Identity Management

Scaling Out and Up

Jan Pazdziora
Principal Software Engineer
Identity Management Engineering, Red Hat
jpazdziora@redhat.com

15th October 2014

Identity

- Users; user groups. Hosts; host groups; services.
- Identities can hold additional attributes and objects, such as certificates or keytabs.
- They can be used to drive behavior.

```
$ id bob
uid=1108923(bob) gid=1108923(bob) groups=1108923(bob),10(wheel)
```

```
$ ssh alice@host1.example.com
[alice@host1 ~]$
```

- Traditional identity sources: `/etc/passwd`, `/etc/group`.
- Goals for large organizations:
  - Manage identities centrally, or consume external identities managed by other departments.
  - Use identities for applications as well, not just for operating system.

Centralized identity sources

- Typically achieved by using directory servers.
- Not exactly trivial.
  - Master `posixAccount`, `organizationalPerson`, `organizationalRole`, learn about `*.schema` files, indexes, ACLs, get `cn=` and `dc=` right...
  - ```
    $ slapcat ... > data.ldif
    $ vi data.ldif
    $ ldapadd -x -D ... -W -f data.ldif
    $ ldapmodify -x -D ... -W -f
    ```
  - Deal with new versions of software and schemas.
- When Kerberos is required, manually edit `/var/kerberos/krb5kdc/kdc.conf`, `/etc/krb5.conf`, get familiar with `kadmin.local`, ...
- Manually editing DNS zone files is not easy either.
    ```

Integrated identity management

- Just external centralized identity source is not enough.
- Solution: FreeIPA (IPA).
- Layer on top of directory server, Kerberos key distribution center, optionally DNS, and certification authority.
- Subsystems are kept in sync via LDAP backend. Creating user means Kerberos principal immediately exists, adding host can add IP address to DNS automatically, creating service means Kerberos keytab can be retrieved.
- With host-based access control rules, sudo rules, automount maps, ...
- With configuration script:

