Proteger para Prevenir: Avaliação de Métodos Anti-Keylogger em Segurança da Informação

Uriel do C. Andrade[1],
Daniel de Oliveira Capanema[1],
Rafael Henriques Noqueira Diniz[1]

[1]Instituto de Ciências Exatas e Informática
Pontifícia Universidade Católica de Minas Gerais (PUC-MG)
Caixa Postal 1.686 – 30535.901 – Belo Horizonte – MG – Brasil

uandrade@sga.pucminas.br

danielcapanema@pucminas.br, 557643@sga.pucminas.br

Abstract. *In this article, we analyzed the effectiveness of detection and mitigation tools for keyloggers, focusing on solutions that are available and widely used by end users. International tests were conducted on malware that uses heuristics to detect threats, including Windows Defender, Avast, AVG, Bitdefender, MalwareBytes, and AV Total. The results revealed differences in detection efficiency depending on the malicious file format (e.g.,* `.exe`*,* `.zip`*,* `.xlsm`*). While some antivirus programs perform well in blocking keyloggers, others show significant weaknesses, especially in detecting malicious files and macros. The research highlighted vulnerabilities in these tools and proactively emphasized the importance of raising user awareness and improving security systems to reduce the risks of attacks.*

Resumo. *Neste artigo, analisamos a eficácia das ferramentas de detecção e mitigação de keyloggers, com foco em soluções que estão disponíveis e são amplamente utilizadas pelos usuários finais. Testes internacionais foram realizados em vírus que usam heurística para detectar ameaças: Windows Defender, Avast, AVG, Bitdefender, MalwareBytes e AV Total. Os resultados mostraram diferenças na eficácia da detecção dependendo do formato do arquivo malicioso (por exemplo,* `.exe`*,* `.zip`*,* `.xlsm`*). Embora alguns antivírus façam um bom trabalho no bloqueio de keyloggers, outros apresentam sérias fraquezas, especialmente quando se trata de encontrar arquivos e macros maliciosos. A pesquisa destacou vulnerabilidades nessas ferramentas e reforçou de forma proativa importância de conscientizar os usuários e aprimorar sistemas de segurança para reduzir os riscos de ataques.*

1. Introdução

Nos últimos anos, temos visto avanços na tecnologia digital. Da proliferação dos *smartphones* à introdução da inteligência artificial na nossa vida quotidiana, a sociedade moderna está imersa num mundo conectado e digital. A facilidade de acesso à informação, comunicação e automação tornaram-se um núcleo central do mundo tecnológico. No entanto, estes avanços tecnológicos também trouxeram muitos desafios e ameaças à segurança cibernética.

Neste ambiente dinâmico e em constante evolução, o *malware* tem sido notícia como uma ameaça à segurança cibernética. Cada avanço tecnológico traz novas falhas de segurança que os malfeitores podem explorar. Seja por rendimentos ilegais, espionagem industrial ou violações de privacidade, os criminosos digitais estão sempre à procura de novas formas de comprometer sistemas e roubar informações sensíveis.

Entre as diversas ameaças que aparecem neste contexto, os *keyloggers* destacam-se como ferramentas maliciosas de alta furtividade. Muitas vezes operando em segredo, esses programas capturam e registram cada pressionamento de tecla, expondo usuários a riscos graves, como o roubo de senhas, dados bancários e outras informações confidenciais.

Essa realidade ressalta a importância de compreender as diferentes estratégias utilizadas pelos *keyloggers*, bem como os métodos de defesa disponíveis. A evolução contínua dessas ameaças exige soluções cada vez mais eficazes, capazes de proteger usuários contra ataques cibernéticos em um cenário digital em constante transformação.

1.1. Problema

A tecnologia está se desenvolvendo rapidamente, proporcionando recursos mais fáceis para trabalhar, estudar e para a vida diária. Mas esses avanços ainda colocam os usuários em grande risco, principalmente quando se trata de segurança digital. Um grande problema neste caso é a proliferação de *keyloggers*, programas maliciosos que podem capturar informações confidenciais, como senhas e informações bancárias.

Keyloggers são uma ameaça crescente que afeta pessoas e organizações em todo o mundo, causando sérios danos e comprometendo informações importantes. De acordo com estatísticas do (Intelligence, 2023) do FBI, as perdas com crimes cibernéticos chegarão a USD 10,3 bilhões em 2022, com foco em fraudes de e-mail comercial (BEC) e surtos de ransomware para evitar impactar empresas e usuários finais. O phishing, frequentemente associado a keyloggers, é o crime cibernético mais comum, com mais de 300 mil reclamações recebidas no mesmo ano.

Dada a confidencialidade destes serviços e a sua capacidade de operar em muitas áreas, é importante desenvolver estratégias eficazes para reduzir estes riscos. É importante analisar e comparar métodos de combate aos *keyloggers* para garantir que os benefícios desta ferramenta não sejam comprometidos pelos perigos que pode causar. Compreender os perigos destes riscos e implementar medidas de segurança fortes é crucial para proteger dados e informações importantes.

