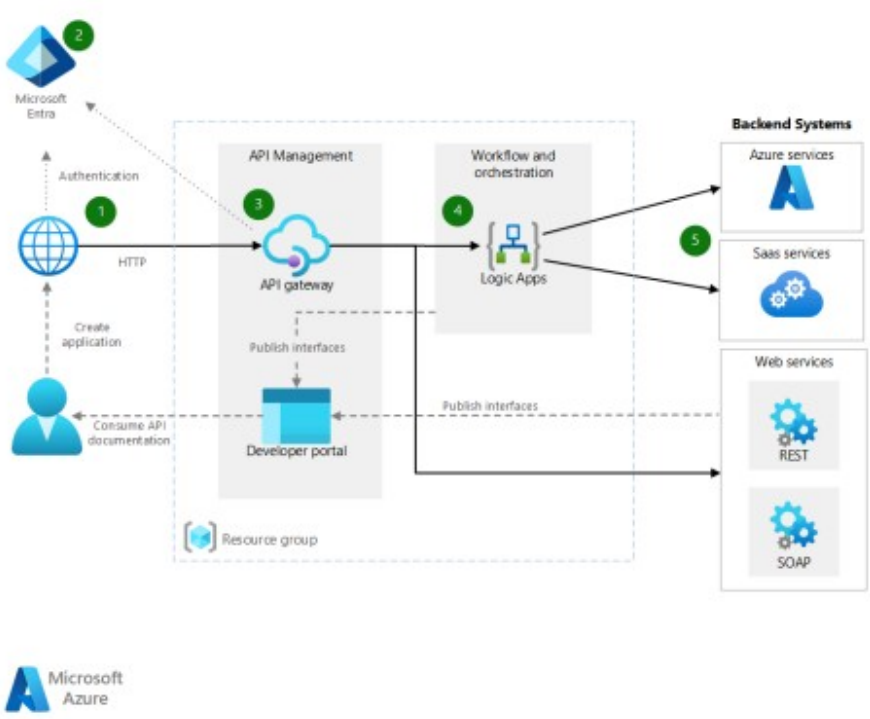
Relatório de análise da solução: tmppg7s02h_



Análise completa da solução atual

Modelo de cloud:
- Microsoft Azure

Lista com os componentes:
- Cliente/Consumidor da API (aplicativo ou serviço chamador via HTTPS)
- Microsoft Entra ID (Azure AD) para autenticação/autorização
- Azure API Management (APIM)
  - API Gateway
  - Developer Portal
- Azure Logic Apps (Workflow and Orchestration)
- Backend Systems
  - Azure services (por exemplo: Azure Functions, Service Bus, Storage, Cosmos DB etc.)
  - SaaS services (conectores para terceiros)
  - Web services (REST e SOAP)
- Resource Group (contenção e governança de recursos)

Interação entre os componentes:
1) O Cliente envia requisições HTTPs para endpoints expostos no API Gateway (APIM).
2) A autenticação é feita via Microsoft Entra ID (OAuth2/OpenID Connect). O APIM valida o token (e opcionalmente subscription keys).
3) O APIM aplica políticas (throttling, validação de JWT, transformação, CORS, schema validation) e roteia a chamada.
4) Para orquestração, o APIM invoca Logic Apps, que coordena fluxos e integrações usando conectores.
5) Logic Apps e/ou APIM chamam os Backend Systems:
   - Serviços Azure via SDK/REST/Private Link
   - SaaS via conectores gerenciados
   - Web services REST/SOAP externos
O Developer Portal publica documentação das APIs e gerencia o onboarding de desenvolvedores.

O que esse sistema faz:
- Plataforma de exposição e mediação de APIs com orquestração de integrações. Centraliza autenticação/autorização, aplica políticas de segurança/escala, converte/transforma payloads, integra serviços Azure, SaaS e web (REST/SOAP) por meio de Logic Apps.

