Artigo
Gabriel Vellasq... · Mar. 22, 2021 6min de leitura

Proteja sua API REST aplicando OWASP Top 10

Olá comunidade,

Você sabia sobre OWASP e os dez principais riscos de segurança de aplicativos da Web para sua API da Web ou aplicativos da Web?

OWASP é uma fundação comunitária criada para nos ajudar a melhorar a segurança de aplicativos / APIs da web. O OWASP torna os aplicativos da web mais seguros por meio de seus projetos de software de código aberto liderados pela comunidade, centenas de capítulos em todo o mundo, dezenas de milhares de membros e hospedando conferências locais e globais.

Para resumir os principais procedimentos para tornar seu aplicativo da web / API da web mais seguro, o OWASP publicou as recomendações "Top 10 Web Application Security Risks" (fonte: https://owasp.org/www-project-top-ten/):

1- **Injection**.. Falhas de injeção, como injeção de SQL, NoSQL, OS e LDAP, ocorrem quando dados não confiáveis são enviados a um intérprete como parte de um comando ou consulta. Os dados hostis do invasor podem induzir o intérprete a executar comandos indesejados ou acessar dados sem a autorização adequada.
2- **Broken Authentication**.. As funções de aplicativo relacionadas à autenticação e gerenciamento de sessão são frequentemente implementadas incorretamente, permitindo que os invasores comprometam senhas, chaves ou tokens de sessão ou explorem outras falhas de implementação para assumir as identidades de outros usuários temporária ou permanentemente.
3- **Sensitive Data Exposure**. Muitos aplicativos da web e APIs não protegem adequadamente dados confidenciais, como finanças, saúde e PII. Os invasores podem roubar ou modificar esses dados fracamente protegidos para conduzir fraude de cartão de crédito, roubo de identidade ou outros crimes. Os dados confidenciais podem ser comprometidos sem proteção extra, como criptografia em repouso ou em trânsito, e requerem precauções especiais quando trocados com o navegador.
4- **XML External Entities (XXE)**. Muitos processadores XML mais antigos ou mal configurados avaliam referências de entidades externas em documentos XML. Entidades externas podem ser usadas para divulgar arquivos internos usando o manipulador de URI de arquivo, compartilhamentos de arquivos internos, varredura de porta interna, execução remota de código e ataques de negação de serviço.
5- **Broken Access Control**. As restrições sobre o que os usuários autenticados têm permissão para fazer muitas vezes não são aplicadas de forma adequada. Os invasores podem explorar essas falhas para acessar funcionalidades e / ou dados não autorizados, como acessar contas de outros usuários, visualizar arquivos confidenciais, modificar dados de outros usuários, alterar direitos de acesso, etc.
6- **Security Misconfiguration**. A configuração incorreta de segurança é o problema mais comum. Isso geralmente é o resultado de configurações padrão inseguras, configurações incompletas ou ad hoc, armazenamento em nuvem aberta, cabeçalhos HTTP configurados incorretamente e mensagens de erro detalhadas contendo informações confidenciais. Não apenas todos os sistemas operacionais, estruturas, bibliotecas e aplicativos devem ser configurados com segurança, mas também devem ser corrigidos / atualizados em tempo hábil.
7- **Cross-Site Scripting (XSS)**. As falhas de XSS ocorrem sempre que um aplicativo inclui dados não confiáveis em uma nova página da web sem validação ou escape adequado, ou atualiza uma página da web existente com dados fornecidos pelo usuário usando uma API do navegador que pode criar HTML ou JavaScript. O XSS permite que os invasores executem scripts no navegador da vítima, que podem sequestrar as sessões do usuário, desfigurar sites ou redirecionar o usuário para sites maliciosos.
8- **Insecure Deserialization**. A desserialização insegura geralmente leva à execução remota de código. Mesmo que as falhas de desserialização não resultem na execução remota de código, elas podem ser usadas para realizar ataques, incluindo ataques de repetição, ataques de injeção e ataques de escalonamento de privilégios.
9- **Using Components with Known Vulnerabilities**. Componentes, como bibliotecas, estruturas e outros módulos de software, são executados com os mesmos privilégios do aplicativo. Se um componente vulnerável for explorado,

