CS 447/647

User Management

Goals

What is a user?

How is account information found?

How do you add, modify and delete users?

What are some common authentication backends?

What is the role of /etc/passwd, /etc/group and /etc/shadow?

What are the parts of an encrypted password in /etc/shadow?

User Management

- Modern Systems
  - Physical
  - Virtual
  - Cloud
- Account Models
  - Classic?
  - Network-based (>1990’s)
- Security

Account Mechanics

- What is a user?
  - A UID, unsigned 32 bit integer
  - Companion GID, also an unsigned 32 bit integer
- System API for users
  - `getpwuid` (man 3 getpwuid), get password file entry
  - `getpwnam` (man 3 getpwnam), get password file entry by username
  - Traditionally uses /etc/passwd
- How does the OS find users?
  - Modern systems use Name Service Switch (/etc/nsswitch.conf)
    - ordered, /etc/passwd & /etc/group first
    - `getent passwd $USER`

Pluggable Authentication Modules (PAM)

- Provides an interface to authentication
  - Login utility calls the pam library
  - Iterates over a stack composed of modules
- Configuration in /etc/pam.d/*

Example:

```
module-type control-flag module-path [arguments]
```

pam_unix.so - /etc/passwd file

- List of local users
- Traditionally contained passwords
  - Now in /etc/shadow

File Format:

```
Login name
Encrypted password placeholder (x = /etc/shadow)
User ID
Group ID
GECOS - Full name, office, phone number
Home directory
Login Shell  (/bin/bash)
```

/etc/passwd - Login Name

- Must be unique
- Limited to 32 characters
- Lowercase
  - Case-sensitive
  - Email RFC5321 - everything before the @ should be case sensitive.
- Easy to remember
  - Traditionally initials, IE: rms (Richard M Stallman)
- Generating usernames often creates duplicates
  - newellz2
- Should be the same across machines
  - More difficult than you think, IE: eadmin
  - Often requires orchestration. Ansible

/etc/passwd - Encrypted Password

- Originally DES
  - Cracked with brute-force in 1998
- MD5 for while
- Currently salted SHA-512
- Changing algorithm does not update existing passwords
  - `chage -d 0 $USER`
- Algorithm in `/etc/pam.d/common-password`
  - password [success=1 default=ignore] pam_unix.so **obscure sha512**
  - `md5, bigcrypt, sha256, blowfish`
  - `rounds=n`
  - `obscure` - palindrome, similar, case, simple, rotated
  - `man 8 pam_unix`
- Moved to /etc/shadow


MARIO
058700
×20
WORLD
1-4
TIME
200
THANK YOU MARIO!
BUT OUR PRINCESS IS IN
ANOTHER CASTLE!

Requirements

System	Default requirements	Where set
Red Hat CentOS	8+ characters, complexity enforced	**/etc/login.defs** **/etc/security/pwquality.conf** **/etc/pam.d/system-auth**
Debian Ubuntu	6+ characters, complexity enforced	**/etc/login.defs** **/etc/pam.d/common-password**
FreeBSD	No constraints	**/etc/login.conf**

/etc/passwd - Encrypted Password

testuser:**$**6**$**YaimArFO**$**Irky4U6vstXuRs3vm.:

1 - MD5

2 - Blowfish

5 - SHA-256

6 - SHA-512

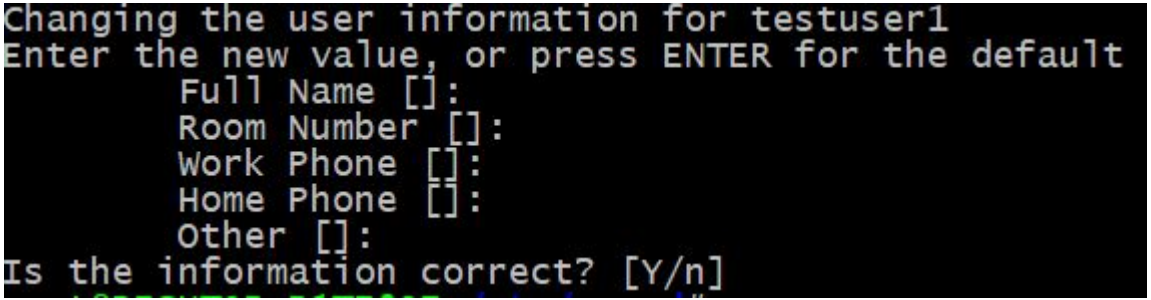
/etc/passwd - UID

- UID 0 for root
- System users < 1000
  - /bin/false shell
- Real users >= 1000
  - We used 5000+ for orchestration accounts
  - 3,000,000+ for AD users
- Do not recycle UIDs
  - Backups
- Should be globally unique
  - Assign ranges to groups, IE CSE = 100,000 - 200,000
  - Directory Server - LDAP, AD, FreeIPA (All LDAP)

/etc/passwd - GID

- GID 0 for root
- System < 1000
- No consistencies across OS or distros
  - bin
    - GID1 on Redhat\CentOS
    - GID2 on Debian\Ubuntu
    - GID7 on BSD
- Often used for accounting and access control
  - files, quotas, SLURM partitions, CPU resources
- Group information stored in /etc/group

/etc/passwd - GECOS



- GEneral Comprehensive Operating System
  - 1962 GE Operating System
  - Printing
- Stored as comma separated value
  - chfn - allows users to modify
- User’s can modify
  - chfn
  - usermod -c “Zach Newell SEM245B” newellz2

