NIS-Authentifizierung mit Kerberos

Misstrauische Tickets

Die Kombination aus Network Information Service (NIS) und Kerberos erlaubt es, viele User zu verwalten und gleichzeitig einen sicheren Login zu gewährleisten. Der Schlüssel dazu liegt in den kryptographischen Fähigkeiten von Kerberos, das zudem sicheren Ersatz für unsichere IP-Dienste mitbringt. Thorsten Scherf

Wer den Network Information Service für die zentrale Benutzerverwaltung einsetzt, hat ein Sicherheitsproblem, vor allem wenn NIS die Shadow-Passwortdatenbank ebenfalls unter seine Fittiche nimmt. Sinn der Shadow-Datei ist es, dafür zu sorgen, dass kein Benutzer Zugriff auf die verschlüsselten Passwörter erhält. Verwaltet jedoch NIS diese Datei, dann fließt bei jeder Benutzeranmeldung der verschlüsselte Passwort-String über das Netzwerk. Sollte jemand den Datenverkehr mithören, kann er diesen String mit einem Sniffer wie Ethereal abfangen und einen Brute-Force- oder einen Wörterbuch-Angriff durchführen. Auch dafür gibt es zahlreiche Programme wie John oder Crack.

An dieser Stelle greift Kerberos [1] ein. Es schützt die Passwörter mit einem besonderen Mix aus Kryptographie und Transportphilosophie. Dieser Artikel bezieht sich auf NIS in der Version 2. Die Authentifizierung übernimmt Kerberos 5. Zwar gibt es NIS bereits in der Version 3, aber für Linux existiert bisher nur ein Client. Die Server-Variante befindet sich noch in Entwicklung. Die Konfiguration von NISv3, auch NIS+ genannt, ist zudem deutlich schwieriger als bei der hier vorgestellten Version. Weitere Informationen zu NIS+ finden sich auf der NIS-Howto-Website [2].

Der dreiköpfige Hund

Viele Admins versuchen ihre Sicherheitsprobleme durch Firewalls zu lösen. Allerdings drohen Gefahren nicht nur von außen. Die meisten nicht autorisierten Zugriffe kommen aus dem lokalen Netzwerk, in dem ein Lauscher einfach Klartextpasswörter abfängt. Diese Lücke schließt die Authentifizierungs-Software Kerberos. Programmierer des Massachusetts Institute of Technology (MIT) haben Kerberos entwickelt, um die sichere Authentifizierung für Client- und Server-Anwendungen zu gewährleisten. Dazu nutzt die Software starke Kryptographie und ein auf Tickets basierendes Authentifizierungsprotokoll.

Typische Netzwerkdienste wie FTP oder POP3 gefährden die Sicherheit im Netzwerk, denn Benutzername und Passwort wandern im Klartext vom Client zum Server. Kerberos bricht diese Tradition, indem es Passwörter erst gar nicht über das Netz sendet. Somit kann es der Sniffer auch nicht abfangen. Andere Angriffe wie Spoofing und Replay-Attacken erschwert Kerberos ebenfalls. Zudem muss sich ein Client nur einmal im Netzwerk authentifizieren und erhält damit Zugang zu allen mit Kerberos gesicherten Diensten.

Kerberos kümmert sich lediglich um den eigentlichen Vorgang der Authentifizierung. Es stellt keine Benutzerinformationen wie beispielsweise die User-ID, eine Login-Shell oder das Heimatverzeichnis bereit. Diese Informationen verwaltet ein Verzeichnisdienst wie NIS (siehe Kasten „NIS aufsetzen“).

Verteilter Mechanismus

Die hohe Sicherheit erhält der Administrator durch einen ausgefeilten Mechanismus (siehe Kasten „Ablauf einer Kerberos-Session“). Der Client baut eine Verbindung zu einem Key Distribution Center (KDC) auf (Abbildung 2). Das geschieht entweder für den User transparent über das Anmeldeprogramm »login« oder mit Hilfe des Clients »kinit«. Das KDC besteht aus zwei Teilen: dem Authentication Server (AS) und dem Ticket Granting Server (TGS). Der Authentication Server empfängt die Anfrage des Clients und prüft im Namensraum (Realm), ob der Benutzername (User Principal) überhaupt berechtigt ist, auf die Dienste zuzugreifen.

