Devops

Capítulo 7. Segurança

Prof. Thiago Chierici

Aula 7.1. Desafios

Nesta aula

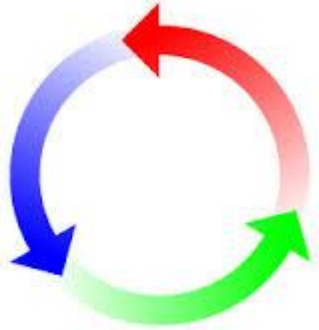
- Como trabalhar a segurança no Devops.
- Os desafios comuns para incluir segurança no processo.

DevXXXOps - Novos paradigmas e o mundo real

* Adaptado de Geek and Poke

Onde entra a segurança no Devops?

- Desde o início.
- Em ciclos contínuos, como o desenvolvimento e a qualidade.



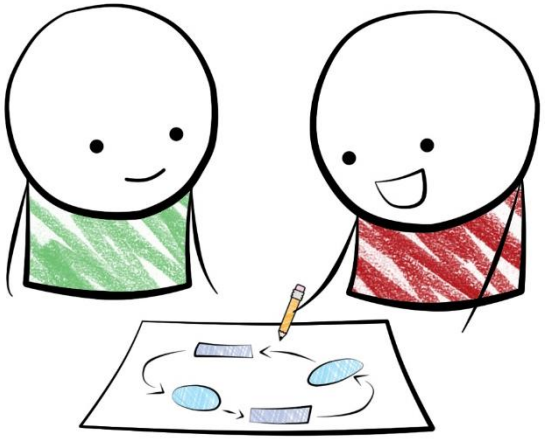
Os desafios

- Velocidade:
  - Os modelos de desenvolvimento estão cada vez mais rápidos.
  - Amazon executou 50 milhões de alterações em 2014.
    - Uma por segundo.
  - Como conseguimos garantir a segurança em cada uma delas?

Fonte: https://www.safaribooksonline.com/library/view/devopssec/9781491971413/ch02.html

Os desafios

- Design (ou ausência dele):
  - Princípio YAGNI.
  - Pensamento Lean.
  - Experimentação.



- Menos planejamento e mais ciclos curtos de criação e validação.

Os desafios

- Desperdício:
  - Lean thinking.
  - Não podemos evitar 100% dos riscos o tempo inteiro.
  - É preciso encontrar o ponto ideal.
    - 80/20.

Os desafios

- Nuvem:
  - Abordagem traz várias vantagens.
  - Também gera alguns riscos comuns, por exemplo:
    - Concentração dos recursos.
    - Gestão de credenciais.
    - Indisponibilidade eventual.

Os desafios

- Microservices:
  - Superfície de ataque.
  - Programação poliglota.
  - Padronização.

Conclusão

☑ Segurança é um importante requisito que precisa ser endereçado de forma contínua, durante o desenvolvimento e com muita automação.

Próxima aula

- Detalhes sobre como desenvolver com segurança.

Aula 7.2. Desenvolvendo com segurança

Nesta aula

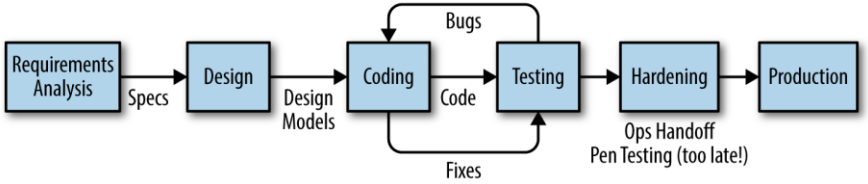
- Como incorporar a segurança no ciclo de desenvolvimento.

Por que incorporar a segurança no desenvolvimento

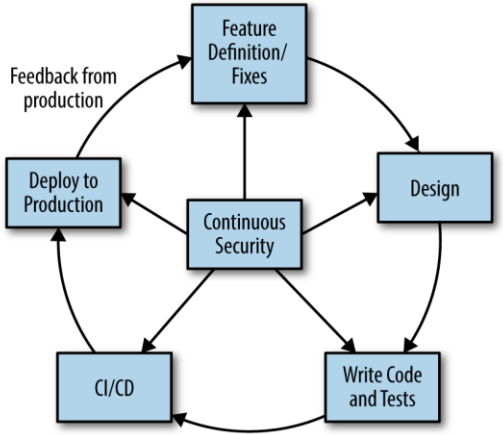
- Obter as mesmas vantagens dos outros ciclos contínuos já abordados na disciplina.
- Integrar os times e profissionais em um único time e propósito.
- Antecipar o trabalho de segurança e garantir um código sempre pronto.

