Escape from the Identity crisis with FreeIPA
Identity Management

- **What is Identity Management?**
  - “Identity management (IdM) describes the management of individual principals, their authentication, authorization, and privileges within or across system and enterprise boundaries with the goal of increasing security and productivity while decreasing cost, downtime and repetitive tasks.”

- **This is the theory, but what does it mean?**
  - **Identities:** users, machines, services/applications
  - **Authentication:** password, 2FA (OTP), biometry…
  - **Authorization:** Policies, ACLs, DAC, MAC
  - Can be configured **locally**, but what about 10000+ machine network? Synchronization nightmare…

Wikipedia
Why Should I Care?

- **Infrastructure**
  - Every networked machine needs **accounts** and **authentication** services
  - Required by both small startups and enterprises; by both **cloud** and **on-premise infrastructure**
  - IdM reduces errors, improves **productivity** for both admins and users by simplifying management

- **Applications**
  - Applications usually have users too and a need to hook in central IdM servers
  - Do we really need to re-invent user table and authentication for every new application?
Tough Life in the FOSS World

● There is a lot of integrated and easy to use proprietary solutions
  ○ Active Directory, IBM Tivoli, Directory Server, Novell eDirectory…
    ▪ Simple configuration, hides complexity from users
● Open Source world was always better in providing tools
  ○ LDAP (different flavours), Kerberos, Samba/Samba 4 DC
  ○ NIS, NIS+
● This is great, so we just need to deploy the tools, configure and profit!
● Easy, right?
Not so easy...
Integrating LDAP, Kerberos together

- **Main challenges:**
  - OS tools usually solve *individual problems*
    - “Do one thing and do it well”
  - Bag of technologies lacking integration
  - Lot of configuration options, but lack of good user interfaces

- **Is the situation really that bad?**
Introducing FreeIPA
FreeIPA – Open Source IdM Project

- **Identity Management system**
  - Centrally manage **identities** (users, hosts, services)
  - **Authentication** (password, OTP) and **authorization** services

- **Control&UI for native Linux services**
  - SUDO, Automount, SELinux user roles...

- **Control&UI for core infrastructure services**
  - DNS, Certificates

- **Acts as a gateway between Linux and Windows realms**

- **Not a monolite application**
  - Integrates many open source projects and technologies with own communities
The Core

- **Directory Server**
  - Main **data backend** for all other services
  - Custom plugins: authentication hooks, password policy, compatibility tree (slapi-nis), validation extended operations...

- **Kerberos KDC**
  - Provides **SSO** authentication for entire FreeIPA realm

- **PKI Server**
  - **Certificates** for services (web, LDAP, TLS) and users

- **HTTP Server**: Provides management interfaces

- **SSSD**: Client

- **Samba**: Communication with Windows realms
Architecture

PKI, KDC, HTTP, CLI/UI, LDAP, NTP, DNS

FreeIPA Server

Admin

Web UI, CLI, API (RPC)

Identity, Authentication, Authorization, Certificates

Solaris, Fedora, Debian

Kerberos Cross-Realm Trust

Active Directory

Windows, Windows, Windows

SSSD, SSSD
Easy Deployment

● Hides complexity of LDAP+Kerberos+CA+... deployment

● We have few requirements though:
  ○ Sane DNS environment (reverse records for servers)
    ▪ DNS is crucial to identify machines
    ▪ Kerberos service principals, X.509 Certificates use DNS names
  ○ Static FQDN hostnames

● Configuration with one command
  ○ ipa-server-install, ipa-replica-install ipa-client-install

● Supports replicas
  ○ Essential for redundancy and fault protection

● Available in Fedora, RHEL, CentOS, Debian, Docker image
Features: Identity 1/2

- **Users, groups:**
  - Manage **SSH public keys, certificates** (even for authentication)
  - Role-based access control, self-service, delegation

- **Hosts, host groups, netgroups:**
  - Manage host life-cycle, enrollment

- **Services/applications**
  - Manage keytab, certificates

- **Automatic group membership based on rules**
Features: Identity 2/2

- 2FA support
  - Support for both password and 2FA/OTP
  - Native HOTP/TOTP support with FreeOTP and Yubikey
  - Proxied 2FA authentication over RADIUS for other solutions
$ kinit admin
Password for admin@EXAMPLE.COM:

$ klist
Ticket cache: KEYRING:persistent:1000:1000
Default principal: admin@EXAMPLE.COM

