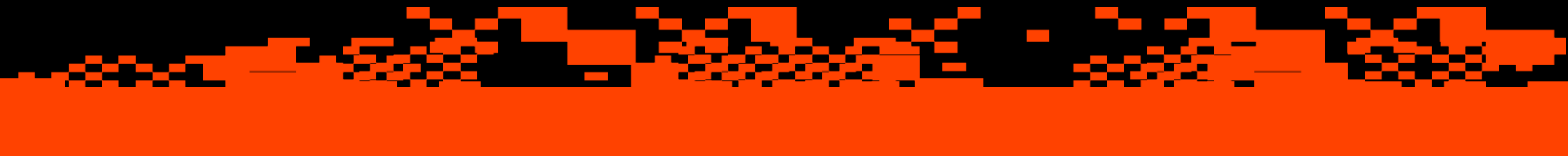


HACKA/FLAG

** Security From Scratch /*



\$ whoami _



Thau0x01 Santos

Professional Bug Maker

@thau0x01 | hackware.tech

HACKA/FLAG

Why Thau0x01?

```
> var suffix = 0x01  
> suffix.toString()  
> "1"
```

Quando pronunciado em inglês fica

[/'wʌn/](#)

HACKA/FLAG

Why Bug Maker?



WHAT'S NEW

- Updated third-party libraries
- Many changes under the hood
- Fixed bugs 🐛
- Added more bugs to fix later

“ Observações >

AVISO: Nada do conteúdo apresentado nesta palestra representa a opinião do meu empregador, de seus clientes ou qualquer pessoa ou empresa ligada direta ou indiretamente a ele.

Eu só usei o template de Slides porque o Marketing olhou o anterior e disse que uma criança de 10 anos faria melhor.

HACKA/FLAG

" Observações - parte 2 >

Gostaria de dizer para todos que estão vendo esta palestra, que se você se ofender com alguma coisa do que eu disser durante a apresentação.

NÃO É A INTENÇÃO.

HACKA/FLAG

“ Observações Finais >

Não tem mais nenhuma observação, foi só pra fazer um começo dramático.

HACKA/FLAG

“ Desenvolvimento de Software e Segurança

>

Resumo: A LGPD tá chegando, o mundo tá girando e os vacilões vão rodando!

HACKA/FLAG

“ Motivações >

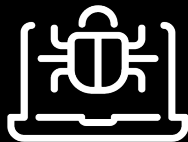
Mercado. No ambiente em que eu me formei, como desenvolvedor de software, uma prática muito comum das empresas era simplesmente fingir que a segurança não existia, a cobrança era focada apenas em operacionalidade do projeto e custo baixo, nunca em qualidade ou segurança, principalmente.

HACKA/FLAG

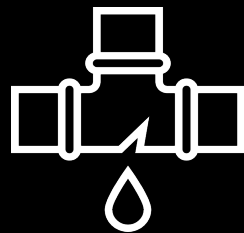
“ Resultado de tais práticas >



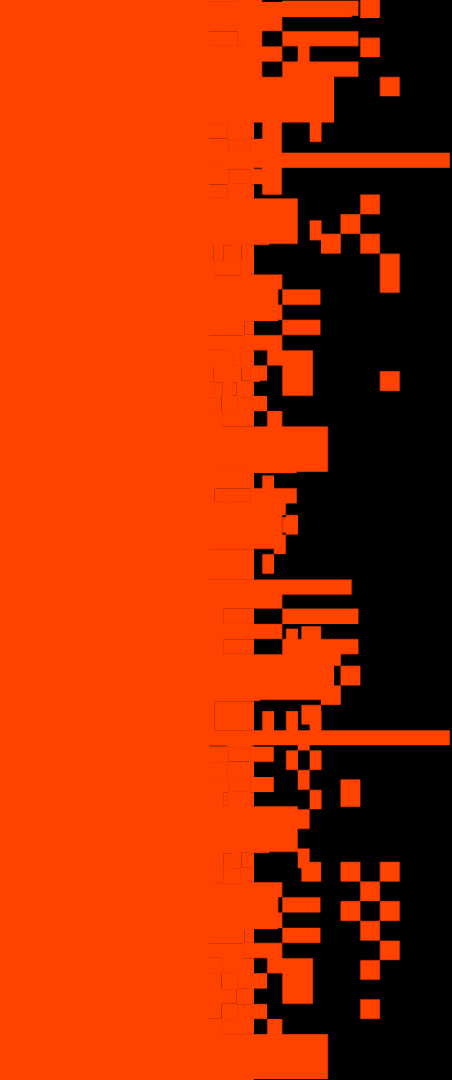
Bugs.
Bugs Everywhere!



Sistemas Ownados.
Nada novo sob o sol!



Dados Vazados.
E clientes ferrados!



Eventualmente...
Alguém era
responsabilizado!

HACKA/FLAG

"Uma coisa é tu vazar dado pessoal de preto, pobre e favelado. Que **ninguém** liga!"

"Outra coisa é tu vazar dado pessoal desse povo da **PUC**, que abraça árvore no parque Ibirapuera e que **tem dinheiro** pra pagar advogado."

Sgt. Peçanha



Pré LGPD - A Responsabilidade é Zero

“ Panorama Atual >



Desenvolvedores mal preparados.

Basicamente um monte de devs que nunca ouviram falar de segurança ou não sabem como aplicá-la corretamente no dia-a-dia!



