OWASP
Open Web Application
Security Project

OWASP TOP 10

Open Web Application Security Project (OWASP)

Leader Owasp Chapter Cuiabá // 2011-2017.

Home: owasp.org

Email: kembolle@owasp.org

FUG-BR (Grupo Brasileiro de Usuários FREEBSD)

Home: bsd.com.br

Email: kembolle@bsd.com.br

Research Information Security and Psychoanalysis Forensic.

Home: kembolle.com.br

Email: contato@kembolle.com.br

IT Security Analyst at Vice-Governadoria - MT

Home: mt.gov.br

Email: kembolleoliveira@vicegovernadoria.mt.gov.br

Open Web Application Security Project (OWASP)

È uma comunidade aberta, dedicada a capacitar as organizações a desenvolver, adquirir e manter aplicações confiáveis na Internet.

A Fundação OWASP é uma organização internacional sem fins lucrativos, registrada sob o 501c3 (IRS) US e não possui nenhuma associação com produtos ou serviços comerciais.

Todas as ferramentas, documentos, fóruns, e capítulos da OWASP são livres e abertos para qualquer pessoa que estiver interessada em melhorar a segurança de aplicações.

Principais Valores

ABERTO – Tudo no OWASP é radicalmente transparente, das finanças ao código.

INOVAÇÃO – OWASP encoraja e apoia inovações/experimentos para solucionar os desafios da segurança de aplicações.

GLOBAL – Qualquer pessoa no mundo é encorajada a participar da comunidade da OWASP

INTEGRIDADE – OWASP é uma comunidade global, honesta e confiável e um fornecedor neutro.

A História da Comunidade

2001: Foi inciado em 9 de Setembro Mark Curphey e Dennis Groves
2003: Jeff Williams vem servindo voluntariamente como Chair do OWASP
2004: Foi estabelecida em 2004 como uma organização sem fins lucrativos nos EUA (501(c)(3) organization) .
Milhares de membros hoje em dia • Mais de 80 capítulos locais ativos :)

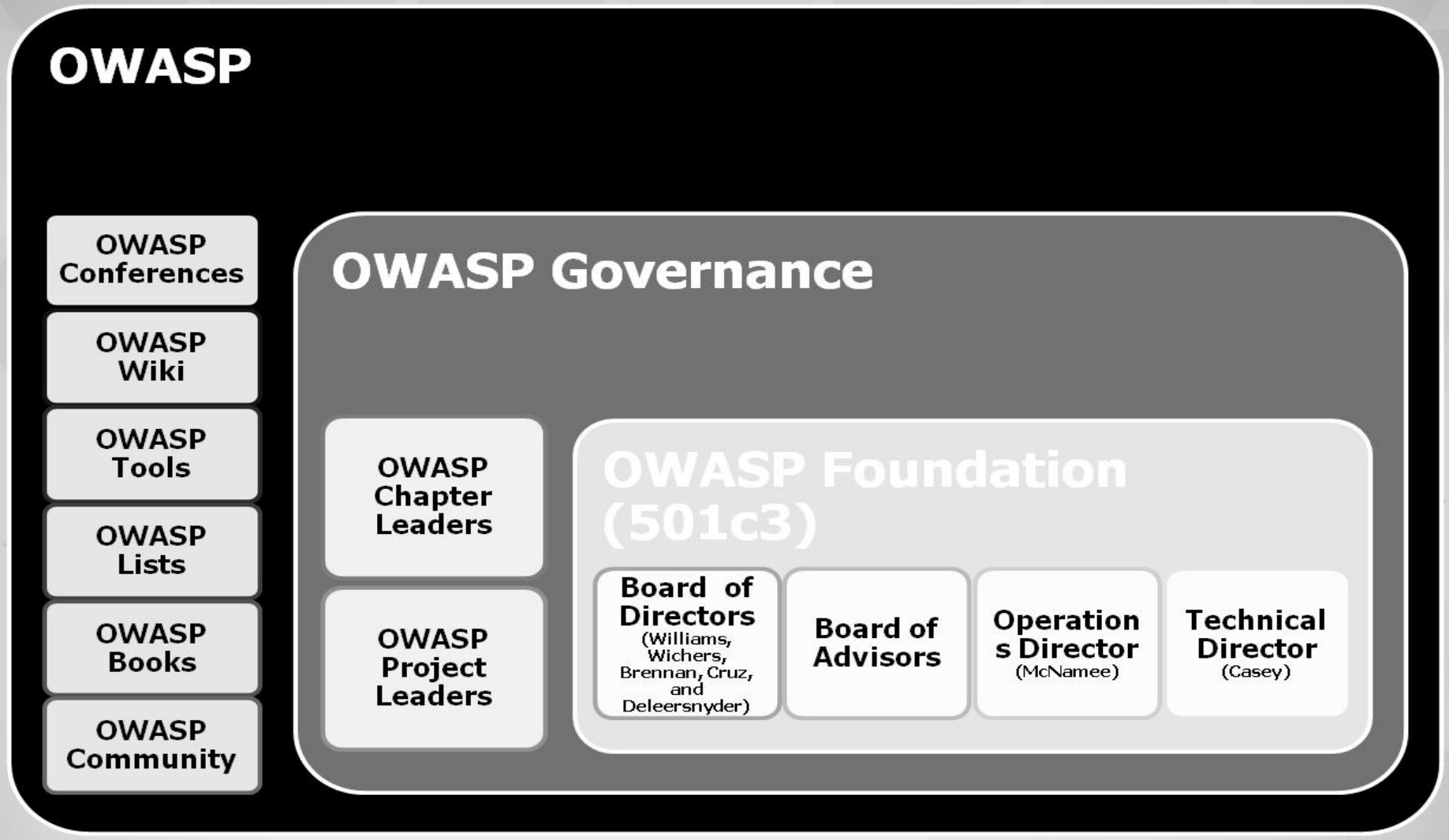
Sinergia da Comunidade

Voluntários:

