IN1020: Løsningsforslag uke 47 (18.-22. november 2019)

Oppgave 1:

a) Snike seg inn etter andre ("tailgating"). Bære store esker og og få hjelp til å åpne døra. Produsere og benytte et falskt tilgangskort.

b) Sende tilpasset phishing-epost med skadevare vedlagt Sende tilpasset phishing-epost med lenke til webside som inneholder en exploit for en zero-day-sårbarhet som nnes på direktørens maskin.

Oppgave 2:

a) Phishing-angrep utnytter menneskelige svakheter: Uvitenhet, godtroendehet, mangel på bevissthet rundt farene ved også digital svindel.

b) Sikkerhetstiltak for å forebygge phishing angrep: Opplæring, slik at folk kan klare å identisere potensielle farer, f.eks. personer som utgir seg for å være legitime personer, og falsk epost. Praktiske sikkerhetstiltak kan være god ltrering av epost, benytte epost-autentisering, samt forsøke å avdekke og deretter tydelig varsle om falske tjener-sertikater.

Oppgave 3:

Se avsnittet om "Metadata i filer" s. 121 i Heide/Nätt.

a) Tips: Kommandoen på en Unix-maskin. Google maps el. for å plassere GPS-koordinatorer på kartet.

b) Informasjon om enheten benyttet til å ta fotoet, inkludert OS-versjon. Er dette noe som potensielt kan utnyttes? Tidspunkt bildet er tatt.

Oppgave 4: Kryptografi

a) Sikre trygg lagring av data i utrygge lagringsenheter, sikre trygg overføring av data i f.eks åpne nett.

b) Tapt eller kompromittert nøkkel gjør det mulig for en angriper å bryte konfidensialitet, integritet og/eller ektehet for informasjon som er beskyttet av denne nøkkelen. En trygg administrasjon av nøkler som beskytter mot kompromittering eller tap av nøkkel er derfor en faktor som i stor grad påvirker styrken i en kryptografisk sikkerhetsløsning.

c) Hovedformålet med PKI er å sikre autentiske offentlige nøkler. PKI er et rammeverk for å legge til rette for å binde en offentlig nøkkel til en navngitt enhet, og denne bindingen kan bekreftes av en betrodd myndighet som utsteder et sertifikat. Et sertifikat forteller/bekrefter følgende: "Offentlig nøkkel K eies av X".

Oppgave 5: Kryptering

Kryptering

- Anne skriver en melding til Per.
- Anne krypterer meldingen ved å benytte den symmetriske nøkkelen de har utvekslet
- Anne sender den krypterte meldingen til Per.

Dekryptering

- Per mottar kryptert melding fra Anne
- Per dekrypterer meldingen ved å bruke den symmetriske nøkkelen de har utvekslet
- Per leser den dekrypterte meldingen

Oppgave 6: Kryptering

a) Ved digital signatur benytter avsender sin private nøkkel til kryptering, mens mottager benytter avsenders offentlig nøkkel til dekryptering.

b) Ved kryptering av innhold i en melding benytter avsender mottages offentlige nøkkel til kryptering av meldingen, hvorpå meldingen så kun kan dekrypteres ved å benytte mottagers private nøkkel.

Oppgave 7: Case-oppgave

Mulige scenarier: Brukere havner på en falsk nettside (følger en falsk lenke, Googler bankens navn istedenfor å huske adressen el.). Bruker må selv vite forskjell på www.dnb.no, www.d-nb.com, d-nb.org, osv. TLS-basert autentisering er syntaktisk, og gir ikke brukeren bevis for at en nettside er ekte. Bruker benytter et åpent trådløst nett: DNS-forfalskning i et kompromittert eller falskt aksesspunkt kan også sende brukeren til en falsk nettside.

Oppgave 8: Case-oppgave

Nei. Kanksje brukeren har forlatt maskinen sin for å få en kaffe eller gå på toalettet? Om maksinen da ikke er låst kan en annen person bruke denne, utgi seg for å vÃ¦re brukeren og sende data til tjeneren. En annen mulighet er at brukerens datamaskin er infiert med en trojaner som genererer og sender data til serveren uten at brukeren vet om det (til tross for at brukeren fysisk sitter foran datamaskinen og aktivt utfører transaksjoner, f.eks. til en nettbank).

