ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEAPP03

CONVERTING FROM NIS TO RED HAT IDENTITY MANAGEMENT*

T. S. McGuckin, R. J. Slominski,
Thomas Jefferson National Accelerator Facility, Newport News, USA

Abstract

The Jefferson Lab (JLab) accelerator controls network has transitioned to a new authentication and directory service infrastructure. The new system uses the Red Hat Identity Manager (IdM) as a single integrated front-end to the Lightweight Directory Access Protocol (LDAP) and a replacement for NIS and a stand-alone Kerberos authentication service. This system allows for integration of authentication across Unix and Windows environments and across different JLab computing environments, including across firewalled networks. The decision making process, conversion steps, issues and solutions will be discussed.

INTRODUCTION

For more than a decade the JLab Accelerator Computing Environment (ACE) network relied on a combination of Network Information System (NIS) and custom scripts to manage user accounts, groups, aliases, ssh host keys and sudo rules. This was a legacy configuration used to support access across multiple, differing and aging platforms (first HP-UX, then Sun Solaris and finally Red Hat Linux).

Several security, scalability, and interoperability limitations make NIS undesirable. NIS's password hashing only pays attention to the first eight characters (further characters are ignored). Directory lookups can suffer from scalability problems with NIS because a client query to an NIS server results in the entire database being transferred to the client upon which the client must then filter out the portion of the database (file) of interest. With LDAP directory service lookups filtering is done server-side and only the portion of data requested is returned. Integrating off-the-shelf software and network devices with NIS often requires scripting whereas integration with LDAP servers is nearly ubiquitous. Finally, NIS is end-of-life and no longer actively maintained. An updated network service solution was needed.

A first step in this upgrade process was performed in 2012 when password storage and authentication was moved to a Kerberos database. Following the success of this work a more complete directory services solution was sought.

Upgrades and/or retirements of old architectures also allowed ACE to update to a more modern/secure service without having to deal with a lot of legacy/backward compatibility. Finally, a solution was needed that could integrate many of ACE's disparate services into one management interface (Fig. 1).

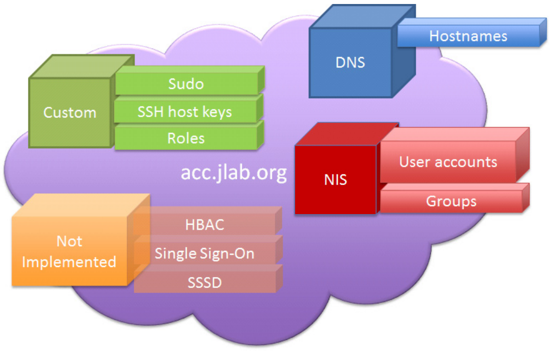


Figure 1: ACE's services spread out across multiple solutions (or not implemented) prior to upgrade.

In investigating a solution it was important to consider that JLab's ACE network (acc.jlab.org) exists as a firewalled sub-domain of the lab's larger network (jlab.org) which is maintained by the separate Computing & Networking Infrastructure (CNI) group (Fig. 2). A core requirement for the ACE network is that it be able to continue operations even if connection with the CNI network was lost. Any solution that was implemented therefore had to be able function in a stand-alone capacity and also be able to asynchronously duplicate user and host information from CNI's servers automatically.

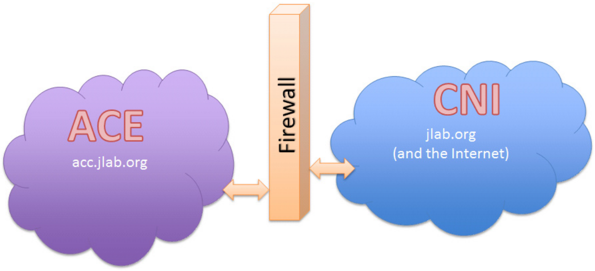


Figure 2: ACE-network relation to JLab domain.

To this end two main solutions were investigated: open source LDAP and Microsoft Active Directory. Active Directory was ultimately eliminated as an option because almost all ACE systems were Red Hat/Linux architecture with only a handful of Windows systems. Therefore, investigation began on an LDAP solution but progress was slow due to difficulties in configuration and maintenance of a bare LDAP installation. The solution that was determined to be the best fit for ACE's environment was found to be Red Hat's Identity Management software suite.

Red Hat Identity Management (IdM) is a suite of tools available as part of Red Hat Enterprise Linux (RHEL) distributions for setting up a collection of multi-master replicated clustered network domain controllers which provide services similar to Active Directory, but for all platforms, including Linux. The services provided are collectively

* This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics under contract DE-AC05-06OR23177.

named Identity Management (IdM) services, and are integrated in an easy to install and administer package with user friendly interfaces. The services include:

Kerberos - User authentication and SSO
LDAP - Authorization and directory lookup
PKI - Digital certificates and trusts
NTP - Network Time Protocol
SSH - Secure Shell host and user key management
OTP - One Time Passwords (Two-Factor) support

The goal of Red Hat Identity Manager is to provide a simple, centralized, and unified identity management system in a network domain and to minimize the administrative overhead of individually managing each component [1].

