DOI: 10.5748/20CONTECSI/PSE/SEC/7242

eLocator: e207242

HARDENING EM DESKTOP LINUX

Pedro Henrique Paiva De Souza – https://orcid.org/0009-0006-6965-2869
Ipt - Instituto De Pesquisas Tecnológicas

Igor Cunha Teixeira – https://orcid.org/0000-0001-5928-8406
Ipt - Instituto De Pesquisas Tecnológicas

Vagner Luiz Gava – https://orcid.org/0000-0001-5965-957X
Ipt - Instituto De Pesquisas Tecnológicas

HARDENING ON LINUX DESKTOP

ABSTRACT

In cybersecurity, the term "hardening" is used to describe the activity of employing techniques, tools, and best practices to reduce the attack surface, commonly applied to servers. However, a personal computer (desktop) also needs to be secure, and the Linux operating system offers users a wide range of configurations and tools that can assist in this process. This research aims to apply and evaluate hardening techniques that can be used in a Linux desktop environment. In the experiment, a method was proposed to check vulnerable components in a Linux distribution. This method was based on specific metrics and went through the steps of identifying components and vulnerabilities, validating and categorizing the found vulnerabilities, checking security update patches, and identifying unpatched vulnerabilities for risk analysis. The result is an automated report that identifies which vulnerabilities truly affect the operating system and whether they have available fixes or not. The report is used to assist the protection process and decision-making regarding security controls in Linux desktop environments.

Keywords: Linux; *Operating System Security*; *Hardening*; CIS Benchmark.

HARDENING EM DESKTOP LINUX

RESUMO

Emsegurançacibernética, o termo*hardening* é utilizado para descreveraatividade de utilizartécnicas, ferramentas e boas práticas a fim de diminuir a superfície de ataque e, geralmente, é utilizadaemservidores. Porém, um computadorpessoal (*desktop*) tambémprecisa ser seguro e o sistemaoperacional Linux possibilitaaousuárioumagrandevariedade de configurações e ferramentas que podemauxiliarnesseprocesso. Estapesquisatemcomoobjetivoaplicar e avaliar as técnicas de *hardening* que podem ser utilizadasemambiente*desktop* Linux. No experimento, foiproposto um método para verificaroscomponentesvulneráveiscontidosemumadistribuição Linux. Essemétodobaseou-se emmétricasespecíficas e percorreu as etapas de identificação de componentes e vulnerabilidades, validação e categorização das vulnerabilidadesencontradas, verificação de patches de atualização de segurança e identificação de vulnerabilidadessemcorreção para análise de risco. Como resultado é gerado um relatórioautomatizado no qual é possívelidentificarquaisvulnerabilidadesrealmenteafetam o sistemaoperacional e se elaspossuemcorreçõesdisponíveisounão. O relatório é utilizado para auxiliar o processo de proteção e a tomada de decisãosobrecontroles de segurançaemambientes*desktop* Linux.

Palavras-chave:Linux; Segurança de sistemaoperacional; *Hardening*; CIS Benchmark.

1. INTRODUÇÃO

Linux é um sistema operacional gratuito, que foi desenvolvido com base nos princípios do UNIX. Seu design modular oferece uma grande flexibilidade e personalização, além de suportar um ambiente multiusuário. Por ser um sistema *open source*, o Linux permite que seu código seja revisado e aprimorado por qualquer pessoa, incluindo desenvolvedores e organizações (SEDANO; SALMAN, 2021).

De acordo com o estudo conduzido pela *International Data Corporation* (IDC), o Linux é o único sistema operacional de *endpoint* que está crescendo globalmente, com uma participação de mercado que aumentou de 30% em 2015 para 35% em 2017 em todo o mundo. Essa tendência de crescimento pode ser atribuída à ampla variedade de dispositivos e sistemas que utilizam o sistema Linux, tais como serviços de nuvem, smartphones, televisões, sistemas embarcados, sistemas de controle automotivo, sinalização digital, sistemas operacionais para desktop, entre outros (KALBERG, 2018).

No entanto, a crescente popularidade do sistema operacional Linux tem chamado a atenção de cibercriminosos, como revelado por uma análise realizada pela *Global ResearchandAnalysis Team* (GReAT) em diversas ameaças sofisticadas (APT) nos últimos anos. Muitos grupos de invasores têm demonstrado interesse significativo neste sistema operacional, com alguns deles desenvolvendo ferramentas para atacar máquinas baseadas em Linux (AVER, 2020).

Neste contexto, a preocupação com a segurança do sistema operacional tem se intensificado. De acordo com Melo (2014), a segurança de um sistema depende de uma série de configurações utilizadas tanto no sistema operacional quanto nas aplicações instaladas. A correta configuração do sistema operacional é um processo complexo, especialmente para sistemas baseados no kernel Linux, que oferecem amplas possibilidades de configuração.

Neste sentido, o processo de *hardening* torna-se relevante, visando a reduzir as vulnerabilidades do sistema operacional e aumentar sua resistência a ameaças cibernéticas. *Hardening* em Linux é um processo que visa a proteger e blindar o sistema operacional, incluindo o kernel e as aplicações, contra ameaças. Esse processo envolve a aplicação de técnicas específicas de controle, minimização de riscos e execução de atividades corretivas (MELO, 2014).

As CIS Benchmarks, desenvolvidas pelo *Center for Internet Security* (CIS), são um conjunto de boas práticas reconhecidas globalmente. Essas práticas foram elaboradas por especialistas e profissionais de segurança, e possuem conformidade regulatória alinhada a *frameworks* de segurança cibernética, como o *NationalInstituteof Standards and Technology* (NIST), entre outros. As diretrizes de segurança da CIS Benchmarks são detalhadas e atualizadas para diversos tipos de sistemas, possuindo três níveis de perfil de acordo com o nível de segurança a ser implementado (AWS, 2022).

