Especificação de arcabouço para testes de escalação de privilégio em sistemas Linux

Gustavo Zanzin Guerreiro Martins[1], **Rodrigo Campiolo**[1]

[1]Departamento Acadêmico de Ciência da Computação
Universidade Tecnológica Federal do Paraná (UTFPR)
Campo Mourão – PR – Brasil

gustavozanzin@alunos.utfpr.edu.br, rcampiolo@utfpr.edu.br

Abstract. *This work proposes the especification and evaluation of a framework focused on privilege escalation in Linux systems, targeting entry-level professionals. The framework's definition employs a narrative review guided by specific criteria and an adapted ontology to model its initial structure. An experiment will assess the framework's impact on reducing the omission of attack vectors during security assessments. The framework is expected to contribute both to the operational systematization of privilege escalation procedures and to the technical training of beginners.*

Resumo. *Este trabalho propõe a especificação e avaliação de um arcabouço voltado à escalação de privilégios em sistemas Linux, com foco em profissionais iniciantes. A especificação do arcabouço emprega uma revisão narrativa baseada em critérios e uma ontologia adaptada para modelar sua estrutura inicial. Um experimento avaliará o impacto do arcabouço na redução de omissões de vetores de ataque em avaliações de segurança. Espera-se que o arcabouço possa contribuir na sistematização operacional e também na formação técnica do iniciante.*

1. Introdução

Testes de intrusão (também conhecidos como pentestes) são avaliações de segurança computacional em que especialistas em segurança cibernética emulam ataques reais para identificar métodos que contornem as medidas de segurança de uma aplicação, de um sistema ou de uma rede (Scarfone et al. 2008). No contexto de sistemas computacionais, um penteste pode envolver uma etapa de verificação de presença de vetores que permitam o ataque de escalação de privilégio.

O ataque de escalação (ou elevação) de privilégio visa alterar os privilégios de um usuário para os privilégios de outro usuário em um sistema computacional ou rede de computadores. Geralmente, é almejado o nível de privilégio associado a um usuário que possui permissões administrativas (Ahmed 2021).

Durante um teste de intrusão, o profissional precisa verificar exaustivamente múltiplos vetores que possam levar à elevação de privilégios. No entanto, dois desafios principais dificultam essa tarefa.

O primeiro desafio relaciona-se com a diversidade das checagens necessárias. A matriz ATT&CK (The MITRE Corporation 2025) categoriza em sua base 14 técnicas divididas em 95 subtécnicas de elevação de privilégio. Isso implica em dezenas de categorias de potenciais vetores (más configurações em permissões, programas executáveis,

bibliotecas, tarefas agendadas, variáveis de ambiente, vulnerabilidades no núcleo, etc.), cada uma subdividida em múltiplas subtécnicas, o que resulta em centenas de verificações.

O segundo desafio é referente à ausência de uma metodologia unificada. Existem diversos recursos (como listas de verificação, guias e ferramentas) que visam auxiliar o processo de enumeração de informações sobre maneiras de elevar privilégios, porém estão dispersos, apresentam lacunas conceituais, entre outras limitações discutidas na Seção 2.

Nesse sentido, até onde se tem conhecimento, não há, atualmente, um único arcabouço integrado disponível publicamente que estruture essas checagens hierarquicamente por tópicos, estabeleça uma ordem sequencial de verificação, apresente o propósito de cada etapa, oriente a execução das verificações, seja amplamente mantido e reconhecido como referência pela comunidade de cibersegurança.

A constatação da carência de um arcabouço unificado surge por consequência de uma revisão narrativa estruturada – baseada em critérios qualitativos (Seção 4) – dos principais projetos sobre enumeração de vulnerabilidades que permitem a elevação de privilégios em Linux. Os resultados, apresentados na Tabela 1, mostram que cada recurso carece de pelo menos um requisito fundamental, demonstrando essa lacuna e reforçando a necessidade de um novo arcabouço integrado.