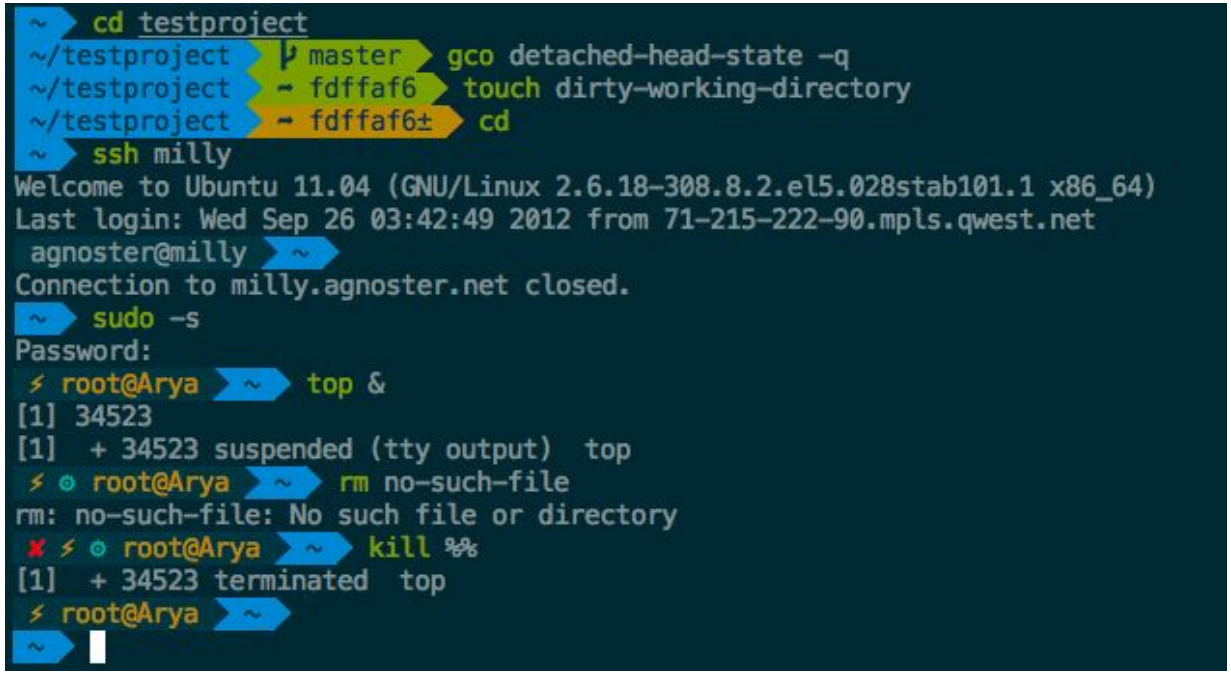
/etc/passwd - Home directory

- Default directory at login
  - echo $HOME
- Stores dotfiles
  - .bashrc
  - .bash_profile
  - .ssh/authorized_keys, .ssh/config
  - .xfce4 or .gnome
- Created as part of adduser
  - Network users home directory not automatically created
    - Solved with pam_mkhomedir.so
- Large organizations often use Network File System (NFS) home directories
  - automount can mount home directories at login.

/etc/passwd - Login shell

