FreeIPA Training Series

Centralized Management of SELinux User Mappings

Rob Crittenden
Jakub Hrozek

January 22, 2013

What are SELinux User maps?

- When a user logs in the PAM stack assigns an SELinux context.
- By default this context is unconfined_u.
- This can be controlled on each system with the file /etc/selinux/targeted/seusers
- It is more efficient to centrally manage this. More powerful too.

Defining the mapping rules

- FreeIPA configuration contains two default values:
  - Default SELinux user context
    - This may be blank. In this case sssd applies the host default.
  - Ordered list of available SELinux contexts, from most restrictive to least restrictive
- Rules are used to define the context a user will have when logging into a host consist of the following:
  - List of Users and/or Groups
  - List of Hosts and/or Hostgroups
  - Category defining all Users or all Hosts
  - Link to existing Host-Based Access Rule (HBAC)

Rules on rules

- A rule may either point to existing HBAC rule or define its members directly, but not both.
- A rule must define both the users and hosts that apply to the rule. If either is missing the rule is skipped during evaluation.

Creating a rule

- Creating a rule takes as many as 3 steps:
  - Create the rule and set the context:
    - ipa selinuxusermap-add rule1 –selinuxuser=staff_u
  - Add users and groups to the rule:

    ipa selinuxusermap-add-user rule1 –users=joe,admin
  - Add hosts and hostgroups to the rule:
    - ipa selinuxusermap-add-host rule1 –hosts=web1.example.com

Evaluating Rules

- A list of maps can be thought of as a triple:
  - (host, user, selinux context)
    - Host can be a host, hostgroup or host category of ALL
    - User can be a user, group or user category of ALL
- Matching is done from the most-specific to the least-specific.
  - Host > Hostgroup > host category ALL
  - User > Group > user category ALL
  - If two rules are equivalent then the SELinux context order defined in FreeIPA config is used, granting the least restrictive context.

Evaluating Rules, cont.

- Rules are stored in FreeIPA on the server and evaluated in the client in sssd.
- The rules are stored in persistent on-disk cache and applied even in case the FreeIPA server is not available
- No configuration changes are necessary on the client side. The FreeIPA provider of the SSSD would apply the SELinux context out of the box.

Evaluation Examples

- We have defined two rules:
  - (client.example.com, *, staff_u)
  - (*, joe, guest_u)
- If joe logs in to client.example.com he will get staff_u because hosts are evaluated first.
- If joe logs in to any other host he gets guest_u.

Evaluation Examples

- We define two rules:
  - (webservers, joe, staff_u)
  - (webservers, admins, unconfined_u)
- webservers is a hostgroup consisting of web1.example.com and web2.example.com
- joe is a member of the admins group
- If joe logs in to web2.example.com he will get staff_u. This is because the rule containing his uid is more specific than the group rule.

Example resolving using context order

- We define two rules:
  - (webservers, joe, guest_u)
  - (webservers, joe, staff_u)
- webservers is a hostgroup consisting of web1.example.com and web2.example.com
- Our default map order is: guest_u:xguest_u:staff_u:unconfined_u
- If joe logs in to web2.example.com he will get staff_u. Both rules evaluate the same, using a hostgroup and a specific user. The context is determined by the map order.

Under the covers – Server side

- Rules are stored in cn=selinux,$SUFFIX
- Sample rule in LDAP:

```
dn: ipaUniqueID=89fa03e8-3ebd-11e2-ac7d-
000c2989f613,cn=usermap,cn=selinux,dc=
 example,dc=com
objectClass: ipaassociation
objectClass: ipaselinuxusermap
ipaSELinuxUser: staff_u:s0-s0:c0.c1023
cn: rule1
ipaEnabledFlag: TRUE
ipaUniqueID: 89fa03e8-3ebd-11e2-ac7d-000c2989f613
memberUser: uid=joe,cn=users,cn=accounts,dc=example,dc=com
memberHost: cn=webservers,cn=hostgroups,cn=accounts,dc=example,dc=com
```

Under the covers - Client side

- sssd works in cooperation with pam_selinux to set the context
- sssd queries the rules from the server with each authentication. Caching is done for offline purposes only.
- On each login sssd creates the file /etc/selinux/<policy_name>/logins/<login> which contains the SELinux context to assign.