Objetiva-se nesta pesquisa em andamento, o desenvolvimento e a avaliação de um arcabouço para averiguação de elevação de privilégios em sistemas Linux. Este artigo descreve os métodos e os primeiros resultados do desenvolvimento do arcabouço.

Parte-se da hipótese de que um arcabouço integrado que unifique aspectos como estrutura hierárquica de checagens, fluxo sequencial para verificação, instruções conceituais e práticas sobre cada etapa e manutenção contínua, pode reduzir a quantidade de vetores de elevação de privilégios omitidos por avaliadores de segurança iniciantes em comparação com o uso de recursos fragmentados.

A hipótese é sustentada pelos resultados do estudo empírico realizado por (Elder et al. 2022), que indicam que abordagens sistemáticas baseadas em metodologias padronizadas são mais eficazes (em relação a abordagens exploratórias) na identificação de vulnerabilidades críticas. Assim, supõe-se que um arcabouço semelhante, porém voltado especificamente à elevação de privilégios, possa oferecer ganhos equivalentes.

2. Trabalhos Relacionados

Diversos métodos reconhecidos pela comunidade de cibersegurança e por organizações orientam a condução de testes de intrusão. Entre outros, alguns exemplos são o *Information Systems Security Assessment Framework* (ISSAF) (OISSG 2006) e o *Penetration Testing Execution Standard* (PTES) (PTES Organization 2014). No entanto, métodos como esses não detalham especificamente checagens sobre elevação de privilégio em Linux. Por outro lado, há uma diversidade de recursos que têm em vista a enumeração de possíveis vetores desse ataque. Nesta seção são apresentados alguns exemplos desses recursos.

O HackTricks organiza sua estrutura por tópicos que atuam como uma lista de verificação, oferecendo comandos para orientações práticas e explicações sucintas sobre cada item (Polop 2019). Apesar de sua utilidade prática e abertura a contribuições, o HackTricks não apresenta expressamente um fluxo sequencial de execução e possui inconsistências nas explicações conceituais quando compara-se verificações distintas. Por

exemplo, em alguns itens, são descritas explicitamente informações como qual é o objetivo da verificação em questão, enquanto em outros itens o padrão não se mantém. Assim, embora seja uma referência para consultas pontuais, sua lacuna conceitual e ausência de um guia ordenado limitam sua aplicação como metodologia didática ou como ferramenta de sistematização completa do processo de verificação.

O The Pentesting Guide é um recurso estruturado por tópicos, com destaque para a seção dedicada à elevação de privilégios em sistemas Linux (Marmeus 2023). Para cada subtema, o material apresenta explicações introdutórias e comandos práticos de verificação, além de sugerir ferramentas automatizadas auxiliares. Embora útil como referência técnica e disponível de forma aberta para contribuições, o guia também possui explicações inconsistentes, não propõe um fluxo sequencial de execução das checagens e apresenta ritmo de atualização mais esporádico. Esses fatores podem limitar seu uso como recurso de aprendizado ou como instrumento para a condução integral de verificações.

O recurso Linux Privilege Escalation consiste em um guia introdutório sobre técnicas comuns de elevação de privilégios em sistemas Linux (Mil0 2018). Apesar de sua utilidade para iniciantes e de sua organização em tópicos, o guia visa fornecer uma visão geral prática do tema, por isso aborda apenas um subconjunto das técnicas possíveis. Assim, o recurso demonstra lacunas em sua abrangência de assuntos, além da falta de um fluxo sequencial sugerido, não se propondo a funcionar como um arcabouço sistemático para avaliações completas.

G0tmi1k elenca em seu artefato Basic Linux Privilege Escalation uma coleção de perguntas associadas a instruções práticas de verificação (G0tmi1k 2011). Embora seja citado como referência por vários projetos análogos, o material não fornece explicações conceituais sobre os objetivos ou fundamentos de cada verificação, o que limita sua utilidade como recurso formativo.

