ATIVIDADE PARCIAL

Conhecimentos necessários para o desenvolvimento da Atividade:

- Conhecimentos sobre técnicas de invasão;
- Conhecimentos sobre outras áreas de segurança como:
- ◦ Teste de software seguro;
- ◦ Técnicas de implementação de software seguro;
- ◦ Atividades de configurações de ambiente seguro. .

Objetivo da proposta da Atividade:

- Desenvolver uma aplicação utilizando metodologias, técnicas e ferramentas com base no ciclo de vida de desenvolvimento seguro de software, tais como TDD, SDL.

Descrição da atividade:

- Os discente deverão desenvolver aplicações seguras que estejam de acordo com as necessidade de um público-alvo. Os projetos têm por natureza a extensão, então os alunos deverão buscar um grupo de pessoas com uma necessidade específica para realizar as tarefas do processo de desenvolvimento. Além de aplicarem processo, técnicas, metodologias que visem o desenvolvimento de aplicações seguras.

Passo a passo para desenvolvimento da atividade:

- • Formar equipes de 4 até 6 integrantes;
- • A equipe desenvolverá uma ação de extensão, que será conduzida conforme as orientações deste documento.
- • Realizar o levantamento de requistos funcionais e não funcionais, priorizando aspectos e funcionalidades seguras (min. de 5 requisitos funcionais), bem como realizar a análise de riscos dos mesmo.
- ◦ Planejamento e Requisitos
- ▪ Definição dos requisitos funcionais e não funcionais (incluindo segurança).
- ▪ Identificação de normas, leis e compliance (ex.: LGPD, PCI-DSS, HIPAA).
- ▪ Criação de políticas de segurança do software.
- ▪ Análise de riscos iniciais: quais dados serão tratados? onde estão as maiores ameaças?
- • Realizar a modelagem de diagramas de arquitetura, e de modelos pertinentes às plataforma que subsidiarão a execução das aplicações, sejam web, mobile ou desktop. (min. De 3 modelos)
- ◦ Análise e Design
- ▪ Modelagem de ameaças (Threat Modeling): prever ataques como SQL Injection, XSS, DoS.
- ▪ Escolha de arquiteturas e padrões que reforcem segurança (ex.: microserviços com autenticação forte).
- ▪ Definição de controles de segurança (criptografia, autenticação multifator, segregação de funções).

- • Implementar a aplicação utilizando frameworks, linguagens de programação, e IDE's que possam dar suporte a técnicas de programação modernas e seguras.
- ◦ Implementação (Codificação Segura)
- ▪ Programação seguindo guidelines de segurança (ex.: OWASP Secure Coding Practices).
- ▪ Revisões de código com foco em vulnerabilidades.
- ▪ Ferramentas automáticas de análise estática (SAST) para encontrar falhas no código antes da entrega.
- • Validar e verificar as funcionalidades utilzando dos processo estabelecidos pela metodologia de desenvolvimento escolhida.
- ◦ Teste e Verificação
- ▪ Testes dinâmicos (DAST) simulando ataques durante a execução.
- ▪ Testes de penetração (Pentests) manuais ou automatizados.
- ▪ Análise de dependências externas (SCA – Software Composition Analysis) para evitar bibliotecas com falhas conhecidas.
- ▪ Verificação de conformidade com requisitos de segurança.
- • O versionamento da aplicação de ocorrer usando o GITHUB, bem como a gerência de projetos também deve ser desenvolvido na tela de "Projetos" do github.
- ◦ As tarefas devem estar distribuídas entre os participantes e o acesso ao repositório deve ser associado ao professor da disciplina, segue o e-mail e user do github do professor, ronnison.reges@gmail.com, @ronnison.
- ◦ Os requisitos e as modelagens devem ser apresentadas no readme do github, como documentação usufruindo livremente de hyperlinks, como a equipe desejar.
- • Os links dos projetos devem ser enviados pela plataforma do AVA bem como os códigos-fontes como comprovação de execução das devidas tarefas.
- • O público-alvo da ação de extensão a ser escolhido poderá ser a comunidade científica/acadêmica ou microempreendedores que tenham interesse em utilzar aplicações seguras.
- • Durante a realização das reuniões de levantamento de requisitos ou de acompanhamento com os stakeholders a equipe deve pedir a permissão dos participantes para fazer registro para evidenciar as ações. A equipe responsável deverá registrar os eventos com fotos e produzir um relatório com o quantitativo de participantes das reuniões, dia e horário da ação e se os objetivos da aplicação foram alcançados. Além de aplicar uma avaliação de feedback com o público ao final do desenvolvimento, com intuito de verificar se a aplicação desenvolvida atende as necessidades esperadas.
- • Elaborar um relatório para descrever e evidenciar a apresentação final do produto aos usuários. É importante apresentar a frequência dos participantes, registros fotográficos ou capturas de telas da reunião final, o relaório de apresentar em forma de gráficos as perguntas do formulário de feedback, além de escrever as conclusões sobre cada pergunta e gráfico resultante, e enviar pelo AVA o relatório.
- • O relatório deve conter as seguintes informações:
- ◦ Imagens do momento da realização da reunião final, virtual ou presencial.
- ◦ Feedback aplicado para os participantes da equipe.

- ◦ Quantitativo de participantes.
- ◦ Relato se os objetivos da oficina foram alcançados.
- ◦ Anexos: Questões do formulário de feedback e link do projeto github, e link da aplicação;

Orientações para envio da atividade:

- ◦ O nome completo e matrícula dos integrantes da equipe devem ser inseridos na capa do arquivo, 0,5 pts serão deduzidos caso não esteja em conformidade;
- ◦ O relatório deve ser enviado em arquivo PDF, 0,5 pts serão deduzidos caso não esteja em conformidade.
- ◦ Todos os membros devem estar agrupados na ferramenta de grupos do AVA, mesmo aqueles que estejam trabalhando individualmente, também devem estar atribuídos a um grupo, 0,5 pts serão deduzidos caso não esteja em conformidade
- ◦ O envio deve ser realizado por todos os membros da equipe, para fins de correção, 0,5 pts serão deduzidos caso não esteja em conformidade.
- ◦ Envios de trabalhos vazios serão desconsiderados e a nota '0', zero, será atribuída à equipe.
- ▪ No caso de algum membro não estar referenciado no documento, mas esteja cadastrado no gihub, e vice-versa, 0,5 pts serão deduzidos de todos os membros da equipe caso não esteja em conformidade.

Critérios a serem avaliados:

- • Processo de Documentação [0,0 a 4,0 pontos]:
- ◦ Documentação no GITHUB, desenvolvida no readme (1 pts):
- ▪ Todos membros da equipe devem estar devidamente cadastrados no repositório do github, inclusive o professor;
- ▪ Requisitos;
- ▪ Modelagem;
- ◦ Gestão de Projetos (1 pts):
- ▪ As atividades do projeto devem estar cadastradas e atribuídas no github;
- ◦ Implmentação (1 pts);
- ◦ Testes (1 pts);
- • Relatório da ação [0,0 a 1,0 pontos] (Obrigatório):
- ◦ Qualidade do conteúdo, clareza, linguagem formal e originalidade; - Conteúdo teórico;
- ◦ Evidências da ação de extensão e feedback do beneficiário.
- ◦ Anexo 1: Formulário de Perguntas;
- ◦ Anexo 2: Link do github;
- ◦ Anexo 3 (Opcional): Link da aplicação;

Total de pontuação: 5,0 pts.
Prazo para envio da atividade: 29/09/2025 (Módulo A).