ISSN: 1983 7402 ITA, 26 a 28 SET 2023

Primeiros Passos na Identificação de *Packers* Utilizando Assinaturas Semânticas

Isabelle Cecília de Andrade[1], Rafael Oliveira da Rocha[2], Lourenço Pereira Jr.[2] e Idilio Drago[3]
[1]Núcleo do Centro de Defesa Cibernética da Aeronáutica, Brasília/DF - Brasil
[2]Instituto Tecnológico de Aeronáutica, São José dos Campos/SP - Brasil
[3]Politecnico di Torino, Turin - Itália

***Resumo*— *Malwares* continuam representando uma grande ameaça para a segurança cibernética de infraestruturas críticas e a utilização de técnicas como *packers* dificultam a sua detecção. As técnicas existentes para identificação de arquivos empacotados, como assinaturas tradicionais e análise de entropia, não são suficientemente eficazes no reconhecimento de variantes desconhecidas de *packers*. Para abordar esse problema, este estudo propõe uma nova abordagem, baseada em assinaturas semânticas, capaz de capturar a funcionalidade do *packer*. Por não utilizar informações da estrutura do arquivo empacotado, que é facilmente mutável, a abordagem proposta torna a detecção mais robusta às variações dos *packers*. Os experimentos realizados neste trabalho produziram resultados promissores, mostrando que a extração de assinaturas semânticas de arquivos empacotados não só é viável como eficaz para detectar *packers* desconhecidos.**

***Palavras-Chave*— *malwares*, *packers*, detecção**

I. Introdução

Notícias recentes têm evidenciado como softwares maliciosos vêm sendo empregado em ataques cibernéticos contra Forças Armadas de diversos países[1], [2]. Nesse contexto de guerra digital, a detecção desse tipo de artefato (*malwares*) pode ser considerada como um desafio contínuo, sobretudo na área de segurança nacional em que qualquer falha de detecção pode ter consequências desastrosas.

Em um contexto operacional cibernético no qual o ator adverso é capacitado e com alta motivação no êxito do ataque, é de se esperar que, cada vez mais, TTPs (técnicas, táticas e procedimentos) sejam aprimorados para contornar as medidas de proteção. Como consequência, tem sido observada uma crescente adoção de mecanismos de proteção e ofuscação de código, tais como *packers*, que tornam difícil a utilização dos métodos tradicionais de detecção de *malwares*[3].

Para lidar com essa realidade, uma boa prática das equipes de segurança é identificar arquivos empacotados (*packed*) e direcioná-los para análise antes que os usuários possam executá-los. Nesse sentido, o Centro de Tratamento de Incidentes de Rede da Forca Aérea Brasileira, CTIR.fab[1] monitora o espaço cibernético da Força empregando ferramentas líderes de mercado, que utilizam, essencialmente, assinaturas, entropia e técnicas dinâmicas para a identificação de *malwares* empacotados.

Embora as assinaturas sejam muito eficazes em identificar um arquivo empacotado com um *packer* já conhecido, elas possuem dificuldades em lidar com variantes de *packers*, por necessitarem de atualização constante do banco de assinaturas. Por outro lado, embora as técnicas dinâmicas apresentem sucesso na detecção de arquivos empacotados com *packers* novos, essa abordagem possui outras desvantagens, como a maior utilização de recursos computacionais e a suscetibilidade em serem contornadas por técnicas *anti-debugging*[4]. Além disso, a utilização de outros parâmetros estáticos na detecção de *malwares* empacotados, como a entropia, também vem caindo em desuso, visto que a correlação entre entropia e empacotamento é discutível em um cenário em que *malwares* de baixa entropia são cada vez mais presentes[5].

Como forma de estudar uma alternativa para detecção de arquivos empacotados que seja eficaz e robusta a variações do *packer*, este trabalho propõe a extração de assinaturas semânticas de arquivos empacotados. Esse tipo de assinatura, por capturar a funcionalidade do *packer* ao invés de de sua estrutura, tende a ser mais robusta do que as assinaturas tradicionais. Os resultados obtidos nos experimentos realizados são promissores, indicando que, de fato, é possível extrair uma assinatura semântica de arquivos empacotados genérica o suficiente para detectar arquivos empacotados com *packers* desconhecidos.