Além disso, para realizar o processo de blindagem do sistema operacional por meio de *hardening*, é necessário levar em consideração alguns aspectos, uma vez que as técnicas utilizadas geralmente são implementadas em servidores e utilizam conceitos baseados no Princípio do Mínimo Privilégio (PoLP, sigla em inglês), que nem sempre são aplicáveis em *desktops*. É fundamental fazer a escolha correta do nível de proteção e dos controles que podem ser aplicados.

O objetivo deste trabalho é aplicar as técnicas de *hardening* que podem ser implementadas em ambiente *desktop* Linux, assim como o nível de *hardening* a ser aplicado e como automatizar o processo.

De modo a alcançar este objetivo, esta pesquisa procura responder à seguinte questão:

Qual metodologia e nível de *hardening* devem ser aplicados a fim de diminuir a superfície de ataque em ambiente *desktop* Linux?

Espera-se que o conhecimento gerado por este artigo contribua para o processo de proteção de ambientes *desktop* Linux auxiliando na tomada de decisão de controles de segurança.

O artigo está dividido em cinco seções. A segunda seção traz a fundamentação teórica, com uma visão geral sobre segurança da informação, Linux, *hardening* do sistema operacional e CIS Benchmark. A terceira seção apresenta a metodologia. A quarta seção descreve o experimento realizado. A quinta seção fornece os resultados obtidos e a sexta seção traz as conclusões, limitações e sugestões para futuros trabalhos de pesquisa.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção será apresentada uma visão geral sobre segurança da informação, *hardening* de sistema operacional e CIS Benchmark.

2.1 SEGURANÇA DA INFORMAÇÃO

A segurança da informação aborda a proteção de ativos de informação e sistemas de informação. Já a cibersegurança se desenvolveu da segurança da informação e possui uma abordagem mais ampla, contemplando não apenas os ativos de informação, mas também as tecnologias, processos, pessoas e melhores práticas (ALTHONAYAN; ANDRONACHE, 2022).

Segundo Mcilwraith (2021), a segurança da informação é regida por 3 princípios básicos:

- Confidencialidade – Apenas pessoas autorizadas podem visualizar a informação;
- Integridade – Lida com a validade e precisão dos dados, ou seja, os dados não podem ser alterados por pessoas não autorizadas e
- Disponibilidade – A informação deve estar acessível para os usuários autorizados a qualquer momento e sempre que for solicitada.

Já a segurança cibernética de acordo com Edgar e Manz (2017) possui alguns conceitos bem definidos como:

- Vulnerabilidade – É uma fraqueza ou falha no sistema que o torna suscetível a uma ameaça, não sendo somente uma falha de *hardware* ou *software*, já que pessoas também podem representar uma vulnerabilidade do sistema;
- Ameaça – É um perigo em potencial, que pode causar impactos adversos no sistema. Uma pessoa, organização ou governo entre outros, que possuem a intenção e os meios para executar ataques são considerados agentes de ameaça. E o método ou meio utilizado para realizar o ataque é chamado de vetor de ameaça ou vetor de ataque e
- Ataque – É uma tentativa, bem-sucedida ou não, de explorar uma vulnerabilidade especifica com a finalidade de comprometer ou obter acesso não autorizado a um sistema, recurso ou informação.

Com relação às vulnerabilidades, existe o programa *Common VulnerabilitiesandExposures*(CVE) que atribui identificadores exclusivos a vulnerabilidades associadas a componentes de *software* e *hardware*, permitindo que várias partes usem um único identificador ao discutir ou compartilhar informações sobre uma vulnerabilidade específica (NIST, 2023).

Ainda seguindo essa mesma linha de raciocínio, é importante entender o conceito de superfície de ataque que segundo Zhang et al (2021), representa o grau de exposição de um *software*, sistema ou dispositivo, ou seja, diz respeito à soma de todas as vulnerabilidades e

dos vetores de ataque que um indivíduo mal-intencionado pode utilizar para realizar um ataque.

Nesse contexto, tornou-se necessário a criação de normas de cibersegurança, sendo a família ISO 27000 responsável por tratar sobre a segurança cibernética dentro das organizações e tem como objetivo:

- Estabelecer controles de segurança da informação;
- Reduzir o risco de segurança na organização;
- Simplificar os processos de segurança cibernética e
- Selecionar, gerir, implementar e monitorar os controles de segurança da informação (THAKUR; PATHAN, 2020).

Além da ISO, existem outras organizações que versam sobre o assunto de cibersegurança e as informações sobre as técnicas e práticas para diminuir a superfície de ataque e que serão tratadas nos tópicos a seguir.

2.3 HARDENING

O termo "*hardening*" significa endurecimento, porém no contexto de cibersegurança significa blindagem do sistema. De acordo com Madureira (2021), *hardening* de sistemas é realizado por meio da utilização de ferramentas e aplicação de técnicas e práticas recomendadas para reduzir vulnerabilidades em *softwares*, *hardwares*, etc., com objetivo de diminuir a superfície de ataque e reduzir os riscos de segurança, dificultando a ação de cibercriminosos.

Segundo Santos e Nobre (2019), esse processo pode ser dividido em três etapas: planejamento, configuração e manutenção. A primeira etapa envolve a análise e planejamento dos requisitos e funcionalidades que o sistema deve atender. A segunda etapa consiste na configuração segura de *softwares* e sistemas operacionais, incluindo a aplicação de configurações enxutas, a remoção de pacotes e serviços desnecessários, o alinhamento da política de controle de acesso aos recursos do sistema, a instalação de rotinas de coletas de logs e opcionalmente, a adesão de tecnologias de detecção ou prevenção de intrusão.

