globo.com

Desafios no desenvolvimento de ferramentas de segurança

Vitória Rio

Como era…

- Processo de desenvolvimento isolado
  - A manutenção de cada projeto depende de quem o desenvolveu
  - Não abre espaço para discussões

Code Review

- Trancar a branch “master" dos projetos
- A aprovação depende de um segundo membro da equipe

Code Review

4 meses

9 integrantes

Code Review

- Processo de desenvolvimento perde a velocidade…

5 dias

Tempo médio de aprovação

Code Review

Merge requests

90

33

(36,6%)

Tiveram discussões

Code Review

Tópicos de discussão

64

- Log (20,3%)
- Documentação (18,75%)
- Decisões de arquitetura (17,75%)
- Lógica (10,9%)

Code Review

Mudanças

37

(57,8%)

Issues abertas

6

(9,3%)

Code Review

- Vantagens:
  - Reduz possibilidade de erros
  - Compartilhamento de conhecimento
  - Melhoria na manutenção de projetos
- Desvantagem:
  - Perda de velocidade

Técnicas

- Garantir a integridade dos dados
- Permitir somente acesso autorizado
- Ser capaz de avaliar os danos
- Ser capaz de se recuperar dos danos sofridos

Cenário

- Sistema de autenticação multi-fator (geração de tokens)
- Ferramenta de gestão de vulnerabilidades

Banco de dados

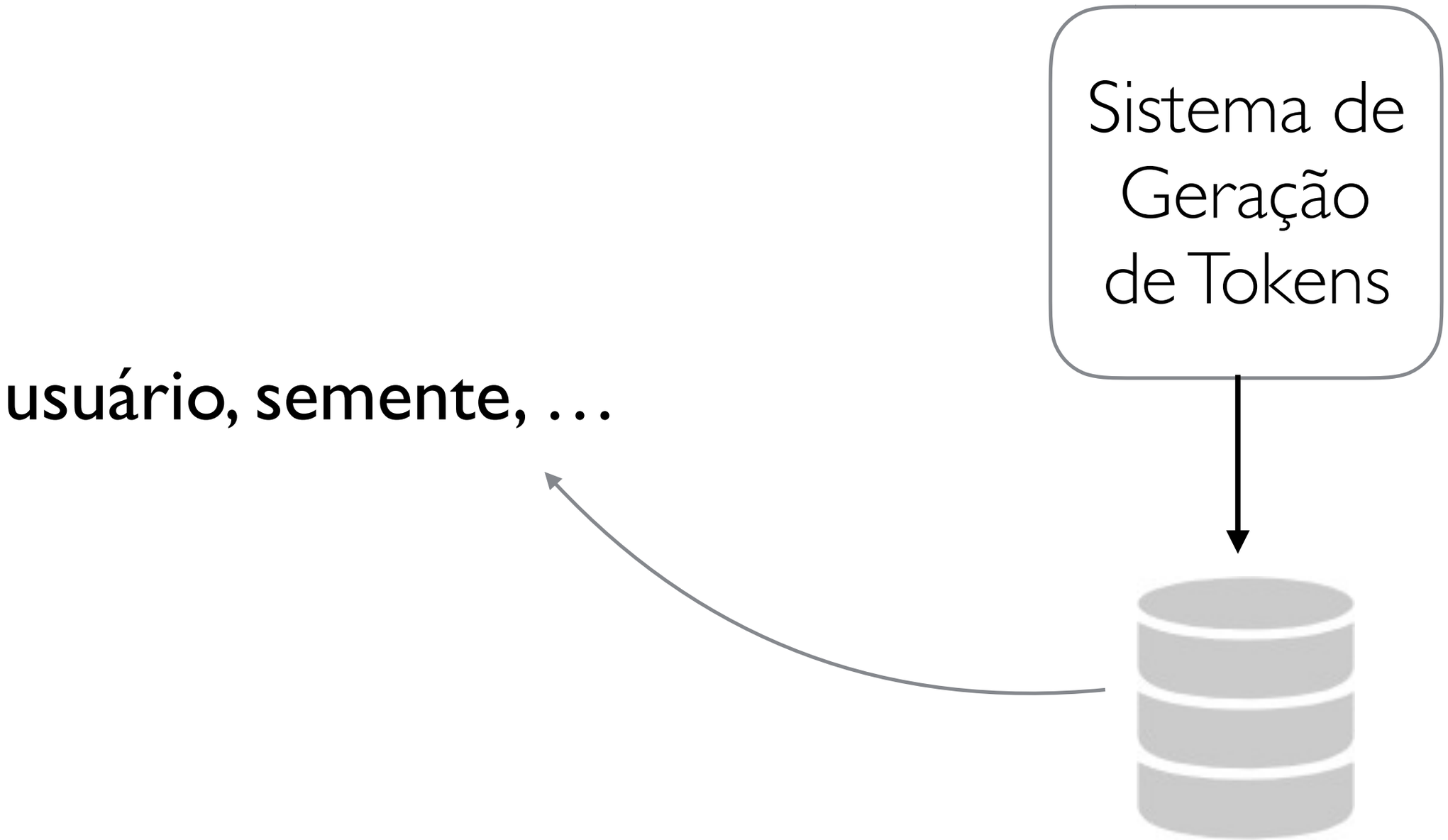
- Como proteger dados sensíveis do banco?

