Integrated identity management with FreeIPA

IDENTITY CHECK

FreeIPA offers integrated identity management and big ideas for the future. **BY THORSTEN SCHERF**

Enterprise Linux systems employ a set of standard tools for security, auditing, and identity management. These tools work well independently, once you get them all configured, but when it comes to integration, the admin often must improvise. Features such as central management of audit logs from multiple machines, as well as the ability to distribute SELinux policy modules to multiple machines, are often the domain of home-grown scripts. Although many proprietary solutions exist, they are typically expensive and inflexible.

The FreeIPA [1] project is an effort to combine a number of popular open source projects into a common, unified system. IPA stands for Identity, Policy, and Audit, but the developers clearly use this abbreviation with an eye on future goals. The current emphasis is on identity management, with support for Kerberos and LDAP. Future releases will offer central configuration and management of certificates, as well as policy and auditing features.

Figure 1 shows the individual FreeIPA version 1 components and how they cooperate. The combination of LDAP and Kerberos means that FreeIPA is easy to integrate with Microsoft's Active Directory System. Although the Linux world offers other options for Active Directory integration (such as Samba or Likewise [2]), Active Directory itself is only part of the solution for a fully integrated security and auditing tool. For instance, Active Directory does not offer anything in the line of policy or audit management for Linux systems, thus forcing admins to turn to other sources for these functions. Many Linux users must also consider whether it is a good idea to place their network security infrastructure in the hands of a proprietary technology like Microsoft Active Directory.

The Path Ahead

The focus of the current FreeIPA version 1 is on managing user and group identities. Migrating existing NIS solutions to FreeIPA for a secure LDAP environment with Kerberos passwords is easy. The developer version already offers synchronization with an existing Active Directory server; in fact, Active Directory integration should be available in the official FreeIPA version by the time this issue leaves the press.

Version 2, which is due for release early next year, will add more features. The identity management feature will be extended to handle machine accounts.

Another feature on the roadmap is a Certificate Authority (CA) for issuing user and service certificates. Of course, the two missing IPA components, Policy (P) and Audit (A), still need to be included. The policy component will not just handle SELinux rule set management. The developer's roadmap also includes central management of PAM settings, with *pam_access*, *pam_time*, and *pam_limits*. It will also be easier to assign user privileges via *sudo* because admins will be able to manage these settings centrally with FreeIPA.

The audit components primarily access the functions of popular audit daemons to ensure compliance with existing identity policies. Of course, a central audit rule rollout will include collecting audit events on individual machines. These audit events will be recorded on the FreeIPA server for reporting and evaluation.

FreeIPA is still a work in progress; the developers haven't yet achieved the full potential of this promising tool (see the box titled "The Path Ahead"). The current version, however, does provide LDAP and Kerberos support, as well as many other useful features. Here, I show you how to get started with FreeIPA.

Server Installation

Before you start installing the FreeIPA server itself, make sure all of the machines support DNS name resolution. Adding a couple of service (*SRV*) records to the existing DNS server will simplify later client configuration by allowing a DNS request to discover the responsible server and the Kerberos realm.

When you install the FreeIPA server, it will create a sample DNS zone file with all the required entries, and you can base your own DNS server extensions on this file (Listing 1).

To install the FreeIPA server on a Fedora system, just type *yum -y install ipa-server*. The server and all the required packages are available from the standard repositories and have been since Fedora 8. After installing, you need to call *ipa-server-install* to configure. If you prefer to create a matching DNS zone file directly, you can call the tool with the *--setup-bind* parameter. This step drops a zone file into the *tmp* folder.

Calling the setup routine installs the following components on your machine:

- NTP
- Fedora Directory Server
- MIT Kerberos
- Apache/TurboGears
- SELinux-targeted policy for FreeIPA

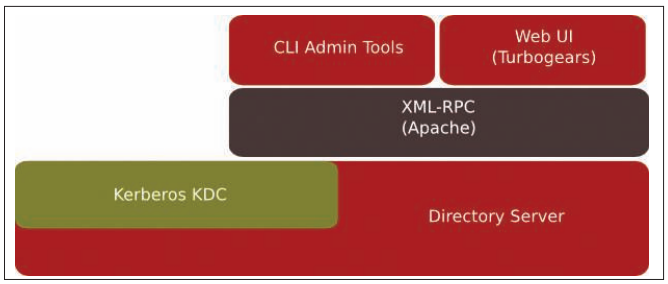

Figure 1: Popular open source tools running under the common umbrella of FreeIPA.

