Oppgave 1

Informasjonssikkerhet er det vi snakker om når det kommer til å holde informasjon sikker. Det kan være å ha en hemmelig familieoppskrift som ikke deles med naboen, eller koder til atomvåpen fordelt på flere personer som sammen må finne den virkelige koden.

Siden det finnes forskjellig nivå av fare hvis informasjonen kommer ut, må man også ha forskjellige nivåer med sikring av informasjonen. Det er kanskje ikke så farlig hvis storesøster smugleser en dagbok, men det kan ha store konsekvenser hvis navn på spioner lekkes til feil land. Vi kan dermed ta utgangspunkt I CIA-modellen for å beskrive hvordan informasjonssikkerheten skal fungere best mulig.

CIA-modellen:

Confidentiality

Informasjonen skal være konfidensiell for de som ikke skal ha tilgang til den. Det går på å beskytte data mot uvedkommende. Det man hovedsakelig gjør for å holde datainformasjon konfidensiell er å kryptere data som sendes. Det finnes forskjellige måter å kryptere informasjonen på, men hovedsakelig består det av en melding, en kryptert melding, og en nøkkel som kan brukes for å "låse" opp meldingen til klartekst.

Man kan også holde informasjon konfidensiell ved å begrense tilgang, eller gi tilgang på informasjon til brukere som skal ha det. Dette kan gjøres ved å gi roller til personer eller datasystemer, som igjen får tilgang (eller mister tilgang) til spesifikke områder. Man kan også sikre informasjonen fysisk, men låser, lyddemping, og rom som stenger signaler ute.

En tredje måte å sikre konfidensialitet er å ha verktøy som kan bevise (til stor grad) noens identitet. Vi kan dele dette I tre:

Noe en person **har**:

Nøkler, mobil, adgangskort, dongel, fysisk kodegenerator (f.eks Bank ID-brikke)

Noe en person **vet**:

Passord, ukjent informasjon

Noe en person **er**:

Fingeravtrykkavlesning, iris-scanning, annsiktsscanning, DNA-test

Integrity

Integritet betyr at informasjonen ikke blir endret av en person som ikke skal ha tilgang på informasjonen. Hvis man sender en mail fra A til B, må man vite at det ikke er en tredje person som kan se, lese, og en endre på epostinnholdet I transitt. Her bruker vi ofte sjekksummer som ser om dataen som sendes, er lik dataen som kommer frem. Hvis innholdet er endret på veien, vil sjekksummene være ulike. Man kan også ta backups for å følge med på endringer som er gjort mellom hver gang.

Availability

Det siste punktet I CIA-modellen er tilgjengelighet. Det vil si at de som skal ha tilgang på dataen, skal få tilgang, og ha mulighet til å endre, på informasjonen de skal ha tilgang på. Hvis et dokument skal signeres, så skal det være mulig uten at noen overfører dokumentet på en harddisk, legger den I en safe, som så sendes til mottaker, som igjen får koden til safen en uke senere I en separat konvolutt. Sikkerhet og integritet skal ikke stikke kjepper I hjulene for arbeid, bare fordi det er mest sikkert.

Oppgave 2

Det finnes mange trusler I dag. Mange ser for seg at det sitter er hacker som spesifikt går etter din personlige datamaskin eller mobil, og gjør alt de kan for å komme seg inn. I realiteten er den største truslen menneskelige feil. Passord og sikkerhet har muligheten til å være så sikre at det vil ta tiår og komme seg inn - og det er hvis man kun prøver å komme inn på den maskina. De største truslene jeg som privatperson har, er egen uaktsomhet, eller dårlig hell. Siden jeg deler nettverket mitt med en samboer, kan jeg også være uheldig og få malware gjennom nettverket fra hans side.

Det er ofte lite som skjer med en privatperson med mindre man selv setter seg i en situasjon man blir et offer. Det kan være ved å trykke på linker som folk sender deg uten å være sikker på hva det er, eller å skrive passord ned på et sted andre har tilgang. Som oftest vil informasjonssikkerheten ligge i hendene til store aktører som ikke nødvendigvis følger alle retningslinjene – jeg fikk selv mail for noen uker siden om at NAV hadde brutt GDPR og sendt ut CV'en min uten å ha lov, og uten at det fikk noen konsekvenser for deres del. Jeg kan også følge med på nettsider som finner ut hvilke nettsider som har fått passordene lekket – og det er ikke få. Med en gang en ekstern aktør har tilgang til informasjonen min, må jeg akseptere at det er en mulighet for at noe kan skje med den. Det er derimot noen ting man kan gjøre for å gjøre sin egen hverdag litt sikrere, og minimere risikoen for at man selv har skyld i lekking av informasjon.