Banco de dados

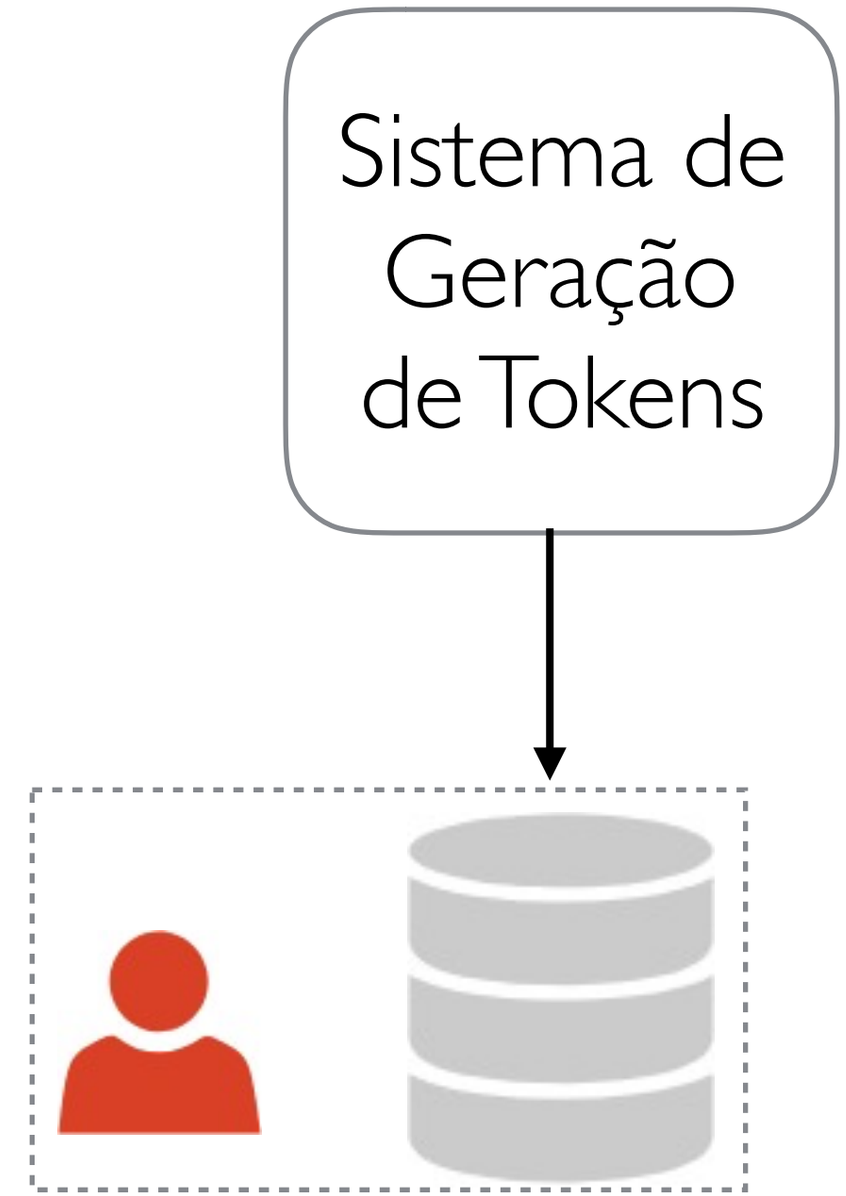
- Como proteger dados sensíveis do banco?

