FreeIPA hands-on tutorial

Fedora 18 update: Active Directory trusts and more

Alexander Bokovoy
Jakub Hrozek
Martin Košek

Red Hat Inc.

LinuxCon Europe

November 5^{th}, 2012

Section 1

Preparation

fedora

Lab structure

- Use cases will use 3 VMs and the host machine
  - **server**: `server.ipa-X.example.com` - will host an IPA server
  - **replica**: `replica.ipa-X.example.com` - will host an IPA replica
  - **client**: `client.ipa-X.example.com` - will host IPA client with a web server
- Instructor machine hosts:
  - **IPA**: `server.ipa-0.example.com` - will host an IPA server
  - **IPA**: `replica.ipa-0.example.com` - will host an IPA replica
  - **IPA**: `client.ipa-0.example.com` - will host an IPA client with a web server
  - **AD**: `ad.example.com` - Active Directory domain
  - **AD**: `dc.ad.example.com` - Active Directory domain controller

Section 2

Installation

fedora

Install IPA server

- Check install options in `ipa-server-install --help`
  - Core options: `--external-ca`, `--setup-dns`, `--selfsign`
- Most common install issues:
  - broken DNS, bad `/ect/hosts` configuration
    - `--no-host-dns`, `--setup-dns`
  - Remains after the last unsuccessful install
    - `/var/lib/ipa/sysrestore/`
  - Time issues (Kerberos time sensitive) - on clients, replicas

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

fedora

Install IPA server (cont.)

- `kinit` as *admin*, check tickets with `klist`
- Check logs (useful for debugging):
  - `/var/log/pki-ca/debug`
  - `/var/log/pki-ca-install.log`
  - `/var/log/dirsrv/` (permissions!)
  - `/var/log/messages`
- Try `ipactl` command
- See certificates tracked in certmonger:
  - `ipa-getcert list`
- Check main IPA configuration: `/etc/ipa/default.conf`
  - `base DN, realm`
- Check automatically created DNS records (A, SRV)
  - `ipa dnszone-find`
  - `ipa dnsrecord-find`
- Check Web UI interface

Section 3

Active Directory trusts

fedora

Kerberos cross-forest trusts

FreeIPA deployment is a fully managed Kerberos realm

- Can be integrated with Windows as RFC4120-compliant Kerberos realm
- Traditional Kerberos trust management applies:
  - on GNU/Linux side `~/.k5login` should be defined to impersonate users with identities
  - on Active Directory side manual mapping is performed with special tools in a similar way
- Does not scale well for thousands of users and hosts:
  - a foreign realm principal impersonates our realm's user
  - requires additional management of special users to impersonate doubling the management effort
  - mapping has to happen on every single machine. Manually?

