XXI Escola Regional de Redes de Computadores / VIII Workshop de Segurança (ERRC). DOI: 10.5753/errc.2024.4558

ARTIGO COMPLETO/FULL PAPER

Silly Putty: Análise Estática e Dinâmica de remote access Trojan (RAT)

Silly Putty: Static and Dynamic Remote Access Trojan (RAT) Analysis

Artur Flores • arturcfoo@gmail.com
Universidade Federal do Rio Grande (FURG)

Giancarlo Lucca • giancarlo.lucca@ucpel.edu.br
Universidade Católica de Pelotas (UCPel)

Ricardo L. dos Santos • ricardo.santos@restinga.ifrs.edu.br
IFRS - Restinga (IFRS)

André Riker • ariker@ufpa.br
Universidade Federal do Pará (UFPA)

Bruno L. Dalmazo • dalmazo@furg.br
Universidade Federal do Rio Grande (FURG)

RESUMO. Este trabalho propõe uma abordagem específica para a análise do *malware* Silly Putty, combinando técnicas estáticas e dinâmicas, utilizando ambientes controlados como SandBox e Máquinas Virtuais. A utilização de SandBox e Máquinas Virtuais permite a criação de ambientes isolados e controlados, possibilitando a observação segura e eficaz do *malware* em ação. A análise estática fornece uma visão detalhada da estrutura do código, facilitando a identificação de características específicas e potencialmente maliciosas. Por outro lado, a análise dinâmica revela comportamentos do *malware* durante a execução, como tentativas de comunicação com servidores remotos, manipulação de arquivos e interações com o sistema operacional hospedeiro. A integração dessas abordagens visa aprimorar a detecção e compreensão de ameaças virtuais, contribuindo para o desenvolvimento de estratégias mais eficientes de prevenção e resposta a incidentes de segurança.

PALAVRAS-CHAVE: Malware • Análise • Sandbox • Silly Putty

1 Introdução

A análise de *malware* é uma área que se dedica a examinar e entender as ameaças cibernéticas que podem afetar sistemas e dispositivos. O *malware*, ou *software* malicioso, pode assumir várias formas, desde cavalos de Troia até *ransomware* e *spyware* [1]. Esses programas maliciosos são projetados para causar danos, roubar informações e interromper a operação de computadores e redes. A análise de *malware* almeja detectar, classificar e desarmar essas ameaças, permitindo que empresas e indivíduos protejam seus sistemas e dados contra ataques. Essa área da ciência da computação envolve uma variedade de técnicas e ferramentas, incluindo engenharia reversa, análise de tráfego de rede e técnicas de *sandboxing*. O objetivo final da análise de *malware* é ajudar a criar defesas robustas e eficazes contra ameaças cibernéticas e garantir a segurança digital [2, 3].

A análise de *malware* tem sido um tema de pesquisa ativo na comunidade acadêmica há muitos anos, resultando em uma ampla variedade de trabalhos acadêmicos relevantes. Um exemplo é o artigo "*A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis*" de Rami Sihwail, que trata das técnicas e ferramentas utilizadas na análise de *malware*, incluindo a engenharia reversa, a análise dinâmica e a análise de tráfego de rede. Outro trabalho acadêmico importante é "*Sandnet: Network Traffic Analysis of Malicious Software*" de Marco Cova que apresenta uma técnica de análise de tráfego de rede que permite a identificação de comunicações maliciosas entre *malware* e seus servidores de comando e controle.

