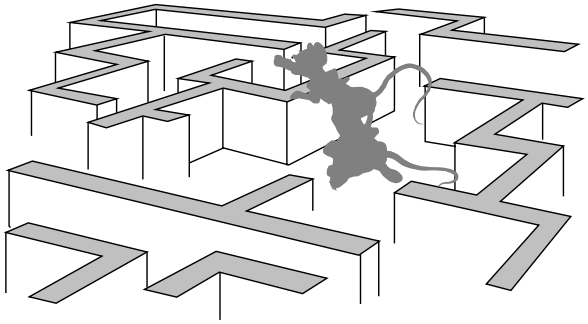
Vilseledande system

Ett examensarbete i datalogi

Utfört av:
Klas Evertsson 661125, GU
Patrik Josefsson 690626, GU

Handledare:
Martin Jigstam, EMW
Professor Erland Jonsson, CTH

Examinator:
Professor Erland Jonsson, CTH



Sammanfattning

Vilseledande system är ett begrepp som på senare år har börjat användas inom området datasäkerhet. Ett vilseledande system är ett komplement till brandväggar och intrångsdetekteringssystem. Det är ett system med en eller flera komponenter vars uppgift är att vilseleda en inkräktare i ett nätverk. Detta examensarbete behandlar huruvida användning av vilseledande system ökar skyddet mot intrång i nätverk. Syftet med vilseledande system är att de ska göra angripare osäkra på vad som är riktig respektive falsk information, de ska förbruka angriparens resurser, uppehålla angriparen så att säkerhetspersonal har möjlighet att spåra angriparen och reagera med motåtgärder samt att samla information om hur angriparen går till väga när de hackar sig in i system. I rapporten tar vi upp vilka metoder det finns för att vilseleda, vilka komponenter som ingår i ett vilseledande system och vad som tidigare har gjorts inom området.

Vi har dragit slutsatsen att med kunnig personal och med ett väl genomtänkt vilseledande system så kan säkerheten ökas. Det är dock inte trivialt att sätta upp ett sådant system och ett felkonstruerat system kan istället leda till sämre säkerhet.

Abstract

Deception systems is a concept that has been used in computer security for a few years. A deception system is a complement to firewalls and intrusion detection system. It is a system with one or several components whose purpose is to deceive an intruder in a network. This master thesis investigates whether the use of deception system gives a higher protection against intrusions in a network. The purpose with a deception system is to make the intruder unsure whether the information is false or true. It should exhaust the intruder's resources, detain the intruder so that security personnel has the opportunity to track the intruder and react with countermeasures and to collect information about how the intruders act when they hack the system. This report discusses deception methods, which components that are included in a deception system and what has been done in the area previously.

One conclusion is that security can be increased using competent personnel and a well-planned deception system. It is not trivial to set up a system like this and a badly constructed system may very well introduce new vulnerabilities.

Förord

Detta är ett magister examensarbete i datalogi vid Institutionen för Datavetenskap vid Chalmers Tekniska Högskola/Göteborgs Universitet. Examensarbete är utfört på Ericsson Microwave i Mölndal, avd FT/K. Handledare från institutionen har varit professor Erland Jonsson och från Ericsson har det varit Martin Jigstam. Vi vill tacka våra handledare för god handledning. Vi vill också tacka Torbjörn Andréasson för att vi fick göra vårt examensarbete på Ericsson.

Vi vill tacka Linda Barman och Kristina Hedenblad för att de har tagit sig tid till den språklig bearbetningen av vårt examensarbete.

Vi vill också tacka våra familjer för det stöd och den avlastning vi har fått från våra dagliga sysslor under vår utbildning.

Slutligen tackar vi Maria Forsberg och Mikael Hedström för alla era glada tillrop och onödiga kommentarer.

Innehåll

1. INLEDNING 11

1.1. BAKGRUND 11
1.2. SYFTE OCH PROBLEMFORMULERING 12
1.3. AVGRÄNSNINGAR 12

2. SÄKERHET OCH DEFINITIONER 13

2.1. VAD ÄR SÄKERHET? 13
2.1.1. HUR MÄTER MAN SÄKERHET? 14
2.2. DEFINITIONER 15

3. BRANDVÄGGAR 17

3.1. PAKETFILTRERING 18
3.2. PROXYSERVER 19
3.3. BRANDVÄGG MED DUBBLA NÄTVERKSKORT 20
3.4. PAKETFILTRERANDE VÄRDDATOR 20

4. INTRÅNGSDETEKTERINGSSYSTEM 22

4.1. VAD ÄR INTRÅNGSDETEKTERING? 22
4.2. VARFÖR SKA DU ANVÄNDA DIG AV IDS? 22
4.3. INTRÅNGSDETEKTERINGSMODELLER 23
4.4. INFORMATIONSKÄLLOR 23
4.4.1. NÄTVERKSBASERAT IDS 24
4.4.2. VÄRDDATORBASERAT IDS 25
4.4.3. DISTRIBUERAD DETEKTERING 26
4.5. HUVUDPRINCIPER FÖR IDS 26
4.5.1. AVVIKELSEDETEKTERING 27
4.5.2. MISSBRUKSDETEKTERING 29
4.6. RESPONS 32
4.6.1. PASSIV RESPONS 32
4.6.2. AKTIV RESPONS 32
4.7. VAR SKA IDS PLACERAS? 33
4.8. ÖNSKVÄRDA EGENSKAPER FÖR IDS 34

5. OLIKA TYPER AV ATTACKER 36

5.1. SKANNINGATTACK 36
5.2. DENIAL OF SERVICE (DOS) 36
5.3. PENETRATIONSATTACKER 37

6. VILSELEDANDE SYSTEM 38

6.1. TEKNIKER FÖR ATT VILSELEDA 38
6.1.1. VISA INFORMATION 38
6.1.2. DÖLJA INFORMATION 40
6.2. VAD ÄR ETT VILSELEDANDE SYSTEM? 41
6.2.1. HONEYPOT 41
6.2.1.1. Fördelar med honeypot 43
6.2.1.2. Nackdelar med honeypot 44

