

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

Uriel do C. Andrade<sup>1</sup>,  
Daniel de Oliveira Capanema<sup>1</sup>,  
Rafael Henriques Noqueira Diniz<sup>1</sup>

<sup>1</sup>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, .xls). 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, .xls). 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, os avanços tecnológicos, como a popularização dos smartphones e a introdução da inteligência artificial, transformaram a sociedade, tornando-a mais conectada e digital. No entanto, com essas inovações, surgiram também desafios significativos para a segurança cibernética, especialmente com o aumento das ameaças digitais.

Os *malwares*, como os *keyloggers*, são exemplos dessas ameaças. Esses programas maliciosos, que funcionam de forma furtiva, capturam pressionamentos de tecla,

expondo os usuários a riscos como roubo de senhas, informações bancárias e outros dados sensíveis. Dada a constante evolução dessas ameaças, é fundamental implementar métodos de defesa cada vez mais eficazes para proteger a privacidade e a segurança dos usuários em um mundo digital em constante transformação.

### **1.1. Problema**

Apesar dos inúmeros benefícios proporcionados pelos avanços tecnológicos, os riscos aumentaram proporcionalmente quando o assunto envolve os dados de seus usuários e organizações neste ambiente digital.

*Keyloggers* são uma ameaça crescente que afeta pessoas e organizações em todo o mundo, causando sérios danos e comprometendo informações importantes. Estima-se que 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, impactando 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 (Intelligence, 2023).

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**

O objetivo principal deste trabalho é investigar detalhadamente o funcionamento de *keyloggers* e realizar uma análise comparativa entre diferentes métodos de detecção e mitigação. Busca-se compreender o comportamento desses programas maliciosos e avaliar a eficácia de cada método na identificação e neutralização de *keyloggers*. Por meio de um ambiente controlado, pretende-se fornecer 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), são definidos como códigos ou programas desenvolvidos com a intenção de causar danos a sistemas, roubar informações ou prejudicar seu funcionamento sem o consentimento do usuário. Representam uma ameaça crescente e complexa no mundo digital, sendo disseminados por diversos meios, como e-mails, sites ou dispositivos infectados (Singh et al., 2021).

Neste mesmo contexto, é relatado pelo (Arun and Singh, 2018) que *"aproximadamente 1 milhão de arquivos de malware são criados todos os dias, e o crime cibernético prejudica a economia mundial em aproximadamente USD 6 trilhões anualmente, até 2021"*.

### 2.3. Keylogger

De acordo com (Wajahat et al., 2019), *"keyloggers são ferramentas 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. Heurísticas 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: (Ekele et al., 2023) "*É ferramenta capaz de atrair e capturar o invasor, servindo para enganar e monitorar suas atividades*". Ele opera como um honey-pot, criando iscas no sistema para atrair invasores, registrando suas interações suspeitas e coletando informações sobre seus comportamentos.

Detecção Baseada em Assinaturas: É uma técnica usada por antivírus para identificar malware com base em padrões específicos já conhecidos. É eficaz para detectar ameaças registradas, mas não consegue identificar malware novo ou 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.

Para (Solairaj et al., 2016) 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.

Avanços significativos têm sido feitos na detecção e bloqueio de *keyloggers* nos últimos anos. Estudos como os de (Singh et al., 2021), (Wajahat et al., 2019), (Solairaj et al., 2016) e (Aslan and Samet, 2020) destacam técnicas inovadoras nessa área. Este trabalho, no entanto, foca na avaliação de ferramentas que utilizam 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**

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 escolhido pelos benefícios de estabilidade, facilidade de uso e compatibilidade com diversos sistemas. Sua natureza de código aberto, gratuita e com atualizações regulares oferece um ambiente seguro e isolado, ideal para testes de métodos anti-keylogger.

#### **4.1.3. Sistema Operacional Virtualizado**

O Windows 10 foi escolhido para os testes devido à sua ampla utilização global, com mais de 60% de participação no mercado, o que o torna um alvo comum para *keyloggers* e *malware* (Tom's, 2023). Essa popularidade torna os testes mais relevantes, refletindo cenários próximos da realidade para análise de proteção contra ameaças.