```
# ipa-server-install ...
```

- With CLI.
- With WebUI.

IPA CLI

```
$ ipa user-add --random --first David --last Smith david
------------------
Added user "david"
------------------
  User login: david
  First name: David
  Last name: Smith
  Full name: David Smith
  Display name: David Smith
  Initials: DS
  Home directory: /home/david
  GECOS: David Smith
  Login shell: /bin/sh
  Kerberos principal: david@EXAMPLE.COM
  Email address: david@example.com
  Random password: -VyDwrTgXKXK
  UID: 830600007
  GID: 830600007
  Password: True
  Member of groups: ipausers
  Kerberos keys available: True
```

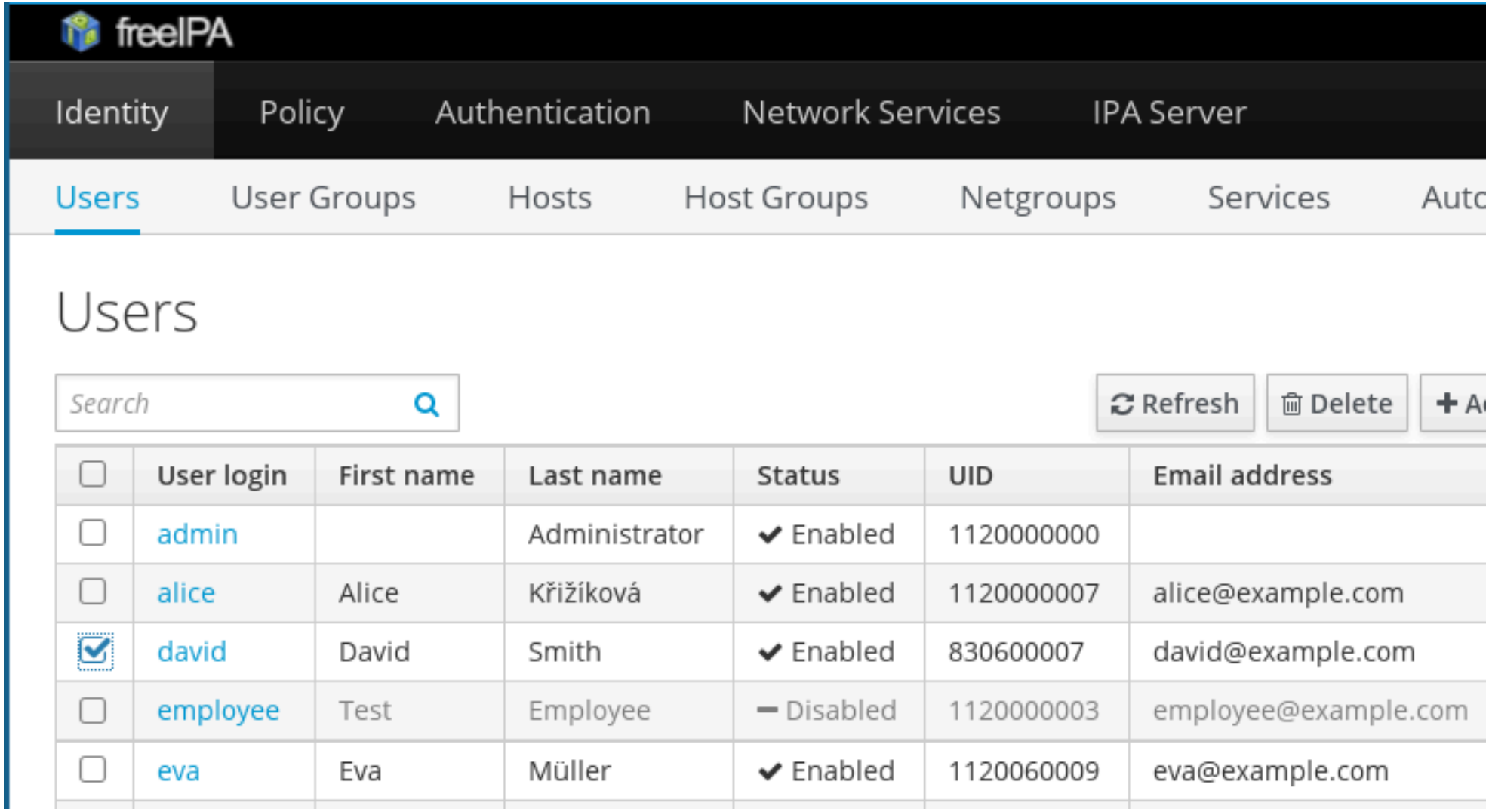
IPA CLI (continued)

```
$ ipa host-add --ip-address 10.0.0.34 wiki2.example.com --no-reverse
--------------------------------
Added host "wiki2.example.com"
--------------------------------
  Host name: wiki2.example.com
  Principal name: host/wiki2.example.com@EXAMPLE.COM
  Password: False
  Keytab: False
  Managed by: wiki2.example.com
```

```
$ ipa dnsrecord-find example.com wiki2
  Record name: wiki2
  A record: 10.0.0.34
----------------------------
Number of entries returned 1
----------------------------
```

```
$ host wiki2
wiki2.example.com has address 10.0.0.34
```

IPA WebUI


freeIPA
Identity
Policy
Authentication
Network Services
IPA Server
Users
User Groups
Hosts
Host Groups
Netgroups
Services
Users
Search
Refresh
Delete
User login
First name
Last name
Status
UID
Email address
admin
Administrator
Enabled
1120000000
alice
Alice
Křižíková
Enabled
1120000007
alice@example.com
david
David
Smith
Enabled
830600007
david@example.com
employee
Test
Employee
Disabled
1120000003
employee@example.com
eva
Eva
Müller
Enabled
1120060009
eva@example.com

Simple use on Linux clients

- sssd: System Security Services Daemon.
- Caching for speed and offline use, failover support, multiple domains.
- Integration to PAM via `pam_sss.so`.
- Sudo rules, automount maps, SELinux user mapping, handling of ssh public keys, ...
- With configuration script:

```
# ipa-client-install ...
```

- IPA is not mandatory on the server side — sssd can be configured to use other provider types than just IPA servers.

Replicating IPA

- For failover.
- For high availability.
- Create GPG-encrypted replica information file.