O *Silly Putty* é considerado uma ameaça representativa por ser um*Remote Access Trojan* (RAT) que explora a popularidade do programa legítimo PuTTY, permitindo ao atacante controle remoto do sistema infectado para capturar dados, manipular arquivos e monitorar o sistema. Embora a verificação de *hash* possa identificar se o executável foi alterado ao compará-lo com o *hash* do programa original, esse método tem limitações, pois o *Silly Putty* pode usar técnicas de modificação e ofuscação para evitar detecção por simples comparação de hashes, gerando hashes únicos a cada instalação por meio de polimorfismo e metamorfismo. Além

Artur é graduando em Engenharia de Computação e Dalmazo é Prof. Adjunto do Centro de Ciências Computacionais na FURG.

disso, análises estática e dinâmica são essenciais para entender a fundo como o *malware* realiza suas ações maliciosas, possibilitando a criação de assinaturas de detecção avançadas e o desenvolvimento de contramedidas eficazes, que vão além dos hashes e incluem padrões de comportamento. Essas análises ajudam na compreensão detalhada da ameaça e na detecção de novas variantes, tornando a segurança mais robusta e proativa.

Nesse contexto, esse trabalho busca fornecer uma análise de *malware* utilizando técnicas de análise dinâmica e estática para identificação e reconhecimento de comportamento malicioso. Estas abordagens são importante, pois muitas vezes o *malware* é projetado para evitar a detecção por técnicas de análise estática, tornando a análise dinâmica uma abordagem eficaz para identificar comportamentos maliciosos e mitigar ameaças em potencial, e vice-versa. A aplicação dessa proposta será apresentada em um estudo de caso detalhado utilizando tais técnicas para o *malware* Silly Putty.

2 Trabalhos Relacionados

Essa seção está organizada como revisão sistemática. Dessa forma, um pesquisador interessado verificar ou avançar nesse estudo, pode seguir exatamente os mesmos passos e critérios que utilizamos para incluir artigos de nossa pesquisa. Devido a restrições de página, serão colocados apenas os artigos mais relevantes.

2.1 Metodologia e Critérios

Nossa revisão sistemática teve como foco abordar artigos de pesquisa que tratam de mecanismos de defesa contra *malware* no contexto de análise de *malware*. Formamos nossa sequência de busca e a utilizamos para consultar conferências e periódicos-chave de segurança da computação publicados pelo IEEE, em busca de manuscritos que contivessem, em seus resumos, pelo menos uma ocorrência dos termos selecionados.

> (cybersecurity) AND (*sandbox*) AND (*malware* analysis) AND (*malware* classification) AND (*malware* detection) OR (*malware* behavior)

A seleção dos artigos para esta revisão sistemática foi realizada seguindo métodos rigorosos e bem definidos. Como o foco do trabalho são softwares malignos que atacam computadores, utilizamos como critério a inclusão de artigos relacionados especificamente a vírus de computador, tanto em conferências quanto em periódicos.

A escolha dos artigos da IEEE como fonte se deu devido à sua reputação e reconhecimento como uma das principais organizações acadêmicas e técnicas no campo da engenharia e ciência da computação [4]. A IEEE publica um grande número de conferências e periódicos de alta qualidade, abrangendo diversas áreas de pesquisa, incluindo análise de *malware*.

Após a aplicação dos termos de pesquisa, obtivemos inicialmente 1.259 resultados. Em seguida, aplicamos os critérios de inclusão (IC), que consistiam em selecionar artigos relacionados a vírus de computador, publicados em conferências no período de 2018 a 2024. Após a aplicação desses critérios, obtivemos um total de 24 artigos que atendiam a todas as condições mencionadas. Para restringir nosso escopo e usar apenas dados validados utilizamos critérios de exclusão (EC) para a seleção dos artigos que são fundamentais garantindo a qualidade e a relevância das fontes utilizadas neste trabalho. O critério de exclusão refere-se à não publicação dos artigos em conferências, revistas e periódicos de Q1 (Quartil 1) SCOPUS.

Outro critério de exclusão estabelecido é a não inclusão de artigos que sejam manuscritos de livros ou artigos de acesso antecipado. Essa restrição é importante para assegurar a utilização de fontes que tenham passado por um processo de revisão por pares ou que estejam em sua forma final de publicação. Livros ainda não publicados podem apresentar informações não validadas ou passíveis de alterações, o que pode comprometer a confiabilidade dos resultados obtidos. Da mesma forma, artigos de acesso antecipado podem conter resultados preliminares que ainda não foram devidamente analisados e validados pela comunidade científica.