Noen ting man kan gjøre for å øke egen sikkerhet:

- Ha lange passord, de må ikke nødvendigvis være kompliserte med tall og tegn.
- Two Factor Authentication. Hvis man har to måter for å bekrefte identitet, er man sikker selv om en av metodene blir brukt av en annen person.
- Ikke skriv ned passordene et sted andre har tilgang. En post-it under laptopen er ikke sikkert, selv med en annen post-it over for å skjule passordet (dette skjedde faktisk på jobb).
- Ikke trykk på linker du får på epost, med mindre du vet det er en sikker kilde. Sjekk avsender. Hvis du får mail fra en bank eller annet "sikkert" nettsted, er det like greit å gå direkte til nettsiden deres og logge inn, Istedenfor å trykke på linken.
- Ikke bruk samme passord alle steder. Det er spesielt viktig å ha unikt passord for tjenester som har tilgang til kort- og/eller personlig informasjon, eller informasjon som brukes for identifisering, slik som bank, e-post, og mobiltelefon.
- Man kan bruke VPN for å skjule nettverksinformasjonen sin for nettsider som ikke trenger tilgang til informasjonen.
- Man har aldri vunnet en iPhone ved å gå inn på en nettside.

Oppgave 3

Det finnes to hovedtyper med malware - programmer som er laget for å ta over eller bruke en uvitende persons datamaskin. Man omtaler som oftest malware I dagligtalen som "datavirus", men det er egentlig en samling av forskjellige typer skadelig programvare. Malware brukes ofte for å få sensitiv informasjon, penger, "hacktivism", spionering eller annen data som kan være relevant for personen(e)

Den ene typen malware er et faktisk datavirus. Virus kommer som oftest skjult I en fil (vert) som en bruker må kjøre (execute) for at maskinen skal bli infisert. Virus fungerer ved at den infiserte koden kobler seg på fungerende kode. Virus kan ikke kjøre eller spre seg videre uten at en bruker aktivt gjør noe, da man aktivt må kjøre koden for at maskinen blir infisert. Noen vanlige måter man kan få virus på er ved å laste ned programmer og filer fra utrygge sider, eller åpne vedlegg fra ukjente epostadresser.

En annen type malware er dataormer. I morsetning til datavirus, så må ikke en dataorm spre seg fra en vert. Dataormer inneholder nemlig kode for å spre seg videre på egenhånd, og kan ofte bevege seg gjennom nettverk. Hvis en dataorm først har klart å komme inn I nettverket, vil den klare å komme seg videre uten problem. Dataormer kan komme på en PC på forskjellige måter, USB og CD, linker, programmer og nettsider er bare noen av dem. Ofte benytter dataormer seg av utdaterte operativsystemer og mangel av antivirus på maskinene. Dataormer kan duplisere seg selv og sprer seg fort på nettverket.

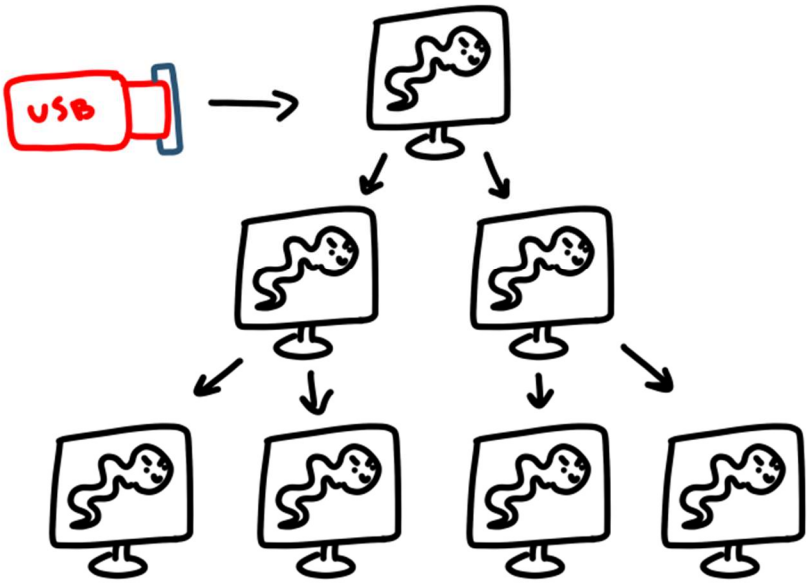
Det finnes flere typer ormer, noen eksempler kan være

Epost-orm: Sender automatisk email til alle på epost-listen din, gjerne med en link eller vedlegg som infiserer videre.

Messenger-orm: Sender melding til alle på vennelista, gjerne med en click-bait link for å infisere videre.

Internett-orm: Ligger I nettsidens HTML, så når man laster siden vil maskinen bli infisert.

Nettverk-orm: Ligger ofte I nettverkspakker, men kan også spre seg til enheter på samme nettverk.