<table>
<thead>
<tr>
<th>Valid starting</th>
<th>Expires</th>
<th>Service principal</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/01/15 12:00:00</td>
<td>01/02/15 12:00:00</td>
<td>krbtgt/EXAMPLE.COM@EXAMPLE.COM</td>
</tr>
<tr>
<td>renew until</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01/03/15 12:00:00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CLI Example 2/2

```bash
$ ipa user-add --first=John --last=Doe jdoe --random
-----------------
Added user "jdoe"
-----------------

<table>
<thead>
<tr>
<th>User login:</th>
<th>jdoe</th>
</tr>
</thead>
<tbody>
<tr>
<td>First name:</td>
<td>John</td>
</tr>
<tr>
<td>Last name:</td>
<td>Doe</td>
</tr>
<tr>
<td>Full name:</td>
<td>John Doe</td>
</tr>
<tr>
<td>Display name:</td>
<td>John Doe</td>
</tr>
<tr>
<td>Initials:</td>
<td>JD</td>
</tr>
<tr>
<td>Home directory:</td>
<td>/home/jdoe</td>
</tr>
<tr>
<td>GECOS field:</td>
<td>John Doe</td>
</tr>
<tr>
<td>Login shell:</td>
<td>/bin/sh</td>
</tr>
<tr>
<td>Kerberos principal:</td>
<td><a href="mailto:jdoe@EXAMPLE.COM">jdoe@EXAMPLE.COM</a></td>
</tr>
<tr>
<td>Email address:</td>
<td><a href="mailto:jdoe@example.com">jdoe@example.com</a></td>
</tr>
<tr>
<td>Random password:</td>
<td>xMc2XkI=ivVM</td>
</tr>
<tr>
<td>UID:</td>
<td>1998400002</td>
</tr>
<tr>
<td>GID:</td>
<td>1998400002</td>
</tr>
<tr>
<td>Password:</td>
<td>True</td>
</tr>
<tr>
<td>Kerberos keys available:</td>
<td>True</td>
</tr>
</tbody>
</table>
```
## Web UI Example

### Users

<table>
<thead>
<tr>
<th>User login</th>
<th>First name</th>
<th>Last name</th>
<th>Status</th>
<th>UID</th>
<th>Email address</th>
<th>Telephone Number</th>
<th>Job Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td></td>
<td></td>
<td>Enabled</td>
<td>1120000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>employee</td>
<td>Test</td>
<td>Employee</td>
<td>Enabled</td>
<td>1120000003</td>
<td><a href="mailto:employee@demo1.freeipa.org">employee@demo1.freeipa.org</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>helpdesk</td>
<td>Test</td>
<td>Helpdesk</td>
<td>Enabled</td>
<td>1120000004</td>
<td><a href="mailto:helpdesk@demo1.freeipa.org">helpdesk@demo1.freeipa.org</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>manager</td>
<td>Test</td>
<td>Manager</td>
<td>Enabled</td>
<td>1120000001</td>
<td><a href="mailto:manager@demo1.freeipa.org">manager@demo1.freeipa.org</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Showing 1 to 4 of 4 entries.
Policy - Authorization

- **HBAC – Host Based Authentication Control**
  - Who
  - Where
  - How
  - Main authorization feature, enforced by SSSD

- **SUDO - privileged user commands**
  - Very popular extension
  - Host and Group based rules
Policy – SUDO Example 1/3

```
# ipa user-add --first Foo --last Bar fbar --random
--------------
Added user "fbar"
--------------
    User login: fbar
...

# ipa sudocmd-add `which cat`
----------------------------------
Added Sudo Command "/usr/bin/cat"
----------------------------------
    Sudo Command: /usr/bin/cat

# ipa sudorule-add can_cat
--------------------------
Added Sudo Rule "can_cat"
--------------------------
    Rule name: can_cat
    Enabled: TRUE
```
Policy – SUDO Example 2/3

```bash
# ipa sudorule-add-allow-command can_cat --sudocmds `which cat`
Rule name: can_cat
Enabled: TRUE
Sudo Allow Commands: /usr/bin/cat

# ipa sudorule-add-host can_cat --hosts `hostname`
Rule name: can_cat
Enabled: TRUE
Hosts: ipa.example.com
Sudo Allow Commands: /usr/bin/cat

# ipa sudorule-add-user can_cat --users fbar
Rule name: can_cat
Enabled: TRUE
Users: fbar
Hosts: ipa.example.com
Sudo Allow Commands: /usr/bin/cat
```
# sudo su - fbar

$ sudo -l
...
[sudo] password for fbar:
Sorry, try again.
[sudo] password for fbar:
Matching Defaults entries for fbar on ipa:
   !visiblepw, env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE INPUTRC KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
...

User fbar may run the following commands on ipa:
   (root) /usr/bin/cat

$ sudo cat /etc/shadow
root:$1$aDa.rvGl$lPTu/xlriXaFpLcUzX8fy/::0:99999:7:::
bin:*:16489:0:99999:7:::
...
Other Policies

- **Automount**
  - Keep automount information centrally, not locally

- **SELinux policy user**
  - For very confined environments leaning heavily on SELinux
  - Assign **SELinux user role** by host/user based rules
Features: Certificates

- **Manage life-cycle for service and host certificates**
  - Can be used for both encryption and authentication

- **From version 4.2, FreeIPA can also can:**
  - Issue user certificates
  - Use different certificate profiles
Certificate Example 1/2

# ipa-getcert request -d /etc/httpd/nssdb -n My-Cert -K foo/`hostname` -N CN=`hostname`, O=EXAMPLE.COM -g 2048 -p /etc/httpd/alias/pwdfile.txt
...

