### **ESTUDO DE CASO**

O objetivo deste estudo de caso é trabalhar técnicas estudadas durante o curso no contexto de um software real.

**Escolha do Software**

A escolha do software que será utilizado no estudo de caso deve seguir algumas regras:

* Um software que seja utilizado em produção.
* O código fonte do software deve estar disponível, já que uma das atividades vai envolver a análise estática do código.
* O software deve ter documentação pública que dê informações suficientes para elaboração de casos de uso, abuse cases, misuse cases e diagrama de arquitetura.
* Um software que tenha um certo nível de complexidade. Caso contrário, o estudo de caso vai ficar muito simplificado.
* O software deve conter vulnerabilidades que sejam encontradas pelo Snyk Code. Por isso, recomenda-se que ele seja analisado pelo Snyk antes de ser escolhido para a atividade.

Uma dica é procurar por softwares comerciais (para dispositivos móveis e/ou Web) cujo código esteja disponível em repositórios como o GitHub. Esta página, por exemplo, traz uma lista de aplicativos Android com código no GitHub: <https://github.com/LinuxCafeFederation/awesome-android>.

**Instalação e utilização do Snyk (Análise Estática de Código)**

Você deve, primeiramente, criar uma conta para utilizar a ferramenta de análise estática de código Snyk no site <https://snyk.io>. A ferramenta contém um plano de uso gratuito que atende perfeitamente às necessidades desta atividade. O próximo passo é instalar o Snyk CLI em seu computador para fazer a análise de código. As instruções para instalá-lo estão disponíveis no link <https://docs.snyk.io/snyk-cli/install-the-snyk-cli> .

A maneira mais comum de instalá-lo é usando o gerenciador de pacotes npm. Se você ainda não tem o npm em sua máquina, pode instalar o Node.js disponível neste link <https://nodejs.org/en/download/>. O npm e o Node também podem ser instalados por meio do gerenciador de versões nvm. As instruções para este procedimento são apresentadas a seguir para diferentes sistemas operacionais.

MacOS ou Linux

nvm (<https://github.com/nvm-sh/nvm>)

Comando para instalar o nvm:

>> curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash

NodeJs

Comando para instalar o Node:

>> nvm install -–lts

Windows

nvm e Node

Para máquinas Windows, é melhor instalar o nvm-windows. Instruções disponíveis em: <https://docs.microsoft.com/en-us/windows/dev-environment/javascript/nodejs-on-windows>

No caso do nodeJS, independente do sistema operacional, instale sempre a versão LTS mais recente.

Para instalar o Snyk, execute o comando npm install -g snyk

Após instalar o npm e o Snyk, você deve habilitar a utilização do Snyk Code no site do Snyk. Faça o seu login no site e acesse a opção *Settings*. Um menu será aberto com diversas opções. Acesse a opção Snyk Code e dentro dela ative o *Enable Snyk Code*. Antes de sair, salve as modificações utilizando o botão *Save Changes*.

Com o Snyk instalado e o Snyk Code habilitado, siga os passos abaixo para inspecionar o código de uma aplicação:

1- Faça download do código-fonte da aplicação escolhida.

2- Abra um terminal no seu computador e navegue até o diretório raiz contendo todo o código-fonte da aplicação.

3- Dentro do diretório, digite o seguinte comando snyk code test e pressione enter. O Snyk fará a análise de todos os arquivos presentes nesta pasta e nas pastas internas. No final, ele apresentará um relatório de vulnerabilidades.

**Estrutura do Estudo**

O estudo vai ser dividido nas seguintes etapas: Abuse/Misuse Cases, Threat Modeling, Análise Estática de Código. As etapas são detalhadas a seguir.

*Abuse/Misuse Cases*

Nesta parte do estudo, você deve:

1- Analisar a documentação disponível sobre o software e escrever três casos de uso para ele. Estes casos de uso devem conter apenas o fluxo principal.

2- Escrever pelo menos um abuse case ou um misuse case para cada caso de uso. Lembre-se que os abuse/misuse cases devem, entre outras coisas, sempre incluir os danos causados. Ao final, devemos ter pelo menos um abuse case e um misuse case. Em outras palavras, você não deve fazer só abuse cases ou só misuse cases.

3- Escrever os requisitos de segurança que cubram os misuse e abuse cases relatados.

*Threat Modeling*

Nesta seção, você deve:

1- Utilizando uma ferramenta de *threat modeling*, construir um diagrama que represente a arquitetura do software analisado. Lembre-se de utilizar os elementos discutidos em sala de aula como *external interactors*, *processes*, fluxos de dados, *data stores*, e *boundaries*.

2- Usando o STRIDE, levantar ao menos duas ameaças para cada um dos elementos do diagrama. Descreva estas ameaças. Ao final, devemos ter ao menos uma ameaça de cada um dos tipos STRIDE.

3- Para cada ameaça, propor uma ação de mitigação.

*Análise Estática de Código*

Nesta seção, você deve:

1- Executar o Snyk Code para analisar todo o código da aplicação.

2- Escolher uma vulnerabilidade encontrada pelo Snyk.

3- Verificar a ocorrência desta vulnerabilidade no código.

4- Escrever um parágrafo explicando a vulnerabilidade e como ela poderia ser mitigada.

**Estrutura do documento a ser entregue**

O documento a ser entregue vai ser dividido nas seguintes seções: Identificação do aluno, Apresentação do Software, Abuse/Misuse Cases, Threat Modeling, Análise Estática de Código. As seções são detalhadas a seguir.

*Identificação do aluno*

Nome e número de matrícula

*Apresentação do software*

Seção contendo breve descrição do software escolhido e local onde o código fonte pode ser encontrado/acessado.

*Abuse/Misuse Cases*

Seção contendo os casos de uso, abuse cases, misuse cases e requisitos de segurança.

*Threat Modeling*

Seção contendo o diagrama da arquitetura do software e as descrições das ameaças levantadas e suas possíveis mitigações. Para cada ameaça, deve haver uma descrição e sua classificação no modelo STRIDE.

*Análise Estática de Código*

Seção contendo um parágrafo que explica a vulnerabilidade encontrada pelo Snyk e propõe uma maneira de mitigá-la.