Hänsel und Gretel: algoritmo para melhoria da resiliência cibernética pela diversificação de ativos através de aprendizado de máquina

Fernando Nunes de Almeida[1], **Antonio Eduardo Carrilho da Cunha**[1], **Anderson Fernandes Pereira dos Santos**[2], **Paulo César Pellanda**[1]

[1]Programa de Pós-graduação em Engenharia de Defesa
[2]Programa de Pós-graduação em Sistemas e Computação
Instituto Militar de Engenharia
CEP 22.290-270 – Rio de Janeiro – RJ – Brasil

{nunes.fernando,carrilho,anderson,pellanda}@ime.eb.br

Abstract. *Cybersecurity is crucial in all sectors of modern society due to the constant emergence of new threats. In this context, asset diversification is a valuable tool to limit or even prevent the spread of malware. This work presents an ongoing research for the development of an approach that aims to improve the cyber resilience of an industrial system by diversifying network resources using machine learning to find critical paths and safer alternatives.*

Resumo. *A segurança cibernética é crucial em todos os setores da sociedade moderna devido ao surgimento constante de novas ameaças. Neste contexto a diversificação de ativos é uma ferramenta valiosa para limitar ou mesmo impedir a propagação de um malware. Este trabalho apresenta uma pesquisa em andamento para o desenvolvimento de uma abordagem que almeja melhorar a resiliência cibernética de um sistema industrial diversificando os recursos de redes utilizando aprendizado de máquina para encontrar os caminhos críticos e as alternativas mais seguras.*

1. Introdução

Com a crescente dependência da sociedade moderna em sistemas de tecnologia da informação, a segurança cibernética se tornou uma questão crítica em todos os setores. Novas ameaças à segurança da informação surgem constantemente na forma de *malware*, com um grande número de ocorrências relatadas [Raj Samani 2021].

Na revisão deste estudo, identificaram-se algumas técnicas de proteção para sistemas industriais de controle, incluindo Sistemas de Detecção de Intrusões (IDS), Avaliação de Riscos e Métricas e Simulação de Segurança [Asghar et al. 2019]. Cada técnica tem vantagens e limitações relacionadas à implementação, custos e especificidade industrial. A solução de diversificação proposta aqui não requer aquisição de novos equipamentos ou mudanças significativas, apresentando um baixo impacto para o negócio.

A resiliência cibernética, conforme discutida aqui, refere-se à análise da capacidade de um sistema de computador para continuar operando ao longo do tempo, mesmo quando sujeito a ataques e problemas. Isso pode ser avaliado por meio de uma função

$F(t)$, que essencialmente demonstra o desempenho do sistema no cumprimento de sua missão principal ao longo do tempo [Ellis et al. 2022].

Nesse contexto, a diversificação de ativos, englobando desde recursos de rede até *software*, torna-se uma estratégia que pode tornar a infraestrutura de tecnologia da informação resiliente a ataques, dificultando a propagação de ameaças. Esta pesquisa propõe a aplicação de técnicas de diversificação de ativos para tornar uma infraestrutura resiliente a ataques, minimizando os danos causados por *malware*.

2. Trabalhos Relacionados

Nesta seção, é apresentado um breve panorama dos estudos anteriores relevantes sobre o objeto deste estudo e sua aplicação em segurança cibernética em sistemas industriais, abordando aspectos de cada abordagem.

Em [Li et al. 2018], é apresentada a métrica de similaridade de vulnerabilidade entre dois ativos, definida a partir da similaridade de *Jaccard* [Choi et al. 2010]. Os autores modelaram o problema em um processo de decisão de Markov e aplicaram o algoritmo *Sequential Tree-Reweighted Message Passing - TRW-S* para minimizar a função de energia. O valor mínimo desta função resultaria da atribuição ideal de *software* para a execução de cada serviço em cada *host*.

