Fileless Malware: Uma revisão e análise sobre as formas de deteção

André Zhan†, Miguel Pombeiro†, Miguel Rocha†

†Estes autores contribuíram igualmente para este trabalho.

Resumo

Com o avanço da tecnologia, as ameaças à cibersegurança tanto dos indivíduos como das organizações estão em constante evolução. Entre essas ameaças, temos o *fileless malware*, que é o tema central deste artigo. Esta revisão inicia-se com uma breve análise do impacto deste tipo de ataque na economia, mais especificamente, no mercado da cibersegurança, e com uma definição clara do que é este *malware*. De seguida, temos uma visão geral do seu ciclo de vida, desde a sua distribuição até à execução e formas de persistência. São ainda abordadas as principais diferenças entre *malware file-based* e *fileless*, que se diferenciam pela forma como operam e persistem no sistema. Adicionalmente, são exploradas as técnicas mais comuns que atacantes costumam utilizar e explorar, para persistir no sistema. Por fim, são discutidas as estratégias modernas de deteção, que procuram combater estas grandes ameaças. Ao apresentar uma análise estruturada e breve do *fileless malware*, o artigo oferece uma compreensão abrangente deste tipo avançado de ataque.

Palavras-chave: Malware, Fileless Malware, Malware residente na memória, Deteção de Fileless Malware, Cibersegurança, Living-off-the-land

1 Introdução

Malware - abreviatura para *Malicious Software* - é o nome dado a *software* malicioso que é desenvolvido com o objetivo de comprometer dispositivos e sistemas informáticos. São programas de computador que infetam outros programas instalados num sistema de forma a obter algum tipo de vantagem, representando assim, uma ameaça à segurança de indivíduos e organizações [1]. O *malware* pode ser utilizado para vários fins, nomeadamente para roubo de informação confidencial ou causar danos a sistemas, perturbando as operações que estes estão a executar [2] [3].

Nesta era digital, que está em constante desenvolvimento, as ameaças à cibersegurança transcendem cada vez mais os *malware* tradicionais, recorrendo a vetores de ataque cada vez mais sofisticados e dissimulados [4]. Dentre estes, o aumento da proeminência dos *fileless malware* apresenta um grande desafio devido ao facto de residirem apenas na memória principal do sistema e conseguirem atuar sem deixar qualquer tipo de rasto no disco, iludindo assim, os tradicionais mecanismos de deteção baseados em assinaturas [5]. Esta mudança no paradigma deu origem à necessidade de desenvolver novas e mais inteligentes formas de detetar ameaças em tempo real para melhor monitorizar os sistemas. Hoje-em-dia, como as organizações dependem cada vez mais de sistemas interligados e infraestruturas em *cloud*, torna-se imperativo que os investigadores se foquem nesta ameaça invisível [6].

O *malware* tradicional funciona à base de ficheiros executáveis que estão guardados em disco para conseguir levar a cabo as suas intenções maliciosas, o que os torna facilmente detetáveis pelos antivírus convencionais. O funcionamento dos antivírus consiste em fazer uma análise aos discos das máquinas para procurar por ameaças desconhecidas que podem causar danos aos computadores, eliminando-as na maioria dos casos [5] [7]. Contudo, os atacantes adaptaram-se a estes métodos através de utilização de *fileless* ou *non-malware*, que tiram partido de outras ferramentas legítimas dos sistemas como a PowerShell[1], a Windows Management Instrumentation (WMI)[2], ou até mesmo *software* mais conhecido dos utilizadores como visualizadores de PDF ou *web-browsers*, para permanecerem ocultos e carregarem *scripts* maliciosas para o sistema [8]. Este comportamento furtivo, levou a que fossem exploradas outras técnicas tais como a deteção baseada no comportamento das aplicações (*behavior-based detection*), *machine learning* ou *memory forensics* [6] [9].

Este artigo faz uma análise exploratória do que são *fileless malware*, com ênfase nas técnicas por eles utilizadas, bem como nos vários mecanismos de deteção utilizados para mitigar os riscos que estes apresentam.

