

CSIS0230A Principle of Operating System (Class A)

Tutorial 11

Unix Security

3rd December 2001

In this tutorial, you are required to setup a simple message board system for your Linux machine. The requirements are:

1. Every user can read the message board through the “msgboard” program;
2. Every user can append a message to the message board through “msgboard” program.

The data of the message board is stored as a text file named “data” in the directory `/usr/local/message/`, and the “msgboard” program is located in `/usr/bin/`, available for any user to run. Due to the security reason, the data file is owned by root and cannot be written by any other users (i.e. the attribute of the file should be `-rw-r--r--`). The msgboard program (you can find it at `/root/msgboard.cc`) have been written for you as follows:

```
#include <stdlib.h>
#include <iostream>
#include <fstream>
#include <sys/types.h>
#include <unistd.h>
#include <wait.h>
#include <errno.h>
#include <string.h>

#define DATA_FILE "/usr/local/message/data"
int main() {
    char c;
    bool cont=true;
    while (cont) {
        //ask the user whether read or write is needed
        cout << "Read or Write message? (r/w) " << endl;
        cin >> ws; //remove all the white space
        cin.get(c);
        while (c != 'r' && c != 'w') { //input checking
            cout << c << "Invalid input! Try again." << endl;
            cin >> ws;
            cin.get(c);
        }
        if (c == 'r') {
            // read requested
            pid_t pid;
            if ((pid=fork()) > 0) {
                waitpid(pid,0,0);
            } else {
                //use the "less" program to display message board
                if (execlp("less","less",DATA_FILE,0) == -1) {
                    cerr << strerror(errno) << endl;
                }
            }
        }
        } else {
```

```

// write requested
cout << "Please enter your message. Press enter 3 times when finished" << endl;
int entercount=0;
ofstream of;
of.open(DATA_FILE,ios::app); //open file in append mode
if (!of.is_open())
    cerr << "File open fail: " << strerror(errno) << endl;
else {
    while (cin.get(c) && entercount<3) {
        if (c=='\n')
            entercount++;
        else
            entercount=0;
        of.write(&c,sizeof(c));
    }
    of.close();
}

// ask user whether to continue or not
cout << "Exit or Continue? (e/c)" << endl;
cin >> ws;
cin.get(c);
while (c != 'e' && c != 'c') {
    cout << c << "Invalid input! Try again." << endl;
    cin >> ws;
    cin.get(c);
}
if (c == 'e')
    cont=false;
else
    cont=true;
}
return 0;
}

```

Your Tasks:

1. Set up the message board system using the above program. The root password is `tutorial`. The location of related files should be:

FILE NAME	LOCATION
<code>msgboard</code>	<code>/usr/bin/</code>
<code>data</code>	<code>/usr/local/message/</code>

(Hints: you are required to set the `setuid`-bit for the “`msgboard`” program) The following steps is checks whether your system have been setup correctly:

- (a) login as normal user (username: `guest`; password: `guest1`)
 - (b) try to delete the data file by “`rm /usr/local/message/data`”. (A normal user should NOT be able to delete the data file)
 - (c) run the “`msgboard`” program and add a new message. (You should be able to add a new message while using a normal user account)
 - (d) use the same program to view the message. (You should find your message have been added to the message board)
2. There is a *security hole* in the system: the program executed might net be the less program of the system, and a normal user subsequently execute any command with root privilege. Where is the security hole, and how to effect an attack? How to make sure that the correct less is executed?
 3. There is an another problem of the system: `less` is too powerful a program to be used this way. Read the manpage of `less`. What type of attacks can be done to the current system? How to solve the problem? (Hint: is special privilege really needed?)