05 - Unix Security

2024年8月26日 13:39

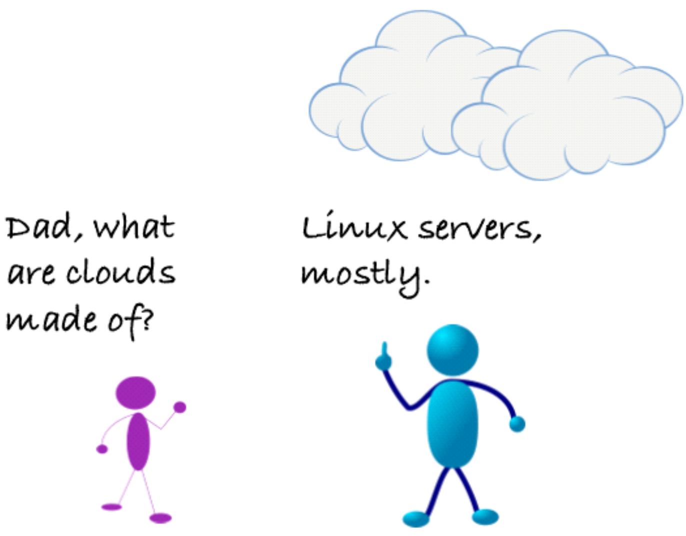
05 - Unix Security

Australian National University

Unix Security

COMP2700 Cyber Security Foundations

Slides prepared based on Chapter 7 of Gollmann's "Comptur Security", 3rd edition.



Australian National University

Objectives

- Understand the security features provided by a typical operating system. 典型操作系统
- Introduce the basic Unix security model.
- See how general security principles are implemented in an actual operating system.
- This is not a crash 速成 course on Unix security administration.

Australian National University

Outline

- Unix security – background
- Principals, subjects, objects
- Access rules
- Security patterns
  - Controlled invocation (SUID programs)
  - Securing memory and devices
  - Importing data
  - Finding resources
- Managing Unix security

Overview of Unix

- Unix was developed for friendly environments like research labs or universities.
- Security mechanisms were quite weak and elementary; improved gradually.
- Several flavours of Unix; vendor versions differ in the way some security controls are managed & enforced.
  - Commands and filenames used in this lecture are indicative of typical use but may differ from actual systems.

Overview of Unix

- Unix designed originally for small multi-user computers in a network environment; later scaled up to commercial servers and down to PCs.
- Linux and Mac OS X are perhaps the most well-known modern Unix-like operating system.
- But lesser known, though more pervasively used, examples of Unix-like systems are (the core) of Android and iOS, running in billions of devices.

Australian National University

Unix Design Philosophy

- Security managed by skilled administrator, not by user. Focus on:
  – protecting users from each other.
  – protecting against attacks from the network.

自主访问控制 粒度

- Discretionary access control with a granularity of **owner**, **group**, **other**.

供应商

- Vendor-specific solutions for managing large system and user-administered PCs.

安全 用户管理的

- "Secure" versions of Unix: Trusted Unix or Secure Unix often indicates support for multi-level security.
  – E.g., Security-Enhanced Linux (SELinux) supports multi-level security.

- **厂商特定的解决方案**：
  - 为了管理**大型系统和用户管理的** PC，不同厂商提供了专门的 Unix 版本。
- **"安全"版本的 Unix**：
  - **受信任的 Unix（Trusted Unix）或安全 Unix（Secure Unix）**通常支持**多级安全**（Multi-level Security，MLS）。
  - 例如：SELinux（Security-Enhanced Linux）支持多级安全控制，可以根据敏感度级别进行更精细的访问控制。

Australian National University

Principals

- Principals: **user identifiers** (UIDs) and **group identifiers** (GIDs).
- A UID (GID) is a 16-bit number; examples:

  0: root
  1: bin
  2: daemon
  8: mail
  9: news
  1001: alice

- UID values differ from system to system
- Superuser (**root**) UID is always zero.