Vulnerabilidades e Solução para cara vulnerabilidade:

- Tokens mal validados (JWT/OIDC): usar validate-jwt no APIM, validar iss/aud/exp/nbf, rotação de chaves (JWKS), reforçar escopos e roles no Entra.
- Exposição de chaves (subscription keys/segredos de conectores): armazenar em Azure Key Vault; usar Managed Identity para Logic Apps/APIM; evitar segredos em configs/variáveis de pipeline.
- Tráfego não criptografado: impor HTTPS/TLS 1.2+; considerar mTLS entre APIM e backends sensíveis; HSTS.
- DoS/DDoS e abuso de API: rate-limit e quota policies no APIM; proteção DDoS na VNet; caching onde aplicável; circuit breaker/retries com jitter.
- Injeções/validações insuficientes (REST/SOAP): schema validation no APIM; sanitização; bloquear XXE/DTD em XML; limitar tamanho/profundidade de XML/JSON.
- CORS mal configurado: evitar comodins em origins; whitelists específicas; bloquear credenciais se desnecessário.
- Orquestração com privilégios excessivos: princípio do menor privilégio com Managed Identities; RBAC por recurso; segmentação por Resource Group.
- Exfiltração de dados via conectores SaaS: DLP e classificação de dados; políticas de exportação; mascaramento/tokenização; restringir conectores.
- Vazamento em logs: evitar logar PII/segredos; mascaramento no APIM/Logic Apps; retenção conforme LGPD; Storage com imutabilidade quando aplicável.
- Exposição pública do APIM/backends: usar APIM em modo Internal com VNet + Application Gateway/WAF; Private Link para backends; IP allowlists.
- Configuração de políticas fraca no APIM: política de segurança base (headers, CSP, anti-cache sensível), validação de content-type, rejeição de métodos não usados.
- Supply chain dos conectores/terceiros: due diligence de fornecedores; SLA/DPAs; monitoramento contínuo; isolamento de integrações por subscrição.
- Falhas de monitoração e resposta: Azure Monitor/Log Analytics/Defender for Cloud; alertas de anomalia; dashboards; runbooks de resposta.
- Erros de CI/CD: validação de políticas como código (APIM DevOps, ARM/Bicep/Terraform); varredura SAST/DAST; revisão de PR; escaneamento de secrets.
- Configuração de SOAP específica: bloquear SOAPAction inesperado; limites de anexos; desabilitar redirecionamentos automáticos.

Gere um Relatório de Modelagem de Ameaças, baseado na metodologia STRIDE:
- Escopo e ativos
  - Escopo: Exposição de APIs via Azure API Management com orquestração em Logic Apps integrando serviços Azure, SaaS e Web (REST/SOAP).
  - Ativos: Endpoints de API, tokens JWT/refresh tokens, chaves/subscriptions, segredos de conectores, dados sensíveis em trânsito/repouso, definições de políticas do APIM, runbooks/workflows de Logic Apps, logs e métricas.
- Atores
  - Consumidor legítimo de API; Desenvolvedor via Developer Portal; Administradores de APIM/Logic Apps; Serviços back-end; Atacante externo; Terceiros de SaaS.
- Fronteiras de confiança
  - Internet → APIM (perímetro público/WAF).
  - APIM → Logic Apps (intra-Azure, preferencialmente VNet/Private Link).
  - Logic Apps/APIM → Backends (Azure/SaaS/Web, diversas políticas).
  - Dev Portal/DevOps → Administração (RBAC/Azure AD).
- Fluxos de dados (referência 1–5 acima)
  - Dados: HTTPs requests, tokens OIDC/JWT, payloads JSON/XML, segredos de conectores, logs/telemetria.
- STRIDE por categoria
  - Spoofing
    - Riscos: falsificação de identidade do cliente; tokens forjados; backends falsos.
    - Mitigações: validate-jwt; mTLS onde crítico; DNSSEC/Private DNS; Managed Identities; validação de certificado de saída; pinning de host em policies.
  - Tampering
    - Riscos: alteração de payloads/políticas; manipulação de workflows.
    - Mitigações: TLS forte; schema validation; assinaturas de payload (JWS) quando necessário; controle de mudanças com IaC; RBAC + PIM; versionamento/approval gates.
  - Repudiation
    - Riscos: negação de autoria de chamadas/admin; logs incompletos.
    - Mitigações: logging imutável (Storage com Immutability/Legal Hold); correlação (trace-id); auditoria do Entra; carimbo de tempo; retenção e proteção contra edição.
  - Information Disclosure
    - Riscos: vazamento de tokens/segredos/PII; erro detalhando stack; logs sensíveis.
    - Mitigações: Key Vault; mascaramento; mensagens de erro genéricas; data classification; criptografia em repouso (CMK opcional); Private Link; mínimo escopo de tokens.
  - Denial of Service
    - Riscos: flood no APIM; fan-out em Logic Apps; endpoints SOAP/REST pesados; picos em backends.