II. Abordagem proposta

Nesta seção, será apresentada a proposta de modelo de detecção utilizando assinaturas semânticas. Além disso, para facilitar o entendimento da proposta, alguns conceitos de arquivos empacotados (*packers*) e de aprendizado de representação serão abordados.

A. Packers

Um *packer* é um programa que protege o conteúdo original de binário executável por meio de determinadas técnicas como compressão ou criptografia e que descomprime/desencripta o arquivo durante a sua execução[6]. *Packers* são, legitimamente, utilizados para proteção de código, porém são também utilizados por agentes maliciosos para dificultar a detecção e análise de *malwares*[3].

O resultado da execução de um binário empacotado (*packed*) é o mesmo do binário original, porém o código do binário empacotado é ofuscado pelo *packer*, apresentando um padrão de funções *assembly* bastante diverso do conteúdo original.

Exitem quatro principais variedades de funcionamento de *packers*: compressores (*compressors*), criptografadores (*crypters*), protetores (*protectors*) e empacotadores (*bundlers*)[3]. Por questões de simplicidade e sem perda de generalidade, o escopo dos experimentos deste trabalho foi restrito aos *packers* da variedade compressores.

[1]https://www2.fab.mil.br/ctir/

B. ASM2VEC

Word2vec[7] e sua extensão Paragraph Vector[8] são técnicas publicadas nos anos de 2013 e 2014 por pesquisadores da empresa Google e que têm a capacidade de aprender relações entre palavras por meio de aprendizado não supervisionado sobre um corpo de textos suficientemente grande. O modelo de aprendizado produzido por essas técnicas é capaz de representar palavras (ou parágrafos, no caso do Paragraph Vector) por meio de vetores de tamanho fixo, que capturam informação semântica das palavras. Dessa forma, quanto mais parecidos forem os significados de duas palavras, mais próximos no espaço latente estarão os seus respectivos vetores gerados pelo modelo.

Baseado no funcionamento do Paragraph Vector, foi proposto o ASM2VEC[9] para gerar representações vetoriais de funções *assembly*. Assim como as técnicas que o inspiraram, o ASM2VEC não requer nenhum conhecimento prévio sobre o a linguagem e é capaz de aprender relações semânticas lexicais do código *assembly*. Em experimentos prévios realizados, a técnica foi capaz de identificar códigos com a mesma funcionalidade que possuíam estruturas completamente diferentes (como no caso de códigos ofuscados).

C. Conceito da solução

Uma vez que *packers* precisam desempacotar o conteúdo do arquivo em tempo de execução, é necessário que todo arquivo empacotado contenha uma função específica (ou um conjunto delas) responsável por realizar esse desempacotamento. A presença desse tipo de função em um binário pode ser entendida como a sua assinatura.

Entretanto, a utilização direta do código *assembly* da função desempacotadora seria semelhante às assinaturas tradicionais, sem capacidade de generalização. Isso ocorreria pelo fato de uma mesma função poder ser representada de diversas formas em código *assembly*, seja pela forma de funcionamento do compilador ou seja por modificações deliberadas no código fonte que modifiquem sua estrutura preservando a semântica. A Figura 1 ilustra um desses exemplos: um código simples em linguagem C e as estruturas diferentes dos códigos *assembly* gerados pelo mesmo compilador (gcc) mas com níveis de otimização diferentes (O0 e O3).

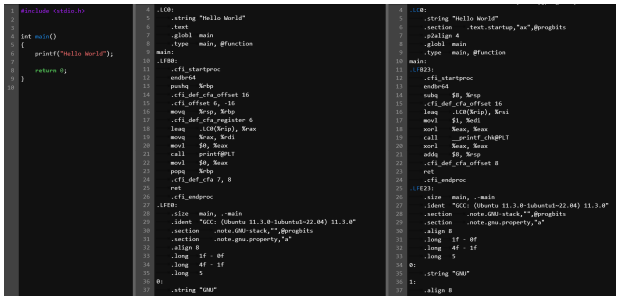

Fig. 1. Códigos *assembly* gerados para dois níveis diferentes de otimização.

Para contornar esse problema, pode-se utilizar o ASM2VEC para gerar representações vetoriais das funções. Como as representações geradas por esse algoritmo são semânticas, ainda que a estrutura do código seja modificada, a assinatura será capaz de identificar uma função desempacotadora em um binário.

