




Identity-Management mit FreeIPA 2



FreeIPA, das steht für Identity, Policy und Audit unter einem Dach und für Linux. Mit der Version 1.0 dieser freien Software wurde bereits das Identity- Management für Benutzer eingeführt, mit der nun anstehenden Version 2.0 kommen Maschinen- und Service-Identitäten sowie Systemmanagement-Funktionen hinzu. Später sollen weitere Erweiterungen folgen: Policy- und Audit- Einstellungen wird IPA an zentraler Stelle konfigurieren, verwalten und auf die Client-Sy- steme verteilen können. Die bereits bestehende Microsoft Active-Directory Integration wird mit Hilfe von Samba 4 weiter ausgebaut. Thorsten Scherf

Die Idee des noch recht jungen FreeIPA (1) war, eine Vielzahl bekannter Open Source-Projekte unter einen Hut zu bringen und die verschiedenen Kommandozeilen-Tools und Webfrontends in einem System zu vereinen. Das Resultat sollte besonders leicht bedienbar sein und den Administrator nicht mit Konfigurationsorgien überfordern. Am Ausgangspunkt gab es bereits für jede Teilaufgabe, der sich FreeIPA stellen will, mindestens eine eingeführte Lösung: Ein LDAP-Server eignet sich für die Benutzerverwaltung, zum sicheren Management von Passwörtern existiert seit langer Zeit das Kerberos-Protokoll, der Audit-Daemon kann sämtliche Benutzer- und Prozessaktivitäten nachverfolgen und schließlich lässt sich mit SELinux eine fein granulierte Sicherheitspolicy auf Grundlage der Mandatory-Access-Control einrichten.

Der Nachteil all dieser Einzellösungen besteht allerdings darin, dass sie nicht ohne Weiteres in eine zentrale Verwaltung integrierbar sind. So ist es beispielsweise nicht möglich, die Audit-Logs von mehreren Maschinen zentral zu verwalten. Auch das Verteilen einmal erzeugter SELinux Policy-Module auf mehrere Maschinen ist ohne selbst gebaute Skripte bisher nicht möglich. Zwar existiert eine Vielzahl von proprietären Lösungen, die jedoch meist recht teuer und unflexibel sind. Erschwerend kommt hinzu, dass die Komponenten nicht immer reibungslos zusammenarbeiten.

Eine weitere Alternative besteht im Einsatz von Microsofts Active-Directory Server (ADS) als

erprobter Directory-Lösung auf LDAP/Kerberos-Basis. Für das Andocken von Linux-Umgebungen an das ADS gibt es bereits das ebenfalls erprobte Samba oder beispielsweise Likewise (2). Ein großer Nachteil besteht jedoch darin, dass im ADS keinerlei Policy- oder Audit-Management für Linux-Systeme existiert. Hierfür müsste der Admin dann wieder auf andere Programme zurückgreifen, was den Produkt-Dschungel erneut vergrößerte, und genau dies möchte man ja vermeiden.

Zusätzlich ließe sich natürlich auch noch darüber diskutieren, ob man sicherheitsrelevante Daten wirklich in die Hand eines Anbieters geben möchte, dessen Software für ihre Anwender undurchschaubar und verschlossen bleibt. Wer das verneint oder die sonstigen Nachteile des Active Directory höher als dessen Nutzen bewertet, dem bietet sich FreeIPA als ein noch recht junges, durch Red Hat gesponsertes Projekt an, das die Open-Source-Community intensiv weiterentwickelt.

Kurz vor Weihnachten 2010 überraschten die FreeIPA-Entwickler mit dem ersten Beta Release der künftigen Version 2.0. Trotz einiger fehlender Features – die Audit- und SELinux-Policy Verwaltung ist auf ein späteres Release verschoben – hat sich einiges im Vergleich zur Version 1.0 getan. Die Policy-Komponente wurde so weit umgesetzt, dass IPA nun auch Host-Based-Access-Control-, Sudo- und Passwort-Regeln verwaltet. Der Identify-Bereich wurde weiter ausgebaut, so unterstützt IPA nun

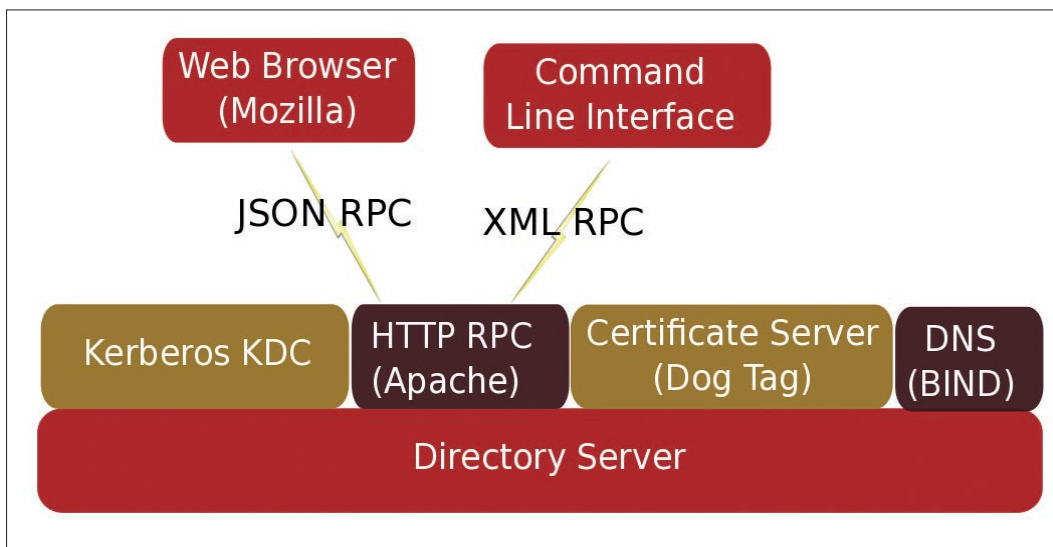


Abbildung 1: Bekannte Open Source-Tools integriert FreeIPA unter einer Haube und erleichtert so wesentlich ihre Bedienung.

```

[root@ipal ~]# ipa help user-add
Purpose: Add a new user.
Usage: ipa [global-options] user-add LOGIN

Options:
-h, --help          show this help message and exit
--first=STR        First name
--last=STR         Last name
--cn=STR           Full name
--displayname=STR Display name
--initials=STR     Initials
--homedir=STR     Home directory
--gecos=STR        GECOS field
--shell=STR        Login shell
--principal=STR   Kerberos principal
--email=STR        Email address
--password         Prompt to set the user password
--uid=INT          User ID Number (system will assign one if not provided)
--gidnumber=INT   Group ID Number
--street=STR       Street address
--city=STR         City
--state=STR        State
--postalcode=INT  ZIP
--phone=STR        Telephone Number
--mobile=STR       Mobile Telephone Number
--pager=STR        Pager Number
--fax=STR          Fax Number
--orgunit=STR      Org. Unit
--title=STR        Job Title
--manager=STR      Manager
--carlicense=STR   Car License
--addattr=STR     Add an attribute/value pair. Format is attr=value. The
                  attribute must be part of the schema.
--setattr=STR     Set an attribute to an name/value pair. Format is
                  attr=value. For multivalued attributes, the command
                  replaces the values already present.
--all             retrieve and print all attributes from the server.
--raw            Affects command output.
                  print entries as stored on the server. Only affects
                  output format.

[root@ipal ~]#