Em [Lin et al. 2017] utilizando técnicas de aprendizado de máquina, um Sistema de Detecção de Intrusões (IDS) incorpora pacotes de camada de link (camada 2) para estabelecer padrões normais em redes de controle industrial reconhecendo anomalias na rede, já o estudo de [Prieto et al. 2021] aborda a exploração simultânea de vulnerabilidades *zero-day* em nós de rede, propondo uma estratégia de migração para criar uma rede heterogênea e minimizar impactos de ataques *zero-day*.

Em um estudo mais recente, [Zhang et al. 2021] apresentam um *framework* de adaptação de rede DREVAN (*Deep REinforcement Learning-based Vulnerability-Aware Network Adaptations*). O objetivo desse *framework* é construir uma topologia de rede robusta contra ataques epidêmicos, levando em consideração a vulnerabilidade de segurança causada pela monocultura de *software*.

A maioria dos outros trabalhos não utiliza técnicas de inteligência artificial. Essa constatação indica uma lacuna na literatura e sugere uma oportunidade para pesquisas que explorem a aplicação de aprendizado de máquina em conjunto com diversificação de recursos de rede para prover resiliência cibernética sem, no entanto, implicar em alteração na topologia da rede.

3. Algoritmo Hänsel und Gretel

Assim como na fábula de Hänsel und Gretel, onde as crianças deixaram migalhas de pão para encontrar o caminho de volta para casa, o algoritmo proposto no presente trabalho busca encontrar um caminho com as melhores combinações de *software* afim de atingir a segurança desejada.

Considere uma rede $N = \langle H, L, S, P \rangle$, em que H representa um conjunto de *hosts* $\{h_1, \ldots, h_n\}$, L define as conexões diretas entre pares de *hosts* ($L \subseteq H \times H$). Por sua vez, S define um conjunto de serviços disponíveis $S = \{s_1, \ldots, s_m\}$, em que $S_{h_i} \subseteq S$ indica o conjunto de serviços disponíveis no *host* h_i. Esses serviços são executados por um

conjunto de produtos $P = \{p_1, \ldots, p_q\}$, em que $p(s_j) = \{p^1_{s_j}, \ldots, p^l_{s_j}\}$, e onde $p^x_{s_j} \in P$. A seguir, encontra-se a definição da atribuição de produtos a um *host*.

DEFINIÇÃO 1 *(**ATRIBUIÇÃO DE PRODUTOS A UM Host**). Dada uma rede $N = \langle H, L, S, P\rangle$, uma atribuição de produtos é definida por $\alpha' : H \times S \to P$, de modo que: $\alpha'(h_i, s_j)$ é a atribuição de produto para um serviço $s_j \in S_{h_i}$ no host $h_i : \alpha'(h_i, s_j) = p^x_{s_j}$. A atribuição para todos os serviços em um host $h_i \in H$ pode ser definida por $\alpha : H \times 2^S \to 2^P$:*

$$\alpha(h_i, S_{h_i}) = (\alpha'(h_i, s_1), \ldots, \alpha'(h_i, s_k))$$

$$\alpha(h_i, S_{h_i}) = \{p^m_{s_1}, \ldots, p^n_{s_k}\}$$

$$\textit{em que } p^m_{s_1} \in p(s_1), \ldots, p^n_{s_k} \in p(s_k).$$

Esta atribuição também pode ser interpretada como uma das diversas combinações possíveis para o conjunto S_{h_i}.

DEFINIÇÃO 2 *(**ATRIBUIÇÃO GLOBAL DE PRODUTOS**). Dada uma rede N, previamente definida, o conjunto de atribuições para todos os serviços em cada host $h \in H$ pode ser definido como:*

$$A(N) = \{\alpha(h_1, S_{h_1}), \ldots, \alpha(h_n, S_{h_n})\}$$

O conjunto $A(N)$ oferece diversas combinações de produtos para executar serviços de cada *host* em $S_{h_i} \in 2^S$. Encontrar o conjunto $A(N)$ que reduza a propagação de *malware* em redes densas é um problema de otimização objetivando aumentar a dificuldade para o *malware* m comprometer os ativos valiosos. O Algoritmo 1 implementa a abordagem deste artigo para mitigar impactos de vulnerabilidades *zero-day* de acordo com o padrão de propagação do *malware*.