The installation program prompts you to enter the required information (e.g., the LDAP Base DN, Kerberos realm, server name); just a couple of minutes later, the server and all its components are ready to rumble. Next, type *kinit admin* and ask for a user ticket for *admin* to check that the Kerberos server is working properly.

The following call adds a new user to the directory/Kerberos server:

```
# ipa-adduser ➥
-f Thorsten -l Scherf tscherf
Password:
Password (again):
tscherf successfully added
```

If your password entry is in line with the password complexity policy, *ipa-finduser* will find the user account, which now exists on the directory server:

```
# ipa-finduser tscherf
Full Name: Thorsten Scherf
Home Directory: /home/tscherf
Login Shell: /bin/sh
Login: tscherf
```

If you need more information on the LDAP attributes, you can, of course, set up a Kerberos-authenticated connection to the LDAP server and query it for the information you need (Listing 2).

The *klist* tool now displays the transferred service ticket for the LDAP server:

```
[root@devel-srv1 ~]# klist -5
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: ➥
admin@VIRT.FOO.DE

Valid starting Expires ➥
Service principal
09/19/08 13:57:28 09/20/08 ➥
13:57:26 krbtgt/VIRT.FOO.DE@➥
VIRT.FOO.DE
09/19/08 13:57:42 09/20/08 ➥
13:57:26 ldap/devel-srv1.➥
virt.foo.de@VIRT.FOO.DE
```

Listing 1: DNS Extensions

```
01 $TTL 86400
02 @ IN SOA devel-srv1.virt.foo.de. root.devel-srv1.virt.foo.
   de. (
03 ; Dont forget to increment the serial number
04         2003040100 ;serial number
05         1H   ;refresh slave
06         5M   ;retry refresh
07         1W   ;expire zone
08         5M   ;cache time-to-live for negative answers
09 )
10 ; Name server resource records ( NS )
11 ; owner     TTL CL   type  RDATA
12 @        IN NS       devel-srv1.virt.foo.de.
13
14
15 ; Internet address resource records( A )
16 ; owner     TTL CL   type  RDATA
17 devel-srv1     IN A      192.168.122.100
18
19 ; ldap servers
20 _ldap._tcp   IN SRV  0 100 389 devel-srv1.virt.foo.de.
21
22 ;kerberos realm
23 _kerberos    IN TXT      VIRT.FOO.DE
24
25 ; kerberos servers
26 _kerberos._tcp     IN SRV 0 100 88     devel-srv1.virt.foo.
   de.
27 _kerberos._udp     IN SRV 0 100 88     devel-srv1.virt.foo.
   de.
28 _kerberos-master._tcp  IN SRV 0 100 88     devel-srv1.virt.
   foo.de.
29 _kerberos-master._udp  IN SRV 0 100 88     devel-srv1.virt.
   foo.de.
30 _kpasswd._tcp      IN SRV 0 100 464    devel-srv1.virt.foo.
   de.
31 _kpasswd._udp      IN SRV 0 100 464    devel-srv1.virt.foo.
   de.
32
33 ;ntp server
34 _ntp._udp        IN SRV 0 100 123    devel-srv1.virt.foo.de.
```

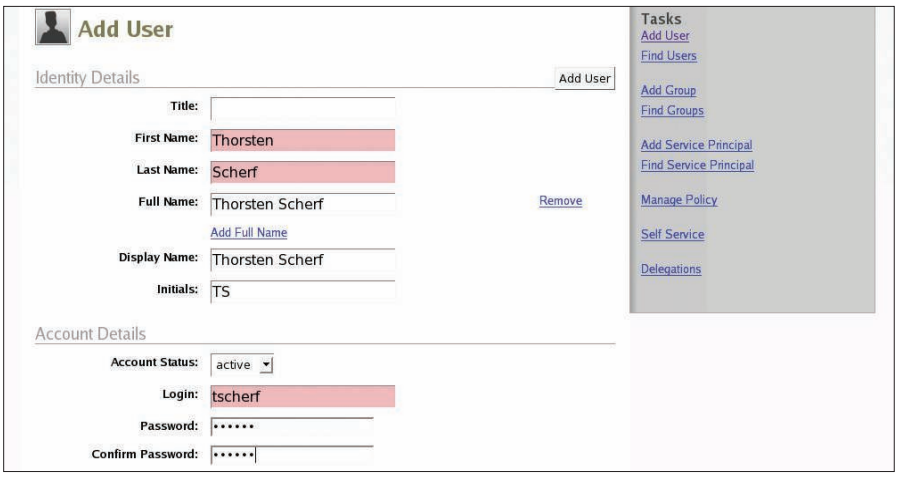

Figure 2: The web interface makes it easy to add users to the directory.