```

Abbildung 2: Alle notwendigen Parameter zum Anlegen eines Benutzers lassen sich als Optionen mit übergeben.

auch Maschinen- und Service-Identitäten. Dank der Integration der offenen Dogtag PKI (3) jetzt sogar mittels X.509-Zertifikaten. Das Zusammenspiel der einzelnen Komponenten ist in **Abbildung 1** dargestellt. Auch die IPA Kommandozeilen-Tools haben sich im Vergleich zur Version 1.0 geändert. Statt einer Vielzahl von unterschiedlicher Tools, existiert für die allermeisten Aufgaben nun einzelnes Tool namens »ipa«. Je nachdem welches Argument

Listing 1: Yum-Repository

```

01 [freeipa-devel]
02 name=FreeIPA Development
03 baseurl=http://freeipa.com/downloads/devel/rpms/
   F$releasever/$basearch
04 enabled=1
05 gpgcheck=0

```

Listing 2: Installation verifizieren

```

01 # kinit admin
02 Password for admin@VIRT.TUXGEEK.DE:
03
04 [root@ipal ~]# klist
05 Ticket cache: FILE:/tmp/krb5cc_0
06 Default principal: admin@VIRT.TUXGEEK.DE
07
08 Valid starting Expires Service principal
09 01/06/11 23:18:01 01/07/11 23:17:59
10 krbtgt/VIRT.TUXGEEK.DE@VIRT.TUXGEEK.DE

```

das Tool erhält, führt es entsprechende Aktionen aus. Beispielsweise verwendet der Admin zum Anlegen eines neuen Benutzers nun nicht mehr das Tool »ipa-adduser«, sondern ruft stattdessen »ipa user-add« auf. Welche Aktionen möglich sind zeigt die Ausgabe von »ipa help«.

Server Installation

Bevor es an die Installation des eigentlichen FreeIPA-Servers geht, ist sicherzustellen, dass sich die Namen aller verwendeten Maschinen via DNS auflösen lassen. Erweitert man den vorhanden DNS-Server um einige Serviceeinträge (SRV-Records), erleichtert das später auch die Konfiguration der Client-Maschinen, weil sie dann über eine DNS-Abfrage Informationen über ihre zuständigen Server und die Kerberos-Realm erhalten. Eine manuelle Konfiguration ist so kaum mehr nötig. Bei der Installation des FreeIPA-Servers ist optional die Installation und Konfiguration eines DNS-Servers möglich. Da die DNS-Server-Daten, wie sämtliche anderen IPA-Daten auch, im LDAP liegen, ist zwingend vor dem IPA-Server das Paket »bind-dyndb-ldap« zu installieren.

In den Standard Fedora-Repositories befindet sich aktuell lediglich die stabile FreeIPA-Version 1.2.2, in diesem Artikel geht es jedoch ausschließlich um die Beta-Version 2.0. Die Pakete hierfür sind über ein eigenes Repository zu installieren. Dazu erzeugt der Installateur im Ordner »/etc/yum.repos.d/« eine Datei »freeipa-devel.repo« wie in **Listing 1**. Statt aus dem Repository kann sich der Admin die Datei auch einfach mittels »wget« herunterladen:

```
# wget http://freeipa.org/downloads/freeipa-devel.repo -P /etc/yum.repos.d/
```

Anschließend installiert er den FreeIPA- und DNS-Server auf seinem 32- oder 64-Bit Fedora-13 oder Fedora-14 mittels:

```
# ipa-server-install --setup-dns
```

Durch den Aufruf der Setup-Routine werden nun die folgenden Komponenten auf der Maschine installiert:

- NTP
- Fedora Directory Server
- Fedora Dogtag PKI
- MIT Kerberos
- Apache Webserver
- SELinux Policy für die diversen FreeIPA Komponenten

Das Installationsprogramm fragt jetzt die notwendigen Informationen wie LDAP Base DN, Kerberos Realm, Servername und so weiter ab und schon nach einigen Minuten ist der Server inklusive aller Komponenten einsatzbereit. Soll IPA innerhalb einer virtuellen Maschine laufen, so bietet es sich an, die automatische Installation des NTP-Servers mittels der Option »--no-ntp« zu deaktivieren. Auch für die anderen Komponenten existieren eine Vielzahl von Optionen die man dem Setup-Tool übergeben kann. Das Kommando »man ipa-server-install« erzeugt eine Übersicht aller verfügbaren Optionen. Soll die Einrichtung des FreeIPA Servers komplett automatisiert ablaufen, beispielsweise als Teil einer Kickstart-Installation, so ist dies mit Hilfe einiger weiterer Optionen möglich:

```
# ipa-server-install --setup-dns ?
-U -u ldap -r VIRT.TUXGEEK.DE -p ds-?
password -a ipa-admin-password
```

Die Option »-u« bestimmt hier den Benutzer unter dem der FreeIPA-Server laufen soll, »-r« den Kerberos Realm, »-p« das Passwort für den Directory Manager und »-a« das Passwort für den FreeIPA-Admin Benutzer, den IPA per default erzeugt.

Hat die Installation soweit geklappt, lässt sich die korrekte Funktionsweise recht leicht mittels

»kinit admin« verifizieren. Hiermit fordert man ein Benutzer-Ticket vom Kerberos-Server an. Hat dies funktioniert, so zeigt »klist« das empfangene Kerberos TGT an (**Listing 2**).

