

Cyber Security Lab 1

Due

In two parts, Jan 26 and Feb 2 in class.

Goal

Use techniques for least privilege and multiple users to solve similar problems in Windows XP and Linux.

Scenario

You have been contracted by YoYoDyne, Inc. to implement a data management package. They are not certain about which platform they want to build on, so they want to better understand the security mechanisms provided by Windows and Linux. They hired you to prototype the solution on both Windows and Linux.

Their application needs to perform some privileged operations (to access very secret files) regardless of who invokes it. After this initial privileged operation, the program must read and write files that are only accessible to the invoking user.

Windows Requirements

On Windows, you could implement this as a server will be running as a service. It will listen on a named pipe for client requests. The client will pass in the full path name of the file it wants the server to access. The server will try to access the file as itself, and then as the user. The server should log the results of these access attempts to a log file. Assuming it is successful; the server will return the first 512 bytes of the file. Instead of actually installing a service, you can use the “runas” utility to invoke the server and client as different users for testing.

Run your Server as a member of the administrative group. The program should examine and disable all unnecessary privileges at the start of the program. It should log the privileges and their original states to a log file.

Linux Requirements

On Linux, you can use setuid implement a program that is initiated as root user. Then use the libcap/setcap calls to drop unnecessary privileges. Finally use the setresuid call to change the user ID to the invoking user. This program should perform the same operations as the windows programs to fetch files for the invoking user. The program should implement similar logging as identified in the Windows requirements. Since the program is invoked directly, the requested file can be passed through the command line or standard I/O rather than through a pipe.

Alternate Requirements

For this scenario, a modular solution as demonstrated in the qmail design could also be applied. You may chose to use the modular solution for either Linux or Windows. But

the implementation on one platform must use the least privilege and user adjusting OS mechanisms.

Things you will need to know

Windows Base Code

Very simple client and server Windows code will be posted to the web site under the assignments tab. The server can run from the command shell or as a service.

If you are operating in the lab, you will want to change the name of the pipe to avoid conflicts with other student's services.

The sample code was developed with Visual Studio and that is what is installed in the lab. The .mak files are exported, so you should be able to use other compilers if you so desire.

The list at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secauthz/security/authorization_functions.asp includes the functions you will need for impersonation and privilege manipulation. Specifically, the following functions should be of interest to you:

- ImpersonateNamedPipeClient
- RevertToSelf
- AdjustTokenPrivileges
- GetTokenInformation

Linux

Look at the man pages for setresuid, chmod, and the "Setuid Demystified" paper to determine how to set the executable to take on a new user ID on execution and to change the user ID during execution.

For information about restricting privileges in a running program look at the capabilities article referenced in class <http://www.linuxjournal.com/article/5737>, and the man pages for capabilities, capget, capset, cap_set_proc, and cap_get_proc.

Your tasks

You will need to perform the following tasks.

1. Create the client and server programs for windows that match the requirements of the scenario. Create an application on Linux that match the requirements of the scenario. For both operating systems, run the following cases where the service is running as an administrative user (Ellen or root) and the client or unprivileged portion of the application is running as an unprivileged user (Alice):
 - a. Client asks for file that both have access to.
 - b. Client asks for file that the client has access to.
 - c. Client asks for file that the administrative user has access to.

2. Identify three secure coding issues that you did do or would do if this were a real project.
3. For Windows, how would you change the program, to restrict access to the pipe to only members of the Engineering group?

Hand-in Items

This lab can be submitted in two pieces. On Jan 26, the solution for the Linux platform or the Windows platform can be submitted. On Feb 2, the solution for the other platform is due.

Electronic hand in is fine. Provide the following items for each platform:

- Server log from running the operations in the first task.
- Code for the client and server
- Answers to questions 2 and 3