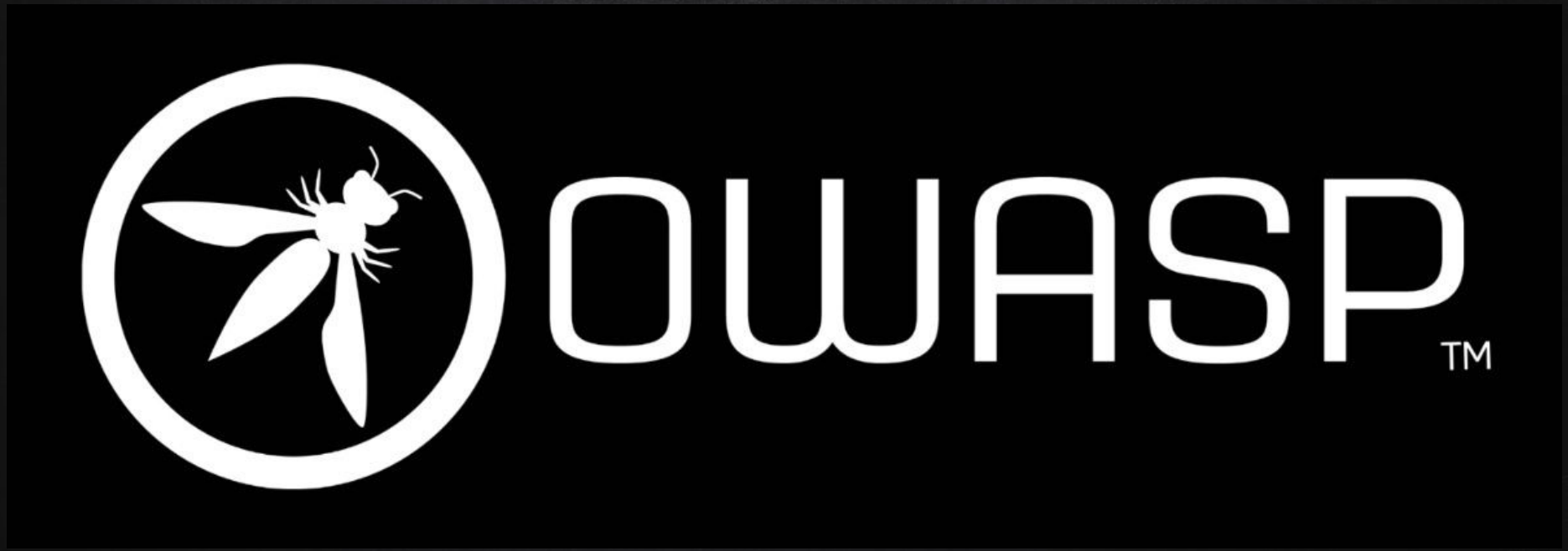

UNAERP
Universidade de Ribeirão Preto
Campus Ribeirão Preto · Campus Guarujá

Segurança da Informação

Tópicos Especiais em Engenharia

Maiores dúvidas

- ✘ Como Tor também serve para navegar na surface ele também cobre rastros como na deep web ou ele é menos criterioso nesse ponto?
  - Ele dificulta muito os rastros, mas não é 100% seguro
- ✘ O que é, e como funciona o ataque do vírus stuxnet ?
  - https://rasoolirfan.com/2014/12/23/stuxnet-are-we-prepared-to-defend/
- ✘ Qual foi a brecha de segurança encontrada no app Zoom ?
  - vazamento de dados / compartilhamento sem permissão
  - instalação de servidor do Zoom
- ✘ Qual o vírus considerado o mais prejudicial e de difícil detecção atualmente?

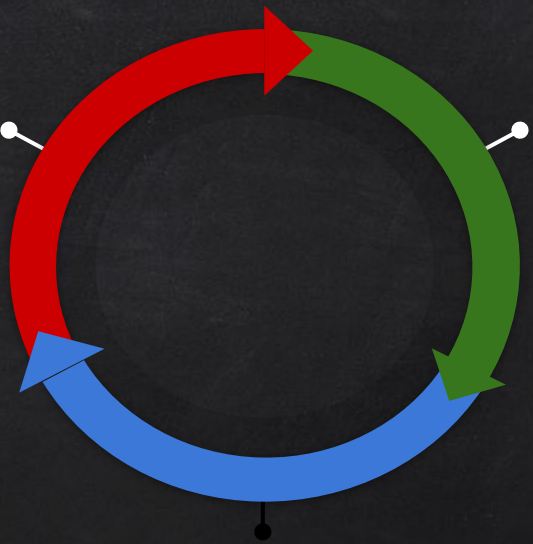

OWASP
TM

Mudanças 2013 - 2017

- ✘ Principais motivadores
  - Microserviços
  - SPA – Single Page Applications

Como desenvolver softwares mais seguros?