D. Arquitetura da solução

A Figura 2 ilustra a arquitetura da solução proposta. Inicialmente é necessário uma fase de treinamento, utilizando um *dataset* com binários sabidamente empacotados (*packed*). A partir desses dados, os seguintes passos são executados:

1) **Extração do código *assembly* & cálculo dos vetores de representação** - é necessário que as funções de cada binário sejam identificadas e extraídas no formato de código *assembly*. Em seguida, as funções do *dataset* são fornecidas ao ASM2VEC para que ele produza os respectivos vetores de representação. Pela característica do ASM2VEC, vetores gerados que estejam próximos no espaço representam funções semanticamente parecidas. Vetores distantes, por outro lado, representam funções que não guardam semelhança sobre o seu funcionamento.
2) **Identificação de vetores comuns** - gerados os vetores de cada função dos binários, é necessário identificar quais funções estão presentes em todos os binários do *dataset*. Para isso, determina-se um valor de semelhança $\mathcal{S}$ e verifica-se, em cada binário, quais funções f_c têm a propriedade de $Semelhança(f_c, f_i) \geq S$ para, ao menos, uma função f_i em cada um dos outros binários do *dataset*.
3) **Assinatura** - por fim, analisando as funções comuns nos binários do *dataset*, é possível utilizar operadores lógicos AND e OR para criar uma assinatura $\mathcal{A}$ que represente o comportamento do *packer*. Ex: $\mathcal{A} = f_1 \wedge f_2 \vee f_3 AND f_4$, significa que, se existirem simultaneamente as funções f_1 e f_2 ou as funções f_3 e f_4, trata-se de um arquivo empacotado.

Uma vez criada a assinatura, é possível verificar se um binário está ou não empacotado conforme os passos a seguir:

1) **Extração do código *assembly* & cálculo dos vetores de representação** - é necessário extrair, do arquivo, as funções em formato *assembly*. Em seguida, é utilizado o ASM2VEC para gerar os vetores que representam cada uma de suas funções e também os vetores que representam as funções f_c constantes na assinatura $\mathcal{A}$.
2) **Compara com assinatura** - define-se um valor de semelhança $\mathcal{S}$ para verificar se as funções f_c da assinatura estão presentes ou não no binário. Por fim, ao se substituir as variáveis da assinatura $\mathcal{A}$ pela informação a respeito da presença ou ausência de cada função f_c no binário, a operação lógica é resolvida e o binário será classificado como empacotado caso o resultado seja 1 e classificado como não empacotado caso resulte em 0.

III. Implementação e Experimentos

A. Implementação da solução

1) Dataset:

a) Fonte de amostras: optou-se por não utilizar *malwares* nos experimentos, pois, nesse caso, haveria incertezas sobre se a amostra estaria ou não empacotada, prejudicando a confiabilidade do *groundtruth* do experimento. Em vez disso, optou-se por coletar aleatoriamente binários de *goodwares* do *site* PortableApps[2]. No total, foram baixados 182 binários

[2]https://portableapps.com/

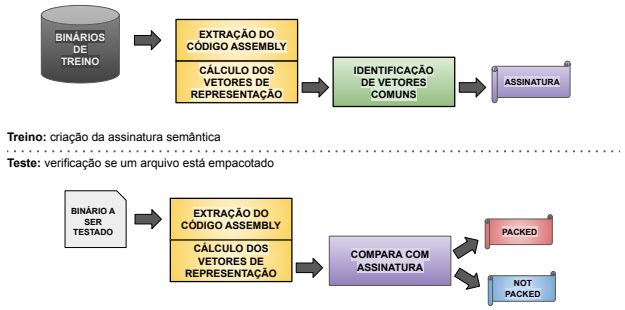

Fig. 2. Arquitetura proposta.

e a ferramenta Unipacker [3] foi utilizada para garantir que os binários não estivessem empacotados. Após essa etapa, consolidou-se um conjunto amostral de 130 binários não empacotados. Em seguida, essas amostras foram empacotadas com quatro *packers* frequentemente encontrados em *malwares*: UPX[4], ASPACK[5], MPRESS[6] e Petite[7]. Algumas amostras não puderam ser empacotadas em alguns casos. A Tabela I resume a quantidade de amostras do *dataset*.

