ANEXO 18

PADRÕES DE SEGURANÇA NO DESENVOLVIMENTO DE SISTEMAS PROCERGS

Este anexo tem o objetivo de descrever os padrões de segurança mínimos dos sistemas desenvolvidos e mantidos pela PROCERGS.

1. O desenvolvimento de softwares na PROCERGS deve estar aderente a Política de Segurança da Informação da PROCERGS, mais especificamente com as diretrizes do item 6.16 que trata do desenvolvimento seguro de softwares. Na qual, são tratadas questões como: o uso de repositórios oficiais, a segregação de ambientes de desenvolvimento, entre outros.

2. Riscos de Segurança em Aplicações Web

Os sistemas desenvolvidos para a PROCERGS e seus clientes devem ter o máximo cuidado contra os seguintes riscos de segurança em aplicações Web baseados na compilação elaborado pelo projeto *OWASP* com as 10 principais vulnerabilidades (https://owasp.org/Top10).

A seguir a lista do OWASP Top 10 - 2021:

2.1. A01:2021 - Broken Access Control (Quebra de Controle de Acesso):
O controle de acesso impõe uma política de modo que os usuários não possam agir fora de suas permissões pretendidas. As falhas normalmente levam à divulgação, modificação ou destruição de informações não autorizadas de todos os dados ou ao desempenho de uma função comercial fora dos limites do usuário.

2.2. A02:2021 - Cryptographic Failures (Falhas Criptográficas):
A primeira coisa é determinar as necessidades de proteção dos dados em trânsito e em repouso. Por exemplo, senhas, números de cartão de crédito, registros de saúde, informações pessoais e segredos comerciais exigem proteção extra, principalmente se esses dados se enquadrarem nas leis de privacidade, por exemplo, Lei Geral de Proteção de Dados Pessoais (LGPD) e Regulamento Geral de Proteção de Dados da UE (GDPR) ou regulamentos, por exemplo, proteção de dados financeiros como PCI Data Security Standard (PCI DSS).

2.3. A03:2021 - Injection (Injeção):
As falhas de injeção, tais como injeção de SQL, de OS e de LDAP, ocorrem quando dados não confiáveis são enviados para um interpretador como parte de um comando ou consulta. Os dados manipulados pelo atacante podem iludir o interpretador para que este execute comandos indesejados ou permita o acesso a dados não autorizados.

2.4. A04:2021 - Insecure Design (Design Inseguro):
O design inseguro é uma categoria ampla que representa diferentes pontos fracos, expressos como "design de controle ausente ou ineficaz". O design inseguro não é a fonte de todas as outras 10 categorias de risco principais. Há uma diferença entre design inseguro e implementação insegura. Nós diferenciamos entre falhas de design e defeitos de implementação por um motivo, eles têm

PROCERGS – CENTRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO DO ESTADO DO RIO GRANDE DO SUL S/A
Praça dos Açorianos, s/nº – Centro Histórico – CEP 90.010-340 – Porto Alegre/RS – Brasil
PABX + 55 51 3210-3100 – http://www.procergs.rs.gov.br – procergs@procergs.rs.gov.br

diferentes causas raiz e remediação. Um design seguro ainda pode ter defeitos de implementação que levam a vulnerabilidades que podem ser exploradas. Um design inseguro não pode ser corrigido por uma implementação perfeita, pois, por definição, os controles de segurança necessários nunca foram criados para a defesa contra ataques específicos. Um dos fatores que contribuem para um design inseguro é a falta de perfis de risco de negócios inerentes ao software ou sistema que está sendo desenvolvido e, portanto, a falha em determinar o nível de design de segurança necessário.