***Empresas Desesperadas.** Um monte de empresas com medo de serem multadas e correndo atrás do problema que elas mesmas criaram.*



***Desenvolvedores ruins falando de Segurança.** Isso é uma autocrítica, tá?!*

HACKA/FLAG

Um Pequeno Resumo

Sobre os ataques e descobertas recentes



O Ponto de Entrada
era uma vulnerabilidade
conhecida publicamente

Misconfiguration
É também um
grande vilão

Devs não conhecem
a **OWASP**

Em
>50%

Dos Ataques

**“Mas é
Infra”**

E as
10

**Vulnerabilidades
Mais Comuns.**

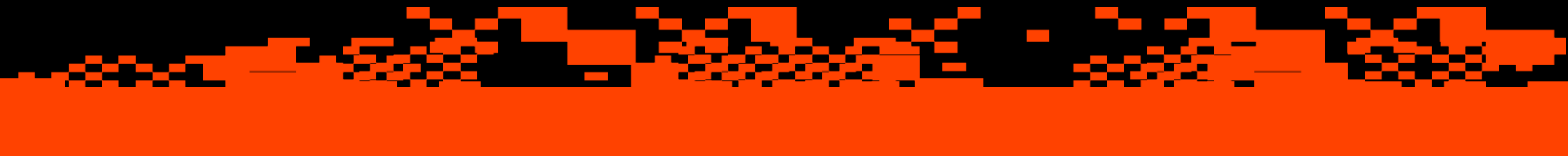
OK. Já podemos começar a montar um plano de ação.

Como mudar este cenário

Para não sofrer com as novas legislações?



Empresas e Gestores



“ Repensar as práticas >



Desenvolver software não é mágica, é necessário aumentar os prazos, capacitar melhor os profissionais e oferecer oportunidades melhores para quem é melhor qualificado, e parar de jogar a segurança, somente para o time de sec.



É possível pensar em formas de otimizar o processo de desenvolvimento, sem perder a segurança ou ter aumentos nos custos!



Conectar todos os times ligados ao produto final é muito importante, é necessário fazer as equipes trabalharem em conjunto.

HACKA/FLAG

Técnicos e Devs



“ Avaliando as ameaças >



Que tipo de desenvolvedor você é?

Quais tecnologias você utiliza no seu dia-dia no trabalho? Qual é a infraestrutura onde essa tecnologia roda?



Quais features eu vou implementar no meu app? Meu usuário vai carregar arquivos, inserir e-mails, vai submeter formulários?



Vou consumir APIs de terceiros?

Eu posso confiar no conteúdo que estas APIs me servem?

HACKA/FLAG

“ Ataques comuns em Aplicações WEB >

1. XSS - Cross Site Scripting
2. SSRF - Server Side Request Request Forgery
3. CSRF - Cross Site Request Forgery
4. Injection *
 - a. SQL, Code, CRLF, SMTP, Command, Xpath, LDAP, HH, etc...
5. Desserialização Insegura
6. Sensitive Information Disclosure
 - a. Source Code
 - b. Filename and Path
7. Broken Authentication