2.2 Resumo dos Artigos selecionados

Esta seção se dedica a revisar estudos relacionados ao escopo desta proposta. Foram selecionados trabalhos que fizeram contribuições na área de detecção de assinaturas e anomalias. Por questões de restrição de espaço, abordaremos somente a discussão sobre as principais contribuições e limitações dos artigos.

Com base nas leituras ([5–7]), podemos observar algumas características em comum nos artigos encontrados. Embora os estudos discutam a importância de desenvolver sistemas antimalware resilientes, combinando abordagens automatizadas de *machine learning* com engenharia robusta ([8, 9]), e mencionem o uso de aprendizado de máquina para detecção de malwares ([10, 11], não há uma discussão específica sobre a análise de vírus específicos para o sistema operacional Windows por meio de *sandbox*.

Tabela 1. Tabela de Critérios de Inclusão e Exclusão

Critérios de Inclusão	
IC1	O artigo foi publicado entre 2018-2023
IC2	O artigo está no contexto de Vírus de computador e propõe uma abordagem para análise e detecção do vírus
Critérios de Exclusão	
EC1	O artigo não foi publicado em uma conferência, periódico ou revista

Tabela 2. Resultado dos critérios

Critérios	**Artigos subtraídos**
IEEE Explorer	1259
IC1	697
IC2	538
EC1	10
Total de Manuscritos	14

Assim, podemos destacar que a lacuna existente na análise de vírus específicos via *sandbox* é a falta de abordagens específicas que explorem o uso de ambientes controlados para executar e observar o comportamento de vírus específicos. Embora os estudos mencionem diferentes técnicas de detecção e análise de malwares, não há uma discussão específica sobre a análise detalhada de vírus específicos em ambientes isolados e controlados [12]. A utilização de sandboxes permitiria uma análise mais aprofundada e segura desses vírus, fornecendo informações valiosas para aprimorar os sistemas de detecção e defesa contra ameaças cibernéticas.

3 Proposta

Neste trabalho, apresentamos a proposta de realizar uma abordagem detalhada de um *malware* RAT, denominado Silly Putty, como estudo de caso, mostrando o uso das metodologias desenvolvidas até hoje para a análise dinâmica e estática de *malware*. Através de uma investigação dos atributos e funcionalidades, almejamos contribuir para a compreensão mais aprofundada da natureza dos vírus e, assim, fornecer subsídios para o desenvolvimento de estratégias efetivas de detecção, prevenção e mitigação de futuras ameaças similares.

O método conceitual se baseia na representação visual e organizada dos elementos-chave envolvidos na investigação dessa ameaça virtual, com o objetivo de estabelecer conexões claras e precisas entre suas características e o comportamento apresentado. Por meio de uma abordagem sistêmica, buscaremos elucidar as principais etapas da análise, definir os conceitos essenciais e suas relações, proporcionando uma base sólida para o desenvolvimento de estratégias eficientes de detecção e mitigação. Essa abordagem metodológica se mostra fundamental para a compreensão mais profunda do vírus Silly Putty e suas possíveis implicações na segurança digital contemporânea.

O modelo conceitual proposto para a análise estática e dinâmica de *malware* é estruturado em quatro etapas. Primeiramente, a "Preparação do ambiente" é executada, envolvendo a configuração de um ambiente controlado e seguro para a análise. Em seguida, na etapa "Obtenção das amostras", as amostras de *malware* são adquiridas de fontes confiáveis com várias opções de códigos-fonte e de plataformas, garantindo uma representação abrangente de ameaças. Essa amostra é colocada no ambiente controlado (A). A terceira etapa se divide em duas vertentes: "Análise dinâmica"(B2) e "Análise estática"(B1). Na análise dinâmica, o *malware* é executado em um ambiente virtual monitorado, permitindo a observação do seu comportamento em tempo real e a captura de atividades maliciosas. Por outro lado, na análise estática, o código do *malware* é examinado sem execução, revelando características como estrutura, strings embutidas e recursos suspeitos. Por fim, os resultados das análises dinâmica e estática são discutidos em um relatório (C), com informações detalhadas sobre o *malware*, auxiliando na compreensão de padrões, na detecção futura e no desenvolvimento de contramedidas.

