Log Analysis in Linux and Splunk

Introduction

Objective

Log analysis is one of the most widely used skills for an analyst. Logs can give indicators of what might be happening to systems on the network. Logs can indicate precursors to compromise, can contain information about a compromise, and also can contain information relevant to post exploitation activities. There are various tools that can be used to analyze logs. There are commercial tools like Splunk as well as free tools like awk, gawk, and grep. The important thing is to be able to look at the logs and parse the relevant information that you're looking for regardless of the tool used.

Overview

Splunk is a widely used commercial log aggregation tool. It is a great tool for ingesting data and then being able to help you analyze network incidents. There are other ways to view log files besides using Splunk. For instance, using grep, gawk, and awk can provide you with similar log parsing results, but those are more arduous methods and those tools are more command line based and require the analyst to remember many various commands and switches.

Outcomes

In this lab, you will learn to:

1. Use Linux commands to search Linux logs.
2. Use Splunk to analyze network traffic and logs.

Key Term	Description
cat	A Linux command used to show the output of data

Key Term	Description
Splunk	A commercial tool with the ability to analyze large log files
grep (global regular expressions print)	A Linux tool to parse information
Leafpad	A GUI text editor program for Linux (similar to Windows Notepad)
tail	The command to show the last few lines of a file in Linux

Reading Assignment

Introduction

Log analysis is one of the most widely used skills for an analyst. Logs can give indicators of what might be happening to systems on the network. Logs can:

- Indicate precursors to compromise
- Contain information about a compromise
- Contain information relevant to post exploitation activities

There are various tools that can be used to analyze logs. There are commercial tools like Splunk as well as free tools like the Linux commands awk, gawk, and grep. The important thing is to be able to look at the logs and parse the relevant information that you're looking for regardless of the tool used.

Figure 1 shows the lab topology.

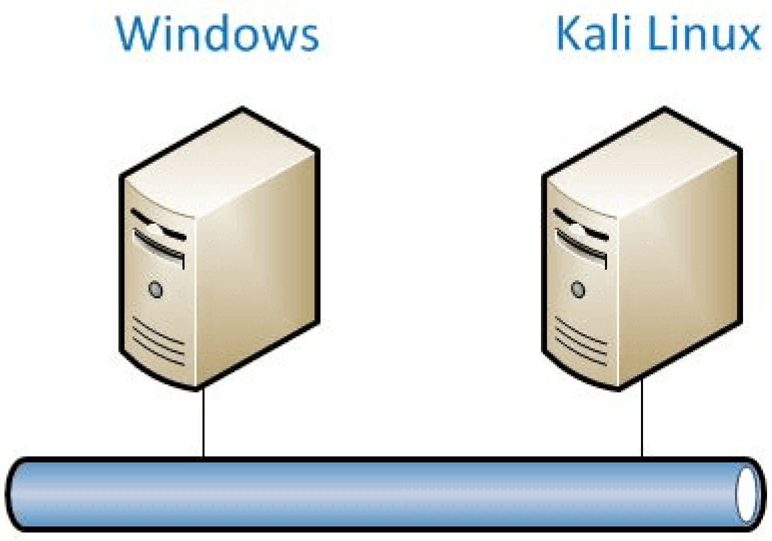


Figure 1 - Lab Topology

In this lab, you will learn to:

- Use Linux commands to search Linux logs.
- Use Splunk to analyze network traffic and logs.

Linux Log Analysis

Linux keeps a large number of log files that keep track of events on the system. One of the most important log files is the log "secure" which logs security incidents. The secure file is located in /var/log on a Red Hat variant of Linux. The Ubuntu/Debian equivalent to secure is auth.log. This log file tracks the following information:

- Creation of new user accounts
- Creation of new group accounts
- Logging on from a remote system
- Changing a user's password

Another important file on a Linux system is the access_log file, which keeps track of web server connections. In Linux, the most common web server used is Apache.

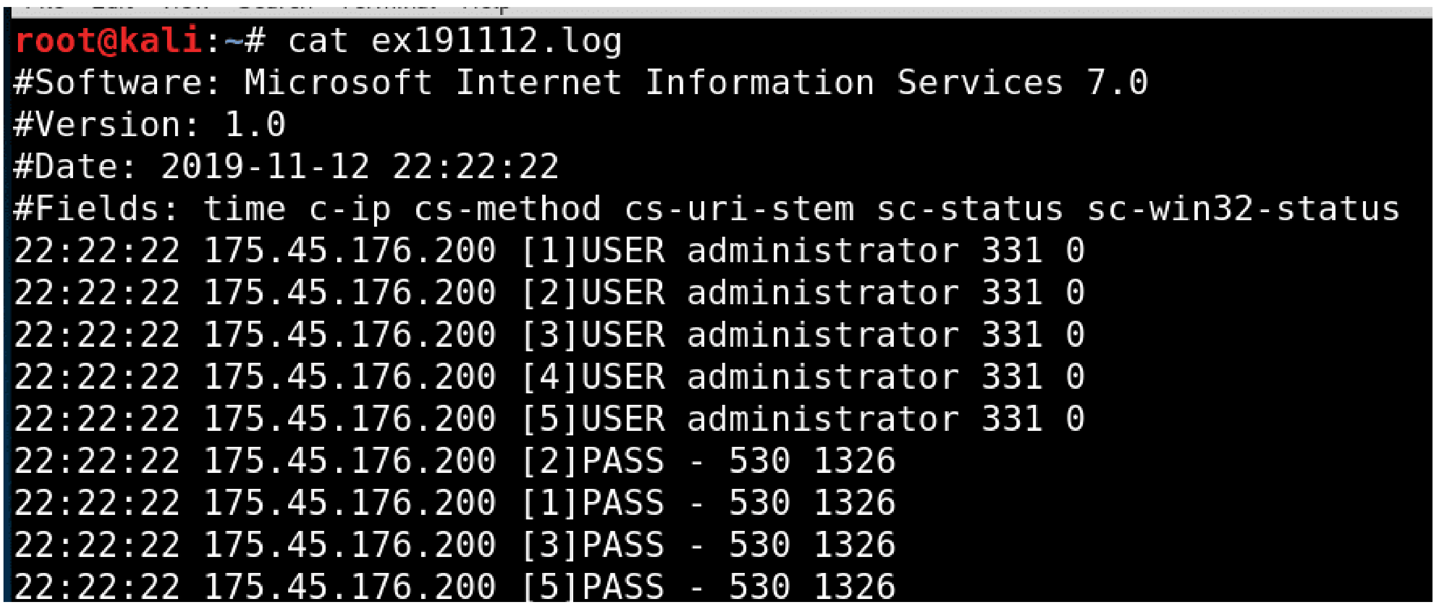
The Apache Server keeps records of the connections made to the website, including:

- IP addresses
- User agents
- Date/timestamps

In this lab, you will use various utilities to search through logs in Linux.

cat

The cat command displays the content of files to the screen. It works well with small files but will scroll off the screen for larger files. The command cat is short for concatenate. Figure 2 shows a cat example.

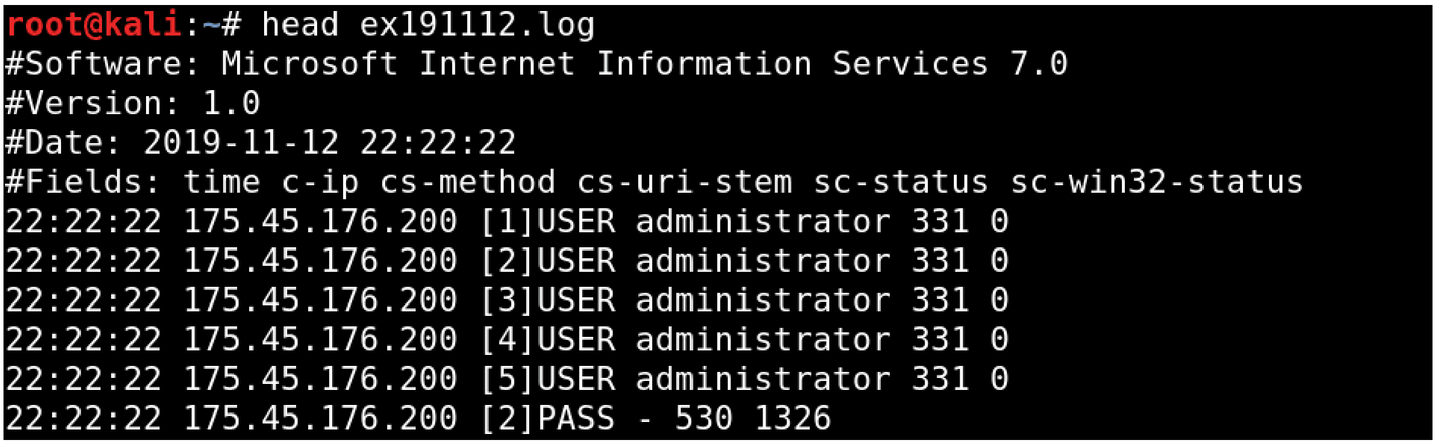


```
root@kali:~# cat ex191112.log
#Software: Microsoft Internet Information Services 7.0
#Version: 1.0
#Date: 2019-11-12 22:22:22
#Fields: time c-ip cs-method cs-uri-stem sc-status sc-win32-status
22:22:22 175.45.176.200 [1]USER administrator 331 0
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [3]USER administrator 331 0
22:22:22 175.45.176.200 [4]USER administrator 331 0
22:22:22 175.45.176.200 [5]USER administrator 331 0
22:22:22 175.45.176.200 [2]PASS - 530 1326
22:22:22 175.45.176.200 [1]PASS - 530 1326
22:22:22 175.45.176.200 [3]PASS - 530 1326
22:22:22 175.45.176.200 [5]PASS - 530 1326
```

Figure 2 - cat Command

head

The head command displays the top 10 lines of a file by default. Different switches can be used with head to show a different number of lines at the beginning of a file. Figure 3 shows a head example.

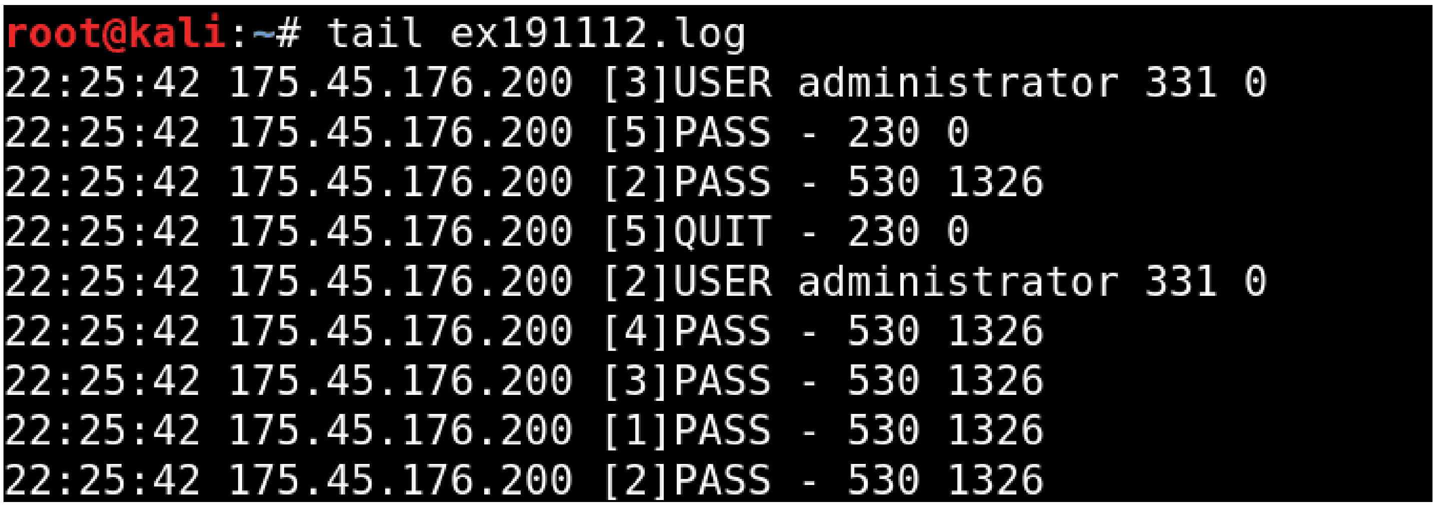


```
root@kali:~# head ex191112.log
#Software: Microsoft Internet Information Services 7.0
#Version: 1.0
#Date: 2019-11-12 22:22:22
#Fields: time c-ip cs-method cs-uri-stem sc-status sc-win32-status
22:22:22 175.45.176.200 [1]USER administrator 331 0
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [3]USER administrator 331 0
22:22:22 175.45.176.200 [4]USER administrator 331 0
22:22:22 175.45.176.200 [5]USER administrator 331 0
22:22:22 175.45.176.200 [2]PASS - 530 1326
```

Figure 3 - head Command

tail

The tail command displays the last 10 lines of a file by default. You can also select the number of lines at the end of a file using the -n option. Figure 4 shows an example of the tail command.

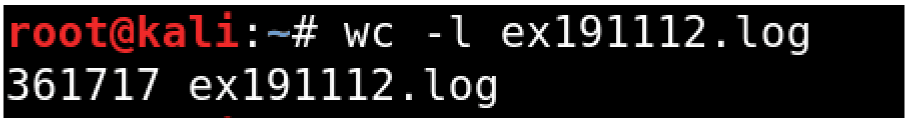


```
root@kali:~# tail ex191112.log
22:25:42 175.45.176.200 [3]USER administrator 331 0
22:25:42 175.45.176.200 [5]PASS - 230 0
22:25:42 175.45.176.200 [2]PASS - 530 1326
22:25:42 175.45.176.200 [5]QUIT - 230 0
22:25:42 175.45.176.200 [2]USER administrator 331 0
22:25:42 175.45.176.200 [4]PASS - 530 1326
22:25:42 175.45.176.200 [3]PASS - 530 1326
22:25:42 175.45.176.200 [1]PASS - 530 1326
22:25:42 175.45.176.200 [2]PASS - 530 1326
```

Figure 4 - tail Command

wc

The wc command displays the number of lines, words, and characters of a file. Figure 5 shows an example of the wc command.



```
root@kali:~# wc -l ex191112.log
361717 ex191112.log
```

Figure 5 - wc Command

Regular Expressions

Regular expressions (RegEx) are defined as a search pattern that allows you to do powerful searches and replacements in strings and files. You also can use regular expressions for input validation in programs and web applications. Regular expressions are one of the most powerful tools in a system administrator's toolbox. Once mastered, system administrators can write very powerful scripts to automate repetitive tasks.

History of Regular Expressions

Regular expressions concept arose in the 1950s by American Mathematician Stephen Cole Keene. Keene developed a regular language. Regular expressions are used in programming languages for input validation, search engines, word processors, text editors, and other applications.

grep

The grep (stands for global regular expressions print) command in Linux is very powerful for indexing text files. The grep command is used to search for a search pattern (regular expression) in a file.

Literal Matching

The most basic search using grep is using the following command:

grep <search pattern> <file-name>

For example, to find the root in the /etc/passwd file, you would use the following command: grep root /etc/passwd. You will get back all the instances of root in the lines in the /etc/passwd file. This is what is called literal matching.

Boolean "or"

Let's say you want to search for root or mail. You would use the following command: grep 'root\|mail' /etc/passwd. This will return all the instances of root or mail in /etc/passwd. You use \| for the or and you put the regular expression in single quotes to tell the command that this is a regular expression. Figure 6 shows an example grep using the Boolean "or."

