# Nwachukwu Edumanichukwu

# CYBR430, Penetration Testing and Incident Response Module 6 Lab – sed, 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 scanning and enumeration we did in module 4. 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, sed, 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 your assigned virtual environment. 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? /root/DesktopA screenshot of a computer

Description automatically generated with medium confidence**

One of the things you found in module 4 was the tool Legion provides lots of good information on the server message block (SMB) protocol. Targeting some of our pentesting activity on SMB shares makes a lot of sense as we may be able to find system and usernames as well as resources shared on the network. When using Legion you had to scroll through the text windows to find information you were looking for, it’s possible that you may have missed some valuable information. You will find that many tools such as Legion use scripts (remember the topic for the week?) to accomplish their work. It’s possible for us to use these, or similar tools, individually from the command line. We may want to do this for specialized searches or to combine 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 goal in this lab will be to accomplish three things: enumerate host names, enumerate share names, and enumerate usernames from the hosts we have found at Happy Accident Labs. To do this we will use two tools, nbtscan which you haven’t used before, and nmap which we did use earlier. This time we will use a capability of nmap called the nmap scripting engine. We could write our own nmap scripts but in this lab we will use two scripts included with our nmap distribution.

First, let’s attempt to enumerate host names with nbtscan. Our previous scans have shown hosts on the Happy Accident Lab network at these addresses:

* 10.19.99.1
* 10.19.99.5
* 10.19.99.10
* 10.19.99.12
* 10.19.99.14
* 10.19.99.16
* 10.19.99.18

Since we are enumerating through use of the SMB protocol we can only gain results from those systems using SMB. Our earlier port scans showed all these systems, except 10.19.99.1 are using SMB. Therefore, we will not include 10.19.99.1 in our scan scope. We can do a simple scan with nbtscan by entering the command name followed by the IP addresses to be scanned. Enter the below command in your terminal window:

**nbtscan 10.19.99.5-20**

You will notice we are provided a short, useful, list of the NetBIOS names associated with the hosts at the indicated IP addresses. Next we will try to enumerate SMB shares using nmap. For this we use the below command. Note that we tell nmap to use a script through the use of the script parameter followed by the name of the script, followed by the ports to be used by the scan (these are standard SMB ports), followed by the scan range. Enter the following command in your terminal window:

**nmap –-script smb-enum-shares -p 139,445 10.19.99.5-20**

Did you catch all of that as it scrolled by? Probably not. Let’s try enumerating users next. We will use nmap again but with a different script and only port 445. Enter the following command in your terminal window:

**nmap --script smb-enum-users -p 445 10.19.99.5-20**

Another long output listing. Once again there is potentially quite a bit of valuable information here but we may be missing it. Fortunately, there are several tools in Linux we can use to search and extract key information from text files.

Our first step will be to capture scan data from the multiple scans and combine it into a single file. To do that we follow the first command we enter with a ‘>’ character and the name of a file to store the data. Enter the command:

**nbtscan 10.19.99.5-20 > HAL\_scan**

The character ‘>’ is an output redirect. It tells the terminal to put the output of nbtscan 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 fileTimeline

Description automatically generated with medium confidence**

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:

**nmap –-script smb-enum-shares -p 139,445 10.19.99.5-20 >> HAL\_scan**

Finally, let’s complete our custom enumeration by appending the user enumeration to the scan file.

**nmap –script smb-enum-users -p 445 10.19.99.5-20 >> 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.Text

Description automatically generated**

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 that we have our scan complete, we need to find a list of host names, a list of shares, and a list of usernames. You could use the cat and more commands to find the information you require, and if this was the only time you were going to do these enumerations that may be a good process. In our pentests we will be using these tools many times and building custom searches to extract the needed information is much more efficient. The key to searching is to look for patterns and unique characters in your data. Let’s start by extracting the Happy Accident Labs host names which were found using nbtscan.

