



Denne forelesningsøkten vil bli tatt opp og lagt ut i emnet i etterkant.

Hvis du ikke vil være med på opptaket:

 Start Video	La være å delta med webkameraet ditt.
 Unmute	La være å delta med mikrofonen din.
To: Marianne Sundby (Privately) Type message here...	Still spørsmål i Chat i stedet for som lyd. Hvis du ønsker kan spørsmålet også sendes privat til foreleser.



Høyskolen
Kristiania

TK2100: Informasjonssikkerhet

12. Forelesning

Eksamensforberedelse

Dagens tema

- Tradisjonelt er siste forelesning en ren repetisjon, men jeg har valgt å legge inn noen nye spennende temaer i dag for å «runde av» faget
- Repetisjon
- Eksamensforberedelser

Eksamen 2020

Om årets eksamen i TK2100

- **24 timers hjemmeeksamen**
- **Vurderingsform: Bestått/Ikke Bestått**
- 8-10 oppgaver, fritekst drøftingsoppgaver, praktiske oppgaver, kanskje noe regning?
- OBS; fristen er på 24 timer, men det kreves ikke mer enn 4-6 timers jobbing!
- Formålet med oppgavene er å vise forståelse for faget
- Øvingstimer med fritekst oppgaver er veldig relevante for denne eksamensformen
- 2-3 oppgaver er praktiske, så gå gjennom notatene deres fra praktiske labøvinger 😊

Poeng setting på eksamen

- På en deloppgave som gir maksimalt 5 poeng foretas sensur slik:
 - 1 poeng: Studenten har svart feil, men «inne på noe»
 - 2 poeng: Veldig kort svar som kun delvis svarer på oppgaven
 - 3 poeng: Et kort og riktig svar, men oppfattes som «tynt»
 - 4 poeng: Et godt svar, men ikke utfyllende eller komplett
 - 5 poeng: Fullgodt svar på oppgaven
- Svarer du KUN på Oppgave 1 og 2 (totalt 50%), må alle deloppgavene ha «et godt svar» for å ha forhåpninger om å bestå. Moral: SVAR PÅ ALT 😊

For å full score på en oppgave kreves det at svaret underbygges av kilder. Kildehenvisninger og sitater skal angis løpende i tekst (APA7 eller Chicago forfatter-år standardene anbefales), samt i referanseside på slutten av besvarelsen. (Merk; ikke et krav for å stå på eksamen, men et krav for full score på en oppgave.)

Nye temaer for å «runde av» faget

Er login navn sensitivt?

- Tidligere var det vanlig at når man opprettet en konto på en tjeneste måtte man også finne et brukernavn, brukernavnet måtte være unikt på tjenesten
- I dag er normalen at man bruker sin EPOST ADRESSE
- Dette resulterer i at
 - Det er lett å gjette brukernavnet til en person
 - Hvis et passord er lekket fra en tjeneste kan man ofte bruke samme passord på andre tjenester
 - I verste fall brukes også det lekkede passordet på epost adressen selv...
- Dette kombinert med «reset my password» muligheter kan øke risikoen for kompromitterte kontoer (husk Birthday Paradox hvis det er en 6 siffer reset-kode)
- Reflekter over dette når man implementerer nye løsninger 😊

Passord reset brukt til konto hijacking

- Passord reset er fint for de som glemmer passord, men for sikkerhet er det en HÅPLØS feature (men; den er ikke valgfri!)
 - I tillegg er reset koden hos noen en 6 sifret kode (ref forrige slide)
- Passord reset betyr at hvis en angriper får tilgang til målets epost eller mobiltelefon, kan angriperen hijacke ALLE andre kontoer
 - Hvor mange kontoer ber om noe MER enn epost for å resette passord?
 - Paypal ber om «security questions» – en annen håpløs feature da det ligger på internett...
 - Paypal har også en metode for å resette passord som krever tilgang til kredittkort mm (bedre)
- Duplisering eller hijacking av mobiltelefon – er det mulig?
 - Bestille nytt SIM kort
 - Kjøpe nytt abonnement av en konkurrent; behold mobilnummer + få nytt SIM kort...
- Reflekter over hva en angriper kan gjøre med tilgang til din epost!

«Security questions» brukt til konto hijacking

- Er «security questions» fint for de som glemmer passord?
 - Gjør at man ikke bare trenger tilgang til epost konto...
- Dette er ofte spørsmål man kan finne på internett
 - Din mors pikenavn
 - Navnet på ditt første kjeledyr
 - Hvilken ungdomsskole gikk du på
 - ... alt dette ligger jo på Facebook 😊
- Personlig oppgir jeg enten ikke svar på security questions, eller jeg oppgir 40 random tegn! 😊
 - Ble et problem når jeg glemte passordet mitt på paypal...
- Jeg vurderer også å ta i bruk «midlertidige» epost adresser, for å forhindre password reset utfordringene på forrige slide...

Vi har pratet om fysisk sikkerhet i løpet av kurset, og jeg har diskutert det med flere av dere i øvingstimene, men la oss formalisere litt mer:

- Dørlåser, låsdirking, forskjellige grader av sikkerhet, laptop låsekabler
- USB Rubber Ducky og andre fysiske enheter
- RFID og NFC kort kopiering (aka stjeling)
- Full-disk kryptering – hva har det med fysisk sikkerhet å gjøre

Jeg har pratet litt om fysiske låser i løpet av emnet, og som dere vet har jeg låsdirking som hobby. Det er viktig å ha et bevisst forhold til den fysiske sikkerheten, og som sikkerhetsansvarlig spørre seg – hvor god lås har jeg kjøpt for å beskytte denne ressursen?

Basic Lock (nivå 1): åpnes med enkle verktøy, også av en utrent angriper

Resistant Lock (nivå 2): noe motstand mot å bli dirket opp, kan ikke «bumpes» eller «shimmes» - laveste nivå som kan brukes på et kontorbygg

High Security (nivå 3): flere mekanismer mot å bli dirket opp, utrent angriper har ingen mulighet til å åpne, erfaren låsdirker vil bruke minimum 5 minutter på å åpne

Unpickable (nivå 4): en erfaren låsdirker med spesialutviklet verktøy kan kanskje åpne låsen, men det vil ta minimum 30 minutter

Rubber Ducky

Hvis en angriper har fysisk tilgang til en maskin er det mange ting han/hun kan gjøre – for eksempel hva kan en USB stick gjøre (annet enn å lagre filer)?

HAK5 er blitt markedsledende og defacto standard for fysisk penetrasjonstesting, og jeg antar at de brukes av de fleste angripere også :-)

Rubber Ducky; USB enhet som sender tastetrykk (simulerer et USB tastatur), som om en hacker sitter der fysisk og skriver kommandoer, veldig fort...

Bash Bunny; USB enhet som simulerer mange USB enheter, inklusive nettverk, kan stjele NTLM hasher og knekke PCens passord selv om skjermen er låst!

Shark Jack; kan utføre mye av det samme, over Ethernet port :-)

<https://shop.hak5.org/>

RFID og NFC

Nesten alle kontorbygninger bruker adgangskort, adgangskort som holdes inntil en kortleser bruker RFID eller NFC teknologi (vi pratet litt om det i TK1104)

Men hva hvis disse kan kopieres?

Kan en angriper komme seg inn i kontorbygget da?

For dårlige kort:



For bra kort:



Full-disk kryptering

Hvis en angriper stjeler en fysisk maskin, hva kan han/hun gjøre? Trenger passord for å logge seg inn i operativsystemet?

1. Skru ut harddisk
2. Koble harddisk i en annen maskin
3. Lese alt som er på maskinen

Med mindre harddisken er full-disk kryptert!

Sikreste måten å gå fra en PC i et kontorlandskap er 1) “hibernate” 2) full-disk kryptering med boot password.

Stjele harddisk virker ikke, Bash Bunny virker ikke, Shark Jack virker ikke :-)

Viktige emner / repetisjon

(Spørsmål jeg har akkumulert de sist årene fra studenter.)

- Digitale signaturer baserer seg på asymmetrisk kryptering (public/private key kryptering)
- Man lager en hash (en sjekksum) av data, og krypterer dette med sin PRIVATE key – dette kalles signering
- Nå kan alle i hele verden «dekryptere» sjekksummen (så den er ikke kryptert i betydningen konfidensiell), men ved at man klarer å dekryptere sjekksummen så beviser man at den var signert (kryptert) med din PRIVATE key

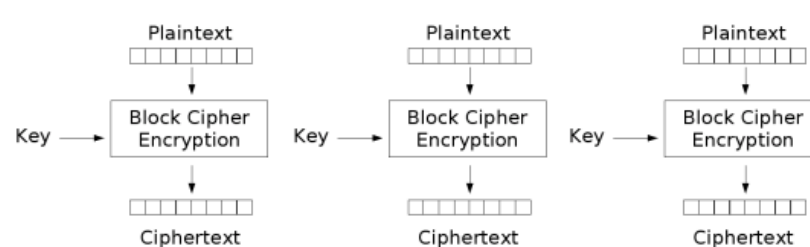
Digital signatur #2

- RSA algoritmen kan brukes både til kryptering og signering
- Hvis man krypterer med en PUBLIC key kan kun eieren av private key lese
- Hvis man signerer med en PRIVATE key kan alle lese, og alle vet at meldingen ble skrevet av den som har den private nøkkelen
- Man har alltid 2 nøkkelpar, ett par for signering og ett annet par for kryptering – ikke bruk samme nøkkelpar til begge deler
- Et digitalt SERTIFIKAT er en fil som beviser at en public key tilhører en bestemt person eller firma, dette er håndtert ved at en kjede av signaturer godkjenner public key'en – hvor typisk rotsertifikatet er allment kjent for alle (feks innebygget i Windows)

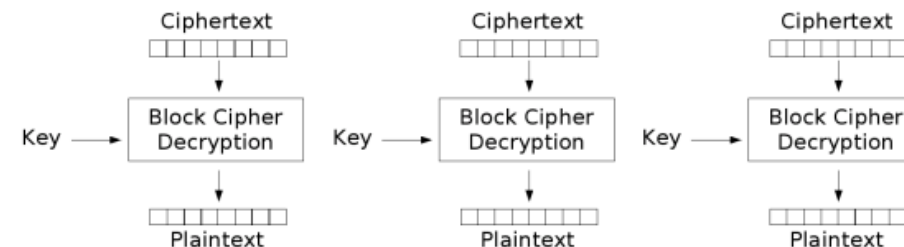
Krypteringsmoduser (AES)

- De to viktigste modusene for AES kryptering er ECB og CBC
- La oss bruke noen minutter på å repetere slidene fra den forelesningen:

- AES og andre blokk-ciffre kan kjøres i ulike **moduser**
- Modusen handler om **måten** og **rekkefølgen** kryptering og dekryptering foregår på/l meldings-blokkene
- **Electronic Code Book (ECB) Mode** (er den enkleste):
 - Blokk $P[i]$ krypteres til kryptogramblokk $C[i] = E_K(P[i])$
 - Block $C[i]$ dekrypteres til klartext blokk $M[i] = D_K(C[i])$
- M.a.o. Dette kan foregå i **parallell**