2 Fileless Malware

2.1 Análise do mercado de segurança para estes ataques

Com o rápido avanço das tecnologias, os *fileless malware* têm ficado cada vez mais comuns e perigosos. De acordo com Obeegadoo et al. [9], é esperado que o mercado para o desenvolvimento de novas soluções de segurança cresça bastante durante esta década. Como é possível verificar na Figura 1, está previsto um crescimento de 16.46 milhares de milhões de USD, em 2021, para cerca de 41.93 milhares de milhões de USD, em 2029.

[1] PowerShell é uma aplicação de automação pré-criada do Windows que os atacantes costumam aproveitar.
[2] É uma ferramenta do Windows que permite administradores configurarem as definições do sistema.

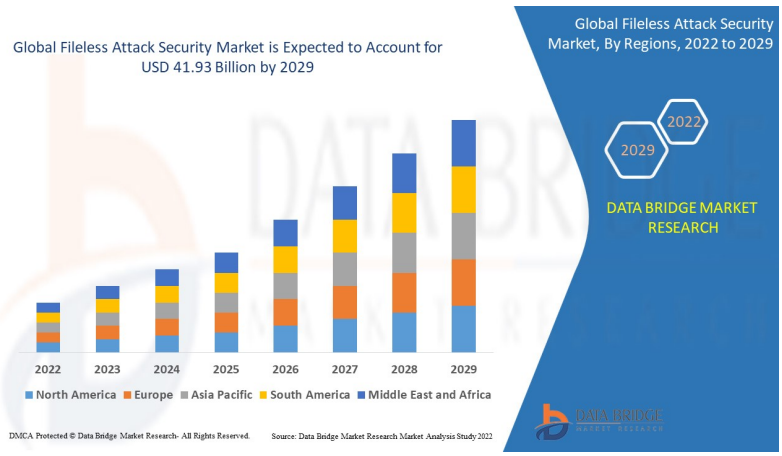

Fig. 1 Previsão de investimento na indústria de segurança de *fileless malware*. [9]

2.2 O que é um Fileless Malware?

Ao contrário do que acontece com os tradicionais ataques com *file-based malware*, que utilizam ficheiros executáveis maliciosos, os *fileless malware* tiram partido de processos legítimos do sistema como ferramentas dos sistemas operativos ou *Living-of-the-Land Binaries* (LoLBins) para atacarem e se esconderem. Uma análise e comparação mais detalhada entre ambos os tipos de *malware* é feita na Secção 3.

Ataques por *fileless malware* não requerem que seja instalado qualquer tipo de *software* ou escrito qualquer tipo de conteúdo nos discos das máquinas das vítimas, de forma a comprometer os seus sistemas. Ao invés disso, os *fileless* tiram partido de aplicações fidedignas e muito utilizadas (e.g. Microsoft Office), ou mesmo ferramentas dos sistemas (e.g. PowerShell) para correr *scripts* maliciosos e carregar código na memória primária [6]. Uma vez que estas ferramentas e as operações que executam são, normalmente, confiáveis por todos os programas de deteção, torna-se particularmente difícil distinguir estes tipos de atividades maliciosas de atividades legítimas. Esta deteção é ainda, mais dificultada pelo facto de nada ser guardado no disco, existindo apenas na memória primária e não deixando qualquer tipo de vestígio. Por outro lado, e uma vez que apenas é utilizada a memória volátil, quaisquer operações nefastas deste tipo de ataques podem ser terminadas reiniciando os sistemas. [2] [10] [11]

2.3 Ciclo de vida de Fileless Malware

De acordo com Khalid et al. [3] e Obeegadoo et al. [9] o ciclo de vida dos *fileless malware* é composto por três fases distintas. A Figura 2 representa o ciclo de vida dos *fileless malware*.