- Compartilhamento de conhecimento
- Liderança de projetos e pessoas
- Apresentações em eventos
- Administração

Sustentado por:

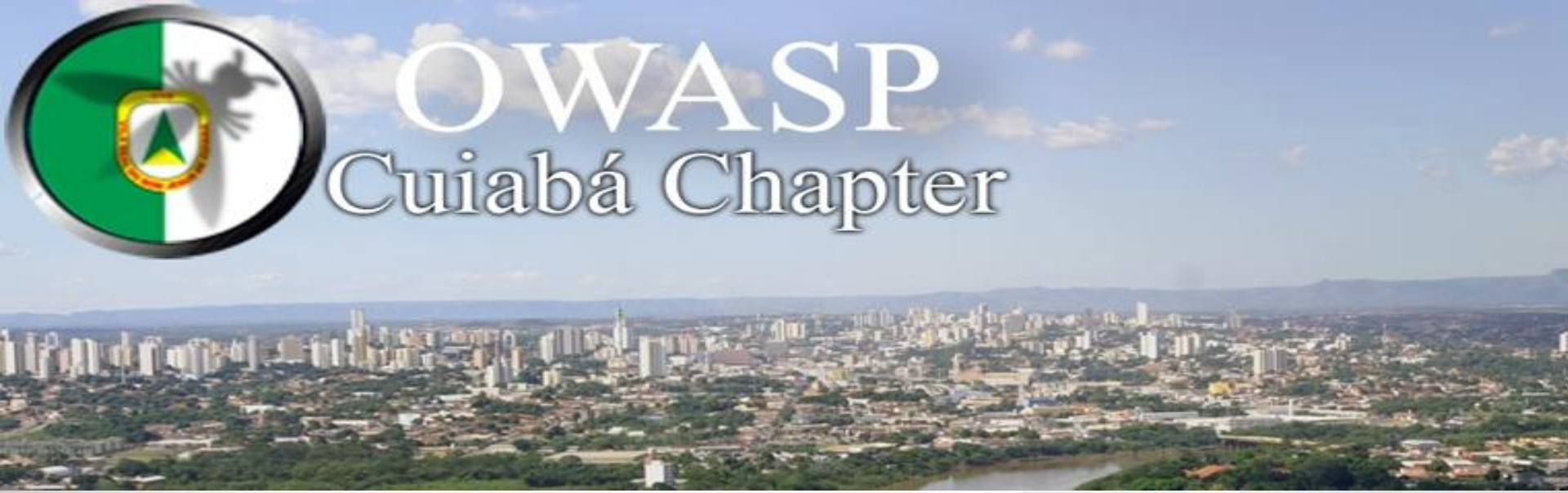
- Conferências
- Anuidades de membros
- Propagandas no site
- Patrocinadores Corporativos


OWASP
OWASP Conferences
OWASP Wiki
OWASP Tools
OWASP Lists
OWASP Books
OWASP Community
OWASP Governance
OWASP Chapter Leaders
OWASP Project Leaders
OWASP Foundation (501c3)
Board of Directors (Williams, Wichers, Brennan, Cruz, and Deleersnyder)
Board of Advisors
Operations Director (McNamee)
Technical Director (Casey)

OWASP GLOBAL COMMITTEES							
OWASP GLOBAL COMMITTEE	**Projects**	**Membership**	**Education**	**Conferences**	**Industry**	**Chapters**	**Connections**
Committee Chair	Jason Li	Dan Cornell	Martin Knobloch	Mark Bristow	Joe Bernik	Tin Zaw	Jim Manico
Members	▪ Brad Causey ▪ Chris Schmidt ▪ Justin Searle ▪ Larry Casey	▪ Michael Coates ▪ Tony UcedaVelez ▪ Ofer Maor ▪ Helen Gao	▪ Eduardo Neves ▪ Cecil Su ▪ Fabio Cerullo ▪ Kuai Hinjosa ▪ Sebastien Gioria ▪ Nishi Kumar	▪ Lucas Ferreira ▪ John Wilander ▪ Richard Greenberg ▪ Ralph Durkee ▪ Neil Matatall ▪ Cassio Goldschmidt	▪ Lorna Alamri ▪ Rex Booth ▪ David Campbell ▪ Alexander Fry ▪ Georg Hess ▪ Colin Watson ▪ Mauro Flores ▪ Mateo Martinez	▪ Andrew van der Stock ▪ Seba Deleersnyder ▪ Puneet Mehta ▪ Matthew Chalmers ▪ Mandeep Khera ▪ L. Gustavo C. Barbato	▪ Justin Clarke
Applicants	▪ Keith Turpin (pending confirmation)	▪ Mateo Martínez ▪ Aryavalli Gandhi	▪ Tony Gottlieb	▪ Benjamin (Ben) Tomhave ▪ Mohd Fazli Azran ▪ Zhendong Yu	▪ Jerry Hoff ▪ Sherif Koussa ▪ Michael Scovetta		▪ Jerry Hoff ▪ Doug Wilson
Committee Looking For	New Members with OWASP Project Leadership Experience	More Members	New Members with Education Background	More Members Outside U.S.	More Members Outside U.S. and Europe	More Members Outside U.S.	More Members

Centenas Capitulos Locais mas somente por volta de 80 estão ativos;
http://www.owasp.org/index.php/Category:Brasil

Belo Horizonte,
Brasília,
Campinas,
Cuiabá,
Curitiba,
Fortaleza,
Goiânia,
Maceió,
Manaus,
Natal,
Paraíba,
Porto Alegre,
Recife,
Rio de Janeiro,
São Luís,
São Paulo,
Vitoria,
Florianópolis.