Im nächsten Schritt sollte der Admin die ersten Benutzerkonten anlegen. Wie bereits erwähnt

Listing 3: Benutzer anlegen

```
01 # ipa user-add --password tscherf
02 First name: Thorsten
03 Last name: Scherf
04 Password:
05 Enter Password again to verify:
06 -----
07 Added user "tscherf"
08 -----
09 User login: tscherf
10 First name: Thorsten
11 Last name: Scherf
12 Full name: Thorsten Scherf
13 Display name: Thorsten Scherf
14 Initials: TS
15 Home directory: /home/tscherf
16 GECOS field: tscherf
17 Login shell: /bin/sh
18 Kerberos principal: tscherf@VIRT.TUXGEEK.DE
19 UID: 1215000004
```

Listing 4: »ldapsearch«

```
01 # ldapsearch -Y GSSAPI -b
    dc=virt,dc=tuxgeek,dc=de -LLL uid=tscherf
02 SASL/GSSAPI authentication started
03 SASL username: admin@VIRT.TUXGEEK.DE
04 SASL SSF: 56
05 SASL data security layer installed.
06 dn: uid=tscherf,cn=users,cn=accounts,dc=virt,dc=
    tuxgeek,dc=de
07 displayName: Thorsten Scherf
08 cn: Thorsten Scherf
09 objectClass: top
10 objectClass: person
11 objectClass: organizationalperson
12 objectClass: inetorgperson
13 objectClass: inetuser
14 objectClass: posixaccount
15 objectClass: krbprincipalaux
16 objectClass: krbticketpolicyaux
17 objectClass: ipaobject
18 objectClass: mepOriginEntry
19 loginShell: /bin/sh
20 sn: Scherf
21 gecos: tscherf
22 homeDirectory: /home/tscherf
23 krbPwdPolicyReference:
24 cn=global_policy,cn=VIRT.TUXGEEK.DE,cn=kerberos
    ,dc=virt,dc=tuxgeek,dc=de
25 krbPrincipalName: tscherf@VIRT.TUXGEEK.DE
26 givenName: Thorsten
27 uid: tscherf
28 initials: TS
29 uidNumber: 1215000004
30 gidNumber: 1215000004
31 ipaUniqueID: 4a17c99a-1f02-11e0-ab26-525400cbd
    f57
32 krbLastPwdChange: 20110113104551Z
33 krbPasswordExpiration: 20110113104551Z
34 mepManagedEntry: cn=tscherf,cn=groups,cn=accoun
    ts,dc=virt,dc=tuxgeek,dc=de
35 memberOf: cn=ipausers,cn=groups,cn=accounts,dc=
    virt,dc=tuxgeek,dc=de
```

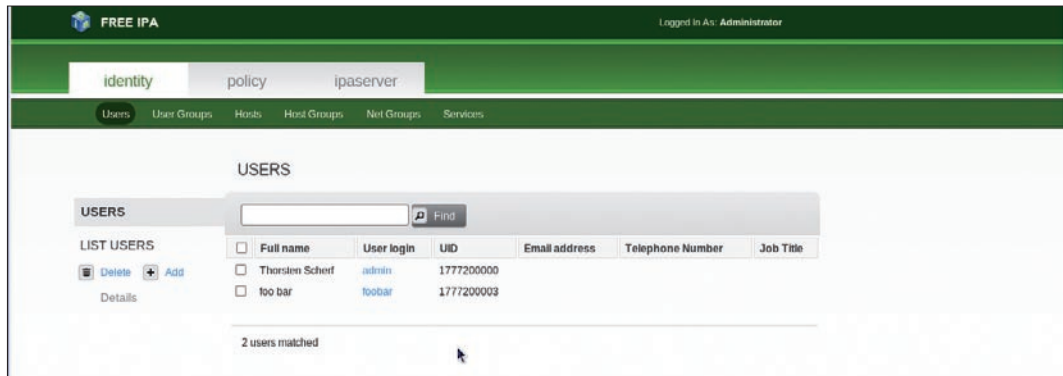


Abbildung 3: Das neue Webinterface verzichtet, im Gegensatz zum Vorgänger, vollständig auf Turbo-

funktioniert dies nun mittels »ipa user-add«. Das Tool fragt die notwendigen Attribute interaktiv ab (Listing 3).

Alternativ lassen sich sämtliche Informationen für den Benutzer-Account natürlich auch als Option übergeben (Abbildung 2). Wer sämtliche Attribute für den Benutzer sehen möchte kann mittels »ldapsearch« das Benutzer-Objekt direkt im LDAP-Baum abfragen (Listing 4).

Das für den Zugriff auf den LDAP-Server bereits das Kerberos-Protokoll verwendet wurde, zeigt der Aufruf von »klist« neben dem initial ausge-

Anweisungen sind individuell anzupassen:

Nachdem eine HTTPS-Verbindung zum FreeIPA-Server aufgebaut ist, lassen sich sehr komfortabel Benutzer über das Webinterface einrichten oder abfragen. **Abbildung 3 und 4** zeigen, dass das komplett überarbeitete Web-GUI im Vergleich zur Vorgängerversion sehr viel mehr Konfigurationsmöglichkeiten anbietet.

FreeIPA Server Upgrade

FreeIPA 2.0 unterstützt kein Upgrade von einer älteren Version. Allerdings existiert mit »ipa migrate-ds« ein Tool, mit dem sich die Daten einer älteren FreeIPA-Installation oder eines anderen Directory-Servers in den aktuellen Directory-Server von FreeIPA 2.0 laden lassen. Hierfür ist dann noch der Aufruf von »ipa config-mod --enable-migration=TRUE« notwendig. Dabei sind die jeweiligen Container, in denen sich die Benutzer- und Gruppen-Objekte befinden, entsprechend anzugeben. Eine Migration ausschließlich von User-Daten ist derzeit nicht möglich:

Noch Fragen?
Unsere Autoren beantworten Ihre Fragen gern in unseren Foren.

stelltem Kerberos TGT nun auch das Service-Ticket für den Zugriff auf den LDAP-Server an:

```
[root@ipa1 ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin@VIRT.TUXGEEK.DE

Valid starting Expires Service principal
01/13/11 11:33:20 01/14/11 11:33:18 krbtgt@VIRT.TUXGEEK.DE@VIRT.TUXGEEK.DE
01/13/11 11:33:22 01/14/11 11:33:18 HTTP@ipa1.virt.tuxgeek.de@VIRT.TUXGEEK.DE
01/13/11 11:36:12 01/14/11 11:33:18 ldap/ipa1.virt.tuxgeek.de@VIRT.TUXGEEK.DE
```

```
# ipa migrate-ds
--user-container=cn=users,
cn=accounts --group-container=cn=groups,
cn=accounts ldap://virt.foo.de:389
```

Viele administrative Aufgaben lassen sich natürlich auch bequem über das FreeIPA-Webinterface erledigen. Hierfür ist jedoch zuerst der Webbrowser für den entsprechenden Kerberos-Realm zu konfigurieren. Beim Firefox lassen sich die aktuellen Konfigurationseinstellungen über »about:config« anzeigen. Die folgenden

Kerberos-Einstellungen für den Firefox

- 01 network.negotiate-auth.trusted-uris .virt.tuxgeek.de
- 02 network.negotiate-auth.delegation-uris .virt.tuxgeek.de
- 03 network.negotiate-auth.using-native-gsslib true