- Fase 1: Distribuição
  O atacante tenta distribuir o *malware* para as máquinas das vítimas, recorrendo às mais variadas técnicas, incluindo engenharia social para persuadir os utilizadores

a clicar em *links* ou enviando o vetor inicial do *malware* num anexo de email de *phishing* [6]. O objetivo do atacante é fazer com que o *malware* inicial seja entregue na máquina das vítimas sem ser detetado pelos programas de antivírus. Este contém, normalmente, *scripts* ou macros que irão ser escritos e executados diretamente na memória principal da máquina. [9]

- Fase 2: Persistência
  Uma das principais vantagens dos *fileless malware* é o facto de residirem sempre na memória principal do computador, o que faz com que seja também, uma das suas maiores fraquezas. Como a memória principal (RAM) é volátil, após o sistema ser reiniciado, os *scripts* que não foram guardados em nenhum ficheiro são apagados. Para acautelar esta limitação, os atacantes utilizam técnicas que alteram o registo do sistema[3], configurando-o para correr os *scripts* automaticamente, com recurso a ferramentas legítimas como o WMI, ou mesmo o escalonador de tarefas. [3] [9]
- Fase 3: Exploração
  O *malware* vai executar as ações para as quais foi desenhado, com recurso às ferramentas legitimas instaladas no sistema. Se o *malware* contém *scripts* ou macros, estas poderão ser executadas diretamente a partir da PowerShell ou do terminal. Os *fileless malware* usam ferramentas e aplicações legitimas como o Microsoft Office, o registo do Windows, ou mesmo o escalonador de tarefas para executar as suas operações [3]. Durante todo o processo, a vitima, provavelmente, nunca se vai aperceber do ataque a não ser que procure por sinais de que a memória do sistema foi comprometida [9].

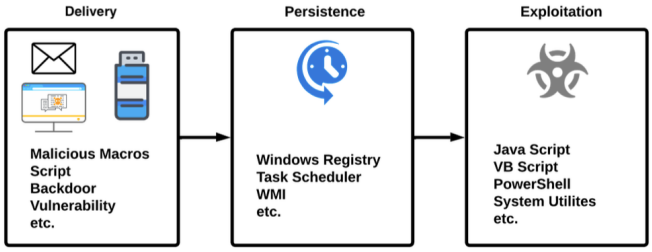

Fig. 2 Ciclo de vida de *fileless malware* descrito na literatura [3].

3 Fileless vs. File-based Malware

Os *file-based malware* infetam os sistemas, armazenando ficheiros maliciosos no disco das máquinas alvo. A sua propagação é feita através de ficheiros executáveis que são, tipicamente, provenientes de *downloads* de *sites* maliciosos, por anexos no email ou até por ficheiros transferidos fisicamente [2]. Uma vez executados, o código malicioso é escrito no disco rígido do sistema podendo assim, roubar os dados, corromper ficheiros

[3] Base de dados no sistema operativo Windows que contém configurações tanto do sistema operativo como de aplicações. Mantém dados relativos a preferências do utilizador, aplicações instaladas, e outras definições do sistema.

ou obter acesso não autorizado ao sistema. Os *softwares* de antivírus tradicionais são projetados para detetar estes tipos de *malwares*, com muita facilidade. [3]

As principais diferenças entre o *file-based malware* e o *fileless malware* podem ser verificadas na forma como ambos operam e persistem no sistema [4]. Os *file-based* são um tipo de *malware* que criam novos ficheiros no computador da vítima, como parte de um processo de instalação [2]. Já os *fileless* não dependem deste método, mais tradicional, de serem instalados no computador da vítima. Ao invés, operam diretamente a partir da memória (RAM), com recurso a ferramentas e processos já existentes, não deixando, assim, nenhum rasto permanente. Isto faz com que sejam muito mais difíceis de detetar e prevenir. [4] [9]

De acordo com Khalid et al. [3], *"Fileless and file-based malware could be equally dangerous and can cause harm to a victim's computer or network. For file-based malware detection, there exists a large state-of-the-art, while fileless malware detection is still in its early stages."*. Perante isso, é de extrema importância continuar a investir na investigação de métodos eficazes para detetar *fileless malware*.

Uma comparação detalhada entre os mais tradicionais *file-based malware* e os *fileless malware* pode ser encontrada na Tabela 1.

