

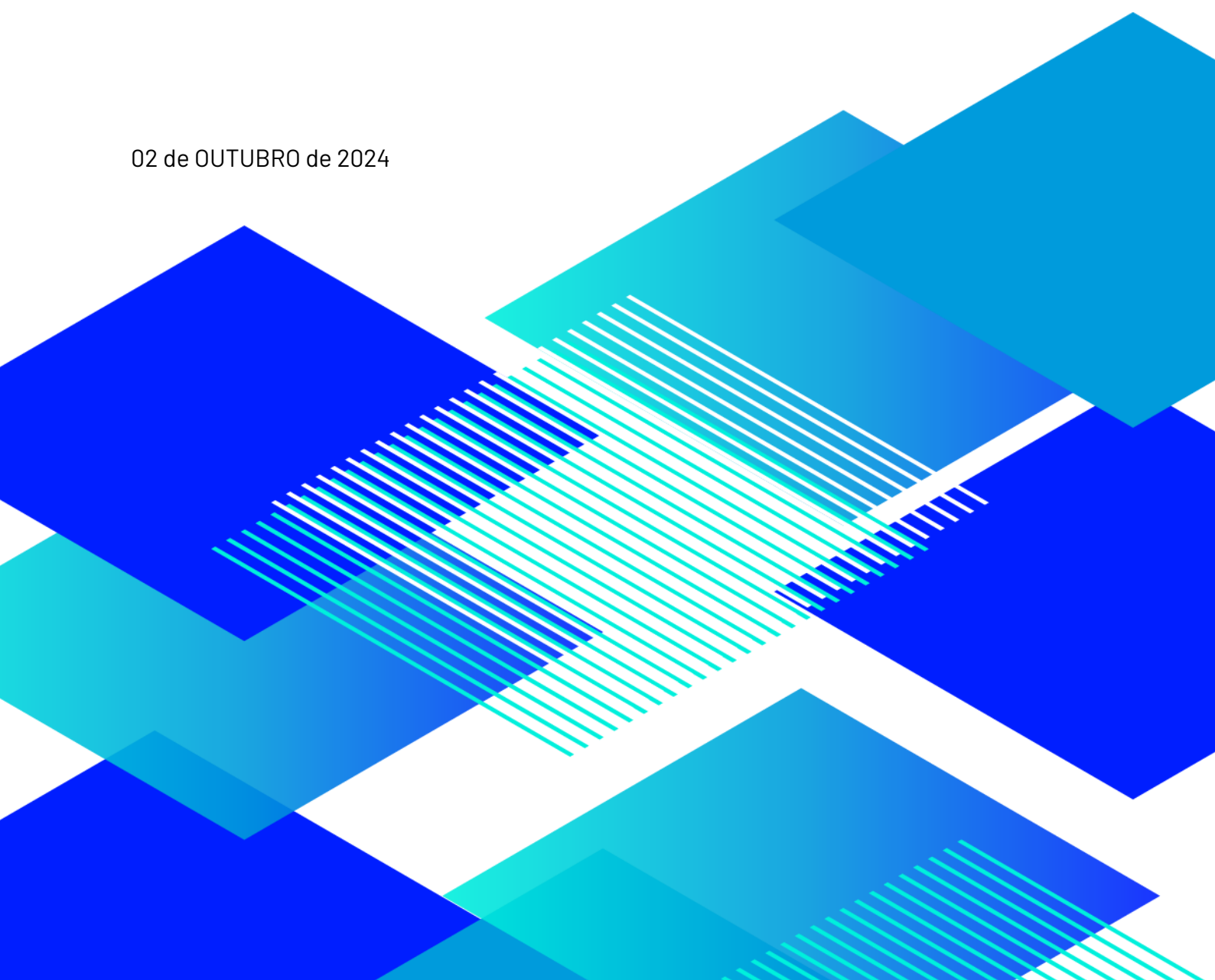


Inteligência em  
Cibersegurança

# Gestão de Vulnerabilidades com Ferramentas Abertas

**Matheus Camargo**

02 de OUTUBRO de 2024





## Sobre a RNP

Somos a rede brasileira para educação e pesquisa. Disponibilizamos internet segura e de alta capacidade, serviços personalizados e promovemos projetos de inovação. Nosso sistema inclui universidades, institutos educacionais e culturais, agências de pesquisa, hospitais de ensino, parques e polos tecnológicos. Com isso, beneficiamos 4 milhões de alunos, professores e pesquisadores brasileiros. Fomos os pioneiros, ao trazer a internet para o Brasil, e hoje nossa rede chega a todas as unidades da federação. Também estamos conectados às demais redes de educação e pesquisa na América Latina, América do Norte, África, Europa, Ásia e Oceania por meio de cabos de fibra óptica terrestres e submarinos. Somos qualificados como uma organização social vinculada ao Ministério da Ciência, Tecnologia e Inovações (MCTI) e mantida por esse, em conjunto com os ministérios da Educação (MEC), das Comunicações (MCom), Turismo, Saúde (MS) e Defesa (MD), que participam do Programa Interministerial RNP (PRO-RNP).

## Sobre o CAIS

Na RNP, temos o papel de zelar pela segurança da nossa rede e das instituições conectadas. Com esse objetivo, surgiu o Centro de Atendimento a Incidentes de Segurança – CAIS. Com 25 anos de atuação, o CAIS foi um dos primeiros grupos de resposta a incidentes de segurança a atuar em nível nacional na detecção, resolução e prevenção de incidentes que trafegam pela rede acadêmica e suas instituições usuárias.

## Sobre esta oficina

Esta oficina foi desenvolvida pelo CAIS para contribuir com a comunidade de segurança, demonstrando teorias e práticas acerca do processo de gestão de vulnerabilidades e como ele pode ser implementado inicialmente com ferramentas abertas.

O material desta oficina inclui este arquivo PDF com o conteúdo teórico e um arquivo OVA para virtualização de VM em ambiente Virtual Box.

A máquina virtual está com criptografia em disco. A chave para desbloqueio é `ubuntu`.

Para logar na VM, o usuário e senha são: `ubuntu/ubuntu`.

O login na aplicação OpenVAS é feito em <https://127.0.0.1:9392>.

