REVISTA UNIFESO – CADERNO UNIFESOTEC
v. 1, n. 1, 2023, Teresópolis

Artigo científico

ALLSAFE: DESENVOLVIMENTO DE UM ANTIVÍRUS DE BAIXO CUSTO COMPUTACIONAL

ALLSAFE: DEVELOPMENT OF AN ANTIVIRUS WITH LOW COMPUTATIONAL COST

Enos Gabriel Melo Da Silva; Eugênio Silva

RESUMO

Atualmente, há diversos programas de antivírus no mercado que se apresentam como mecanismos eficazes de proteção de sistemas computacionais contra a infecção por vírus. Contudo, de maneira geral, esses programas apresentam alta demanda por recursos computacionais, especialmente memória e processamento, para alcançar seus objetivos. Com isso, o uso desses programas em máquinas mais antigas, em que esses recursos computacionais são mais escassos, se torna praticamente inviável. Assim, tem-se uma parcela bastante significativa de equipamentos que acabam ficando desprotegidos. Diante disso, este trabalho propôs e desenvolveu um programa de antivírus, onde o processamento é feito em nuvem, garantindo um baixo consumo de recursos computacionais no computador onde o antivírus está instalado. O programa desenvolvido foi testado em máquinas virtuais e máquinas reais e os resultados obtidos mostram a viabilidade da proposta.

Palavras-chave: Antivírus; Segurança; Vírus;

ABSTRACT

Currently, there are several antivirus programs on the market that present themselves as effective mechanisms for protecting computer systems against virus infection. However, in general, these programs have a high demand for computational resources, especially memory and processing, to achieve their goals. As a result, the use of these programs on older machines, where these computational resources are scarce, becomes practically unfeasible. Thus, there is a very significant portion of equipment that ends up being unprotected. In view of this, this work proposed and developed an antivirus program, where the processing is done in the cloud, guaranteeing a low consumption of computational resources, on the computer where the antivirus is installed. The developed program was tested in virtual machines and real machines and the obtained results show the viability of the proposal.

Keywords: Anti-viruses; Security; Virus;

INTRODUÇÃO

A segurança da informação é um assunto muito importante na área de tecnologia, pois é responsável pela proteção dos dados e sistemas computacionais. Um dos softwares relacionados com a segurança da informação são os antivírus, que são extremamente recomendados para todos os tipos de usuários. Os antivírus têm como objetivo proteger o sistema de vírus e *scripts* maliciosos. Quando um arquivo é baixado no computador o antivírus deve verificar se o arquivo baixado pode conter vírus e notificar o usuário. Os antivírus podem também efetuar filtragens na rede, ou seja, limitar acesso a sites que tenham certificados duvidosos, *feedbacks* negativos e que façam parte do histórico de sites perigosos. As filtragens na rede são muito interessantes para empresas, pois permitem que o administrador limite o tráfego de rede fazendo com que apenas sites que estejam de acordo com a política da empresa possam ser acessados pela rede.

Nos dias atuais, há diversos antivírus que, em comum, apresentam como aspecto negativo o alto consumo de recurso computacional, em especial memória RAM e processador. O processo de escaneamento de um antivírus pode deixar os computadores mais lentos, conforme explica o *supporter*, Sebastião Almeida:

> *"O antivírus está sempre buscando por vírus e atualizações enquanto está online.*

> *Se esse processo for executado durante o horário em que estamos trabalhando, sequentemente, ele deixará o computador lento"* (Sebastião, Almeida, 2017).

Segundo o suporte do antivírus Avast, em caso de lentidão:

> *"O ideal e a melhor solução seria comprar um novo computador com múltiplos processadores. No entanto, se o seu orçamento não permite comprar um novo computador, então pelo menos, acrescentar memória RAM"* (Avast, 2015).

Diante do exposto, o desenvolvimento de um antivírus que proporcione eficácia, com baixo custo computacional, seria de bastante utilidade para atender a demanda por proteção em computadores mais antigos.