Befindet sich der Principal in der Kerberos-Datenbank, erzeugt der AS einen zufälligen Session Key und ein so genann-

tes Ticket Granting Ticket (TGT). Dieses TGT enthält verschiedene Informationen wie den Hostnamen und die IP-Adresse des Clients sowie die Gültigkeitsdauer des Tickets, einen Zeitstempel und den eben erzeugten Session Key.
Dieses TGT kodiert Kerberos mit einem Schlüssel, der nur dem Authentication Server und dem Ticket Granting Server bekannt ist. Zusammen mit dem eben erzeugten Session Key schickt der Server das Ticket an den Client. Um Mitleser zu enttäuschen, ist das Ticket mit einem Schlüssel kodiert, den Kerberos aus dem Passwort des Clients berechnet hat.

Jetzt kommt das Passwort

Nachdem der Client die Antwort des Authentication-Servers erhalten hat (kodiertes TGT und Session Key), schaltet das lokale System den Passwortprompt zur Anmeldung frei. Das Passwort konvertiert der Client zu einem DES-Schlüssel. Dieser dient zur Dekodierung des eben empfangenen TGT. Der Client speichert das TGT in seinem Credential Cache und löscht das eingegebene Passwort aus dem Speicher. Der Benutzer weist durch das TGT seine Identität nach, allerdings nur solange das Ticket gültig ist. Läuft das Ticket ab, muss er sich neu einloggen und die ganze Prozedur beginnt wieder von vorne.
Die Authentizität des Users ist mit Passwort und TGT für die lokale Workstation verifiziert. Möchte der Benutzer auf einen Netzwerkdienst wie FTP zugreifen, muss er ein Service Ticket vom Key Distribution Center anfordern. Dazu wendet sich der Client an den Ticket Granting Server (TGS). Dieses Service Ticket (ST) ist für genau diesen einen Dienst zuständig, für den es der Client anfordert, beispielsweise für FTP.

Service Ticket verlangen

Die Anforderung des Service Ticket ist um einiges komplexer als beim TGT: Der Client sendet eine Anfrage an den TGS. Sie besteht aus dem Namen des Dienstes, auf den der Client zugreifen möchte (Authenticator), und dem gespeicherten TGT. Der Authenticator enthält den Namen des Clients, seine IP-Adresse sowie die aktuelle Zeit des Clients und ist mit dem Session Key des TGT kodiert.
Das verschlüsselte TGT sendet der Client zusammen mit dem Authenticator an den Ticket Granting Server. Dieser dekodiert den Authenticator und das TGT und vergleicht den Inhalt sowie die IP-Adresse und die Uhrzeit. Sind die Informationen identisch, generiert der Server einen neuen Session Key, den in Zukunft der Client und der angesprochene Dienst (hier der FTP-Server) benutzen. Der neue Session Key ist im Service Ticket enthalten, das der Ticket Granting Server nun ausstellt. Alles zusammen sendet der Server mit dem Session Key des TGT verschlüsselt an den Client.
Daraufhin beginnt das Ticketgeschiebe wieder von vorne. Der Client empfängt das Service Ticket und reicht es an den gewünschten Server (FTP) weiter, um seine Identität nachzuweisen. Zusätzlich zum Service Ticket erzeugt der Client wieder einen Authenticator und schickt diesen an den Server. Stimmen alle Informationen des ST und des Authenticators überein, stuft der Server den Client als echt ein. Der Client ist somit authentifiziert und muss sich nicht noch einmal mit Usernamen und Passwort gegenüber dem Server ausweisen.

Kein Schutz ohne NTP

Der Authenticator schützt effektiv davor, dass ein Angreifer den Netzwerkverkehr mitliest und ein Service Ticket abfängt. Denn er könnte dies später einem Server anbieten, um Zugang zu erhalten (Replay-Attacke). Damit die Authentifizierung des Clients funktioniert, ist es zwingend erforderlich, dass alle Rechner im lokalen Netzwerk die gleiche Zeit benutzen. Das lässt sich mit einem NTP-

