Sumário

Histórico					
Versão	Elaborado por	Data	Revisado por	Data	Principais Alterações
1.0	Edenilton Soares - ejsoares	19.04.2021	Maria das Graças - mgsmfernandes		

Objetivo

Prover aos membros do CDS um framework para avaliar o nível de requerimentos de segurança que deve ser utilizado e implementado, garantindo assim uma maior segurança.

Introdução

A implementação de uma metodologia que possa auxiliar no desenvolvimento seguro é um dos principais pontos dentro de um processo de Desenvolvimento Seguro. Os requisitos apresentados neste documento foram definidos com base nas especificações propostas pelo Open Web Application Security Project® (OWASP)[1], mais precisamente, pelo projeto da OWASP Application Security Verification Standard (ASVS) desenvolvido para assegurar parâmetros para cuidar da qualidade e da segurança da aplicação durante seu desenvolvimento.

A utilização do ASVS proporciona ao desenvolvedor a possibilidade de criar uma estrutura de avaliação de controles de segurança de aplicações web para, desta forma, atender aos requisitos necessários para proteção das informações que serão tratadas pela aplicação.

Níveis de Implementação

O ASVS apresenta três níveis de verificação que devem ser observados durante o processo de desenvolvimento de aplicações web. Cada um dos níveis aqui apresentados tem sua particularidade e profundidade.

Nível 1 - Segurança Mínima Necessária	Nível 2 - Padrão para web	Nível 3 - Alto Valor, Garantia, Segurança e Resiliência
Para níveis baixos de garantia de segurança. Apresenta o nível mínimo de segurança esperado para qualquer aplicação web.	Para aplicações que contêm dados sensíveis. É o nível recomendado para a maioria das aplicações.	Para as aplicações mais críticas, que requerem o mais alto nível de confiança.

[1] OWASP é uma fundação sem fins lucrativos que trabalha para melhorar a segurança do software com o suporte da comunidade formada por desenvolvedores, profissionais de segurança e corporações interessadas em promover um ambiente mais seguro para todos na web.

Nível 1 - Segurança mínima

Para se adequar a este nível, a aplicação não deve possuir vulnerabilidades de fácil descoberta e exploração, bem como as listadas no TOP 10 da OWASP ou mesmo em outras listas de vulnerabilidades similares classificadas como críticos e altos.

O nível 1 é considerado o mínimo de segurança para qualquer aplicação, no entanto, se os dados que devem ser protegidos por sua aplicação são críticos, raramente somente o Nível 1 trará o nível de proteção recomendado de segurança.

Nível 2 – Segurança Recomendada

Uma aplicação atinge o nível 2 do ASVS se tiver resiliência suficiente contra a maioria dos riscos associados a ataques através de recursos externos. Portanto, deve ser utilizado como o nível padrão de segurança para as aplicações web. Este nível, é apropriado para aplicações que tratam dados sensíveis, garante que os controles de segurança existem, funcionam e são utilizados dentro da aplicação.

Ameaças a aplicações neste nível serão tipicamente feitas por atacantes habilidosos e motivados, focando em alvos específicos, utilizando ferramentas e técnicas efetivas para o descobrimento e exploração de fraquezas dentro de aplicações.

Características da Aplicação
Armazena, transmite ou processa dados pessoais, confidenciais ou sensíveis de funcionários, parceiros de negócio, terceiros ou prestadores de serviço
Aplicações cuja integridade é crítica.

Nível 3 – Segurança Alto valor, garantia e resiliência

Este nível é o mais alto utilizado para avaliação utilizando o ASVS e é normalmente reservado para aplicações que requerem níveis significativos de verificações de segurança, como em situações de infraestrutura crítica, quando realizam processos críticos para o negócio ou funções sensíveis, quando é necessária a garantia da sua integridade para o funcionamento do negócio. ou seja, nível de segurança 3 é exigido em cenários onde uma falha pode provocar um impacto significativo nas operações da organização.

Características da Aplicação
Aplicações que executam funções críticas, onde a falha pode afetar significativamente as operações. Ex: uma aplicação interna que seja utilizada por várias outras, Sefaz Identity, Portal de Assinaturas, etc Aplicações que manipulam informações sigilosas ou que realizam processos críticos para o negócio ou funções sensíveis. Ex.: manipulação de informações sujeitas ao sigilo fiscal.

Seleção dos Requisitos de Segurança

Para o processo de seleção dos requisitos de segurança ser eficiente, deve-se aplicar uma avaliação mista, levando em consideração os ativos que devem ser protegidos e os riscos mapeados.

O anexo 1 apresenta uma lista de requisitos de segurança propostos, divididos nos três níveis de segurança. Além disso, os requisitos propostos nos anexos, não deverão ser tratados como únicos. Caso seja notada a necessidade de adição de um requisito, o mesmo pode e deve ser adicionado à sua respectiva tabela dentro do respectivo anexo.

A partir destes anexos, o PO irá gerar a Lista de Requisitos de Segurança, que será passada para a fase de sprint de desenvolvimento. Para gerar esta Lista de Requisitos de Segurança do Projeto, considere utilizar ferramentas, Kanban boards, Scrum ou tabelas que facilitarão na seleção e acompanhamento da implementação dos requisitos.

Anexo 1 – Requisitos de Segurança para Aplicações Web

Este anexo apresenta os controles que devem ser avaliados nos cenários de aplicações web, e caso atendam às necessidades de segurança da solução, devem ser aplicados.