TABELA I

DISTRIBUIÇÃO DAS AMOSTRAS DO *dataset*.

Categoria	Quantidade
Amostras benignas não empacotadas	130
Amostras empacotadas com UPX.	129
Amostras empacotadas com ASPACK.	121
Amostras empacotadas com MPRESS.	97
Amostras empacotadas com PETITE.	30

b) Separação do dataset: para preparar o *dataset* de treino, foram selecionados, de maneira aleatória, 60 dos 129 binários que puderam ser compactados pelo UPX. A essas 60 amostras empacotadas foram adicionados os respectivos 60 binários originais, de modo que esses 120 objeto pudessem ser utilizados para treinar o modelo ASM2VEC.

2) Cálculo dos vetores de representação & Assinatura: neste trabalho foi utilizada uma implementação do ASM2VEC disponível em github[8]. Além do próprio ASM2VEC, essa implementação disponibiliza um *script* baseado em RADARE2[9] que foi utilizado para extrair o código *assembly* dos binários. Para criação do modelo ASM2VEC, foram configurados os seguintes hiperparâmetros: número de épocas = 4.000; número de dimensões do vetor de *embedding* = 40;e taxa de aprendizado = 0,0002. Uma vez que o modelo foi criado, este foi utilizado para gerar as representações vetoriais das funções dos 60 binários UPX do *dataset* de treino.

3) Identificação de vetores comuns: após os vetores de representação terem sido gerados, buscou-se identificar quais funções estariam presentes em todos os binários empacotados do *dataset* de treino. Para isso, foi definida como métrica a similaridade por cosseno e o valor de 0,8 foi arbitrado como limite mínimo de similaridade para que duas funções fossem consideradas semelhantes entre si. Definidas tais premissas, foram identificas quatro funções presentes em todos os binários UPX do *dataset* de treino, chamadas de funções comuns.

Após a análise manual supracitada, adotou-se como assinatura $\mathcal{A}$ a verificação da existência simultânea das quatro funções comuns. Dessa forma, seja f_1, f_2, f_3 e f_4 variáveis indicando a presença (1) ou ausência (0) de cada uma das quatro funções comuns identificadas em todos os binários UPX do *dataset* de treino, a assinatura pode ser representada conforme a seguinte expressão lógica:

$$\mathcal{A} = (f_1 \wedge f_2 \wedge f_3 \wedge f_4)$$

B. Experimentos realizados

1) Eficácia com packer conhecido: para avaliar a eficácia da assinatura semântica em detectar binários empacotados com *packers* conhecidos, foi feito um teste com o conjunto total de amostras originais (130 amostras) e empacotadas com o UPX (129 amostras). Cabe notar que, desse total, 60 amostras empacotadas com UPX foram utilizadas na fase de treino para criar a assinatura semântica.

Foram feitos experimentos com diversos níveis de de similaridade: 40%, 45%, 50%, 55%, 60%, 65%, 70%, 75% e 80%. A tendência é a de que, quanto maior o nível de similaridade exigido, menos amostras (empacotadas ou não) tenderão a serem classificadas como empacotadas. Em contraste, quanto menor o nível de similaridade exigido, mais amostras (tanto empacotadas como não empacotada) passarão a ser classificadas como empacotadas.

A partir das classificações feitas pela assinatura, o valor de verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN) foram calculados e analisados. Para a assinatura ser considerada bem sucedida, deve haver algum intervalo de similaridade para o qual a assinatura não gere FP nem FN.

2) Eficácia com packers desconhecidos: O procedimento realizado no experimento com *packer* conhecido foi replicado em mais três outros conjuntos de amostras. Cada um desses conjuntos contempla, além das 130 amostras originais, todas as amostas empacotadas com o respectivo *packer*.

Dessa forma, a assinatura gerada a partir de binários empacotados com UPX foi avaliada em cada um dos três conjuntos descritos anteriormente, de modo a avaliar se a assinatura semântica tem potencial de ser generalizada para outros *packers* até então desconhecidos.