  - Mitigações: rate-limit/quota; caching; circuit breaker/backoff; dimensionamento de APIM/Logic Apps; DDoS Protection; fila assíncrona (Service Bus) para workloads intensivos.
 - Elevation of Privilege
  - Riscos: políticas permissivas; identidades gerenciadas com papéis amplos; JWT com escopos mal definidos.
  - Mitigações: RBAC mínimo; PIM para admins; separação por RG/subscrição; escopos/roles granulares no Entra; revisão periódica de acessos; Just-In-Time.

- Riscos principais priorizados
 - Alto: validação insuficiente de JWT; exposição pública de backends sem WAF/Private Link; segredos fora do Key Vault; falta de rate limiting.
 - Médio: CORS amplo; logs com PII; conectores SaaS com permissões excessivas; XXE em SOAP.
 - Baixo: headers de segurança ausentes; mensagens de erro verbosas.
- Plano de tratamento resumido
 - 0–30 dias: validate-jwt em todas as rotas; rate-limit/quota; migração de segredos para Key Vault; WAF + APIM hardening; bloquear XXE/DTD.
 - 30–90 dias: VNet + Private Link; Managed Identities e RBAC mínimo; mascaramento/retention de logs; políticas de schema validation; DDoS Standard.
 - 90–180 dias: automação IaC com políticas como código; SAST/DAST; tabletop de resposta a incidentes; revisão de fornecedores SaaS e DLP.

Additional resourcing needs:
- Network Administrator: desenho de VNet, Private Link, DDoS/WAF, IP allowlists, DNS privado.
- Security Officer: conformidade (LGPD, políticas corporativas), revisão de controles STRIDE, classificação de dados.
- Cloud/DevOps (Azure): IaC (Bicep/Terraform), pipelines, APIM DevOps, integração com Key Vault e monitoramento.
- Software Developer (APIs): contratos OpenAPI, validação de entrada, versionamento, testes de segurança (SAST/DAST), OWASP API Top 10.
- Hardware Developer: apenas se houver integração com dispositivos; hardening de credenciais embarcadas/TPM.
- IAM/Entra Engineer: modelagem de apps, scopes/roles, PIM, rotação de chaves, Conditional Access.
- SRE/Observability: métricas, logs, alertas, SLOs, runbooks, caos/DR.
- Data Protection Officer/Legal: avaliação de PII, retenção, DPAs com SaaS.
- Vendor Management: due diligence de terceiros, SLA e segurança de conectores.
- API Product Owner: governança de ciclo de vida, catálogos, planos de consumo e limites.

Observações finais:
- Este desenho é nitidamente Azure (Microsoft Entra, API Management, Logic Apps e logo Microsoft Azure). A solução é uma camada de gestão de APIs com orquestração de integrações, adequada para expor serviços internos/externos com segurança, governança e observabilidade. Implementar controles recomendados acima reduz significativamente o risco nas seis classes STRIDE.

Sugestões de melhorias

A seguir estão: 1) o diagrama Mermaid da arquitetura com os controles de segurança aplicados e correções às vulnerabilidades, 2) um script Terraform de referência para provisionamento no Azure, e 3) o Relatório de Modelagem de Ameaças (STRIDE).