Tabela 1 Comparação entre *file-based* e *fileless malware*. [3] [9]

Critérios	***File-Based Malware***	***Fileless Malware***
Código-Fonte	Sim	Não
Ficheiro Malicioso	Sim	Não
Processo Malicioso	Sim	Não (Usa processos legítimos do SO)
Complexidade	Moderada	Muito Alta
Complexidade de Deteção	Moderada	Muito Alta
Persistência	Média	Baixo
Tipos de Ficheiro	- Ficheiros executáveis - *Scripts* embutidos (PDF, Word, Excel, etc)	- JavaScript - WMI - PowerShell - Flash - WScript / CScript
Alvos	Ficheiro executável para combinação única de SO/*patch*	Pode visar várias combinações de SO/diretório
Métodos de ofuscação	- Encriptação - Ficheiro de arquivo - Executável disfarçado - Executável embutido noutro ficheiro	- *Strings* fragmetadas - Encriptação - Aleatorização - Ofuscação de dados e lógica - Codificação - ASCII/Unicode escapado
Deteção por antivírus	Possível com assinatura conhecida	Possível
Deteção por *sandbox*	Requer o ficheiro presente	Não é possível

4 Técnicas de ataques Fileless Malware

Os atacantes podem aproveitar-se de diversas técnicas para persistirem no sistema, evitar a deteção e comprometer os sistemas. De acordo com a literatura, estas técnicas podem ser classificadas em diversas categorias, sendo as principais descritas a seguir.

4.1 Malware residente na memória

Esta técnica tem como base a memória principal do sistema (RAM). A memória RAM é uma memória volátil, isto é, os dados são armazenados temporariamente e são apagados quando o sistema é reiniciado. Sendo assim, e como estes tipos de ataque utilizam a memória RAM, não é gravada qualquer parte da sua atividade no disco rígido do computador. Assim sendo, a existência deste *malware* dura apenas até que o sistema operativo reinicie, aumentando assim as suas hipóteses de não ser detetado por antivírus tradicionais [9]. Existem diversos *malwares* deste tipo, como *Code Red* ou o *SQL Slammer*.

O *Code Red* e o *SQL Slammer* foram dos primeiros ataques deste tipo. *Code Red* explorou uma vulnerabilidade no servidor IIS[4] da Microsoft e propagou-se pelo protocolo TCP/IP na porta 80 (HTTP), infetando imensas máquinas. O *SQL Slammer* explorou uma vulnerabilidade no servidor SQL da Microsoft, enviando pedidos na porta 1434, causando ataques DDoS[5]. [6]

4.2 Malware do registo do Windows

O registo do Windows é uma plataforma utilizada para armazenar as definições de baixo nível do sistema operativo Windows e de algumas aplicações críticas [12]. Este *malware* pode alterar o registo do Windows para executar comandos maliciosos assim que o sistema é reiniciado [5] [9]. Esta técnica irá manter controlo total do computador de uma vítima, já que garante que o *malware* continua a ser executado mesmo após o sistema ser reiniciado. Um exemplo deste *malware* é o poweliks, que se infiltra dentro do registo do Microsoft Windows, e quando atinge os seus objetivos se autodestrói-se, não deixando qualquer rasto da sua existência. [13]

4.3 Fileless malware por rootkits

Este é um tipo de ataque baseado no *kernel*[6] e foi concebido para permanecer indetetável no sistema operativo por muito tempo. Apesar de não ser um *malware* totalmente *fileless*, pode ser classificado como tal, porque não deixa qualquer rasto no disco. Como este tipo de *malware* opera ao nível do *kernel*, pode ser difícil para os antivírus tradicionais detetá-lo. [9] [13]

[4] É um servidor web desenvolvido pela Microsoft para hospedar serviços, aplicações e sites.
[5] *Distributed Denial of Service* é um ataque informático cujo objetivo é interromper a disponibilidade de um serviço, e para tal, esse serviço é sobrecarregado com tráfego excessivo.
[6] O kernel é o núcleo do sistema operativo. É responsável por garantir que os programas sejam executados corretamente, que a memória seja alocada de forma adequada e que o processador seja utilizado de forma eficiente.

4.4 Living-off-the-Land

Este tipo de ataque é destacado pela utilização de ferramentas legítimas e integradas no Windows, como WMI e PowerShell, para executar código malicioso sem ser detetado.

