

Linux sadservers

"Saint John": what is writing to this log file?

- `ps auxf`
- `lsof /path/to/file`
- `fuser /path/to/file`
- `pwdx PID: pwd del fichero PID`

1. You can use `ps` to list all processes and see if you see something related, for example with: `ps auxf`. Ignore system processes [in brackets].

A better way is to use the command to list open files: `lsof`.

2. Find the name (first column) and Process ID (PID, second column) of the process related to `/var/log/bad.log` by running `lsof` and filtering the rows to the one(s) containing `bad.log`.

You can also use the "fuser" command to quickly find the offending process: `fuser /var/log/bad.log`.

3. Run: `lsof |grep bad.log` and get the PID (second column). "Saint John": what is writing to this log file?

With the PID of the process, it's not necessary but we can find its current working directory (program location) by doing `pwdx PID` or for more detail: `lsof -p PID` and check the `cwd` row. This will allow us to check its ownership and perhaps inspect its offending code if it's a script (not a binary).

(Open window once more to see the complete solution).

"Saskatoon": counting IPs.

- `awk '{print $1}' ' ' * 'cut -d' ' -f1`
- `sort:`
 - `-h`: numérico humano
 - `-r`: reverso
 - `-k[primer-campo],[ultimo-campo]`
 - `-t <SEP>`: separador
- `uniq -c`: cuenta elementos únicos

1. To get the first field (IP) of the file, you can do `awk '{print $1}' access.log` or using "cut" with delimiter of space (`-d' '`) and picking the first field (`-f1`): `cat access.log |cut -d' ' -f1`. You may want to append a pipe `| head` or `| tail` as you construct the command to see how your filters are working.

2. After the previous step, you want to sort the IPs so they are together and can be counted: `cat access.log | awk '{print $1}' |sort`

3. Now you want to do the count with "`uniq -c`", so we have so far: `awk '{print $1}' access.log |sort|uniq -c`

4. Finally you want to sort the results with "sort" (goes in ascending order) and get the latest one (with "tail -1" for example), or sort in reverse order with "sort -r" and get the top result: `awk '{print $1}' access.log|sort|uniq -c|sort -r|head -1`.

(Open window once more to see the complete solution).

Solution: One possible way is `awk '{print $1}' access.log|sort|uniq -c|sort -r|head -1|awk '{print $2}' > /home/admin/highestip.txt`

"Santiago": Find the secret combination

- `grep -rc`
- `grep -A 1`
- `find ... | xargs grep -c`

1. Use `grep` recursively or use `find` and pass the results to `grep` via `xargs`

(Open window once more to see the solution to the first part).

2. (Solution to 1) `cd /home/admin/` and then for example: `grep -rc Alice *.txt` or `find . -type f -name "*.txt" |xargs grep -c 'Alice'`, adding the results from the three files: `echo -n 411 > /home/admin/solution`

(Open window once more to see the solution to the second part).

3. (Solution to 2) The file with exactly one Alice occurrence is `1342-0.txt`: `grep Alice -A 1 /home/admin/1342-0.txt` (or open the file with `less` or `vi` and enter `/Alice`). Appending this result: `echo 156 >> /home/admin/solution` (The solution is `411156`).

"The Command Line Murders"

- `knock localhost 3000`
- `nmap -p- localhost`

1. You can use the `knock` utility, for example to knock on port 3000: `knock localhost 3000`. `Netcat (nc)` and `nmap` are also available. Note that `nmap` has some options where you'd need to be root (not possible here)

2. You can also write a `BASH` script that knocks sequentially on all ports.

3. Solution. Probably the fastest is using `nmap` against all ports, for example: `nmap -p- localhost`.

"Resumable Server": Linux Upskill Challenge

"Bucharest": Connecting to Postgres

- `sudo systemctl restart postgresql@13-main`

1. The issue might be related to the configuration of the PostgreSQL server. (See the error message when attempting the test). The configuration files are usually located in the `/etc/postgresql/$version/main/` directory. You might want to start by checking these files. (You'll need to use "sudo").

2. The `pg_hba.conf` file controls client authentication. This file is read on start-up and when the main server process receives a SIGHUP signal. If you're having trouble connecting to the database, this file could be a good place to look. (Click again "Next Clue/Solution" to reveal the final step)

Solution: In the `/etc/postgresql/13/main/pg_hba.conf` file, delete or comment out the lines with a reject keyword from all. Then restart the PostgreSQL service: `sudo systemctl restart postgresql@13-main`

From:

<https://matebcn.eu/> - miguel angel torres egea

Permanent link:

<https://matebcn.eu/linux:seguridad:sadservers>

Last update: **28/05/2024 15:51**