Criptografia dos dados

Banco de dados

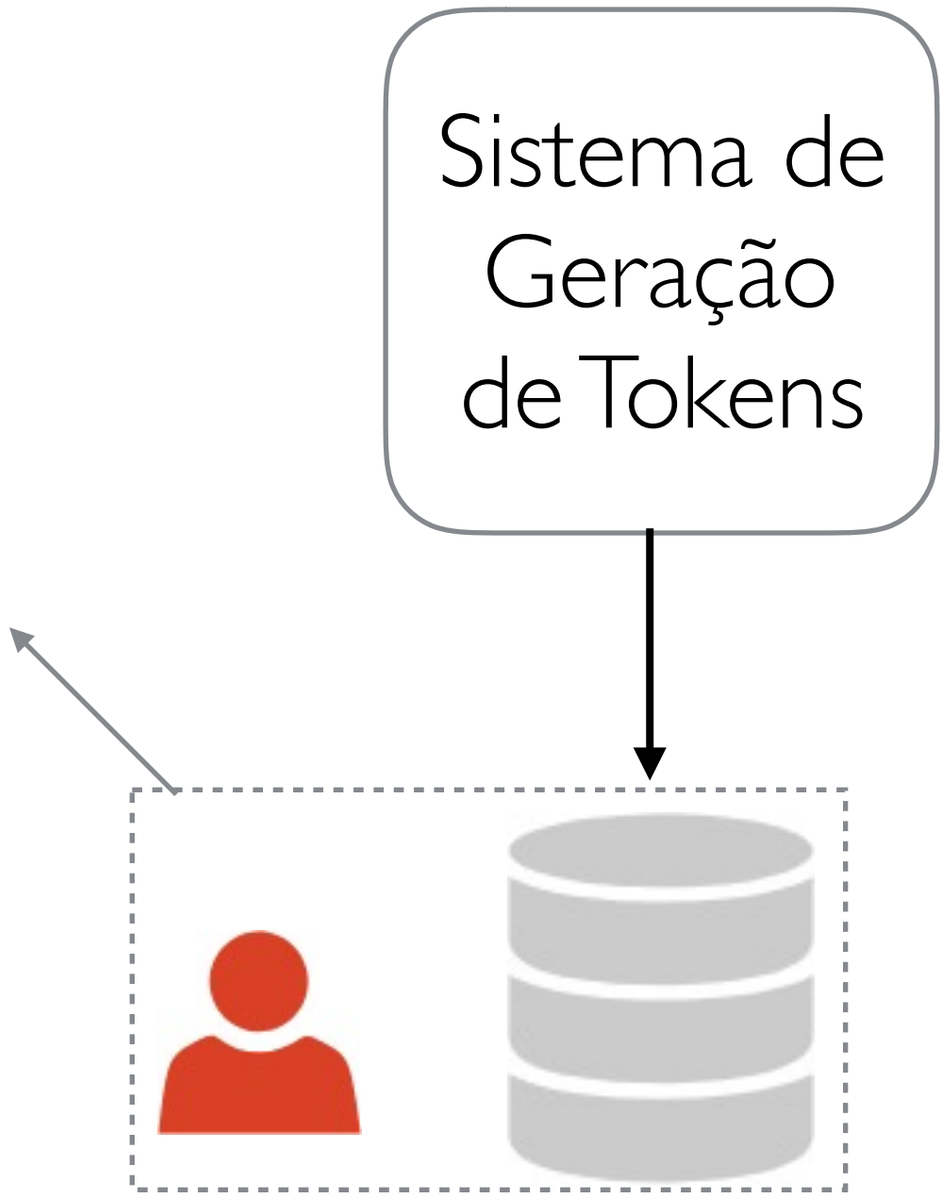


Banco de dados

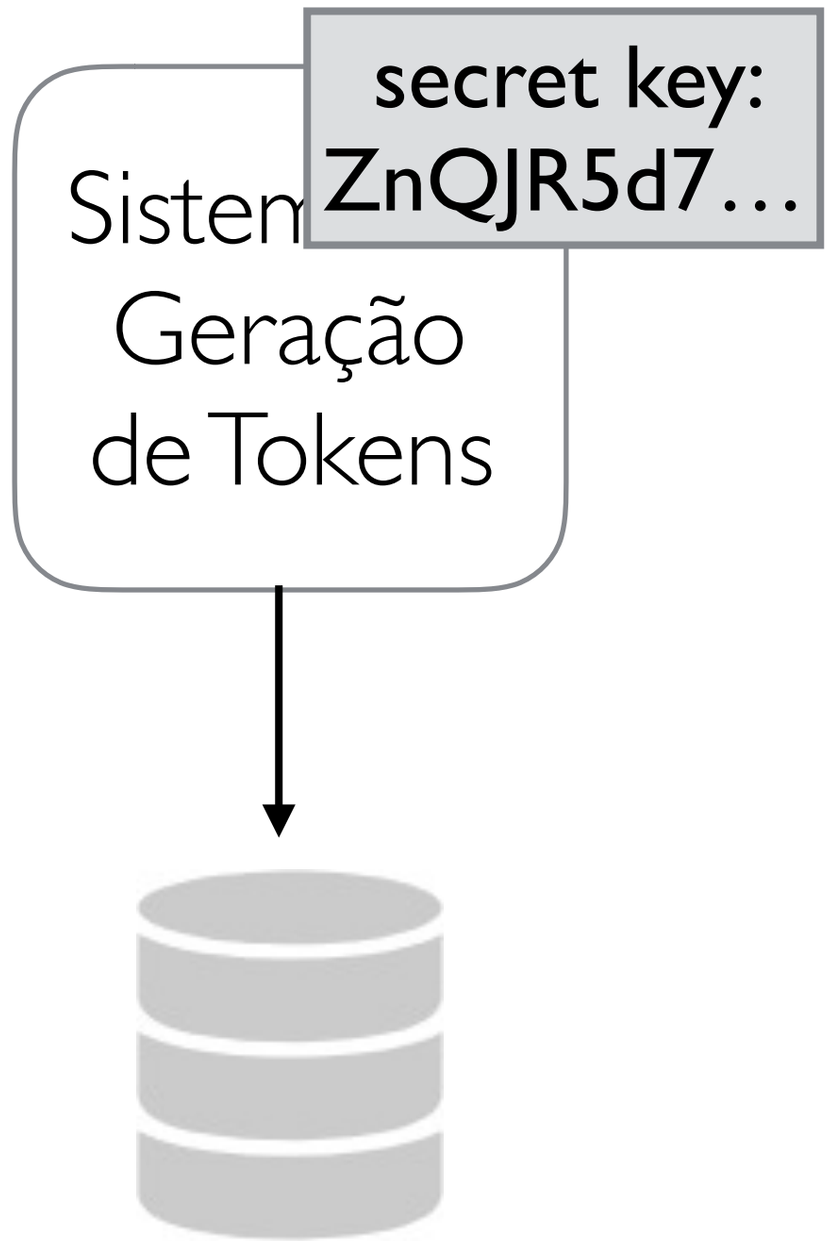


Banco de dados

usuario	semente	…
user1	abcdef	
user2	ghijkl	
user3	mnopqr	
…	…	



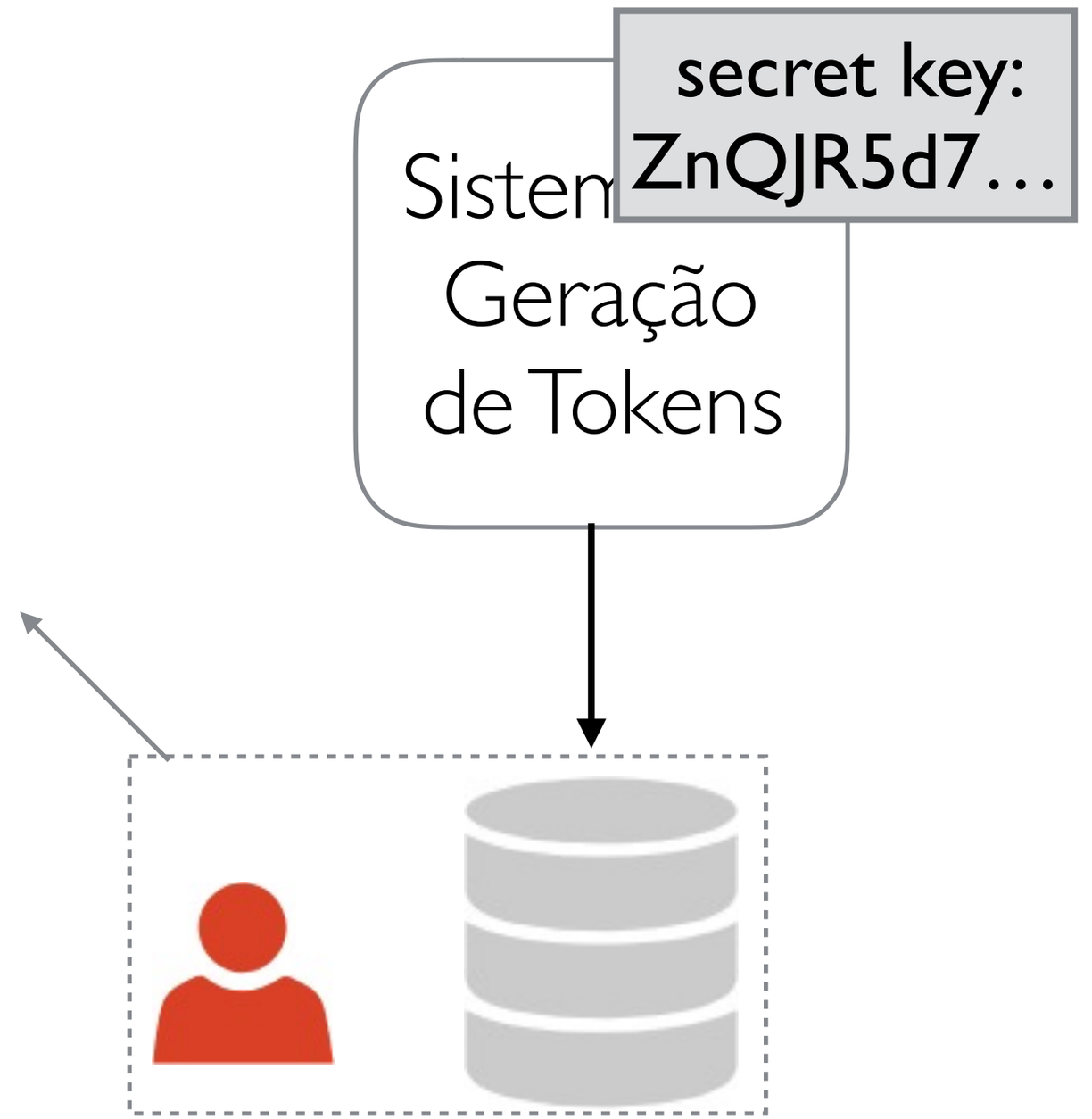
Banco de dados



Banco de dados

???

usuario	semente	…
user1	H6asdUla8	
user2	60aHis1Fdl	
user3	O966aTgD	
…	…	



Permissionamento

- Modelos de controle de acesso:

Mandatório (MAC)

- Permissões localizadas no sistema
- Somente o sistema concede permissões
- Hierarquia de classificação

Discricionário (DAC)

- Permissões localizadas no objeto
- Usuário/Grupo pode conceder suas próprias permissões (Herança)

D., Bokefode Jayant; A., Ubale Swapnaja; S., Apte Sulabha. Analysis of DAC MAC RBAC Access Control based Models for Security. International Journal of Computer Applications, v. 104, n. 5, p. 6–13, 2014. Disponível em: <https://research.ijcaonline.org/volume104/number5/pxc3899115.pdf>

Permissionamento

Qual escolher?

Permissionamento

Qual escolher?

Híbrido

Permissionamento

- **Permissões localizadas no sistema**

Permissionamento

- Permissões localizadas no sistema
- **Baseado no usuário**

Permissionamento

Papéis

- Mais simples (fácil manutenção)
- Conjunto de permissões pré-definido
- Mais geral

Usuário

- Complexo
- Maior garantia de que cada usuário terá acesso apenas ao necessário
- Mais granular