1.2. Objetivo principal

Por meio de um ambiente controlado, o objetivo principal deste trabalho é investigar detalhadamente o funcionamento de *keyloggers* e realizar uma análise comparativa entre diferentes métodos de detecção e mitigação. Além de compreender o comportamento desses programas maliciosos, busca-se avaliar a eficácia de cada método na identificação e neutralização de *keyloggers*, fornecendo uma visão clara sobre quais abordagens são mais eficazes para proteção contra essa ameaça.

Esta análise será conduzida comparando diversas ferramentas de proteção, avaliando aspectos como eficiência na detecção, impacto no desempenho do sistema, facilidade de uso, frequência de atualizações, e modelo de licenciamento. Dessa forma, o

estudo pretende identificar as soluções mais completas e acessíveis, visando fortalecer as estratégias de segurança digital tanto para profissionais quanto para usuários comuns.

1.3. Objetivos específicos

Os objetivos específicos desta pesquisa são:

- Criar um ambiente controlado para analisar os mecanismos de operação dos *keyloggers*, observando como eles capturam dados e se mantêm ocultos.
- Identificar e avaliar a eficácia de diversos métodos de detecção e mitigação de *keyloggers*, considerando diferentes abordagens técnicas.
- Realizar uma análise comparativa entre os métodos de detecção, destacando quais são mais eficientes na prevenção de *keyloggers* em diferentes cenários.
- Fornecer recomendações baseadas em resultados empíricos sobre as melhores práticas e estratégias para a detecção e neutralização de *keyloggers*, contribuindo para o aprimoramento da segurança digital.

1.4. Contribuições Esperadas

A partir da elaboração do trabalho e obtenção dos resultados, espera-se que este estudo contribua para a comunidade técnica e científica ao oferecer uma análise detalhada da eficácia de métodos anti-keylogger. Dado que as pesquisas sobre este tema, especificamente em sistemas operacionais de uso comum, são limitadas, este trabalho possibilitará a comparação entre diferentes ferramentas de detecção e mitigação de *keyloggers*, oferecendo insights sobre as soluções mais eficientes para o problema em questão.

2. Fundamentação teórica

2.1. Segurança da informação

A segurança da informação é uma preocupação crítica para organizações no cenário digital contemporâneo. Como destaca (Bhaharin et al., 2019) , a proteção e segurança das informações organizacionais têm se tornado cada vez mais desafiadoras na era da Indústria 4.0. Isso é atribuído ao surgimento de ameaças sofisticadas à segurança.

O surgimento da Indústria 4.0 trouxe conectividade sem precedentes e integração de tecnologias digitais nos processos organizacionais. Embora essa conectividade tenha facilitado a eficiência e a inovação, também expôs as organizações e seus usuários a uma ampla gama de ameaças cibernéticas. Desde software malicioso e ataques de *phishing* até violações de dados e ameaças internas, o cenário de ameaças moderno é caracterizado por sua complexidade e diversidade.

2.2. *Malware*

Malwares, abreviação de *malicious software* (software malicioso), é definido pelos autores (Singh et al., 2021) como:"*qualquer código ou programa escrito por hackers com a intenção de causar danos a um sistema sem o consentimento do usuário*", ele representa uma ameaça complexa e crescente no mundo digital.

Nesse mesmo contexto, Steve Morgan(2021) nos traz a informação com relatórios científicos e empresariais, que aproximadamente 1 milhão de arquivos de *malware* são criados todos os dias, e o crime cibernético prejudica a economia mundial em aproximadamente US 6 trilhões anualmente até 2021.

2.3. *Keylogger*

De acordo com os autores (Wajahat et al., 2019), *um keylogger é uma ferramente de monitoramento que possui uma certa ou total capacidade de capturar cada pressionamento de tecla de um teclado e armazenar em arquivos de log.*

Também conhecidos como registrador de pressionamentos de tecla, os *keyloggers* é um tipo de *spyware*, que podem ser utilizados tanto em contextos éticos quanto em situações que desconsideram a ética. Em contextos éticos, eles são empregados em empresas onde *"o monitoramento é um fator importante para manter a estabilidade da rede"* (Tuli and Sahu, 2013) e para realizar a vigilância de crianças na navegação à internet. No entanto, em cenários onde a ética não é levada em conta, *keyloggers* podem ser usados de forma maliciosa para roubo de informações pessoais, como senhas e dados bancários, espionagem ou outras atividades ilícitas.

Existem dois tipos principais de variações dessa ferramenta, cada uma com suas peculiaridades e mecanismos de funcionamento distintos. O primeiro tipo é o *keylogger* de hardware, (Singh et al., 2021). Após os dados serem salvos na memória interna eles podem ser acessados de várias maneiras, dependendo do design do dispositivo e das intenções do atacante.