O objetivo da terceira etapa do processo de *hardening* é manter o sistema operando em conformidade com os requisitos de segurança, estabelecendo políticas de monitoramento, revisão e atualização do sistema operacional. O processo de *hardening* não é estático e deve ser continuamente avaliado e atualizado à medida que novas ameaças e vulnerabilidades são descobertas (SANTOS; NOBRE, 2019).

Para Boelen (2018), esse processo leva em consideração 3 pilares:

- Princípio do Mínimo Privilégio (PoLP – *PrincipleofLeastPrivilege*) – Trata-se de reduzir os privilégios de usuários, limitando o acesso aos recursos necessários para o desempenho de seu trabalho;
- Segmentação – Diz respeito ao uso da memória, no qual cada processo pode acessar apenas seus próprios segmentos de memória e
- Redução – Consiste em eliminar *softwares* e aplicativos desnecessários, minimizando potenciais vulnerabilidades e também melhorando o desempenho e estabilidade do sistema.

Existem modelos de *hardening* do sistema que são disponibilizados por organizações como o NIST ou Center for Internet Security (CIS) e que podem ser utilizados como referência do estado de segurança do sistema que se pretende alcançar (ZLOTNIK, 2021).

Porém, de acordo com Nguyen e Dupuis (2019), existe um conflito entre usabilidade e segurança que afeta diretamente a Experiencia do Usuário (UX – *User Experience*). Exemplos disso são as políticas de senha e a autenticação multifator (MFA), que geram resistência por parte dos usuários, interferindo no uso de seus dispositivos pessoais e demandando tempo adicional para autenticação, o que, por sua vez, prejudica a interação entre o usuário e o sistema.

2.4 CIS BENCHMARK

O *Center for Internet Security* (CIS) é uma organização sem fins lucrativos que se dedica ao desenvolvimento, promoção e manutenção das melhores práticas em defesa cibernética. Essas práticas são baseadas na experiência de profissionais das áreas de TI e segurança cibernética, com o objetivo de proteger organizações contra ameaças cibernéticas. Um dos principais conjuntos de práticas desenvolvido pela CIS é o CIS Benchmark, que oferece diretrizes detalhadas para configurar com segurança diversos sistemas (ORTIZ-GARCES; ECHEVERRIA-LOPEZ; ANDRADE, 2020).

Conforme Paya et al (2022), as CIS Benchmarks são as melhores práticas recomendadas e servem como linha de base para a realização de *hardening* de sistemas. Cada recomendação segue as diretrizes do CIS *Controls* e também está em conformidade com estruturas regulatórias e padrões estabelecidos pelo NIST e ISO 27000 entre outros.

Para Sedano e Salman (2021),as CIS Benchmarks possuem dois níveis de configuração, sendo que cada nível descreve as configurações recomendadas conforme descrito abaixo:

- Nível 1 – Possuí as configurações básicas de segurança e foi projetada para reduzir a superfície de ataque causando o mínimo de impacto no desempenho das máquinas e não prejudicando a funcionalidade dos negócios e
- Nível 2 – É considerado “defesa em profundidade” sendo recomendado para ambientes que necessitam de maior segurança. Nesse nível, as configurações podem gerar efeitos adversos caso sejam implementadas inadequadamente, ou sem o devido cuidado.

Segundo CIS (2022), as CIS Benchmarks também possuem o nível de configuração STIG que apresenta todas as recomendações do *Security TechnicalImplementationGuides* (STIG).

O STIG é um padrão de configuração de sistemas e aplicativos desenvolvido originalmente para Departamentos de Defesa (DoD) com o objetivo de dificultar o acesso de invasores aos sistemas. Abrange uma ampla variedade de produtos, incluindo sistemas operacionais, dispositivos de rede e dispositivos móveis, entre outros. Os STIGs são referências para a configuração segura de sistemas e aplicativos e são amplamente utilizados em organizações governamentais e em setores que exigem altos níveis de segurança (ROSE; ZHOU, 2020).

3. METODOLOGIA

Para a seleção dos artigos utilizados como base para o desenvolvimento das discussões apresentadas neste trabalho, foi realizada uma revisão sistemática da literatura. Essa abordagem é baseada no relatório técnico de Kitchenham (2004), que define a revisão sistemática como um meio de identificar, avaliar e interpretar todas as pesquisas relevantes disponíveis para uma questão de pesquisa específica, área de tópico ou fenômeno de interesse. A estratégia de busca utilizada permitiu avaliar a completude da busca para garantir que todos os estudos relevantes foram incluídos na revisão sistemática da literatura.

3.1 PROTOCOLO DA REVISÃO SISTEMÁTICA

As bases de dados utilizadas neste estudo foram selecionadas por seu reconhecimento mundial e por incluírem as principais publicações científicas, revistas e eventos na área de computação. Essa escolha foi feita com o objetivo de garantir que a revisão sistemática da literatura fosse abrangente e envolvesse a literatura mais relevante e atualizada disponível, conforme quadro 1.

A pesquisa foi realizada em fevereiro de 2023.

Quadro 1 - Bases de Busca

Base de Busca	Endereço virtual (URL)
Web of Science	http://www.webofscience.com/
IEEE *Xplore*	https://ieeexplore.ieee.org/
Scopus*DigitalLibrary*	http://www.scopus.com/

Fonte: Autor

A seguir estão os critérios de inclusão e exclusão dos artigos para esta revisão sistemática:

Critérios de Inclusão

- Deve conter Linux;
- Deve conter *hardening* de sistema operacional;
- Deve conter padrões de *hardening*.