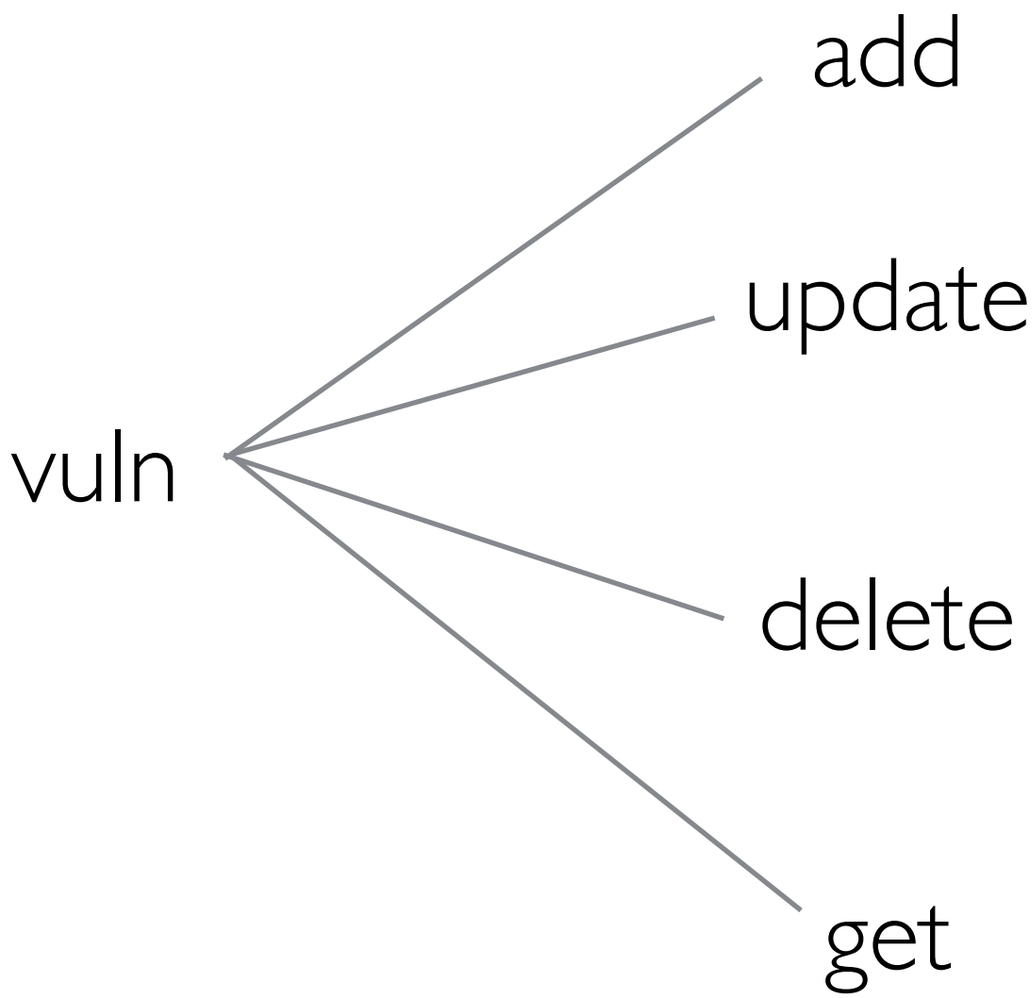
Permissionamento

- Permissões localizadas no sistema
- Baseado no usuário
- **Permissões em contextos**
  - Granularidade de permissões e contextos
  - Hierarquia de permissões e contextos

