Universitetet i Oslo
IN2120 Cybersikkerhet
Høst 2026
Teoretiske oppgaver – løsningsforslag

Teori del 5: Pentesting og angrepsmetodikk

Oppgave 1: Juridiske og etiske rammer for pentesting

a. *Hva skiller en pentester (white hat) fra en ondsinnet hacker (black hat), sett i lys av motivasjon?*
Motivasjonen: black hat vil ha uautorisert tilgang for å lese, manipulere eller ødelegge/kryptere data, eller utnytte ressurser – ofte opportunistisk. Pentesteren har tillatelse og vil avdekke og fjerne sårbarheter for å sikre systemet.

b. *Hvorfor er skriftlig tillatelse på forhånd avgjørende for en lovlig pentest? Hva kalles denne tillatelsen i dagligtale, og hvilken lovbestemmelse gjør uautorisert datainnbrudd straffbart?*
Uten tillatelse er datainnbrudd straffbart etter strl. § 204 (bot eller fengsel inntil 2 år). Skriftlig, forhåndsinnhentet tillatelse fra noen med myndighet dokumenterer at tilgangen er autorisert – kalles gjerne et «get out of jail free card».

c. *Forklar legalitetsprinsippet slik det gjelder for en privatperson kontra en ansatt i forvaltningen, og hvorfor dette er relevant når en offentlig virksomhet skal pentestes.*
Privatperson: alt er lov med mindre eksplisitt forbudt. Forvaltningsansatt: alt er forbudt med mindre eksplisitt hjemlet (legalitetsprinsippet, jf. Grunnloven). Tillatelsen til pentesten må derfor være forankret i et gyldig vedtak/fullmakt fra noen med reell myndighet.

Oppgave 2: Hvit-, grå- og svartbokstesting

a. *Forklar forskjellen mellom hvitboks-, gråboks- og svartbokstesting.*
Svartboks: ingen forhåndsinformasjon om systemet. Hvitboks: all relevant informasjon gis på forhånd. Gråboks: en mellomting, hvor pentesteren får noe informasjon.

b. *Diskuter minst to fordeler og to ulemper ved hvitbokstesting sammenlignet med svartbokstesting.*
Hvitboks: mer effektiv og bedre planlagt (mindre risiko), finner ofte flere avvik – men lett å bli «blind» på informasjonen, og mindre realistisk sammenlignet med en reell angriper. Svartboks: enklere å selge inn og gir «nye øyne» – men tar mer tid og finner ofte færre feil.

c. *Du er sikkerhetsansvarlig i en bedrift med begrenset budsjett og stramme tidsfrister. Hvilken testtype vil du anbefale, og hvorfor?*
Hvitboks (evt. gråboks), siden man slipper å bruke tid/budsjett på ren rekognosering og får mer testdekning per time.

Oppgave 3: Stadiene i en pentest

a. *List opp og forklar kort stadiene i en pentest, fra planlegging til rapport.*
Planlegging → rekognosering → innledende tilgang → utvide tilgang → sikre fotfeste → gevinst → rapport.

b. *Hva er OSINT, og gi tre eksempler på kilder en pentester kan bruke i rekognoseringsfasen.*
OSINT = rekognosering via åpne, offentlig tilgjengelige kilder. Eksempler: søkemotorer/nettsider, LinkedIn/stillingsannonser, sosiale medier og Google Maps.

c. *Forklar hvorfor en pentester ofte utfører mesteparten av testen med vanlig brukertilgang, fremfor å bruke lang tid på å skaffe seg mer tilgang selv.*
Initial tilgang er den vanskeligste og mest tidkrevende delen. Siden testen er kort (dager–uker, ikke måneder/år som en reell angriper), gis brukertilgang tidlig for å spare tid og finne flere reelle feil.