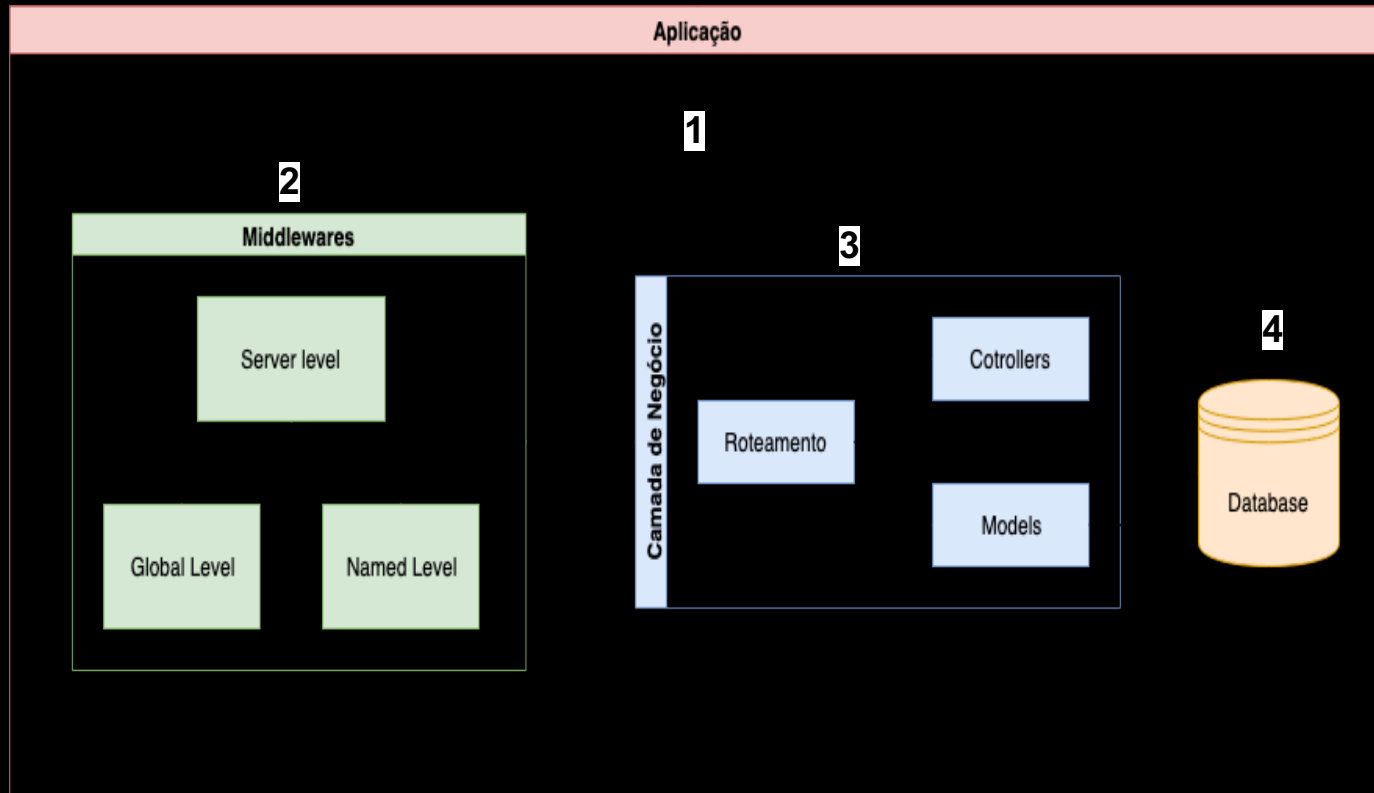
HACKA/FLAG

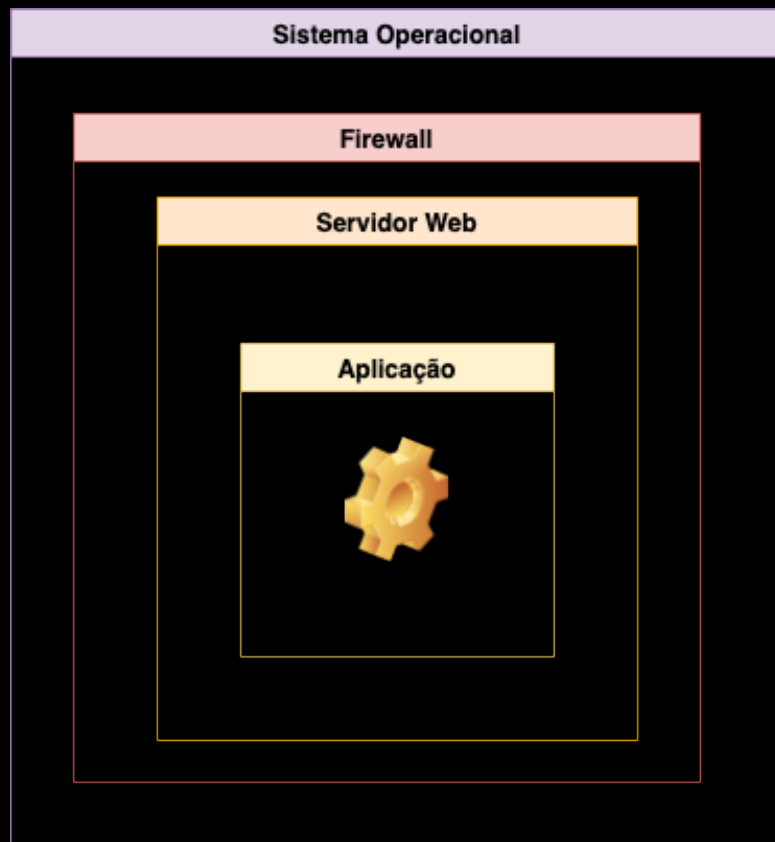
“Falando de Arquitetura >

Você conhece o Lifecycle da sua aplicação?

Eu não falo de MVC em si, mas do fluxo que os dados percorrem, do momento em que são inseridos pelo usuário, até a hora em que são armazenados no Banco de Dados.

