OverTheWire's Bandit Hints n' Tricks

hey guys, I made this guide, let me know what you think

1 Bandit Introduction

The Bandit series of wargames are a simple set of sequential levels which have the purpose of introducing new users to shell operations. Early levels are trivially easy, but consist of some fundamentals for UNIX shell. These are somewhat simple for those who have used some "terminal power-tools" before, but they can be challenging and fun for someone unfamiliar with the environment.

1.1 Terminal Fundamentals

Before the advent of window-management systems, graphics, and other modern niceties, there was a time when a computer would only be used through an 80-character wide text display (no, I don't believe it either, ask Dr. Wilson, Prof. Snider, or Dr. Hansen ;]). In this time when dinosaurs roamed the Earth, the terminal became a trove of advanced, well written tools. Instead of a file-browser, commands like `ls`[1], `cd`[2], `mv`[3], `mkdir`[4] and `find`[5] were used to complete the operations that we now use graphical file browsers like Windows' Explorer or OSX's Finder.

As you continue to use basic commands like `find`, or `ls`, you may come across special sequences, or *flags*, which usually look like `-f` or `--flag`. These

[1] LiSt files
[2] Change Directory (aka folder)
[3] MoVe file (can also be used rename a file)
[4] MaKe DIRectory (aka folder)
[5] FIND a file

flags allow for extra functionality with the command. For example, when using `ls` to list files, adding the `-l` flag will display extra metadata about files, and the `-a` flag will also display hidden files. Flags can be combined, so to list all files (including hidden files) and their metadata, use this command: `ls -a -l`. Finding what flags can be used for a command is usually simple, just enter the command with no following characters (or *arguments*). This will print the *usage* message, which will contain all the possible flags alongside a short description of each flag.

In some cases (like `ls`), a no-argument command (or *call*) will not print a usage message. If this is the case, checking the *man page* can reveal the possible flags and usually, a more detailed description of the tool than just a usage message. To open a man page for a particular command, just pass in the command name to `man` like this: `man <command>`. For example, to check the man-page for `ls`, enter `man ls`.

1.2 SSH

Bandit runs through SSH (Secure SHell), which is a wrapper term that usually refers to a remote-login service that allows users to login to a specific machine somewhere on the internet and enter commands as that user. SSH runs on Windows, OSX, and UNIX-based systems, (though Windows requires the use of PuTTY instead of a native terminal), allowing an outgoing connection to a hosting computer, or possibly some number of incoming connections.

2 General Tips for Getting Around Bandit

As you progress through levels in Bandit, it may become cumbersome to copy the correct password and open a new terminal or PuTTY session with the correct username/hostname combination. A workaround to this annoyance is to chain SSH sessions together, for example:

```
bandit0@bandit.labs.overthewire.com $ < redacted password >
bandit0@bandit.labs.overthewire.com $ ssh bandit1@localhost
< redacted key and other information >
Are you sure you want to  continue connecting? (yes/no): yes
bandit1@bandit.labs.overthewire.com's password:
```

3 Bandit Hints and Lessons Learned

3.1 Bandit Level 0

This is a great way to learn how to use flags. Make sure you check the man page or usage of the `ssh` command before googling. As a rule of thumb, it's better to check the usage before a man-page, because the information in the usage message should be more concise.

Lesson Learned You learned how to SSH into a remote machine operating on a non-standard port and how to login with specific credentials.

3.2 Bandit Level 0→1

Welcome to the Matrix, Neo! This level requires you to copy and paste text in terminal, you should know that `cntrl + c` and `cntrl + v` do not usually work. However, the middle-mouse button will paste the selected text to where your cursor (blinking box) is.

Lesson Learned You learned how to read a text file, you'll do this a lot!

3.3 Bandit Level 1→2

Here is where it gets interesting, click around the OverTheWire page on this level to get better tips.

Lesson Learned You learned (a) common character used for I/O operations!

3.4 Bandit Level 2→3

Another filename trick, google around a little for a nice trick!

Lesson Learned You learned (hopefully) about tab-completion! As you get faster with terminal, you'll end up using tab-completion much more.

3.5 Bandit Level 3→4

Take a look at the introduction in this document if you aren't sure how to find a hidden file, or how you might learn the proper parameters or flags yourself.

Lesson Learned Now you learned how to use a (basic but helpful) flag!

3.6 Bandit Level 4→5

Time to use what you learned in 1→2! This "needle in a haystack" is not too bad, searching for the file manually is possible.

Lesson Learned Now you know what a non human-readable file looks like!

3.7 Bandit Level 5→6

Another "needle in a haystack", but this time it's not especially easy to manually search. There are 20 directories, each with about 7 or more files, which leads to 140 files to read! Look at the find command (GNU find, by the way), and keep your eye out for block sizes. The size you are passing through find may not be the size you want, bytes are not the only unit used in filesystems.

Lesson Learned You either learned about the find command (very helpful) and some helpful flags, or you spend a long time checking through every file.

3.8 Bandit Level 6→7

This "needle" is hidden somewhere, but you *really* can't manually find it this time! Since it's somewhere on the server, the best place to start a find from is the *root*, which in Unix based systems is termed with the single forward-slash '`/`'. Keep in mind that the flags for this find may be different, but the `-exec` flag may be extremely useful!

Lesson Learned You learned how to combine several flags to get a very specific result. You also may have learned that grep only works on stdout, and stderr will still be displayed alongside stdout!

3.9 Bandit Level 7→8

Another needle, but this time it's in a huge file, with about 98,000 lines! Even using a utility like less, you'll have a really hard time finding the one unique line by hand, so use your best Google-fu to find a simple solution.

Lesson Learned You learned how to filter text, and hopefully how to pipe i/o streams.

3.10 Bandit Level 8→9

This file is a lot smaller – only about 1,000 lines. But the garbled, *unsorted* passwords are harder to track. If you haven't already, look up pipes in UNIX (you're looking for this: |), they're one of the best tools to have on your belt.

Lesson Learned You learned how to pipe i/o streams.

3.11 Bandit Level 9→10