- Often an interpreter
  - bash - Bourne Again SHell
    - Most common
  - sh - Shell
    - Widely support and simplest
  - ksh - Korn Shell
    - “a standard/restricted command and programming language”
  - zsh
    - Newest shell
    - Fancy - https://ohmyz.sh/
- Changed with chsh
  - /bin/false - do nothing, unsuccessfully
  - /bin/nologin - Stops login and displays “This account is currently not available.”

zsh



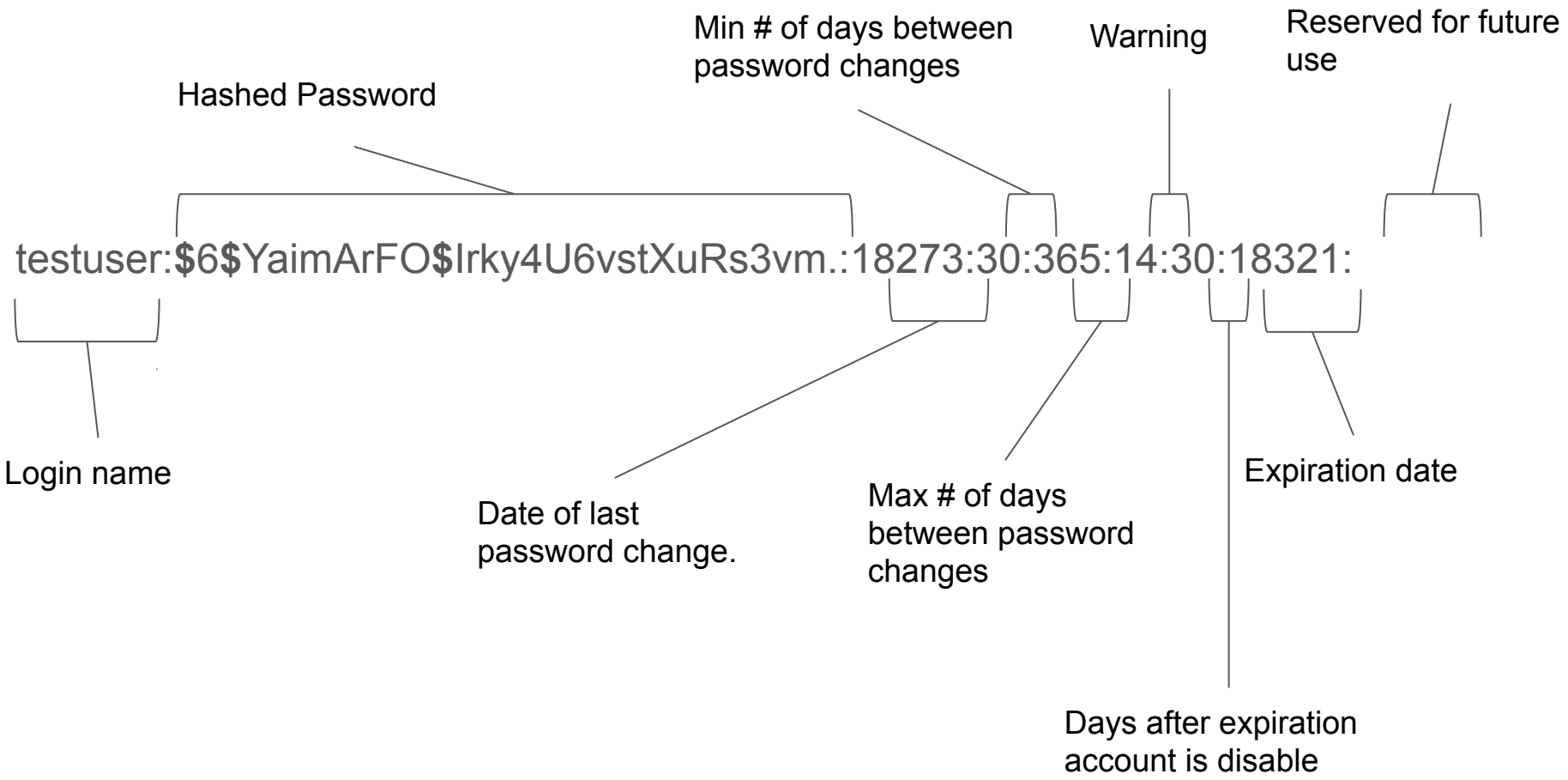
https://github.com/agnoster/agnoster-zsh-theme