-- **Administração**: Este núcleo tem como finalidade dar suporte necessário para realização das atividades de todos os outros GT's.

-- **Projetos**: Este núcelo tem como finalidade, Organizar a Execução de todas as atividades relacionadas a comunidade, com intuito de dar “norte” na execução dos projetos proposto pelos membros.

-- **Comunicação**: Este núcleo tem como finalidade Organizar a Execução de todas as Atividades relacionadas a Comunicação da Comunidade.

-- **E-learning**: Este núcleo tem como finalidade Organizar a Execução de todas as atividades relacionadas a educação/conscientização dos Membros com relação as novas tecnologias e demais atividades correlatas a área.

Atividades da Comunidade Capitulo cuiabá

CaguetaOS - Distribuição Forense criada por membros da comunidade, com intuito de auxiliar toda atividade a ser executada neste campo de pesquisa.

Palestras e Workshop's - Encontros em todas as Universidades de Cuiabá, onde buscamos disseminar a segurança da informação e segurança em aplicações web.

Cursos - Cursos das quais disponibilizamos gratuitamente a toda Comunidade de TI, visando a parte pratica da Segurança da informação utilizando conteudo da OWASP.

Tradução de Material - Realizamos tradução de Materiais oficiais da Owasp para língua portuguesa,tornando-as de facil acesso e entendimento a Comunidade de TI.

EX: OWASP_Snakes_and_Ladders - pt-br .

Patrocinadores Corporativos

http://www.owasp.org/index.php/Membership

Eu não entendo nada de Segurança da Informação , mas quero implementar estas melhorias da minha empresa ou aplicações, como faço?

Projetos Owasp

Tools [Reviewed February 2015]Health Check February 2016]

O-Saft
OWASP Dependency Track Project
OWASP EnDe Project
OWASP Hackademic Challenges Project
OWASP Mantra Security Framework
OWASP Mobile Security Project
OWASP O2 Platform
OWASP Passfault
OWASP Security Ninjas Appsec Training
OWASP Security Shepherd
OWASP WebGoat Project
OWASP Xenotix XSS Exploit Framework
OWASP Code Pulse Project

Documentation [In Progress-Results by February/March 2015]Health Check February 2016]

OWASP Application Security Guide For CISOs
OWASP Cheat Sheets Project
OWASP CISO Survey OWASP Code Review Guide
OWASP Codes of ConductOWASP Cornucopia
OWASP Development Guide
OWASP Podcast
OWASP Proactive Controls
OWASP Internet of Things Top Ten
OWASP Top 10 Privacy Risks
OWASP Reverse Engineering and Code Modification Prevention

Code [Reviewed March 2015 - Health Check February 2016]

OWASP Java Encoder Project
OWASP Java HTML Sanitizer Project
OWASP Node.js Goat Project
OWASP Security Logging Project
OWASP Mth3l3m3nt Framework Project
OWASP WebGoat PHP aProject
OWASP Secure Headers Project
OWASP Vicnum Project

Research
OWASP WASC Distributed Web Honeypots Project

Tools [Reviewed last: May 2015 - Health Check February 2016]

OWASP Benchmark
OWASP Wordpress Vulnerability Scanner
OWASP Threat Dragon
OWASP Security Knowledge Framework
OWASP Faux Bank Project
OWASP Droid

WAP Web Application_Protection
OWASP Mutillidae 2 Project
OWASP SeraphimDroid Project
OWASP WebSpa Project
OWASP Pyttacker Project
OWASP SonarQube Project
OWASP Rainbow Maker Project
OWASP ZSC Tool Project
OWASP DefectDojo Project
OWASP_Web Malware Scanner Project
OWASP Basic Expression Lexicon Variation Algorithms (Belva) Project
OWASP VBScan

