Um Estudo Acerca da Seleção de *Features* para a Detecção dos *Ransomwares* WannaCry, Ryuk e CryptoLocker

Gabriel O. Souza[1], **Silvio E. Quincozes**[1], **Juliano F. Kazienko**[2]
Vagner E. Quincozes[3] **e Nícolas Naves R. Faria**[1]

[1] FACOM – Universidade Federal de Uberlândia (UFU), Monte Carmelo, Brasil

[2]CTISM – Universidade Federal de Santa Maria (UFSM), Santa Maria, Brasil

[3]Universidade Federal do Pampa (UNIPAMPA), Alegrete, Brasil

{gabrieloliveirasouza620, sequincozes, nicolasnaves}@ufu.br,
kazienko@redes.ufsm.br, vagnerquincozes.aluno@unipampa.edu.br

Abstract. *Ransomware is a malicious software that encrypts files and demands a ransom payment. Due to their late detection, they pose a significant threat to individuals and organizations, resulting in substantial financial losses and data breaches. This paper identifies relevant features for the early detection of three well-known types of ransomware: WannaCry, Ryuk, and CryptoLocker. We use the Cuckoo environment to carry out experiments in which each ransomware activity is logged. Afterward, we create and make available specialized datasets for each studied ransomware attack containing normal and attack samples. Finally, we apply feature ranking methods to identify the 20 most representative features for detecting each ransomware family from our datasets, providing a valuable contribution towards improving ransomware detection mechanisms.*

Resumo. *Ransomware é um malware que sequestra dados e devido a sua detecção frequentemente tardia apresentam uma ameaça significativa para indivíduos e organizações, causando perdas financeiras substanciais e violações de dados. O objetivo deste trabalho consiste em identificar as features relevantes para detecção de três tipos conhecidos de ransomwares: WannaCry, Ryuk e CryptoLocker. Para tanto, é utilizado o ambiente Cuckoo Sandbox no qual a atividade de tais ransomwares são registradas. Com isso, são criados datasets contendo amostras normais e de ataque para cada tipo de ransomware estudado. Por fim, aplicou-se métodos de ranqueamento de features nesses datasets para a identificação das 20 features mais relevantes para a detecção dos ransomwares estudados, fornecendo uma contribuição valiosa para a melhoria dos mecanismos de detecção de ransomware.*

1. Introdução

A pandemia de COVID-19 impactou severamente a forma de trabalho das pessoas. Pesquisas realizadas em mais de 200 empresas mostram que 81% delas adotaram o modelo de trabalho remoto [Curran 2020]. Com isso, a ocorrência de ataques por meio de *softwares* maliciosos (*malwares*) aumentou mais de 75% no primeiro semestre de 2022 [Micro 2022][Razaulla et al. 2023]. Dentre os *malwares* mais devastadores, estão os *ransomwares*. No cenário atual, esse tipo de *malware* surge como uma das principais

ameaças, com um impacto devastador [Abbasi et al. 2022][Brewer 2016]. Esses *malwares* sequestram e criptografam dados valiosos, exigindo o pagamento de um resgate para sua liberação [Urooj et al. 2022]. Alguns tipos de *ransomwares* têm ganhado destaque pela sua popularidade, grau de ocorrências e pelos prejuízos principalmente financeiros causados, como o *WannaCry*, o *Ryuk* e o *CryptoLocker* [Beaman et al. 2021].

Apesar de ser a comum incidência de *ransomwares* em ambientes corporativos atualmente, a maioria das empresas atacadas ainda está sujeita a perder seus dados ou ser compelida a efetuar o pagamento do resgate. Até mesmo a perda do controle ou inativação de dispositivos podem ter efeitos danosos especialmente em sistemas de controle industriais, do inglês, *Industrial Control Systems* (ICSs) e dispositivos típicos desses sistemas, como *Human Machine Interfaces* (HMIs) e *Programmable Logic Controllers* (PLCs) [Micro 2020]. Um relatório da empresa russa de cibersegurança *Kaspersky* revela que cerca de 88% das empresas que já sofreram um ataque de *ransomware* admitem que pagariam o resgate caso voltassem a ser atacadas [Kaspersky 2021]. Isso se deve principalmente à falta de soluções adequadas para mitigar *ransomwares*. Existem esforços na literatura para detectar *ransomwares* através do processamento de características (*features*) que representam o padrão operacional desses *malwares*. Contudo, não há um consenso sobre quais características são mais eficazes na detecção de *ransomwares*.

Na literatura, existem autores que concentram esforços em estudar o comportamento de *ransomwares* como o *WannaCry* [Chen and Bridges 2017][Chen et al. 2019], Ryuk [Masid et al. 2022] e CryptoLocker [Abbasi et al. 2020][Abbasi et al. 2022]. Tais trabalhos empregam ferramentas como o *Cuckoo* [Oktavianto and Muhardianto 2013] para a extração de *features* e, em alguns casos, algoritmos de aprendizados de máquina na intenção de selecionar *features* representativas. Contudo, tais trabalhos apresentam como principais limitações a modelagem de comportamentos de usuários legítimos que é usada como contraste às operações realizadas por *ransomwares*. Em particular, tal modelagem contempla pouca diversidade de operações e é potencialmente tendenciosa já que tais operações (*e.g.,* compra de passagens online) não são similares àquelas executadas por um *ransomware*, o que pode resultar em amostras normais não representativas impactando negativamente na representatividade das *features* selecionadas.