Algorithm 1 Hänsel und Gretel

1: **algorithm** HANSELUNDGRETEL(N)
2: $G \leftarrow convertToGraph(N)$
3: $G' \leftarrow toLineGraph(G)$
4: $G'' \leftarrow addCombinations(G')$
5: $G''' \leftarrow applyWeight(G'')$
6: $c \leftarrow findCriticalPathWithRL(G''')$
7: $s \leftarrow findSafePathWithRL(G''', c)$
8: $N \leftarrow changeCombs(s, c, N)$
9: **end algorithm**

Inicialmente, é modelada para Algoritmo 1 a representação de uma rede real (Rede N) e após a obtenção do grafo que reflete a topologia mais similar à rede original, transforma-se mais uma vez essa modelagem utilizando o que é conhecido em teoria dos grafos como *Line Graph*, ou grafo linha, em tradução livre [Beineke and Bagga 2021].

Suponha um grafo $G = \langle H, L\rangle$ com *hosts* $H = \{h_1, \ldots, h_n\}$ e arestas $L = \{(h_i, h_j), \ldots, (h_k, h_l)\}$. Cada conexão entre *hosts* é representada por (h_i, h_j). O *Line*

Graph $G' = \langle L, K \rangle$ de G tem L como vértices, onde vértices em G' são adjacentes se as arestas em G compartilham um vértice. Isso permite representar possíveis combinações de *software* $\alpha(h_i, S_{h_i})$ como arestas. A próxima etapa envolve adicionar múltiplas arestas entre nós, o que requer definir equivalência entre vértice e aresta, bem como a lógica para acrescentá-las.

DEFINIÇÃO 3 *(***EQUIVALÊNCIA VÉRTICE-ARESTA***). Seja* $G' = \langle L, K \rangle$ *o grafo resultante da chamada ao método toLineGraph no Algoritmo 1. Com* $K \subseteq L \times L$*, em que* $((h_i, h_j), (h_j, h_k)) \in K$ *se e somente se* h_i, h_j *e* $h_k \in H$ *e* $(h_i, h_j) \in L$ *e* $(h_j, h_k) \in L$*, desta maneira assume-se que uma aresta em* K *equivale à um host em* H*:*

$$((h_i, h_j), (h_j, h_k)) \equiv h_j$$

Várias arestas seriam conectadas aos mesmos pares de vértices $((h_i, h_j), (h_j, h_k))$ distinguindo-se pela combinação de *software* $\alpha(h_j, S_{h_j})$ que cada aresta representa.

Os pesos serão calculados para cada combinação de *software* por meio da métrica similaridade média. A definição da função $sim(.,.)$, que calcula a similaridade entre dois conjuntos, baseia-se no conceito de similaridade de *Jaccard* [Li et al. 2018].

DEFINIÇÃO 4 *(***SIMILARIDADE MÉDIA DE VULNERABILIDADES***). Para cada combinação de produtos* $\alpha^l(h_j, S_{h_j}) \in \Pi_j$ *a similaridade de vulnerabilidades média entre cada software executando cada serviço com relação a um malware específico* m *pode ser calculada pela equação abaixo:*

$$avg\_sim(\alpha(h_j, S_{h_j}), m) = \frac{1}{|\alpha^l(h_j, S_{h_j})|} \sum_{p_k \in \alpha^l(h_j, S_{h_j})} sim(p_k, m)$$

Ao utilizar os valores calculados a partir da similaridade média como pesos das arestas, configura-se então o ambiente para a execução do modelo de aprendizado por reforço que é um tipo de aprendizado de máquina em que um agente interage com o ambiente, aprendendo a tomar ações que maximizem uma recompensa numérica, enquanto recebe feedback de recompensa ou penalidade após cada ação. O objetivo é que o agente aprenda a tomar ações que resultem em maiores recompensas ao longo do tempo [Russell and Norvig 2004].

