**CYBR430, Penetration Testing and Incident Response  
Week 6 Lab – grep and regular expressions**

Complete the below actions, answering the questions (**in red**). This week provide screen dumps with your answers.

This week we will be returning to the basic scans we did in week four. This time rather than scrolling through the results in a terminal window (or GUI) we will be capturing the output of commands you enter in the terminal window (no GUI this week) in a text file. We will then search that data using tools such as grep and regular expressions also known as regex. HAL has a small network but when you are scanning a network with hundreds of hosts with thousands of ports open you will find this a useful skill. Refer back to your resource list for this week as well conduct your own research while completing this lab.

This week’s lab will be conducted on your Kali vm in the toxic lab. Log into your Kali vm.

Let’s start by changing your current directory to make sure you are putting files on your desktop where you can find them.

Open a terminal window and enter the following command:

**cd ~/Desktop**

The ~ character is a shortcut to your home directory. By using the cd (change directory) command you are changing you current directory to your home desktop. Remember linux commands and paths are case sensitive. You can confirm this is your location by looking at the prompt in the terminal or enter the command to print your working directory – pwd

**pwd**

**What is returned by the pwd command?**

One of the things you found in week four was the tool Sparta provides lots of good information on smb shares. You did have to scroll through the text windows to find information you were looking for, you may even have missed some valuable information. You will find that many tools such as Sparta use shell scripts (remember the topic for the week?) to do their commands. It’s usually possible to call these scripts outside of the specific tool and sometimes you may want to do this for specialized searches or to combine scripts from different tools into your own custom scan. That’s what we are going to do this week. As always in linux there are many ways to complete some of these tasks. We are going to walk through one way this week but don’t be surprised if you come across others.

Our first step will be to capture scan data from multiple scans and combine it into a single file. The script we will be running is called smbenum.sh and it’s located in /usr/share/sparta/scripts/. To conduct a scan on the file server you found enter the following at a command prompt:

**/usr/share/sparta/scripts/smbenum.sh 10.19.99.5**

Did you catch all that as it scrolled by? We need to dump the output into a file. To do that enter the command:

**/usr/share/sparta/scripts/smbenum.sh 10.19.99.5 > HAL\_scan**

The character ‘>’ is a output redirect. It tells the terminal to put the output of smbenum.sh into the file HAL\_scan. Since you changed your directory to the Desktop you should see the file on your desktop. You can also use the terminal to list your directory and you should see the file. Enter the following command to list your current directory in a long format to show permissions and file size:

**ls –l**

**Provide a screen shot of your directory listing showing the HAL\_scan file**

You can see what is in the file by using the cat command. Enter the following:

**cat HAL\_scan**

The cat command will scroll the entire file. You can also use the more command to scroll a page at a time. The space bar advances the file and hitting q will exit the view.

**more HAL\_scan**

Now we will add additional scans onto that same file. To concatenate information on the end of the file use ‘>>’ rather than ‘>’. Enter the following command to add another scan (by the way you can use the up arrow and down arrow to scroll through your command history to save on typing):

**/usr/share/sparta/scripts/smbenum.sh 10.19.99.10 >> HAL\_scan**

**Provide another screen shot of your directory listing showing the HAL\_scan file and its file size. You should note the file size has gotten larger.**

Let’s complete the rest of the HAL scans and append them to the scan file. Remember you can scroll through your command history to save some typing.

**/usr/share/sparta/scripts/smbenum.sh 10.19.99.12 >> HAL\_scan**

**/usr/share/sparta/scripts/smbenum.sh 10.19.99.14 >> HAL\_scan**

**/usr/share/sparta/scripts/smbenum.sh 10.19.99.16 >> HAL\_scan**

**/usr/share/sparta/scripts/smbenum.sh 10.19.99.18 >> HAL\_scan**

**Provide another screen shot of your directory listing showing the HAL\_scan file and its file size. You should note the file size has gotten larger again.**

Now that we have our scan complete we need to find a list of usernames, a list of host names, and a list of shares. Let’s start with usernames.

The format of the grep command we will be using is **grep regex file.** So the command grep, followed by the regular expression, followed by the file we want to scan. One of the tricks to being successful with grep and regex is looking for patterns, regex is all about patter matching. This means you may need to look at a sample of your file first to determine any identifying patterns you can use. Take a look at your scan file – remember you can do this with’ cat HAL\_scan’ or ‘more HAL\_scan’. You will notice at one point in the scan file usernames always are on their own line and the line starts with “user:” This means we can search on that. This one is pretty simple. The special character ‘^’ is called an anchor and it denotes something occurring at the start of a line. So to find the usernames in our scan we can use the following command.

**grep ‘^user:’ HAL\_scan**

This tells grep to search the file HAL\_scan for the word user: occurring at the start of a line and to print out that line.

**Enter the command to search your scan file for usernames and provide a screen shot of the results.**

Now let’s try something a bit more difficult. We want to find the host names. Again looking at the file we can see that host names always occur on a line with the words ‘File Server Service’ not at the start of a line. The words ‘File Server Service’ are a pattern in and of themselves. We can search for that now.

**grep ‘File Server Service’ HAL\_scan**

Try that command now. When you look at the output you will see something like the below

HAL-PC-001 File Server Service

That’s not exactly what we want, we really want the host names only. What we can do is take the output of one grep command and use it as the input for another grep command. We do this with a symbol called a pipe. That symbol is the |, or vertical bar. Placed between two commands it will feed the output of one into the input of the other. Looking at the above again we can see all host names are at the start of a line and are separated from the remainder of the line by white space. That’s our pattern. Our command would be:

**grep ‘File Server Service’ HAL\_scan | grep –Eo ‘^ \S{1,}’**

The part before the pipe you’ve seen before, the output of that part is sent to the second grep. The parameters –E tell grep that what follows needs to be interpreted as a pattern and the parameter o tells grep to not return the whole line but only the part that matches. Let’s break the pattern down

^ The pattern we are looking for starts at the beginning of a line  
\S The pattern is not white space  
{1,} The pattern continues for 1 or more characters

So return everything that isn’t whitespace at the beginning of a line, until you hit whitespace. We don’t have to give a file name to search as the pipe provides the file input.

**Enter the above combination grep command and provide a screen shot of your results**

Finally we need to find the shares. If we look at the file we will see that all shares are listed in the form ‘\\ ip address’. So we need to write a regex we can feed to grep that looks for \\ at the start of a line followed by an IP address. Our command will be of the form:

**grep –E ‘regex’ HAL\_scan**

We now need to build the regex that goes between the ‘’ To start we need to look for two backslashes in a row. If you remember from earlier a backslash is used for special commands such as \S. To look for a backslash we need to use two backslashes. Since we need to find two backslashes in a row we will need to start our regex with four backslashes. ‘\\\\’ We now need to add a pattern for an IP address. An IP address is of the form ‘3 digits . 3 digits . 3 digits . 3 digits

To do that first grouping 3 digits followed by a period the regex would be

\\\\[0-9]{1,3}\.

\\\\ We start with our four backslashes meaning we are looking for two backslashes   
[0-9] Any digit zero to nine  
{1,3} Repeat the preceding 1 to 3 times – so I am looking for 1 to 3 digits, zero to nine  
\. Followed by a period, here again we need to use the backslash as an escape

We now have the two initial backslashes and the first three digits and period. What’s left to complete the regex is 3 digits, a period, 3 digits, a period, and 3 digits.

**Complete the regex and execute the completed grep command. Provide a screen shot of the completed command and output.**