MATERIAL E MÉTODOS

O antivírus é um software que identifica e protege os dispositivos de malwares, também conhecidos como vírus. Esse programa pode ser instalado em computadores e dispositivos móveis, como celulares e tablets. Atualmente, há diversos tipos de antivírus no mundo da tecnologia e todos têm a sua particularidade e metodologia de funcionamento. A função mais simples de um antivírus é monitorar arquivos e outros programas de um dispositivo para detectar vírus. Quando novas aplicações são instaladas, o programa faz a verificação delas para saber se existe alguma ação suspeita. Se algo foi identificado, a instalação é bloqueada ou a nova aplicação é encaminhada para a quarentena (ESET,2022).

No processo de verificação de um arquivo, a maioria dos antivírus segue um padrão de verificação que pode ser dividido em duas camadas. Na primeira camada, o antivírus recebe o arquivo e verifica se a sua assinatura é de algum vírus conhecido. A busca e comparação são realizadas por meio de uma base de dados de vírus e malwares. As bases de dados para comparação tendem a ser muito robustas, já que elas têm que armazenar todos os tipos de vírus conhecidos até o momento. Na segunda camada o antivírus faz uma busca mais minuciosa. Ao receber o arquivo para verificação o antivírus particiona o arquivo e observa todo o seu código binário, verificando se existe alguma macro ou algum *script* malicioso. Após esse processo o antivírus emite um laudo. Se o laudo informar que o arquivo verificado está infectado, o antivírus deve realizar ações para neutralizar o vírus.

Com base nas informações de como os antivírus funcionam, é necessário identificar a forma mais adequada para desenvolver o projeto. Ao pesquisar sobre o assunto foram observadas duas opções. A primeira opção seria desenvolver um *back-end* que fizesse as principais camadas de verificação de vírus. No entanto, isso é algo que consome muito tempo, pois para validar o seu funcionamento serão necessários testes e esses testes são complexos, já que em muitos casos é necessário testar o projeto com vírus reais para validar a sua efetividade.

Como segunda opção há uma biblioteca desenvolvida em linguagem Python, disponibilizada pela empresa CLOUDMERSIVE, que realiza os processos de verificação. Tendo essas opções foi escolhida a segunda. Foi desenvolvido um sistema *desktop* Python integrado com a biblioteca CLOUDMERSIVE, que faz a comunicação com *endpoints* de uma API. Nesta API, são executados processos de verificação de vírus e uma das vantagens de usar uma API é que para a verificação dos vírus, não se utiliza a máquina local para processar as camadas de verificação, o que garante o uso reduzido de seus recursos (memória e processamento). Esse comportamento de uso de API pela rede é algo comum entre APIs.

Para o desenvolvimento do antivírus foi utilizada a linguagem Python, que é uma linguagem de programação de alto nível dinâmica, interpretada, modular, multiplataforma e orientada a objetos (PYTHON,2022). Por ser uma linguagem de sintaxe relativamente simples, ganhou popularidade entre os desenvolvedores. Uma de suas maiores vantagens é a grande oferta de

bibliotecas, nativas e de terceiros, tornando-a muito robusta e útil. O Python é utilizado em diversas áreas, principalmente na área de segurança da informação, devido a sua flexibilidade e fácil legibilidade.

Para o desenvolvimento da interface gráfica foi escolhida a biblioteca PySimpleGUI. Trata-se de uma biblioteca gráfica de fácil utilização e que apresenta uma demanda bastante baixa por recursos computacionais. Com isso tem-se uma ferramenta que atende, ao mesmo tempo, aos requisitos desempenho e de facilidade criação de uma interface de comunicação com a aplicação desenvolvida em Python.

Para a verificação de vírus foi utilizada a biblioteca CLOUDMERSIVE, que disponibiliza serviços como, verificar se um arquivo contém vírus, verificar se um *website* é malicioso, até mesmo verificar se existe algum vírus em nuvem. Alguns destes serviços possuem custos, mas alguns são gratuitos. Para o desenvolvimento deste antivírus, por exemplo, foi utilizada uma licença gratuita que permite fazer mais de 800 chamadas por mês, o que é um número que provavelmente não vai ser utilizado no período de 30 dias. No desenvolvimento do antivírus foram utilizados apenas dois recursos, o "Vírus Scan Files" e o "Vírus Scan URLS".