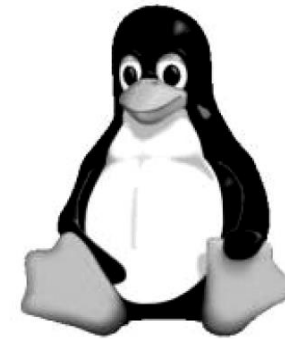
Electronic Codebook (ECB) mode encryption



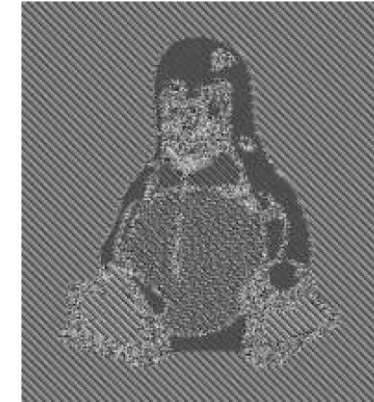
Electronic Codebook (ECB) mode decryption

ECB: Styrker og svakheter

- Styrker:
 - Enkelt å gjennomføre
 - Tillater parallell kryptering/deryptering av blokker
 - Taps-tollerant fordi om en blokk går tapt er de andre intakte
- Svakheter:
 - Noen dokumentformater og bilder egner seg ikke for ECB kryptering fordi mønstre i “klarteksten” kan reflekteres på “avstand” selv om hver blokk er “godt kryptert”



(a)



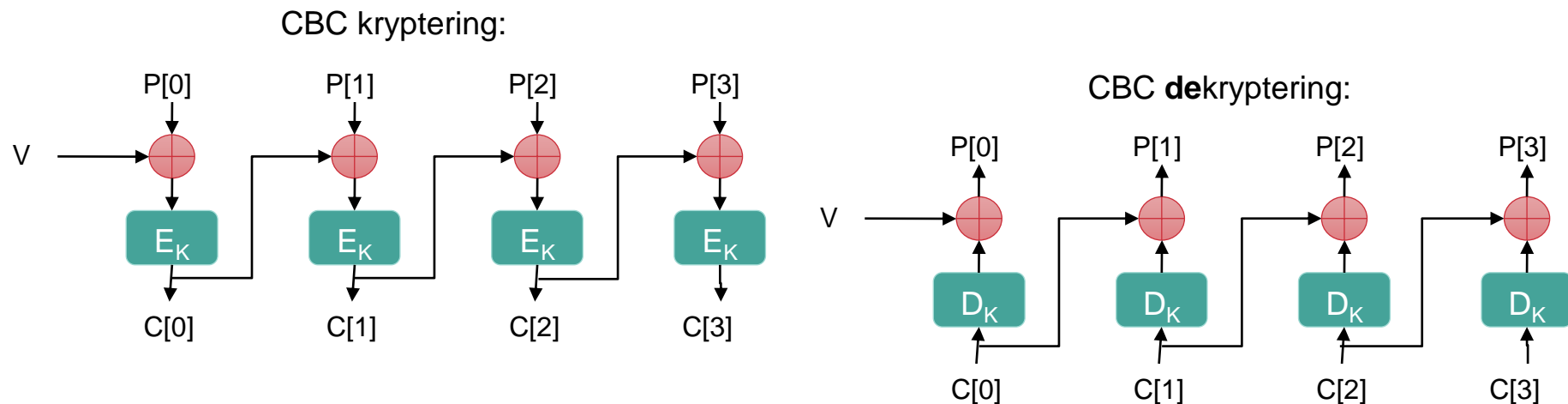
(b)

Figure 8.6: How ECB mode can leave identifiable patterns in a sequence of blocks: (a) An image of Tux the penguin, the Linux mascot. (b) An encryption of the Tux image using ECB mode. (The image in (a) is by Larry Ewing, lewing@isc.tamu.edu, using The Gimp; the image in (b) is by Dr. Juzam. Both are used with permission via attribution.)

Chiffer Blokk Lenking (CBC)

- I Cipher Block Chaining (CBC) Modus

- kombineres forrige chiffer-text blokk med neste klartext-blokk $C[i]$
 $= E_K (C[i - 1] \oplus P[i])$
- Initialiserer med $C[-1] = V$, en random blokk som er separat kryptert
- Dekryptering: $P[i] = C[i - 1] \oplus D_K (C[i])$



CBC: Styrker og svakheter

- Styrker:

- Avslører ikke mønstre i klarteksten
- Er den vanligst brukte
- Relativt rask og enkel

- Svakheter:

- CBC krever pålitelig overføring av alle blokker, i riktig rekkefølge
- CBC egner seg ikke for anvendelser som tillater pakketap (f.ex. Musikk- og video-streaming, telekonferanse, ...)

Blockchain – samme som CBC?



- Et oppfølgingsspørsmål om moduser er om CBC egentlig er en block-chain
- Blockchain har ikke vi hatt om, men det er det som er basis for feks BitCoin
- En Blockchain er bygget opp slik at man har en trestruktur av blokker, hvor block inneholder en HASH av forrige blokk – altså ikke en XOR av hele den krypterte forrige blokken med plaintekst fra den nye
- Man kan da si at selv om metodene kan ha noen (små) felles trekk så er AES CBC en kjedet krypteringsmetode – Blockchain er en protokoll («ledger») for å forhindre at noen endrer data som er registrert i ettertid
- TDLR; ikke bland sammen disse to begrepene

DH og RSA sikre i dag?

- Spørsmålet er om Diffie-Hellman og RSA er ansett som sikre i dag
- Forutsatt at nøklene er store nok er både Diffie-Hellman og RSA definert som sikre, men det er noen svakheter i Diffie-Hellman som gjør at man i praksis sier at «ren» DH ikke er sikker
- I praksis brukes en kombinasjon av de to for å oppnå best sikkerhet; i TLS heter det DH_RSA
- Det forutsettes også at nøklene er tilfeldig valgt, og at implementasjonen ikke inneholder feil eller svakheter
- Jeg trekker frem denne sliden fra forelesningen om kryptering:

Hvor trygt er egentlig RSA?

- RSA-768
- Et semi-primtall (et tall som er ett produkt av to nesten jamstore primtall)
- 768 bit – 232 desimalsiffer (skriver du et tall i sekundet bruker du ca 4 minutter på å punche det...)
- Faktoriseringen ville tatt ca 2000 år på en 2,2 GHz prosessor; men ble gjennomført som et samarbeid mellom flere forskningsinstitutt over to år.
- Kan også få indikasjoner ved å måle Cache-bruk, tid det tar å beregne ut fra offentlig nøkkel mm
- **INGEN VET SIKKERT HVOR VANSKELIG DETTE EGENTLIG ER!**
- Merk: Vi bruker ikke så små nøkler som 768 bits, vi bruker minimum 8196 bits 😊

Hvordan holde OS trygt

- Bruke et operativsystem som blir vedlikeholdt (Windows XP og Vista er ikke lenger supportert, Windows 7 er ute av mainstream support, men har extended support til 14. januar 2020)

<https://support.microsoft.com/en-us/help/13853/windows-lifecycle-fact-sheet>

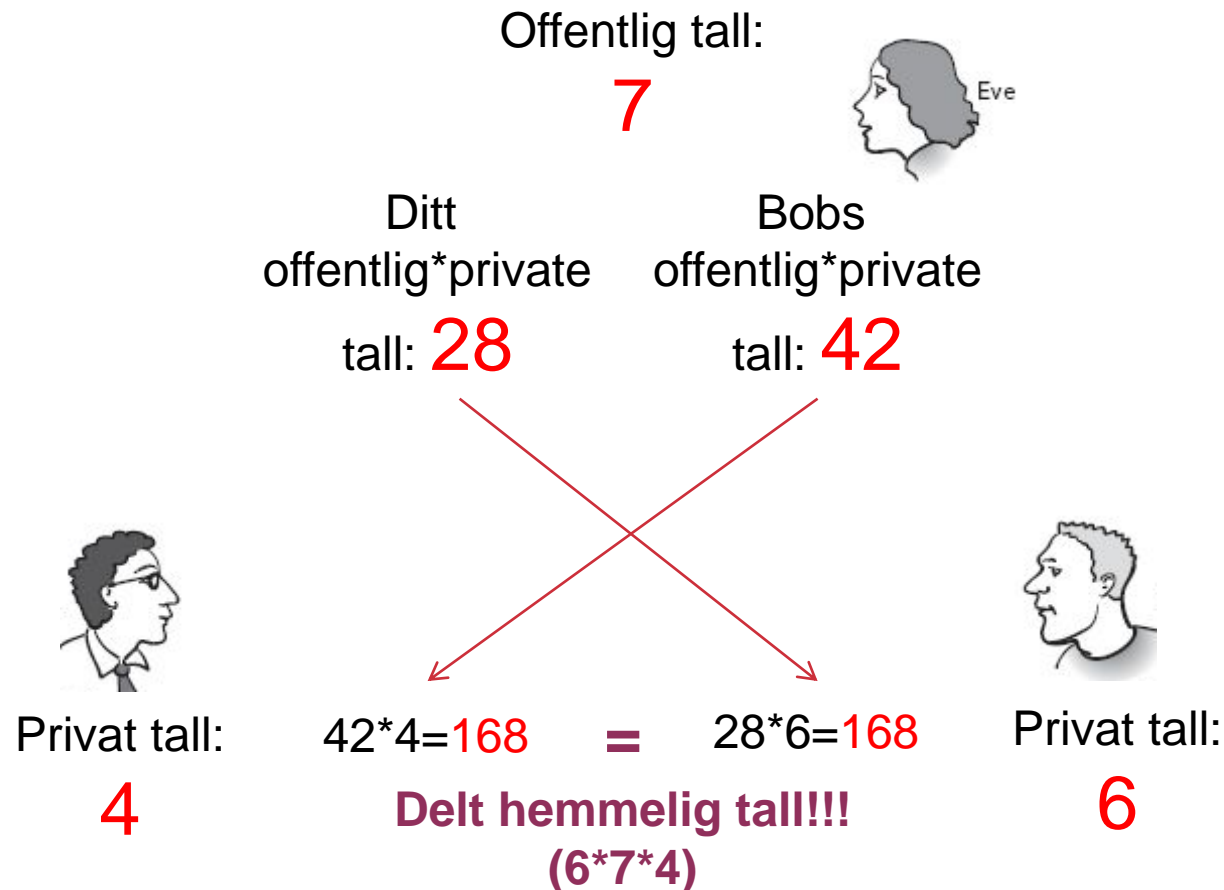
- Installere siste patcher, helst automatisk
- Ikke installer ukjent software, eller software fra ukjente / usikre kilder
- Ha anti virus program og en personal FW
- Som siste forsvar; ta backup (er du paranoid tar du backup til en online tjeneste, og så tar du i tillegg regelmessig kopi av alle filer til den extern harddisk som du oppbevarer i en safe)

Kryptering basert på random tall

- Nesten all (moderne) kryptering baserer seg på at nøkkelen er tilfeldig, unntaket er nøkler avledet av brukerens passord
- Vanlige krypteringsalgoritmer baserer seg ikke «internt» på noe random, men nøkkelen er laget basert på et random tall

Eksempel: Basis for D-H nøkkelutv.