Low Activity (LABS)[Reviewed July 2015] Health Check February 2016

* OWASP Broken Web Applications Project

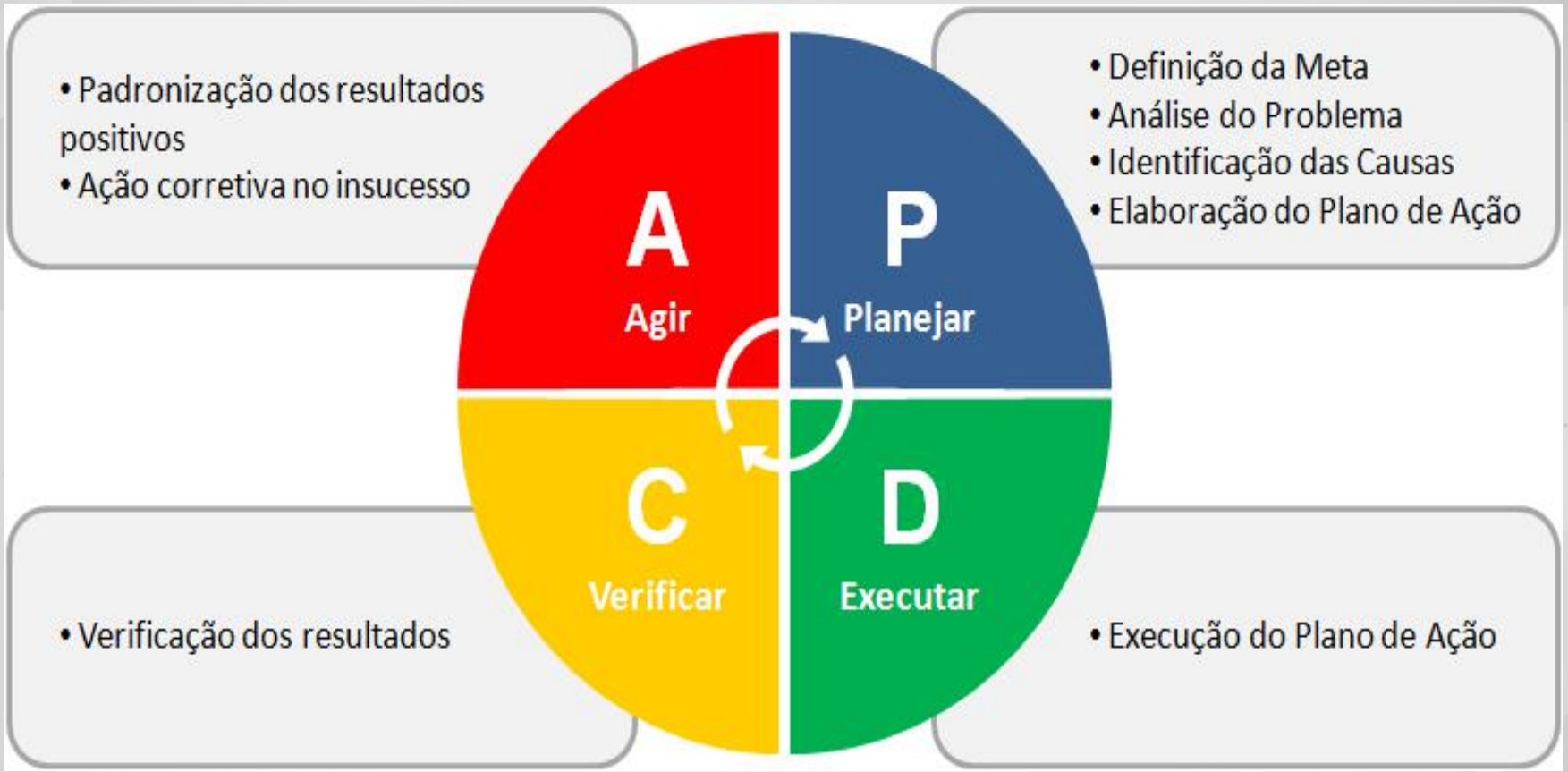
Tools Health Check February 2016

* WebScarab
* OWASP HTTP POST Tool

Documentation [Low Activity] Health Check February 2016

* OWASP AppSec Tutorial Series
* OWASP Legal Project
* Virtual Patching Best Practices
* OWASP Secure Coding Practices - Quick Reference Guide

PDCA


• Padronização dos resultados positivos
• Ação corretiva no insucesso
A
Agir
P
Planejar
• Definição da Meta
• Análise do Problema
• Identificação das Causas
• Elaboração do Plano de Ação
C
Verificar
D
Executar
• Verificação dos resultados
• Execução do Plano de Ação

"PDCA" utilizando OWASP?

OWASP SAMM: 4 Business functions

Define | **Design** | **Develop** | **Deploy** | **Maintain**

Governance

Software development management activities and organisation-wide business processes

Construction

Goal definition and software creation processes

Verification

Checking, evaluation and testing of software development artifacts

Deployment

Software release management and normal operational management

STRIDE

Uma categorização ameaça tais como STRIDE é útil na identificação de ameaças ao classificar metas atacante, tais como:

* **Spoofing**
* **Tampering (adulteração)**
* **Repudiation**
* **Information Disclosure**
* **Denial of Service**
* **Elevation of Privilege.**

1) Requisitos e Análise

Nesta fase os analistas consideram os requisitos e objetivos da aplicação, bem como os possíveis problemas.

A OWASP TOP 10 pode ser usado como um guia para a possíveis ataques; e examinando como os dez riscos podem afetar o seu software irá ajudá-lo a moldar o design do aplicativo para minimizar as ameaças mais críticas.

https://www.owasp.org/index.php/Application_Threat_Modeling

2) Arquitetura e Designer (SDLC)
Você pode seguir as diretrizes de design específicas que são comprovadas soluções para os Top 10 por exemplo ESAPI, especialmente aqueles que você identificou durante os Requisitos e fase de análise como sendo vulnerabilidades particulares.

https://www.owasp.org/index.php/Project_Information:_OWASP_Enterprise_Security_API_Project

3) Desenvolvimento
Na fase de desenvolvimento você pode adotar padrões de codificação seguras específicas que têm sido comprovada para se defender contra os OWASP Top 10 riscos.

A fase de desenvolvimento é também quando as revisões de código ocorrem normalmente; bem como revisar o código para garantir que ele tem as características e funções especificadas, os desenvolvedores devem ser treinados para procurar por vulnerabilidades no código relativas à OWASP Top 10.

https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide

4) Teste

Se você está ciente dos riscos de segurança mais comuns, durante a fase de testes, você pode garantir que os testes específicos são executados para simular ataques relacionados com a OWASP Top 10.