Este estudo visa identificar características importantes para detectar três tipos de ransomware: *WannaCry*, *Ryuk* e *CryptoLocker*. A partir da ferramenta *Cuckoo Sandbox*, foram capturados os comportamento de arquivos normais e dos ransomwares estudados, gerando três conjuntos de dados correspondentes. Os resultados obtidos através do algoritmo de seleção de *features Releaf* indicam quais são as 20 *features* mais relevantes para detecção dos três tipos de *ransomwares* estudados. O estudo dos *ransomwares* em questão, é fundamental para a compreensão de ameaças cibernéticas mais presentes no cenário atual. A motivação para estudar esses *malwares* é a necessidade de entender como ambos funcionam, como se propagam e como podem ser detectados e prevenidos de maneira eficiente. Cada *ransomware* tem suas próprias características e técnicas de ataque [IBM 2022].

2. Fundamentação Teórica

Esta seção introduz o conceito de *ransomware* e suas categorias, bem como seus tipos estudados neste trabalho. Adicionalmente, a seleção de *features* e sua importância no

processo de detecção desse *malware* são discutidas. Ao estudar os *ransomwares* em questão, é possível compreender as técnicas utilizadas pelos atacantes, como a exploração de vulnerabilidades de sistemas, o uso de engenharia social e o emprego de técnicas de criptografia. Além disso, é possível conhecer as APIs do *Windows* e outras tecnologias utilizadas pelos *ransomwares*, que serão mencionadas posteriormente.

2.1. Ransomware

Ransomware é um tipo de *malware* que é capaz de impedir o acesso a dados pessoais, a menos que um resgate seja pago. Normalmente, o resgate dos dados é feito através de criptomoedas, onde ao mesmo tempo, gera um grande impasse referente ao rastreamento da transação solicitada pelos cibercriminosos [Razaulla et al. 2023].

Os *ransomwares* podem ser divididos em três categorias, sendo elas: *Scareware*, que usa anúncios a fim de manipular os usuários a baixar um determinado tipo de software que contém trechos de códigos maliciosos; o *Locker*, que visa bloquear qualquer funcionalidade do computador (e.g., tela, teclado) não permitindo que o usuário navegue ou utilize a máquina; e o *Crypto*, que cifra os arquivos do disco do usuário, porém não interfere nas funções básicas do computador. De forma geral, as três categorias utilizam e-mails de *phishing* como principal forma de disseminação, visando facilitar a contaminação de máquinas-alvo e se espalhando de forma gradativa na rede [Beaman et al. 2021].

Neste trabalho, são analisadas três variações de *ransomwares*: *WannaCry*, *CryptoLocker* e *Ryuk*. O primeiro tipo trata-se do *WannaCry*, que é um dos mais populares e tradicionais *ransomwares* da categoria *crypto*, onde seu principal método de propagação é através de *worms* [Beaman et al. 2021]. *Worms* consistem em uma categoria de *malware* capaz de se copiar de máquina para máquina, geralmente explorando algum tipo de falha de segurança em um software ou sistema operacional, não exigindo interação do usuário para operar. Outro tipo, o *Ryuk*, também encontra-se na categoria *crypto*. Seu principal método de propagação é através de e-mails de *phishing* [Beaman et al. 2021]. Os e-mails que são enviados ao usuário geralmente têm sua origem forjada. O ataque do *ransomware* se inicia, por exemplo, quando o alvo realiza o *download* de arquivo anexo em sua máquina e o executa. Segundo relatório da empresa Trend Micro [Micro 2020], o *Ryuk* respondeu pela maior parte dos ataques a ICSs no ano de 2020. Por último, o *ransomware CryptoLocker* cifra os arquivos do usuário e exige um resgate em troca da chave de decifração. Ele pertence à categoria *crypto*. Normalmente, a sua disseminação se dá através de e-mails de *phishing*, os quais contém anexos maliciosos ou *links* para *websites* forjados. Esse *ransomware* é apontado como o sétimo *malware* com maior ocorrência pelo *Center for Internet Security* em 2022 [CIS 2022].

2.2. Seleção de Features

A seleção de *features* consiste na escolha das características, do inglês, *features*, que melhor descrevem uma situação em particular, como um ciberataque. Particularmente, as características são destinadas à identificação de ataques provocados pelos *ransomwares* estudados neste trabalho. A escolha das *features* impactam significativamente na taxa de acurácia, entre outras métricas, em um processo de detecção. Por isso, torna-se importante um esforço seleção minucioso [Quincozes et al. 2018].

O Relief é um algoritmo de que usa um método estatístico para identificar aquelas *features* mais relevantes em conjuntos de dados. Ele é eficiente em termos de tempo

de execução e tende a selecionar um subconjunto pequeno de características estatisticamente relevantes. Para isso, o algoritmo usa uma abordagem baseada em amostras, selecionando aleatoriamente uma instância e, em seguida, encontrando a instância mais próxima que pertence a uma classe diferente. Em seguida, atualiza os pesos de todos os atributos com base nas diferenças entre essas duas instâncias. A comparação com outros algoritmos de seleção de atributos mostra as vantagens do Relief em termos de tempo de aprendizado e precisão do conceito aprendido, o que sugere a praticidade do Relief [Kira and Rendell 1992]. A seguir, no Algoritmo 1, o pseudocódigo do algoritmo Relief (baseado em [Megchelenbrink 2010]) é mostrado.

