Aula 1

Introdução a Cyber Security: Conhecendo as vulnerabilidades da aplicação (parte 1)

PÓSPUCPRDIGITAL | PUCPR GRUPO MARISTA

Quem é o professor?

Izidio Rosa

Formado em processamento de dados

Pós graduado em Segurança da Informação e outros muitos cursos na área

Faz tempo que sou apaixonado por tecnologia (década de 90).

Vivi o início da internet no Brasil

Trabalho profissionalmente com TI desde 2.000, sempre estudando segurança da informação

Estou há 10 anos em uma empresa do ramo financeiro

Há 4 anos iniciei minha transição de líder técnico para gestor de time

Sou casado a 15 anos

Pai de uma linda menina e um lindo menino

Cyber Security: Conhecendo vulnerabilidades

Agenda – Aula 1

- Termos e Conceitos
- Por que devemos conhecer vulnerabilidades?
- Relembrar é viver:
  - Regras de SI
  - Pilares de SI
  - O que é criptografia?
- O que é ou quem é OWASP?
  - Quais são as categorias de vulnerabilidades que abordaremos?
  - Sugestões de estudo de ferramentas open

Termos e conceitos

1. **Ativos**:
    a. Depende da área que aplica:
        a. REDES: ativos são máquinas conectadas no ambiente de trabalho.
        b. SI é mais amplo: tudo que possui valor para a organização.
    b. Como exemplo,
        i. o quadro de funcionários: possui expertise sobre o negócio da empresa e seus processos.
        ii. as propriedades físicas ou digitais: as casas, os edifícios, os computadores, os domínios digitais, as páginas de internet e os sistemas produzidos
        iii. As informações: é a inteligência do negócio, os segredos de negócio, a carteira de clientes, etc.

Em resumo: é tudo que possui valor intrínseco ou real

Termos e conceitos

2. **Vulnerabilidades**: são as fraquezas que um ativo possui.
   - resulta na quebra de um dos pilares de SI. (veremos logo mais quais são)

3. **Ameaça**: Ameaças é tudo que pode afetar negativamente um ativo.
   - Não é limitado as vulnerabilidades da aplicação.
   - Pode ser um erro no processo
   - Pode ser intencional ou acidental.

4. **Incidente**: É a exploração da vulnerabilidade
   a. Gerar um impacto negativo para a organização, sistema ou indivíduo.
   b. Perda financeira, um dano a imagem, um vazamento de dados, a destruição ou sequestro dos dados ou simplesmente a indisponibilidade

Termos e conceitos

5. **Impacto**: A forma de medir os possíveis resultados da exploração de uma vulnerabilidade.
   - Para avaliarmos o impacto, algumas perguntas precisam ser feitas.
   - Exemplos - aumentar conforme necessidade.

   a. O sistema que está vulnerável é essencial ao negócio?
   b. A vulnerabilidade causaria constrangimento a terceiro ou colaborador?
   c. Qual o custo ao negócio?
   d. Vulnerabilidade Fácil de ser explorada?
   e. Qual o custo para consertar?

Termos e conceitos

Resultado (medir): impacto

- muito baixo, baixo,
- médio, alto,
- muito alto ou crítico.

Impacto alto, muito alto ou crítico:

uma das respostas causaria um dano sério.

Exemplo: A vulnerabilidade – se explorada – causaria constrangimento a terceiro ou colaborador (vazamento de dados onde pessoas de saúde, dívidas, etc).

6. **Risco**: probabilidade de ocorrer e impacto determinado.

Por que conhecer vulnerabilidades?

- Conhecer os problemas, ajuda a determinar onde devemos investir.
  - Ex.: solução para monitoramento de conformidade
- Conhecer os problemas, ajuda a conhecer nossas deficiências
  - Ex.: meus códigos sempre são muito vulneráveis, invisto em treinamentos
- Conhecer os problemas, ajuda a na eficiência do negócio (agrega valor ao negócio)
  - Ex.: quanto menor o número de vulnerabilidades, mais estável/disponível é

Reflexão: Na prática!

Imagine: - Sistema de saúde (dados de saúde)

Vulnerabilidade: Brute force - login

Risco: Impacto (se ocorrer) vs Probabilidade

Impacto: Vazamento de dados de saúde – alto

Probabilidade: custo baixo, pode ser automatizado, comum/fácil

Vamos piorar!

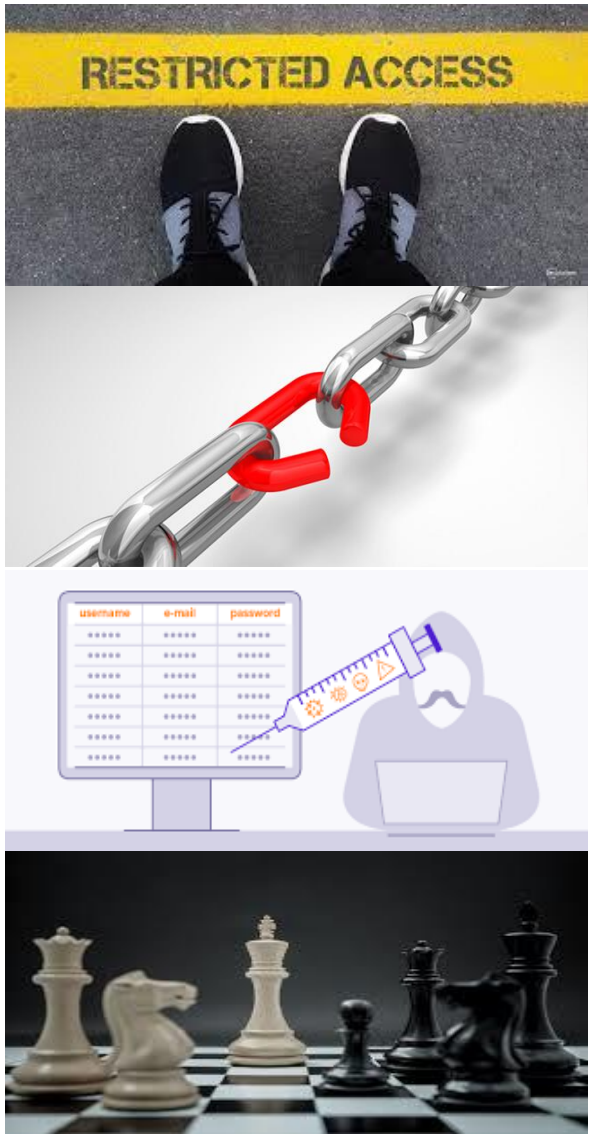
Fatores externos: Nesta mesma situação hipotética

- Adicione campanhas de marketing para uso do sistema ou atração de cliente.
  - Aumento de visibilidade
  - Possibilidade de phishing

Relembrar é viver:

Regras:

- Princípio do menor privilégio (PoLP - Principle of Least Privileged)
- Sua segurança é tão forte quanto o seu elo mais fraco (pessoas, processos e tecnologia)
- Todo acesso ou envio de dados é mal intencionado, até que se prove o contrário
- Dividir para conquistar



Relembrar é viver:

3 principais pilares (CID)

1. Confidencialidade
2. Integridade
3. Disponibilidade

+3 pilares importantes

1. Conformidade
2. Autenticidade
3. Não repúdio

O que é Criptografia?

- em grego: kryptós, "escondido", e gráphein, "escrita")

- A criptografia envolve o uso de uma chave criptográfica, um conjunto de valores matemáticos com os quais tanto o remetente quanto o destinatário concordam.

- Existem muitos tipos de criptografia, mas relembraremos apenas 2.
  - Simétrica e Assimétrica

Chave simétrica

Característica:

Chave única para ambos os processos

Ponto forte:

Baixo custo computacional

Ponto fraco:

Precisa garantir o meio de transmissão da chave simétrica

Chave Assimétrica

Característica:

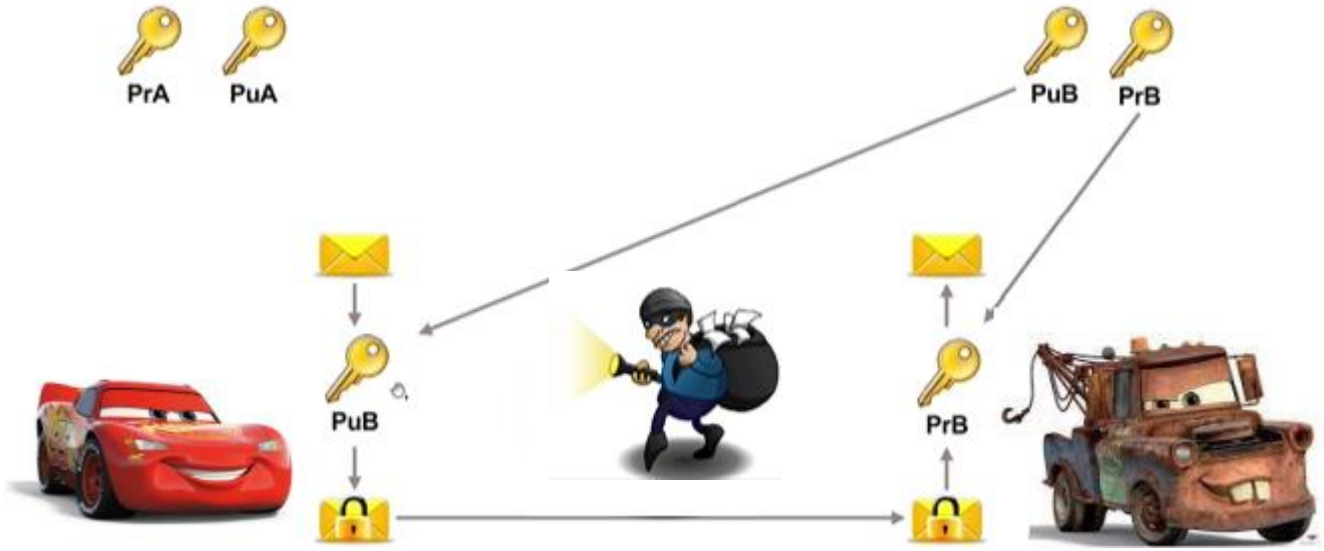
Par de chaves (pública e privada)

Ponto forte:

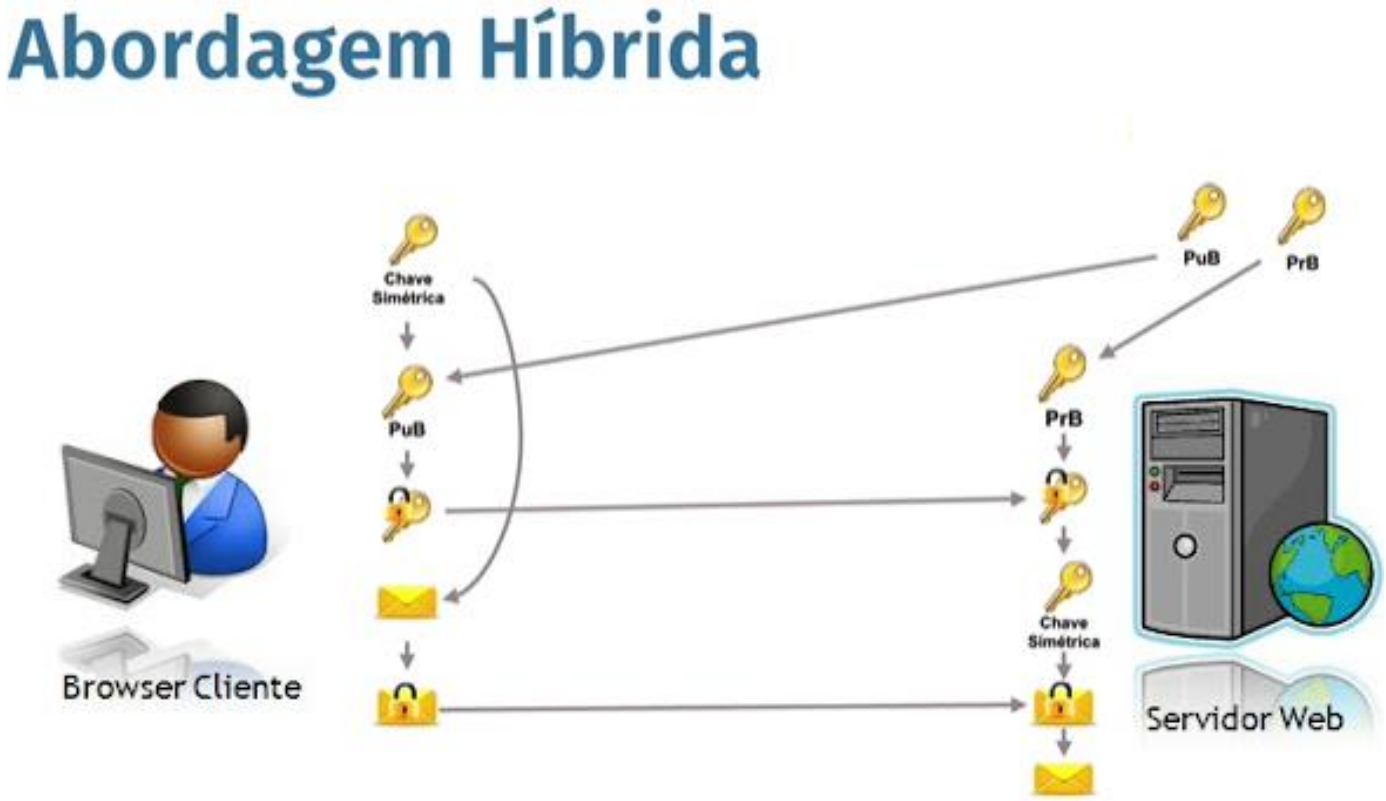
Meio de transmissão não é uma preocupação

Ponto fraco:

Alto custo computacional



E o HTTPS?