Critérios de Exclusão

- Trabalhos duplicados;
- Trabalhos indisponíveis para leitura na íntegra por meio da plataforma CAFe;
- Não deve focar em *Soft Error*;
- Não deve focar em *TrustedComputing*;
- Não deve focar em segurança de hypervisor;
- Não deve focar em IDS (sistema de detecção de intrusão);
- Não deve abordar somente chamadas de sistema;
- Não deve abordar somente segurança de fluxo de controle.

Após a pesquisa inicial, a seleção dos artigos seguiu o seguinte protocolo:

- Etapa 1: Exclusão de artigos duplicados;
- Etapa 2: Leitura dos títulos, resumos e palavras-chave;
- Etapa 3: Leitura da introdução e conclusão;
- Etapa 4: Leitura completa dos artigos.

3.2 ANÁLISE DOS ARTIGOS

Nesta seção, serão abordados os principais pontos relacionados ao processo de *hardening* do sistema operacional, a partir dos artigos selecionados. Serão discutidas as

recomendações e práticas sugeridas pelos autores para reduzir as vulnerabilidades e aumentar a segurança do sistema, destacando as medidas mais efetivas e os resultados obtidos.

Para a seleção inicial dos artigos, foram utilizadas as seguintes *strings* de busca restritas aos metadados dos trabalhos (título, resumo e palavras-chave), com o objetivo de encontrar estudos publicados a partir de 2018 ou mais recentes:

- ***hardening AND Linux AND (workstation OR desktop OR guide OR benchmark OR standard)***

A tabela 1 mostra o número de artigos encontrados de acordo com base de busca:

Tabela 1 - Artigos por Base de Busca

Base de Busca	Quantidade de artigos
Web of Science	**5**
IEEE *Xplore*	**14**
Scopus*DigitalLibrary*	**22**

Fonte: Autor

Após a realização das 4 etapas de seleção (figura 1), foram escolhidos 5 artigos que atendiam aos critérios definidos para esta revisão sistemática, conforme quadro 2 abaixo.

Figura 1 - Etapas de seleção de artigos

Fonte: Autor

Quadro 2 – Artigos selecionados

Artigo	Autores
Auditing Linux Operating System with Center for Internet Security CIS Standard	**Sedano e Salman**
Automation of Server Security Assessment	**Varun e Agarwal**
Methodologicalproposal for the optimization oftheinstallation times ofhardenized Linux operating systems throughtheSpacewalksolution in criticalinfrastructures	**Ortiz-Garces, Echeverria-Lopez e Andrade**
System Hardening for Infrastructure as a Service IaaS	**Rose e Zhou**
VulnerabilityIdentificationonGNULinuxOperatingSystemsthrough Case-BasedReasoning	**Santos e Nobre**

Fonte: Autor

No estudo de Sedano e Salman (2021), foi abordado o processo de *hardening* do sistema operacional Linux Ubuntu 20.04 por meio da utilização da ferramenta de automação Chef Inspec, seguindo as recomendações do CIS Benchmark. Os controles implementados

foram organizados em seis grupos: Configuração inicial, Serviços, Configuração de rede, Registro e auditoria, Acesso, autorização e autenticação, Manutenção do sistema.

Do total de 241 controles sugeridos, nove não puderam ser automatizados, e sete controles não foram completamente inspecionados.

Já no estudo de Varun e Agarwal (2022), é discutido o uso de ferramentas para automação de verificação de segurança do sistema operacional Linux Ubuntu, com foco na utilização da ferramenta Ansible para automação e da ferramenta Lynis para coleta de informações de configuração do sistema operacional. O objetivo é identificar áreas de risco e gerar um relatório que forneça ao usuário uma imagem clara da superfície de ataque e sugestões de *hardening* do sistema.

No trabalho de Ortiz-Garces, Echeverria-Lopez e Andrade (2020), os autores descrevem a aplicação automatizada de *hardening* no sistema operacional Linux CentOS 7 e 8, seguindo as diretrizes recomendadas pelo CIS Benchmark. O processo de implementação das configurações de *hardening* é realizado durante a instalação do sistema operacional por meio do aplicativo Spacewalk. Esse método resulta em uma significativa redução de tempo gasto para instalação e aplicação de *hardening* no sistema, além de possuir alta taxa de conformidade com o CIS Benchmark, conforme tabela 2.

Tabela 2 - Percentual de conformidade com o CIS Benchmark níveis 1 e 2

OS	*Version*	*Level 1 Server*	*Level 2 Server*	*Level 1 Workstation*	*Level 2 Workstation*
CentOS Minimal	7	92%	79%	92%	80%
CentOS Minimal	8.2	90%	78%	90%	79%
CentOS GUI Server	7	95%	82%	95%	83%
CentOS GUI Server	8.2	93%	79%	93%	80%

Fonte: Ortiz-Garces et al, 2020

Em outro trabalho relevante de Rose e Zhou (2020), as autoras apresentam um experimento de *hardening*do sistema operacional em ambiente de nuvem, utilizando benchmarks de sistema operacional padrão. O sistema operacional utilizado foi o RedHat Enterprise Linux 7 (RHEL 7), o benchmark selecionado foi o STIG e a ferramenta utilizada para auditoria e *hardening*do sistema foi o OpenSCAP. Os resultados dos testes demonstraram que a aplicação de técnicas de *hardening*pode reduzir significativamente a superfície de ataque do sistema, mitigando os ataques discutidos nos estudos de caso apresentados no artigo. As autoras alcançaram um índice de conformidade de 98% com o benchmark selecionado, evidenciando a eficácia do processo de *hardening* e a importância da utilização de ferramentas automatizadas para auditoria e aplicação de controles de segurança.