Algoritmo 1 Seleção de Features baseada no Algoritmo Relief

Entrada: Dataset D, número de features nf
Saída: Conjunto de feature selecionadas C
$C \leftarrow \emptyset$ ▷ Inicializa com valor vazio o conjunto de features selecionadas
$P \leftarrow \emptyset$ ▷ Inicializa o vetor de pesos P
for $f \leftarrow 1$ to nf **do**
    $P_f \leftarrow 0$ ▷ Inicializa o peso da feature f
end for
for $i \leftarrow 1$ to $|D|$ **do**
    $x_i \leftarrow random(D)$ ▷ Seleciona aleatoriamente (x) uma instância i de D
    $amp \leftarrow$ acertoMaisProximo(x_i) ▷ Instância com mesma classe mais próxima
    for $f \leftarrow 1$ to nf **do**
        $P_f \leftarrow P_f - \frac{1}{|D|}(|x_{i,f} - x_{amp,f}|)$ ▷ Atualiza o peso da f baseada no amp
        $emp \leftarrow$ erroMaisProximo(x_i) ▷ Instância com classe diferente mais próxima
        $P_f \leftarrow P_f + \frac{1}{|D|}(|x_{i,f} - x_{emp,f}|)$ ▷ Atualiza o peso de f baseada no emp
    end for
end for
for $f \leftarrow 1$ to amp **do**
    $i \leftarrow \arg\max_{i \in 1,\ldots,nf \setminus C} P_i$ ▷ Seleciona a feature com maior peso
    $C \leftarrow C \cup f$ ▷ Adiciona a feature selecionada ao conjunto
end for
return C

3. Trabalhos Relacionados

Na literatura existem diversos trabalhos relacionados ao estudo de *ransomwares*, porém ainda existem lacunas no que tange à análise de *features* para a sua detecção.

Em [Razaulla et al. 2023], é apresentada uma visão abrangente do estado da arte no contexto de *malwares* do tipo *ransomware*, incluindo sua evolução, taxonomia, detecção, vetores de infecção. Com isso, esse trabalho fornece uma compreensão atualizada do campo de segurança cibernética do *ransomware* e estimula o combate a essa ameaça em constante evolução. No entanto, não são abordados aspectos relacionados às *features* que podem ser relevantes para a detecção desse *malware*.

Outro estudo do tipo *survey* é conduzido por [Urooj et al. 2022], o qual visa contribuir para o entendimento abrangente da evolução e classificação do *ransomware*, bem como destacar as principais pesquisas realizadas atualmente nessa área. Por meio de uma análise detalhada de mais de 150 referências, observou-se que a maioria dos estudos (72,8%) concentrou-se na detecção do *ransomware*, com uma proporção significativa (70%) aplicando técnicas de Aprendizado de Máquina. No entando, observa-se uma lacuna em relação à falta de estudos sobre *ransomwares* mais recentes.

Em [Chen and Bridges 2017] é apresentada uma análise comportamental automatizada do *ransomware WannaCry*. A metodologia adotada pelos autores consiste na geração de *features* baseada na ferramenta *Cuckoo* [Oktavianto and Muhardianto 2013] e seleção de *features* com o algoritmo de aprendizado de máquina *Term Frequency–Inverse Document Frequency* (TF-IDF). Em particular, foram usados binários executáveis do *WannaCry* e desenvolvidos *scripts* na linguagem de programação Python para a simulação do comportamento normal. As principais limitações deste trabalho são decorrentes da modelagem do comportamento legítimo, o qual contempla pouca diversidade de operações e é potencialmente tendencioso já que tais operações (*e.g.,* compra de passagens online) não são similares àquelas executadas por um *ransomware*.

Em [Chen et al. 2019] os autores apresentam uma ferramenta automatizada para a extração de padrões e detecção precoce de *ransomwares*, incluindo o *WannaCry*. Para a etapa de geração de *features*, os autores executam a ferramenta *Cuckoo* [Oktavianto and Muhardianto 2013] seguindo a mesma metodologia que foi usada em [Chen and Bridges 2017]. Em seguida, três algoritmos de aprendizado de máquina foram empregados para a discriminação das *features* mais relevantes: TF-IDF, *Fisher's Linear Discriminant Analysis* (LDA) e *Extremely Randomized Trees* (ET). Contudo, embora a principal contribuição seja a seleção de *features*, a geração de padrões que representam comportamento normal estão limitadas a operações simples e previsíveis, tal qual em [Chen and Bridges 2017].

Em [Gera et al. 2021], os autores propõem uma abordagem híbrida para identificar e mitigar *ransomware* para plataforma Android. É empregado um novo algoritmo de seleção de características para extrair o conjunto de características dominantes. Os resultados experimentais mostram que o modelo proposto pode identificar *ransomware* com maior precisão. No entanto, apesar do estudo envolver em plataforma móvel amplamente utilizada, os autores não avaliam os tipos de *ransomwares* estudados neste trabalho.

