dSik - System security and models

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 Systemsikkerhed

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 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.

4 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.

5 Malware

Trojanske heste

Vira

Orme

6 Access Control

Access Control kontrollerer hvad en bruger har adgang til at gøre, således at ondsindede brugere ikke sletter vigtige file og lignende.

En meget simpel løsning er en Access Control Matrix, hvor indgang $A[s, o]$ er alle opeartioner en bruger s har ret til at gøre på objekt o. Denne løsning skalerer dog elendigt, så ofte bruges andre løsninger.

Access Control List

Her deler man rettighederne op i lister af brugere. F.eks. i *nix systemer hvor man gemmer rettighederne i, ejer, ejergruppe og alle andre. Disse rettigheder bliver så gemt på filen.

User Capabilities

Her gemmer man rettighederne på brugerne istedet og giver dem roller som f.eks. "normal bruger", "superbruger", "administratorer".

Opdatering af ACM

Ofte vil der være interesse i at kunne ændre disse rettigheder løbende i et system. Så der skal både være en måde til at ændre rettighederne, og en måde at bestemme hvem der har ret til at ændre rettighederne.

Generelt er der to fremgangsmåder:

- Mandatory Access Control
- Discretionary Access Control

I den førstnævnte er det ikke muligt at ændre i rettighederne af brugerne, men i den anden har man mulighed for at ændre rettighederne.

Man er selvfølgelig nødt til at sørge for at man ikke kan give højere rettigheder end man selv har fået. (I hvert fald ikke uden hjælp fra en administrator). Man kan (forsøge) at teste om dette er tilfældet ved at se om der er et begrænset antal kommandoer c for hvilket matrixen A ændres til matrixen A', hvor operationen $r \in A'[s, o]$ og $r \notin A[s, o]$. Hvis svaret er nej kaldes matrixen A for sikker i forhold til r. Følgende kan det vises at:

- Hvis sættet af kommandoer kun består af en enkelt operation, så er det bestemmeligt at A er sikker i forhold til r.

- Hvis sættet af kommandoer består af mere end en operation, så er det ubestemmeligt om A er sikker i forhold til r.
- Hvis antallet af brugere er endeligt, så er det altid bestemmeligt om A er sikker i forhold til r

Derfor kan vi generelt ikke sige om A er sikker i forhold til r, men nogle gange kan vi bevise lidt.

7 Sikkerhedspolitikker

Sikkerhedspolitik En sikkerhedspolitik er en kort beskrivelse af de sikkerhedsmål vi har for systemet og en højniveau strategi for at opnå disse mål.

Trusselsmodel En trusselsmodel er en beskrivelse af de mulige angreb vi vil have vores system er beskyttet imod.

Sikkerhedsmekanismer Sikkerhedsmekanismerne er de tekniske og administrative løsninger vi bruger til at opnå vore sikkerhedsmål.

Indtil videre har vi primært set på Trusselsmodeller og Sikkerhedsmaknismer, nu er det Sikkerhedspolitikker.

I en sikkerhedspolitik definerer man, formelt set, hvilke "states"vi ønsker systemet skal kunne være i, og vores højniveau strategi beskriver hvordan vi har tænkt os at garantere systemet kun kan gå ind i disse sikre "states".

Sikkerhedspolitikken skal også omhandle den såkaldte "Trusted Computing Base"(TCB) som er den del af systemet man altid skal kunne regne med opfører sig "ordentligt".

Bell-Lapadula

Først definerer vi en lattice, som er et begrænset sæt S og en relation $\leq$ hvor vi får elementer $a, b, c \in S$ hvor der skal gælde:

- $a \leq a$
- $a \leq b$ og $b \leq a$ betyder at $a = b$
- $a \leq b$ og $b \leq c$ betyder at $a \leq c$

Man bruger en lattice til at beskrive bruger privilegier i forhold til hinanden.

Der er andre aspekter til en lattice som greatest lower bound og least upper bound, men det springer vi over.

Bell-Lapadula er beregnet til at opnå confidentiality. Dette gør den ved at definere en række klassificerings niveauer som en mængde lattices f.eks. $public \leq secret \leq topsecret$ med to regler:

No read up Subject s må læse fra object o kun hvis $C(s) \geq C(o)$.

No write down Subject s må skrive tilobject o kun hvis $C(s) \leq C(o)$.

Biba går efter integritet af information fremfor confidentiality, så den omdefinerer reglerne til at være de omvendte:

No read down

No write up

Chinese Wall

F.eks. konsulentfirma:
Man definerer et predikat $compete(c_1, c_2)$ hvor c_1 og c_2 så evaluerer predikatet til true hvis disse klienter er konkurrenter.

En ansat s kan så kun tilgå c_1 hvis der ikke findes et c_i således at $compete(c_1, c_i) = true$.

Seperation of Duty

Handler generelt om at dele beslutningen ud på flere personer. To måder: *Dual Control* og *Functional Seperation*. Dual Control skal have tilladelse fra alle på en gang, mens Functional Seperation spredes ud på seperate tidspunkter.