NIS-Client konfigurieren

Die Konfiguration eines NIS-Clients ist unter Red Hat Linux recht einfach, da das Tool »authconfig« beim Setup hilft (Abbildung 1). Hier sind die NIS-Domäne sowie der NIS-Server einzutragen, die das Werkzeug in die Datei »/etc/yp.conf« schreibt (Listing 1). Danach passt Authconfig die Datei »/etc/sysconfig/network« an. Hier fügt es nochmals den NIS-Domänennamen ein, den das System beim nächsten Start ausliest.
Im dritten Schritt konfiguriert es die Nameserver-Switchdatei »/etc/nsswitch.conf«. In ihr ist festgelegt, wo und in welcher Reihenfolge der Client nach Informationen wie Passwort- oder Host-Dateien suchen soll. Folgende Einträge sollten vorhanden sein:

```
passwd: files nis
shadow: files
group: files nis
```

Der Client konsultiert hier zuerst die lokalen Dateien »/etc/passwd« und »/etc/group« und wendet sich dann den NIS-Maps »passwd« und »group« zu. Ist keine lokale Authentifizierung gewünscht, kann der Eintrag »files« entfallen. Allerdings ist dann auch kein lokaler Root-Login mehr möglich.
Zuletzt bearbeitet Authconfig die PAM-Datei »/etc/pam.d/system-auth«. Hier stellt es ein, dass ein Benutzerpasswort auf dem NIS-Server geändert wird, sofern es sich um einen NIS-Account handelt. Ein User gibt ein neues Passwort danach einfach mittels »yppasswd« ein. Zum Schluss startet das Tool den NIS-Client-Dienst »ypbind« im Hintergrund, der die Verbindung zum NIS-Server herstellt.

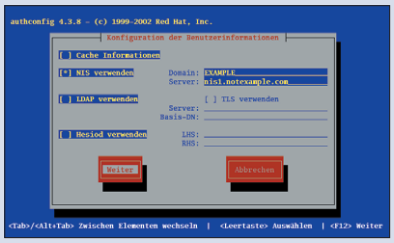


Abbildung 1: Mit Hilfe von Authconfig trägt der Administrator den NIS-Domänennamen und den NIS-Server in die entsprechenden Dateien ein.

Listing 1: Konfiguration des NIS-Clients

```
01 # Valid entries are
02 #
03 # domain NISDOMAIN server HOSTNAME
04 #       Use server HOSTNAME for the domain NISDOMAIN.
05 #
06 # domain NISDOMAIN broadcast
07 #       Use  broadcast  on  the local net for domain
   NISDOMAIN
08 #
09 # ypserver HOSTNAME
10 #       Use server HOSTNAME for the  local  domain. The
11 #       IP-address of server must be listed in /etc/hosts.
12 #
13 # broadcast
14 #       If no server for the default domain is specified
15 #        ornone of them is rechable, try a broadcast call
16 #        tofind a server.
17 #
18 domain EXAMPLE server nis1.notexample.com
```

Server (Network Time Protocol) sicherstellen [3]. Allerdings ist NTP selbst ein unsicherer IP-Dienst, sodass er ebenfalls mit Kryptographie vor neugierigen Augen geschützt sein muss. Über die Möglichkeiten, NTP mit Schlüsseln zu sichern, klärt ebenfalls [3] auf.

Der Kerberos-Server

Der Kerberos-Server verwaltet die Datenbank, in der die Principals gespeichert sind. Die Kerberos-Maschine sollte besonders gesichert sein und am besten in einem eigenen verschlossenen Raum oder Rack residieren. Auf keinen Fall dürfen zusätzliche Dienste auf diesem Rechner laufen, um sich möglichst keine Schwachstelle einzuhandeln.

Principals existieren sowohl für Benutzer als auch für die auf Kerberos basierenden Dienste und Hosts. Ein Principal hat folgenden Aufbau: »*Primary/instance@REALM*«. Instance ist optional; es dient lediglich zur Gruppierung des Primary. Ein User Principal könnte wie »thorsten/admin@NOTEXAMPLE.COM« aussehen, ein FTP-Server hätte beispielsweise »ftp/station1.notexample.com@NOTEXAMPLE.COM« als Principal.

Der Realm fasst alle Principals eines Bereichs zusammen und entspricht dem großgeschriebenen DNS-Domänennamen. Zusammen mit den Principals nimmt die Kerberos-Datenbank auch die Passwörter der Benutzer sowie die Dienste in der Datenbank auf.