Australian National University

User Accounts

主体（用户和组）

- Information about principals is stored in user accounts and *home directories*.
- User accounts stored in the `/etc/passwd` file

```
$ cat /etc/passwd
```

- User account format:

username:password:UID:GID:name:homedir:shell

- username：用户的登录名。
- password：存储用户密码的哈希值（通常存储在 /etc/shadow 文件中）。
- UID：用户标识符。
- GID：组标识符。
- name：用户的全名（可选）。
- homedir：用户的主目录路径。
- shell：用户的默认 shell 程序（例如 /bin/bash）。

UID 和 GID 是 Unix 系统用于管理权限的核心机制，确保用户和组之间的隔离和访问控制。

Australian National University

User Accounts Details

- Username: up to eight characters long
- Password: password hash (in older versions of Unix); in modern Unix the password hash is stored elsewhere.
- User ID: user identifier for access control
- Group ID: user’s primary group
- ID string: user’s full name
- home directory
- Login shell: program started after successful log in

Australian National University

Examples

From the lab VM:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
admin2700:x:1000:1000:Ubuntu,,,:/home/admin2700:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
alice:x:1001:1001:Alice,,,:/home/alice:/bin/bash
bob:x:1002:1002:Bob,,,:/home/bob:/bin/bash
charlie:x:1003:1003:Charlie,,,:/home/charlie:/bin/bash
dennis:x:1004:1004:Dennis,,,:/home/dennis:/bin/bash
eve:x:1005:1005:Eve,,,:/home/eve:/bin/bash
felix:x:1006:1006:Fong,,,:/home/fong:/bin/bash
```

Australian National University

Superuser

- The **superuser** is a special privileged principal with **UID 0** and usually the username **root**.
- 几乎没有限制
  There are few restrictions on the superuser:
  - All security checks are turned off for superuser.
  - The superuser can become any other user.
  - The superuser can change the system clock.
- Superuser cannot write to a read-only file system but can remount it as writeable.
  重装
- Superuser cannot decrypt passwords but can reset them.
  解密

Australian National University

Groups

- Users belong to one or more groups.
- /etc/group contains all groups; file entry format:

  groupname:password:GID:list of users
- Every user belongs to a primary group (主组); group ID (GID) of the primary group stored in **/etc/passwd**.
- Collecting users in groups is a convenient basis for access control decisions. 访问控制决策
  - For example, put all users allowed to access email in a group called **mail** or put all operators in a group **operator**

Australian National University

Examples

显示信息、

搜索文本的命令

From the lab VM: groups where user bob belongs to

通过以下命令可以查看用户 bob 所属的所有组

```
$ cat /etc/group | grep bob
bob:x:1002:
tutors:x:1007:alice,bob,charlie
```

用来查找所有包含字符串 bob 的行

组ID

- tutors:x:1007:alice,bob,charlie：表示 bob 还属于 tutors 组，该组的 GID 为 1007，成员包括 alice、bob 和 charlie。

Australian National University

Examples

Some commands to display user id and groups:

```
$ whoami
alice
```

显示当前登录用户的用户名

```
$ id
uid=1001(alice) gid=1001(alice)
groups=1001(alice),6(disk),1007(tutors)
```

显示当前用户的 UID、GID **以及所属组**

表示用户 alice 属于 alice、disk 和 tutors 组。

```
$ groups
alice disk tutors
```

列出当前用户所属的所有组

Australian National University

替代用户

substitute user do

Sudo-ers

- In some linux distributions (such as Ubuntu), one cannot login as the root user directly.
- Instead, a special group, called ‘sudo’, is created, such that its members are allowed to become ‘root’ using the ‘sudo’ command.
- Example:

```
$ sudo whoami
root

$ grep sudo /etc/group
sudo:x:27:admin2700
```

default. → root.

Australian National University

Subjects

- The subjects in Unix are processes; a process has a process ID (PID).
- New processes generated with **exec** or **fork**.
- Processes have a real UID/GID and an effective UID/GID.
- Real UID/GID: inherited from the parent; typically UID/GID of the user logged in.
- Effective UID/GID: inherited from the parent process or from the file being executed.

核心系统调用

Australian National University

Examples

The ps command can be used to query information about processes.

For example, to display PID, real user and effective user of all processes running in the system:

```
$ ps -eo pid,ruser,euser,command
```

用于查询系统中运行的进程信息

Example of (selected) output:

```
 PID RUSER     RUID EUSER     EUID COMMAND
2818 alice     1001 alice     1001 bash
3150 alice     1001 root         0 passwd
```

Australian National University

Passwords

- Users are identified by username and authenticated by password.
- In legacy Unix systems, passwords stored in `/etc/passwd` hashed with the algorithm crypt(3).
- crypt(3) is really a one-way function: slightly modified DES algorithm repeated 25 times with all-zero block as start value and the password as key.
- Salting: password encrypted together with a 12-bit random salt that is stored in the clear.

DESData Encryption Standard对称加密算法

将密码与一个全零块一起加密 25 **次；防止反向破解**

Australian National University

Passwords

- When the password field for a user is empty, the user does not need a password to log in.
- To disable a user account, let the password field starts with an asterisk; applying the one-way function to a password can never result in an asterisk.
- `/etc/passwd` is world-readable as many programs require data from user accounts.
- Shadow password files: hashed passwords are not stored in `/etc/passwd` but in a shadow file **/etc/shadow** that can only be accessed by root.

通过将 **密码字段设置为星号（*）** 来禁用账户。由于密码经过单向哈希加密，因此**哈希函数的结果不可能为星号**，这使得禁用后的账户无法使用密码登录

shadow 文件专门用于存储 **哈希后的密码**，而不是 /etc/passwd 文件中明文存储

Australian National University

Shadow password file

- Shadow password file location: /etc/shadow
- Also used for password aging and automatic account locking; file entries have nine fields:
  - username
  - user password
  - days since password was changed
  - days left before user may change password
  - days left before user is forced to change password
  - days to "change password" warning
  - days left before password is disabled
  - days since the account has been disabled
  - reserved

/etc/shadow 文件的作用：

- 除了存储密码哈希之外，还用于**密码过期管理** 和 **账户自动锁定**

Australian National University

Objects

- Files, directories, memory devices, I/O devices are uniformly treated as **resources**.
- These resources are the objects of access control.
- Resources organized in a tree-structured file system.
- Each file entry in a directory is a pointer to a data structure called **inode**.

这些资源即为**访问控制的对象**，受系统权限管理机制的保护

每个目录中的文件条目是指向 inode（**索引节点**）的指针

Inode

Unix 系统中的每个文件都有一个 inode，用于存储文件的**元数据**（而不是文件的实际数据内容）。通过 inode，系统能够管理和控制文件的访问权限、文件属性等信息

Fields in the inode relevant for access control

mode	type of file and access rights rwx
uid	username of the owner
gid	owner group
atime	access time
mtime	modification time
itime	inode alteration time
block count	size of file
	physical location

Examples

stat 是 Unix/Linux 系统中用于**显示文件或目录的 inode 信息**的命令

The command stat displays the inode information of a file, e.g.,

```
alice@comp2700-lab:~$ stat /etc/passwd
  File: /etc/passwd
  Size: 2034        Blocks: 8          IO Block: 4096   regular
file
Device: 811h/2065d Inode: 8043        Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid:
(    0/    root)
Access: 2021-08-16 05:52:56.121875300 +0000
Modify: 2021-07-25 11:51:47.543481900 +0000
Change: 2021-07-25 11:51:47.583482399 +0000
 Birth: -
```

You can also use ls command to show the inode number:

```
alice@comp2700_lab:~$ ls -il /etc/passwd
8043 -rw-r--r-- 1 root root 2034 Jul 25 11:51 /etc/passwd
```

使用 ls -il 命令显示文件的 inode 信息

Australian National University

Information about Objects

- Example: directory listing with **ls -l**

```
-rwxr-x--- 1 alice alice 4807960 Aug 12 10:34 lab1.pdf
drwxr-xr-x 2 alice staff    4096 Aug 15 10:33 lectures
```

- File type: first character 7 kinds.

```
- file
d directory                    s socket
b block device file            l symbolic link
c character device file        p FIFO
```

first in first out.

- File permissions: next nine characters
- Link counter:
  - the number of links (i.e. directory entries pointing to) the file

显示硬链接数量，即指向同一文件 inode 的目录条目数量

Australian National University

Information about Objects