Credenciais para OpenVAS são: `admin/ubuntu`.

O login na aplicação DefectDojo é feito em <https://127.0.0.1:8080>.

Credenciais para DefectDojo: `admin:Senha123!@#`.



## Sumário

<b>1. Gestão de Vulnerabilidades Técnicas com Ferramentas Abertas .....</b>	<b>4</b>
1.1. <i>Sobre a Gestão de Vulnerabilidades Técnicas .....</i>	5
1.2. <i>Gestão de Vulnerabilidades: Ativos .....</i>	6
1.3. <i>Gestão de Vulnerabilidades: Escopo .....</i>	8
1.4. <i>Gestão de Vulnerabilidades: Ciclo de Vida .....</i>	9
1.5. <i>Gestão de Vulnerabilidades: Aferir .....</i>	10
1.6. <i>Gestão de Vulnerabilidades: Priorizar .....</i>	11
1.7. <i>Gestão de Vulnerabilidades: Remediar .....</i>	12
1.8. <i>Gestão de Vulnerabilidades: Monitorar .....</i>	13
1.9. <i>Alguns Benefícios do Processo de Gestão de Vulnerabilidades .....</i>	14
1.10. <i>Gestão de Vulnerabilidades – Análise de Vulnerabilidades Técnicas .....</i>	15
<b>2. OpenVAS e Defect Dojo .....</b>	<b>16</b>
2.1. <i>OpenVAS .....</i>	16
2.2. <i>OpenVAS Instalação .....</i>	17
2.3. <i>Defect Dojo .....</i>	18
2.4. <i>Defect Dojo – Instalação .....</i>	19
<b>3. Utilizando OpenVAS .....</b>	<b>20</b>
3.1. <i>OpenVAS - Configurando Novas Credenciais .....</i>	21
3.2. <i>OpenVAS - Configurando Novas Listagens de Portas .....</i>	22
3.3. <i>OpenVAS - Configurando Novos Agendamentos .....</i>	23
3.4. <i>OpenVAS - Configurando Novos Alvos .....</i>	25
3.5. <i>OpenVAS – Configurando Novas Tarefas .....</i>	26
3.6. <i>OpenVAS – Executando Scans e Gerando Relatórios .....</i>	27
<b>4. Utilizando Defect Dojo .....</b>	<b>28</b>
4.1. <i>DefectDojo – Entendendo sua Estrutura .....</i>	28
4.2. <i>DefectDojo – Configurando Tipos de Produto .....</i>	29
4.3. <i>DefectDojo – Configurando Novos Produtos .....</i>	30
4.4. <i>DefectDojo – Importando Relatórios do OpenVAS via Interface Web .....</i>	31
4.5. <i>DefectDojo – Importando Relatórios do OpenVAS via API .....</i>	33
4.6. <i>DefectDojo – Obtendo Métricas e Estatísticas .....</i>	34



# 1. Gestão de Vulnerabilidades Técnicas com Ferramentas Abertas



Olá! Este é o treinamento de Gestão de Vulnerabilidades Técnicas com Ferramentas Abertas, apresentado na Semana de Capacitação do Nic.br de 2024. Meu nome é Matheus Camargo e eu te guiarei hoje, apresentando um pouco de teoria e prática sobre a gestão de Vulnerabilidades Técnicas.



## 1.1. Sobre a Gestão de Vulnerabilidades Técnicas



### Sobre a Gestão de Vulnerabilidades

- **Conjunto de atividades coordenadas que tem por objetivo a redução, a níveis aceitáveis, das vulnerabilidades de segurança encontradas durante o processo de “Análise de Segurança” ou “Análise de Vulnerabilidades” em um determinado ativo, conjunto de ativos ou ambiente.**

Guia Gestão de Vulnerabilidades Técnicas - RNP

- **Establish and maintain a documented vulnerability management process for enterprise assets. Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.**

CIS Controls v8, Safeguards 7.1

Falando um pouco sobre teoria, existem diversas definições para a Gestão de Vulnerabilidades Técnicas mundo afora. Para nosso treinamento de hoje, citaremos a definição segundo o Guia de Gestão de Vulnerabilidades Técnicas da RNP. Ele diz:

**Conjunto de atividades coordenadas que tem por objetivo a redução, a níveis aceitáveis, das vulnerabilidades de segurança encontradas durante o processo de “Análise de Segurança” ou “Análise de Vulnerabilidades” em um determinado ativo, conjunto de ativos ou ambiente.**

Perceba que nessa definição da RNP, é tratado como um conjunto de atividades coordenadas. É muito importante entendermos que não existe bala de prata em segurança da informação. As ferramentas precisam estar alinhadas com um processo bem estruturado e pessoas capacitadas para executarem esse processo.

Também trazemos uma definição existente dentro da documentação do Center for Internet Security – CIS. Nos controles do CIS, há uma salvaguarda que diz, de forma traduzida:

**Estabelecer e manter um processo documentado de gestão de vulnerabilidades. Reavaliar e atualizar essa documentação anualmente ou quando houverem mudanças significativas na empresa que impactem esta salvaguarda.**

Portanto, o processo precisa ser executado de forma periódica. Mas também é necessário que seja reavaliado para melhorias de forma periódica. Além disso, em toda e qualquer mudança drástica na instituição que o adotou, é necessário que este processo seja levado em conta também, a fim de realização de ajustes.

Um exemplo que podemos citar é quando uma empresa passa a incluir alguns ativos em infraestrutura de nuvem. O processo deverá ser alterado de forma a contemplar os novos ativos de nuvem, inclusive novas ferramentas para analisar nuvem.