Der Server ist relativ leicht konfigurierbar. In der Datei »/etc/krb5.conf« ist der Kerberos-Realm einzutragen (Abbildung 3). Mit dem Kommando »kdb5_util create« erzeugt der Administrator die Datenbank im Verzeichnis »/var/kerberos/krb5kdc«. Die Verwaltung der Datenbank erfolgt entweder lokal mit dem Tool »kadmin.local« oder remote mittels

Listing 2: Securenets

```
01 # Zugang für »localhost« explizit erlauben:
02 host            127.0.0.1
03
04 # Zugang für Rechner aus dem Netz 192.168.0.0/24
   erlauben:
05 255.255.255.0   192.168.0.0
06
07 # Zugang jedem Rechner gestatten (Vorsicht!):
08 #0.0.0.0        0.0.0.0
```

NIS aufsetzen

Ein NIS-Server verwaltet Benutzer- und Host-Informationen, die Clients in derselben NIS-Domäne abfragen können. Der Administrator erhält durch NIS eine Zentrale für die Benutzer- und Host-Verwaltung. Anwender greifen von einem beliebigen NIS-Client auf diese Datenbank zu, unter anderem um sich zu authentifizieren (siehe Kasten „NIS-Client konfigurieren"). Dazu lässt der Administrator einfach die Dateien »/etc/passwd« und »/etc/group« von NIS verwalten. Auch mit Shadow-Passwörtern kommt der NIS-Server zurecht, sofern eine Glibc installiert ist. Ältere Bibliotheken wie Libc5 verweigern die Zusammenarbeit.

NIS-Server konfigurieren

Bevor es an die eigentliche Konfiguration des Servers geht, wählt der Administrator den Namen der neuen NIS-Domäne. Er sollte nicht den Namen der DNS-Domäne verwenden, da Einbrecher diesen oft als zuallererst ausprobieren. In diesem Artikel heißt die NIS-Domäne »example«. Sie ist mit der Anweisung »NISDOMAIN=example« in die Datei »/etc/sysconfig/network« einzutragen. Danach erst geht es an die eigentliche Arbeit: die Konfiguration des Makefiles, das im Verzeichnis »/var/yp« liegt.

Die meisten Einträge muss der Verwalter nicht anpassen. Mit der Anweisung »MINUID« legt er fest, ab welcher User-ID NIS die Benutzer verwaltet. Die Option »MINGID« bestimmt entsprechend die Gruppen. »MERGEPASSWD« ist in diesem Beispiel zwingend auf »false« zu setzen, da NIS keine Passwörter verifizieren soll. Im letzten Schritt ist mit Hilfe von »all: passwd group« anzugeben, dass NIS die Benutzerkonten und Gruppen verwalten soll. NIS kann zwar mehr, das ist für die Beispielkonfiguration aber nicht relevant.

NIS arbeitet nicht mit den eigentlichen Ascii-Dateien »/etc/passwd« und »/etc/group«, sondern generiert mit Hilfe des Tools »makedbm« GDBM-Dateien, die im NIS-Jargon Maps heißen. Makedbm erzeugt aus jeder Datei zwei NIS-Maps, die nach verschiedenen Kriterien sortiert sind. Die Passwd-Map ordnet es zum Beispiel nach Login-Namen (»passwd.byname«) und nach User-IDs (»passwd.byuid«). Die NIS-Maps befinden sich nach der Initialisierung des Servers mit dem Kommando

```
/usr/lib/yp/ypinit -m
```

in einem Ordner unterhalb von »/var/yp«. Der Ordner trägt den Namen der NIS-Domäne. Die NIS-Datenbank enthält lediglich jene Benutzer, die zu diesem Zeitpunkt vorhanden sind. Kommen später weitere User oder Gruppen hinzu, müssen sie in die NIS-Maps einfließen. Das erreicht der Administrator mit dem Kommando »make -C /var/yp«.

Der NIS-Server nimmt seine Arbeit auf, sobald die Dienste »portmap« und »ypserv« laufen. Den Ablauf stellt der Administrator in den verschiedenen Runlevel-Verzeichnissen unter »/etc/init.d/« ein.

Redundanz und Sicherheit

Bei einem zentralisierten Verzeichnisdienst wie NIS ist es immer geschickt, einen zusätzlichen Server aufzusetzen. Auf diese Weise erreicht der Verwalter, dass sich die Benutzer weiterhin am Netzwerk anmelden können, selbst wenn der primäre NIS-Server ausfällt. Der sekundäre Server lässt sich einfach aufsetzen, beim »ypinit«-Kommando ist die IP-Adresse des Master-Servers anzugeben:

```
/usr/lib/yp/ypinit -s Master-Server-IP
```

Fortan bezieht der zweite Rechner alle Maps vom Master-Server. Vorher sind allerdings noch zwei Sachen zu erledigen: Der sekundäre Server muss als NIS-Client arbeiten und die Datei »/var/yp/ypservers« auf dem Master muss auf diesen verweisen.