No estudo referência de Santos e Nobre (2019), é abordado o tema de identificação de vulnerabilidades do sistema operacional por meio da utilização de um sistema de Inteligência Artificial de Raciocínio Baseado em Casos (CBR). Por meio deste sistema, é possível que um profissional inexperiente atinja um nível de identificação de vulnerabilidades muito próximo ao de um profissional experiente. A etapa de identificação de vulnerabilidades é fundamental para a aplicação de *hardening* no sistema operacional, uma vez que a correta identificação das vulnerabilidades é o primeiro passo para sua mitigação e prevenção.

Dentre os artigos encontrados três merecem destaque sendo eles os trabalhos de: Sedano e Salman (2021), Ortiz-Garces, Echeverria-Lopez e Andrade (2020) e Rose e Zhou (2020). Tais artigos são relevantes por apresentarem os benchmarks utilizados em conjunto com os controles de segurança aplicados, além de fornecerem dados precisos sobre o percentual de conformidade atingido em cada experimento, contribuindo para a avaliação do impacto das medidas de *hardening* na segurança do sistema operacional e permitindo uma comparação mais precisa entre diferentes abordagens de *hardening*.

Os artigos de Varun e Agarwal (2022) e Santos e Nobre (2019) apresentam uma abordagem mais específica em relação à identificação de vulnerabilidades em sistemas operacionais Linux, sendo assim, possuem menor relevância para este estudo que tem como foco a aplicação de benchmarks de segurança e *hardening* de sistema operacional Linux.

4. EXPERIMENTO

O experimento concentra-se na redução de vulnerabilidades em sistemas operacionais *desktop* baseados em Linux, empregando técnicas de *hardening* de pacotes. Esse tipo de abordagem está inserido no processo de *hardening*, seguindo as diretrizes do CIS Benchmark, conforme utilizado por Sedano e Salman (2021) e Ortiz-Garces, Echeverria-Lopez e Andrade (2020). O mesmo princípio é aplicado no STIG, utilizado por Rose e Zhou (2020).

Entretanto, o procedimento apresentado por esses artigos tem como objetivo principal atualizar os pacotes que possuem patches de atualização disponíveis, sem identificar e considerar as vulnerabilidades existentes no sistema para as quais não há correção disponível. Como complemento, o método proposto desempenha as funções de identificar as vulnerabilidades relacionadas a cada componente do sistema operacional, validar e classificar as vulnerabilidades encontradas, verificar a existência de patches de atualização de segurança e identificar as vulnerabilidades sem correção para a análise de risco.

Esse processo é realizado a partir de uma imagem ISO contendo um sistema operacional Linux baseado em Debian. Inicialmente, a ISO é submetida a uma análise binária utilizando a ferramenta *Black DuckBinaryAnalysis*, que de acordo com Synopsys (2023), é uma ferramenta que oferece transparência sobre as dependências de código aberto e de terceiros presentes em executáveis, bibliotecas e firmware. Essa análise gera um relatório detalhado que lista todos os pacotes instalados, suas versões e as vulnerabilidades conhecidas associadas a cada pacote (CVEs). A figura 2 mostra a análise realizada pela ferramenta *Black DuckBinaryAnalysis*.

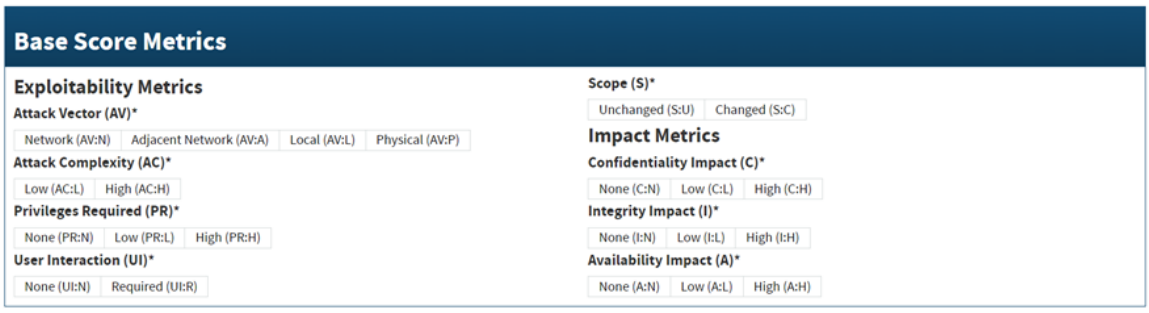
Figura 2 - Análise Black DuckBinaryAnalysis

Fonte: Autor

Posteriormente, submete-se esse relatório a uma segunda análise por meio da ferramenta Gost. O Gost é responsável por criar uma réplica local do *Security Tracker*, que é o registro contínuo e atualizado das informações de segurança para sistemas operacionais Debian e RedHat, entre outros (VULSIO, 2023). Essa etapa verifica a aplicabilidade das CVEs ao sistema operacional selecionado e a existência de correções para os problemas identificados. Além disso, necessita-se de informações adicionais da CVE, que podem ser consultadas no site https://www.cve.org/ e que possui uma API para consultas.

Para classificar as vulnerabilidades identificadas, são adotados critérios baseados nas métricas do *Common VulnerabilityScoring System* (CVSS), um método utilizado para avaliação qualitativa da gravidade de vulnerabilidades (NIST, 2023), e que podem ser observadas no quadro 3. No contexto desta análise, foram estabelecidas duas categorias de maior preocupação: *Blocker*e *High*.

Quadro 3 - Métricas CVSS versão3



Fonte: NIST, 2023

*Blocker*é a categoria de maior risco para o usuário, caracterizada pelas métricas AV:N (Vetor de Ataque: Rede), AC:L (Complexidade de Ataque: Baixa) e UI:N (Interação do Usuário: Nenhuma). A categoria *High* é semelhante ao *Blocker*, exceto pela complexidade de ataque, que é alta (AC:H), mantendo as demais métricas como AV:N e UI:N.