- Tenk deg at multiplikasjon er en enveisfunksjon!
 - Lett å multiplisere («gange»)
 - Veldig, veldig vanskelig å dividere («dele»)
- Du vil dele en felles krypteringsnøkkel med Bob, men ikke Eve



Diffie-Hellman

- Divisjon er ikke vanskelig nok
- Vi *tror* at diskrete logaritmer er tilstrekkelig vanskelig.
 - Å finne verdien til x når du vet at f.eks. $2^x \% 11 = 3$ krever masse testing, og blir vanskelig når tallene er store!
 - Og se gangetabellen for $\%11$ øverst til venstre...
- Starter med å velge en basis (f.eks. 2) og en modulus (f.eks. 11)
 - Modulus må være et primtall:

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	1	3	5	7	9
3	3	6	9	1	4	7	10	2	5	8
4	4	8	1	5	9	2	6	10	3	7
5	5	10	4	9	3	8	2	7	1	6
6	6	1	7	2	8	3	9	4	10	5
7	7	3	10	6	2	9	5	1	8	4
8	8	5	2	10	7	4	1	9	6	3
9	9	7	5	3	1	10	8	6	4	2
10	10	9	8	7	6	5	4	3	2	1

n	2^n	3^n	6^n
1	2	3	6
2	4	9	3
3	8	5	7
4	5	4	9
5	10	1	10
6	9	3	5
7	7	9	8
8	3	5	4
9	6	4	2
10	1	1	1

Offentlige tall:

$\% = 11, b = 2$



Ditt offentlig*private tall: $(2^8)\%11 = 3$

Bobs offentlig*private tall: $(2^9)\%11 = 6$



Privat tall:

8



Privat tall:

9

$(6^8)\%11 = 4 = (3^9)\%11 = 4$

Delt hemmelig tall!!!

Hva med kryptering?

- Diffie-Hellman utveksler (symmetriske) nøkler med private/public nøkler – men det er ikke noe kryptering
- Whitfield Diffie og Martin Hellman publiserte metoden i 1976
- Det hadde vært bra med samme "teori" for kryptering, det jobbet på det tidspunktet mange med...
- I 1977 kom Ron Rivest, Adi Shamir og Leonard Adleman opp med en metode:

1. Velg to store primtall p, q .
(f.eks. 1024 bit, dvs min. 100 desimale siffer hver)
2. Beregn $n = pq$, og $z = (p-1)(q-1)$
3. Velg krypteringsnøkkelen e (der $e < n$) slik at den har ingen felles faktorer med z . (e, z er "relative primtall").
4. Velg dekrypteringsnøkkelen d slik at $ed-1$ er "exakt delbar" på z . (m.a.o.: $ed \bmod z = 1$).
5. Public key er $\underbrace{(n, e)}_{K_B^+}$. Private key er $\underbrace{(n, d)}_{K_B^-}$.

RSA: Kryptering, dekryptering

0. Gitt off. krypteringsnøkkel (n,e) og privat (n,d) dekrypteringsnøkkel

1. For å **kryptere** bit mønsteret, m , beregn

$$c = m^{e \bmod n}$$

2. For å **dekryptere** mottatt bit mønster, c , beregn

$$m = c^{d \bmod n}$$

Magi?!

$$m = \underbrace{(m^{e \bmod n})}_c^{d \bmod n}$$

RSA (leke-)eksempel:

- Gitt at dere har en public key hvor $n = 3233$ og $e = 17$, og deres privat key er $n = 3233$ og $d = 2753$. Dekrypter min melding til dere:
- 3000 28 2726 2726 1307 1992 641 2726 2790 2680 2680

$$M = C^d \bmod n = 3000^{2753} \bmod 3233 = 72 = \text{bokstaven 'H'}$$

72	69	76	76	79	32	67	76	65	83	83
H	E	L	L	O		C	L	A	S	S

- Ved implementasjon av TLS og/eller krypto systemer (i C/C++) anbefales det å bruke OpenSSL biblioteket
- Hvis du har kontroll på både server og klient kan du begrense cipher suites til de sikreste
- Eksempel på "cipher suite" format i TLS 1.2
 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
 - Key exchange: ECC (ECDHE)
 - (Server) authentication: RSA
 - Encryption: AES 128, GCM mode
 - Message Authentication Code: SHA256

Usikre TLS ciphers

- Gjennom ulike pen-tester i 2020 og 2021 kan jeg fortsatt se TLS 1.0 støttet, og servere som støtter 3DES og til og med RC4-algoritmer, Diffie Hellman-nøkkeltutveksling er også ansett som ikke sikker
- Test alltid en webserver ved hjelp av:
ssllabs.com
- Hvis du ikke får en A eller A+ rating, starter du på nytt og prøver igjen...

Secure Hash Algorithm (SHA)

- Utviklet ved NSA og godkjent av NIST
- SHA-0 og SHA-1 (1993)
 - 160-bit
 - Regnes som usikker
 - Fremdeles i bruk
 - Mindre sårbar enn MD5
- SHA-2 (2002)
 - 256 bits (SHA-256) eller 512 bits (SHA-512)
 - Finnes publiserte angrepsteknikker, men regnes fremdeles som sikre
- Offentlig konkurranse om SHA-3 startet i 2007
 - Keccak algoritmen ble valgt som SHA-3, men det er mange kontroverser rundt interne endringer som ble tvunget inn av NIST, som gjør at mange i dag ikke stoler på den
 - Brukes ikke, av frykt for at den er bevisst svekket (av NSA?)

Klassifikasjon («historisk»)

- Vi kan dele **malware** opp i ulike **typer** ut fra hvordan den spres og hvordan den skjuler seg.
- Spredning
 - **Virus**: Virus endrer eksisterende filer eller systemer, koden kan ikke leve eller spre seg alene
 - **Orm**: automatisk spredning fra maskin til maskin over nettet
- Skjuler seg
 - **Rootkit**: endrer OS for skjule nærvær
 - **Trojaner**: Nytteprogram som skjuler ondsinnede operasjoner (f.eks. keylogger)
- «Nyttelast» (payload)
 - Alt fra humor/irritasjon til ran av maskinkraft og identitetstyveri

Hva er et computer virus?

- Et program som kan **replisere** seg selv
 - ved å endre andre filer/program
 - ved å **infisere** dem med kode
 - som kan **formere** seg videre
- Det er evnene til å **formere seg LOKALT** som skiller virus fra andre typer malware
- Krever vanligvis innledende **brukermedvirkning** for å formere seg
 - Klikke på en link og godta installasjon
 - Åpne epost-vedlegg
 - Dele en **minnepinne**, eller annet USB-utsyr

Hva er en orm («Worm»)?

- Malware som **sprer kopier** av seg selv **uten å infisere** andre program, og vanligvis uten menneskelig medvirkning
- Ikke virus siden de ikke infiserer eller endrer LOKALT (filer eller bootsektorer)
 - men begge deler spres ved selv-replisering
- I de fleste tilfeller vil ormen ha en ondsinnet **nyttelast (payload)**
 - Installere bakdør
 - Slette filer

Anti-virus Signaturer

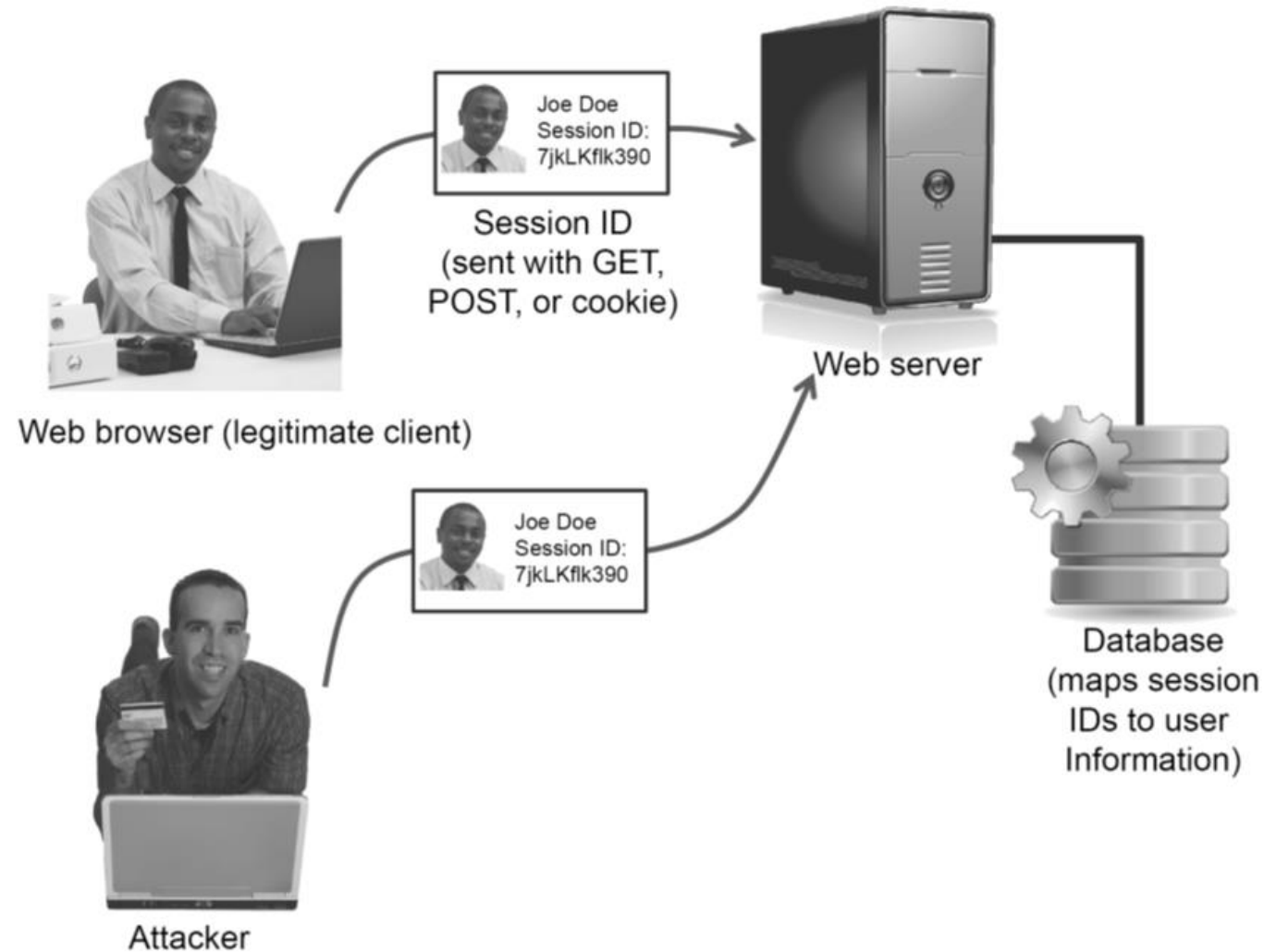
- Scanning sammenligner det analyserte objektet med en database av **signaturer**
- En signatur er «fingeravtrykket» til et virus, eller annet stykke malware
 - M.a.o. en streng med sekvens instruksjoner som er spesifikke for hvert enkelt
 - Ikke det samme som en digital signatur
- En fil flagges som infisert dersom signaturen finnes inne i den
 - Rask **mønster-gjenkjenning** («pattern matching») teknikker benyttes for å søke etter signaturer
- Alle signaturene utgjør i felleskap en malware database som vanligvis er proprietær for produsenten av Anti-virus programvaren
- Signaturer beskytter kun mot malware som har blitt oppdaget og rapportert!