Além disso, ferramentas de análise estática, pode ler através de código de software programados para procurar pistas no código que podem apontar para vulnerabilidades - coisas que os desenvolvedores podem não ter pego durante suas revisões de código.

https://www.owasp.org/index.php/Category:OWASP_Testing_Project

5) Implantação

Software e sistemas de computadores que não estão configurados com segurança pode se expor ser alvo facil na Intetrnet. OWASP pode ser muito útil na fase de implantação do SDLC - ajudando a reduzir o risco através da verificação de erros de configuração e implantação física relativas ao Top 10.

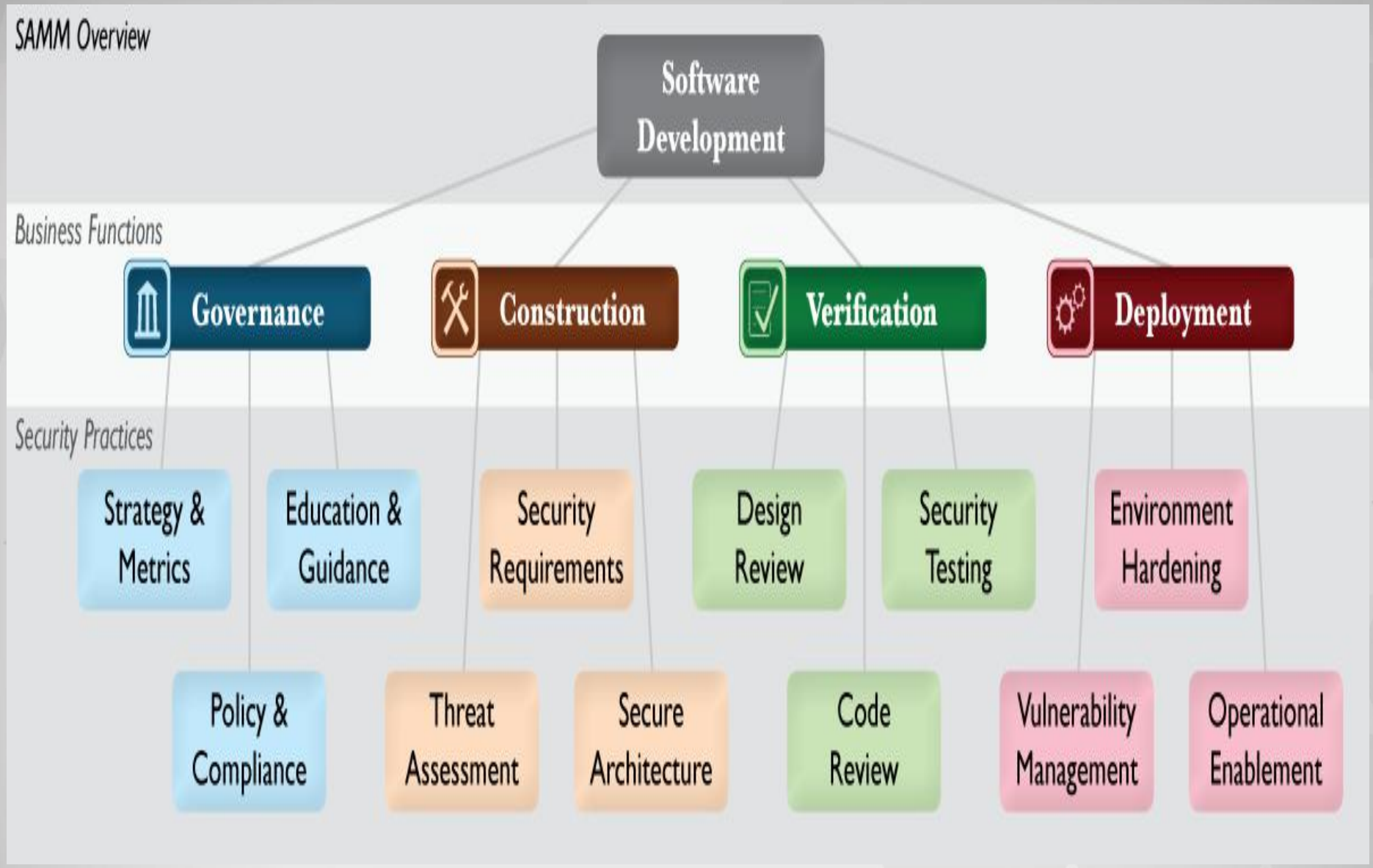
https://www.owasp.org/index.php/Security_Code_Review_in_the_SDLC

6) Manutenção

Centrando-se no OWASP Top 10 irá garantir que, mesmo depois que as organizações implementarem a metodologia, poderam descobrir se as brechas exploradas e resolvidas ainda continuam.