Red Hat Identity Management is the supported version of Fedora's FreeIPA project.

IMPLEMENTATION

Upgrades were done in two major phases.

In the first phase the ACE team upgraded to using Kerberos as a password and single sign-on authentication service. This provided updated password controls, including more robust password length and choice enforcement and encrypted password storage using modern hashing techniques. Single sign-on was also implemented, allowing users to be issued an authentication ticket for a pre-set length of time (defaulting to twenty-four hours) negating the need to login multiple times across common applications.

Once a successful Kerberos password solution was implemented, an investigation began of extending this model to fully replace NIS. At this point NIS was still being used for account maintenance with Kerberos added on as a password manager. Full implementation of a modern, directory service solution for user accounts and host information was the next goal. And Red Hat Identity Management was found to be the best fit for ACE's architecture.

In the second phase Red Hat Identity Management software (IdM) was implemented as a complete directory services solution. IdM server was installed on one physical and two virtual machines with built-in synchronization between the servers to ensure reliability.

IdM itself was installed using Red Hat's built in yum commands with a minimal amount of configuration required. Data was then copied to the database from our existing NIS server quite easily by harvesting existing data from NIS and piping this data through IdM's "ipa" ("Identity, Policy, Audit") command (Fig. 3).

This initial process populated the IdM database with user-data (username, UID, fullname, homedir, etc.), network groups (group name, GID, members, etc.) and host data (hostnames, architecture and ssh keys). At this point it was necessary for all users to update their password information because existing passwords were encrypted in a separate Kerberos Berkley database module (db2). The db2 architecture proved incompatible with LDAP's database architecture so attempts to transfer the encrypted passwords were unsuccessful and another solution to update everyone's password had to be found.

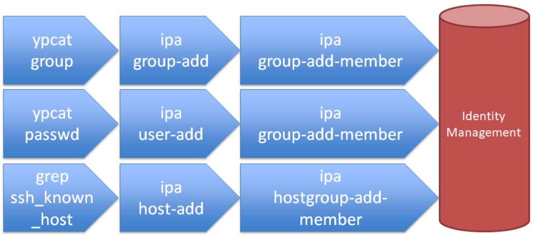


Figure 3: Initial IdM database population.

To facilitate password updates, a "user audit" was performed. During this audit all users wishing to keep their ACE account were required to login using their existing credentials and provide a new and unique password. At this point the newly provided (and encrypted) password was inserted into IdM. This process served the dual purposes of ensuring all accounts had passwords set in the new system and that only active accounts were transferred.

As the new IdM servers were brought online, shell scripts were written to automatically sync data between the IdM server on ACE's side and user and host data on CNI's side [2]. These included adding new users and hosts, disabling users locked out or removed on CNI's side and removing hosts that were deleted from CNI.

Once the scripts were sufficiently tested they were installed on all three IdM hosts. The scripts were maintained between hosts via a local git repository. Cronjobs were then implemented to run these scripts nightly and report results to the ACE team.

RESULTS

Table 1 summarizes the services that were (and in a couples instances weren't) converted to IdM. This table also summarizes services that had yet to be implemented until IdM was put into service.

Table 1: Summary of Service Changes

Service	**Initial**	**Final**
User Accounts	NIS	**IdM**
Host keys	NIS	**IdM**
Groups	NIS	**IdM**
Roles	Oracle	**IdM**
HBAC	N/A	**IdM**
Single Sign-On	N/A	**IdM**
SSSD	N/A	**IdM**
Sudo	Custom	Custom
DNS	DNS	DNS

IdM allowed the ACE team to not only continue using the features of Kerberos (including single sign-on), but to utilize additional upgrades to user account control, groups, host control, and host-based access control (HBAC). IdM also added a convenient suite of command-line tools (largely wrappers for more convoluted LDAP commands) and an intuitive, web-based graphical interface.

User account controls largely focused on maintaining and expanding on the gains seen from Kerberos. Having an industry standard LDAP directory server has allowed us to integrate with other network device servers easily, including CNI's. Part of this integration involved using Keycloak, a Red Hat sponsored open source project to extend Single Sign On from hosts in the domain to all web based applications via the Open ID Connect OAuth protocol. The additional functionality of being able to asynchronously copy user information from CNI allowed access to web resources (such as logbooks and test plan information) without requiring the creation and maintenance of a separate ACE account for every user. This was one of the most widely visible (and welcome) upgrades to users.