```
[root@ipa ~]# ipa-replica-prepare ipa2.example.com
```

- Copy the encrypted file to the replica machine.
- Configure the replica

```
[root@ipa2 ~]# ipa-replica-install replica-info-ipa2.example.com.gpg
```

- Multi-master replication means either of the master or replica can be used for write operations.
- sssd and other OS-libraries are able to failover to replicas.

Cross-realm trust

- Active Directory users accessing Linux machines and services run in Linux realm.
- And vice versa.
- Enable trust support in IPA

```
# ipa-adtrust-install --netbios-name=EXAMPLE -a password
```

- Set up DNS forwarding in IPA to AD

```
# ipa dnsforwardzone-add addomain.test \
        --forwarder=10.1.2.3 --forward-policy=only
```

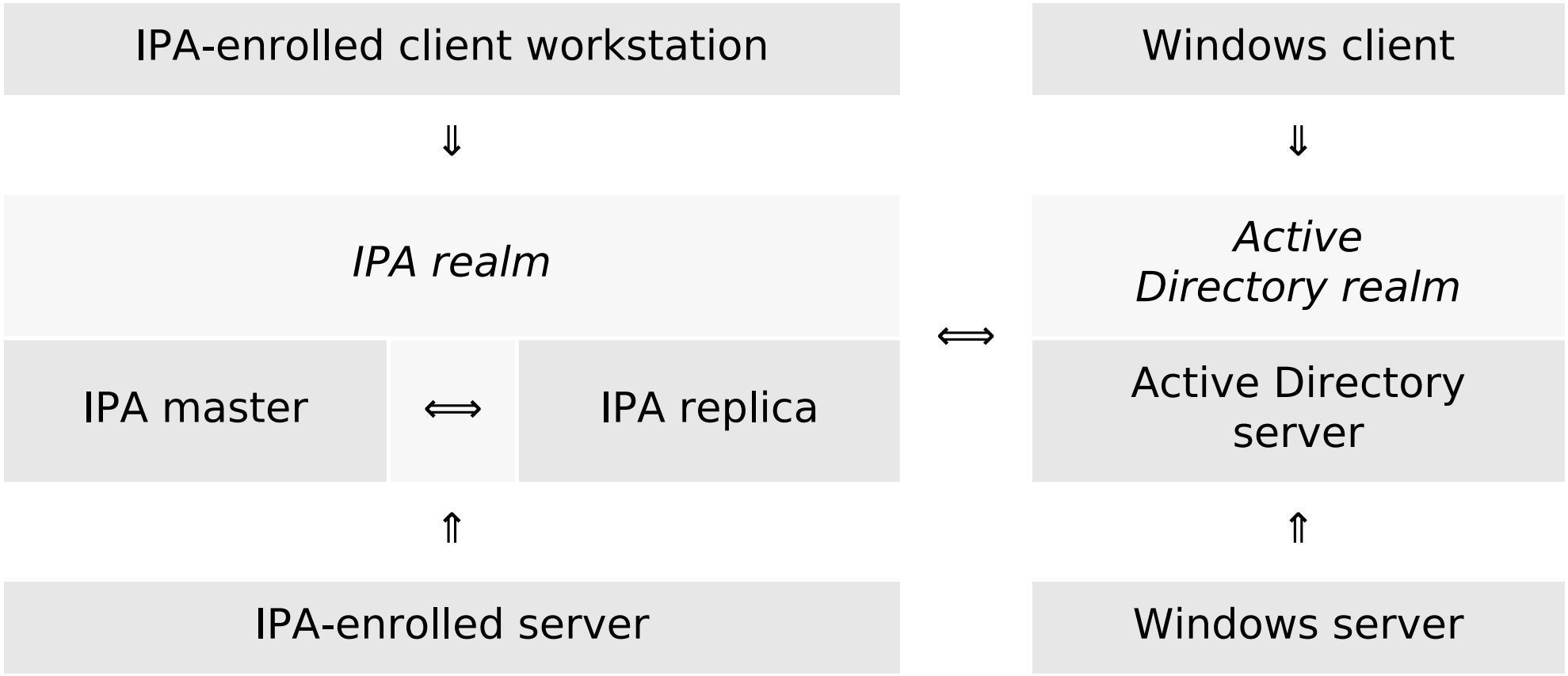
- Set up DNS forwarding in AD to IPA

