dSik - Network security

Lukas Peter Jørgensen, 201206057, DA4

16. juni 2014

1 Sikkerhedsmål

CAA

CAA står for:

Confidentiality: Information skal holdes hemmelig for uvedkommende, gælder både under forsendelse, opbevaring og behandling af data

Authenticity: Informationen er autentisk, den er ikke blevet manipuleret af en uautoriseret person.

Availability: Systemer skal være tilgængelig når de skal bruges

Definition af et sikkert system

Det er typisk svært eller umuligt at bevise et system er sikkert. Tit bliver man nødt til at lave mange antagelser om angriberens muligheder for at bevise at et system er sikkert og disse antagelser er typisk forkerte.

Vi bruger udtrykket "sikret system"istedet for "sikkert system"da det blot indikerer at vi har sikret systemet mod nogle bestemte tilfælde. Disse tilfælde defineres ved en *Sikkerhedspolitik* på baggrund af en *Trusselmodel*, og implementerer herefter denne sikkerhedspolitik vha. nogle *Sikkerhedsmekanismer*. På den måde får vi, at et sikret system kan beskrives som:

$$Sikretsystem = Sikkerhedspolitik + Trusselsmodel + Sikkerhedsmekanismer$$

2 Netværkssikkerhed

Netværkssikkerhed er mange ting, og overlapper ofte med Systemsikkerhed. Jeg har valgt at fokuserer på de sikkerhedsmekanismer der direkte arbejder med netværkstrafik.

Prevent-Detect-Recover Policy

Selvom emnet ikke handler om sikkerhedspolitikker er det en god idé at nævne at i netværkssikkerhed typisk arbejder med Prevent-Detect-Recover politikken. Da man har et first line of defence hvor man prøver at forhindre angriberen i at komme ind i systemet, sker det alligevel forsøger man at samle så mange informationer som muligt om ham. Til sidst sørger man for at der er et system til at undgå at systemet bryder sammen hvis angriberen kommer igennem sikkerheden.

3 Authenticated Key Exchange

For at bruger A og bruger B kan kommunikere med hinanden skal de udveksle sessionsnøgler. For at dette kan ske sikkert skal der overordnet set gælde tre ting:

- Hvis A ønsker at kommunikere med B og B ønsker at kommunikere med A, og begge konkluderer at protokollen til nøgleudveksling var succesfuld, så er de enige om sessionsnøglen K.
- Hvis A ønsker at kommunikere med B, og konkluderer at protokollen til nøgleudveksling var succesfuld, så skal det være tilfældet at B deltog med intentionen om at kommunikere med A og at ingen andre end A og B har adgang til sessionsnøglen K. Samme regel gælder for B i forhold til A.
- Den aftalte sessionsnøgle K skal være "frisk"i den forstand at den ikke må have været brugt før. En eventuel angriber må altså ikke kunne tving A eller B til at bruge en gammel sessionsnøgle.

Generelt kan vi dog ikke kræve at begge brugere er enige om hvorvidt protokollen var succesfuld. Angriberen kan eks. forhindre en verifikation i at dukke op. - "Two Army Problem".

Needham-Schroeder

Dårligt eksempel:

1. A vælger et nonce n_A og sender $E_{pk_B}(ID_A, n_A)$ til B.
2. B dekrypterer beskeden, tjekker at ID_A er gyldig (ud fra en CA) og sender $E_{pk_A}(n_A, n_B)$ til A.
3. B dekrypterer og tjekker at den korrekte værdi for n_B er i resultatet.
4. Herefter kan A og B bruge n_A, n_B til at skabe en sessionsnøgle K.

Sårbar overfor man-in-the-middle attacks.

4 SSL/TLS

SSL standarden kræver at begge parter har et certifikat. Dette er dog upraktisk så i implementationer har typisk kun 1 af parterne et certifikat.

SSL består af adskillige protokoller som f.eks.

Handshake Protocol Handshake protokollen sørger for at begge parter har udvekslet nøgler og har et fælles *cipher spec*.

Change Cipher Spec Protocol Denne protokol er blot en enkelt meddelelse en part sender til den anden part for at indikere de nu skal til at bruge den aftalte *cipher spec* og de aftalte nøgler.