```
-rwxr-x--- 1 alice alice 4807960 Aug 17 10:34 lab1.pdf
drwxr-xr-x 2 alice tutor    4096 Aug 17 10:33 lectures
```

- Username of the owner: usually the user that has created the file.
- Group: depending on the version of Unix, a newly created file belongs to its creator's group or to its directory's group.
- File size, modification time, filename.
- Owner and root can change permissions (**chmod**); root can change file owner and group (**chown**).
- Filename stored in the directory, not in inode.

Australian National University

File and Directory Permissions

- File permissions are internally represented by a sequence of bits, consisting of 4 groups of 3-bits.
- The first group represents *special modes* (to be discussed later).
- The next three groups define read, write, and execute access for owner, group, and other.

Australian National University

Special modes

- The first group of three bits represents special modes.
- The first bit is also called the SUID bit.
- The second bit is called the SGID bit.
- And the third is called the sticky bit.
- The SUID and SGID bits are used to implement controlled invocation (to be discussed later).
- These bits are rarely used – most files will have these bits set to 0.

SUID 和 SGID：

- 用于**受控调用**（controlled invocation），即当程序由普通用户执行时，以文件拥有者或组的权限运行。
- 常用于需要**临时提升权限**的程序，如 /usr/bin/passwd，用户可以修改自己的密码，而无需 root 权限

Australian National University

Special modes

- The sticky bit is used for different purposes in different implementations.
- In some legacy Unix systems, it is used to indicate a program file should be ‘cached’ in swap space.
- In Linux, a sticky bit on a directory means that a user may not delete files owned by other users.
  - This is usually used in a world-writeable directory, such as /tmp
  - Every user can create files/directories in /tmp, but they cannot delete files/directories created by other users.

在目录上的使用：

- 当 Sticky Bit 应用于**目录**（如 /tmp）时，意味着：
  - 任何用户都可以在目录中**创建文件。**
  - 但是，**只有文件的所有者**（或 root 用户）可以删除文件，即使其他用户对目录有写权限。
- 这对于公共临时目录（如 /tmp）非常有用，防止用户意外删除其他用户的数据

示例操作

1. **查看文件权限**：ls -l /path/to/file

输出可能显示特殊模式，如 rwsr-xr-x（表示 SUID 被设置

2.**设置 SUID**：sudo chmod u+s /path/to/program

3.**设置 SGID**：sudo chmod g+s /path/to/program

4.**设置 Sticky Bit**：sudo chmod +t /path/to/directory

5.**验证 Sticky Bit**：

ls -ld /tmp

输出：drwxrwxrwt 10 root root 4096 Nov 9 12:34 /tmp

t 表示 Sticky Bit 已被设置。

Australian National University

File and Directory Permissions

The three bits in the second, third and fourth groups are interpreted as follows: when the bit is set (i.e., its value is 1), its interpretation is as follows:

- First bit: read access granted
- Second bit: write access granted
- Third bit: execute access granted.

Example:

000 111 101 101

- 000 → No special modes
- 111 → Owner read,write, exec
- 101 → Group read, exec
- 101 → Other read, exec

Australian National University

Textual representation of permissions

- Permission bits are commonly displayed using a textual notation that is easier to understand.
- When the first group is 000 (i.e., no special modes), the remaining groups are represented textually as follows: if a bit in the group is 0, it's represented by '-'. Otherwise, depending on the position of the bit:
  - First bit: represented by 'r' (read)
  - Send bit: represented by 'w' (write)
  - Third bit: represented by 'x' (exec)
- Examples:
  - `rw-r--r--` **represents** `000 110 100 100`
  - `rwxrwxrwx` **represents** `000 111 111 111`

Australian National University

Special modes in textual representation

When special modes are present, the bits in the special modes change the display of the executable bits of the remaining groups.

- If SUID bit is set: display 's' if the owner exec bit is set; otherwise display 'S'.
- If SGID bit is set: display 's' if the group exec bit is set; otherwise display 'S'.
- If sticky bit is set: display 't' if the 'other' exec bit is set; otherwise display 'T'.

Australian National University

Special modes in textual representation

Examples:

- 110 111 110 100 can be represented as
  `rwsrwSr--`

- 011 111 101 101 can be represented as
  `rwxr-sr-t`

- 101 110 110 100 can be represented as
  `rwSrw-r-T`

Australian National University

Octal Representation

八进制权限表示法

- Another representation of permission bits that is commonly used is the octal notation.
- Each group of three bits can be represented as an octal.
- For example:
  - 000 110 100 100 in octal notation is 0644. (八进制)
  - 011 111 101 101 in octal notation is 3755.
- A 3-digit octal permissions means the special modes are absent, e.g., 644 is the same as 0644.
- Octal notations are used in some commands to set permissions ('chmod') and permission masks ('umask').

Australian National University

Default Permissions

- Unix utilities typically use default permissions 0666 when creating a new file and permissions 0777 when creating a new program.
- Permissions can be further adjusted by the umask:
  - a four-digit octal number specifying the rights that should be withheld.
- Actual default permission is derived by masking the given default permissions with the umask: compute the logical AND of the bits in the default permission and of the inverse of the bits in the umask.

can't access to it

open.

→ what permission I'm going to revoke / what won't available