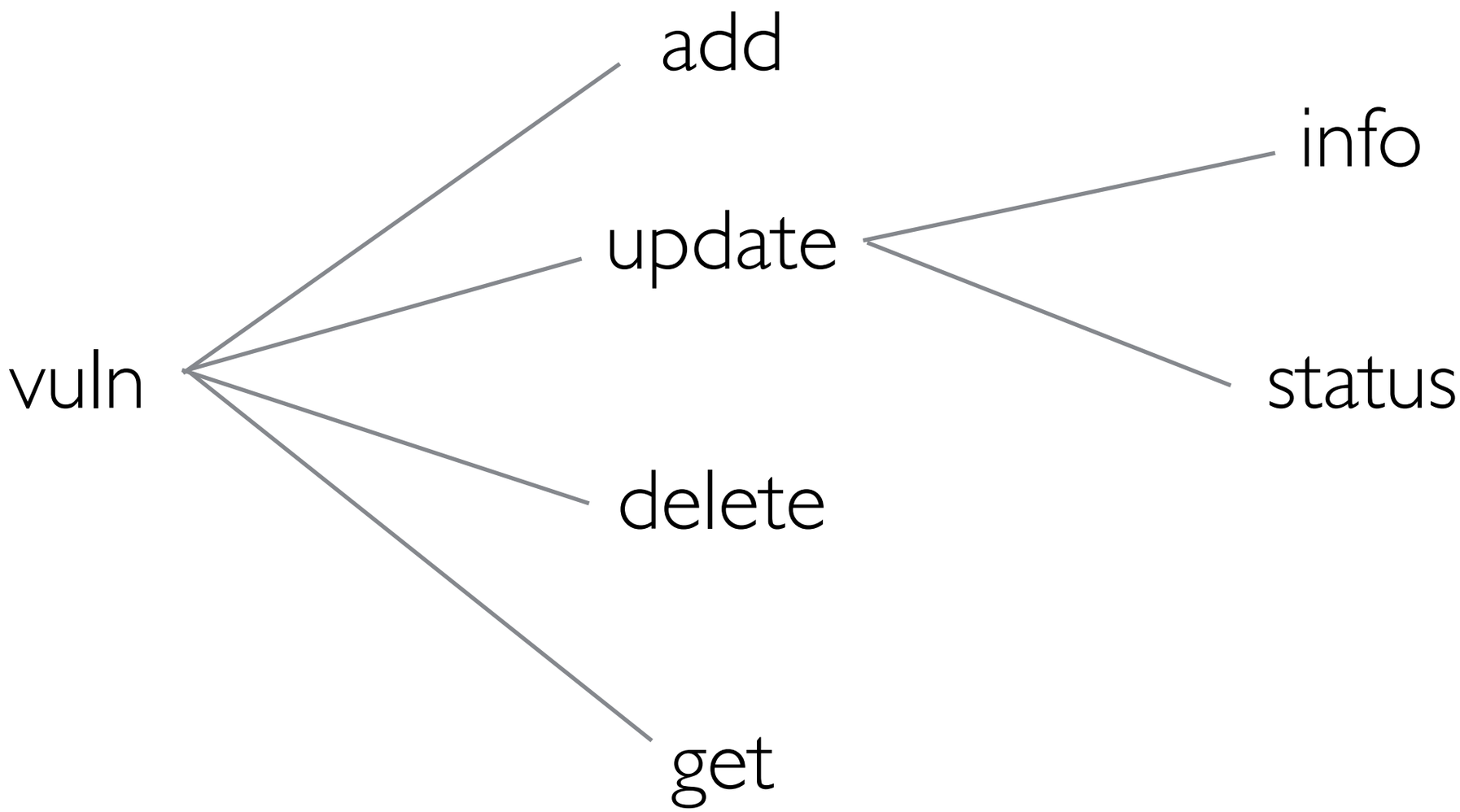
Permissionamento

vuln

Permissionamento

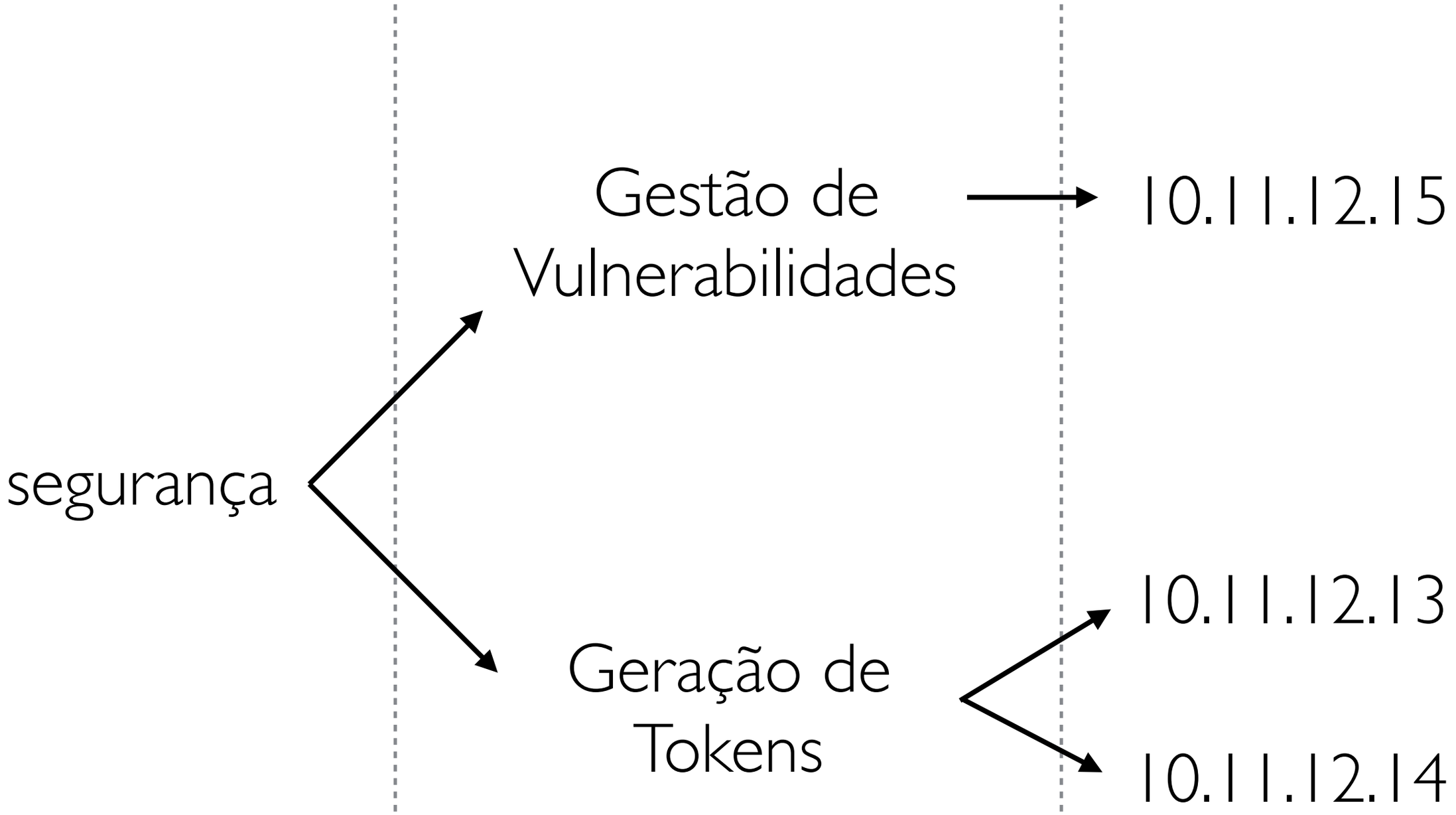


Permissionamento



vuln.update = vuln.update.info + vuln.update.status

Permissionamento



Permissionamento

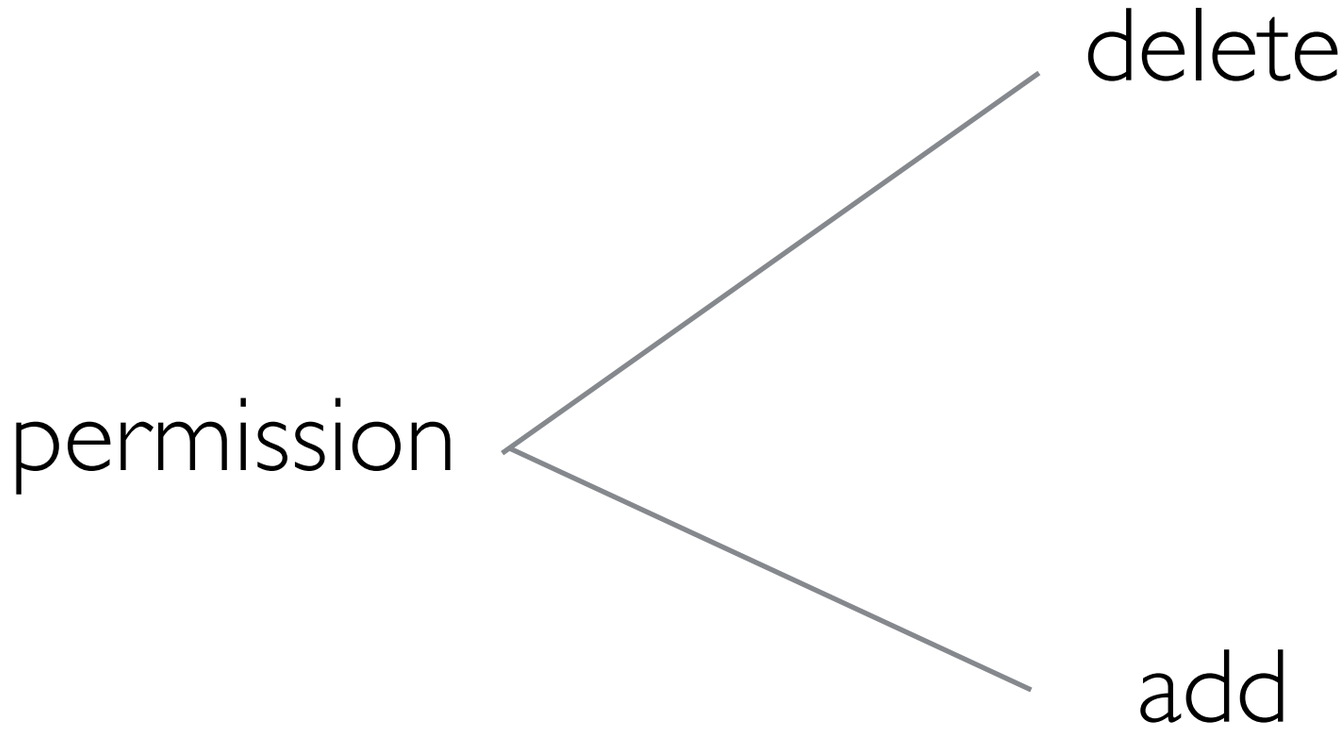
```
{
  user: "user123",
  permissions:[
    {
      name: "vuln.get",
      ctx: "project",
      value: “Geração de Tokens",
    }
  ]
}
```

Permissionamento

- Permissões localizadas no sistema
- Baseado no usuário
- Permissões em contextos
  - Granularidade de permissões e contextos
  - Hierarquia de permissões e contextos
- **Conceder permissões dinamicamente**
  - Permissão para conceder permissão

Permissionamento

- Concedendo permissões:



Permissionamento

```
{
  user: "user123",
  permissions:[
    {
      name: "vuln.get",
      ctx: "project",
      value: “Geração de Tokens",
    }
  ]
}
```

Permissionamento

```
{
  user: "user123",
  permissions:[
      {
        name: "vuln.get",
        ctx: "project",
        value: “Geração de Tokens"
      },
      {