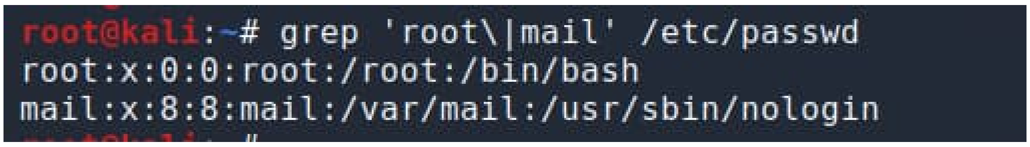

```
root@kali:~# grep 'root\|mail' /etc/passwd
root:x:0:0:root:/root:/bin/bash
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
```

Figure 6 - grep Command

Anchoring

There are meta-characters that tells you where in a line to match. For example, the command grep '^root' /etc/passwd will only return items that have root in the beginning of the string. The dollar sign ($) is another meta-character. With the dollar sign meta-character, grep will match anything at the end of a string. So, the command grep 'bash$' /etc/passwd will match strings with bash at the end. Now, if you use ^$ this will match empty lines in a file such as grep '^$' /etc/passwd. In the case that there are no empty lines in /etc/passwd, so nothing will return.

Single Character Matching

The period is a meta-character that allows you to match any single character. For example, the command grep ‘r.0t’ /etc/passwd will match any character in the second position of the search pattern where the period is located.

Bracket Expressions

Bracket expressions allows you to match a group of characters by putting them into []. For example, if you want to match all strings that have either a small d or a capital D, you could use grep ‘[dD]’ /etc/passwd. This will return all lines with a small d or capital D in them. If you want to search for all lines that begin with a lowercase letter, you would use this command grep ‘^[a-z]’ /etc/passwd.

There are special bracket expressions that can be used in your regular expressions.

Quantifier	Character Classes
[:alnum:]	Alphanumeric characters
[:alpha:]	Alphabetic characters
[:blank:]	Space and tab
[:digit:]	Digits
[:lower:]	Lowercase letters
[:upper:]	Uppercase letters

Quantification

The following table shows the different quantifiers, descriptions, and regular expression examples.

Quantifier	Description	Example
*	Matches the preceding item zero or more times	‘n*login’
?	Matches the preceding item zero or one time	‘n\?login’
+	Matches the preceding item one or more times	‘no+’
{n}	Matches the preceding item exactly n times	‘[:digit:]{3}’
{n,}	Matches the preceding item at least n times	‘[:digit:]{9,}’
{,m}	Matches the preceding item at most m times	‘[:digit:]{,2}’
{n,m}	Matches the preceding item from n to m times	‘[:digit:]{9,12}’

Grouping

Grouping allows you to group patterns together and reference them as a single entity using (). For example, grep '(\(no\)' /etc/passwd will show everything that has no in the word.

Backslash Expressions

The following table shows special backslash expressions and examples.

Expression	Description
\b	Matches a word boundary
\w	Matches a word
\s	Matches a space

For example, grep '\b[nN]fs\b' /etc/passwd will return anything that is N or n from the term nfs (network file system).

gawk

The gawk command is a program that does pattern matching, record processing, and other forms of text manipulation.

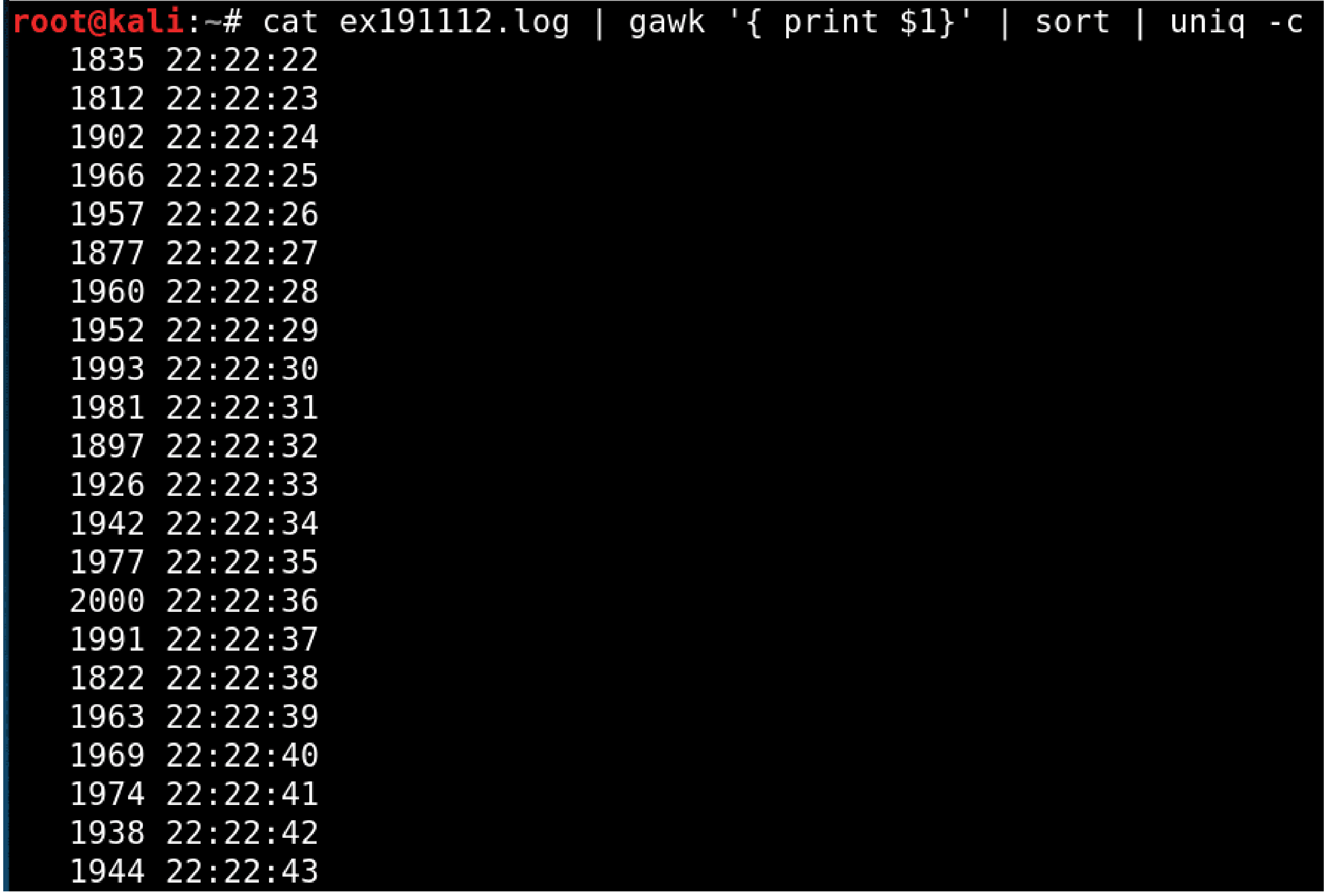
sort

The sort command will sort a list of values in a file.

uniq

The uniq command removes duplicate adjacent links from sorted file or from standard input.

Figure 7 shows an example of using gawk, sort, and uniq. The command will display the total number of unique values in the first column of the log file.



```
root@kali:~# cat ex191112.log | gawk '{ print $1}' | sort | uniq -c
   1835 22:22:22
   1812 22:22:23
   1902 22:22:24
   1966 22:22:25
   1957 22:22:26
   1877 22:22:27
   1960 22:22:28
   1952 22:22:29
   1993 22:22:30
   1981 22:22:31
   1897 22:22:32
   1926 22:22:33
   1942 22:22:34
   1977 22:22:35
   2000 22:22:36
   1991 22:22:37
   1822 22:22:38
   1963 22:22:39
   1969 22:22:40
   1974 22:22:41
   1938 22:22:42
   1944 22:22:43
```

Figure 7 - cat Command

Splunk

Splunk is a log analysis tool that will allow you to ingest several different types of log formats, including web logs, system logs, event viewer files, and Packet Capture Application (PCAP) data. Splunk is a commercial product, but it has a free edition. With the free edition of Splunk, there is a parse limit of 500 MB per day within a 24-hour time frame. Figure 8 shows the log in screen of Splunk.

Splunk helps "capture, index, and correlate real-time data in a searchable repository from which it can generate graphs, reports, alerts, dashboards, and visualizations." Source: Link

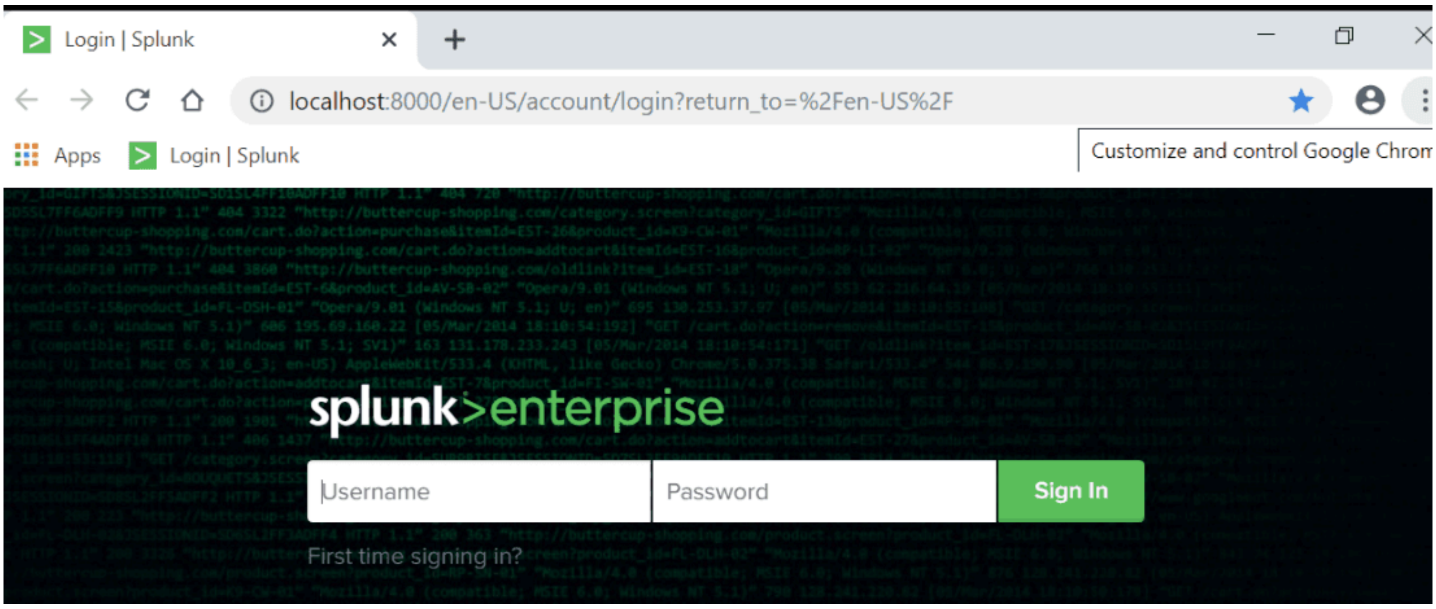


Figure 8 - Splunk

CONCLUSION:

Splunk and Linux command-line log analyses provide the same result, but the Linux results are provided faster. The disadvantage to the Linux method is remembering the use and syntax of the commands. Frequent use and practice with the Linux commands and the available switches will make you more familiar with them. Using Splunk is always another option.

Using the Command Line in Linux to Parse Log files

1. **Click** on the Kali virtual machine. **Log in** with the username of `root` and the password of `toor`.

⚠ Note: The password of **toor** will not be displayed when you type it for security purposes.

2. **Open** a terminal on Kali. **Type** the following command and **press** Enter to get the version of Linux.

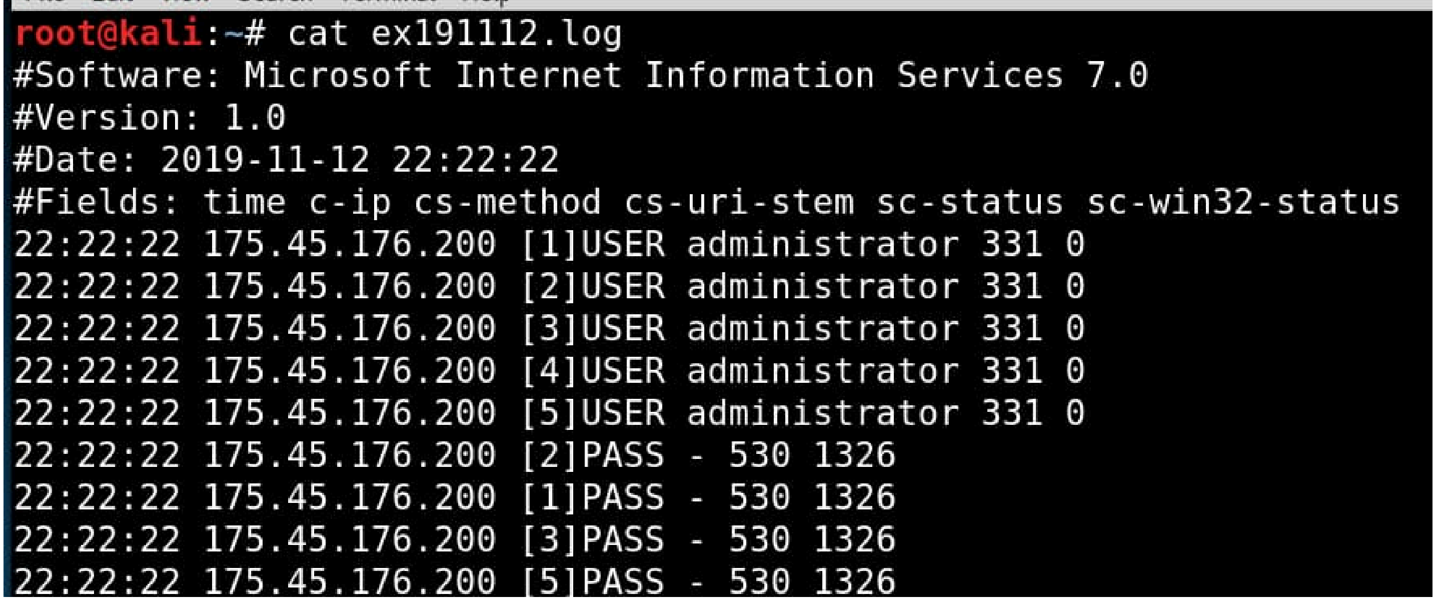
root@kali:~# `uname -a`

3. **Type** the following command and **press** Enter to list files and folders in the root directory.

root@kali:~# `ls`

4. **Type** the following command and **press** Enter to view the log files.

root@kali:~# `cat ex191112.log`



```
root@kali:~# cat ex191112.log
#Software: Microsoft Internet Information Services 7.0
#Version: 1.0
#Date: 2019-11-12 22:22:22
#Fields: time c-ip cs-method cs-uri-stem sc-status sc-win32-status
22:22:22 175.45.176.200 [1]USER administrator 331 0
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [3]USER administrator 331 0
22:22:22 175.45.176.200 [4]USER administrator 331 0
22:22:22 175.45.176.200 [5]USER administrator 331 0
22:22:22 175.45.176.200 [2]PASS - 530 1326
22:22:22 175.45.176.200 [1]PASS - 530 1326
22:22:22 175.45.176.200 [3]PASS - 530 1326
22:22:22 175.45.176.200 [5]PASS - 530 1326
```

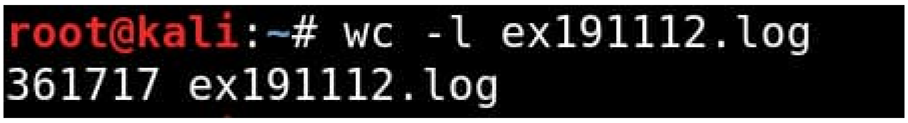
Note: The file is very long and has a lot of lines. We will use the wc command in Linux to determine the number of lines.