fedora

Kerberos cross-forest trusts

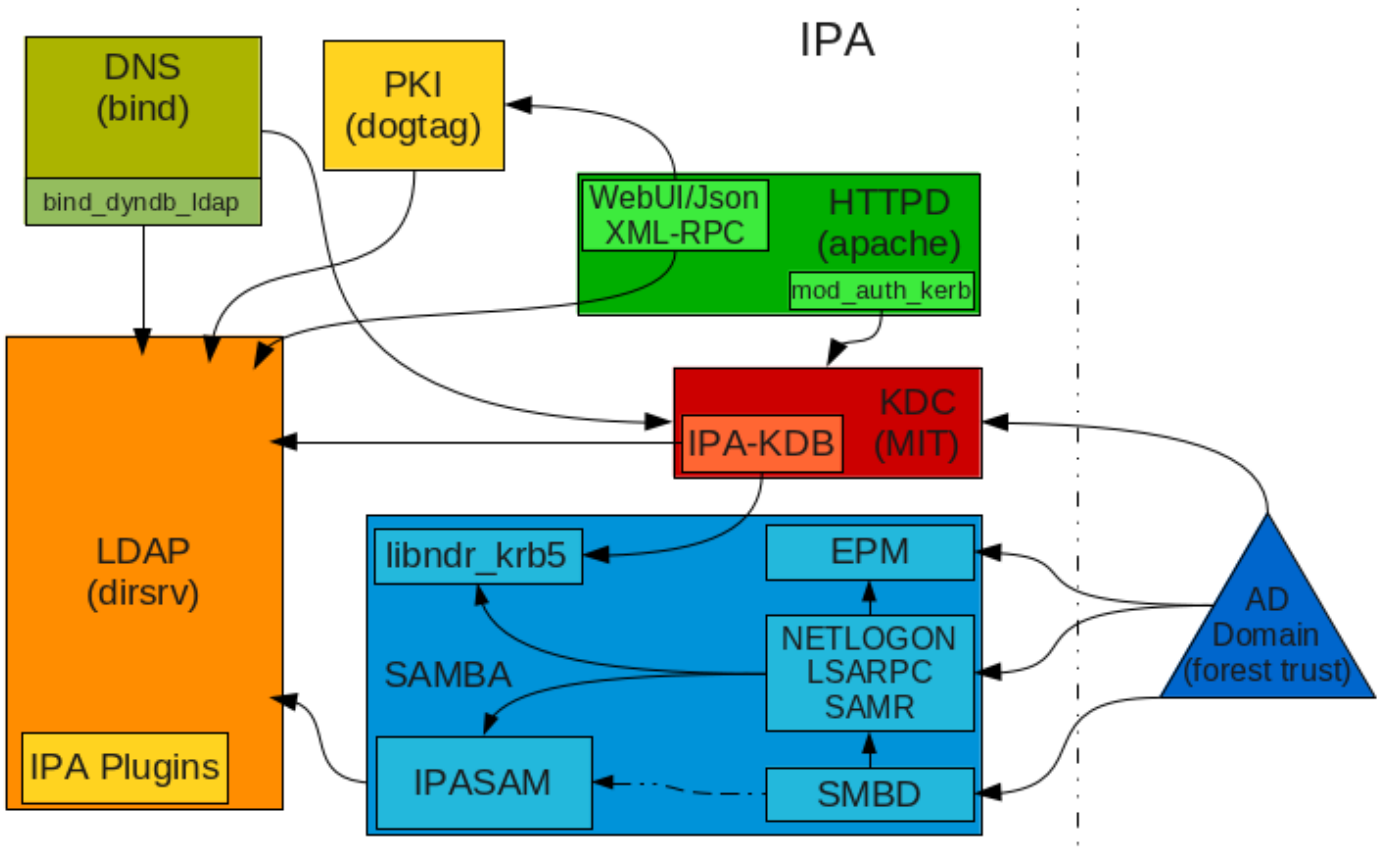
Active Directory native cross forest trusts

- Require two Active Directory domains
- AD domain establishes trust with another AD domain via LSA RPC
- AD uses LSA RPC and Global Catalog to map incoming principals to SIDs
  - technically: KDC + CLDAP + LSA RPC + LDAP on a port 3268
  - FreeIPA provides CLDAP, KDC and LDAP, Samba provides LSA RPC
- Stage 1: allow AD users to connect to FreeIPA services
  - e.g. PuTTY from Windows machine connecting to FreeIPA ssh service
- Stage 2: allow FreeIPA users to interactively log in into AD machines
  - Requires support for Global Catalog on FreeIPA server side
  - not implemented yet

fedora

FreeIPA v3 architecture

Full overview is available at
`http://freeipa.org/page/IPAv3_Architecture`



fedora

Kerberos cross-forest trusts

What was missing?