A Seção 4 compara esses trabalhos/recursos de acordo com critérios qualitativos especificados na mesma seção.

3. Método de Pesquisa

A pesquisa está dividida em duas principais etapas: a especificação do arcabouço e a avaliação do arcabouço.

3.1. Especificação do arcabouço

Para descrever o arcabouço, empregou-se (i) uma revisão narrativa estruturada, na qual extraiu-se e comparou-se seis critérios a partir de quatro recursos de referência, e (ii) um processo de modelagem conceitual que gerou uma ontologia adaptada sobre técnicas de elevação de privilégio em sistemas Linux.

Os critérios para a especificação do arcabouço foram definidos por meio da análise dos recursos comumente usados para enumeração de vetores de elevação de privilégio. Foram identificadas características fundamentais e, com isso, agregadas para compor cada critério.

Na especificação, buscou-se atender o critério *organização tópica* por meio de uma modelagem orientada à ontologia. Nesta etapa, as técnicas de elevação de privilégio

foram categorizadas em classes e subclasses a fim de estruturá-las hierarquicamente e estabelecer uma terminologia comum.

Na etapa de modelagem conceitual, foi empregado um método incremental para a construção de ontologias adaptado de (Noy and McGuinness 2001). Assim, objetivou-se simplificar o processo e atenuar o rigor ontológico ao contexto aplicado.

3.2. Avaliação do arcabouço

A avaliação do arcabouço será conduzida por meio de um experimento com avaliadores de segurança iniciantes, divididos em dois grupos: um utilizando o arcabouço proposto e outro utilizando recursos diversos, porém fragmentados. Cada participante receberá um cenário em máquina virtual Linux contendo vetores de elevação de privilégios previamente definidos, devendo documentar os vetores identificados. A principal métrica analisada será a quantidade de vetores omitidos, complementada por dados sobre o tempo de execução e autopercepção de desempenho. Com isso, espera-se avaliar a eficácia tanto operacional quanto educativa do arcabouço. Esta etapa ainda não foi realizada.

4. Resultados Preliminares

A seguir são apresentados os critérios para a especificação do arcabouço:

A *Organização tópica*: apresentar estrutura hierárquica por tópicos.
B *Ordenação sequencial*: estabelecer uma sequência ordenada de etapas a ser seguida sistematicamente durante as verificações.
C *Explicações conceituais*: apresentar explicações conceituais em cada verificação que incluam o objetivo da verificação, os elementos de suas instruções práticas e o resultado esperado.
D *Instruções práticas*: descrever, para cada verificação, pelo menos um método para afirmar a presença ou ausência da vulnerabilidade que essa técnica explora.
E *Abertura*: ser aberto para que a comunidade possa contribuir.
F *Manutenção ativa*: receber atualizações frequentes à medida que novas técnicas de elevação de privilégio são identificadas.

A Tabela 1 compara cada recurso em termos de conformidade aos critérios (representados por letras), além de tornar evidente a carência de um arcabouço completo.

Tabela 1. Comparação de recursos sobre elevação de privilégios em termos de critérios qualitativos.

Recurso	A	B	C	D	E	F
HackTricks (Polop)	✓		✓*	✓	✓	✓
Pentesting Guide (Marmeus)	✓		✓*	✓		
Linux Priv. Esc. (Mil0)	✓		✓*	✓		
Basic Linux PE (G0tmi1k)	✓*			✓		

Nota: A marcação ✓ indica que o critério é atendido. A marcação ✓* indica que o critério é atendido parcialmente, ou seja, o recurso fornece conteúdo relacionado ao critério, mas de forma incompleta, superficial ou inconsistente em algum tópico.

Em certo nível, todos os recursos analisados compartilham a deficiência no critério explicações conceituais. Quando compara-se verificações distintas pertencentes

ao mesmo recurso, é possível constatar algumas inconsistências. Por exemplo, em alguns itens, há descrições como aquelas que explicam o objetivo da verificação ou apontam possíveis interpretações de seus resultados, enquanto em outros itens esse padrão não se repete. Isso ocorre, entre outros fatores, devido a falta de definição da abrangência desse critério durante a concepção do projeto.