Außerdem sollte sich der Administrator Gedanken über die Sicherheit seiner NIS-Sever machen. Jeder NIS-Client kann sich jede von NIS verwaltete Datei anzeigen lassen. Ein einfacher Aufruf von »ypcap passwd« genügt, schon präsentiert der Server die Passwd-Map. Passwd ist hier ein Alias für »passwd.byname« und »passwd.byuid«; weitere Aliase befinden sich in der Datei »/var/yp/nicknames«.

Zwar erhält der Client nicht die verschlüsselten Passwörter der User, da dies die Anweisung »MERGEPASSWD=false« im Makefile verhindert. Was der Server zeigt, reicht jedoch aus, um bereits umfangreiche Informationen über Benutzernamen zu sammeln. Daher sollte der Administrator den Zugriff auf die NIS-Maps unbedingt einschränken.

Speziell für diese Aufgabe gibt es die Datei »/var/yp/securenets« (Listing 2). Die zugelassen Netzwerke finden hier einen Platz. Auf IP-Ebene ist mit entsprechenden Paketfilterregeln eingeschränkter Zugriff auf den Portmapper zu erreichen. Eine Netfilter-Regel auf der Firewall könnte beispielsweise so aussehen:

```
iptables -t filter -A FORWARD -p udp ➥
  -dport 111 -s 192.168.0.0/24 ➥
  -d 192.168.0.100 -j ACCEPT
```

Dabei entspricht 192.168.0.0/24 dem zugelassenen Netzwerk und auf 192.168.0.100 residiert der Portmapper. Connection Tracking [7] und eine Anpassung der Standardpolicy für diese Chain wird hier vorausgesetzt. Allerdings ist daran zu denken, dass jetzt auch alle anderen Dienste, die vom Portmapper abhängig sind, mit einer Zugangsbeschränkung belegt sind. Gleiches gilt, wenn der TCP-Wrapper den Portmapper schützt.

Ablauf einer Kerberos-Session

❶ Der User meldet sich am System an und sendet eine Login-Anfrage an den Authentication Server (AS).
❷ Der Authentication Server antwortet mit einem Ticket Granting Ticket (TGT).
❸ Der Client möchte eine Verbindung zu einem Kerberos-Dienst aufbauen und verlangt deshalb ein Service Ticket (ST).
❹ Der Ticket Granting Server (TGS) stellt das angeforderte Service Ticket aus.
❺ Der Client reicht das Service Ticket an den Kerberos-Dienst weiter.
❻ Der User ist authentifiziert, die Verbindung aufgebaut und der User muss sich für die Gültigkeitsdauer des ST nicht mehr anmelden.

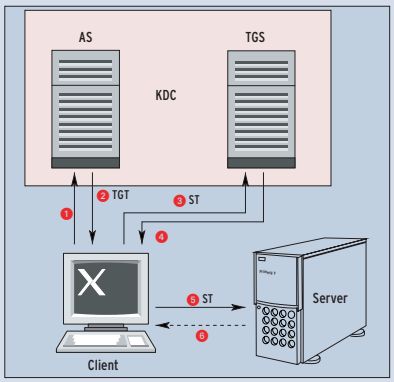


Abbildung 2: In einer Kerberos-Session wandert das Passwort nie im Klartext über das Netzwerk.

»kadmin«. Allerdings müssen hierfür der »kadmin«-Dienst auf dem KDC aktiv und ein gültiger Admin-Principal in der Datei »/var/kerberos/krb5kdc/kadm5.acl« verfügbar sein.
Innerhalb des Verwaltungstools trägt der Administrator mit »add_principal« einen neuen Principal in der Datenbank ein, zum Beispiel:

```
add_principal -pw Passwort thorsten
```

Für einen Service oder eine Workstation funktioniert dies ähnlich:

```
add_principal -pw ftp ➥
  ftp/ftp.notexample.com
add_principal -pw host ➥
  host/station1.notexample.com
```

Die Passwörter der Service Principals müssen auf den entsprechenden Servern bekannt sein. Dazu ist das Passwort für einen Service aus der Kerberos-Datenbank zu extrahieren:

```
ktadd -k /etc/krb5.keytab ➥
  host/station1.notexample.com
```

Die Datei »/etc/krb5.keytab« ist anschließend sicher auf den enstprechenden Service-Rechner zu kopieren, beispielsweise mit »scp«. Mit dem Start des KDC via »service krb5kdc start« (nur Red Hat) steht der Kerberos-Server im Netzwerk zur Verfügung.