5 **Type** the following command and **press** Enter to view the options for wc. wc is a program that displays the line, word, character, and byte count of a file.

root@kali:~# `wc --help`

6 **Type** the following command and **press** Enter to view the number of lines in the file.

root@kali:~# `wc -l ex191112.log`



```
root@kali:~# wc -l ex191112.log
361717 ex191112.log
```

Note: The number to the left of the file indicates that this log file has 361,717 lines. That is a large amount of data.

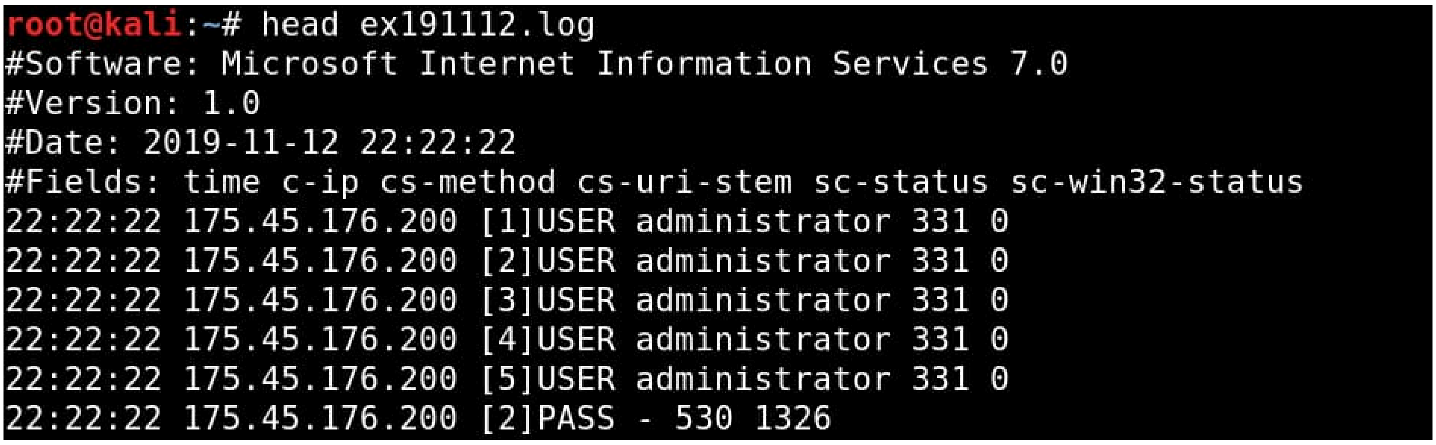
The head command can be used in Linux to display the beginning of a file.

7 **Type** the following command and **press** Enter to view the options for head. head is a program that displays first lines of a file.

root@kali:~# `head --help`

8 **Type** the following command and **press** Enter to view the first 10 lines of the log files.

root@kali:~# `head ex191112.log`



```
root@kali:~# head ex191112.log
#Software: Microsoft Internet Information Services 7.0
#Version: 1.0
#Date: 2019-11-12 22:22:22
#Fields: time c-ip cs-method cs-uri-stem sc-status sc-win32-status
22:22:22 175.45.176.200 [1]USER administrator 331 0
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [3]USER administrator 331 0
22:22:22 175.45.176.200 [4]USER administrator 331 0
22:22:22 175.45.176.200 [5]USER administrator 331 0
22:22:22 175.45.176.200 [2]PASS - 530 1326
```

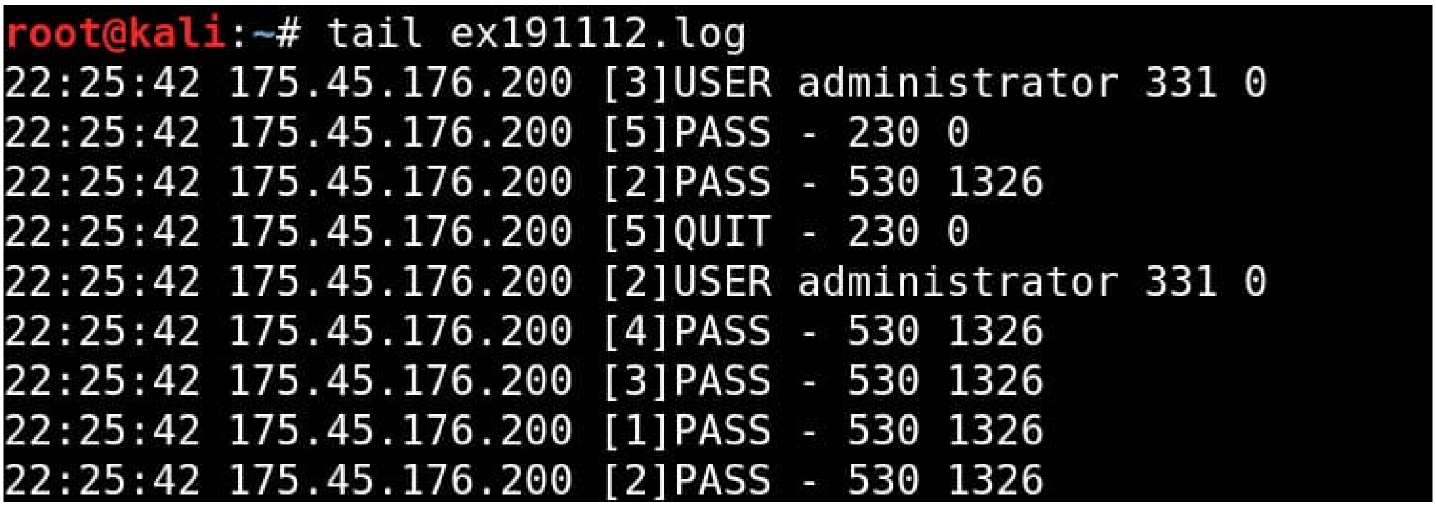
The tail command can be used in Linux to display the end of a file.

9 **Type** the following command and **press** Enter to view the options for tail. tail is a program that displays a number of lines at the end of a file.

root@kali:~# `tail --help`

10 **Type** the following command and **press** Enter to view the last 10 lines of the log files.

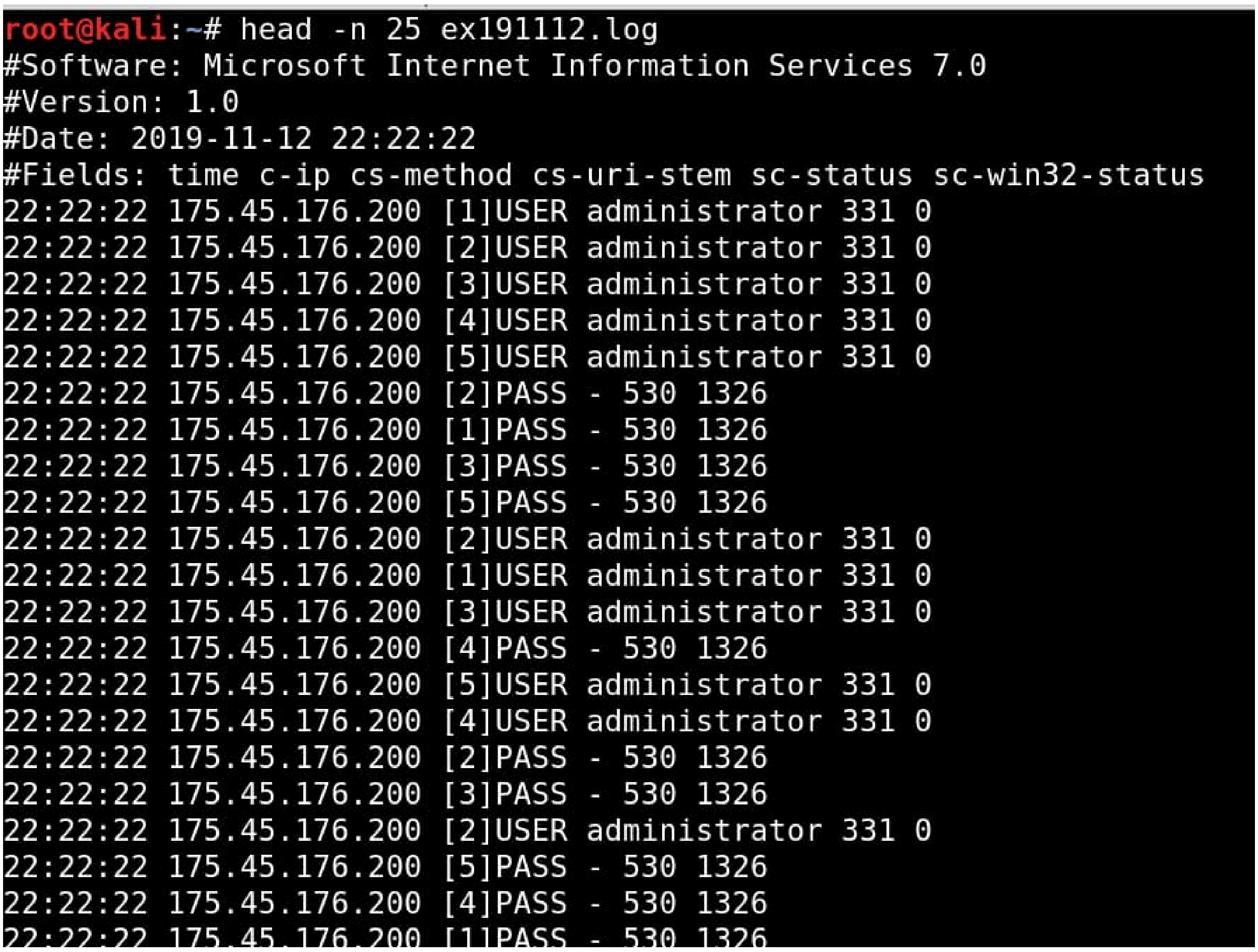
root@kali:~# `tail -n 10 ex191112.log`



```
root@kali:~# tail ex191112.log
22:25:42 175.45.176.200 [3]USER administrator 331 0
22:25:42 175.45.176.200 [5]PASS - 230 0
22:25:42 175.45.176.200 [2]PASS - 530 1326
22:25:42 175.45.176.200 [5]QUIT - 230 0
22:25:42 175.45.176.200 [2]USER administrator 331 0
22:25:42 175.45.176.200 [4]PASS - 530 1326
22:25:42 175.45.176.200 [3]PASS - 530 1326
22:25:42 175.45.176.200 [1]PASS - 530 1326
22:25:42 175.45.176.200 [2]PASS - 530 1326
```

11 **Type** the following command and **press** Enter to view the first 25 lines of the log files.

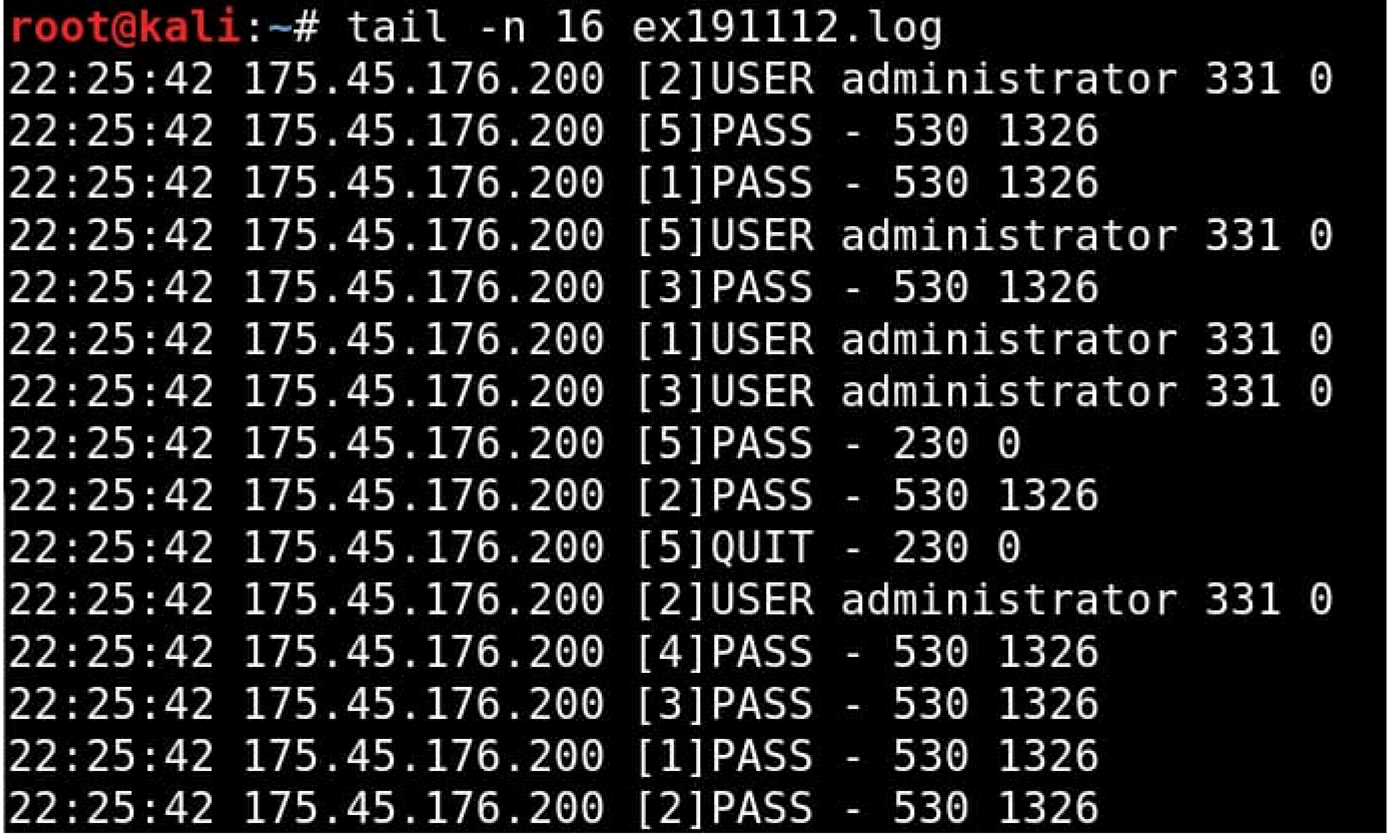
root@kali:~# `head -n 25 ex191112.log`



```
root@kali:~# head -n 25 ex191112.log
#Software: Microsoft Internet Information Services 7.0
#Version: 1.0
#Date: 2019-11-12 22:22:22
#Fields: time c-ip cs-method cs-uri-stem sc-status sc-win32-status
22:22:22 175.45.176.200 [1]USER administrator 331 0
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [3]USER administrator 331 0
22:22:22 175.45.176.200 [4]USER administrator 331 0
22:22:22 175.45.176.200 [5]USER administrator 331 0
22:22:22 175.45.176.200 [2]PASS - 530 1326
22:22:22 175.45.176.200 [1]PASS - 530 1326
22:22:22 175.45.176.200 [3]PASS - 530 1326
22:22:22 175.45.176.200 [5]PASS - 530 1326
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [1]USER administrator 331 0
22:22:22 175.45.176.200 [3]USER administrator 331 0
22:22:22 175.45.176.200 [4]PASS - 530 1326
22:22:22 175.45.176.200 [5]USER administrator 331 0
22:22:22 175.45.176.200 [4]USER administrator 331 0
22:22:22 175.45.176.200 [2]PASS - 530 1326
22:22:22 175.45.176.200 [3]PASS - 530 1326
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [5]PASS - 530 1326
22:22:22 175.45.176.200 [4]PASS - 530 1326
22:22:22 175.45.176.200 [1]PASS - 530 1326
```