Antecipar

Waterfall



DevOps



Fonte: https://www.safaribooksonline.com/library/view/devopssec/9781491971413/ch03.html#keys_to_injecting_security_into_devops

Controles OWASP

- Verificar segurança cedo e sempre.
- Parametrizar queries (Injection).
- Codificar dados (XSS).
- Validar entradas.
- Implementar detecção de intrusão.
- Tratamento de exceções.

Fonte: https://www.safaribooksonline.com/library/view/devopssec/9781491971413/ch03.html#keys_to_injecting_security_into_devops

Infra As Code *

- Bloqueio por padrão.
- Padronização.
- Clareza.
- Processos de validação:
  - Code review.
  - Testes.

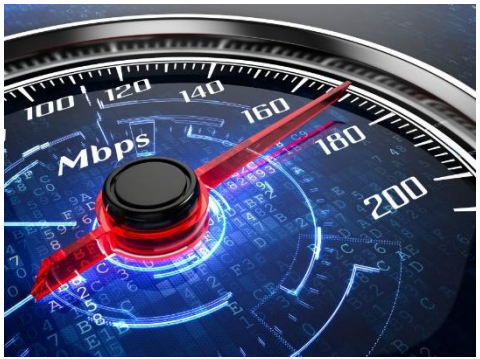
* Mais detalhes na próxima aula

Evoluções pequenas e incrementais

- Ciclos de validação mais frequentes.
- Menores unidades a serem validas a cada ciclo.
- Mais fácil de resolver os problemas encontrados.
  - Geralmente não precisarão ser adiados.
- Técnicas de deploy já estudadas também podem contribuir para reduzir a exposição ao risco.

Velocidade a seu favor

- Em um primeiro momento a velocidade e a quantidade de entregas podem assustar.
- Por outro lado, em caso de brechas encontradas, já temos uma capacidade plena de fazer correções com grande velocidade.



Conclusão

- ☑ A segurança pode ser trabalhada de forma contínua ao longo do desenvolvimento.
- ☑ Ferramentas e processos podem nos ajudar a garantir que esse esforço tenham um retorno ótimo.

Próxima aula

- Como garantir que todo o trabalho focado na segurança também seja entregue como código.

Aula 7.3. Security As Code

Nesta aula

- Como implementar a segurança usando a mesma abordagem utilizada no desenvolvimento – a segurança como código.

Antecipação

- “Não deixe para amanhã o que pode fazer hoje.”
  - Autor desconhecido.

Incluindo segurança na entrega contínua

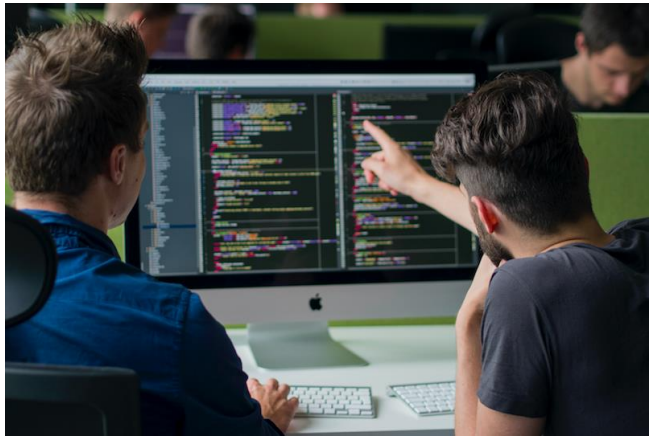
- Pre-commit:
  - Análise estática com ferramentas.
  - Code-review.
- Commit stage (CI):
  - Validação de brechas no código.
- Acceptance stage:
  - Validação de brechas no ambiente montado:
- Deploy em produção e pós-produção:
  - Monitoração contínua.

Design seguro

- Método ágeis não trabalham com grande fase de design.
- É preciso incluir uma validação de segurança durante o detalhamento técnico:
  - Checklist.
  - Entrevista com especialista.
  - Solution review.

Escrevendo código seguro

- Code-reviews para a segurança.
- Pair-programming.
- SAST - Static Application Security Testing.

