Software-sikkerhed – Eksamensprojekt.

Dette dokument beskriver kravene til eksamensprojektet i faget Software-sikkerhed på professionsbacheloruddannelsen i it-sikkerhed, efteråret 2025.

Faglige krav til projektet.

Eksamensprojektet skal tage udgangspunkt i en analyse og løsningsforslag til forbedring af applikationens sikkerhed beskrevet i afsnittet *Den analyserede* Den analyserede applikation.

Analysen og forbedringsforslag skal tage udgangspunkt i de metoder og teori, der er blevet gennemgået i faget Software-sikkerhed. Emner fra faget *Software-sikkerhed* vægtes højest, men der kan også inddrages emner relateret til Software- sikkerhed fra andre fag, såsom *webapplikationssikkerhed*, *introduktion til IT-sikkerhed* og Sikkerhed i *IT-governance*.

Husk, at alt tekst bliver fortolket forskelligt – især metodebeskrivelser, krav og begreber. Brug vejledning til at få afklaret jeres forståelse tidligt i processen. 5 minutters vejledning tidligt i processen kan spare jer for 5 timers fejlretning senere.

Vejledende anvendelse af metoder og teorier

Projektets analyse og løsningsforslag bør tage udgangspunkt i de metoder og teorier, der er blevet gennemgået i undervisningen. Den følgende tilgang kan med fordel anvendes:

1. **Analyse af applikationen**
   Analysen kan med fordel tage udgangspunkt i den overordnede metode, der er beskrevet i dokumentserien *"Introduktion til software sikkerhed "*. Det indebærer først en klarlægning af systemmål, fx gennem identifikation af centrale brugstilfælde, der udgør systemets ønskede funktionalitet og dermed værdiskabelse. Herefter udledes sikkerheds mål som misbrugstilfælde – altså uønskede hændelser, der kompromitterer fortrolighed, integritet eller tilgængelighed i forhold til systemmålene. Hvert misbrugstilfælde analyseres med hensyn til risiko, ved at vurdere konsekvens og sandsynlighed, hvilket danner grundlag for prioritering. Endelig gennemføres en trusselsmodellering, hvor man identificerer angrebsflader og mulige angrebsvektorer, fx via STRIDE-modellen, og relaterer de identificerede områder til systemets tekniske arkitektur. Derudover kan der foretages en analyse af kodekonstruktionen og de anvendte praksisser. Statisk kodeanalyse og andre værktøjer kan desuden bidrage til analysen af applikationen. Denne analyse danner grundlag for valg af relevante løsningsforslag til forbedring af applikationen.

2. **Løsningsforslag og implementering**
   De valgte løsninger bør tage udgangspunkt i principperne fra *Secure by Design* (Johnsson et al.). Det kan indebære blandt andet anvendelse af domæneprimitiver til at sikre, at data er valide og konsistente i hele applikationens levetid, samt implementering af inputvalidering, fejltolerant adfærd (fail secure) og eksplicit styring af tilstande i systemet. Der kan desuden indtænkes automatiserede sikkerhedsaktiviteter i udviklingsprocessen, såsom statisk kodeanalyse, scanning for tredjeparts-sårbarheder og automatiserede tests i en CI/CD-pipeline og meget mere. Det anbefales,

at løsningsforslag og implementering kan kobles til specifikke krav i f.eks. *OWASP Application Security Verification Standard (ASVS)*, som supplement til mere generelle retningslinjer og kontrolkataloger.

3. **Dokumentation af ændringer og resultater**
   Løsningsforslagets implementerede ændringer og sikkerhedstiltag bør dokumenteres i rapporten og/eller i en tilknyttet kildekoderepository (fx GitLab eller GitHub). Repositoryet skal indeholde kode, CI/CD-konfiguration og eventuel yderligere dokumentation, og det skal refereres som kilde i rapporten på lige fod med anden anvendt litteratur.

Hvis det ikke er muligt at ændre i den originale kildekode (fx pga. tekniske eller praktiske begrænsninger), kan der i stedet udarbejdes et selvstændigt proof of concept, som illustrerer, hvordan de foreslåede ændringer kunne implementeres i praksis.

Bemærk: Ovenstående **tilgang er vejledende og ikke et krav**. Den skal ses som inspiration til, hvordan projektets analyse og løsning kan struktureres metodisk og praksisnært i overensstemmelse med fagets indhold og læringsmål.

