Graduate Program in Information Science and Telecommunications and Networking
School of Information Sciences
University of Pittsburgh

TEL2821/IS2150: INTRODUCTION TO SECURITY
Lab 2: Forensics & Port Re-direction

Version 1.2, Last Edited: September 24, 2007
contact: Saubhagya Joshi {srjoshi at mail.sis.pitt.edu}

Group Members: _____

_____

_____

_____

Date of Experiment: _____

Graduate Program in Information Science and Telecommunications and Networking
School of Information Sciences
University of Pittsburgh

Part I: Objective

The objective of this laboratory exercise is twofold:

1. Introduce you to some of the tools and techniques used for forensic analysis.
2. Demonstrate some of the mechanisms used by malicious attackers as well as forensic experts to disrupt computer networks and manipulate information access.

This lab session will cover data storage and access, bypassing filtered [blocked] ports, reviewing Internet activity, and the use of steganography. Open-source forensic tools will be introduced and demonstrated for each exercise.

The lab has been setup for all of the exercises and the required executables are available in the lab machines. Instructions are available throughout the lab handout.

Part II: Equipment/Software

Most of the tools used for this lab exercise is freely available for non-commercial testing purposes and open-source software, either freeware or shareware. All the executables required for the lab is available in the lab or as temporary download as a zip-compressed file at: http://www.sis.pitt.edu/~srjoshi/public/d/lab2.zip

The zip-compressed file contains three folders: (1) Data, which contains all the original data you need for the lab exercises, (2) Exercises, which contains all of the four exercises in this lab, and, (3) Installers, which contains all the executables you will need for completing the exercises (Please make sure your system has a text editor like Notepad. Also you need to be able to access the command prompt).

Note that Exercise 1 has parts that use different executables for hex editor and hash function, and hence, different folders. Also, any data that needs to be manipulated are available within the respective exercise folders. If at any point, you corrupt the working data, or you need to re-do the exercises, you can copy fresh copies of the data files from the "Data" folder.

Windows Vista PCs are provided for the purposes of the lab. The access accounts and passwords are given on the screen. Some of the exercises use GUI, but some use the command line interface. When the command line is required, a shortcut to the command prompt is given in the exercise folder.

The executables that are used in this lab exercise are as follows:

Hidden Files:

- XVI32 hex editor (Shareware renamed to hex.exe for the lab purposes)
- "fciv.exe" is a ***F**ile **C**hecksum **I**ntegrity **V**erifier* from Microsoft.
- Text editor (Notepad is good enough)
- Data files: "sol.exe" and "sol.modified.exe"

Port Redirection:

- Quick 'n Easy FTP Server (Freeware download from http://www.pablovandermeer.nl)
- FPIPE (Freeware download from http://www.foundstone.com)
- Batch executable file: "checkPort.bat" used to display relevant port information.
- User database file for the FTP server: "users.xml"

IE Activity analysis:

- Pasco (Freeware download from http://www.foundstone.com)
- Galleta (Freeware download from http://www.foundstone.com)
- Internet Explorer cache file (index.dat)
- Internet Explorer cookie files

Steganography:

- JPHS (Jpeg Hide and Seek) v0.5 (Freeware download from www.stegoarchive.com)
- Text editor (i.e. Notepad)
- Image file in *jpeg* format: "KaalBhairava.jpg"

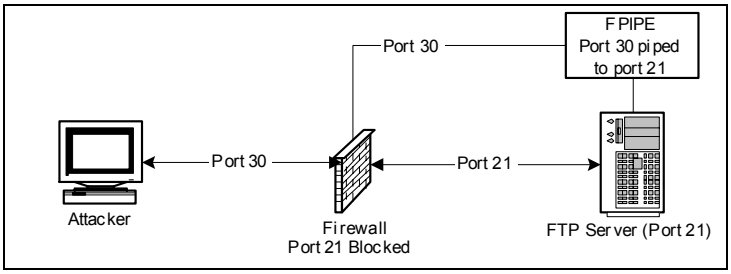
Part III: Exercises

You can do the following exercises either in laboratory in the Windows Vista machines, or re-create the exercise environment in any other Windows 2000 (or later) environment of your choice. Instructions are provided in Part IV: Appendix A. After you complete all the exercises, make sure you roll back to the original state. Refer Part IV: Appendix B.

