Verificando a segurança computacional do servidor de uma organização

Daniel Carvalho de Oliveira[1], **Carlos Alberto de Oliveira**[2]

[1]Curso de Bacharelado em Ciência da Computação - Faculdade de Computação (FACOM) ,
[2] Faculdade de Computação (FACOM)
Universidade Federal do Mato Grosso do Sul (UFMS)
79070-900 – Campo Grande – MS – Brasil

daniel.carvalho@ufms.br, carlos.silva@ufms.br

Abstract. *This article presents an analysis of the security vulnerabilities present in a server of a private company, working in conjunction with a public agency in the state of Mato Grosso do Sul, Brazil. The vulnerabilities present on this server were identified and evaluated, with the aim of providing an overview of the security conditions and proposing mitigation and prevention measures. The analysis revealed the existence of several vulnerabilities, highlighting the importance of proactive approaches to ensure system protection and prevent potential cyber invasions.*

Resumo. *Este artigo apresenta uma análise das vulnerabilidades de segurança presentes em um servidor de uma empresa privada, trabalhando em conjunto com um órgão público no estado de Mato Grosso do Sul, Brasil. As vulnerabilidades presentes nesse servidor foram identificadas e avaliadas, com o objetivo de fornecer uma visão geral das condições de segurança e propor medidas de mitigação e prevenção. A análise revelou a existência de diversas vulnerabilidades, destacando a importância de abordagens proativas para garantir a proteção dos sistemas e evitar possíveis invasões cibernéticas.*

1. Introdução

A segurança cibernética é um tema de grande e crescente importância, devido à contínua e montante dependência de organizações e indivíduos em relação a sistemas computacionais. No contexto empresarial e público, servidores próprios podem ser essenciais, principalmente quando se pensa em segurança de dados. Embora a identidade da organização envolvida permaneça confidencial por razões de segurança, esta pesquisa tem como objetivo analisar as vulnerabilidades de segurança de acesso de um servidor. Neste cenário, foram utilizadas as ferramentas Nmap e OpenVAS.

A análise foi realizada com base nos seguintes objetivos específicos: identificar as vulnerabilidades de segurança do servidor da empresa e avaliar a criticidade das vulnerabilidades identificadas; produzir um relatório técnico das vulnerabilidades encontradas com seus respectivos níveis de criticidade, para que medidas corretivas possam ser tomadas, a fim de evitar que uma invasão cibernética ocorra e cause inconvenientes, como interrupções de serviços, perda de dados ou outras ações que danifiquem a reputação da entidade.

2. Fundamento Teórico

Esta seção apresenta os principais conceitos e trabalhos relacionados que fornecem embasamento teórico para o desenvolvimento deste estudo. A análise de vulnerabilidades realizada neste trabalho tem como base o sistema CVE (*Common Vulnerabilities and Exposures*) [1], um repositório padronizado de vulnerabilidades de segurança conhecidas publicamente. O CVE é mantido pela MITRE

Corporation, com apoio do Departamento de Segurança Interna dos Estados Unidos, e tem como objetivo fornecer uma nomenclatura comum para facilitar a identificação e o compartilhamento de informações sobre falhas de segurança.

Complementarmente, é utilizado o padrão CVSS (*Common Vulnerability Scoring System*) [2], um sistema de pontuação que classifica a gravidade das vulnerabilidades com base em diversas métricas técnicas, atribuindo um valor numérico entre 0 e 10. Essa pontuação permite avaliar o impacto potencial de uma vulnerabilidade de forma objetiva e padronizada.

Além dos conceitos técnicos, este trabalho também se apoia em estudos prévios com abordagens metodológicas semelhantes. Destacam-se, por exemplo, os trabalhos ”Análise de vulnerabilidades em domínios WordPress: um estudo de caso em uma universidade pública”, de Gabriel Pastorello de Oliveira [3], e ”Análise de vulnerabilidades em domínios”, de Raissa Rinaldi Yoshioka [4], que contribuíram para a definição da abordagem adotada na presente pesquisa.

3. Ferramentas

As ferramentas utilizadas nessa pesquisa são descritas em maiores detalhes nesta seção.

3.1. Kali Linux

Kali Linux [5] é uma distribuição Linux de código aberto, baseada em Debian Linux, anteriormente conhecida como BackTrack Linux. Ela permite que usuários executem testes avançados de penetração e auditoria de segurança. Essa distribuição contém centenas de ferramentas, configurações e *scripts* que permitem a usuários focarem em computação forense, engenharia reversa e detecção de vulnerabilidades. O Kali Linux pode ser implantado em diferentes configurações, incluindo instalação convencional em um dispositivo de armazenamento, execução em ambiente virtualizado ou utilização a partir de um dispositivo USB com uma *live image*. No presente estudo, o sistema operacional foi configurado em um ambiente de *dual boot* em um notebook, permitindo a coexistência com outro sistema previamente instalado e garantindo acesso direto ao hardware sem camadas de virtualização.

3.2. Nmap

Nmap (*Network Mapper*) [6] é uma ferramenta de código aberto para exploração de redes e auditoria de segurança. Foi projetado para escanear rapidamente grandes redes, mas funciona bem para *hosts* únicos. O Nmap usa pacotes IP brutos para determinar quais *hosts* estão disponíveis na rede, quais serviços esses *hosts* oferecem, quais sistemas operacionais estão executando, entre outras características. Normalmente, o Nmap é usado para auditorias de segurança, como no caso deste artigo, mas pode ser usado também para inventário de rede, cronograma de atualizações de serviços e monitoramento do tempo de atividade.

3.3. OpenVAS

OpenVAS [7] é uma ferramente de escaneamento de vulnerabilidades completo. Ele consegue executar testes autenticados e não autenticados, vários protocolos de internet, ajuste de desempenho para varreduras em grande escala e possui uma linguagem de programação interna para implementação de testes para as vulnerabilidades conhecidas e documentadas pelo CVE e CVSS. O OpenVAS obtém os testes de vulnerabilidade de um *feed* atualizado diariamente.

Ao identificar uma vulnerabilidade, o OpenVAS apresenta o NVT (*Network Vulnerability Test*) [8], um *script* executado para encontrar a vulnerabilidade e também informa um valor para a qualidade de detecção de cada vulnerabilidade, QoD (*Quality of Detection)* [9], onde normalmente vulnerabilidades com valor de QoD acima de 70% são consideradas confiáveis. Além disso, o OpenVAS também

traz o código CVE, quando existente, e o valor CVSS da vulnerabilidade, especificado em nível de severidade alto, médio ou baixo, de acordo com a seguinte escala:

- Alto: 7.0 - 10.0
- Médio: 4.0 - 6.9
- Baixo: 0.1 - 3.9

4. Metodologia

De acordo com o *X-Force Threath Intelligence Index* [10], um relatório anual elaborado pela equipe **IBM X-Force**, que reúne e analisa dados de segurança cibernética coletados por meio de centenas de bilhões de eventos monitorados em redes de clientes, bem como dados extraídos da *dark web* [11], *honeypots* [12] e incidentes reais de resposta a ataques, a exploração de vulnerabilidades é o segundo maior vetor de ataques cibernéticos depois de *phishing* [13]. O principal objetivo desse trabalho é encontrar vulnerabilidades no sistema analisado, de acordo com os seguintes passos:

1. **Definição do escopo e preparação do ambiente:** Inicialmente, definiu-se o escopo da análise, que compreendia um servidor específico da organização. Para isso, foi criado um ambiente de testes com a instalação do Kali Linux em uma máquina dedicada, onde foram configuradas e testadas as ferramentas Nmap e OpenVAS
2. **Varredura inicial com Nmap:** Após a configuração do ambiente, realizou-se uma varredura de rede utilizando o Nmap, com o objetivo de identificar os hosts ativos, serviços em execução, portas abertas e possíveis sistemas operacionais em uso.
3. **Identificação e escaneamento do servidor com OpenVAS:** Com base nas informações obtidas pelo Nmap, o IP do servidor alvo foi identificado. Em seguida, foi realizada uma varredura detalhada com o OpenVAS, que gerou um relatório técnico contendo as vulnerabilidades encontradas, suas classificações CVSS, QoD e, quando disponíveis, os respectivos códigos CVE.
4. **Análise dos resultados:** Os dados coletados pelo OpenVAS foram cuidadosamente analisados. As vulnerabilidades foram classificadas quanto à sua criticidade e descritas tecnicamente, com base nos padrões CVSS e CVE.
5. **Proposição de medidas de mitigação:** Por fim, com base nas vulnerabilidades identificadas, foram sugeridas ações corretivas e preventivas, visando à redução dos riscos e a implantação de novas políticas de segurança para o servidor.

5. Resultados

Nesta seção, são apresentados os resultados obtidos a partir das análises realizadas com as ferramentas Nmap e OpenVAS sobre o servidor avaliado. As vulnerabilidades encontradas foram identificadas pelos NVT's (*Network Vulnerability Test*) usados para encontrá-las, classificadas com base no sistema CVSS (*Common Vulnerability Scoring System*) e, sempre que possível, associadas aos códigos CVE (*Common Vulnerabilities and Exposures*). A Tabela 1 resume as principais vulnerabilidades confirmadas, com suas respectivas classificações de severidade e indicadores de confiabilidade (QoD – *Quality of Detection*).