#### 4.1.4. Visual studio code

O Visual Studio Code (VS Code 1.94), criado pela Microsoft, foi utilizado para analisar o código do keylogger devido à sua versatilidade. Com recursos como realce de sintaxe, pesquisa inteligente e depuração integrada, facilitou a revisão, testes e avaliação do keylogger escolhido para os testes.

#### 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

Nesta pesquisa, 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 (*ISO* é um arquivo que contém uma cópia exata do sistema operacional, funcionando como um "arquivo de instalação" que pode ser usado para configurar o sistema). 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 busca por um código de *keylogger*, foi encontrado um que atendia aos padrões e objetivos do trabalho. Desenvolvido em Python e disponível no GitHub, o código, criado pelo autor (Aydin, 2024), foi submetido a um processo de testes, depuração e compilação para garantir sua eficácia e segurança. Este *keylogger* pertence à categoria de *keyloggers* de software, sendo capaz de monitorar entradas do teclado e capturar cliques do mouse. Projetado para roubar os dados coletados, o *keylogger* envia automaticamente as informações por meio de e-mails.

#### 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

A funcionalidade do keylogger foi testada e depurada no Visual Studio Code, garantindo a captura de teclas, movimentos do mouse, sincronização entre threads, estabilidade do sistema e envio de dados.

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.

### 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 *anti-malware* selecionadas para testes e suas heurísticas

**Windows Defender, Microsoft, 2005:** Uma ferramenta de segurança integrada ao Windows, oferecendo proteção contra ameaças, gerenciamento de vulnerabilidades e atualização constante para garantir a segurança do sistema (Microsoft, 2024).

**Avast, Avast Software, 1988:** Famoso por proteger contra ransomware e phishing, oferece monitoramento em tempo real e foco em segurança online, detectando ameaças de forma eficiente (Software, 2024).

**AVG, AVG Technologies, agora parte da Avast, 1991:** Focado em segurança online, oferece proteção contra sites e downloads maliciosos, compartilhando a mesma tecnologia do Avast (Technologies, 2024).

**Malwarebytes, Malwarebytes Inc., 2006:** Especializado em detectar malware desconhecido, oferece proteção avançada contra exploits e monitora atividades suspeitas (Inc., 2024).

**Total AV, Protected.net Group, 2016:** Conhecido pela precisão na detecção de ameaças e interface amigável, destacando-se pela facilidade de uso e eficácia (Group, 2024).

**Bitdefender, LLC 2001:** Fornece proteção abrangente, incluindo recursos adicionais como proteção para redes Wi-Fi e gerenciamento de senhas, destacando-se pela eficácia na detecção de ameaças (LLC, 2024).

Essas ferramentas foram escolhidas para os testes, representando uma amostra das soluções contra ameaças como keyloggers, sem incluir todas as principais opções do mercado.

## 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.

Avast	AVG	Bkav Pro	CrowdStrike Falcon
DeepInstinct	Elastic	ESET-NOD32	Kaspersky
Malwarebytes	SecureAge	SentinelOne (Static ML)	Skyhigh (SWG)
Zillya			

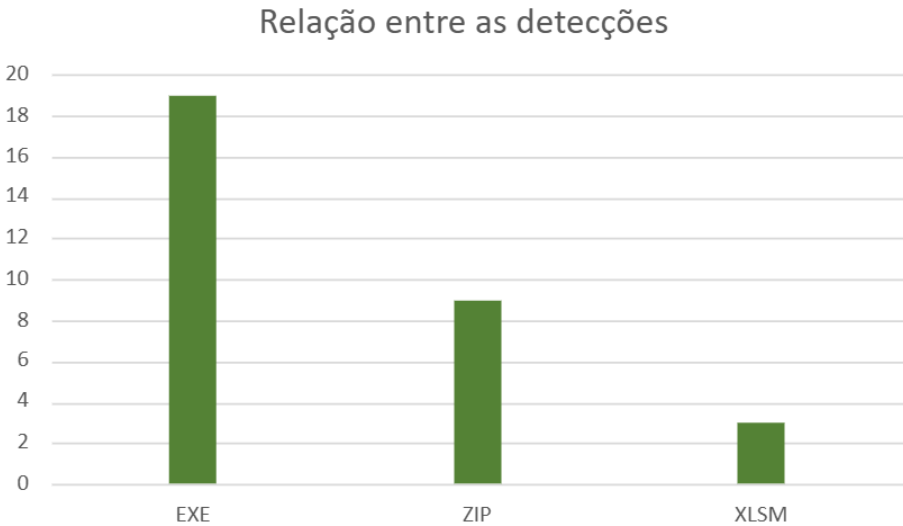
**Tabela 1. Viruz Total: Ferramentas que detectaram o keylogger exe**