1) Diagrama Mermaid (arquitetura e controles)
```mermaid
flowchart LR
  %% Perímetro e identidade
  U[Cliente / Consumidor de API\n(App/Web/M2M)] -->|HTTPS| WAF[Application Gateway WAF (opcional)]
  U -->|OIDC/OAuth2| AAD[Microsoft Entra ID (Azure AD)\nAutenticação/Autorização]
  WAF -->|TLS 1.2+ / WAF rules| APIM[(Azure API Management\nGateway + Developer Portal)]
  U -.->|Onboarding/Docs| DEVPORTAL[[Developer Portal]]

  %% Grupo de Recursos e VNet
  subgraph RG[Resource Group]
    subgraph VNET[VNet (DDoS Standard habilitado)]
      subgraph SUBAPIM[Subnet APIM (Internal)]
        APIM
      end
      subgraph SUBPE[Subnet Private Endpoints]
        PEVault[(Private Endpoint\nKey Vault)]
      end
```

```
    %% (Opcional) Subnet AppGW se WAF for implantado
  end

  LA[Azure Logic Apps\n(Orquestração de integrações)]:::svc
  KV[(Azure Key Vault\nSegredos/Certs/Keys)]:::sec
  MON[(Azure Monitor / Log Analytics)]:::mon
end

%% Backends
subgraph BACK[Backend Systems]
  AF[Azure Functions]:::svc
  SB[Service Bus]:::svc
  STG[Storage]:::svc
  COS[Cosmos DB]:::svc
  SAAS[SaaS (Conectores)]:::ext
  WS[Web services\nREST/SOAP externos]:::ext
end

%% Conectividade e políticas
APIM -->|Chama| LA
APIM -->|Private Link / VNet| AF
APIM -->|Private Link / VNet| SB
APIM -->|Private Link / VNet| STG
APIM -->|Private Link / VNet| COS
LA -->|Conectores| SAAS
LA -->|HTTP/SOAP| WS

%% Segredos e identidade
APIM ==Managed Identity==> KV
LA ==Managed Identity==> KV

%% Observabilidade
APIM -. Diagnósticos .-> MON
LA -. Logs/Métricas .-> MON

%% Autenticação e políticas no APIM
AAD -->|Tokens JWT (iss/aud/exp/nbf, scopes/roles)| APIM

%% Notas de segurança
classDef sec fill=#e6f2ff,stroke=#2f5597,stroke-width=1px,color=#0e2a4c
classDef mon fill=#f4f9f4,stroke=#2f8f2f,stroke-width=1px,color=#1d4d1d
classDef svc fill=#fffaf0,stroke=#c5861a,stroke-width=1px
classDef ext fill=#fff0f0,stroke=#a61b1b,stroke-width=1px

note over APIM: Políticas principais:\n- validate-jwt (iss/aud/exp/nbf, scopes/roles)\n- rate-limit/quota & caching\n- schema validation (JSON/XML), anti-XXE/DTD\n- CORS restrito (origens específicas)\n- Segurança de headers + HSTS\n- Rejeição de métodos/content-type inválidos\n- Circuit breaker/retries com jitter
note over VNET: Endpoints privados (Private Link) para backends\nAPIM em modo Internal (sem IP público)\nDDoS Standard ativo
note over KV: Segredos/keys externos ao app\nRotação via KV + Managed Identity (APIM/Logic Apps)\nSem segredos em código/pipeline
note over WAF: Publicação segura do APIM Internal via App Gateway WAF\nRegras OWASP + IP allowlists
note over LA: Princípio do menor privilégio (RBAC)\nDLP para conectores SaaS\nMascaramento/retention de logs (LGPD)
```

