Raspberry PI Pen Testing

The aim of this lab is to introduce you to the world of penetration testing with a basic demo. Following this lab sheet, you will learn about SQL injection (As well as some basic SQL) and buffer overflows in a C program. **Beware**, trying any of these techniques on any system without permission is a **crime and you will be caught and prosecuted for it**. With that out of the way, at the end of the cheat sheet is some helpful resources to safely and legally practice pen testing in your own time.

# Task 1: Network Mapper

First, we need to find out what services are running on the system. To do this, open a new console and scan ‘localhost’. Use the results to answer the following questions…

1. Name the services running on port 80 and 3306

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Task 2: SQL Injection

Having scanned the localhost, you’ve found there’s two services running on this system. One of them is a website being hosted on port 80. Type the following address into the browser:

localhost:80

Once you’re finished exploring, travel to the search page. Here there is a search bar that corresponds to the database running on the system. It’s important to verify that this search bar is secured against SQL injection attacks. To test this, type the following into the search bar and hit search:

' UNION (SELECT 1,2,3 FROM dual);#

The results don’t look good. The dual table is present in databases for testing, and the fact we can access this means there is no protection against and SQL injection attack. The way this works is as follows. Using the search normally would place the term into an SQL query as such where Search\_Name is what we put in the search box:

SELECT \* FROM ? WHERE ? LIKE ‘%<Search\_Name>%’;

However, by inserting a single quotation mark, writing our code then commenting out anything after that using the hash (#) character we can manipulate the results to show us other tables.

SELECT \* FROM ? WHERE ? LIKE ‘%’ UNION (SELECT 1,2,3 FROM dual);#%’;

Using this, search for and create SQL queries and answer the questions below. (There are hints available on the hint sheet if need be)

1. Find the name of the second ‘secret’ table

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Find the name of the secret file and decode the password

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Task 3: Buffer Overflow

Inside of this file there is a C program (myHelper.c). Why this file is here we have no idea, but we figure its likely this is giving users that root access. With some research and help from the cheat sheet, open a terminal with root access to the system.

Once you think you have it, type ‘whoami’ to confirm that you are the root user.

Congratulations! You have gained root access to your first system.