Como há um grande desbalanceamento de classes (empacotado *versus* não empacotado) em alguns dos conjunto de dados, optou-se por utilizar a métrica F1-Score para medir o desempenho de classificação da assinatura. Para considerar que a solução de assinaturas semânticas tem potencial de generalização, é necessário que, para algum nível de similaridade, o F1-Score obtido para algum dos *packers* desconhecidos seja alto (indicando baixa taxa de falsos positivos e falsos negativos).

IV. RESULTADOS E DISCUSSÕES

A. Achados paralelos

Após a extração do código *assembly* dos arquivos do *dataset* de treino, verificou-se uma propriedade interessante:

[3]https://github.com/unipacker/unipacker
[4]https://upx.github.io/
[5]https://http://www.aspack.com/
[6]https://mpress.apponic.com/
[7]https://www.un4seen.com/petite/
[8]https://github.com/oalieno/asm2vec-pytorch
[9]https://rada.re/n/

TABELA II

ESTATÍSTICA DA QUANTIDADE DE FUNÇÕES POR CLASSE

	ASPACK	MPRESS	PETITE	UPX	ORIGINAIS
Média	12,98	6,41	91,00	2,04	95,75
Mediana	13	4	91	2	99
Moda	13	3	91	2	99
Máximo	14	12	91	3	99
Mínimo	11	3	91	1	83

os arquivos empacotados, em geral, possuem um número de funções muito menor do que os originais. Além disso, os arquivos empacotados com o Petite possuem um padrão muito específico, contendo sempre 91 funções. A Tabela II sintetiza os números de funções em ambas as situações.

Caso as propriedades identificas se repitam em um *dataset* mais representativo (com maior quantidade e variedade de amostras), o número de funções em um arquivo pode vir a se tornar um indicador para arquivo empacotado. Por ser um assunto ortogonal ao objeto deste trabalho, não foram realizados maiores experimentos para validar tal hipótese. Entretanto, trabalhos futuros podem ser realizados no sentido de validar a propriedade descoberta.

B. Eficácia para packer conhecido

A Tabela III sintetiza os resultados obtidos para binários empacotados com UPX (o mesmo *packer* utilizado para criação da assinatura) para diversos valores de limite (*threshold*) de similaridade. São exibidos os valores de TP, TN, FP e FN.

TABELA III

RESULTADOS DE DETECÇÃO DE BINÁRIOS EMPACOTADOS COM UPX.

Threshold	TP	TN	FP	FN
40%	129	119	11	0
45%	129	130	0	0
50%	129	130	0	0
55%	129	130	0	0
60%	129	130	0	0
65%	129	130	0	0
70%	129	130	0	0
75%	120	130	0	9
80%	102	130	0	27

Conforme pode ser observado, para os valores de *threshold* de 45% a 70% a assinatura funcionou perfeitamente, detectando todos os binários empacotados e sem classificar erroneamente nenhum arquivo não empacotado. Utilizando o valor de 40% de similaridade, a assinatura passa a classificar binários não empacotados como se estivessem empacotados (em torno de 8% de erro). No outro extremo, ao utilizar 75% para similaridade, a assinatura deixa de identificar alguns binários empacotados (cerca de 7% dos binários deixaram de ser identificados).

Em síntese, os dados mostram que houve uma larga faixa de similaridade (de 45% a 70%) na qual a assinatura semântica é capaz de identificar arquivos empacotados perfeitamente, sem ocorrência de falsos positivos ou falsos negativos. Tais resultados evidenciam que é possível criar uma assinatura semântica que represente um *packer* conhecido.

Essa faixa extensa de possíveis *thresholds* que geram uma resultado perfeito para a assinatura semântica propiciam uma significativa discricionariedade operacional para a implementação da assinatura semântica. Por exemplo, caso a utilização operacional da assinatura demande que binários empacotados com ligeiras variações do *packer* sejam detectados, a melhor opção seria utilizar valores de *threshold* próximos a 45%. Por outro lado, caso o ambiente de monitoramento possua muitos binários não empacotados que possuam funções parecidas com as funções comuns da assinatura, a escolha de um *threshold* próximo a 70% poderia evitar a produção de falsos positivos.