3.1 Setup

Primeiro é necessário criar um ambiente controlado e seguro para a análise de um *malware*. Como discutido anteriormente, usamos uma metodologia de Sandbox. Para construirmos um ambiente *sandbox* foi utilizada a ferramenta VitualBox. A máquina virtual utiliza o sistema operacional Windows 7 de 64 bits. Após configurar a máquina virtual, instalamos o FLARE VM. FLARE VM é uma coleção de scripts de instalação de software para sistemas Windows que permite configurar e manter facilmente um ambiente de engenharia reversa em uma máquina virtual (VM).

Para obtermos as amostras de vírus, acessamos um grande repositório de amostras no Github, que contém

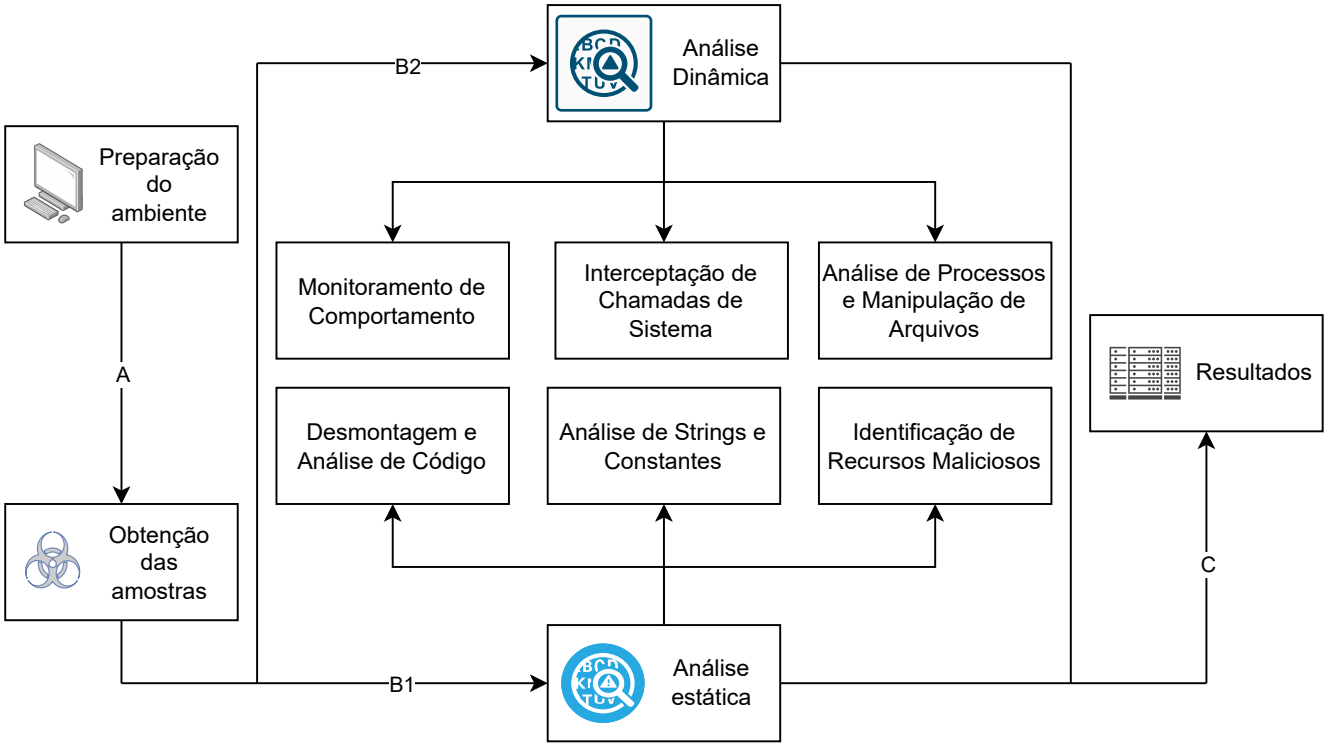

Figura 1. Modelo conceitual da análise.

mais de 2.500 amostras de *malware* e códigos-fonte para diversas plataformas. Após garantirmos que o ambiente Sandbonx está configurado corretamente fazemos o download de um dos malwares mais conhecidos e estudados para garantirmos que a metodologia e as ferramentas geram soluções satisfatórias [1].