O segundo tipo é o de software, um programa que é instalado no sistema do computador. *O keylogger de software intercepta dados que viajam pelo teclado e pelo sistema operacional. Ele coleta eventos de pressionamento de tecla, armazenando-as em um local remoto e depois os transmite ao invasor que instalou o keylogger.* (Ahmed et al., 2014)

2.4. Heuristicas de detecção

Heurísticas de detecção são técnicas avançadas usadas para identificar ameaças que não são detectadas por métodos tradicionais. Algumas das principais são:

Técnica de anti-hook: "Esta tecnologia é baseada no uso de ganchos de API, que são utilizados por todos os processos (visíveis ou ocultos) para implementar ganchos. Hooks são um conjunto de métodos usados para alterar o comportamento de um sistema operacional ou aplicativo. Isto pode interromper o fluxo de chamadas de função ou mensagens entre diferentes componentes do computador. O sistema verifica todos os processos, executáveis estáticos e DLLs (bibliotecas de vínculo dinâmico) para identificar processos ou arquivos suspeitos que usam ganchos."(Solairaj et al., 2016)

Técnica HoneyID: Descrito como uma ferramenta capaz de "atrair e capturar o invasor"(C. et al., 2023), servindo para enganar e monitorar suas atividades. Ele opera como um honeypot, criando iscas no sistema para atrair invasores, registrando suas interações suspeitas e coletando informações sobre seus comportamentos.

Detecção Baseada em Assinaturas: *É um recurso de malware que encapsula a estrutura do programa e identifica cada malware exclusivamente. A abordagem de detecção baseada em assinatura é amplamente usada em antivírus comerciais. Essa abordagem é rápida e eficiente para detectar malware conhecido, mas insuficiente para detectar malware desconhecido.* (Aslan and Samet, 2020)

3. Trabalhos relacionados

A análise sobre a ação dos *keyloggers* fornece uma visão detalhada sobre a definição, os diferentes tipos e as técnicas de detecção e prevenção, conforme discutido por (Singh et al., 2021). O autor aborda amplamente a importância de proteger informações confidenciais contra esses programas maliciosos e apresenta métodos proativos, como o uso de softwares *anti-malware* e a verificação de entradas de inicialização. O trabalho também destaca estratégias específicas para dispositivos móveis, considerando suas características distintas.

Focando em *keyloggers* de software, (Wajahat et al., 2019) aprofundaram a análise de suas operações tanto em modo kernel quanto no espaço do usuário, ilustrando exemplos históricos de uso malicioso, como o roubo de credenciais bancárias. Esses autores propuseram uma solução baseada na simulação de teclas e padrões de entrada/saída, demonstrando eficácia na detecção e aplicabilidade prática, especialmente em dispositivos móveis.

(Solairaj et al., 2016) exploraram uma ampla gama de técnicas de detecção, incluindo abordagens baseadas em assinaturas e comportamentos, como Anti-Hook e HoneyID. Além disso, propuseram o uso de Support Vector Machines (SVM) para aprimorar a detecção de *keyloggers*. Essa técnica, particularmente relevante para dispositivos móveis, mostrou-se eficaz na superação de desafios específicos, proporcionando maior precisão na identificação de ameaças. Por outro lado, (Aslan and Samet, 2020) voltaram-se para métodos avançados de detecção de *malware*, como o aprendizado profundo, aplicando-os ao combate de *keyloggers*. A pesquisa evidenciou o potencial dessas técnicas para identificar padrões complexos e oferecer soluções robustas e adaptáveis a diferentes plataformas, com especial foco em dispositivos móveis.

(Singh et al., 2021), (Wajahat et al., 2019), (Solairaj et al., 2016) e (Aslan and Samet, 2020) relataram avanços na detecção e bloqueio de *keyloggers*. Este estudo, contudo, avalia ferramentas que aplicam essas técnicas, priorizando sua acessibilidade e eficácia em cenários do mundo real.

4. Metodologia

4.1. Ambiente

Um passo crucial para garantir o sucesso do monitoramento, adaptação e codificação é a seleção cuidadosa das ferramentas adequadas para segurança e implementação.

4.1.1. Sistema Operacional Raiz

Para testes, o Ubuntu (24.04) foi selecionado como sistema operacional base devido à sua robustez, estabilidade e atualizações regulares de segurança. A arquitetura baseada em permissões, quando combinada com o VirtualBox, fornece um ambiente seguro e garante os procedimentos necessários para avaliar os métodos de detecção de *keyloggers*.

4.1.2. Virtual Box

O VirtualBox (7.1.2) foi utilizado no teste devido à sua estabilidade, facilidade de uso e compatibilidade com diversos sistemas. O software permite implantação rápida e eficiente e recursos como suporte para vários sistemas convidados, o que é fundamental para o gerenciamento de vários eventos.

Sua natureza de código aberto, gratuita e com atualizações regulares, garante um ambiente seguro e robusto. A virtualização isola os testes do sistema host, proporcionando segurança e controle nos cenários avaliados, sendo uma escolha ideal para estudos de métodos anti-*keylogger*.