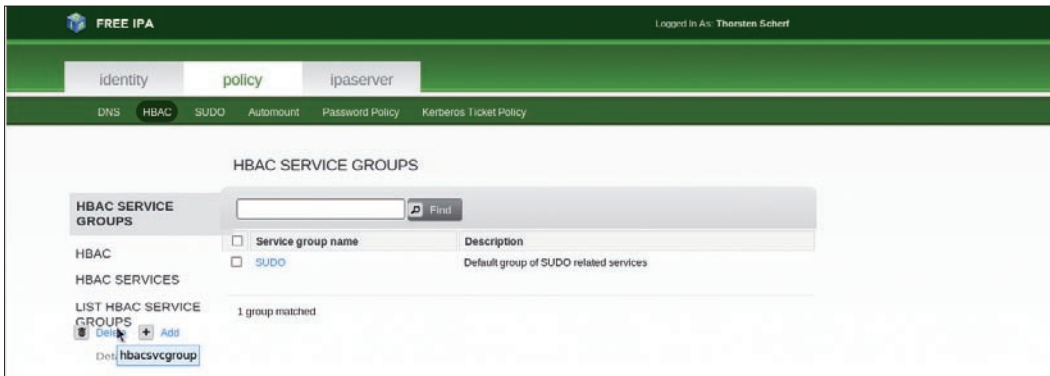


Abbildung 4: Neben Einstellungen zur Benutzerverwaltung finden sich auch Möglichkeiten zur Konfiguration von Policy-Regelsätzen.

Das Tool versucht nun sich mit dem Directory Manager Account auf dem angegebenen Server anzumelden, die Daten entsprechend umzuwandeln um diese schließlich auf dem lokalen Server zu importieren. Optional erlaubt es die Option »-bind-dn«, einen alternativen Account

für den Zugriff auf den Directory-Server anzugeben. Bei der Daten-Migration von einem regulären Directory-Server sind natürlich keine Kerberos-Keys für die entsprechenden Benutzer-Accounts vorhanden, sodass die Client-Systeme nach

System Security Service Daemon (SSSD)

Der System Security Service Daemon (SSSD) stellt verschiedene Funktionen zur Verfügung. Hiervon sind drei besonders interessant. Zum einen löst das Tool das Problem der Offline-Authentifizierung eines Benutzers. Der SSSD hält empfangene Credentials eines zentralen Servers einfach in einer lokalen Cache vor. Meldet sich ein Benutzer also im Firmennetzwerk mit einem Firmen-Konto an seinem Notebook an, so gelangen die Credentials automatisch in den SSSD-Cache. Wie lange sie dort liegen bleiben und gültig sind, lässt sich natürlich in einer zentralen Konfigurationsdatei festlegen. Desweiteren unterstützt der SSSD die Abfrage mehrerer LDAP- oder auch NIS-Server. Hiermit lassen sich dann eine Vielzahl von unterschiedlichen Benutzerdatenbanken

abfragen. Auch aus Performance-Sicht bietet der Einsatz des neuen Daemons einen Vorteil. Anstatt für jede Abfrage an den LDAP-Server eine eigene Verbindung aufzubauen, ist lediglich ein Socket vom SSSD zum LDAP-Server notwendig. Der Daemon bietet dabei eine eigene NSS- und PAM-Schnittstelle an für anfragende Client-Systeme. Im Backend sorgen sogenannte Identity-Provider für einen Zugriff auf den entsprechenden Identity und Authentication-Server (Abbildung 5). Meldet sich nun also ein Benutzer mit einem migrierten Konto am FreeIPA-Server an, so wird dessen Identität über den integrierten LDAP-Server verifiziert. Die eigentliche Authentifizierung findet anschließend über den ebenfalls integrierten Kerberos-Server statt. Nun existiert für diesen Benutzer natürlich noch kein Eintrag in der Kerberos-Datenbank, da das Konto zuvor ja lediglich auf einem LDAP-Server gepflegt wurde somit auch das Passwort Teil des Benutzer-Objekts war. In diesem Fall schlägt die Authentifizierung mittels Kerberos also fehl. Ist der SSSD jedoch, wie oben beschrieben, für die Migration von Benutzer-Konten konfiguriert, so führt der integrierte Security-Provider automatisch einen »simple bind« gegen den integrierten LDAP-Server durch. Hierzu sendet der SSSD das Benutzer-Passwort im Klartext an den LDAP-Server wo es entsprechend gehasht und mit dem hinterlegten Passwort-Hash des Benutzers verglichen wird. Bei einer erfolgreichen Authentifizierung generiert der SSSD automatisch einen Eintrag in der Kerberos-Datenbank des FreeIPA-Servers.

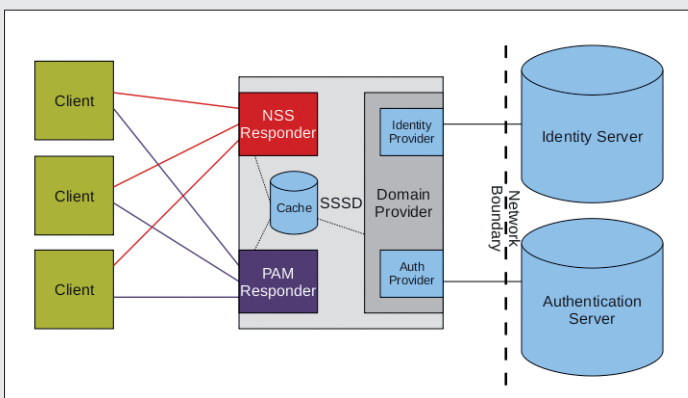


Abbildung 5: Beim System Security Service Daemon hat der Client nur eine Schnittstelle für mehrere Backends.

einer Migration auf den FreeIPA-Server noch nicht für eine Kerberos-basierte Authentifizierung konfiguriert werden dürfen. Stattdessen ist hier eine reine LDAP- oder SSSD-Authentifizierung (siehe Kasten SSSD) mittels »simple bind« und »startTLS« einzurichten – dies ist notwendig, da die sensiblen Benutzer-Daten bei einem Simple Bind im Klartext über das Netzwerk fließen. Nach der Anmeldung eines Benutzers, erzeugt IPA die fehlenden Kerberos-Keys automatisch. Reine LDAP-basierte Clients lassen sich dann auf eine SSSD-basierte Authentifizierung umstellen.