A análise estática é uma etapa fundamental na compreensão e identificação de ameaças cibernéticas, permitindo que os pesquisadores de segurança e analistas de *malware* revelem detalhes importantes sobre o funcionamento do código malicioso sem a necessidade de executá-lo, reduzindo assim os riscos potenciais associados à análise dinâmica.

A análise dinâmica permite que pesquisadores e analistas compreendam melhor o comportamento do *malware* e, assim, desenvolvam contramedidas eficazes. Ela oferece informações valiosas sobre as táticas e técnicas do *malware*, bem como seu impacto potencial, auxiliando na identificação de ameaças e no desenvolvimento de estratégias de defesa mais robustas.

O FLARE VM abrange disassembladores e debuggers e utilitários específicos, proporcionando um ambiente completo e eficiente para análise de ameaças. Essas ferramentas são fundamentais para observação do funcionamento interno de códigos maliciosos, permitindo que pesquisadores compreendam as nuances das ameaças digitais.

4 Avaliação

Neste capítulo, iremos realizar a detonação controlada do *malware* Silly Putty e analisá-lo dentro da máquina virtual. O Silly Putty é um *malware* conhecido por suas capacidades de infecção e evasão, tornando-se um desafio significativo para analistas de segurança. Este *malware* se destaca por utilizar técnicas de polimorfismo, alterando seu código a cada infecção para evitar a detecção por softwares antivírus.

Para iniciar a análise estática preliminar do *malware*, é usado duas ferramentas principais: o PowerShell para obter o endereço hash do arquivo e o FLOSS para extrair strings ocultas e ofuscadas do *malware*. Primeiramente, é verificado o hash SHA-256 do arquivo *putty.exe* para verificar a integridade do arquivo e para identificação única do *malware*. Utilizamos o seguinte comando no PowerShell:

```
Get-FileHash -Algorithm SHA256 putty.exe
```

Este comando gera o hash SHA-256, que é um identificador exclusivo para o arquivo, essencial para compa-

[1] Devido as restrições de página, mais detalhes da amostragem, dos resultados do estudo, juntamente com imagens e o código fonte estão disponíveis no repositório: https://github.com/36thChamber/silly-putty-analysis

rações e verificações posteriores. Em seguida, o FLOSS (FireEye Labs Obfuscated String Solver) é empregado para extrair strings do *putty.exe*. O FLOSS é uma ferramenta para detectar strings que podem estar escondidas dentro do *malware*. O seguinte comando foi utilizado:

```
floss -n 8 putty.exe > floss.txt
```

Este comando extrai as strings do *putty.exe* e as salva no arquivo *floss.txt*. O parâmetro `-n 8` especifica que o FLOSS deve procurar por strings de pelo menos 8 caracteres de comprimento. A análise das strings extraídas serve para identificar comportamentos suspeitos e funções ocultas no *malware*, fornecendo uma base para investigações mais aprofundadas.

A combinação dessas ferramentas permite identificar aspectos críticos da estrutura e comportamento do *malware* antes mesmo de executá-lo em um ambiente controlado. A análise das *strings* revelou que o arquivo executável possui 159 strings possivelmente maliciosas, e também 52 possíveis funções do Windows API com finalidade maliciosa.

O putty.exe é um programa gratuito e de código aberto para Windows que suporta vários protocolos de rede que permitem estabelecer uma conexão segura para transferir informações sensíveis. Portanto, este *malware* funciona através de um programa legítimo por uma Backdoor. Diversas funções e strings levantadas como flags maliciosas são necessárias para o funcionamento do programa legítimo. Assim, é impossível definir de forma concreta, com as informações levantadas até o momento que este executável é de fato malicioso.

