UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GUSTAVO ZANZIN GUERREIRO MARTINS

ESPECIFICAÇÃO DE ARCABOUÇO PARA TESTES DE ELEVAÇÃO DE PRIVILÉGIO EM SISTEMAS LINUX

CAMPO MOURÃO

2025

GUSTAVO ZANZIN GUERREIRO MARTINS

ESPECIFICAÇÃO DE ARCABOUÇO PARA TESTES DE ELEVAÇÃO DE PRIVILÉGIO EM SISTEMAS LINUX

Framework specification for privilege escalation testing in Linux systems

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.
Orientador(a): Prof. Dr. Rodrigo Campiolo

CAMPO MOURÃO

2025

GUSTAVO ZANZIN GUERREIRO MARTINS

ESPECIFICAÇÃO DE ARCABOUÇO PARA TESTES DE ELEVAÇÃO DE PRIVILÉGIO EM SISTEMAS LINUX

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 02 / dezembro / 2025

Rodrigo Campiolo
Prof. Dr.
Universidade Tecnológica Federal do Paraná

Luiz Arthur Feitosa dos Santos
Prof. Dr.
Universidade Tecnológica Federal do Paraná

Rogério Aparecido Gonçalves
Prof. Dr.
Universidade Tecnológica Federal do Paraná

CAMPO MOURÃO

2025

AGRADECIMENTOS

Agradeço ao meu orientador, professor Dr. Rodrigo Campiolo, pelas contribuições técnicas e pela orientação consistente ao longo de todo o desenvolvimento deste trabalho.

À minha família, em especial à minha mãe e à minha esposa, cujos nomes opto por preservar, registro minha profunda gratidão pelo apoio incondicional, pela compreensão nos momentos de maior dedicação acadêmica e pelo incentivo constante, sem os quais esta etapa não teria sido concluída.

Aos colegas de graduação, com quem compartilhei desafios, aprendizados, dores e conquistas, expresso meu sincero reconhecimento. Mais do que colegas de profissão, levo-os comigo como amigos que marcaram de forma significativa essa trajetória.

Por fim, agradeço a todos os docentes do Departamento Acadêmico de Computação da Universidade Tecnológica Federal do Paraná (UTFPR), Câmpus Campo Mourão, cujo trabalho ultrapassa a formação técnica e contribui de maneira decisiva para a transformação pessoal e profissional de seus alunos.

RESUMO

Este trabalho apresenta a especificação, a implementação e a avaliação preliminar de um arcabouço voltado a testes de elevação de privilégios em sistemas GNU/Linux, com foco no auxílio a profissionais iniciantes em segurança cibernética. A pesquisa parte da constatação de que, embora existam metodologias gerais para testes de intrusão, há uma carência de métodos padronizados especificamente para a etapa de elevação de privilégios, resultando no uso de recursos fragmentados e em possíveis omissões de vulnerabilidades. A metodologia adotada consistiu em uma revisão narrativa para levantar requisitos gerais e em uma modelagem conceitual inspirada em ontologias para organizar o conhecimento e estabelecer uma terminologia comum. Como resultado, foi estabelecida uma especificação estrutural para um arcabouço de testes de elevação de privilégio em GNU/Linux. Baseado nessa especificação, foi desenvolvido o LinPE Framework, um produto mínimo viável de código aberto que estrutura as verificações de segurança de forma hierárquica e sequencial, combinando instruções práticas com fundamentação teórica. A avaliação preliminar, realizada por meio de um experimento com profissionais iniciantes e envolvendo ambientes virtuais, indicou que a estrutura proposta favorece a compreensão dos mecanismos de ataque e a sistematização do processo, contribuindo tanto para a eficácia operacional quanto para a formação educacional de profissionais iniciantes.

Palavras-chave: administração de sistemas; teste de intrusão; cibersegurança.

ABSTRACT

This work presents the specification, implementation, and preliminary evaluation of a framework aimed at privilege escalation testing in GNU/Linux systems, with a focus on assisting entry-level cybersecurity professionals. The research starts from the observation that, although general methodologies for penetration testing exist, there is a lack of standardized methods specifically for the privilege escalation stage, resulting in the use of fragmented resources and potential omission of vulnerabilities. The adopted methodology consisted of a narrative review to gather general requirements and a conceptual modeling inspired by ontologies to organize knowledge and establish a common terminology. As a result, a structural specification for a GNU/Linux privilege escalation testing framework was established. Based on this specification, the LinPE Framework was developed, an open-source minimum viable product that structures security checks hierarchically and sequentially, combining practical instructions with theoretical foundation. The preliminary evaluation, conducted through an experiment with beginner professionals and involving virtual environments, indicated that the proposed structure favors the understanding of attack mechanisms and the systematization of the process, contributing both to operational effectiveness and to the educational formation of entry-level cybersecurity professionals.

Keywords: system administration; penetration testing; cybersecurity.

LISTA DE QUADROS

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

EUID	*Effective User Identifier*
GNU	GNU *GNU is Not Unix*
ISSAF	*Information Systems Security Assessment Framework*
LinPE	*GNU/Linux Privilege Escalation Testing Framework*
OTG	*OWASP Testing Guide*
OWASP	*Open Web Application Security Project*
PTES	*Penetration Testing Execution Standard*
RUID	*Real User Identifier*
SGID	*Set Group Identifier*
SSH	*Secure Shell*
SUID	*Set User Identifier*
UID	*User Identifier*
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1 INTRODUÇÃO

Testes de intrusão (também conhecidos como pentestes) são avaliações de segurança computacional em que especialistas em segurança cibernética emulam ameaças reais para identificar métodos que contornem as medidas de segurança de aplicações, de sistemas ou de redes (Scarfone *et al.*, 2008).

Diversos métodos padronizados e reconhecidos tanto pela comunidade de cibersegurança, quanto por organizações, orientam a condução de testes de intrusão. Entre os principais exemplos, destacam-se o *Information Systems Security Assessment Framework* (ISSAF) (OISSG, 2006), o *OWASP Testing Guide* (OTG) (OWASP Foundation, 2014), e o *Penetration Testing Execution Standard* (PTES) (PTES Organization, 2014).

Um método bem definido é primordial em qualquer atividade que exija resultados replicáveis (Frankland, 2009). Shanley e Johnstone (2015) afirmam que a metodologia é o alicerce de um teste de intrusão bem-sucedido. Segundo os autores, um procedimento precisamente especificado ocupa um papel crítico para a obtenção de resultados que possam ser verificados e estudados com o intuito de proteger dados, aplicações e infraestrutura. Sem uma metodologia ou um arcabouço estabelecido para conduzir um penteste, identificar vulnerabilidades pode se tornar difícil ou até prover falsa sensação de segurança (Shanley; Johnstone, 2015; Frankland, 2009). Essa constatação sugere que abordagens sistemáticas – baseadas em métodos – tendem a aumentar a confiabilidade dos resultados e reduzir a ocorrência de omissões de vulnerabilidades, quando comparadas a avaliações conduzidas de forma improvisada.

No contexto de sistemas computacionais, um penteste pode envolver uma etapa para averiguar a presença de vetores que permitam o ataque de elevação de privilégio. O ataque de 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 se traduz em diversas categorias de potenciais vetores (más configurações em permissões, programas executáveis, bibliotecas, tarefas agendadas, variáveis de ambiente, vulnerabilidades no núcleo, etc.), cada uma dividida 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 estruturais, entre outras limitações discutidas na Seção 2.5.

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 como resultado de uma revisão narrativa estruturada – ou seja, baseada em critérios qualitativos (Subseção 4.1.1) – que comparou os principais projetos sobre elevação de privilégios em GNU/Linux. Os resultados mostram que cada recurso carece de pelo menos um requisito fundamental, demonstrando essa lacuna e reforçando a necessidade de um novo arcabouço integrado.

A soma do fator diversidade de vetores que permitem a elevação e devem ser verificados em uma avaliação com o fator carência de um recurso didático que oriente profissionais durante testes de segurança pode resultar em omissões de vulnerabilidades durante um penteste, principalmente por parte de avaliadores iniciantes.

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.

Objetiva-se nesta pesquisa a especificação, o desenvolvimento de uma versão inicial e a avaliação preliminar de um arcabouço para averiguação de elevação de privilégios em sistemas GNU/Linux, com foco em profissionais iniciantes.

Têm-se como objetivos específicos:

- Especificar o arcabouço integrado, detalhando seus requisitos (estrutura hierárquica, fluxo, instruções, abertura) e sua estrutura.
- Implementar uma versão inicial de arcabouço como guia para a busca por vetores de elevação de privilégios em sistemas GNU/Linux.
- Avaliar o arcabouço implementado, por meio de um experimento com profissionais de segurança iniciantes, sob a ótica de sua estrutura especificada e de suas possíveis contribuições educacionais e operacionais.

Portanto, ao fornecer aos profissionais de segurança um arcabouço integrado, a contribuição esperada deste trabalho é dupla: (1) auxiliar na condução dos testes de elevação de

privilégios, reduzindo omissões de vulnerabilidades e aumentando a eficiência dos testes manuais, por meio da sistematização do processo; e (2) fomentar o desenvolvimento do raciocínio crítico dos avaliadores, promovendo a compreensão dos fundamentos teóricos por trás das ferramentas e técnicas utilizadas. Assim, além de servir como guia operacional, o arcabouço proposto também contribui para a formação técnica do profissional.

O trabalho é apresentado em cinco capítulos e dois apêndices. O Capítulo 2 apresenta o referencial teórico, abordando conceitos fundamentais de cibersegurança, testes de intrusão, ataques de elevação de privilégios, ontologias e trabalhos relacionados. O Capítulo 3 descreve os métodos adotados, detalhando o processo de especificação do arcabouço e o procedimento de avaliação preliminar da implementação de um arcabouço inicial. O Capítulo 4 expõe os resultados obtidos, contemplando a especificação do arcabouço, a implementação de um arcabouço mínimo e a análise dos dados da avaliação preliminar. Por fim, o Capítulo 5 apresenta as considerações finais, discutindo as contribuições do trabalho, suas limitações e possíveis direções para trabalhos futuros. Por fim, os apêndices reúnem materiais que complementam a pesquisa, incluindo as instruções e o questionário utilizados na avaliação preliminar, bem como exemplos de uso do arcabouço implementado.

2 REFERENCIAL TEÓRICO

Este capítulo aborda os fundamentos teóricos para melhor compreensão do trabalho. Inicialmente, trata-se sobre Cibersegurança e testes de intrusão, seguida por aspectos relacionados a elevação de privilégios. Na sequência, é feita uma introdução às ontologias e ao processo de sua construção. Por fim, são apresentados trabalhos relacionados que contextualizam o trabalho desenvolvido.

2.1 Cibersegurança

Cibersegurança está relacionada com a proteção de sistemas computacionais, redes de computadores, dispositivos e dados digitais de ameaças. Segundo Maziero (2019), a segurança de um sistema computacional está relacionada à garantia de propriedades fundamentais associadas às informações e recursos presentes no sistema. Dentre as propriedades destacam-se:

- Confidencialidade: Recursos do sistema podem ser acessados somente por usuários autorizados a isso.
- Integridade: Recursos do sistema podem ser alterados ou removidos somente por usuários autorizados a isso.
- Disponibilidade: Recursos devem estar disponíveis para os usuários que têm o direito de usá-los, a qualquer momento.
- Autenticidade: Garantia de que as entidades (usuários, dispositivos ou sistemas) envolvidas em uma interação são legítimas e possuem identidade verificada, assegurando que não há falsificação.
- Não-repúdio: Todas as ações no sistema são conhecidas e não podem ser escondidas ou negadas por seus autores; também é conhecida como irretratabilidade ou irrefutabilidade.

O ato de explorar alguma vulnerabilidade de modo a violar alguma propriedade de segurança em sistemas computacionais é chamado de ataque (Maziero, 2019). Incidentes de segurança cibernética demonstram os impactos que ataques bem-sucedidos podem causar em organizações de diversos setores. Em outubro de 2025, uma instituição brasileira de tecnologia financeira foi alvo[1] de um ataque que resultou no desvio de aproximadamente R$ 26 milhões (Temóteo, 2025). Incidentes como esse ilustram a magnitude de consequências que falhas de

[1] A vulnerabilidade explorada estaria, na verdade, em um sistema de um prestador de serviços terceirizado que intermedia operações financeiras para a empresa. Com isso, os atacantes conseguiram acesso a credenciais de contas, permitindo a realização de centenas de transações fraudulentas via o sistema de pagamentos instantâneos.

segurança em sistemas computacionais podem acarretar e evidenciam a importância de práticas relacionadas à cibersegurança em organizações.