2) Terraform (referência)
Observações:
- Provisiona: Resource Group, VNet (com DDoS), APIM (Internal) com Managed Identity, Logic App (consumption) com Managed Identity, Key Vault com Private Endpoint, Named Value no APIM referenciando segredo no KV, e Diagnósticos do APIM no Log Analytics.
- Em produção, recomenda-se incluir Application Gateway WAF v2 para publicar o APIM Internal; para simplificar o exemplo, o App Gateway não está no script.
- Substitua valores de variáveis conforme seu tenant/domínio. Políticas APIM com CORS especificam origens de exemplo.

```terraform
```

```
terraform {
  required_version = ">= 1.5.0"
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = ">= 3.108.0"
    }
  }
}

provider "azurerm" {
  features {}
}

#########################
# Variáveis
#########################
variable "location" {
  type    = string
  default = "eastus"
}

variable "resource_group_name" {
  type    = string
  default = "rg-apim-logicapps"
}

variable "apim_name" {
  type    = string
  default = "apim-internal-sec"
}

variable "apim_sku" {
  type    = string
  # Developer_1 para não-prod; Premium_1 para prod com SLAs e multi-região
  default = "Developer_1"
}

variable "tenant_id" {
  type        = string
  description = "Tenant ID do Microsoft Entra ID (GUID)."
}

variable "jwt_audience" {
  type        = string
  description = "Audience (Application ID URI ou client_id do app) esperado nos tokens."
}

variable "allowed_origins" {
  type        = list(string)
  default     = ["https://app.example.com", "https://portal.example.com"]
  description = "Origens permitidas no CORS (evitar '*')."
}

variable "tags" {
  type = map(string)
  default = {
    env = "dev"
    app = "apim-logicapps"
  }
}

#########################
# Resource Group
#########################
resource "azurerm_resource_group" "rg" {
  name     = var.resource_group_name
  location = var.location
  tags     = var.tags
```

```
}

#########################
# Observabilidade
#########################
resource "azurerm_log_analytics_workspace" "law" {
  name                = "${var.resource_group_name}-law"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  sku                 = "PerGB2018"
  retention_in_days   = 30
  tags                = var.tags
}

#########################
# DDoS Protection + VNet
#########################
resource "azurerm_network_ddos_protection_plan" "ddos" {
  name                = "${var.resource_group_name}-ddos"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  tags                = var.tags
}

resource "azurerm_virtual_network" "vnet" {
  name                = "${var.resource_group_name}-vnet"
  address_space       = ["10.10.0.0/16"]
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  tags                = var.tags

  ddos_protection_plan {
    id     = azurerm_network_ddos_protection_plan.ddos.id
    enable = true
  }
}

resource "azurerm_subnet" "subnet_apim" {
  name                 = "snet-apim"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.vnet.name
  address_prefixes     = ["10.10.1.0/24"]
}

resource "azurerm_subnet" "subnet_private_endpoints" {
  name                 = "snet-priv-endpoints"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.vnet.name
  address_prefixes     = ["10.10.2.0/24"]
}

#########################
# Key Vault + Secret + Private Endpoint
#########################
resource "azurerm_key_vault" "kv" {
  name                        = "kv-${var.resource_group_name}"
  location                    = azurerm_resource_group.rg.location
  resource_group_name         = azurerm_resource_group.rg.name
  tenant_id                   = var.tenant_id
  sku_name                    = "standard"
  soft_delete_retention_days  = 90
  purge_protection_enabled    = true
  public_network_access_enabled = false
  tags                        = var.tags
}

# Exemplo de segredo (não coloque segredos reais no estado/TF)
resource "azurerm_key_vault_secret" "example_secret" {