O método *findCriticalPathWithRL(*G'''*)* no Algoritmo 1 percorre do *host* de entrada até um *host* alvo de valor para o atacante, revelando combinações de software críticas. O método *findSafePathWithRL(*G''', c*)*, por sua vez, calcula o caminho mais seguro, representando o oposto, já o método *changeCombs(*s, c, N*)* assinala a correção das combinações críticas em *hosts* sensíveis.

Assim, o Algoritmo 1 aprimora a resiliência do sistema de controle industrial com um processo flexível, evitando alterações na topologia e permitindo combinações de *software* legado em *hosts* vitais. A preferência de combinações pode ser feita removendo arestas de K. As restrições e limitações do mundo real devem ser consideradas, de modo que nem todos os softwares e *hosts* podem ser diversificados devido a diversas razões, como configurações restritas, sistemas legados e incompatibilidade entre determinados softwares, portanto é importante destacar que a estratégia adotada visa diversificar elementos do sistema onde essa alteração não seja um entrave

4. Avaliação de Resultados

Com o objetivo de avaliar a viabilidade técnica da proposta apresentada, nesta seção, avaliam-se experimentos utilizando o *software* NetLogo[1]. Essa ferramenta de modelagem, conhecida por sua capacidade de simular modelos de sistemas complexos e programáveis, é utilizada para construir um sistema de controle industrial em rede, conforme ilustrado na Figura 1.a. Por meio da simulação, analisa-se a propagação de *malware* e avalia-se o esforço necessário para atingir um alvo pré-determinado.

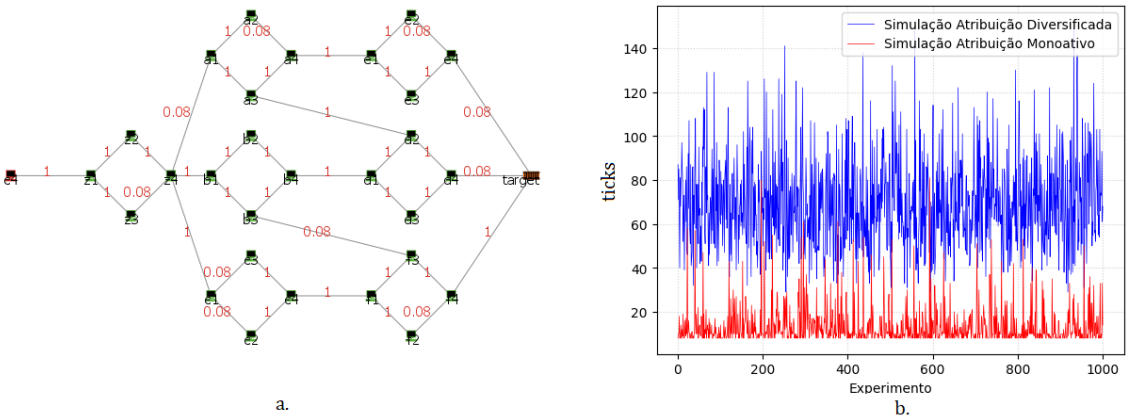

Figura 1. a. Rede simulada no NetLogo; b. Gráfico experimento $\times$ *ticks*.

4.1. Métrica utilizada

O *Mean-time-to-compromise (MTTC)*, ou Tempo Médio para Comprometimento, é a métrica utilizada para avaliar o tempo médio no qual um determinado *host* alvo será comprometido em um ataque. Essa métrica é amplamente empregada na área de segurança da informação como uma medida de vulnerabilidade e eficácia das medidas de proteção implementadas.