Trusler ved oppstart av en maskin

- Dette er et ganske komplekst tema, og vi har bare veldig kort vært inne på det når vi hadde operativsystemsikkerhet
- Jeg antar at det ble spurt om hvordan malware kan infisere oppstarten av en maskin, da det når man skruer på maskinen ikke kan gå så mye galt
- Malware kan infisere både CPU, BIOS og bootloader med malicious kode
- Det er også mulig for en malicious angriper «stjeler» din maskin, setter inn en bootbar CD og bruker den for å ta over maskinen din
 - For å beskytte mot det må man sørge for at man i BIOS setter kun boot fra harddisk
 - Og så setter passord for å gå inn i BIOS
 - Nå kan en angriper ikke lenger boote fra et infisert medie, hvis man i tillegg har boot passord, feks ved å ha full kryptering av primærpartisjon med TrueCrypt, så vil boot sekvensen være sikker
 - Når man er logget inn kan malware selvfølgelig fortsatt infisere deg, også med CPU og bootloader rootkits, og også BIOS faktisk

- Phishing er epost eller andre meldinger som forsøker å lure brukere til å oppgi personlig informasjon
- Typisk utgir en phishing epost seg for å være fra en bank, fra facebook, fra paypal osv – en aktør vi stoler på
- Det vanligste er å be om brukernavn og passord, men andre varianter finnes

Session Hijacking (sniffing)

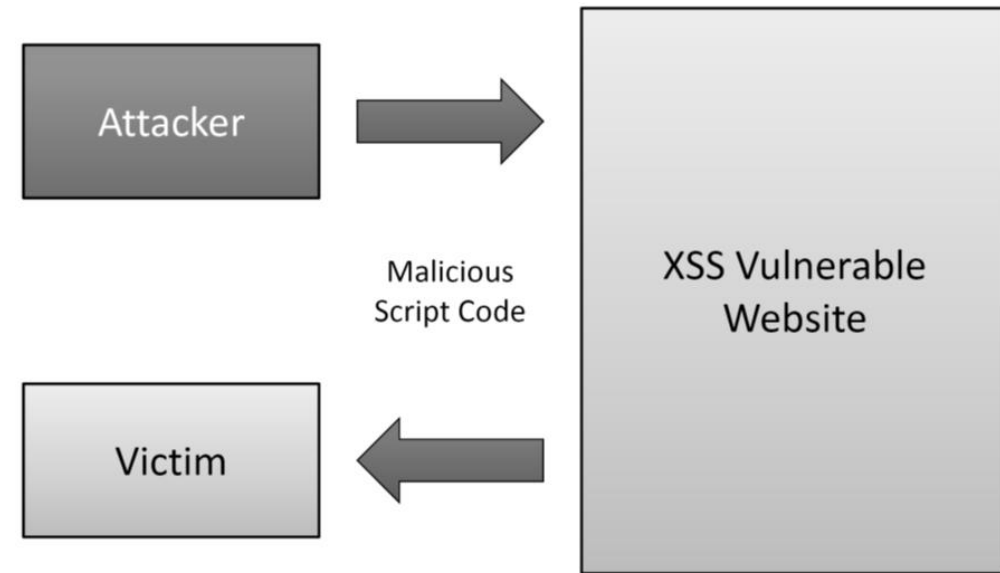


Session Hijacking (sniffing)

- I et Session Hijacking angrep stjeles brukerens (session) cookie
- Da cookie er serverens «state» kan man kopiere cookie til en annen maskin og serveren tror at dette er den opprinnelige maskinen
- Dette er en veldig vanlig sårbarhet i web applikasjoner

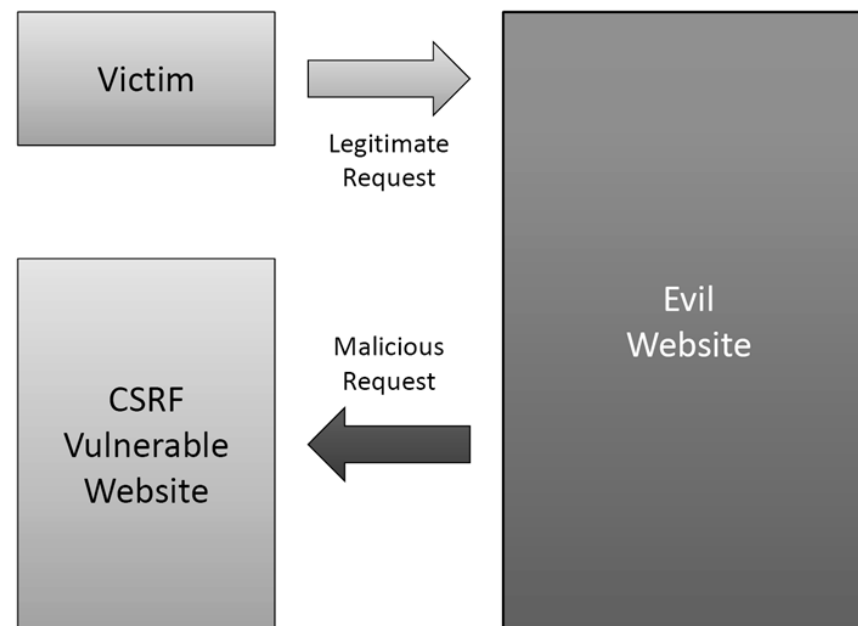
Cross Site Scripting (XSS)

- Injiserer script på webserver i andres web-applikasjoner
- Trusler
 - Phishing, hijacking, endre brukerinnstillinger, cookie tyveri/forgiftning, falsk reklame, kjøre kode på klient.



Cross Site Request Forfalskning

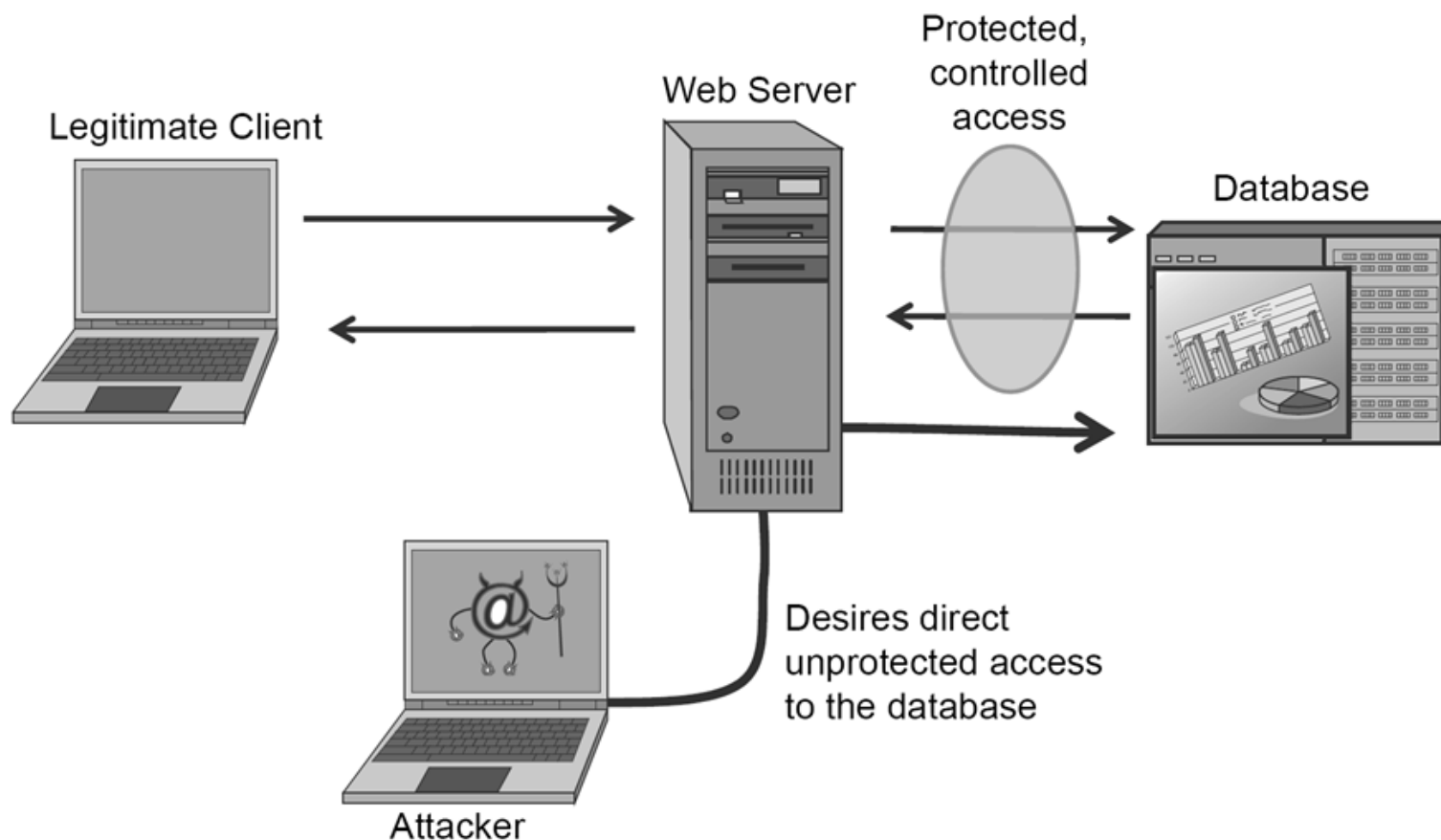
- **CSRF** er det motsatte av XSS
- Utnytter en site's tillit til en bruker, ikke brukerens tillit til site'n
- Naivt eksempel:
 - Bruker er pålogget «bank»
 - Besøker samtidig «slemt» nettsted



```
<script>
  document.location="http://www.naivebank.com/
  transferFunds.php?amount=10000&fromID=1234&toID=5678";
</script>
```