Listing 5: Replika-Konfiguration

```
01 # ipa-replica-prepare ipareplica.virt.tuxgeek.de
02 Directory Manager (existing master) password:
03
04 Preparing replica for ipareplica.virt.tuxgeek.de from ipal.
   virt.tuxgeek.de
05 Creating SSL certificate for the Directory Server
06 Creating SSL certificate for the Web Server
07 Exporting RA certificate
08 Copying additional files
09 Finalizing configuration
10 Packaging replica information into
11 /var/lib/ipa/replica-info-ipareplica.virt.tuxgeek.de.gpg
```

Listing 6: »ipa-client-install«

```
01 # ipa-client-install
02 Discovery was successful!
03 Realm: VIRT.TUXGEEK.DE
04 DNS Domain: virt.tuxgeek.de
05 IPA Server: ipal.virt.tuxgeek.de
06 BaseDN: dc=virt,dc=tuxgeek,dc=de
07
08 Continue to configure the system with these values? [no]:
   yes
09 Enrollment principal: admin
10 Password for admin@VIRT.TUXGEEK.DE:
11 Enrolled in IPA realm VIRT.TUXGEEK.DE
12 Created /etc/ipa/default.conf
13 Configured /etc/sss/sss.conf
14 Configured /etc/krb5.conf for IPA realm VIRT.TUXGEEK.DE
15 SSSD enabled
16 Kerberos 5 enabled
17 NTP enabled
18 Client configuration complete.
```

FreeIPA Replica

Nachdem nun eine grundlegende Konfiguration des Servers fertig gestellt ist, sollte man die Daten des Directory-Servers auf eine zweite Maschine replizieren. Da FreeIPA auch die komplette Kerberos-Konfiguration und -Datenbank im LDAP vorhält, hat man somit im Handumdrehen einen zweiten Master-Server aufgesetzt. Der ist nun ebenfalls in der Lage, Änderungen von Clients entgegenzunehmen und diese auf den jeweils anderen Server zu replizieren. Nach Ausfall eines Masters wären die Daten so trotzdem noch verfügbar und ließen sich dort sogar ändern. Ist der ausgefallene Server wieder online, werden die geänderten Daten auf ihn zurückrepliziert. Auch zur Lastverteilung bietet sich der Einsatz mindestens zweier Server an. Speichert man die Daten an mehreren, geografisch voneinander getrennten Standorten, so empfiehlt es sich, unter Umständen noch weitere Server zu konfigurieren und als Replicas einzurichten, damit nicht für jede Abfrage oder Änderung am Directory eine WAN-Verbindung nötig ist.

Die Konfiguration geht auch hier sehr schnell von der Hand. Auf dem ersten Master ist eine Konfigurationsdatei mit allen notwendigen Informationen für den zusätzlichen Server zu erzeugen (**Listing 5**)-

Nun kopiert man einfach die so erzeugte Datei auf den Replica-Host und startet dort die Installation mittels »ipa-replica-install«. Es ist darauf zu achten, dass zuvor alle FreeIPA Pakete installiert wurden:

```
# scp /var/lib/ipa/replica-info-?
ipareplica.virt.tuxgeek.de.gpg ?
root@ipareplica:/var/lib/ipa/
# ipa-replica-install /var/lib/ipa?
/replica-info-ipareplica.virt.?
tuxgeek.de.gpg
```

Ist das Installationsprogramm ohne Fehler durchgelaufen, startet im Anschluss eine Replikation der LDAP-Datenbank. Fügt man diesen Replica nun noch seiner DNS-Zonendatei hinzu, stehen zwei unterschiedliche Server bereit. Die so eingerichteten Replizierungsvereinbarungen (replication agreements) lassen sich mittels »ipa-replica-manage« anzeigen oder auch nachträglich ändern.

Client-Konfiguration

Ein FreeIPA-Client existiert nicht nur für Fedora

und Red Hat Enterprise Linux (RHEL), sondern daneben gibt es Clients für eine Vielzahl verschiedener Unix-Varianten wie beispielsweise Solaris, AIX, HP-UX oder auch Mac OS X. Auf einem Fedora-System gelingt die Installation wieder ganz einfach, nämlich nur mit einem schlichten Yum-Aufruf:

```
# yum install ipa-client
```

Administratoren installieren auf Ihrer Arbeitsstation zusätzlich noch das Paket »ipa-admin-tools«. Da der FreeIPA-Server auch als DNS-Server konfiguriert wurde, ist dieser nun auch auf den Client-Systemen in der Datei »/etc/resolv.conf« einzutragen. Die eigentliche Konfiguration des Clients erfolgt dann über den Aufruf von »ipa-client-install« (Listing 6). Dank der

?

Noch Fragen?

Unsere Autoren beantworten Ihre Fragen gern in unseren Foren.

Service-Records im DNS, findet der Client alle Server durch ein Service-Discovery.

Für den Host wird hierbei direkt ein Principal in der Kerberos-Datenbank auf dem FreeIPA-Server erzeugt. Dies ist wichtig, damit der Admin später einen IPA-Service, wie beispielsweise einen NFS-, SSH- oder Web-Server mit einem Kerberos-Principal oder einem X.509-Zertifikat einrichten kann. Alternativ lässt sich der Client auch leicht mit Hilfe des Tools »system-config-authentication« einrichten (Abbildung 6).

Sollte der Client noch über keinen DNS-Eintrag auf FreeIPA-Server verfügen, so lässt sich dieser Eintrag mit Hilfe des FreeIPA Kommandozeilen-Tools sehr leicht nachholen (listing 7).

Bei einer Directory-Server Migration sollten man anschliessend noch für eine sichere Kommunikation mit dem LDAP-Server sorgen. Hierfür ist, wie erwähnt, die Option startTLS in der Datei /etc/ldap.conf zu aktivieren:

```
# cat /etc/ldap.conf
uri ldap://ipal.virt.tuxgeek.de
base dc=virt,dc=tuxgeek,dc=de
ssl start_tls
tls_checkpeer yes
tls_cacertdir /etc/cacerts/
```

Anschliessend lädt der Admin das FreeIPA RootCA-Zertifikat auf die Client und speichern es im passenden Format:

```
# wget http://ipal.virt.tuxgeek.de
/ipa/config/ca.crt -P /etc/cacerts/
# cacertdir_rehash /etc/cacerts/
```

Damit ist die Konfiguration des Clients abgeschlossen und Zugriffe auf den LDAP-Server mittels »pam_ldap« finden dank der angepassten Konfiguration mittels startTLS nun auch verschlüsselt statt.

