

Unix Security

Cunsheng Ding
HKUST, Hong Kong, CHINA
cding@cs.ust.hk

Agenda

- A short history of Unix
- Login and user accounts
- access control
- Instances of general security principles
- Audit configuration and management

A Brief History

- Originated in 1969 and early 70's as a prototype in Bell Labs (part of AT&T).
- In 1973 Unix was rewritten in C and successfully ported.
- AT&T freely gave away Unix in source to many universities, most notably to UC Berkeley.
- 1993 first release of Unix-like OS, called Linux.

What is Unix

- Multi-user, multi-process operating system.
- Hierarchical file system.
- Consistent byte-oriented access to files and devices.

Login and User Account

Login

- identification + authentication: = (username, password)
- password length: 8 characters
- password protection: encrypted with `Crypt(3)`, and stored in `/etc/passwd` file in earlier versions (I can read my entry in the password file) and in a secure location `/.secure/etc/passwd` in recent versions of Unix

Format of the Password File

- Format: Username: encrypted password: user ID: Group ID: ID string: home directory: login shell
- ID string = user's full name
- User ID and group ID = explained later.
- Login shell: the Unix shell available to the user after successful login.

Format of the Password File ctd.

- Displaying the password file: `cat /etc/passwd`

```
dieter:RT.QsZEEsxT92:100026:53:Dieter  
Gollman:/home/staff/dieter:usr/local/bin/bash
```

- When the password field is empty, the user does not need a password for login.
- If the password field starts with an asterisk, the user cannot login, because such values cannot be the results of $F(\text{cleartext password})$. Account disable

Other Issues

- Passwd(1): change password by supplying old one twice
- Shadow password file: in security-conscious versions of Unix, it is stored in `/.secure/etc/passwd`
- Expiry date and control of old password: set
- Root login: can be restricted to terminals nominated in `/etc/ttys`

Users and Superusers

- Users by *user name*, up to 8 characters
- Users by *user ID* (UID) internally, a 16-bit number
- UIDs are linked to user names in `/etc/passwd`.
- Unix does not distinguish between users having the same UID.

Special User IDs

- **Superuser** has UID 0, and the name **root**.
 - The **root** account is used by the operating system for essential tasks like login, recording the audit log, or access to I/O devices.
- | | | |
|---|----|--------|
| • | -2 | nobody |
| • | 0 | root |
| • | 1 | daemon |
| • | 2 | uucp |
| • | 3 | bin |
| • | 4 | games |
| • | 9 | audit |

Nobody account is for NFS (network file system) anonymous connections and configuring anonymous FTP

Special User IDs

- Almost all security checks are turned off for the superuser.
- The root account performs also certain administrative tasks.
- The systems manager should not use root as his personal account.
- When necessary, changing to root can be requested by typing `/bin/su` without specifying a user name.

Superuser and Protections

- Remark: The superuser can do almost everything.
- Remark: Every precaution has to be taken to control access to superuser status.
- Question: How?



Control of Access to Superuser Status

- The files `/etc/passwd` and `/etc/group` have to be write-protected. [UID => 0 in `/etc/passwd`]
- Record all `su` attempts in the audit log together with the user (account) who issued the command.
- Separate the duties of the system manager, e.g., by having special users like `uucp` or `daemon` to deal with networking. If one of these special users is compromised, not all is lost.