Exercise 1: ***Port Redirection***

Objective

The purpose of this lab is to demonstrate how an attacker could exploit a machine and obtain access to a server with a filtered port by piping another unfiltered port. Because of sophisticated Trojans, it could be hard for a virus detection program to detect the problem. Because of that, a port scanner/listener must be used to determine if/what ports are actively carrying traffic.



Scenario

Imagine that an IT department has an FTP server on an IBM server that they use to share source code between other departments within the organization in various locations throughout the US on the same LAN/WAN. By default, the information security department blocks certain known ports from being exposed to the internet through a firewall. Some of these ports include the well known 21, 23, 80, 8080, etc.

Graduate Program in Information Science and Telecommunications and Networking
School of Information Sciences
University of Pittsburgh

A users logs onto this IBM server with Windows Vista through Windows Remote Desktop Connection and accidentally downloads a Trojan that is meant to get access to and FTP server. However, if port 21 is blocked through the Firewall, how could the attacker connect to the FTP server? There is a very simple technique known as port redirection. Port redirection is a sophisticated way of bypassing port filtering, firewalls, and IPSEC.

Steps

i. Make sure you are within the "Exercises" folder.

ii. Get the FTP server running
- Double Click the "FTPServer.exe" file in the subfolder "1.0 PortRedirection" to open the FTP Server configuration tool.
- Check if the users include "anonymous". You need to create it if it does not exist.
- Click the **Start** button on the top left of the FTP Server configuration panel.

iii. Confirm that the FTP server is running on port 21.
- Which port is the FTP Server running on (check EasyFTP)? (……..)
- What is the IP address of the PC (check EasyFTP)? (…….…...)
- Use `netstat` to verify the FTP port. For this, you can start a command prompt and type `netstat -n`. The output displayed gives the following: *Protocol*, *Local Address*, *Foreign Address*, and *State.*
- Can you verify that the FTP server is running on your IP? (…….)
- Do not close the terminal. This terminal will be referred to later as "netstat terminal."

iv. Redirect the network traffic on port 21 to port 30 (or any arbitrary port number).
- Open another terminal with a command prompt.
- Determine the current active IP address of the PC by using the command `ipconfig` at the command prompt. Note that if you have multiple IP addresses, you might have to consult the FTP server configuration to determine the IP that is being used.
- Enter command: `fpipe -l 30 -s 30 -r 21 -v` <*ip-address*>
- Do not close the terminal. This terminal will be referred to later as "FPIPE terminal."
- Check the "netstat terminal" by entering command: `netstat -n`
- What port is the executable "fpipe" running on? (……..)

- NOTE: In case you need to stop and restart the server, you might get an error saying an FTP server is already running at port 21. In this case you need to go to the task manager and end the "FTPServer.exe" process.

v. Start a ftp-client session and connect to the server (Assume that port 21 is blocked)

- Open up a new command prompt terminal.
- At the prompt enter command: `ftp`
- If you are connected, check the FPIPE terminal. What is the response.
  (…….)
  (…….)
  (…….)

- Enter command: `open`
- At the "to" prompt, type: <*ip-address*> `30`
- At the "username" prompt, enter: `anonymous`
- At the "password" prompt, enter: (no password, just press **Enter**)
- Type command: dir
- Check the FPIPE terminal. What is the response?

  (…….)
  (…….)
  (…….)
  (…….)

vi. What sort of security problems can occur due to port redirection?

(…….)
(…….)
(…….)

vii. Can port redirection be used for any useful purpose?

(…….)
(…….)

viii. Close all open windows.

Exercise 2: ***Data Storage***

Objective
The purpose of this lab is to demonstrate the use of a hex editor and hash tool in computer forensics. This lab will also demonstrate how data can be modified within a file or hidden on a disk without the data being saved as a file.

Graduate Program in Information Science and Telecommunications and Networking
School of Information Sciences
University of Pittsburgh

Description

The lab will be using a hash value to find initial evidence of tampering within a file. A hex editor will be used to compare the two files to find the exact differences. Also, the hex editor should demonstrate that hidden data could be stored onto the storage device without actually saving as a file in the operating system.