Tabela 1. Tabela das Vulnerabilidades Confirmadas

NVT	QoD	CVE	CVSS
DCE/RPC and MSRPC Services Enumeration Reporting	80%	-	Médio (5.0)
Cleartext Transmission of Sensitive Information via HTTP	80%	-	Médio (4.8)
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	98%	CVE-2011-3389 / CVE-2015-0204	Médio (4.3)
TCP Timestamps Information Disclosure	80%	-	Baixo (2.6)
ICMP Timestamp Reply Information Disclosure	80%	CVE-1999-0524	Baixo (2.1)

Uma breve descrição das vulnerabilidades será descrita a seguir, incluindo suas possíveis consequências e recomendações de mitigação.

5.1. DCE/RPC and MSRPC Services Enumeration Reporting

Essa vulnerabilidade afeta serviços DCE/RPC (*Distributed Computing Environment / Remote Procedure Call*) e MSRPC [14] que estão em execução no *host* remoto, podendo ser enumerados por meio de conexões à porta TCP 135 [15] . Essa exposição permite que um atacante obtenha informações detalhadas sobre os serviços em execução, ampliando o conhecimento sobre a infraestrutura e facilitando a elaboração de ataques direcionados [16].
Recomendação: Restringir o acesso externo à porta 135/TCP utilizando regras de *firewall* [17] ou segmentação de rede, limitando o escopo apenas a sistemas autorizados.

5.2. Cleartext Transmission of Sensitive Information via HTTP

Esse vulnerabilidade está relacionada a transmissão de informações sensíveis usando texto simples, como nomes de usuários e senhas, através do protocolo HTTP [18]. Essa prática torna a comunicação vulnerável a interceptações, especialmente em ataques do tipo *man-in-the-middle* [19] .
Recomendação: Implementar o uso obrigatório de conexões seguras (HTTPS), com certificados digitais válidos e atualizados, assegurando a criptografia de dados sensíveis em trânsito.

5.3. SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection

Essa vulnerabilidade afeta o protocolo TLS [20] versão 1.0 e 1.1, considerados obsoletos e inseguros por apresentarem fraquezas conhecidas. Essas configurações podem expor a comunicação a ataques que exploram falhas criptográficas, como o BEAST (CVE-2011-3389) e o FREAK (CVE-2015-0204) [21].
Recomendação: Desabilitar os protocolos TLSv1.0 e TLSv1.1, configurando os serviços para aceitarem apenas versões mais recentes e seguras, como o TLSv1.2 ou TLSv1.3.

5.4. TCP Timestamps Information Disclosure

Essa vulnerabilidade atinge o uso de carimbos de tempo TCP (TCP *timestamps*), permitindo que atacantes estimem o tempo de atividade do sistema [22]. Essa informação pode ser utilizada para inferir padrões de uso ou identificar alvos vulneráveis para ataques de reconhecimento.
Recomendação: Desabilitar a funcionalidade de TCP *timestamps* nas configurações de rede do sistema operacional sem comprometer a funcionalidade de aplicações críticas.

5.5. ICMP Timestamp Reply Information Disclosure

Essa vulnerabilidade impacta solicitações ICMP [23] de carimbo de data e hora (*ICMP Timestamps*), as quais disponibilizam informações passíveis de exploração em ataques relacionados à previsibilidade temporal ou à análise de rede [24]. Tal prática também auxilia no mapeamento do sistema por invasores.
Recomendação: Restringir ou desativar respostas a pacotes ICMP do tipo *timestamp* por meio de regras no *firewall* ou nas configurações do sistema.

6. Conclusão

A realização desta pesquisa permitiu identificar e classificar diversas vulnerabilidades presentes em um servidor, demonstrando na prática a importância da análise preventiva de segurança em ambientes computacionais. Através da utilização das ferramentas Nmap e OpenVAS, foi possível obter uma visão clara das fragilidades existentes, com base em padrões reconhecidos como CVE e CVSS. As vulnerabilidades detectadas, embora classificadas como de criticidade média ou baixa, evidenciam brechas que podem ser exploradas por atacantes e, portanto, não devem ser ignoradas.

A correção ou mitigação dessas falhas é essencial para reforçar a segurança da organização, prevenir incidentes cibernéticos e garantir a integridade, confidencialidade e disponibilidade das informações. Este estudo reforça a relevância de boas práticas de auditoria de segurança e da adoção de protocolos atualizados e configurações adequadas de *firewall*. A empresa responsável pelo servidor foi informada das falhas encontradas no sistema, e as medidas possíveis de mitigação foram executadas.

Concluímos, diante do crescente número de ameaças no cenário digital atual, a aplicação de ferramentas automatizadas de varredura de vulnerabilidades, aliada a uma análise crítica e contextualizada dos resultados, é uma etapa fundamental no ciclo de segurança da informação em qualquer organização.

Referências

[1] MITRE Corporation. *CVE - FAQs and resources*. URL: https://www.cve.org/ResourcesSupport/FAQs. Acessado: 10.06.2025.