Para enfrentar esse cenário, existem organizações que estruturam suas equipes de segurança em abordagens especializadas de defesa e ataque. As equipes defensivas, conhecidas como *blue teams*, exercem atividades como monitoramento, detecção e resposta a incidentes. Enquanto as equipes ofensivas, ou *red teams*, simulam o comportamento de agentes maliciosos em busca de vulnerabilidades que poderiam ser exploradas. Equipes ofensivas, em suas operações, adotam táticas para avaliar aspectos como a segurança física do alvo, o preparo de pessoas envolvidas e a segurança de sistemas computacionais (esse último, por meio de testes de intrusão).

2.2 Testes de intrusão

Os testes de penetração ou testes de intrusão (também referidos como *pentest*, forma reduzida do termo inglês *penetration testing*, ou como penteste, forma adaptada ao português) são avaliações de segurança de sistemas computacionais ou de redes de computadores. O objetivo dos testes é que o avaliador simule o comportamento de ameaças reais de modo que encontre-se a maior quantidade possível de métodos para contornar as medidas de segurança do alvo (Scarfone *et al.*, 2008). Dessa forma, os mantenedores do sistema alvo têm a oportunidade de melhorar sua segurança antes que uma ameaça real explore esses pontos fracos.

Existem diversos métodos que visam orientar a condução de testes de intrusão. São métodos padronizados e reconhecidos tanto pela comunidade de cibersegurança, quanto por organizações. Alguns exemplos incluem o *Information Systems Security Assessment Framework* (ISSAF) (OISSG, 2006), o *OWASP Testing Guide* (OTG) (OWASP Foundation, 2014), e o *Penetration Testing Execution Standard* (PTES) (PTES Organization, 2014).

Porém, esses métodos tendem a ser generalistas, de modo a guiar as avaliações de maneira ampla. Em seu estudo, Happe, Kaplan e Cito (2025) avaliaram a eficiência de modelos de linguagem de grande escala na realização de ataques de elevação de privilégio. Em determinada etapa da pesquisa, os pesquisadores constataram que, embora haja métodos padronizados para orientar testes de intrusão, não há um método semelhante projetado especificamente para guiar testes de elevação de privilégio.

2.3 Ataques de elevação de privilégios

No contexto de sistemas computacionais, elevação de privilégio refere-se ao processo por meio do qual um atacante modifica os privilégios associados a uma conta de usuário, assumindo os privilégios de outro usuário – frequentemente com permissões mais elevadas, como as de administrador – ao realizar uma ação em um sistema computacional ou rede de computadores (Ahmed, 2021).

O sucesso em um ataque de elevação de privilégio não representa, por si só, o objetivo final do atacante, mas sim um meio de adquirir controle ainda maior sobre o sistema comprometido. Ao obter privilégios elevados, o invasor é capaz de comprometer quaisquer propriedades da segurança da informação supracitadas, conforme seu intuito: pode violar a confidencialidade ao acessar dados sensíveis, comprometer a integridade ao modificar configurações ou arquivos, e eliminar a disponibilidade por meio da desativação de serviços, por exemplo. Além disso, com privilégios administrativos, torna-se possível desativar ferramentas de defesa, ampliar o acesso a outros sistemas da rede e manter persistência no ambiente atacado. Por essas razões, a elevação de privilégios é considerada uma etapa de alto impacto em cadeias de ataque, como as descritas na matriz MITRE ATT&CK (The MITRE Corporation, 2025).

Os ataques de elevação de privilégio estão relacionados a contas de usuários e suas permissões. Com base nisso, segundo Ahmed (2021), a partir da execução bem-sucedida de ataques de elevação de privilégio, seus resultados são categorizados em dois tipos principais: horizontal e vertical.

2.3.1 Elevação de privilégio horizontal

Ataque de elevação de privilégio horizontal é o processo de exploração de vulnerabilidades ou de más configurações em sistemas, serviços ou utilitários que permite que atacantes realizem ações em sistemas usando privilégio de outras contas de usuários que não possuem permissões administrativas. Nesse sentido, geralmente, tais contas de usuários detém níveis de permissões semelhantes em relação a conta de usuário já comprometida no momento do acesso inicial.

No entanto, existem casos em que o usuário alvo da elevação horizontal possui privilégio superior em algum contexto específico, como executar um utilitário do sistema de forma privilegiada, por exemplo. Desse modo, o atacante pode, primeiro, realizar uma elevação horizontal e, na sequência, realizar um novo ataque para obter acesso ao usuário administrativo. Ou seja, também é possível utilizar a elevação de privilégio horizontal como um meio para alcançar o controle irrestrito.

2.3.2 Elevação de privilégio vertical

Ataque de elevação de privilégio vertical é o processo de exploração de vulnerabilidades ou de más configurações em sistemas, serviços ou utilitários que possibilita que atacantes realizem ações em sistemas usando privilégio administrativo indevidamente. A elevação de privilégio vertical é o resultado mais visado por atacantes, já que provê acesso e controle total sobre o sistema alvo.

Para melhor compreensão do funcionamento desse ataque, considere o seguinte exemplo de ataque de elevação de privilégio vertical que envolve uma má configuração de permissões de arquivos.

No contexto de controle de acesso em sistemas Unix-like, Set User Identifier (SUID) e Set Group Identifier (SGID) são bits que representam permissões especiais. Em sistemas Unix-like, cada processo possui dois identificadores de usuário principais que influenciam diretamente seu comportamento: o Real User Identifier (RUID) e o Effective User Identifier (EUID). O RUID representa o usuário que iniciou o processo, enquanto o EUID determina quais permissões o processo possui durante sua execução. Quando um programa possui o bit SUID ativado, seu EUID é alterado para o User Identifier (UID) do proprietário do arquivo, permitindo que ele seja executado com privilégios diferentes – e, potencialmente, mais elevados – do que o usuário que o executou.

Em um sistema GNU/Linux, considere o interpretador do Python com o bit SUID ativado e pertencente ao usuário administrativo `root`. Quando um usuário comum executa esse interpretador, o processo resultante assume o EUID do proprietário do binário – ou seja, zero (do `root`). Um usuário comum pode executar um código como `import os; os.system("/bin/sh")`, o qual faz com que o interpretador inicie um processo filho (um shell). Como o processo filho herda o EUID do processo pai, ele será iniciado com privilégios administrativos também. Dessa forma, ao inserir o comando `id` no terminal recém-criado, constata-se que a sessão foi iniciada com o usuário `root`, evidenciando a elevação de privilégio vertical.

2.4 Ontologias e sua construção

Em áreas técnicas, onde o conhecimento está distribuído em fontes heterogêneas e, com frequência, despadronizadas, a ausência de uma terminologia comum pode comprometer a comunicação entre especialistas. Ontologias visam resolver problemas como esse. Ontologias são representações explícitas de uma conceitualização compartilhada sobre determinado domínio (Gruber, 1993). Uschold e Gruninger (1996) afirmam que “esse tipo de entendimento [compartilhado] pode funcionar como um arcabouço unificador dos diferentes pontos de vista”. Uma das principais funções de uma ontologia está na sua capacidade de tornar o conhecimento mais acessível, reutilizável e interoperável, servindo tanto para a construção de sistemas computacionais quanto para o alinhamento conceitual entre pessoas ou organizações (Uschold; Gruninger, 1996).

As ontologias são compostas por um conjunto de elementos que permitem representar formalmente o conhecimento de um domínio. O elemento central é a classe, que descreve um conceito geral do domínio; classes podem ser organizadas hierarquicamente por meio de subclasses, que especializam conceitos mais amplos. As propriedades descrevem características das classes ou relações entre elas, podendo indicar atributos internos ou vínculos com outros conceitos. Já as instâncias representam ocorrências concretas das classes, isto é, elementos específicos que materializam os conceitos abstratos definidos na ontologia. Além desses componentes principais, ontologias podem incluir outros elementos, como restrições e axiomas, utilizados para expressar regras lógicas, enriquecer a representação do domínio e assim por

diante. Em conjunto, esses componentes permitem estruturar o domínio de forma explícita e extensível, favorecendo a organização, o compartilhamento e a reutilização do conhecimento.

Noy e McGuinness (2001) propõem um método incremental composto por sete etapas para a construção de ontologias. O método inicia-se com a definição do domínio e do escopo da ontologia, seguida pela reutilização de ontologias existentes quando possível. Em seguida, procede-se à listagem dos termos relevantes para o domínio, organização desses termos em uma hierarquia de classes e definição de propriedades (atributos e relações) para as classes identificadas. Por fim, são determinadas as restrições de valor para as propriedades, caso necessário, e criadas instâncias das classes.

Diante do exposto, observa-se que ontologias são instrumentos adequados para organizar domínios caracterizados por conhecimento disperso, heterogêneo e pouco padronizado, como é o caso de assuntos relacionados à elevação de privilégios em sistemas GNU/Linux. Ao fornecer uma conceitualização explícita e compartilhada, ontologias permitem reduzir ambiguidades terminológicas, o que favorece a reutilização e evolução do conhecimento. Nesse contexto, a adoção de métodos inspirados em ontologias neste trabalho justifica-se como meio de estruturar os mecanismos, condições e técnicas envolvidas em ataques de elevação de privilégio, servindo de base conceitual para a especificação do arcabouço proposto.

2.5 Trabalhos relacionados

Existem várias metodologias reconhecidas por organizações que visam orientar profissionais durante a operação de testes de intrusão (como ISSAF, OTG e PTES), mas não detalham especificamente checagens de elevação de privilégio em GNU/Linux (Happe; Kaplan; Cito, 2025). Por outro lado, há uma diversidade de recursos que têm em vista a enumeração de possíveis vetores desse ataque. Alguns desses são apresentados nesta seção.

O *HackTricks* (especificamente sua seção *Linux Privilege Escalation*) organiza sua estrutura por tópicos (como “System Information”, “Drives”, “Processes”, entre outros) que funcionam como uma lista de verificação, oferecendo comandos para orientações práticas e explicações sucintas sobre cada item (Polop, 2019). Apesar de sua utilidade prática, reconhecimento na comunidade e abertura a contribuições, o *HackTricks* não apresenta explicitamente um fluxo sequencial de execução e apresenta inconsistências estruturais entre suas seções: resultado da ausência de padronização dos elementos que deveriam compor suas seções. Essa última limitação reflete principalmente no aspecto explicações conceituais, utilizado, em geral, por iniciantes. Assim, embora seja uma referência para explorações específicas, suas lacunas estrutural e conceitual e ausência de um fluxo sequencial limitam sua aplicação como guia didático ou como instrumento de sistematização do processo de verificação.

O *The Pentesting Guide* é um recurso organizado por assuntos, com destaque para a seção dedicada à elevação de privilégios em sistemas GNU/Linux (Marmeus, 2023). Para cada tema, 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 estrutura inconsistente, 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 GNU/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 (2011) elenca em seu artefato *Basic Linux Privilege Escalation* uma coleção de perguntas associadas a instruções práticas de verificação. 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 Subseção 4.1.1 compara esses trabalhos/recursos em termos de critérios qualitativos especificados na mesma subseção.

Em síntese, embora os recursos apresentados forneçam contribuições relevantes para a identificação de vetores de elevação de privilégio em sistemas GNU/Linux, observa-se que eles, em sua maioria, se concentram em abordagens predominantemente operacionais e carecem de padronização dos componentes de suas seções, resultando em inconsistências estruturais entre as diferentes seções do mesmo material. Isso dificulta o uso desses recursos como instrumentos didáticos, especialmente por avaliadores iniciantes. Diante desse cenário, o presente trabalho diferencia-se ao propor um arcabouço de apoio a testes de elevação de privilégio em GNU/Linux com estrutura padronizada. O arcabouço busca não apenas apoiar a condução operacional dos testes, mas também favorecer a compreensão pelo avaliador da necessidade de cada verificação, atendendo simultaneamente às perspectivas operacional e educacional.

3 MÉTODOS

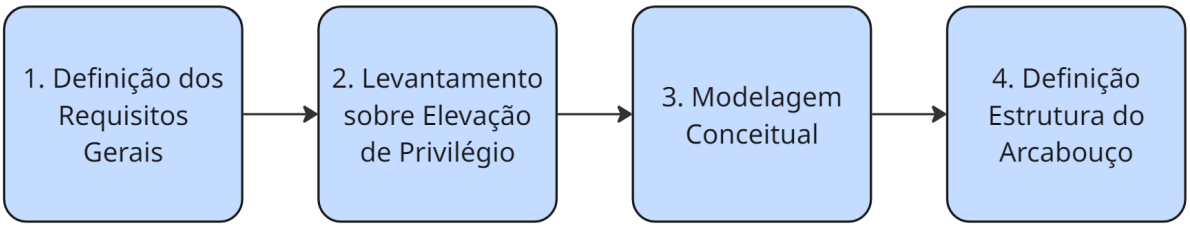
A pesquisa está dividida em três etapas principais: a especificação do arcabouço, a implementação inicial de um exemplar de arcabouço e a avaliação preliminar dessa implementação.

