Øvelse 45 – CVE-gruppeøvelse

Formål

Formålet med denne øvelse er at styrke jeres forståelse af sårbarheder (CVE'er) gennem fælles research og dialog. I skal som gruppe undersøge specifikke kendte sårbarheder og diskutere deres tekniske betydning og alvor. Øvelsen træner jeres evne til at analysere trusler og forstå deres potentielle påvirkning i praksis.

Baggrund

En **CVE** (Common Vulnerabilities and Exposures) er en standardiseret identifikator for kendte sårbarheder. CVE'er bruges af både udviklere, sikkerhedsfolk og værktøjer til at identificere og klassificere svagheder i software og systemer.

Ved at undersøge konkrete CVE'er får I erfaring med:

- Hvordan sårbarheder beskrives og klassificeres
- Hvilken type trussel de udgør (f.eks. remote code execution, privilege escalation, DoS)
- Hvordan man vurderer alvoren via CVSS-score og kontekst

> I praksis bruges CVE'er af både Blue Team (forsvar), udviklere og compliance-ansvarlige. De er centrale i patch cycles, vulnerability scanning og risikovurderinger – og danner grundlag for prioritering af udbedringer og afværgeforanstaltninger.

Instruktioner

I jeres gruppe skal I undersøge og diskutere følgende sårbarheder:

- CVE-2023-32269
- CVE-2023-31436
- CVE-2014-0160
- CVE-2022-47509
- CVE-2021-44228
- CVE-2022-26903

For hver sårbarhed skal I finde svar på følgende:

- Hvilken type sårbarhed er der tale om?
- Hvornår er sårbarheden et problem?
- Hvor alvorlig er den (fx via CVSS-score)?

Brug gerne CVE Search – Mitre

Tid til rådighed: 30 minutter

> I kan bruge både Mitre og NIST's databaser samt evt. tekniske blogindlæg eller producentens advisories.

Bonusopgave (valgfri)

Find en CVE, der er offentliggjort inden for den seneste måned – og vurder:

- Hvilket system påvirkes?
- Er der kendt exploit tilgængelig?
- Hvor hurtigt bør en organisation reagere?

Nyttige links

- CVE Search – Mitre
- National Vulnerability Database (NIST)

Refleksionsspørgsmål

- Hvad har sårbarhederne tilfælles – og hvad adskiller dem?
- Hvilke sårbarheder har størst potentielt impact, og hvorfor?
- Hvordan kan denne viden bruges i fx patch management eller risikovurdering?
- Hvad er forskellen på en "teoretisk" og en "aktivt udnyttet" sårbarhed – og hvordan ændrer det prioritering?

CIS Controls – Kobling

CIS Control	Titel	Relevans
07 – Continuous Vulnerability Management	Identificér og håndtér sårbarheder	Øvelsen træner systematisk CVE-forståelse og prioritering
04 – Secure Configuration of Enterprise Assets	Kend og reducer risiko fra fejlkonfigurationer	Mange CVE'er udspringer af fejlkonfigurerede eller usikrede komponenter
06 – Access Control Management	Begrænsning af adgang ved udnyttelse	Flere sårbarheder i opgaven relaterer til eskalering og brud på adgangskontrol

---

Videre perspektiv: CVE'er i automatiseret sikkerhed

Moderne SIEM- og sårbarhedsscanningsværktøjer – fx **Wazuh**, **Nessus**, **OpenVAS** eller **Qualys** – anvender CVE-databaser som grundlag for deres detektion og rapportering.

Det betyder, at:

- Filer og pakker på et system kan matches mod kendte CVE'er
- Kritiske sårbarheder (fx RCE eller 0-days) kan udløse alarmer
- Resultater kan prioriteres baseret på CVSS-score og aktiv udnyttelse
- Organisationer kan automatisere patch cycles og risikooverblik

 Når I arbejder med CVE'er i hånden, lærer I at forstå det datafundament, som mange automatiske værktøjer bygger videre på.

2025-04-29 08:32:06