Vírus Scan files é um *endpoint* da CLOUDMERSIVE. Um *endpoint* da API é o local em que essas solicitações (conhecidas como chamadas de API) são atendidas (CLOUDFLARE, 2022). Para fazer uma chamada adequada para este endereço é necessário informar o nome do arquivo para verificação e a chave da API. Após isso, a API recebe o arquivo e realiza os procedimentos para verificar sua integridade.

A CLOUDMERSIVE possui uma base de dados sobre assinaturas de vírus com mais de 17 milhões de resultados. Segundo a documentação, o arquivo submetido passa por duas camadas de testes. Na primeira camada, chamada de "Virus and Malware Signature Scanning" a API verifica se o arquivo submetido faz parte de algum vírus conhecido, por meio de buscas de assinaturas em uma base de dados. Após a busca, caso o arquivo seja um vírus desconhecido, será necessário realizar a segunda camada de testes, chamada "Content Scanning".

Na segunda camada a API faz uma leitura minuciosa no arquivo submetido, verificando se o seu conteúdo contém *scripts* ou macros maliciosos. Esta leitura funciona sobre todos os formatos de arquivos, até mesmo binários (.exe) e comprimidos (rar, zip e etc...). Após realizar estas etapas a API retorna os resultados, informando se o arquivo possui vírus ou não. Com base nesse retorno, cabe ao usuário decidir o que será feito com o arquivo.

Virus Scan URLS também é um *endpoint* da CLOUDMERSIVE, mas foi desenvolvido para verificação de *links* e *websites*. Para fazer uma chamada adequada para este endereço é necessário informar uma URL para verificação e a chave da API. Ao fazer uma requisição, é acionada uma única etapa chamada "Content Scanning". O procedimento de verificação é o mesmo do *endpoint* "Vírus Scan Files" com uma única diferença, que neste caso será verificada uma URL em vez de um arquivo real, sendo assim é descartada a necessidade de *upload*. Após a verificação, a API vai informar se o *link* é seguro ou não.

Para realizar os testes foi utilizado o software VirtualBox. Trata-se de um software *opensource*, aplicativo gratuito, multi-plataforma para criar, gerenciar e executar máquinas virtuais (VMs) – computadores cujos componentes de hardware são emulados pelo computador host, o computador que executa o programa. VirtualBox pode ser executado em Windows, Mac OS X, Linux e Solaris (ORACLE,2022).

Após reunir essas informações, foi desenvolvido um sistema desktop Python integrado com a biblioteca CLOUDMERSIVE, que faz a comunicação com *endpoints* de uma API. No processo de *back-end* o objetivo principal é integrar o código em Python com a API CLOUDMERSIVE, que é responsável

por verificar a existência de vírus em arquivos. Para essa integração, existe uma biblioteca Python, disponibilizada pela própria Cloudmersive, chamada de "cloudmersive-virus-api-client". Por meio desta biblioteca é possível estabelecer uma comunicação entre a aplicação Python e o *endpoint* de verificação de vírus.

Para realizar essa integração é necessário instalar a biblioteca "cloudmersive-virus- api-client" no Python e passar dois parâmetros. O primeiro parâmetro é a chave da API, que pode ser obtida gratuitamente no *website* da Cloudmersive, sendo necessário apenas fazer um cadastro. O segundo parâmetro é o caminho onde se encontra o arquivo a ser verificado, se for um *website* o parâmetro será URL. Feito isso, se os dados foram informados corretamente, um retorno da API será recebido informando o resultado da verificação.

No retorno da API é informado um campo "Clean Result" com um valor booleano. Caso o resultado seja falso, significa que o arquivo está infectado com algum vírus.

Após fazer a integração com a API é necessário criar uma interface mais amigável para que o usuário possa selecionar o arquivo para ser verificado. Para fazer isso utilizando a biblioteca PySimpleGUI, foi desenvolvido um layout utilizando a sintaxe padrão disponível na documentação do (PySimpleGUI, 2022). Após criar o layout é necessário chamá-lo por meio de uma função e estará concluído o antivírus em Python. A Figura 1 mostra um exemplo de código Python realizando uma chamada para API. Na Figura 2, é apresentado um exemplo de retorno da API, quando o arquivo submetido não possui vírus. Na Figura 3 é exibido um exemplo de retorno quando o arquivo submetido possui vírus. Já na Figura 4 é apresentada a tela principal do antivírus.

