TECHCOLLEGE

Undervisningsmateriale til AMU-kursus

Kursusoplysninger

Kursusnavn	**Programmering: Softwaresikkerhed**
AMU-kursusnummer	49507
Varighed	5 dage
Målgruppe	Faglærte personer inden for det datatekniske område og andre inden for AMU-målgruppen med tilsvarende kvalifikationer, der skal eller ønsker at arbejde med softwaresikkerhed i forbindelse med programudvikling. Det anbefales, at deltageren inden kursusstart har erfaring med et objektorienteret programmeringssprog.
Dato for udarbejdelse	December 2025
Udarbejde af	Johnny Kure Jakobsen

Indhold i materialet:

Støttet af

BØRNE- OG UNDERVISNINGS-MINISTERIET
STYRELSEN FOR UNDERVISNING OG KVALITET

Formål

Formålet med kurset er, at deltagerne opnår forståelse for sikkerhed i softwareudvikling og sammenhængen mellem programmering, datahåndtering og sikkerhed.

Kurset giver deltagerne kompetencer til at identificere og forebygge almindelige sikkerhedsfejl i kode samt til at anvende kryptografiske mekanismer korrekt i udviklingsarbejde.

Deltagerne opnår indsigt i:

- Hvorfor kode bliver sårbar
- Hvordan input og data udgør en angrebsflade
- Hvordan kryptografi anvendes i praksis
- Hvordan software testes for sikkerhed

Efter kurset kan deltageren bidrage til udvikling af software, der anvender best-practice kryptografiske sikkerhedsalgoritmer og grundlæggende sikre programmeringsprincipper.

Målet med kurset er, at deltager kan:

- Redegøre for principperne bag kryptografiske hashing-funktioner, som eksempelvis SHA
- Redegøre for principperne bag symmetriske og asymmetriske krypteringsalgoritmer, som eksempelvis AES
- Redegøre for principperne bag authentication- og integrity-algoritmer anvendt i TLS/SSL
- Redegøre for forskellige former for cyberangreb, som eksempelvis SQL-injection og XSS
- Opsøge information om best-practice kryptografiske sikkerhedsalgoritmer hos troværdige tredjeparter
- Udvikle programmer, der anvender hashing, kryptering og authentication
- Udvikle scriptede tests, der tester applikationens sikkerhed
- Anvende netværksanalysator til at kontrollere, at kommunikation er sikker
- Anvende opnået viden i forbindelse med udvikling af sikker software

Støttet af
BØRNE- OG UNDERVISNINGS-MINISTERIET
STYRELSEN FOR UNDERVISNING OG KVALITET

Indhold

Kurset indeholder en blanding mellem teori og praktisk undervisning i følgende tematikker:

Introduktion til softwaresikkerhed

- Hvorfor kode bliver sårbar
- OWASP og typiske fejltyper
- Trusselsmodeller

Input, data og tillid

- Inputvalidering
- Sanitization
- Data som angrebsflade

Kryptografi i software

- Hashing (SHA)
- Symmetrisk kryptering (AES)
- Asymmetrisk kryptering
- TLS/SSL-principper

Sikker programmeringspraksis

- Secure coding principles
- Fejlhåndtering
- Logging uden informationslækage

Test og verifikation

- Scriptede sikkerhedstests
- Test for injection
- Netværksanalyse af krypteret trafik

Undervisningen er sprog-uafhængig i principper, men praktiske øvelser kan gennemføres i et valgt programmeringssprog.

Støttet af
BØRNE- OG UNDERVISNINGS-MINISTERIET
STYRELSEN FOR UNDERVISNING OG KVALITET

Aktiviteter

Dag 1 – Sårbar kode og trusselsforståelse

OWASP
SQL-injection
XSS
Analyse af kode

Dag 2 – Inputvalidering og datasikkerhed

Validering
Fejlhåndtering
Secure coding principles

Dag 3 – Kryptografi i praksis

Hashing
Symmetrisk og asymmetrisk kryptering
TLS-principper

Dag 4 – Implementering og test

Udvikling af sikker funktionalitet
Scriptede tests
Test for injection

Dag 5 – Verifikation og anvendelse

Netværksanalyse
Kontrol af TLS
Mini-projekt med sikker implementering

Kurset afsluttes med opsamling på læring, fælles refleksion og evaluering.

Deltagerne arbejder med at perspektivere softwaresikkerhed til egen udviklingspraksis og vurderer:

1. Hvordan sikkerhed kan integreres tidligt i udviklingsprocessen
2. Hvordan kryptografi anvendes korrekt
3. Hvordan test kan indarbejdes systematisk

Der lægges vægt på, at deltageren kan forbinde teoretiske sikkerhedsprincipper med praktisk programudvikling.