SQL (Injection)

- I tillegg benytter site'n typisk en (relasjons-) database



SQL Injection eksempel

Kode på server:

```
statement = "SELECT * FROM users WHERE name  
= ' " + userName + " ' ;"
```

Injection kode:

```
bengt ' OR '1'='1
```

Resultat:

```
SELECT * FROM users WHERE name = ' ' OR  
'1'='1' ;
```

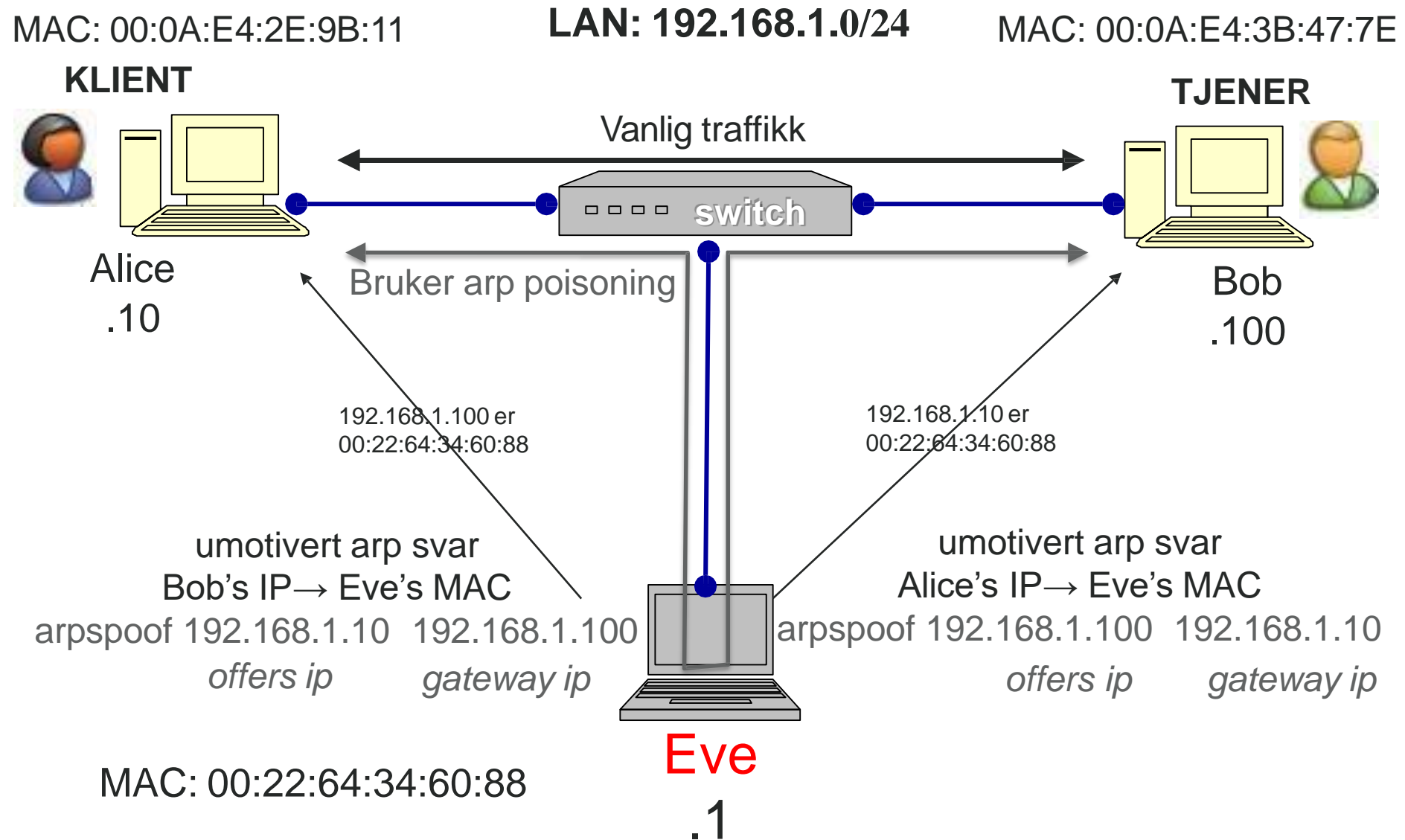
SQL Injection eksempel



Sikkerhet (C.I.A.) og TCP/IP

- **Konfidensialitet?**
 - Ingen krav om det.
 - Kan støttes ved *kryptering* i applikasjon (f.eks. https) eller IPSec/VPN
- **Integritet?**
 - *Sjekksummer* sikrer en viss *pålitelighet*, men ikke på noen måte sikkerhet mot endring
- **Tilgjengelighet**
 - Har vært målet, men ofte vanskelig å *skalere* opp.

ARP Spoofing



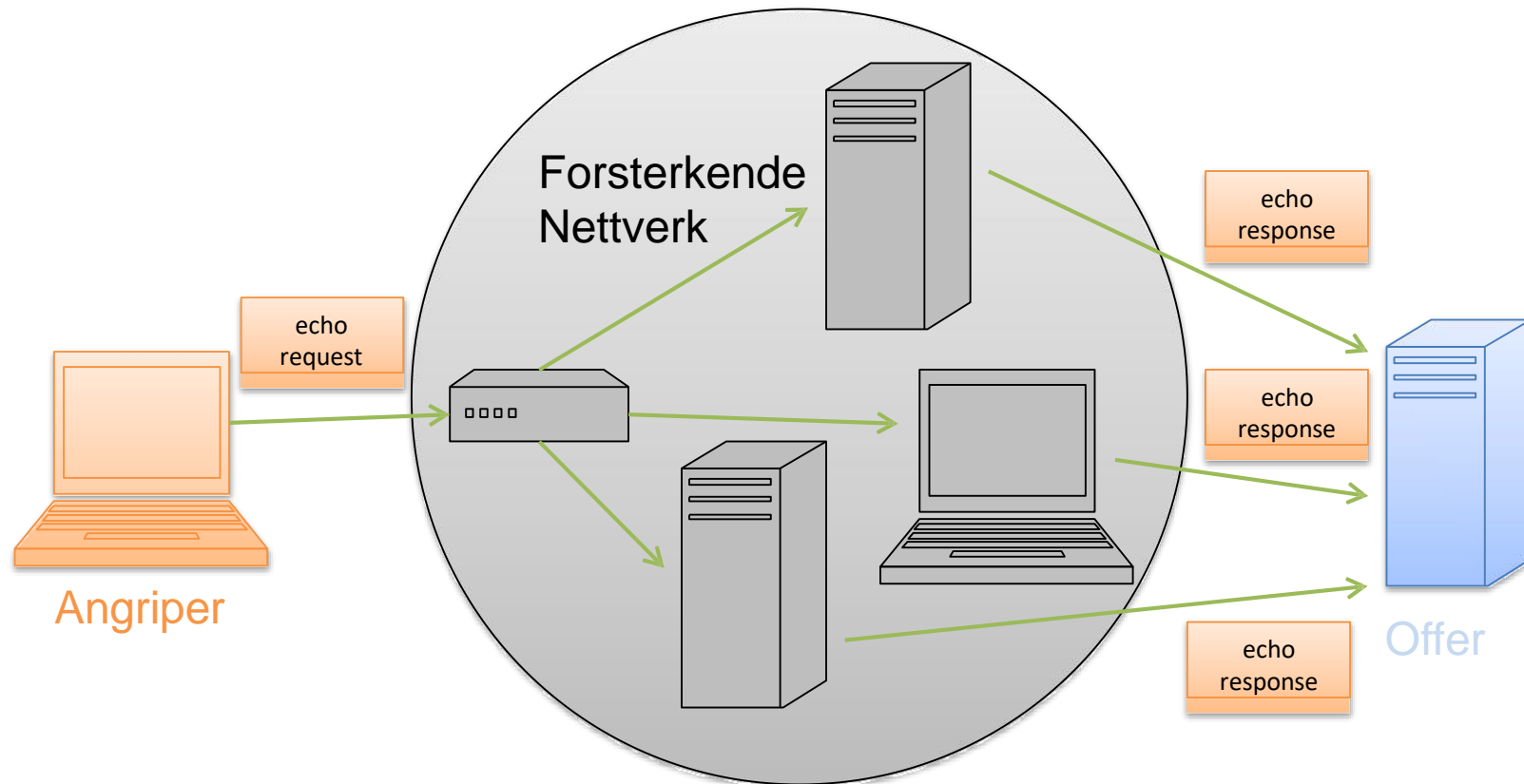
- **Ping of Death**
 - ICMP-standarden sier at en ICMP-melding er på max 64kB
 - Lag ping-pakker som benytter muligheten for å fragmenterte IP-pakker
 - Dele ICMP-«data» ut over flere IP-pakker
 - Resulterer i IP pakker større enn max limit
 - Mange OS kræsjet når de satte sammen igjen ICMP-pakken pga buffer-overflow!
- Tiltak
 - Patche OS
 - Legge grenser i ping og filtrer på routere

Smurfe -angrep



- Broadcast ICMP-echo pakker i nettverket med offerets spoofede IP-adresse som avsender

```
ping -S 10.21.24.1 10.21.27.255
```



- **Ukryptert** overføring
 - Kan avlyttes hele veien fra avsender til mottager
 - Løses stort sett på applikasjonsnivå
- Ingen **avsender-autentisering**
 - Avsender-adresse kan spoofes
 - Gjør det vanskelig å spore opp gjerningsmannen
- Ingen **integritets-testing**
 - Pakken som helhet kan modifieres og innholdet endres; omdirigeres; mao MITM-angrep
- Ingen **bitrate-restriksjoner**
 - Kan injisere vilkårlige mengder pakker inn i nettet og starte DoS-angrep
 - Broadcast gjør DoS enda enklere