Kerberos-Dienste nutzen

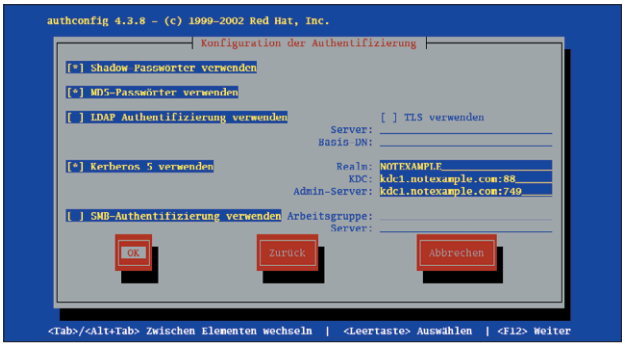
Um auf Kerberos basierende Dienste im Netzwerk zu nutzen, sind diese Services noch auf den entsprechenden Rechnern zu installieren. Kerberos kann mit mehreren Diensten umgehen. Unter »/usr/kerberos/sbin« steht zum Beispiel eine Kerberos-Variante des FTP-Daemon »ftpd« zur Verfügung. Die passende Konfigurationsdatei »gssftp« liegt in dem Verzeichnis »/etc/xinetd.d«. Darin ist unter anderem anzugeben, ob Xinetd den Dienst starten soll.
Kerberos sichert jeden Dienst, der mit einem GSS-API (Generic Security Service, [4]) ausgestattet ist. Dabei sollte der Administrator berücksichtigen, dass er einen TCP/UDP-Port immer nur an einen einzigen Service binden kann. Entscheidet er sich also dazu, die Kerberos-Version des »ftpd« einzusetzen, muss er den bisherigen FTP-Daemon deaktivieren oder am besten ganz aus dem System entfernen. Zudem ist auf den Service-Rechnern ebenfalls die Datei »/etc/krb5.conf« (Listing 3) zu konfigurieren.

Kerberos-Clients einstellen

Die Konfiguration der User-Workstation nimmt der Verwalter wie bei NIS (siehe Kasten „NIS aufsetzen“) mit Hilfe des Red-Hat-Tools Authconfig vor (Abbildung 3). Es sind der Kerberos-Realm so-

Eingesetzte Software

Für den Artikel kam Red Hat 9 zum Einsatz. Alle Beispiele funktionieren aber auch mit anderen Distributionen oder Versionen wie RHEL 3 und Fedora Core 1. Folgende RPM-Pakete [6] sind in jedem Fall zu installieren:
- krb5-workstation-1.3.1-6
- krb5-libs-1.3.1-6
- krb5-server-1.3.1-6
- pam_krb5-2.0.4-1




```
[root@kermit root]# kinit thorsten
Password for thorsten@EXAMPLE.COM:
[root@kermit root]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: thorsten@EXAMPLE.COM

Valid starting     Expires            Service principal
01/15/04 20:53:43  01/16/04 06:53:43  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until 01/15/04 20:53:43


Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
[root@kermit root]#
```

▲ **Abbildung 4:** So sieht das Benutzer-Ticket nach einem »kinit« aus.

◀ **Abbildung 3:** Das Konfigurationsprogramm Authconfig enthält eine Maske, in der sich Realm und Kerberos Domain Controller für den Server eintragen lassen.

wie der KDC und der Admin-Server einzutragen. Beide laufen hier auf dem gleichen Rechner. Das Werkzeug Authconfig trägt die Parameter in die Dateien »/etc/krb5.conf« (Listing 3) und »/etc/pam.d/system-auth« ein.

Aus »system-auth« ruft der Kerberos-Client das PAM-Modul »pam_krb5.so« auf, das die Benutzernamen an den KDC (genauer den Authentication Server) weiterleitet, um ein TGT zu erhalten. Alternativ lässt sich das Anmeldeprogramm »/bin/login« durch die Kerberos-Version austauchen, die im Verzeichnis »/usr/kerberos/sbin« steht.