Este processo é tão importante quanto os outros, pois basicamente revisa todos os anteriores. :)

https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project


SAMM Overview
Software Development
Business Functions
Governance
Construction
Verification
Deployment
Security Practices
Strategy & Metrics
Education & Guidance
Security Requirements
Design Review
Security Testing
Environment Hardening
Policy & Compliance
Threat Assessment
Secure Architecture
Code Review
Vulnerability Management
Operational Enablement

Top 10 Mobile Security

M1 - Uso inadequado da plataforma
M2 - Armazenamento de dados inseguros
M3 - Comunicação insegura
M4 - Autenticação insegura
M5 - Criptografia insuficiente
M6 - Autorização insegura
M7 - Qualidade do Código do Cliente
M8 - Proibição de código
M9 - Engenharia reversa
M10 - Funcionalidade Extrangeira

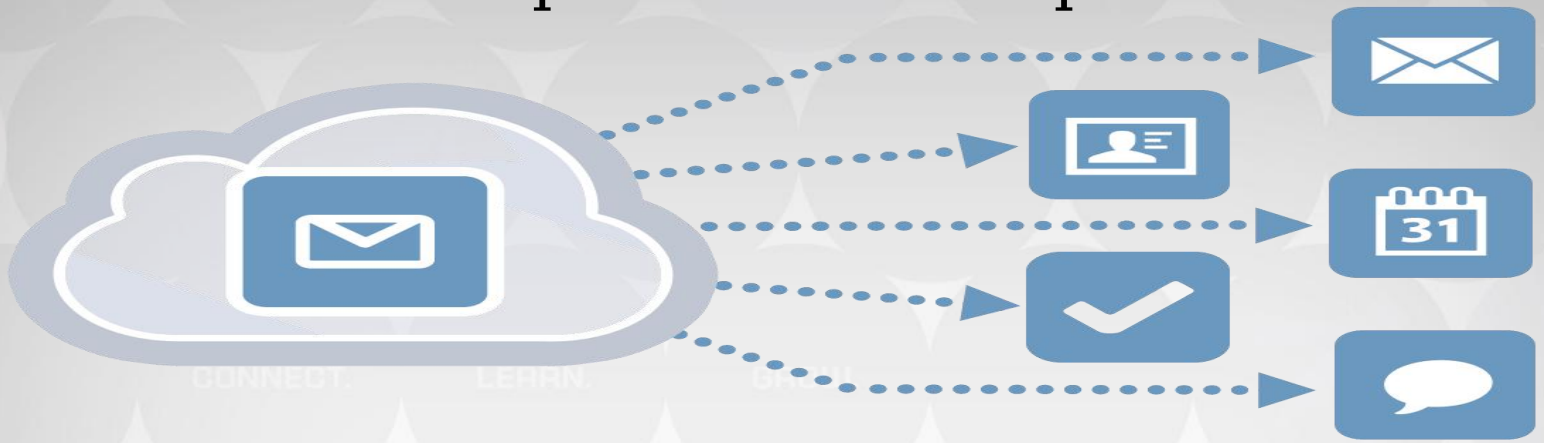
https://www.owasp.org/index.php/OWASP_Mobile_Security_Project

Top 10 Privacy Risks

P1 Vulnerabilidades de aplicativos da Web
P2 Vazamento de dados do lado do operador
P3 Resposta de violação de dados insuficiente
P4 Apagamento insuficiente de dados pessoais
P5 Políticas, termos e condições não transparentes
P6 Coleção de dados não necessários para o objetivo principal
P7 Compartilhamento de dados com terceiros
P8 Dados pessoais desatualizados
P9 Falta ou insuficiência de sessão insuficiente
P10 Transferência de dados inseguras

https://www.owasp.org/index.php/OWASP_Top_10_Privacy_Risks_Project

Top 10 Cloud Computer



R1 - Responsabilidade e propriedade de dados
R2 - Federação de Identidade do Usuário
R3 - Conformidade Regulatória
R4 - Continuidade do Negócio e Resiliência
R5 - Privacidade do usuário e uso secundário de dados
R6 - Integração de serviços e dados
R7 - Multi Tenancy e Segurança Física
R8 - Análise de Incidência e Suporte Forense
R9 - Segurança de infra-estrutura
R10 - Exposição ao ambiente não produtivo

https://www.owasp.org/index.php/Category:OWASP_Cloud_%E2%80%90_10_Project

OWASP TOP10

O objetivo do projeto OWASP Top 10 Controles Proativo é aumentar a consciência sobre a segurança do aplicativo, descrevendo as áreas mais importantes da preocupação de que os desenvolvedores de software deve estar ciente e aprender com os erros das outras organizações.

OWASP TOP10/2015/2016

A1- Verifique para a Segurança cedo e frequentemente
A2- Parametrizar consultas
A3- Dados Encode
A4- Validar Todas as Entradas
A5- Implementar controles de identidade e autenticação
A6- Implementar controles de acesso apropriados
A7- Proteger Dados
A8- Implementar Log e Detecção de Intrusão
A9- Estruturas de segurança de alavancagem e bibliotecas
A10- Erro e Tratamento de Exceções

Visão Geral do conceito Seguro de Ambientes.

Primeiro passo de todos é exatamente realizar planejamento da aplicação em todos os contextos.

0x01- Rede e Infraestrutura. (Ensaio de Intrusão Networking)
0x02- Servidor Hospedagem da Aplicação. (Pentesting S.O.)
0x03- Banco de Dados e Aplicação. (OWASP TOP 10)
0x04- Projeto de Gerenciamento Segurança da Informação.(ISO's)
0x05- Gerenciamento de "U.S.B." até perfis de Crackers. (SEING)

A**1- Verifique para a Segurança cedo e frequentemente**

A1-Injection
A2-Broken Authentication and Session Management
A3-Cross Site Scripting (XSS)
A4-Insecure Direct Object References
A5-Security Misconfiguration
A6-Sensitive Data Exposure
A7-Missing Function Level Access Control
A8-Cross-Site Request Forgery (CSRF)
A9-Using Components with Known Vulnerabilities
A10-Unvalidated Redirects and Forwards

A**2- Parametrizar Consultas**

As consultas parametrizadas são uma forma de alavancar a Data Access Abstraction Layer como os parâmetros são interpretados antes de executar uma consulta SQL.