(D)DoS

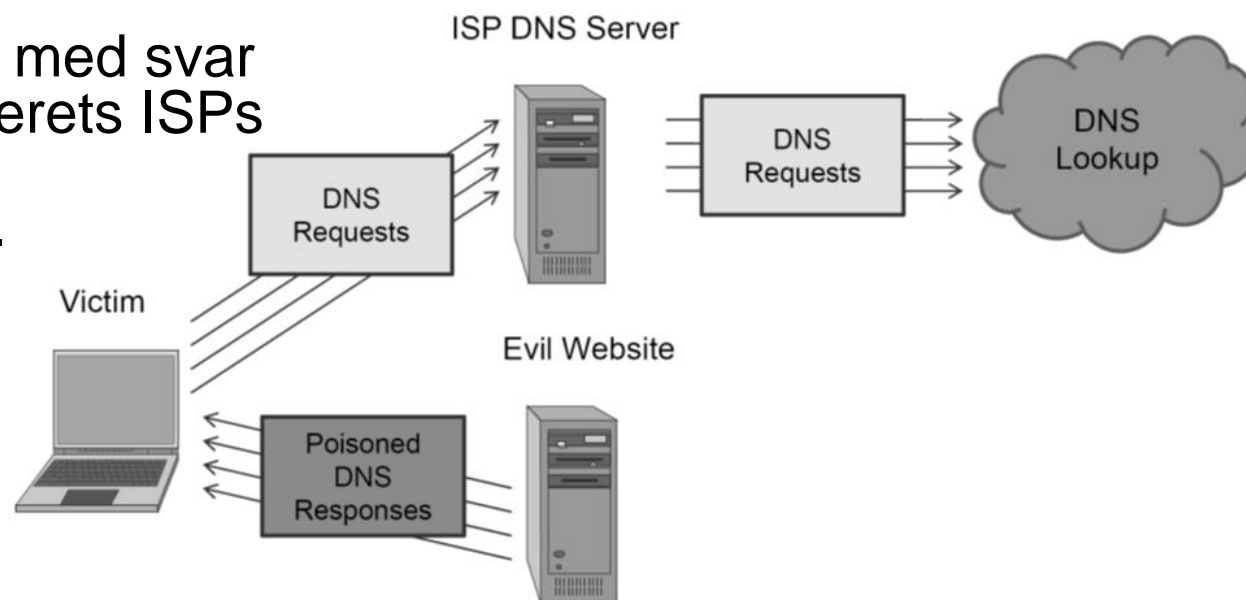
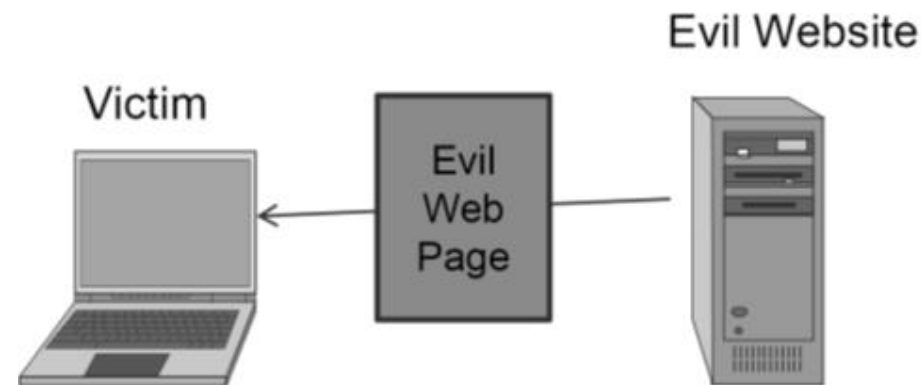
- **D**enial **o**f **S**ervice handler om å forhindre andre (legitime) brukere fra å få tilgang til en tjeneste
 - enten ved å **kræsje** tjenesten
 - eller ved å **overbelaste** tjenesten
- De mest kjente **Distribuerte** DoS de seneste årene har enten vært
 - mot Web-servere: regjering + bank
 - enten politisk motivert: Anonymous; Nord vs Sør Korea
 - benyttet BotNet (ZombieNet)

DNS cache-forgiftning

- Gi DNS tjenere (eller resolveere) falske svar og få dem cachet
- DNS benytter en 16 bits Request ID
 - Samordner spørsmål mot svar ut fra ID
- Cache kan dermed f.eks. forgiftes dersom en NS:
 - Ser bort fra ID
 - Har forutsigbare ID
 - Aksepterer DNS RR som den ikke har spurt om (jf Bonjour multicast)
 - MITM: Noen fanger opp request og sender reply som ankommer før det «ekte»

Forgiftning av klient-cache

- Mange ulike teknikker, en enkel er å lage en webside full av bilder med `height= «0»`, `width=«0»`; som så har en `src=«http://falsktunderdomene.domenet-vi-vil-forgifte.com/bildefinnesikke/»`
- Så pøser angriper på med svar på DNS-spørsmål offerets ISPs navnetjener aldri får til å besvare..

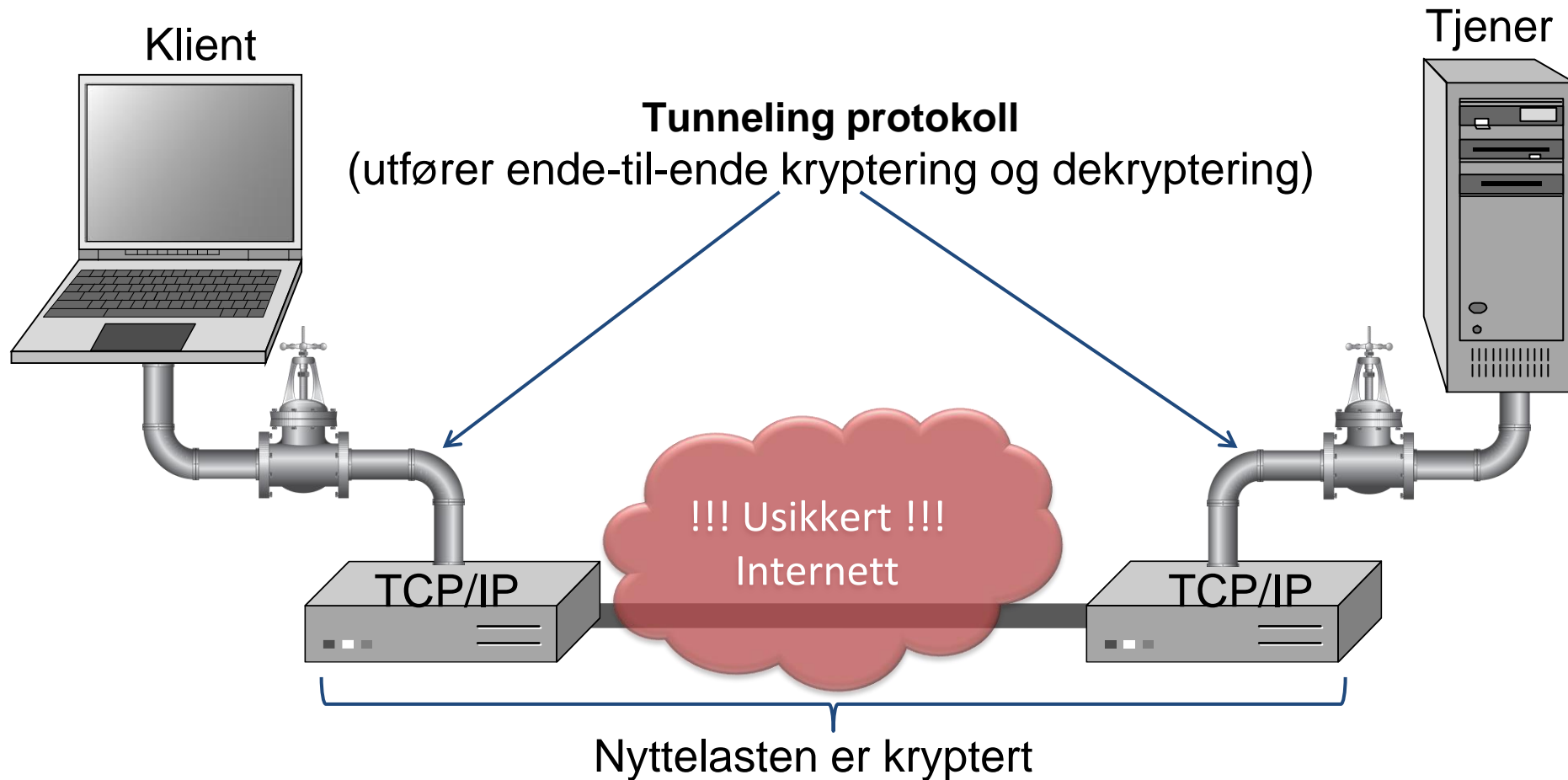


Brannmur TYPEN

- **Pakke filter** («tilstandsløse»)
 - Dropper eller aksepterer hver enkelt ankommen pakke kun ut fra regelen
- **«Tilstandsorienterte» filtre**
 - Holder oversikt over alle forbindelser
 - Kan avgjøre om en pakke er starten på en ny forbindelse, del av en etablert, eller ikke akseptabel
- **Applikasjonslag**
 - Fungerer som en «proxy» og kjenner reglene for protokoller og bestemte applikasjoner
 - Inspiserer innholdet og blokkerer det som er definert som uakseptabelt (websteder, virus, sårbarheter, ...)

Tunneling forhindrer avlytting

- Pakker sendt over Internett kan automatisk krypteres med *tunneling*



Virtual Private Networking (VPN)

- **VPN** er en fellesbetegnelse på ulike teknologier som tillater sikker tilgang til private nettverk over Internett
- **VPN** skal garantere data konfidensialitet, integritet og autentisering, til tross for usikkert transport-nettverk
- To hovedtyper
 - Remote Access VPN
 - Site-to-site VPN
- Typisk sikret med enten IPSec eller SSL/TLS

Creepy eye on the wall

- I November 2016 ble de slått opp en sak fra USA hvor en mor kom inn i rommet til barnet, og hørte en stemme i baby monitoren si «Wake up little boy, daddy's looking for you»...
- Barnet hadde klaget over at han var redd på natten fordi «telefonen pratet til ham»
- Noen hadde hacket baby monitoren og fulgt med på – og pratet med barnet over lengre tid, hva har han sagt og hva var planen?!
- <http://sfglobe.com/2016/01/06/stranger-hacks-familys-baby-monitor-and-talks-to-child-at-night/>
- <https://www.buzzfeed.com/craigsilverman/creeps-hack-baby-monitors-and-say-terrifying-thing>

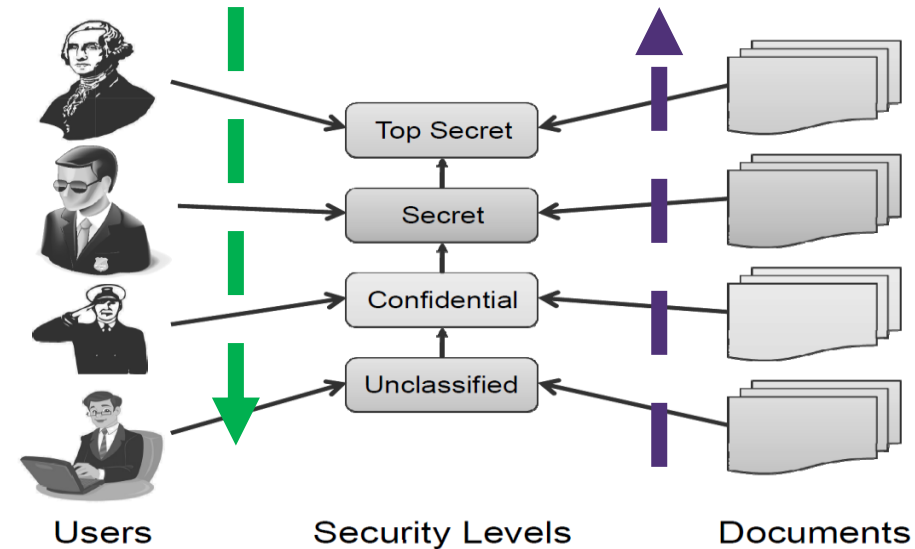
Kaffemaskiner – med mer enn kaffe

- Land har alltid spionert på hverandre
- Avlyttingsutstyr har vært forsøkt plantet hos «fienden» siden de ble oppfunnet, også gjemt i gaver – ref «The Thing» fra August 1945 til US Embassy i Moskva
- Men når vi er så paranoide i dag, så sjekkes alt grundig, spesielt gaver – men hva med ting ansatte kjøper privat og tar med seg?
- I Russland fant man avlyttingsutstyr i kaffemaskiner og strykejern som var kjøpt i butikk – kineserne hadde tilsynelatende avlyttet ALLE modellene som ble produsert i Kina for det russiske markedet!

