Vista Least Privilege Lab

Due

February 18, 2010 in compass by midnight.

Goal

Use techniques for least privilege to implement a simple client server program.

Scenario

CosaNostra Pizza has retained you for another assignment. CosaNostra Pizza needs to implement a delivery routing package, and they are considering using Windows 7 as the OS, but they are unsure of how well the security mechanisms of Windows 7 would support them. They hired you to prototype the solution on Windows 7 and evaluate the security support.

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.

Requirements

CosaNostra Pizza already has a simple client server program that implements the basic communication but minimal security and no core functionality. On Windows, you would install the server as a service. It listens on a named pipe for client requests. The client passes in the full path name of the file it wants the server to access. For testing purposes, 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 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.

Windows Base Code

Very simple client and server Windows code is posted to the web site under the assignments tab. This code is also on Alice's desktop on your VM image. The server can run from the command shell or as a service. If run from the command line (or from the debugger) you must pass it an argument such as -noService. If the server is invoked with no arguments, it assumes it is being invoked as a service.

The sample code was developed with Visual Studio 2008 professional and that is what is installed on the VMs. Initially you have access to the VirtualBox Virtual Machines in the lab. Use the same Windows 7 VM's that you used for the ACL lab. Possibly additional Windows 7 VM's will be made available from a larger Vmware server, but they are not yet available at this time.

You may develop on your own machine if you have access to a Windows 7 system, and you want to reproduce the development environment. As students you should have access to Visual Studio 2008 from the University's WebStore.

Windows Documentation

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

Notes on using MS Visual Studio (MSVC)

In the sample server and client directories you will see .sln and .vcproj files. Double click on these files to open up MSVC.

Look at Build menu to compile. Error messages will appear in the bottom window and compile errors should be highlighted in the code.

Look at debug menu to run your program. Running without debugger will leave the console window up after executing. Running with debugger (F5) will remove window after program has exited. You can of course also run the program from cmd shell directly. By default, the executable will be in the Debug subdirectory.

Look at Project->Properties to set arguments for the program to use while debugging. Specifically look at the Configuration Properties->Debugging window. Fill in the Command Arguments. You will need to pass the -noService argument to the sample server so it will not start up as a service.

Consider setting breakpoints to help your debugging. To set a breakpoint, select a line and right-click.

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. Run the following cases where the service is running as an administrative user (Alice or Gus) and the client or unprivileged portion of the application is running as an unprivileged user (e.g., Bob):
    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. Write up a final recommendation report for Uncle Ezno that addresses the following questions.
    a. What you would recommend to CosaNostra Pizza if they decide to proceed with an implementation on Windows 7?
    b. How would you implement similar requirements in a Linux system? Which do you think would be better (more secure, maintainable, etc)?
    c. What is their exposure if a code injection exploit is found against the client or server?

Hand-in Items

Hand in on compass. Provide the following.

- Server log from running the operations in the first task.
- Code for the client and server
- Final recommendation report. (Task 2)