Por outro lado, de maneira geral, nenhum destes recursos é completo em si mesmo. O que há são materiais complementares: HackTricks e The Pentesting Guide fornecem verificações práticas e algumas explicações conceituais, mas não estabelecem um fluxo claramente sequencial; sites de *cheatsheets* como o de G0tmi1k ou ferramentas automatizadas (que não foram mencionadas neste trabalho) oferecem comandos práticos sem a teoria. Por isso, avaliadores de segurança iniciantes muitas vezes precisam combinar vários desses recursos para cobrir os pontos ausentes em cada um.

Adicionalmente, foram obtidos resultados preliminares da modelagem conceitual. Os componentes básicos são classes, subclasses e propriedades. Todas as classes e subclasses possuem um conjunto de propriedades. Para cada propriedade há uma descrição textual que a define. A ideia desta modelagem conceitual é mapear classes e subclasses para técnicas de elevação de privilégio e variações dessas, respectivamente.

A seguir, é definido o conjunto de propriedades esperadas de uma classe pertencente à modelagem conceitual:

- `preConditions`: O que deve existir ou qual o estado de algum componente do sistema para que a técnica funcione.
- `exploitationVector`: Quais recursos ou mecanismos são explorados.
- `outcome`: Descreve a consequência final da técnica.
- `interactionLevel`: Indica quanto envolvimento manual do atacante é necessário para que a técnica seja bem-sucedida.
- `tools`: Ferramentas que identificam ou exploram essa técnica.

Além disso, foi possível definir uma versão preliminar da modelagem conceitual, a qual é apresentada no Apêndice A.

5. Considerações Finais

Uma revisão narrativa baseada em critérios indicou a ausência de um arcabouço integrado para testes de elevação de privilégio em sistemas Linux. Desse modo, os próximos passos desta pesquisa em desenvolvimento incluem a definição de um arcabouço de acordo com os critérios estabelecidos. Com isso, será possível iniciar os experimentos para testar a hipótese elaborada.

Referências

[Ahmed 2021] Ahmed, A. (2021). *Privilege Escalation Techniques: Learn the Art of Exploiting Windows and Linux Systems*. Packt Publishing, Birmingham.

[Elder et al. 2022] Elder, S., Zahan, N., Shu, R., Metro, M., Kozarev, V., Menzies, T., and Williams, L. (2022). Do i really need all this work to find vulnerabilities? an empirical case study comparing vulnerability detection techniques on a java application. Disponível em: https://arxiv.org/abs/2208.01595. Acesso em: 13 mar. 2025.

[G0tmi1k 2011] G0tmi1k (2011). Basic linux privilege escalation. Disponível em: https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/. Acesso em: 26 fev. 2025.

[Marmeus 2023] Marmeus (2023). Linux local privilege escalation. The Pentesting Guide. Disponível em: https://the-pentesting-guide.marmeus.com/local_privilege_escalation/linux/. Acesso em: 1 mar. 2025.

[Mil0 2018] Mil0 (2018). Linux privilege escalation. Disponível em: https://percussiveelbow.github.io/linux-privesc/. Acesso em: 25 fev. 2025.

[Noy and McGuinness 2001] Noy, N. F. and McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. Relatório Técnico KSL-01-05 and SMI-2001-0880, Stanford Knowledge Systems Laboratory and Stanford Medical Informatics, Stanford, CA. Disponível em: https://protege.stanford.edu/publications/ontology_development/ontology101.pdf. Acesso em: 29 abr. 2025.

[OISSG 2006] OISSG (2006). Information systems security assessment framework (ISSAF).

[Polop 2019] Polop, C. (2019). Linux privilege escalation. HackTricks. Disponível em: https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html. Acesso em: 25 fev. 2025.

