sehr genau auf die korrekte Version geachtet werden (Version 2.13.16-2).

Weiterhin müssen vor der Verwendung noch einige Umgebungsvariablen eingerichtet und ein wenig mit den Einstellungen getrickst werden. Dazu wählt man **Start -> Einstellungen -> Systemsteuerung** und dort **System -> Erweitert -> Umgebungsvariablen**.

Hier gibt es zwei Vorgehensmöglichkeiten, Es können alle Variablen Lokal, das heist nur für den gerade angemeldeten User gesetzt werden, oder ein Teil (der Userunabhängige Teil) wird allgemein für alle User auf dem System gesetzt. Auf euren Rechnern könnt ihr die Variablen besser für alle User setzen.

Je nach dem müssen die entsprechenden Variablen im oberen (user) oder unteren (system) Teil gesetzt werden, für Systemweite Variablendefinition benötigt man Administratorrechte.

PATH hier sollte der Pfad zu dem modifizierten Putty-Verzeichniss angehängt werden. Wichtig, dieser Pfad darf keine Leerzeichen enthalten (Leerzeichen in anderen Pfaden machen aber keinerlei Probleme).

UNISON (optional) gibt den Namen (und Pfad) des Verzeichnisses an, welches von Unison für interne Daten und Profile verwendet wird. Ist dies nicht gesetzt wird $HOME/.unison verwendet.

HOME (optional) gibt den von einigen Programmen verwendeten Pfad zu dem Heimatverzeichniss an. Dies wird von Windows nicht standardmäßig gesetzt, und sollte rpo Konto manuell gesetzt werden. Wenn dieser Pfad definiert ist, werden die Unison dateien und Profile in dem unterverzeichniss .unison in eben diesem Pfad abgelegt (empfohlene Variante). Ist diese Variable auch nicht definiert, so wird C:\.unison verwendet.

Unison benötigt einen SSH Client um eine abgesicherte Verbindung zum Zielhost herzustellen. Eine Cygwin-Installation oder eine gewisse unfreie SSH Implemntierung sind vorgesehen, um die Vorzüge von Kerberos auch hier nutzen zu können muss aber das modifizierte Putty verwendet werden. Da dies nicht mit allen standard-ssh Komandozeilenoptionen zurechtkommt muss ein wenig getrickst werden. Es sollte in dem unison Verzeichnis eine Profildatei mit dem Namen **connection** (ohne Endung, einfache Textdatei) mit folgendem Inhalt angelegt werden:

`http://g4t3.de/download/connection`

```
rshcmd = C:/Programme/PuTTY_GSSAPI/plink.exe
rshargs = -load unison -batch
```

Dabei ist der Pfad bei **rshcmd** natürlich durch den entsprechend gültigen Pfad auf die **modifizierte plink.exe** zu ersetzen.

Weiterhin muss in dem modifizierten Putty noch eine Session für den Zielhost, **dev.g4t3.de** und den in section2 beschriebenen Kerberos Optionen (beide) angelegt werden. Die Option **„Attempt Keyboard-interactive auth“** (gleiches Submenü) sollte deaktiviert werden. Das ganze muss unter dem sessionnamen **unison** abgespeichert werden.

Es sollten Profildateien für die tatsächliche Synchronisierung eingerichtet werden. Informationen dazu findet man im Unison Handbuch (`http://www.g4t3.de/download/unison-2.13.16-manual.pdf`) als Beispiel sei hier folgende gegeben

(`http://www.g4t3.de/download/home-sync-linux.prf`)

```
root = /home/torian/
root = ssh://dev.g4t3.de/home/torian/lin-home
include connection

ignore = Name *.aux
ignore = Name *.bbl
ignore = Name *.blg
ignore = Name *.log
ignore = Name *.toc
ignore = Name {,.}*{.aux}
ignore = Path {.bash_history}
```

8.3 Test

Die Synchronisierung kann entweder per GUI angestoßen werden (da passende Profil auswählen), oder mit der Kommandozeile.

9 SQL Datenbank

Als Datenbank wird PostgreSQL verwendet, welches eine ausgereifte Kerberos-Anbindung ermöglicht, und mit pgAdmin steht auch ein Administrationswerkzeug zur Verfügung, welches kerberosfähig ist.

9.1 Voraussetzungen

- Es muss eine Datanbankberechtigung existieren (im full-account enthalten)
- **data.g4t3.de** muss erreichbar sein
- Kerberos muss eingerichtet sein

9.2 Programe

- pgAdmin 3
  `http://www.pgadmin.org`
- psql

9.3 Konfiguration

Der Datanbankserver läuft auf **data.g4t3.de** auf dem standard PostgreSQL Port **5432**. Die Wartungs-DB ist **template1**.

9.3.1 Konfiguration pgadmin3

Datei-> neuen Server hinzufügen. Der Name kann frei Gewählt werden, z.B. „G4T3 Main“, Server und Port wie oben aufgeführt. Unter Username gebt ihr euren User an, das Passwort-Feld bleibt leer und das Häcken bei Passwort speichern kann gesetzt werden. Der Rest kann auf den Standardwerten verbleiben.