6.2.1.3. Vad ska man tänka på när man bygger en honeypot? *44*
6.2.2. IRONBOX 45
6.2.3. KAMELEONTMASKIN 46
6.2.4. HONEYNET 46
6.3. VAD BÖR MAN TÄNKA PÅ NÄR MAN BYGGER ETT VILSELEDANDE SYSTEM? 47
6.3.1. VAR SKA INSAMLAD DATA SPARAS? 48
7. EXISTERANDE VILSELEDANDE SYSTEM 49
7.1. AN EVENING WITH BERFERD …..... 49
7.2. THE HONEYNET PROJECT 49
7.3. PRODUKTER SOM FINNS UTE PÅ MARKNADEN 50
7.3.1. DTK 50
7.3.2. SNET 50
7.3.3. MANTRAP 51
7.3.4. SPECTER 52
7.3.5. CYBERCOP STING 53
8. ANALYS I.....54
8.1. INLEDNING 54
8.2. HONEYPOT I DET PRODUKTIVA NÄTET 54
8.2.1. VAD UPPTÄCKER BRANDVÄGGEN? 54
8.2.2. VAD UPPTÄCKER IDS? 56
8.2.3. VAD UPPTÄCKER EN HONEYPOT? 56
8.3. HONEYPOT I BRANDVÄGG 57
8.3.1. ANALYS AV IRONBOX 58
8.3.2. NACKDELAR MED LOCKBETE I BRANDVÄGG 59
8.4. HONEYPOT AVSKILJT FRÅN DET PRODUKTIVA DELNÄTET 59
8.4.1. TEKNIKEN MED HONEYNET 60
8.4.2. TEKNIKEN MED KAMELEONTMASKINER 62
8.5. DISKUSSION 65
8.5.1. TÄNKT MÅLGRUPP FÖR VILSELEDANDE SYSTEM 65
8.5.2. VISA PORT 365 65
8.5.3. INSTALLATION OCH ADMINISTRATION 65
8.5.4. FÖRDELAR MED VILSELEDANDE SYSTEM 66
8.5.5. NACKDELAR MED VILSELEDANDE SYSTEM 66
8.5.6. OVERHEAD 67
8.5.7. OMOGEN MARKNAD 67
9. ANALYS II 68
9.1. TEST AV HONEYNET MOT INTERNET 68
9.2. TEST I VERKLIG MILJÖ 68
9.3. TEST I SLUTEN MILJÖ 69
10. RESULTAT OCH SLUTSATSER.....70
11. ETISKA SYNPUNKTER PÅ VILSELEDNING.....72
12. REFERENSER OCH LITTERATURFÖRTECKNING.....74
13. ORDLISTA OCH INDEX.....77

14. FÖRKORTNINGAR 80

Figurer

Figur 1. Datasäkerhet och dess aspekter 13
Figur 2. En brandvägg som separerar Internet från det lokala nätverket 17
Figur 3. Datapaket 18
Figur 4. Proxyserver 19
Figur 5. Dual home gateway 20
Figur 6. Paketfiltrerande värddator 20
Figur 7. Nätverks- och värddatorbaserat IDS 24
Figur 8. Avvikelsedetektering 27
Figur 9. Missbruksdetektering 30
Figur 10. Placering av IDS i ett nätverk 34
Figur 11 Honeypot i LAN 55
Figur 12 Brandvägg med honeypot 57
Figur 13 Vilseledande system med parallella brandväggar 60
Figur 14 Vilseledande system 61
Figur 15 Kameleontmaskin 63
Figur 16 Brandvägg med kameleontportar. 64

1. Inledning

1.1. Bakgrund

Det finns mycket som är bra med Internet. Uppkopplad mot Internet får du tillgång till en mängd olika tjänster: e-post, informationssökning, filöverföring, chatt, spel, nyhetsbrev, bankärenden m.m. Det finns också baksidor med Internet. När du är uppkopplad mot Internet kan andra, som också är uppkopplade, nå din dator. Det kan då vara någon med illvilliga avsikter som försöker komma åt din dator.

Den ökade användningen av datorer och datorbaserade system har gjort att det blir allt mer viktigt att skydda datorer och datornät från intrång. De flesta företag och många privatpersoner sköter nu sina bankärenden via Internet. Företag har känslig och värdefull information lagrade i sina datorsystem. Allt fler människor är i sitt dagliga arbete beroende av datorer och att kunna koppla upp sig mot Internet. Därför är det viktigt att ha skydd mot olika typer av intrång i datasystem.

För några år sedan var det bara så kallade knäckare[1] med god kunskap om datorsystemets uppbyggnad som gjorde intrång. Idag kan även personer med relativt lite kunskap om datorns uppbyggnad ta sig in med hjälp av olika program som finns att ladda ner från Internet.

Inom försvarsmakten har man länge använt sig av lockbeten och skenmanövrar, för att försöka vilseleda sina motståndare. Den förmodligen största och mest kända vilseledning som har gjorts, är den som gjordes under andra världskriget inför och under ”Operation Overlord”, när de allierade 1944 landsteg i Normandie. De allierade lyckades bl.a. få tyskarna att tro, i flera dagar, att landstigningen i Normandie bara var en skenmanöver och att den riktiga invasionen skulle ske vid Calais [KOC 92].

Knäckare har länge använt sig av förvillning och vilseledning för att komma in i datorsystem. Med så kallad social ingenjörskonst (ung. ljuga sig till information) kan t.ex. angriparen låtsas vara en systemansvarig som ringer en nyanställd på ett företag och frågar efter dennes lösenord [SLA 95].

Ett säkerhetskonsultsföretag i USA blev anlitat för att kontrollera ett företags sårbarhet mot knäckare [ZON 01]. De började med att använda sig av social ingenjörskonst. De ringde upp tio stycken av sin klients kunder och på ett övertygande sätt låtsades de vara klienten. De fick ut värdefull information om användarkonton och lösenord från alla tio kunderna.

[1] Knäckare (cracker) är en person som försöker forcera datorsystems säkerhetsspärrar i syfte att sabotera programvara eller att komma åt, utnyttja eller eventuellt förändra skyddad information. Gränsen för om en person ska kallas hackare eller knäckare kan ibland vara svår att dra, men vi tycker ändå att det finns skäl att försöka skilja entusiaster från brottslingar.

För att öka datasäkerheten ytterligare har konceptet vilseledning börjat användas. Meningen är att de ”vilseledande systemen” ska göra angriparen osäker på vad som är riktig och vad som är falsk information, förbruka angriparens resurser, uppehålla angriparen så att säkerhetspersonal har möjlighet att spåra angriparen och reagera med motåtgärder samt att samla information om angriparens tillvägagångssätt.

