Aula TP - 01/Abr/2019

Cada grupo deve colocar a resposta às perguntas dos seguintes exercícios na área do seu grupo no Github até ao final do dia 16/Abr/2019. Por cada dia de atraso será descontado 0,15 valores à nota desse trabalho.

Exercícios

1. Vulnerabilidade de codificação

Experiência 1.1 - *Common Weakness Enumeration* (CWE)

A *Common Weakness Enumeration* (CWE) classifica classes de vulnerabilidade, atribuindo a cada classe de vulnerabilidades um identificador.

Aceda a https://cwe.mitre.org/ e veja quais as vulnerabilidades que são identificadas como:

- vulnerabilidades de projeto, introduzidas durante a fase de projeto do software (obtenção de requisitos e desenho) - https://cwe.mitre.org/data/definitions/701.html. Explore um dos membros dessa classe de vulnerabilidades e veja (e perceba) um dos exemplos demonstrativos.
- vulnerabilidades de codificação, introduzidas durante a programação do software - https://cwe.mitre.org/data/definitions/702.html. Explore um dos membros dessa classe de vulnerabilidades e veja (e perceba) um dos exemplos demonstrativos.
- vulnerabilidade operacional, causadas pelo ambiente no qual o software é executado ou pela sua configuração - https://cwe.mitre.org/data/definitions/16.html. Explore um dos membros dessa classe de vulnerabilidades e veja (e perceba) um dos exemplos demonstrativos..

Veja ainda quais as vulnerabilidades identificadas como:

- *CWE/SANS Top 25 Most Dangerous Software Errors* (2011) - https://cwe.mitre.org/top25/
- *OWASP Top Ten* (2017) - https://cwe.mitre.org/data/definitions/1026.html

Experiência 1.2 - *Common Vulnerabilities and Exposures* (CVE)

A *Common Vulnerabilities and Exposures* (CVE) identifica vulnerabilidades (de projeto e codificação) existentes em software comercial ou aberto, com identificador com formato CVE-AAAA-NNNN, sendo AAAA o ano em que a vulnerabilidade foi catalogada e NNN o seu número.

Aceda a https://cve.mitre.org/ e verifique:

- o detalhe da vulnerabilidade mais recente;
- as vulnerabilidades identificadas no Google Chrome;
- as vulnerabilidades identificadas no Facebook.

Experiência 1.3 - *Common Vulnerability Scoring System* (CVSS)

O *Common Vulnerability Scoring System* (CVSS) disponibiliza um modelo quantitativo para definir as características e impacto das vulnerabilidades, garantindo uma medição precisa e repetível para gerar pontuações de impacto de vulnerabilidade.

Dois usos comuns do CVSS são a priorização das atividades de correção de vulnerabilidades e, o cálculo da gravidade das vulnerabilidades descobertas.

Explore o calculador de vulnerabilidades em https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator.

Experiência 1.4 - *National Vulnerability Database* (NVD)

A *National Vulnerability Database* (NVD) é o repositório de vulnerabilidades gerido pelo NIST. Baseia-se no CVE, mas inclui a gravidade da vulnerabilidade, de acordo com o CVSS (*Common Vulnerability Scoring System*)

Aceda a https://nvd.nist.gov/ e verifique:

- qual é a vulnerabilidade mais recente identificada?
- essa vulnerabilidade é a mesma vulnerabilidade mais recente encontrada na experiência 1.2 (CVE)? Qual poderá ser o motivo?
- as vulnerabilidades identificadas no Google Chrome;
- as vulnerabilidades identificadas no Facebook.

Pergunta P1.1

Em https://informationisbeautiful.net/visualizations/million-lines-of-code/ encontra (algumas são estimativas) o número de linha de código (*SLOC - Source Lines Of Code*) de alguns pacotes/plataformas de software.

1. Estime o número de bugs do Facebook, software de automóveis, Linux 3.1 e de todos os serviços Internet da Google.
2. Quantos desses bugs são vulnerabilidades?

Pergunta P1.2

Considere os três tipos de vulnerabilidades: de projeto, de codificação e operacional. Apresente para cada um deles dois exemplos e discuta a dificuldade de correção.

Pergunta P1.3

O que é que distingue uma vulnerabilidade dia-zero de outra vulnerabilidade de codificação que não seja de dia-zero?

Experiência 1.5 - *Exploit Database*

O *Exploit Database* contém um arquivo de *exploits* públicos, identificando o CVE da vulnerabilidade e/ou software que explora, para utilização por investigadores de vulnerabilidades e *penetration testers*.

Aceda a https://www.exploit-db.com/ e verifique:

- qual é o *exploit* mais recente identificado?
- o que é que o *exploit* relativo ao CVE-2019-6780 lhe permite fazer?

Experiência 1.6 - *Google Hacking Database*

O *Google Hacking Database* é um arquivo de Google *dorks* (*query* de pesquisa que retorna a informação sensível) que embora sendo uma forma de *exploits*, são também utilizados para criar novos *exploits*.

Aceda a https://www.exploit-db.com/google-hacking-database/ e verifique:

- qual é o *dork* mais recente identificado?
- explore alguns *dorks*.

Experiência 1.7 - Linguagem C

Tal como visto na aula teórica, a linguagem C é uma linguagem compilada em que cada instrução C é traduzida para instruções em linguagem máquina.

Utilizando a máquina virtual, compile um programa C com o `gcc` com a opção `-S`, de forma a produzir o código assembly correspondente ao código binário/executável (o código assembly fica no ficheiro com extensão .s). Verifique o ficheiro .s e compare-o com o ficheiro .c original.

Como exemplo utilize o seguinte código C:

```
#include <stdio.h>

void main()
{
  printf("Hello World\n");
}
```

Projeto de Engenharia de Segurança

Pode utilizar o resto da aula para o projeto de Engenharia de Segurança