/etc/shadow

- Readable only by root
- Stores encrypted passwords
  - “x” /etc/passwd entries

9 fields:

1. Login name
2. Encrypted password
3. Date of last password change
4. Minimum number of days between password changes
5. Maximum number of days between password changes
6. Number of days in advance to warn users about password expiration
7. Days after password expiration that account is disabled
8. Account expiration date
9. A field reserved for future use which is currently always empty


Min # of days between password changes
Warning
Reserved for future use
Hashed Password
testuser:6YaimArFO$Irky4U6vstXuRs3vm.:18273:30:365:14:30:18321:
Login name
Date of last password change.
Max # of days between password changes
Expiration date
Days after expiration account is disable

Managing password age and expiry

```
NAME
    chage - change user password expiry information
SYNOPSIS
    chage [options] LOGIN
DESCRIPTION
    The chage command changes the number of days between password
changes and the date of the last password change. This information
is used by the system to determine when a user must change his/her
password.
```

```
chage \ #
-m 30 \          #Min days before password change
-M 365 \         #Max days for valid password
-W 14 \          #Warn 14 days before
-I 30 \          #Inactivity disable in days
-E 2020-02-29 \ #Expiry date
testuser
```

/etc/groups

- Stores UNIX groups
- Can be password protected
  - gpasswd
  - newgrp - User enters a group with a password
- adduser creates a private group
  - adduser $USER $GROUP
- Default Privileges
  - Audio - sound
  - Serial Ports - dialout
  - Display - vga
  - Sudo - wheel or sudo

/etc/groups

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,zachn
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
dialout:x:20:zachn
```

1. Group name
2. Encrypted password or a placeholder
3. GID number
4. List of members, separated by commas (be careful not to add spaces)

Groups

- sudo
  - /etc/sudoers gives sudo permissions
- dialout
  - /dev/ttyS{0..10} permissions. Serial ports.
- audio
  - Audio permissions
- lpadmin
  - Printer permissions
- libvirtd
  - virt-manager permissions

Adding a group

```
addgroup [options] [--gid ID] group

addgroup --gid 5000 engr-admins

adduser someuser engr-admins #Debian\Ubuntu

usermod -G dialout -a newellz2sa #RHEL\CentOS


groupadd, groupmod, and groupdel #Universal
```

Adding users

1. Define account
   a. Add /etc/passwd entry
2. Add private group
   a. /etc/group
3. Set password
4. Create home directory
5. `chown` home directory
6. Configure roles and permissions

Adding users

```
adduser --home /home/someuser --shell --uid 6000 \
--gecos "Some User" --gid 6000 someuser

passwd someuser #interactive change

echo “someuser:${PW}” | chpasswd # Batch mode password change

adduser someuser somegroup

chage -d 0 testuser1 #force password change at next login

echo “umask 077” >> ~someuser/.bash_profile # 700 permissions
```

umask

- /etc/login.defs
  - UMASK 022
- libpam_mkhomedir.so
  - Essential for network users.