Krav til projektrapportens struktur

Projektrapporten anvendes til at dokumentere projektets udførelse. Der er følgende krav til projektets overordnede kapitler og afsnit:

1. Forside
2. Abstrakt
3. Indledning
   3.1. Introduktion
   3.2. Baggrund
   3.3. Problemformulering
   3.4. Afgrænsninger
4. Metode og teori
   4.1. Definitioner & begreber *(Må gerne flyttes til indledningskapitlet)*
   4.2. Anvendt litteratur
   4.3. Projektets metode
   4.4. Anvendte metoder & teori
5. Analyse
   5.1. Problemanalyse
6. Løsningsforslag
   6.1. Overordnet løsningsforslag
   6.2. Implementering af løsningsforslag
7. Konklusion & anbefalinger
   7.1. Konklusion
   7.2. Fremadrettede anbefalinger
8. Kildehenvisninger og litteraturliste

Til de enkelte afsnits indhold, kan der finde vejledning i bogen *Projekter og rapporter på tekniske uddannelser* (Larsen, 2021) samt ved at spørge den vejledende underviser. Det er vigtigt at understrege, at

der ikke findes en enkelt korrekt løsning, da rapporter er individuelle argumentation, og læsbarheden samt formidling er derfor det der vægtes højst i rapporten.

Til kilde og litteraturhenvisning, skal *Harvard standarden* anvendes med *parentes henvisninger*. Beskrivelsen af denne kan findes her. Henvisninger til implementeringer i et offentligt tilgængeligt repository (eksempelvis Github eller Gitlab) bør medtages og behandles på lige fod med øvrige kildehenvisninger.

Projektrapportens struktur som angivet ovenfor skal overholdes i den afleverede rapport. **Manglende overholdelse af strukturen kan medføre, at rapporten afvises.**

Den analyserede applikation

Applikationen, der danner grundlag for analysen, kan findes på følgende URL-adresse på Github:

https://github.com/mesn1985/WebGoat.NET

Som udgangspunkt bør hver gruppe oprette sit eget offentlige repository med en kopi af applikationen, hvis der foretages ændringer i kildekoden, eller implementering af *proof of concept* eksempler.

Husk, at repository skal angives i litteraturlisten, og henvises i rapporten, på lige fod med øvrige henvisninger.

Vejledning under projektperioden

Der vil i løbet af projektperioden være afsat specifikke dage til vejledning som en del af undervisningen i faget *Software-sikkerhed*. **Det anbefales kraftigt, at grupperne benytter sig af disse vejledningstilbud** til at drøfte fremdrift, metodiske valg og afklaringer i forhold til kravene til eksamensprojektet.

Erfaringer fra tidligere semestre viser, at grupper som ikke gør brug af vejledningen, ofte har sværere ved at afgrænse deres problemstilling, dokumentere metodeanvendelse tilstrækkeligt og udarbejde en sammenhængende rapport. **Dette gælder især i projektets opstartsfase**, hvor mange grundlæggende beslutninger skal træffes. Vejledning tidligt i processen kan være afgørende for at få formuleret en klar og realistisk problemformulering samt for at planlægge en hensigtsmæssig arbejdsproces.

Som udgangspunkt tilbydes der kun vejledning på de planlagte vejledningsdage. Der kan derfor ikke forventes individuel vejledning uden for disse tidspunkter, medmindre andet er aftalt med underviseren på forhånd. Det er gruppens ansvar at forberede spørgsmål og materiale til vejledningen, så tiden anvendes bedst muligt.

Eksamensbeskrivelse

Eksamen er beskrevet i den institutionelle studieordning, afsnit 3.2.7 og er en mundtlig eksamination der tager udgangspunkt i den obligatoriske skriftlige rapport. Den studerende skal til eksamen lave en 10 minutters mundtlig præsentation af projektet, og eksamineres herefter af eksaminator i 10 minutter.

Udover kravene beskrevet i eksamensbeskrivelsen, skal rapporten følge kravene for skriftlige eksamener opstillet i den institutionelle studieordning, afsnit 3.1.5.

Bedømmelsen afgives ud fra 7-trinsskalaen, og er en helhedsvurdering af den studerendes opfyldelse af fagets læringsmål ud fra den skriftlige rapport, mundtlige præsentation samt besvarelse af spørgsmål under eksaminationen.