4.1.3. Sistema Operacional Virtualizado

O sistema operacional selecionado para a virtualização e para rodar os testes foi o Windows 10. Essa escolha foi feita devido à sua ampla utilização em todo o mundo, o que o torna um alvo comum para *keyloggers* e outras ameaças de *malware*. A popularidade do Windows 10 é comprovada, pois ainda detém mais de 60% de participação no mercado, apesar do lançamento do Windows 11. (Hardware, 2023)

Essa vasta adoção permite que os testes realizados reflitam cenários mais próximos da realidade, aumentando a relevância das análises e conclusões obtidas em relação à proteção contra ameaças em ambientes reais

4.1.4. Visual studio code

Para análise do código *keylogger* foi utilizado o Visual Studio Code (VS Code 1.94), escolhido por sua versatilidade e adequação para revisão, teste e modificação de código. Esta ferramenta fornece funções como realce de sintaxe e pesquisa inteligente, permitindo compreender mais facilmente a estrutura e lógica do *keylogger*. Além disso, suas ferramentas de depuração integradas facilitam o monitoramento do comportamento do sistema em tempo real enquanto realizam testes controlados em um ambiente dedicado para garantir uma operação segura. Mudanças de teste foram implementadas no código, facilitando o teste de configurações e diversos recursos. Portanto, o VS Code desempenha um papel importante na análise do *keylogger* de forma clara e segura, ajudando a avaliar os métodos anti-textitkeylogger examinados neste estudo.

4.2. Seleção do *keylogger*

Para esta parte do projeto, envolve uma busca e seleção de um código de *keylogger* já existente, que será utilizado como base para a realização dos testes e validações necessárias. O objetivo é encontrar um código que implemente as funcionalidades básicas de um *keylogger*, permitindo assim concentrar esforços nas etapas de testes, depuração e validação, sem a necessidade de desenvolvimento inicial do código. Uma vez identificado o código, as etapas do processo incluem:

- **Análise do Código**: Será realizada uma análise detalhada do código escolhido, buscando entender sua estrutura, funcionalidades e fluxo de execução.

- **Teste e Depuração**: Serão realizados testes abrangentes para garantir a funcionalidade adequada do *keylogger*, além de ajustes nas partes do código que exigirem correções.
- **Implementação de Funcionalidades**: Caso necessário, novas funcionalidades poderão ser implementadas, como a otimização do envio de dados e a melhoria da captura de informações.
- **Build e Distribuição**: Após a validação do código, será realizada a compilação utilizando o *PyInstaller* para gerar um arquivo executável (.exe) que poderá ser utilizado em sistemas Windows.
- **Teste de Eficiência**: Será realizado um teste de eficiência para verificar o desempenho do *keylogger* e sua capacidade de evadir detecção por antivírus.

4.3. Seleção das ferramentas para teste

Serão avaliados métodos para interceptar *keyloggers*, com a instalação de ferramentas *anti-malware*. A seleção das ferramentas focará nas mais utilizadas no mercado, especialmente as acessíveis ao público geral, para garantir uma análise representativa da eficácia e usabilidade dessas soluções na detecção de *keyloggers* e outras ameaças de *malware*.

5. Análise

A análise será conduzida com base nos resultados obtidos por meio da aplicação da metodologia Real World Prevention Test, que simula cenários realistas para avaliar a eficácia dos métodos e ferramentas na detecção de *keyloggers*. Essa abordagem permite examinar, de forma prática e contextualizada, o desempenho das soluções em situações que refletem ameaças reais.

O objetivo é identificar quais ferramentas apresentam o melhor desempenho na detecção e bloqueio de *keyloggers* e destacar a importância de essas soluções estarem em constante evolução para acompanhar as novas ameaças e garantir uma proteção eficaz aos usuários em cenários reais.

6. Desenvolvimento

6.1. Configuração do Ambiente

Durante esta fase, foi instalado o VirtualBox, escolhido para fornecer um ambiente controlado e seguro para os testes. Em seguida, uma imagem ISO do Windows 10 foi instalada com sucesso na máquina virtual. A instalação foi concluída sem problemas, permitindo que o sistema operacional funcionasse adequadamente. Dessa forma, nosso ambiente já estava preparado para rodar os testes.

6.2. Sobre o código *Keylogger*

Na procura de um código de *keylogger*, um atendia os padrões e a proposta do trabalho. Desenvolvido em Python, o código do autor (Yunus, 2024) foi encontrado no GitHub. Este código foi submetido a um rigoroso processo de testes, depuração e compilação para garantir sua eficácia e segurança. O *keylogger* em questão pertence à família de *keyloggers* de software, com funcionalidades de monitoramento do teclado e cliques do mouse. Ele é projetado para exfiltrar os dados coletados por meio de envio automático por email. Além disso, o código implementa técnicas de persistência e ocultação, tentando se auto-deletar ou fechar ao ser executado em sistemas operacionais específicos, o que o caracteriza como um *malware* sofisticado e voltado para a furtividade.