## 1.2. Gestão de Vulnerabilidades: Ativos



### Gestão de Vulnerabilidades: Ativos

*The data, personnel, devices, systems, and facilities that enable the organization to achieve business purposes.*

NISTIR 8286 - NIST

A definição de gestão de vulnerabilidades do slide anterior cita o termo ativo. Basicamente, sempre que esse processo é implementado, ele visa analisar um ativo. Uma boa definição de ativo seria, segundo a publicação NISTIR 8286 do *National Institute of Standards and Technology - NIST*, de forma traduzida:

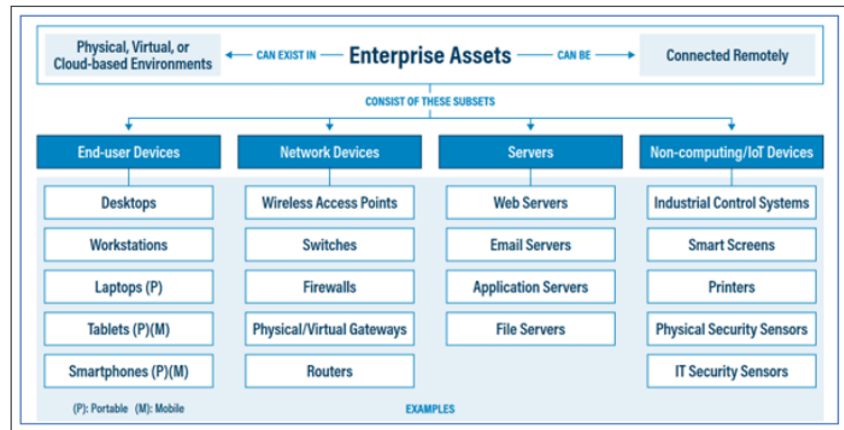
***Dados, pessoas, dispositivos, sistemas e instalações que habilitam a organização a alcançar os objetivos do negócio.***

Então, um ativo é pode ir muito além de apenas um computador ou uma rede. Por isso, é fundamental entender que o processo de gestão de vulnerabilidades precisa ir além de apenas controles físicos.



## Gestão de Vulnerabilidades: Ativos

- Dispositivos de Usuário Final;
- Dispositivos de Rede;
- IoT;
- Servidores.



Disponível em: <https://www.cisecurity.org/insights/white-papers/vulnerability-management-policy-template-for-cis-control-7>

Voltando um pouco aos ativos que são equipamentos físicos, afinal esse será o foco deste treinamento, o CIS define alguns ativos que são mais comuns nas empresas de hoje. Nesta listagem temos desde ativos de ICS até telas inteligentes.

A depender do nicho de sua instituição, você pode se deparar mais com alguns ativos dessa lista. No entanto, para o escopo de treinamento de hoje, vamos deixar um pouco de lado alguns ativos e vamos focar em outros. A nossa proposta para hoje é atender a alguns ativos em específico. Afinal, a ferramenta utilizada para escaneamento de vulnerabilidades, o OpenVAS, é mais forte no escaneamento de servidores e desktops.



### 1.3. Gestão de Vulnerabilidades: Escopo

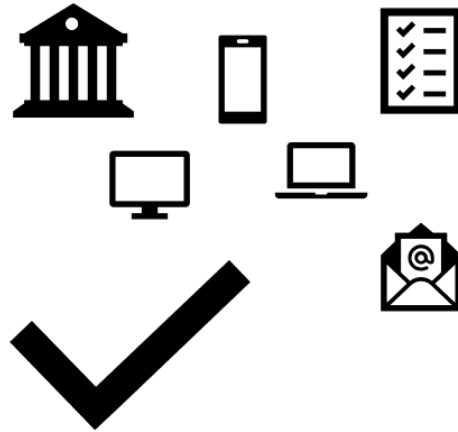


#### Gestão de Vulnerabilidades: Escopo

A execução de um processo de Gestão de Vulnerabilidades é melhor realizada quando se tem um escopo bem definido

Normalmente, o escopo precisa ser o mais abrangente possível, contendo todos os ativos da instituição

Dessa forma, a definição do escopo pode estar atrelada ao processo de inventariado. Se fizermos essa associação, garantiremos que todos os ativos, incluindo novos ativos no parque, estão incluídos no processo de Gestão de Vulnerabilidades



Geralmente, o escopo da gestão de vulnerabilidades são todos os ativos da instituição. Obviamente, para alguns ativos, a atividade é facilitada, pois existem mais meios tecnológicos para realização da atividade.

Para outros ativos, talvez nem venham a existir meios tecnológicos criados. Nesse caso, um desdobramento deverá ser feito a fim de levantamento dos riscos e possibilidades.

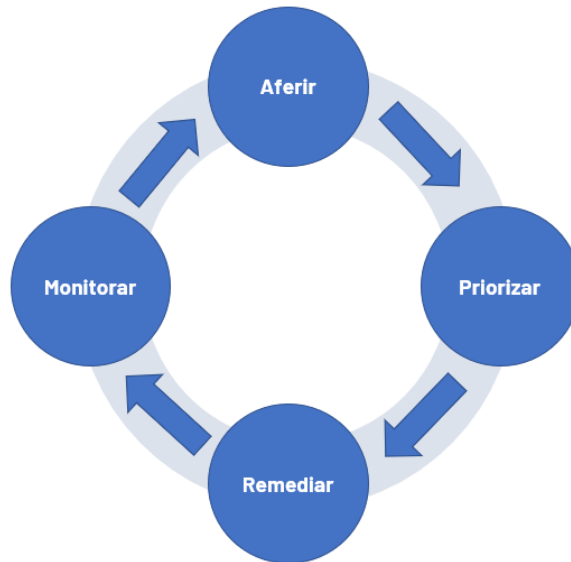




## 1.4. Gestão de Vulnerabilidades: Ciclo de Vida



### Gestão de Vulnerabilidades: Ciclo de Vida



A partir de agora, vamos falar um pouco sobre o ciclo de vida de gestão de vulnerabilidades.

Basicamente, o ciclo tem 4 fases principais: Aferir, Priorizar, Remediar e Monitorar.

Conforme bem especificado nessa imagem, e de acordo com os conceitos já apresentados, o processo é algo que está sempre acontecendo. Daí vem as setas em loop infinito.

## 1.5. Gestão de Vulnerabilidades: Aferir



### Gestão de Vulnerabilidades: Aferir

- Busca ativa por vulnerabilidades
- Ações automatizadas e/ou manuais
- Padrão de Severidade:
  - Crítica
  - Alta
  - Média
  - Baixa
  - Informativa
- Obtenção de listagem inicial



Nessa fase são realizadas ações para verificar a existência de vulnerabilidades nos ativos

Idealmente, devem ser combinadas ações de escaneamento automatizado e também testes manuais

Por padrão, as vulnerabilidades são classificadas por severidade. É utilizado um score CVSS que resulta nas seguintes classificações: Crítica, Alta, Média, Baixa, Informativa.