3.1 Especificação do arcabouço

Para especificar o arcabouço, realizou-se: (i) uma revisão narrativa na qual, a partir de recursos de referência, extraíram-se critérios qualitativos que originaram os requisitos gerais do arcabouço; (ii) um levantamento sobre técnicas de elevação de privilégio em sistemas GNU/-Linux, (iii) um processo de modelagem conceitual que gerou uma ontologia adaptada sobre técnicas de elevação de privilégio em sistemas GNU/Linux e (iv) uma definição da estrutura do arcabouço em nível mais granular.

O resumo do processo de quatro etapas para a especificação do arcabouço apresentado no parágrafo anterior pode ser observado na Figura 1.

Figura 1 – Processo de especificação do arcabouço.



Fonte: Autoria própria (2025).

3.1.1 Definição dos requisitos gerais

Os seis requisitos gerais que norteiam o arcabouço proposto são: organização tópica, ordenação sequencial, explicações conceituais, instruções práticas, abertura e manutenção ativa. Esses requisistos foram definidos por meio da análise de quatro recursos comumente usados para enumeração de vetores de elevação de privilégio. Tais recursos são elencados na Seção 2.5 sobre trabalhos relacionados: *HackTricks* (Polop, 2019), *The Pentesting Guide* (Marmeus, 2023), *Linux Privilege Escalation* (Mil0, 2018) e *Basic Linux Privilege Escalation* (G0tmi1k, 2011). Foram identificadas características fundamentais em cada um e, então, agregadas para compor os requisitos gerais.

Especificamente esses quatro recursos foram selecionados porque, além de serem amplamente conhecidos pela comunidade, são os que apresentam as características mais semelhantes ao arcabouço proposto por este trabalho: orientam o processo de enumeração de

informações sobre sistemas GNU/Linux com foco em vetores de elevação de privilégio, e não são de cunho totalmente operacional, pois apresentam pontualmente aspectos educacionais.

Em seguida, os quatro recursos mencionados foram comparados do ponto de vista de alguns critérios. Esses critérios são os mesmos seis requisitos gerais que norteiam o arcabouço proposto. Nessa comparação, observou-se, para cada recurso, sua conformidade com cada critério. Os resultados dessa comparação são apresentados na Tabela 1 e discutidos na Subseção 4.1.1.

3.1.2 Levantamento sobre técnicas de elevação de privilégio

Com base em informações apresentadas em fontes não acadêmicas (incluindo os trabalhos relacionados), realizou-se um levantamento preliminar sobre técnicas de elevação de privilégio em sistemas GNU/Linux.

O levantamento consistiu em catalogar e agregar informações sobre o tema. Para cada técnica foram elaborados: (i) descrição conceitual sobre o funcionamento da técnica de elevação de privilégio; (ii) descrição de um ambiente vulnerável a essa técnica de exploração; (iii) instruções práticas para identificação da vulnerabilidade que permite o ataque; e (iv) instruções práticas para a exploração da vulnerabilidade em questão.

Os materiais das fontes não acadêmicas utilizados para o levantamento incluem artigos, postagens em *blogs* pessoais, módulos em plataformas de treinamento em cibersegurança e assim por diante. Todas as fontes utilizadas durante o levantamento são citadas nas seções de referências do arcabouço mínimo implementado[1].

3.1.3 Modelagem conceitual

No início do desenvolvimento do arcabouço, buscou-se atender os critérios de organização tópica, explicações conceituais e capacidade de evolução. Para isso, adotou-se um método para modelagem conceitual orientado a ontologia. Conforme Noy e McGuinness (2001), ontologias visam compartilhar entendimento comum e viabilizar a reutilização de conhecimento de domínio. Por isso, a fim de reduzir a ambiguidade, definiu-se uma terminologia comum sobre técnicas de elevação de privilégios em GNU/Linux, por meio de classes e suas propriedades. A modelagem conceitual organiza os assuntos de forma coerente e hierárquica. Cada classe e subclasse corresponde a seções do arcabouço, como *Contexto Privilegiado* e *Permissões Especiais: Bits SUID e SGID*. A modelagem conceitual permite sua extensão quando surgem novos vetores, promovendo a manutenção contínua do arcabouço.

Para realizar a modelagem conceitual, foi empregado um método incremental para a construção de ontologias adaptado de Noy e McGuinness (2001) (o método original é brevemente apresentado na Seção 2.4). O processo adaptado inicia-se com a definição do domínio e

[1] Disponível no GitHub: https://github.com/cybersec-utfpr-cm/linpe-framework

do escopo da modelagem conceitual, seguida pela busca de ontologias existentes para reuso. Na sequência, lista-se os termos importantes do domínio e organiza-se tais termos hierarquicamente através de classes. Por fim, definiu-se as propriedades das classes.

Com a adaptação, objetivou-se simplificar o processo de formação do modelo e atenuar o rigor ontológico ao contexto aplicado. As etapas do procedimento adaptado resultam nos objetivos pretendidos pela modelagem conceitual (expostos no primeiro parágrafo desta subseção). Desse modo, por serem suficientes para o objetivo dessa modelagem conceitual, realizaram-se as adaptações que simplificam o procedimento de construção de ontologias proposto pelas autoras.

Na Subseção 4.1.3 apresenta-se uma visão geral da modelagem conceitual obtida. A modelagem conceitual alcançada está disponível no repositório[1] do projeto.

3.1.4 Definição da estrutura do arcabouço

A estrutura mais granular do arcabouço foi especificada com base: (i) na estrutura de alto nível (de abstração) fornecida pela modelagem conceitual; (ii) na tentativa de atenuação de problemas identificados nos trabalhos relacionados; e (iii) nos requisitos gerais.

Em relação a modelagem conceitual, as duas classes do topo da hierarquia (*Contexto Privilegiado* e *Técnicas de Elevação de Privilégio Específicas*) influenciaram o estabelecimento de dois agrupamentos para abranger suas respectivas subclasses.

Após comparar-se os recursos elencados como trabalhos relacionados, discute-se um problema presente em alguns deles: a ausência de padronização dos elementos que deveriam compor suas seções, o que resulta em inconsistências estruturais entre as diferentes seções do mesmo material.

Com o objetivo de atenuar inconsistências como essas e de atender aos requisitos gerais (especialmente explicações conceituais, ordenação sequencial e instruções práticas), definiu-se uma estrutura padronizada para os componentes do arcabouço. Assim, torna-se necessário que o arcabouço forneça, para cada classe, (i) uma seção para explicações conceituais sobre o tema em questão e (ii) um conjunto de instruções práticas (método), ordenadas de modo que possam ser seguidas sequencialmente, visando guiar as verificações de segurança. Dada a perspectiva educacional do arcabouço, cada etapa de verificação deveria apresentar o que seria verificado (objetivo) e por que deveria ser verificado (motivo).

Além disso, pensando na perspectiva educacional que o arcabouço busca fornecer, definiu-se uma seção sobre pré-requisitos relacionados aos assuntos envolvidos em determinada classe para que o leitor do arcabouço possa informar-se previamente ou aprofundar-se no estudo desse tema.

Desse modo, a estrutura em nível mais granular do arcabouço proposto é apresentada na Subseção 4.1.4.

[1] https://github.com/cybersec-utfpr-cm/linpe-framework/tree/master/thesis/conceptual-modeling

3.2 Avaliação preliminar do arcabouço

A Seção 4.2 apresenta o GNU/Linux Privilege Escalation Testing Framework (LinPE Framework): um arcabouço mínimo implementado de acordo com a especificação proposta por este trabalho. A avaliação preliminar teve como objetivo analisar o arcabouço sob duas perspectivas principais: (i) sua contribuição operacional, ou seja, sua capacidade de orientar iniciantes na condução de avaliações de segurança em sistemas GNU/Linux; e (ii) sua contribuição educacional, considerando se a estrutura especificada favorece a compreensão dos mecanismos envolvidos e das configurações que podem viabilizar ataques de elevação de privilégio.

Para essa avaliação preliminar, adotou-se um procedimento envolvendo quatro participantes iniciantes em cibersegurança (que nunca trabalharam formalmente no setor, mas estudam sobre há aproximadamente um ou dois anos), portanto, perfis de participantes alinhados ao público-alvo do arcabouço.

O experimento foi conduzido por meio de três ambientes virtuais construídos com Docker, cada um simulando um sistema GNU/Linux distinto. Dois deles continham configurações inseguras que permitiam ataques de elevação de privilégio, enquanto o terceiro não possuía vulnerabilidades inseridas intencionalmente, permitindo observar eventuais falsos positivos.

A tarefa solicitada aos participantes consistiu em utilizar exclusivamente o arcabouço LinPE para identificar se cada ambiente apresentava ou não condições que permitissem ataques de elevação de privilégio. O uso de materiais externos relacionados a técnicas de elevação de privilégio ou de ferramentas automatizadas de enumeração foi explicitamente proibido, sendo permitido apenas o uso de consultas para esclarecer dúvidas periféricas, por exemplo dúvidas sobre sintaxe de comandos do sistema.

Além das instruções sobre a tarefa principal do experimento, os participantes receberam um diretório compactado. Nele, haviam os arquivos Docker com os ambientes previamente configurados (o acesso a tais arquivos foi explicitamente proibido). No diretório também havia um documento com instruções adicionais sobre: como criar as imagens e contêineres Docker, e como acessar os ambientes virtuais via Secure Shell (SSH).

Após instanciar os contêineres, acessar o arcabouço LinPE e realizar a tarefa, cada participante respondeu a um questionário via Formulários Google. O questionário foi composto por 11 questões: três quantitaitvas, voltadas à mensuração da quantidade de vetores identificados e tempo utilizado; e oito qualitativas, sendo cinco elaboradas em escala de Likert e três abertas. De modo geral, essas questões buscaram avaliar a estrutura especificada, inteligibilidade do conteúdo, utilidade operacional e percepção subjetiva sobre o arcabouço. As instruções fornecidas aos participantes e o questionário completo utilizado na avaliação preliminar encontram-se no Apêndice A.

O procedimento de condução da avaliação seguiu um fluxo simples: disponibilização aos participantes das instruções para a tarefa junto com o questionário pelo formulário; disponibilização aos participantes, em um diretório compactado, dos arquivos Docker com os ambientes previamente configurados e suas instruções de inicialização; execução individual das avaliações

por cada participante, conforme as orientações estabelecidas; preenchimento do questionário; e análise das respostas ao questionário. Não houve intervenção ou assistência durante a execução da tarefa.

Para análise dos dados, foram considerados critérios quantitativos e qualitativos. Do ponto de vista quantitativo, avaliou-se a quantidade de vetores corretamente identificados por cenário, o tempo utilizado e a presença de falsos positivos ou negativos. No âmbito qualitativo, analisaram-se os comentários abertos apresentados pelos participantes e as pontuações obtidas nas escalas de Likert. A partir desses elementos, buscou-se identificar padrões recorrentes que indicassem problemas ou aspectos positivos do arcabouço.

4 RESULTADOS

Os resultados desta pesquisa são apresentados em três fragmentos principais: especificação do arcabouço, implementação inicial de um exemplar de arcabouço e os resultados da avaliação preliminar dessa implementação.

4.1 Especificação do arcabouço

Os resultados desta etapa incluem: (i) requisitos gerais que norteiam o arcabouço originados a partir da revisão narrativa dos trabalhos relacionados e sua comparação; (ii) base de conhecimento preliminar advinda de levantamento sobre técnicas de elevação de privilégio em sistemas GNU/Linux; (iii) modelagem conceitual sobre o tema e inspirada em ontologias; (iv) estabelecimento de uma estrutura para o arcabouço proposto.

4.1.1 Definição dos requisitos gerais e comparação de recursos

Apresentam-se as definições dos requisitos gerais que orientam o arcabouço, seguidas de uma comparação entre recursos comumente utilizados para a enumeração de informações em sistemas GNU/Linux, com foco em vetores de elevação de privilégio. Por fim, discute-se essa comparação com base nos requisitos definidos.