6.2.1. Funcionamento

O código do *keylogger* desenvolvido por *Yunus* utiliza técnicas de multithreading, permitindo a execução simultânea de várias tarefas sem comprometer o desempenho do sistema. A captura de dados ocorre de maneira eficaz e em tempo real, sendo gerenciada por quatro *threads* principais, cada uma com uma tarefa específica:

- **Captura do Mouse**: Uma thread dedicada à captura dos movimentos e cliques do mouse, garantindo que todas as interações do usuário sejam monitoradas.
- **Captura do Teclado**: Responsável por registrar as teclas pressionadas, armazenando esses dados ou enviando-os posteriormente.
- **Thread Principal**: Controla a execução do programa e a sincronização entre as demais threads, assegurando que o *keylogger* funcione corretamente.
- **Envio de Dados**: Responsável por enviar os dados capturados para o 'hacker'.

Para simular o envio de e-mails com os dados capturados, o *keylogger* se integra com a API Mailtrap, uma ferramenta que simula a rede de e-mails e permite testar a funcionalidade de envio sem afetar destinatários reais. As etapas de envio são as seguintes:

1. A thread de envio começa a conexão com a API Mailtrap após capturar uma quantidade definida de dados.
2. Os dados (teclas digitadas e registros de mouse) são formatados em um e-mail.
3. O e-mail é enviado para a caixa de entrada simulada do Mailtrap, onde pode ser inspecionado para verificar o funcionamento correto da função de envio.

O uso do Mailtrap garante que o comportamento do *keylogger* seja testado de forma segura, sem comprometer a privacidade dos dados ou violar leis de segurança cibernética.

6.2.2. Teste e Depuração

Durante o processo de testes, a funcionalidade do *keylogger* foi garantida por meio de depuração detalhada no *Visual Studio Code*, com foco em vários aspectos, como a captura das teclas e movimentos do mouse, a sincronização entre threads, a estabilidade do sistema e o envio de dados.

Testes realizados:

1. **Captura**: Cada thread foi testada para garantir que as capturas de teclas e movimentos ocorressem corretamente, sem interferências.
2. **Sincronização**: Foi verificado se o fluxo das threads não causava bloqueios ou perda de dados.
3. **Estabilidade**: O *keylogger* foi testado por períodos variados para assegurar sua continuidade sem falhas.
4. **Envio de Dados**: A integração com *Mailtrap* foi testada para verificar a formatação e o envio correto dos dados.
5. **Eficiência**: A detecção por antivírus foi analisada no *VirusTotal* para avaliar a evasão e o impacto no sistema.

Esses testes permitiram ajustes importantes, especialmente na sincronização entre as threads e na captura eficiente dos dados.

6.2.3. Build e Distribuição

O código foi compilado com o *PyInstaller* para gerar um arquivo executável (`.exe`), funcional em sistemas Windows sem necessidade de interpretador Python. O comando:

```
pyinstaller --onefile --noconsole keylogger.py
```

O parâmetro `--noconsole` garantiu que o executável fosse gerado sem abrir janelas de console, aumentando a discrição da aplicação.

6.3. Métodos de Infecção do *Keylogger*

Três métodos principais de infecção (`.exe`, `.zip` e `.xlsm`) foram utilizados para descrever a situação real e avaliar o desempenho auditivo do dispositivo especificado.

6.3.1. .exe: Download Direto do Executável

O método de download direto do executável envolve o fornecimento do *keylogger* por meio de um arquivo executável baixado da internet. Embora simples, enfrenta desafios devido à detecção por antivírus.

- **Link Malicioso**: O usuário é direcionado a um link malicioso, frequentemente por phishing ou e-mails enganosos, que hospeda o executável do *keylogger*.
- **Download**: Ao clicar no link, o arquivo executável é baixado para o sistema da vítima, sendo monitorado por ferramentas de segurança.
- **Execução**: O usuário precisa abrir manualmente o arquivo, o que pode gerar desconfiança se não for disfarçado adequadamente. Após aberto, o *keylogger* começa a capturar dados.
- **Risco de Detecção**: Este método é altamente arriscado, pois antivírus frequentemente bloqueiam o download de arquivos executáveis maliciosos.

6.3.2. .xlsm: Planilha Excel com Código VBA

Foi criada uma planilha Excel com código VBA para instalar e executar o *keylogger*. Ao abrir a planilha, o código VBA instala e executa o *keylogger* silenciosamente, sem alertar o usuário, permitindo a captura de dados de forma discreta.

- **Execução do Código VBA**: O código VBA é ativado automaticamente ao abrir a planilha e instala o *keylogger*.
- **Instalação Silenciosa**: O código executa a instalação sem alertar o usuário, evitando detecção.
- **Execução do *Keylogger* Silenciosamente**: O *keylogger* é iniciado imediatamente após a instalação e começa a capturar dados.