Ao término dessa fase é esperado a obtenção de uma listagem inicial das vulnerabilidades identificadas, com suas severidades atreladas e também o ativo no qual a vulnerabilidade reside



## 1.6. Gestão de Vulnerabilidades: Priorizar



### Gestão de Vulnerabilidades: Priorizar



- Atividade de classificação de vulnerabilidades
- Podem haver alterações em severidades
- Obtenção de uma lista pós priorização

Nessa fase é feita uma atividade de classificação das vulnerabilidades apontadas inicialmente, a fim de uma priorização

Sempre devem ser tratadas as vulnerabilidades de maior criticidade. No entanto, há cenários específicos no qual uma vulnerabilidade pode ter sua severidade alterada, devido a fatores específicos do ambiente

Ao término dessa fase, é esperado uma listagem organizada e com apontamentos de priorizações nas aplicações de mitigações

## 1.7. Gestão de Vulnerabilidades: Remediar

### Gestão de Vulnerabilidades: Remediar



- Aplicação de correções e/ou mitigações
- Sempre tentar remover a vulnerabilidade
- Aplicar mitigações diferenciadas em casos específicos
- Falhas tratadas

Na fase Remediar, correções e/ou mitigações são aplicadas

O cenário ideal é a remoção por completo da vulnerabilidade. No entanto, existem cenários nos quais isso não é possível. Nestes casos, mitigações podem ser aplicadas, como por exemplo, impedir o acesso a tal componente vulnerável

Ao término dessa fase, é esperado que as falhas identificadas sejam tratadas

## 1.8. Gestão de Vulnerabilidades: Monitorar

### Gestão de Vulnerabilidades: Monitorar



- Verificar a eficácia das tratativas
- Tratar outros problemas que venham a aparecer
- Descobrir a raiz do problema
- Reaplicar controles ou aplicar novas mitigações

Nessa fase, as correções e mitigações aplicadas são verificadas a fim de identificar a eficácia de sua aplicação. Além disso, essa verificação adicional é necessária para identificar eventuais problemas em consequência das alterações aplicadas.

Quando são identificados problemas na fase de monitoria, é necessário verificar a raiz do problema. Se a raiz do problema for erro humano na aplicação da correção, o procedimento deverá ser refeito cuidadosamente. Se a raiz do problema for estrutural, a natureza da correção deverá ser alterada. Por exemplo: uma vulnerabilidade apontada diz que um software que é crucial para o funcionamento do negócio está em uso está numa versão vulnerável. A correção realizada foi atualizar o software para uma versão mais recente. No entanto, após atualizar para a última versão disponibilizada pelo fabricante, verificou-se que esta versão possui vulnerabilidades. Nesse cenário, será necessário aplicar uma mitigação adicional, que provavelmente consistirá em isolar o componente vulnerável, ou aplicar controles de acessos severos a ele.

## 1.9. Alguns Benefícios do Processo de Gestão de Vulnerabilidades



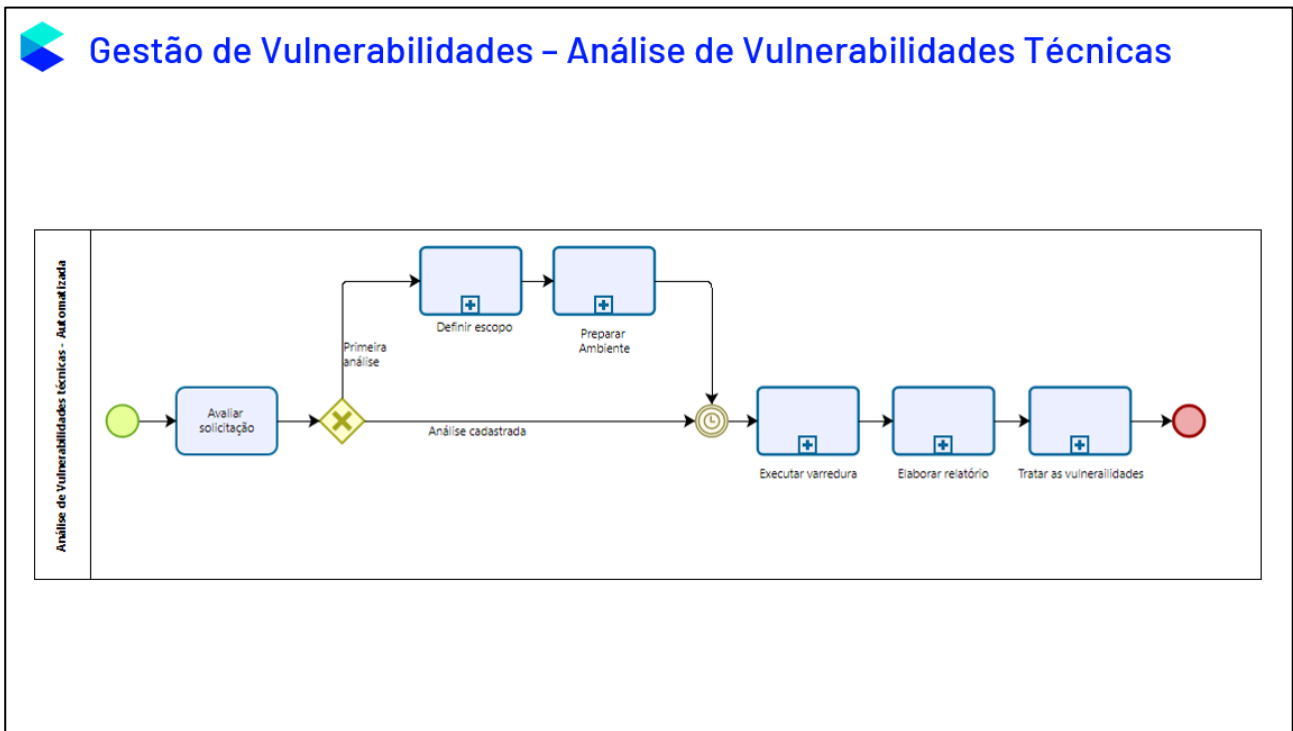
### Alguns Benefícios do Processo de Gestão de Vulnerabilidades

- Conhecimento do ambiente;
- Apoio no processo de Inventário de software e hardware (responsabilidade por ativos);
- Transparência;
- Informações claras sobre cada ativo e o que é necessário implementar;
- Auxílio para tomada de decisão;
- Priorização de ações;
- **Ambiente mais seguro.**

Os benefícios deste processo são variados. A depender de quão maduro estão os processos internos da TI, o processo de gestão de vulnerabilidades auxiliará no aumento da organização e gestão, pois poderá facilitar a identificação de ativos. É muito importante que este processo esteja associado ao processo de inclusão de um novo ativo, por exemplo.