```
grep mkhomedir /etc/pam.d/*
common-session:session    optional   pam_mkhomedir.so umask=077
```

```
#skel=/path/to/skel/directory
#Indicate an alternative skel directory to override the default /etc/skel.
```

Adding users

Target	Filename	Typical uses
all shells	**.login_conf**	Sets user-specific login defaults (FreeBSD)
sh	**.profile**	Sets search path, terminal type, and environment
bash[a]	**.bashrc**	Sets the terminal type (if needed) Sets **biff** and **mesg** switches
	.bash_profile	Sets up environment variables Sets command aliases Sets the search path Sets the **umask** value to control permissions Sets CDPATH for filename searches Sets the PS1 (prompt) and HISTCONTROL variables
csh/tcsh	**.login**	Read by "login" instances of **csh**
	.cshrc	Read by all instances of **csh**
vi/vim	**.vimrc/.viminfo**	Sets **vi/vim** editor options
emacs	**.emacs**	Sets **emacs** editor options and key bindings
git	**.gitconfig**	Sets user, editor, color, and alias options for Git
GNOME	**.gconf**	GNOME user configuration via **gconf**
	.gconfpath	Path for additional user configuration via **gconf**
KDE	**.kde/**	Directory of configuration files

a. **bash** also reads **.profile** or **/etc/profile** in emulation of **sh**. The **.bash_profile** file is read by login shells, and the **.bashrc** file is read by interactive, non-login shells.

/etc/skel

```
ls -lha /etc/skel
```

```
total 12K
drwxr-xr-x 1 root root  512 Jan 12 08:49 .
drwxr-xr-x 1 root root  512 Jan 12 09:17 ..
-rw-r--r-- 1 root root  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 root root 3.7K Apr  4  2018 .bashrc
-rw-r--r-- 1 root root 2.2K May 31  2017 .kshrc
-rw-r--r-- 1 root root  807 Apr  4  2018 .profile
```

Disabling and removing accounts

```
usermod -L someuser # Lock

usermod -U someuser # Unlock

deluser someuser    # Delete user
```

Centralized Account Management

- rsync /etc/passwd, /etc/group, /etc/shadow
  - Ansible
- NIS - Networking Information Server
  - Network access based authentication and authorization
- LDAP
  - Lightweight Directory Access Protocol
- LDAP+KRB5 (Kerberos)
  - SSO
  - Ticket-based

Ansible

```
- name: Create cadmin user
  tags:
    - server-auth

  user: name=cadmin state=present
        uid=8000
        shell=/bin/bash
        groups="sudo"
        home='/usr/local/home/eadmin'
        password='$6$9dcUsl0p$siuhsd.ffewg3dss8
```

Centralized Account Management - NIS

- Network Information Service (NIS)
- Originally developed by Sun Microsystems in the 1980’s
- Used to be named Yellow Pages
  - Renamed due to trademark
  - Services retain the yp*
    - ypbind - finds the NIS master
    - ypserv - Primary NIS master service
- Exports
  - Groups
  - Users
  - Hostnames

Setting up NIS

```
time apt install nis #Wait forever

nano /etc/default/nis #master

nano /etc/yp.conf #client

systemctl restart nis #wait…

cd /var/yp && make

ypcat passwd

zcat /usr/share/doc/nis/nis.debian.howto.gz | less #2003!
```

Centralized Account Management - LDAP

- Lightweight Directory Access Protocol
  - Small to large organizations
- Enforces unique UIDs and GIDs
- Largely replaced NIS
- Mixed *nix and Windows infrastructure means AD
  - Kerberos
  - NIS additions not default
- `apt install -y ldap-utils`

Attribute	Stands for	What it is
o	Organization	Often identifies a site's top-level entry[a]
ou	Organizational unit	A logical subdivision, e.g., "marketing"
cn	Common name	The most natural name to represent the entry
dc	Domain component	Used at sites that model their hierarchy on DNS
objectClass	Object class	Schema to which this entry's attributes conform

a. Typically not used by sites that model their LDAP hierarchy on DNS

Searching LDAP