- Samba passdb backend to FreeIPA supporting trust storage and retrieval
- CLDAP plugin to FreeIPA to respond on AD discovery queries
- FreeIPA KDC backend to generate MS PAC and support case-insensitive searches
- Configuration tools to setup trusts

fedora

Kerberos cross-forest trusts

FreeIPA passdb backend:

- Expansion of traditional LDAP passdb backend
- New schema objects and attributes to support trusted domain information
- Support for uid/gid ranges for multi-master replicas
- Kerberos principal creation for foreign domain account

FreeIPA KDC backend:

- Generates MS PAC information out of LDAP info and add to the ticket
- Allows to accept principals and tickets from a trusted cross forest realm
- Verifies and sign MS PAC coming from a trusted cross forest realm

fedora

Kerberos cross-forest trusts

FreeIPA configuration tools:

- FreeIPA has command line (CLI) and Web user interfaces
- `ipa trust-add` creates new cross-forest trust
  - CLI operates with Kerberos authentication
  - Request is sent to FreeIPA server via XML-RPC over HTTPS with Kerberos auth
  - FreeIPA uses S4U2Proxy Kerberos feature to allow constrained delegation
  - Samba 4 Python bindings are used to establish trust
    - Code runs under non-privileged account (apache)
    - Uses Kerberos ticket obtained via XML-RPC with the help of mod_kerb_auth
    - Issues Kerberos-authenticated LSA RPC requests to a local smbd
    - Uses AD credentials or shared secret passed via XML-RPC request to talk to AD DC

fedora

Install Active Directory trusts support

- Pre-requisites: `freeipa-server-trust-ad` package
- Run `ipa-adtrust-install` as *root* to
  - define NetBIOS (short) name for the domain
  - set up Samba to use FreeIPA for searching users and storing trusted domains information
  - set up DNS service records expected by the Active Directory clients
  - set up additional plugins in 389-ds directory server
  - configure Kerberos KDC to handle MS-PAC in trusted tickets
- Caveats:
  - Check that DNS resolution works from both Active Directory and FreeIPA sides
  - If there is no common upstream DNS server, set up DNS forwarders
- After `ipa-adtrust-install` run we are ready to create AD trusts!

fedora

Create trust to Active Directory domain

Following command creates the trust

- `ipa trust-add` *`[--type=ad]`* *`ad.example.com`* `--admin` *`Administrator`* `--password`
- Explanation:
  - *`--type=ad`* — trust type (defaults to Active Directory trust)
  - *`ad.example.com`* — DNS name of Active Directory domain
  - *`--admin NAME`* — name of administrative account from Active Directory capable to create trusts
  - *`--password`* — ask for the AD admin password
- Typical issues:
  - Active Directory is relying on DNS SRV records on resource discovery.
  - Kerberos is relying on clock synchronization — more than 5 minutes skew will break things
  - `ipa-adtrust-install` suggests firewall configuration update, don't ignore it!

Section 4

Users

fedora

User management, account lockout

- As *admin*:
  - Add new user we will use for tests:
  - `ipa user-add --first=Test --last=User ruser`
  - Try editing user's attribute:
  - `ipa user-mod ruser --shell=/bin/bash`
- As *ruser*:
  - kinit 3 times with incorrect password
  - You can check `ipa user-status ruser` in the process (as *admin*)
  - Fourth kinit will be rejected
- As *admin*:
  - `ipa user-status ruser` - account is locked
- As *admin*, you have 2 ways to unlock user:
  - `ipa unlock ruser`
  - `ipa passwd user` - resets the password
- *ruser* is now able to log in

fedora

Allow access from trusted domain

- Trusted domain users authenticate with Kerberos principals *user@ad.example.com*
- These principals are mapped automatically to uids with SSSD, Winbindd, and 389-ds plugins
- Each trusted domain user receives its own primary group equal to the **uid** of the user
- To apply access controls, trust domain users and groups should belong to POSIX groups
  - First map security identifier (SID) to non-POSIX group in FreeIPA
  - Then include non-POSIX group to existing POSIX group in FreeIPA
  - Use the POSIX group in all access controls (HBAC and SUDO rules)

fedora

Allow access from trusted domain

Following sequence will map group *AD\Domain users* to POSIX group *strangers* with use of *strangers_ext* non-POSIX group:

- `ipa group-add strangers_ext --external`
- `ipa group-add-member strangers_ext --external 'AD\Domain users'`
- `ipa group-add strangers`
- `ipa group-add-member strangers --groups strangers_ext`

fedora

Allow access from trusted domain (cont.)