12 **Type** the following command and **press** Enter to view the last 16 lines of the log files.

root@kali:~# `tail -n 16 ex191112.log`



```
root@kali:~# tail -n 16 ex191112.log
22:25:42 175.45.176.200 [2]USER administrator 331 0
22:25:42 175.45.176.200 [5]PASS - 530 1326
22:25:42 175.45.176.200 [1]PASS - 530 1326
22:25:42 175.45.176.200 [5]USER administrator 331 0
22:25:42 175.45.176.200 [3]PASS - 530 1326
22:25:42 175.45.176.200 [1]USER administrator 331 0
22:25:42 175.45.176.200 [3]USER administrator 331 0
22:25:42 175.45.176.200 [5]PASS - 230 0
22:25:42 175.45.176.200 [2]PASS - 530 1326
22:25:42 175.45.176.200 [5]QUIT - 230 0
22:25:42 175.45.176.200 [2]USER administrator 331 0
22:25:42 175.45.176.200 [4]PASS - 530 1326
22:25:42 175.45.176.200 [3]PASS - 530 1326
22:25:42 175.45.176.200 [1]PASS - 530 1326
22:25:42 175.45.176.200 [2]PASS - 530 1326
```

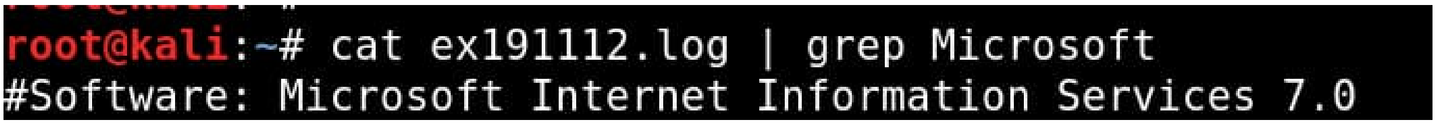
The grep (global regular expressions print) command can be used to parse data for a certain term or search expression.

13 **Type** the following command and **press** Enter to view the options for grep. grep is a program that finds pattern matches in a file.

root@kali:~# `grep --help`

14 **Type** the following command and **press** Enter to view all lines with the Word Microsoft.

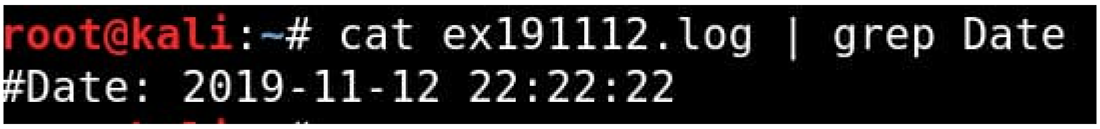
root@kali:~# `cat ex191112.log | grep Microsoft`



```
root@kali:~# cat ex191112.log | grep Microsoft
#Software: Microsoft Internet Information Services 7.0
```

15 **Type** the following command and **press** Enter to view all lines with the word Date.

root@kali:~# `cat ex191112.log | grep Date`

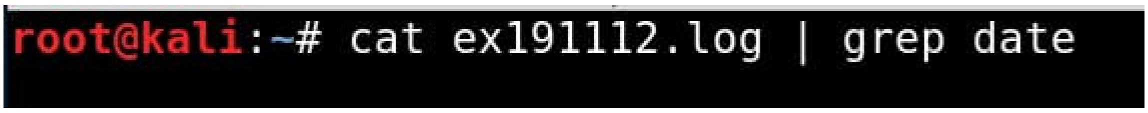


```
root@kali:~# cat ex191112.log | grep Date
#Date: 2019-11-12 22:22:22
```

The grep command is case sensitive. When we change the word date to all lowercase, there are no results.

16 **Type** the following command and **press** Enter to view all lines with the word date.

root@kali:~# `cat ex191112.log | grep date`

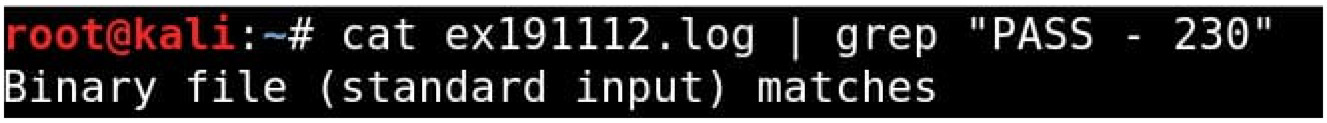


```
root@kali:~# cat ex191112.log | grep date
```

Note: There are no results.

17 **Type** the following command and **press** Enter to grep for the particular expression.

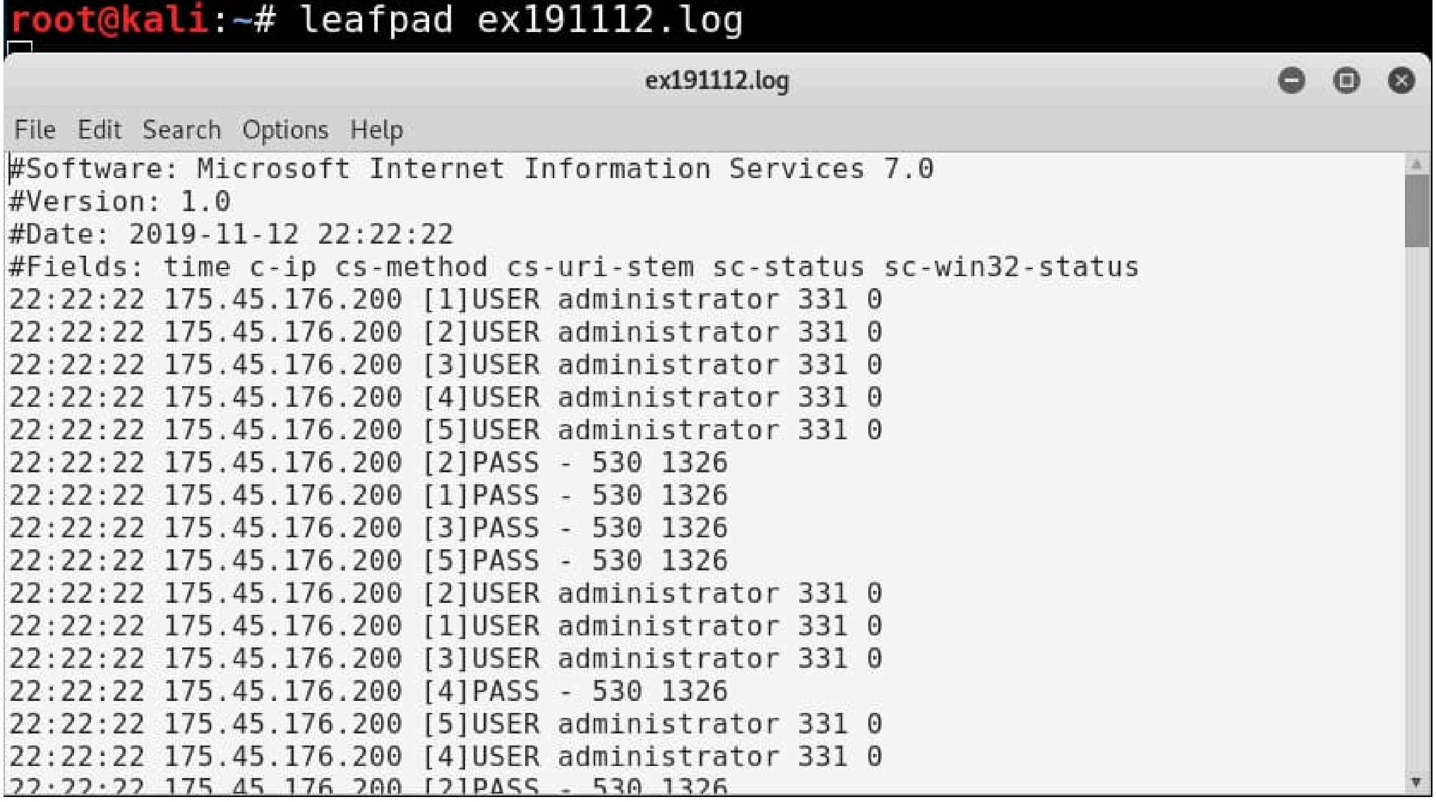
root@kali:~# `cat ex191112.log | grep “PASS - 230”`



```
root@kali:~# cat ex191112.log | grep "PASS - 230"
Binary file (standard input) matches
```

18 **Notice** the error you are receiving. This is due to the information at the beginning of the file. We will need to remove that information to remove the error. **Open** the file using Leafpad. Leafpad is an open-source text editor in Linux.

root@kali:~# `leafpad ex191112.log`



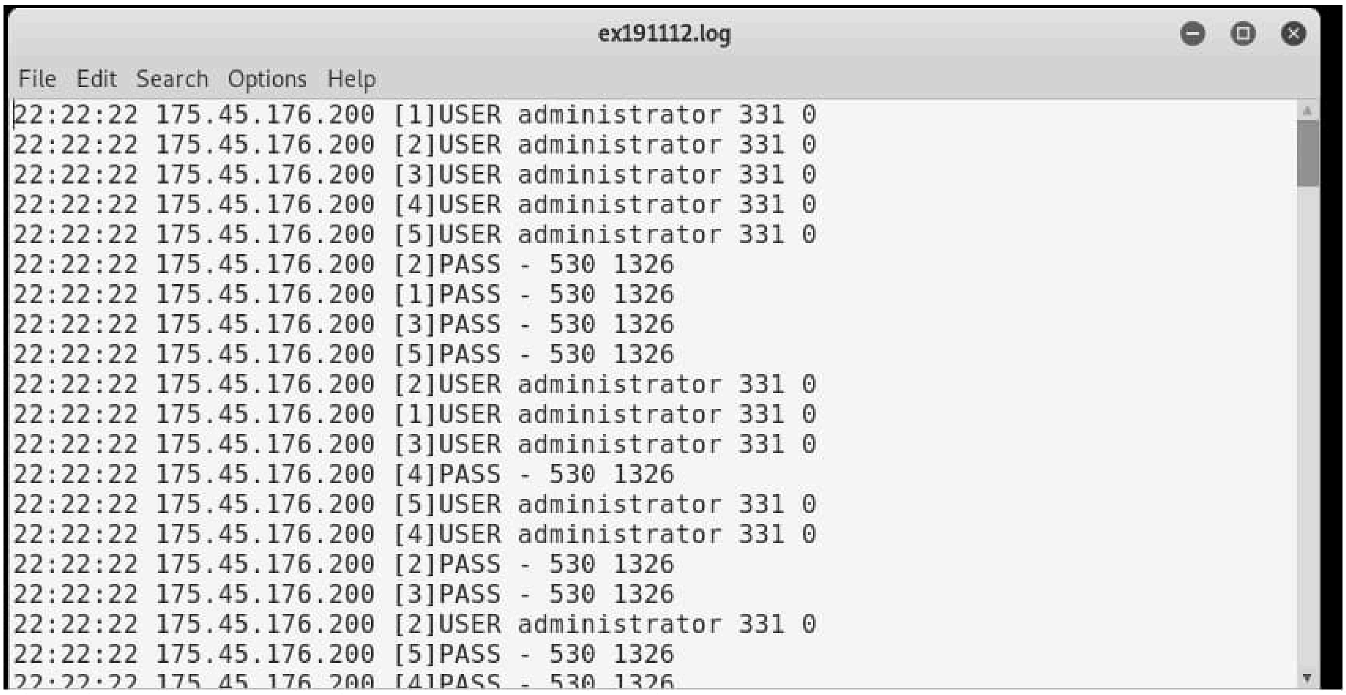
```
root@kali:~# leafpad ex191112.log
```

ex191112.log

File Edit Search Options Help

```
#Software: Microsoft Internet Information Services 7.0
#Version: 1.0
#Date: 2019-11-12 22:22:22
#Fields: time c-ip cs-method cs-uri-stem sc-status sc-win32-status
22:22:22 175.45.176.200 [1]USER administrator 331 0
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [3]USER administrator 331 0
22:22:22 175.45.176.200 [4]USER administrator 331 0
22:22:22 175.45.176.200 [5]USER administrator 331 0
22:22:22 175.45.176.200 [2]PASS - 530 1326
22:22:22 175.45.176.200 [1]PASS - 530 1326
22:22:22 175.45.176.200 [3]PASS - 530 1326
22:22:22 175.45.176.200 [5]PASS - 530 1326
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [1]USER administrator 331 0
22:22:22 175.45.176.200 [3]USER administrator 331 0
22:22:22 175.45.176.200 [4]PASS - 530 1326
22:22:22 175.45.176.200 [5]USER administrator 331 0
22:22:22 175.45.176.200 [4]USER administrator 331 0
22:22:22 175.45.176.200 [2]PASS - 530 1326
```

19 **Remove** the first four lines (as seen below) and then **save** the file. **Close** the log file.