4.2. Simulação

Um experimento comportando a simulação de uma rede composta por 30 *hosts* foi realizado. A partir de um *host* de entrada específico, foi analisado o tempo necessário em *ticks* para alcançar um *host* alvo. O resultado do experimento pode ser visualizado no gráfico na Figura 1.b.

Ao realizar a simulação de atribuição monoativa, obteve-se um valor de MTTC de 13,51 *ticks*, representando o tempo médio para comprometimento. Por outro lado, na simulação de atribuição diversificada, o MTTC registrado foi de 69,25 ticks. Esses resultados evidenciam diferenças significativas nos esforços necessários para comprometer um determinado *host* alvo, dependendo da estratégia de atribuição adotada.

5. Considerações Finais

Este artigo aborda uma pesquisa em andamento para aprimorar a resiliência cibernética por meio de diversificação de recursos de rede usando aprendizado de máquina. Estão

[1]https://ccl.northwestern.edu/netlogo

planejados experimentos com várias topologias, variações de nós, graus e técnicas de conexão durante o treinamento por simulações, explorando um ambiente heterogêneo.

Os resultados preliminares mostram que essa abordagem possui um potencial de melhoria na mitigação de ameaças cibernéticas e no fortalecimento da segurança das redes. Entre as oportunidades de pesquisas futuras estão a otimização dos algoritmos de aprendizado de máquina empregados e a realização de experimentos em uma infraestrutura real.

Agradecimentos

Este trabalho foi desenvolvido no âmbito do Acordo de Parceria para Pesquisa, Desenvolvimento e Inovação, firmado entre o Exército Brasileiro, a ITAIPU Binacional e a Fundação Parque Tecnológico Itaipu-Brasil, no Projeto Centro de Estudos Avançados em Proteção de Estruturas Estratégicas (Ceape²), na cooperação científica nas áreas de Segurança de Infraestruturas Críticas (IEC), Segurança da Informação e Cibernética e Sistemas Elétricos de Potência e foi parcialmente financiado pelo Programa de Apoio à Pós-Graduação (PROAP) — CAPES.

Referências

Asghar, M. R., Hu, Q., and Zeadally, S. (2019). Cybersecurity in industrial control systems: Issues, technologies, and challenges. Computer Networks, 165:106946.

Beineke, L. W. and Bagga, J. S. (2021). Line graphs and line digraphs. Springer.

Choi, S.-S., Cha, S.-H., Tappert, C. C., et al. (2010). A survey of binary similarity and distance measures. Journal of systemics, cybernetics and informatics, 8(1):43–48.

Ellis, J. E., Parker, T. W., Vandekerckhove, J., Murphy, B. J., Smith, S., Kott, A., and Weisman, M. J. (2022). An experimentation infrastructure for quantitative measurements of cyber resilience. In MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM), pages 855–860. IEEE.

Li, T., Feng, C., and Hankin, C. (2018). Improving ics cyber resilience through optimal diversification of network resources. arXiv preprint arXiv:1811.00142.

Lin, C.-T., Wu, S.-L., and Lee, M.-L. (2017). Cyber attack and defense on industry control systems. In 2017 IEEE Conference on Dependable and Secure Computing, pages 524–526. IEEE.

Prieto, Y., Figueroa, M., and Pezoa, J. E. (2021). Maximizing network reliability to 0-day exploits through a heterogeneous node migration strategy. IEEE Access, 9:97747–97759.

Raj Samani (2021). McAfee labs threats report. Technical report, McAfee.

Russell, S. J. and Norvig, P. (2004). Inteligência artificial. Elsevier.

Zhang, Q., Cho, J.-H., Moore, T. J., and Nelson, F. F. (2021). Drevan: Deep reinforcement learning-based vulnerability-aware network adaptations for resilient networks. In 2021 IEEE Conference on Communications and Network Security (CNS), volume ., pages 137–145. IEEE.