**Activity 7-6: Creating a Perl Script that uses the Win32 API**

**Description**: In this activity, you write a basic Perl script, using the formatting functions you have already learned and the win32 API functions in Table 7-11. If possible, work in groups of three to four students.

1. If necessary, open a command prompt window, and switch to the C:\Perl directory. Type **notepad Wind32.pl** and press **Enter**. Click **Yes** when prompted to create a new file.
2. In the new Notepad document, type **# Win32.pl** on the first line and press Enter.
3. Use what you’ve learned in this chapter to write comments for documenting the program. Be sure to enter the author name, date, and a brief description of what the program does, such as the functions it accesses from the Win32 API.
4. After your lines of documentation, press **Enter** several times to create blank lines for separating your comments from the program code. Then type **use win32;** and press Enter. (note: don’t forget the semicolon)
5. You need five pieces of information (noted in the bulleted list before this activity) from the Win32 API. Attempt to write the code for getting this information, and then save the program. If you need assistance, use the following steps.
6. Type **$login = Win32::LoginName();** and press Enter. This line populates the $login variable with the information gathered from LoginName().
7. Next, type the following lines to populate the other variables needed to complete the task, pressing Enter after each line:

$NetBIOS = Win32::NodeName();

$filesystem = Win32::FsType();

$Directory = Win32::GetCwd();

$os\_name = Win32::GetOSName();

1. The following variables need to be displayed onscreen. Type the lines of code as shown, pressing Enter after each line.

Print “$login\n”;

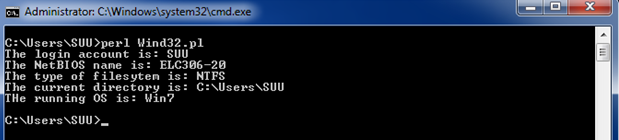
Print “$NetBIOS\n”;

Print “$filesystem\n”;

Print “$Directory\n”;

Print “$os\_name\n”;

1. After typing all the code, save the program, run it, and debug any errors.
2. Spend time improving the report’s formatting so that anyone reading the output could understand its meaning.



1. Are there any improvements your group thinks should be made to the script? Explain. What other information might be beneficial for a security professional to get from this report?

We think this script could be improved by showing additional information, such as IP address. This is useful to us as well as the other information. This script can help a security professional because it can quickly grab information the professional may need for tasks.