- En sikkerhetspolicy er et sett med regler som definerer og beskriver
- **Subjekter**
 - Hvem som samhandler med systemet: individer, roller, grupper
 - Ex: Navn (og tittel), bruker/administrator/angriper/gjest...
- **Objekter**
 - Informasjons- og behandlings-ressursene som policyen er laget for å beskytte og administrere
 - Ex: dokumenter, filer, databaser, servere, arbeidstasjoner, programvare
- **Handlinger**
 - Hva en subjekt kan og ikke kan gjøre med objektene
 - Ex: Lese/skrive dokument, installere software, bruke database,...
- **Tillatelser**
 - Oversikt over sammenhengen mellom subjekter, handlinger og objekter, som klart sier hvilke handlinger som tillatt og ikke.

Bell-La Padula (BLP) modellen

- Klassisk tvungen adgangskontroll (MAC) modell for å beskytte **konfidensialitet**
- Basert på det **militære** flernivå sikkerhetsparadigmet for dokument**klassifisering** og personell**klarering**
- Strengt **lineær** orden av brukernivåer for dokumenter og adgang:



- En bruker kan **lese** alle dokumenter på eget klareringsnivå og under
- En bruker kan **ikke skrive** til lavere nivå («forhindre lekkasje»), men en bruker på lavere nivå kan skrive oppover...
- «Informasjonsendringer flyter bare oppover»

For en fullstendig informasjon om lover og forskrifter i Norge vises det til
<http://www.lovdata.no>

- *Virksomheten leder har ansvar for å påse at regelverket overholdes og skal sørge for at krav til sikkerhet innarbeides i avtaler med partnere, leverandører og andre det utveksles informasjon med*

- Standardene er en form for internkontroll, men garanterer primært **systematikken** i sikkerhetsarbeidet – ikke nødvendigvis hvor effektivt det faktisk er.
- Sikkerheten bør derfor også testes fra angriperes perspektiv.
- Slik «simulert hacking» omtales som penetrasjonstesting
- Forutsetter juridisk bindende avtale med offeret for ikke å være straffbart.
- Kan benytte metodikken fra OSSTMM
 - <http://www.osstmm.org>

PenTest «faser» ligner mye...

1. Samle informasjon
2. Scan IP-adresser
3. Fingerprint OS
4. Identifiser sårbare tjenester
5. Utnytt sårbarheten (?)
6. Fix problemene

Samle informasjon

- Finn mest mulig ekstern informasjon
 - IP-adresser
 - ripe.net
 - Domenenavn
 - Kan f.eks. bruke <http://online-domain-tools.com/>
 - Personlig info om ansatte
 - Social Engineering
 - Facebook
 - Google
 -

- Gir **enerett** til **kommersiell** utnyttelse av en **oppfinnelse** for et begrenset **tidsrom** (20 år) innenfor et juridisk område (stat).
 - **hindrer andre** i å produsere, importere og selge oppfinnelsen
 - kan lisenseres/leies ut
- Skal sikre alle tilgang til oppfinnelsen og kunnskapen bak, ved at virkemåten offentliggjøres.
 - Formålet er dermed primært å gi et tidsbegrenset (rettsbeskyttet) monopol til oppfinneren,
 - mot at h@n offentliggjør/publiserer...

Hva kan patenteres?

- En konkret, praktisk løsning på et problem der løsningen
 - har teknisk karakter
 - har teknisk effekt
 - er reproducerbar
- Oppfinnelsen må være **ny**
 - Kan ikke være omtalt i tidligere patenter, tidsskrifter, eller annet noe sted (i verden!), før datoen søknad innleveres.
- Oppfinnelsen må ha **oppfinnelseshøyde**
 - Må **skille seg vesentlig** fra tidligere kjent teknikk
 - Kan ikke bare være en logisk videreføring av tidligere teknikk...
 - Sett i sammenheng med tidligere publikasjoner kan den **ikke være opplagt** for en kyndig person

Principal of Defensive Programming

- General quality – reducing the number of software bugs and problems
- Making code comprehensive – the source code should be readable and understandable so it is approved in a code audit
- Making the software behave in a predictable manner despite unexpected input or user actions
- Vi vil fokusere i dette foredraget på det siste punktet; noen ganger referert til som et delsett som heter «sikker programmering»
- Mest fokus på dette i praksis i dag, ikke så mye terping på teorien og metodikken
- Vil også nevne begrepet "offensive programmering"



Defensive Programming

- Beskytt koden mot feil data «fra utsiden»
- Etabler sikre grensesnitt
- Valider alt ved input data gjennom grensesnittene; type, lengde, innhold i data, struktur på data
- Etabler en strategi for hvordan forskjellig typer feil data skal håndteres (erstatt med default data, fast fail, logging, osv)
- Ikke forvent at en ekstern metode kan kalles og vil oppføre seg slik den er dokumentert
- Kode for diagnoser, logging og tracing
- Forsiktig bruk av exception handling; exception handling skal ikke være «kode flyt» – men kritisk feilhåndtering

Input validering

- All input må valideres
- Ikke stol på språket eller miljøet, selv om du kjører Azure Web-applikasjoner som har en viss beskyttelse innebygd - legg til spesifikke og spesialtilpasset inndatavalidering
- Ikke stol på en Web Application Firewall, dette er en ekstra forsvarslinje, så ikke forlat din første forsvarslinje (som er inndatavalidering)
- Hvis et inndatafelt inneholder et telefonnummer; BARE tillat tall og +-tegnet
- Hvis et inndatafelt inneholder et navn; BARE tillat tegn som vanligvis brukes i et navn på språkene du antar å støtte

Eksamensforberedelse

Om årets eksamen i TK2100

- **24 timers hjemmeeksamen**
- **Vurderingsform: Bestått/Ikke Bestått**
- 8-10 oppgaver, fritekst drøftingsoppgaver, praktiske oppgaver, kanskje noe regning?
- OBS; fristen er på 24 timer, men det kreves ikke mer enn 4-6 timers jobbing!
- Formålet med oppgavene er å vise forståelse for faget
- Øvingstimer med fritekst oppgaver er veldig relevante for denne eksamensformen
- 2-3 oppgaver er praktiske, så gå gjennom notatene deres fra praktiske labøvinger 😊

Eksempel oppgave

COVID-19 krisen utnyttet av kriminelle til å utføre cyber angrep mot personer og selskaper, hvilke nye trusler ser du for deg kan oppstå som en følge av dette – hvordan kan cyber kriminelle utnytte en slik krise?

Hvordan ville du som sikkerhetsleder i et selskap gått frem for å sikre ansatte og selskapet gjennom denne krisen?

- Det er 8-10 oppgaver, så estimer 30 minutter skrijving per oppgave. Det gir deg rammene for å besvare denne oppgaven
- Dette er en oppgave som er ment å BRUKE kunnskapen du har lært i dette kurset, prøv å få med så mange elementer fra pensum som mulig – men samtidig er det drøfting så du kan ta med egne tanker og forutsetninger
- Vi har pratet om phishing angrep, det er MEST interessant å diskutere her, DNS spoofing kan også være interessant, samme kan web sikkerhet hvor angriperen tar over myndighetssider (for eksempel stored XSS), fysisk sikkerhet er interessant

Eksempel oppgave

Drøft hva som er de største truslene mot informasjonssikkerhet i et middels selskap med 50 ansatte og 100 millioner NOK i omsetning. Beskriv de viktigste tiltakene for å redusere risiko.

- Igjen - estimer 30 minutter skrijving per oppgave. Det gir deg rammene for å besvare denne oppgaven
- Det er viktig å fokusere på hvilket trusselbilde selskapet har, et vanlig selskap av denne størrelsen har ingen store aktører som går etter dem spesifikt, så de må forsvare seg mot de vanlige truslene
 - Malware angriper alle i verden uten å diskriminere (som oftest)
 - Port scan utenfra må ikke avdekke noe «spennende», har du en åpen port med Telnet vil noen bruke den – sørg for at alt ser normalt og trygt ut hvis selskapet blir portscannet...
 - Insidetrusler er alltid en trussel, hva kan en ansatt som føler seg urettferdig behandlet eller lite verdsatt finne på – har alle fulle administratorrettigheter til hele systemet?
 - Tofaktor autentisering på remote access er ansett som best practice i dag

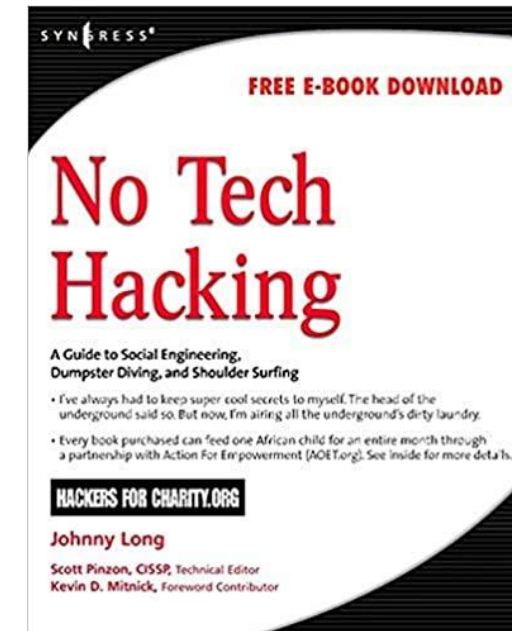
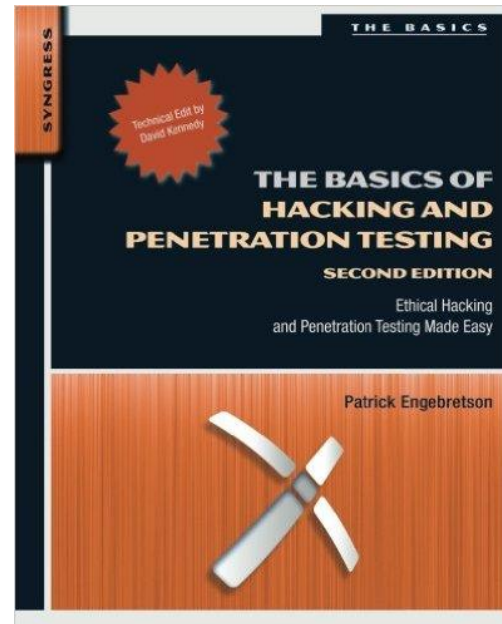
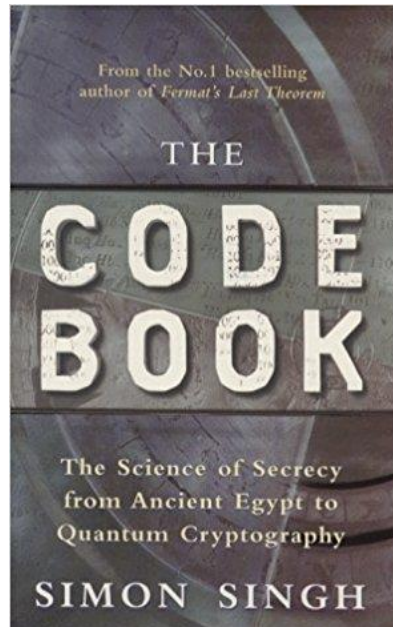
Eksempel oppgave