  name         = "backend-api-key"
```

```
  value        = "REPLACE_THIS_VALUE_IN_PIPELINE"
  key_vault_id = azurerm_key_vault.kv.id
}

# Private DNS zone para Key Vault
resource "azurerm_private_dns_zone" "kv_priv_dns" {
  name                = "privatelink.vaultcore.azure.net"
  resource_group_name = azurerm_resource_group.rg.name
  tags                = var.tags
}

resource "azurerm_private_dns_zone_virtual_network_link" "kv_priv_dns_vnet_link" {
  name                  = "kv-dnslink"
  resource_group_name   = azurerm_resource_group.rg.name
  private_dns_zone_name = azurerm_private_dns_zone.kv_priv_dns.name
  virtual_network_id    = azurerm_virtual_network.vnet.id
}

resource "azurerm_private_endpoint" "kv_pe" {
  name                = "pe-kv"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  subnet_id           = azurerm_subnet.subnet_private_endpoints.id
  tags                = var.tags

  private_service_connection {
    name                           = "kv-priv-conn"
    private_connection_resource_id = azurerm_key_vault.kv.id
    is_manual_connection           = false
    subresource_names              = ["vault"]
  }

  private_dns_zone_group {
    name                 = "kv-zone-group"
    private_dns_zone_ids = [azurerm_private_dns_zone.kv_priv_dns.id]
  }
}

#########################
# API Management (Internal + MI + baseline policy)
#########################
resource "azurerm_api_management" "apim" {
  name                = var.apim_name
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  publisher_name  = "API Team"
  publisher_email = "api-owner@example.com"

  sku_name = var.apim_sku

  identity {
    type = "SystemAssigned"
  }

  virtual_network_type = "Internal"
  virtual_network_configuration {
    subnet_id = azurerm_subnet.subnet_apim.id
  }

  tags = var.tags
}

# Conceder acesso do APIM ao Key Vault (para Named Values via KV)
resource "azurerm_key_vault_access_policy" "kv_apim" {
  key_vault_id = azurerm_key_vault.kv.id
  tenant_id    = var.tenant_id
  object_id    = azurerm_api_management.apim.identity[0].principal_id
```

```
  secret_permissions = ["Get", "List"]
}

# Named Value do APIM referenciando segredo do KV
resource "azurerm_api_management_named_value" "nv_backend_key" {
  name                = "backend-api-key"
  resource_group_name = azurerm_resource_group.rg.name
  api_management_name = azurerm_api_management.apim.name
  display_name        = "backend-api-key"
  key_vault_secret_id = azurerm_key_vault_secret.example_secret.versionless_id
}

# Política global do APIM (baseline de segurança)
# Ajuste allowed-origins para sua lista; evite '*' em produção
resource "azurerm_api_management_policy" "apim_global_policy" {
  api_management_id = azurerm_api_management.apim.id

  xml_content = <<POLICY
<policies>
  <inbound>
    <base />
    <set-variable name="clientKey" value="@(context.Subscription?.Key ??
context.Request.IpAddress)" />
    <set-header name="X-Content-Type-Options" exists-action="override">
      <value>nosniff</value>
    </set-header>
    <set-header name="Strict-Transport-Security" exists-action="override">
      <value>max-age=31536000; includeSubDomains</value>
    </set-header>
    <set-header name="X-Frame-Options" exists-action="override">
      <value>DENY</value>
    </set-header>
    <set-header name="Content-Security-Policy" exists-action="override">
      <value>default-src 'none'; frame-ancestors 'none';</value>
    </set-header>

    <check-header name="Content-Type" failed-check-httpcode="415"
failed-check-error-message="Unsupported content type">
      <value>application/json</value>
      <value>application/xml</value>
    </check-header>

    <validate-jwt header-name="Authorization" require-scheme="Bearer"
failed-validation-httpcode="401" failed-validation-error-message="Invalid or missing token">
      <openid-config
url="https://login.microsoftonline.com/${var.tenant_id}/v2.0/.well-known/openid-configuration" />
      <required-claims>
        <claim name="aud">