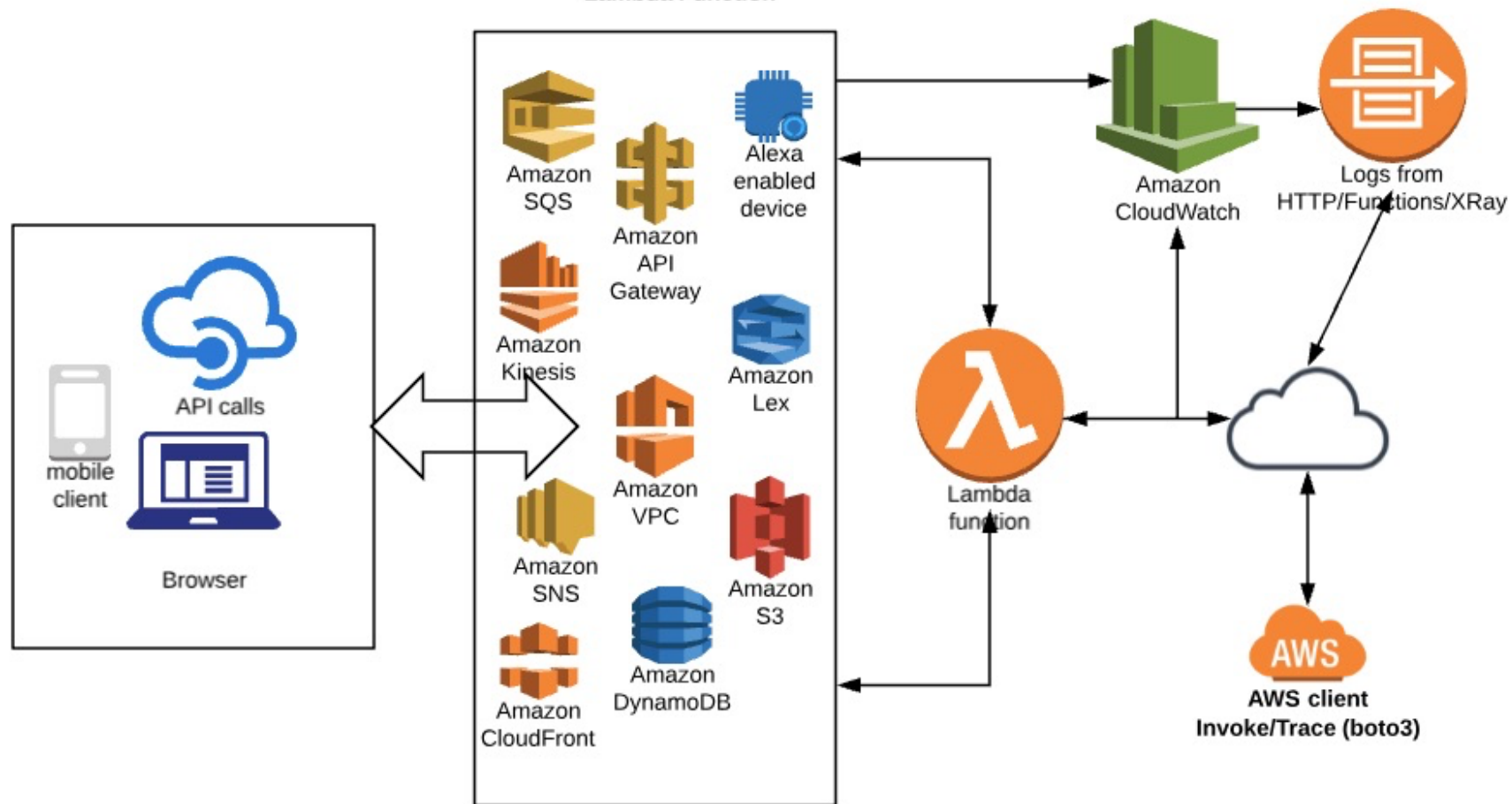
HACKA/FLAG





HACKA/FLAG

Events calling Lambda Function



“Framework é legal, mas nem tanto! >

Frameworks são uma mão na roda para a produtividade e manutenção.
Não há problema algum em utilizar um framework, o problema está no fato de que muitos desenvolvedores atualmente não sabem programar sistemas, só sabem utilizar frameworks.

Um ponto de atenção, é observar as atualizações de segurança, para fazer upgrade nas aplicações o mais rápido possível.

Como mitigar as ameaças?

Secure Development Lifecycle



HACKA/FLAG

O que é SDL?

SDL é uma forma de desenvolver software, baseando-se nos princípios SD3+C

- **Design** - Seu software deve ser seguro por projeto
- **Default** - É impossível ser 100% seguro, então você precisa adotar uma postura padrão para implementar segurança
- **Deployment** - Seguro na Implantação
- **Comunicação** - O time desenvolvimento deve estar preparado para as vulnerabilidades e estar aberto aos usuários e admins.

HACKA/FLAG



Entendendo o SDL

O SDL tem um total de 12 fases que englobam toda a equipe envolvida no processo, incluindo gestores, técnicos, compliance, pentesters, e profissionais de resposta a incidentes.

#1 Promover Treinamentos

Segurança é um trabalho de todo mundo, dev, arquiteto, gerente, etc. E todos eles precisam entender o básico de segurança.

Se não, como é que o software ou serviço que vocês entregam, será seguro?

Como vou ser gerenciado por uma pessoa que não sabe como gerenciar isso?

HACKA/FLAG



#1 Promover Treinamentos

1. Técnicas de anonimato
2. Discrição nas redes sociais
3. Cuidado e atenção e e-mails duvidosos
4. Criação de Senhas fortes
5. Criar Hábitos de segurança no dia-a-dia
 - a. bloquear a tela ao se afastar do pc
 - b. não tirar foto da tela do computador
 - c. não colocar senhas em post-it

#2 Definir os requerimentos de Segurança

Considerar a segurança e privacidade dos dados neste processo é muito fundamental e independente da metodologia de desenvolvimento utilizada, estes requisitos devem ser sempre atualizados de acordo com o cenário de ameaças.

Definindo-se estas necessidades no começo do projeto, ajuda a diminuir os impactos e interrupções causados por falhas e claro, diminui a incidência de falhas também.

#2 Definir os requerimentos de Segurança

1. Quais dados preciso proteger?
2. Quão críticos esses dados são?
3. Qual o impacto de um vazamento destas informações.
4. Onde vou armazenar estas informações?
5. Por onde estes dados irão trafegar?
6. Quem irá manipular e processar tais dados?