Med tanke på at mer og mer blir heldigitalt, står vi forran en stor "katt og mus"-situasjon når det kommer til malware, og bekjempelse av disse. Vi så allerede I 2010 en dataorm, Stuxnet, som hadde I oppdrag å ødelegge atomreaktorene I Iran, og som I en viss grad var vellykket I dette. Malware har mulighet til å infisere, overta, og endre dataprogrammer, så alt som er bygget på et digitalt nettverk er I risikosonen. Hvis noen klarer å infiltrere vannprossesering, kan samfunn kollapse I løpet av få dager. Det er dermed viktig å ha opplæring av etiske hackere, som aktivt jobber for å tette sikkerhetshull før de blir funnet og misbrukt.

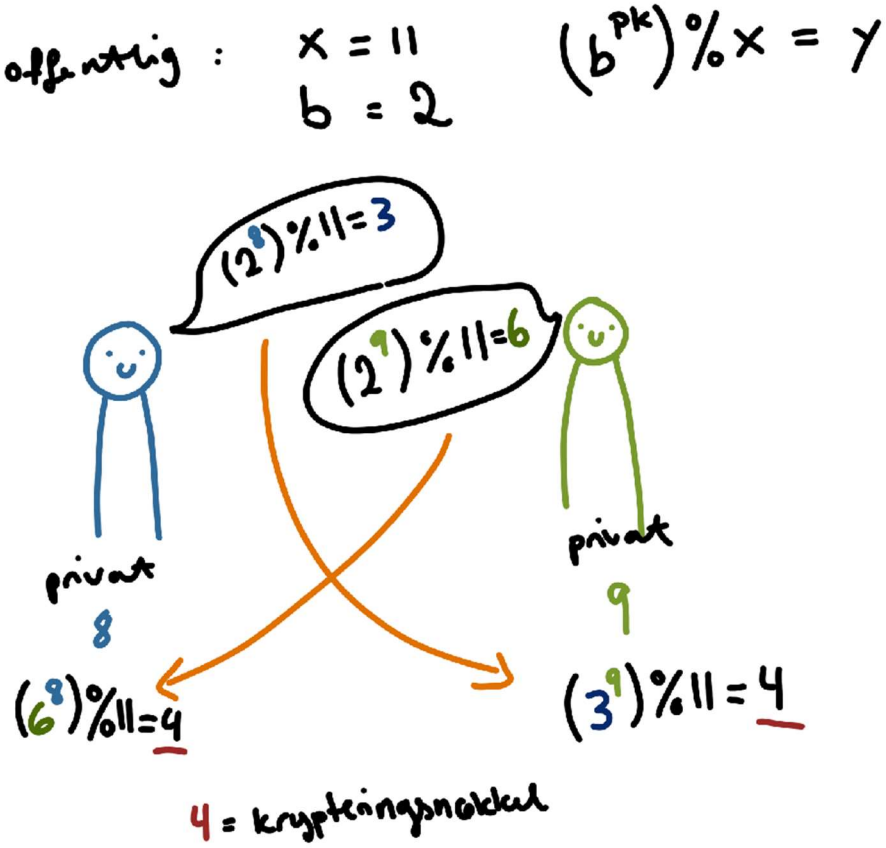
For å beskytte oss mot fremtidens datavirus er det også viktig at man styrker kunnskapen om informasjonssikkerhet blandt den generelle befolkningen. Et nettverk er så trygt som den svakeste linken, og mennesklige feil er den største sikkerhetstrusselen vi har. Mange får høre at man må ha et sterkt passord, og ikke trykke på linker man ikke vet hva er - men man ser også daglig på sosiale medier folk som sender dataormer nettopp fordi de trykket på "JEG KAN IKKE TRO AT DETTE ER DEG?!!"-linken.

Oppgave 4

Asymmetrisk kryptering er når krypteringen benytter både privat og offentlig nøkkel for å dekrypteres. Som oftest tar man utgangspunkt I det som kalles enveisfunksjoner, som er funksjoner der man kan finne a gitt b, men ikke b gitt a. Hvis man har regnestykket (1123^11)%17, så kan man regne seg frem til at resultatet er 1. Får man derimot oppgitt tallet 1, finner man ikke tilbake til basetallet 1123 uten å drive med faktorisering av tall. I 2021 tar disse regneopperasjonene (faktorisering) såpass lang tid at krypteringen regnes som trygg - men dagen man klarer å knekke koden for utregning, faller det sammen.