1.2. Syfte och problemformulering

Syftet med denna rapport är att definiera ”vilseledande system”, vidare att undersöka vilka tekniker det finns för att vilseleda, vilka för- och nackdelar som finns med dessa tekniker. Vi ska även sätta upp ett färdigt system som använder sig av en eller flera av dessa tekniker.

Vår frågeställning är: Kan säkerheten ökas med hjälp av vilseledande system och i så fall hur?

1.3. Avgränsningar

Eftersom vi endast har 20 veckor på oss att göra detta examensarbete, innebär det att vi inte hinner behandla alla områden inom säkerhet, även om dessa är intressanta i sig. Vi kommer bland annat inte att gå in på intrångstoleranta system, antivirusprogram, autentisering och kryptering i denna rapport.

Vi kommer att göra mer specifika avgränsningar efterhand i rapporten. Detta för att läsaren ska få en bättre förståelse för vår avgränsning.

Vi går inte in på att beskriva hur implementering går till av ett vilseledande system utan vi diskuterar endast detta på en abstrakt nivå.

En större testning av ett vilseledande system har inte gjorts eftersom vi bedömt att det skulle hamna utanför tidsramen för detta examensarbete.

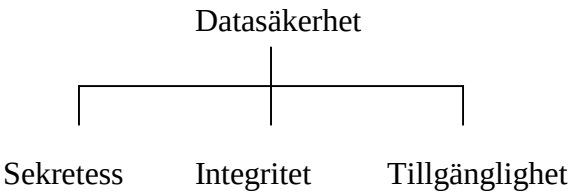
2. Säkerhet och definitioner

I det här kapitlet går vi igenom några viktiga termer som används i denna rapport (avsnitt 2.1). För att läsaren ska få en bättre förståelse för hur de vilseledande systemen fungerar ges först en övergripande beskrivning av hur brandväggar (kapitel 3) respektive intrångsdetekteringssystem (kapitel 4) är uppbyggda och hur de fungerar.

2.1. Vad är säkerhet?

Datasäkerhet är ett datorsystems eller ett nätverkssystems egenskaper som specificerar att systemen i fråga och deras komponenter går att lita på och att de agerar som förväntat när det gäller att skydda deras ägare och användares information. Datasäkerhet brukar delas upp i tre olika grupper, se figur 1, som i vanligt tal kallas för ”CIA”. Vi använder ITSEC’s (Information Technology Security Evaluation Criteria) [ITS 91] definition av ”CIA”:

Confidentiality:	Sekretess, förhindrande av att obehöriga får tillgång till information.
Integrity:	Integritet, förhindrande av att obehöriga modifierar informationen.
Availability:	Tillgänglighet, förhindrande av obehörigt undanhållande av information eller resurser.



Figur 1. Datasäkerhet och dess aspekter

Datasäkerhet beskriver inte enbart den service systemet levererar till sin omgivning (tillgänglighet), utan även systemets möjligheter att skydda sig mot externa attacker (integritet). CIA täcker det mesta om säkerhet men inte allt; autentisering och oavvisbarhet finns inte med eftersom dessa representerar mekanismer med vars hjälp CIA kan uppnås. Med autentisering avses verifikation av identiteter mellan de kommunicerande parterna och med oavvisbarhet menas förhindrande av möjligheten för kommunicerade parter att vägra sändning eller kvittering av ett meddelande [LIN 99].

2.1.1. Hur mäter man säkerhet?

Hur mäter man då säkerhet? Ja, det är inte så enkelt som man kanske skulle kunna tro och det finns ännu ingen generell metod för att mäta säkerhet, än mindre finns det någon enhet för säkerhet. Det gamla ordspråket ”ingen kedja är starkare än dess svagaste länk” belyser problemet något. Ta t.ex. ett hus med tre likadana ytterdörrar. Om man byter ut två av de gamla dörrarna mot två nya dörrar med bättre lås m.m. Är det då riktigt att säga att husets skydd mot inbrott har ökat? Det finns ju fortfarande en gammal dörr kvar. Får man däremot bara göra ett försök så kan man statistiskt påvisa; att det två gånger av tre försök, har blivit svårare d.v.s. att det tar längre tid eller kräver bättre verktyg m.m.

Ofta relaterar man säkerhet i hur svårt det är att ta sig förbi säkerhetsmekanismerna d.v.s. hur lång tid det tar, vilka resurser som krävs, hur stor sannolikheten är m.m. Vid jämförelser mellan olika system vill man veta om system A är säkrare än system B. Men som Edward Schneider påpekar är säkerhet flerdimensionell, så system A kan vara säkrare än system B i en dimension samtidigt som den är svagare i en annan dimension [SCHN 01]. A kanske kan motstå vissa attacker som B inte klarar samtidigt som B kan motstå attacker som A inte klarar av.

Enligt Schneider vore det bästa om vi kunde mäta hur bra säkerhets-mekanismerna i ett system skyddar systemet från förväntade hot mot säkerhetspolicyn. En säkerhets-policy är ett antal regler som talar om vad som är tillåtet respektive förbjudet att göra i ett system. Två möjligheter skulle kunna vara att titta på hur bra en mekanism är som skydd och den ansträngning det krävs av motståndare för att ta sig förbi mekanismen.

Olika säkerhetsmekanismer inom en klass jämförs ofta relativt med varandra. T.ex. när det gäller autentisering anses biometriska mekanismer vara starkare än smartcard som i sin tur anses vara starkare än lösenord. Likaså anses en 128 bitars krypteringsnyckel vara starkare än en 56 bitars nyckel. Att pröva alla nycklarna av 128 bitars längd tar 2^{72} ggr längre tid än att pröva alla nycklarna av 56 bitars längd. Vi kan däremot inte säga att ett system som använder sig av 128 bitars nyckel är 2^{72} gånger säkrare än ett system som använder sig av 56 bitars nyckel. Det finns ju andra och bättre metoder för att knäcka nycklar än att använda sig av uttömmande sökning [STI 95]. Om en motståndare inte kan knäcka en 56 bitars nyckel så ökar man inte säkerheten genom att förlänga nyckeln. Jämför med en höjdhoppare som inte klarar att hoppa över 3 meter, det blir inte svårare för honom bara för att man höjer ribban till 5 meter. Det vi vill ha sagt är att det inte är lätt att rangordna olika säkerhetsmekanismer även om de är inom samma klass.