        name: “permission.add",
        ctx: "project",
        value: “Geração de Tokens"
      }
  ]
}
```

Auditoria

- Escrita de eventos
  - Avaliar danos
  - Possibilita a recuperação dos danos

Auditoria

- Busca responder as seguintes perguntas:
  - O quê?
  - Quem?
  - Onde?
  - Quando?
  - Por quê?
  - Como?

Auditoria

- Busca responder as seguintes perguntas:
  - **O quê?**
  - Quem?
  - Onde?
  - Quando?
  - Por quê?
  - Como?

- Ação
- Resposta

Auditoria

- Busca responder as seguintes perguntas:
  - O quê?
  - **Quem?**
  - Onde?
  - Quando?
  - Por quê?
  - Como?

- IP de origem
- Identificador de usuário

Auditoria

- Busca responder as seguintes perguntas:
  - O quê?
  - Quem?
  - **Onde?**
  - Quando?
  - Por quê?
  - Como?

- Rota
- Aplicação
- Função

Auditoria

- Busca responder as seguintes perguntas:
  - O quê?
  - Quem?
  - Onde?
  - **Quando?**
  - Por quê?
  - Como?

- Data/hora
- Timestamp

Auditoria

- Busca responder as seguintes perguntas:
  - O quê?
  - Quem?
  - Onde?
  - Quando?
  - **Por quê?**
  - Como?

