freeIPA
identity | policy | audit

FreeIPA Client and Server

Improvements in FreeIPA 3.3

Martin Košek <mkosek@redhat.com>

2014-04-03

Focus of FreeIPA 3.x versions

- FreeIPA 3.3 introduced cross-realm Trusts with Active Directory
- Since then, several stabilization versions released with following main themes:
  - *FreeIPA 3.1*: Migration to new PKI server - Dogtag 10
  - *FreeIPA 3.2*: CA-less installation, dropped --selfsign option (*covered in other presentation*)
  - *FreeIPA 3.3*: Supporting subdomains in AD forest (*covered in other presentation*)
- This slide deck introduces FreeIPA Server and Client changes not covered in specialized presentations

Dogtag 10

Introduction

- FreeIPA integrates Dogtag PKI as a Certificate System of the choice – FreeIPA 3.0 PKI was based on *Dogtag 9*
- FreeIPA 3.1 introduced Dogtag 10 with major changes:
  - Infrastructure changes – PKI running on Tomcat 7, installers rewritten to Python, major code refactoring and cleanup
  - REST interface – planned to be leveraged by FreeIPA in the future, currently it is only used by *cert-find* command
  - CLI – *pki* command
  - New directory layout enabling future configuration of multiple PKI subsystem on FreeIPA server (CA, KRA, TKS)

Migrating from Dogtag 9 to Dogtag 10

- Dogtag 10 does not allow migration from Dogtag 9
- Thus, FreeIPA servers with PKI cannot be automatically upgraded from 3.0 to 3.1, they need to be **migrated**
- Easiest way to upgrade FreeIPA PKI servers is to follow a migration procedure. In a nutshell:
  - Install a FreeIPA 3.3 replica with CA
  - Test that replica and CA works
  - Configure FreeIPA replica as primary one and decommission the old Dogtag 9 replica

Migrating from Dogtag 9 to Dogtag 10 (2)

- FreeIPA prior to 3.3.5 needs a manual change before migration can start (Red Hat Bug #1083978):
  - Open `/etc/httpd/conf.d/ipa-pki-proxy.conf`
  - Locate section titled *matches for ee port*
  - Update `LocationMatch` and add `/ca/ee/ca/profileSubmit` `URI`:

```
<LocationMatch "^/ca/ee/ca/checkRequest|^/ca/ee/ca/getCertChain|
^/ca/ee/ca/getTokenInfo|^/ca/ee/ca/tokenAuthenticate|^/ca/ocsp|
^/ca/ee/ca/updateNumberRange|^/ca/ee/ca/getCRL|^/ca/ee/ca/profileSubmit">
```

  - Restart `httpd` service
  - Proceed with migration

Command System Changes

Dropped CSV support

- FreeIPA 3.0 supported CSV in multivalue options:

```
ipa dnsrecord-add example.com --a-rec=10.0.0.1,10.0.0.2
```

- However, CSV parsing was suboptimal and caused limitations – it was therefore removed
- Use multiple arguments or BASH expansions instead:

```
ipa dnsrecord-add example.com --a-rec=10.0.0.1 --a-rec=10.0.0.2
ipa dnsrecord-add example.com --a-rec={10.0.0.1,10.0.0.2}
```

New command – cert-find

- Utilizes new Dogtag 10 REST interface
- Searches for all FreeIPA certificates, based on given criteria passed as *cert-find* options
- Simply run *ipa cert-find* command and see the results:

```
  Serial number (hex): 0x9
  Serial number: 9
  Status: VALID
  Subject: CN=ipa.example.com,O=EXAMPLE.COM

  Serial number (hex): 0xB
  Serial number: 11
  Status: REVOKED
  Subject: CN=oldipa.example.com,O=EXAMPLE.COM
```

Kerberos flags for Services and Hosts

- Under special circumstances, admin may want to set special Kerberos flags for service principals
- FreeIPA framework now allows 2 flags to be set:
  - *OK_AS_DELEGATE*: service tickets trusted for delegation
    - AD will forward TGT only to services with this flag set
    - With the flag set, SSSD can add AD user TGT to the default Kerberos credentials cache on the FreeIPA client machine
  - REQUIRES_PRE_AUTH: pre-authentication is required
    - Can be used to disable pre-authentication for selected services or hosts (lowers the load on KDC, slightly increases possibility of a brute force attack on a long term key)

Kerberos flags for Services and Hosts (2)

- Example - adding OK_AS_DELEGATE flag for *test/ipa.example.com@EXAMPLE.COM* principal:
  - Focus on the *O* flag for this principal in klist output

```
$ ipa service-mod test/ipa.example.com –ok-as-delegate=1
$ kvno test/ipa.example.com@EXAMPLE.COM
$ klist -f
Ticket cache: KEYRING:persistent:0:0
Default principal: admin@EXAMPLE.COM

Valid starting       Expires              Service principal
02/19/2014 09:59:02  02/20/2014 08:21:33  test/ipa.example.com@EXAMPLE.COM
    Flags: FATO
...
```