Além disso, após aplicar um processo de Gestão de Vulnerabilidades será possível perceber aumento da postura de segurança, com resultados positivos na proteção dos ativos.

## 1.10. Gestão de Vulnerabilidades – Análise de Vulnerabilidades Técnicas



Demonstraremos agora um exemplo de modelo de processo para análise de Vulnerabilidades Técnicas. Em cada organização, este processo poderá seguir com suas particularidades. Entretanto, esse modelo macro dá uma ideia geral do processo, e como ele geralmente possui uma complexidade exequível.

**Avaliar Solicitação:** Esta fase serve para entender se se trata de uma análise que já aconteceu antes, ou se trata-se de uma nova.

**Definir escopo:** Aqui faz-se a avaliação do escopo da análise. Após interações e discussões, é esperado que se tenha em mãos uma lista com os ativos que serão analisados.

**Preparar o Ambiente:** Aqui após interações e discussões, é esperado que haja um consenso entre o período no qual os scans serão executados. Também é esperado uma definição dos tipos de liberação necessários para execução e a previsão de tais liberações.

**Executar Varredura:** Aqui é onde acontece o scan de fato. Não há muita interação, entretanto, verifica-se se os scans foram executados com sucesso.

**Elaborar relatório:** Verifica-se o relatório e o mesmo é encaminhado para os responsáveis pelo sistema. Os responsáveis pelo sistema são encarregados de aplicar na ferramenta de estatísticas os relatórios no formato adequado.

**Tratar as Vulnerabilidades:** Nessa fase, os responsáveis pelo sistema aplicam as correções correspondentes. O processo ideal visa garantir que as vulnerabilidades são mitigadas. Portanto, a equipe é responsável por garantir isso. Inclusive, novos scans podem ser executados para verificar a validade das mitigações. Perceba que, essa fase deverá incluir um subprocesso para verificação das mitigações aplicadas, a fim de verificar a eficácia das correções, bem como efeitos colaterais de sua aplicação.

## 2. OpenVAS e Defect Dojo

Agora que já pincelamos as teorias por detrás da Gestão de Vulnerabilidades. A partir de agora, vamos enfatizar em duas ferramentas abertas que podem ser utilizadas para Gestão de Vulnerabilidades: OpenVAS e DefectDojo.

### 2.1. OpenVAS



#### OpenVAS

- Open Vulnerability Assessment System
- OpenVAS = Greenbone Community Edition
- Código Aberto
- Interface Web
- Scan de Vulnerabilidades
- Muito bom para varreduras em Servidores e Desktops
- Atualizações em Feeds de Vulnerabilidades são frequentes
- Gera relatórios em formatos diferentes




**Greenbone**  
Sustainable Resilience


Fonte: <https://greenbone.github.io/docs/latest/index.html>

O OpenVAS é um scan de vulnerabilidades open-source. Também sendo chamado de Greenbone Community Edition, ele é a versão free do software pago da Greenbone. Sua maior força reside nos scans de sistemas operacionais de servidores e desktops. Ele possui uma interface web para facilitar a interação e realização das atividades e a capacidade de gerar relatórios em alguns formatos, incluindo csv e pdf. Ele é suportado pela comunidade e está atualmente na versão 22.4



## 2.2. OpenVAS Instalação

 **OpenVAS Instalação**

  
**Greenbone**  
Sustainable Resilience  
Fonte: <https://greenbone.github.io/docs/latest/index.html>

- **Informações em:**
  - <https://greenbone.github.io/docs/latest/>
- **Algumas formas de instalação incluem:**
  - Build from Source;
  - Utilizando Contêineres;
  - Instalação de pacotes.

Existem algumas maneiras de realizar a instalação. Você pode se debulhar nas formas de instalação do OpenVAS em:

<https://greenbone.github.io/docs/latest/>

Para este treinamento, optamos por realizar a instalação através do *Building from Source*. Esse método de instalação é bom para aqueles que querem entender um pouco mais dos componentes internos do OpenVAS. Hoje, no início de outubro de 2024, para este método de instalação, os Sistemas Operacionais suportados/recomendados são:

- Debian stable (bookworm)
- Ubuntu 22.04 LTS
- Fedora 38
- CentOS 9 Stream

No entanto, optamos por testar a instalação no Ubuntu 24.04. A instalação foi bem sucedida, com apenas algumas alterações durante o processo:

- Instalar Postgre16. A sugestão para Ubuntu 22.04 é postgre14;
- Quando Instalando GVMD, foi necessário instalar adicionalmente o *libcjson*, que não é indicado para instalação em outras etapas;

Alguns comandos precisaram ser executados com `sudo`, por problemas de permissionamento.

## 2.3. Defect Dojo

### Defect Dojo



Fonte: <https://defectdojo.github.io/django-DefectDojo/>

- Plataforma para Gestão de Vulnerabilidades
- Código Aberto
- Boas Integrações com Ferramentas de Segurança
- API robusta e muito funcional
- Provê grande auxílio em SecDevOps
- Aceita diversos tipos de relatório, incluindo de ferramentas específicas para nuvem

Enquanto que o OpenVAS é um scan de vulnerabilidades. O DefectDojo é uma plataforma para Gestão de Vulnerabilidades. Portanto, ambas as ferramentas trabalham muito bem em conjunto. Além disso, o DefectDojo também é open-source.