A partir desses resultados obtidos, espera-se que estudos mais profundos sejam executados, no sentido de desenvolver assinaturas semânticas eficazes para os *packers* já conhecidos pelo CTIR.fab. Tal tipo de assinatura teria a vantagem em relação às já existentes pelo fato de, em tese, ser robusta a modificações na estrutura do binário que busquem ofuscar o *packer* utilizado.

C. Eficácia para diversos packers

A Figura 3 ilustra o resultado de F1-Score para cada um dos grupos de *packers*. Conforme pode ser observado, a assinatura gerada com o UPX apresentou bons resultados para as amostras empacotadas com ASPACK e MPRESS quando o *threshold* de similaridade é definido como 40% e 45%. Tal resultado indica uma potencial capacidade de generalização da regras para *packers* desconhecidos.

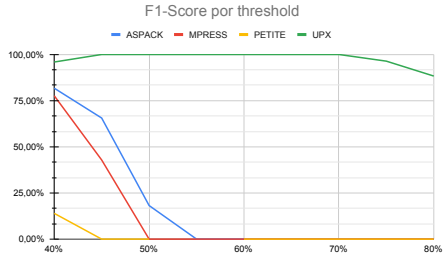

Fig. 3. F1-score para cada grupo de *packer*.

Para avaliar melhor o impacto do uso da assinatura na identificação de amostras empacotadas com *packers* desconhecidos, foi elaborada a Tabela IV com as matrizes de confusão dos resultados deste experimento para os *thresholds* de 40% e de 45%. Nessa matriz, os resultados dos quatro experimentos foram consolidados de forma que as classes UPX, ASPACK, MPRESS e PETITE se tornassem uma única classe ”*packed*”e os binários originais (não empacotados), a classe ”*not packed*”.

Cabe ressaltar que, para evitar o problema de *Out of Vocabulary Token*, o ASM2VEC atualiza o modelo antes de gerar o cálculo de vetores de representação. Por esse motivo, uma mesma função de um binário não empacotado acaba tendo diferentes representações vetoriais, a depender do contexto em que está sendo comparada. Dessa forma, nas matrizes de confusão geradas, são considerados que os binários não empacotados em cada um dos quatro experimentos (UPX, ASPACK, MPRESS e PETITE) são distintos, o que resulta em um total de 520 ”binários”não empacotados para a matriz de confusão.

Conforme pode ser observado, a precisão da assinatura é razoavelmente alta (87% para o *threshold* de 40% e 96% para o caso do *threshold* de 45%), o que representa uma baixa

TABELA IV

MATRIZES DE CONFUSÃO PARA OS *thresholds* 40% E 45%.

Threshold	Classe	Predito Packed	Predito Not Packed
40%	Real Packed	292	84
	Real Not Packed	44	476
45%	Real Packed	216	160
	Real Not Packed	8	512

taxa de falsos positivos. Por outro lado, a taxa de sensibilidade (*recall*) obtida foi menor: 78% para o *threshold* de 40% e 57% para o *threshold* de 45%, o que significa que a taxa de falsos negativos não foi tão baixa quanto a de falsos positivos.

Para ilustrar melhor o impacto dessas taxas, a Figura 4 ilustra o percentual de amostras empacotadas (real *packed*) e o de amostras não empacotadas (real *not packed*) que foram classificadas como *packed* pela assinatura utilizando os *thresholds* de 40% e de 45%. Conforme pode ser observado, para o *threshold* de 40%, quase 78% das amostras empacotadas (o que inclui os três *packers* não utilizados para criação da assinatura) foram identificadas com a contrapartida de pouco mais de 8% das amostras não empacotadas terem sido erroneamente classificadas como *packed*. Para o *threshold* de 45%, esses percentuais passam para 57% e 2%, respectivamente.

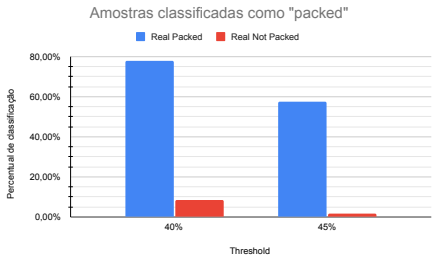

Fig. 4. F1-score para cada grupo de *packer*.