- Hvis man har tid ville jeg tatt noen forutsetninger for å utdype
- Behandler selskapet finansielle data?
 - Da er organiserte kriminelle en mulig trussel, de vil «investere» basert på forventet fortjeneste, det setter visse krav til styrke på kryptering, sikkerhet kan også være i flere lag, kanskje outsource sikkerhet til en tredjepart som kan gi SIEM og SOC kapabiliteter?
- Behandler selskapet helsedata?
 - Da kan andre nasjoner vise interesse, man kan bli utsatt for targeted phishing
 - Kanskje awareness, sikkerhetsopplæring og phishing øvelser er mer viktig pga trusselaktør?
- Gjør selskapet noe NOEN kan mene er skadelig?
 - Oljeselskaper kan være utsatt for aktivisme og «hacktivisme»
 - Noe som kan oppfattes som å svekke personvern; Anonymous kan være en trusselaktør?
 - Spesielle trusselaktører kan medføre et høyere nivå av sikkerhet enn størrelsen og verdien på selskapet skulle tilsi, kanskje er de fleste trusselaktører fysiske – da må man styrke den fysiske sikkerheten (tradisjonelle aktivister er i denne kategorien)

Etter eksamen

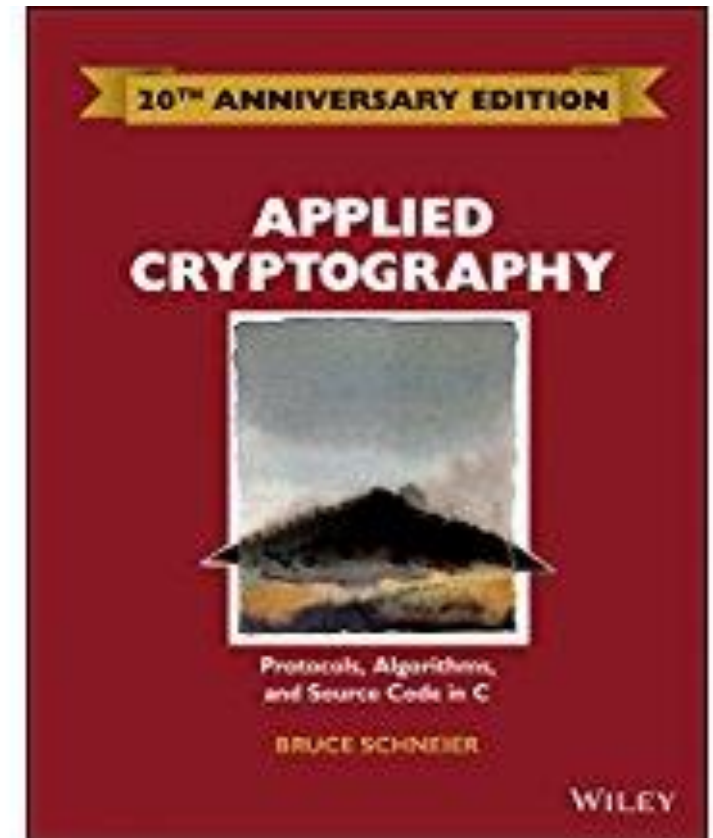
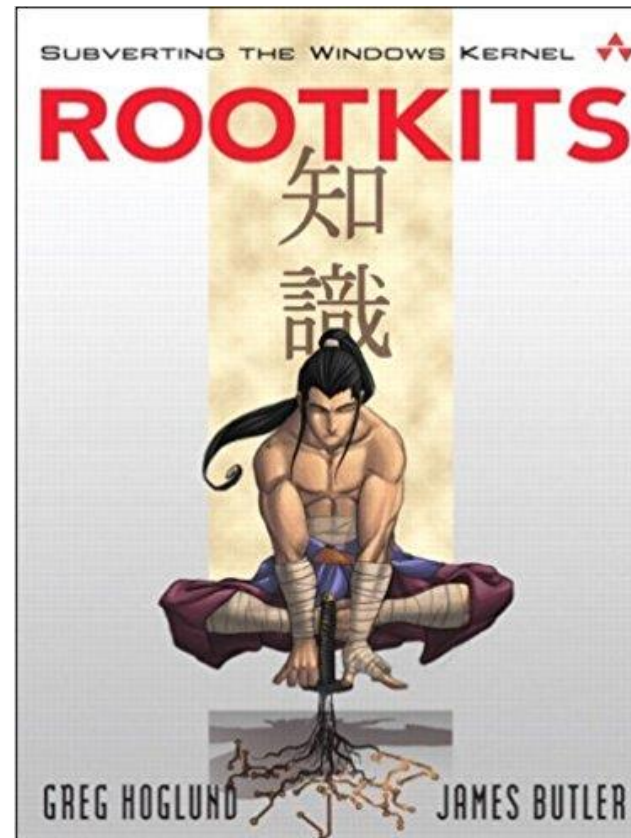
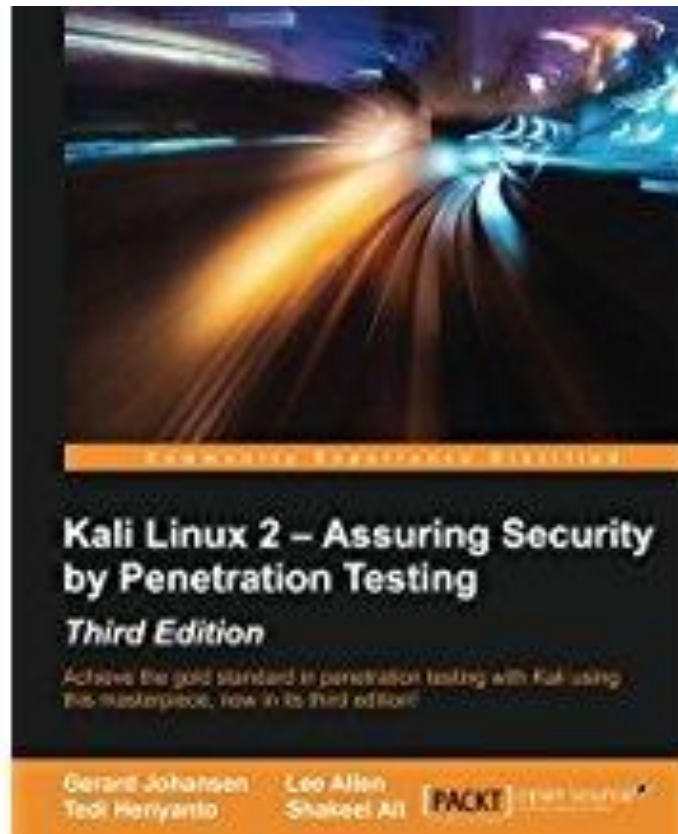
Anbefalt til **etter** eksamen

- Noen har spurt meg hvilke bøker jeg leser/har lest som jeg vil anbefale?



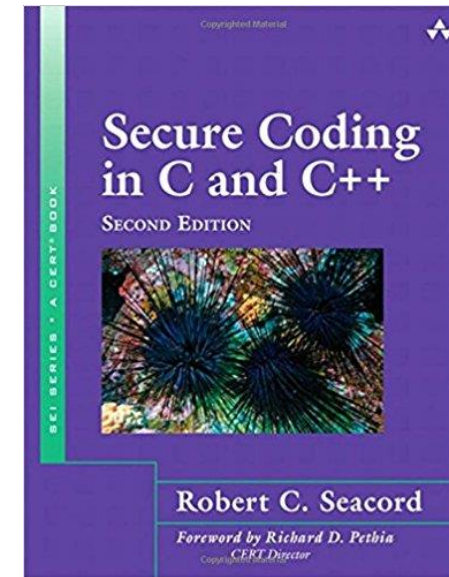
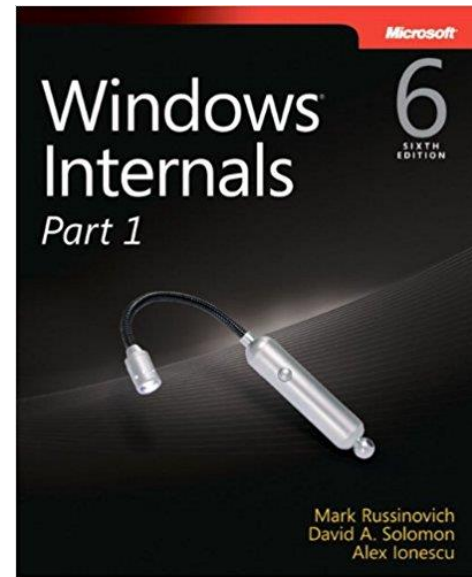
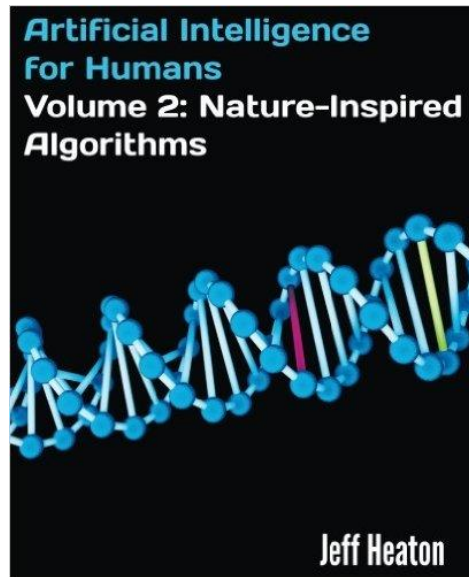
Anbefalt til etter eksamen

- Noen flere



Anbefalt til **etter** eksamen

- Ikke sikkerhets-relaterte bøker





IT = livslang læring
😊

LYKKE TIL PÅ EKSAMEN

TAKK FOR ET GODT
SEMESTER, DERE HAR VÆRT
FLINKE OG ENGASJERTE
STUDENTER!