RESUMO: Ele fornece proteção de injeção SQL.

Na comunidade você pode consultar varios exemplos de realização de parametrização de dados nas aplicações em diversas linguagens.

https://www.owasp.org/index.php/Query_Parameterization_Cheat_Sheet

Ataques a SGDB mais Conhecidos.

Posição	2013 - Principais Ameaças	2015 - Principais Ameças
1	Privilégios excessivos ou esquecidos.	Privilégios excessivos ou esquecidos.
2	Abuso de privilégio	Abuso de privilégio
3	SQL Injection	Input Injection
4	Malware	Malware
5	Auditoria fraca	Auditoria fraca
6	Exposição de mídia de storage	Exposição de mídia de storage
7	Exploração de vulnerabilidades e configurações fracas de banco de dados	Exploração de vulnerabilidades e configurações fracas de banco de dados
8	Dados sensíveis sem politicas de segurança.	Dados sensíveis sem politicas de segurança.
9	DoS - Negação de Serviço	DoS - Negação de Serviço
10	Pouca experiência dos profissionais na área de segurança.	Pouca experiência dos profissionais na área de segurança.

A3- Dados Encode

A codificação é um mecanismo poderoso para ajudar a proteger contra muitos tipos de ataques, especialmente ataques de injeção.

A Codificação é necessária para impedir diversas formas de injeção incluindo a injeção de comando (comando de codificação de Unix, o comando a codificação do Windows), a injeção de LDAP (codificação LDAP) e injeção XML (XML codificação).

Outro exemplo de codificação é a codificação de saída que é necessário para prevenir Cross Site Scripting (codificação HTML entidade, hex codificação JavaScript, etc).

Fonte: https://www.owasp.org/index.php/OWASP_Java_Encoder_Project

A4- Validar Todas as Entradas

A aplicação deve realizar o tratamento de todo input direcionado ao sistema, ou de instruções maliciosas direcionadas ao servidor.

Considere todas as entradas a partir do exterior da aplicação como não confiável. Para aplicações web que inclui cabeçalhos HTTP, cookies e GET e POST parâmetros: qualquer um ou todos esses dados podem ser manipulados por um invasor.

Ex:
A1-Injection.
A3-Cross Site Scripting (XSS).
A10-Unvalidated Redirects and Forwards.

https://www.owasp.org/index.php/OWASP_Mantra_-_Security_Framework

A5- Implementar controles de identidade e autenticação

A autenticação é o processo de verificar se um indivíduo ou uma entidade é quem afirma ser, enquanto o gerenciamento de identidade é um tópico mais amplo que inclui não só autenticação, gerenciamento de sessão, mas também abrange tópicos avançados como a federação de identidade, single sign on, password-management ferramentas, repositórios de identidade etc..

Fonte: A2-Broken Authentication and Session Management
https://www.owasp.org/index.php/Authentication_Cheat_Sheet

Multifator de Autenticação

A autenticação multifator garante que os usuários são quem dizem ser, obrigando-os a identificar-se com uma combinação de:

- algo que eles sabem - senha ou PIN
- Algo que eles próprios - símbolo ou telefone
- Algo que eles são - biometria, como uma impressão digital

Mobile Application: Autenticação TokenBased

Quando a construção de aplicações móveis, é recomendado evitar o armazenamento / autenticação persistindo credenciais localmente no **dispositivo**. Em vez disso, executar a autenticação inicial usando o nome de usuário e senha fornecida pelo usuário e, em seguida, gerar um token de acesso de curta duração que pode ser usado para autenticar uma solicitação do cliente sem enviar as credenciais do usuário.

https://www.owasp.org/index.php/Authentication_Cheat_Sheet

Implementar o armazenamento de senha de segurança

A fim de proporcionar controles de autenticação forte, um aplicativo deve armazenar de forma segura as credenciais do usuário. Além disso, controles de criptografia deve ser posto em prática de forma que se uma credencial (por exemplo, uma senha) está comprometido, o atacante não tem imediatamente acesso a esta informação.

Implementar Mecanismo de Recuperação de senha de segurança

É comum para um aplicativo para ter um mecanismo para um usuário para ter acesso a sua conta em caso eles esquecem sua senha. Um fluxo de trabalho de design bom para um recurso de recuperação de senha usará elementos de autenticação multifatoriais (por exemplo, fazer a pergunta de segurança algo que eles sabem, e em seguida, enviar um sinal gerado em um dispositivo de algo que eles próprios)

Sessão: Geração e Expiração

Em qualquer autenticação bem sucedida e uma nova autenticação do software deve gerar uma nova sessão e sessão id. A fim de minimizar o período de tempo que um invasor pode lançar ataques sobre as sessões ativas e sequestrar-los, é obrigatório definir o tempo limite de expiração para cada sessão, após um determinado período de inatividade. O comprimento de tempo limite deve ser inversamente proporcional com o valor dos dados protegidos.

Exigir Reauthentication para recursos sensíveis

Para transações sensíveis, como a mudança de senha ou alterar o endereço de envio para uma compra, é importante exigir que o usuário se autenticar novamente e, se possível, para gerar um novo ID de sessão após a autenticação bem-sucedida.