撤回

- **文件的默认权限**：
  - 创建 **新文件** 时的默认权限为 0666 (即读写权限，但没有执行权限)。
  - 创建 **新程序** 时的默认权限为 0777 (即读、写、执行权限)。
- **umask (用户文件创建掩码)**：
  - **umask** 用于调整文件和目录的默认权限。
  - 通过 **掩码** (mask) 控制权限。例如：
    - umask 022 将默认权限 0666 调整为 0644 (即去掉组和其他用户的写权限)。

默认权限与 umask 掩码进行 **按位取反** 后的结果作为实际权限

Australian National University

Default Permissions

- Example: default permission 0666, umask 0077
- Invert 0077: gives 7700, then AND:

0077 = 000000111111 INVERT
7700 = 111111000000

0666 = 000110110110
7700 = 000111000000 **AND**
⇒ 0600 = 000110000000

only owner
no one else can execute

- Owner of the file has read and write access, all other access is denied.
- umask 7777 denies every access, umask 0000 does not add any further restrictions .

Australian National University

Some umask Settings

- 0022: withhold none from owner, withhold write permission for group and for other.
- 0027: withhold none from owner, withhold write permission from group, withhold all from other.
- 0037: withhold none from owner, withhold write and execute from group, withhold all from other.
- 0077: withhold none from owner, withhold all from group and other.

Australian National University

Permissions for Directories

- Every user has a home directory; to put files and subdirectories into, the correct permissions for the directory are required.

在 Unix 系统中，目录权限和文件权限的意义略有不同。对于目录而言，权限控制主要包括以下三种：

- Read permission: to find which files are in the directory, e.g. for executing `ls`.
- Write permission: to add files to and remove files from the directory.
- Execute permission: to make the directory the current directory (`cd`) and for opening files inside the directory.

Australian National University

Permissions for Directories

- To access your own files, you need execute permission in the directory.
- Without read permission on the directory, but with execute permission, you can still open a file in the directory if you know that it exists but you cannot use `ls` to see what is in the directory.

Australian National University

Permissions for Directories

- To stop other users from reading your files, you can either set the access permissions on the files or prevent access to the directory.
- You need write and execute permission for the directory to delete a file; no permissions on the file itself are needed, it can even belong to another user.

文件本身的权限不影响删除操作，即使该文件归属于其他用户，只要拥有相应目录的写和执行权限，用户就能删除文件

Australian National University

Changing Permissions

change permission

- ~~Access rights can be altered~~ with `chmod` command:
  - `chmod 0754 filename`
  - `chmod u+wrx,g+rx,g-w,o+r,o-wx filename`

take away

add those permission to user

- The first octal number from the left (representing special modes) is optional, e.g.,
  - `chmod 754 filename`

  achieves the same thing as `chmod 0754 filename.`

Australian National University

Changing Ownership

- Ownership can be altered with the `chown` command:
  - `chown <Owner>:<Group> <filename>`

- <Owner>：新的文件所有者（用户）。
- <Group>：新的文件所属的组。
- <filename>：要更改的文件名。

- For example:
  - `chown alice:tutors  foo.txt`

  changes the owner of foo.txt to user alice in group tutors.

如果只想更改文件的组，可以使用：chown :tutors foo.txt

可以使用 -R 选项递归更改目录及其所有子文件的所有者和组：sudo chown -R alice:tutors /path/to/directory

Australian National University

Permissions: Order of Checking

- Access control uses the effective UID/GID:
  - If the subject's UID owns the file, the permission bits for owner decide whether access is granted.
  - If the subject's UID does not own the file but its GID does, the permission bits for group decide whether access is granted.
  - If the subject's UID and GID do not own the file, the permission bits for other (also called world) decide whether access is granted.
- Permission bits can give the owner less access than is given to the other users.
  - But the owner can always change the permissions.

权限控制基于有效的 UID/GID

在 Unix 系统中，文件权限的检查遵循特定的顺序，这取决于请求访问的用户和文件的拥有者的关系：

1. **检查文件所有者（Owner）权限**：
   - 如果当前用户（主体）的 UID 与文件的所有者 UID 匹配，那么系统会根据**所有者权限（Owner bits）**来决定是否授予访问权限。

2**检查组（Group）权限**：
- 如果当前用户的 UID 与文件所有者不匹配，但用户所属的 GID 与文件的 GID 匹配，则系统会依据**组权限（Group bits）**来决定是否允许访问。

- 3**检查其他用户（Other/World）权限**：
  - 如果用户的 UID 和 GID 都不匹配，则系统会依据**其他用户权限（Other bits）**来决定是否允许访问。

文件所有者可以随时更改文件的权限（使用 chmod 命令），因此即便初始权限较少，也可以根据需要调整。

Australian National University

Security Patterns

Some general security principles implemented in Unix.

- Controlled invocation: SUID programs.
- Physical and logical representation of objects: deleting files.
- Access to the layer below: protecting devices.
- Search path
- Importing data from outside world: mounting filesystems.

Australian National University

Controlled Invocation受控调用

- Superuser privilege is required to execute certain operating system functions.
- Example: only processes running as root can listen at the "trusted ports" 0 – 1023.
- Solution adopted in Unix: SUID (set userID) programs and SGID (set groupID) programs.
- SUID (SGID) programs run with the effective user ID or group ID of their owner or group, giving controlled access to files not normally accessible to other users.

Australian National University

Displaying SUID Programs

- When **ls -l** displays a SUID program, the execute permission of the owner is given as **s** instead of **x**:

```
$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 59640 Mar 23  2019 /usr/bin/passwd
```

执行 ls -l 命令时，如果程序设置了 SUID，则会看到权限为 s 而不是 x

- When **ls -l** displays a SGID program, the execute permission of the group is given as **s** instead of **x**:

```
$ ls -l /usr/bin/ssh-agent
-rwxr-sr-x 1 root ssh 362640 Mar  4  2019 /usr/bin/ssh-agent
```

Australian National University

SUID to root

- When root is the owner of a SUID program, a user executing this program will get superuser status during execution.
- Important SUID programs:

`/bin/passwd`	change password
`/bin/sudo`	escalate privilege to root
`/bin/su`	change UID

- As the user has the program owner's privileges when running a SUID program, the program should only do what the owner intended

SUID 程序必须谨慎使用，因为它们可能被滥用来获取 root 权限

Australian National University

SUID Dangers

- By tricking a SUID program owned by root to do unintended things, an attacker can act as the root (confused deputy attack).
- All user input (including command line arguments and environment variables) must be processed with extreme care.
- Programs should have SUID status only if it is really necessary.
- The integrity of SUID programs must be monitored (e.g., using tripwire).

SUID Dangers（SUID 的危险）

- **Confused Deputy Attack（困惑副手攻击）**：
  - 攻击者可以通过欺骗 root 拥有的 SUID 程序执行非预期操作，从而冒充 root 用户。这种攻击通常利用 SUID 程序中不受信任的输入来达到权限提升的目的。
- **用户输入处理**：
  - 所有用户输入（包括命令行参数和环境变量）必须经过严格验证，以防止攻击者利用恶意输入操纵 SUID 程序。
- **SUID 状态的谨慎使用**：
  - 只有在确实必要的情况下，才应为程序设置 SUID 权限，以最小化潜在的安全风险。
- **监控 SUID 程序的完整性**：
  - 应使用工具（如 Tripwire）定期监控 SUID 程序的完整性，以防止未经授权的修改。

Applying Controlled Invocation

- Sensitive resources, like a web server, can be protected by combining ownership, permission bits, and SUID programs:
- **Least privilege:** Create a new UID that owns the resource and all programs that need access to the resource.
- Only the owner gets access permission to the resource.
- Define all the programs that access the resource as SUID programs.

Applying Controlled Invocation（应用受控调用）

- **保护敏感资源**：
  - 例如，Web 服务器等敏感资源可通过结合所有权、权限位以及 SUID 程序来保护。
- **最小特权原则（Least Privilege）**：
  - 创建一个新的 UID，使其拥有资源及需要访问该资源的所有程序。确保仅资源所有者具备访问权限，从而减少被滥用的可能性。
- **SUID 程序的定义**：
  - 将所有需要访问特定资源的程序定义为 SUID 程序，以确保它们能够在受控权限下执行。

Managing Security

- Beware of overprotection; if you deny users direct access to a file they need to perform their job, you have to provide indirect access through SUID programs.
- A flawed SUID program may give users more opportunities for access than wisely chosen permission bits.
- This is particularly true if the owner of the SUID program is a privileged user like root.

Australian National University

Deleting Files

- General issue: logical vs physical memory
- Unix has two ways of copying files.
  - **cp** creates an identical but independent copy owned by the user running **cp**.
  - **ln** creates a new filename with a pointer to the original file and increases link counter of the original file; the new file shares its contents with the original.

– If a process has opened a file which then is deleted by its owner, the file remains in existence until that process closes the file.

Australian National University

Deleting Files

- Once a file has been deleted the memory allocated to this file becomes available again.
- Until these memory locations are written to again, they still contain the file's contents.
- To avoid such memory residues, the file can be wiped by overwriting its contents with random patterns before deleting it.
- But advanced file systems (e.g. defragmenter) may move files around and leave copies.

Australian National University

Protection of Devices

- General issue: logical vs physical memory
- In Unix, "everything is a file".
  - Unix treats devices like files; access to memory or to a printer is controlled like access to a file by setting permission bits.
- Devices commonly found in directory **/dev**:

/dev/console	console terminal
/dev/kmem	kernel memory map device (image of the virtual memory)
/dev/tty	terminal