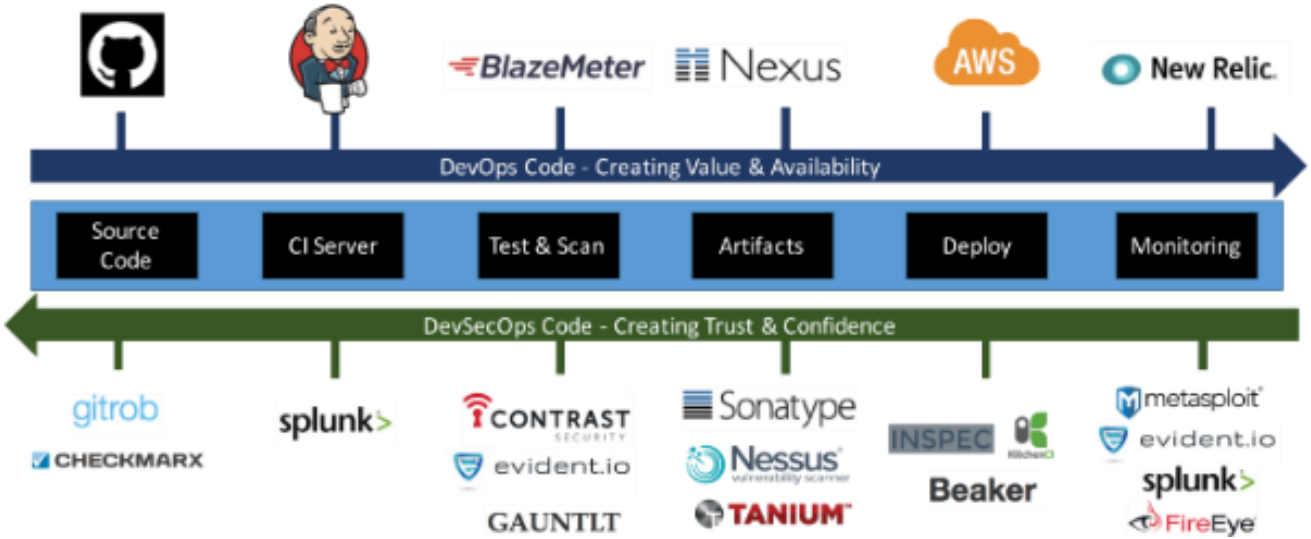


Testes

- DAST – Dynamic scanning.
- Fuzzing – Entradas anômalas.
- Testes unitários e integração em segurança.
- Ataques automatizados.
- Pen testing.
- Gestão de vulnerabilidades.

Segurança da infraestrutura

- Gestão de configuração automatizada.
- Segurança de pipeline de entrega.



Segurança no ambiente de produção

- Verificações de execução (monkeys).
- Acompanhamento constante.

Conclusão

☑ Também é possível implementar a segurança como código.

☑ Precisamos considerar diferentes abordagens em conjunto para garantir a segurança da informação.

Próxima aula

- Outro case Devops, desta vez com ênfase nos resultados relacionados à segurança.

Aula 7.4.1. Inspiração – Netflix – Segurança (Parte 1)

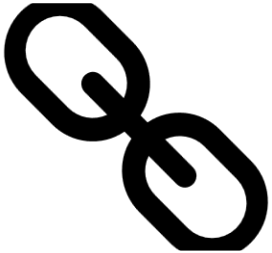
Nesta aula

- Vamos conhecer mais sobre um dos mais famosos ‘unicórnios’ Devops: a Netflix e sua cultura da liberdade e responsabilidade.

Histórico

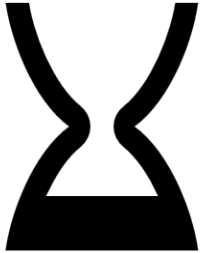
- Chegaram a tentar cuidar da própria infra.
- Voltaram pra AWS:
  - Talvez o maior cliente do mundo.
- Chamam sua estratégia de ‘NoOps’.
  - Não existe time de operação / infra.