2.5. A05:2021 - Security Misconfiguration (Configuração Incorreta de Segurança):
Uma aplicação pode estar vulnerável por falta de proteção de segurança apropriada em qualquer parte da pilha de aplicativos ou permissões configuradas incorretamente em serviços na nuvem, ou por ter recursos desnecessários ativados ou instalados (por exemplo, portas, serviços, páginas, contas ou privilégios desnecessários), ou por estar com a conta padrão, usuário e senha inalterados. Também o tratamento de erros revela rastreamentos de pilha ou outras mensagens de erro excessivamente informativas aos usuários, e para sistemas atualizados, os recursos de segurança mais recentes estão desabilitados ou não estão configurados com segurança. As configurações de segurança nos servidores de aplicativos, estruturas de aplicativos (por exemplo, Struts, Spring, ASP.NET), bibliotecas, bancos de dados, etc., não são definidas para proteger os valores, e o servidor não envia cabeçalhos ou diretivas de segurança, ou eles não estão configurados para proteger os valores, muitas vezes com o software desatualizado ou vulnerável.

2.6. A06:2021 - Vulnerable and Outdated Components (Componentes Vulneráveis e Desatualizados):
Componentes tais como, bibliotecas, frameworks e outros módulos de software, são executados com os mesmos privilégios que a aplicação. O comprometimento de um componente vulnerável pode provocar uma grande perda de dados ou do controle completo de um servidor. Aplicações e APIs que utilizam componentes com vulnerabilidades conhecidas podem enfraquecer as defesas da aplicação possibilitando ataques e impactos diversos.

2.7. A07:2021 - Identification and Authentication Failures (Falhas de Identificação e Autenticação):
A confirmação da identidade do usuário, autenticação e gerenciamento de sessão é fundamental para proteção contra ataques relacionados à autenticação. As funções da aplicação que estão relacionadas com a autenticação e gestão de sessões são muitas vezes implementadas incorretamente, permitindo que um atacante possa comprometer senhas, chaves, tokens de sessão, ou explorar outras falhas da implementação que lhe permitam assumir a identidade de outros usuários (temporária ou permanentemente).

2.8. A08:2021 - Software and Data Integrity Failures (Falhas de software e integridade de dados):
As falhas de software e integridade de dados estão relacionadas ao código e à infraestrutura que não protegem contra violações de integridade. Um exemplo disso é quando um aplicativo depende de plug-ins, bibliotecas ou módulos de fontes não confiáveis, repositórios, e redes de entrega de conteúdo (CDNs). Um pipeline de CI/CD inseguro pode apresentar o potencial de acesso não autorizado, código malicioso ou comprometimento do sistema. Por último, muitos aplicativos agora incluem a funcionalidade de atualização automática, em que as atualizações são baixadas sem verificação de integridade suficiente e aplicadas ao aplicativo anteriormente confiável. Os invasores podem potencialmente carregar suas próprias atualizações para serem distribuídas e executadas em todas as instalações. Outro exemplo é onde objetos ou dados são codificados ou serializados em uma estrutura que um invasor pode ver e modificar sendo vulnerável à desserialização insegura.

PROCERGS – CENTRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO DO ESTADO DO RIO GRANDE DO SUL S/A
Praça dos Açorianos, s/nº – Centro Histórico – CEP 90.010-340 – Porto Alegre/RS – Brasil
PABX + 55 51 3210-3100 – http://www.procergs.rs.gov.br – procergs@procergs.rs.gov.br

2.9. A09:2021 - Security Logging and Monitoring Failures (Falhas de Registro e Monitoramento de Segurança):
Voltando ao OWASP Top 10 - 2021, esta categoria serve para ajudar a detectar, escalar e responder às violações ativas. Sem registro e monitoramento, as violações não podem ser detectadas. Registro, detecção, monitoramento e resposta ativa insuficientes ocorrem a qualquer momento, e é passível de se ficar vulnerável ao vazamento de informações, tornando os eventos de registro e alerta visíveis para um usuário ou invasor.