Hvis sesjonen mellom klient og server ikke er beskyttet med TLS (Transport Layer Security) eller f.eks. VPN (Virtual Private Network), kan sesjonen også vÃ¦re et offer for et Man-in-themiddel-angrep. I et Man-in-the-middle-angrep bryter uvedkommende inn i kommunikasjonen og kan endre eller slette data utvekslet mellom klienten (brukerens datamaskin) og tjener

Oppgave 9: Sjekksumalgoritme (hash-funksjon)

a) Nei, sjekksummen blir en annen når fila endres. Dette er en grunnleggende egenskap ved sjekksumalgoritmer. Men observer gjerne at sjekksummen blir den samme når du bruker programmet sha256sum på samme fil flere ganger.

b) Man kan sammenligne sjekksum av en datafil opp mot en kjent sjekksum av denne fila. Er den lik? Isåfall er fila ikke endret mellom første og andregangskjøring av sjekksum.

c) Sjekksumalgoritmer brukes ofte ved deling av programvare (f.eks. nedlasting fra nett). Tilbyderen av programvare genererer en sjekksum av orignalen (ofte et filarkiv (f.eks. .zip-fil), som publiseres sammen med programvaren. Brukere som laster ned programvare kan enkelt selv kjøre en sjekksumalgoritme etter nedlasting, og kan med det finne ut om programvaren det den utgir seg for å vÃ¦re, eller er den kanskje endret etter at sjekksummen ble generert?

Vil også kunne avdekke hvorvidt en nedlasting har gått greit eller ikke (pakketap?). Helt konkret eksempel: https://www.postgresql.org/ftp/source/v12.0/, nedlasting av kildekode PostgreSQL databaseserver. Her finner du både programvaren og tilhørende sjekksummer for både MD5 og SHA256.

Oppgave 10: SSL

Praktisk oppgave uten løsningsforslag.

Oppgave 11: Tilgangskontroll i Unix

a) og b) Praktiske oppgaver uten løsningsforslag, men:

- Kommandoen ls -l <mappenavn> viser deg innholdet i mappen, inkludert hvilke rettighetsbit som er satt på en l eller undermappe, hvilken bruker som eier de ulike ler og mapper, osv.
- Kommandoen ls -ld <mappenavn> viser deg informasjon om mappen selv, dvs. hvilke rettighetsbit som er satt og hvilken bruker som eier mappen. Noen observasjoner:
- Du kan ikke se (lese) innholdet i en mappe uten å lesetilgang (r-bit satt).
- Hvis eksekveringsrettigheter (x-bit) ikke er satt på en mappe kan du ikke eksekvere innholdet i den, dvs. heller ikke se innholdet eller lese en l som ligger i mappen (uavhengig av hvilke retigheter som er satt på la).
- Du må ha skrivetilgang (w-bit satt) for å kunne skrive i en mappe.

c) Kondensialitet: Man kan bidra til å sikre kondensialitet ved at kun de brukerne som skal ha tilgang gis tilgang. Dette kan styres ved å sette lesetilgang (r-bitet) riktig. Ved å kun sette eksekvering-tilgang (x-bit) på en mappe kan heller ikke innholdet i mappen listes, mens en l som ligger i mappen nt kan både leses og eksekveres (bruker må dog kjenne full sti (path) til la). Dvs. at kun de som har fått kjennskap til full sti har tilgang (skjule informasjon, Security by obscurity). Dette kan være nyttig hvis man ønsker å gi en bruker tilgang til enkeltelementer (en bestemt l/et program), samtidig som man fortsatt vil skjule alt annet som ligger under samme mappe.

Dataintegritet: Ved å styre og begrense hvem som kan skrive til mapper og ler i lsystemet bidrar man til å sikre at uvedkommende ikke kan endre innholdet i ler.