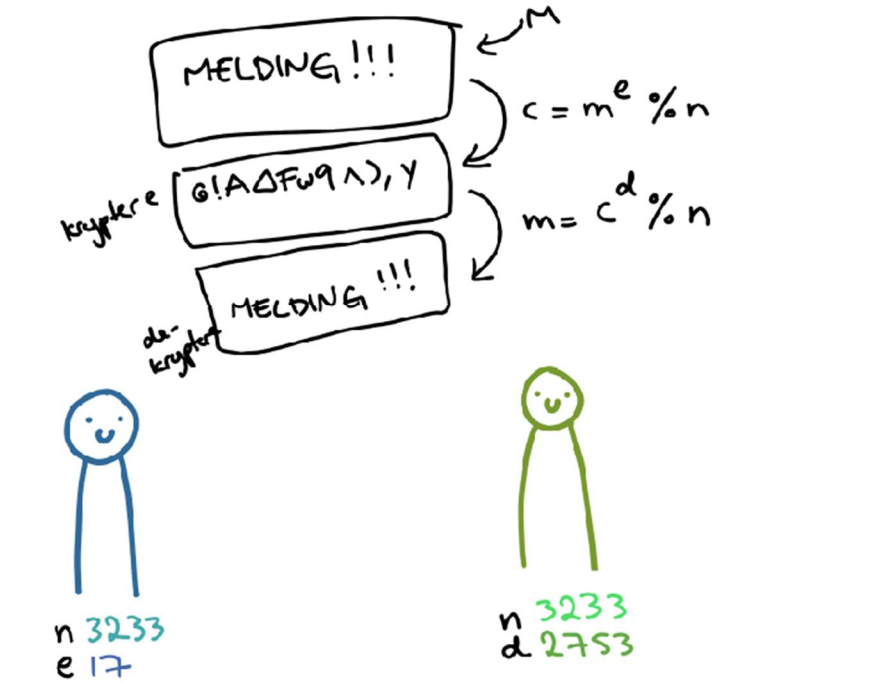
Diffie-Hellman

Diffie-Hellman er ikke en krypteringsmetode, men en metode å utveksle krypteringsnøkler. Den foregår på måten at man har en offentlig modulus (x) som må være et primtall, en offeltig base (b), og et privat tall. Hver bruker tar basen opphøyd I sitt private tall, og tar moduloen av det offentlige modulo-tallet. De får ta et tall (y) som de sender til den andre personen. Den andre personen vil da kunne ta tallet den fikk, og bruke det som base I samme regnestykke. Resultatet vil være likt hos begge, og det felles tallet er da krypteringsnøkkelen.



RSA

RSA bruker en lignende metode som Diffie-Hellman, men har klart å videreutvikle det til å bli en krypteringsmetode, og ikke bare utveksling av krypteringsnøkler. RSA fungerer på måten at man har en melding (m), public nøkkel, og public encryption-nøkkel. For å kryptere meldingen brukes formelen c = (m^e)%n, som vil si at hvert tegn I meldingen endrer verdi. Siden tegnene endrer verdi, vil meldingen bestå av nye tegn, og være uleselig. Når meldingen skal dekrypteres, trenger man en privat nøkkel, samt en dekrypteringsnøkkel. Man kjører det gjennom formelen m = (c^d)%n, og meldingen vil da returnere til samme verdi som før.



Oppgave 5

Internett (world wide web) er den største plattformen for sikkerhetstrusler innen datasikkerhet. Det er der malware spres hyppigst, og stedet med flest angrep. Siden internett ikke er bygget på en måte som ivaretar sikkerheten, så er det opp til hvert enkelt selskap eller person å tilrettelegge for datasikkerhet selv. Dette kan være alt fra å hashe passord til å sikre input-felter for cross site scripting. Brukeren selv må også ta forhåndsregler når det kommer til nedlastning av programvare og filer, eller klikking på linker.

En av grunnene til at internett er så usikkert I dag, er at det opprinnelig var interne nettverk som var koblet sammen - og dermed hadde lite påvirkning fra utsiden. Når man satte opp et

verdensdekkende internett, fortsatte oppbyggingen som om det ikke var fiendtlige brukere på nettverket.

TCP/IP-modellen har som oppdrag å overføre data I riktig rekkefølge, til riktig maskin. Det ligger ikke noe innebygget sikkerhet I TCP/IP, og kan dermed misbrukes. Siden TCP bruker en "three way handshake", som vil vente på svar fra klienten, så kan man kjøre et DoS-angrep. Hvis klienten aldri sender en ACK-melding tilbake, så vil den bli hengende. Skjer dette nok ganger, vil alle portene bli brukt opp, og applikasjonen kan ikke ta i mot flere henvendelser.

Man har også et angrep som heter «Session Hijacking». Det vil si at en person har etablert en kobling mot en server, og angriperen overtar sesjonen – så TCP vil sende pakkene til angriperen samtidig som den tror det går til offeret. Angriperen må da finne ut sekvensnummeret og ACK-nummeret til brukeren.