Det är ännu svårare att rangordna säkerhetsmekanismer mellan olika klasser. Det traditionella förhållningssättet att mäta säkerhet har varit att kräva att vissa säkerhetsmekanismer används och sedan argumenterat för hur säkert systemet är. Ett system som använder sig av både brandvägg och IDS kan anses vara säkrare en ett system som bara använder sig av IDS (men det behöver det inte vara). Hur jämför man två system där det ena systemet bara har en brandvägg som skydd och det andra systemet bara har ett IDS som skydd?

När det gäller att mäta datasäkerhet så har det på Chalmers Tekniska Högskola, under Erland Jonsons ledning, forskats om hur man skulle kunna mäta ett systems förmåga att motstå attacker [JON 95]. De skapade en modell av intrångsprocessen där *angriparens ansträngning* var en variabel. Variabeln antogs innefatta angriparens utbildning, skicklighet och erfarenhet, vilka resurser som användes i intrångsprocessen såväl som olika tidsparametrar d.v.s. CPU tid, on-line tid och hur många man-timmar som lades ner under intrångsprocessen. De utförde två experiment där en grupp av personer, under kontrollerade, former fick göra intrångsförsök i ett givet system. Ett resultat av experimenten var att de, med statistiska tester, kunde visa att det sannolikt går att göra en modell av intrångsprocessen som är exponential-fördelad givet vissa villkor.

Vi har valt att endast föra en mer informell diskussion huruvida säkerheten kan öka vid användning av vilseledande system. Vi har då valt att fokusera på om användningen av vilseledande system leder till att antal lyckade intrång minskar per tidsenhet (kalendertid). Skulle det vara så, kan man säga att säkerheten har ökat med användning av vilseledande system.

2.2. Definitioner

När vi skriver om angripare eller inkräktare så avses personer som medvetet utför illegitima handlingar mot datorsystem och därigenom bryter mot säkerhets-policyn som råder i det aktuella systemet (s.k. intrång). Hur säkerhetspolicyn är utformad beror på hur stor vikt som läggs på de olika aspekterna i säkerhets-policyn, se figur 1, t.ex. hög tillgänglighet kan medföra sämre integritet och sekretess.

En angripare kan vara en extern angripare d.v.s. en person (outsider) som inte har tillgång till någon intern kunskap om företagets struktur. En intern angripare är t.ex. en anställd (insider) som missbrukar eller överskrider sina befogenheter [LIN 99]. Säkerhetspersonal fruktar mest interna angripare eftersom de har störst potential att göra intrång och stor skada. Det finns undersökningar gjorda i USA som visar att 70 procent av alla intrång sker från insiders [THO 00]. Ur teknisk synvinkel kan en extern angripare, som har etablerat sig i det lokala nätverket d.v.s. LAN (Local Area Network), ses som en intern angripare.

För att skydda system mot intrång från externa angripare används traditionellt brandväggar. Brandväggar fungerar som en första försvarsvall. Som skydd mot interna angripare eller externa angripare (som har tagit sig igenom brandväggen), används oftast intrångsdetekteringssystem (IDS) men även intrångstoleranta system, antivirusprogram och kryptering av information används.

Vad är då ett intrång? Med ett intrång menas ett försök att kompromettera sekretess, integritet och tillgänglighets aspekter eller att kringgå säkerhets-mekanismer i en dator eller ett nätverk.

Amoroso definierar ett intrång som:

> Ett intrång är en sekvens av relaterade handlingar utförda av en illvillig motståndare som resulterar i en händelse av obehöriga säkerhetshot mot ett data- eller nätverksdomän [AMO 00].

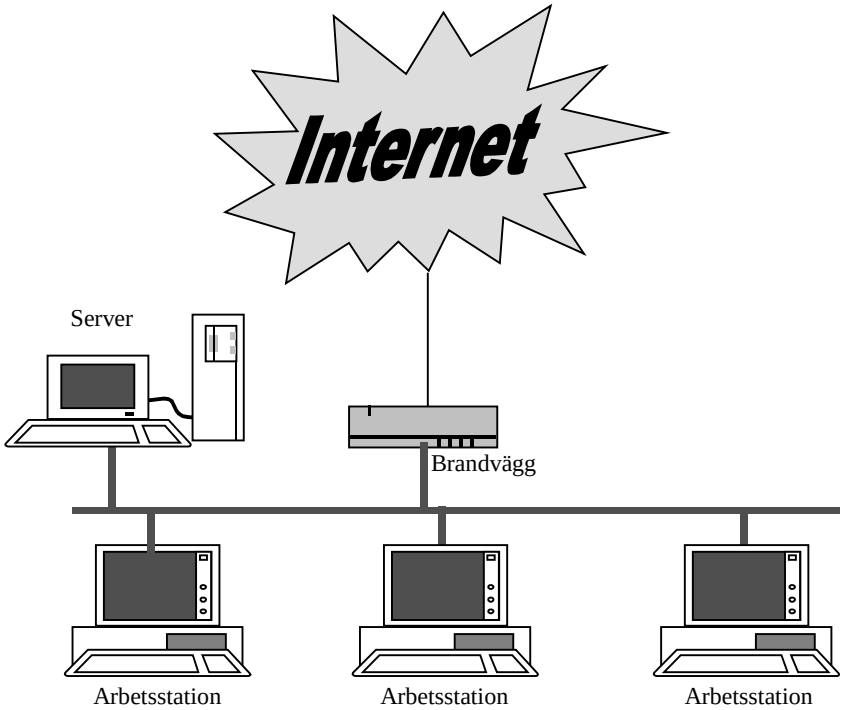
3. Brandväggar

I början av Internets uppbyggnad fungerade omdirigeringen i routrarna mycket dåligt, vilket gjorde att många datapaket hamnade i fel nätverk. För att få bort felaktig trafik i näten installerades brandväggar. I dag installeras en brandvägg mellan ett lokalt nätverk och Internet istället som skydd mot intrång från Internet, se figur 2.