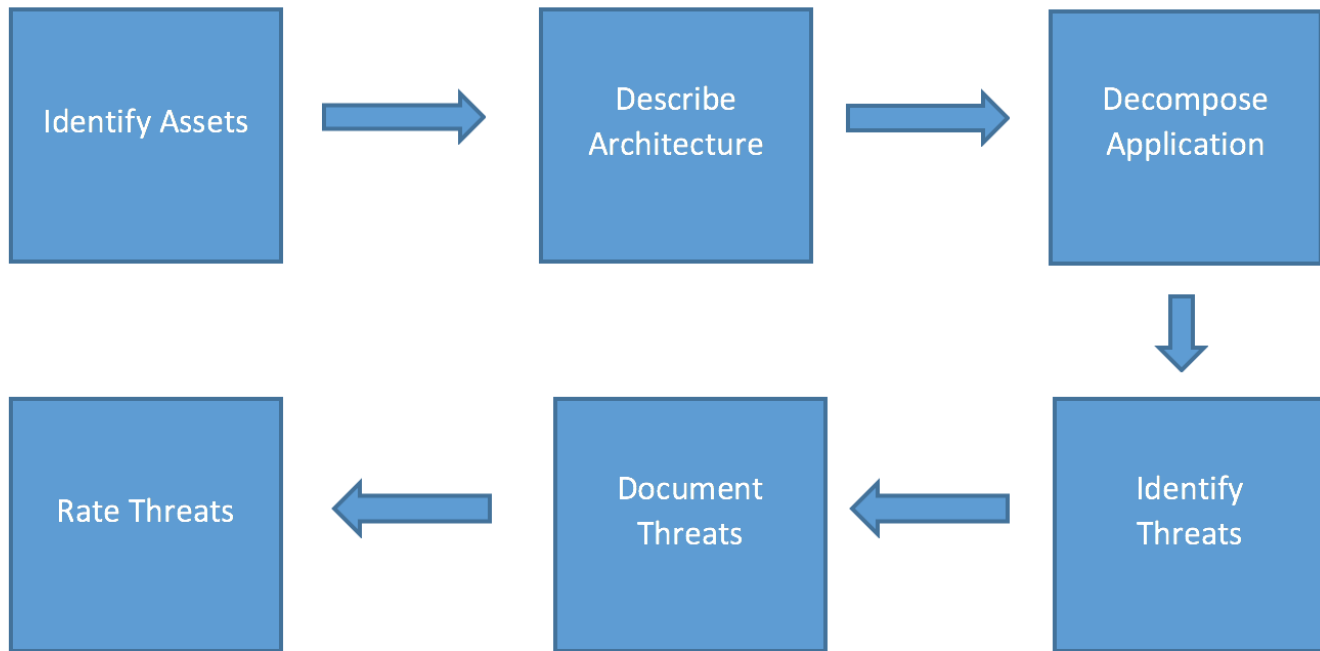
#3 Definir Métricas

É essencial definir os níveis mínimos aceitáveis de segurança e responsabilizar a equipe de engenharia pelo cumprimento desta missão.

É necessário mapear e classificar as ameaças conhecidas antecipadamente, para corrigir problemas e bugs ainda no desenvolvimento.

Isso também envolve definir a gravidade e o tempo de resposta para cada ameaça.

Steps to Threat Modeling



#4 *Executar a modelagem de ameaças*

Sistema

- Quais aplicativos terão acesso a este sistema?
- O que estes aplicativos processam?
- Onde meu sistema está hospedado?
- Estes apps são confiáveis?

Aplicativo

- Quais dados este aplicativo irá processar?
- Quem terá acesso a este app?
- Quais são as medidas de segurança em App providas?

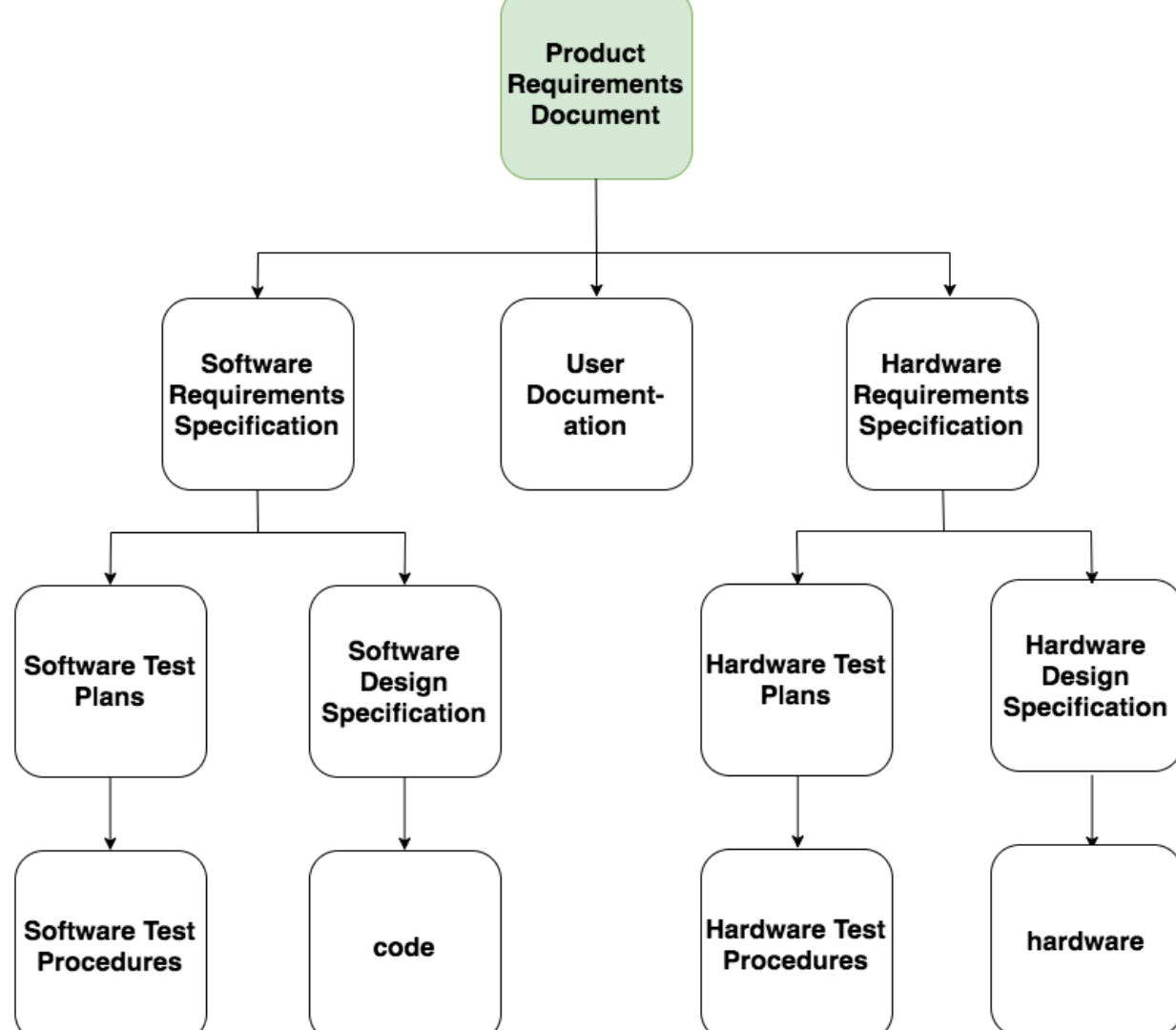
#5 Estabelecer requerimentos de Design

Com o SDL, este estágio fica um pouco mais fácil, pois o produto passa a ser projetado, baseando-se em segurança.