Figura 1 – Exemplo de chamada da API

```
def scanningFile(filepath):
    # Configure API key authorization: Apikey
    configuration = cloudmersive_virus_api_client.Configuration()

    # ------ API KEY
    configuration.api_key['Apikey'] = '[redacted]'



    # create an instance of the API class
    api_instance = cloudmersive_virus_api_client.ScanApi(cloudmersive_virus_api_client.ApiClient(configuration))
    #input_file = 'C:\\Users\\Enos\\Downloads\\vtex.json' # file | Input file to perform the operation on.
    input_file = filepath

    try:
        # Scan a file for viruses
        api_response = api_instance.scan_file(input_file)
        pprint(input_file)
        pprint(api_response)
        return(api_response.clean_result)
    except ApiException as e:
        print("Exception when calling ScanApi->scan_file: %s\n" % e)
```

Fonte: Autoria Própria

Figura 2 - Retorno quando um arquivo está limpo

```
'C:/Users/Enos/Downloads/DiagramaEmBranco.pdf'
{'clean_result': True, 'found_viruses': None}
```

Fonte: Autoria Própria

Figura 3 - Retorno quando um arquivo está infectado com vírus

```
'C:/Users/Enos/Downloads/eicar_com.zip'
{'clean_result': False,
 'found_viruses': [{'file_name': 'stream',
                    'virus_name': ' Win.Test.EICAR_HDB-1'}]}
```

Fonte: Autoria Própria

Figura 4 - Tela principal do antivírus

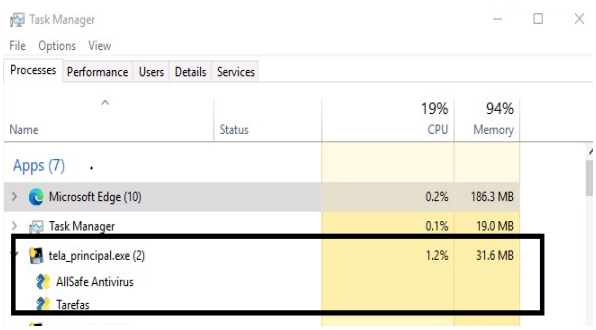
Fonte: Autoria Própria

RESULTADOS

Para realizar os testes de vírus é necessário primeiramente se preocupar com a segurança do computador, onde vão ser realizados os testes. Para isso, foi utilizada uma máquina virtual (vitualbox) para realização dos testes, já que é necessário desligar todos os meios de

segurança do sistema operacional. Para a realização dos testes foi utilizado um vírus de teste chamado EICAR. Esse vírus de teste é projetado para verificar a operação dos aplicativos de antivírus e foi desenvolvido pelo European Institute for Computer Antivirus Research (EICAR). O vírus de teste não é um objeto malicioso e não contém um código executável para o seu dispositivo, mas os aplicativos antivírus da maioria dos fornecedores o identificam como uma ameaça ("Sobre o vírus de teste EICAR", KASPERSKY, 2022). O antivírus conseguiu verificar que o arquivo baixado estava infectado com um vírus de teste. Também é importante informar que API que está sendo utilizada, já foi testada pela própria CLOUDMERSIVE descartando possíveis preocupações em relação à confiabilidade do resultado.

Figura 5 - Gerenciador de tarefas do Windows



Fonte: Autoria Própria

Para os testes de desempenho foi utilizado o próprio gerenciador de tarefas do sistema operacional Windows que mostra o consumo de CPU, memória e disco. Durante os testes, o antivírus consumiu 1.2% do processador e um máximo de 34MB de memória RAM. Este baixo consumo, está relacionado ao processo de verificação de arquivos, que não é realizado na máquina local, mas na API em nuvem, economizando recursos computacionais no computador onde o antivírus estiver instalado. Na Figura 5 é exibido a o gerenciador de tarefas com o consumo de recursos do antivírus.

DISCUSSÃO

Mesmo o antivírus desenvolvido tendo atingido o objetivo de detectar um vírus em um arquivo infectado, é importante destacar alguns aspectos que merecem atenção. Um deles está relacionado a arquivos grandes. Como o antivírus precisar fazer o *upload* do arquivo para ser verificado, em casos de arquivos muito grandes o usuário precisará dispor de uma conexão de internet de melhor qualidade. Outro aspecto está relacionado à necessidade de o usuário estar sempre conectado à internet, para que as ferramentas de detecção de vírus possam ser utilizadas.