En definition av brandvägg [CHA 95]:

> ”En komponent eller en mängd av komponenter som begränsar tillgången mellan ett skyddat nätverk och Internet eller mellan olika mängder av nätverk.”

Idén med brandväggar är att all trafik mellan LAN och Internet ska passera igenom brandväggen, det finns då möjligheter att kontrollera trafiken och att endast godkänna tillåten trafik. Vad som avses med tillåten trafik beror på hur säkerhetspolicyn är utformad. Säkerhetsnivån på brandväggar skiljer sig åt mellan olika brandväggar, vissa är väldigt restriktiva medan andra släpper igenom en hel del.



Figur 2. En brandvägg som separerar Internet från det lokala nätverket

Brandväggar är ofta konfigurerade så att de inte släpper in paket från Internet till adresser som tyder på att de kommer från datorer på nätverkets insida. Likaså

filtrerar routrar vanligtvis bort paket från interna nätverk ifall avsändaradresserna inte finns i just det nätverket. Detta är ett skydd mot så kallad adressförfalskning[2] vid t.ex. ”denial of service” (DoS) attacker [DAH 01].

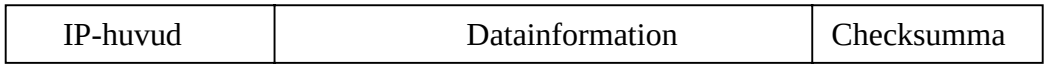
En brandvägg är en effektiv typ av nätverkssäkerhet men den ger inget skydd inne i det lokala nätverket[3], eftersom den endast kontrollerar information som passerar igenom brandväggen. Därför bör det lokala nätverket även kompletteras med ett inre skydd t.ex. IDS, antivirusprogram, osv. Om någon på insidan av brandväggen kopplar upp ett modem internt mot Internet går denna information via modemet och inte genom brandväggen. Detta ger möjligheter för illvilliga personer att sända otillåten information i båda riktningarna, det ger också möjligheter för utomstående att komma in i systemet.

Brandväggar är i huvudsak designade för att skydda mot kända hot. En brandvägg som endast tillåter förbindelser som vi vet är säkra, skyddar till viss del mot okända typer av attacker men om någon skulle upptäcka en bugg i förbindelsen, kan detta utnyttjas för att forcera brandväggen. Det är därför viktigt att en brandvägg uppdateras och säkerhetstestas regelbundet.

De vanligaste sätten att bygga brandväggar på idag, baseras på en kombination av paketfiltrering och proxyservrar.

3.1. Paketfiltrering

Paketfiltreringssystem, kallas även för screening router, vidarebefordrar paket mellan datorer i det lokala nätverket och externa nätverk.



Figur 3. Datapaket

Varje datapaket innehåller: datainformation som ska överföras (Data-information), styrinformation för att paketet ska komma fram till rätt mottagare (IP-huvud) och en checksumma som används för att kontrollera om det har uppstått några fel vid överföringen (Checksumma), se figur 3.

När en paketfiltrerad brandvägg vidarebefordrar ett paket kollas paketets IP-huvud mot säkerhetsinställningarna i brandväggen. Beroende på vilket protokoll som används kan informationen i IP-huvudet bestå av:

- Avsändarens och mottagarens IP-adresser

[2] Författarnas översättning av spoofing – förfalskande av IP-adresser.
[3] Förutsatt att inte nätverket är partitonerat. I så fall kan varje partition ses som ett LAN.

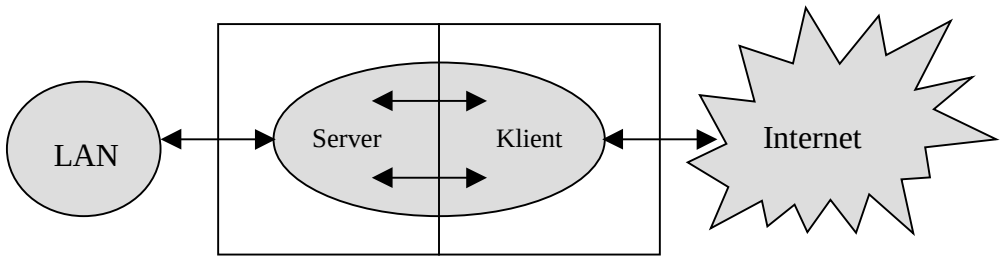
- Avsändarens och mottagarens portnummer
- Protokoll (TCP, UDP, ICMP, FTP, m.m.)
- Paketets längd
- Paketets nummer
- Felkorrigerande information

Det är endast denna information som paketfiltreringssystem undersöker, när de ska avgöra om de ska tillåta ett paket eller inte.

Att en del portnummer används av vissa förutbestämda Internettjänster gör det möjligt att tillåta eller förbjuda tjänsten genom att specificera portnumret, t.ex. port 23 används av Telnet. Vill man inte använda Telnet stänger man av demonen till port 23 [CHA 95]. Paketfiltreringsystem kan endast tillåta eller förbjuda tjänster och de erbjuder heller inte några loggnings möjligheter.

3.2. Proxyserver

Proxyservern är en speciell applikation eller ett serverprogram som körs på en dator med hög säkerhet som har förbindelse till både Internet och det lokala nätverket. En proxyserver har hand om en användares begäran av en tjänst, t.ex. FTP eller Telnet och vidarebefordrar den till Internet. Detta görs endast om begäran uppfyller säkerhetskraven.

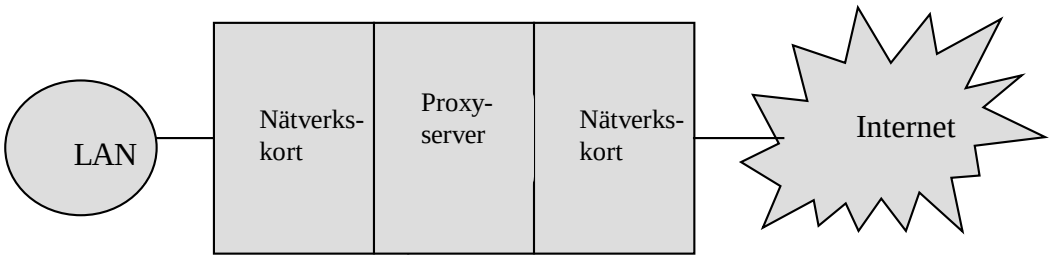


Figur 4. Proxyserver

Proxyservern kan ses som en genomskinlig sluss mellan en användare från insidan i det lokala nätverket och en service på utsidan, se figur 4. Proxyservern har hand om all kommunikation mellan användaren och Internet, vilket gör det möjligt för den att logga och analysera datainformationen som passerar brandväggen, se figur 3. En proxyserver kan även analysera vad användaren vill göra med tjänsten t.ex. förbjuda import men tillåta export av filer med FTP eller att filer endast kan importeras från speciella platser på Internet.