É necessário então, elaborar todos os recursos de segurança necessários para cada módulo da aplicação, a fim de implementar estes módulos da forma mais segura possível.

É necessário muita atenção nesta parte, pois implementar isto se provou tão complexo quanto ao ponto de apresentar mais vulnerabilidades.

HACKA/FLAG



#6 Definir padrões de Criptografia

Quais são as cifras e algoritmos mais adequados para o meu problema?

Como proteger:

- Senhas
- Arquivos
- Comunicação

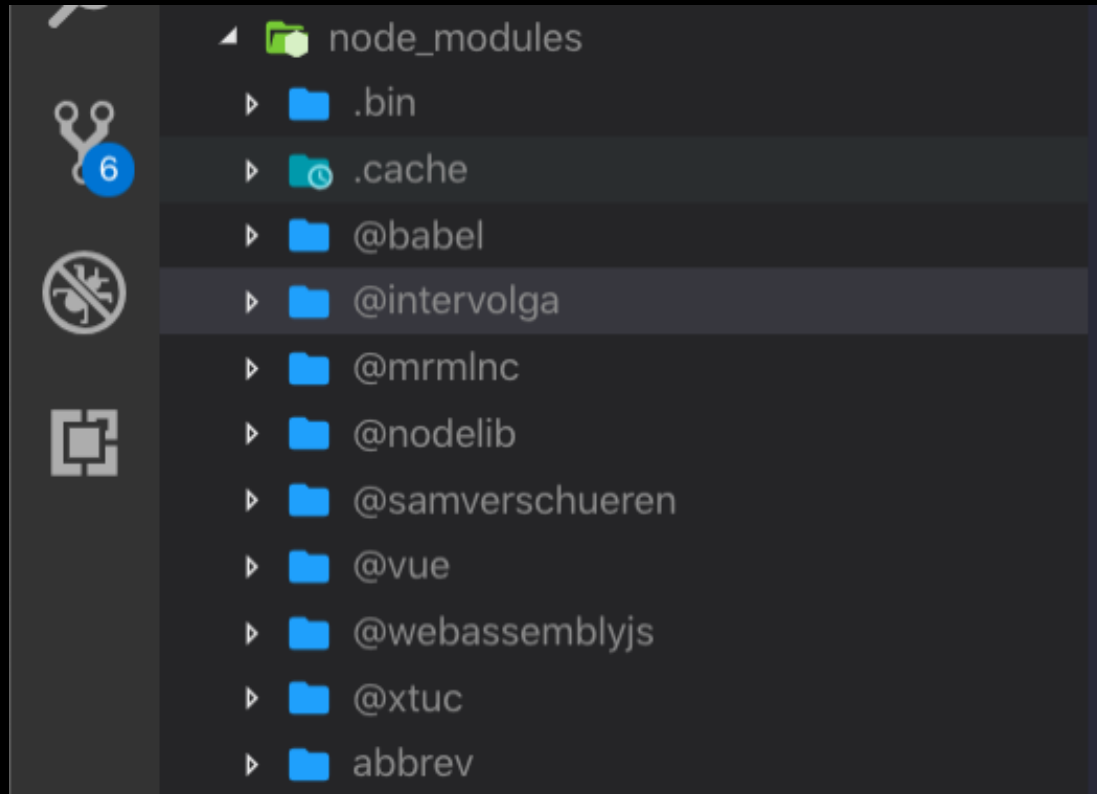


HACKA/FLAG

#7 Gerenciamento de Riscos por utilizar componentes de terceiros

Muitos projetos são construídos em cima de componentes de terceiros, ao utilizar uma biblioteca ou componente de terceiros, é necessário mensurar o impacto de uma vulnerabilidade no seu projeto.

Basicamente precisamos montar um inventário de todos os componentes de terceiros que estão sendo utilizados no projeto.

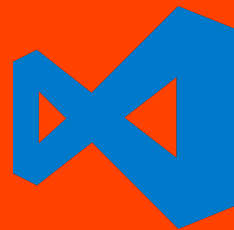
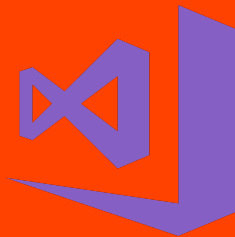
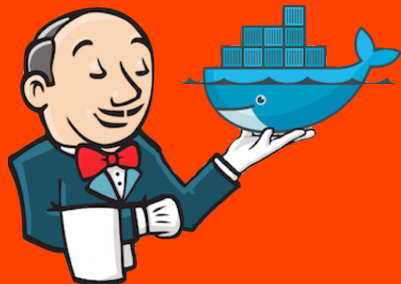


HACKA/FLAG

#8 Utilizar ferramentas aprovadas

Esta parte envolve a definição e publicação de uma lista de ferramentas aprovadas para a utilização no desenvolvimento e compilação da aplicação.