For at man skal sikre seg for angrep, så må man gå til verks. Siden det ikke ligger noen grunnleggende sikkerhet på selve internettet, så må enten brukeren, eller sideleverandøren stå for dette selv – gjerne i en kombinasjon. En bruker kan velge å bruke VPN for å skjule nettverksaktiviteten for en leverandør, samt være påpasselig med linker som blir klikket på og filer som lastes ned. Man må også være bevisst på hvem man gir informasjonen sin til. Leverandøren må passe på at ingen kan hente ut sensitiv informasjon eller få tilgang på data de ikke skal ha. De må også passe på at ingen har mulighet til å endre på- eller legge til kode som ikke skal være der. Det finnes også flere programmer, som antivirus, som hjelper med trygg surfing på nett.

Oppgave 6

Under en penetrasjonstest skal man prøve å finne muligheter å hacke seg inn på et system på en etisk måte - det vil si at man leier inn, eller ansetter, noen som skal finne sikkerhetshull slik at selskapet kan tette dem. Man skal altså FINNE sårbarhetene, ikke utnytte dem. Det følger en lignende formula man ville brukt for å faktisk hacke selskapet. For at den etiske hackeren ikke skal gjøre noe straffbart, må det være en juridisk avtale mellom personen og bedriften

Man bruker ofte samme fremgangsmåte for å finne svakhetene i systemet, som en hacker ville brukt for å komme seg inn i systemet.

Steg 1: Samle Informasjon

I denne oppgaven får vi beskjed om at de bruker en frontend-portal som kommuniserer med REST API og en SQL-server. Det finnes mange verktøy man kan bruke for å finne mer informasjon om systemet og selskapet. Det kan være mot software og hardware, eller mot personene som jobber med programmene. Det kan ofte være lettere å lure menneskene enn å komme seg rundt databeskyttelsen.

Steg 2: Scan IP-adresser og OS «fingerprint»

Ved å finne IP-adressene på nettverket, kan man finne ut om det er noen enheter som det kanskje er lettere å komme seg inn på enn andre. Det finnes verktøy man kan bruke som skanner etter tilkoblinger og porter som blir benyttet, og hva slags operativsystem som kjøres -samt hvilken versjon. Dette kan være essensielt for å vite hva slags kjente sikkerhetshull systemet har.

Steg 3: Identifiser sårbare tjenester

For å finne sårbarheter kan man bruke programmer som skanner ip-adressen, og den oppgir alle sårbarheter som finnes på nettverket.

Siden det er en nettside vi skal se etter sårbarheter i, kan vi bruke sslabs.com og securityheaders.com. Disse vil sjekke sårbarheten på nettsiden og gi deg en rating på hvor sikker nettsiden er.

Det er en rekke sårbare områder i en webapplikasjon som kjører mot et API og en SQL-server. Noen av de vi må passe ekstra godt på er:

Validering av input-data

Hvis en bruker kan skrive inn hva som helst i input-felter, så kan de også skrive ting som kan ødelegge eller korruptere nettsiden, eller kjøre egen kode via feltene. Dette kan man gjøre

ved å kun la noen skrive bokstaver i et «navn»-felt, siffer i et «nummer»-felt, eller ha regler for hvordan en epostadresse skal se ut.

Skriv kode basert på hva en bruker skal få lov til

Det å bruke whitelisting som base for hvordan man setter opp logikken, gjør at man har full kontroll over det brukeren får lov til å gjøre. Det vil si at man spesifikt sier man skal skrive bokstaver i et «navn»-felt, istedenfor å si at man ikke får skrive siffer eller spesialtegn.

XSS (cross site scripting)

Cross site scripting betyr at en angriper kan kjøre sin egen kode på din nettside. Dette må man ta høyde for, og stoppe det før det skjer.

SQL-injections

SQL-injections gjør at en angriper kan hente ut informasjon fra din database uten å ha tilgang. Dette kan angriperen gjøre ved å skrive kode som henter ut tabeller fra input-felt man vet er koblet mot en database.

Hashing / kryptering av sensitiv informasjon

Hvis man glemmer å hashe eller kryptere sensitiv informasjon, kan man plutselig få passord oppgitt i klartekst.

Brute force angrep

Hvis noen prøver å komme seg inn på nettsiden ved å taste inn alle passord de kommer på, eller at en bruker logger på med forskjellig IP bare noen millisekunder etter hverandre, så kan man ta til høyde for at det ofte ikke er en person som skal ha tilgang til denne informasjonen. Man setter ofte opp regler (prøv passord 3 ganger, så vent x-sekunder, neste gang tar det lenger tid) for å stoppe dette.

Steg 3.5: Utnytt sårbarheten

Hvis man finner en sårbarhet kan man velge å utnytte denne. Dette må være i tråd med kontrakten til selskapet.