3.3. Brandvägg med dubbla nätverkskort

En brandvägg med två nätverkskort, där varje nätverkskort har en egen IP-adress, kallas för Dual Home Gateway. Brandväggen har ett nätverkskort mot Internet och ett nätverkskort mot det lokala nätverket. För att kunna implementera en brandvägg med dubbla nätverkskort måste routerfunktionen stängas av.

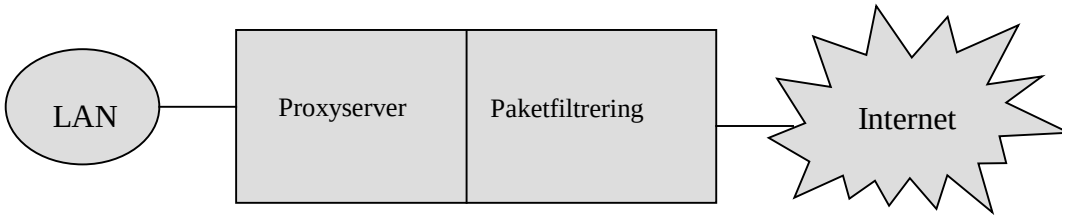


Figur 5. Dual home gateway

Det lokala nätverket kan kommunicera med nätverkskortet som sitter på insidan. Internet kopplas upp mot nätverkskort som sitter på utsidan, se figur 5. För att kommunicera mellan dessa nätverkskort används en proxyserver. De IP-paket som kommer från det ena nätet och ska till det andra nätet dirigeras alltså inte direkt, utan proxyservern vidarebefordrar den trafik som uppfyller säkerhets-policyn. Detta innebär att all direkt trafik är blockerad mellan de olika näten.

3.4. Paketfiltrerande värddator

En paketfiltrerande värddator använder sig av både en proxyserver och paketfiltrering men endast ett nätverkskort. Syftet med att kombinera olika typer av system är att uppnå en högre säkerhet på systemet.



Figur 6. Paketfiltrerande värddator

Från Internet går all trafik in i så kallad paketfiltrering, där trafiken filtreras. Den trafik som uppfyller säkerhetspolicyn får fortsätta till proxyservern. Proxyservern analyserar trafiken och sänder den trafiken som är godkänd vidare in i det lokala nätverket, se figur 6. Det är viktigt att paketfiltreringen och proxyservern befinner sig i serie med varandra. Om det finns en bugg i t.ex. paketfiltreringen finns proxyservern kvar som skydd.

4. Intrångsdetekteringssystem

I det här avsnittet ges en definition på vad vi menar med intrångsdetektering och intrångsdetekteringssystem. Vi kommer också att diskutera varför det behövs intrångsdetektering och vilka tekniker som används för intrångsdetektering.

4.1. Vad är intrångsdetektering?

James P Anderson introducerade 1980 konceptet med intrångsdetektering [AND 80]. Intrångsdetektering är en process som innefattar att övervaka händelser som sker i ett datorsystem eller ett nätverkssystem, analysera händelserna efter tecken på intrång och reagera med motåtgärder mot eventuella intrång. Ett IDS är mjukvara eller hårdvara som automatiserar denna process av övervakning, analys och respons. Några möjliga funktioner som ett IDS kan ha, är att upptäcka intrång, larma, avbryta förbindelse och spåra angripare samt att undersöka vilken skada ett intrång har medfört.

Amoroso definierar intrångsdetektering på följande sätt [AMO 99]:

> ”Intrångsdetektering är processen av identifiering och motåtgärder mot illvilliga aktiviteter riktad mot data och nätverksresurser.”

4.2. Varför ska du använda dig av IDS?

Det är viktigt att ett systems säkerhetsmekanismer är designat för att förhindra obehöriga tillträden till systemets resurser och data. Att helt förhindra intrång i system är i dagsläget ganska orealistiskt, däremot kan man försöka upptäcka dessa intrång och försöka minska de skador som ett intrång kan medföra. Därför bör det för de säkerhetsansvariga inte vara en fråga, om de ska använda sig av IDS eller inte, utan snarare vilken typ och kapacitet IDS ska ha.

IDS finns för att varna oss när den skyddande mekanismen provoceras, kringgås eller penetreras. Några anledningar till att använda sig av IDS är:

- Ett IDS avskräcker dem som kan tänkas bryta mot säkerhetspolicyn, eftersom risken för att bli upptäckt och straffad ökas.

- Ett IDS kan upptäcka problem som inte förhindras av andra säkerhetsmekanismer. IDS kan upptäcka när en angripare har utnyttjat ett säkerhetshål (sårbarhet) som tidigare varit okänt eller inte åtgärdat.

- IDS kan upptäcka då en attack inleds. Vanligtvis inleds en attack med att angriparen försöker undersöka nätverket eller systemet och testa det för kända sårbarheter, s.k. skanning av systemet. Systemet kan då larma säkerhetspersonal och blockera angriparen.

- IDS dokumenterar existensen av hot. Om du vet hur ofta och på vilket sätt attackerna görs, är det lättare för dig att avgöra vilka säkerhetsåtgärder som är nödvändiga. Även om IDS inte lyckas förhindra intrånget, kan det ge relevant, detaljerad och tillförlitlig information som hjälper till att återställa systemet.