Para categorizar o putty.exe como malicioso, é necessário a análise dinâmica preliminar. Após a detonação do executável é possível notar a abertura de um terminal *powershell* por alguns instantes. Em comparação, o programa original não executa nenhum comando via terminal. A primeira análise necessária é a identificação do *payload* gerado imediatamente após a detonação do putty.exe.

Utilizando o procmon, uma ferramenta de monitoramento avançado para Windows, que fornece informações em tempo real sobre o sistema de arquivos, registro, processos e atividades de *threads*, é possível identificar a primeira operação do executável, um PROCESS START com ID 2948. E novamente utilizando o procmon e filtrando pelo ID do processo podemos identificar a execução de um powershell como primeira operação do executável após a detonação. É possível notar que há uma Base64String encriptada e enviada para o objeto comprimido. Para identificação dessa *string* é utilizado a segunda VM Remnux. Esse comando gera um arquivo comprimido denominado out. Após a extração desse arquivo é recebido o algoritmo do payload que está agindo no powershell.

O próximo passo é definir o endereço DNS que foi demandado após a detonação. Para isso é utilizado o wireshark, que permite capturar pacotes de dados que trafegam pela rede e analisá-los em detalhe. Após a reinicialização da sessão do Putty podemos detectar uma query suspeita através do log do Wireshark. Porém, essa query não obtém resposta porque o InetSim não está em execução e, portanto, não há nenhum manipulador ou resposta para essa solicitação de DNS. Após o InetSim ser executado e feito o teste de rede, é possível definir qual o objetivo da chamada DNS feita. Após refeita a chama DNS segue-se um protocolo TCP detectado pelo Wireshark. Nota-se que uma porta ostensiva e efêmera comunica-se com outra de número 8443 que possivelmente é a porta de retorno de chamada.

Portanto é usado TCPview e após reiniciada a sessão do Putty é possível perceber que filtrando as chamadas powershell e TCP acontece uma chamada vinda da porta 8443. A partir disso é se confirma que há uma chamada para comunicação remota. O protocolo de callback é HTTPS, portanto não é possível interceptar essa chamada sem um certificado x509. Uma tentativa de interceptação é possível ao trocamos o arquivo host para local. Utilizando um sessão ncat na porta que recebe a chamada é possível uma tentativa de interceptação da chamada.

5 Conclusão

A análise do *malware* Silly Putty apresentou uma série de desafios devido às suas técnicas avançadas de evasão e polimorfismo. Através de uma abordagem sistemática utilizando diversas ferramentas de análise estática e dinâmica, foi possível identificar comportamentos suspeitos e funções maliciosas no executável putty.exe.

A análise estática inicial, utilizando PowerShell e FLOSS, revelou que o arquivo continha várias strings e funções potencialmente maliciosas. PEview permitiu uma inspeção detalhada da estrutura interna do executável, reforçando as suspeitas iniciais de atividades maliciosas. A detonação do *malware* em um ambiente virtual revelou a execução de um comando PowerShell logo após a inicialização, algo que o programa legítimo putty.exe não realiza. Utilizando o procmon, foi possível rastrear a atividade inicial do *malware*, destacando a criação de um objeto para manipulação de um pacote GZipStream, sugerindo compressão e potencial exfiltração de dados.

Além disso, a captura de pacotes de rede com Wireshark e a análise das chamadas de rede demonstraram tentativas de comunicação com um servidor de comando e controle (C2). A detecção de uma query DNS suspeita e subsequente comunicação via porta 8443 indicou a utilização de um canal HTTPS para comunicação segura, dificultando a interceptação e análise do tráfego sem um certificado TLS legítimo.

Enquanto as ferramentas e técnicas utilizadas forneceram *insights* valiosos sobre o funcionamento de malwares em geral, a encriptação robusta utilizada pelo *malware* impede a análise completa de suas atividades pós-conexão. Isso ressalta a necessidade contínua de aprimoramento das ferramentas de análise e da implementação de estratégias de segurança em camadas para defender contra tais ameaças avançadas.