Oppgave 4: Passordangrep

a. *Forklar forskjellen mellom passordspraying og et vanlig bruteforce-angrep mot én bruker. Hvorfor er passordspraying ofte tryggere for en pentester å utføre, med tanke på kontosperring?*
Spraying: ett passord mot mange brukere. Bruteforce: mange passord mot én bruker – trigger fort kontosperring. Spraying holder forsøk per bruker lavt og unngår dermed lockout.

b. *Hva menes med «pass the hash», og hvorfor er dette et argument mot å gjenbruke (gode) passord?*

Pass the hash = autentisere med selve passordhashen, uten å kjenne klartekstpassordet. Selv sterke passord hjelper ikke hvis hashen stjeles og gjenbrukes på andre maskiner – derfor bør man ikke gjenbruke passord/hasher.

c. *Et passord består kun av 8 små engelske bokstaver (26 tegn). Regn ut hvor lang tid et uttømmende søk i snitt vil ta ved 14 milliarder forsøk per sekund. Hvorfor er reelle passord i praksis mye raskere å knekke enn den teoretiske kompleksiteten skulle tilsi?*
$26^8 \approx 208{,}8$ mrd. Snittid = $(26^8/2) / (14 \cdot 10^9) \approx 7{,}5$ sekunder. Reelle passord knekkes enda raskere fordi folk følger forutsigbare mønstre (ord + tall + tegn), og knekkeverktøy som Hashcat bruker ordlister/regelbaserte angrep i stedet for rent uttømmende søk.

Oppgave 5: Fra sårbarhet til gevinst

a. *Forklar forskjellen mellom å kartlegge sårbarheter (f.eks. med OpenVAS) og å utnytte dem (f.eks. med Metasploit).*
Kartlegging (OpenVAS) identifiserer mulige svakheter, f.eks. via versjonsnumre på tjenester. Utnyttelse (Metasploit) kjører faktisk exploit-kode for å utnytte svakheten og skaffe tilgang.

b. *Hva er «lateral movement», og hvorfor foretrekker en pentester ofte dette fremfor å kjøre nye exploits på hver enkelt maskin?*
Å gjenbruke tilganger (brukernavn/passord, AD- eller gruppemedlemskap) for å bevege seg mellom maskiner, fremfor å kjøre nye exploits – reduserer risikoen for driftsforstyrrelser.

c. *Hva er fuzzing, og hvorfor beskrives det som den eneste praktiske måten å finne feil som buffer overflow?*
Fuzzing = sende uventet/ekstrem input for å avdekke feil («jakt på feilmeldinger»). Slike feil krever systematisk utprøving av et enormt antall input-varianter, noe bare automatisert fuzzing i praksis klarer.

Oppgave 6: Verktøy i pentesting

a. *I foilsettet nevnes blant annet nmap, Wireshark/tcpdump, Hydra og Metasploit. Forklar kort hva hvert av disse verktøyene brukes til i en pentest, og i hvilken fase de typisk brukes.*
nmap: portscanner som kartlegger maskiner, IP-adresser og åpne tjenester/porter (rekognosering). Wireshark/tcpdump: lytter på og analyserer nettverkstrafikk, gir kjennskap til nettverket (rekognosering/MiTM). Hydra: prøver ordlister/bruteforcer passord mot innloggingstjenester for å skaffe initial tilgang. Metasploit: rammeverk for å utnytte kjente sårbarheter og få tilgang (utnyttelsesfasen).

b. *Hva er forskjellen mellom John the Ripper og Hashcat til passordknekking, og når vil du velge det ene fremfor det andre?*
John the Ripper er raskest på uttømmende søk og enklest å bruke. Hashcat er best på regelbasert/maskebasert knekking og gir større kontroll, men er mer komplisert. Velg JtR for rask, enkel knekking; Hashcat når man vil utnytte kjente passordmønstre for å knekke raskere.

c. *Verktøy som Mimikatz og Impacket/psexec brukes til henholdsvis passorddumping og «pass the hash». Hvorfor krever slike verktøy typisk administrator- eller debug-rettigheter, og hvorfor er dette en risiko dersom en angriper får slike rettigheter ett sted i nettverket?*
De henter ut klartekst-/hashpassord fra systemets minne, noe som krever utvidede rettigheter for å lese. Får en angriper admin-/debugrettigheter på én maskin, kan de dermed hente ut creds/hasher der og bruke disse til lateral movement videre i nettverket.