No entanto, o DefectDojo também pode trabalhar em conjunto com diversas outras ferramentas. O foco do DefectDojo é muito mais em segurança de software, e ele possui capacidades notáveis no apoio a DevSecOps.

Outro ponto forte do DefectDojo está em suas automações e como ele proporciona uma administração facilitada da Gestão de Vulnerabilidades através de sua API.

## 2.4. Defect Dojo – Instalação

 **Defect Dojo - Instalação**

- Informações em:
  - [https://defectdojo.github.io/django-DefectDojo/getting\\_started/installation/](https://defectdojo.github.io/django-DefectDojo/getting_started/installation/)
- Instalação baseada em Docker:
  - <https://hub.docker.com/r/defectdojo/defectdojo-django>



Fonte: <https://defectdojo.github.io/django-DefectDojo/>

A instalação do DefectDojo é extremamente fácil. O único método disponibilizado para a versão *free* é através dos contêineres Docker e são disponibilizados scripts que fazem o build dos contêineres.

Maiores informações sobre a instalação do DefectDojo em:

[https://defectdojo.github.io/django-DefectDojo/getting\\_started/installation/](https://defectdojo.github.io/django-DefectDojo/getting_started/installation/)

Maiores informações sobre os contêineres Docker utilizados em:

<https://hub.docker.com/r/defectdojo/defectdojo-django>

Durante o processo de instalação para este treinamento, não tivemos nenhum problema com o processo de instalação no Ubuntu 24.04. Levou em torno de 01 hora para concluir a instalação.

### 3. Utilizando OpenVAS

Agora vamos de fato utilizar o OpenVAS. Durante a instalação, foi configurado o seguinte usuário e senha: `admin/ubuntu`.

Basicamente vamos preencher as páginas associadas à seção `Configuration` do OpenVAS.

### 3.1. OpenVAS - Configurando Novas Credenciais

#### OpenVAS - Configurando Novas Credenciais



## Greenbone

Sustainable Resilience

Fonte: <https://greenbone.github.io/docs/latest/index.html>

- Credenciais são usadas para autenticação em sistemas alvo durante os scans
- São associadas aos alvos

No OpenVAS, vá para a seção `Credentials` e adicione as novas credenciais. Essas credenciais são usadas para autenticação em sistemas alvo durante os scans. Você pode criar credenciais para diferentes protocolos, como SSH, RDP ou SNMP, e definir o escopo e permissões apropriadas para cada uma.

Para nosso teste, incluiremos uma nova credencial do tipo `username + password`:

`ubuntu/ubuntu`. Para isso, clique em  no canto superior esquerdo. Nomeie essa credencial como `Ubuntu_Local`. Mantenha as opções `Allow insecure use` e `Auto-Generate` como `No`. Na sequência, basta clicar em `Save`.

## 3.2. OpenVAS - Configurando Novas Listagens de Portas

### OpenVAS - Configurando Novas Listagens de Portas




**Greenbone**  
Sustainable Resilience

Fonte: <https://greenbone.github.io/docs/latest/index.html>

- Criar uma listagem de portas que serão escaneadas
- São associadas aos alvos

Acesse a página de `Port Lists` no OpenVAS. Aqui, você pode adicionar novas regras de porta para ajustar quais portas serão verificadas durante o scan. É possível definir os intervalos de portas e protocolos específicos que desejamos incluir ou excluir dos scans.



Clique em  para criar uma nova lista de portas. Para nosso teste, nomearemos essa listagem como `SSH_HTTP` e no campo `Manual` devemos inserir `T:22,80,443`. Isso criará uma nova lista de portas com as portas TCP 22, 80 e 443.

### 3.3. OpenVAS - Configurando Novos Agendamentos

#### OpenVAS - Configurando Novos Agendamentos



**Greenbone**  
Sustainable Resilience

Fonte: <https://greenbone.github.io/docs/latest/index.html>

- Criar novos agendamentos que serão associados a tarefas

Para configurar novos agendamentos, vá até `Schedules` no OpenVAS. Crie novos agendamentos para definir quando os scans devem ser executados automaticamente. É possível programar scans diários, semanais ou mensais. Também é possível ajustar os horários. Isso é muito útil para cenários no qual as aplicações ou redes envolvidas no scan não podem ser estressadas em determinados períodos do dia.

Clique em `+` para criar um novo agendamento. Nomeie este novo agendamento como `Testes na Madrugada`. Mantenha o `Timezone` como `Coordinated Universal Time/UTC` e configure o `First Run` como `03/10/2024`, às `02h30m`. Isso configurará uma primeira execução na data especificada. A recorrência dos scans é configurada através do campo `Recurrence`. Selecione `Monthly` e clique em `Save`.

## OpenVAS - Configurando Novas Modalidades de Alertas



### OpenVAS - Configurando Novas Modalidades de Alertas



**Greenbone**


Sustainable Resilience

Fonte: <https://greenbone.github.io/docs/latest/index.html>

- Criar modalidades para envio de alertas, como por exemplo encaminhar informações via email

No OpenVAS, navegue até Alerts e configure as modalidades de alertas que deseja receber. Aqui, temos algumas opções de encaminhamento de alertas. você pode definir como as notificações serão enviadas, o que será encaminhado e será encaminhado, como exemplo via e-mail ou via webhook.



Para adicionar uma nova modalidade de alerta, basta clicar em .

Em nosso treinamento no dia de hoje, não utilizaremos esta opção. No entanto, é uma opção muito útil disponibilizada pelo OpenVAS.



### 3.4. OpenVAS - Configurando Novos Alvos

#### OpenVAS - Configurando Novos Alvos




**Greenbone**  
Sustainable Resilience

Fonte: <https://greenbone.github.io/docs/latest/index.html>

- Configurar o ativo que será efetivamente escaneado

A seção de Targets permite que você adicione e configure novos alvos para os seus scans. Os alvos são os ativos que serão escaneados, podendo ser indicados como somente um dispositivo ou um bloco de IPs a ser escaneado. Então, podemos inserir detalhes dos alvos, como endereços IP ou mesmo intervalos de IP. É aqui também que associamos um alvo a uma credencial. É aqui também que associamos um alvo a uma listagem de portas.