Test Wenn ihr euch mit dem Server verbindet, sollte (nach dem Anklicken des Servers) dieser geladen und „Aufklappbar sein“.

9.3.2 Konfiguration psql

Es ist keinerlei weitere Konfiguration für psql erforderlich (neben der Kerberos installation). Es muss allerdings stets der Datenbankserver mittels der **-h** Option mitgeliefert werden, auch wenn man sich auf **data.g4t3.de** eingeloggt hat.

10 Andrew File System (AFS)

10.1 Voraussetzungen

- Kerberos muss eingerichtet sein.
- Ein entsprechender Account muss eingerichtet sein.

- **data.g4t3.de** muss erreichbar sein.

10.2 Programme

Debian Etch und Lenny Die Pakete **openafs-client**, **openafs-krb5** sowie **openafs-modules-source**. Die installation des Moduls sollte über den module assistant (**module-assistant**) vorgenommen werden.

Gentoo Der stable **openafs-kernel-1.4.6** und **openafs-1.4.6** kompiliert (zumindest auf 64-Bit Systemen) nicht gegen aktuelle Kernel Sourcen (2.6.25). Statt die unstable Version zu nehmen sollte hier ein anderer Kernel verwendet werden (2.6.24). Die unstable Version dieses ebuild hat sich auf Intel und AMD 64-Bit systemen als wirklich instabiel erwiesen und führte bisher immer nach kurzer nutzung zu einem einfrieren der entsprechenden Prozesse.

Ubuntu 8.04 LTS (Hardy Heron) Das mitgelieferte openafs-kernel Paket lässt sich nicht kompilieren (siehe Bug #129480).

Ubuntu 8.10 (Intrepid Ibex) Benötigt die Installation der Pakete **openafs-client**, **openafs-krb5** sowie **openafs-modules-source**. Zusätzlich ist es ratsam, den module assistant (**module-assistant**) zum installieren des Modules zu benutzen.

10.3 Konfiguration

10.3.1 OpenAFS auf Debian

Als erstes muss das Modul für Debian kompiliert werden. Falls er noch nicht installiert ist, dazu mit **$> apt-get install module-assistant** einspielen.

Dann mit **$> m-a prepare openafs** das herunterladen von allen zum kompilieren des opeafs-Kernelmoduls notwenidgen Paketen veranlassen. Hiernach kann mit **$> m-a a-i openafs** das ensprechende module erstellt werden.

Anschließend noch mit **$> apt-get install openafs-client openafs-krb5** den OpenAFS client für Linux installieren und konfigurieren.

Der Zellenname für G4T3 ist **g4t3.de**, der primäre File- und Volumeserver ist **data.g4t3.de**. Der Client kann ruhig mit dem Rest des Systems gestartet werden.

10.3.2 OpenAFS auf Gentoo

Ein **$> emerge net-fs/openafs net-fs/openafs-kernel** mit dem aktivierten USEFLAG **kerberos** sollten für die Installation ausreichen. Dabei sollten beide pakete nicht in **/etc/portage/package.keywords** aufgeführt sein.

Wenn es hierbei zu einem Compile-Fehler kommt wird vermutlich eine inkompatible Version des Linux-Kernel verwendet. In diesem Falle kann für die Version **1.4.6_p20080222** der letzte Kernel der Version **2.6.24** verwendet werden. In Jedem Falle NICHT die unstable (maskierte) Version der openafs Pakete wählen.

Anschließend kann der OpenAFS client noch in die Autostartliste eingetragen werden mit **$> rc-update add openafs-client default**.

10.3.3 OpenAFS auf Ubuntu 8.04

In Ubuntu 8.04 (Hardy Heron) kompiliert das OpenaAFS Paket der Version 1.4.6.dfsg1-2 nicht gegen den verwendeten Kernel 2.6.24-21-rt, wie in Bug #129480 beschrieben (`https://bugs.launchpad.net/ubuntu/+source/openafs/+bug/129480`). Hintergrundinformationen zu dem Begriff Tainted-Kernel und den GPLONLY Flags findet man unter `http://www.kernel.org/pub/linux/docs/lkml/#s1-18`.

Die Paketversion 1.4.7.dfsg1- behebt diesen Fehler, ist aber erst in Ubuntu 8.10 (Intrepid Ibex) enthalten (Release am 30.10.2008).

10.3.4 OpenAFS auf Ubuntu 8.10

Als erstes muss das Modul für Ubuntu kompiliert werden. Falls es noch nicht installiert ist, dazu mit **$> aptitude install module-assistant** einspielen.

Dann mit **$> m-a prepare openafs** das herunterladen von allen zum kompilieren des opeafs-Kernelmoduls notwenidgen Paketen veranlassen. Hiernach kann mit **$> m-a a-i openafs** das ensprechende module erstellt werden.