Nos trabalhos [Abbasi et al. 2020] e [Abbasi et al. 2022], os autores apresentam um método de seleção de *features* baseado em *wrapper* para detectar e classificar *ransomwares* — incluindo o CryptoLocker — com base em seu comportamento. É utilizada uma estratégia orientada a grupos e *Particle Swarm Optimization* (PSO) para selecionar um número adequado de *features* de cada grupo, com base em sua relevância. No entanto, os autores não indicam quais foram os binários benignos utilizados na pesquisa. Com isso, não é possível saber o grau de similaridade das ações executadas pelo binário legítimo em relação ao binário maligno podendo.

Uma metodologia para a análise de *malwares* em geral é proposta em [Masid et al. 2022]. Como prova de conceito os autores aplicam tal metodologia na forma de um estudo de caso sobre o *ransomware Ryuk*. Para tanto, informações além de seu comportamento são capturados através de uma análise híbrida. Contudo, em contraste com a proposta deste trabalho, os autores não apresentam uma indicação de quais são as *features* mais relevantes para a detecção do *Ryuk*.

Dessa forma, embora a literatura atual contemple os trabalhos discutidos nesta seção, ainda existem lacunas que precisam ser abordadas. Por exemplo, uma das principais limitações da literatura consiste na modelagem de comportamentos de usuários legítimos que é usada como contraste às operações realizadas por *ransomwares*. Com

isso, há pouca diversidade de operações legítimas sendo estudadas em paralelo ao comportamento dos *ransomwares*, o que pode comprometer uma eficiente identificação das informações mais pertinentes para a detecção desse tipo de *malware*.

4. Materiais e Métodos

Nesta seção serão apresentados os materiais e métodos adotados neste trabalho. Na Seção 4.1, o cenário estudado é apresentado. Em seguida, na Seção 4.2, são apresentadas as etapas do estudo baseado no cenário apresentado anteriormente.

4.1. Cenário

De modo a reproduzir um ambiente com operações legítimas e também maliciosas, foi configurado um ambiente virtualizado por meio da ferramenta *Cuckoo Sandbox* [Oktavianto and Muhardianto 2013]. Tal ferramenta consiste em um *software* gratuito de código aberto que permite a análise automatizada de binários (*malwares*). Optou-se por tal ferramenta pela (i) oferta de uma variedade de opções para análise de *malwares*, superando *sandboxes* como o Drakvuf, e (ii) a possibilidade de ser aprimorado com ferramentas adicionais, como YARA e Suricata [Ilić et al. 2022]. A partir do ambiente configurado no Cuckoo Sandbox, foram executados 35 binários de aplicações legítimas (*e.g.,* Ruffus, Chrome, Firefox, Putty, Tetris, etc.) e 3 *ransomwares*, quais sejam: *WannaCry*, *Ryuk* e *CryptoLocker*. Os *ransomwares* foram executados três vezes no *Cuckoo Sandbox*. Cada binário executado teve seu tempo de execução definido como dez minutos.

Visando facilitar à reprodutibilidade dos resultados apresentados na Seção 5, os binários benignos e malignos, *datasets* e *scripts* utilizados nos experimentos estão disponíveis em repositório criado na plataforma *GitHub*[1]. A Figura 1 ilustra o cenário de experimentação, que envolve o *upload* de binários (*binary.exe*) na ferramenta e a criação de máquinas virtuais por meio do software VirtualBox. A ferramenta *Cuckoo* possui um agente, criado em Python, que monitora e captura todo o conteúdo e mudanças realizadas pelo binário durante sua execução. A ferramenta em questão é instalada em um computador Dell Inspiron 15 3000 com sistema operacional Ubuntu 18.04.

4.2. Metodologia

Uma vez construído o ambiente virtualizado para a execução de binários, foram adotadas análises desses binários a partir de abordagens estática e dinâmica. A principal diferença dessas abordagens é que a análise estática se baseia apenas na análise referente ao código-fonte dos binários e não requer a execução do mesmo, em contraste com análise dinâmica que analisa o fluxo de informações e chamadas de funções de um binário em execução. Quando ambas as técnicas são usadas em conjunto, esse processo é chamado de análise híbrida [Damodaran et al. 2017][Uppal et al. 2014].

Os passos ilustrados no cenário da Figura 1 demonstram a metodologia empregada na execução dos experimentos realizados. No Passo 1, o usuário insere o binário a ser analisado, definindo as especificações de tempo de execução e permissão de acesso à internet para outras ações durante a execução do mesmo. Em seguida, no Passo 2, a aplicação utiliza a *Virtual Network* para enviar e inicializar a execução do binário na máquina virtual configurada para o *Cuckoo Sandbox* realizar sua análise. Durante a execução do binário,