2.10. A10:2021 - Server-Side Request Forgery (SSRF) (Falsificação de solicitação do lado do servidor):
As falhas de SSRF ocorrem sempre que um aplicativo da web busca um recurso remoto sem validar a URL fornecida pelo usuário. Ele permite que um invasor force o aplicativo a enviar uma solicitação criada para um destino inesperado, mesmo quando protegido por um firewall, VPN ou outro tipo de lista de controle de acesso à rede (ACL). Como os aplicativos da web modernos fornecem aos usuários finais recursos convenientes, buscar uma URL se torna um cenário comum. Como resultado, a incidência de SSRF está aumentando. Além disso, a gravidade do SSRF está se tornando mais alta devido aos serviços em nuvem e à complexidade das arquiteturas.

Nota: os Riscos de Segurança em Aplicações Web não se limitam somente ao Top 10.

3. Checklist de validação de segurança no desenvolvimento:

3.1. Validar a entrada de dados.
De preferência filtrar, validar e limitar os dados que são inseridos pelo usuário.
3.2. Atentar para *warnings* no *build*.
Pode gerar uma vulnerabilidade como o *buffer Overflow,* por exemplo.
3.3. Manter o código mais simples possível.
Facilita a leitura e a detecção das ameaças.
3.4. Por padrão, negar.
No caso de controles de acesso às páginas, botões, campos, etc., é uma boa prática negar o acesso ou ocultar.
3.5. Aderir ao princípio do “menor privilégio”.
3.6. Sanitizar dados enviados para outros sistemas.
Evitar o envio de dados desnecessários ou que contenham informações sensíveis para integrações.
3.7. Funções devem ser intrinsecamente seguras.
Priorizar a verificação de segurança na menor porção de código possível.
3.8. Verificar códigos de erros retornados por funções ou métodos.
3.9. Atentar para os limites dos tipos usados na codificação.
Erros nas conversões e *castings* podem gerar vulnerabilidades.
3.10. Documentar o código.
Auxilia na depuração e localização de possíveis erros e não conformidades.
3.11. Sempre construir conforme especificação.

PROCERGS – CENTRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO DO ESTADO DO RIO GRANDE DO SUL S/A
Praça dos Açorianos, s/nº – Centro Histórico – CEP 90.010-340 – Porto Alegre/RS – Brasil
PABX + 55 51 3210-3100 – http://www.procergs.rs.gov.br – procergs@procergs.rs.gov.br

PADRÕES DE SEGURANÇA NO DESENVOLVIMENTO DE SISTEMAS MOBILE PROCERGS

Esta seção tem o objetivo de descrever os padrões de segurança mínimos dos sistemas mobile desenvolvidos e mantidos pela PROCERGS.

4. Riscos de Segurança em Aplicações Mobile

Os sistemas mobile desenvolvidos para a PROCERGS e seus clientes devem ter o máximo cuidado contra os seguintes riscos de segurança em aplicações mobile baseados na compilação elaborado pelo projeto OWASP Mobile com as 10 principais vulnerabilidades (https://owasp.org/www-project-mobile-top-10/) **A seguir a lista do OWASP Mobile Top 10 - 2016:**

4.1. M1:2016 - Improper Platform Usage (Uso Inapropriado da Plataforma):

Esta categoria abrange o uso indevido de um recurso da plataforma ou a falha na utilização dos controles de segurança da plataforma. Pode incluir intenções do *Android*, permissões da plataforma, uso incorreto do *TouchID*, *Keychain* ou algum outro controle de segurança que faça parte do sistema operacional móvel.

4.2. M2:2016 - Insecure Data Storage (Armazenamento de Dado Inseguro):

Os agentes dessa ameaça incluem: um atacante que obteve um dispositivo móvel perdido / roubado, *malwares* ou outro aplicativo “reempacotado” que executa no dispositivo móvel, agindo em nome do atacante.