          <value>${var.jwt_audience}</value>
        </claim>
      </required-claims>
    </validate-jwt>

    <cors allow-credentials="false">
      <allowed-origins>
        <origin>${element(var.allowed_origins, 0)}</origin>
        <origin>${length(var.allowed_origins) > 1 ? element(var.allowed_origins, 1) :
"https://app.example.com"}</origin>
      </allowed-origins>
      <allowed-methods preflight-result-max-age="120">
        <method>GET</method>
        <method>POST</method>
        <method>PUT</method>
        <method>DELETE</method>
        <method>OPTIONS</method>
      </allowed-methods>
      <allowed-headers>
        <header>Content-Type</header>
        <header>Authorization</header>
```

```
        <header>Ocp-Apim-Subscription-Key</header>
      </allowed-headers>
      <expose-headers>
        <header>Content-Length</header>
        <header>Content-Type</header>
        <header>ETag</header>
      </expose-headers>
    </cors>

    <rate-limit-by-key calls="100" renewal-period="60" counter-key="@(context.Subscription?.Id ??
context.Request.IpAddress)" />
    <quota-by-key calls="10000" renewal-period="86400" counter-key="@(context.Subscription?.Id ??
context.Request.IpAddress)" />

    <!-- Exemplo de proteção XML (anti-XXE/DTD) e limites; aplique por rota quando usar XML -->
    <xml-threat-protection>
      <max-attributes>256</max-attributes>
      <max-attribute-length>1024</max-attribute-length>
      <max-depth>16</max-depth>
      <max-element-name-length>256</max-element-name-length>
      <max-elements>4096</max-elements>
      <max-entity-expansions>0</max-entity-expansions>
      <max-characters>1048576</max-characters>
      <deny-dtd>true</deny-dtd>
      <ignore-comments>true</ignore-comments>
      <ignore-processing-instructions>true</ignore-processing-instructions>
    </xml-threat-protection>
  </inbound>

  <backend>
    <base />
    <forward-request />
  </backend>

  <outbound>
    <base />
    <set-header name="Cache-Control" exists-action="override">
      <value>no-store</value>
    </set-header>
  </outbound>

  <on-error>
    <base />
    <return-response>
      <set-status code="500" reason="Internal Server Error" />
      <set-header name="Content-Type" exists-action="override">
        <value>application/json</value>
      </set-header>
      <set-body>{"message":"Unexpected error"}</set-body>
    </return-response>
  </on-error>
</policies>
POLICY
}

# Diagnósticos do APIM -> Log Analytics (logs principais; ajuste categorias conforme necessidade)
resource "azurerm_monitor_diagnostic_setting" "apim_diag" {
  name                       = "apim-diagnostics"
  target_resource_id         = azurerm_api_management.apim.id
  log_analytics_workspace_id = azurerm_log_analytics_workspace.law.id

  enabled_log {
    category = "GatewayLogs"
  }

  enabled_log {
    category = "WebSocketConnectionLogs"
  }
```

```
  metric {
    category = "AllMetrics"
    enabled  = true
  }
}

#########################
# Logic App (Consumption) + MI
# Em produção, considerar Logic App Standard em VNet
#########################
resource "azurerm_logic_app_workflow" "la" {
  name                = "la-orchestrator"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  tags                = var.tags

  identity {
    type = "SystemAssigned"
  }
}

# Acesso do Logic App ao Key Vault
resource "azurerm_key_vault_access_policy" "kv_la" {
  key_vault_id = azurerm_key_vault.kv.id
  tenant_id    = var.tenant_id
  object_id    = azurerm_logic_app_workflow.la.identity[0].principal_id

  secret_permissions = ["Get", "List"]
}

#########################
# Saídas
#########################
output "apim_private_name" {
  value = azurerm_api_management.apim.name
}

output "key_vault_uri" {
  value = azurerm_key_vault.kv.vault_uri
}

output "logic_app_id" {
  value = azurerm_logic_app_workflow.la.id
}

output "log_analytics_workspace_id" {
  value = azurerm_log_analytics_workspace.law.id
}
```