4.3. Intrångsdetekteringsmodeller

Det finns flera olika sorters IDS. Dessa skiljer sig ifrån varandra genom att de använder olika tekniker för övervakning och analys. De flesta IDS kan beskrivas i termer av tre grundläggande funktionella komponenter [BAC 01]:

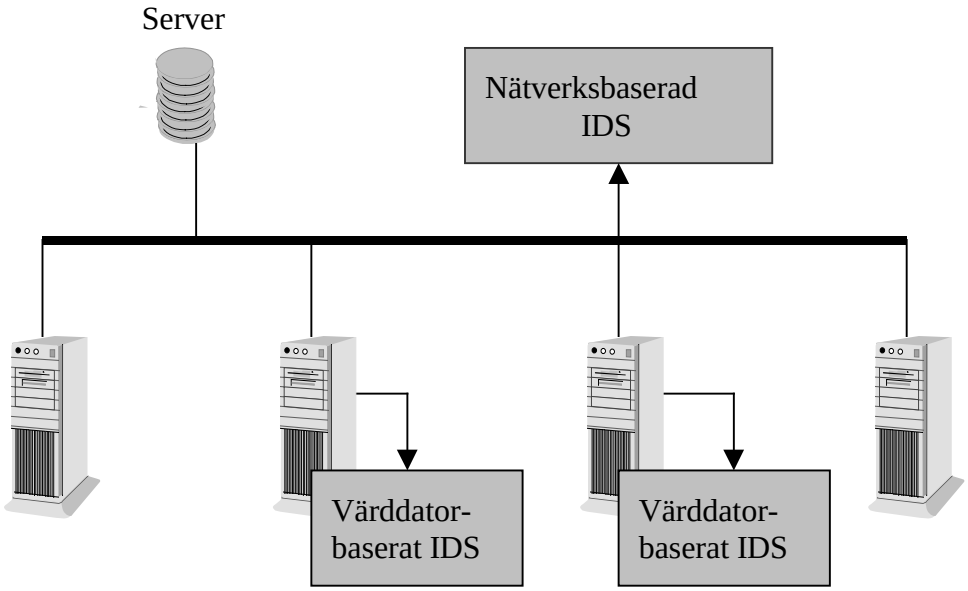
- *Informationskällor* – det är därifrån information om händelser fås. Dessa kan tas från olika nivåer i systemet där nätverks-, värddator- och applikations-övervakning är de mest vanliga källorna.

- *Analys* – den del av intrångsdetekteringen som analyserar händelserna och gör så att de får en innebörd. Händelserna fås från informationskällorna. Analysen avgör om händelserna ska betraktas som intrång eller inte.

- *Respons* – den mängd av åtgärder systemet vidtar när ett intrång upptäcks. Dessa delas ofta upp i aktiva och passiva åtgärder. Aktiva åtgärder innefattar något slag av automatiskt ingripande i systemet. Passiva åtgärder kan vara att rapportera till säkerhetspersonal som sedan förväntas agera utefter vilken information de fått.

4.4. Informationskällor

Det vanligaste sättet att klassificera IDS är att dela in dem i grupper, beroende på vilka informationskällor de använder sig av. Det finns två huvudgrupper av informationskällor: nätverksbaserat och värddatorbaserat, se figur 7. De nätverksbaserade analyserar nätverkspaket som trafikerar LAN och de värddatorbaserade analyserar data som genererats från operativsystem (operativsystemsloggar) eller mjukvaruprogram (applikationsloggar). Det finns även distribuerade IDS som använder sig av både nätverksbaserade och värddatorbaserade informations källor.



Figur 7. Nätverks- och värddatorbaserat IDS

4.4.1. Nätverksbaserat IDS

Dessa system upptäcker intrång genom att analysera paket som trafikerar nätverket. Genom att avlyssna ett nätverkssegment eller en switch, kan en nätverksbaserat IDS övervaka trafiken som berör flera värddatorer (i ett LAN) och därmed skydda dessa.

Nätverksbaserat IDS (NIDS) består ofta av en mängd sensorer eller värddatorer som är placerade på olika platser i nätverket. Dessa enheter övervakar nätverkstrafiken och gör en lokal analys på trafiken och rapporterar attacker till en central ledningskonsol. Många av dessa sensorer körs i ”smygläge” för att göra det svårare för en angripare att avgöra om de är närvarande och var de är lokaliserade.

Fördelar med nätverksbaserat IDS:

- Ett fåtal välplacerade nätverksbaserat IDS kan övervaka ett stort nätverk.
- Nätverksbaserat IDS har liten inverkan på ett existerande nätverk. De är ofta passiva enheter som lyssnar på trafiken utan att störa de normala nätverksoperationerna.
- De kan göras mycket säkra mot attacker och även göras osynliga (för många angripare).
- Nätverksbaserat IDS kostar inte lika mycket att administrera som värddatorbaserat. Det går snabbare att installera och det finns ingen risk att en användare av en stationär dator kommer att ändra inställningarna på IDS.

Nackdelar med nätverksbaserat IDS:

- Är det hög trafik på nätverket kan ett nätverksbaserat IDS få problem att hinna med att analysera alla paket och därmed missa ett intrång. Detta kan delvis lösas med att implementera hela IDS i hårdvara vilket är betydligt snabbare.

- Nätverksbaserat IDS kan inte analysera krypterad information. Detta är ett stort problem eftersom allt fler organisationer (och angripare) använder sig av virtuella privata nätverk (VPN).

- De flesta nätverksbaserade IDS kan inte avgöra om en attack lyckades, de kan endast tala om att en attack påbörjades. Detta innebär att när en nätverks-baserat IDS upptäcker en attack får säkerhetspersonal manuellt undersöka varje värd som angripits, för att se om den penetrerats.

- Vissa nätverksbaserade IDS har problem med nätverksbaserade attacker där fragmentering av paket ingår. Dessa paket med felaktig form kan få IDS att krascha.

4.4.2. Värddatorbaserat IDS

Ett värddatorbaserat IDS (HIDS = Hostbased IDS) placeras på datorn den ska övervaka. En fördel med värddatorbaserat IDS är att den med stor precision och tillförlitlighet kan analysera aktiviteter (behöver inte nödvändigtvis ske i värddatorn) och avgöra exakt vilka processer och användare som är involverade i en attack mot operativsystemet. Till skillnad mot nätverksbaserat IDS kan värddatorbaserat IDS kontrollera vad resultatet av en attack blev eftersom de kan övervaka datafiler och systemprocesser, som ofta är målet för angripare.

Information som värddatorbaserat IDS analyserar kommer vanligen från operativsystemsloggar eller från systemloggar men kan även komma från applikationsloggar. Ett värddatorbaserat IDS, som endast har applikationsloggar som informationskälla, brukar benämnas för **applikationsbaserade IDS** [BAC 01]. De är med andra ord endast en delmängd av värddatorbaserat IDS.