Alert Protocol Signalerer fejl til den anden part.

SSL Key Exchange

1. C sender en "hello"besked indeholdende en nonce n_C den har valgt.
2. S sender en nonce n_S tilbage, samt dens certifikat $Cert_S(ID_S, pk_S)$.
3. C verificerer certifikatet ved at kontakte en CA og vælger herefter en *pre master secret*(pms) tilfældigt. C sender så $E_{pk_S}(pms)$, dens certifikat $Cert_C(ID_C, pk_C)$ og dens signatur af konkateringen af de to nonces og pms krypteret, altså $sig_C(n_C n_S E_{pk_S}(pms))$.
4. S verificerer så $Cert_C$ og sig_C vha. en CA og hvis de er iorden, dekrypteer den pms.
5. S sender herefter en "finished"besked til C, indeholdende et MAC på alle beskeder sendt mellem parterne i handshaket, alt sammen krypteret med pms som nøgle.

Trin nummer 6, kaldes *Final Authentication of Views* metoden og gør at en angriber ikke kan agere "man in the middle".

5 IPSec VPN

Sker mellem transport laget og IP laget. Modsat SSL hvor programmet skal understøtte kryptering for at lave en sikker forbindelse, krypterer IPSec alt information på computeren. IPSec bruger *Internet Key Exchange* (IKE) protokollen.

Det fungerer ved at parterne bruger offentlige nøgler, eller en pre-shared secret til authentikere sig i forhold til hinanden og bruger så Diffie-Hellman Key Exchange algoritmen.

1. Man vælger et tal g i intervallet $0 \ldots p-1$ hvor p er et stort primtal.
2. A vælger nu et tilfældigt tal a og sender $g^a \mod p$ til B.
3. B vælger ligeledes et tilfældigt tal b og sender $g^b \mod p$ til A.
4. A udregner nu $(g^b \mod p)^a \mod p$ og B udregner tilsvarende $(g^a \mod p)^b \mod p$
5. Det viser sig så at dette betyder at begge parter nu har $g^{ab} \mod p$ og kan bruge det som en fælles nøgle.

Diffie-Hellman baserer sig på idéen om at diskrete logaritmere er et svært beregneligt problem a ud fra $g^a \mod p$.

6 Firewalls

Firewalls bestemmer hvad og med hvem vi vil kommunikere.

Packet Filtering Firewalls

Blokerer packets baseret på afsender, modtager etc. F.eks. kunne den blokere al kommunikation bortset fra port 80 til webserver el.lign.

Hvis der er et sikkerhedshul, så ville en angriber kunne bryde ind på denne og derved komme uden om firewallen gennem webserveren. For at løse dette kan man bruge 2 packet filtre der isolerer webserveren. Webserveren vil så ligge i en *Demilitarized Zone* (DMZ)

Packet filtre er for simple til at være sikre uden at gøre dem ulidelige.

Proxy Firewalls

Man tillader kun kommunikation gennem proxy firewallen. Udefra ligner systemet en enkelt maskine, angribere kan ikke kontakte andet end firewallen.

Er et problem hvis applikationer ikke understøtter at bruge en proxy.

Stateful Inspection Firewalls

Kan tage højde for f.eks. om en pakke hører til en eksisterende forbindelse og kan afvise pakker på dette grundlag. F.eks. kan man tillade mange udadgående forbindelser, men få indadgående.

Kan også fungere som en Proxy Firewall.

7 IDS

Intrusion Detection Systems (IDS), minder lidt om Stateful Firewalls, da de ligeledes overvåger og analyserer f.eks. processer, brugerer, trafik og lignende.

Regel-baseret IDS

Regel-baserede IDS'er definerer et kæmpe sæt af regler for hvad normal adfærd er, og reagerer hvis adfærden afviger kraftigt fra disse regler.

Statistik-baseret IDS

Baserer opfattelsen af normal adfærd, ud fra statistikker af brug over en længere periode.

Firewalls $\rightarrow$ Prevent.
IDS $\rightarrow$ Detect.
Recover kan ske ved database backup og IP blokering.