Of course, a web interface is available for easier handling of all of these tasks (Figure 2), but you do need to configure the web browser. Firefox shows the current configuration when you type *about:config*. The following commands are those you need to customize:

```
network.negotiate-
auth.trusted-uris .virt.foo.de
network.negotiate-auth.
delegation-uris .virt.foo.de
network.negotiate-auth.
using-native-gsslib true
```

After opening an https connection to the FreeIPA server, you can easily create or query user accounts via the web interface.

Client Configuration

A FreeIPA client is available for Fedora, Red Hat Enterprise Linux (RHEL), and a variety of Unix variants, including Solaris, AIX, HP-UX, and Mac OS X. Installing on a Fedora system is easy with a simple *yum* command line:

```
yum install ipa-client
ipa-admintools
```

If you then transfer the */etc/krb5.conf* Kerberos configuration file from the server to the client, all you need to do is call *ipa-client-install* to start the client installation. Thanks to the *dns_lookup_realm = true* entry in */etc/krb5.conf*, the client will ask its DNS server for all the necessary configuration information (Listing 3).

To test the server connection, you can use *kinit admin* on the client; if everything is working, the next step is to set up a host principal for the client in the Kerberos database and store the password locally on the client side:

```
# ipa-addservice host/
devel-client.virt.foo.de
# ipa-getkeytab host/
devel-client.virt.foo.de -k
/etc/krb5.keytab

Keytab successfully retrieved
and stored in: /etc/krb5.keytab
```

Kerberos Services

The next step is to configure a service to work with Kerberos. First, consider the example of an NFS server that the client machines can access via the secure NFSv4 protocol with Kerberos authentication. The server will ensure data integrity and privacy. To allow this to happen, you need to set up an NFS share on the IPA server:

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

Entering *echo SECURE_NFS = yes > /etc/sysconfig/nfs* activates all the required NFS services after the *service nfs start* command is issued.

Now you need to set up a service principal for the NFS service in the Kerberos database and export it to the server's *keytab* file:

```
# ipa-addservice
nfs/devel-srv1.virt.foo.de
# ipa-getkeytab
nfs/devel-srv1.virt.foo.de -k
/etc/krb5.keytab
Keytab successfully retrieved
and stored in: /etc/krb5.keytab
```

The client configuration is fairly similar. If you follow the same steps to create an NFS service principal and store it locally in the */etc/krb5.keytab* file, the *ipa-findservice* command will tell you whether or not it has worked. The *ipa-findservice* command will list all the host and service principals in the *keytab* file.

To make sure that the required NFS client services, *rpcgssd* and *rpcidmapd*, start correctly, you need to add a *SECURE_NFS = yes* entry to the *etc/sysconfig/nfs* file. Now you are ready for a secure NFSv4 mount:

Listing 2: Querying the Server

```
01 [root@devel-srv1 ~]# ldapsearch -Y
   GSSAPI uid=tscherf -LLL
02 SASL/GSSAPI authentication started
03 SASL username: admin@VIRT.FOO.DE
04 SASL SSF: 56
05 SASL installing layers
06 dn: uid=tscherf,cn=users,cn=accounts
   ,dc=virt,dc=foo,dc=de
07 uid: tscherf
08 objectClass: top
09 objectClass: person
10 objectClass: organizationalPerson
11 objectClass: inetOrgPerson
12 objectClass: inetUser
13 objectClass: posixAccount
14 objectClass: krbPrincipalAux
15 objectClass: radiusprofile
16 loginShell: /bin/sh
17 gidNumber: 1002
18 gecos: tscherf
19 sn: Scherf
20 homeDirectory: /home/tscherf
21 krbPrincipalName: tscherf@VIRT.FOO.
   DE
22 givenName: Thorsten
23 cn: Thorsten Scherf
24 uidNumber: 1100
25 memberOf: cn=ipausers,cn=groups,cn=a
   ccounts,dc=virt,dc=foo,dc=de
```

```
# mount -v -t nfs4 ➥
-o sec=krb5p devel-srv1:/ ➥
/mnt/nfs4
```

Note that FreeIPA stores the complete Kerberos configuration in LDAP (Figure 3). Because native Kerberos tools such as *kadmin* or *kadmin.local* do not offer a native LDAP interface, you cannot use them to manage the Kerberos database. Instead, administrators will always need to use the FreeIPA tools for administrative tasks.

Highly Available Data

After completing the basic server configuration, you should replicate the directory server data on a second machine. Because FreeIPA stores the complete Kerberos configuration and the Kerberos database in LDAP, this replication gives you a second master server in next to no time.

