

User Management

- System security depends on account management
- User's ID UID is a 32 bit number.
- */etc/passwd* – list of authorized users in a world readable file
- Contains pro-forma entry (x most often, * as well) where user passwd used to be.
- */etc/shadow* – contains encrypted password in the file that is not world readable.

/etc/passwd

- Login name
 - Unique, lowercase, start with letter, <= 32 characters long;
 - No colons or newlines
 - Need a clever scheme to avoid duplicates
- Password holder, hashed string in */etc/shadow*
 - DES was easy to crack.
 - MD5 had a weakness.
 - Salted SHA-512 is now the standard
 - Passwd standard on Ubuntu – 6+ chars, complexity enforced
 - Use system tools to add users, do not edit passwd file!!!!
- UID
 - Root has UID of 0.
 - UID determines who is superuser, not account name.
 - You can have account *bobby* with UID 0 who is a superuser, and account *root* with UID 1006 who is not a superuser.
 - Pseudo-users bin, daemon with low UID.
 - Real user ID start with 1000 or higher.
 - Useradd config file specifies the range.
 - Two accounts with the same UID are the same user (even if names and passwds are different) – BAD IDEA!
 - Can kill each other's processes and read and delete each other's files
 - Do not recycle user ids. (restored backups????)
 - Should be consistent across all systems in the organization.
- Default GID
 - 32 bit int.
 - GID of 0 is for root, system or wheel (on some systems).
 - Groupid are used to share file access

- `/etc/group` defines groups
- `/etc/passwd` provides default GID at login
- Optional GECOS info: name, office, number
 - Relic from the UNIX past
 - General Electric Comprehensive Operating System services
 - NO well-defined syntax
 - Users can change their info with **chfn**
 - Good idea to disable **chfn**!!!!
- Home directory
 - Login default
 - Shell aliases, environmental variables
- Login shell
 - Command interpreter
 - Default in Linux is bash, FreeBSD and others use sh
 - Sysadmin can change user's shell in `/etc/passwd` file.

`/etc/shadow`

- Readable by superuser only.
- Default on all systems.
- Has additional info not provided in **`/etc/passwd`** file.
- Not a superset of the **`/etc/passwd`** file.
- Use tools to maintain it! Do not add users by adding entry to the file by hand.
- Contains fields:
 - Login name.
 - Same as in **`/etc/passwd`** file.
 - Encrypted password (86 chars, SHA-512, starts with \$6\$).
 - Same as was in previous **`/etc/passwd`** file.
 - Date of last passwd change.
 - Filled by **passwd** command.
 - Min # days between password changes.
 - 0 is recommended.
 - Max # days between passwd changes.
 - This number + grace period.
 - Number of days in advance to warn users of passwd expiration.
 - **login** will warn users
 - Days after expiration for account to be disabled (grace period).
 - Account expiration date.
 - Admin has to reset before user can login.

- Can use **usermod** to set expiration field in the format *yyyy-mm-dd*.
 - Reserved field for future use.
- Required: username and password.
- Date fields are specified in days (not seconds) since Jan 1, 1970.

Reconciling differences between passwd and shadow:

pwconv

/etc/group file

- Lists groups and their members.
 - Group name.
 - 8 char long in most systems
 - Encrypted passwd placeholder (usually an asterisk).
 - GID number.
 - List of members, separated with commas.
 - Group membership is granted at login file, so /etc/passwd has the last word if user is not listed in a specific group in /etc/group.
 - GID should start at 1000.
 - User can be in unlimited number of groups.
 - **useradd** and **adduser** default to putting each user to his own group.
 - Linux provides **groupadd**, **groupmod**, and **groupdel** commands.

Changes Made by Hand

- **Error prone!**
- User agreement and policy statement.
- Do this before creating an account.
- If doing manual changes to passwd or shadow files, use **vipw**, and **vigr** if making changes to group file.
- This invokes editor in your EDITOR env variable, not necessarily a vi.

Setting a passwd

Automated systems is a security risk, as usernames can be looked up in passwd file.

Other (non-root and non-user) accounts on the system

- Low-UID users are pseudo-users.
- UIDs under 10 are system accounts.
- UIDs under 100 are associated with some installed software.
- Bin and Daemon are assigned OS files and processes that do not need to be owned by root – avoids danger associated with root ownership.
- Advantage of pseudo-groups and pseudo-users:
 - Can be used more safely than the root account to provide access to resources(“mysql” owns database-related resources)
 - “nobody” pseudo-user is used by NSF for remote roots – remote roots are mapped to this account not to have root powers on the local system. Owns no files.

Commands and Config Files for User Management

- **useradd, usermod, userdel** live in /etc/login.defs, /etc/default/useradd
- **adduser, deluser** live in /etc/adduser.conf, /etc/deluser.conf
- useradd disables new account by default.
- To see all current values:

\$ sudo useradd -D

Can you add users in bulk ?

- Yes, but...
- **newusers** creates accounts from a text file with usernames similar to /etc/passwd, but with passwd in clear text?!?!?
- Does not copy all start up files.