[1]Disponível em: `https://github.com/gabrielolivs/Deteccao-de-Ransomware`

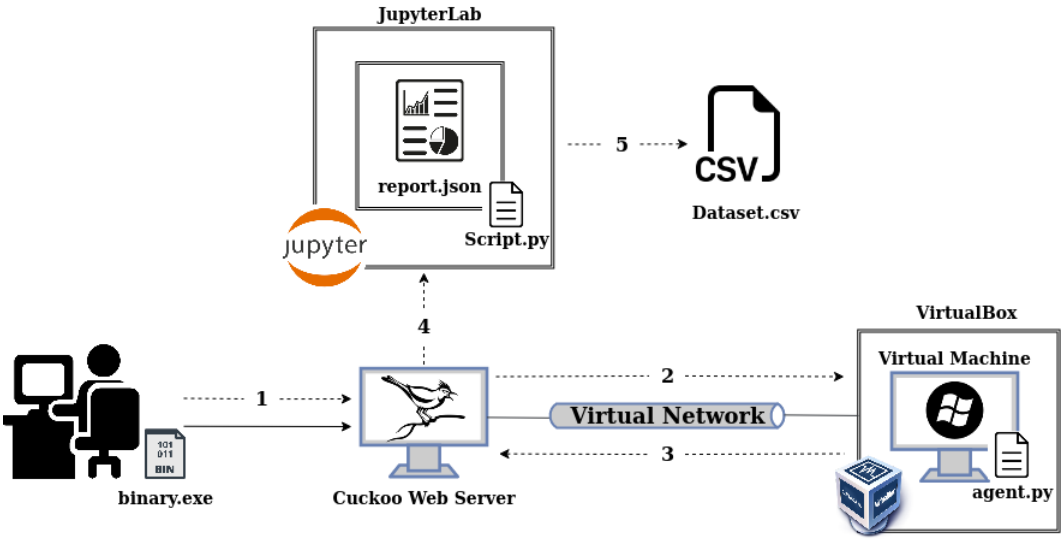

Figura 1. Cuckoo Sandbox - Fluxo

um programa chamado `agent.py` é executado dentro da máquina virtual, capturando todas as ações do binário.

Após a conclusão da execução do binário, no Passo 3, o agente `agent.py` captura todas as informações relevantes que ocorreram durante a execução e as envia para a aplicação por meio da *Virtual Network*. O *Cuckoo Sandbox* coleta todas essas informações e as armazena em um arquivo de formato *JavaScript Object Notation* (JSON), o qual é enviado no Passo 4 para a plataforma JupyterLab. Nessa plataforma, foi implementado um código que realiza a coleta e filtragem das informações relevantes para o ranqueamento das *features*, como descrito neste artigo. A fim de processar nesse componente o relatório resultante da análise do *Cuckoo*, foi implementado um *script* na linguagem *Python* chamado `script.py` o qual é disponibilizado no repositório recém mencionado. Ele é responsável pela criação de um arquivo CSV que contém todas as features filtradas, representado no Passo 5 na imagem.

A partir do arquivo resultante do processo ilustrado na Figura 1, foram coletadas 226 *features* para o *WannaCry*, 211 para o *Ryuk* e também 211 para o *CryptoLocker*. Tais *features* representam informações como chamadas de sistemas e operações realizadas por cada binário executado. Dessa forma, foi possível a construção de 3 *datasets* com amostras maliciosas (*WannaCry, Ryuk, CryptoLocker*) e legítimas (aplicações típicas de usuários comuns). É importante destacar que cada *dataset* gerado contém amostras relativas aos 35 binários legítimos e 1 binário malicioso. Portanto, o que muda de um *dataset* para outro são as amostras do binário malicioso. Ou seja, cada *dataset* possui amostras de somente um dos binários maliciosos, além de amostras relativas ao 35 binários legítimos, que são as mesmas para cada base de dados.

Por fim, utilizou-se o algoritmo Relief[2] para medir a representatividade de cada uma das *features* geradas, descartando assim as menos representativas. Tal algoritmo é detalhado na Seção 2.2.

[2]Disponível na biblioteca WEKA: `https://www.cs.waikato.ac.nz/ml/weka/`

5. Resultados e Discussão

Nesta seção são apresentados os resultados obtidos para cada *ransomware* estudado, bem como as devidas discussões.

5.1. Estudo do WannaCry

A partir das 226 *features* geradas, apenas 20 possuem um peso superior a 0,6. Portanto, como resultados preliminares, são apresentadas e discutidas tais *features* que compõem o *Top 20*. A Figura 2 ilustra o peso computado pelo algoritmo Relief para cada uma das *features* entre aquelas 20 mais bem avaliadas.

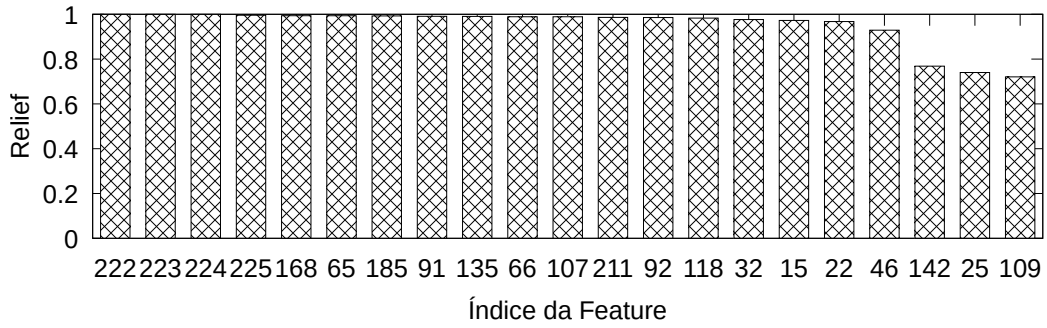

Figura 2. Top 20 features selecionadas para o WannaCry.