Esta técnica, como todos as outras já faladas, também é de difícil deteção. Esta dificuldade deve-se ao facto dos antivírus raramente fazerem *scan* destas ferramentas nativas do sistema operativo, facilitando assim estes ataques [14]. Os atacantes podem utilizar o WMI para carregar *scripts* prejudiciais diretamente para a memória e, como se trata de um programa legítimo do Windows, o antivírus não irá restringir ou bloquear as suas ações. Já o PowerShell, que também acede às definições do sistema, pode ser utilizado para realizar estes ataques, e poderá até alterar a palavra-passe do utilizador [9]. Segundo Side Liu et al. [6], estudos recentes mostram que estas técnicas são muito utilizadas para ataques de ameaça persistente avançada, com uma taxa de prevalência de 26.26% nas amostras desse tipo.

5 Técnicas de deteção de Fileless Malware

A deteção de *fileless malware* representa um grande desafio para a cibersegurança, devido à sua natureza dissimulada, em que operam na memória primária dos sistemas, não deixando praticamente nenhum vestígio no disco. Ao contrário do *malware* dito tradicional, que guarda ficheiros executáveis, com assinaturas características, que podem ser detetadas por antivírus, os *fileless malware* utilizam ferramentas legítimas para executar o seu *payload*[7] e residem na memória do sistema. Isto faz com que sejam bastante difíceis de detetar usando estas soluções, mais tradicionais, dos antivírus. Consequentemente, novas estratégias de deteção e prevenção foram desenvolvidas por investigadores, com recurso à monitorização do comportamento dos sistemas, do conteúdo da memória (*memory dump*[8]), entre outras. Compreender e melhorar estas técnicas é muito importante, dado o crescimento e impacto económico dos *fileless malware* em ambientes empresariais e governamentais.

5.1 Deteção por análise comportamental

A análise comportamental é uma técnica que se foca na monitorização de ações e padrões de processos dentro do sistema em vez de depender de assinaturas de ficheiros. Ao estabelecer uma linha base de comportamento do sistema, desvios e anomalias podem levantar suspeitas sendo potencialmente maliciosos. Isto envolve, frequentemente, algoritmos de inteligência artificial e *machine learning*, de forma a analisar vários atributos dos sistemas, tal como a criação de processos, ligações à rede, modificações nos registos, utilização da memória e *system calls* [6]. Este tipo de análise comportamental e preditiva consegue detetar tanto *malware* conhecido, bem como desconhecido que ainda não constam das bases de dados, através da identificação de atividades suspeitas. No entanto, pode gerar falsos positivos se atividades legitimas menos comuns forem sinalizadas, exigindo assim a necessidade de uma análise por profissionais. Sendo assim, é muito importante para equipas de segurança ajustar os

[7] Parte do ataque que executa a ação nociva para o sistema.
[8] Cópia instantânea da memória volátil, RAM, de um computador, num dado instante.

parâmetros do sistema de análise comportamental, de modo a reduzir a probabilidade destes falso positivos. [3] [9] [15]

5.2 Deteção por análise forense da memória

Como os *fileless malware* residem na memória e executam a partir desta, analisar a memória volátil de um sistema é um método direto de deteção. Isto envolve capturar uma imagem instantânea da memória do sistema em execução (*memory dump*) e utilizar ferramentas especializadas como Volatility, Memory Acquisition e CAINE para examinar os seus conteúdos [9]. Os profissionais de segurança podem procurar processos ocultos, código injetado[9] (em DLLs[10], por exemplo), ligações de rede suspeitas e outros artefactos maliciosos dentro da imagem da memória [13]. A análise forense de memória permite então, revelar *malware* que está ou esteve ativo. Contudo, este método requer frequentemente conhecimentos e ferramentas muito especializadas para obter as imagens da memória. É, portanto, uma técnica mais utilizada no contexto de análise pós-incidente do que em deteção em tempo-real e de monitorização contínua. [9]