Request ID '20150624062211':
  status: MONITORING
  stuck: no
  key pair storage: type=NSSDB,location='/etc/httpd/alias',nickname='My-Cert',token='NSS Certificate DB',pinfile='/etc/httpd/alias/pwdfile.txt'
  certificate: type=NSSDB,location='/etc/httpd/alias',nickname='My-Cert',token='NSS Certificate DB'
  CA: IPA
  issuer: CN=Certificate Authority,O=EXAMPLE.COM
  subject: CN=ipa.example.com,O=EXAMPLE.COM
  expires: 2017-06-24 06:22:16 UTC
  principal name: foo/ipa.example.com@EXAMPLE.COM
  key usage: digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment
  eku: id-kp-serverAuth,id-kp-clientAuth
  track: yes
  auto-renew: yes
Certificate Example 2/2

# certutil -L -d /etc/httpd/alias/ -n 'My-Cert'
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number: 11 (0xb)
      Signature Algorithm: PKCS #1 SHA-256 With RSA Encryption
      Issuer: "CN=Certificate Authority,O=EXAMPLE.COM"
   Validity:
      Not After : Sat Jun 24 06:22:16 2017
   Subject: "CN=ipa.example.com,O=EXAMPLE.COM"
...

Certificate Trust Flags:
   SSL Flags:
      User
   Email Flags:
      User
   Object Signing Flags:
      User
Core Infrastructure - DNS

- Optional feature
- DNS data stored in LDAP
- Plugin for BIND9 name server (bind-dyndb-ldap)
  - Bridge between LDAP and DNS worlds
- Integration of DNS records with the rest of the framework
Active Directory Integration

Indirect

Direct
Direct vs. Indirect

**Direct**
- SSSD joins AD directly
- Linux systems are the **second class** citizens
- Need to cover per-host license
- No native management for Linux services or POSIX data
  - There is *Identity Management for Unix* extension
    - Deprecated anyway
- Popular for simple infrastructures with few Linux systems

**Indirect**
- FreeIPA is involved, manages Linux hosts natively
- SSSD joins FreeIPA server, FreeIPA server has trust with AD
- Number of hosts can expand without license costs
SSSD

The Client Part

freeIPA

identity | policy | audit
SSSD

- **SSSD is a service running on each client**
  - Used to retrieve information from a central identity management system.
- **Connects a Linux system to a central identity stores**
  - Active Directory
  - FreeIPA
  - Any other directory server
- **Provides authentication and access control**
- **Top technology in the evolution chain of the client side IdM components**
  (nss_ldap, nss-pam-ldapd)
SSSD Features

- **Multiple parallel sources of identity and authentication – domains**
- **All information is cached locally for offline use**
  - Remote data center use case
  - Laptop or branch office system use case
- **Advanced features for**
  - FreeIPA integration
  - Active Directory integration
Architecture

- **SSSD**
  - Cache
  - NSS Responder
  - Backend Process
  - Identity Provider
  - Auth Provider

- **Identity Server**
- **Authentication Server**

**Client**
- nss_sss
  - Client "id"
- nss_sss
  - Client "login"
Web Applications
Integration for Web Applications 1/2

● **Standard way of creating Web App:**
  ○ Working on a concept - forget about users, just use static user+pass
  ○ Before shipping - we need users, let us hack some user table
  ○ Later - we also need to let AD/LDAP users log in! Let us hack some more
  ○ …

● **Repetitive user management and authentication coding**

● **We have tools to provide OS-level IdM services**
  ○ Why not using them for Web Applications also?
  ○ We get Kerberos, OTP, AD Trusts, authorization, user information... for free!
Integration for Web Applications 2/2

- Apache modules implemented for the FreeIPA project:
  - `mod_auth_gssapi`: Kerberos authentication
  - `mod_authnz_pam`: system-level authorization
  - `mod_intercept_form_submit`: form based login via SSSD
  - `mod_lookup_identity`: additional user info to application
Resources
Resources

● **FreeIPA**
  ○ Project wiki: [www.freeipa.org](http://www.freeipa.org)
    ▪ Web applications: [http://www.freeipa.org/page/Web_App_Authentication](http://www.freeipa.org/page/Web_App_Authentication)
  ○ Project trac: [https://fedorahosted.org/freeipa/](https://fedorahosted.org/freeipa/)

● **SSSD**:  
  ○ Project trac and wiki: [https://fedorahosted.org/sssd/](https://fedorahosted.org/sssd/)

● **Developer blogs**  