As *features* classificadas como as 20 mais representativas pelo método de seleção *relief* são elendadas a seguir: *CryptDecrypt* (F222), *CryptGenKey* (F223), *CryptExportKey* (F224), *CryptEncrypt* (F225), *SetFileAttributesW* (F168), *FindFirstFileExW* (F65), *MoveFileWithProgressW* (F185), *SetFileTime* (F91), *GetFileSizeEx* (F135), *NtQueryDirectoryFile* (F66), *CopyFileW* (F107), *CopyFileA* (F211), *DeleteFileW* (F92), *NtQueryInformationFile* (F118), *NtCreateFile* (F32), *CryptAcquireContextA* (F15), *GetFileAttributesW* (F22), *NtDuplicateObject* (F46), *RemoveDirectoryA* (F142), *NtEnumerateValueKey* (F25), *SHGetFolderPathW* (F109). A definição das mesmas pode ser encontrada no site Microsoft Learn[3].

A maioria das *features* representativas para a identificação da execução do *ransomware WannaCry* consistem em chamadas de funções da *Windows* API (WinAPI). Tal API consiste em um conjunto base de interfaces de programação para o sistema operacional *Microsoft Windows*.

A função *CryptDecrypt* (F222) foi identificada como a mais representativa. A API *CryptDecrypt* é uma função de criptografia da biblioteca do *Windows* utilizada para decifrar dados cifrados. Embora possa ser usada para garantir a segurança dos dados, a API *CryptDecrypt* também pode ser usada por *malwares* para decifrar dados cifrados que foram protegidos por um software de segurança ou por um usuário comum. Isso pode permitir que o *malware* acesse informações confidenciais, como senhas, números de cartões de crédito e outras informações pessoais.

Em seguida, as *CryptGenKey* (F223), *CryptExportKey* (F224), *CryptEncrypt* (F225) e *SetFileAttributesW* (F168) aparecem nas posições de 2 a 5 do *Top 20*. As funções

[3]Disponível em: `https://learn.microsoft.com/en-us/windows/win32/api/`

CryptDecrypt, *CryptGenKey*, *CryptExportKey* e *CryptEncrypt*, embora essas não sejam funções necessariamente maliciosas, elas são conhecidas por serem usada por *malwares*[4]. As mesmas são funções amplamente utilizadas em aplicativos que exigem segurança e criptografia de dados. No entanto, essas funções também podem ser usadas por *malwares* para realizar atividades maliciosas, como cifrar dados de vítimas ou criar novos *threads* para realizar operações maliciosas em segundo plano.

5.2. Estudo do Ryuk

O *ransomware Ryuk* invoca a chamada de várias API's para realizar suas atividades maliciosas. Algumas delas são usadas para cifrar arquivos do sistema, enquanto outras podem ser usadas para se comunicar com servidores de comando, que, por sua vez, realizam o ataque cifrando os dados do alvo. A partir da Figura 3, é possível visualizar o ranqueamento das *features* utilizadas pelo *malware*. A seguir as *features* selecionadas são listadas e as mais relevantes são discutidas.

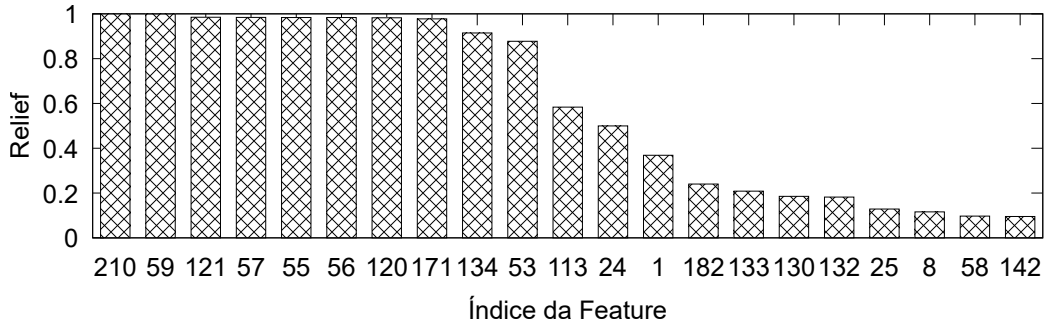

Figura 3. Top 20 features selecionadas para o Ryuk.

A partir das 211 *features* existentes no *dataset*, o algoritmo de seleção Relief rankeou as 20 mais representativas. São elas: *EnumServicesStatusW* (F210), *LookupAccountSidW* (F59), *NtOpenKeyEx* (F121), *Process32NextW* (F57), *CreateToolhelp32Snapshot* (F55), *Process32FirstW* (F56), *NtQueryKey* (F120), *ShellExecuteExW* (F171), *GetVolumePathNamesForVolumeNameW* (F134), *GlobalMemoryStatusEx* (F53), *NtTerminateProcess*(F113), *NtQueryValueKey* (F24), *score* (F1), *LookupPrivilegeValueW* (F182), *RegEnumKeyW* (F133), *OleInitialize* (F130), *GetVolumeNameForVolumeMountPointW* (F132), *NtClose* (F25), *SetUnhandledExceptionFilter* (F8), *NtOpenProcess* (F58), *CreateProcessInternalW* (F142).

