Programação Segura (24/25)
Prof Ronaldo Moreira Salles

Projeto Final - OP1

Ataque: execução de shellcode com privilégios de root

O objetivo final do projeto é conseguir executar um *shellcode* com provilégios de root por meio da exploração de uma ou mais vulnerabilidades (*string* de formato, *buffer overflow*) em um código vulnerável. A avaliação levará em conta o grau de dificuldade do ataque e o seu resultado. Por exemplo, ataque via string de formato e *shellcode* armazenado na *heap* (maior dificuldade), *buffer overflow* e *shellcode* na stack, . . ., *shellcode* executado simplesmente por chamada *system* (menor dificuldade).

Parte 1

1) Desenvolver um programa em C (python – pontuação menor) contendo uma vulnerabilidade de string de formato (buffer overflow – pontuação menor).
2) Desabilitar todas os mecanismos de proteções de stack e heap.
3) Compilar o prorgama usando as opções de compilação necessárias.
4) Converter o programa em SetUID para o escalonamento de privilégios de root.

Parte 2

5) Realização do ataque ao programa vulnerável para atingir o objetivo.
    - Entendimento de como criar um *shellcode* e injetá-lo na *heap* (*stack*).
    - Entendimento do processo de execução do *shellcode* pela exploração da *string* de formato (*buffer overflow*).
    - Esta etapa é a mais importante e deve ser detalhadamente explicada no relatório final, pois representa a maior parte da pontuação do projeto.
    - Tanto o *shellcode* quanto a *string* devem ser entradas externas ao programa por meio da interface com o utilizador

Entregáveis:

- Relatório técnico detalhando todos os passos acima, com ênfase especial no passo 5.
- Vídeo de 3 a 5 minutos com a demonstração do ataque.
- Apresentação do ataque e defesa do trabalho em sala de aula.

Projeto Final - OP2

Aplicação Web Vulnerável com Módulo de Correções (OWASP Top 10)

Objetivo: Desenvolver uma aplicação web com vulnerabilidades intencionais e, posteriormente, aplicar as correções necessárias com base nas boas práticas de segurança recomendadas pela OWASP (Open Worldwide Application Security Project), abordando vulnerabilidades comuns presentes no relatório **OWASP Top 10**.

Descrição Geral: O projeto consiste na criação de uma aplicação web simples (por exemplo, um sistema de registo de utilizadores e troca de mensagens), passando pelas seguintes fases:

Etapas do Projeto

1. Desenvolvimento de uma versão insegura da aplicação

A primeira versão da aplicação deve conter, de forma intencional, pelo menos **quatro vulnerabilidades** comuns descritas na OWASP Top 10. Exemplos de vulnerabilidades a incluir:

- **Injeção**
  Exemplo: Falta de validação ou sanitização de entradas que permite **Injeção SQL** ou **Injeção de comandos** no servidor.
- **Cross-Site Scripting (XSS)**
  Exemplo: Campos de formulário que permitem a injeção e execução de scripts maliciosos no navegador de outro utilizador.
- **Gestão incorreta de sessões e autenticação**
  Exemplo: Tokens de sessão previsíveis, falta de invalidação de sessões, ausência de timeout, ou dados de autenticação armazenados em texto plano.
- **Exposição de dados sensíveis**
  Exemplo: Transmissão de dados sem encriptação (ex: passwords via HTTP), má gestão de dados confidenciais (ex: ficheiros de configuração acessíveis publicamente).
- (Opcional) Outras vulnerabilidades do OWASP Top 10 podem ser integradas, como: controlo de acesso falho, configuração insegura, dependências vulneráveis, etc.

2. Desenvolvimento da versão segura (corrigida)

Nesta fase, o estudante deverá aplicar as correções apropriadas para mitigar as vulnerabilidades implementadas anteriormente, seguindo as **boas práticas de segurança** recomendadas pela OWASP. Deverá ser apresentada uma **nova versão funcional e segura** da aplicação.

3. Relatório técnico

O relatório técnico deve documentar todo o processo, com as seguintes secções:

- **Descrição geral da aplicação desenvolvida.**
- **Lista das vulnerabilidades implementadas** com:
  - Explicação do tipo de falha (com referência à OWASP Top 10).
  - Código vulnerável com explicação do problema.
  - Demonstração (com capturas de ecrã) da exploração da vulnerabilidade.
- **Medidas corretivas aplicadas**, com:
  - Código corrigido com explicação das alterações.
  - Justificação técnica da solução.
  - Capturas de ecrã demonstrando o comportamento seguro.
- **Reflexão crítica** sobre os riscos envolvidos, o impacto das falhas e a importância da validação de segurança no ciclo de desenvolvimento.

Entregáveis

1. Código-fonte das duas versões (insegura e segura).
2. Relatório técnico em formato PDF.
3. Vídeo (3-5 minutos) com demonstração das vulnerabilidades e respetivas correções.

Regras:

- O trabalho deve ser desenvolvido individualmente/dupla.
- O aluno vai escolher **UM** dos temas para desenvolver o projeto final (Op1 **OU** Op2).
- A deteção de trabalho fraudulento invalida o trabalho desenvolvido. Serão considerados fraudulentos os trabalhos que apresentem, total ou parcialmente, conteúdo desenvolvido por terceiros.

Datas e apresentação:

O relatório e vídeo devem ser entregues até ao dia 31 de maio de 2025 - 23h59 em um ficheiro .zip submetido na plataforma Moodle. O nome do ficheiro deve corresponder ao número de estudante (ex.: 8149999_8148888.zip).

No dia 03 de junho de 2025, decorrerá a apresentação do projeto global (durante a aula), e o estudante deverá preparar uma apresentação de 10 minutos mostrando as 5 etapas realizadas, desde a conceção inicial até a execução do ataque.

Cálculo da nota final:

100% × (Relatório + Apresentação)