A seguir são apresentados os seis requisitos gerais que orientam o arcabouço proposto:

A *Organização tópica*: apresentar estrutura hierárquica por assuntos.

B *Ordenação sequencial*: estabelecer sequências ordenadas de etapas de verificação a serem seguidas sistematicamente.

C *Explicações conceituais*: apresentar uma descrição conceitual sobre o funcionamento do assunto abordado; e apresentar, para cada etapa de verificação, descrições conceituais que incluam aspectos como o seu objetivo e o seu motivo.

D *Instruções práticas*: descrever, para cada assunto, pelo menos um método para identificar as vulnerabilidades que essa técnica explora; ou, quando isso for inviável, apresentar, para cada assunto, aspectos com potencial de risco em relação ao que pode ser descoberto durante a verificação.

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). Os critérios são os mesmos seis requisitos gerais norteadores do arcabouço proposto. Com essa comparação, evidencia-se 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 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: em alguns tópicos, elementos como os objetivos e/ou motivos das verificações são descritos, mas em outros tópicos, tais elementos não são apresentados. Isso ocorre devido a falta (no desenho do projeto) da definição de uma estrutura a ser usada ao longo do desenvolvimento do projeto de maneira padronizada.

Por outro lado, avaliando o cenário completo, 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 (2011) ou ferramentas automatizadas (que não foram mencionadas neste trabalho) oferecem comandos práticos sem a teoria; recursos como de G0tmi1k (2011) e de Mil0 (2018) são publicações estáticas em *sites* pessoais, o que impossibilita sua manutenção. Por isso, avaliadores de segurança iniciantes muitas vezes precisam combinar vários desses recursos para cobrir os pontos ausentes em cada um.

4.1.2 Levantamento sobre técnicas de elevação de privilégio

Durante o levantamento sobre técnicas de elevação de privilégio em sistemas GNU/Linux, consultaram-se aproximadamente 20 fontes não acadêmicas. Realizando os procedimentos apresentados na Subseção 3.1.2, foi catalogada em uma base de conhecimento preliminar 9 assuntos relacionados à elevação de privilégio em sistemas GNU/Linux.

Os assuntos agregados no levantamento incluem: exploração do núcleo Linux; mecanismo de controle de acesso Sudo; permissões especiais SUID e SGID; injeção de comandos em binários privilegiados; exploração de variáveis de ambiente em contexto privilegiado (em especial, as variáveis `PATH` e `LD_LIBRARY_PATH`); exploração de *links* simbólicos em contexto privilegiado; manipulação de *glob* (expansão de coringas) em contexto privilegiado; e exploração de arquivos graváveis em contexto privilegiado.

É importante destacar que, devido a escassez de literatura branca e a dispersão do conhecimento a cerca do tema por fontes não acadêmicas (onde não são estabelecidos padrões conceituais e/ou terminológicos), o levantamento produz uma lista não exaustiva. Logo, sua

abrangência atual não deve ser tratada como uma linha de chegada, mas sim como um ponto de partida.

4.1.3 Modelagem conceitual

Adicionalmente, foram obtidos resultados referentes a modelagem conceitual. Os componentes básicos que compõem a modelagem 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. Dessa forma, cada classe e subclasse é definida como um conceito por meio de suas propriedades, originando uma terminologia comum sobre técnicas de elevação de privilégio em sistemas GNU/Linux.

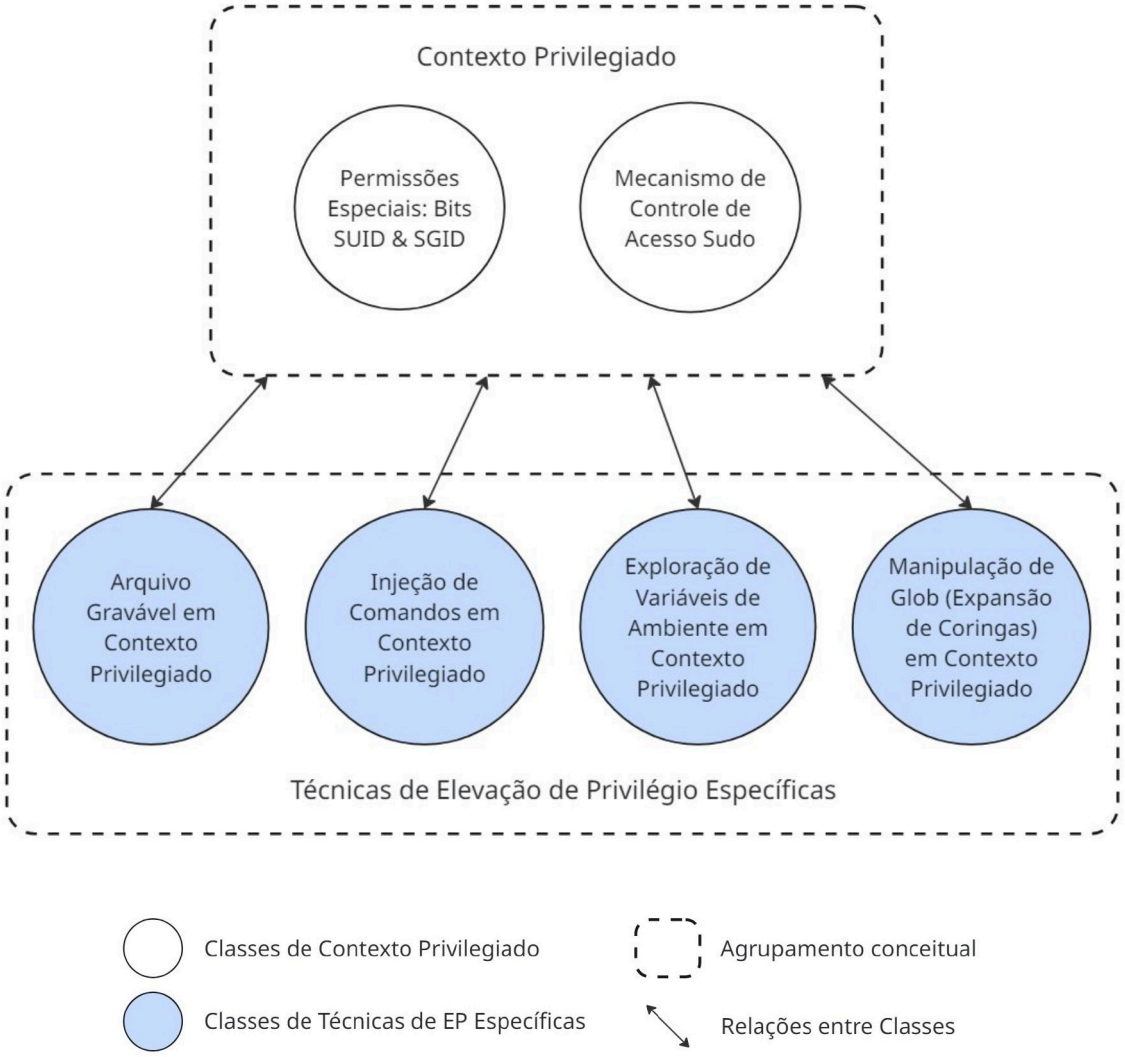
O parágrafo anterior apresentou os componentes da modelagem conceitual. Este parágrafo, sobre a ausência de um componente da modelagem conceitual, faz uma breve tangência ao raciocínio desta subseção para mencionar um aspecto positivo da modelagem conceitual: um possível caso de uso. Um dos componentes comuns em ontologias são as instâncias (que, a grosso modo, são indivíduos pertencentes a uma classe, com valores específicos de propriedades). A modelagem conceitual elaborada por este trabalho não abrange instâncias. Porém, com a modelagem de classes estabelecida, é possível utilizá-la para classificar exemplares de ataques de elevação de privilégio em GNU/Linux como uma instância de determinada classe.

A modelagem conceitual categorizou o tema em duas classes principais: *Contexto Privilegiado* e *Técnicas de Elevação de Privilégio Específicas*. Essas duas classes ocupam o topo da hierarquia e cumprem função de agrupadoras conceituais. Assim, elas são meramente um conjunto que contém suas subclasses e não podem possuir instâncias associadas diretamente a elas. Em analogia a programação orientada a objetos, elas seriam como classes abstratas. Por outro lado, continuando com a analogia, suas subclasses seriam classes concretas e portanto instanciáveis.

Na Figura 2 é possível observar uma representação em diagrama da modelagem conceitual elaborada por este trabalho e sua respectiva legenda. Além das classes conceituais já apresentadas, também foram catalogadas algumas de suas subclasses. As subclasses para a classe *Contexto Privilegiado* são: *Permissões Especiais: Bits SUID e SGID* e *Mecanismo de Controle de Acesso Sudo*. As subclasses para a classe *Técnicas de Elevação de Privilégio Específicas* são: *Arquivo Gravável em Contexto Privilegiado*, *Injeção de Comandos em Contexto Privilegiado*, *Exploração de Variáveis de Ambiente em Contexto Privilegiado* e *Manipulação de Glob (Expansão de Coringas) em Contexto Privilegiado*.

Assim como salientado anteriormente em relação ao levantamento, é importante destacar aqui também que, devido a (i) escassez de literatura branca a cerca do tema; (ii) a dispersão do conhecimento a cerca do tema por fontes não acadêmicas (onde não são estabelecidos padrões conceituais e/ou terminológicos); e (iii) a limitações de tempo; a modelagem conceitual alcançada não produz um apanhado exaustivo. Logo, sua abrangência atual não deve ser tratada como uma linha de chegada, mas sim como um ponto de partida.

Figura 2 – Representação em diagrama da modelagem conceitual sobre elevação de privilégio em sistemas GNU/Linux



Fonte: Autoria própria (2025).

O Quadro 1 apresenta o conjunto de propriedades definido para as subclasses pertencentes à classe *Contexto Privilegiado*. O Quadro 2 apresenta o conjunto de propriedades definido para as subclasses pertencentes à classe *Técnicas de Elevação de Privilégio Específicas*.

Quadro 1 – Propriedades que compõem a classe *Contexto Privilegiado*

Propriedade	**Descrição**	**Exemplos**
Condições de risco	Quais configurações ou mecanismos do sistema que, quando presentes, sinalizam risco de serem aproveitados em ataques de elevação de privilégio.	Binário configurado com `sudo` para permitir execução sem senha (opção `NOPASSWD`).
Condições em comum	Este mecanismo é condição necessária para duas ou mais técnicas de elevação de privilégio específicas.	Permissões especiais SUID e SGID; Mecanismo de controle de acesso Sudo.

Fonte: Autoria própria (2025).

Quadro 2 – Propriedades que compõem a classe *Técnicas de Elevação de Privilégio Específicas*

Propriedade	Descrição	Exemplos
Condições de risco	Quais configurações ou mecanismos do sistema que, quando presentes, sinalizam risco de serem aproveitados em ataques de elevação de privilégio.	Binários em contexto privilegiado que chamam outros comandos passando a eles argumentos que contêm coringas (como o asterisco).
Vetor de exploração	Quais recursos ou mecanismos do sistema são explorados nessa técnica de elevação de privilégio específica.	Expansão de coringas (*globbing*); Variável de ambiente `LD_PRELOAD`.

Fonte: Autoria própria (2025).

Durante a elaboração do levantamento e da modelagem conceitual foram identificadas algumas classes que mostravam-se presentes em múltiplos processos de identificação de vulnerabilidades que permitem o ataque de elevação de privilégio. Identificou-se, portanto, etapas de verificação comuns entre processos de verificação distintos. Até onde a modelagem conceitual deste trabalho alcançou, as chamadas *Técnicas de Elevação de Privilégio Específicas* necessitam de aspectos relacionados a tais etapas de verificação comuns. A modelagem conceitual representa essas etapas de verificação comuns nas subclasses de *Contexto Privilegiado* por meio da propriedade *Condições em comum*.

Além disso, a modelagem conceitual alcançada está disponível no repositório[1] do projeto.

4.1.4 Definição da estrutura do arcabouço

A definição estrutural do arcabouço foi influenciada pelos requisitos gerais, pela modelagem conceitual e pela tentativa de atenuação de problemas de inconsistências identificados nos trabalhos relacionados.

O arcabouço é organizado por páginas digitais, onde cada página é um arquivo. Com base na modelagem conceitual, o arcabouço possui dois tipos de páginas: uma para as classes pertencentes ao agrupamento *Contexto Privilegiado* e outra para para classes pertencentes ao agrupamento *Técnicas de Elevação de Privilégio Específicas*.