Additionally, every host that will provide services to trusted domain users, needs to have SSSD and `krb5.conf` configured to recognize them.

Listing 1: krb5.conf

```
[libdefaults]
....
 dns_lookup_kdc = true
 ....

[realms]
 IPA-X.EXAMPLE.COM = {
 ....
   auth_to_local = \
     RULE:[1:$1@$0](^.*@AD.EXAMPLE.COM$)s/@AD.EXAMPLE.COM/@ad.example.com/
   auth_to_local = DEFAULT
 }
```

fedora

Allow access from trusted domain (cont.)

In `sssd.conf` 'subdomains_provider = ipa' ensures that sssd will be able to look up users in trusted domains. 'services = ..., pac' ensures that user membership information from MS PAC (`http://tools.ietf.org/html/draft-brezak-win2k-krb-authz-01`) is evaluated as well.

Listing 2: sssd.conf

```
[domain/ipa-X.example.com]
...
  subdomains_provider = ipa
...
[sssd]
  services = nss, pam, ssh, pac
```

Section 5

Certificates, keytabs

fedora

Building a secured web server

Let's now build a secured web server that would accept trusted users

- Log in to **client** machine
- Make sure that IPA server is resolvable
- `ipa-client-install`
- Prepare content for `httpd`:
  - `cp workshop.conf /etc/httpd/conf.d/workshop.conf`
  - `cp workshop.wsgi /var/www/cgi-bin/workshop.wsgi`
- Create the IPA service entry for `httpd`:
  - `ipa service-add HTTP/‘hostname‘`
- Configure SSSD and Kerberos to accept trusted users

fedora

Building a secured web server: SSL certificate

- Create NSS certificate database for `httpd`
  - `mkdir /etc/httpd/alias; cd /etc/httpd/alias`
  - `certutil -N -d .`
  - `chown :apache *.db && chmod g+rw *.db`
  - `certutil -G -d .`
  - `certutil -A -d .  -n 'IPA CA' -t CT,, -a < /etc/ipa/ca.crt`

fedora

Building a secured web server: SSL certificate (cont.)

- Reqest a signed certificate for the service
  - `certutil -R -d . -a -g 2048 -s CN=‘hostname‘,O=IPA-X.EXAMPLE.COM > web.csr`
  - `ipa cert-request --principal=HTTP/‘hostname‘ web.csr`
  - `ipa cert-show $SERIAL_NO --out=web.crt`
  - `certutil -A -d . -n Server-Cert -t u,u,u -i web.crt`
  - `certutil -L -d . -n Server-Cert`
  - `certutil -V -u V -d . -n Server-Cert` - valid certificate
- Check `/etc/httpd/conf.d/nss.conf` and start `httpd`

fedora

Building a secured web server: Kerberos authentication

- Retrieve a keytab for `httpd` service
  - `ipa-getkeytab -p HTTP/‘hostname‘ -k http.keytab -s server`
  - `ipa service-show HTTP/‘hostname‘` - keytab indicator is True
  - `klist -kt http.keytab`
- Configure `httpd` to use the keytab
  - `mv http.keytab /etc/httpd/conf/`
  - `chown apache:apache /etc/httpd/conf/http.keytab`
  - `chmod 0400 /etc/httpd/conf/http.keytab`
  - Update `/etc/httpd/conf.d/workshop.conf` and enable Kerberos authentication
  - Restart `httpd`
- Test the page secured via Kerberos authentication
  - Open with web browser, OR
  - `curl -k --negotiate -u :  https://‘hostname‘`