4.3. M3:2016 - Insecure Communication (Comunicação Insegura):

Essa categoria abrange um *handshake* ruim, versões incorretas do SSL, negociação fraca, comunicação em texto claro de ativos sensíveis, etc. Ao projetar um aplicativo móvel os dados geralmente são trocados num modelo cliente-servidor. Quando a solução transmite seus dados, ela deve percorrer a rede da operadora do dispositivo móvel e a Internet. Os agentes de ameaça podem explorar vulnerabilidades para interceptar dados confidenciais enquanto estão em trânsito. Os agentes de ameaça mais comuns são atacantes que compartilham sua rede local (Wi-Fi monitorada ou comprometida), dispositivos de rede ou portadoras (roteadores, torres de celulares, *proxy*, etc), ou *malwares* em seu dispositivo móvel.

4.4. M4:2016 - Insecure Authentication (Autenticação Insegura):

Essa categoria abrange as noções de autenticação do usuário final ou o gerenciamento incorreto de sessão, o que pode incluir falha ao identificar o usuário quando for necessário, falha na manutenção da identidade do usuário quando necessário e fraquezas no gerenciamento de sessões. Os agentes de ameaças que exploram vulnerabilidades de autenticação normalmente fazem isso através de ataques automatizados que usam ferramentas disponíveis ou customizadas.

4.5. M5:2016 - Insufficient Cryptography (Criptografia Insuficiente):

O código deve aplicar uma criptografia adequada a um ativo de informação confidencial. Esta categoria aplica-se a problemas decorrentes de criptografia implementada incorretamente. Os agentes de ameaça são: qualquer pessoa com acesso físico a dados que foram criptografados de forma incorreta ou *malware* móvel agindo em nome do atacante.

4.6. M6:2016 - Insecure Authorization (Autorização Insegura):

Esta é uma categoria para tratar quaisquer falhas na autorização como decisões de autorização no

PROCERGS – CENTRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO DO ESTADO DO RIO GRANDE DO SUL S/A
Praça dos Açorianos, s/nº – Centro Histórico – CEP 90.010-340 – Porto Alegre/RS – Brasil
PABX + 55 51 3210-3100 – http://www.procergs.rs.gov.br – procergs@procergs.rs.gov.br

lado do cliente ou navegação forçada, e se distingue dos problemas de autenticação como identificação do usuário. Se o aplicativo não autenticar o usuário numa situação em que deveria (como conceder acesso anônimo a algum recurso ou serviço quando o acesso autenticado e autorizado é necessário), isso é uma falha de autenticação e não de autorização. Os agentes de ameaça que exploram as vulnerabilidades de autorização normalmente o fazem através de ataques automatizados que usam as ferramentas disponíveis ou personalizadas.

4.7. M7:2016 - Poor Code Quality (Código de Baixa Qualidade):

Esta categoria busca capturar *buffer overflows*, vulnerabilidades no formato de *strings* e vários outros erros de nível de código, onde a solução é reescrever algum código que esteja sendo executado no dispositivo móvel. Os agentes de ameaça incluem entidades que possam passar entradas não confiáveis para chamadas de método feitas no código móvel. Estes não são necessariamente problemas de segurança em si, mas levam às vulnerabilidades de segurança. Problemas de baixa qualidade de código são normalmente explorados por *malware* ou golpes de *phishing*.

4.8. M8:2016 - Code Tampering (Adulteração de Código):

Esta categoria cobre *patching* binário, modificação de recurso local, *hooking* de método, swizzling de método e modificação de memória dinâmica. Uma vez que o aplicativo é entregue ao dispositivo móvel, o código e os recursos de dados ficam lá residentes. Um invasor pode modificar diretamente o código, alterar o conteúdo da memória dinamicamente, alterar ou substituir as APIs do sistema que o aplicativo usa ou modificar os dados e os recursos do aplicativo. Isso pode fornecer ao atacante um método direto para subverter o uso pretendido do software para ganhos pessoais ou monetários. Normalmente, um invasor irá explorar a adulteração de código através de aplicativos mal-intencionados hospedados em lojas de aplicativos de terceiros. O invasor também pode enganar o usuário para que instale o aplicativo utilizando ataques de *phishing*.