The idea is that there is lots of *slack space* (shown as dots using the hex editor tool) on the storage device that runs to the end of the sector that the file is saved in. This is the space on a disk that is unused when a file smaller than a sector is saved into that sector. This slack space can only be used if the file saved in that sector is made large enough to take up all of the space in the sector. A hex editor can be used to directly store data directly onto this slack space, as will be demonstrated in this exercise.

The MD5 hashing tool uses an algorithm to derive a hash value for any given file. Each file has a unique hash value. Therefore slight changes to a file can generate totally different hash values.

Steps

i. Make sure you are within the "Exercises" folder. Then navigate to the "2.1 IntegrityCheckTool" folder.

ii. Try out Windows Solitaire games in the "data" sub-folder.

- In the "data" sub-folder, double-click on "sol.exe" to open a new game of Windows Solitaire.
- Does the game of solitaire function as intended? (……..)
- Next, in the "data" sub-folder again, double-click on "sol.modified.exe" to open a new game of Windows Solitaire.
- Does the game of solitaire function as intended? (……..)

iii. Actually, the original file "sol.exe" has been modified to "sol.modified.exe"

- Open a new command prompt terminal
- Use "fciv" to check the hash values of the two files.

iv. What is the hash values displayed for the original and modified files?
(Original: ……..……..……..……)
(Modified: ……..……..……)

v. Close all windows.

- NOTE: Hex-editors and other editors that can write directly to a storage device must be used with extreme caution.

vi. Make sure you are in the "Exercises" folder. Then navigate into the "2.2 HexEditor" sub-folder and double-click on "hex.exe" to open the hex editor.

vii. Confirm that the "sol.modified.exe" file has been altered.

- Click and drag the file "sol.modified.exe" into the Hex Editor window.
- Search and find out the phrase that has been added to the file. Hint: the phrase contains the word: "Brutus"
- What is the total phrase that has been added? Hint: starts and ends with: "
  (…….)
- Try to make other changes (possibly random) to the file and save the file with another name, but retaining the extension.
- Can you execute the new file? Explain why or why not?
  (…….)
  (…….)
  (…….)
  (…….)
  (…….)

viii. Close all windows.

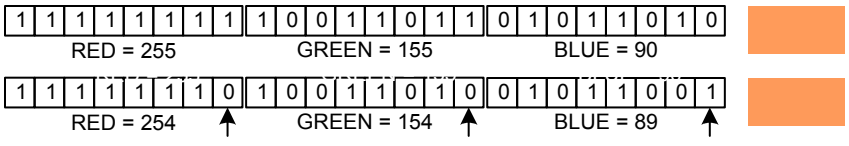
Exercise 3: ***Steganography***

Objective

The purpose of this lab will be to demonstrate the typical use of steganography. This lab will require the user to use a steganography tool to store and retrieve hidden data from a jpeg file.

Description

Steganography is the mechanism to hide relatively small amount of data in other data files that are significantly larger. A simple example of steganography may be the storage of text files in bit-mapped images. Each pixel in a bitmapped image is defined by three bytes representing red, blue and green colors. Each byte is made up of eight bits and represents a shade red, blue or green. Random changes to the least priority bit generally produce only slight changes of shade, typically undistinguishable by the normal human eye as shown by the figure (the arrow shows the changed least significant bit and the resulting color is shown on the side. Therefore, if the image file is 1 kilobyte, the number of bits that may be changed is given by the total number of least significant bits available (one bit from each byte) divided by the number of bits required for one text character (we consider 8 bits). Therefore, 1 kilobyte of image file can accommodate `1024/8 bits = 128 bytes`. Hence, a text file of 128 bytes could be hidden in a bit-mapped image of 1 kilobyte.



Steps

i. Make sure you are in the "Exercises" folder. Then navigate into the "3.0 Steganography" sub-folder and double-click on the "Jphswin.exe" file to open the Steganography tool.

ii. Hide data into "jpeg" file.