Isso envolve situações onde uma ferramenta (compilador por exemplo) contém vulnerabilidades que podem resultar na quebra de segurança do projeto final.



GitLab

HACKA/FLAG

#9 Executar Análise de Segurança Estática (SAST)

Analisar o código-fonte de forma estática, permite detectar possíveis falhas de segurança, no momento em que o código é criado.

O SAST geralmente é executado em cada *commit* que é feito no projeto, assim fica mais escalável, e simples de ser feito e qualquer problema que venha a ser detectado nesta fase, elimina o risco de que o código defeituoso seja integrado ao projeto final.

```

115 115     lineElements.css 'background-color',
116 116
117 117     if lineRange.length is 1
118 118 -   $("\#{anchorPrefix}LC#\{lineRange[0]}\").css 'background-color', "#ffc"
119 119 +   $("\#{anchorPrefix}LC#\{lineRange[0]}\").css 'background-color', "#f8eec7"
120 120
121 121     else if lineRange.length > 1
122 122         i = lineRange[0]
123 123         while i <= lineRange[1]
124 124 -   $("\#{anchorPrefix}LC#\{i}\").css 'background-color', "#ffc"
125 125 +   $("\#{anchorPrefix}LC#\{i}\").css 'background-color', "#f8eec7"
126 126         i++
127 127
128 128     # Highlight and scroll to the lines in the current location hash.

```

24 ■■■■■ app/assets/javascripts/github/pages/diffs/linkable-line-number.coffee



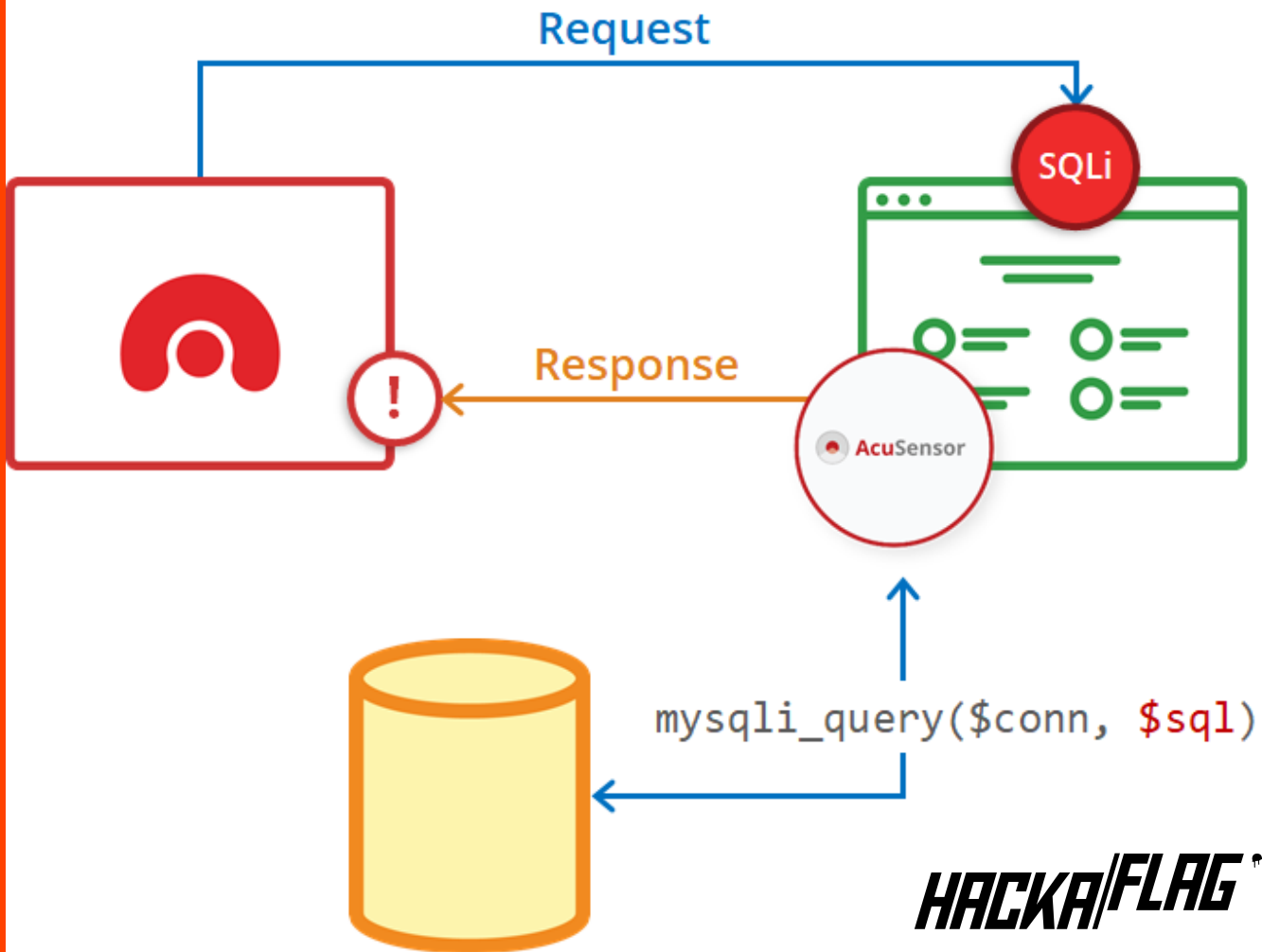
@@ -9,3 +9,27 @@

HACKA/FLAG

#10 *Executar Análise de Segurança Dinâmica (DAST)*

Esta etapa deve ser executada após a compilação, durante a execução do projeto, que é quando todos os componentes estão totalmente integrados e em execução.