4.9. M9:2016 - Reverse Engineering (Engenharia Reversa):

Um invasor normalmente baixa o aplicativo de destino de uma loja de aplicativos e o analisa em seu próprio ambiente local usando um conjunto de ferramentas diferentes. Esta categoria inclui a análise do binário principal para determinar seu código fonte, bibliotecas, algoritmos e recursos incorporados ao aplicativo. Softwares como o IDA Pro, Hopper, otool, strings e outras ferramentas de inspeção binária fornecem ao invasor informações sobre o funcionamento interno da aplicação. Isso pode ser usado para explorar outras vulnerabilidades nocivas no aplicativo, bem como revelar informações sobre servidores de *back-end*, cifras, constantes criptográficas e propriedade intelectual.

4.10. M10:2016 - Extraneous Functionality (Funcionalidade Estranha):

Muitas vezes, os desenvolvedores incluem funcionalidades de *backdoor* ou outros controles internos de segurança de desenvolvimento que não se destinam a serem lançados em um ambiente de produção. Por exemplo, um desenvolvedor pode acidentalmente incluir uma senha como um comentário em um aplicativo híbrido. Outro exemplo inclui a desativação da autenticação de dois fatores durante os testes. Normalmente, um invasor procura por funcionalidades estranhas dentro de um aplicativo móvel para descobrir alguma funcionalidade oculta nos sistemas de *back-end*. O invasor normalmente irá explorar essas funcionalidades estranhas diretamente de seus próprios sistemas sem qualquer envolvimento de usuários finais.

Nota: os Riscos de Segurança em Aplicações Mobile não se limitam somente ao Top 10.

PROCERGS – CENTRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO DO ESTADO DO RIO GRANDE DO SUL S/A
Praça dos Açorianos, s/nº – Centro Histórico – CEP 90.010-340 – Porto Alegre/RS – Brasil
PABX + 55 51 3210-3100 – http://www.procergs.rs.gov.br – procergs@procergs.rs.gov.br

5. Os 10 principais controles e princípios de segurança mobile que precisam ser respeitados:

5.1. Identificar e proteger dados sensíveis em dispositivos móveis.
5.2. Gerenciar credenciais de forma segura no dispositivo.
5.3. Assegurar que as informações sensíveis sejam protegidas em trânsito.
5.4. Implementar a autenticação do usuário, autorização e gerenciamento de sessão corretamente.
5.5. Manter as APIs de *Back-end* (serviços) e a plataforma (servidor) seguras.
5.6. Integração segura de dados com serviços e aplicativos de terceiros.
5.7. Atenção especial à coleta e armazenamento do consentimento para a coleta e uso dos dados do usuário.
5.8. Implementar controles para impedir o acesso não autorizado a recursos pagos (carteira, SMS, chamadas telefônicas, etc.).
5.9. Assegurar o provisionamento e a distribuição segura de aplicações móveis.
5.10. Verificar cuidadosamente qualquer erro de interpretação de código em tempo de execução.

6. Referências:

Política de Segurança da Informação da PROCERGS
https://owasp.org/Top10
https://owasp.org/www-project-top-ten/
https://owasp.org/www-project-mobile-top-10/
https://owasp.org/www-project-mobile-security/

1- https://www.owasp.org/index.php/Category:OWASP_Java_Project
2-http://www.asp.net/web-api/overview/security/preventing-cross-site-request-forgery-%28csrf%29-attacks
3 -http://msdn.microsoft.com/en-us/library/ms972969.aspx
5 - https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Top_10_Mobile_Controls

PROCERGS – CENTRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO DO ESTADO DO RIO GRANDE DO SUL S/A
Praça dos Açorianos, s/nº – Centro Histórico – CEP 90.010-340 – Porto Alegre/RS – Brasil
PABX + 55 51 3210-3100 – http://www.procergs.rs.gov.br – procergs@procergs.rs.gov.br