Esta lista de requisitos foi baseada primordialmente no ASVS. Um dos documentos relacionados com este Guia de Requisitos de Segurança é o OWASP Top 10 **Proactive Controls**, que apresenta uma lista das técnicas de segurança que deveriam ser incluídas em todo projeto de desenvolvimento de software, organizadas por grau de relevância, escrita por desenvolvedores, para desenvolvedores.

1. Requisitos de Verificação de Tratamento de Erro
1.1. Tratamento de Erros

#	Descrição	1	2	3	CWE
1.1.1	Recomenda-se que uma mensagem genérica seja mostrada quando ocorrer um erro inesperado ou sensível à segurança, potencialmente com um ID exclusivo que a equipe de suporte possa usar para investigar. O software identifica uma condição de erro e cria seu próprio diagnóstico ou mensagens de erro que contêm informações confidenciais. Quando ele gera a exceção, está passando o ID do usuário que é chave primária da tabela de usuários. https://docs.microsoft.com/en-us/aspnet/web-forms/overview/older-versions-getting-started/deploying-web-site-projects/displaying-a-custom-error-page-cs	✓			210
1.1.2	Recomenda-se que ocorra o tratamento de exceções (ou um equivalente funcional) é usado na base de código para lidar com condições de erro esperadas e inesperadas. Se o aplicativo tratar as mensagens de erro individualmente, uma a uma, é provável que isso resulte em um tratamento de erros inconsistente. As causas dos erros podem ser perdidas. Além disso, informações detalhadas sobre as causas de um erro podem ser devolvidas involuntariamente ao usuário. Pode-se manipular erro padrão e erro HTTP adicionando uma customError ao arquivo Web.config. A customError permite que você especifique uma página padrão para a qual os usuários serão redirecionados quando ocorrer um erro. Ele também permite que especifique páginas individuais para erros de código de status específicos. Um manipulador de nível de página retorna o usuário à página onde ocorreu o erro, mas como as instâncias de controles não são mantidas, não haverá mais nada na página. Para fornecer os detalhes do erro ao usuário do aplicativo, você deve escrever especificamente os detalhes do erro na página. https://docs.microsoft.com/en-us/aspnet/web-forms/overview/getting-started/getting-started-with-aspnet-45-web-forms/aspnet-error-handling	✓			544

1.1.3	Recomenda-se que seja definido um tratador de erros de "último recurso" que captura todas as exceções não tratadas. Frequentemente, quando funções ou loops se tornam complicados, algum nível de limpeza de recursos é necessário durante a execução. As exceções podem perturbar o fluxo do código e impedir que a limpeza necessária aconteça. O método do controlador tem um tratamento de erro genérico para impedir que erros internos do sistema passem para o usuário. https://docs.microsoft.com/pt-br/dotnet/standard/exceptions/best-practices-for-exceptions	✓			460
1.1.4	Recomenda-se que o tratamento de erros para manusear uma requisição de API, se existir um erro na requisição do usuário ou se alguma coisa inesperada ocorrer no servidor, pode-se simplesmente lançar uma exceção para notificar o usuário que algo deu errado. https://www.yiiframework.com/doc/guide/2.0/pt-br/rest-error-handling https://docs.microsoft.com/pt-br/dotnet/api/system.servicemodel.faultexception-1?view=dotnet-plat-ext-5.0	✓			
1.1.5	Recomenda-se que ocorra o tratamento do retorno ou qualquer parte da resposta HTTP para não expor informações confidenciais durante a comunicação ou na camada de aplicação. Existem muitos tipos diferentes de erros que introduzem exposições de informações. A gravidade do erro pode variar amplamente, dependendo do contexto em que o produto opera, do tipo de informação sensível que é revelada e dos benefícios que ela pode oferecer a um invasor. Exemplo: Páginas de autenticação que retornam "**Falha no login - nome de usuário incorreta**" ou "**Falha no login - senha incorreta**" https://cwe.mitre.org/data/definitions/200.html	✓			200

1.2. Requisitos de Divulgação não Intencional de Segurança

#	Descrição	1	2	3	CWE
1.2.1.	Recomenda-se que se as mensagens de erro da Web ou do servidor de aplicação e do framework estão configuradas para fornecer respostas personalizadas e acionáveis pelo usuário para eliminar quaisquer divulgações de segurança não intencionais. As informações confidenciais podem ser valiosas por si mesmas (como uma senha) ou podem ser úteis para lançar outros ataques mais sérios.	✓			209
1.2.2.	Verificar se os modos de depuração da Web ou do servidor de aplicação e do framework estão desativados na produção para eliminar recursos de debug, consoles de desenvolvedor e divulgações de segurança não intencionais. https://vulncat.fortify.com/en/weakness?q=Disable%20debug	✓			497
1.2.3.	Assegurar que todas as mensagens de erro sejam registradas em arquivos de logs, inclusive as de erros inesperados. Considerar os seguintes aspectos: - Cuidar para neutralizar (escape) ou não corretamente a saída gravada nos registros de log; - Não truncar informação de segurança relevante - Não omitir informação de segurança relevante - Não registrar informações relevantes para a segurança de acordo com um nome alternativo da entidade afetada, usar o nome canônico Não registrar informações sensíveis no arquivo de log Verificar se a informação a informação a ser registrada no log é suficiente Verificar se o log não está com informações excessivas desnecessariamente	✓			1210