TDD – Test Driven Development



RED

Escrever testes que irão falhar

GREEN

Desenvolver código para passar o teste

REFACTOR

Refatorar código

TOP 10 Controles Pró-Ativos

by OWASP

C1: Defina seus requisitos de segurança

- ✘ OWASP Application Security Verification Standard
  - Catálogo de verificações e critérios de segurança
  - https://github.com/OWASP/ASVS#latest-released-version

C2: Utilize Frameworks e Libs Seguras

- ✘ Use bibliotecas e frameworks de fontes seguras
- ✘ Crie e mantenha um catálogo de todas bibliotecas externas
- ✘ https://owasp.org/www-project-dependency-check/
- ✘ https://retirejs.github.io/retire.js/

C3: Garanta Segurança no Acesso a Bancos de Dados

- ✘ Consultas
- ✘ Configuração
- ✘ Autenticação
- ✘ Comunicação
- ✘ https://bobby-tables.com/
- ✘ https://cheatsheetseries.owasp.org/cheatsheets/Query_Parameterization_Cheat_Sheet.html

C4: Encode / Escape

- ✘ Transformar < em <
- ✘ \ antes de ”
- ✘ https://owasp.org/owasp-java-encoder/
- ✘ https://docs.microsoft.com/en-us/aspnet/core/security/cross-site-scripting?view=aspnetcore-3.1
- ✘ https://framework.zend.com/manual/2.4/en/modules/zend.escaper.theory-of-operation.html

C5: Valide Todas as Entradas

- ✘ Validação sintática
  - ID da conta tem que ter 4 dígitos
- ✘ Validação semântica
  - Data precisa estar num intervalo
- ✘ Whitelist / Blacklist
- ✘ ClientSide / ServerSide
- ✘ https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html

C6: Implemente Identidade Digital

- ✘ Autenticação
- ✘ Armazenamento de Senhas
- ✘ Recuperação de Senha
- ✘ Gerenciamento de Sessão
- ✘ https://pages.nist.gov/800-63-3/sp800-63-3.html

C7: Reforce Controles de Acesso

- ✘ Discretionary Access Control (DAC)
- ✘ Mandatory Access Control (MAC)
- ✘ Role Based Access Control (RBAC)
- ✘ Attribute Based Access Control (ABAC)
- ✘ Bloqueie por padrão
- ✘ Princípio do privilégio mínimo
- ✘ Tenha log de todos eventos de controle de acesso

C8: Proteja todos os dados

- ✘ Classifique seus dados
- ✘ Criptografe na transmissão
- ✘ Criptografe no armazenamento
- ✘ Mobile: Armazenamento Local
- ✘ https://github.com/google/tink

C9: Implemente Logs de segurança e Monitoramento

- ✘ Detectar intrusões
- ✘ Análise e auditorias
- ✘ Requisitos para legislações (compliance)
- ✘ https://owasp.org/www-pdf-archive/OWASP_Logging_Guide.pdf