Característica:

Par de chaves (pública e privada) + chave simétrica

Ponto forte:

Meio de transmissão não é uma preocupação

Baixo custo computacional

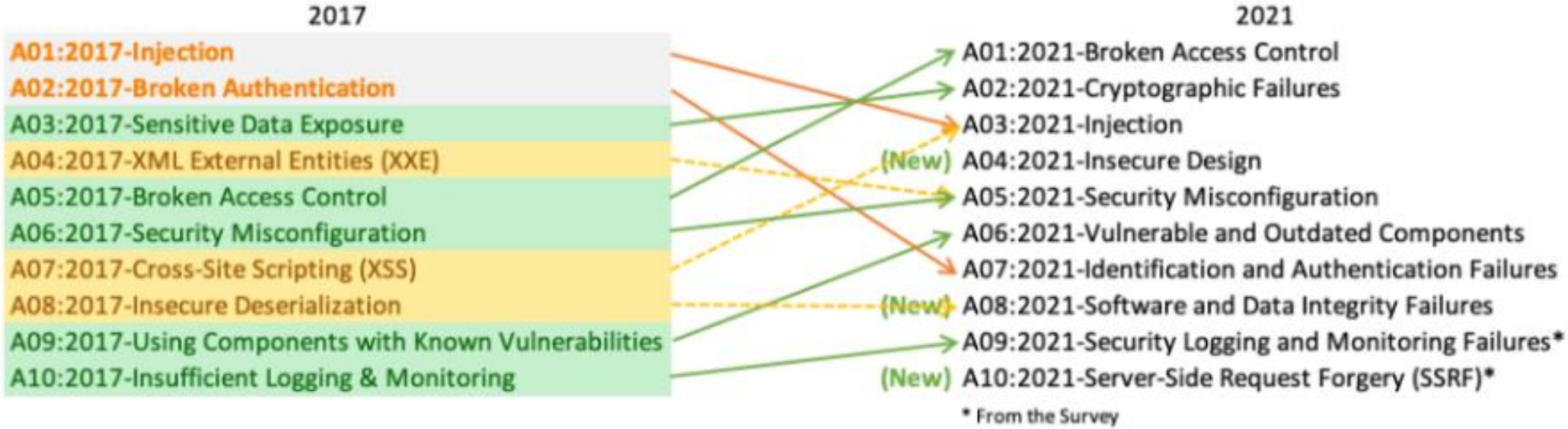
Quem é OWASP?

- “Open Web Application Security Project” ou Projeto Aberto de Segurança em Aplicações Web
- Comunidade online
    - Presente mundialmente – dividida em capítulos
    - Sem fins lucrativos
- Objetivo
    - Conscientizar
- Como?
    - Realiza treinamentos
    - Disponibiliza ferramentas Open
    - Disponibiliza documentação (ex.: TOP 10)
    - Organiza vulnerabilidades por categoria

TOP TEN

Top 10 Web Application Security Risks

There are three new categories, four categories with naming and scoping changes, and some consolidation in the Top 10 for 2021.



Categorias de vulnerabilidades

Ano de referência: 2021

1. Quebra de controle de acesso
2. Falhas de criptográficas
3. Injeções
4. Design Inseguro
5. Configurações incorretas de segurança
6. Componentes vulneráveis e desatualizados
7. Falhas de identificação e autenticação
8. Falhas de integridade de software e dados
9. Monitoramento de segurança e falhas de registro
10. Falsificação de solicitação do lado do servidor

Ferramentas e documentos

OWASP Dependency-Check

- Software Composition Analysis (SCA): ferramenta que testa e detecta vulnerabilidades contidas em um projeto de acordo com a cadeia de dependências. Determinada se há Common Platform Enumeration (CPE) e identifica fazendo vinculo a CVE correspondente. Usado para fazer um “SCAN” (varredura) na aplicação e suas bibliotecas. Útil para controles as vulnerabilidades de 2 categorias já mencionadas

OWASP Dependency-Track Monitors Component

Plataforma inteligente para análise de componente que permite as organizações identificar e reduzir risco.

A plataforma traz vários benefício e é compatível com Software Bill Of Material (SBOM).

Traz mais informações e visão mais profunda que o SCA e pode ser integrada em esteira CI/CD, pois possui integrações por API.

Ferramentas e documentos

ZAP (Zed Attack Proxy)

O OWASP ZAP é muito utilizado de forma manual por quem está executando o teste de vulnerabilidade, mas ele também fornece um conjunto de APIs permitindo que o programador automatize através de scripts a execução dos testes.

Documentos:

- OWASP TOP 10 – web site
- TOP 10 Mobile Risks
- TOP 10 Privacy Risk
- CISO Report
- CISO Guide

Aula 2

Introdução a Cyber Security: Conhecendo as vulnerabilidades da aplicação (parte 2)

PÓSPUCPRDIGITAL | PUCPR GRUPO MARISTA

Cyber Security: Conhecendo vulnerabilidades

Agenda – Aula 2

Visão do CISO

1. Missão
2. Ameaças e riscos
3. Investimentos
4. Desafios

Riscos à Privacidade

Modelagem de ameaça

1. Por que fazer?
2. Quais são as premissas?

Gestão de vulnerabilidade

1. Qual o ganho?
2. Rotinas?

Visão do CISO: Missão

- Definir estratégia de Cyber (ex.: application security) para atender o Business
- Incentivar o conhecimento em Cyber (treinamento e cultura)
- Promover as boas práticas (configurações e arquitetura)
- Atender Compliance e auditorias internas
- Gerenciar riscos (onde sou vulnerável)
- Mitigar riscos
- Gerenciar controles
- Gerenciar incidentes
- Desenvolver um time forte de Security

Ameaças e vulnerabilidades

- Ameaças externas estão crescendo
- O negócio agora é digital (via aplicativo) e os riscos de aplicativos aumentaram
- Impacta o negócio: risco a imagem / perda de cliente ou receita
- Preocupação: não quero ser a próxima notícias.
- APPs pode ser a fonte de vazamentos (não só phishing, ransonware)
- **05 problemas apontados por CISOS (em 2013)**
  - Falta consciência dos problemas de segurança
  - O desenvolvimento é inseguro
  - Metodologias de testes fracas ou inadequadas
  - Falta de orçamento
  - Falta de habilidade da equipe

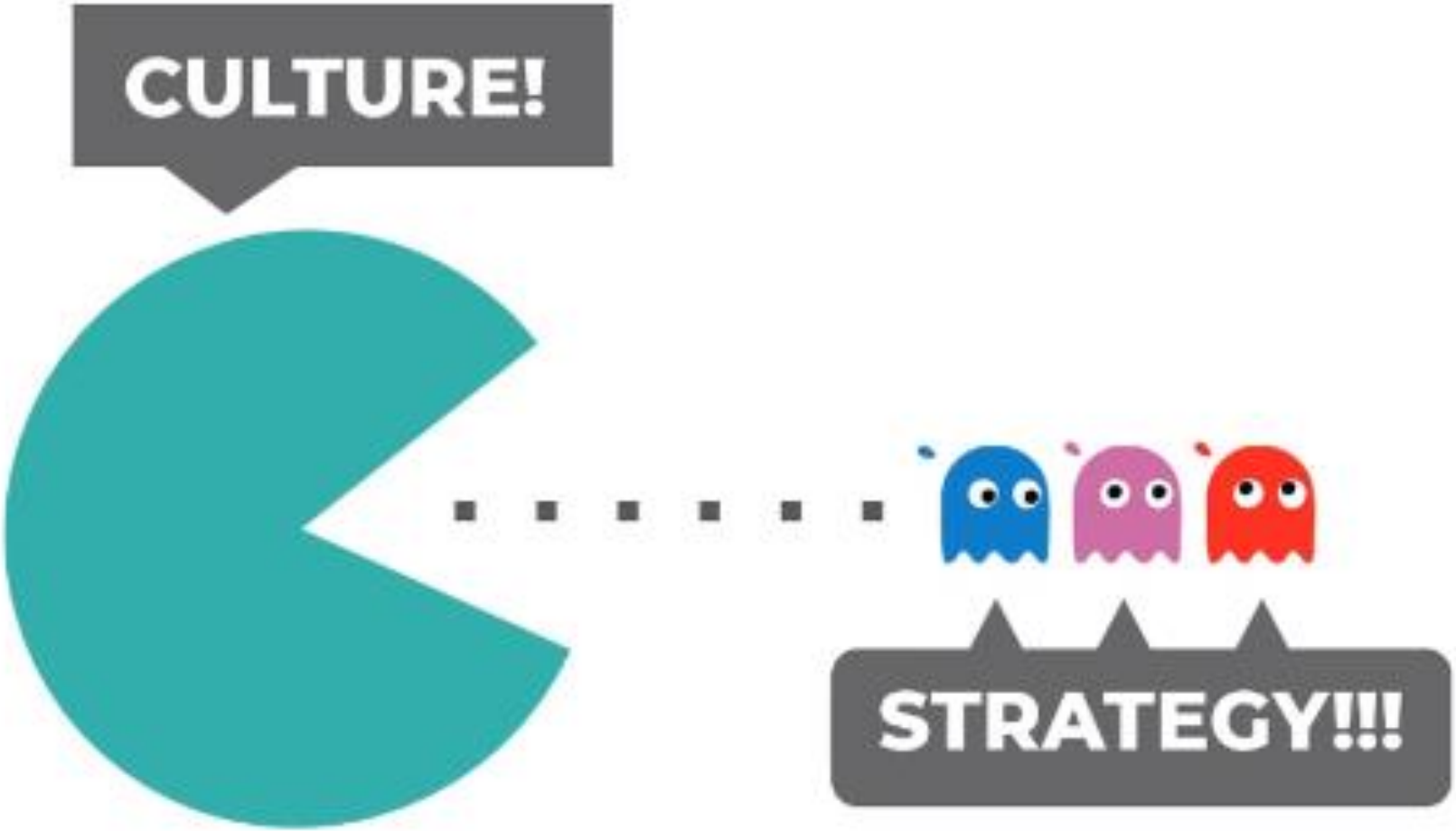
Investimentos

- Promover o conhecimento e treinamento para desenvolvedores
- Investir em testes de segurança em aplicativos (durante o desenvolvimento, antes de produção e pós produção)
- Aprimorar tecnologias e processos de gerenciamento de vulnerabilidades da camada do aplicativo
- Aprimorar controles e logs

Desafios

- Promover o investimento contínuo em application security
- Especificar requisitos de segurança aderentes ao negócio
- Gerenciar riscos
- Documentar e aplicar as diretrizes de segurança
- Torna-se aderente à security frameworks
- Promover arquitetura segura
- Promover o ambiente de implementação bem configurado
- Estabelecer um processo de gestão de vulnerabilidades
- Revisões de códigos
- Testar casos de uso comuns de segurança (Top 10)
- Promover o ciclo de vida de desenvolvimento seguro
- Realizar a modelagem de ameaça do aplicação

ATENÇÃO:



Riscos à Privacidade

Picture source: thedailydose.com

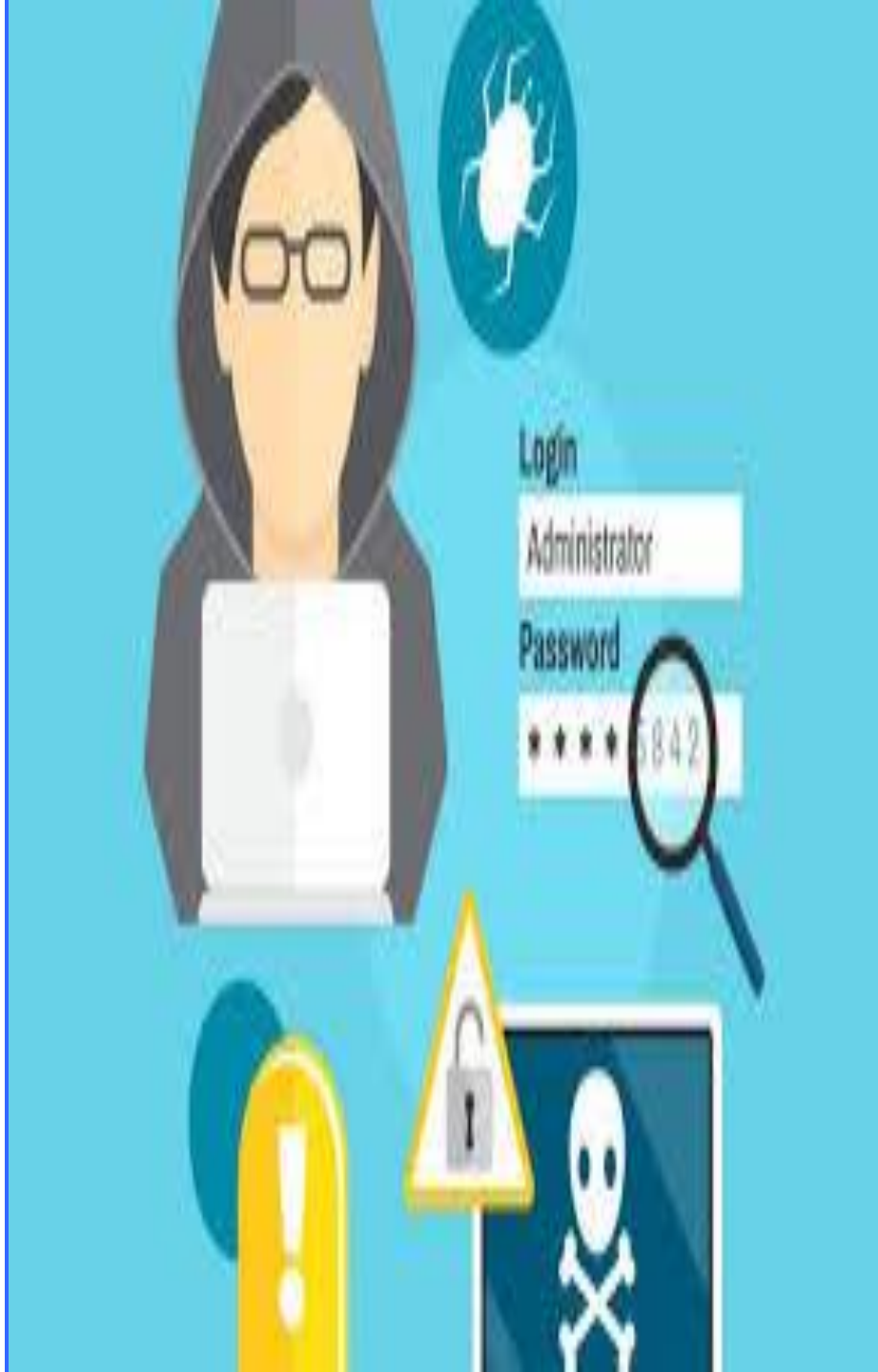
Riscos à Privacidade

Problemáticas: Aplicações web com riscos de privacidade

- TOP 10 (by OWASP 2014)
  1. Aplicações web vulneráveis
  2. Vazamento de dados pelo operador
  3. Resposta de violação de dados insuficiente
  4. Exclusão de dados pessoais insuficiente
  5. Políticas, termos e condições sem transparência
  6. Coleta de dados não exigida para o propósito principal
  7. Compartilhamento de dados com terceiros
  8. Dados pessoais desatualizados
  9. Ausência de expiração de sessão ou controle insuficiente
  10. Transferência de dados insegura

Recomendação: faça modelagem de ameaça

A “Modelagem de Ameaças” é um procedimento para otimizar a segurança do aplicativo, sistema ou processo de negócios, identificando objetivos e vulnerabilidades e, em seguida, definindo contramedidas para prevenir ou mitigar os efeitos das ameaças ao sistema.