Fontes:

https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management
https://www.owasp.org/index.php/Mobile_Top_10_2014-M5
https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet
https://www.owasp.org/index.php/Session_Management_Cheat_Sheet
https://www.owasp.org/index.php/IOS_Developer_Cheat_Sheet

6- Implementar controles de acesso apropriados

Autorização (Controle de Acesso) é o processo em que os pedidos de acesso a um determinado recurso ou recurso deve ser concedido ou negado. Os seguintes requisitos de projeto de controle de acesso "positivos" deve ser considerada nas fases iniciais de desenvolvimento de aplicações:

* Forçar todos os pedidos que passar por verificações de controle de acesso.
* Negar por padrão.
* Evite verificações de controle de acesso baseado em políticas codificados em código.
* Verifique no servidor quando cada função é acessado.

Fontes:
* https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References
* https://www.owasp.org/index.php/Top_10_2013-A7-Missing_Function_Level_Access_Control

A7- **Proteger Dados**

A criptografia de dados em Transporte
Ao transmitir dados sensíveis, em qualquer nível da sua aplicação ou arquitetura de rede, encryptionintransit de algum tipo deve ser considerada.

TLS é de longe o modelo mais comum e amplamente apoiada usado por aplicações web para criptografia em trânsito. Apesar fraquezas publicados em implementações específicas (por exemplo heartbleed - http://heartbleed.com/), ainda é o de fato o método recomendado para implementar a criptografia de camada de transporte.

Fonte: *https://tools.ietf.org/html/rfc5246*
https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet
https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet
https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet
https://www.owasp.org/index.php/Testing_for_Weak_SSL/TLS_Ciphers,_Insufficient_Transport_Layer_Protection_(OTG-CRYPST-001)

A8- Implementar Log e Detecção de Intrusão

Logging de Aplicação não deve ser uma reflexão tardia ou limitado a depuração e solução de problemas.

O Log é também usado em outras actividades importantes como:

- Monitoramento de aplicativos.
- A analítica de negócios e visão.
- auditoria de Actividade e controle do cumprimento.
- Sistema de detecção de intrusão.
- Forensics.

Fontes:

https://www.owasp.org/index.php/Logging_Cheat_Sheet
https://www.owasp.org/index.php/IOS_Developer_Cheat_Sheet#Sensitive_Information_Disclosure_.28M10.29
https://www.owasp.org/index.php/Error_Handling,_Auditing_and_Logging

A9- Estruturas de segurança de alavancagem e bibliotecas

Utilizar de bibliotecas que possam auxiliar na construção de arquiteturas de software mais seguras , criando-se uma camada a mais de segurança dos sistemas utilizando recursos de alavancagem em Sistemas Operacionais ou aplicação.

Fonte:
http://projects.spring.io/spring-security/
http://shiro.apache.org/
https://docs.djangoproject.com/en/1.8/topics/security/
https://pythonhosted.org/Flask-Security/

A10- Erro e Tratamento de Exceções

Muitas vezes, durante um teste de penetração em aplicações web, nos deparamos com muitos códigos de erro gerados a partir de aplicativos ou servidores web. É possível fazer com que esses erros para ser exibido ao usar uma determinada pedidos, seja especialmente criados com ferramentas ou criados manualmente.

Estes códigos são muito úteis à penetração testadores durante as suas actividades, porque revelam um monte de informações sobre bancos de dados, bugs e outros componentes tecnológicos directamente relacionadas com as aplicações web.

Fontes:
https://www.owasp.org/index.php/Testing_for_Stack_Traces_(OWASP-IG-XXX)
https://www.owasp.org/index.php/Error_Handling
https://www.owasp.org/index.php/Testing_for_Error_Code_(OWASP-IG-006)
https://github.com/diy1/aspirator

Alguns Exemplos Como:

- Web Server Errors: Not Found
The requested URL /page.html was not found on this server.
Apache/2.2.3 (Unix) mod_ssl/2.2.3 OpenSSL/0.9.7g DAV/2 PHP/5.1.2 Server at localhost Port 80

Fonte: https://www.owasp.org/index.php/Fingerprint_Web_Server_(OTG-INFO-002)

ASP-net Error's versões e Caminhos

https://www.saotn.org/net-framework-4-6-allows-side-loading-windows-api-set-dll/
https://www.securify.nl/advisory/SFY20160201/_net_framework_4_6_allows_side_loading_of_windows_api_set_dll.html

SEFAZ
SECRETARIA DE
ESTADO DE FAZENDA

GOVERNO DE
MATO
GROSSO

CONNECT. LEARN. GROW.

Sefaz - MT

SEFAZ
SECRETARIA DE
ESTADO DE FAZENDA

GOVERNO DE
MATO
GROSSO

Visualização de Dominios:
sefaz.mt.gov.br
nfce.sefaz.mt.gov.br
cte.sefaz.mt.gov.br
nfe.sefaz.mt.gov.br
homologacao.sefaz.mt.gov.br

Conectividade:
https://www.shodan.io/search?query=sefaz.mt.gov.br
https://www.censys.io/ipv4?q=sefaz.mt.gov.br

Desenvolvimento:
https://www.sefaz.mt.gov.br/javascript/miscelanea/BrowserType.js?

é importante ressaltar que todos as vulnerabilidades encontradas foram **Corrigidos** pelas equipes de:
Desenvolvimento, Infraestrutura de Redes e Gerencia de Riscos e Segurança.

Vide documentação: pentest/

SEFAZ
SECRETARIA DE
ESTADO DE FAZENDA

GOVERNO DE
MATO
GROSSO

CONNECT. LEARN. GROW.

Obrigado!