Os dois tipos de página são majoritariamente iguais, com apenas três diferenças sutis: (i) presença do elemento *Indicadores de risco* em *Contexto Privilegiado*; (ii) diferença nas definições do elemento *Objetivo*; e (iii) diferença nos títulos dos guias, refletindo seus diferentes propósitos, devido a distinção entre as duas classes de agrupamento conceitual.

A Figura 3 mostra a estrutura de seções e elementos que compõem dos dois tipos de páginas. O conteúdo que se espera encontrar em cada seção e elemento especificados é descrito a seguir. Primeiro apresentam-se os elementos comuns a ambas as páginas e, em seguida, os elementos exclusivos de cada página.

[1] https://github.com/cybersec-utfpr-cm/linpe-framework/tree/master/thesis/conceptual-modeling

Conteúdos esperados para as seções e elementos comuns aos dois tipos de páginas:

- *Pré-requisitos*: elenca assuntos fundamentais sobre sistemas operacionais cujo conhecimento prévio é necessário para o pleno entendimento dos próximos elementos.
- *Explicações conceituais*: mostra o funcionamento da técnica de elevação de privilégio em questão ou uma visão geral do funcionamento do mecanismo em questão.
- *Instruções práticas*: contempla os guias práticos que consistem em sequências de instruções, incluindo o uso de ferramentas/comandos, de modo a permitir ao administrador avaliar de forma objetiva aspectos de segurança do sistema. Ambos os guias são compostos por um conjunto de etapas de verificação sequenciais (enumeradas), onde cada etapa leva seu título e possui estrutura de elementos padronizada.
- *Motivo* (por que verificar): apresenta descrição sobre por que essa verificação deve ser feita.
- *Método* (como verificar): apresenta comandos ou instruções operacionais (textuais) para efetivamente identificar o que se busca nessa etapa.
- *Referências*: lista as fontes usadas para a elaboração do material, bem como recomendações para estudo aprofundado.

Conteúdos esperados para as seções e elementos das páginas pertencentes a subclasse *Contexto Privilegiado*:

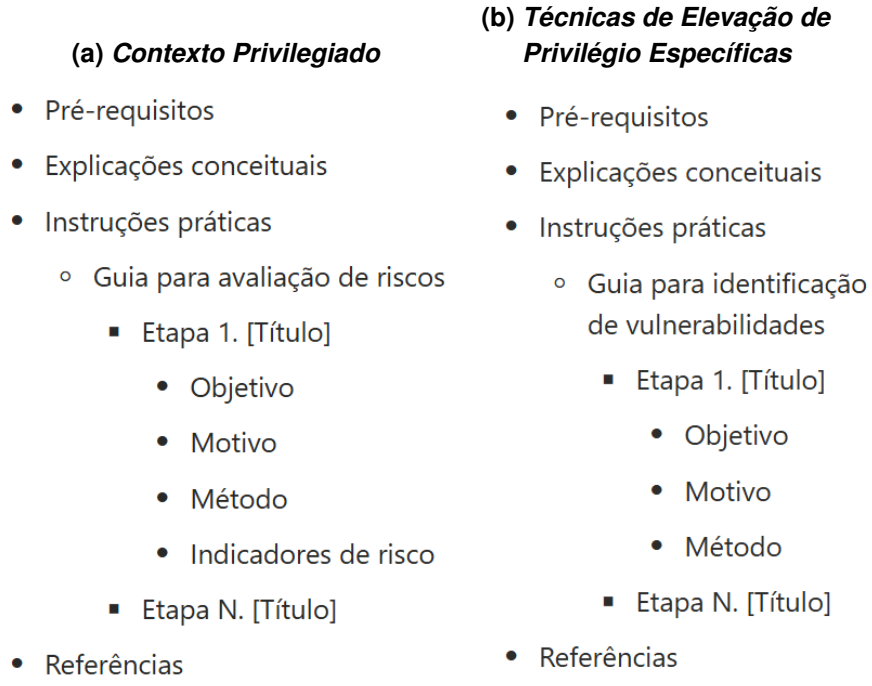
- *Guia para avaliação de riscos*: apresenta verificações direcionadas exclusivamente a identificar más configurações do mecanismo em si, sem análise de explorabilidade específica. O propósito é avaliar configurações que podem sinalizar algum grau de risco.
- *Objetivo* (o que verificar): apresenta descrição sucinta sobre o que será verificado nesta etapa.
- *Indicadores de risco*: apresenta uma lista de aspectos com potencial de risco em relação ao que pode ser descoberto durante a verificação.

Conteúdos esperados para as seções e elementos das páginas pertencentes a subclasse *Téncicas de Elevação de Privilégio Específicas*:

- *Guia para verificação de vulnerabilidade*: apresenta uma sequência de instruções cujo objetivo é identificar a presença de uma vulnerabilidade específica em sistemas GNU/-Linux que permita um ataque de elevação de privilégio. Essas instruções podem envolver, quando necessário para fins de confirmação técnica, a exploração controlada da falha como prova de conceito.

- *Objetivo* (o que verificar): apresenta aspecto/estado/configuração do sistema que viabilize essa etapa do processo de exploração através da técnica de elevação de privilégio em questão.

Figura 3 – Estruturas de páginas do arcabouço para subclasses de (a) *Contexto Privilegiado*, (b) *Técnicas de Elevação de Privilégio Específicas*.



Fonte: Autoria própria (2025).

4.2 Implementação de arcabouço mínimo

Com base na especificação do arcabouço apresentada em seções anteriores, implementou-se um exemplar mínimo denominado LinPE Framework (acrônimo de GNU/Linux Privilege Escalation Testing Framework).

O LinPE Framework foi desenvolvido como um produto mínimo viável, atuando como uma implementação inicial simplificada em relação a sua abrangência de conteúdos, mas seguindo a especificação de arcabouço proposta por este trabalho. Com isso, foi possível utilizá-lo no processo de avaliação preliminar. O LinPE Framework foi elaborado integralmente em Markdown (Gruber, 2004) e disponibilizado publicamente em repositório no GitHub[2].

O Apêndice B apresenta dois exemplos de uso do LinPE Framework.

[2] https://github.com/cybersec-utfpr-cm/linpe-framework

4.3 Avaliação preliminar do arcabouço

O questionário completo fornecido aos participantes da avaliação preliminar encontra-se no Apêndice A. Todas as respostas ao formulário feitas pelos participantes do experimento, o código-fonte de todos os ambientes virtuais Docker usados durante a avaliação preliminar e outros artefatos relacionados a avaliação preliminar estão disponíveis no repositório[3] do projeto.

A análise das respostas evidencia que os participantes identificaram corretamente, no máximo, um vetor de elevação de privilégio, com alguma variação entre eles. Um participante detectou apenas uma vulnerabilidade (avaliando corretamente 2 dos 3 cenários), outro identificou erroneamente três vetores distribuídos nos três cenários (avaliando corretamente 1 de 3), enquanto um terceiro encontrou apenas um vetor e não avaliou os demais cenários; um quarto participante não identificou nenhum vetor por dificuldades na execução das verificações. O tempo de execução variou em torno de 1 hora a 1 hora e 30 minutos.

Os comentários abertos mostram percepções majoritariamente positivas sobre organização estrutural e principalmente para a combinação entre teoria e prática. Os principais pontos de dificuldade referem-se à profundidade conceitual exigida para o pleno entendimento dos assuntos e à ausência de exemplos de saída dos comandos de verificação. Nas escalas de Likert, Q5 e Q6 apresentaram valores altos (entre 4 e 5), indicando percepção favorável da estrutura especificada quanto aos aspectos operacional e educacional. A compreensão dos termos (Q7) e a suficiência das instruções (Q8) variaram entre 3 e 5, sugerindo que, embora adequadas, as instruções não foram uniformemente suficientes para todos os usuários.

4.3.1 Discussão

Considerando a identificação de vulnerabilidades, uma das possíveis razões para o desempenho geral abaixo do esperado é a falta de conhecimento em relação ao que se trata o projeto (que poderia ter sido solucionado com a leitura atenta da página inicial do arcabouço LinPE; o seu `README.md`). Um exemplo que ilustra isso foi o participante que identificou erroneamente três vetores pois confundiu indicadores de risco com vulnerabilidades.

Os resultados revelam três padrões principais. Primeiro, há um desequilíbrio significativo entre os participantes na capacidade de identificar vetores de escalonamento de privilégio, evidenciando uma curva de aprendizagem acentuada: participantes com maior familiaridade técnica extraíram valor mais completo do arcabouço, enquanto outros demonstraram dificuldades conceituais básicas. Segundo, observou-se que a adição de exemplos de saídas reais ou de orientações adicionais sobre a interpretação dos resultados são aspectos que alguns participantes gostariam de ver no arcabouço. Terceiro, os comentários positivos recorrentes destacam a estrutura modular, a combinação entre teoria e prática e o benefício da sequência de verificações, indicando que tais elementos constituem pontos fortes a serem preservados.

[3] https://github.com/cybersec-utfpr-cm/linpe-framework/tree/master/thesis/preliminary-evaluation

Quanto às limitações da avaliação preliminar, o tamanho reduzido da amostra e seu caráter exploratório impedem qualquer generalização estatística acerca da eficácia do arcabouço. O objetivo, de fato, foi apenas identificar problemas iniciais e oportunidades de melhoria (propósito atendido pelos achados). Além disso, devido a restrições de tempo, reconhece-se que a ausência de uma divisão do experimento em dois grupos (um utilizando o arcabouço e outro utilizando recursos fragmentados, como MITRE ATT&CK ou HackTricks) inviabiliza o teste da hipótese elaborada no início desta pesquisa, impedindo, por consequência conclusões robustas sobre ganhos proporcionados pelo LinPE. Novos estudos com grupo controle e amostras maiores poderão fornecer evidências mais sólidas sobre o impacto do arcabouço no desempenho dos avaliadores iniciantes.

5 CONSIDERAÇÕES FINAIS

Esta pesquisa teve como objetivo principal a especificação de um arcabouço para testes de elevação de privilégio em sistemas GNU/Linux, com foco em profissionais iniciantes, complementado pela implementação de uma versão inicial de arcabouço baseada na sua especificação e por uma avaliação preliminar dessa implementação.

Inicialmente, uma revisão narrativa baseada em critérios indicou a ausência de um arcabouço integrado para testes de elevação de privilégios em sistemas GNU/Linux. Dessa revisão, extraíram-se requisitos gerais do arcabouço: organização tópica, ordenação sequencial, explicações conceituais, instruções práticas, abertura e manutenção ativa. Visando atender alguns desses requisitos, empregou-se uma modelagem conceitual inspirada em ontologias que iniciou o estabelecimento de uma terminologia comum sobre elevação de privilégio em GNU/Linux. A modelagem conceitual influenciou parte da estrutura especificada para o arcabouço, outra parcela da estrutura foi definida a partir dos requisitos gerais e, ainda, houve porção definida visando evitar problemas identificados em trabalhos relacionados.

Com isso, foi possível implementar um arcabouço mínimo conforme a especificação: o LinPE Framework. Embora sua abrangência de assuntos seja limitada, a padronização alcançada pela sua especificação permite que, através de trabalhos futuros e/ou de contribuições da comunidade, sua abrangência cresça gradualmente.

Em avaliação preliminar, foi considerado estruturalmente organizado de modo a favorecer profissionais iniciantes em operações de avaliação e aspectos educacionais ao unir conceitos e prática em um formato padronizado. O LinPE é implementado em Markdown e disponibilizado em repositório público no GitHub[1], o que facilita receber contribuições para incrementar sua amplitude de assuntos e se transformar em um projeto de referência na comunidade de cibersegurança, de maneira análoga aos métodos para condução de testes de intrusão.

Para trabalhos futuros, a primeira sugestão é aumentar o *corpus* do projeto de modo a abranger mais técnicas para que possa ser feita uma avaliação mais robusta, dividindo os participantes em dois grupos e comparando o uso do LinPE para verificação de vetores de elevação de privilégio ao uso de outros recursos. Outra sugestão é incrementar o arcabouço (ou utilizar apenas a estrutura especificada) para temáticas além da identificação de vetores: um exemplo seria adicionar, na seção *Instruções práticas*, um *guia para mitigação de vulnerabilidades* que trate de boas práticas e forneça instruções para melhorar a segurança do sistema em relação à elevação de privilégios. Também é considerável o aproveitamento da estrutura especificada para temas além da elevação de privilégios que envolvam a combinação entre conceitos e prática.