Essas informações permitem a construção de um relatório contendo os dados necessários para a análise e aplicação do *hardening* de pacotes. No entanto, realizar esse

processo manualmente demandaria muito tempo e esforço. Diante disso, decidiu-se criar um script em Python para automatizar grande parte do processo.

O script em Python atua com base no relatório proveniente da análise realizada pelo *Black DuckBinaryAnalysis*, extraindo dados cruciais como componentes, versões, CVEs e métricas do CVSS v3. Durante esse processo, as CVEs duplicadas são eliminadas e é estabelecida uma função para classificar as vulnerabilidades de acordo com critérios pré-definidos.

Em seguida, são efetuadas requisições para a API do Gost, verificando cada CVE identificada anteriormente. Destacam-se algumas observações importantes neste ponto:

1. As CVEs não encontradas nas requisições ao Gost são adicionadas ao relatório final com a informação "status: Não encontrado".

2. Houve discrepância nos nomes de alguns componentes ao comparar as informações do *Black DuckBinaryAnalysis* e do Gost. A validade da informação proveniente do Gost foi confirmada e incorporada ao relatório.

3. Cada CVE identificada pelo Gost é acompanhada pela informação "release", que corresponde às versões da distribuição Debian, como Bookworm e Trixie. Adicionalmente, é fornecida a informação "status", indicando se a CVE possui correção (*resolved*) ou não (*open*).

Para obter informações complementares, como o responsável pelo registro da CVE, foi necessária a consulta à API do CVE.ORG. Essa consulta é realizada apenas para as CVEs identificadas pelo Gost.

Com todos os dados reunidos, é elaborado um relatório contendo os seguintes campos e informações:

• CVE: Número da CVE identificada.

• Pacote: Nome do pacote ou componente identificado.

• Status: Versão da distribuição Debian seguida pela indicação da disponibilidade ou não de correção.

• Versão Instalada: Versão do pacote encontrada na ISO.

• Versão Corrigida: Versão do pacote que soluciona a vulnerabilidade.

• Última Versão do Repositório: Versão atual do pacote disponível no repositório do Debian para a versão selecionada.

• Severidade: Classificação da vulnerabilidade de acordo com os critérios estabelecidos.

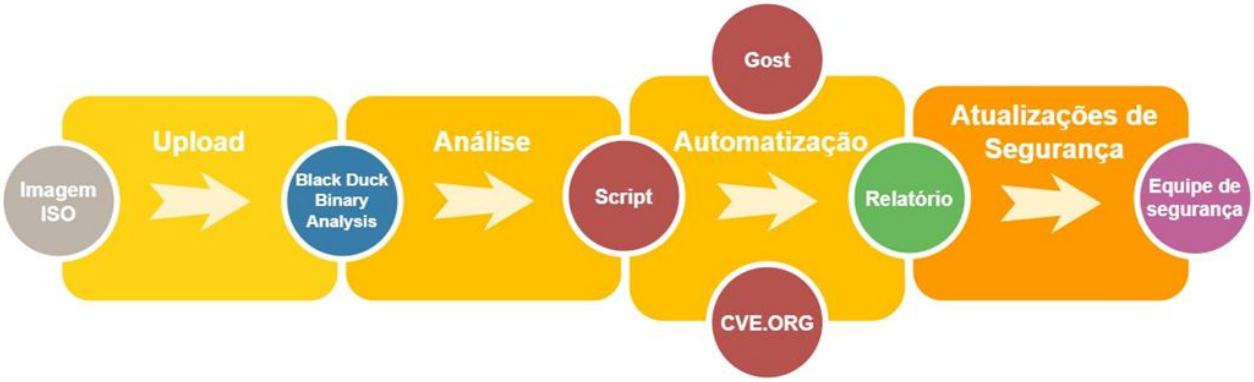
• Vetor de Ataque: Métricas do CVSS v3 associadas à vulnerabilidade.

• Registro: Responsável pelo registro da CVE.

• Descrição: Informação descritiva da CVE.

A figura 3 mostra as fases percorridas até a produção do relatório.

Figura 3 - Fluxograma de fases para geração do relatório



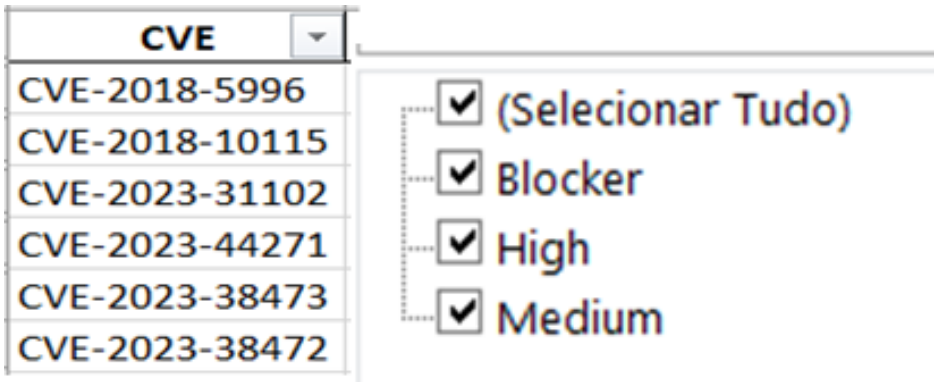
Fonte: Autor

5. RESULTADOS

Nesta seção, serão abordados os principais pontos relacionados ao experimento realizado de *hardening* do sistema operacional desktop Linux.