5.3 Deteção baseada em regras

Este tipo de deteção é baseado em regras criadas com base na experiência, em dados recolhidos noutros ataques ou mesmo outros padrões já conhecidos. Estas regras são então utilizadas para detetar situações suspeitas, em que programas legítimos executam comandos na linha de comandos ou na PowerShell, tipicamente criadas por *fileless malware*. Este método de deteção é muito rápido e bastante fácil de expandir, bastando, para isso, continuar a estudar estes tipos de ataques. Este tipo de deteção é utilizada por várias organizações pela facilidade na sua implementação e baixo custo computacional. No entanto, como esta técnica depende exclusivamente de padrões já identificados, é falível quando os ataques utilizam técnicas novas e mais sofisticadas, que fogem às regras já estabelecidas. Assim, é frequentemente necessária a intervenção de especialistas e de manter o seu repositório de regras atualizado com os dados catalogados de ataques mais recentes. [6] [9]

5.4 *Sandboxing* e Emulação de Execução

Replicar ataques por *fileless malware* numa máquina virtual, com um ambiente controlado e isolado (*sandbox*), permite observar o seu comportamento sem colocar em risco um sistema computacional real. Ao correr os *scripts* num ambiente isolado, se nenhum dos sistemas for afetado, então não se trata de um *malware*, caso contrário, o ambiente virtual pode facilmente ser reinicializado [4]. A partir de *sandboxes* é possível fazer a recolha de *memory dumps* de forma segura, durante a execução dos processos maliciosos, permitindo a sua posterior análise por técnicas forenses (Secção 5.2). Nesta *sandbox* podem, ainda, ser integrados modelos de IA/ML que permitem detetar e alertar de forma autónoma para a existência de ficheiros maliciosos. [9]

[9] Sequência de instruções que o atacante insere no espaço de endereçamento de um processo legítimo.
[10] Ficheiro binário do Windows que contém recursos que vários programas podem carregar.

Khalid et al. [3] propuseram um método para a deteção de *fileless malware* que utilizava várias das técnicas já descritas (IA, *sandbox*, análise forense da memória), de forma a impedi-los de atingir os seus fins. Neste método, o programa, seja malicioso ou não, é executado numa *sandbox* que pode posteriormente ser restaurada ao seu estado original. Depois de executado o programa, será extraída a *memory dump* da máquina virtual, que é enviada para a ferramenta Volatility. Esta ferramenta vai extrair as características do *fileless malware* que serão posteriormente utilizadas para treinar modelos de inteligência artificial. A Figura 3 contém um esquema representativo deste método de deteção.

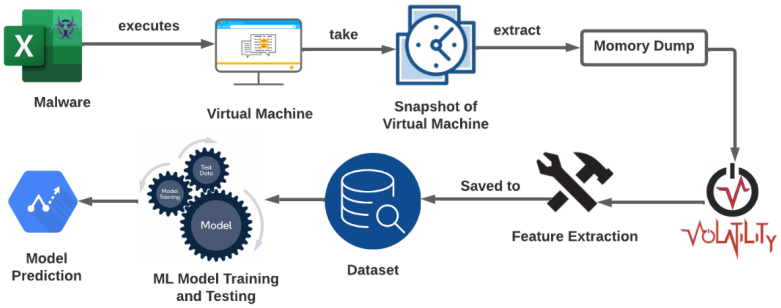

Fig. 3 Método de deteção de *fileless malware* proposto por Khalid et al. [3].

6 Conclusão

Este artigo apresentou uma revisão de várias técnicas de ataques *fileless* e os respetivos métodos de deteção. O foco incidiu maioritariamente em *malware* que tem como alvo as plataformas dos sistemas Windows passando pelas várias eras deste tipo de ataques informáticos: desde a altura em que estes residiam nos discos das máquinas até às atuais técnicas *fileless* de *Living-of-the-Land*.

As técnicas de ataque exploradas, como o *malware* residente na memória, *malware* do registo do Windows, ou ataques *Living-of-the-Land*, demonstram como os atacantes conseguem tirar proveito de funcionalidades legítimas, já existentes nos sistemas para executar código malicioso, evitando assim, as técnicas de deteção tradicionais.

De forma a mitigar esta ameaça, foram abordados vários métodos de deteção, desde a análise comportamental, que se baseia na identificação de padrões, passando pela análise forense da memória, que examina o conteúdo da memória volátil dos sistemas ou o uso de *sandboxing*, que permite aos investigadores especializados fazer uma análise do comportamento do *malware*. Contudo, cada uma destas técnicas tem as suas limitações tanto ao nível de precisão, como da sua escalabilidade, pelo que normalmente são utilizadas de forma complementar.