A seguir são apresentadas as definições das principais *features* identificadas: A *feature EnumServicesStatusW* (F210) enumera serviços em execução no *Windows*, obtendo informações que incluem o seu nome, status e tipo. O *Ryuk* utiliza a *EnumServicesStatusW* (F210) para verificar se certos serviços de segurança, como os de soluções antivírus populares, estão em execução na máquina alvo, evitando detecção. Também pode utilizar o *Windows Defender* como solução padrão. A *feature LookupAccountSidW* (F59) é usada para obter o nome da conta associada a um determinado SID de usuário, permitindo determinar o usuário conectado na máquina alvo. A função *NtOpenKeyEx* (F121) é usada para abrir chaves do registro do *Windows* e modificar suas configurações, permitindo ao *Ryuk*

[4]Disponível em: `https://malapi.io/`

ser executado automaticamente na inicialização do sistema. A função *Process32NextW* (F57) é usada para identificar processos específicos que podem interferir na execução ou operações do *malware*.

5.3. Estudo do CryptoLocker

O *CryptoLocker* possui uma grande escala e popularidade. Na Figura 4 é possível observar o resultado da seleção de *features* baseada nos estudos em cima desse *ransomware*.

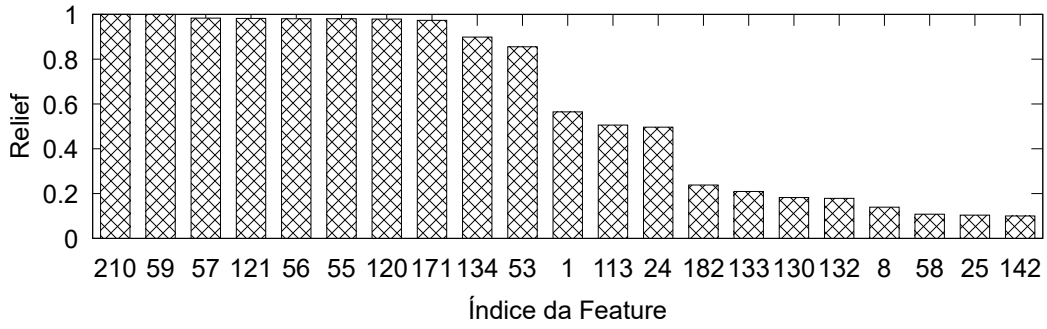

Figura 4. Top 20 features selecionadas para o CryptoLocker.

A partir das 211 *features* geradas, aquelas classificadas como as 20 mais representativas por meio da aplicação do algoritmo *Relief* são: *EnumServicesStatusW* (F210), *LookupAccountSidW* (F59), *Process32NextW* (F57), *NtOpenKeyEx* (F121), *Process32FirstW* (F56), *CreateToolhelp32Snapshot* (F55), *NtQueryKey* (F120), *ShellExecuteExW* (F171), *GetVolumePathNamesForVolumeNameW* (F134), *GlobalMemoryStatusEx* (F53), *score* (F1), *NtTerminateProcess* (F113), *NtQueryValueKey* (F24), *LookupPrivilegeValueW* (F182), *RegEnumKeyW* (F133), *OleInitialize* (F130), *GetVolumeNameForVolumeMountPointW* (F132), *SetUnhandledExceptionFilter* (F8), *NtOpenProcess* (F58), *NtClose* (F25), *CreateProcessInternalW* (F142).

Durante sua execução, o *malware* se utiliza de algumas APIs, tendo sua principal utilização a função *EnumServicesStatusW* (F210). Como dito anteriormente, essa é a mesma utilizada por *malwares* para realizar a enumeração de serviços em execução do *Windows*. As outras três funções que o *malware* utiliza são, a função *LookupAccountSidW* (F59), que permite que o *malware* possa obter o nome da conta associada a um determinado SID (identificador de segurança) do usuário, a função *Process32NextW* (F57) que, como dito anteriormente, é utilizada para listar processos em execução no sistema e permitir que o *malware* identifique e desative serviços e por fim a função *NtOpenKeyEx* (F121), que é usada para abrir uma chave específica no registro do *Windows*.

O *ransomware CryptoLocker* a utiliza para abrir a chave do registro *"Run"* e adicionar um valor que aponta para seu próprio arquivo de execução, garantindo que o *malware* seja executado automaticamente sempre que o sistema é iniciado. Essa técnica é conhecida como "persistência de inicialização" e é usada por muitos tipos de *malwares* para garantir sua execução contínua. É importante monitorar as chaves críticas do registro para detectar e impedir tentativas de modificação maliciosa por parte de *malwares*.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou um estudo sobre a seleção de *features* para a detecção dos *Ransomwares WannaCry, Ryuk e CryptoLocker*. É importante destacar a relevância do estudo apresentado neste trabalho para a compreensão dos *malwares* e para o desenvolvimento de novas técnicas de prevenção aos seus ataques. Esses *ransomwares* são exemplos de ameaças cibernéticas que podem causar danos significativos a organizações, desde a perda de dados e informações valiosas até prejuízos financeiros. Por isso, o estudo desses *malwares* é fundamental para aprimorar a segurança cibernética e evitar potenciais danos. O estudo e conhecimento de cada API citadas nesta pesquisa pode auxiliar na criação de técnicas de medidas de segurança mais eficazes para combater ataques de *malware*.

Como trabalhos futuros, pretende-se (i) utilizar o GRASP-FS [Quincozes et al. 2022] a fim de refinar o processo de seleção de *features*, (ii) aplicar algoritmos de aprendizado de máquina para avaliar o desempenho na detecção, (iii) estudar acerca do comportamento de *ransomwares* em outros sistemas operacionais e (iv) analisar o comportamento de outros *ransomwares* centrados na infecção de ICSs.