O relatório final, elaborado ao término das etapas, abrange todas as informações pertinentes para a implementação de *hardening*de pacotes. Destacando todas as CVEs identificadas que impactam o sistema operacional, possibilitando a filtragem com base nas categorias sugeridas (figura 4).

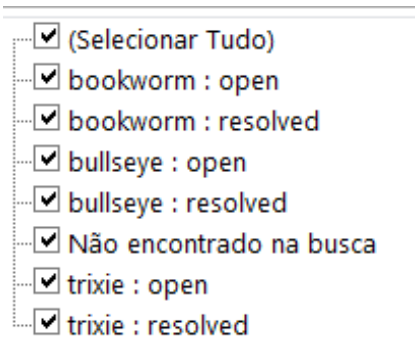
Figura 4 – CVEs identificadas e Categorias



Fonte: Autor

Além disso, é possível filtrar os resultados com base na versão da distribuição Debian utilizada, ou ainda verificar quais CVEs não obtiveram resposta na consulta a ferramenta Gost, conforme figura 5.

Figura 5 – Status da vulnerabilidade com base na versão da distribuição Debian



Fonte: Autor

Dentro de cada categoria, o relatório simplifica a identificação das vulnerabilidades já corrigidas (quadros 4 e 5) daquelas que ainda necessitam de soluções (quadros 6 e 7).

Quadro 4 – Vulnerabilidades com correção parte 1

CVE	Pacote	Status	Versão Instalada	
CVE-2018-5996	p7zip-rar	trixie : resolved	16.02+dfsg-8	1
CVE-2018-10115	p7zip-rar	trixie : resolved	16.02+dfsg-8	1

Fonte: Autor

Quadro 5 - Vulnerabilidades com correção parte 2

Versão Corrigida	Ultima versão do Repositór	Severida	Vetor de Ataque	Registro	
16.02-2	16.02-3	Medium	AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H	mitre	I
16.02-3	16.02-3	Medium	AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H	mitre	I

Fonte: Autor

Quadro 6 - Vulnerabilidades sem correção parte 1

CVE	Pacote	Status	Versão Instalada
CVE-2023-28450	dnsmasq	trixie : open	2.89-1
CVE-2023-40569	freerdp2	trixie : open	2.10.0+dfsg1-1

Fonte: Autor

Quadro 7 - Vulnerabilidades sem correção parte 2

Versão Corrigida	Ultima versão do Repositório	Severidade	Vetor de Ataque	Registro
	2.89-1	Blocker	AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H	mitre
	2.11.2+dfsg1-1	Blocker	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	GitHub_M

Fonte: Autor

O relatório é utilizado para a verificação das vulnerabilidades detectadas e que foram classificadas como *Blocke*r e *High,* destacando as vulnerabilidades que não possuem correção. Neste estágio, torna-se essencial analisar se o componente exerce uma função crucial no sistema, se há outra alternativa de pacote não vulnerável com funcionalidade similar e se o risco pode ou não ser aceitável.

Esse procedimento pode ser incorporado ao ciclo de desenvolvimento de sistemas operacionais Linux, podendo ser implementado em conjunto com outros requisitos de segurança, tais como desenvolvimento seguro e testes de segurança, entre outros.

6. CONCLUSÃO

Considerando a análise realizada, é possível notar que há uma carência de estudos acadêmicos recentes sobre o tema do *hardening* em ambiente desktop Linux, conforme evidenciado pela quantidade limitada de artigos encontrados nas bases de pesquisa. No entanto, há um grande volume de informações disponíveis em literatura cinzenta, que inclui padrões e guias produzidos por empresas e organizações especializadas em segurança. De acordo com Dudziak (2021), a literatura cinzenta refere-se a informações produzidas por diferentes fontes, incluindo governos, acadêmicos, empresas e indústrias, em formato eletrônico ou impresso e que não são controladas por publicações comerciais.

A seguinte questão de pesquisa foi formulada: "Qual metodologia e nível de *hardening* devem ser aplicados para diminuir a superfície de ataque em ambiente desktop Linux?".

Com base na revisão sistemática e no experimento conduzidos neste estudo, pode-se inferir que o método de *hardening*de pacotes, é aquele que gera o menor impacto na experiência do usuário ao aplicar medidas de blindagem em sistemas operacionais desktop Linux. Por outro lado, a adoção de padrões e benchmarks citados na revisão sistemática da

literatura e que são centrados na redução dos privilégios do usuário, contrariam a proposta deste artigo, e são principalmente usados por empresas.

Além disso, é importante destacar que o processo de *hardening* deve ser mantido de forma contínua e, entre outras tarefas, deve envolver a aplicação de atualizações no sistema operacional, bem como a utilização de ferramentas destinadas a mitigar as novas ameaças que surgem constantemente.

Com relação as limitações, esse estudo possui contexto diferente dos trabalhos analisados, uma vez que sua ênfase reside na redução da superfície de ataque em sistemas desktop sem impactar a experiência de uso, resultando em restrições nas ações de *hardening* que podem ser aplicadas. Além disso, uma limitação crítica é a dependência exclusiva da ferramenta *Black DuckBinaryAnalysis*, exigindo a busca por uma ferramenta que possa oferecer resultados equiparáveis ao *Black DuckBinaryAnalysis*.

Espera-se que o conhecimento gerado por este artigo contribua para o processo de proteção de ambientes desktop Linux. Além disso, espera-se que este estudo possa ser utilizado como base para futuras pesquisas relacionadas ao assunto, a fim de avançar no desenvolvimento de estratégias e soluções para a proteção de sistemas operacionais desktop Linux.

REFERÊNCIAS

ALTHONAYAN, Abraham; ANDRONACHE, Alina. Shifting from information security towards a cybersecurity paradigm. In: **Proceedings of the 2018 10th International Conference on Information Management and Engineering**. 2018. p. 68-79.