Groups

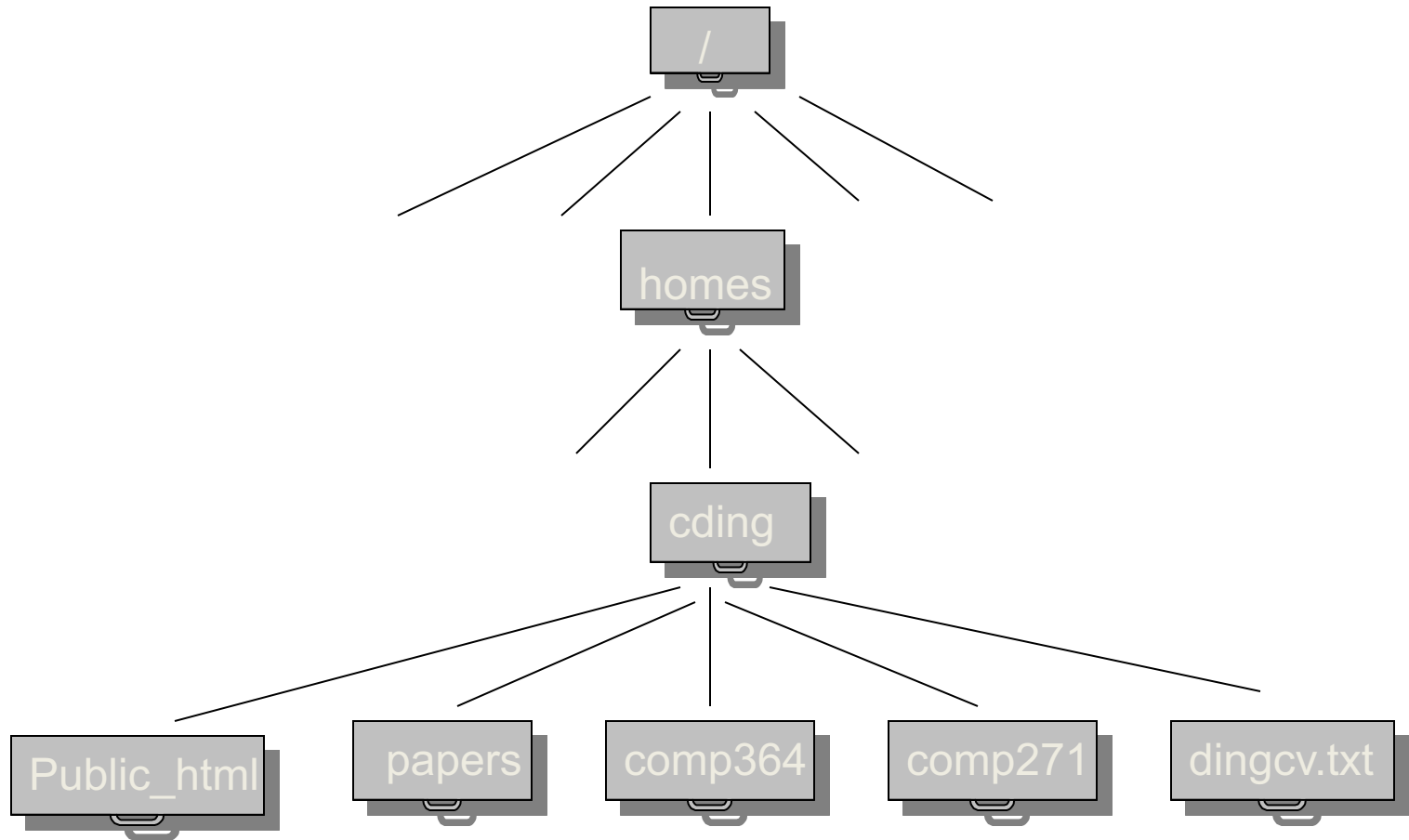
- Fact: Users belong to one or more groups.
- Why? Collecting users in groups is a convenient basis for access control decisions.
- Example: put all users allowed to access email in a group called mail.
- Primary group: contains every user. The *group ID (GID)* of the primary group is stored in `/etc/passwd`.

Set UserID and Set GroupID

- Question: If you can only read `/etc/passwd` or do not have any access right to `/.secure/etc/passwd`, how can you change your password?
- Answer: *controlled invocation*,
Set UserID Program (SUID).
- Remark: *temporarily* take on the UID of the owner of the password file. (i.e., root)

Access Control

Tree Structure for files, directories and devices



Access Control: Unix File Structure

- Each directory contains ["ls -a" gives all]
 - a pointer to itself, the file '.'
 - a pointer to its parent directory, the file '..'
- Each file
 - has an owner, usually the user who created the file;
 - belongs to a group (its owner's or directory's group).
- A newly created file belongs either to its creator's group or its directory's group.

Access Control: Unix File Structure

- Each file entry in the directory is a pointer to a data structure, inode.
 - use "ls -l" to find
- Fields in the inode that are relevant to access control.

FIELDS in inode relevant to security

- mode : type of file access rights
- uid : user who owns the file
- gid : group which owns file
- mtime: modification time
- block count: size of file

Fields in inode (part 1)

Inspect a directory with command `ls -l`

```
-rw-r--r-- 1 dieter staff 1617 Oct 28 11:01 d.tex  
drwx----- 2 dieter staff 512  Oct 25 17:44 ads/
```

- The 1st character gives the type of file. '-' a file, 'd' a directory.
- The next nine characters give the *file permission* (to be discussed later).