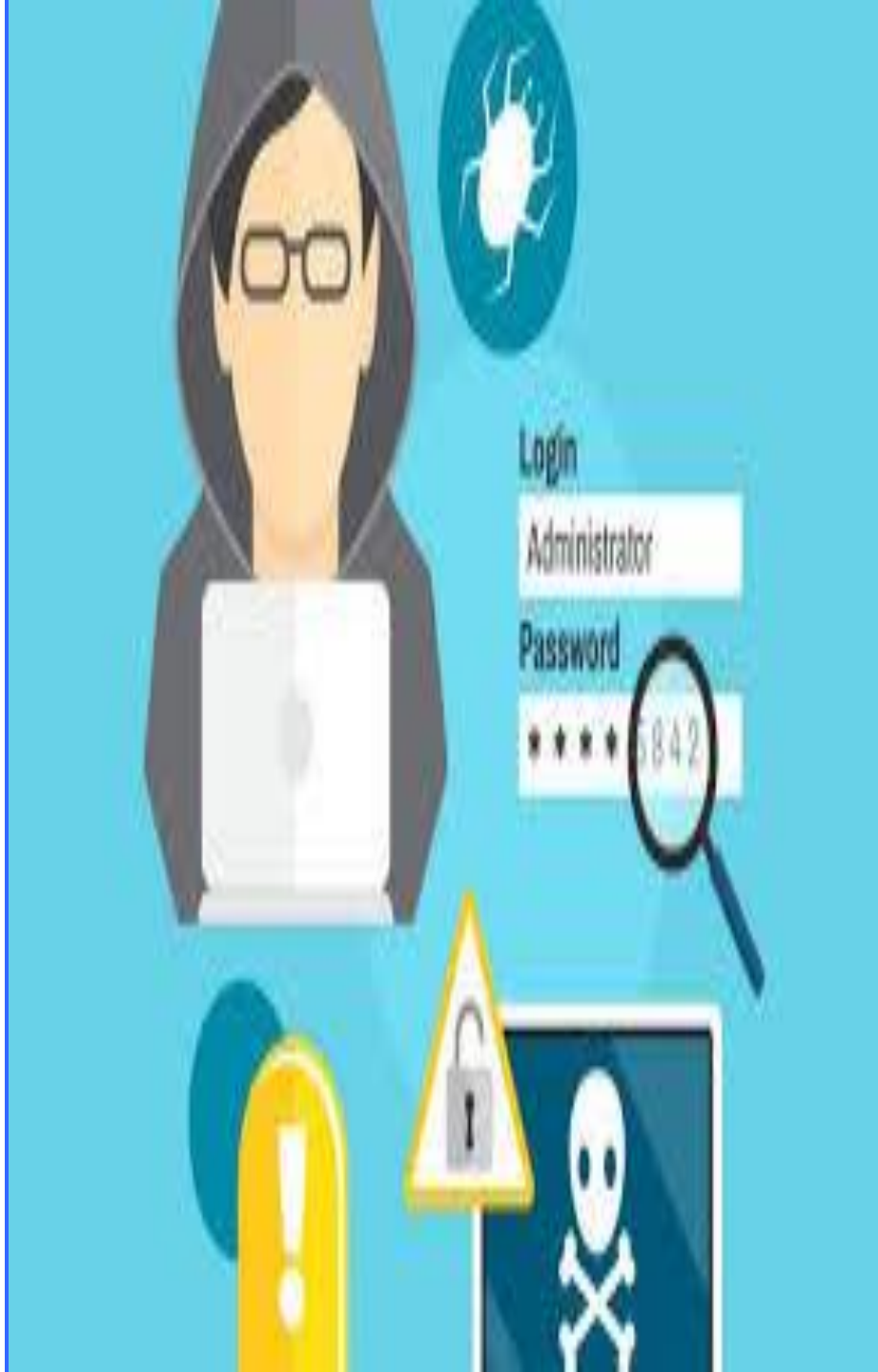
Modelagem de ameaça

Por que fazer?

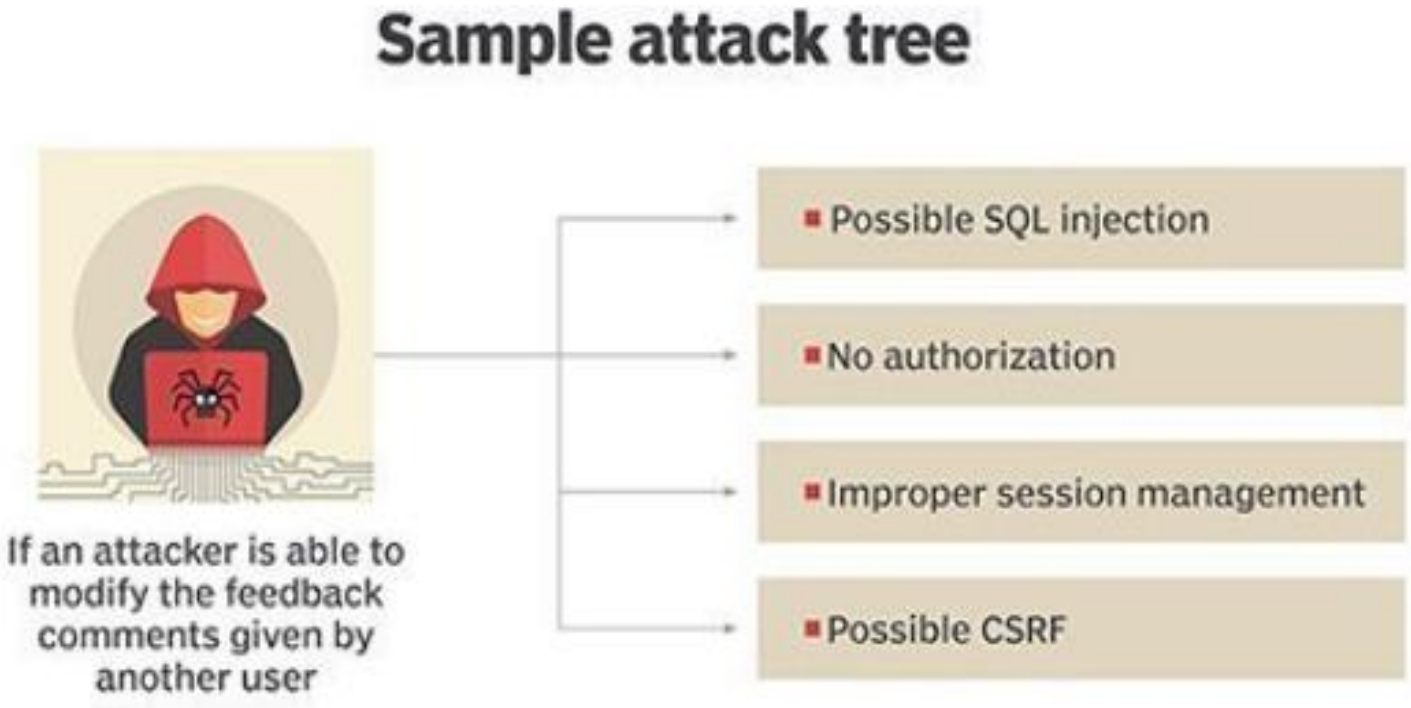
- Atacantes pensam diferentes
- Precisamos determinar controles
- Fortalece a cultura
- Otimiza os testes

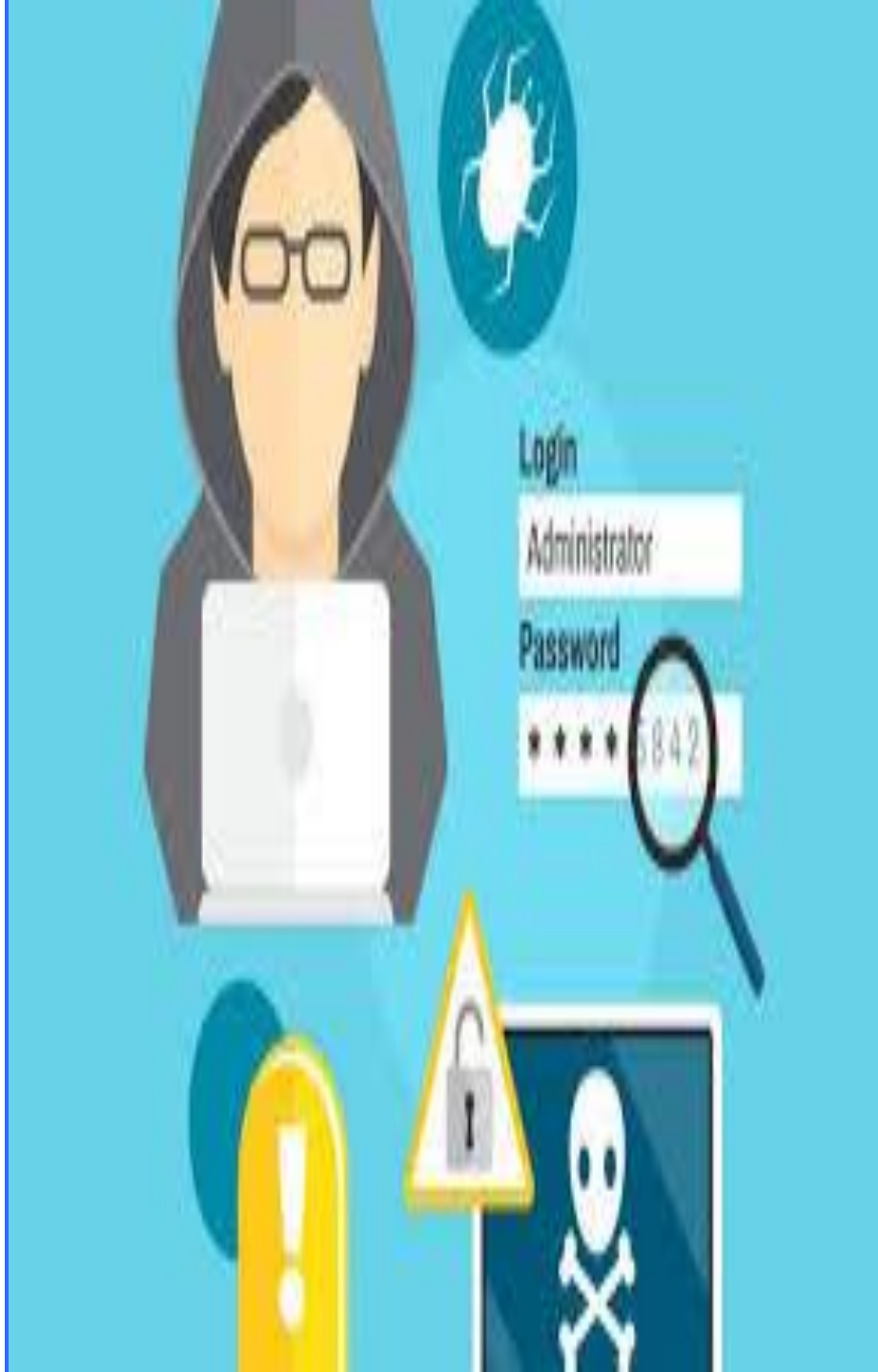
Quais são as premissas?