Assim, tendo em conta este cenário, é crucial que empresas e organizações invistam em soluções de prevenção e deteção de *malware*, para proteger não só os seus clientes, mas também toda a sua infraestrutura interna. Estas soluções devem recorrer

a abordagens leves e que se adaptem facilmente a novos vetores de ataque, podendo ser implementadas com *machine learning* e inteligência artificial. Quanto aos governos, devem, também, investir na segurança das suas infraestruturas informáticas de forma a proteger não só os dados dos seus cidadãos, mas também os seus mecanismos de defesa. Devem desenvolver a sua legislação para obrigar as empresas que operam nos respetivos países a investirem na segurança para dados dos consumidores. Espera-se, ainda, que incentivem e apoiem a investigação de novas técnicas mais avançadas para deteção e prevenção, bem como o reforço das técnicas mais promissoras, já existentes, de modo combater e prevenir estes ataques. Os programadores e pessoal especializado devem ser sensibilizados para a importância de boas práticas de segurança e desenvolvimento de *software* seguro e sem falhas.

Através desta revisão, pretendeu-se consciencializar e aumentar a notoriedade das ameaças dos *fileless malware* no domínio atual do cibercrime.

Referências

[1] Tahir, R.: A study on malware and malware detection techniques. International Journal of Education and Management Engineering **8**(2), 20 (2018)

[2] Kara, I.: Fileless malware threats: Recent advances, analysis approach through memory forensics and research challenges. Expert Systems with Applications **214**, 119133 (2023)

[3] Khalid, O., Ullah, S., Ahmad, T., Saeed, S., Alabbad, D.A., Aslam, M., Buriro, A., Ahmad, R.: An insight into the machine-learning-based fileless malware detection. Sensors **23**(2), 612 (2023)

[4] Afianian, A., Niksefat, S., Sadeghiyan, B., Baptiste, D.: Malware dynamic analysis evasion techniques: A survey. ACM Comput. Surv. **52**(6) (2019) https://doi.org/10.1145/3365001

[5] Sudhakar, Kumar, S.: An emerging threat fileless malware: a survey and research challenges. Cybersecurity **3**(1), 1 (2020)

[6] Liu, S., Peng, G., Zeng, H., Fu, J.: A survey on the evolution of fileless attacks and detection techniques. Computers & Security **137**, 103653 (2024) https://doi.org/10.1016/j.cose.2023.103653

[7] Khushali, V.: A review on fileless malware analysis techniques. International Journal of Engineering Research & Technology (IJERT) **9**(05) (2020)

[8] Microsoft Docs: PowerShell Scripting Overview. https://learn.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.5&viewFallbackFrom=powershell-7.3. Acedido: 30-04-2025

[9] Obeegadoo, P., Bekaroo, G.: Effective detection of fileless malware: A review and comparative analysis of detection techniques. In: 2024 International Conference

on Next Generation Computing Applications (NextComp), pp. 1–7 (2024). IEEE

[10] Microsoft Corporation: Fileless threats - Microsoft Defender for Endpoint. Acedido: 02-05-2025 (2024). https://learn.microsoft.com/en-us/defender-endpoint/malware/fileless-threats

[11] Baker, K.: Fileless Malware Explained. https://www.crowdstrike.com/en-us/cybersecurity-101/malware/fileless-malware/. Acedido: 02-05-2025 (2024)

[12] Microsoft Docs: Windows Registry for Advanced Users. https://learn.microsoft.com/en-us/troubleshoot/windows-server/performance/windows-registry-advanced-users. Acedido: 30-04-2025

[13] Afreen, A., Aslam, M., Ahmed, S.: Analysis of fileless malware and its evasive behavior. In: 2020 International Conference on Cyber Warfare and Security (ICCWS), pp. 1–8 (2020). https://doi.org/10.1109/ICCWS48432.2020.9292376

[14] Kavadias, N.: Silent intruders: Understanding living-off-the-land techniques, threats, countermeasures and emerging solutions (2024)

[15] Dopamu, O.: Updates on malware detection and analysis. IJSER **15**, 1 (2024)