Additional resources

- Upstream feature pages:
  - http://www.freeipa.org/page/V3/Drop_CSV
  - http://www.freeipa.org/page/V3/Cert_find
  - http://www.freeipa.org/page/V3/Kerberos_Flags
- Kerberos protocol tutorial:
  - http://www.kerberos.org/software/tutorial.html

Host provisioning and installation

Host provisioning – userClass attribute

- FreeIPA 3.0 did not offer a *host* attribute suitable *annotating* hosts according to their class
  - The only option was to misuse the description attribute
  - Host annotation is useful for host provisioning tools to set class of the machine in FreeIPA realm
- FreeIPA 3.3 introduces *userClass* attribute to be assigned
- The attribute can be used with Directory Server *Automatic Membership* plugin to automatically assign annotated host to hostgroups
- Hostgroups can be used in HBAC rules, SELinux user mapping rules or SUDO and thus applying the right policy for the new host according to it's class

Host provisioning – Example

- Any host attribute can be used (*userClass*, *FQDN*, ...) in the rule – the example below will use the new *userClass* attribute
- Prepare an automember rule to place all hosts with webserver class to specific host group
- If there are more than one matching rules, all are applied

```
$ ipa hostgroup-add webservers --desc "Web Servers"

$ ipa automember-add --type=hostgroup webservers

$ ipa automember-add-condition webservers --key=userclass --type=hostgroup \
    --inclusive-regex=^webserver$

$ ipa automember-show webservers --type=hostgroup
  Automember Rule: webservers
  Inclusive Regex: userclass=^webserver$

$ ipa host-add web.example.com --class webserver

$ ipa hostgroup-show webservers
  Host-group: webservers
  Description: Web Servers
  Member hosts: web.example.com
```

Host re-enrollment

- Previously installed client may be **re-enrolled**
  - Can be used to reset a system to a known state
  - Can be used after a restore from a backup/snapshot
  - Existent FreeIPA client need to be uninstalled first
- 2 options to authenticate:
  - **Host keytab**: Use `--keytab` option and pass path to backed up `/etc/krb5.keytab`
  - **Administrator credentials/OTP**: Use `--principal` and `--password` options

Host re-enrollment (2)

- Use `--force-join` option in `ipa-client-install` if the host was not properly uninstalled and host entry is still active on the server
- `--force-join` effects:
  - New **certificate** is generated, old certificate is revoked
  - New **keytab** is generated
  - Public **SSH keys** are re-uploaded on the server
- `ipaUniqueID` of the host entry stays the same

Additional resources

- Upstream feature pages:
  - http://www.freeipa.org/page/V3/Forced_client_re-enrollment
  - http://www.freeipa.org/page/V3/Integration_with_a_provisioning_systems

DNA range management

Introduction to DNA

- FreeIPA users Distributed Number Assignment (DNA) plugin to automatically manage UID/GID assignment
  - Per server configuration: *cn=Posix IDs,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config*
  - Replicated DNA plugin status: *cn=posix-ids,cn=dna,cn=ipa,cn=etc,SUFFIX*
- Plugin manages the ranges across all replicas
  - *On-deck* range actively used by the replica (*dnaNextValue*, *dnaMaxValue* in per server configuration)
  - *Next* range used when on-deck range is depleted (*dnaNextRange* in per server configuration)

Introduction to DNA (2)

- DNA makes sure there are no duplicates even when replication link is down by allocating different
  - Achieved by reserving different ranges for different replicas
  - New range assigned when a number is assigned for the first time on given replica
- When a replica is deleted, it's range was not recovered

Use cases

- Live replica is being deleted
  - Free number range can be still retrieved from the replica
  - ipa-replica-manage is capable of saving the range
- Range is depleted or lost
  - Assigned range was exhausted, there is no free range in other replicas
  - Replica may have died for any reason and was deleted
    - It's range or a range of a replica connected only to the dead replica is lost
  - Administrator will need to assign a new range

Use case: Live replica is being deleted

- Previously, it's range was simply lost
- In FreeIPA 3.3, *ipa-replica-manage del* was enhanced:
  - Connects to removed replica before deleting it
  - Makes it read only
  - Retrieves the dead ranges (*on-deck* and *next* range)
  - Tries to add the ranges as a *next range* to available FreeIPA replicas
- Useful commands:
  - *ipa-replica-manage dnarange-show*
  - *ipa-replica-manage dnanextrange-show*

Use case: Range is depleted or lost

- Range needs to be set or extended manually
- Useful commands:
  - *ipa-replica-manage dnarange-set*
  - *ipa-replica-manage dnanextrange-set*
- Be cautious when extending the range manually:
  - Make sure that FreeIPA ID range contains the extended range (check *ipa idrange-find*)
  - Make sure that no number from the recovered range was never used (audit UID/GID of existent users and groups)
    - Duplicate UID/GID could be assigned otherwise.

Additional resources

- Upstream feature page:
  - http://www.freeipa.org/page/V3/Recover_DNA_Ranges
- Man page:
  - man ipa-replica-manage


freeIPA
identity | policy | audit