```
ldapsearch \
-h cs447.cse.unr.edu \
-p 389 -x -b \
"dc=cs447,dc=cse,dc=unr,dc=edu" \
"(uid=rhyolite)"
```

```
# rhyolite, users, cs447.cse.unr.edu
dn: uid=rhyolite,ou=users,dc=cs447,dc=cse,dc=unr,dc=edu
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
cn: rhyolite
sn: Amargosa
uid: rhyolite
uidNumber: 1000000
gidNumber: 1000000
homeDirectory: /home/rhyolite
loginShell: /bin/bash
gecos: Rhyolite Amargosa
```

Searching UNR LDAP

```
ldapsearch \
-W \
-h unrdc4.unr.edu \
-p 389 -x -D "CN=Zachary A
Newell,ou=Employees,ou=Users,ou=IT,dc=unr,dc=edu" \
-b "dc=unr,dc=edu" \
"sAMAccountName=newellz2"
```

```
CN=$NAME,ou=$FLETTER,ou=NetID,ou=IT,dc=unr,dc=edu
```

Adding a user the hard way...

ldapadd -x -W -D "cn=admin,dc=cs447,dc=cse,dc=unr,dc=edu" -f rhyolite.ldif

```
dn: uid=rhyolite,ou=users,dc=cs447,dc=cse,dc=unr,dc=edu
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
cn: rhyolite
sn: Amargosa
uid: rhyolite
uidNumber: 1000000
gidNumber: 1000000
homeDirectory: /home/rhyolite
loginShell: /bin/bash
gecos: Rhyolite Amargosa
userPassword: {SSHA}xAOqDys+VX3WbjsF6C0xZ6lomLNAIRyJ
```

Centralized Account Management - SSO

- **Shibboleth** - Open-source
  - UNR uses Shib
- JOSSO - SSO Server
- CAS - Central Authentication Service

Centralized Account Management - IAM

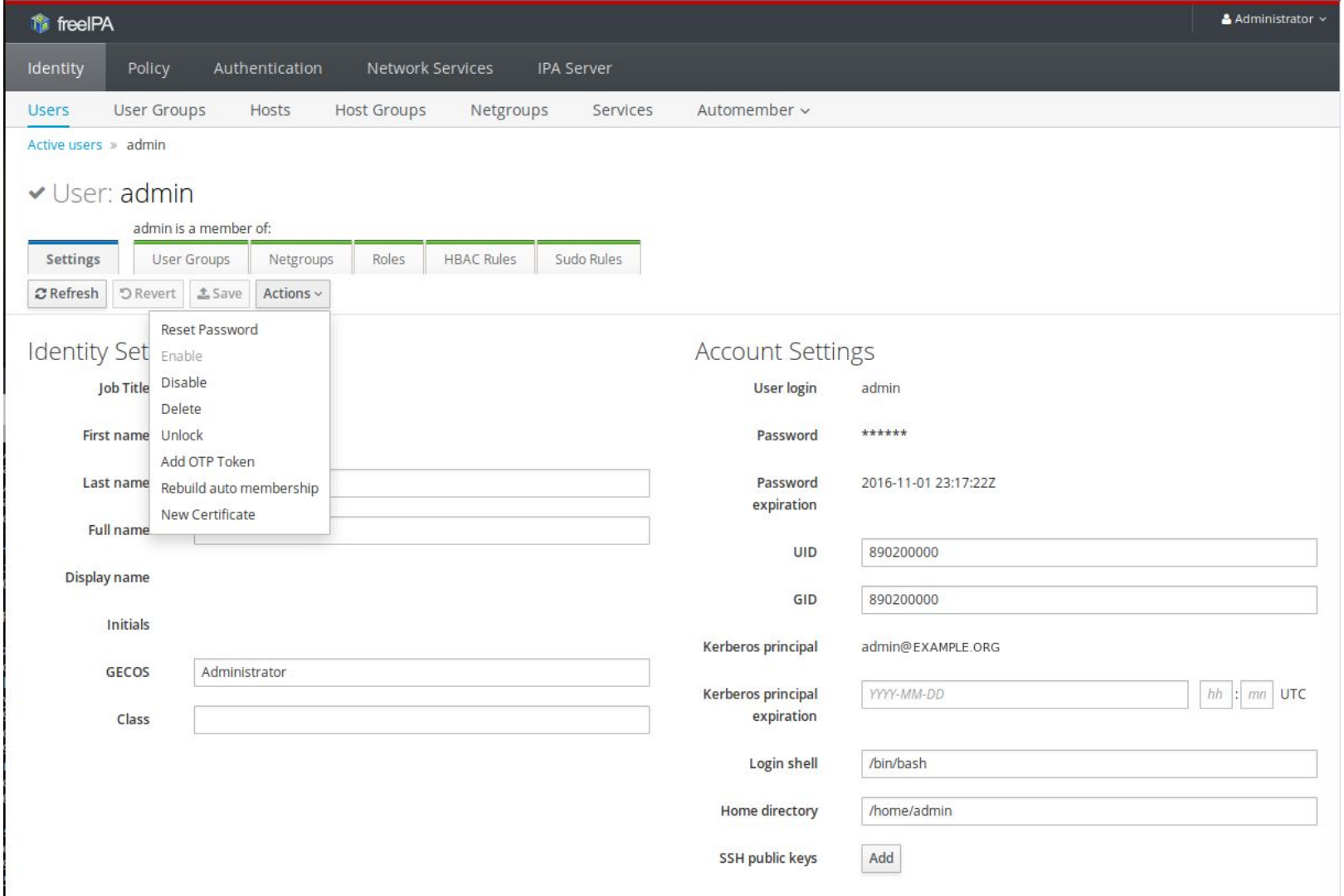
- Identity Management Systems\Identity Access Management
  - Authentication
  - Granting Privileges
- Largely Commercial
  - Microsoft
  - Oracle
  - Redhat
- Web-based Management Interface
- Role-based provisioning
- *Offboarding*
  - Removing users from groups and access.

Centralized Account Management - IAM

- Account Management
  - Unique login names, UIDs and GIDs
  - CRUD across an organization
    - Regardless of OS
  - Approval workflows
  - Reporting
  - Logging of all administrative actions
- Ease of use
  - Users can update their own information and password
  - Global password changes
  - https://security.unr.edu/

Centralized Account Management - FreeIPA

- Open-source version of Redhat IdM
- Manages
  - Users
  - Groups
  - Machine Accounts
  - SSO
- Uses
  - LDAP - Directory Server 389
  - Kerberos - MIT Kerberos 5
  - WWW - Python-based Web Application
  - Certificates - dogtag
  - DNS - bind9
  - Sudo


freeIPA
Administrator
Identity
Policy
Authentication
Network Services
IPA Server
Users
User Groups
Hosts
Host Groups
Netgroups
Services
Automember
Active users » admin
User: admin
admin is a member of:
Settings
User Groups
Netgroups
Roles
HBAC Rules
Sudo Rules
Refresh
Revert
Save
Actions
Reset Password
Enable
Disable
Delete
Unlock
Add OTP Token
Rebuild auto membership
New Certificate
Identity Set
Job Title
First name
Last name
Full name
Display name
Initials
GECOS
Administrator
Class
Account Settings
User login
admin
Password
******
Password expiration
2016-11-01 23:17:22Z
UID
890200000
GID
890200000
Kerberos principal
admin@EXAMPLE.ORG
Kerberos principal expiration
YYYY-MM-DD
hh
mn
UTC
Login shell
/bin/bash
Home directory
/home/admin
SSH public keys
Add