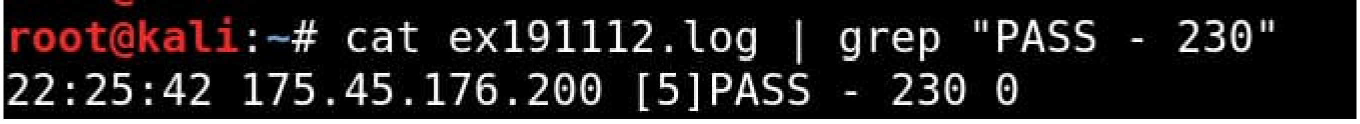
ex191112.log

File Edit Search Options Help

```
22:22:22 175.45.176.200 [1]USER administrator 331 0
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [3]USER administrator 331 0
22:22:22 175.45.176.200 [4]USER administrator 331 0
22:22:22 175.45.176.200 [5]USER administrator 331 0
22:22:22 175.45.176.200 [2]PASS - 530 1326
22:22:22 175.45.176.200 [1]PASS - 530 1326
22:22:22 175.45.176.200 [3]PASS - 530 1326
22:22:22 175.45.176.200 [5]PASS - 530 1326
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [1]USER administrator 331 0
22:22:22 175.45.176.200 [3]USER administrator 331 0
22:22:22 175.45.176.200 [4]PASS - 530 1326
22:22:22 175.45.176.200 [5]USER administrator 331 0
22:22:22 175.45.176.200 [4]USER administrator 331 0
22:22:22 175.45.176.200 [2]PASS - 530 1326
22:22:22 175.45.176.200 [3]PASS - 530 1326
22:22:22 175.45.176.200 [2]USER administrator 331 0
22:22:22 175.45.176.200 [5]PASS - 530 1326
22:22:22 175.45.176.200 [4]PASS - 530 1326
```

20 **Type** the following command and **press** Enter to grep for the particular expression.

root@kali:~# `cat ex191112.log | grep “PASS - 230”`


```
root@kali:~# cat ex191112.log | grep "PASS - 230"
22:25:42 175.45.176.200 [5]PASS - 230 0
```

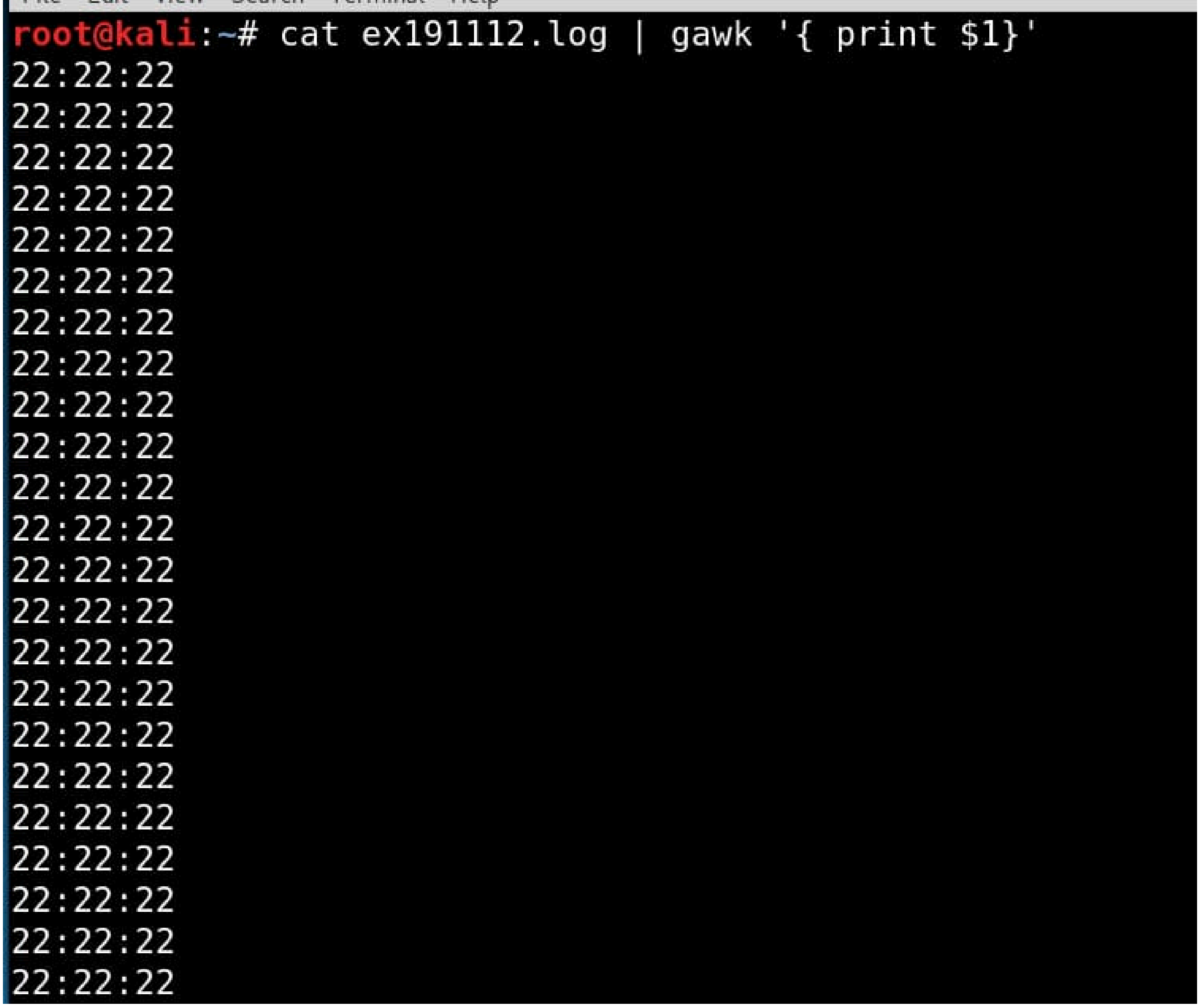
Note: The command line expression gawk can be used to display a particular column of your output. The default delimiter is the space, but you can change that by typing the gawk command at the terminal to see all of the available options.

21 **Type** the following command and **press** Enter to view the options for the gawk command. gawk is a program is a powerful pattern-matching and processing language.

root@kali:~# `gawk --help`

22 **Type** the following command and **press** Enter to use the gawk command and display the first column in the log file.

root@kali:~# `cat ex191112.log | gawk ‘{print $1}’`



```
root@kali:~# cat ex191112.log | gawk '{ print $1}'
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
22:22:22
```

The sort command can be used in Linux to put the output of the columns in order.

23 **Type** the following command and **press** Enter to view the options for the sort command. sort program allows you to print lines in sorted order.

root@kali:~# `sort --help`

The uniq command can be used in Linux to show all of the columns with unique output.

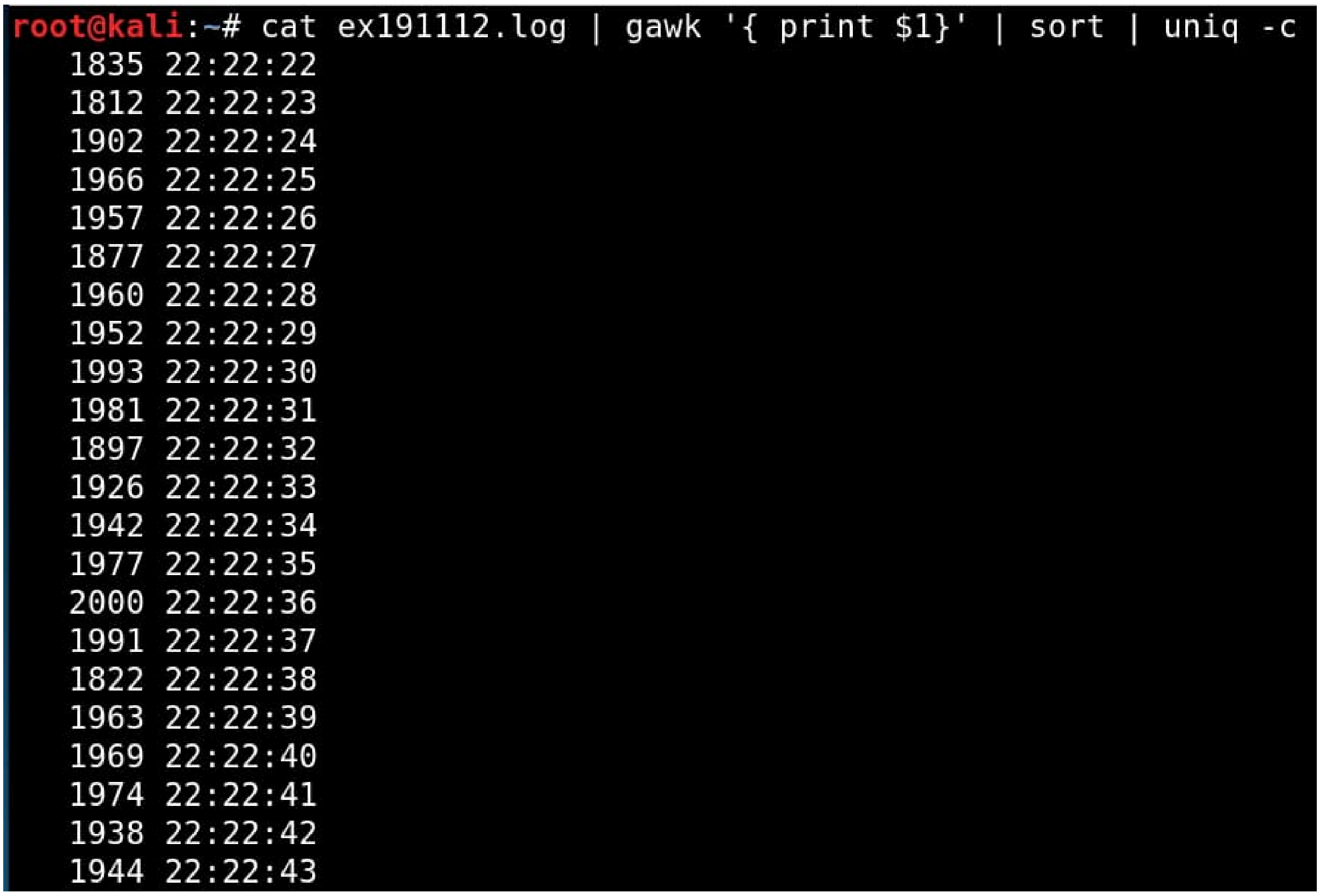
24 **Type** the following command and **press** Enter to view the options for the uniq command. uniq program discards duplicate lines from a listing.

root@kali:~# `uniq --help`

25

Type the following command and **press** Enter to use the gawk command and display the total number of each of the unique values in the first column in the log file.

root@kali:~# `cat ex191112.log | gawk ‘{print $1}’ | sort | uniq -c`

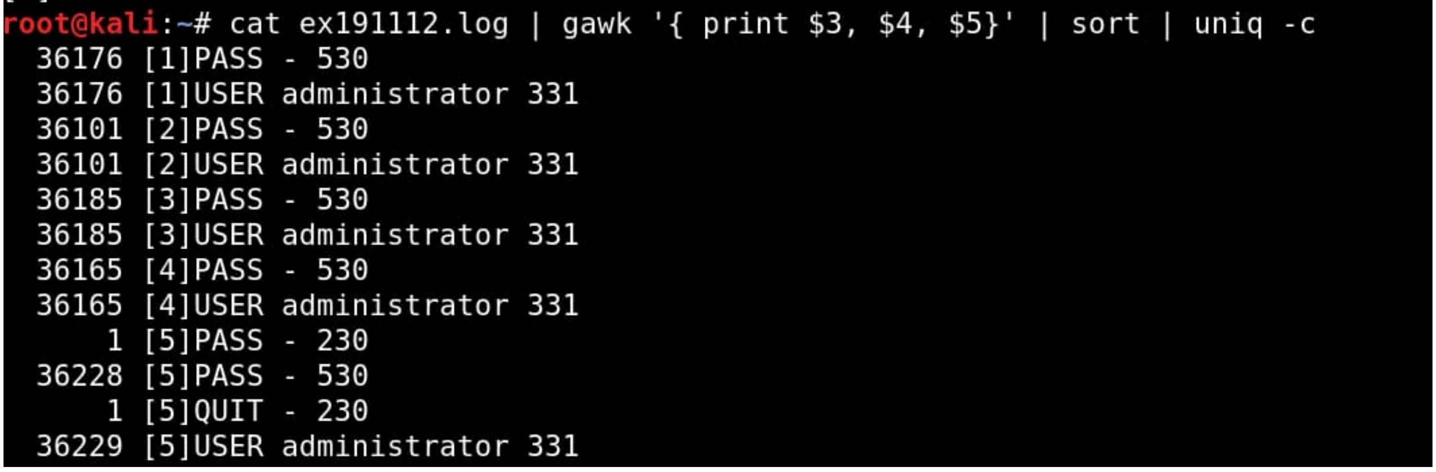


```
root@kali:~# cat ex191112.log | gawk '{ print $1}' | sort | uniq -c
   1835 22:22:22
   1812 22:22:23
   1902 22:22:24
   1966 22:22:25
   1957 22:22:26
   1877 22:22:27
   1960 22:22:28
   1952 22:22:29
   1993 22:22:30
   1981 22:22:31
   1897 22:22:32
   1926 22:22:33
   1942 22:22:34
   1977 22:22:35
   2000 22:22:36
   1991 22:22:37
   1822 22:22:38
   1963 22:22:39
   1969 22:22:40
   1974 22:22:41
   1938 22:22:42
   1944 22:22:43
```

26

Type the following command and **press** Enter to use the gawk command and display the total number of each of the unique values in the three columns in the log file. In Linux, you can send the output of a command to another command using a | character.

root@kali:~# `cat ex191112.log | gawk ‘{print $3,$4,$5}’ | sort | uniq -c`



```
root@kali:~# cat ex191112.log | gawk '{ print $3, $4, $5}' | sort | uniq -c
  36176 [1]PASS - 530
  36176 [1]USER administrator 331
  36101 [2]PASS - 530
  36101 [2]USER administrator 331
  36185 [3]PASS - 530
  36185 [3]USER administrator 331
  36165 [4]PASS - 530
  36165 [4]USER administrator 331
      1 [5]PASS - 230
  36228 [5]PASS - 530
      1 [5]QUIT - 230
  36229 [5]USER administrator 331
```

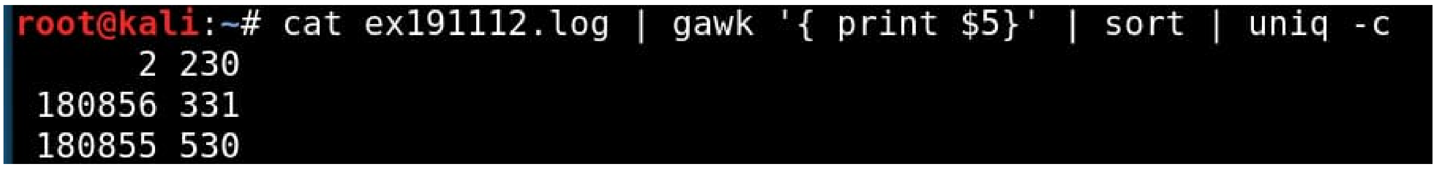
FTP log codes include:

- 331 – attempted log in
- 530 – failure to log in
- 230 – successful login/logout

Here is a reference: https://en.wikipedia.org/wiki/List_of_FTP_server_return_codes

27 **Type** the following command and **press** Enter to use the gawk command and display the total number of each of the unique values in the fifth column in the log file.