[PTES Organization 2014] PTES Organization (2014). Penetration testing execution standard (PTES).

[Scarfone et al. 2008] Scarfone, K., Souppaya, M., Cody, A., and Orebaugh, A. (2008). Technical guide to information security testing and assessment. Technical Report NIST SP 800-115, National Institute of Standards and Technology, Gaithersburg, MD.

[The MITRE Corporation 2025] The MITRE Corporation (2025). MITRE ATT&CK Framework. Disponível em: https://attack.mitre.org/. Acesso em: 15 jan. 2025.

Apêndice A – Modelagem Conceitual

A construção desta modelagem conceitual sobre técnicas de elevação de privilégio em Linux orientada a ontologia seguiu o processo iterativo proposto por (Noy and McGuinness 2001) e realiza simplificações pertinentes.

Os componentes básicos desta modelagem conceitual são classes, subclasses e propriedades. Todas as classes e subclasses possuem um conjunto de propriedades. Para cada propriedade há uma descrição textual que a define. Esta modelagem mapeia classes e subclasses para técnicas de elevação de privilégio e variações dessas, respectivamente.

A seguir é apresentada a versão inicial de uma das classes modeladas, a saber, *SudoExploitation*. Essa, por sua vez, abrange três subclasses: *ShellEscaping*, *FeatureExploitation* e *EnvironmentVariableExploitation*.

Classe: SudoExploitation

Subclasse: ShellEscaping

`preConditions` Permitido o direito de execução com `sudo` de um programa escapável (ex.: `vim`, `less`, `find`, `python`), que deve ser capaz de invocar um shell e que não esteja restrito por opções seguras no arquivo `sudoers`.

`exploitationVector` Comando interno do programa usado para invocar um shell (ex.: `:!sh` no `vim`, `!sh` no `less` ou `os.system()` no `python`).
`outcome` Execução de um shell interativo com os privilégios que o `sudo` está configurado para conceder.
`interactionLevel` Alto. Envolve iniciar o programa via `sudo`, navegar até o modo de comando ou entrada interativa e executar manualmente o escape.
`tools` `GTFOBins`, `LinPEAS`, entre outras.

Subclasse: FeatureExploitation

`preConditions` Acesso `sudo` a programas cujas funcionalidades legítimas permitem ações privilegiadas. Além disso, o programa não pode estar restrito por opções seguras no arquivo `sudoers`.
`exploitationVector` Funções nativas de programas como `tee`, `tar`, `find`, `cp`, `cat`, `apache` que permitem, por exemplo, redirecionar saídas, ler arquivos protegidos ou sobrescrever arquivos do sistema.
`outcome` Múltiplos: leitura de arquivos protegidos (como `/etc/shadow`), escrita em arquivos (como `authorized_keys`) ou execução indireta de comandos com privilégio elevado.
`interactionLevel` Médio. Exige conhecimento da funcionalidade do comando e montagem adequada de entradas e saídas, mas não requer interação contínua com o programa.
`tools` `GTFOBins` (casos documentados), `LinPEAS` (detecta possíveis abusos de comandos permitidos), entre outras.

Subclasse: EnvironmentVariableExploitation

`preConditions` Permissão para criar ou modificar variáveis de ambiente. Em alguns casos, também é necessário ter permissão para compilar e/ou escrever arquivos `.so`.
`exploitationVector` Manipulação de variáveis de ambiente que alteram o comportamento do carregador dinâmico (`ld.so`) ou do interpretador da linguagem. Inclui o carregamento de bibliotecas maliciosas ou a redefinição do caminho de execução (`PATH`).
`outcome` Execução arbitrária de código com privilégios elevados, podendo resultar em um `root shell` ou na execução de comandos com permissões elevadas.
`interactionLevel` Média. Exige a preparação de um *payload* (por exemplo, compilar um arquivo `.so`) e a configuração adequada da variável de ambiente, mas a execução em si é simples.
`tools` `LinPEAS`, entre outras.