Avast	AVG	DeepInstinct	Elastic
ESET-NOD32	Kaspersky	Malwarebytes	SentinelOne (Static ML)
Zillya			

**Tabela 2. Viruz Total: Ferramentas que detectaram o keylogger zip**

Elastic	Google	VirIT
---------	--------	-------

**Tabela 3. Viruz Total: Ferramentas que detectaram o keylogger xlsm**



**Figura 1. Relação representativa da quantidade de ferramentas que detectaram o keylogger nos 3 tipos de arquivos**

Os resultados obtidos pelo VirusTotal confirmam os testes realizados neste estudo, reforçando a confiabilidade da plataforma na detecção de *malware*. A diversidade de motores antivírus presentes na ferramenta demonstra sua eficácia na análise de ameaças. Apesar disso, o *keylogger* se mostrou eficiente em evadir a detecção de diversos antivírus,

evidenciando sua capacidade de operar de forma furtiva em cenários variados. Além disso, pode-se observar que, quanto mais complexo o método de infecção, menor a capacidade dos antivírus em detectar a ameaça. Isso destaca a importância de uma avaliação abrangente das ferramentas de segurança para detectar e mitigar novas ameaças.

## 7.2. Tabela de resultados dos testes em máquina

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 4. 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. Quanto mais símbolos, mais rápida e eficaz é a detecção.

Nenhum + indica o sistema menos eficiente, e ++++ representa o mais eficiente.

## 7.3. Observações dos resultados

Windows Defender:

- **Arquivo .exe:** A detecção é inconsistente. Em 40% dos casos, o arquivo malicioso é detectado, enquanto nos 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 com cerca de 60% dos casos positivos, 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. Consequentemente o *keylogger* não é executado.
- **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 arquivo é detectado) *Keylogger* não é executado.

#### 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 download 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.

**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*. Para o tipo de malware utilizado neste estudo, o resultado encontrado foi que a detecção foi mais eficaz em ferramentas com técnicas de heurística aprimoradas. 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

- 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.
- Arun and Vaishali Singh. Infringement of prevention technique against keyloggers using sift attack. In *2018 International Conference on Advanced Computation and Telecommunication (ICACAT)*, pages 1–4. IEEE, 2018.
- Aslan and Refik Samet. A comprehensive review on malware detection approaches. *IEEE access*, 8:6249–6271, 2020.
- Aydin. Keylogger detection and mitigation methods: Code for research and analysis url:<https://github.com/aydinnyunus/keylogger>. GitHub repository, 2024.
- Baharin, 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, Adebisi Ayodele A., and Adebisi 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.
- Protected.net Group. Total av antivirus. <https://www.totalav.com/>, 2024. Accessed: Nov. 2024.
- Malwarebytes Inc. Malwarebytes antivirus. <https://www.malwarebytes.com/>, 2024. Accessed: 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.
- Bitdefender LLC. Bitdefender antivirus. <https://www.bitdefender.com/pt-br/>, 2024. Accessed: Nov. 2024.
- Microsoft. Microsoft security blog. <https://www.microsoft.com/pt-br/security/blog/>, 2024. Accessed: Nov. 2024.
- Singh, Pushpa Choudhary, et al. Keylogger detection and prevention. In *Journal of Physics: Conference Series*, volume 2007, page 012005. IOP Publishing, 2021.
- Avast Software. Avast antivirus. <https://www.avast.com/pt-br/index>, 2024. Accessed: Nov. 2024.
- 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.
- AVG Technologies. Avg antivirus. <https://www.avg.com/pt-br/homepage>, 2024. Accessed: Nov. 2024.
- Tom's. 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.
- Tuli and Priyanka Sahu. System monitoring and security using keylogger. *International Journal of Computer Science and Mobile Computing*, 2(3):106–111, 2013.
- Wajahat, 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.