CONCLUSÃO

O alto consumo de recursos computacionais, característico dos antivírus atuais, inviabiliza a utilização desses mecanismos de proteção em computadores mais antigos, deixando vulnerável uma parcela significativa dos equipamentos em uso. Para contornar essa limitação e oferecer proteção a esses equipamentos, foi desenvolvido um antivírus de baixo custo computacional que utiliza apenas três tecnologias essenciais: a linguagem de programação Python, a biblioteca gráfica (PySimpleGUI) e a API de verificação de vírus (Cloudmersive APIs). A ideia de utilizar uma API foi para que o processamento não ocorresse na máquina local, economizando assim recursos computacionais. O antivírus foi desenvolvido com um design minimalista e com o foco em baixo consumo de memória RAM e processador. O antivírus apresenta funções como: verificação se um arquivo está infectado por um vírus e verificação se um site é malicioso. Foram realizados testes de verificação de vírus, além da utilização de uma API (Cloudmersive APIs) que garante a confiabilidade de suas verificações de vírus. Com isso conclui-se que é possível desenvolver um antivírus simples e funcional com baixo custo e que consuma poucos recursos computacionais. Ainda que seja uma solução que dependa de uma conexão de internet que seja constante e de boa qualidade, trata-se de uma solução capaz de oferecer proteção a computadores mais antigos e que ainda estão em uso.

REFERÊNCIAS

1. CLOUDMERSIVE. Home. Disponível em: <https://cloudmersive.com/>. Acesso em: 21 out 2022.
2. Welcome to Python.org. Disponível em: <https://www.python.org/>. Acesso em: 21 out 2022.
3. PySimpleGUI. Disponível em: <https://www.pysimplegui.org/en/latest/>. Acesso em: 21 out 2022.
4. Antivirus e Soluções de Segurança para Internet. Disponível em: <https://www.eset.com/br/>.
5. Acesso em: 21 out 2022.
6. O que é um vírus de computador? Disponível em: <https://br.norton.com/blog/malware/what- is-a-computer-virus>. Acesso em: 21 out 2022.
7. Oracle VM VirtualBox. Disponível em: <https://www.virtualbox.org/>. Acesso em: 21 out 2022.
8. Computador lento? em: <https://netsupport.com.br/computador-lento-fatores/>.
9. Acesso em: 7 nov 2022.
10. Avast deixa o PC lento?. Disponível em: <https://blog.avast.com/pt-br/2015/07/28/o- avast-deixa-o -computador-lento/>. Acesso em: 7 nov 2022.
11. Avast | Antivírus grátis e fácil. Disponível em: <https://www.avast.com/pt-br/index-tcpc>. Acesso em: 7 nov 2022.
12. QUE, O. O que é API e para que serve? Cinco perguntas e respostas. Disponível em:
13. <https://www.techtudo.com.br/listas/2020/06/ o-que-e-api-e-para-que-serve-cinco-perguntas- e-respostas.ghtml>. Acesso em: 7 nov 2022.
14. O que é um endpoint de API? Disponível em: <https://www.cloudflare.com/pt-br/learning/security/api/what-is-api-endpoint.>. Acesso em: 7 nov 2022.
15. ALEXEY MALANOV. Manual básico do antivírus: Assinaturas, vírus e desinfecção. Disponível em: <https://www.kaspersky.com.br/blog/signature-virus-disinfection/6708/.>. Acesso em: 7 nov 2022.
16. Sobre o vírus de teste EICAR. Disponível em: <https://support.kaspersky.com/KESS/3.0/pt- BR/147734.htm>. Acesso em: 7 nov 2022.
17. Eicar – Y. Disponível em: <https://www.eicar.org/>. Acesso em: 7 nov 2022. LUCAS THIAGO. TÉCNICAS ULTRALEVES PARA DETECÇÃO DE MALWARE BASEADA EM ASSINATURAS PARA REDES DE COMPUTADORES. Disponível em: <https://mostra- de-tccs-bcc.github.io/TCC-BCC-Bauru-2016/gouvea/thesis-gouvea.pdf>. Acesso em: 20 nov 2022