root@kali:~# `cat ex191112.log | gawk ‘{print $5}’ | sort | uniq -c`

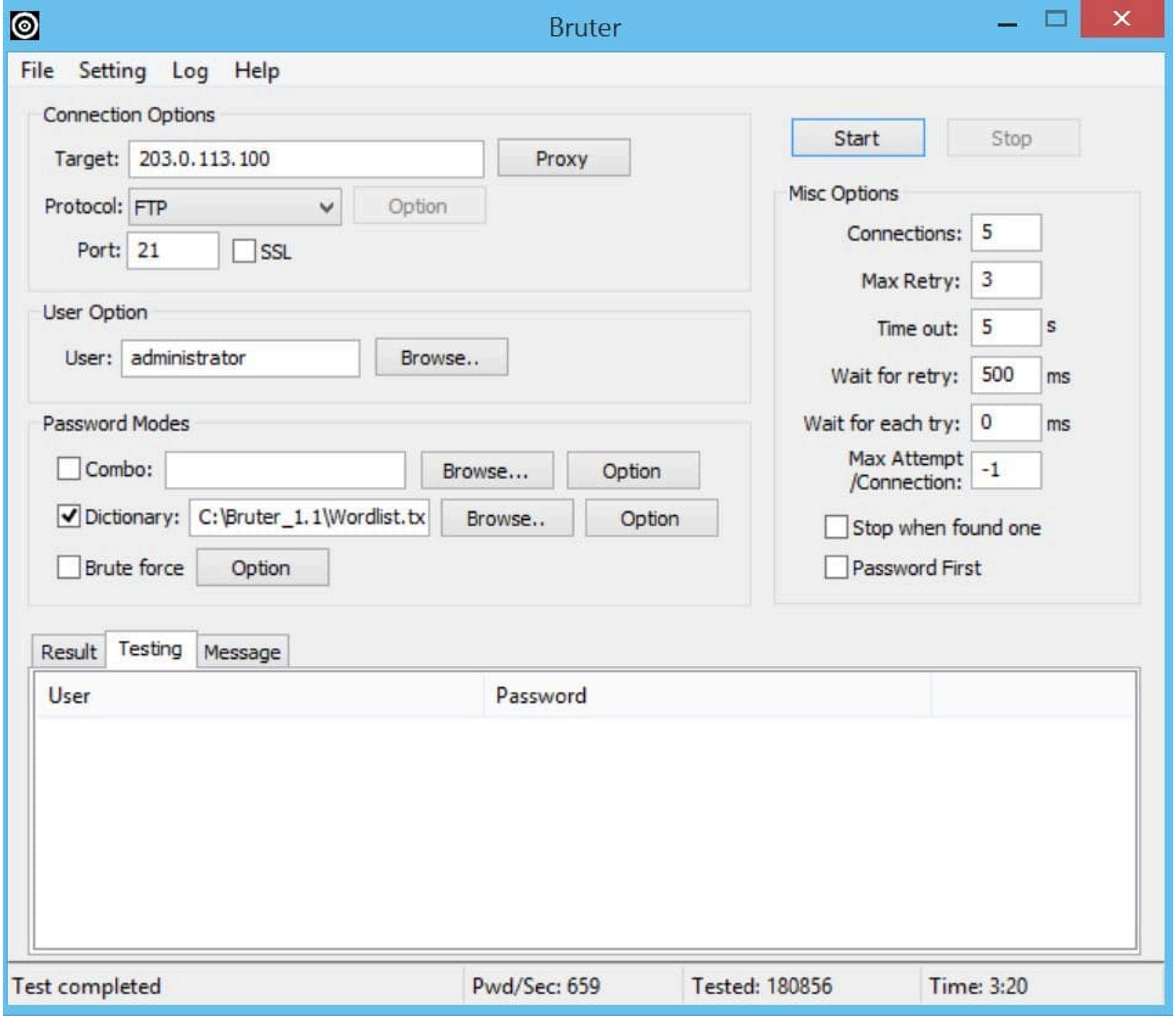


```
root@kali:~# cat ex191112.log | gawk '{ print $5}' | sort | uniq -c
      2 230
 180856 331
 180855 530
```

According to the attacker, there were 180,856 attempted logins by the administrator account.

The 331 code shows this to be the case.

There were 180,855 failed logins shown by the error code of 530.



Connection Options
Target: 203.0.113.100
Protocol: FTP
Port: 21 ☐ SSL

User Option
User: administrator

Password Modes
☐ Combo:
✓ Dictionary: C:\Bruter_1.1\Wordlist.tx
☐ Brute force

Misc Options
Connections: 5
Max Retry: 3
Time out: 5 s
Wait for retry: 500 ms
Wait for each try: 0 ms
Max Attempt /Connection: -1
☐ Stop when found one
☐ Password First

User	Password

Test completed | Pwd/Sec: 659 | Tested: 180856 | Time: 3:20

Summary

Log files can be extremely large. One way to view their output is by using the command line in Linux. The cat command in Linux will allow you to view a file, and the grep command can be used to parse through the information. The head command will show you the first lines in a file, and the tail command will show you the last few lines. Using a combination of the gawk, sort, and uniq commands, you can find out specific information about events in log files.

DISCUSSION QUESTIONS:

1. What is the command to view a file in Linux?

2. What is the command to count the number of lines in a file in Linux?

3. What is the default delimiter in gawk?

4. What is the command to rename a file in Linux?

5 What is the command to print a column in Linux?

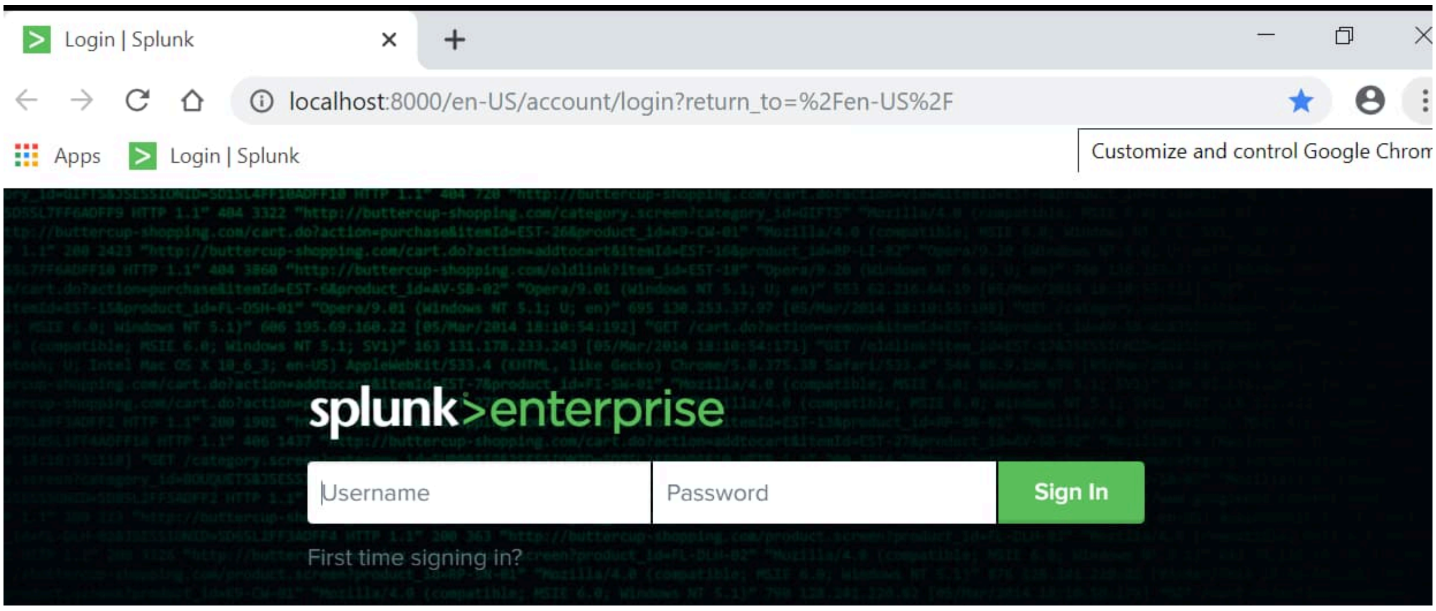
Splunk

Splunk is a log analysis tool that will allow you to ingest several different types of log formats, including web logs, system logs, event viewer files, and PCAP data. Splunk is a commercial product, but it has a free edition. With the free edition of Splunk, there is a parse limit of 500 MB per day within a 24-hour time frame.

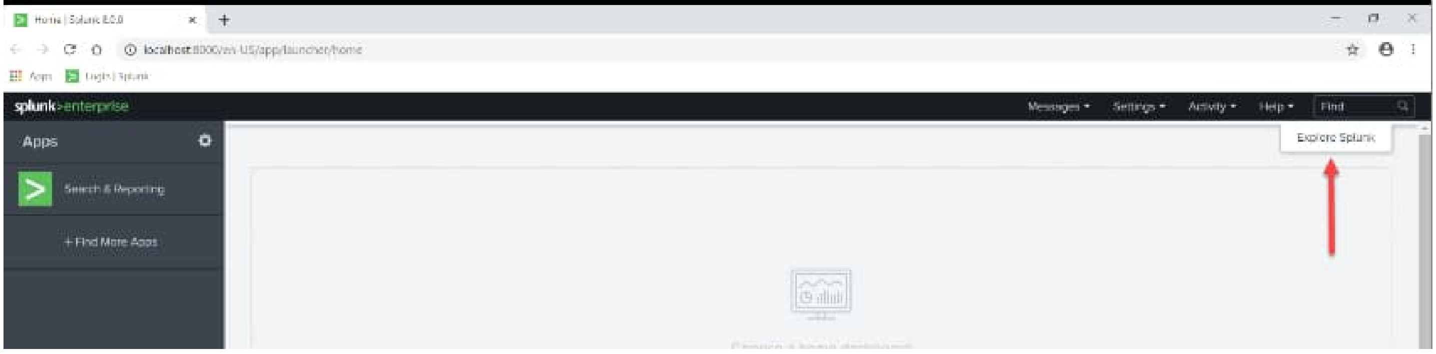
1 **Log on** to the Windows 10 VM as Student with the password of `P@ssw0rd`. **Double-click** the Chrome icon on the desktop.

Note: The password of **P@ssw0rd** will not be displayed when you type it for security purposes.

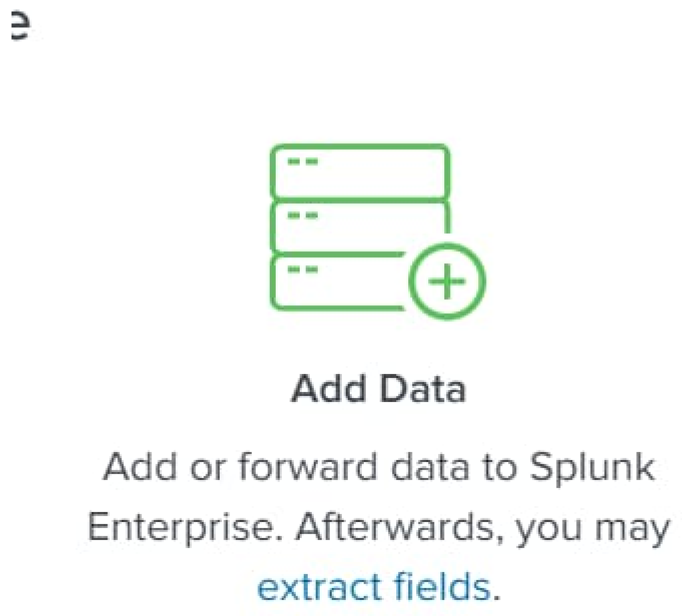
2 You will automatically be logged in to Splunk.



3 **Click** Explore Splunk in the top right of the window.

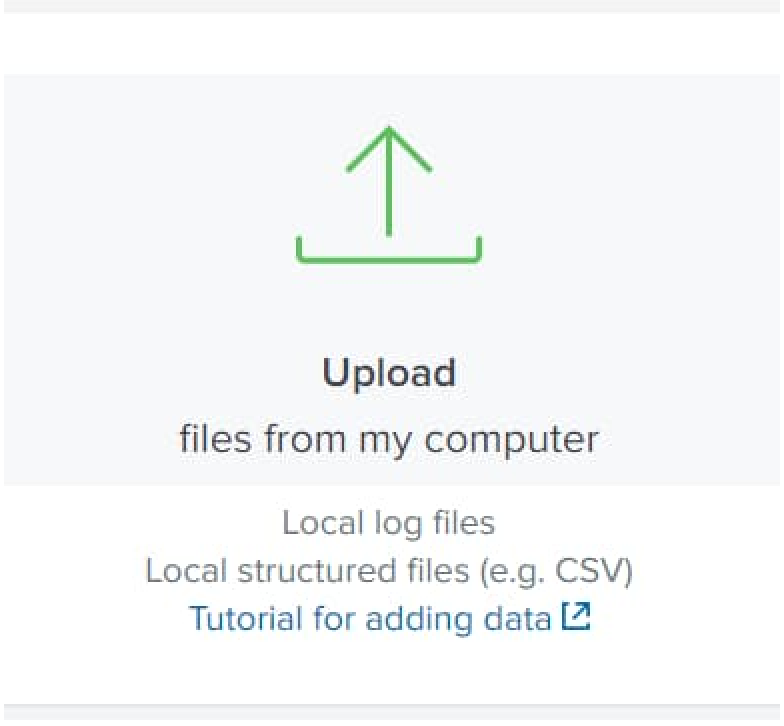


4 **Click** Add Data. **Click** Skip Tour if asked.


Add Data

Add or forward data to Splunk Enterprise. Afterwards, you may extract fields.

5 **Ignore** any errors about not being connected to the Internet. **Click** Upload files from my computer.


Upload

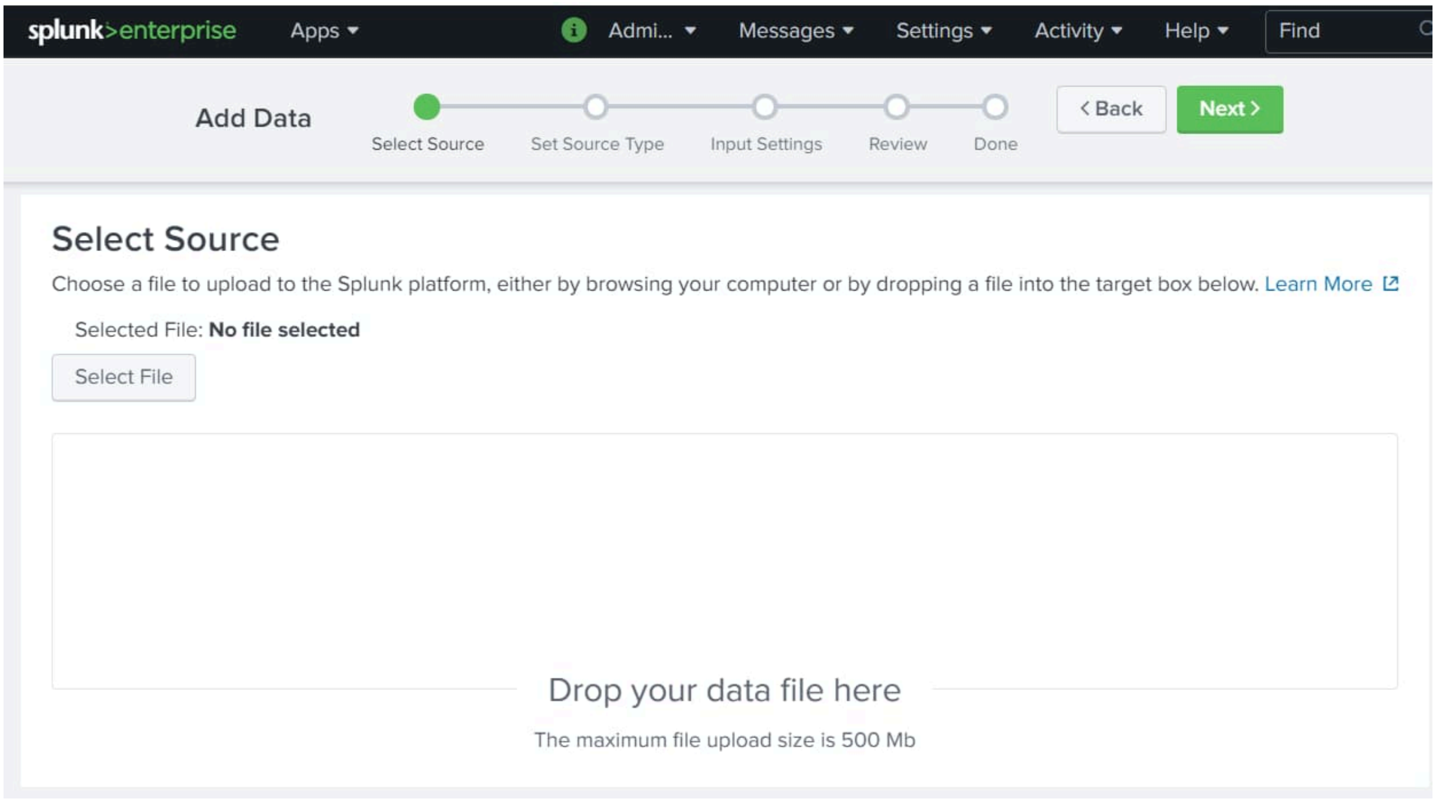
files from my computer

Local log files

Local structured files (e.g. CSV)

Tutorial for adding data

6 **Click** Select File.



Select Source

Choose a file to upload to the Splunk platform, either by browsing your computer or by dropping a file into the target box below. Learn More

Selected File: **No file selected**

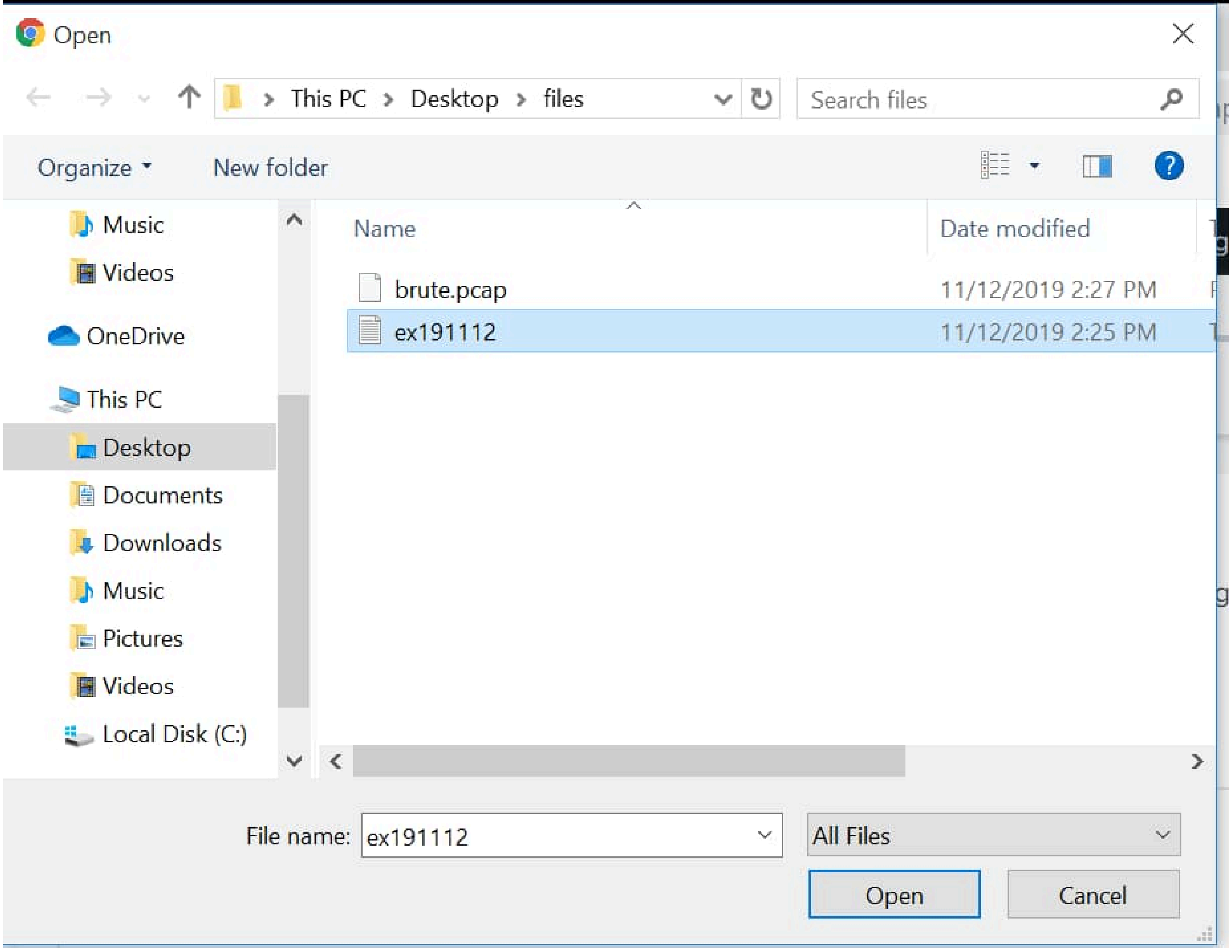
Select File

Drop your data file here

The maximum file upload size is 500 Mb

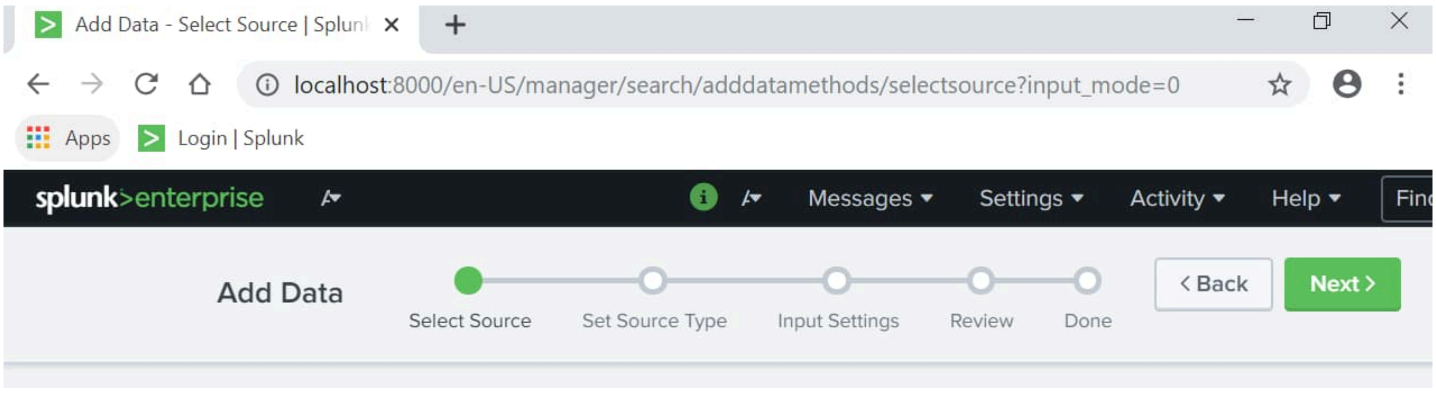
Note: There is a limit of 500 MB per day of data with the free version of Splunk.

7 **Browse** to This PC > Desktop > files. **Double-click** on the ex19112 file.

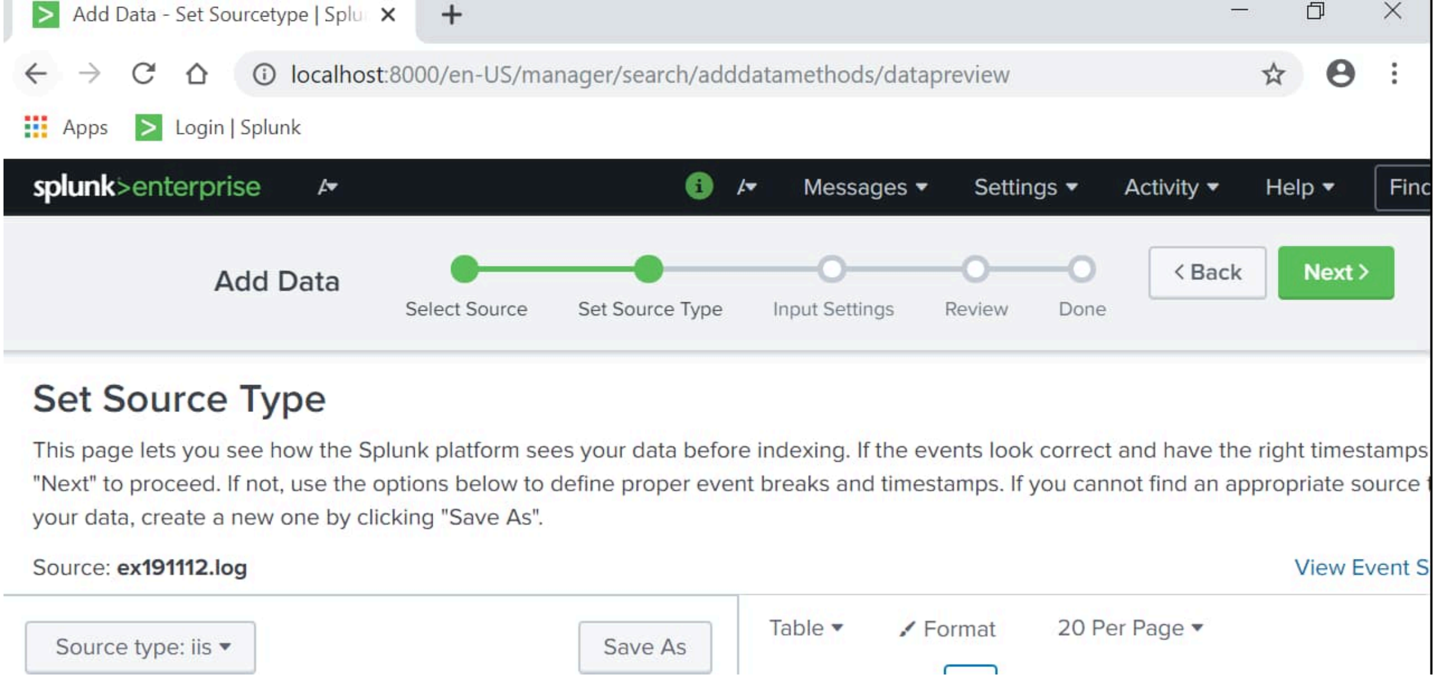


Note: This is the same file we parsed through with the Linux command line tools.

8 **Click** Next.



9 **Verify** that the Source type is iis and **select** Next.



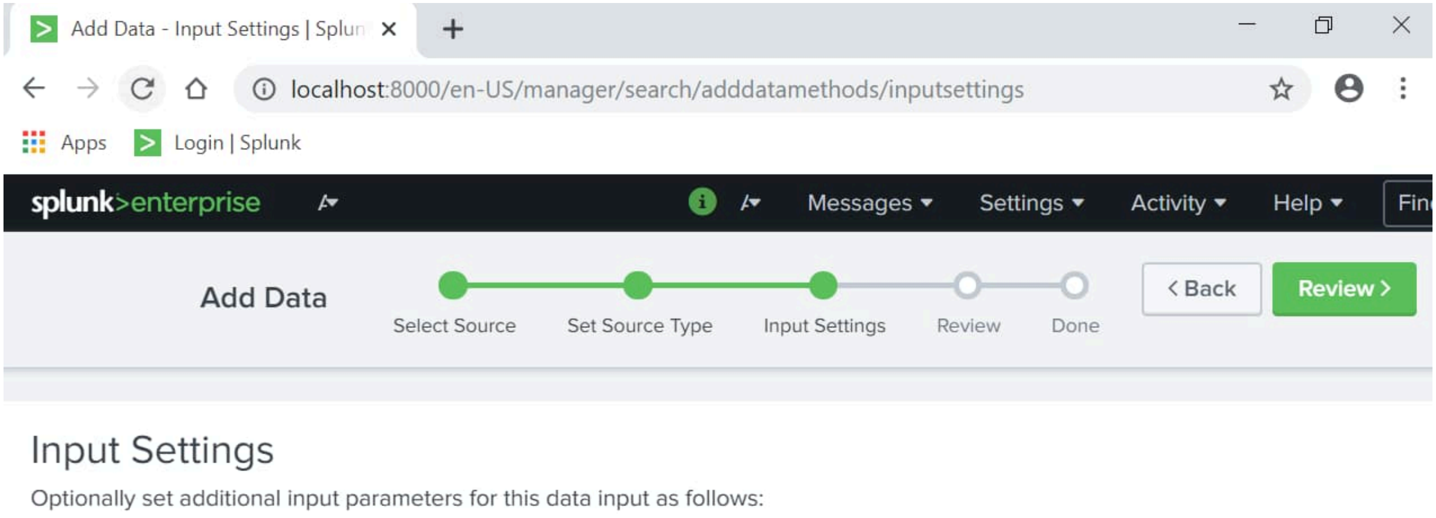
Set Source Type

This page lets you see how the Splunk platform sees your data before indexing. If the events look correct and have the right timestamps "Next" to proceed. If not, use the options below to define proper event breaks and timestamps. If you cannot find an appropriate source your data, create a new one by clicking "Save As".

Source: **ex191112.log**

NOTE: The source type is the type of log file that Splunk is going to analyze. iis is a web server on Windows web servers.

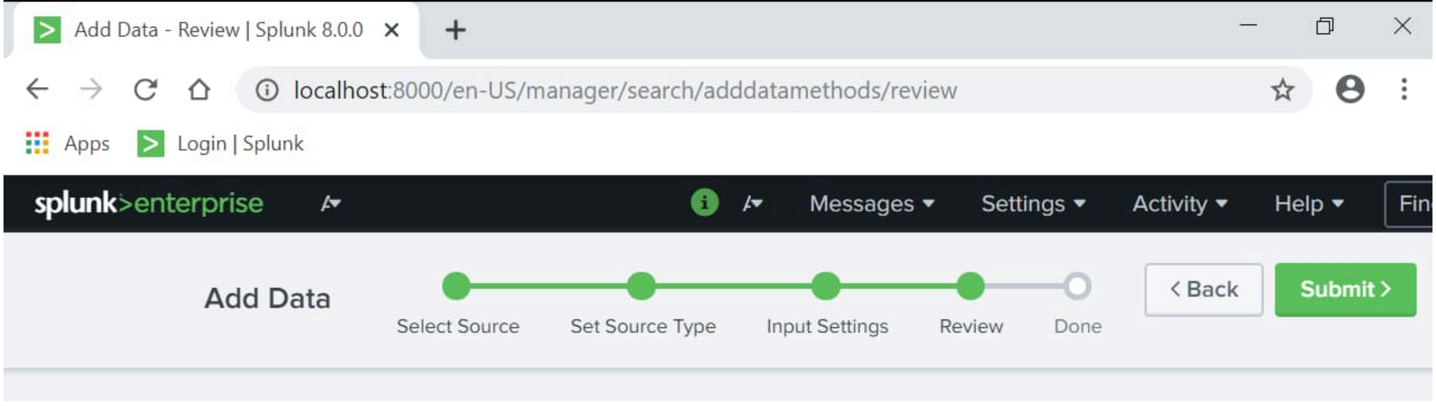
10 **Click** Review to Input Settings.