Safe Account Removal

- Make backups first
- Use **userdel** or **rmuser**
- Checklist to ensure account and everything related was removed:
 - Remove from local db and phonelist
 - Mail databases
 - Remove any pending jobs
 - Kill any running processes
 - Remove user from **passwd, shadow, gpasswd** and **gshadow** files
 - Remove home dir
 - Remove mail spool
 - Find orphaned files with: **sudo find filesystem -xdev -nouser**

- Ubuntu's **deluser** is a Perl script that calls **userdel** and undoes everything **adduser** does. Lives in `/etc/deluser.conf`

Lockout

- Used to temporarily disable.
- Can put a * in front of the user passwd in shadow. OR:
`usermod -L user` (puts ! instead of * described above)
`usermod -U user` (unlocks account)
- Some commands still function: ssh.
- BETTER: replace shell with program that prints message and terminates the session.

Centralized Management

- Large organizations benefit from centralized account management.
- Use LDAP (Lightweight Directory Access Protocol)
 - Generalized, database-like repository that stored user management data.
 - Hierarchical client-server model
 - Supports multiple servers and multiple clients.
 - Enforces unique UIDs and GIDs across systems.
 - Gets along with Windows.
- Microsoft's Active Directory wants to be the boss in mixed systems.
- Use Linux/Unix LDAP databases as secondary servers.
- The Lightweight Directory Access Protocol is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol network (Wikipedia)
- Active Directory (AD) is a Microsoft technology used to manage computers and other devices on a network. It is a primary feature of Windows Server, an operating system that runs both local and Internet-based servers.
- Active Directory is a directory services product and LDAP is a protocol which provides directory and indexing services. AD function is based on LDAP protocol.

PAM – Pluggable Authentication Modules

- Programs often use GUI logins without passwords, but with alternative methods of authentication: biometrics, network identity system, two-factor authentication.
- PAM – wrapper for different method-specific authentication libraries.
- Authentication framework
- Is an authentication technology, not access control technology
- Does not check what user is authorized to do on the system
 - Admins chose methods of authentication + context
 - Program that require authentication call PAM system
 - PAM calls authentication library specified by the sysadmin
- File `/etc/pam.d/common-password` contains password policies and hashing algorithm to use.
- Changing hashing algorithm does not update existing user passwords, need to manually update passwords before new algorithm takes effect:
`change -d 0 username`
- Password standards are set in `/etc/login/defs` and `/etc/pam.d/common-password`

Can you id hashing algorithm?

SHA512 starts with \$6\$, 86 chars
SHA256 starts with \$5\$, 40 chars
MD5 starts with \$1\$ or \$md5\$, 34 chars
Blowfish \$2\$, 32-448 chars
DES \$xx\$, where xx are 2 salt chars, 13 chars

Kerberos

- Deals with authentication, not access control.
- Authentication method
- PAM is the wrapper, and Kerberos is the implementation
- Uses 3rd party server to perform authentication for the entire network.
- Users do not authenticate themselves to a machine, but provide credentials to Kerberos service
- Kerberos issues cryptographic credentials that users present to other services as evidence of their identity
- Standard in Windows and part of MS Active Directory system.

Filesystem ACL

- Generalization of the traditional user/group/world permission model.
- Allows permissions to be set for multiple users and groups at once.

Useful Commands

1. **who** utility displays a list of users logged in into the local system. Remote users logged into the local system have a machine name in parenthesis.

```
$ who
sam      tty4      2019-08-25    17:18
max      tty1      2019-08-25    16:22
zach     tty1      2019-08-25    16:39
max      pts/4     2019-08-25    17:27        (dragon)
```

If you need to know which terminal you are on, or what time you logged in:

```
$ who am i
max      pts/4     2019-08-25    17:27        (dragon)
```

2. **finger** utility displays similar info + full name + idle time + if messages are disabled (*):

```
$finger
Login      Name      TTY      Idle Login time  ...
max        Max Wild  *tty2    5      Jul  25 16:42
sam        Sam Smith tty4     29     Jul  25 17:33
```

- You can use finger to get information about one specific user:

```
$finger max
Login: max          Name: Max Wild
Directory: /home/max Shell: /bin/bash
On since Wed Jul 25 16:42 (PDT) on tty2 (messages off)
On since Wed Jul 25 16:42 (PDT) on pts/4 from dragon
  3 minutes 7 seconds idle
New mail received Wed Jul 25 17:16 2018 (PDT)
  Unread since Wed Jul 25 16:44 2018 (PDT)
Plan:
  I will be at the conference in Hawaii next week.
```

- Can also find user's username.

```
$finger HELEN
Login: hls          Name: Helen Simpson
```

- Info is collected from system files. .plan was written by max in .plan file in his home directory.
- **Finger can be a security risk!!!!** Why do you think it may be a security risk?
- Some systems disable it.

3. System load and duration info: **uptime**

```
$ uptime
09:48:13 up 2 days, 23:14, 3 users, load average: 0.00,
0.01, 0.05
```

4. **w** Lists users on the system

- Displays same info as above, plus a list of users currently logged n.
- Useful if you want to communicate with someone locally.

5. Communicate with other users: **write** utility.

- Both users have to execute
write username [terminal]
\$ write max
Hi Max, what is the answer to question 5?
- Press Ctrl-D to exit.
- You can block messages by typing: **mesg n**
Or enable by: **mesg y**