Øvelse 46 – CWE-gruppeøvelse

Formål

Formålet med denne øvelse er at give jer en introduktion til konceptet **CWE** (Common Weakness Enumeration), og hvordan svagheder i software design og implementation klassificeres. Øvelsen træner jeres evne til at forstå og forklare typiske sikkerhedsmæssige svagheder – og diskutere deres betydning i praksis.

Baggrund

Hvor CVE'er identificerer **konkrete sårbarheder i specifik software**, beskriver CWE'er **generelle svagheder i design eller implementation**, som ofte ligger til grund for sårbarhederne. CWE'er bruges i secure development, trusselsmodellering, code reviews og automatiske analyseværktøjer.

Ved at forstå CWE'erne kan man bedre:

- Genkende mønstre i sårbare systemer
- Undgå at gentage typiske fejl i egne løsninger
- Forstå hvordan en svaghed kan føre til en alvorlig sårbarhed

> CWE'er danner grundlag for mange sikkerhedsstandarder, sikkerhedstests og værktøjer som fx SAST (statisk analyse), OWASP SAMM og DevSecOps-pipelines.
> De bruges til at **forbygge fejl tidligt i udviklingen** – før de bliver til CVE'er.

Instruktioner

I jeres gruppe skal I undersøge og diskutere følgende svagheder:

- CWE-287
- CWE-272
- CWE-1329

For hver svaghed skal I finde svar på følgende:

- Hvad dækker svagheden over?

- Hvilke typer fejl eller situationer kan føre til denne svaghed?
- Hvilke risici opstår, hvis svagheden udnyttes?
- Hvordan kan man undgå eller afbøde denne type svaghed?

Tid til rådighed: 25 minutter

> I kan bruge MITRE'sCWE-database og evt. andre sikkerhedsressourcer.

Nyttige links

- CWE-database (MITRE)

Refleksionsspørgsmål

- Hvordan adskiller CWE'er sig fra CVE'er?
- Hvilken rolle spiller CWE'er i secure software development?
- Hvilke svagheder er sværest at opdage i udviklingsfasen – og hvorfor?
- Hvordan kan man bruge CWE'er til at forbedre kvaliteten og sikkerheden i et projekt?

CIS Controls – Kobling

CIS Control	Titel	Relevans
16 – Application Software Security	Udvikling med sikkerhed i fokus	CWE'er understøtter sikker softwarearkitektur og fejlforebyggelse
04 – Secure Configuration of Enterprise Assets	Fejlkonfigurationer	Mange CWE'er handler om forkerte standardindstillinger og privilegier
18 – Penetration Testing	Validering mod kendte svaghedsmønstre	CWE'er bruges som baggrund for mange testcases i SAST/DAST/pen-test

---

Videre perspektiv: CWE'er i automatisering og værktøjer

CWE'er danner grundlag for mange sikkerhedsværktøjer, som automatiserer test og analyse af kode og konfigurationer:

- **SAST-værktøjer** (fx SonarQube, CodeQL) identificerer kildesvagheder baseret på CWE-mønstre
- **EDR/SIEM-løsninger** (fx Wazuh) kan matche adfærds- og konfigurationsfund mod CWE-kategorier
- **Trusselsmodellering og secure design** anvender CWE'er til at kortlægge mulige fejltyper og afværge dem i arkitekturen

 Når du forstår CWE'erne, lærer du ikke bare at lukke sårbarheder – men at undgå at skabe dem.

2025-04-28 10:45:51