Similar to user controls, user groups were migrated and expanded upon in IdM. Because of the ease of updating IdM additional groups were added as needed to facilitate secure user work. Some examples of this include adding "super-groups" to existing groups, giving some members elevated privileges or dividing an existing group into more finely defined sub-groups with different privileges within a single organization.

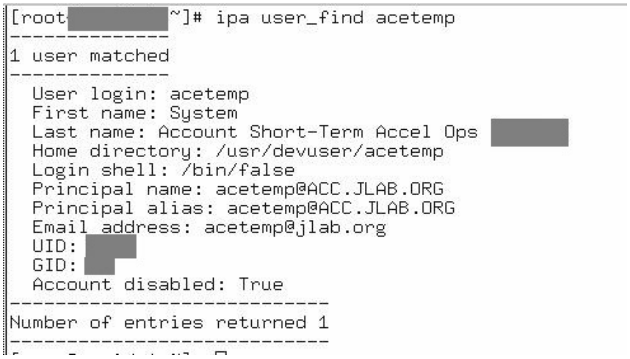
In addition, group-level control in IdM allowed for varying password policy by group. This allowed the ACE team to apply more stringent password controls on groups that were deemed higher-risk from a security perspective without changing the standard password policy on a user-by-user basis.

Host control functions of secure shell (SSH) maintenance were migrated into IdM and largely automated. This allowed for automatic update and distribution of host ssh keys to all systems when new hosts were brought online or an existing host's ssh keys were changed. Previously any changes required manual updates of a master ssh key that then had to be synchronized with CNI's ssh key file. That list would then have to be distributed to every computer system on the network via scripts that were slow and cumbersome.

Host-based access control (HBAC) defines which users (or user groups) can access specified hosts (or host groups) by using specified services (or services in a service group). This allowed the ACE group to limit which machines a given user or group can access and to give privileged users wider access [3].

System Security Services Daemon (SSSD) is a service that provides access and authentication resources, including cached/offline support for client hosts. This provides the host client with access to identity and authentication remote services provided by IdM and removes the need for users to provide login credentials (username & password) multiple across services within the ACE domain.

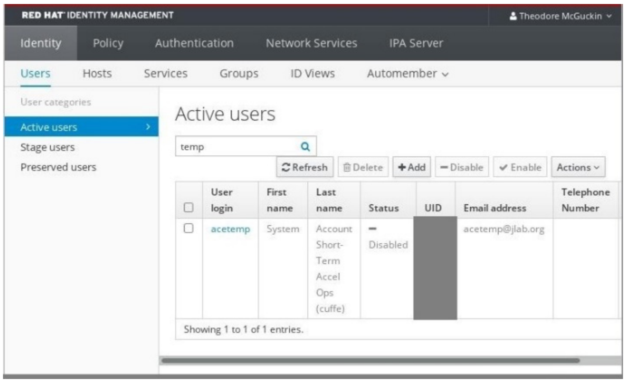
All of these additions of services and functionality could have easily increased the amount of work required by the ACE team to implement and maintain without an effective interface. IdM provided a useful set of command-line tools [4] available from any user/system with sufficient privileges via the "ipa" command (Fig. 4). This allows functionality similar to the "ypcat" or "getent" commands, but querying against the IdM database.



```
[root[redacted]~]# ipa user_find acetemp
--------------
1 user matched
--------------
  User login: acetemp
  First name: System
  Last name: Account Short-Term Accel Ops [redacted]
  Home directory: /usr/devuser/acetemp
  Login shell: /bin/false
  Principal name: acetemp@ACC.JLAB.ORG
  Principal alias: acetemp@ACC.JLAB.ORG
  Email address: acetemp@jlab.org
  UID: [redacted]
  GID: [redacted]
  Account disabled: True
----------------------------
Number of entries returned 1
----------------------------
```

Figure 4: Sample output from IdM's ipa command.

In addition to command-line tools, IdM also provides a robust web-based graphical-user interface (GUI) with admin-level access [5]. From this web interface searches are possible on any data type, such as Users, Groups, Policy, etc. (Fig. 5). By selecting a given record an admin can then investigate or update any fields associated with that record (Fig. 6).