Existem diversas ferramentas que executam uma série de ataques pré-construídos contra a sua aplicação, também há algumas que monitoram problemas ligados a corrupção de memória, privilégios de usuários, e muitos outros.



HACKA/FLAG™

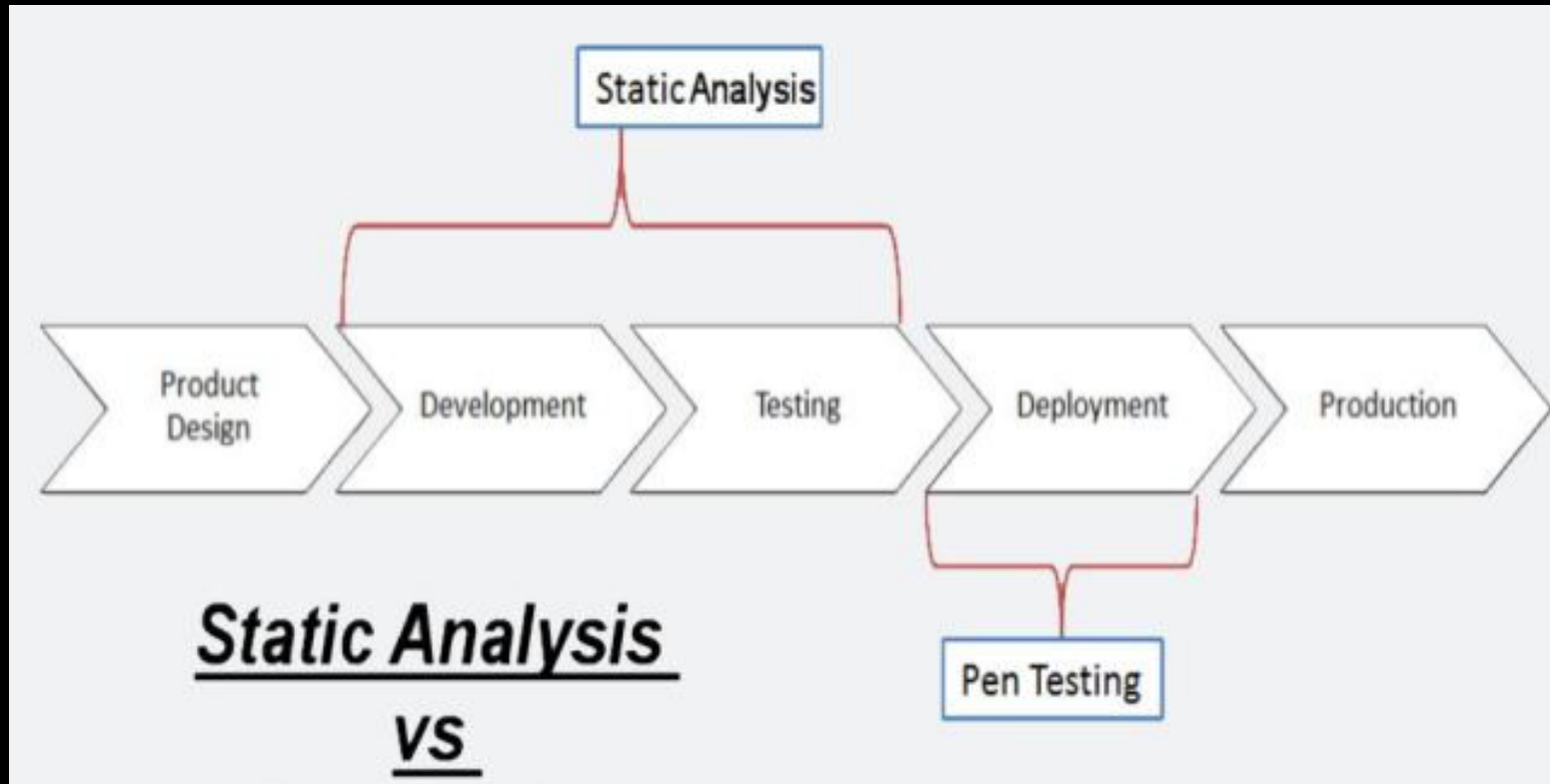
#11 Executar um Pentest

Essa etapa é executada por um profissional de segurança especializado.

O foco do pentest é identificar problemas de codificação (lógica por exemplo), erros de configuração da aplicação e da infraestrutura onde ela roda.

O processo é executado manualmente e também automatizado, para possibilitar uma análise maior e mais aprofundada do sistema.

HACKA/FLAG



Pentest

HACKA/FLAG

#12 Definir um processo padrão de resposta a incidentes

Na última fase do SDL, é necessário definir como será tratado cada incidente que possa surgir ao longo do tempo.

É necessário criá-lo em conjunto com um time de dedicado de resposta a incidentes.

Geralmente o plano inclui até quem deve ser contactado em caso de incidente, e protocolos para como iremos tratar a falha caso ela esteja em códigos herdados de outros times da organização ou de terceiros.

É necessário testar o plano de resposta a incidentes antes de ele ser necessário.

Bibliografia

- **Trilhas em Segurança da Informação**
Caminhos e Ideias para a proteção de dados.
ELIAS, Wagner; CABRAL, Carlos e CAPRINO, Willian
isbn:9788574526867
- **Microsoft SDL Framework**
<https://www.microsoft.com/en-us/securityengineering/sdl/practices>
- Experiência diária como desenvolvedor de aplicações.



% Obrigado! _

Dúvidas?

Thau0x01 Santos

Professional Bug Maker

@thau0x01 | thauan@hackware.tech

HACKA/FLAG