If you look at the first page of the HAL\_scan file (remember, you use the ‘more’ command to do that) you see this:

A picture containing text, black, plaque, screenshot

Description automatically generated

The host names are listed here in a table, starting with the heading ‘IP address’ and ending with a line that is the start of the next scan we did ‘Starting Nmap…’ We can use this pattern in a program called sed. We will ask sed to look for a line that starts “IP address” and print everything up to a line that says “Starting Nmap.” Let’s look at the command format we will use

**sed -n ‘/IP address/, /Starting Nmap/ {/Starting Nmap/!p}’ HAL\_scan**

sed is the name of the program

-n tells sed to not use its default printing behavior but to let us specify what to print

‘/IP address/, /Starting Nmap/ {/Starting Nmap/!p}’ We enclose the pattern we are looking for to sed Inside the single quotes. The actual patterns to look for are enclosed inside //.

/IP address/ tells sed a line with IP address will start the selected text.

/Starting Nmap/ tells sed a line with Starting Nmap marks the end of the selected text.

{/Starting Nmap/!p} Remember we told sed to ignore it’s default printing behavior with the -n parameter? This line tells sed to not print the line with Starting Nmap. Not being told otherwise it will print the line starting the search pattern.

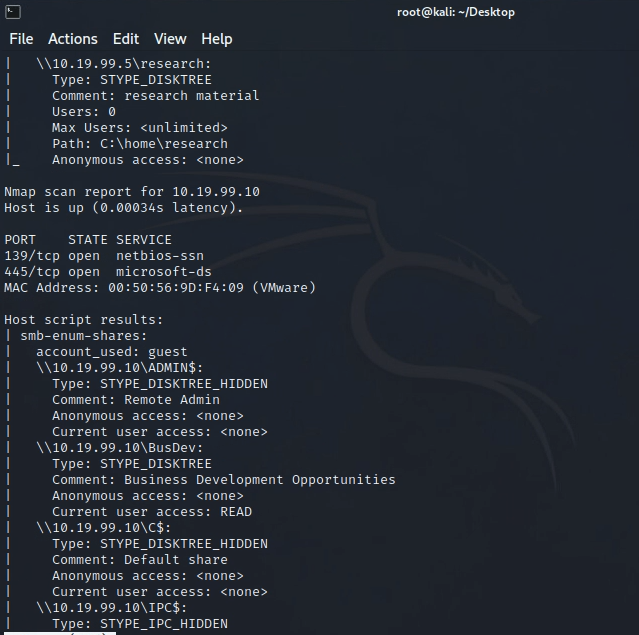
It’s OK if this seems complex. sed is a very powerful program and there are entire chapters and books dedicated to all its features. For now, just know it can help you search on very specific patterns and is a good tool to use for that purpose. Let’s see how well it works. Enter the below command in your terminal:

**sed -n ‘/IP address/, /Starting Nmap/ {/Starting Nmap/!p}’ HAL\_scan**

**Provide a screen shot of the results of this command.Graphical user interface, text

Description automatically generated**

Next we want to find file shares in the scan results. Using the more command to look at our full scan results file we see this pattern



Notice that each share listed starts with \\ followed by an IP address and the share name. That line and the following two lines which state the type of share and comments on the share are valuable to us. We can extract this information using the grep command.

The format of the grep command we will be using is **grep parameters regex file.** So, the command grep, followed by parameters, followed by the regular expression, followed by the file we want to scan. REGEX or regular expressions are a way to specify search patterns. We are going to keep this one simple.

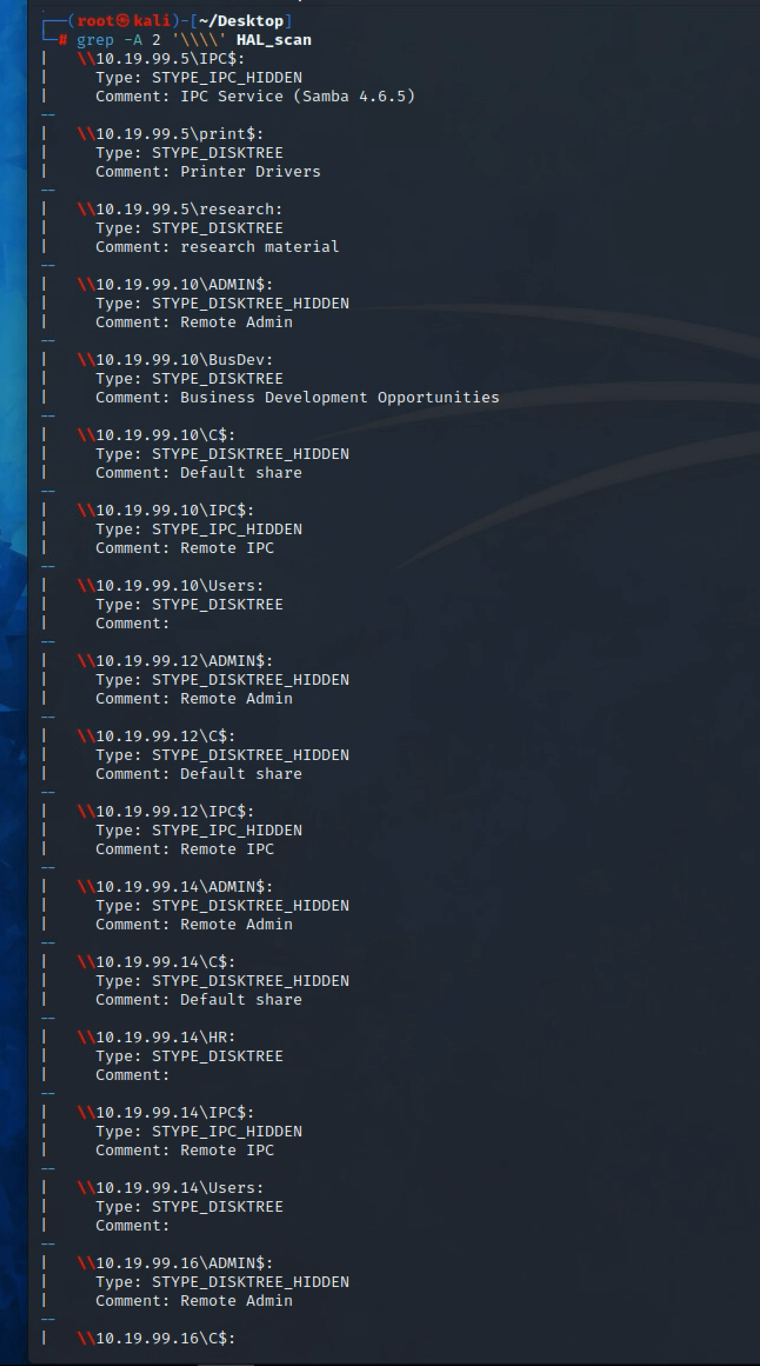
grep -A 2 ‘\\\\’ HAL\_scan

grep is the name of the command

-A 2 tells grep to print the line that matches the search and the two lines after

‘\\\\’ We want to tell grep to look for a line with \\ in it. But the character \ has special meaning in REGEX so we need to tell grep to ignore the special meaning. To tell grep to ignore the special meaning we precede the character with the escape character which is \. That may be confusing, but it just so happens that the character we need to ignore is the same character that tells grep to ignore characters. This means that the first pair \ tells grep to ignore the special meaning and treat what follows as a normal character which is \. The second pair of \ tells grep the same. This means grep will look for \\

Let’s give that command a try for displaying shared. Enter the below command in your terminal:

grep -A 2 ‘\\\\’ HAL\_scan  
**Provide a screen shot of the results of this command.**

We’ve extracted the host names and share names. Now we want to extract usernames. Looking at our capture file once again we see a pattern for discovered usernames.

* The first line of the information we want has the text smb-enum-users
* The last line of the information we want has two unique characters, a vertical bar and an underscore, like this: |\_
* We would like our output to include both the above lines.

Very similar to the earlier example the command we would use to do this is:

**sed -n ‘/smb-enum-users/, /|\_/p’ HAL\_scan**

sed is the name of the program

-n tells sed to not use its default printing behavior

‘/smb-enum-users/, /|\_/p’ The starting and ending search pattern, and the p tells sed to print both

Enter the below command in your terminal:

**sed -n ‘/smb-enum-users/, /|\_/p’ HAL\_scan**

**Provide a screen shot of the results of this command.Text

Description automatically generated**

Be sure and keep a copy of these results, they will be needed in future labs.