- Click on **Open Jpeg** on the menu bar and open the file "KaalBhairava.jpg" in "data" sub-folder.
- Create a text file "input.txt" with some text in the "data" folder.
- Click on Hide on the menu bar and give a password of your choice as prompted. Then, as prompted, point to the file "input.txt" that you intend to hide. Lastly, save the image as "hidden.jpg" in the "data" sub-folder. The message text in "input.txt" has been hidden in the jpeg image file "hidden.jpg".
- Note: Sometimes the Steganography tool gives "illegal operation" error when files need to be replaced. In such cases close the application and try again after deleting the files "hidden.jpg" and "output.txt".
- Close all open files.

iii. Retrieve the hidden message from the "jpeg" file.

- Open the file "hidden.jpg" using the Steganography tool "Jphswin.exe".
- Click on **Seek** on the menu bar. Then, as prompted, save the file as hidden "output.txt" into the "data" folder; replace if necessary.
- Is the message the same in "input.txt" and "output.txt" ? (……)

iv. What other types files be used to hide text data using stenography?

(……..……)
(……..……)

v. What are possibly some useful uses of stenography?

(……..……)
(……..……)

vi. Close all windows.

Exercise 4: ***Viewing Microsoft Internet Explorer Cache***

Objective
The objective of this exercise is to show how the encrypted Internet Explorer cache may be viewed using some freely available tools.

Description
Pasco and Galleta are to DOS-based executables that can decrypt the Internet Explorer cache. The use of these tools is demonstrated in this exercise.

Steps

i. Make sure you are in the "Exercises" folder. Then navigate into the "4.0 PascoGalleta" sub-folder and open a new command prompt terminal.

ii. Open the Internet Explorer cache called "index.dat," located in the "data" subfolder, using a text editor.

- At the prompt, enter:
  ```
  notepad data\index.dat
  ```
- What is the content like? (…….)
- Close **notepad** window.

iii. Use **pasco** to decrypt the Internet explorer cache called "index.dat"

- At the prompt, enter:
  ```
  pasco data\index.dat > index.txt
  ```
- At the prompt, enter:
  ```
  notepad index.txt
  ```
- What is the content like? (…….)
- Close **notepad** window.

iv. Use galleta to decrypt cookies.

- At the prompt, enter:
  ```
  galleta "data\bassel@advertising[2].txt"
  ```
- What is the result? Try with other cookies also. What is the result?

(……..……)
(……..……)

v. How can Pasco and Galleta be useful?

(……..……)
(……..……)

Part IV: Appendix A :: Installation of software required for the lab

1. Download the "lab2.zip" file from:
   http://www.sis.pitt.edu/~srjoshi/public/d/lab2.zip
2. Create a new folder and extract the contents of "lab2.zip" into the new folder.

You should have three folders: (1) Data, (2) Exercises (3) Installers

3. Make sure you have access to the command prompt.
4. Make sure you have access to Notepad
5. You are all set to start the exercises. Just follow the instructions in the handouts. Executables required for each exercise is available within the "Exercises" folder. For example, if you are working on Exercise 2, the folder you should look into, within the Exercises folder is "2.xxxxxxx" Remember that exercise 1 has two different folders, one for the hex editor and the other for getting hash functions.

Part IV: Appendix B :: Roll back PC to original state.

There are two ways to restore original state:

1. Delete the folder in which the Lab 2 is installed. In the lab PCs, the folder name is "Lab2.Forensics". Delete this folder along with its sub-folders. Then create the folder again and extract the contents of "lab2.zip" into the folder. The file "lab2.zip" is available at: http://www.sis.pitt.edu/~srjoshi/public/d/lab2.zip

2. Provided that you have followed instructions strictly and have not altered any of the executable files, or, their accessories, you can selectively delete the following six files created during the exercises:

    a. Exercise 1.0
        i. port.txt
        ii. show.txt

    b. Exercise 2.1
        i. modified version of file "sol.modified.exe"

    c. Exercise 3.0
        i. input.txt
        ii. hidden.jpg
        iii. output.txt

\- END of LAB 2 –

Bonus Question: *The hash values of the installation files provided for this lab are located in the "installers" folder. The hash values for these files are given in the file "integrity.txt", which is located in the "\Exercises\2.1 IntegrityCheckTool" folder. Using* ***fciv*** *verify that the installation files are the original distributions.*