[1] https://github.com/cybersec-utfpr-cm/linpe-framework

REFERÊNCIAS

AHMED, A. **Privilege Escalation Techniques: Learn the Art of Exploiting Windows and Linux Systems**. Birmingham: Packt Publishing, 2021. Disponível em: https://research.ebsco.com/linkprocessor/plink?id=a8c22d55-ceda-3b19-b686-dabb2b171d82. Acesso em: 16 set. 2024.

ELDER, S. *et al.* **Do I really need all this work to find vulnerabilities? An empirical case study comparing vulnerability detection techniques on a Java application**. [*S.l.*]: , 2022. Disponível em: https://arxiv.org/abs/2208.01595. Acesso em: 13 mar. 2025.

FRANKLAND, J. The importance of standardising methodology in penetration testing. **Database and Network Journal**, „, 2009. Disponível em: https://www.thefreelibrary.com/The+importance+of+standardising+methodology+in+penetration+testing.-a0202562264. Acesso em: 14 mar. 2025.

G0TMI1K. **Basic Linux Privilege Escalation**. [*S.l.*]: , 2011. Disponível em: https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/. Acesso em: 26 fev. 2025.

GitBook. **GitBook – Documentação e guia da plataforma**. [*S.l.*]: , 2025. Disponível em: https://www.gitbook.com/. Acesso em: 24 nov. 2025.

GRUBER, J. **Markdown**. [*S.l.*]: , 2004. Disponível em: https://daringfireball.net/projects/markdown/. Acesso em: 10 dez. 2025.

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowledge Acquisition**, v. 5, n. 2, p. 199–220, 1993. Acesso em: 29 abr. 2025.

HAPPE, A.; KAPLAN, A.; CITO, J. LLMs as Hackers: Autonomous Linux Privilege Escalation Attacks. **arXiv preprint arXiv:2310.11409v5**, „, 2025. Disponível em: https://arxiv.org/pdf/2310.11409v5. Acesso em: 17 jan. 2025.

MARMEUS. **Linux Local Privilege Escalation**. [*S.l.*]: , 2023. The Pentesting Guide. Disponível em: https://the-pentesting-guide.marmeus.com/local_privilege_escalation/linux. Acesso em: 1 mar. 2025.

MAZIERO, C. A. **Segurança computacional.** [*S.l.*]: , 2019. Material didático da disciplina CI1007. Universidade Federal do Paraná. Disponível em: https://wiki.inf.ufpr.br/maziero/doku.php?id=sc:start. Acesso em: 30 nov. 2024.

MIL0. **Linux Privilege Escalation**. [*S.l.*]: , 2018. Disponível em: https://percussiveelbow.github.io/linux-privesc/. Acesso em: 25 fev. 2025.

NOY, N. F.; MCGUINNESS, D. L. **Ontology Development 101: A Guide to Creating Your First Ontology**. Stanford, CA: , 2001. Disponível em: https://protege.stanford.edu/publications/ontology_development/ontology101.pdf. Acesso em: 29 abr. 2025.

OISSG. **Information Systems Security Assessment Framework (ISSAF)**. [*S.l.*]: , 2006. Disponível em: https://www.oissg.org/. Acesso em: 17 mar. 2025.

OWASP Foundation. **OWASP Testing Guide**. [*S.l.*]: , 2014. Disponível em: https://owasp.org/www-project-web-security-testing-guide/. Acesso em: 17 mar. 2025.

POLOP, C. **Linux Privilege Escalation**. [*S.l.*]: , 2019. HackTricks. Disponível em: https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html. Acesso em: 25 fev. 2025.

PTES Organization. **Penetration Testing Execution Standard (PTES)**. [*S.l.*]: , 2014. Disponível em: http://www.pentest-standard.org/index.php/Main_Page. Acesso em: 17 mar. 2025.

SCARFONE, K. *et al.* **NIST SP 800-115 Technical Guide to Information Security Testing and Assessment**. Gaithersburg, MD: , 2008. Disponível em: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf. Acesso em: 12 mar. 2025.

SHANLEY, A.; JOHNSTONE, M. N. Selection of penetration testing methodologies: A comparison and evaluation. *In*: PROCEEDINGS OF THE 13TH AUSTRALIAN INFORMATION SECURITY MANAGEMENT CONFERENCE. 2015, Perth, Western Australia. **Anais [...]** Perth, Western Australia: Edith Cowan University, 2015. p. 65–72. Disponível em: https://ro.ecu.edu.au/ism/182. Acesso em: 14 mar. 2025.

TEMÓTEO, A. **Novo ataque hacker rouba milhões de instituição financeira e desafia BC**. [*S.l.*]: , 2025. Disponível em: https://platobr.com.br/novo-ataque-hacker-rouba-milhoes-de-instituicao-financeira-e-desafia-bc. Acesso em: 10 dez. 2025.

The MITRE Corporation. **MITRE ATT&CK Framework**. [*S.l.*]: , 2025. Disponível em: https://attack.mitre.org/. Acesso em: 15 jan. 2025.

USCHOLD, M.; GRUNINGER, M. Ontologies: Principles, methods and applications. **Knowledge Engineering Review**, v. 11, n. 2, p. 93–136, 1996. Acesso em: 29 abr. 2025.

APÊNDICES

APÊNDICE A – Avaliação preliminar: instruções e questionário

A.1 Instruções fornecidas aos participantes

A seguir expõe-se as instruções fornecidas aos participantes antes da realização do experimento de avaliação preliminar do arcabouço LinPE:

- O objetivo principal é avaliar o arcabouço para testes de elevação de privilégio em sistemas GNU/Linux (LinPE Framework) sob a ótica de sua estrutura especificada e de suas possíveis contribuições educacionais e operacionais para profissionais iniciantes.
- Para o experimento, foram elaborados três ambientes virtuais Docker. Cada ambiente simula um sistema GNU/Linux. Alguns deles contêm configurações inseguras que, se presentes, permitiriam ataques de elevação de privilégio.
- O experimento consiste em usar somente o arcabouço LinPE como guia para conduzir avaliações de segurança nos ambientes de modo a identificar se o cenário está vulnerável ou não. Note que o objetivo da atividade é apenas identificar as vulnerabilidades, e não explorá-las.
- Instancie e acesse os contêineres, acesse o arcabouço, inicie um cronômetro, faça a atividade, desligue o cronômetro e responda ao questionário.
- Observação: Se julgar conveniente, você pode manter três terminais abertos (um para cada ambiente). O modo de execução das verificações (sequencial ou paralelo entre os ambientes) é livre.
- As instruções para instanciação e acesso aos contêineres estão no arquivo `README.md` presente no arquivo compactado enviado a você (junto com os arquivos `Dockerfile`). Não acesse nenhum outro arquivo presente nessa pasta além do `README.md`.
- Acesso ao arcabouço LinPE: https://linpe-framework.gitbook.io/linpe-framework-docs/.
- Note também que o objetivo é avaliar o arcabouço LinPE. Por isso, não utilize outros recursos que abordem a temática elevação de privilégios (tais como materiais didáticos, *blog posts*, livros, artigos, notas pessoais ou modelos de linguagem de grande escala).
- Porém, o uso de materiais externos é permitido somente para esclarecer dúvidas sobre comandos GNU/Linux (como sintaxe, argumentos, comportamento, etc.) ou sobre funcionamento geral do shell.
- Tendo em vista o objetivo da avaliação, é proibido o uso de qualquer tipo de ferramenta automatizada de enumeração de informações sobre sistemas GNU/Linux (como LinPEAS, LinEnum, LSE, PrivChecker, etc.). Ferramentas padrão já instaladas no sistema (como `ls`, `find`, `ps`, `cat`, `grep`, `sudo`, etc.) podem ser usadas livremente.

Além do formulário (que continha tanto as instruções acima quanto o questionário), foi enviado aos participantes um diretório compactado. Nele, haviam os arquivos Docker com os ambientes previamente configurados e um documento (nomeado `README.md`) com instruções adicionais sobre: como criar as imagens e contêineres Docker, e como acessar os ambientes virtuais. Esses materiais estão disponíveis no repositório[1] do projeto.

A.2 Questionário da avaliação preliminar

O questionário foi utilizado para coletar dados quantitativos e qualitativos referentes à percepção dos participantes sobre o arcabouço LinPE.

As questões de 1 a 3 envolvem aspectos quantitativos, enquanto as questões de 4 a 8 foram elaboradas em escala de Likert (de um a cinco, onde um representa *nunca* ou *não favorece* e cinco representa *o tempo todo* ou *favorece muito*) e as questões de 9 a 11 foram abertas.

Visão geral em relação aos temas das perguntas: as questões de 1 a 4 envolviam a confirmação da realização da tarefa como solicitado, além da métrica de quantidade vulnerabilidades encontradas. Já as questões 5 e 6 abordaram a estrutura e organização do arcabouço, enquanto 7 e 8 avaliaram a inteligibilidade de seu conteúdo. Por fim, as questões abertas buscaram capturar a percepção subjetiva dos participante sobre o LinPE.

1. Usando o arcabouço, ao total, você conseguiu identificar quantas vulnerabilidades que permitem ataques de elevação de privilégio?
2. Como justificativa da resposta anterior, descreva brevemente a(s) vulnerabilidade(s) encontrada(s) em cada cenário.
3. Quanto tempo você levou para realizar o experimento?
4. Durante o experimento, você consultou o arcabouço LinPE com qual frequência?
5. Aspecto operacional: A estrutura especificada favorece a operacionalização de avaliações de segurança em sistemas GNU/Linux?
6. Aspecto educacional: A estrutura especificada favorece seus estudos sobre mecanismos de sistemas GNU/Linux e sobre possíveis abusos nesses mecanismos que comprometem a segurança desse tipo de ambiente?
7. Com qual frequência os conceitos e termos utilizados foram compreensíveis?
8. Com qual frequência as instruções fornecidas eram suficientes para que você realizasse as verificações?

[1] https://github.com/cybersec-utfpr-cm/linpe-framework/tree/master/thesis/preliminary-evaluation

9. Qual sua percepção sobre o arcabouço LinPE em relação a outros recursos (especialmente MITRE ATT&CK e HackTricks)?
10. Pontos positivos: O que você gostou do arcabouço LinPE?
11. Pontos negativos: Que aspectos você mudaria ou adicionaria visando a melhoria do arcabouço? Algum aspecto do arcabouço dificultou a realização da tarefa?

APÊNDICE B – Exemplos de uso do arcabouço LinPE

Este apêndice apresenta dois exemplos de uso do arcabouço LinPE para a condução de uma avaliação de segurança em um ambiente GNU/Linux simulado. O primeiro exemplo envolve avaliação de riscos, enquanto o segundo relaciona-se a identificação de vulnerabilidades.

No LinPE, os dois tipos de guias são compostos por etapas de verificação. As etapas são enumeradas com o objetivo de fornecer ao avaliador instruções sequenciais a serem seguidas sistematicamente. Entretanto, visando manter a concisão, as Seções B.2 e B.3 deste apêndice abordam apenas as etapas essenciais relacionadas ao exemplo proposto para demonstração. Apesar disso, destaca-se que o uso ideal do arcabouço consiste na verificação, com igual atenção, de todas as etapas.

Além disso, o apêndice trata de aspectos básicos relacionados a interface usada para visualização do material que compõe o LinPE.

B.1 Apresentação da interface

O arcabouço LinPE foi construído completamente em Markdown e disponibilizado publicamente visando oferecer sua abertura. Nesse sentido, fica a critério do leitor como fará a interpretação do Markdown para consumir seu conteúdo. Porém, visando facilitar o manuseio do material, realizou-se a sincronização do repositório com a plataforma GitBook (GitBook, 2025). Até o momento, isso é possível por meio de um plano gratuito que a plataforma oferece.

O GitBook oferece serviços de hospedagem e publicação de documentações e possui suporte a Markdown. A Figura 4 mostra a disposição dos principais elementos na interface do GitBook. O destaque do lado esquerdo contém o menu de navegação entre as páginas, onde cada página representa uma classe do arcabouço e é implementada como um arquivo Markdown. No lado direito, ao abrir alguma página, apresenta-se o sumário da respectiva página aberta.

B.2 Exemplo de uso do LinPE para avaliação de riscos

Em subclasses de *Contexto Privilegiado*, os *Guias para avaliação de riscos* apresentam verificações direcionadas exclusivamente a identificar más configurações do mecanismo em si, sem análise de explorabilidade específica. O propósito é avaliar configurações que podem sinalizar algum grau de risco.