6.3.3. .zip: Arquivos Compactados

O método .zip envolve encapsular o *keylogger* em um arquivo compactado para evitar a detecção durante o download.

- **Compactação**: O executável é compactado, dificultando a detecção por antivírus.
- **Descompactação**: O usuário é induzido a descompactar o arquivo, sem levantar suspeitas.
- **Instalação do *Keylogger***: Após a descompactação, o *keylogger* é executado manualmente, capturando dados.

6.4. Ferramentas *antimalware* selecionadas para testes e suas heurísticas

O **Windows Defender**, desenvolvido pela Microsoft, é uma solução integrada ao sistema operacional Windows que oferece proteção contra vírus, *malware* e outras ameaças. Utiliza técnicas como *Hook* e detecção por anomalias para identificar comportamentos suspeitos. A versão mais recente disponível é a 10.8750, com recursos aprimorados de resposta em tempo real e gerenciamento de vulnerabilidades.

O **Avast**, da Avast Software, é amplamente reconhecido por sua especialização na proteção contra ransomware e outras formas de *malware*. Ele combina técnicas de *Hook*, *HoneyID* e análise comportamental para detectar e mitigar ameaças. Sua versão atual, 24.8, inclui melhorias em proteção contra *phishing* e desempenho.

O **AVG**, desenvolvido pela AVG Technologies (agora parte da Avast), é focado em segurança para navegação online. Ele utiliza técnicas como *Hook*, *HoneyID* e análise comportamental. A versão mais recente, 24.8, compartilha a base tecnológica com o Avast, oferecendo funcionalidades robustas contra ataques.

O **Malwarebytes**, da Malwarebytes Inc., é uma ferramenta dedicada à identificação de ameaças com base no comportamento. Faz uso de técnicas de *Hook* e análise heurística para proteger sistemas de infecções avançadas. A versão atual, 5.3, apresenta proteção aprimorada contra exploits e suporte expandido para dispositivos móveis.

O **Total AV**, produzido pela Protected.net Group, fornece proteção em tempo real contra *malware*, incorporando técnicas como *Hook*, análise comportamental e monitoramento em tempo real. A versão mais recente, 2024 Premium 11.0, é conhecida por sua interface amigável e capacidade de detecção precisa.

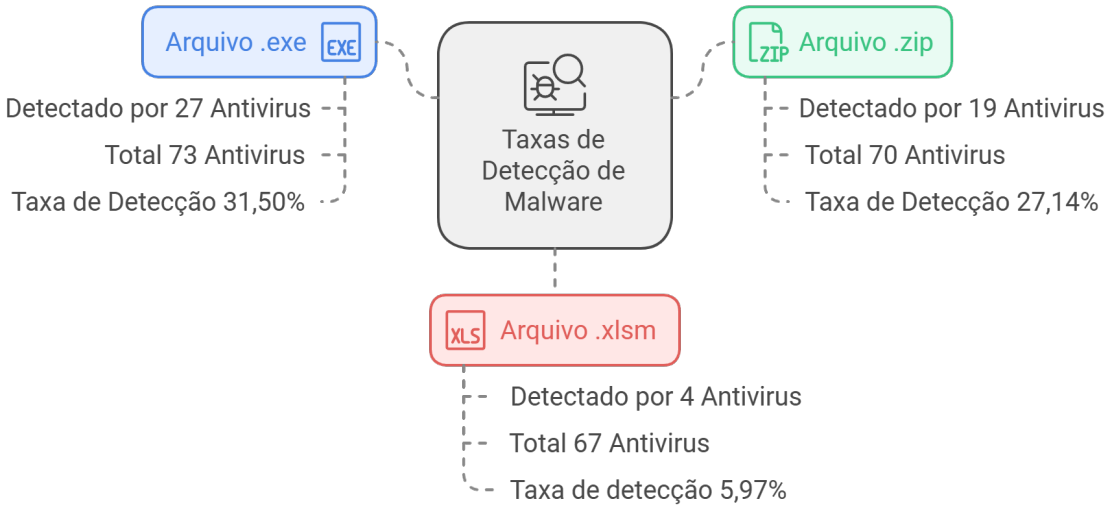
Por último, o **Bitdefender**, desenvolvido pela Bitdefender LLC, é uma solução que combina heurística avançada com análise comportamental, utilizando também técnicas de *Hook* para identificação de ameaças. A versão mais recente, Total Security 28.1, adiciona funcionalidades como proteção para redes Wi-Fi e gerenciamento integrado de senhas.

Essas ferramentas representam o estado atual do mercado e demonstram a necessidade contínua de evolução para enfrentar novas formas de *malware*, incluindo *keyloggers*.

7. Resultados

7.1. Eficiência do *keylogger*

Os testes avaliaram a capacidade do *keylogger* de evadir antivírus, utilizando o *VirusTotal* para analisar arquivos (`.exe`, `.zip`, `.xlsm`) com diversos motores como Arcabit, Avast, Kaspersky, McAfee e ESET-NOD32.