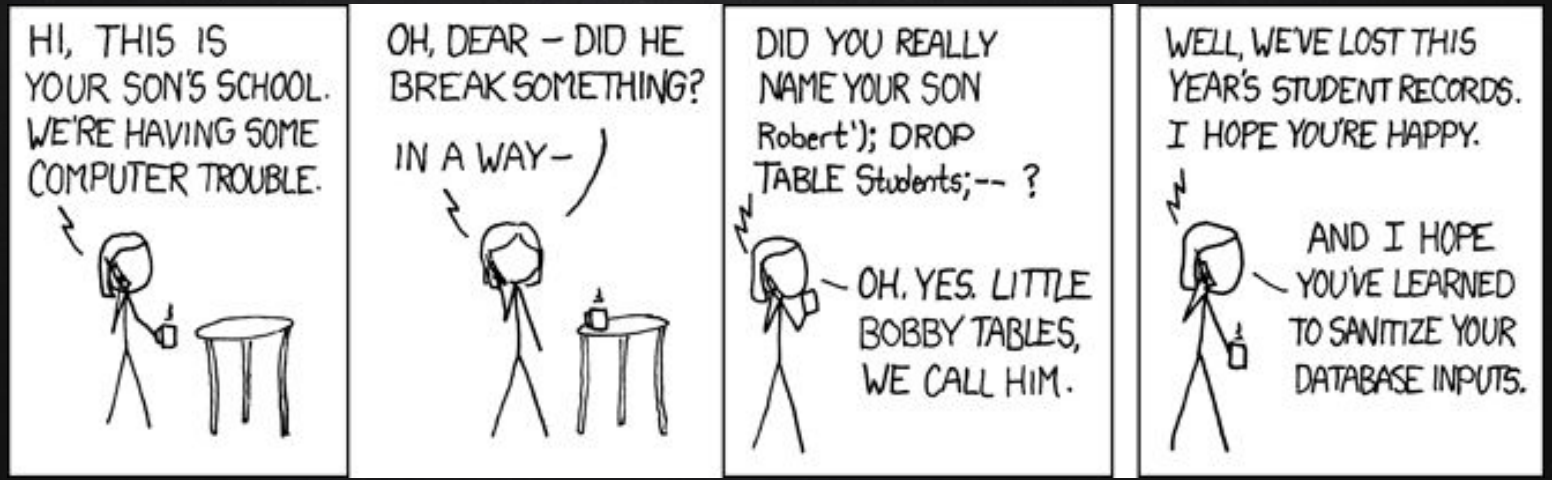
C10: Trate todos erros e exceptions

- ✘ Alguns ataques podem disparar erros que ajudam na detecção de ataques em curso
- ✘ Mantém seu código confiável e seguro
- ✘ https://owasp.org/www-pdf-archive/OWASP_Code_Review_Guide_v2.pdf

Top 10 Vulnerabilidades

A1: Injeção

- ✘ Entrada do usuário é concatenada com código executável
- ✘ SQL Injection



A1: Injeção – Prevenção

- ✘ Não use consultas SQL dinâmicas quando puderem ser evitadas
  - Instruções preparadas
  - Procedimentos Armazenados
- ✘ Atualizações
- ✘ Reduzir superfície de ataque
  - xp_cmdshell
- ✘ Use privilégios apropriados
- ✘ ORM – Object-relational mapping
  - Mapeamento objeto-relacional

A2: Quebra de Autenticação

- ✘ ID's de sessão não são rotacionadas depois de logar
- ✘ Permitir ataque de força bruta ou outros ataques automatizados
- ✘ Usar senhas padrão, senhas fracas ou senhas conhecidas

A2: Quebra de Autenticação – Prevenção

- ✘ Alterar IDs de sessão depois de login com sucesso
- ✘ Implementar proteção de força bruta
- ✘ Implementar complexidade de senhas

A3: Exposição de Dados Sensíveis

- ✘ Dados sensíveis são transmitidos ou armazenados em texto claro
- ✘ Usar criptografias fracas ou antigas

A3: Exposição de Dados Sensíveis - Prevenção

- ✘ Criptografar todos dados sensíveis tanto no armazenamento como no trânsito
- ✘ Usar algoritmos de criptografia, protocolos e chaves atualizados e fortes.

A4: Entidades Externas de XML (XXE)