Steg 4: Fiks problemet

Oppgave 7

I korona har hjemmekontor blitt den nye normalen, noe som har ført med seg nye sikkerhetstrusler. Nå må ansatte ha tilgang til sensitiv informasjon fra steder utenfor et satt kontor, og mange bruker privat utstyr for å holde seg oppdatert på jobb. Steder man før kunne ha et kablet intern-nett må gå over på VPN, og man har mindre oversikt og kontroll over sikkerheten i hjemmet. Det er nå også flere på den nye arbeidsplassen som ikke nødvendigvis skal ha tilgang på informasjonen som kommer på møter, og man kan overhøre sensitiv informasjon lettere enn før.

Vi har tidligere etablert at den største sikkerhetstrusselen er menneskelige feil, noe som ikke kommer til å minske når menneskene befinner seg spredt rundt uten en standard for håndtering av sikkerhet. Siden mye av arbeidsdagen nå er heldigital, har også cyberangrep økt vesentlig. Når mer av kommunikasjonen foregår digitalt, er det også lettere å falle for phishing-angrep. Man mister oversikten over nye kollegaer, og det er lett å utgi seg for andre. Det er også mindre sikkerhetstiltak tilrettelagt på hjemmekontoret, og det varierer fra person til person hvordan de har satt opp sikkerheten i hjemmet. Man må også ta med seg sensitive (fysiske) dokumenter hjem, uten å nødvendigvis ha et godt sted å lagre dem.

Hvis hjemmekontor blir en større del av arbeidshverdagen, vil jeg tro at det vil utvikles et større marked for løsninger der man jobber mot en remote-desktops eller virtual machine, slik at arbeidsplassen kan ha litt mer kontroll på datasikkerheten. Det vil nok også bli strengere krav til et eventuelt hjemmekontor, og flere vil få krav om enheter som er kjøpt og kontrollert av arbeidsplassen, med nulltoleranse for jobbrelevant informasjon på personlige enheter. Det blir nok også krav til låsbare skap og mulig lydisolering og nettverkskontroll for de som jobber med ekstra sensitivt arbeid.

Hvis arbeidsplassen jobber med ekstremt sensitiv informasjon vil det nok være krav om fysisk oppmøte, da hjemmekontor aldri kommer til å være like sikkert for en bedrift som skal oppbevare hemmeligheter.

Oppgave 8

Lage nøkkelpar:

openssl genrsa -aes256 -out privateKey.pem 2048

openssl	Biblioteket vi bruker
genrsa	Jobber med Generering av RSA-nøkkel
-aes256	Hvilken kryptering som skal benyttes
-out	Hvor (private) nøkkelen lagres
2048	Hvor mange bits nøkkelen skal være (2048 er egentlig default, og ikke nødvendig å ha med)

```
C:\Users\marie>openssl genrsa -aes256 -out privateKey.pem 2048
Generating RSA private key, 2048 bit long modulus
.....................................+++
.....................................................+++
unable to write 'random state'
e is 65537 (0x10001)
Enter pass phrase for privateKey.pem:
Verifying - Enter pass phrase for privateKey.pem:
```

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-256-CBC,BBF6C1D51ED8F0397B6A3F4D199616E5