```
C:\> dnscmd 127.0.0.1 /ZoneAdd EXAMPLE.COM /Forwarder 10.0.0.10
```

- Establish two-way trust

```
# ipa trust-add --type=ad ADDOMAIN.TEST --admin Administrator ...
```

The architecture



The arrows show the direction of the enrollment / trust and authentication operations. Communication can place from any client to any server.

Identity management scaling out

- With IPA replicas, robust identity and authentication source setup can be achieved.
- With cross-realm trust, Linux-hosted services on IPA-enrolled machines can be accessed by users authenticated to Windows domain.
- In Windows, users get Kerberos ticket (TGT) just by logging in.
- In Linux, `kinit` or graphical tools can be used.
- The net result is simpler user experience with less passwords to be remembered and easier management of the whole setup.
- IPA or sssd not present in your favorite Linux distribution?
  - While we might not be able to add them there ourselves, we will be happy to help people who would like to package and maintain them.
  - Container images are available for testing purposes.

User identities in applications

- So far we have looked at OS-level authentication and identity services.
  - IPA on server and sssd on clients provide robust and flexible solution.
- Assume organization deploys new application (typically Web-based).
  - How will the user identities be managed?
- Proposal: use what already works for operating system.
- Instead of every application or application framework implementing the same complex support for various types of authentication sources and failover and caching and reestablishing network connections, take advantage of sssd.

Authentication in Apache modules

- mod_authnz_pam
  - PAM authentication.
  - Access control module (even for Kerberos / mod_auth_kerb).
  - With configurable PAM service name and `pam_sss.so`, it can take advantage of IPA's HBAC mechanism.
- mod_intercept_form_submit
  - Intercept logon form POST submission.
  - Attempt PAM authentication with [login, password] pair.
- mod_lookup_identity
  - Retrieve additional attributes of authentication user from sssd.
  - What Web application often need: name, email address, group membership.

Module overview

<table>
<tr><th rowspan="2">Authen. Method</th><th colspan="3">Apache Modules</th></tr>
<tr><th>Authentication</th><th>Access Check</th><th>Extra User Info</th></tr>
<tr><td>Application</td><td colspan="3">None</td></tr>
<tr><td>Kerberos</td><td>mod_auth_kerb</td><td rowspan="3">mod_authnz_pam</td><td rowspan="4">mod_lookup_identity</td></tr>
<tr><td rowspan="2">Certificate</td><td>mod_nss</td></tr>
<tr><td>mod_ssl</td></tr>
<tr><td>Form-Based</td><td colspan="2">mod_intercept_form_submit</td></tr>
</table>

Applications easy to adapt

- Changes to the application or framework code are very small.
- Many already support REMOTE_USER authentication, from HTTP Basic Authentication days.
- Applications become accessible by all users in the organization, in authenticated manner.
  - No more manually managing users in applications' user tables
  - User records can get auto-populated and updated whenever the user logs in.
- Applications still locally manage mapping of groups to roles or authorization permissions.
  - The user group membership information provided by the external identity provider via Apache drives the access check.

Identity management scaling up

- The same code and setups which support large scale deployments on operating system level can be used for Web applications as well.
- Spacewalk, Foreman, or ManageIQ already take advantage of it.
- Django being investigated.
- Your favorite application not supporting Kerberos or IPA's HBAC?
  - While we might not be able to enable them for this type of authentication ourselves, we will be happy to help people who would like to add the feature.

Conclusion and references

- Solutions for flexible and scalable identity management setups exist:
  - Integrated server side, with AD integration.
  - Operating system level.
  - Web applications.
- They make mid-size setups easy, large-size setups possible.
- Explore them, use them, let us know what you think.

- www.freeipa.org
- fedorahosted.org/sssd/
- www.freeipa.org/page/Web_App_Authentication
- github.com/adelton/docker-freeipa