Detecção Inteligente de Malwares em Sistemas Windows

Jessica C. Patricio[1], Carlos H. Paiva[1], Renan L. Rodrigues[1], Rafael L. Gomes[1]

[1]Universidade Estadual do Ceará (UECE), Fortaleza, Ceará, Brasil.

{jessica.cacau,henrique.paiva,renann.rodrigues}@aluno.uece.br

rafa.lopes@uece.br

1. Introdução

A ocorrência de malwares em sistemas operacionais Windows continua sendo uma das principais preocupações para as empresas e instituições, pois este é dominante em ambientes corporativos e domésticos. Os Malwares (tais como Spywares, Trojans e Ransomwares) responsáveis pelos ataques são frequentemente distribuídos online, direcionados a usuários, empresas e entidades governamentais [Costa et al. 2024].

Neste contexto, propõe-se uma solução baseada em Inteligência Artificial e análise de processos, que visa detectar malwares por meio da inspeção do conteúdo da memória principal de dispositivos com sistema operacional Windows. A arquitetura da solução foi concebida por ser portátil, escalável e com impacto mínimo nos dispositivos. O funcionamento baseia-se no monitoramento do dispositivo com a extração de dumps de memória RAM. O conteúdo extraído é compactado e enviado para um ambiente de análise remoto, onde são extraídas características dos processos. Esses dados são então submetidos a um modelo de IA previamente treinado, capaz de classificar como legítimo ou malicioso.

A Figura 1 apresenta uma visão geral da arquitetura proposta composta por dois componentes: o *device*, fazendo referência à sua execução no dispositivo a ser monitorado, responsável pela extração e envio de dados a partir do dispositivo monitorado, e o *service*, fazendo referência à sua função como serviço em borda dedicado à análise, responsável pelo armazenamento dos dados, análise remota e classificação.

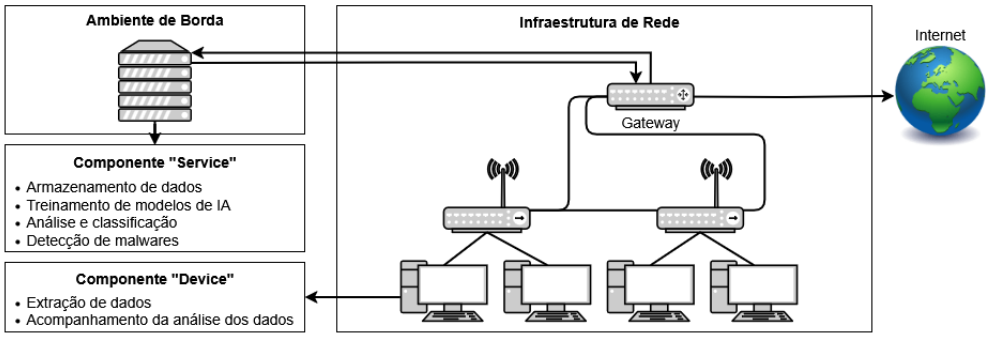

Figura 1. Visão Geral da Solução.

Durante sua operação, o dispositivo realiza a extração do dump de memória com o WinPmem[1], uma ferramenta de aquisição de memória, gerando um arquivo RAW. Esse arquivo é então compactado com o uso da biblioteca zipfile[2], visando à eficiência do envio, visto que o arquivo gerado pode ser muito grande e gerar uma sobrecarga na rede. Por sua vez, o *service* inicia-se com a descompressão do arquivo recebido, seguida da

[1]https://github.com/Velocidex/WinPmem/releases/tag/v4.0.rc1
[2]https://docs.python.org/3.10/library/zipfile.html

análise do conteúdo RAW com o VolMemLyzer[3], que extrai características dos processos da memória. Essas características são a entrada de um modelo de IA previamente treinado (com os dados da base [Carrier et al. 2022]) para detecção de malwares, ou seja, classifica o processo como malicioso ou não.

2. Experimentos

O experimento de eficiência de extração e análise de dumps de memória RAM foi realizado em máquinas virtuais (VMs) executando o sistema operacional Windows 10 com 2GB, 4GB e 8GB de memória RAM. Os dumps gerados pelas ferramentas demandaram, em média, tempos de processamento em torno de $6ms$, $23ms$ e $88ms$ para as VMs de 2GB, 4GB e 8GB, respectivamente, enquanto que a análise destes demandaram um tempo de $360ms$, $679ms$ e $890ms$ para as VMs de 2GB, 4GB e 8GB, respectivamente. Adicionalmente, foi avaliada a capacidade de compressão da solução, onde a duração da compressão demandou cerca de 131 segundos e a taxa de compressão em torno de 70%, viabilizando a eficiência e escalabilidade do sistema (devido a economia de largura de banda), mesmo com múltiplos dispositivos monitorados simultaneamente.

Com relação à capacidade de detecção de malwares, foram analisadas as técnicas KNN, CART, MLP e RF, visto que estas possuem abordagens de aprendizado distintas. Os resultados são apresentados na Tabela 1. A partir desses resultados, é possível notar que as técnicas KNN, CART e RF possuem a melhor taxa de detecção, mas KNN e RF possuem um maior tempo de detecção. Desta forma, o modelo CART se apresentou como a técnica mais viável para integrar a solução, visto que possui a maior taxa de detecção juntamente com o menor tempo de detecção. Com base nos resultados, foi possível avaliar a viabilidade da solução de detecção de Malwares.

Tabela 1. Desempenho de Detecção de Malwares e Processos Benígnos

Caso	Recall	Acurácia	AUPRC	Tempo de Detecção
KNN	0.99	0.99	0.99	171.88
CART	0.99	0.99	0.99	15.66
MLP	0.95	0.95	0.97	31.25
RF	0.99	0.99	0.99	78.09

Agradecimentos

Os autores agradecem ao CNPq (N^o 303877/2021-9 e N^o 405940/2022-0) e a CAPES (N^o 88887.954253/2024-00) pelo apoio financeiro.

Referências

Carrier, T., Victor, P., Tekeoglu, A., and Lashkari, A. H. (2022). Detecting obfuscated malware using memory feature engineering. In Mori, P., Lenzini, G., and Furnell, S., editors, *Proceedings of the 8th International Conference on Information Systems Security and Privacy, ICISSP 2022, Online Streaming, February 9-11, 2022*, pages 177–188. SCITEPRESS.

Costa, M. A., Costa, Y. M., Almeida, Y. O., Cardoso, F. J., and Gomes, R. L. (2024). Connection management using automated firewall based on threat intelligence. In *Proceedings of the 2024 Latin America Networking Conference*, LANC '24, page 32–37, New York, NY, USA. Association for Computing Machinery.

[3]https://github.com/ahlashkari/VolMemLyzer/releases/tag/V2.0.0