Fields in inode (part 2)

```
-rw-r--r-- 1 dieter staff 1617 Oct 28 11:01 d.tex  
drwx----- 2 dieter staff 512 Oct 25 17:44 ads/
```

- The following numerical field is the *link counter*, counting the number of links (pointers) to the file.
- The next two fields are the name of the owner and the group of the file.

Fields in inode (part 3)

```
-rw-r--r-- 1 dieter staff 1617 Oct 28 11:01 d.tex  
drwx----- 2 dieter staff 512 Oct 25 17:44 ads/
```

- The next integer is the size of the file in bytes.
- The date and time is mtime, the time of the last modification.
- The last entry is the name of the file. The '/' after ads indicates a directory.

Fields in inode: File Permissions

```
-rw-r--r-- 1 dieter staff 1617 Oct 28 11:01 d.tex  
drwx----- 2 dieter staff 512 Oct 25 17:44 ads/
```

- The permission bits are grouped in three triples that define read, write and execute access for owner, group, and other.
- '-' indicates no grant of right.
- The uid, gid tell who own the file.

Changing Permissions with chmod by *owner* or *superuser* only

- Absolute mode
 - chmod [-R] absolute file
 - specify the value for all permission bits
- Symbolic mode
 - will not introduced here. For details, see, Dieter Gollmann, Computer Security, Wiley, 1999. [page 91]

Changing Permissions with chmod in *Absolute Mode*

- The file permissions are specified directly by an octal number.
- **Example: 6=110 4=100, 7=111**
 - chmod 644 = 110100100 = rw-r--r--
 - chmod 777 = 111111111 = rwxrwxrwx
 - chmod 755 = 111101101 = rwxr-xr-x
- The option -R applies the specified change recursively to all subdirectories of the current directory.

Default Permissions (1)

- Unix utilities (e.g., editors or compilers):
 - 666 when creating a new file
 - 777 when creating a new program
- Adjust the permissions by `umask`, specifying the rights that should be withheld.
 - `umask 777` denies every access
 - `umask 000` does not add any further restriction.

Default Permissions (2)

Sensitive Default Settings

- 022 *all for owner, r and x for group and other.* [for programs]
- 077 *all for owner, no for group and other.*
- umask value is in /etc/profile
- actual default permission is computed as:
default ^ umask = 666 ^ 077 = 600
 $A \wedge B = A \text{ and } [\text{not}(B)] \quad \text{AND NOT}$

Instances of General Security Principles

Deleting Files (1)

- Question: If we remove (delete) a file from the file system, does it still exist in some form?
- Remark: We have to talk about how a file was constructed!

Deleting Files (2)

- Two types of copying: cp, link and ln
- cp: identical but independent file owned by the user running cp.
- link, ln: only create a new file name with a pointer to the original file and increase the *link counter* of the original file.

Deleting Files (3)

- Conclusion: If a new file shares its content with the original, and if the original is deleted with `rm` or `rmdir`, it disappears from its parent directory, but its contents as well as its copy still exist.
- Question: How do we get rid of a file?
 - The super user runs `ncheck` to list all the links to that file and then deletes those links.

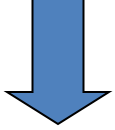
Deleting Files (4)

- Conclusion: Once a file has been deleted, the memory space allocated to this file becomes available again. However, until these memory locations have actually been used again, they will still contain the file's contents.
- Question: How do we get rid of a file?
 - Overwrite its contents with all-zeros before deleting it.

Protection of Devices

- Information: Unix treats devices like files. Thus access to memory or a printer can be controlled like access to a file through setting permission bits.
- How to create devices: use the `mknod` command which should only be executable by root.

Memory Device Must be Protected

- An attacker can bypass the controls set on files and on directories, if they can access to the memory devices holding these files. 
- If the read or write permission bits are set on a memory device, an attacker can **browse** through memory or **modify** data in memory without being affected by the permissions defined for the files stored in this memory.
- Conclusion: Almost all devices should NOT be readable or write-able by other.

Auditing and Administration

Audit Logs & Intrusion Detection

- Auditing: records security relevant events in an *audit log (audit trail)* for later analysis.
- Intrusion detection: detects suspicious events when they happen and inform the system manager by email or by messages sent to the operator console.
- Comment: The audit log should be well protected from writing by an attacker.

Protecting the Audit Log

- Set a “logical protection” on the audit log so that only privileged users have write access.
- Sent the audit log to another computer where root on the audited machine has no superuser privilege.
- Sent the audit log to a secure printer.