- Equipe multidisciplinar
- Precisa ter um escopo
- Crie cenários de ataques
- Classifique
- Mitigue
- Documente



Modelagem de ameaça





Automatização

- Scan de vulnerabilidades para aplicação
  - Revisão de código
  - Testes de aplicativos dinâmicos
  - Testes interativos
- Soluções para avaliar componentes
- Solução para controlar imagens para Containers
- Solução para gestão de multi-cloud

Ferramentas de defesa:

- SOAR – security orchestration automation and response
- WAF – Web Application Firewall
- Firewall
- Sanbox (upload de arquivo)

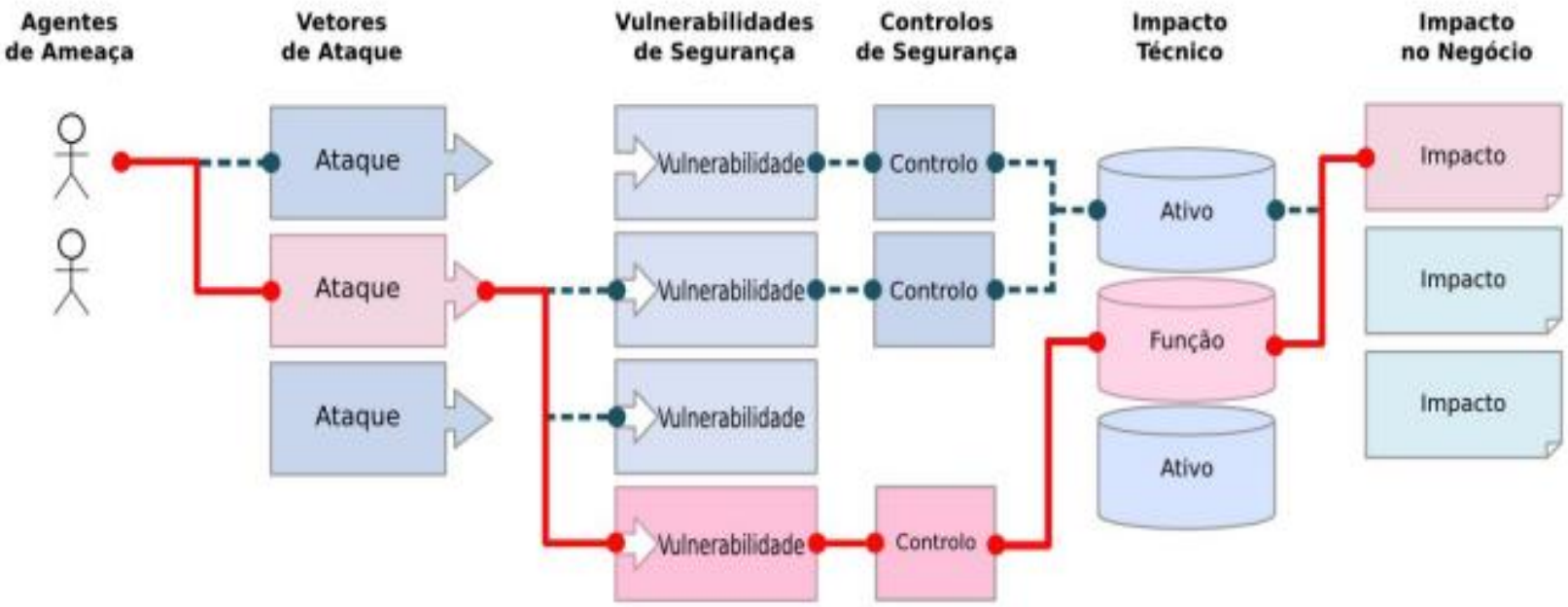
Gestão de vulnerabilidade

Falamos de processo!

- Quais tipos de scan (app, rede, srv)?
- Qual a periodicidade do scan?
- Quem estimará o impacto? E se tiver dados pessoais?
- Quem determinará o risco?
- Qual o prazo de correção?
- Quais controles serão estabelecidos até a correção?
- Quais controles serão estabelecidos após a correção?
- Quem avaliará o processo?
- E o risco do processo?
- E o ambiente com arquitetura incorreta que não aparecerá em um SCAN?

Gestão de vulnerabilidade

Metodologia do ataque



Gestão de vulnerabilidade

Medir o risco

Agente de Ameaça	Vetor de Ataque	Prevalência da Vulnerabilidade	Facilidade de Deteção da Vulnerabilidade	Impacto Técnico	Impacto de Negócio
	Fácil	Generalizada	Fácil	Severo	
?	Médio	Comum	Médio	Moderado	?
	Difícil	Pouco comum	Difícil	Menor	

Aula 3

Categorias: A01 e A02

PÓSPUCPRDIGITAL | PUCPR GRUPO MARISTA

Agenda – Aula 3

Categorias:

- A01 Broken Access Control
- A02 Cryptographic Failures

A01 Broken Access Control

- A quebra de autorização ocorre quando um usuário, mesmo que autêntico, consiga acessar funcionalidades exclusivas de outro usuário de perfil superior, constatando um ataque de escalação de privilégio.

Vulnerabilidade de segurança

Todo site, endpoint ou API que possua acesso exclusivo a perfis de usuários distintos se torna um vetor de ataque.

A quebra pode ser tanto de funcionalidade (acesso a consultas) quanto a recursos (dados)

Exploração: Média

Controles de segurança

1. Processo de gestão de perfis
   - Quão segregado são os perfis?
     - Único perfil
     - Administrador e usuários
     - N perfis
   - Quais ferramentas automatizadas são usadas?
2. Testes de segurança
   - Ataques simulando usuários com perfis diferentes
3. Monitoramento de comportamento
   - Qual é o normal do usuário?

Mitigação: Médio

Controles de segurança

4. Evitar obscuridade.
   Não suponha que o erro não será descoberto
5. Mantenha o Princípio do menor privilégio
   Menos é mais! (menos acesso, mais segurança)
6. Validação abstrata
   Crie classes de segurança
7. Realize auditoria nos processo
   Quem é a fonte autoritativa?
   Quando há a troca de perfil?
   Acúmulo de função?

Mitigação: Médio

Impacto técnico

O sucesso na exploração dessa vulnerabilidade permite a um atacante obter acesso a funcionalidades sem a devida autorização

1. Exposição de dados à pessoa não autorizada
2. Permissão de alteração ou exclusão à pessoa não autorizada
3. Manutenção inapropriada

Impacto: Alto

Impacto ao negócio

1. Exposição de dados protegidos
2. Interrupção do negócio
3. Risco à imagem
4. Multas

Impacto: Alto

Cenário de exemplo

01)

Cadastro: onde há o envio do perfil (role) na requisição, porém, sem as devidas validações no front e back-end.

```
POST        /api/sistema/cadastro/novo    HTTP/1.1
Host:       site1.com
Accept-Language: pt
User-Agent: Mozilla/5.0
Content-Length: 40
Autorization: Basic mvJhujksuJSHXSkjshHSUKJIUHksjkuytgBhgFRhJUY=
{
"username": "jose@email1.com"
"role":"user"
}
```

Cenário de exemplo

01)

Cadastro: onde há o envio do perfil (role) na requisição, porém, sem as devidas validações no front e back-end.

```
POST      /api/sistema/cadastro/novo   HTTP/1.1
Host:     site1.com
Accept-Language: pt
User-Agent: Mozilla/5.0
Content-Length: 40
Autorization: Basic mvJhujksuJSHXSkjshHSUKJIUHksjkuytgBhgFRhJUY=
{
"username": "jose@email1.com"
"role":"user"
}
```

“role”:“administrator”

Cenário de exemplo

02)

A URL para listar todos os usuários não valida se o usuário que está acessando é administrador.

```
POST        /api/sistema/admin/usuário/listar   HTTP/1.1
Host:       site1.com
User-Agent: Mozilla/5.0
Content-Length: 60
Autorization: Bearer [toke]
```

Cenário de exemplo

03)

A aplicação usa dados não validado nas querys do SQL. A chamada visa trazer informações da conta.

Imagine o código:

```
pstmt.setString(1, request.getParameter("acct"));
ResultSet results = pstmt.executeQuery( );
```

Cenário de exemplo

03)

A aplicação usa dados não validado nas querys do SQL. A chamada visa trazer informações da conta.

Chamada

```
https://example.com/app/accountInfo?acct=notmyacct
```


Não há validação para o parâmetro “ACCT”

A02 Cryptographic Failures

- A falha na criptografia ocorre quando o algoritmo utilizado na proteção do dado é reversível sem ter a chave privada. Ou seja, os cálculos usado para a proteção da informação é insuficiente em entropia e/ou complexidade para garantir a proteção.

Vulnerabilidade de segurança

Amplo aspecto!

O problema está no algoritmo! Logo, tudo que faz uso de um algoritmo ruim será afetado.

- Serviços (SMTP, HTTPS, FTPS, SFPT, SCP e outros)
- Criptografia de disco
- Criptografia de arquivos
- Criptografia de dados

Exploração: Difícil

Controles de segurança

1. Mapeamento: saiba onde está sendo utilizado criptografia
2. Força: use os recomendados
3. Armazenamento: armazene o necessário
4. Oriente: traga a visibilidade sobre a fragilidade
5. Audite: valide se o orientado ou o mapeamento estão corretos
6. Centralize: utilize soluções de segurança que aplicam TLS para todos os serviços expostos (API Gateway, WAF, etc)
7. Exigir no cabeçalho o uso de criptografia (HSTS – http strict transport security)
8. Monitore: gere alertas

Mitigação: Médio

Controles de segurança

1. Existem dados transmitidos sem criptografia na Internet?
2. Existem dados sendo trafegados na rede interna sem criptografia? Por exemplo, comunicação entre servidores ou balanceadores?
3. Você faz uso de algoritmo de criptografia fraco ou descontinuado? Considere tanto criptografia em repouso, quanto em trânsito.
4. Sua aplicação possui HEADERs (como o HSTS) que obriga o trânsito de dados com protocolos seguros (como o HTTPS)?
5. Sua empresa possui diretivas de segurança claras quanto aos cabeçalhos de segurança?
6. O certificado digital e sua respectiva cadeia estão devidamente registrados?
7. As senhas dos certificados digitais são armazenadas em “cofres” apropriados – HSM (hardware security module)?
8. Você faz uso de HASH obsoletos?
9. Há versões obsoletas de protocolos?
10. Os dados estão sendo salvos em locais com criptografia?
11. Arquivos gerados a partir do sistema também são salvos em locais com criptografia?

Mitigação: Médio

Impacto técnico

O sucesso na exploração dessa vulnerabilidade permite a um atacante obter acesso a dados que deveriam estar protegidos.

1. Dados cadastrais,
2. Dados sensíveis,
3. senhas,
4. Movimentações financeiros,
5. e-mails
6. Base de dados
7. Disco rígidos e outros.

Impacto: Alto

Impacto ao negócio

1. Risco à imagem
2. Multas

Impacto: Alto

Cenário de exemplo

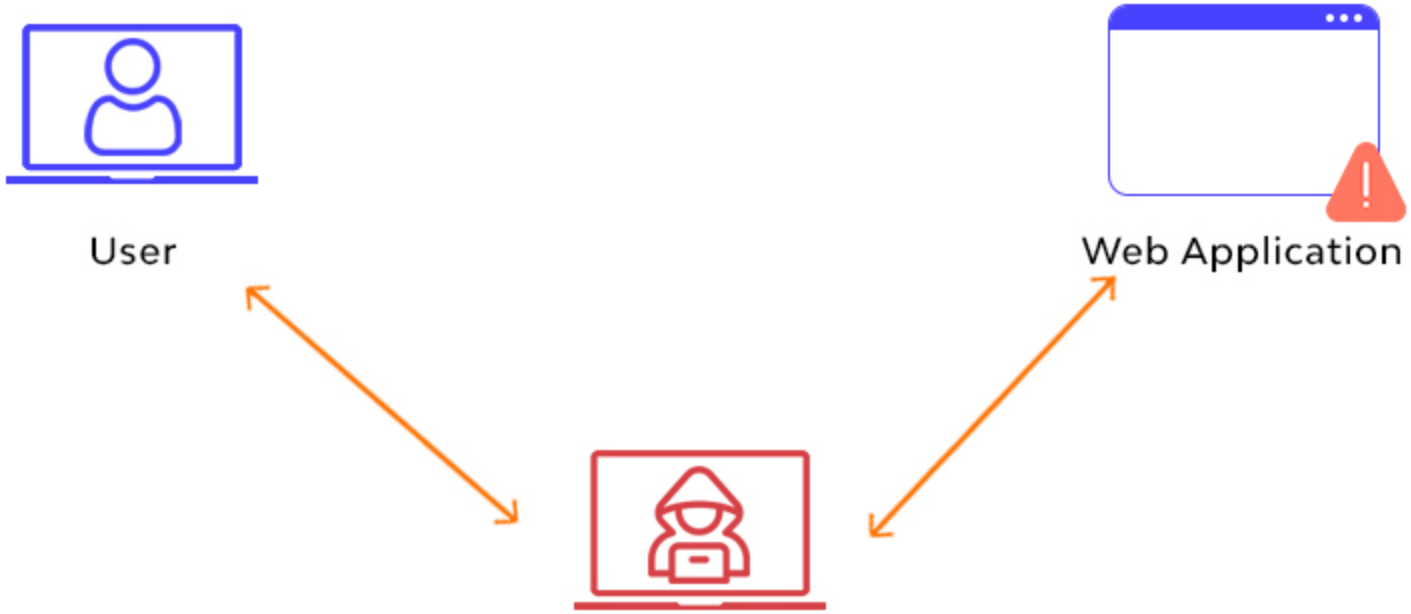
Ambiente:

- https: ligado
- Algoritmo: DES – 50bits
- http: desligado

Cenário de exemplo

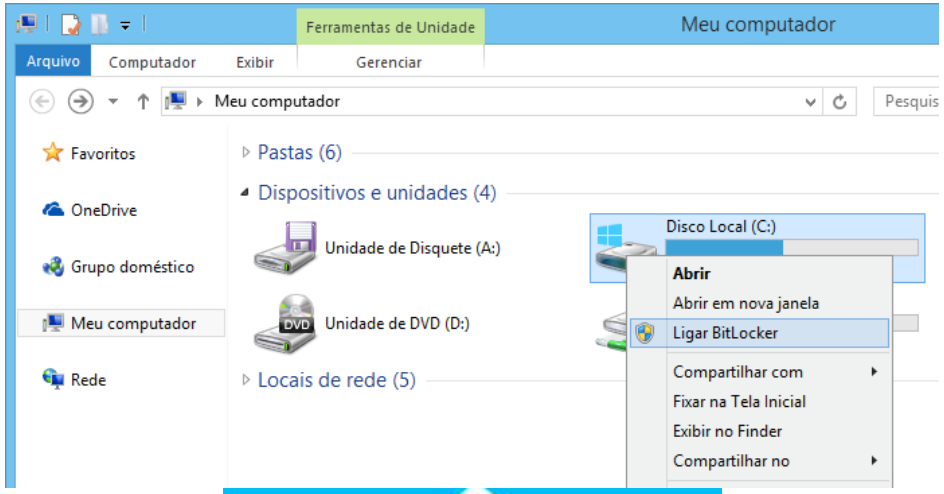
Ambiente:

- https: ligado
- Algoritmo: RSA – 256 bits
- http: ligado
- Servidor: sem cabeçalho HSTS



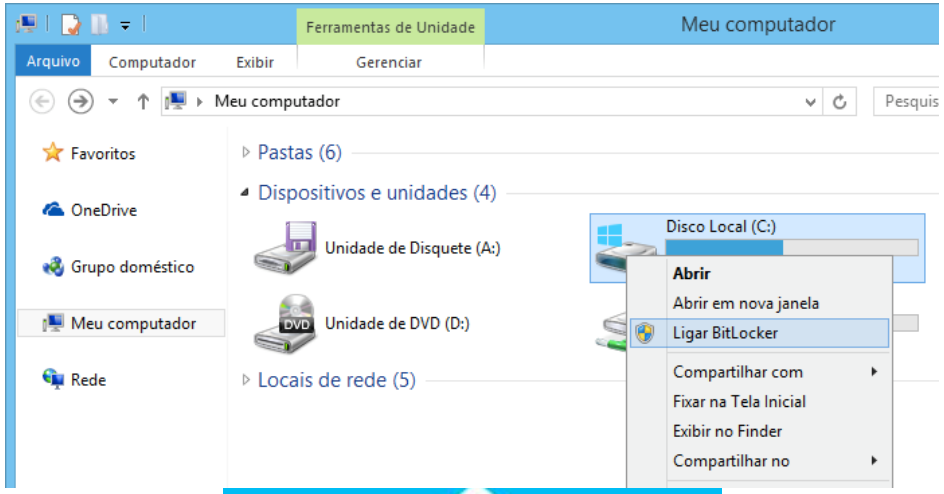
Cenário de exemplo

HD Criptografado e descartado ou interceptação de arquivo criptografado:



Cenário de exemplo

HD Criptografado e descartado ou interceptação de arquivo criptografado:



Cenário de exemplo

Armazenamento de senha:

	(No column name)	
1	0x48699133C5F20DB0CAB52CC203CFFED1	MD5

	(No column name)	
1	0x18CAA0AA39B10AE2AB9CFF8DAEC38619F64CF38D	SHA1

	(No column name)	
1	0xC697E478FC964732DF55417D09DFAFAB7FDAA61EABCB5B8F98B9AA142E684C5A	SHA256

	(No column name)	
1	0x260E1E4E12CBB6994E4E4E335A4BDF10AE446E30C6555DC9D4BF8636C3E3EC993AD4D412D77D22EBF4EA0AFB21F90BA56A6094A64B17D66BC3BF35BA7A69F4C6	SHA512

Aula 4

Categorias: A03, A04 e A05

PÓSPUCPRDIGITAL

PUCPR

GRUPO MARISTA

Agenda – Aula 4

Categorias:

- A03 Injection
- A04 Insecure Design
- A05 Security Misconfiguration

A03 Injection

- Basicamente, são vulnerabilidades que ocorrem quando as entradas passiveis de manipulação e não são validadas.
- Estas entradas ficam na superfície de ataque da aplicação, como um formulário web ou uma URL com passagem de parâmetro.

- Relembre a regra (aula 1 – relembrar é viver):

“Todo acesso ou envio de dados é mal intencionado, até que se prove o contrário”

Vulnerabilidade de segurança

Um formulário de busca, cadastro, login ou passagem de parâmetro na URL.

Todo ambiente que permitir envio de parâmetros (na URL ou através de formulário) será interpretado pelo sistema desenvolvidos ou não acionar outro sistema (ex.: SQL, LDAP, Comandos de OS, etc).

Exploração: Fácil

Controles de segurança

1. Funções de validação (uso de mesmas bibliotecas)
2. Tratamento de entrada de dados no frontend e backend
3. Limite as resposta (ex.: SQL – limit 1)
4. WAF (Web Application Firewall)
5. SDLC – secure development lifecycle
    1. Faça modelagem de ameaça
    2. Automatize processos de avaliação de aplicativo (SAST – static application Security Test, DAST – Dynamic Application Security Test e IAST - Interactive Application Security Test)
    3. Realize testes manuais em toda superfície exposta (red team – pentester)

Mitigação: médio

Impacto técnico

O sucesso na exploração dessa vulnerabilidade poderá resultar:

1. Perca ou vazamento de dados
2. Falhas ou negação de serviço
3. Engenharia social
4. Controle total do ambiente.

Impacto: Alto

Impacto ao negócio

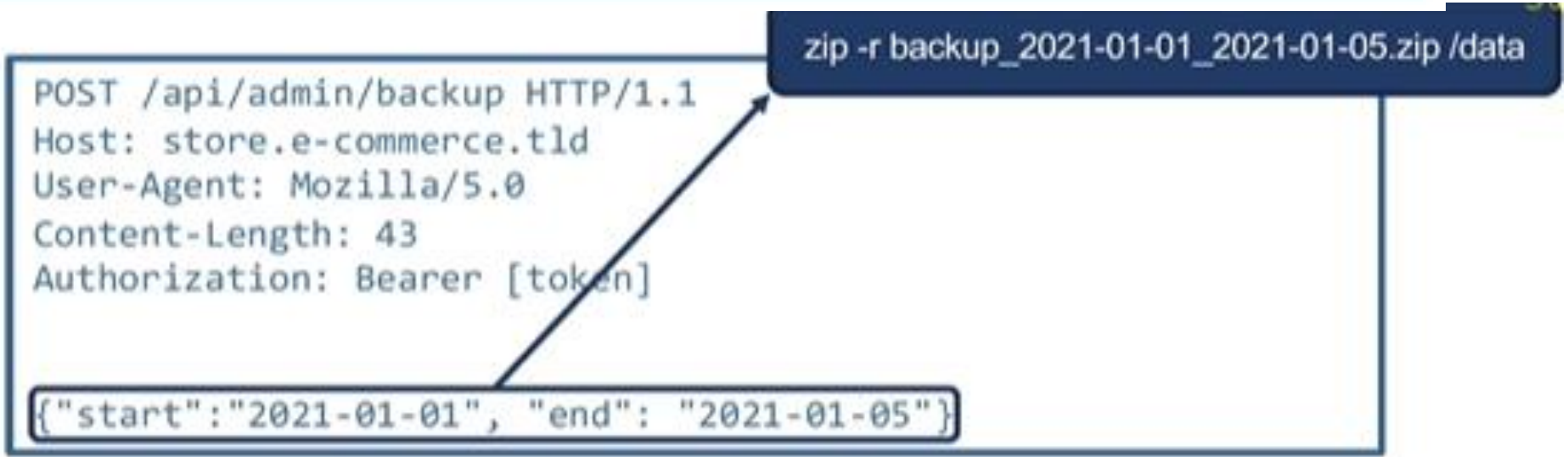
1. Indisponibilidade
2. Chantagem
3. Risco ao cliente
4. Risco à imagem
5. Multas

Impacto: Alto

Cenário de exemplo

1) Sistema para backup:

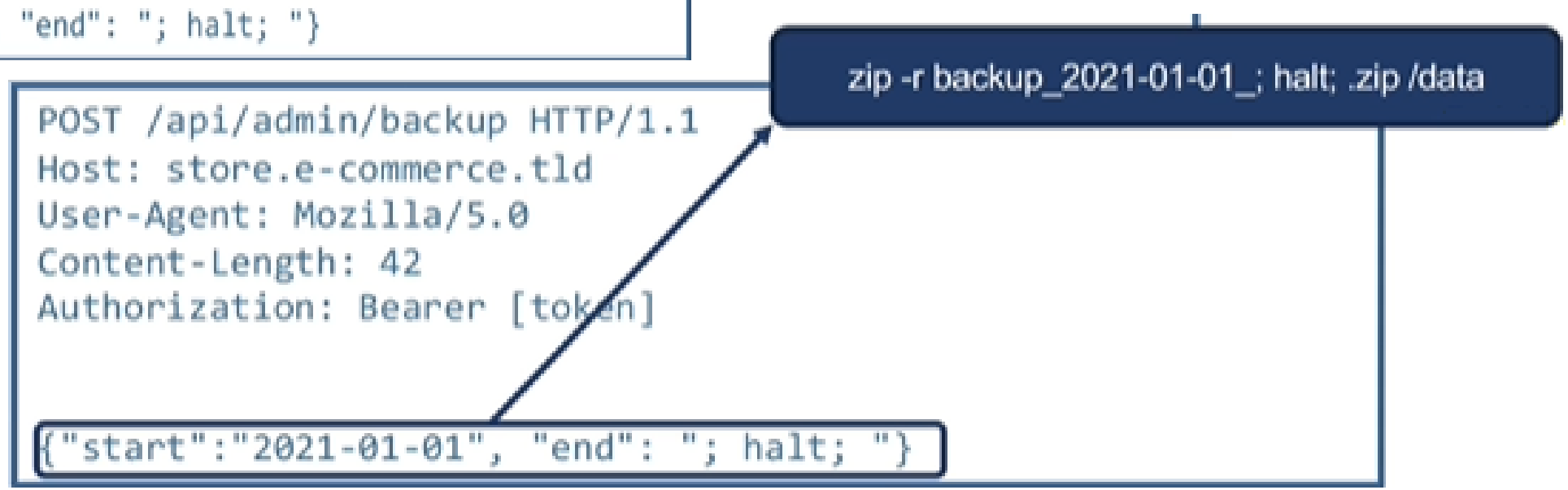
```
POST /api/admin/backup HTTP/1.1
Host: store.e-commerce.tld
User-Agent: Mozilla/5.0
Content-Length: 43
Authorization: Bearer [token]


{"start":"2021-01-01", "end": "2021-01-05"}
```



Cenário de exemplo

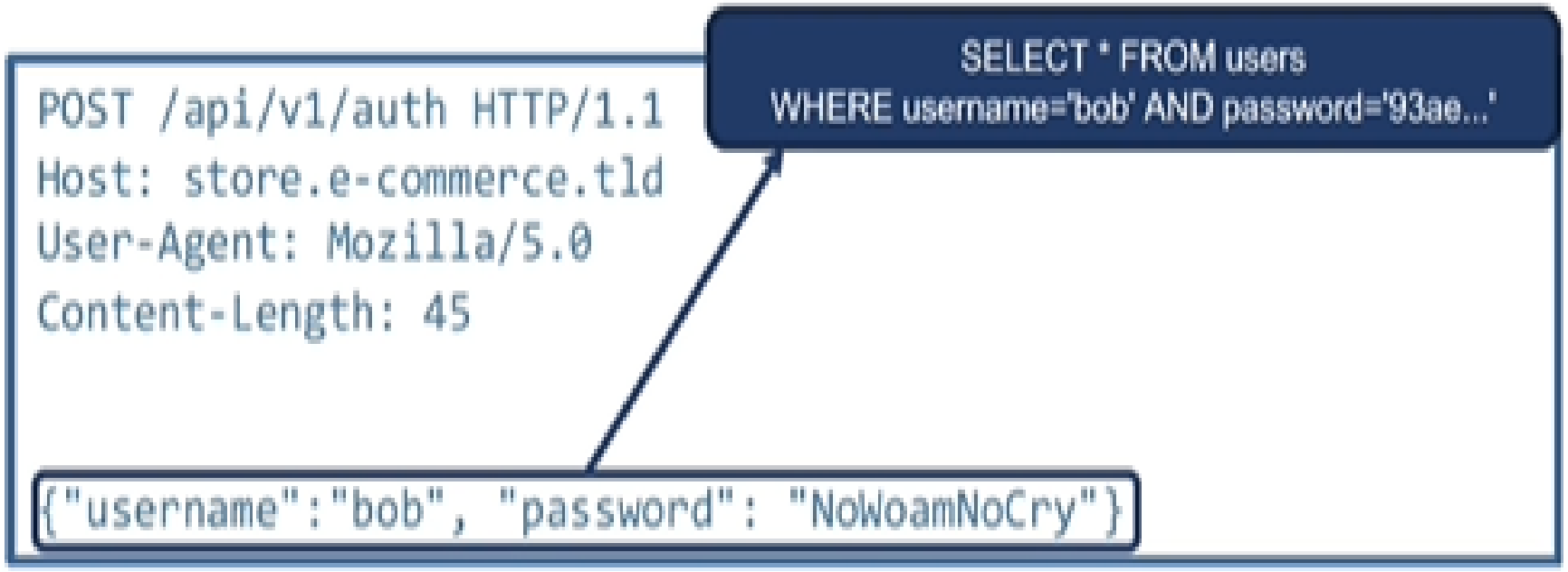
1) Sistema para backup:

```
POST /api/admin/backup HTTP/1.1
Host: store.e-commerce.tld
User-Agent: Mozilla/5.0
Content-Length: 42
Authorization: Bearer [token]


{"start":"2021-01-01", "end": "; halt; "}
```



