Overview de um estudo de caso com as ferramentas Beelzebub, OpenVAS e DefectDojo

Keywords: Beelzebub, OpenVAS, DefectDojo, honeypot, escaneamento, CVE
Autora: Katia Kaori Kaminishikawahara

Sumário

1. Introdução

O artigo tem como objetivo demonstrar o uso das ferramentas Beelzebub, OpenVAS e DefectDojo e fazer um overview de suas funcionalidades, fornecendo uma opção de *honeypot* que pode ser desenvolvida conforme a necessidade dos diferentes testes de vulnerabilidades que queiram ser executados.

O Beelzebub destaca-se como uma ferramenta capaz de construir *honeypots* personalizados, ou seja, armadilhas que simulam sistemas reais com o objetivo de atrair, registrar e analisar atividades maliciosas. Esses *honeypots* não servem apenas como mecanismos de defesa proativa, mas também geram dados valiosos que auxiliam na compreensão do comportamento dos atacantes.

Além disso, o Beelzebub pode ser integrado com o OpenVAS (*Open Vulnerability Assessment System*), um scanner automatizado para identificar falhas de segurança em

sistemas e aplicações, sendo essencial para times técnicos durante o desenvolvimento de soluções seguras.

Complementando essa cadeia de segurança, o DefectDojo atua como uma plataforma de gerenciamento de vulnerabilidades, facilitando o registro, triagem e acompanhamento das falhas detectadas. Ele centraliza os resultados gerados pelo escaneamento do OpenVAS nos *honeypots* do Beelzebub, fornecendo uma visão clara dos riscos, promovendo uma abordagem colaborativa e contínua.

2. Pré-requisitos

2.1. Ferramenta: Beelzebub

O Beelzebub é uma ferramenta de segurança cibernética desenvolvida para facilitar a criação e gestão de *honeypots*. Uma de suas principais funcionalidades é a capacidade de construir *honeypots* personalizados e interativos, simulando serviços reais para observar o comportamento de atacantes em tempo real. Além disso, para contextos avançados, ele pode se integrar com modelos de linguagem (LLMs – *Large Language Models*) como o Ollama, adicionando camadas de interação mais realistas (CANDELA, 2024).

Requisitos Mínimos:

- **Sistema Operacional:** Linux;
- **Memória RAM:** Mínimo de 512 MB;
- **Espaço em Disco:** Pelo menos 100 MB livres.

O passo a passo para sua instalação pode ser obtido pelo *link* (CANDELA, 2024):

https://github.com/mariocandela/beelzebub

2.2. Ferramenta: OpenVAS

De acordo com o *Greenbone Networks* (2025a), o OpenVAS é uma ferramenta *open source* para escaneamento e avaliação de vulnerabilidades em redes, sistemas e aplicações. Permite identificar falhas de segurança que podem ser exploradas por invasores. Além disso, conta com os seguintes recursos:

- Realiza varreduras automatizadas e agendadas;
- Avalia vulnerabilidades conhecidas com uma base de dados constantemente atualizadas;

- Gera relatórios detalhados e orientações para mitigação;
- Integra outras ferramentas de segurança e gerenciamento de vulnerabilidades.

Requisitos Mínimos:

- **Sistema Operacional:** Linux;
- **CPU:** Processador multi-core (recomendado 2 cores ou mais);
- **Memória RAM:** Pelo menos 4 GB (recomendado 8 GB para ambientes maiores);
- **Espaço em Disco:** Mínimo de 20 GB para a instalação e armazenamento de dados;
- **Dependências:** PostegreSQL, Redis e outros pacotes de sistema conforme documentação oficial.

Sua instalação pode ser realizada acompanhando o passo a passo do link *(Greenbone Networks,* 2025b*)*:

https://greenbone.github.io/docs/latest/22.4/container/index.html

2.3. Ferramenta: DefectDojo

O DefectDojo é uma plataforma *open source* que auxilia no gerenciamento de vulnerabilidades. Ele centraliza os resultados de diferentes scanners, organiza as vulnerabilidades encontradas, permite a triagem, priorização e o acompanhamento do ciclo de vida das vulnerabilidades mapeadas.