- ✘ Atacantes podem explorar vulnerabilidades em processadores XML se puderem fazer upload de XML ou incluir conteúdo malicioso num conteúdo de um documento XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE teste [
<!ELEMENT teste ANY>
<!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<teste>&xxe;</teste>
```

A4: Entidades Externas de XML (XXE) - Prevenção

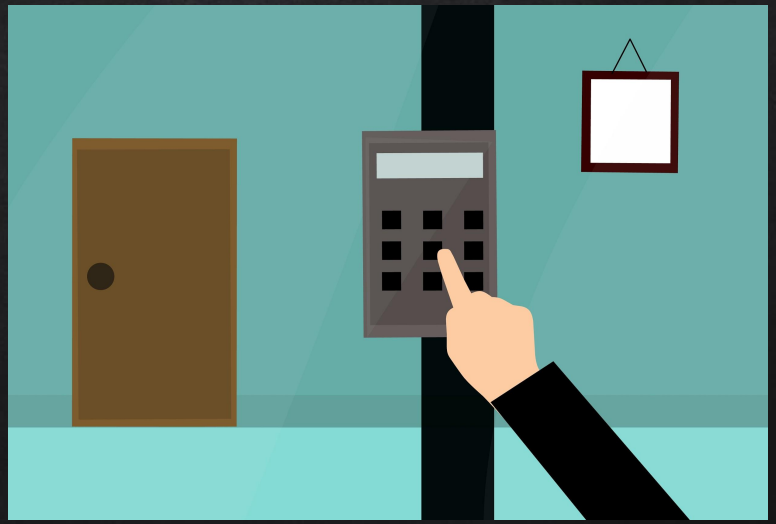
- ✘ Desabilitar entidades externas ao XML e processamento de DTD em todos XML Parsers na aplicação
- ✘ https://cheatsheetseries.owasp.org/cheatsheets/XML_External_Entity_Prevention_Cheat_Sheet.html#Unmarshaller

A5: Quebra de Controle de Acessos

- ✘ Escalar ou elevar privilégios
- ✘ Acesso de usuário regular com permissões de administrador

A5: Quebra de Controle de Acessos – Prevenção

- ✘ Implementar mecanismos de controle de acessos



A6: Configurações de Segurança Incorretas

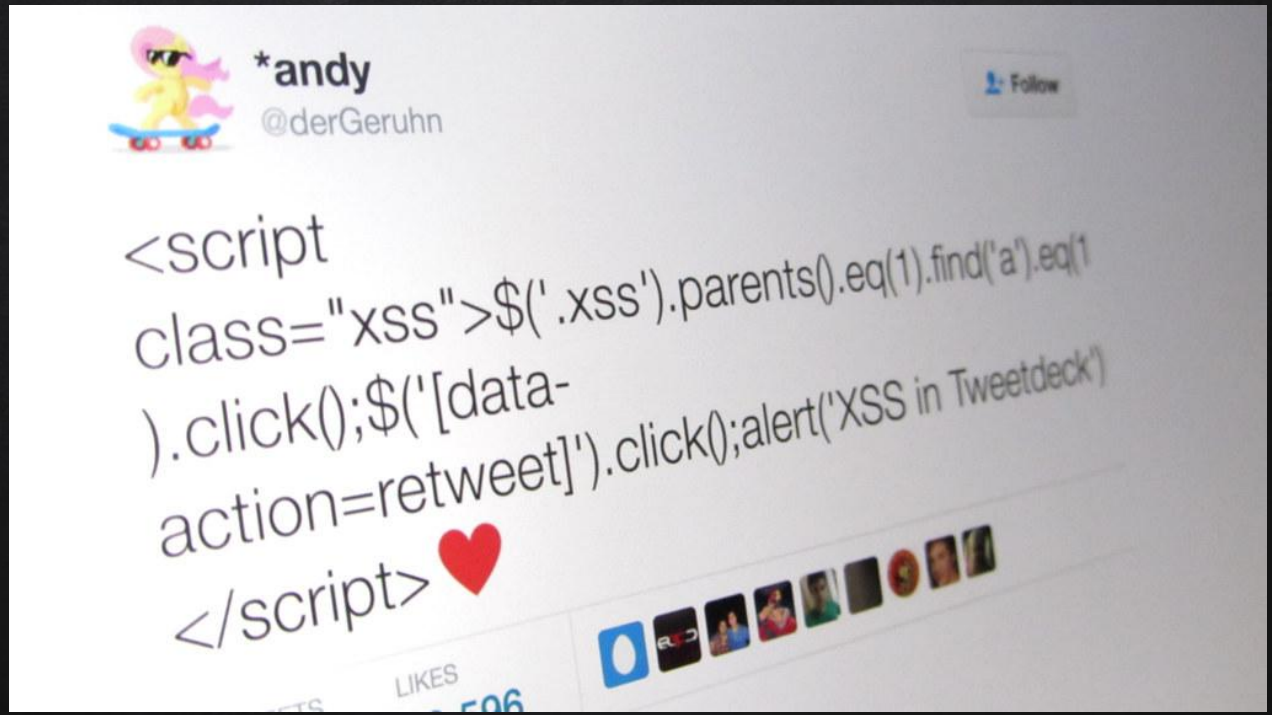
- ✘ Funcionalidades desnecessárias habilitadas ou instaladas
  - Portas
  - Serviços
- ✘ Contas padrão
- ✘ Senhas padrão

A6: Configurações de Segurança Incorretas – Prevenção

- ✘ Fechar portas desnecessárias
- ✘ Desabilitar serviços desnecessários
- ✘ Remover contas padrão
- ✘ Mudar senhas padrão

A7: Cross-Site Scripting (XSS)

- ✘ Atacante consegue executar scripts no navegador da vítima
- ✘ Beef XSS



A7: Cross-Site Scripting (XSS) - Prevenção