Cenário de exemplo

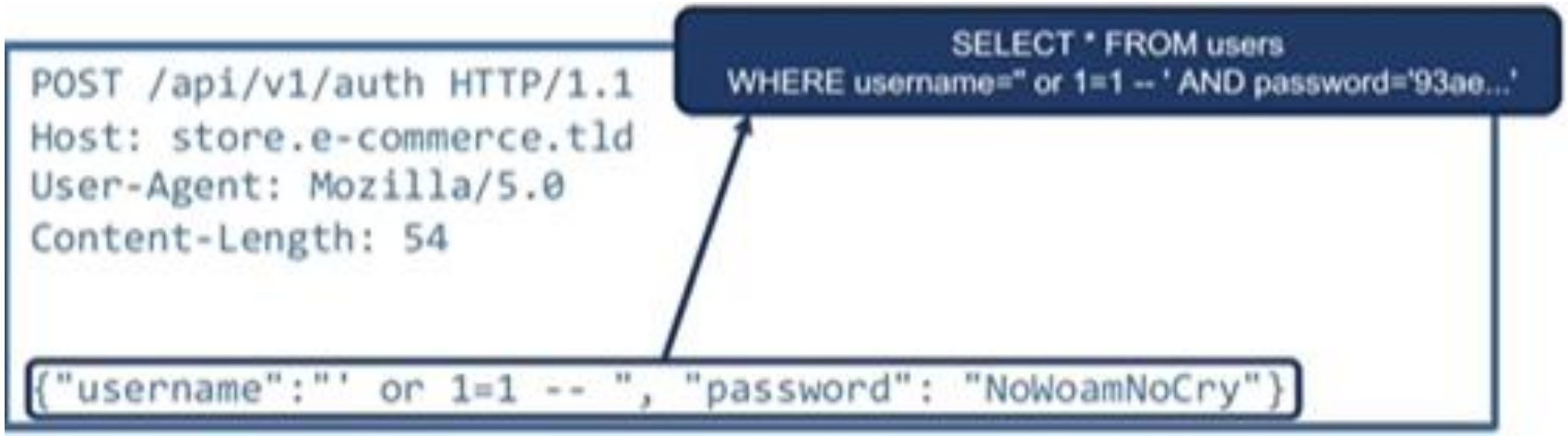
2) Acesso aos dados em banco de dados:



Cenário de exemplo

2) Acesso aos dados em banco de dados:

```
POST /api/v1/auth HTTP/1.1
Host: store.e-commerce.tld
User-Agent: Mozilla/5.0
Content-Length: 54


{"username":"' or 1=1 -- ", "password": "NoWoamNoCry"}
```



A04 Insecure Design

- Ocorre quando a aplicação foi especificada sem considerar as necessidade de segurança da informação (com falta ou controles insuficientes).
- Quando isso ocorre, mesmo uma implementação perfeita (sem erros ou faltas) não serão capazes de corrigir as falhas conceituais.

Vulnerabilidade de segurança

- Pode ocorrer quando a concepção não abrange a cultura/conscientização de segurança. No entanto, podem existir outras causas raízes:
  - tempo de desenvolvimento,
  - reaproveitamento de código ou
  - MVP que vira produção

Resultado:

- Especificação incorreta ou sem contemplar todos os critérios (exemplo: logs)
- Falta de perfis de negócio para especificar os requisitos
- Falta de modelagem de ameaça

Exploração: Médio

Controles de segurança

1. Capacite pessoas e otimize processos
2. Implemente boas práticas, componentes, bibliotecas e frameworks
3. Determine limites de risco
   - MVP
4. Determine controles de segurança necessários
   - ex.: controle de acesso
5. Especifique o fluxo de dados
6. Especifique o estado, mensagens, logs para cada estado
   - fluxo feliz ou infeliz
7. Crie modelagem de ameaça
   - autenticação crítica, controle de acesso, lógica de negócios e fluxos de chave

Mitigação: Fácil

Controles de segurança

8. Errou, corrija, aprenda com os erros e siga em frente
9. Motive os acertos e aprendizados
10. Segregue o ambiente
    - cloud, tenant, rede, serviços, usuários, certificados, etc
11. Especifique limites
    - consumo de usuário, consumo de dispositivos, consumo de API
12. Estabeleça fluxo de testes constantes
    - red team / pentester / appsec
13. Realize medições da sua maturidade no SDLC:

- (OWASP Software Assurance Maturity Model (SAMM))

Mitigação: Fácil

Impacto técnico

1. A correção pode ser inviabilizada pelo custo ou pela tecnologia utilizada
2. A rastreabilidade necessária não será identificada
    - ex.:
        - o que o usuário com acesso consultou?
        - A periodicidade de consulta está correto?

Impacto: Alto

Impacto ao negócio

1. Falta de rastreabilidade
2. Falta de responsabilização
3. Indisponibilidade (retirar a solução do ar)
4. Falta de evidência (*forense*)
5. Multas
6. Operações canceladas (prejuízos)
7. Risco a imagem

Impacto: Alto

Cenário de exemplo

1. Contratação com falta de logs ou rastreabilidade ineficiente

2. Processo de recuperação de senha utilizando “perguntas e respostas” proibida na NIST 800-63b

Cenário de exemplo

3. Uma rede de cinemas que permite reservar até 15 lugares antes de pedir depósito financeiro e recebe um ataque (de poucas requisições) reservando mais lugares em toda a rede

4. Um site de vendas que não está preparado para DDOS e não consegue atender clientes

Cenário de exemplo

5. Um sistema de investimentos que não valida o percentual a ser monetizado ao cliente quando o dinheiro é investido

6. Um processo de cadastro que não valida corretamente o cliente

A05 Security Misconfiguration

- Ocorre quando as configurações, parametrizações ou processos necessárias para garantir a integridade do sistema ou ambiente não são realizadas ou são insuficientes.

Vulnerabilidade de segurança

1. Falha de rede (segregação, ACL ou firewall / logs de controle)
2. Falha de configuração de servidores
3. Falha de configuração de aplicativos
4. Reaproveitamento de usuários
5. Serviços ou funções desnecessárias ativas
6. Mensagens de erro com muitas informações ao usuário
7. Falta de atualização (sistema operacional ou componentes da aplicação)
8. Aplicações instaladas de forma desprotegidas (ex.: admin)
9. Excesso de permissão (ex: usuário perfil admin no desktop)
10. O servidor não envia cabeçalhos ou diretivas de segurança, ou eles não estão configurados para proteger os valores.

Exploração: Fácil

Controles de segurança

PROCESSOS:

1. Padronização de servidores seguros com base na função (hardening)
2. Permissões mínimas para executar a função
3. Utilização de usuários específicos para cada atividade
4. Atualização de servidores e componentes
5. Desabilitar funções não necessárias
6. Revisar configurações de armazenamento

Mitigação: Fácil

Controles de segurança

PROCESSOS:

6. Revisar configurações de armazenamento
7. Revisar permissões atribuídas
8. Revisar permissões herdadas
9. Processo de segregação de função, recursos e acessos eficiente
10. Envio de exigências de segurança no cabeçalho de comunicação
11. Processo de validação (SCAN) eficaz
12. Auditoria contínua com apontamentos de melhorias

Mitigação: Fácil

Impacto técnico

1. Ineficiência na atividade (ex.: erros)
2. Acesso indevido: administração por pessoa não autorizada
3. Identificação (enumeração) de recursos que pode resultar em outras vulnerabilidades
4. Uso de *exploits* conhecidos no ambiente
5. Indisponibilidade de recursos
6. Vazamento de dados
7. Ataques internos: Facilidade para espalhar malware na rede
8. Ataques internos: Acesso a recursos não necessários (servidores)

Impacto: Alto

Impacto ao negócio

1. Multas
2. Risco a imagem
3. Indisponibilidade
4. Lentidão
5. Experiência do usuário

Impacto: Alto

Cenário de exemplo

1. Solução de hospedagem da página web com dados de login do administrador padrão de instalação: permite um atacante acessar a administração web.

2. Possibilidade de listar diretórios do servidor e encontrar arquivos de configuração (ex.: arquivo com dados de conexão, arquivos compilados para engenharia reversa)

3. Erros de acesso que acusam versão da solução de hospedagem, a qual permitirá pesquisar vulnerabilidades conhecidas

Cenário de exemplo

4. Armazenamento na nuvem com permissão pública

5. Estrutura de instalação sem segregação

6. Contas administrativas padronizadas, onde um acesso pode resultar no acesso massivo dos dispositivos

7. Aplicações utilizando o mesmo usuário causando riscos:
   - indisponibilidade (ex.: usuário bloqueado),
   - senhas conhecidas por vários times e
   - rastreabilidade não suficiente.

Aula 5

Categorias: A06, A07 e A08

PÓSPUCPRDIGITAL | PUCPR GRUPO MARISTA

Agenda – Aula 5

Categorias:

- A06 Vulnerable and Outdated Components
- A07 Identification and Authentication Failures
- A08 Software and Data Integrity Failures

A06 Vulnerable and Outdated Components

- Ocorre quando você não conhece a versão de todos os componentes que você usa no servidor ou cliente.
- Inclui OS, Web/Application server, gerenciamento de banco de dados, aplicativos, APIs e todos os componentes executados em tempo de execução e bibliotecas

Vulnerabilidade de segurança

- Componente é vulnerável, não tem suporte ou não tem atualização.
- Não há verificação regular e envio de boletins de segurança relativos aos componentes utilizados
- Você não corrige ou não atualiza plataformas ligadas, frameworks e dependência com base em risco e em tempo hábil.
  - **Comum quando a tarefa de validação é mensal ou trimestral.**
- Desenvolvedores não testam compatibilidades da atualização com as atualizações ou bibliotecas
- Não há configuração de segurança no componente (A05)

Exploração: Fácil

Controles de segurança

Gerenciamento de patches para:

1. Remova dependências não utilizadas, recursos, componentes, arquivos e documentação desnecessários.
2. Inventarie regularmente as versões dos componentes do lado do cliente e do lado do servidor (por exemplo: frameworks e bibliotecas)
3. Inventarie dependências usando ferramentas de versões, OWASP Dependency Check, retire.js, etc.
4. Use ferramentas de análise de “composição” de software para automatizar o processo.
5. Monitore continuamente fontes como vulnerabilidade comum e exposições (CVE) e National Vulnerability Database (NVD)
6. Inscreva-se para receber alertas de e-mail sobre vulnerabilidades de segurança relacionadas aos componentes que você usa.

Mitigação: Médio

Controles de segurança

7. Obtenha componentes apenas de fontes oficiais por meio de links seguros.
   - Prefira pacotes assinados para reduzir a chance de incluir um componente malicioso modificado (A08).
8. Monitore bibliotecas e componentes sem manutenção ou que não criem patches de segurança para versões anteriores.
   - Se o patch não for possível, considere implantar um virtual patch para monitorar, detectar ou proteger contra o problema descoberto.
9. Garanta o plano monitoramento contínuo, triagem e aplicação de atualizações ou alterações de configuração durante a vida útil do aplicativo ou portfólio.
   - Gestão de vulnerabilidades
   - GMUD – Gestão de mudança

Mitigação: Médio

Impacto técnico

Quebra do aplicativo – indisponibilidade

Correções emergenciais (paralização das atividades, retrabalho, ineficiência)

Vulnerabilidades diversas (de informativas à críticas)

- Execução de código remota (crítica)
- DOS – queda da aplicação
- Identificação de ativos
- Acessos não autorizados

Impacto: Alto

Impacto ao negócio

Queda na lucratividade (indisponibilidade)

Vazamento de dados

Exposição de dados

Multas

Imagem da empresa

Impacto: Alto

Cenário de exemplo

1 – Banco de dados OpenSource com componente desatualizado.

CVE-2021-35647: Vulnerabilidade no produto MySQL Server da Oracle (componente: Server Optimizer) da versões 8.0.26 e anteriores permitem:

- invasor obtenha altos privilégios de acesso à rede e resultar no travamento.

2 – Componentes são executados com mesmo privilégio dos aplicativos.

- As falhas (intencionais ou não) que permita a execução de código remoto ou outra pode resultar em violação.

3 – Falta de gerenciamento em mudanças

- Alterações sem testes ou feitas de forma aleatórias podem comprometer o funcionamento do sistema

A07 Identification and Authentication Failures

- Gestão de identidades, também conhecida como gestão de identidades e acessos ou pelo seu termo em inglês, identity and access management está entre as disciplinas de segurança da informação que "habilita os indivíduos corretos à acessar os recursos corretos no momento correto e pelos motivos corretos“

- Ocorre quando um usuário não autorizado consegue realizar uma autenticação por fora de um processo legítimo ou quando consegue identificar usuários do sistema – as contas válidas – para posterior atacá-las.

Vulnerabilidade de segurança

1. técnicas força bruta,
2. password stuffing
3. password spray
4. SQL Injection
5. Engenharia social
6. Falhas de programação que expõem o usuário (ex.: mensagens ao usuário)

Expostos estão: tela de login, recuperação de senha e telas de cadastro.

Exploração: Fácil

Controles de segurança

Mais comuns:

1. WAF (Web Application Firewall): Rate limit, Pinagem, mTLS, validação do source, badlist
2. Tratamentos: contra injeções (aula 3), política de senhas, controla o originador (ex.: para API)
3. Bloqueio por tempo (exemplo: login Iphone)
4. Armazenamento de senhas: criptografadas ou HASH forte + salt

Quando possível:

1. Restrições de IP
2. Geolocation
3. Controle do dispositivo (device Fingerprint)

Indicadores

1. Comportamento anormal (quantidade de deny, horário, origem desconhecida, etc)

Mitigadores: Fácil

Controles de segurança

Mais comuns:

1. Identificar o fluxo de dados
2. WAF (Web Application Firewall): Rate limit, Pinagem, mTLS, validação do source, badlist
3. Tratamentos contra injeções (aula 3)
4. Política de senhas
5. Use MFA
6. Controla o originador (ex.: para API)
7. Bloqueio por tempo (exemplo: login Iphone)
8. Armazenamento de senhas: criptografadas ou HASH forte + salt
9. Validação das mensagens ao usuário
10. Proteção de tokens (ex.: um click está logado)
11. Utilizar frameworks de mercado – não invente uma nova forma de fazer
12. Administre tokens os JWT (cuidados: dados sensíveis armazenados algoritmo de validação de integridade precisa ser do mesmo tamanho da chave)

Mitigadores: Fácil

Impacto técnico

Acesso não autorizado... Quebra de confidencialidade!