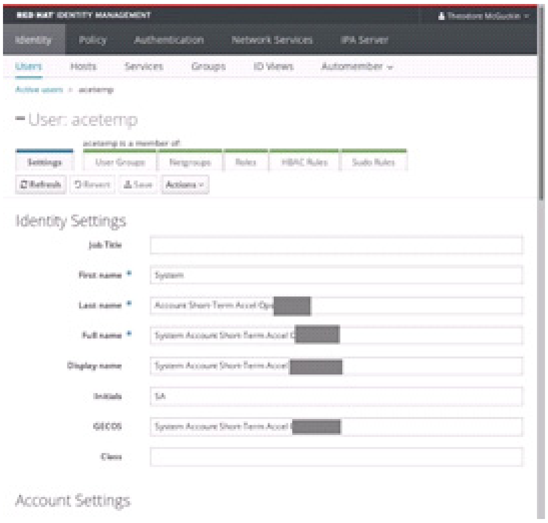
Active users

temp

□	User login	First name	Last name	Status	UID	Email address	Telephone Number
□	acetemp	System	Account Short-Term Accel Ops (cuffe)	− Disabled	[redacted]	acetemp@jlab.org	

Showing 1 to 1 of 1 entries.

Figure 5: Sample IdM search page.



User: acetemp

Identity Settings

Job Title
First name * System
Last name * Account Short-Term Accel Op[redacted]
Full name * System Account Short-Term Accel [redacted]
Display name System Account Short-Term Accel [redacted]
Initials SA
GECOS System Account Short-Term Accel [redacted]
Class

Account Settings

Figure 6: Sample IdM user page.

This polished interface, or simplified command-line structure, allows for simple and fast access to all the data of the underlying LDAP database without having to know or input complex LDAP query/update commands.

FUTURE CAPABILITY

While ACE has implemented many of the features available as part of IdM, there is still more potential to explore in future updates. Future upgrade possibilities include adding more comprehensive user roles, SELinux integration, sudo migration, adding an external git repository for sharing scripts and utilities and expanding HBAC.

Roles applied to users that can pre-define levels of access on different clusters of machines would provide a fast and easy way to configure both internal groups that need similar permissions/access and visiting groups that need to have their access limited to small suites of machines and/or services.

While SELinux is being implemented on some systems within the ACE network, full integration with IdM has not occurred yet. Once configured, IdM will be able to map users to configured SELinux roles on a per-host basis. This is accomplished by having SSSD query the user's access privileges from IdM before the user process is launched.

Currently ACE's sudo rules (used to map users to elevated/privileged commands on designated host groups) are maintained in an NIS-style file that must be distributed to all relevant hosts whenever changes are made. IdM has the capability to fully control and maintain sudo rules across all hosts automatically. The future plan is to leverage this feature and migrate ACE's existing sudo rules into IdM.

ACE wrote several custom scripts and also modified several of IdM's built in scripts to facilitate migration and maintenance of our IdM database. A future plan is to upload these scripts to Jlab's github account [2] and make them publicly available for other organizations to download and modify to their needs. Before these scripts are made available however they must have any sensitive information removed and must undergo an internal review process.

Finally, at present Host-based access control is only being applied to specific groups to allow additional privileges. A future upgrade would be expand HBAC to apply to services as well in order to limit or allow privileges on a wider basis. An example of this would be to limit the use of ssh between sub-domains within the ACE network and then allow for privileged access across domains to specified groups.

CONCLUSION

Upgrading from deprecated directory services like NIS can provide enhanced security, reliability and integrate services into a more manageable interface for your organization. Leveraging advances in encryption, authentication and authorization utilities can greatly increase your network's security. Advances in parallel operation and data synchronization built into modern Directory Services can ensure reliable performance. Additionally, integrating services under a reliable suite of management software can make all of this easier for your organization to maintain.

But the fact remains that working with interfaces like LDAP or Active Directory can be a very cumbersome process with a steep learning curve requiring a large investment of time and resources. This initial investment can result in many organizations opting to remain on older, less secure and efficient services.

The key to using a Directory Service effectively is finding the right software tools for your organization. After an extensive analysis period JLab's ACE group found the right solution for our environment was upgrading to IdM which provided the tools needed to modernize these services.

REFERENCES

[1] T. Scherf, "*Red Hat Identity Management Overview,*" `https://www.RedHat.com/archives/rh-community-de-berlin/2012-November/pdfOlwXB8dm7U.pdf`

[2] JLab Red Hat IdM Scripts Git repository, `https://github.com/JeffersonLab/RedHatIdentityManagementScripts`

[3] Configuring Host-Based Access Control In An IdM Domain, `https://access.RedHat.com/documentation/en-us/red_hat_enterprise_linux/7/html/linux_domain_identity_authentication_and_policy_guide/hbac-configure-domain/`

[4] Red Hat Identity Management Command Line Tools, `https://access.RedHat.com/documentation/en-us/red_hat_enterprise_linux/7/html/linux_domain_identity_authentication_and_policy_guide/managing-idm-cli/`

[5] Red Hat Identity Management Web UI, `https://access.RedHat.com/documentation/en-us/red_hat_enterprise_linux/7/html/linux_domain_identity_authentication_and_policy_guide/using-the-ui`