Os resultados do VirusTotal são consistentes com testes realizados no artigo, apoiando a confiança da plataforma na detecção de *malware*. A diversidade de sistemas antivírus torna o VirusTotal uma ferramenta poderosa para avaliação de ameaças.

7.2. Tabela de resultados

Os resultados a seguir foram obtidos a partir de testes realizados com a metodologia ”Real World Prevention Test”, onde cada método de infecção foi simulado cinco vezes em cada ferramenta *anti-malware*, a fim de avaliar a eficiência de detecção e prevenção do *keylogger*.

Ferramentas	Versão	.exe	.zip	.xlsm
Windows Defender	10.8750	Sucesso	Intermediário +	Fracasso
Avast	24.8	Sucesso +	Intermediário ++++	Sucesso ++
AVG	24.8	Sucesso +	Intermediário ++++	Sucesso +++
Bit Defender	28.1	Sucesso +	Intermediário ++	Sucesso +++
MalwareBytes	5.3	Sucesso	Intermediário	Sucesso ++
Total AV	11.0	Sucesso +	Intermediário +++	Sucesso +++

Tabela 1. Ferramentas de detecção e suas mitigações

Legenda:
Sucesso - Detectou o programa malicioso imediatamente.
Intermediário - Detectou após múltiplas tentativas ou etapas.
Fracasso - De forma alguma conseguiu detectar o programa malicioso.
+ - Indica maior eficiência. Mais símbolos significam detecção mais rápida ou em melhores condições.

7.3. Observações dos resultados

Windows Defender

- **Arquivo `.exe`**: A detecção é inconsistente. Em alguns testes, o arquivo malicioso é detectado, enquanto em outros não. No entanto, em todos os casos, a execução é bloqueada com um aviso de programa malicioso.

- **Arquivo `.zip`**: Não detecta o conteúdo malicioso dentro do arquivo compactado. Após a extração, a detecção é inconsistente, mas sempre bloqueia a execução com um aviso de programa malicioso.
- **Arquivo `.xlsm`**: Não identifica o download do arquivo malicioso feito via código VBA no `.xlsm`, permitindo sua execução sem bloqueios.

Avast

- **Arquivo `.exe`**: Detecta e bloqueia o arquivo malicioso imediatamente, impedindo o download ainda durante o processo de download do arquivo temporário. Também bloqueia posteriormente o download do arquivo `.exe` na fonte original.
- **Arquivo `.zip`**: Após a extração, o arquivo malicioso é detectado e colocado em quarentena, e o próprio arquivo `.zip` é colocado em quarentena após uma segunda tentativa de extração. O Avast posteriormente também bloqueia o download do `.zip` na fonte.
- **Arquivo `.xlsm`**: Detecta o arquivo .exe após a execução de download do VBA e desativa futuras execuções do Macro no arquivo xlsm. Consequentimente o *keylogger* não é executato.
- **Acesso**: Remove o acesso remoto.

AVG

- **Arquivo `.exe`**: Detecta e bloqueia o arquivo malicioso imediatamente, impedindo o download durante o processo de download do arquivo temporário. Também bloqueia o download do arquivo `.exe` na fonte original.
- **Arquivo `.zip`**: Após a extração, o arquivo malicioso é detectado e colocado em quarentena, e o próprio arquivo `.zip` também é colocado em quarentena após uma segunda tentativa de extração. O AVG posteriormente também bloqueia o download do `.zip` na fonte.
- **Arquivo `.xlsm`**: Bloqueia o acesso do código VBA aos diretórios da máquina, impedindo o download e a execução do código.
- **Acesso**: Remove o acesso remoto.

Bitdefender

- **Arquivo `.exe`**: Detecta e bloqueia o arquivo malicioso imediatamente, impedindo o download durante o processo de download do arquivo temporário. Também bloqueia o download do arquivo `.exe` na fonte original.
- **Arquivo `.zip`**: Após a extração, o arquivo malicioso é detectado e bloqueado, mas de maneira lenta, permitindo sua execução momentânea antes do bloqueio.
- **Arquivo `.xlsm`**: Bloqueia o acesso do código VBA aos diretórios da máquina, impedindo o download e a execução do código.
- **Acesso**: Remove o acesso remoto.

MalwareBytes

- **Arquivo `.exe`**: Não detecta o arquivo durante o download, mas bloqueia sua execução quando o identifica como *malware*, colocando-o em quarentena.
- **Arquivo `.zip`**: Não detecta o conteúdo malicioso dentro do arquivo compactado durante o download ou na extração. Apenas bloqueia o arquivo `.exe` ao ser executado, colocando-o em quarentena.
- **Arquivo `.xlsm`**: A detecção e o bloqueio ocorrem apenas durante a execução do `.exe`. O download do `.exe` é permitido, mas sua execução é bloqueada. (Apenas na execução o arquvo é detectado) *Keylogger* não é executato.