zmFM0ia0G3u4DHIJV45eeRJ23XRGI/IUb7IGaOP8haK8vV8v7FYwxvHflqaCzIzI
qLBIKm9CNRSyOXlzL2i8BLpBtYc/O+TU9WBCRYnSe14nHFDebLTSog8Up6tQvIj5
+Pe9Ji4q4hNjn7F1iBmIOeODTkfYdF9dyThDJSGPucjSGkBePvAUr6Hi7vNiriEQ
EXU5F10mAktySU8P1rZLN4XDK4MYs6Qu5b1aO8ddetYuw5L0PWC1rMFcRheeXE+i
Fc43pgKCHElBFVxmFRYpt+7cPEVgWoxq8qOv9VvuubYh1rHdPVGcZw/TLkyJSanT
uBRvSKhokhIcjrJA9pg9TbN5L9K7yCI7/7CogKJ7cWUmWR6Pz7/uVVuFd3G5ZveR
1BLKiV6RohGgGz54xKYmAIDeN50XEobUztzdGhIB6X/Ao4eDmMdLfcWePN0Bbaob
4VBYi4v7ZQ68UD1wMz3Ppu2o5f/GFW236Ys8nr4undsBarjfehOQT+EWHiK34HHD
L4aRyyK0re1rvfPrUvn1wNm1f+mQs8Rcx0z7warBvqipCQx8dPqBH3EOVqJlNOaC
SEsRQOLbusG+mOiBF255AEdhmYMKwb8mzGgis7SgbCy/YLS3Ms+vE/atgWr2QeSM
u/EqFCnFmGNkDnxoRgxlH3s9QSbRQWOnR9N0sBkvXlveTdm19LCFLmlacIatTs9v
qVkARJlbhJ2f5gOA7Dhc7kn0xbmUDFzYJwH29Iqg3ZSwD5ZhfhT5hDwubnP9/Jub
ksAHdizUNBXYv3jiZidnDB6U0kLDEvliZMWN93A9cNmCh9B+wGldK6JcbN6bta3O
0D085PCZSuIjYYTzXnsbB3CmRvyExd0ZwizZBikpFpZT+wWcRv7MUGdAnqbfLNkY
Amhft7TNvCWECNHwrdAFvUNo7fRsLG8IXAsIyrrSlaLc5Ujz1xxK9fYxQbbLkuVg
1ILZ0i0qj68tblFR/e3nDCXq3+px+Hz90n93Nzw4I0MD2YEkTzWJCrEIqYFJisvJ
ctLjsimX6S2glrju5hgsGOvhMVnWxztM20aY4PIPbl7n83KJFAWjfTT2J9U2L5ih
+Oa9XTizO8sV6lnixccgaBIowRTA9xto5mEvegidY7UsgZM5XCK/08qXXWiG8xSg
5wUzUZck3UwNkXF1HrT5leKvHU85ktUCEmejW/PLdM+bnZsVCkbfNFyGgufJAs+z
egdeFHoTQ1It4ryOHfLPe6Jchz2FgVJBz6WanMRopYsJFMK3BJsWlSd0yvULt5Ts
beElIThznhRTgg0jXQiXNYv+wzkSlXP6Yp9vvglqTQxzgNoJwrfllaYrQ+S/ZbFX
pNb44Znc3bh/hd3cc4qG00qOhjBIq0ISyXSH+nSgTp03J7UnsxKm8+oJlxi9V8lT
BRyp2nzPhCX2+UWK3WBt6V+Yp7e6VuRDHI5oKLC/+Ktev2pjvV4KmJgSknRg5zvw
IlMexFtPkdKDOpAUibGldfDcHJMV9HSNWBICylkmyeCk/i0jKJ9mYdoz+G6jQbzw
j39tdXuURButD/Xq1Zp2VlLgFUJez9/faXZtsM8nW0MQQYNDUCdghuCvydanytqK
-----END RSA PRIVATE KEY-----

```

Eksportere public-key:

Openssl rsa -in privateKey.pem -outform PEM -pubout -out publicKey.pem

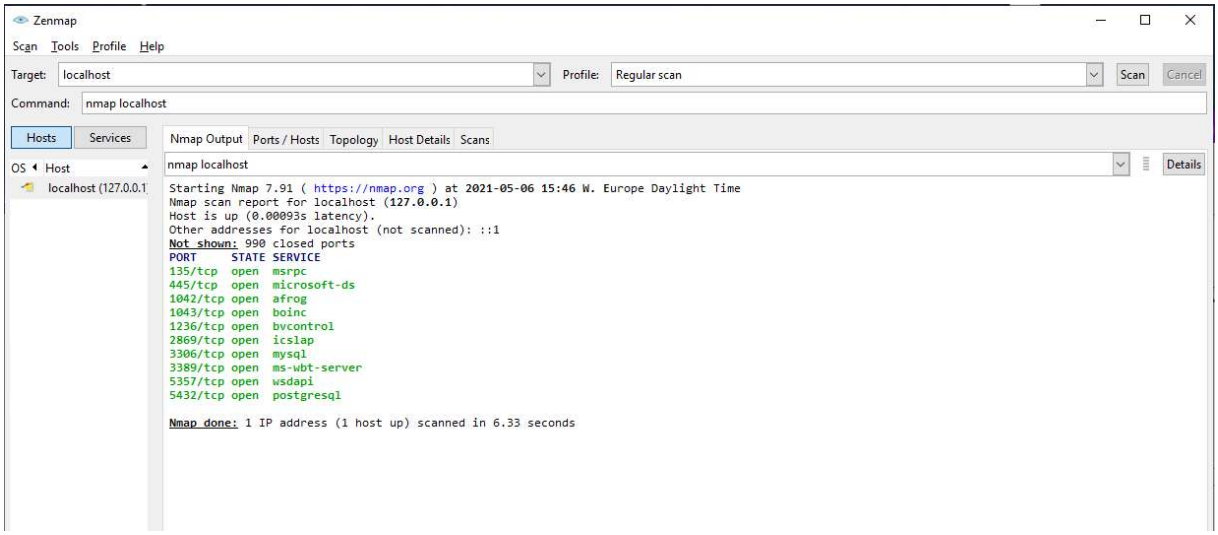
Openssl	Biblioteket vi bruker
RSA	Jobber med RSA-nøkler
-in	Fila vi skal hente inn
-outform	Filtypen vi skal ha ut (PEM)
-pubout	Vi vil ha ut public-nøkkelen I steden for private-nøkkelen
-out	Fila vi vil ha ut