Referências

Abbasi, M. S., Al-Sahaf, H., Mansoori, M., and Welch, I. (2022). Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for feature selection. *Applied Soft Computing*, 121:108744.

Abbasi, M. S., Al-Sahaf, H., and Welch, I. (2020). Particle Swarm Optimization: A Wrapper-Based Feature Selection Method for Ransomware Detection and Classification. In *Applications of Evolutionary Computation: 23rd European Conference, EvoApplications 2020*, pages 181–196. Springer.

Beaman, C., Barkworth, A., Akande, T. D., Hakak, S., and Khan, M. K. (2021). Ransomware: Recent advances, analysis, challenges and future research directions. *Computers & Security*, 111:102490.

Brewer, R. (2016). Ransomware attacks: detection, prevention and cure. *Network security*, 2016(9):5–9.

Chen, Q. and Bridges, R. A. (2017). Automated behavioral analysis of malware: A case study of wannacry ransomware. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 454–460.

Chen, Q., Islam, S. R., Haswell, H., and Bridges, R. A. (2019). Automated ransomware behavior analysis: Pattern extraction and early detection. In *International Conference on Science of Cyber Security*, pages 199–214. Springer.

CIS (2022). Top 10 Malware January 2022. https://www.cisecurity.org/insights/blog/top-10-malware-january-2022. Acesso em: Junho/2023.

Curran, K. (2020). Cyber security and the remote workforce. *Computer Fraud & Security*, 2020(6):11–12.

Damodaran, A., Troia, F. D., Visaggio, C. A., Austin, T. H., and Stamp, M. (2017). A comparison of static, dynamic, and hybrid analysis for malware detection. *Journal of Computer Virology and Hacking Techniques*, 13(1):1–12.

Gera, T., Singh, J., Mehbodniya, A., Webber, J. L., Shabaz, M., and Thakur, D. (2021). Dominant feature selection and machine learning-based hybrid approach to analyze android ransomware. *Security and Communication Networks*, 2021.

IBM (2022). O que é ransomware? Publicado em: https://www.ibm.com/br-pt/topics/ransomware. Acesso em: Junho de 2023.

Ilić, S. Ž., Gnjatović, M. J., Popović, B. M., and Maček, N. D. (2022). A pilot comparative analysis of the cuckoo and drakvuf sandboxes: An end-user perspective. *Vojnotehnički glasnik*, 70(2):372–392.

Kaspersky (2021). Kaspersky Security Bulletin 2021 Statistics. https://go.kaspersky.com/rs/802-IJN-240/images/KSB_statistics_2021_eng.pdf. Acesso em: Junho/2023.

Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In Sleeman, D. H. and Edwards, P., editors, *Ninth International Workshop on Machine Learning*, pages 249–256. Morgan Kaufmann.

Masid, A. G., Higuera, J. B., Higuera, J.-R. B., and Montalvo, J. A. S. (2022). Application of the sama methodology to ryuk malware. *Journal of Computer Virology and Hacking Techniques*, pages 1–34.

Megchelenbrink, W. (2010). Relief-based feature selection in bioinformatics: detecting functional specificity residues from multiple sequence alignments. Master's thesis, Radboud University Nijmegen, the Netherlands.

Micro, T. (2020). 2020 Report on Threats Affecting ICS Endpoints. Disponível em: https://documents.trendmicro.com/assets/white_papers/wp-2020-report-on-threats-affecting-critical-industrial-endpoints.pdf. Acesso em: Junho/2023.

Micro, T. (2022). Defending the expanding attack surface. Disponível em: https://documents.trendmicro.com/assets/rpt/rpt-defending-the-expanding-attack-surface-trend-micro-2022-midyear-cybersecurity-report.pdf. Acesso em: Junho/2023.

Oktavianto, D. and Muhardianto, I. (2013). *Cuckoo malware analysis*. Packt Publishing.

Quincozes, S. E., Kazienko, J. F., and Copetti, A. (2018). Avaliação de conjuntos de atributos para a detecção de ataques de personificação na internet das coisas. In *VIII Simpósio Brasileiro de Engenharia de Sistemas Computacionais*, Porto Alegre. SBC.

Quincozes, S. E., Mossé, D., Passos, D., Albuquerque, C., Ochi, L. S., and dos Santos, V. F. (2022). On the Performance of GRASP-Based Feature Selection for CPS Intrusion Detection. *IEEE Transactions on Netw. and Service Management*, 19(1):614–626.

Razaulla, S., Fachkha, C., Markarian, C., Gawanmeh, A., Mansoor, W., Fung, B. C., and Assi, C. (2023). The age of ransomware: A survey on the evolution, taxonomy, and research directions. *IEEE Access*.

Uppal, D., Mehra, V., and Verma, V. (2014). Basic survey on malware analysis, tools and techniques. *Int. Jrnl on Computational Sciences & Applications (IJCSA)*, 4(1):103.

Urooj, U., Al-rimy, B. A. S., Zainal, A., Ghaleb, F. A., and Rassam, M. A. (2022). Ransomware detection using the dynamic analysis and machine learning: A survey and research directions. *Applied Sciences*, 12(1):172.