Da das Client-System zu diesem Zeitpunkt bereits über einen Host-Principal in der Kerberos-Datenbank verfügt, ist ein Login über die GSSAPI-Schnittstelle des SSH-Servers bereits möglich:

```
# kinit tscherf
# ssh tscherf@ipa2
Last login: Sun Jan 16 13:04:35 2011 from 192.168.122.1
# id
uid=1640400003(tscherf) gid=1640400003(tscherf)
groups=1640400003(tscherf),1640400001(ipausers)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

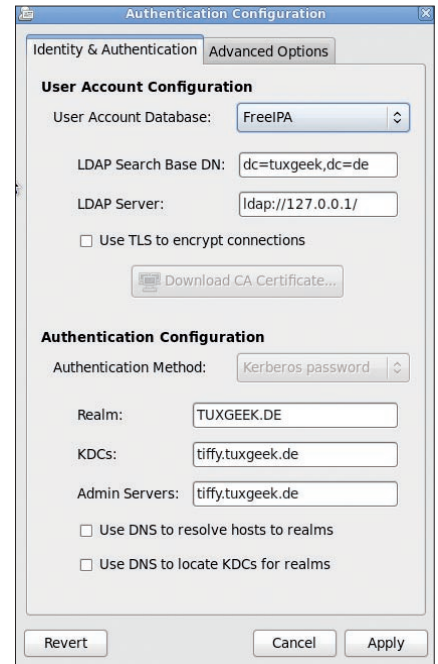


Abbildung 6: Auch das bekannte system-config-authentication Tool bietet eine Konfigurationseinstellung für den FreeIPA-Server.

Listing 7: DNS-Eintrag

```
01 # ipa dns-add-rr virt.tuxgeek.de ipa2 A 192.168.122.74
02 -----
03 dns-add-rr:
04 -----
05 dn: idnsname=ipa2,idnsname=virt.tuxgeek.de,cn=dns,dc=virt,dc=tuxgeek,dc=de
06 arecord: 192.168.122.74
07 objectclass: top
08 objectclass: idnsrecord
09 -----
10 Added DNS resource record "ipa2 A 192.168.122.74" to zone "virt.tuxgeek.de".
11 -----
```

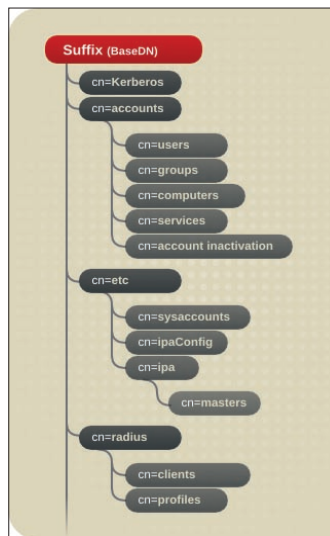



Abbildung 7: FreeIPA speichert die Kerberos-Datenbank in einem LDAP-Container.

Ein Blick in die Logdatei des SSH-Servers bestätigt, dass die Anmeldung tatsächlich mittels Kerberos stattgefunden hat:

```
# tail /var/log/secure Jan 16 13:04:34 ipa2 sshd[2571]: Authorized to tscherf, krb5 principal tscherf@VIRT.TUXGEEK.DE (krb5 kuserok)
Jan 16 13:04:34 ipa2 sshd[2571]: Accepted gssapi-with-mic for tscherf from 192.168.122.1 port 36462 ssh2
Jan 16 13:04:35 ipa2 sshd[2571]: pam_unix(sshd:session): session opened for user tscherf by (uid=0)
```

Auf dem Source-System sollte die Ausgabe von »klist« nun neben dem Ker-

beros TGT und auch das Host-Principal des Client anzeigen.

Kerberos-Services

Nun geht es an die Konfiguration des ersten eigenen Kerberos-Service. Als Beispiel dient ein NFS-Server, der sich von den Client-Maschinen über das sichere NFSv4-Protokoll mit Kerberos-Authentifizierung ansprechen lässt. Zusätzlich soll der Server auch für Datenintegrität und Vertraulichkeit sorgen. Hierfür richten wir auf dem IPA-Server eine NFS-Freigabe ein, die nicht nur eine Benutzerauthentifizierung auf Basis von NFSv4 durchführt, sondern auch für Datenauthentizität und Datenintegrität sorgt:

```
# cat /etc/exports
/data gss/krb5(rw,fsid=0,subtree_check)
/data gss/krb5i(rw,fsid=0,subtree_check)
/data gss/krb5p(rw,fsid=0,subtree_check)
```

In der Kerberos-Datenbank ist nun noch ein Service-Principal für den NFS-Dienst zu erzeugen und die Keytab-Datei des Servers zu exportieren (Listing 8).

Ein »echo SECURE_NFS=yes > /etc/sysconfig/nfs« sorgt dafür, dass nach einem »service nfs start« alle notwendigen NFS-Dienste verfügbar sind.

Die Client-Konfiguration verläuft ähnlich. Auch hier ist ein NFS Service-Principal zu erzeugen und lokal in der »/etc/krb5.keytab«-Datei zu speichern (Listing 9).

Ob dies geklappt hat, lässt sich leicht über den Aufruf von »ipa service-find« herausfinden. Es sollte auf Wunsch alle vorhandenen Host- und Service-Principals aus der Keytab-Datei auflisten:

```
# ipa Service-find NFS/`hostname`
-----
1 service matched
-----
Principal: nfs/ipa2.virt.tuxgeek.de@VIRT.TUXGEEK.DE
Keytab: True
Managed by: ipa2.virt.tuxgeek.de
-----
Number of entries returned 1
-----
```

Damit die notwendigen NFS-Client Dienste »rpcgssd« und »rpcidmapd« korrekt starten, ist auch auf dem Client ein Eintrag »SECURE_NFS=yes« in der Datei »etc/sysconfig/nfs« vor-

Listing 8: Service erzeugen

```
01 # ipa service-add nfs/ipa1.virt.tuxgeek.de
02 -----
03 Added service "nfs/ipa1.virt.tuxgeek.de@VIRT.TUXGEEK.DE"
04 -----
05 Principal: nfs/ipa1.virt.tuxgeek.de@VIRT.TUXGEEK.DE
06 Managed by: ipa1.virt.tuxgeek.de
07
08 # ipa-getkeytab -s ipa1 -p nfs/ipa1.virt.tuxgeek.de -k /etc/krb5.keytab
09 Keytab successfully retrieved and stored in: /etc/krb5.keytab
```

Listing 9: NFS-Client installieren