In the case of a master server failure, the secondary master still has all the data, on which you can even edit the data. Once the primary master goes back online, the modified data is then replicated back to it. The use of at least two servers is also a good idea for load balancing purposes.

If you store data at two distinct geographical locations, you should consider configuring more servers and setting them up as replicas to avoid the use of a WAN connection whenever you query or change the directory.

The primary master has a configuration file with all the information you need to create a second server:

```
# ipa-replica-prepare➥
devel-srv2.virt.foo.de
```

Now just copy the file created at the last step to the replica host and launch the installation there:

```
# scp /var/lib/ipa/replica-➥
info-devel-srv2.virt.foo.de ➥
root@devel-srv2:/tmp/
# ipa-replica-install ➥
/tmp/replica-info-devel-➥
srv2.virt.foo.de
```

Assuming the installation program completes without error, you can start a replication of the LDAP database. If you then assign a separate DNS zone file to the replica, you have two independent servers. With the use of *ipa-replica-manage*, you can view and modify any replication agreements set up in this way.

Active Directory Synchronization

Administrators can use *ipa-replica-manage* to synchronize data between a Windows Active Directory server and a FreeIPA server. The current developer version of FreeIPA already implements this feature. To do so, you need a TLS/SSL certificate on the Windows server; this is mandatory for synchronizing the data on the Active Directory server with FreeIPA. The Fedora Wiki has a HOWTO [3].

Now copy the CA certificate used here to the FreeIPA server to verify the TLS/SSL certificate on the Active Directory server. When you launch the *ipa-server-install* program, the Windows Sync plugin is installed automatically. However, the plugin is not used unless you set up data replication between a Windows server and a FreeIPA server with *ipa-replica-manage*. The tool has several new options:

- *winsync* – defines data replication between a Windows server and a FreeIPA server.
- *binddn* – defines the user account for logging into the Active Directory. This user needs a number of privileges (read, write, search, password change, DirSync).
- *bindpw* – specifies a password for the specified user account.
- *cacert* – defines a path to the ASCII/PEM-encoded CA certificate that is used to sign the Windows server's TLS/SSL certificate. This setting is then stored in the FreeIPA certificate repository.

After entering the required information, the Active Directory user's container is synchronized with the FreeIPA server. All Unix/Linux IPA clients can then access this information via a native interface. Because the synchronization process is unidirectional, new users who have accounts on both Windows and Linux clients must first be created in Active Directory.

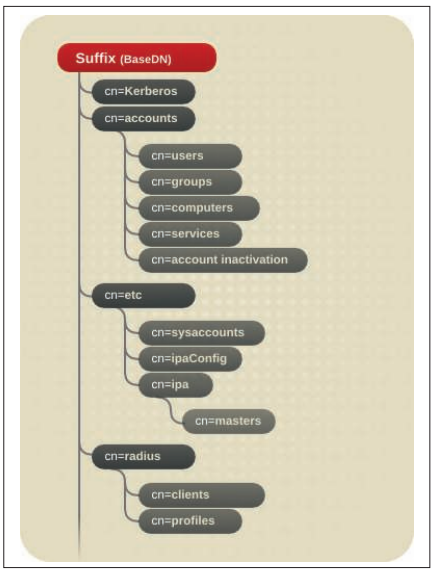

Figure 3: FreeIPA stores the Kerberos database in an LDAP container.

Conclusions

FreeIPA unifies a number of popular tools under a common umbrella. Version 1 focuses on storing identities. Although components such as certificate, audit, and policy management are still missing, it is easy to see where the product is heading. ■

Listing 3: Client Installation

```
01 [root@devel-client ~]#
   ipa-client-install
02 Discovery was successful!
03 Realm: VIRT.FOO.DE
04 DNS Domain: virt.foo.de
05 IPA Server: devel-srv1.virt.foo.de
06 BaseDN: dc=virt,dc=foo,dc=de
07
08 Continue to configure the system
   with these values? [y/N]: y
09
10 Created /etc/ipa/ipa.conf
11 Configured /etc/ldap.conf
12 LDAP enabled
13 nss_ldap is not able to use DNS
   discovery!
14 Changing configuration to use
   hardcoded server name: devel-srv1.
   virt.foo.de
15 Kerberos 5 enabled
16 NTP enabled
17 Client configuration complete.
```

INFO

[1] FreeIPA: *http://www.freeipa.org*

[2] Likewise: *http://www.likewisesoftware.com*

[3] Windows Sync HOWTO: *http://directory.fedoraproject.org/wiki/Howto:WindowsSync*