Além disso, a ferramenta é amplamente utilizada para integrar testes automatizados, facilitar auditorias e melhorar a governança de segurança dentro de processos de desenvolvimento seguro (OWASP, 2025).

Requisitos Mínimos:

- **Sistema Operacional:** Linux;
- **Python:** Versão 3.8 ou superior;
- **Banco de Dados:** PostgreSQL;
- **Servidor Web:** Nginx ou Apache;
- **Dependências:** Docker e Docker Compose são frequentemente usados para facilitar a instalação e gerenciamento;
- **Memória RAM:** Pelo menos 2 GB (recomendado 4 GB para uso mais intensivo);
- **Espaço em Disco**: Mínimo de 10 GB para armazenamento dos dados e relatórios.

As instruções para a sua instalação podem ser verificadas no link (DEFECTDOJO, 2025):

https://docs.defectdojo.com/en/open_source/installation/installation/

3. Configuração e Criação do Honeypot

Após a instalação da ferramenta, foi configurado um arquivo yaml (Figura 1), com o objetivo de simular um serviço HTTP vulnerável, mais especificamente um software personalizado ou inseguro para ser identificado pelo scanner OpenVAS.

Onde foi especificado dados como: a porta na qual o Beelzebub deve escutar, um banner de resposta básico, os manipuladores de requisições HTTP e outras variáveis.

```yaml
# Nome da configuração do honeypot — normalmente usado para identificar o serviço simulado
name: cve

# Descrição do honeypot — explicando que ele simula um serviço HTTP vulnerável
description: Simulação de serviço HTTP vulnerável

# Lista de serviços que o Beelzebub deve escutar
listen:
 - port: 8080
   protocol: tcp
   type: http

   # Banner de resposta básico, simulando um serviço API REST com JSON
   banner:
    server: VulnerableService/1.2.3
    content-type: application/json
    body: '{"status": "ok", "version": "1.2.3"}'

# Manipuladores de requisições HTTP — definem como o honeypot responde
handlers:
 - type: http

   # Banner exibido para enganar ferramentas de varredura
   banner: "VulnerableService/1.2.3 CVE"

   # Cabeçalhos de resposta HTTP simulados
   response_headers:
    Server: "VulnerableService/1.2.3"
    Content-Type: text/html

   # Corpo da resposta HTTP simulando uma página vulnerável
   response_body: |
    <html>
     <head><title>Vulnerable Service</title></head>
     <body>
        <h1>VulnerableService version 1.2.3</h1>
        <p>This service is vulnerable to CVE</p>
     </body>
    </html>

   # Métodos HTTP aceitos pelo honeypot
   methods:
    - GET
    - HEAD
```

Fonte: Autora, 2025

Figura 1. Arquivo yaml desenvolvido para criar um honeypot com vulnerabilidade HTTP no Beelzebub.

4. Configuração de Escaneamento do OpenVAS

Para o escaneamento do *honeypot* criado com o Beelzebub, a seguinte configuração foi testada:

- **Configurações da *Task*:**
  - ***Name:*** Scan_CVE2025
  - ***Target:*** Target_CVE2025
  - **Scan Config:** *Full and fast*
  - **Min QoD:** 80%

- **Configurações do *Target*:**
  - ***Name:*** Target_CVE2025
  - ***Host:*** IP da rede em que se encontra o *honeypot*
  - ***Alive Test:*** *Consider Alive*
  - ***Port List:*** *All IANA assigned TCP*

- **Resultados:**
  - **Tempo de duração do escaneamento:** 14 minutos
  - ***Applications:***
    - *OpenSSH*
    - *Secure Shell Protocol*
    - *HTTP Server*
  - **Sistema Operacional:**
    - *Linux Kernel*
  - **Portas identificadas:**
    - 22
    - 2222
    - 8080
    - 8181
    - 2112
    - 3112
    - 3306
    - 3336
  - **Vulnerabilidades mapeadas:**
    - *SSH Brute Force Logins With Default Credentials Reporting;*
    - *Cleartext Transmission of Sensitive Information via HTTP;*

        - *Weak MAC Algorithm(s) Supported (SSH)*
        - *TCP Timestamps Information Disclosure*
        - *ICMP Timestamp Reply Information Disclosure*
    - **CVEs:**
        - Relacionados a *SSH Brute Force Logins With Default Credentials Reporting:*
            - CVE-1999-0501, CVE-1999-0502, CVE-1999-0507, CVE-1999-0508, CVE-2005-1379, CVE-2006-5288, CVE-2009-3710, CVE-2012-4577, CVE-2016-1000245, CVE-2017-16523, CVE-2020-29583, CVE-2024-22902, CVE-2024-31970 e CVE-2024-46328
        - *ICMP Timestamp Reply Information Disclosure:*
            - CVE-1999-0524
        - *IP Forwarding Enabled Active Check:*
            - CVE-1999-0511

Foi adotada a configuração de scan config (*Full and fast*), a fim de realizar uma varredura completa e rápida, buscando identificar o máximo possível de vulnerabilidades em menor tempo, equilibrando o detalhamento com a velocidade.

Quanto à Min QoD (Qualidade de Detecção) foi adotado um valor de 80%, com a intenção de reduzir falsos positivos, trazendo apenas as vulnerabilidades que o sistema considera confiáveis.

O *Test Alive* do tipo *Consider Alive*, assumiu que o host estava ativo sem precisar testar com pacotes ICMP (*Internet Control Message Protocol*) ou TCP (*Transmission Control Protocol*) preliminares. Isso acelerou o processo de varredura, porém há o risco de escanear hosts inativos.

Para o *Port List* foi atribuído todas as portas TCP consideradas pelo IANA, que abrangem tanto serviços comuns quanto serviços incomuns (IANA, 2025).

4.1. Interpretação dos Resultados do OpenVAS

O tempo de duração de 14 minutos é um indicativo de escaneamento rápido e eficiente para um escopo definido.

As aplicações detectadas como o *OpenSSH* e o *Secure Shell Protocol*, mostram a presença de serviços SSH e o *HTTP Server*, mostrando que os serviços web estão ativos.

Quanto as vulnerabilidades encontradas e possíveis mitigações:

- ***SSH Brute Force Logins With Default Credentials Reporting:*** criticidade alta, ataque de força bruta tentando acessar o sistema via SSH usando credenciais padrão.
    - Mitigação:
        - Desabilitar logins com credenciais padrão;
        - Usar autenticação por chaves públicas;
        - Implementar bloqueio após tentativas falhas;
        - Monitorar logs e usar sistemas de detecção de intrusão.

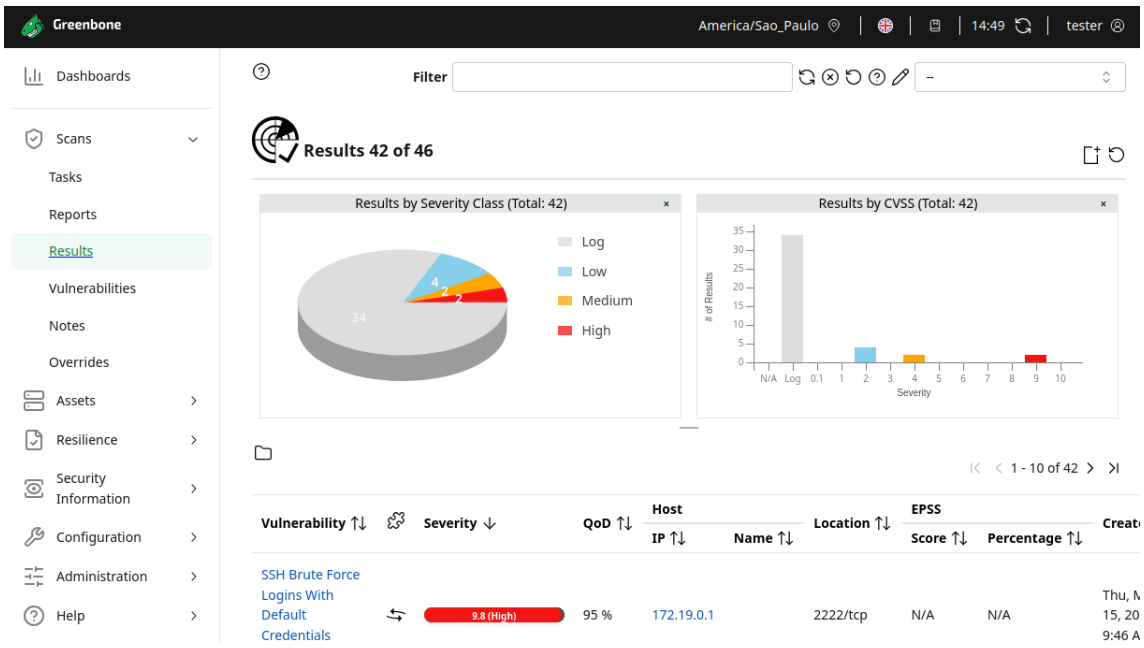
- ***Cleartext Transmission of Sensitive Information via HTTP:*** criticidade média, dados sensíveis são transmitidos sem criptografia via HTTP.
    - Mitigação:
        - Usar HTTPS (TLS/SSL) para proteger a comunicação;
        - Configurar o redirecionamento automático de HTTP para HTTPS.

- ***Weak MAC Algorithm(s) Supported (SSH):*** criticidade baixa, uso de algoritmos de integridade criptográfica considerados fracos no SSH.
    - Mitigação:
        - Atualizar as configurações SSH para usar algoritmos MAC modernos e seguros;
        - Atualizar o software para versões que suportem algoritmos mais robustos.

- ***TCP Timestamps Information Disclosure:*** criticidade baixa, o servidor revela *timestamps* TCP que podem ser usados para ataques de *fingerprinting* ou cálculo do *uptime*.
    - Mitigação:
        - Desabilitar os *timestamps* TCP no sistema operacional;
        - Usar *firewalls* para limitar o acesso.

- ***ICMP Timestamp Reply Information Disclosure:*** criticidade baixa, respostas ICMP podem revelar informações sobre o tempo do sistema.
    - Mitigação:
        - Configurar o *firewall* para bloquear ou limitar respostas ICMP;
        - Ajustar as configurações do sistema para não responder a solicitações de *timestamps*.

A Figura 2, mostra o *dashboard* de vulnerabilidades gerada no OpenVAS após o escaneamento do *honeypot*. Foram no total 42 vulnerabilidades mapeadas, sendo 2 *High*, 2 *Medium*, 4 *Low* e 34 registros. Esses registros não indicam uma vulnerabilidade propriamente dita, mas sim um evento registrado ou informação técnica detectada durante um escaneamento. Ele não afeta diretamente a pontuação de risco, mas são úteis para o entendimento da superfície de ataque ou validar que o ambiente está exposto como o esperado *(honeypots*) (GREENBONE, 2025c).



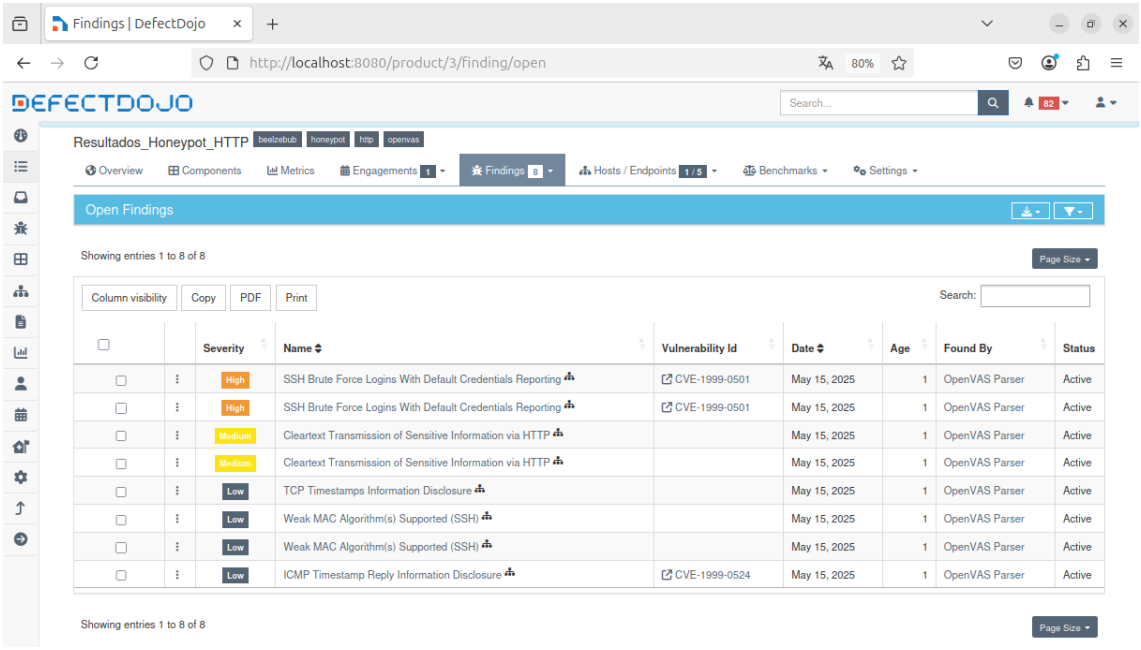
Fonte: Autora, 2025

Figura 2. Dashboard de vulnerabilidade gerado a partir do escaneamento do *honeypot* criado pela ferramenta Beelzebub.

5. Resultados no DefectDojo

O resultado do escaneamento realizado no OpenVAS foi importado para o DefectDojo, fornecendo um overview de todas as vulnerabilidades mapeadas.

Na Figura 3, o dashboard de *Open Findings* (vulnerabilidades abertas), traz os dados do grau de severidade da vulnerabilidade identificada, seu nome associado ao CVE, a data de escaneamento, a idade em dias da vulnerabilidade (enquanto sem mitigação), o scanner utilizado e o status de atividade (se mitigado ou não).

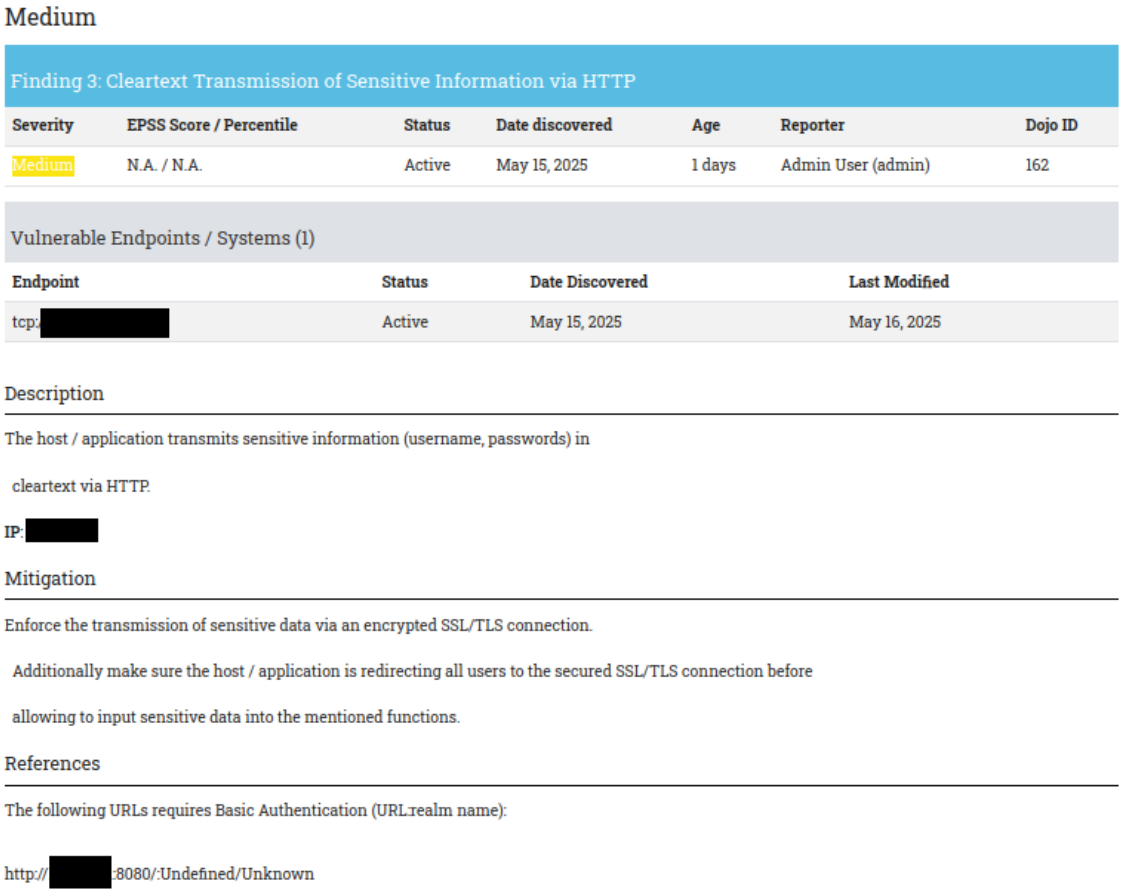


Fonte: Autora, 2025

Figura 3. Dashboard de *Open Findings* do DefectDojo, fornecendo uma visão geral dos **resultados obtidos pelo OpenVAS.**

A ferramenta ainda fornece uma variedade de tipos de relatórios, um exemplo é o relatório de *Product Security Report* (Figura 4), que fornece os dados de cada vulnerabilidade, uma breve descrição de sua característica, a mitigação, referências técnicas e em alguns casos o impacto.

No exemplo da Figura 4, trata da vulnerabilidade de criticidade média, *Cleartext Transmission of Sensitive Information via HTTP*, mapeada pelo OpenVAS no *honeypot* criado no Beelzebub, na qual os dados sensíveis são transmitidos sem criptografia via HTTP. O relatório fornece uma tabela com o grau de severidade, o status, data na qual foi mapeada, idade em dias de atividade, quem reportou e seu ID dentro do DefectDojo. Além do fornecimento de uma descrição breve da vulnerabilidade, forma de mitigação e referências.



Medium

Finding 3: Cleartext Transmission of Sensitive Information via HTTP

Severity	EPSS Score / Percentile	Status	Date discovered	Age	Reporter	Dojo ID
Medium	N.A. / N.A.	Active	May 15, 2025	1 days	Admin User (admin)	162

Vulnerable Endpoints / Systems (1)

Endpoint	Status	Date Discovered	Last Modified
tcp:[redacted]	Active	May 15, 2025	May 16, 2025

Description

The host / application transmits sensitive information (username, passwords) in cleartext via HTTP.

IP: [redacted]

Mitigation

Enforce the transmission of sensitive data via an encrypted SSL/TLS connection.

Additionally make sure the host / application is redirecting all users to the secured SSL/TLS connection before allowing to input sensitive data into the mentioned functions.

References

The following URLs requires Basic Authentication (URL:realm name):