```
01 # kinit admin
02 Password for admin@VIRT.TUXGEEK.DE:
03 # ipa service-add nfs/`hostname`
04 -----
05 Added service "nfs/ipa2.virt.tuxgeek.de@VIRT.TUXGEEK.DE"
06 -----
07 Principal: nfs/ipa2.virt.tuxgeek.de@VIRT.TUXGEEK.DE
08 Managed by: ipa2.virt.tuxgeek.de
09
10 # ipa-getkeytab -s ipa1.virt.tuxgeek.de -p nfs/ipa2.virt.tuxgeek.de -k /etc/krb5.keytab
11 Keytab successfully retrieved and stored in: /etc/krb5.keytab
```

zunehmen. Nun sollte einem erfolgreichen NFSv4-Mount mit der höchstmöglichen Sicherheitsstufe nichts mehr im Wege stehen (**Listing 10**).

Ein Blick in die Logdatei des Kerberos-Servers bestätigt die Übertragung eines NFS-Service-Tickets an den Client (**Listing 11**).

Angemerkt sei noch, dass FreeIPA die komplette Kerberos-Konfiguration im LDAP speichert (**Abbildung 7**). Da die Kerberos-Tools wie »kadmin« oder »kadmin.local« keine native LDAP-Schnittstelle bieten, darf man sie zum Verwalten der Kerberos-Datenbank keinesfalls einsetzen. Für sämtliche Administrationsarbeiten sollte der Admin stattdessen immer die FreeIPA-Tools benutzen. Weitere Kerberos-Services lassen sich auf ähnliche Weise einrichten.

X.509-Zertifikate

Komplett neu ist in der Version 2.0 des FreeIPA-Servers, die Verwaltung von X.509-Zertifikaten. Diese lassen sich einem beliebigen Principal zuordnen, solange ein entsprechender Eintrag für den Service existiert. FreeIPA benötigt zum Ausstellen des Zertifikats eine Zertifikatsanfrage im PEM-Format. Ob diese mit Hilfe von OpenSSL oder auf Basis der Network Security Service (NSS) ausgestellt wird, hängt alleine vom Anwender und seinen Applikation ab, die später das Zertifikat verwenden soll. Zuerst ein Beispiel für eine Zertifikatsanfrage mit Hilfe von OpenSSL (**Listing 12**).

Sollen Ihre Zertifikate stattdessen in einer NSS-Datenbank gespeichert werden, wie sie beispielsweise auch der Apache Webserver oder der Fedora 389-DS verwenden, so erstellt man die Zertifikatsanfrage mittels dem Tool »certutil« und der Angabe, wo sich die NSS-Datenbank befindet:

```
# certutil -R -s "CN=ipa2.virt.2
tuxgeek.de" -d /etc/httpd/alias/ -a2
> webserver.csr
```

Will man eine neue Datenbank erzeugen, so gelingt dies einfach mittels:

```
# certutil -N -d <Pfad zur 2
neuen Datenbank>
```

Anschließend senden der Admin die so erzeugte Zertifikatsanfrage an den FreeIPA-Server der daraus dann ein entsprechendes X.509-Zertifikat erzeugt, das dann in der gewünschten Applikation zum Einsatz kommen kann.

Zugangskontrolle

Mit der aktuellen FreeIPA-Version haben die Entwickler auch erstmals die Unterstützung für Netgroups (**4**) hinzugefügt. Hierbei handelt es sich um einen Mechanismus, mit dem sich recht leicht festlegen lässt, welche Benutzer von welchen Systemen aus, Zugang zu welcher Maschine haben. Die eigentlichen Zugangsregeln hierfür liegen dabei in der Datei »/etc/security/access.conf«. Dies ist die Konfigurationsdatei des zuständigen PAM-Modules »pam_access.so«. Die einzelnen Benutzer/System-Objekte,

Listing 10: Sicherer NFSv4-Mount

```
01 # echo "SECURE_NFS=yes" > /etc/sysconfig/nfs"
02 # service rpcgssd start
03
04 # mount -v -t nfs4 -o sec=krb5p ipa1.virt.tuxgeek.de:/ /
mnt/
05 mount.nfs4: timeout set for Sat Jan 15 12:48:08 2011
06 mount.nfs4: trying text-based options
07 'sec=krb5p,addr=192.168.122.130,clientaddr=192.168.122.74'
08 ipa1.virt.tuxgeek.de:/ on /mnt type nfs4 (rw,sec=krb5p)
09 [root@ipa2 ~]# df -Th /mnt/
10 Filesystem Type Size Used Avail Use% Mounted on
11 ipa1.virt.tuxgeek.de:/ nfs4 3.5G 1.5G 1.8G 46% /mnt
```

Listing 11: NFS-Log

```
01 # tail -f /var/log/krb5kdc.log
02 Jan 15 12:44:47 ipa1.virt.tuxgeek.de krb5kdc[3974](info):
AS_REQ (4 etypes {18
03 17 16 23}) 192.168.122.1: NEEDED_PREAUTH:
04 nfs/ipa2.virt.tuxgeek.de@VIRT.TUXGEEK.DE for
05 krbtgt/VIRT.TUXGEEK.DE@VIRT.TUXGEEK.DE, Additional
pre-authentication required
06
07 Jan 15 12:44:47 ipa1.virt.tuxgeek.de krb5kdc[3974](info):
AS_REQ (4 etypes {18
08 17 16 23}) 192.168.122.1: ISSUE: authtime 1295091887,
etypes {rep=18 tkt=18
09 ses=18}, nfs/ipa2.virt.tuxgeek.de@VIRT.TUXGEEK.DE for
10 krbtgt/VIRT.TUXGEEK.DE@VIRT.TUXGEEK.DE
11
12 Jan 15 12:44:47 ipa1.virt.tuxgeek.de krb5kdc[3974](info):
TGS_REQ (4 etypes {18
13 17 16 23}) 192.168.122.1: ISSUE: authtime 1295091887,
etypes {rep=18 tkt=18
14 ses=18}, nfs/ipa2.virt.tuxgeek.de@VIRT.TUXGEEK.DE for
15 nfs/ipa1.virt.tuxgeek.de@VIRT.TUXGEEK.DE
```

die die genannte Konfigurationsdatei verwenden kann, speichert FreeIPA in seinem LDAP-Baum (Listing 13)

Neben den Netgroups bietet FreeIPA nun auch einen eigenen Host-based-Access-Control (HBAC) Mechanismus an. Eine Konfiguration findet, wie bei FreeIPA gewohnt, entweder über das Webfrontend oder das Kommandozeilentool »ipa« statt. Die Idee hierbei ist, dass man für die gewünschten Benutzer oder Gruppen festlegt, auf welche Dienste diese auf welchen Hosts zugreifen dürfen. Eine abschliessende Catch-All-Regel verbietet dann den Zugang zu allen anderen Diensten. Ein sehr einfaches Beispiel, das den Zugang zum SSH-Dienst auf sämtlichen Systemen für den Benutzer »admin« erlaubt, jeden weiteren Zugang aber unterbindet, zeigt Listing 14.

Listing 12: Zertifikatsanfrage