Notas de implantação e endurecimento adicional
- Publicação segura do APIM Internal: posicione um Application Gateway WAF v2 em subnet dedicada e faça o routing para o APIM via private IP. Aplique regras OWASP, mTLS (se necessário) e IP allowlists.
- Backends: configure Private Link para Storage/Cosmos/Service Bus/Functions conforme usados; use Private DNS Zones correspondentes.
- Logic Apps Standard: para VNet integration e conectores privados, adotar SKU Standard com integração de rede; mover segredos para Key Vault e usar Managed Identities.

3) Relatório de Modelagem de Ameaças (STRIDE)

Escopo e ativos
- Escopo: Exposição de APIs via Azure API Management com orquestração em Logic Apps integrando serviços Azure, SaaS e Web (REST/SOAP).
- Ativos: Endpoints de API, tokens JWT/refresh tokens, chaves/subscriptions, segredos de conectores, dados sensíveis em trânsito/repouso, políticas APIM, workflows Logic Apps, logs/telemetria.

Atores
- Consumidor legítimo; Desenvolvedor (Developer Portal); Admins de APIM/Logic Apps; Serviços

back-end; Atacante externo; Terceiros (SaaS).

Fronteiras de confiança
- Internet → WAF/APIM (perímetro).
- APIM → Logic Apps (intra-Azure, idealmente VNet/Private Link).
- APIM/Logic Apps → Backends (Azure/SaaS/Web).
- Dev Portal/DevOps → Administração (RBAC/Azure AD).

Fluxos de dados
- HTTPs requests, tokens OIDC/JWT, payloads JSON/XML, segredos de conectores, logs/telemetria.

STRIDE por categoria
- Spoofing
  - Riscos: falsificação de cliente; tokens forjados; backends falsos.
  - Mitigações: validate-jwt no APIM (iss/aud/exp/nbf, scopes/roles); mTLS onde crítico; Private DNS/DNSSEC; Managed Identities; validação de certificado/host pinning em policies.
- Tampering
  - Riscos: alteração de payloads/políticas; manipulação de workflows.
  - Mitigações: TLS forte; schema validation (JSON/XML), anti-XXE; assinatura JWS quando necessário; mudanças via IaC + PR/review; RBAC + PIM; versionamento e approval gates.
- Repudiation
  - Riscos: negação de autoria de chamadas/admin; logs incompletos.
  - Mitigações: logging imutável (Storage Immutability/Legal Hold); correlação (trace-id); auditoria Entra; timestamp confiável; retenção/proteção contra edição.
- Information Disclosure
  - Riscos: vazamento de tokens/segredos/PII; erros verbosos; logs sensíveis.
  - Mitigações: Key Vault; mascaramento no APIM/Logic Apps; mensagens de erro genéricas; classificação de dados; criptografia em repouso (CMK se preciso); Private Link; escopos mínimos nos tokens.
- Denial of Service
  - Riscos: flood no APIM; fan-out em Logic Apps; endpoints pesados; picos em backends.
  - Mitigações: rate-limit/quota; caching; circuit breaker/backoff; dimensionamento adequado; DDoS Protection Standard; fila assíncrona (Service Bus) para workloads intensivos.
- Elevation of Privilege
  - Riscos: políticas permissivas; identidades com papéis amplos; JWT com escopos mal definidos.
  - Mitigações: RBAC mínimo; PIM para admins; segmentação por RG/subscrição; escopos/roles granulares no Entra; revisão periódica de acessos; JIT.

Riscos principais priorizados
- Alto: validação insuficiente de JWT; backends expostos sem WAF/Private Link; segredos fora do Key Vault; ausência de rate limiting.
- Médio: CORS amplo; logs com PII; conectores SaaS com permissões excessivas; XXE em SOAP.
- Baixo: headers de segurança ausentes; mensagens de erro verbosas.

Plano de tratamento
- 0–30 dias: aplicar validate-jwt em todas as rotas; rate-limit/quota; migrar segredos para Key Vault; endurecer APIM + WAF; bloquear XXE/DTD.
- 30–90 dias: VNet + Private Link; Managed Identities e RBAC mínimo; mascaramento/retention de logs; schema validation; DDoS Standard.
- 90–180 dias: IaC com políticas como código; SAST/DAST; tabletop de incidentes; revisão de fornecedores SaaS e DLP.

Recursos adicionais necessários
- Network Admin: VNet, Private Link, DDoS/WAF, IP allowlists, DNS privado.
- Security Officer: conformidade (LGPD), revisão STRIDE, classificação de dados.
- Cloud/DevOps: IaC (Terraform), pipelines, APIM DevOps, integração KV/monitoramento.
- Devs de API: OpenAPI, validação de entrada, versionamento, testes de segurança (OWASP API Top 10).
- IAM/Entra: modelagem de apps, scopes/roles, PIM, rotação/CA.
- SRE/Observabilidade: métricas, logs, alertas, SLOs, runbooks, DR/chaos.
- DPO/Legal e Vendor Management conforme aplicável.

Se desejar, posso complementar o Terraform com App Gateway WAF v2 e um exemplo de Private Link para um backend específico (Storage/Cosmos/Service Bus) e/ou Logic App Standard com VNet.