Sobald sich ein User an einer so konfigurierten Workstation mit Benutzernamen und Passwort auf einer virtuellen Konsole authentifiziert hat, kommt er in den Besitz eines gültigen TGT. Das Ticket kann er mit dem Befehl »klist« oder »klist -5« (also nur Kerberos-Version-5-Tickets) prüfen. Das Programm Klist präsentiert nicht nur die empfangenen Tickets, sondern zeigt zudem weitere Informationen über sie an, etwa den Namen des Service Principal und die Gültigkeitsdauer. Sie beträgt in der Regel zehn Stunden, ist aber in »/etc/krb5.conf« auch anders zu definieren. Das User-Passwort ändert der Benutzer auf dem Kerberos-Server selbst. Dafür steht das Programm »kpasswd« bereit.

Und was ist mit Windows?

Auch Microsoft-Clients können den Linux-KDC für die Authentifizierung benutzen [5]. Ein Windows-Domänen-Controller ist letztlich nichts anderes als eine Kombination aus Kerberos- und LDAP-Server (Lightwight Directory Access Protocol). Dazu trägt der Administrator für die Windows-Maschinen Host Principals in der Kerberos-Datenbank ein. Die Windows-Clients benötigen außerdem den benutzten KDC, den Realm und das Host-Passwort. Dies ist mit Hilfe von Ksetup, das im Windows-2000-Ressource-Kit enthalten ist, leicht zu bewerkstelligen.

Damit das Ganze funktioniert, müssen die Windows-Clients alle Benutzerinformationen von einem Verzeichnisdienst (beispielsweise LDAP) beziehen. NIS ist dazu nicht geeignet, da es Server-seitig auf Unix-Systeme beschränkt ist.

Alles oder nichts

Kerberos erhöht die Sicherheit im Netzwerk beträchtlich, da das Passwort nicht im Klartext durch das Netzwerk wandert. Doch hilft auch Kerberos nicht, falls sich die Benutzer weiterhin gegen Dienste authentifizieren, die nicht zu seinem Bereich gehören. Ist es nicht möglich, alle Services auf Kerberos umstellen, dann ist es besser, ganz auf dessen Dienste zu verzichten. Wenn eine Authentifizierung wirklich sicher sein soll, gelingt dies nur nach dem Motto „Alles oder nichts“. *(jre)* ■

Listing 3: Kerberos-Client-Konfiguration

```
01 #/etc/krb5.conf
02 [logging]
03  default = FILE:/var/log/krb5libs.log
04  kdc = FILE:/var/log/krb5kdc.log
05  admin_server = FILE:/var/log/kadmind.log
06
07 [libdefaults]
08  ticket_lifetime = 24000
09  default_realm = NOTEXAMPLE.COM
10  dns_lookup_realm = false
11  dns_lookup_kdc = false
12
13 [realms]
14  NOTEXAMPLE.COM = {
15   kdc = kdc1.notexample.com:88
16   admin_server = kdc1.notexample.com:749
17   default_domain = notexample.com
18  }
19
20 [domain_realm]
21  .example.com = NOTEXAMPLE.COM
22  example.com = NOTEXAMPLE.COM
23
24 [kdc]
25  profile = /var/kerberos/krb5kdc/kdc.conf
26
27 [appdefaults]
28  pam = {
29   debug = false
30   ticket_lifetime = 36000
31   renew_lifetime = 36000
32   forwardable = true
33   krb4_convert = false
34  }
```

Infos

[1] Kerberos: [http://web.mit.edu/kerberos]
[2] NIS-HOWTO: [http://www.linux-nis.org/nis-howto]
[3] NTP: [http://www.eecis.udel.edu/~mills/ntp/servers.html]
[4] GSS-API RFC: [http://www.faqs.org/rfcs/rfc1964.html]
[5] Kerberos 5 Interoperability: [http://www.microsoft.com/windows2000/techinfo/planning/security/kerbsteps.asp]
[6] Red Hat FTP-Server: [ftp://ftp.redhat.com/redhat/linux/9/en/os/i386/RedHat/RPMS]
[7] Connection Tracking: [http://www.sns.ias.edu/~jns/security/iptables/iptables_conntrack.html]

Der Autor

Thorsten Scherf arbeitet für den Linux-Distributor Red Hat und führt Projekte und Schulungen durch. Sein Schwerpunkt liegt im Bereich Netzwerksicherheit.