AVER, Hugh. **Linux é um sistemaoperacionalinvulnerável?** 2020. Disponívelem: https://www.kaspersky.com.br/blog/threats-targeting-linux/16032/. Acessoem: 17 abr. 2023.

AWS. **O que são as CIS Benchmarks?** 2022. Disponível em: https://aws.amazon.com/pt/what-is/cis-benchmarks/. Acesso em: 23 ago. 2022.

BOELEN, Michael. **Linux hardening steps for starters**. 2018. Disponível em: https://linux-audit.com/linux-server-hardening-most-important-steps-to-secure-systems/. Acesso em: 05 jan. 2023.

CIS. **CIS Benchmarks™ FAQ**. 2022. Disponível em: https://www.cisecurity.org/cis-benchmarks/cis-benchmarks-faq. Acessoem: 21 nov. 2022.

DUDZIAK, Elisabeth. O que é literaturacinzenta? AGUIA Blog, 16 ago. 2021. Disponívelem: https://www.aguia.usp.br/noticias/o-que-e-literatura-cinzenta/Acessoem: 03 mai. 2023.
EDGAR, Thomas; MANZ, David. **Research methods for cyber security**. Syngress, 2017.

KALBERG, Jeff. **Five Trends Influencing Linux's Growth at the Endpoint**. 2018. Disponívelem: https://www.linuxjournal.com/content/five-trends-influencing-linuxs-growth-endpoint. Acessoem: 17 abr. 2023.

KITCHENHAM, Barbara. Procedures for performing systematic reviews. Keele, UK, Keele University, v. 33, n. 2004, p. 1-26, 2004.

MADUREIRA, Regis. **O que é Hardening e por que isso é importante?** 2021. Disponível em: https://tripla.com.br/o-que-e-hardening-e-por-que-isso-e-importante/. Acesso em: 29 nov. 2022.

MCILWRAITH, Angus. **Information security and employee behaviour: how to reduce risk through employee education, training and awareness**. Routledge, 2021.

MELO, Sandro. **Hardening em Linux**. Rio de Janeiro: Escola Superior de Redes Rnp, 2014. 282 p. Disponível em: https://kupdf.net/download/hardening-em-linux_59f37d36e2b6f51b0a240fed_pdf. Acessoem: 15 jul. 2022.

NGUYEN, Jessica; DUPUIS, Marc. Closing the feedback loop between ux design, software development, security engineering, and operations. In: **Proceedings of the 20th Annual SIG Conference on Information Technology Education**. 2019. p. 93-98.

NIST. Vulnerabilities. 2023. Disponívelem: https://nvd.nist.gov/vuln. Acessoem: 13 nov. 2023.

NIST. **Vulnerability Metrics**. 2023. Disponívelem: https://nvd.nist.gov/vuln-metrics/cvss. Acessoem: 13 nov. 2023.

ORTIZ-GARCES, Ivan; ECHEVERRIA-LOPEZ, Aaron; ANDRADE, Roberto O..Methodologicalproposal for the optimization oftheinstallation times ofhardenized Linux operating systems throughtheSpacewalksolution in criticalinfrastructures. **2020 InternationalConferenceOnComputational Science**

AndComputationalIntelligence(Csci), Las Vegas, v. , n. , p. 99-104, dez. 2020. IEEE. http://dx.doi.org/10.1109/csci51800.2020.00024.
PAYA, Antonio; COTARELO, Alba; REDONDO, Jose Manuel. Egida: automated security configuration deployment systems with early error detection. **Computers & Security**, [S.L.], v. 116, p. 102638, maio 2022. Elsevier BV. http://dx.doi.org/10.1016/j.cose.2022.102638.

ROSE, Tina; ZHOU, Xiaobo. System Hardening for Infrastructure as a Service (IaaS). In: 2020 IEEE Systems Security Symposium (SSS). IEEE, 2020. p. 1-7.

SANTOS, Douglas; NOBRE, Jéferson Campos. Vulnerability Identification on GNU/Linux Operating Systems through Case-Based Reasoning. **Revista de InformáticaTeórica e Aplicada**, [S.L.], v. 26, n. 3, p. 13-25, 30 nov. 2019. Universidade Federal do Rio Grande do Sul. http://dx.doi.org/10.22456/2175-2745.82079.

SYNOPSYS (Eua). Black Duck Binary Analysis. 2023. Disponívelem: https://www.synopsys.com/software-integrity/software-composition-analysis-tools/binary-analysis.html. Acessoem: 13 nov. 2023.

THAKUR, Kutub; PATHAN, Al-Sakib Khan. **Cybersecurity Fundamentals: A Real-World Perspective**. CRC Press, 2020.

VARUN, C. P.; AGARWAL, Rashmi. Automation of Server Security Assessment. In: 2022 4th International Conference on Circuits, Control, Communication and Computing (I4C). IEEE, 2022. p. 515-518.

VULSIO. Gosto (go-security-tracker). 2023. Disponívelem: https://github.com/vulsio/gost. Acessoem: 13 nov. 2023.

ZHANG, Mengyuan; WANG, Lingyu; JAJODIA, Sushil; SINGHAL, Anoop. Network Attack Surface: lifting the concept of attack surface to the network level for evaluating networks⁚ resilience against zero-day attacks. **Ieee Transactions On Dependable And Secure Computing**, [S.L.], v. 18, n. 1, p. 310-324, 1 jan. 2021. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tdsc.2018.2889086.

ZLOTNIK, Oleg. System Hardening Guidelines for 2022: Critical Best Practices. 2021. Disponível em: https://www.hysolate.com/blog/system-hardening-guidelines-best-practices/. Acessoem: 01 dez. 2022.