Segurança dividida em quatro pilares

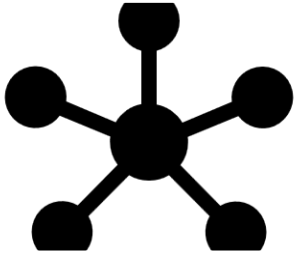


Responsabilidade compartilhada

Rastreabilidade



Segurança contínua



Fragmentação

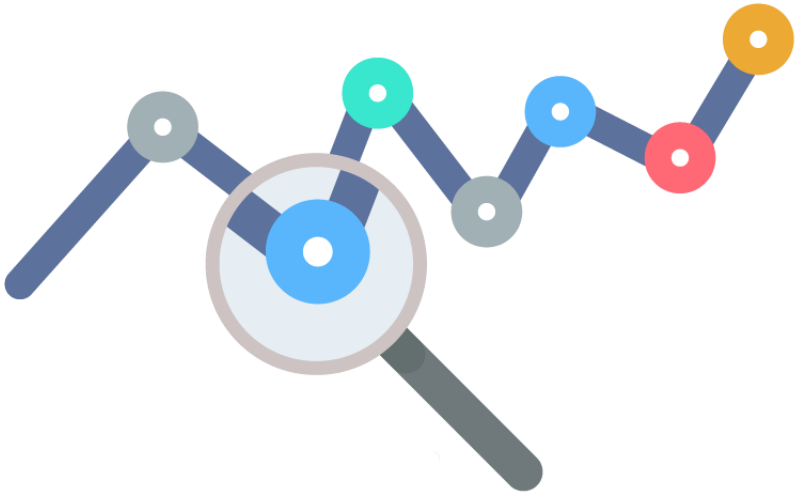
Responsabilidade compartilhada

- AWS (nuvem):
  - Provisionamento de máquinas.
  - Gestão de vulnerabilidades.
  - Armazenamento de dados.
  - Backups.
  - Segurança física.
- Netflix / plataforma:
  - Validações de segurança.
  - Analytics.
  - Monitoramento.

Fonte: https://www.oreilly.com/library/view/devopssec/9781491971413/ch04.html#learning_from_failure_game_dayscomma_red

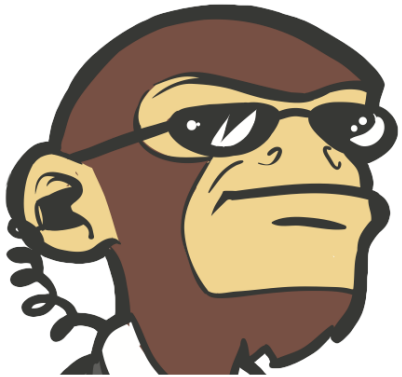
Rastreabilidade

- Código-fonte x Commit x Pull Request x Funcionalidade x Deploy:
  - Ferramental integrado e de fácil uso e recuperação.



Visibilidade – Segurança contínua

- Automação / Ferramentas:
  - Próxima aula mostrará um exemplo.
  - Outras ferramentas:
    - Security Monkey, Conformity Monkey e Penguin Shortbread.
  - Visibilidade permanente.



Fragmentação

- Arquitetura fragmentada -> Menor exposição.
  - Dados e microsserviços.
- Evitar privilégio excessivo:
  - Mínimo de permissões pra fazer o que é preciso.

Conclusão

☑ É importante que o trabalho de Devops esteja alinhado com a cultura da organização:

- Evolua suas técnicas, suas ferramentas e principalmente sua **cultura.**

Próxima aula

- Mais Netflix.
- Uma ferramenta de segurança que pode te ajudar.
- Mindset de contribuição com a comunidade.

Aula 7.4.2. Inspiração – Netflix – Ferramentas (Parte 2)

Nesta aula

- Uma ferramenta de segurança criada, utilizada e compartilhada pelo time Netflix.
- A importância da colaboração / interação com o mercado.

Diffy – Colaboração da Netflix com o mercado

Fonte: https://medium.com/netflix-techblog/netflix-sirt-releases-diffy-a-differencing-engine-for-digital-forensics-in-the-cloud-37b71abd2698

Diffy - O que é

- Uma ferramenta de triagem.
- Busca encontrar máquinas comprometidas.

Diffy - Funcionalidades

- Encontrar portas abertas que não deveriam estar:
  - A partir de um modelo base.
  - A partir da comparação com outras máquinas do cluster.

Netflix – muito mais a ser conhecido

- Build/Delivery:
  - Nebula, Aminator e Spinnaker.
- Libraries:
  - Eureka, Archaius, Ribbon e Frenzo.
- Persistência:
  - Hollow.
- Insights:
  - Atlas e Vector.
- Security:
  - Security monkey.
- Outros: Titus.

NETFLIX | OSS

https://netflix.github.io/

Conclusão

☑ Compartilhar seus resultados com a comunidade é bom para todos:

- Também é importante saber reutilizar as criações dos outros.

Próximos passos

- Esta é a última aula desta etapa, continue estudando!
- Assim como funciona o Devops...