Declarações complementares

Financiamento

Os autores expressam seu agradecimento à FAPERGS (24/2551-0001396-2) e à FAPERGS/CNPq (23/2551-0000126-8; 23/2551-0000773-8) pelo apoio financeiro para a realização deste trabalho.

Referências

1 Ucci, D.; Aniello, L.; Baldoni, R. Survey of machine learning techniques for malware analysis. *Computers & Security*, Elsevier, v. 81, p. 123–147, 2019.

2 Dalmazo, B. L.; Vilela, J. P.; Curado, M. Triple-Similarity Mechanism for alarm management in the cloud. *Computers & Security*, v. 78, p. 33–42, 2018. ISSN 0167-4048. DOI: https://doi.org/10.1016/j.cose.2018.05.016. Disponível em: https://www.sciencedirect.com/science/article/pii/S0167404818306515.

3 Ayres, D. *et al.* Comparando Médias Móveis com Integral de Choquet para Detectar Anomalias no Tráfego de Redes. *In*: ANAIS Estendidos do XXIV Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. São José dos Campos/SP: SBC, 2024. P. 353–357. DOI: 10.5753/sbseg_estendido.2024.243381. Disponível em: https://sol.sbc.org.br/index.php/sbseg_estendido/article/view/30154.

4 Cardoso, F. C. *et al.* Echo state network and classical statistical techniques for time series forecasting: A review. *Knowledge-Based Systems*, v. 293, p. 111639, 2024. ISSN 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2024.111639. Disponível em: https://www.sciencedirect.com/science/article/pii/S0950705124002740.

5 Singhal, M.; Levine, D. Analysis and Categorization of Drive-by Download Malware. *In*: 2019 4th International Conference on Computing, Communications and Security (ICCCS). 2019. P. 1–4. DOI: 10.1109/CCCS.2019.8888147.

6 Chen, L. *et al.* AVMiner: Expansible and Semantic-Preserving Anti-Virus Labels Mining Method. *In*: 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). 2022. P. 217–224. DOI: 10.1109/TrustCom56396.2022.00039.

7 Rohan, A.; Basu, K.; Karri, R. Can Monitoring System State + Counting Custom Instruction Sequences Aid Malware Detection? *In*: 2019 IEEE 28th Asian Test Symposium (ATS). 2019. P. 61–615. DOI: 10.1109/ATS47505.2019.00007.

8 Alghamdi, S. M.; Othathi, E. S.; Alsulami, B. S. Detect keyloggers by using Machine Learning. *In*: 2022 Fifth National Conference of Saudi Computers Colleges (NCCC). 2022. P. 193–200. DOI: 10.1109/NCCC57165.2022.10067780.

9 Ellahi, O.; Shah, M. A.; Usman Rana, M. The ingenuity of malware substitution: Bypassing next-generation Antivirus. *In*: 2021 26th International Conference on Automation and Computing (ICAC). 2021. P. 1–5. DOI: 10.23919/ICAC50006.2021.9594221.

10 Kuruvila, A. P.; Kundu, S.; Basu, K. Analyzing the Efficiency of Machine Learning Classifiers in Hardware-Based Malware Detectors. *In*: 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). 2020. P. 452–457. DOI: 10.1109/ISVLSI49217.2020.00-15.

11 Sharma, D.; Verma, H. K. Malware Signature and Behavior Performance Evaluation utilizing Packers. *In*: 2022 2nd Asian Conference on Innovation in Technology (ASIANCON). 2022. P. 1–8. DOI: 10.1109/ASIANCON55314.2022.9909111.

12 Santo, Y. *et al.* Fault Detection on the Edge and Adaptive Communication for State of Alert in Industrial Internet of Things. *Sensors*, v. 23, n. 7, 2023. ISSN 1424-8220. DOI: 10.3390/s23073544. Disponível em: https://www.mdpi.com/1424-8220/23/7/3544.