```
01 # openssl req -out webserver.csr -new -newkey rsa:2048
    -nodes -keyout
02 webserver.key
03 Generating a 2048 bit RSA private key
04 .....+
    ++
05 .....+++
06 writing new private key to 'webserver.key'
07 -----
08 You are about to be asked to enter information that will be
    incorporated
09 into your certificate request.
10 What you are about to enter is what is called a
    Distinguished Name or a DN.
11 There are quite a few fields but you can leave some blank
12 For some fields there will be a default value,
13 If you enter '.', the field will be left blank.
14 -----
15 Country Name (2 letter code) [XX]:DE
16 State or Province Name (full name) []:NRW
17 Locality Name (eg, city) [Default City]:Essen
18 Organization Name (eg, company) [Default Company
    Ltd]:RedHat
19 Organizational Unit Name (eg, section) []:WORLD
20 Common Name (eg, your name or your server's hostname)
    []:ipa2.virt.tuxgeek.de
21 Email Address []:admin@tuxgeek.de
22 Please enter the following 'extra' attributes
23 to be sent with your certificate request
24 A challenge password []:
25 An optional company name []:
```

Zu erwähnen ist noch, dass FreeIPA per default eine Regel »allow_all« implementiert hat. Hiermit ist also der Zugang von allen Benutzern zu sämtlichen Diensten möglichen. Das Löschen dieser Regel gelingt mittels »ipa hbac-del allow_all«.

Active-Directory-Sync

Mir Hilfe von »ipa-replica-manage« ist es in der FreeIPA Version 2.0 nun auch möglich, Daten zwischen einem Windows Active-Directory-Domänencontroller und dem integrierten 389-Directory-Server zu synchronisieren (5). Der Abgleich der Daten erfolgt dabei standardmässig zwischen dem AD-Container CN=Users,\$SUFFIX und dem 389-DS Container cn=users,cn=accounts,\$SUFFIX, wobei \$SUFFIX dem Root-DN entspricht. Soll IPA einen anderen LDAP-Zweig synchronisieren, so ist dies mit der »ipa-replica-manage« Option »--win-subtree« möglich.

Zur Synchronisation benötigt man auf dem Windows-Rechner ein SSL-Zertifikat, das für eine Replizierung der Daten vom AD-Server nach FreeIPA zwingend notwendig ist. Eine Anleitung hierfür befindet sich im Fedora Wiki [x]. Das verwendete CA-Zertifikat ist nun zur Verifizierung des SSL-Zertifikats des AD-Servers auf den FreeIPA-Server zu kopieren. Ruft man dort das Installationsprogramm »ipa-server-install« auf, so wird das Windows-Sync-Plugin automatisch mitinstalliert. Zum Einsatz kommt es aber nur dann, wenn man später mittels »ipa-replica-manage« eine Daten-Replizierung zwischen einem Windows- und einem FreeIPA-Server einrichtet. Das Tool kennt hierfür einige zusätzliche Optionen:

- --winsync definiert eine Daten-Replizierung zwischen einem Windows- und FreeIPA
- --binddn bestimmt den Benutzer für die Anmeldung am Active-Directory
- --bindpw Passwort für diesen Benutzer
- --passsync Passwort für den PassSync Benutzer auf den Domänencontrollern
- --cacert Pfad zum ASCII/PEM codierten CA-Zertifikat, das zum Signieren des SSL-Zertifikats des Windows-Servers verwendet wurde. Dieses wird anschließend im FreeIPA-Zertifikatspeicher vorgehalten

Ein entsprechender Aufruf zum Einrichten eines »Replications-Agreements« könnte beispielsweise so aussehen wie im folgenden Beispiel gezeigt:

```
# ipa-replica-manage add --winsync --binddn
cn=administrator,cn=users,dc=tuxgeek,dc=de
--bindpw password --passsync password
--cacert /path/to/certfile.cer adserver.
tuxgeek.de -v
```

Hat man alle notwendigen Informationen angegeben, werden alle Daten, die sich im Container des Active Directory-Benutzers befinden, auf den FreeIPA-Server synchronisiert. Auf sie können dann alle Unix/Linux-IPA-Clients über eine native Schnittstelle zugreifen. Wichtig dabei ist, dass neue Benutzer, die sowohl auf Windows wie auch auf Linux-Clients zur Verfügung stehen sollen, im ADS anzulegen sind, weil die Synchronisierung nur in eine Richtung verläuft.

Fazit

Mit der nun erschienen Beta Version der kommenden FreeIPA-2.0 Generation zeigen die Entwickler in welche Richtung es gehen soll. Der Fokus liegt hier noch ganz klar auf dem Identity-Management. Dieses ist allerdings in Vergleich zur Version 1 doch deutlich erweitert worden. Gerade die Integration der Dogtag-PKI erleichtert das Erstellen und Verwalten von Zertifikaten doch ungemein. Im Policy-Bereich sind mit der Integration von HBAC- und Sudo-Regeln die ersten Schritte gemacht, weitere müssen jedoch folgen. Dass die Audit-Komponente leider noch gar nicht umgesetzt wurde ist ärgerlich, viele andere neue Features lassen jedoch über dieses Manko hinwegsehen. ■■■

Infos

- (1) FreeIPA: (<http://www.freeipa.org>)
- (2) Likewise: (<http://www.likewiseoftware.com>)
- (3) Dogtag-PKI: (<http://fedoraproject.pki.org>)
- (4) Fedora Netgroups Howto: (<http://directory.fedoraproject.org/wiki/Howto:Netgroups>)
- (5) Windows Sync-Howto: (<http://directory.fedoraproject.org/wiki/Howto:WindowsSync>)

Der Autor

horsten Scherf arbeitet als Consultant und Trainer bei red Hat EMEA und ist auf den Bereich Security spezialisiert.

Listing 13: »netgroup add«

```
01 # ipa netgroup-add QaAdmins --desc="QA-Admins"
02 -----
03 Added netgroup "qaadmins"
04 -----
05 Netgroup name: qaadmins
06 Description: QA-Admins
07 NIS domain name: virt.tuxgeek.de
08 IPA unique ID: 1d5e4e08-20f4-11e0-a795-525400cbdf57
09
10 # ipa netgroup-add-member qaadmins --users=tscherf
    --hosts=*.tuxgeek.de
11 Netgroup name: qaadmins
12 Description: QA-Admins
13 NIS domain name: virt.tuxgeek.de
14 External host: *.tuxgeek.de
15 Member User: tscherf
16 -----
17 Number of members added 2
18 -----
```

Listing 14: Zugriffsregelungen

```
01 # ipa hbac-add --type=allow --hostcat=all --srchostcat=all
    admin_sshd
02 # ipa hbac-add-user --users=admin admin_sshd
03 # ipa hbac-add-service --hbacsvcs=sshd admin_sshd
04
05 # ipa hbac-add --type=deny --usercat=all --srchostcat=all
    deny_all
06 # ipa hbac-add-host --hosts=all deny_all
```