UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO

RELATÓRIO: **implementação projeto ‘Leva Jeito’**

GABRIEL MARTINS SPÍNOLA

NATAL, RN
2019

GABRIEL MARTINS SPÍNOLA

RELATÓRIO: **implementação projeto ‘Leva Jeito’**

Relatório apresentado como requisito para obtenção de aprovação na disciplina Segurança de Redes, no Curso de Bacharelado em Tecnologia da Informação , na Universidade do Rio Grande do Norte.
Prof. Dr. Silvio Costa Sampaio

NATAL, RN
2019

SUMÁRIO

1. INTRODUÇÃO

O projeto trata-se da implementação de um programa que consiste em monitorar os arquivos de uma determinada pasta, para saber se algum arquivo foi modificado, removido ou adicionado no diretório especificado

O objetivo é utilizar os algoritmos de geração de hash(com e sem criptografia) na implementação do nosso programa de monitoramento que se chama “guarda”. O usuário que utilizar o ‘guarda’ terá a liberdade de escolher entre gerar um hash sem criptografia ou um hash com criptografia.

Esse trabalho foi implementado na linguagem Python 3. Contém apenas um arquivo necessário para rodar, o arquivo ‘guarda.py’. Esse, deve ser executado na linguagem Python 3 exclusivamente. O programa irá gerar um arquivo oculto na pasta em que foi executado, onde guardará a informação das pastas que estão sendo monitoradas pelo ‘guarda’ (OBS: se o arquivo de verificação de monitoramento for apagado ou movido da pasta que se encontra o ‘guarda.py’ o ‘guarda’ irá se comportar como se não tivesse monitorando nenhuma pasta).

As dificuldades apresentadas foram de aprendizado de ferramentas que foram utilizadas ao qual eu nunca tive contato anteriormente. como a ferramenta ‘getopt’, algumas bibliotecas de criação, alteração e remoção de arquivos/pastas, bibliotecas de geração de hash e hmac.

2. COMO EXECUTAR

Para iniciar o monitoramento em uma pasta devemos executar o programa passando o parâmetro -i e em seguida a pasta que será monitorada, e passando também o método de geração de hash que será utilizado --hash ou --hmac. Exemplo: python3 guarda.py --hash -i “pasta”

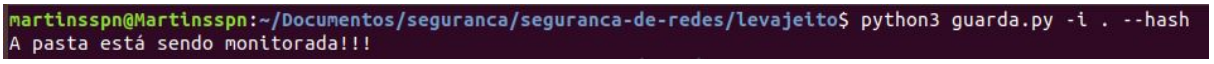

Figura 1 - inicializando monitoramento na pasta atual

Para fazer o tracking da pasta monitorada executaremos o programa utilizando o parâmetro -t e em seguida a pasta que será realizado o tracking. Na imagem abaixo temos uma demonstração de tracking adicionando um novo arquivo na pasta que está sendo monitorada.

Figura 2 - realizado o tracking na pasta selecionada

E se alterarmos um arquivo dentro da pasta? Quando realizarmos o tracking ele avisará que foi realizado uma alteração no arquivo. O exemplo abaixo é realizado uma alteração no arquivo ‘levajeito.txt’ e logo após realizamos o tracking.

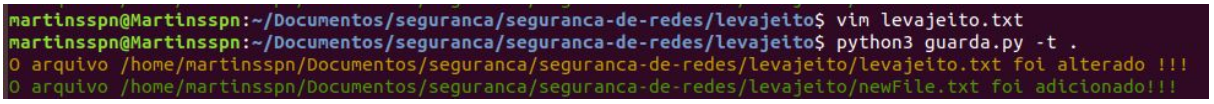

Figura 3 - testando alteração em um arquivo dentro da pasta

Em último caso, se removermos um arquivo da pasta, o tracking avisará que foi realizado a remoção. O exemplo abaixo retiramos o arquivo ‘levajeito.txt’ da pasta monitorada e realizamos o tracking.

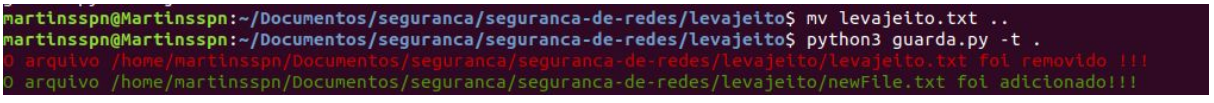

Figura 4 - testando a remoção de um arquivo dentro da pasta

Caso desejar desabilitar o tracking para a pasta selecionada devemos executar o programa com o parâmetro -x e em seguida informar qual pasta será desabilitada.