Os resultados obtidos, principalmente os dos *packers* ASPACK e MPRESS, evidenciam o potencial de generalização da regra semântica. A depender do ambiente onde a regra será empregada, o *threshold* pode ser ajustado para privilegiar a detecção de amostras empacotadas com *packers* desconhecidos a fim de reduzir o número de falsos negativos, a depender do cenário operacional.

V. CONCLUSÕES

Neste trabalho foi proposta uma nova abordagem para detecção de amostras empacotadas: regra semântica. Os resultados obtidos indicaram, primeiramente, que é possível a extração de assinaturas semânticas eficazes a partir de um binário empacotado. Essa assinatura, em tese, é mais robusta que as tradicionais, que levam em consideração a estrutura do binário empacotado em vez da funcionalidade do *packer*. Estudos futuros serão realizados no sentido de comparar diretamente os resultados de detecção da regra semântica com os dos métodos tradicionais (entropia e assinatura comum).

Cabe ressaltar que a assinatura gerada neste trabalho se deu de forma simplória: foi identificado manualmente quais funções estavam presentes em todas as amostras de treino considerando um *threshold* arbitrário (de 80%, no caso). Em consequência, podem ser pesquisadas formas mais inteligentes de se gerar a assinatura, com uso de estatística ou de aprendizado de máquina sobre o *dataset* de treino.

Outro resultado relevante do trabalho foi o de que as assinaturas semânticas possuem capacidade de generalização que permitem a detecção de amostras empacotadas com *packers* desconhecidos. Mais estudos serão realizados no sentido incluir outros tipos de *packers*, categorizá-los de acordo com sua técnica de compressão e realizar mais experimentos de detecção de *packers* desconhecidos.

Por fim, para que esse tipo solução seja adotada por um centro operacional como o CITR.FAB, que necessita que amostras suspeitas sejam analisadas em tempo real, serão ainda necessários experimentos para quantificar o *overhead* introduzido pelo uso de assinaturas semânticas e analisar seu tempo de resposta. Por esse motivo, serão conduzidos trabalhos não só no sentido de medir a eficiência, mas também serão feitos estudos no sentido de verificar possibilidades de otimização da solução.

REFERÊNCIAS

[1] S. Gatlan, "Rhysida ransomware leaks documents stolen from Chilean Army," Jun 2023, [Accessedo em 19-Jul-2023]. [Online]. Available: https://www.bleepingcomputer.com/news/security/rhysida-ransomware-leaks-documents-stolen-from-chilean-army/

[2] S. Lyngaas, "Unsolicited Smartwatches Bearing Malware Target U.S. Service Members: Army CID Raises Alarm — circleid.com," https://circleid.com/posts/20230627-unsolicited-smartwatches-bearing-malware-target-u.s-service-members-army-cid-raises-alarm, Jun 2023, [Accessado em 20-Jul-2023].

[3] T. Muralidharan, A. Cohen, N. Gerson, and N. Nissim, "File packing from the malware perspective: Techniques, analysis approaches, and directions for enhancements," vol. 55, no. 5, dec 2022. [Online]. Available: https://doi.org/10.1145/3530810

[4] Q. Sun, M. Abuhamad, E. Abdukhamidov, E. Chan-Tin, and T. Abuhmed, "Mlxpack: Investigating the effects of packers on ml-based malware detection systems using static and dynamic traits," in *Proceedings of the 1st Workshop on Cybersecurity and Social Sciences*, ser. CySSS '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 11–18. [Online]. Available: https://doi.org/10.1145/3494108.3522768

[5] A. Mantovani, S. Aonzo, X. Ugarte-Pedrero, A. Merlo, and D. Balzarotti, "Prevalence and impact of low-entropy packing schemes in the malware ecosystem," in *NDSS 2020, Network and Distributed System Security Symposium, 23-26 February 2020, San Diego, CA, USA*. Internet Society, 2020.

[6] A. Abusitta, M. Q. Li, and B. C. Fung, "Malware classification and composition analysis: A survey of recent developments," *Journal of Information Security and Applications*, vol. 59, p. 102828, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212621000648

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[8] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, p. II–1188–II–1196.

[9] S. H. Ding, B. C. Fung, and P. Charland, "Asm2vec: Boosting static representation robustness for binary clone search against code obfuscation and compiler optimization," vol. 2019-May. Institute of Electrical and Electronics Engineers Inc., 5 2019, pp. 472–489.