Anschließend noch mit **$> aptitude install openafs-client openafs-krb5** den OpenAFS client für Linux installieren und konfigurieren.

Der Zellenname für G4T3 ist **g4t3.de**, der primäre file- und volumeserver ist **data.g4t3.de**.

Der Client kann ruhig mit dem Rest des Systems gestartet werden, jedoch ist hier noch eine kleine Korrektur vonnöten, wie auch aus folgendem Bugreport zu entnehmen ist: `https://bugs.launchpad.net/ubuntu/+source/openafs/+bug/318605`. Das Setzen der Option **AFS_DYNROOT=true** in der Datei **/etc/openafs/afs.conf.client** ermöglicht das fehlerfreie Funktionieren von AFS.

10.3.5 Manuell Linux Konfiguration

Sollten diese Einstellungen aufgrund des installtionsmechanismus nicht abgefragt werden (Gentoo, Debian mit entsprechenden debconf-Einstellungen), so muss der Zellenname (**g4t3.de**) in **/etc/openafs/ThisCell** eingetragen werden, z.B. mit **$> echo 'g4t3.de' > /etc/openafs/ThisCell**. Weiterhin muss der g4t3.de fileserver in die **/etc/openafs/CellServDB** eingefügt sein.

Der dazugehörige Ausschnitt:
>g4t3.de
193.34.68.212 # data.g4t3.de

Eine passende Datei kann auch direkt von `http://g4t3.de/download/CellServDB` heruntergeladen werden.

Anschließend noch den client starten (mit **$> /etc/init.d/openafs-client start**) und es kann losgehen. Der AFS-Namensraum wird normalerweise in /afs gemountet.

10.4 Test

10.4.1 Linux

1. Mit **$> kinit** ein Kerberos Ticket für den User beschaffen.

2. **$> aklog** aufrufen um mit Hilfe des Kerberos Tickets ein AFS Token zu erwerben.

3. Versuchen in das Persönliche Verzeichnis im AFS zu wechseln. Dieses befindet sich unter **/afs/g4t3.de/user/USERNAME**, USERNAME ist hier durch den eigenen User zu ersetzen. Demnach also **$> ls /afs/g4t3.de/user/USERNAME**. Der Zugriff auf das AFS kann beim ersten Mal einige Momente brauchen (auch schon bei der Tab-Expansion des Pfades).

Start Allgemein gilt, erst ein Kerberos Ticket anfordern (**$> kinit**), dann ein AFS-Token (**$> aklog**).

Status Die AFS-Tokens können mit **$> tokens** angezeigt werden. Die Kerberos Tickets mit **$> klist**.

Stop Die AFS-Tokens werden durch **$> unlog** gelöscht, die Kerbero-Tickets mittels **$> kdestroy**.

10.5 Fehlerbehebung

- Die Systemzeit des eigenen Rechners muss korrekt sein. Eine Differenz von mehr als drei Minuten wird das Anfordern von Kerberos-Tickets oder AFS-Tokens verhindern.
- Die Rechner **auth.g4t3.de** und **data.g4t3.de** müssen erreichbar sein (**ping**).

10.5.1 Linux

Allgemeines Vorgehen:

1. Der OpenAFS Client muss gestartet sein (Gentoo fügt ihn z.B. nicht zum Autostart hinzu).

2. Ein aktuelles und gültiges Kerberos-Ticket muss vorhanden sein. Einmal löschen (**$> kdestroy**) und neu anfordern (**$> kinit**). Mit **$> klist** überprüfen. Dort sollte nun nur ein Eintrag für den Service principal **krbtgt/G4T3.DE@G4T3.DE** aufgeführt sein.

3. Ein neues AFS-Token anfordern (**$> aklog**). Anscließend sollte der die Ausgabe von **$> tokens** genau einen Eintrag für **tokens for afs@g4t3.de** enthalten, mit einem Ablaufdatum (Expires ...) in der Zukunft.

4. AFS verwenden.

Spezielle Fehlerfälle:

aklog bleibt ununterbrechbar stehen Das ist typischerweise der Fall wenn das Openafs-Modul Fehler aufweist und die Prozesse nicht bedient. In diesem Falle hilft kurzfristig nur der Neustart des Rechners und langfristig das austauschen des Moduls gegen eine stabile Version.

aklog: Couldn't determine realm of user:)aklog: unknown RPC error (...) while getting realm Es ist kein Kerberos Ticket vorhanden. Bitte mit **$> kinit** nachholen.

cannot open directory /afs/g4t3.de/user: Permission denied Offensichtlich fehlt das AFS-Token, mit **$>** **tokens** prüfen und mit **$>** **aklog** anfordern.

aklog: Couldn't get g4t3.de AFS tickets: aklog: unknown RPC error (...) while getting AFS ti Hier stimmt vermutlich die Systemzeit nicht mit der Zeit auf dem Fileserver überein. Mit **$>** **ntpdate dome.g4t3.de** kann die Zeit angeglichen werden.