Section 6

HBAC

fedora

Host Based Access Control - Authorization

- Check `allow_all` HBAC rule
  - Allows users to access all services in IPA realm
- Disable `allow_all` HBAC rule → *ruser* cannot log in to **client**
- As *admin*:
  - Create hostgroup `webservers`
  - Add **client** to `webservers`
  - Create HBAC rule `allow_on_client` and then:
    - `ipa hbacrule-add-host allow_on_client --hostgroups=webservers`
    - `ipa hbacrule-add-user allow_on_client --users=ruser`
    - `ipa hbacrule-add-group allow_on_client --groups=strangers`
    - `ipa hbacrule-add-service allow_on_client --hbacsvcs=sshd,login`
- Verify the rule:
  - Log in to **client** with *ruser*, *admin*
  - Check hbactest in CLI and Web UI
  - Try to log in from Windows machine using PuTTY and web interface

Section 7

RBAC

fedora

Role Based Access Control

- As *ruser*:
  - `ipa user-add --first=John --last=Doe jdoe` - fails due to missing privileges
- As *admin*:
  - `ipa role-find`
  - `ipa role-add-member 'User Administrator' --users=ruser`
- As *ruser*:
  - `ipa user-add --first=John --last=Doe jdoe` - suceeds
  - `ipa user-del jdoe`

Section 8

Replication

fedora

Create a replica

- Prepare replica info file and install replica
  - `ipa-replica-prepare`
  - `ipa-replica-install`
- Notice `ipa-replica-conncheck` is run before installation
- Check logs (`ipareplica-install`)
- Install DNS service on replica as well
  - `ipa-dns-install`
- To debug replica agreements:
  - View agreements with `ipa-replica-manage list`
  - Run raw LDAP searches:
  - `ldapsearch -h localhost -Y GSSAPI -b cn=config "(objectclass=nsds5ReplicationAgreement)"`
  - `ldapsearch -h localhost -Y GSSAPI -b cn=config "(nsDS5ReplicaId=*)"`

Section 9

Other features

- SELinux user mappings
  - per-user per-host contexts the user receives
  - requires SSSD on the client side
- Centralized management of SUDO rules
  - SUDO can enforce either directly or via the SSSD
- Centralized management of automounter maps
  - consumable directly or via the SSSD as well

Section 10

SSSD: More than a FreeIPA client

fedora

SSSD

- a client side of the IPA, but can be (and often is!) used standalone
- a system daemon that provides its own NSS and PAM modules
  - the modules just proxy requests to the SSSD
- the daemon is stateful
  - keeps track of network status, server availability
- supports several back ends
  - LDAP
  - Kerberos
  - IPA
  - Active Directory (new in 1.9.0)

fedora

New features of the SSSD in Fedora 18

- a native Active Directory provider
- caching of SUDO rules
- caching of automounter maps
- the ability to act as a client for setups with AD trusts
- notable performance improvements

fedora

The Active Directory Provider

- LDAP + Kerberos underneath
- uses the AD-specific tokenGroups attribute to optimize group lookups
- defaults tailored to match the Active Directory environments
- enrollment is a separate issue
  - provided by the realmd project
  - new in Fedora 18, very easy to use
    - `yum install realmd`
    - `realm join --user Username ad.example.com`
  - both server and desktop use case

fedora

Integration with the SUDO utility

- SSSD acts as a proxy between SUDO and LDAP
- SSSD provides a persistent cache and also smart and configurable refresh rules
- requires sudo 1.8.6p3 or newer
- IPA supports sudoers compat tree in `ou=sudoers,$BASEDN`
  - `ldapsearch -Y GSSAPI -b "ou=sudoers,$BASEDN"`
  - `sudo_provider = ldap` in sssd.conf
  - `sudoers:  files sss` in nsswitch.conf
- Try to create sudo rules in IPA, see results in sudoers compat tree

The end.

Thanks for listening.