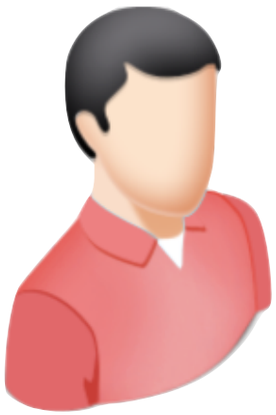
- Causa do erro
- Por que a ação foi bem sucedida?

Auditoria

- Busca responder as seguintes perguntas:
  - O quê?
  - Quem?
  - Onde?
  - Quando?
  - Por quê?
  - **Como?**

- Requisição

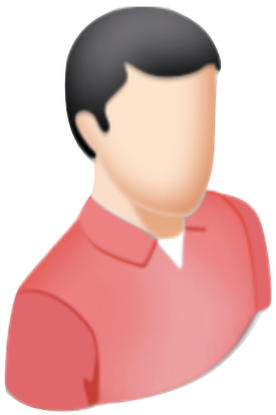
Auditoria



(POST)…/vuln
{id:VLN001, ip: 10.10.10.8, info: xss}

API
Gestão de
Vulnerabilidades

Auditoria

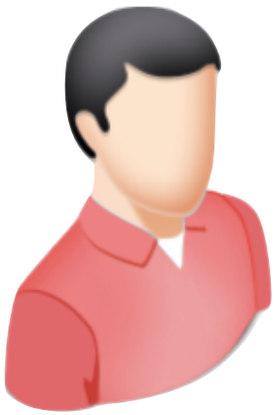


(POST)…/vuln
{id:VLN001, ip: 10.10.10.8, info: xss}

API
Gestão de
Vulnerabilidades

evento

Auditoria



"não autorizado"

API
Gestão de
Vulnerabilidades

evento

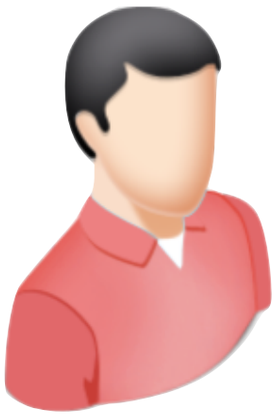
Auditoria

```
{
  source_ip: "192.168.1.10",
  user: “user123",
  path: “/vuln",
  request: "{id: VLN001, ip: 10.10.10.8, info: xss}”,
  action: “cadastro de vulnerabilidade”,
  response: “403”,
  error: “não autorizado”,
  timestamp: 1526760828
}
```

Soft delete

- Técnica para evitar perda de dados
- O dado é removido apenas na visão do usuário
- Exemplos:
  - Email
  - Sistema operacional

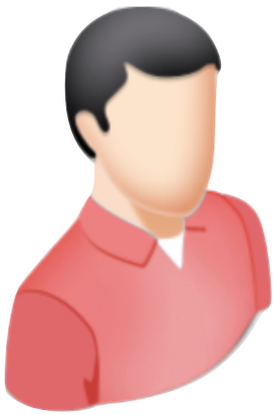
Soft delete



(DELETE)…/vuln/VLN002

API
Gestão de
Vulnerabilidades

Soft delete

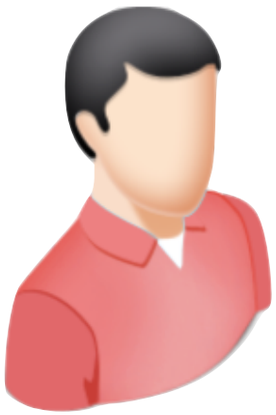


(DELETE)…/vuln/VLN002

API
Gestão de
Vulnerabilidades

```
{
  id:"VLN002",
  ip:”10.10.1.10”,
  info: “xss”,
  status: "removed"
}
```

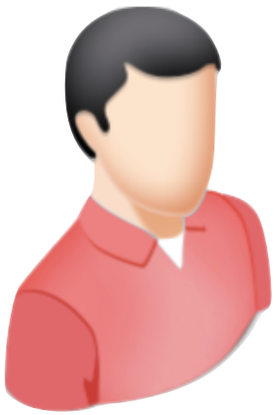
Soft delete



(GET)…/vuln?id=VLN002

API
Gestão de
Vulnerabilidades

Soft delete

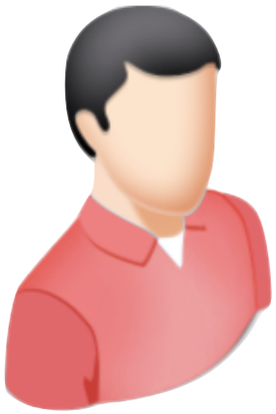


(GET)…/vuln?id=VLN002

API
Gestão de
Vulnerabilidades

id = VLN002
status != "removed"

Soft delete



"Not found"

API
Gestão de
Vulnerabilidades

Próximos passos

- O que melhorar?

Métricas

- Ajudam a definir os pontos que precisam de atenção
  - Performance (tempos de resposta)
  - Uso de recursos
  - Número de falhas

- É necessária uma análise crítica

Métricas

- No exemplo do code review…
  - Métrica: 5 dias para aprovação
  - Análise: O tempo para aprovação está elevado
  - Causas: Merge requests não detalhados; falta de tempo para realizar code review; falta de adesão.
  - Ações: Se dedicar a elaboração dos merge requests; Comprometimento.

Obrigada

vitoria.rio@corp.globo.com