1. Acesso a dados não autorizados (pessoas, sensíveis, movimentações financeiras)
2. Operação do sistema de forma não autorizada (cadastros, movimentações financeiras, exclusões, etc)

3. SUPERFÍCIE DE ATAQUE AUMENTA QUANDO O ACESSO É CONQUISTADO!!!
    1. O acesso pode ser legítimo – cadastro falso
    2. O acesso pode ser ilegítimo – usuário comprometido

Impacto: Alto

Impacto ao negócio

1. Interrupção de sistema
2. Chantagem
3. Risco à imagem
4. Risco ao cliente
5. Multa
6. Prejuízos diversos

Impacto: Alto

Cenário de exemplo

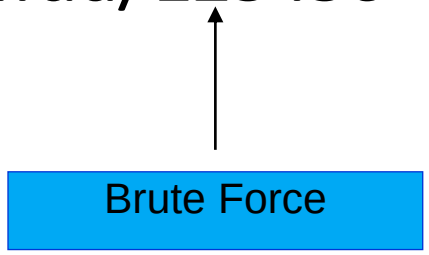
1) Processo de recuperação de senha com token que demora para expirar informado na URL.

```
POST        /api/sistema/recuperar_senha    HTTP/1.1
Host:       site1.com
Accept-Language: pt
User-Agent: Mozilla/5.0
Content-Length: 40

{
"username": "jose@email1.com"
}
```

Cenário de exemplo

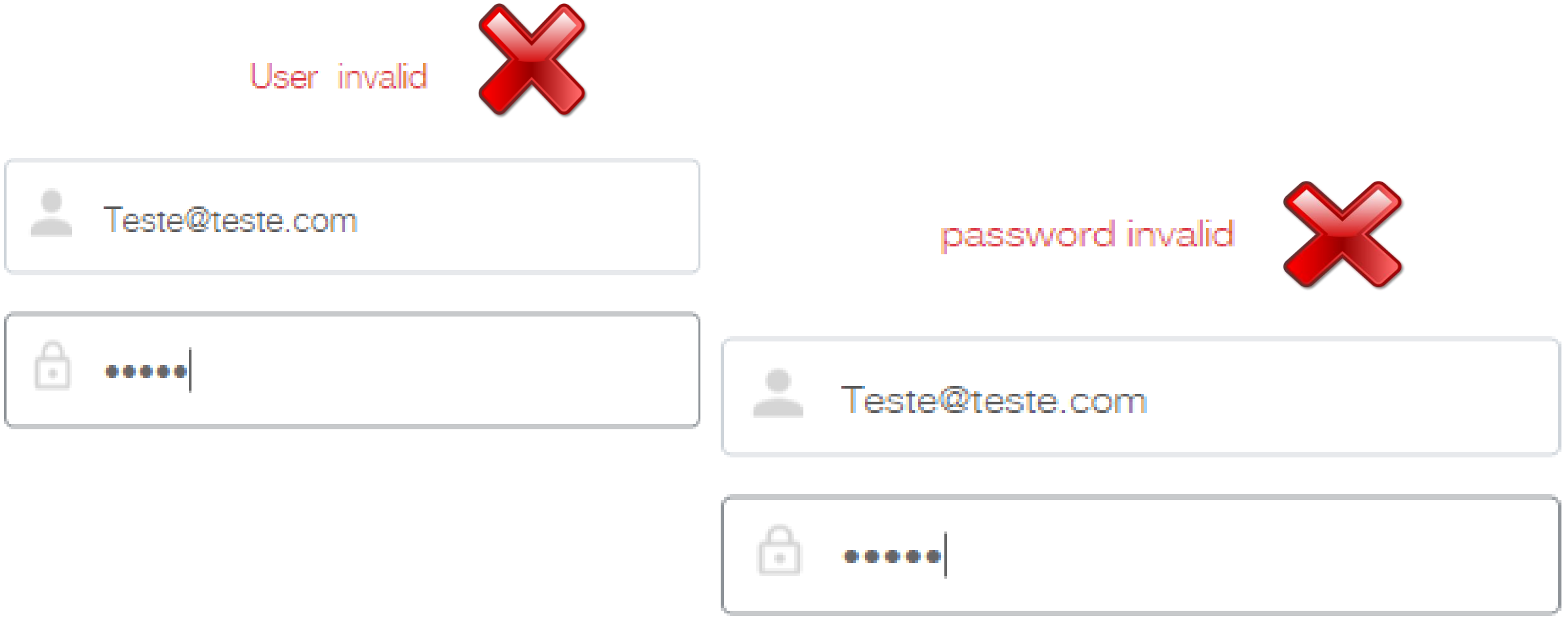
1) Processo de recuperação de senha com token que demora para expirar informado na URL.

```
POST        /api/sistema/valida/123456   HTTP/1.1
Host:       site1.com
Accept-Language: pt
User-Agent: Mozilla/5.0
Content-Length: 40

{
"username": "jose@email1.com"
}
```



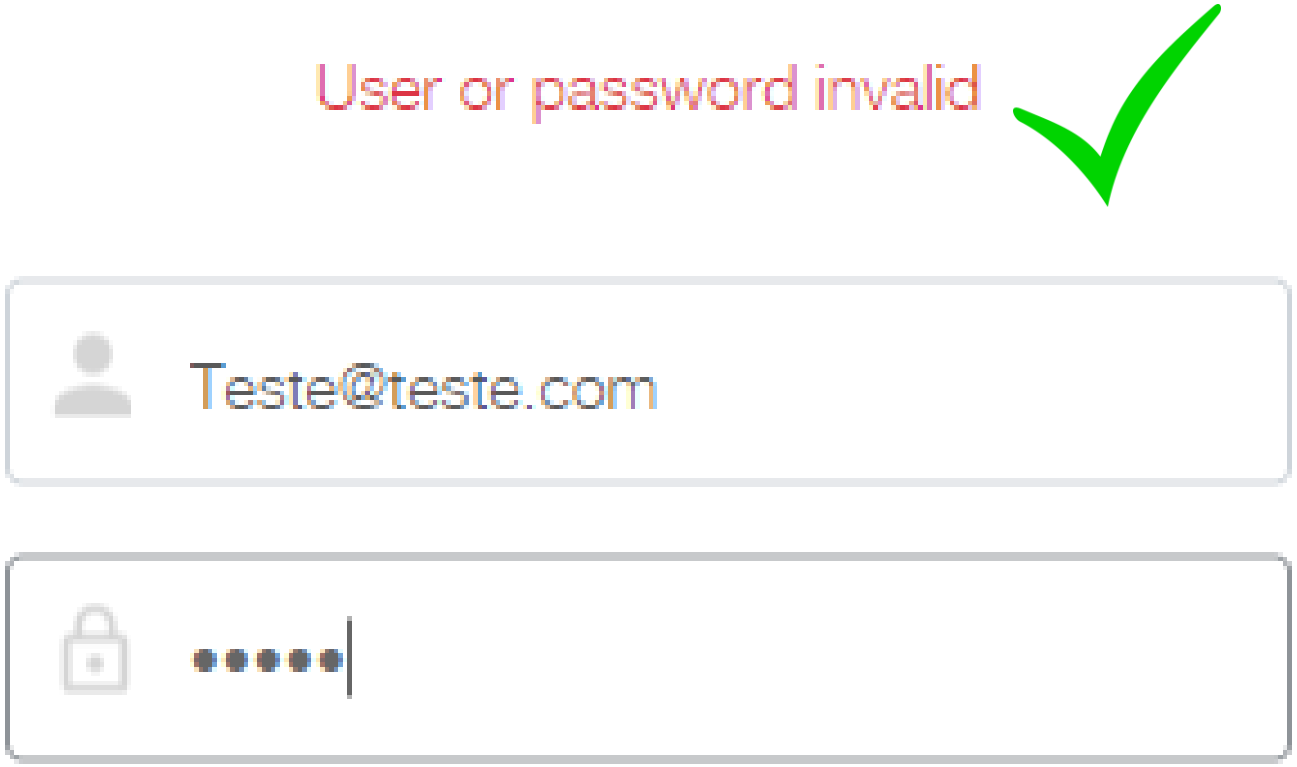
Cenário de exemplo

2) Excesso de detalhe na mensagem:



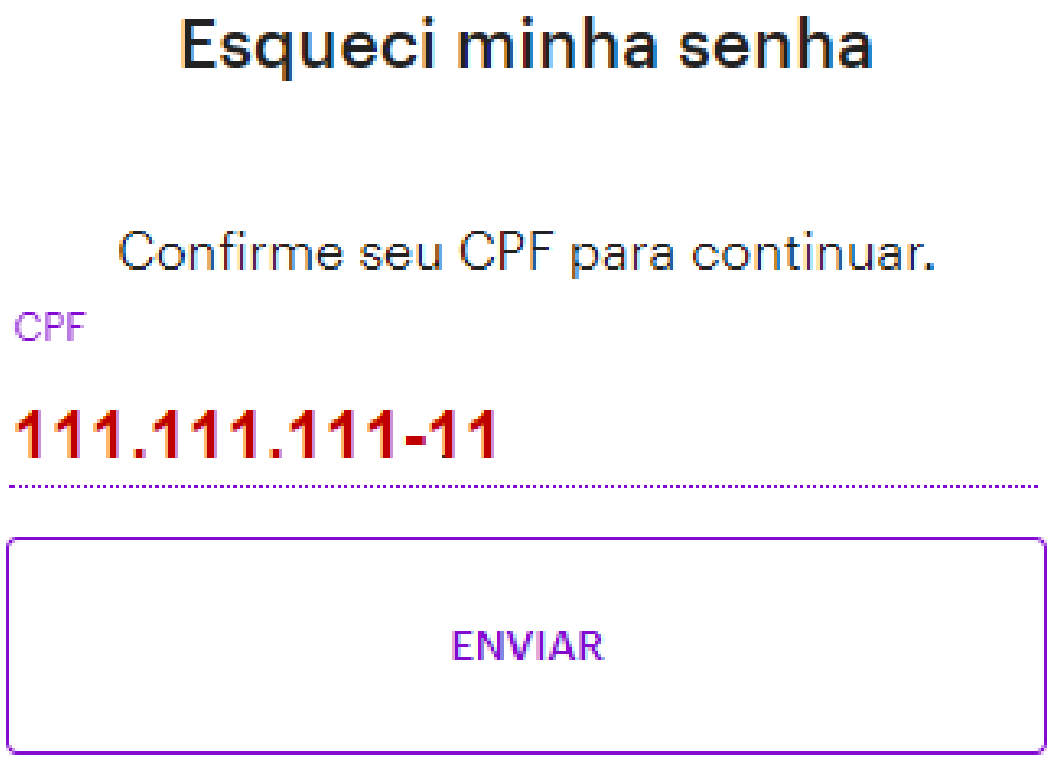
Cenário de exemplo

2) Excesso de detalhe na mensagem:



Cenário de exemplo

2) Excesso de detalhe na mensagem:



Cenário de exemplo

2) Excesso de detalhe na mensagem:

Cenário de exemplo

2) Excesso de detalhe na mensagem:

Se você for cadastro no sistema, enviaremos uma mensagem para o seu e-mail.

Verifique sua caixa.

A08 Software and Data Integrity Failures

- Categoria relacionada a atualização de softwares, dados críticos e pipelines CI/CD sem verificação de integridade.

- Ocorre quando o código ou infraestrutura não é protegido contra violação de software ou os dados.

Vulnerabilidade de segurança

1. Aplicativo depende de plug-ins, bibliotecas ou módulos de fontes não confiáveis, repositórios e redes de entrega de conteúdo (CDNs).

2. Um pipeline de CI / CD inseguro pode apresentar o potencial de acesso não autorizado, código malicioso ou comprometimento do sistema.

3. Muitos aplicativos incluem a funcionalidade de atualização automática e que sem verificação de integridade. Os invasores podem carregar suas próprias atualizações e distribuir nas instalações.

4. Objetos ou dados são codificados ou serializados em uma estrutura que um invasor pode ver e modificar (vulnerável à desserialização insegura)

Exploração: Fácil

Controles de segurança

1. Use assinaturas digitais (ou mecanismos semelhante) para garantir a origem do software ou dados
2. Valide se o npm ou marvem estão consumindo repositórios confiáveis
3. Avalie a possibilidade de manter um repositório interno
4. Use softwares de validações (OWASP Dependency Check ou OWASP CycloneDX) para avaliar vulnerabilidades
5. Valide as permissões da sua pipeline, insira processos de revisão e aprovação de código para minimizar alterações não autorizadas (ou erradas)
6. Valide que as permissões da esteira de CI/CD estão segregadas e possuem controle de acesso adequado para garantir a integridade do código
7. Não envie dados ao cliente “não seguro” com serialização não assinada ou sem criptografia. Você precisa de uma forma de validar adulteração.

Mitigação: Média

Impacto técnico

1. Falta de governança de código
2. Perda de código
3. Falta de conhecimento em riscos
4. Manipulação do código de forma não autorizada
5. Vazamento de dados
6. Aplicativo comprometido

Impacto: Alto

Impacto ao negócio

Perdas financeiras

Risco a imagem

Multas

Impacto: Alto

Cenário de exemplo

1- Atualização sem assinatura:

Muitos roteadores domésticos, decodificadores, firmware de dispositivo e outros não verificam as atualizações por meio de firmware assinado. Firmware sem assinatura é um alvo crescente para invasores e espera-se que só piore. Essa é uma grande preocupação, pois muitas vezes não há mecanismo para remediar a não ser corrigir em uma versão futura e esperar que as versões anteriores se esgotem.

2- Atualização mal-intencionada do SolarWinds:

Estados-nações são conhecidos por atacar os mecanismos de atualização. Caso SolarWinds é o mais abrangente no momento.

Cenário de exemplo

3- Desserialização insegura:

Um aplicativo React chama um conjunto de microsserviço Spring Boot.