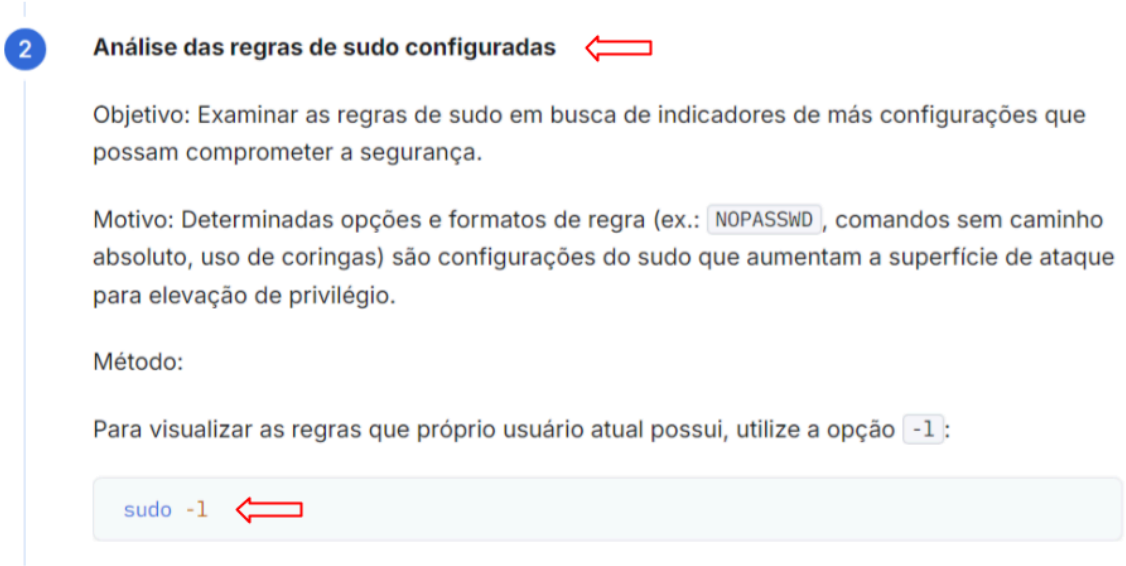
Para manter o exemplo conciso, apenas uma etapa de verificação é exibida: a análise das regras de sudo configuradas. A verificação dessa etapa leva o avaliador ao indicador de risco necessário para a posterior identificação da vulnerabilidade em si (demonstrada na Seção B.3).

A Figura 5 mostra um trecho da segunda etapa de verificação do *Guia de avaliação de riscos* relacionado a classe *Mecanismo de Controle de Acesso Sudo*. Nela, destaca-se o método apresentado para realizar essa verificação: o comando `sudo -l`.

Figura 4 – Interface do LinPE no GitBook: menu de navegação entre páginas e sumário de página.

Fonte: Autoria própria (2025).

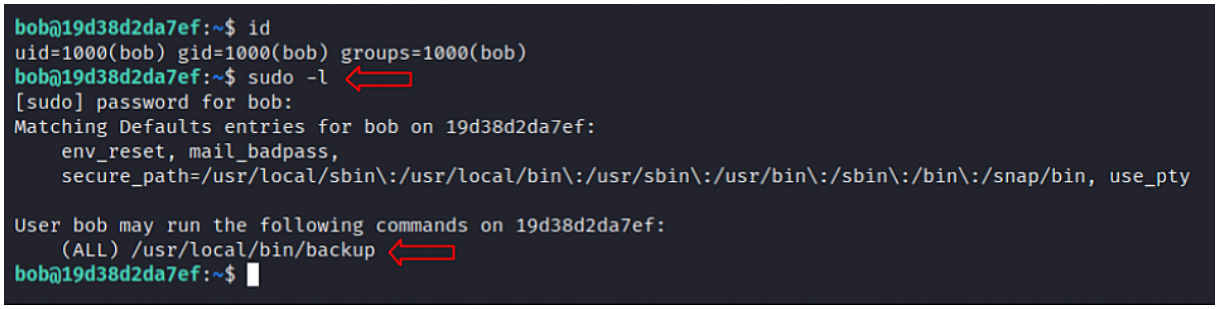
Figura 5 – Trecho da segunda etapa de verificação do *Guia para avaliação de riscos* relacionado a classe *Mecanismo de Controle de Acesso Sudo*.



Fonte: Autoria própria (2025).

Ao executar a verificação no ambiente alvo, recebe-se a saída exibida na Figura 6. O resultado da verificação mostra que o usuário `bob` pode executar o binário personalizado `/usr/local/bin/backup` em nome de qualquer usuário do sistema, inclusive do usuário administrativo `root`. Assim, caso esse binário esteja mal configurado ou vulnerável de alguma forma, isso poderá ser usado em ataques de elevação de privilégio. Por isso, essa situação in-

Figura 6 – Resultado da segunda etapa de verificação do *Guia para avaliação de riscos* relacionado a classe *Mecanismo de Controle de Acesso Sudo*.



Fonte: Autoria própria (2025).

dica um risco e o binário deve ser analisado minuciosamente em outras etapas da avaliação de segurança.

B.3 Exemplo de uso do LinPE para identificação de vulnerabilidades

Em subclasses da *Técnicas de Elevação de Privilégio Específicas*, os *Guias para identificação de vulnerabilidades* apresentam conjuntos de instruções no qual o objetivo é identificar a presença de uma vulnerabilidade específica em sistemas GNU/Linux que permita um ataque de elevação de privilégio. Para esse tipo de guia, suas instruções podem envolver, quando necessário para fins de confirmação técnica, a exploração controlada da falha como prova de conceito.

Com base no indicador de risco encontrado na seção anterior (B.2), esta seção dá continuidade ao exemplo de uso do LinPE. Dessa vez, com o objetivo de encontrar alguma vulnerabilidade (que envolva o indicador de risco encontrado), através da execução das verificações presentes no *Guia para identificação de vulnerabilidade* que pertence à classe *Manipulação de Glob (Expansão de Coringas) em Contexto Privilegiado*.

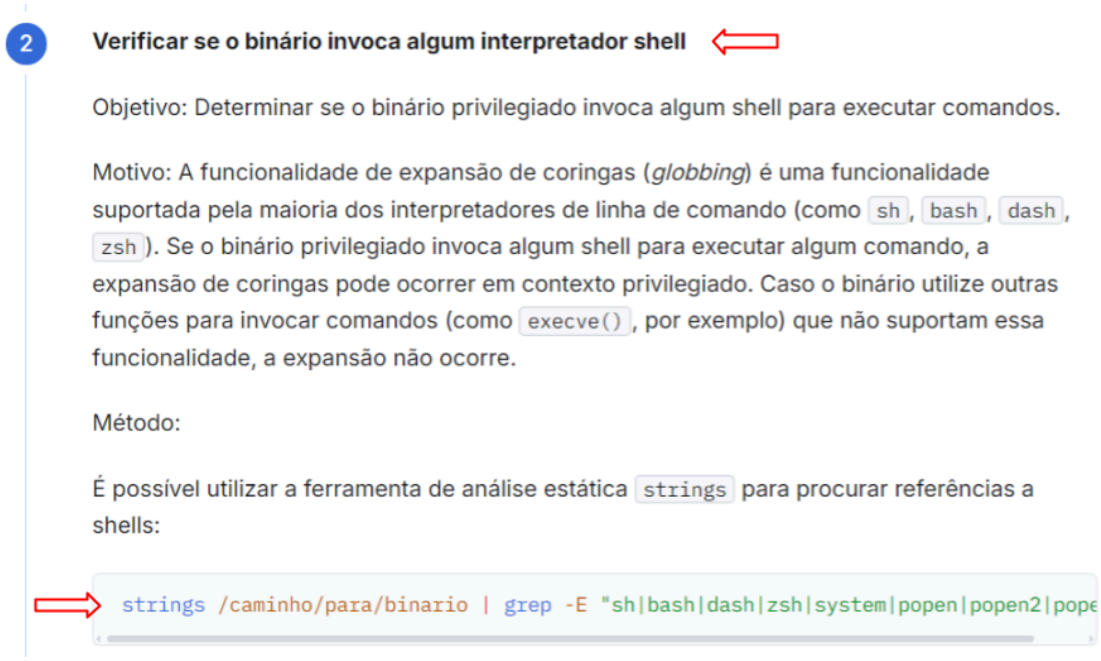
As etapas 4, 5 e 6 do guia visam realizar a análise do binário privilegiado (mencionada no fim da seção anterior), enquanto as etapas 5 e 6 avaliam a ferramenta invocada pelo binário e as permissões no diretório alvo da expansão, respectivamente.

A Figura 7 mostra um trecho da segunda etapa de verificação do guia em questão. Nela, o avaliador é orientado a usar a ferramenta `strings`[1] para verificar se o binário privilegiado invoca algum interpretador *shell*.

Ao executar a verificação no ambiente alvo, recebe-se a saída exibida na Figura 8. O resultado da verificação mostra que o binário privilegiado invoca um *shell* por meio da chamada `system`. Essa situação não descarta a possibilidade de exploração desse binário por meio dessa técnica de elevação de privilégio, então é necessário realizar as próximas verificações.

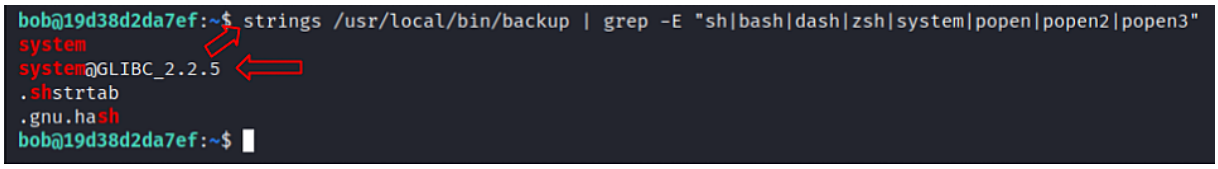
[1] O utilitário `strings` é usado para extrair sequências de caracteres imprimíveis (*strings*) de arquivos binários ou de dados, ignorando código de máquina e dados não textuais, exibindo apenas o texto legível contido no arquivo.

Figura 7 – Trecho da segunda etapa do *Guia para identificação de vulnerabilidades* relacionado a classe *Manipulação de Glob (Expansão de Coringas) em Contexto Privilegiado*.



Fonte: Autoria própria (2025).

Figura 8 – Resultado da segunda etapa de verificação do *Guia para identificação de vulnerabilidades* relacionado a classe *Manipulação de Glob (Expansão de Coringas) em Contexto Privilegiado*.



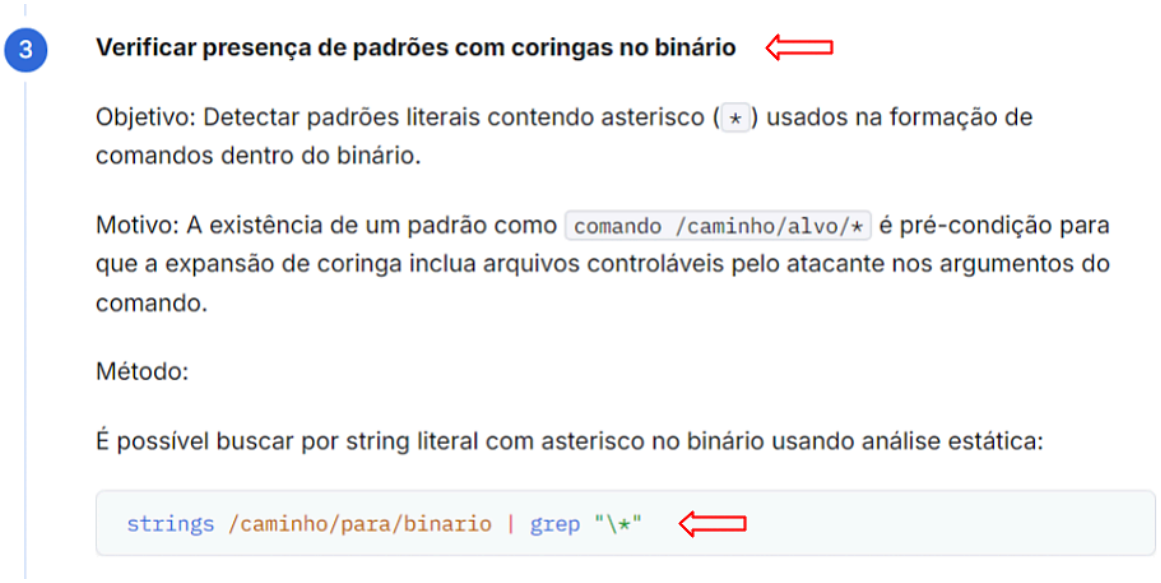
Fonte: Autoria própria (2025).

A Figura 9 mostra um trecho da terceira etapa de verificação do guia em questão. A etapa visa identificar o uso de coringas internamente pelo binário, também através do uso do utilitário `strings`.

Com a execução dessa verificação, identificou-se o uso do coringa asterisco (`*`) pelo binário em um argumento passado à uma ferramenta invocada internamente. Além disso, foi possível identificar o que busca-se na etapa 4: quais diretórios são alvo da expansão de coringas (nesse caso, o diretório `/home/bob/Documentos/`). Ambos os resultados são mostrados na Figura 10. Desse modo, observa-se a construção dos requisitos para a exploração de uma possível vulnerabilidade. Então é necessário continuar as verificações.

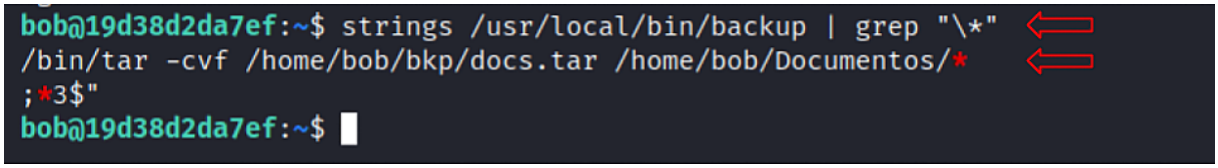
A quarta etapa mostrou que o binário invoca internamente a ferramenta `tar`. A quinta etapa orienta a verificar se essa ferramenta pode receber argumentos de forma a permitir execução de comandos, conforme ilustra a Figura 11. Ao acessar o manual do utilitário `tar`, identifica-se a opção `--checkpoint-action` que permite executar uma ação em um momento específico de sua execução. Com isso, a constatação da vulnerabilidade fica ainda mais

Figura 9 – Trecho da terceira etapa do *Guia para identificação de vulnerabilidades* relacionado a classe *Manipulação de Glob (Expansão de Coringas) em Contexto Privilegiado*.



Fonte: Autoria própria (2025).

Figura 10 – Resultados das etapas de verificação 3 e 4 do *Guia para identificação de vulnerabilidades* relacionado a classe *Manipulação de Glob (Expansão de Coringas) em Contexto Privilegiado*.



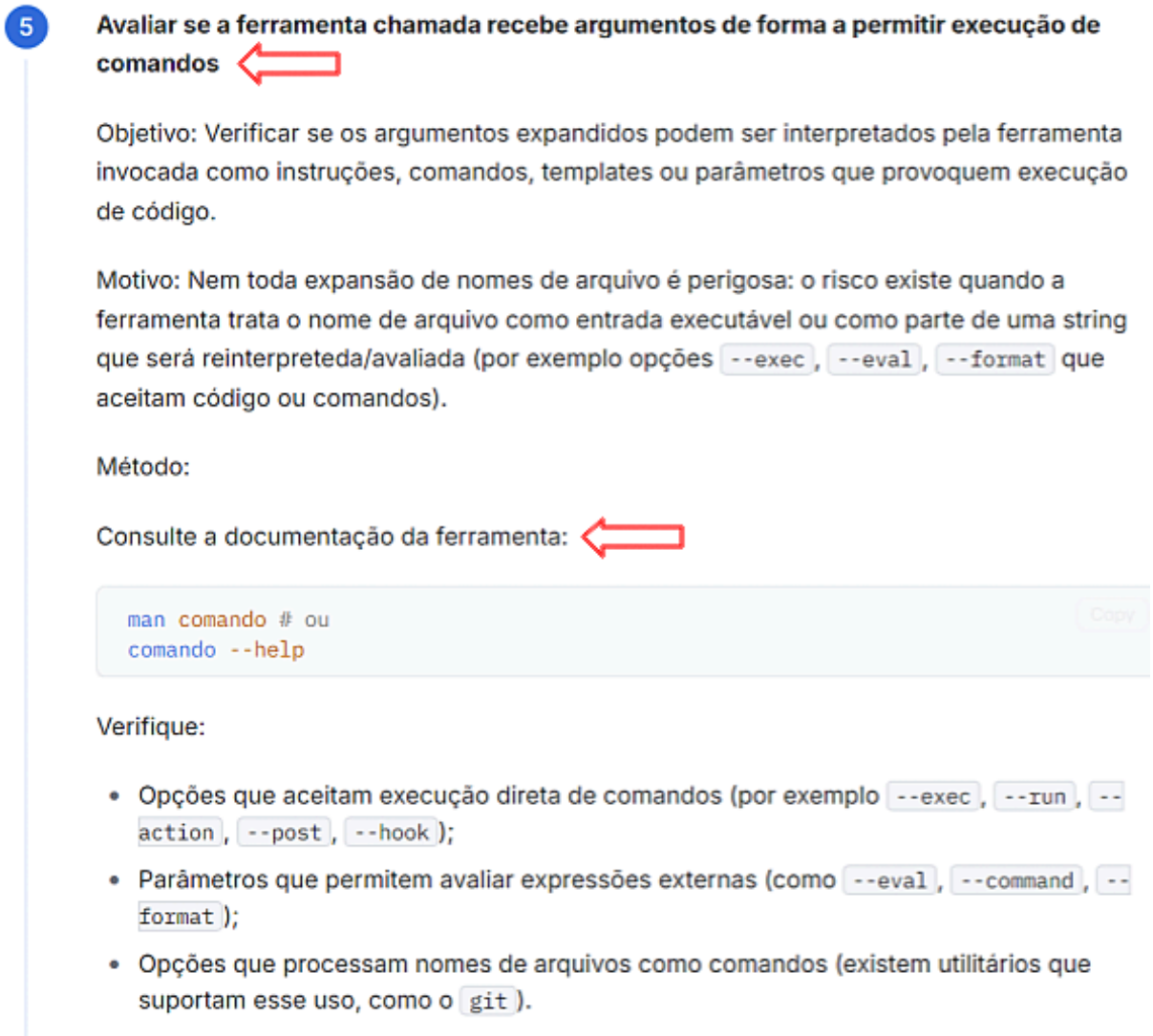
Fonte: Autoria própria (2025).

próxima, provocando a verificação da última etapa descrita no LinPE (até o momento) sobre esse assunto.

A sexta verificação, como exibe a Figura 12, visa responder quais usuários possuem permissão de escrita no diretório alvo da expansão de coringas (identificado na quarta etapa). O resultado da verificação, apresentado na Figura 13, revela que o usuário não administrativo `bob` possui permissão de escrita no diretório.

Após enumerar essas informações sobre o sistema alvo, observa-se que há uma grande possibilidade de que o sistema esteja vulnerável ao ataque de elevação de privilégio através de *Manipulação de Glob (Expansão de Coringas) em Contexto Privilegiado*. Para confirmar a presença da vulnerabilidade, é possível tentar explorar a falha como prova de conceito, sem danificar o ambiente, caso tenha autorização dos mantenedores e/ou responsáveis do sistema para fazê-lo.

Figura 11 – Quinta etapa do *Guia para identificação de vulnerabilidades* relacionado a classe *Manipulação de Glob (Expansão de Coringas) em Contexto Privilegiado*.



5 **Avaliar se a ferramenta chamada recebe argumentos de forma a permitir execução de comandos**

Objetivo: Verificar se os argumentos expandidos podem ser interpretados pela ferramenta invocada como instruções, comandos, templates ou parâmetros que provoquem execução de código.

Motivo: Nem toda expansão de nomes de arquivo é perigosa: o risco existe quando a ferramenta trata o nome de arquivo como entrada executável ou como parte de uma string que será reinterpreteda/avaliada (por exemplo opções `--exec`, `--eval`, `--format` que aceitam código ou comandos).

Método:

Consulte a documentação da ferramenta:

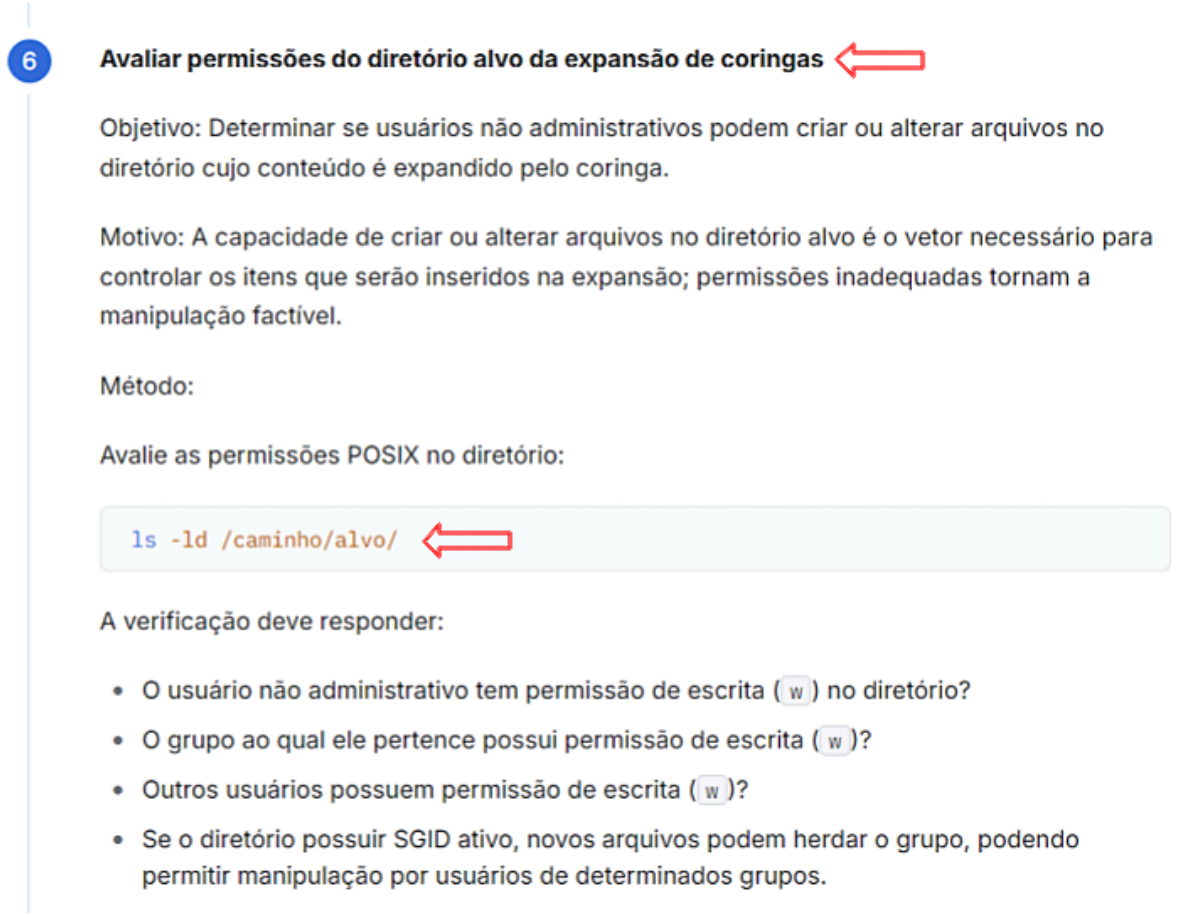
```
man comando # ou
comando --help
```

Verifique:

- Opções que aceitam execução direta de comandos (por exemplo `--exec`, `--run`, `--action`, `--post`, `--hook`);
- Parâmetros que permitem avaliar expressões externas (como `--eval`, `--command`, `--format`);
- Opções que processam nomes de arquivos como comandos (existem utilitários que suportam esse uso, como o `git`).

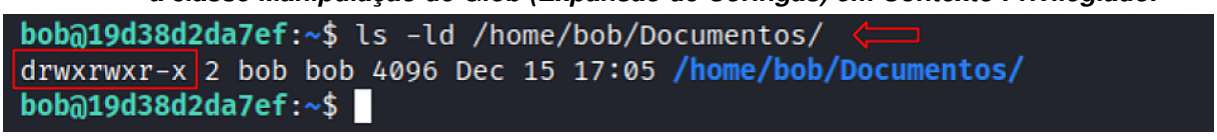
Fonte: Autoria própria (2025).

Figura 12 – Trecho da sexta etapa do *Guia para identificação de vulnerabilidades* relacionado a classe *Manipulação de Glob (Expansão de Coringas) em Contexto Privilegiado*.



Fonte: Autoria própria (2025).

Figura 13 – Resultado da sexta etapa do *Guia para identificação de vulnerabilidades* relacionado a classe *Manipulação de Glob (Expansão de Coringas) em Contexto Privilegiado*.



Fonte: Autoria própria (2025).