/dev/sda1	hard disk
/proc	virtual file system containing system information

Australian National University

Accessing the Layer Below

- Attackers can bypass the controls set on files and directories if they can get access to the memory devices holding these files.
  - In Linux, user group disk has write access to raw devices. Members of this group can bypass file and directory permissions.
- If the read or write permission bit for other is set on a memory device, an attacker can browse through memory or modify data in memory without being affected by the permissions defined for files.
- Almost all devices should therefore be unreadable and unwritable by “other”.

攻击者可以通过直接访问设备来绕过文件系统权限

Australian National University

Example

- The command `passwd` allows any user to change their password, thus modifying the /etc/shadow file.
- Defining `passwd` as a SUID to root program allows `passwd` to acquire the necessary permissions.
- But a compromise of `passwd` would allow an attacker to modify the shadow file, e.g., to reset the administrator password.

Australian National University

Terminal Devices

- When a user logs in, a terminal file is allocated to the user who becomes owner of the file for the session.
- It is convenient to give "other" read and write permission to this file so that the user can receive messages from other parties.
- Vulnerabilities:
  - other parties can now monitor the entire traffic to and from the terminal, potentially including the user's password.
  - Others can send commands to the user's terminal, and execute them using the privileges of another user.

Australian National University

Mounting File Systems

- General issue: When importing objects from another security domain into your system, access control attributes of these objects must be redefined.
- Unix file system is built by linking together file systems held on different physical devices under a single root `/` with the `mount` command.
- Remote file systems (NFS) can be mounted from other network nodes.
- Users could be allowed to mount a filesystem from their own floppy disk (`automount`).
- Mounted file systems could have dangerous settings, e.g. SUID to root programs in an attacker's directory.

Australian National University

Environment Variables

- Environment variables: kept by the shell, normally used to configure the behaviour of utility programs
- Inherited by default from a process' parent.
- A program executing another program can set the environment variables for the program called to arbitrary values.
- Danger: the invoker of setuid/setgid programs is in control of the environment variables they are given.
- Not all environment variables are documented!

Australian National University

Examples

The command env lists all the defined environment variables in the current shell.

Some examples:

PATH	# The search path for shell commands (bash)
TERM	# The terminal type (bash and csh)
DISPLAY	# X11 - the name of your display
LD_LIBRARY_PATH	# Path to search for object and shared libraries
HOSTNAME	# Name of this UNIX host
HOME	# The path to your home directory (bash)

Australian National University

Example: the "Shellshock" bug

- Discovered in September 2014.
- Exploits a vulnerability in parsing of environment variables.
- Allows an attacker to inject arbitrary codes into environment variables.
- The injected codes get executed if the target (victim) executes a bash shell.
- See

    http://en.wikipedia.org/wiki/Shellshock_(software_bug)

Australian National University

Search path

- General principle: execution of programs taken from a ‘wrong’ location.
- Users can run a program by typing its name without specifying the full pathname that gives the location of the program within the filesystem.
- The shell searches for the program following the search path specified by the `PATH` environment variable in the `.profile` file in the user’s home directory.

安全风险:

- 如果 PATH 变量被恶意修改，用户可能会执行到攻击者放置的恶意程序。
- 常见攻击手法包括将恶意程序放置在系统优先搜索的目录中，从而替换合法程序。