A “garantia” de código imutável foi a serialização do estado do usuário a cada solicitação.

O atacante usa a ferramenta Java Serial Killer no objeto Java "rO0" (em base64) e obtém a execução remota de código no servidor de aplicativos.

Aula 6

Categorias: A9 e A10

PÓSPUCPRDIGITAL

PUCPR

GRUPO MARISTA

Agenda – Aula 6

Categorias:

- A09 Security Logging and Monitoring Failures
- A10 Server-Side Request Forgery (SSRF)

Considerações finais

Plano de resposta ao incidente de segurança da informação

- Detecção
- Resposta

A09 Security Logging and Monitoring Failures

- Categoria relacionada a ajudar a detectar, escalar e responder as brechas ativas.
- A ausência de logs e monitoramento, permitem que brechas não sejam detectadas.
- Logs insuficientes, detecções ou monitoramentos falhos não permitiram uma resposta rápida.

Vulnerabilidade de segurança

1. Falhas em eventos auditáveis
   a. Login: excesso de falhas de login podem representar um ataque
   b. Valores de transações: transferências fora do padrão, podem representar um ataque
2. Logs inadequados ou Ausência de logs de erros
   a. Sistemas operacionais: logs não claros ou sem registro, podem: falhar ao identificar falha crítica ou “escalação de privilégio”
   b. Aplicativos: Logs inadequados ou a ausência podem resultar na não rastreabilidade da operação.
3. Logs de aplicativos ou API sem monitoramento de atividade suspeita
   a. Consumo excessivo: uso de BOT não autorizada, tentativa de ataques para quebra de token, ataques de DOS, exploração de vulnerabilidades

Exploração: Fácil

Vulnerabilidade de segurança

4. Logs apenas armazenados
    a. Ausência de inteligência na interpretação – unicamente reativo
5. Alertas de thresholds (limites) e processo de resposta inexistente
    a. Qual o volume de request/min – qual o adequado?
    b. Qual o playbook de resposta se detectado? Captcha, bloqueio, honeypot?
6. Testes de segurança e scans dinâmicos de aplicativo (DAST – ex.: OWASP ZAP) sem alertas ou monitoramento.
    a. Registro sem ação
7. Aplicativo não detecta, escala ou alerta ataques em tempo real ou próximo ao tempo real
    a. Ineficiência do monitoramento
8. **LOGS VAZADOS AO ATACANTE É UMA VULNERABILIDADE (A1)**

Exploração: Fácil

Controles de segurança

Arquitetura da solução prevendo logs, detecção e resposta, como:

1. Logs de login, controle de acesso e validação de entrada de dados (lado do servidor) com contexto do usuário suficientes para identificar contas suspeitas ou mal-intencionadas.
2. Prazo de retenção de logs adequado para investigações forense futuras
3. Padronização de logs para consumo por soluções de SIEM ou antifraude
4. Viabilize o consumo em tempo real ou próximo
5. Verifique se o log está em um formato que inviabilize injeções contra o servidor de monitoramento, siem ou antifraude

Mitigação: Médio

Controles de segurança

6. Garanta a captura de todas as ações para a trilha de auditoria (Quem, Quando, Por onde, Tempo de execução, Aceites)
7. Garanta controles de integridade do log (banco de dados ou arquivo)
8. Times de DevOps (DevSecOps) devem garantir o monitoramento e alertas eficazes para atividades suspeitas e que sejam detectadas e respondidas rapidamente
9. Estabelece um plano de resposta ao incidente – ex: NIST 800-61r2 (ou superior)

Existem diversos softwares (comerciais e open source) para proteger o framework (OWASP ModSecurity Core Rule Set) e de correlação de logs (Elasticsearch, Logstash, Kibana (ELK) stack) que permitem dashboard e alertas

Mitigação: Médio

Impacto técnico

1. Ineficiência na resposta
2. Ambiente “desgovernado”
3. Instabilidade
4. Indisponibilidade
5. Erros sem causa raíz
6. Vazamento de dados
7. Acessos não autorizados

Impacto: Alto

Impacto ao negócio

1. Ambiente desgovernado (o que está ocorrendo?)
2. Cliente impactado (sistema fora do ar)
3. Auditorias prejudicadas
4. Negação da ação executa (ex.: contratações ou transações financeiras)
5. Prejuízos financeiros decorrentes de processos judiciais
6. Impactos com órgãos reguladores
7. Multas
8. Imagem da organização

Impacto: Alto

Cenário de exemplo

1 – Ausência ou ineficiência de monitoramento

Site de plano de saúde não detectou uma invasão devido a falta de interpretação dos logs. Resultou no vazamento de dados de pacientes (crianças e adultos). Em análise forense, foi identificado que vulnerabilidades significativas não foram tratadas e o sistema estaria vulnerável à anos – sem registro das atividades.

2 – Retenção de logs de poucos meses

Uma empresa armazena logs de acesso à Internet de apenas 2 meses e foi notificada judicialmente para apresentar o usuário que realizou acessou ilegítima no ano anterior. Por não apresentar, a empresa responderá pelo acesso.

3 – Ausência de rastreabilidade

Cliente não reconhece a contratação do serviço online, solicita que as cobranças sejam canceladas e pode processar juridicamente o prestador de serviço.

Por falta de comprovação, a prestadora de serviço poderá ficar com o prejuízo.

Cenário de exemplo

4 – Ausência de rastreabilidade

O ambiente web de uma empresa foi acessado de forma indevida. Os logs de acesso à blobs estavam desativados, prejudicando a rastreabilidade da ação.

5 – Alertas ineficientes

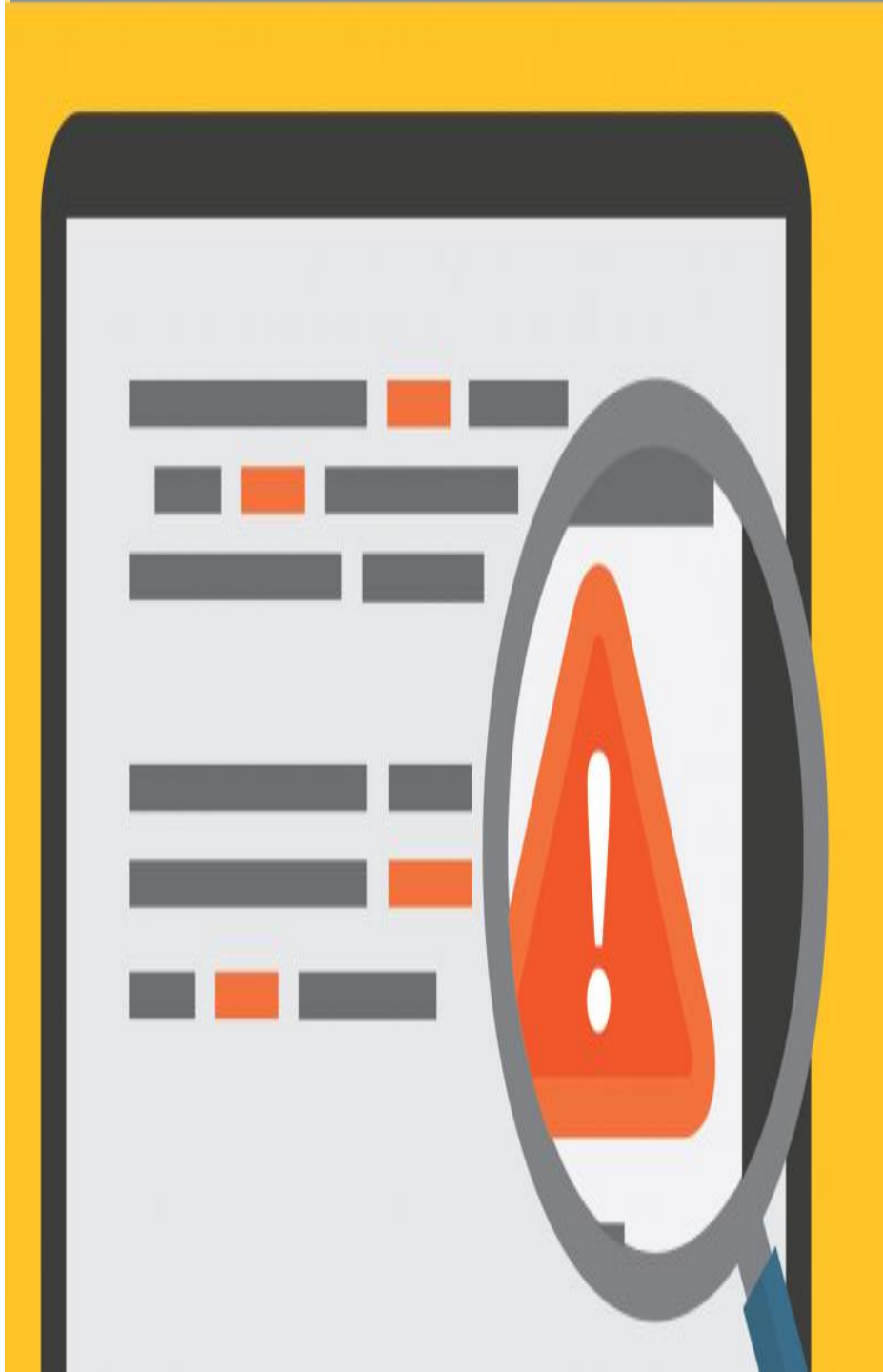
Uma conta corrente possui um cliente residente nos EUA. A conta foi acessada pelo cliente as 2:00PM através de um endereço IP originado nos EUA. As 2:05PM a conta foi acessada com sucesso através de um IP originado na EUROPA. O alerta de comportamento anormal.

6 – Alertas ineficientes

Uma conta corrente foi acessada através de um endereço IP. Em 10 minutos, o mesmo IP acessa mais de 20 contas correntes. O sistema não detectou a anomalia.

A10 Server-Side Request Forgery (SSRF)

- Ocorre quando um aplicativo web aciona uma URL fornecida pelo usuário sem validação
- Firewall não são eficientes contra esta categoria



Vulnerabilidade de segurança

1. Permite ao atacante direcionar chamadas do aplicativo para destinos não validados
2. Os alvos podem ser redes internas ou externas
3. Arquiteturas complexas e serviços em nuvem podem potencializar o ataque

Exploração: Fácil

Controles de segurança

1. Valide e limpe as entradas do lado do cliente:
    1. Crie lista de endereços IP confiáveis para requisições
    2. Crie lista de domínios ou aplicativos confiáveis
2. Force URL schema, porta e destino com lista de permitidos
3. Não envie resposta pura para o cliente
4. Desabilite HTTP redirect
5. Saiba que DNS atacados podem ser utilizados na URL para ataques SSRF

Mitigação: Médio

Controles de segurança

5. Saiba que DNS atacados podem ser utilizados na URL para ataques SSRF
6. Desabilite URL schemas não utilizados
   a. Ex.: FTP, SFTP, HTTP
7. Evitar ou não fazer:
   a. Que o aplicativo faça requisição para qualquer IP e domínio na internet
   b. Que o usuário possa enviar a URL completa
   c. Não faça lista de negação. Atacantes irão contornar a sua lista
   d. Não implemente sistemas relevante em sistemas que ficam expostos na internet (ex.: OpenID) ou controles locais (ex.:localhost)

Mitigação: Médio

Impacto técnico

1. Exposição do ambiente
2. Necessidade de correção imediata
3. Vazamento de dados
4. Vazamento de credenciais
5. Acesso a arquivos de sistema

Impacto: Médio

Impacto ao negócio

1. Acesso não autorizado
2. Vazamento de dados
3. Multa
4. Prejuízos
5. Risco a imagem

Impacto: Médio

Cenário de exemplo

1 – Identificação de portas internas.

Se a arquitetura de rede não for segmentada, os invasores podem mapear as redes internas e determinar se as portas estão abertas ou fechadas em servidores internos a partir dos resultados da conexão ou do tempo decorrido para conectar ou rejeitar as conexões de carga SSRF.

2 – Vazamento de dados

Invasores podem acessar arquivos locais, como ou serviços internos, como: “file:///etc/passwd” e http://localhost:28017/

3 – Acessar armazenamento de metadados em nuvem

Alguns provedores de nuvem armazenam metadados como http://169.254.169.254

4 – Serviço interno comprometido

O invasor realizar outros ataques em serviços internos como execução remota de código (RCE) ou negação de serviço (DoS)

Plano de resposta ao incidente de segurança da informação

Plano de ação voltado a responder um incidente de segurança cibernética.

Tem o intuito de trazer a governança, acompanhamento e visibilidade a todos sobre todas as fases das tratativas.

Fundamental para garantir o rápido retorno do negócio.

Organizando a resposta ao incidente de segurança

O objetivo:

1. Detectar rápido
2. Minimizar prejuízos e roubos
3. Aprender com o incidente para não ser afetado novamente (aprimorar controles e proteções)
4. Estar em conformidade com regulações
5. Prover melhoria contínua da estratégia de defesa contra ameaças digitais

<u>Política de resposta é particular de cada empresa</u>

O que deve ser explicito e inequívoco:

O que deve ser explicito e inequívoco:

1. Missão e estratégia
2. O que é um incidente para a sua organização
3. Declaração de compromisso de gestão
4. Finalidade e objetivos da política
5. Processos de comunicação:
    a. Demais times
    b. Clientes
    c. Fornecedores
    d. Investidores
    e. Formulário de contatos

O que deve ser explicito e inequívoco:

5. Quais equipes podem ser acionadas
   a. Autonomia para tomar decisões
6. Estrutura organizacional e definição de papéis, responsabilidades e níveis de autoridade
   a. Quem “puxar o cabo”?
   b. Quem documenta as tratativas?
   c. Quem fará a comunicação?
7. Priorização de incidentes por classificação
8. Como será medida a eficácia do plano
9. Processo de melhoria do plano
10. Como o plano ajuda a organização

Procedimentos

O que deve conter no plano: