

Introduksjon til Internett og sikkerhet

Olav Skundberg

Opphavsrett: Forfatter og Stiftelsen TISIP

L restoffet er utviklet for faget Internett og sikkerhet

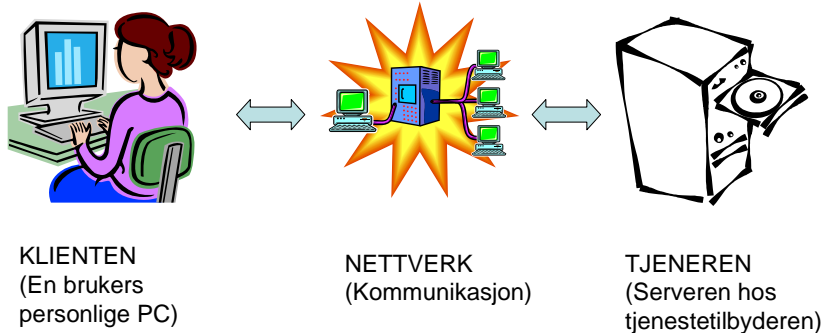
1. Introduksjon til Internett og sikkerhet

1.1. Tiln rming til faget

Internett og sikkerhet kan vinkles   en rekke m ter. Vi stiller oss f lgende sentrale sp rsm l:

- Hva er det som er utsatt for angrep
- Hvordan arter disse angrepene seg
- Hva m  en virksomhet ta hensyn til

Her er en overordnet skisse som viser alminnelig bruk av Internett med tjenester som e-post og weboppslag. Angrep kan rette seg b de mot klienten, tjeneren og det mellomliggende nettverket:



Ut fra de tre sp rsm lene og skissen har vi valgt   vinkle faget mot fire hovedtema:

Tema 1: Skadelig programvare

En type angrep kalles i dagligtalen *virus*, men kan generelt kalles *skadelig programvare* som retter seg b de mot klient og tjener. Virus er en av flere typer skadelig programvare. Her finner vi en sekk full av uhumskheter: virus, ormer, trojanske hester og en rekke eksempler p  utnyttelse av feil i programvare. Det sentrale temaet er   forst  hvordan skadelig programvare arter seg. Vi skal ogs  studere tilbudene fra leverand rer av antivirusprogram.

Tema 2: Kryptering og sikre tjenester

Kryptering er et sentralt tema for kommunikasjon mellom klient og tjener. Hvordan kan man stole p  at den andre parten virkelig er den han utgir seg for? Hvordan kan man unng  at uvedkommende fanger opp eller endrer innhold i en overf ring? Sentrale tema er krypteringsteknologi og bruk av digitale sertifikater.

Tema 3: Nettverk

Vi skal studere sikkerhetsaspekter i lokalnett (LAN) og fjernnett (WAN) og hvordan man kan sikre seg p  ulike vis mot skadelige angrep i nettet. Sentrale tema er   forst  hvordan TCP/IP, Internettets protokollfamilie, virker og hvordan brannmurer og sikre forbindelser motvirker angrep.

Tema 4: Administrative aspekter

En rekke offentlige instanser bidrar med personvernregler og retningslinjer for sikkerhetsstrategier, s  det sentrale temaet her er   gj re seg kjent med og forst  betydningen av dette.

1.2. Om skadelig programvare

Her kommer en kommentar til artikkelen *Malware* fra den engelske wikipedia, og denne artikkelen inng r i pensum: <http://en.wikipedia.org/wiki/Malware>

Hva er Malware:

Det   kunne kategorisere og forst  begrepene i tilknytning til skadelig programvare er halve jobben. Artikkelen legger i begrepet skadelig (malicious): Fiendtlig, innbryter, plagsomt. Programvare er betraktet som skadelig mer ut fra skaperens hensikter enn spesielle egenskaper eller funksjoner i programmet.

Hva er hensikten:

Merk at det i dag i hovedsak ligger former for  konomiske interesser bak angrep.

Infeksjoner: virus og ormer:

Navnene kommer fra spredningsmetoden og ikke hva de kan for rsake. Legg spesielt merke til at virus legger koden sin inn i maskinkoden til program og blir dermed utf rt hver gang programmet kj res. Ormer, derimot, startes som separate prosesser i maskinen. Ormer er spesielt rettet mot sikkerhetshull og s rbarheter i systemene og fors ker   spre seg aktivt gjennom disse.

Skjulte angrep: trojanere/ trojanske hester

En trojansk hest inneholder mer enn du ber om. En skadelig programvare kan skjule seg inni et annet program som virker uskyldig. Det er ingen grense for hva slike skjulte program kan gj re n r de f rst er installert. Aller mest fors ker de   skjule at de kj rer p  maskinen (rootkit og andre teknikker). Programmer som  pner "b kd rer" p  maskinen gj r det mulig for utenforst ende   ta fullstendig kontroll over maskinen slik at den kan fjernstyres. Et mislykket fors k p  bruk av trojansk hest finner du p  YouTube i en sketsj med Monty Python som skal innta en fransk festning:

<http://www.youtube.com/watch?v=rCOVvp-4RBo>

Andre kategorier: spyware, botnets, loggere og dialers:

Disse kategoriene om tales. Dialers er ikke s  aktuelle lenger i og med at de fleste har bredb ndstilknytning til Internett, men for de som fortsatt har oppringte forbindelser vil samtaler satt opp til betaltjenester kunne falle meget dyrt.

Hvor s rbar er man for angrep:

Her kommer man inn p  faktorer som ogs  kan inng  i en sikkerhetsstrategi, for eksempel at man ikke skal la maskiner/brukere har for store rettigheter til   utf re kommandoer.

1.3. SQL-injection

Her skal vi studere en bestemt type angrep, nemlig hvordan lovlig funksjonalitet i SQL-programvare kan utnyttes i u rlige hensikter. En forklaring p  SQL-injection gis i podcast nr 87 av Steve Gibson. Han er en kjendis p  området og produserer ukentlige podcast om forskjellige sikkerhetstema. Det er derfor verdt   lytte til han, og eksemplet med SQL-injection er tatt med for   inspirere til videre lytting p  andre podcasts.

Omtale av Steve Gibson og Security now!, med en kronologisk liste av alle hans podcasts, finner vi (igjen) p  den engelske wikipedia: http://en.wikipedia.org/wiki/Security_now

Hver podcast varer omlag en time og har mye "smalltalk", du kan derfor starte avspilling nr 87 om SQL-injection etter 20 min 40 sek, og spille av 12 minutter for   f  med essensen. Her er noen engelske ord du kanskje trenger oppfriskning p : exploit (utnytte), leverage (vektarmkraft, p virke), compromised (kompromittert, avdekket), cleansing (rense), scrutinizing (granske n ye). Andre ord kan du sl  opp p  for eksempel www.ordnett.no

SQL er forkortelsen for Structured Query Language, hvilket betyr at dette er et sp rrespr k mot databaser. Uttrykket *sp rrespr k* m  tolkes sv rt vidt, for det kan gis kommandoer b de for   opprette og slette innhold og hele filer. Et element rt SQL-uttrykk kan se omtrent slik ut:

```
Select * from Tabell where Brukernavn = ole and Passord = 123  
(Velg ut alt fra Tabell hvor kolonne Brukernavn inneholder verdien 'ole' og kolonne Passord inneholder verdien '123')
```

SQL-sp rring kan brukes til   kontrollere p logging p  webtjenester hvor alle brukerne er registrert i en SQL-database p  tjeneren. Hvis brukernavnet og tilh rende passord finnes, f r vi et gyldig svar fra databasen. Man skriver inn de brukervariable (navn og passord) p  webside som vises p  PC og sender dette til tjeneren ved   trykke Enter. Merk at kontrollen med SQL-sp rringen utf res p  tjenersiden, og m  bruke de to variable som er oversendt. SQL-sp rsm let kan derfor v re programmert slik p  tjeneren:

```
Select * from Tabell where Brukernavn = Variabel1 and Passord = Variabel2
```

SQL-spr ket tillater komplisert logikk. Trikket med SQL-injection g r ut p    skrive lovlige SQL-uttrykk i input til variabelfeltet. N r dette blir satt inn i variabelfeltet kan vi f  f lgende uttrykk (som et meget enkelt eksempel):

```
Select * from Tabell where Brukernavn = ole and Passord = gjetting OR 1=1
```

Effekten av denne modifiserte SQL-sp rringen er at vi alltid vil f  OK-svar fra databasen, fordi $1=1$ alltid er gyldig, man har dermed kommet seg forbi passordkontrollen. Dette er det enkleste eksemplet, man kan bygge ut SQL-uttrykket til   utf re kompliserte operasjoner p  databasen og kompromittere andre tjenester p  maskinen.

For   gardere seg mot SQL-injection m  man ha programvare som stopper dette, det vil si at programmereren kjenner til muligheten og har tatt h yde for slike angrep ved   kontrollere innholdet i variabelfelt.

1.4. Fagwiki

Det er tatt i bruk en wiki spesifikt for dette faget. Fagwikien deles kun av studenter ved AITeL i fjernundervisning og p  campus og er beskyttet av brukernavn og passord. Programvaren heter Mediawiki og er n yaktig den samme som er brukt i den mer kjente offentlige Wikipedia. N yaktig som p  Wikipedia m  du opprette din egen brukerkonto for   kunne skrive egne innlegg og diskutere eller justere i andres innlegg. Hensikten med denne bruken er   l re:

- en metode for   finne og dele informasjon og kunnskap
- hvordan man bruker Mediawiki konkret
- det   formulere egne bidrag

Du finner fagwiki her: http://wiki.aitel.hist.no/ish/index.php/Main_Page
(krever brukernavn og passord – vises i it's learning)