```
C:\Users\marie>Openssl rsa -in privateKey.pem -outform PEM -pubout -out publicKey.pem
Enter pass phrase for privateKey.pem:
writing RSA key
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs0d/VQlZDahy0PRN88Fd
HwNOwC0DNeedPpcrhJdtAgtDK2iLi+34Yp866SdwXeWKm9vDgU8z8/PfumxBjbCk
uVpanejBRheAgPTqM5daVKsdgMOAijz122D0cs3Jod/02/2Ku6DQfo15qHxJn9B1
Dq9FtqLNajmPZIbdJAhym5uk05Jy3Lvjwa6tKW+k2r96TdcJCKVW265hE1PHOC/U
wrLBMvMPzzcNBAIP255Lxt/XFzHMumjkrny2w/bQJGccQ+7Fk6vmdep7Ql/4JJpe
aiRUZeCfz6UAyhdGAxelN+q6K5SuDn2vI15Oo9ruOYxQmvo/+AxNKZY8x7Kgz5Uy
wQIDAQAB
-----END PUBLIC KEY-----
```

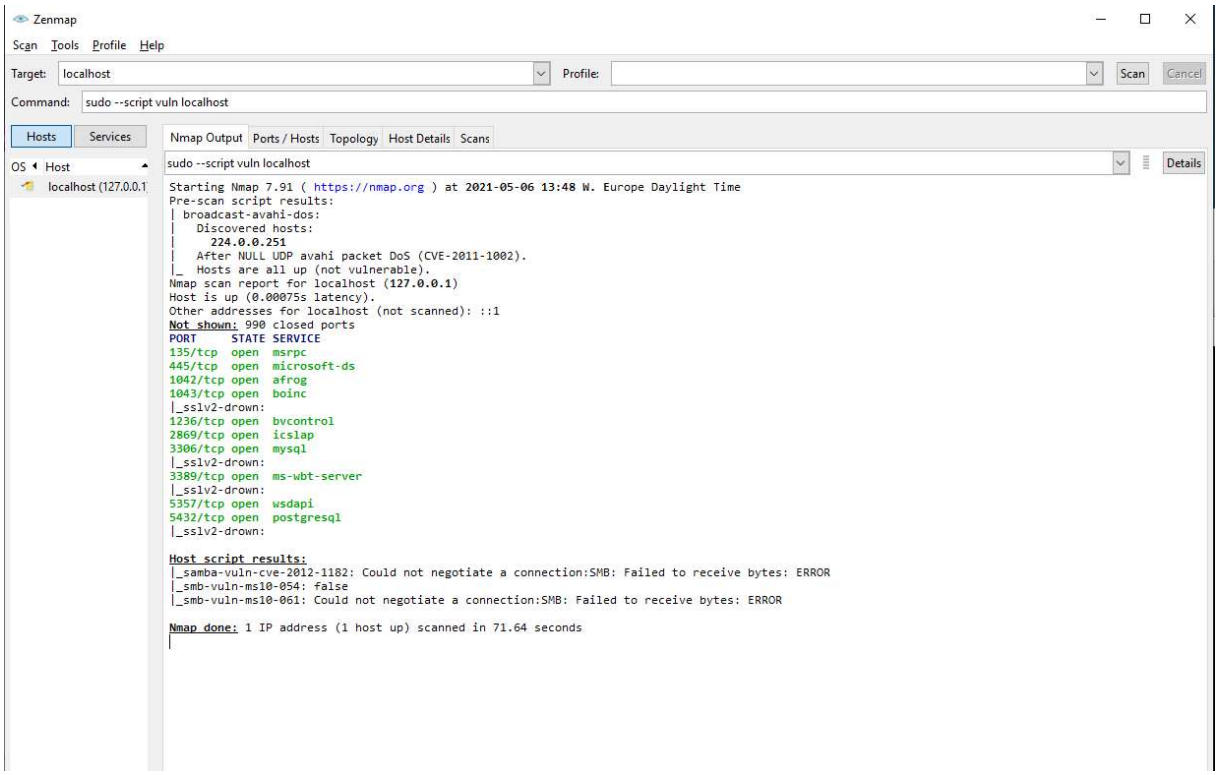
Oppgave 9

Regular Scan av localhost



Vanlig scan av nmap vil vise hvilke porter som er koblet til og tilgjengelige for valgt ip-adresse. Denne sjekker bare localhost, altså ikke alle enheter som er koblet til ruteren

Vuln-scan



Vuln-scan sjekker nettverket for svakheter (vulnerability). Som vi kan se, er alle hoster oppe og det er ingen svakheter.