esse tipo de ataque pode facilitar a perda séria de dados ou o controle do servidor. Aplicativos e APIs que usam componentes com vulnerabilidades conhecidas podem minar as defesas do aplicativo e permitir vários ataques e impactos.
10- **Insufficient Logging & Monitoring**. O registro e o monitoramento insuficientes, juntamente com a integração ausente ou ineficaz com a resposta a incidentes, permitem que os invasores ataquem ainda mais os sistemas, mantenham a persistência, se movam para mais sistemas e adulterem, extraiam ou destruam dados. A maioria dos estudos de violação mostra que o tempo para detectar uma violação é de mais de 200 dias, normalmente detectado por partes externas em vez de processos internos ou monitoramento.

Algumas técnicas podem ser usadas para implementar os dez primeiros OWASP, consulte a tabela:

Recursos de recomendação OWASP	Recursos
Injection	No SQL, use parâmetros de entrada especificados usando o " ? " caractere ou variáveis de host de entrada (por exemplo,: var). Evite concatenar instruções SQL e argumentos de método de entrada. Use a API Sanitazer como https://developer.mozilla.org/en-US/docs/Web/API/HTMLSanitizerAPI
Broken Authentication	Crie senhas fortes, consulte o documento da Microsoft: https://support.microsoft.com/en-us/windows/create-and-use-strong-passwords-c5cebb49-8c53-4f5e-2bc4-fe357ca048eb Não use senhas dentro do código-fonte, criptografe-o, use serviços de cofre ou serviços SSO. Externalize suas senhas usando parâmetros do ambiente Docker / OS Não permita ataques de força bruta usando gateways de API como IAM: https://docs.intersystems.com/components/csp/docbook/Doc.View.cls?KEY=CIAM1.5
Sensitive Data Exposure	Exposição de dados confidenciais Evite retornar dados de cartão de crédito, dados de saúde e outros dados confidenciais em sua API, se não for possível, criptografe os canais de mensagens e o armazenamento de dados confidenciais. Minimize a coleta de dados ao mínimo possível.
XXE	XXE Use% XMLAdaptor
Broken Access Control	Use ferramentas de gerenciamento de acesso como IAM: https://docs.intersystems.com/components/csp/docbook/Doc.View.cls?KEY=CIAM1.5
Security Misconfiguration	Atualize seu software periodicamente e execute o software Pentest em seu ambiente para realizar os procedimentos apropriados. Siga o Guia de administração de segurança da intersystems: https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GCAS
XSS	Faça a sanitização / validação de entrada, consulte dicas OWASP: https://cheatsheetseries.owasp.org/cheatsheets/Input_

	ValidationCheatSh...
Insecure deserialization	Crie hash para arquivos, crie pastas mais seguras sem permissão de execução para armazenar arquivos e validar extensões de arquivos
Using Components with Known Vulnerabilities	Na pilha da InterSystems o Apache Web Server é a ferramenta mais conhecida, portanto tome cuidado ao configurá-la para a produção, seguindo: https://www.tecmint.com/apache-security-tips/. Habilite o firewall do seu sistema operacional para ubuntu executar ufw enable (consulte: https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-18-04)
Insufficient Logging & Monitoring	Use o Log Monitor (https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls...) Siga-o: https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=ALOG Monitore usando SAM: https://docs.intersystems.com/sam/csp/docbook/DocBook.UI.Page.cls?KEY=ASAM

Você pode usar o software WAF em alinhamento com os dez principais OWASP, consulte: https://www.g2.com/categories/api-security ou use o IAM da InterSystems. Da lista G2, gosto do 42 crunch, que permite uma conta gratuita e você indica seu arquivo OpenAPI e ele é analisado para relatar recomendações, veja em https://42crunch.com/owasp-api-security-top-10/.

#InterSystems IRIS

URL de origem:https://pt.community.intersystems.com/post/proteja-sua-api-rest-aplicando-owasp-top-10