Apache Security Log Analysis – Command Reference

The list of commands below includes everything needed to complete the hands-on log analysis exercise, but the arguments and examples listed are not exhaustive. In particular, awk is an entire scripting language and only the barest functionality is used in the examples here.

Additionally, though all of the commands below are shown taking input piped from another process, all of them can also accept a file name as command line argument. For example, the two commands below are equivalent.

- `cat apache_access.log | grep "/tmp"`
- `grep "/tmp" apache_access.log`

If that sounds confusing, feel free to always use cat first, following the pattern shown in this reference.

Command Reference

`cat` - Output the contents of a file to a user's terminal, or as input into another process.

Examples:

- `cat apache_access.log` **(CTL+c will stop it.)**
- `cat apache_access.log | uniq`

`grep` - Output the contents of a file or input that match a given string.

Arguments

- `-v` **(print lines that don't match)**

Examples:

- `cat apache_access.log | grep 26.165.104.85`
- `cat apache_access.log | grep -v " 200 "`

`head` - Output the first $x lines of a file or input, where $x defaults to 10.

Arguments:

- `-n __` **(change the number of lines to output)**

Examples:

- `cat apache_access.log | head`
- `cat apache_access.log | grep "getting-started" | head -n 5`

`tail` - Output the last $x lines of a file or input, where $x defaults to 10.

Arguments:

- `-n __` **(change the number of lines to output)**

Examples:

- `cat apache_access.log | tail`
- `cat apache_access.log | grep "getting-started" | tail -n 5`

`wc` - Count the number of words (or lines) in a file or input.

Arguments:

- `-l` **(count number of lines instead of words)**

Examples:

- `cat apache_access.log | wc -l`
- `cat apache_access.log | head -n 1 | wc`

`awk` - Print specified columns from a file or input.

Arguments

- `'{print $n, $m}'` **(print the nth and mth columns from the input)**
- `-F '__'` **(change the field delimiter - default is whitespace)**

Examples

- `cat apache_access.log | grep 26.165.104.85 | awk '{print $7}'`
- `cat apache_access.log | head | awk -F '"' '{print $6}'`

`sort` - Output a file or input in sorted order.

Arguments

- `-n` **(sort numerically instead of alphanumerically)**
- `-r` **(reverse sort)**

Examples

- `cat apache_access.log | grep " 404 " | awk '{print $7}' | sort`

`uniq` - Output lines that only exist once in a file or input.

Note: the file or input must be sorted prior to sending it to uniq.

Arguments

- `-c` **(print a count of how many times the line was seen)**

Examples

- `cat apache_access.log | grep " 404 " | awk '{print $1}' | sort | uniq -c`
- `cat apache_access.log | awk '{print $1}' | sort | uniq -c | sort -nr`