Troubleshooting ncm-useraccess

Luis Fernando Muñoz Mejías

CERN IT-DI-CSO

2010-07-21

Outline

1. Introduction
   - Replacing ncm-access_control
   - Brief overview of ncm-useraccess
2. Using ncm-useraccess
   - ncm-useraccess schema
   - Configuring the access to an account
3. Limitations and improvements
   - Current situation
4. Conclusions

ncm-access_control

Description

- An ncm-component for controlling how to access a machine
  - Valid credentials for each user
    - Kerberos tokens
    - SSH keys...
  - PAM services a user was allowed to use
  - Sudo!
- No fine grain
- Complex
- CERN-specific
  - The community needed something similar

ncm-access_control

Problems

- A component that tried to do too much
  - Difficult to use
- Bad code
  - And I mean **bad**

ncm-access_control

Replacements

- ncm-sudo
  - Nobody complains about it. :)
- ncm-useraccess
  - Still does too many things
  - Our star for the rest of the presentation

Goals of ncm-useraccess

- Specify the set of credentials that can log into a system account
- Specify the PAM service(s) a system account can log into.

Problems of ncm-useraccess

- Still, we're doing too much with it
  - But we can live with it!!

ncm-useraccess roles

Example

```
“/software/components/useraccess/roles” =

     nlist(

          “Homer”, nlist(...)
          “Kenny”, nlist(...)
          “MrBurns”, nlist(...),
          “Bart”, nlist(...),
          “Lisa”, nlist(...),
          “Patty”, nlist(...))
```

role a name assigned to a set of configuration parameters

ncm-useraccess users

“/software/components/useraccess/users/nuclear_plant” =

user configuration parameters to be written into a system account

Parameters from a role are inlined into the account's configuration as well

Example

```
“/software/components/useraccess/users/nuclear_plant/roles” =

    list(

         “Homer”,
         “Kenny”,
         “MrBurns”)
```

ncm-useraccess users

- They **must** exist in the system
  - There must be a `nuclear_plant` account on the system!!!
- The component complains otherwise
- It's not possible to tell at compilation time if a user exists on the system

ncm-useraccess ACL services

ACL service a PAM service that should have ACLs attached

- Only users listed in the ACL can use the service
  - Be sure you add root if you want root login on the system!!
- Remember that services can be stacked!!
  - The ACL will affect to any service that includes an ACL service
  - For instance, an ACL on `system-auth` will affect all the system!

- ACLs are populated in the user's configuration
- This field enables ACLs on selected PAM services

Which types of credentials will ncm-useraccess control?

- By default, all means but password are controlled by ncm-useraccess
  - SSH public key
  - Kerberos v4
  - Kerberos v5
- Empty fields means the files should be removed
- Edit the `managed_credentials` field for any users that need something different

Which pre-existing settings will have access to the account?

- Set up the roles allowed to use the account

Example

```
“/software/components/useraccess/users/root/roles” =
    list(“munoz”);
```

Which additional credentials will have access to the account?

- Set up additional `kerberos4`, `kerberos5`, `ssh_keys` or `ssh_keys_urls` settings
- If you repeat the same credentials over and over, consider grouping them under a role
  - Roles can be nested!

Which services will the account be allowed to use?

- Decide on which ACLs the user must be present

Example

```
“/software/components/useraccess/users/root/acl_services” =
    list(“sshd”);
```

- This doesn’t mean the PAM service will have any ACLs
    - If you want so, add the service to `.../useraccess/acl_services`
- For CERN use case, we usually want to restrict `sshd` and maybe `login`
    - `system-auth` is too restrictive

What it does so far

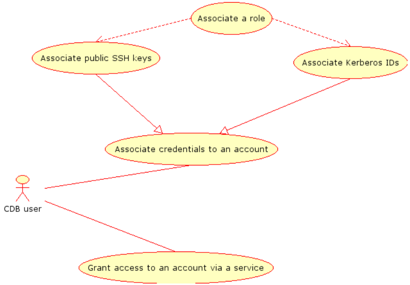
Associate a role
Associate public SSH keys
Associate Kerberos IDs
Associate credentials to an account
CDB user
Grant access to an account via a service

Figure: Use case diagram for ncm-useraccess

Solving current problems

- Ensure the users subtree contains only users that exist on the system
- Fix add_root_access()
- Don't add ACLs to system-auth anymore
  - They cause unexpected side effects
  - Maybe sshd and login?

What it may do

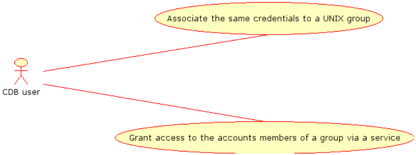
Associate the same credentials to a UNIX group
CDB user
Grant access to the accounts members of a group via a service

Figure: Extending ncm-useraccess to handle UNIX groups

Handling e-groups

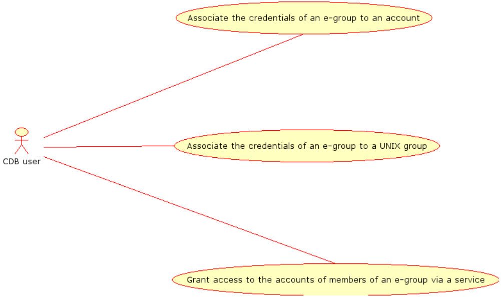
Associate the credentials of an e-group to an account
CDB user
Associate the credentials of an e-group to a UNIX group
Grant access to the accounts of members of an e-group via a service

Figure: E-group handling

- CERN-specific feature
    - Out of the scope of ncm-useraccess

Solutions for e-groups

- CERN-specific component, replacing ncm-useraccess where needed
  - Suggested by Steve Traylen
- Synchronize e-groups into ncm-useraccess roles
  - And re-run ncm-useraccess periodically
- Synchronize e-groups into ncm-useraccess settings

Conclusions

- ncm-useraccess manipulates many parts of the system
  - ~/.klogin
  - `~/.k5login`
  - `~/.ssh/authorized_keys`
  - `/etc/pam.d`
- You can select which parts it should manipulate
- Ensure your profile specifies only `users` present on the node
- `system-auth` shouldn't have PAM ACLs
- Don't forget root on your ACLs