Støttet af
BØRNE- OG UNDERVISNINGS-MINISTERIET
STYRELSEN FOR UNDERVISNING OG KVALITET

Konkrete produkter

a. Pædagogiske overvejelser

Kurset er tilrettelagt med fokus på aktiv deltagelse og refleksion.

Da målgruppen har programmeringserfaring, lægges vægt på:

- Analyse af eksisterende kode
- Identifikation af sårbarheder
- Diskussion af konsekvenser
- Forbedring af kodeeksempler

Målet er at udvikle forståelse for sikkerhedsprincipper frem for blot at lære konkrete kodeopskrifter.

b. Didaktiske overvejelser

Undervisningen er opbygget med progression fra forståelse af sårbarheder til implementering af sikre løsninger.

PowerPoint anvendes til begrebsafklaring, mens hovedindlæringen sker gennem:

- Analyseøvelser
- Praktisk kodning
- Testscenarier
- Netværksverifikation

Der arbejdes bevidst med at koble teori om kryptografi til konkret kodeimplementering.

Støttet af
BØRNE- OG UNDERVISNINGS-MINISTERIET
STYRELSEN FOR UNDERVISNING OG KVALITET

c. Opgaver og cases

Formålet med opgaverne er at understøtte deltagernes forståelse af softwaresikkerhed og træne deres evne til at identificere, analysere og forebygge sikkerhedsfejl i kode.

Opgaverne anvendes både som læringsaktiviteter og som forberedelse til den afsluttende praktiske prøve.

Der arbejdes med progression fra analyse af sårbar kode til udvikling og test af sikre løsninger.

Opgave 1 – Analyse af sårbar kode (Dag 1)

Identifikation af SQL-injection og XSS.
Output: Analyse og risikovurdering.

Opgave 2 – Inputvalidering (Dag 2)

Forbedring af kode med korrekt validering.
Output: Forbedret kode og forklaring.

Opgave 3 – Kryptografisk implementering (Dag 3)

Implementering af hashing og kryptering.
Output: Fungerende kode og dokumentation.

Opgave 4 – Sikkerhedstest (Dag 4)

Udvikling af scriptede tests.
Output: Testscript og testresultater.

Opgave 5 – Integreret mini-projekt (Dag 5)

Udvikling af sikker kommunikation med TLS.
Output: Program, test og dokumentation.

Støttet af
BØRNE- OG UNDERVISNINGS-MINISTERIET
STYRELSEN FOR UNDERVISNING OG KVALITET

d. Praktisk prøve

Praktisk prøve – Procesprøve (Bestået / Ikke bestået)

Prøveform

- Individuel prøve
- Case-baseret
- Praktisk kodning
- Mundtlig redegørelse

Deltageren skal:

- Analysere sårbar kode
- Identificere sikkerhedsfejl
- Implementere sikker løsning
- Anvende hashing eller kryptering
- Udarbejde test
- Verificere sikker kommunikation
- Redegøre for valg af algoritmer

Hjælpemidler

- Egne noter
- Kursusmateriale
- Dokumentation fra troværdige kilder

Tidsforbrug

- Implementering og test: ca. 3 timer
- Mundtlig redegørelse: 10–15 minutter

Støttet af
BØRNE- OG UNDERVISNINGS-MINISTERIET
STYRELSEN FOR UNDERVISNING OG KVALITET

e. Bedømmelsesgrundlag

Deltageren vurderes på evnen til at:

- Identificere sikkerhedsfejl korrekt
- Implementere kryptografi korrekt
- Anvende sikre programmeringsprincipper
- Udvikle og anvende test
- Dokumentere og redegøre fagligt

Vurderingsskema (intern brug)

Vurderingsområde	Opfyldt	Delvist opfyldt	Ikke opfyldt	Bemærkninger
Sårbarhed korrekt identificeret	☐	☐	☐	
Input korrekt valideret	☐	☐	☐	
Hashing/kryptering korrekt anvendt	☐	☐	☐	
TLS korrekt implementeret	☐	☐	☐	
Test udviklet og anvendt	☐	☐	☐	
Dokumentation sammenhængende	☐	☐	☐	
Faglig redegørelse korrekt	☐	☐	☐	

Bestået forudsætter:

- Sårbarhed identificeret korrekt
- Kryptografiske principper anvendt korrekt
- Test dokumenteret
- Deltageren kan redegøre fagligt for valg og implementering

Hvis centrale sikkerhedsprincipper ikke forstås eller implementeres forkert, vurderes prøven som ikke bestået.

Støttet af
BØRNE- OG UNDERVISNINGS-MINISTERIET
STYRELSEN FOR UNDERVISNING OG KVALITET