Total AV

- **Arquivo `.exe`**: Detecta e bloqueia o arquivo malicioso imediatamente, impedindo o download durante o processo de download do arquivo temporário. Também bloqueia o download do arquivo `.exe` na fonte original.
- **Arquivo `.zip`**: Não detecta o conteúdo malicioso durante o download. Após a extração, o arquivo malicioso é detectado e bloqueado. Não exclui o .zip e nem bloqueia seu downloand na fonte.
- **Arquivo `.xlsm`**: Bloqueia o acesso do código VBA aos diretórios da máquina, impedindo o download e a execução do código.

Nenhum dos programas antivírus testados detecta o arquivo `.xlsm` como malicioso, nem o conteúdo do arquivo `.zip` durante o primeiro momento de download.

8. Conclusão

De acordo com os resultados obtidos, verifica-se que a eficácia da proteção varia dependendo do método de infecção utilizado pelo *keylogger* e do tipo de arquivos maliciosos.

Resultados Quantitativos

Arquivos `.exe`: Avast, AVG e Total AV bloquearam 100% dos arquivos, enquanto o Windows Defender teve desempenho inconsistente.

Arquivos `.zip`: Embora algumas ferramentas detectassem o *malware* após extração, a proteção foi insuficiente durante o download do arquivo compactado.

Arquivos `.xlsm`: Nenhum antivírus detectou o arquivo diretamente, mas ferramentas como Avast e AVG bloquearam o código VBA malicioso subsequente.

Interpretação dos Dados

A detecção eficaz depende de métodos heurísticos e comportamentais. A falha na detecção de arquivos `.zip` e `.xlsm` destaca vulnerabilidades.

Síntese Final

Aplicativos como Avast, AVG e Bitdefender se destacaram na detecção de diferentes tipos de ameaças, incluindo *keyloggers*, enquanto o Windows Defender demonstrou vantagens

significativas, especialmente na detecção de arquivos compactados e macros. A análise do comportamento do *keylogger* mostrou que:

A detecção rápida é essencial para prevenir a execução de códigos maliciosos, como *keyloggers*. O método de transferência utilizando arquivos `.xlsm` continua sendo uma vulnerabilidade em muitos programas antivírus, facilitando a propagação de keyloggers. Portanto, o estudo evidenciou a necessidade de melhorias nos métodos de detecção heurística, além do desenvolvimento de estratégias de conscientização do usuário para mitigar os riscos associados a *keyloggers* e outros arquivos maliciosos.

Referências

Yahye Abukar Ahmed, Mohd Aizaini Maarof, Fuad Mire Hassan, and Mohamed Muse Abshir. Survey of keylogger technologies. *International journal of computer science and telecommunications*, 5(2), 2014.

Ömer Aslan Aslan and Refik Samet. A comprehensive review on malware detection approaches. *IEEE access*, 8:6249–6271, 2020.

Surayahani Hasnul Bhaharin, Umi Asma'Mokhtar, Rossilawati Sulaiman, and Maryati Mohd Yusof. Issues and trends in information security policy compliance. In *2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS)*, pages 1–6. IEEE, 2019.

Ekele Victoria C., Adebiyi Ayodele A., and Adebiyi Ayodele A. Keylogger detection: A systematic review. In *2023 International Conference on Science, Engineering and Business for Sustainable Development Goals (SEB-SDG)*, volume 1, pages 1–6, 2023. doi: 10.1109/SEB-SDG57117.2023.10124477.

Tom's Hardware. Windows 11 market share declines as users seemingly shift back to windows 10 url:https://www.tomshardware.com/software/operating-systems/windows-11-market-share-declines-as-users-seemingly-shift-back-to-windows-10, 2023. Acessado em: 23 nov. 2024.

Security Intelligence. 10 billion in cyber crime losses shatters previous totals url:https://securityintelligence.com/news/10-billion-in-cyber-crime-losses-shatters-previous-totals/, 2023. Acesso em: nov. 2024.

Arjun Singh, Pushpa Choudhary, et al. Keylogger detection and prevention. In *Journal of Physics: Conference Series*, volume 2007, page 012005. IOP Publishing, 2021.

A Solairaj, SC Prabanand, J Mathalairaj, C Prathap, and LS Vignesh. Keyloggers software detection techniques. In *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, pages 1–6. IEEE, 2016.

Preeti Tuli and Priyanka Sahu. System monitoring and security using keylogger. *International Journal of Computer Science and Mobile Computing*, 2(3):106–111, 2013.

Ahsan Wajahat, Azhar Imran, Jahanzaib Latif, Ahsan Nazir, and Anas Bilal. A novel approach of unprivileged keylogger detection. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–6. IEEE, 2019.

AYDIN Yunus. Keylogger detection and mitigation methods: Code for research and analysis url:https://github.com/aydinnyunus/keylogger. GitHub repository, 2024. Accessed: Nov. 2024.