

Filtrando dados pelas requisições

Transcrição

Em um novo dia de reunião na empresa *Multillidae*, os diretores falaram que agora eles precisam fazer um novo tipo de filtro nesse arquivo de log baseado na requisição **GET**, **POST**, **PUT** ou **DELETE**. Vamos dar uma olhada nessas requisições no arquivo de log.

Acessaremos o diretório `/apache-log` com o comando `cd apache-log`. Para ver o resultado do arquivo `apache.log`, faremos assim:

```
$ cat apache.log
```

Então, se procurarmos nesse resultado, vamos encontrar as requisições **GET**, **POST**, **PUT** e **DELETE**, como os nossos diretores nos disseram.

A ideia é que seja possível colocar o tipo de requisição que se deseja filtrar. Na etapa anterior, utilizamos o `grep` para fazer o filtro, e tivemos sucesso com ele. Por isso, vamos utilizá-lo agora.

Voltemos ao nosso diretório de scripts.

```
$ cd ~/Scripts
```

E criaremos um novo script para essa tarefa.

```
$ nano filtro-requisicao.sh
```

Como sabemos, a primeira linha do nosso script é o **interpretador**.

```
#!/bin/bash
```

Logo após o interpretador, temos que trocar de diretório, para o diretório `/apache-log`.

```
#!/bin/bash
```

```
cd ~/apache-log
```

Pegaremos o conteúdo do arquivo `apache.log`, e filtraremos o resultado baseado no parâmetro que o usuário irá passar: (**GET**, **POST**, **PUT**, **DELETE**).

```
#!/bin/bash
```

```
cd ~/apache-log
```

```
cat apache.log | grep $1
```

Mas antes, precisamos fazer uma validação. Precisamos saber qual é o tipo da requisição que o usuário irá passar. Utilizaremos o `if`. Vamos comparar o conteúdo desse parâmetro como uma **string**.

```
#!/bin/bash

cd ~/apache-log

if [ $1 == "GET" ]
then
    cat apache.log | grep GET
```

Se o usuário colocou a palavra "GET" como parâmetro, então filtraremos tudo o que tiver a requisição com o nome de "GET".

Caso não seja o "GET", poderá ser o "POST", o "PUT", ou o "DELETE".

```
#!/bin/bash

cd ~/apache-log

if [ $1 == "GET" ]
then
    cat apache.log | grep GET
elif [ $1 == "POST" ]
then
    cat apache.log | grep POST
elif [ $1 == "PUT" ]
then
    cat apache.log | grep PUT
elif [ $1 == "DELETE" ]
then
    cat apache.log | grep DELETE
fi
```

Nesse código, nós testamos todas as quatro opções de requisições, comparando o valor de cada parâmetro com a string correspondente.

Vamos sair e salvar com "Ctrl + X" e "Y". Agora testaremos esse script com as novas informações, passando como parâmetro a requisição `POST`.

```
$ bash filtro-requisicao.sh POST
```

Obtemos esse resultado:

```
rafael@rafael-VirtualBox:~/Scripts$ bash filtrando-novas-requisicoes.sh POST
94.253.13.179 - - [18/Jul/2017:15:15:34 -0300] "POST /list HTTP/1.0" 200 5104 "http://www.torres-cox.com/category/li
st/explore/privacy/" "Mozilla/5.0 (Windows 95; sl-SI; rv:1.9.0.20) Gecko/2012-10-17 17:24:58 Firefox/3.6.6"
178.88.86.139 - - [18/Jul/2017:15:37:16 -0300] "POST /search/tag/list HTTP/1.0" 200 5054 "http://www.martin-byrd.com
/list/search/blog/main/" "Mozilla/5.0 (X11; Linux i686) AppleWebKit/5352 (KHTML, like Gecko) Chrome/14.0.836.0 Safar
i/5352"
```

Como podemos ver, foi retornado duas requisições do tipo "POST". Faremos o teste com a requisição do tipo "DELETE".

Já o resultado desse filtro, foram retornados três requisições do tipo "DELETE". O mesmo acontece com os resultados as requisições do tipo PUT e GET.

Observamos que o script está conseguindo realizar o filtro baseado no parâmetro passado pelo usuário.

Entretanto, o usuário poderia colocar uma palavra qualquer como parâmetro, e o script é executado. O que está faltando é uma mensagem alertando-o que essa palavra não existe. Um outro caso que pode ocorrer, é o usuário colocar o nome das requisições em minúsculo. Nós também não estamos fazendo essa validação!

Se olharmos bem, o código no script não está muito elegante. Vemos que é possível melhorá-lo de várias maneiras.

Faremos essas modificações a seguir, onde também validaremos esses casos específicos sobre palavras que não estão no arquivo, e requisições em minúsculo, mostrando uma mensagem ao usuário.