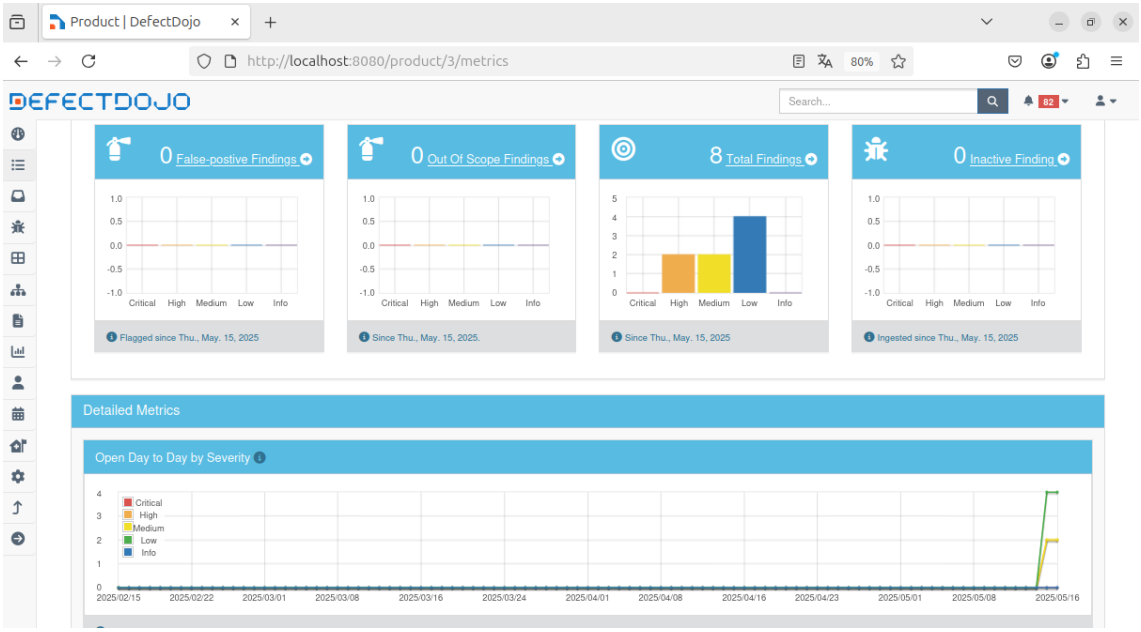
http://[redacted]:8080/:Undefined/Unknown

Fonte: Autora, 2025

Figura 4. *Product Security Report* gerado pela ferramenta DefectDojo, trazendo informações de descrição, mitigação e informações gerais.

Outra funcionalidade interessante no DefectDojo, é a possibilidade de organizar os dados de varredura em forma de métricas (gráficos).

Na Figura 5, são criados vários tipos de gráficos, como o *Total Findings* (Total de vulnerabilidades mapeadas) e o tempo que a vulnerabilidade está em aberto.



Fonte: Autora, 2025

Figura 5. Dashboard de métricas do honeypot com gráficos, por exemplo de, falsos positivos, total de vulnerabilidades mapeadas e o tempo em que se encontra sem mitigação.

6. Considerações Finais

O objetivo em simular um serviço HTTP vulnerável e detectável pelo OpenVAS foi alcançado, permitindo que a varredura identificasse aplicações, portas e vulnerabilidades gerais. Evidenciados, pela identificação de um servidor HTTP em funcionamento, detecção de portas típicas de serviços expostos (exemplo: 8080, 8181 e 22), reconhecimento de protocolos como SSH e HTTP e apontamento de vulnerabilidades genéricas (transmissão de dados em texto sem criptografia via HTTP, algoritmos fracos de autenticação SSH e informações sensíveis reveladas por *timestamps* e respostas ICMP).

A importação dos dados no DefectDojo, permitiu ter uma visão mais clara dos dados mapeados organizando-os em forma de gráficos, tabelas e possibilitando a geração de relatórios, como o do tipo *Product Security Report*.

7. Bibliografia

CANDELA, M. *Beelzebub*. GitHub, 2024. Disponível em: https://github.com/mariocandela/beelzebub. Acesso em: 16 maio 2025.

DEEFECTDOJO. *Installation (Open-Source)*, 2025. Disponível em: https://docs.defectdojo.com/en/open_source/installation/installation/. Acesso em: 17 maio 2025.

GREENBONE NETWORKS. *Greenbone Community Containers.* 2025b. Disponível em: https://greenbone.github.io/docs/latest/22.4/container/index.html. Acesso em: 17 maio de 2025.

GREENBONE NETWORKS. *Greenbone Vulnerability Manager (GVM) Documentation.* 2025a. Disponível em: https://greenbone.github.io/docs/. Acesso em: 16 maio 2025.

GREENBONE NETWORKS. *Greenbone Community Edition – Documentation.* 2025c. Disponível em: https://docs.greenbone.net. Acesso em: 17 maio 2025.

IANA - INTERNET ASSIGNED NUMBERS AUTHORITY. *Service Name and Transport Protocol Port Number Registry, 2025*. Disponível em: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml. Acesso em: 17 maio 2025.

OWASP. *OWASP DefectDojo Project*. Disponível em: https://owasp.org/www-project-defectdojo/. Acesso em: 15 maio 2025.