- ✘ Validar entrada de dados para todas entradas
  - Tanto back quanto front
  - White-lists
- ✘ Codificar (Encode) saída

A8: Desserialização Insegura

- ✘ Serialização é o processo de traduzir estruturas de dados ou objetos em formatos que podem ser armazenados ou transmitidos e reconstruídos depois (desserialização)
- ✘ Desserialização Insegura
  - Atacante muda o objeto entre a serialização e a deserialização

A8: Desserialização Insegura – Prevenção

- ✘ Não desserializar objetos de origem não confiáveis

A9: Utilização de Componentes Vulneráveis

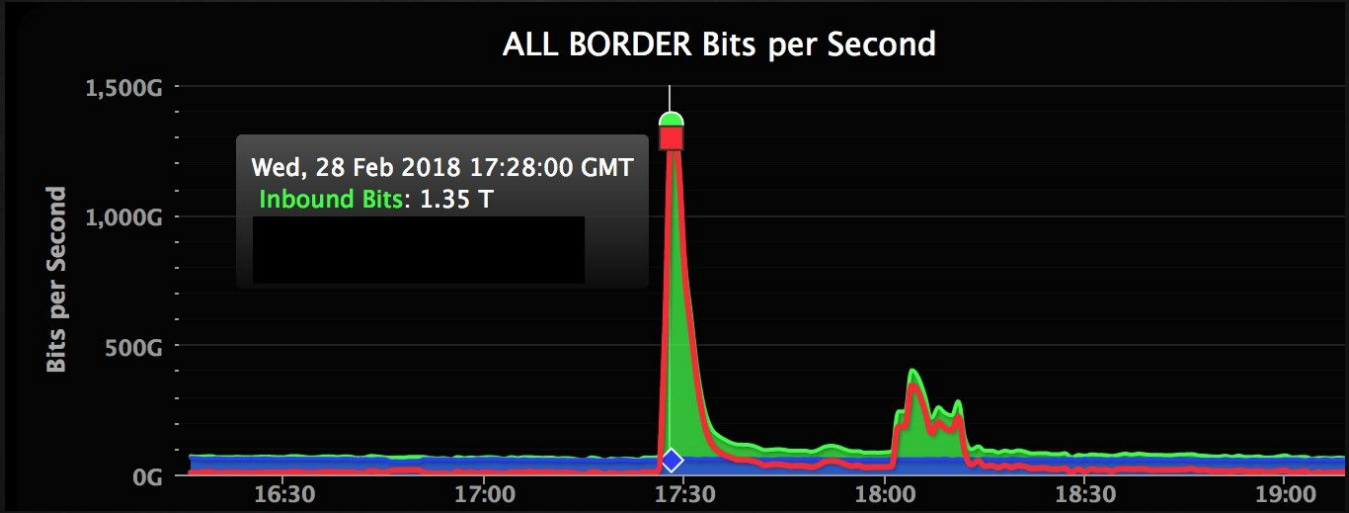
- ✘ Software são vulneráveis, desatualizados e ficam sem suporte

A9: Utilização de Componentes Vulneráveis – Prevenção

- ✘ Mantenha componentes atualizados

A10: Registro e Monitorização Insuficiente

- ✘ Log insuficientes
  - Logins
  - Falhas de logins
  - Transações de alto valor



A10: Registro e Monitorização Insuficiente – Prevenção

- ✘ Gravar Log de eventos importantes com contexto do usuário
  - Username
  - Endereço IP
  - Data / hora

DÚVIDAS?

- ✘ Não esqueça de enviar sua maior dúvida pelo link que está no ClassRoom.
- ✘ Críticas/Sugestões: ezarpelao@unaerp.br

Credits

Special thanks to all the people who made and released these awesome resources for free:

- ✘ Presentation template by SlidesCarnival
- ✘ Photographs by Unsplash