[2] National Institute of Standards and Technology. *CVSS metrics – National Vulnerability Database*. URL: https://nvd.nist.gov/vuln-metrics/cvss. Acessado: 10.06.2025.

[3] Gabriel Pastorello de Oliveira e Carlos Alberto da Silva. "Analise de vulnerabilidades em domínios WordPress: um estudo de caso em uma universidade publica". Trabalho de Conclusão de Curso. Universidade Federal do Mato Grosso do Sul - UFMS, 2023.

[4] Raissa Rinaldi Yoshioka e Carlos Alberto da Silva. " Análise de Vulnerabilidades em domínios". Trabalho de Conclusão de Curso. Universidade Federal do Mato Grosso do Sul - UFMS, 2024.

[5] Offensive Security. *Kali Linux - Frequently Asked Questions (FAQ)*. URL: https://www.kali.org/faq/. Acessado: 10.06.2025.

[6] Gordon Lyon. *Nmap reference guide*. URL: https://nmap.org/book/man.html. Acessado: 10.06.2025.

[7] Greenbone Networks. *GSM – Manual do Greenbone Security Manager (GOS 24.10)*. URL: https://docs.greenbone.net/GSM-Manual/gos-24.10/en. Acessado: 10.06.2025.

[8] Greenbone Networks. *GSM – Manual do Greenbone Security Manager (GOS 24.10) - NVT - Glossary*. URL: https://docs.greenbone.net/GSM-Manual/gos-24.10/en/glossary.html#vulnerability-test-vt. Acessado: 10.06.2025.

[9] Greenbone Networks. *GSM – Manual do Greenbone Security Manager (GOS 24.10) - 10.2.6 Quality of Detection Concept*. URL: https://docs.greenbone.net/GSM-Manual/gos-24.10/en/reports.html#quality-of-detection-concept. Acessado: 10.06.2025.

[10] IBM. *Vulnerability scanning – IBM Security*. URL: https://www.ibm.com/think/topics/vulnerability-scanning. Acessado: 10.06.2025.

[11] Adam Volle. *Dark web — Definition, The Onion Router, History, Examples*. URL: https://www.britannica.com/technology/dark-web. Acessado: 10.06.2025.

[12] Klaus Steding-Jessen e Marcelo H. P. C. Chaves Cristine Hoepers. *Honeypots e Honeynets: Definições e Aplicações*. URL: https://www.cert.br/docs/whitepapers/honeypots-honeynets/. Acessado: 10.06.2025.

[13] Matthew Kosinski. *O que é phishing?* URL: https://www.ibm.com/br-pt/think/topics/phishing. Acessado: 10.06.2025.

[14] Ben Barnea. *Uma visão geral do MS-RPC e seus mecanismos de segurança*. URL: https://www.akamai.com/pt/blog/security-research/msrpc-security-mechanisms. Acessado: 10.06.2025.

[15] Computer Security Resource Center (CSRC). *Transmission Control Protocol (TCP) - Glossary — CSRC*. URL: https://csrc.nist.gov/glossary/term/transmission_control_protocol. Acessado: 10.06.2025.

[16] Greenbone Networks. *DCE/RPC and MSRPC Services Enumeration Reporting*. URL: https://secure1.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.10736. Acessado: 10.06.2025.

[17] Computer Security Resource Center (CSRC). *firewall - Glossary — CSRC*. URL: https://csrc.nist.gov/glossary/term/firewall. Acessado: 10.06.2025.

[18] MITRE Corporation. *CWE-319: Cleartext transmission of sensitive information via HTTP*. URL: https://cwe.mitre.org/data/definitions/319.html. Acessado: 10.06.2025.

[19] Greeg Lindemulder e Matthew Kosinski. *"What Is a Man-In-The-Middle (MITM) Attack? — IBM*. URL: www.ibm.com/think/topics/man-in-the-middle. Acessado: 10.06.2025.

[20] Andrei Popov. *Protocolo TLS*. URL: https://learn.microsoft.com/pt-br/windows-server/security/tls/transport-layer-security-protocol. Acessado: 10.06.2025.

[21] Greenbone Networks. *SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection*. URL: https://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.117274. Acessado: 10.06.2025.

[22] Michel Arboi. *TCP Timestamps Information Disclosure*. URL: https://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.80091. Acessado: 10.06.2025.

[23] Nikolay Kartashev. *Entender as mensagens de redirecionamento ICMP*. URL: https://www.cisco.com/c/pt_br/support/docs/ios-nx-os-software/nx-os-software/213841-understanding-icmp-redirect-messages.html. Acessado: 10.06.2025.

[24] MITRE Corporation. *CVE-1999-0524*. URL: https://www.cve.org/CVERecord?id=CVE-1999-0524. Acessado: 10.06.2025.