Para adicionar um novo alvo, clique em . Nomeie esse novo alvo como `Ubuntu_Local`. Preencha a seção de Hosts no campo `Manual` com o IP `::1`. Selecione a lista de portas conforme preenchido anteriormente: `SSH_HTTP`. Selecione a credencial para SSH conforme criado anteriormente: `Ubuntu_Local`. Mantenha as outras opções disponíveis como estão e clique em `Save`.

### 3.5. OpenVAS – Configurando Novas Tarefas

#### OpenVAS – Configurando Novas Tarefas



**Greenbone**


Sustainable Resilience

Fonte: <https://greenbone.github.io/docs/latest/index.html>

- Configurar o scan em si, associando alvos e agendamentos


As tarefas são os scans de fato. Para configurar uma nova tarefa, clicamos em `Tasks` que está dentro do subgrupo `Scans`. Em `Tasks` é possível criar e configurar novas tarefas de scan.



Clique em  e `New Task` para criar uma nova tarefa. Aqui também temos várias opções para serem modificadas. No entanto, normalmente as opções pré-selecionadas já atendem nossas necessidades. A única opção que geralmente precisamos alterar é a `Alterable Task`. Essa opção vem selecionada como `No` e caso seja mantida dessa forma, não será possível alterar essa tarefa futuramente.

Para nosso treinamento, nomeie essa tarefa como `Scan_Imediato_Ubuntu_Local`. Selecione os alvos em `Scan Targets` com o alvo que criamos anteriormente: `Ubuntu_Local`. Na seção `Schedule` selecione o campo `Once`. Essa configuração dirá que essa tarefa só será executada uma única vez. Mantenha as outras opções com as seleções padrão e clique em `Save`.



Após criar uma nova tarefa, clique em  para executar essa tarefa.

### 3.6. OpenVAS – Executando Scans e Gerando Relatórios

#### OpenVAS – Executando Scans e Gerando Relatórios




**Greenbone**  
Sustainable Resilience

Fonte: <https://greenbone.github.io/docs/latest/index.html>

- Gerar relatórios em:
  - CSV;
  - PDF;
  - TXT

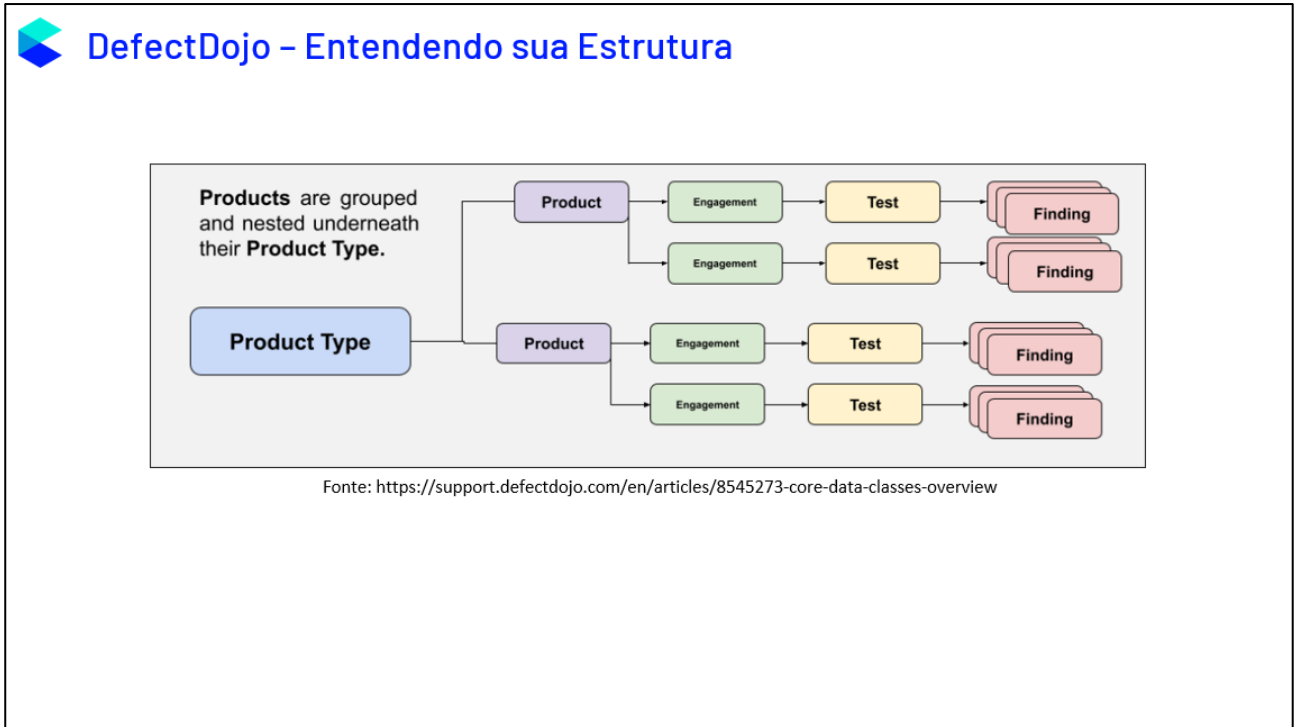
O OpenVAS permite a geração de relatórios em alguns formatos. Vou demonstrar para vocês um relatório em PDF, no entanto, para fins de utilização com o DefectDojo, o formato utilizado é CSV.

Para gerar um novo relatório, acesse a página da Tarefa desejada e clique em algum relatório. Após ser

direcionado à página do relatório, clique em  para selecionar o formato desejado e em seguida baixar.

## 4. Utilizando Defect Dojo

### 4.1. DefectDojo – Entendendo sua Estrutura



Antes de prosseguir para a demonstração prática no DefectDojo, é importante uma breve explicação de como o DefectDojo se organiza no que diz respeito às informações das vulnerabilidades.

O DefectDojo tem uma subdivisão, o que eles chamam de classes de dados, ou data classes. Existem cinco classes principais de dados: *Product Types*, *Products*, *Engagements*, *Tests* e *Findings*.