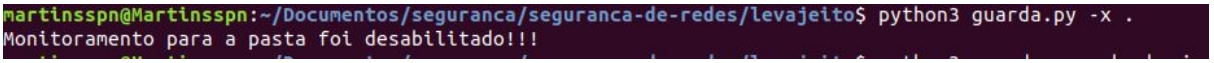

Figura 5 - desabilitando o monitoramento para a pasta atual

Caso desejar desabilitar o tracking para todas as pastas que estão sendo monitoradas pelo ‘guarda’ devemos utilizar a palavra “all” no lugar onde seria a pasta.

Figura 6 - desabilitando monitoramento para todas as pastas

E se modificarmos algo na pasta e quiser que o ‘guarda’ monitore a pasta após a mudança? Nesse caso podemos atualizar a lista de arquivos monitorados pelo guarda. Para isso basta utilizarmos o parâmetro ‘--push’ seguido da pasta que se deseja atualizar.

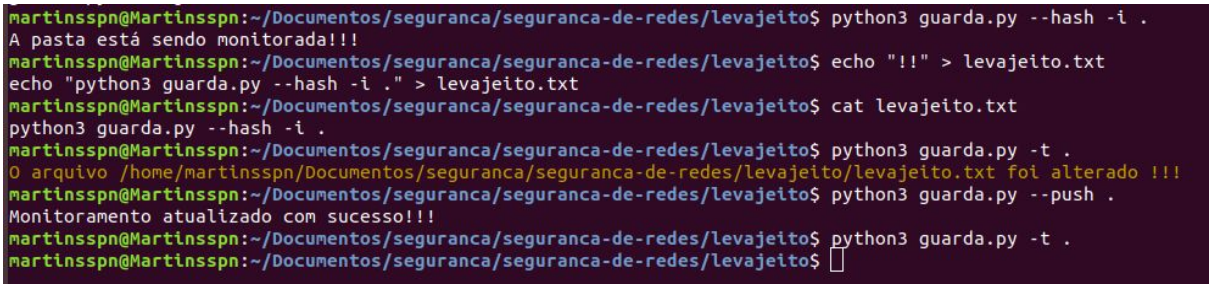

Figura 7 - atualizando lista de arquivos monitorados

Se adicionarmos o parâmetro -o seguido de um arquivo de saída em qualquer dos comandos apresentados acima o programa imprimirá a saída não mais no terminal, mas imprimirá no arquivo passado por parâmetro.

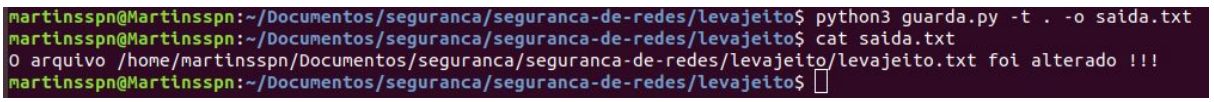

Figura 8 - imprimindo saída do programa em um arquivo passado por parâmetro

CONSIDERAÇÕES FINAIS

Este trabalho apresentou a implementação do algoritmo de monitoramento de arquivos por meio de geração de hash dos arquivos. Sendo possível verificar se os arquivos de uma pasta, que será passada por parâmetro na inicialização do programa, foram alterados, removidos ou adicionados. O objetivo principal da aplicação é utilizar os conceitos de função hash vistos em sala e criar um programa que seja possível de ser usado não só no âmbito acadêmico mas também na vida fora da faculdade, ou seja, possível de usar no dia a dia.

3. REFERÊNCIAS

CORES No Terminal. [*S. l.*]. Disponível em: https://wiki.python.org.br/CoresNoTerminal. Acesso em: 1 out. 2019

PYTHON os.mkdir() Method. [*S. l.*]. Disponível em: https://www.tutorialspoint.com/python/os_mkdir.htm. Acesso em: 1 out. 2019.

MD5 hash in Python. [*S. l.*]. Disponível em: https://www.geeksforgeeks.org/md5-hash-python/. Acesso em: 1 out. 2019.

HASHING an array or object in python 3. [*S. l.*], 1 jul. 2013. Disponível em: https://stackoverflow.com/questions/17412304/hashing-an-array-or-object-in-python-3/17441989#17441989. Acesso em: 1 out. 2019.

PYTHON: Uma função recursiva para listar arquivos de uma pasta e das subpastas. [*S. l.*], 14 dez. 2016. Disponível em: http://graciomar.com.br/blog/35/python-uma-funcao-recursiva-para-listar-arquivos-de-uma-pasta-e-das-subpastas. Acesso em: 1 out. 2019.

ENDEREÇO Caminho path Criar Arquivo Diretório Pasta módulo os. [*S. l.*]. Disponível em: https://www.pythonprogressivo.net/2018/10/Endereco-Caminho-path-Criar-Arquivo-Diretorio-Pasta-modulo-os.html. Acesso em: 1 out. 2019.

C-STYLE parser for command line options. [*S. l.*]. Disponível em: https://docs.python.org/2/library/getopt.html. Acesso em: 1 out. 2019.