Input Settings

Optionally set additional input parameters for this data input as follows:

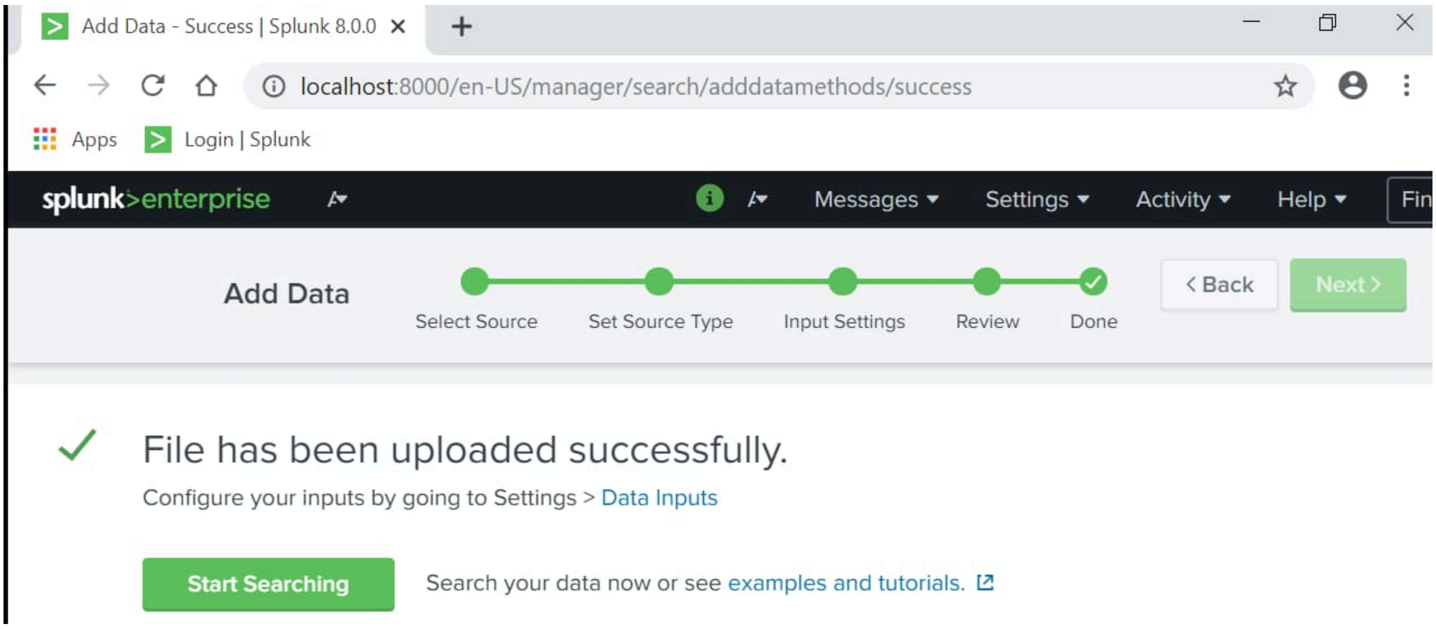
11 **Click** Submit.



Review

Input Type	Uploaded File
File Name	ex191112.log
Source Type	iis
Host	DESKTOP-A2KB24I
Index	Default

12 **Click** Start Searching.

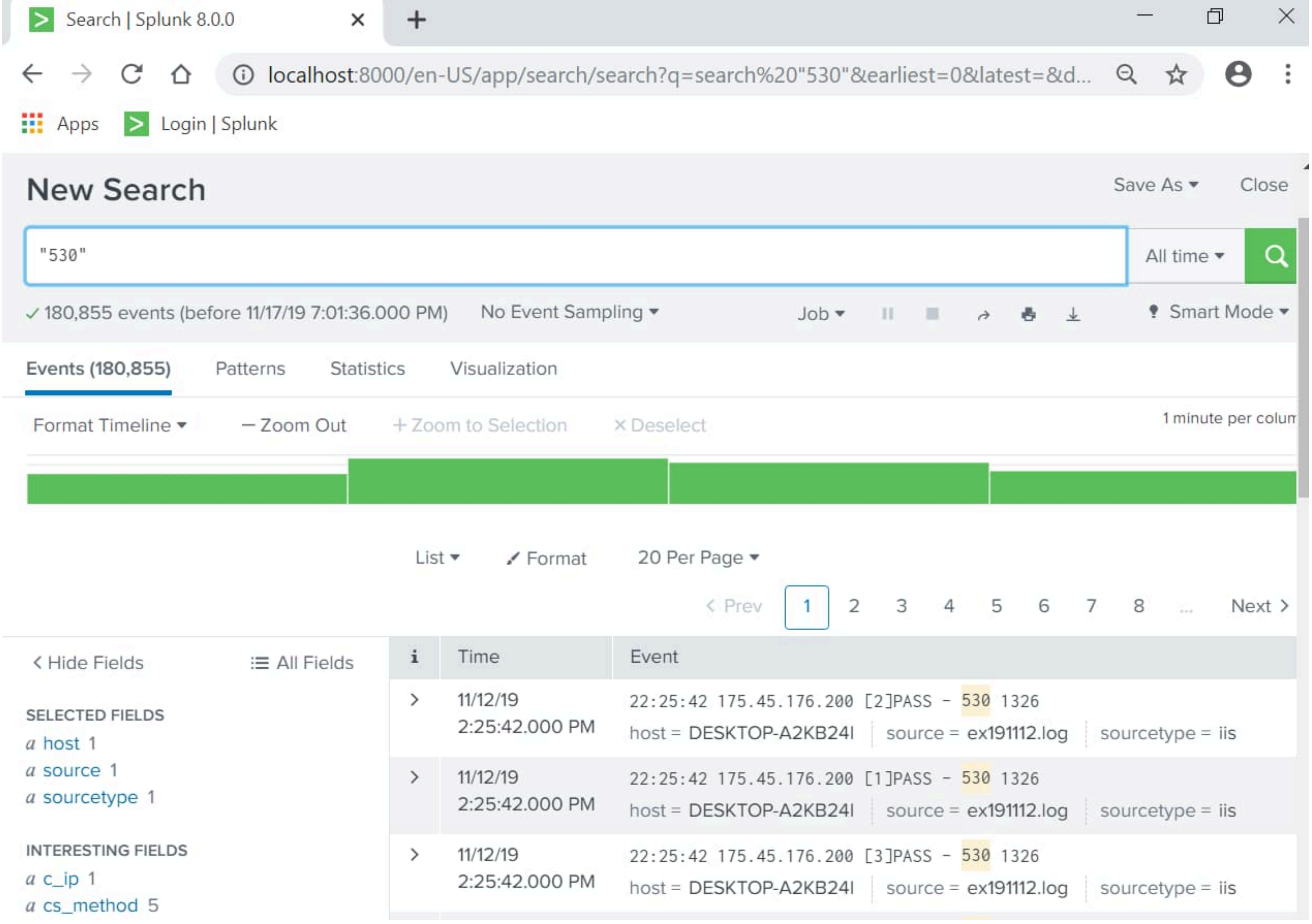


File has been uploaded successfully.

Configure your inputs by going to Settings > Data Inputs

Search your data now or see examples and tutorials.

13 In the New Search bar, **type** `“530”` and then **click** the search button. The "530" event code in IIS is a User cannot login error.

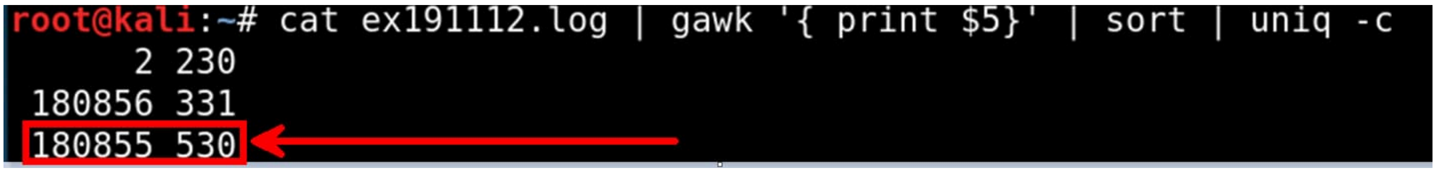


```
"530"
✓ 180,855 events (before 11/17/19 7:01:36.000 PM)
```

i	Time	Event
>	11/12/19 2:25:42.000 PM	22:25:42 175.45.176.200 [2]PASS - 530 1326 host = DESKTOP-A2KB24I source = ex191112.log sourcetype = iis
>	11/12/19 2:25:42.000 PM	22:25:42 175.45.176.200 [1]PASS - 530 1326 host = DESKTOP-A2KB24I source = ex191112.log sourcetype = iis
>	11/12/19 2:25:42.000 PM	22:25:42 175.45.176.200 [3]PASS - 530 1326 host = DESKTOP-A2KB24I source = ex191112.log sourcetype = iis

The number of events with the code of 530 is 180,855. This was the exact same number of these event codes when we used all of the Linux commands below to parse the information.

root@kali:~# `cat ex191112.log | gawk '{print $5}' | sort | uniq -c`

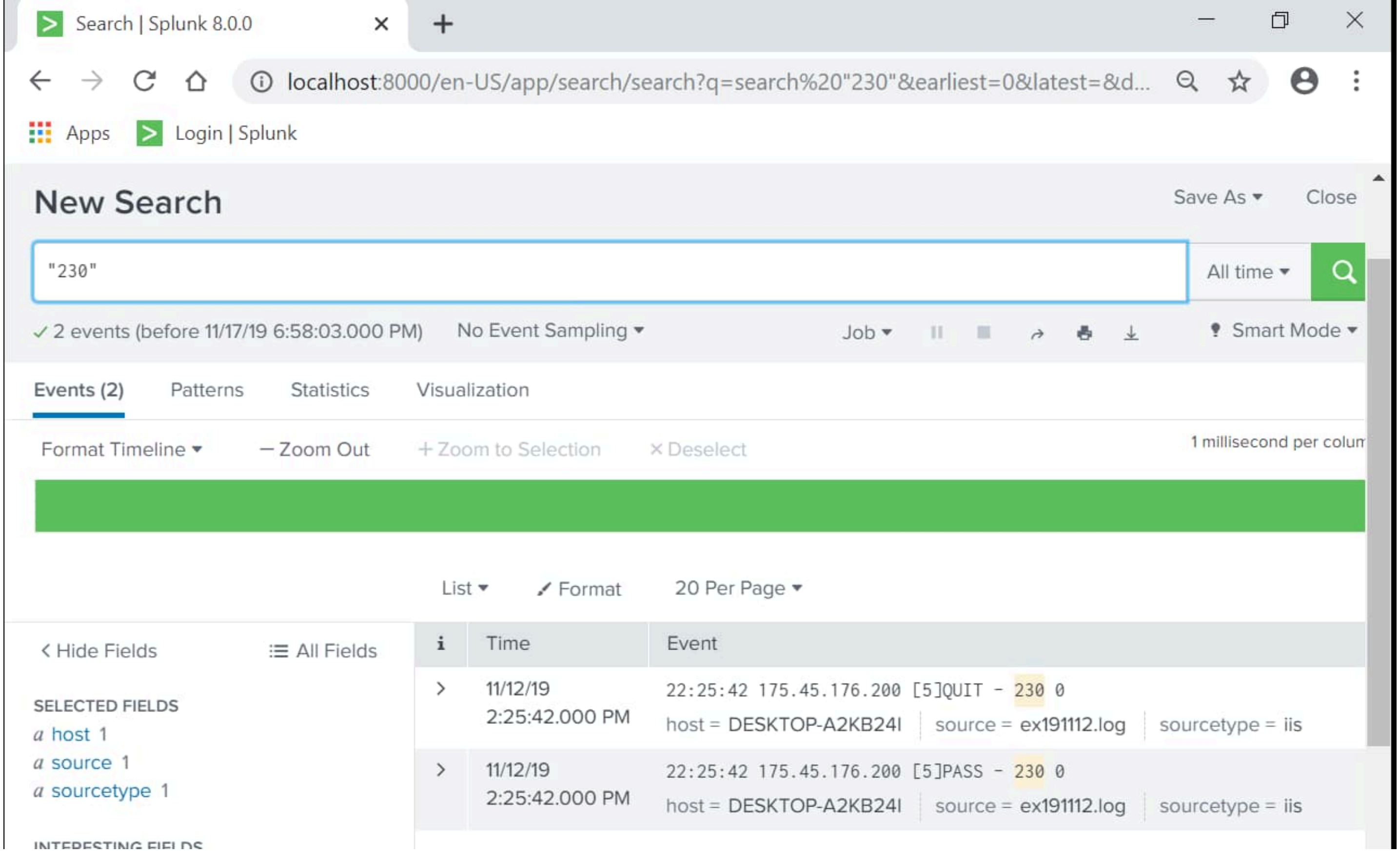


```
root@kali:~# cat ex191112.log | gawk '{ print $5}' | sort | uniq -c
      2 230
 180856 331
 180855 530
```

Notice that both methods provide the same result, but the Linux results were provided faster. The disadvantage to the Linux method is remembering the use and syntax of the commands.

14 In the New Search bar, **type** `"230"` and then **click** the search button.

"230" event code is a successfully log in in IIS.



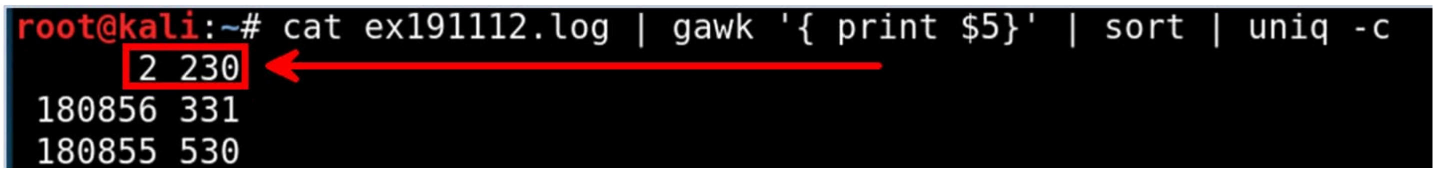
```
"230"
```

✓ 2 events (before 11/17/19 6:58:03.000 PM)

i	Time	Event
>	11/12/19 2:25:42.000 PM	22:25:42 175.45.176.200 [5]QUIT - 230 0 host = DESKTOP-A2KB24I | source = ex191112.log | sourcetype = iis
>	11/12/19 2:25:42.000 PM	22:25:42 175.45.176.200 [5]PASS - 230 0 host = DESKTOP-A2KB24I | source = ex191112.log | sourcetype = iis

The number of events with the code of 230 is 2. This was the exact same number of these event codes when we used all of the Linux commands below to parse the information.

root@kali:~# `cat ex191112.log | gawk ‘{print $5}’ | sort | uniq -c`



```
root@kali:~# cat ex191112.log | gawk '{ print $5}' | sort | uniq -c
      2 230
 180856 331
 180855 530
```

Again, both methods provide the same result, but the Linux results were provided faster. As we started before, the major disadvantage to the Linux method is remembering the use and syntax of the commands. Frequent use and practice with the Linux commands and the available switches will make you more familiar with them. Using Splunk is always another option.

⚠ Note: **Press** the STOP button to complete the lab.

Summary

Splunk is a log analysis tool that will allow you to ingest several different types of log formats, including web logs, system logs, event viewer files, and PCAP data. Splunk is a commercial product, but it has a free edition. With the free edition of Splunk, there is a parse limit of 500 MB per day within a 24-hour time frame. Splunk and Linux command line log analysis provide the same result, but the Linux results are provided faster. The disadvantage to the Linux method is remembering the use and syntax of the commands. Frequent use and practice with the Linux commands and the available switches will make you more familiar with them. Using Splunk is always another option.

DISCUSSION QUESTIONS:

1. What is Splunk?
2. What are some of the advantages of using Splunk?
3. What are some advantages of using the Linux command line over Splunk?
4. Is Splunk free?
5. What limitation will you face if you do not pay for Splunk?