Embora possa parecer avançado, na verdade essa estrutura é muito flexível e permite modificações para uma adequação ao ambiente da sua instituição.

## 4.2. DefectDojo – Configurando Tipos de Produto

 **DefectDojo – Configurando Tipos de Produto**

  
Fonte: <https://defectdojo.github.io/django-DefectDojo/>

- Tipos de Produtos ou Product Types;
- Agrupamento macro para Products
- Podem ser marcados como:
  - Critical product
  - Key product
- Podem ter controles de acesso baseados em Roles.

Os Tipos de Produtos, ou ProductTypes, são o topo da cadeia na estrutura das classes de dados do DefectDojo. Portanto, é a primeira coisa que você irá configurar relacionado à vulnerabilidades.

A ideia por detrás dos ProductTypes é ter uma forma macro para organização dos Products. Por exemplo, você pode configurar um Product Type para uma área específica de sua empresa. Novamente digo, a ferramenta é bem flexível e permite que você faça a organização da forma que achar mais conveniente e atrativa para suas necessidades.

É possível associar controles de acesso baseados em roles e ProductTypes. Portanto, é possível, por exemplo, definir que determinados usuários terão acesso apenas a um ProductType.

Para adicionar um novo ProductType, acessa o menu lateral à esquerda e clique em Add Product Type.

Vamos criar um Tipo de Produto chamado Segurança.

Podemos colocar uma descrição e selecionar os campos `Critical Product` e `Key Product`. Estas marcações são utilizadas principalmente nas métricas e estatísticas.

### 4.3. DefectDojo – Configurando Novos Produtos

 **DefectDojo – Configurando Novos Produtos**

  
Fonte: <https://defectdojo.github.io/django-DefectDojo/>

- **Podem ser:**
  - Programas;
  - Projetos;
  - Servidores;
  - Aplicações Web;
- **Precisam:**
  - Ter um nome único;
  - Ter uma descrição;
  - Estar associados a um ProductType;
  - Estar associados a um SLA.
- **É possível configurar controles de acesso baseados em Roles.**

Idealmente, segundo os desenvolvedores, os Produtos ou *Products*, são utilizados para representar algum programa ou projeto. Nas utilizações que fiz do DefectDojo, cadastrei como *Products* aplicações Web e Servidores. A flexibilidade da plataforma permite isso.

Existem algumas regras para os *Products*:

- Precisam ter um nome único;
- Precisam de uma descrição;
- Precisam estar associados a um *ProductType*;
- Precisam ter um SLA associado.

Assim como nos *Product Types*, também é possível configurar regras de controle de acesso baseados em Roles nos *Products*.

Para adicionar um novo *Product*, acesse o menu lateral à esquerda e clique em Add Product.

Vamos criar um *Product* chamado `Ubuntu_Local`.

## 4.4. DefectDojo – Importando Relatórios do OpenVAS via Interface Web



### DefectDojo – Importando Relatórios do OpenVAS via Interface Web



Fonte: <https://defectdojo.github.io/django-DefectDojo/>

- **Regras para criação de novo Engagement:**
  - Nome único;
  - Data de início e Fim;
  - Status;
  - Responsável pelo testes;
  - Um produto associado.
- **Ao importar um scan, um engagement AD-HOC será criado;**

Os *engagements* do DefectDojo é a classe de dados nos quais define-se períodos nos quais testes serão executados na plataforma.

Sempre que formos criar um *engagement* será necessário preencher:

- Um nome único;
- Datas de início e fim dos testes;
- Status;
- Um responsável pelos testes;
- Um produto associado.

Existem dois principais tipos de *engagement*: *interactive engagement* e *CI/CD engagement*.

Eu, particularmente, nunca adicionei um novo *engagement* na aplicação. Isso porque, quando você importa um novo relatório, já é criado um novo *engagement*.


Para criar um novo *engagement*, clique em *Engagements* e *Add New* de dentro da página de um *Product*. Ou, você pode importar um relatório e associá-lo a esse *Product*. Dessa forma, será criado um *engagement* automaticamente. Para fazer isso, clique em *Findings* e *Import Scan Results*. Na página seguinte, faça o preenchimento conforme a importação sendo realizada.

No nosso treinamento, iremos importar o relatório do openvas em CSV. Portanto, será necessário selecionar o arquivo e escolher o *Scan Type* como *OpenVAS Parser*.


Após essa importação, já poderemos visualizar as vulnerabilidades no DefectDojo.



## 4.5. DefectDojo – Importando Relatórios do OpenVAS via API

 **DefectDojo – Importando Relatórios do OpenVAS via API**

- **Necessário token de acesso à API**
  - Disponível em: `/api/key-v2`
- [documentation.defectdojo.com/integrations/api-v2-docs/](https://documentation.defectdojo.com/integrations/api-v2-docs/)
- `/api/v2/oa3/swagger-ui`



Fonte: <https://defectdojo.github.io/django-DefectDojo/>

Sobre a API do DefectDojo: quando precisamos importar uma grande quantidade de relatórios, por exemplo, podemos usar a API.

Um ponto que pode nos ajudar com essas automações é trabalhar num esquema padronizado de nomeação dos *Products* dentro do DefectDojo. Dessa forma, ficará muito mais fácil realizar a importação correta dos novos scans.

Para utilizar a API, será necessário fornecer o token de acesso único para o usuário realizando a requisição. Sem esse token não será possível comunicar com a API.

Preparei vários scripts em python para comunicação com a API.

A documentação da API está disponibilizada em:

- <https://documentation.defectdojo.com/integrations/api-v2-docs/>

[https://<SEU\\_DEFECTDOJO>/api/v2/oa3/swagger-ui](https://<SEU_DEFECTDOJO>/api/v2/oa3/swagger-ui)

## 4.6. DefectDojo – Obtendo Métricas e Estatísticas



### DefectDojo – Obtendo Métricas e Estatísticas



Fonte: <https://defectdojo.github.io/django-DefectDojo/>

Em se tratando de métricas e estatísticas de vulnerabilidades, o DefectDojo também é muito forte.

Ele disponibiliza diversas métricas, que podem ser contabilizadas em conjunto e separadas. Por exemplo, você pode obter métricas relacionadas a um *Product Type* específico.