```
# 检查系统中使用的 ls 命令实际路径
which ls
type ls
```

Australian National University

Search path

- A typical search path (it may differ across different systems):

系统会按照从左到右的顺序搜索路径，找到匹配的程序后立即执行。

```
PATH=.:$HOME/bin:/bin:/usr/bin:/usr/local/bin
```

- Directories in the search path are separated by ‘:’; the first entry ‘.’ is the current directory.
- Search paths are read from left to right.
- When a directory is found that contains a program with the name specified, the search stops and that program will be executed.

如果 .（当前目录）在 PATH 的开头或中间，攻击者可以利用该漏洞:

- **路径劫持**：攻击者在当前目录下创建一个恶意程序，与常用命令（如 ls）同名。当用户在当前目录运行 ls 时，实际上执行的是攻击者的恶意程序。
- 确保 . 不在 PATH 中，尤其是对 root 用户来说

PATH=/usr/local/bin:/usr/bin:/bin

使用绝对路径

- 调用关键命令时使用完整路径，以避免意外调用恶意程序。例如：/bin/ls

Australian National University

Search path

- To insert a Trojan horse, give it the same name as an existing program and put it in a directory that is searched before the directory containing the original program.
- As a defence, call programs by their full pathname, e.g. **`/bin/ls`** instead of **`ls`**.
- Make sure that the current directory is not in the search

- As a defence, call programs by their full pathname, e.g. `/bin/ls` instead of `ls`.
- Make sure that the current directory is not in the search path of programs executed by root.

Australian National University

Management Issues

- Brief overview of several issues relevant for managing Unix systems
  - Protecting the root account
  - Networking: trusted hosts
  - Auditing

Australian National University

Protecting the root Account

- The root account is used by the operating system for essential tasks like login, recording the audit log, or access to I/O devices.
- The root account is required for performing certain system administration tasks.
- Superusers are a major weakness of Unix; an attacker achieving superuser status effectively takes over the entire system.
- Separate the duties of the systems manager; create users like `uucp` or `daemon` to deal with networking; if a special users is compromised, not all is lost.

Australian National University

Superuser

- Systems manager should not use root as their personal account. 而是应通过普通用户账户切换到 root
- Change to root from a user account using `/bin/su`; the O/S will not refer to a version of `su` that has been put in some other directory.
- Record all `su` attempts in the audit log with the user who issued the command.
- `/etc/passwd` and `/etc/group` have to be write protected; an attacker who can edit `/etc/passwd` can become superuser by changing its UID to 0.

Australian National University

Trusted Hosts

- In legacy Unix systems, commands such as rlogin or rsh allows users to login remotely.
  - Both rlogin and rsh transmit passwords in plain text
  - In modern Linux systems they are replaced by 'secure shell' (ssh)
- Users from a trusted host can login without password authentication; they only need to have the same user name on both hosts.
- Trusted hosts of a machine are specified in `/etc/hosts.equiv`.
- Trusted hosts of a user are specified in the `.rhosts` file in the user's home directory.
  - User can either access all hosts in the system or nothing; exceptions difficult to configure.

- **受信任主机**配置简化了跨主机的登录，但应谨慎使用，以防止未经授权的访问。
- **审计日志**是系统安全的重要组成部分，应定期检查，以及时发现潜在的安全问题。

Australian National University

Audit Logs

In modern Linux systems, log files are located in /var/log/. For example:

- /var/log/auth.log: all authentication related events, including wrong passwords, attempts to 'sudo', etc.
- /var/log/dmesg: information related to hardware and device drivers
- /var/log/kern.log: information logged by the kernel
- /var/log/syslog: global system activity data

Australian National University

Audit Logs

- Audit logs may sometimes contain sensitive information.
  - Be careful of what information you log and the permissions to the log files.
- Example: bugs in Mac OS X (version 10.3.3) cause system encryption software to record disk encryption password in plaintext in installation logs.
  - See /var/log/install.log in the affected Mac OS X
  - Log accessible by normal (non-root) use. See:
    - https://www.mac4n6.com/blog/2018/3/30/omg-seriously-apfs-encrypted-plaintext-password-found-in-another-more-persistent-macos-log-file
- Example: In Android (prior to 'Jelly Bean' version), apps can request permission to read system logs.
  - See, e.g., William Enck, et. al. : A Study of Android Application Security. USENIX Security Symposium 2011

Australian National University

Summary

- Unix served as a case study to see how core security primitives can be implemented.
- Illustrate a number of general security issues.
- Also relevant, but not covered yet: network security, software security.
- For practical security, it does not suffice to have a "secure" operating system; the system also has to be managed securely.