Fördelar med värddatorbaserat IDS:

- Med deras möjlighet att övervaka lokala händelser kan de upptäcka angrepp som nätverksbaserade IDS inte kan.

- De kan operera i en omgivning där nätverkstrafiken är krypterad, förutsatt att IDS har tillgång till datan innan och efter kryptering.

- När en värddatorbaserat IDS opererar på operativsystemsloggar kan de hjälpa till att upptäcka trojanska hästar eller andra attacker som inbegriper integritetsintrång. Dessa visar sig som inkonsistenser i processexekveringen.

Nackdelar med värddatorbaserat IDS:

- De är svårare att administrera eftersom information måste konfigureras och handhas på varje enskild värddator som övervakas.

- Eftersom informationskällorna (systemloggar mm) finns på värddatorn som utsätts för angreppet kan IDS angripas och oskadliggöras.

- De har svårt för att upptäcka attacker som riktar sig mot hela nätverket, avsökning av nätverket och liknande, eftersom de bara ser paket som är ämnade för värddatorn.

- Värddatorbaserat IDS kan göras odugliga med vissa DoS attacker.

- När värddatorbaserat IDS använder sig av operativsystemsloggar som informationskälla, kan mängden av information bli överväldigande vilket medför att det behövs mer lagringsminne.

- Värddatorbaserat IDS förbrukar CPU-tid på den värddator den övervakar, vilket medför att det övervakade systemets prestanda sjunker.

4.4.3. Distribuerad detektering

När informationskällorna är distribuerade eller när analysen av informationen sker på flera olika ställen så kallas det för distribuerad detektering.

Ofta är distribuerad detektering en hybrid av nätverks- och värddatorbaserad detektering. Deras informationskällor är distribuerade men där analysen av datainformationen sker centralt. De använder sig av agenter som samlar in information som sedan skickas till en monitor, där informationen sammanställs och analyseras. Vissa agenter är värddatorbaserade medan andra är nätverks-baserade.

På CERIAS, som håller till på Purdue Univeristy i USA, bedrivs en hel del forskning om intrångsdetektering som bl.a. använder sig av autonoma agenter [SPA 01]. De som är intresserade av intrångsdetektering rekommenderas besöka CERIAS hemsida: www.cerias.purdue.edu.

4.5. Huvudprinciper för IDS

Två huvudprinciper används för att analysera händelser för att detektera intrång: **avvikelsedetektering** och **missbruksdetektering**. Missbrukdetektering är den metod som används mest i de kommersiella systemen. Det finns fördelar och nackdelar med båda metoderna och det visar sig att de bästa av dagens IDS använder sig i huvudsak av missbruksdetektering men med ett inslag av avvikelsedetektering.

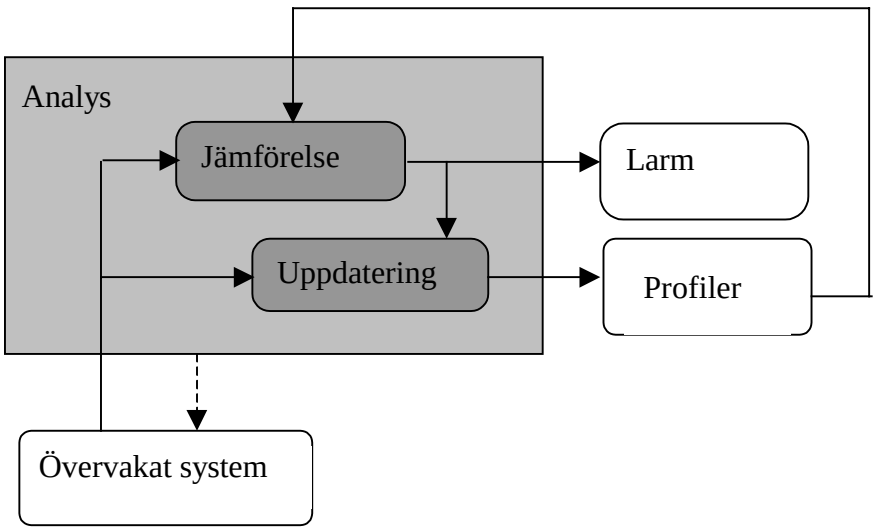
Intrångsdetekteringskomponenter analyserar system och användares operationer i datorsystem och nätverkssystem på jakt efter aktiviteter som betraktas som

oönskade ur ett säkerhetsperspektiv. Finns det för lite data, så kan det medföra att IDS missar intrång och har man för mycket data så drunknar den i information.

4.5.1. Avvikelsedetektering

Avvikelsedetektering går ut på att systemet antar att alla intrångsaktiviteter är avvikelser från hur processer och användare normalt beter sig i systemet. Det innebär att systemet vet eller kan lära sig hur användare eller processer normalt brukar bete sig och slår larm vid för stor avvikelse från tidigare beteende. Det gör också att systemet kan upptäcka tidigare okända intrång och man behöver inte uppdatera det med de allra senaste kända intrångsmetoderna.

En nackdel är att det är svårt att veta på vilken nivå man ska sätta tröskeln för larm, d.v.s. när en avvikelse ska betraktas som intrång. En för lågt satt tröskel kan medföra många *falskalarm*, d.v.s. avvikande aktiviteter flaggas som intrång fastän de inte är det. Ännu allvarligare är det om tröskeln sätts för högt, då kan systemet tolka intrång med små avvikelser som normalt beteende, så kallad *miss*. En avvikelse som inte genererar ett larm uppdateras i systemets profil(er) för normalt beteende, se figur 8. Detta innebär att en inkräktare successivt kan ”lära” systemet att acceptera stora avvikelser genom att hela tiden ligga på gränsen till att systemet ska larma [SUN 96].



Figur 8. Avvikelsedetektering

Nedan går vi igenom tre olika detekteringsmetoder för avvikelsedetektering: statistisk profilering, förutspå händelser och neurala nätverk.

Statistisk profilering
Statistiska profiler skapas för systemobjekt (d.v.s. användare, filer, bibliotek, processer, etc.) genom att mäta olika attribut vid normal användning (d.v.s. hur mycket CPU-tid som används, antal nätverksuppkopplingar under en viss tid, hur ofta en operation misslyckas, vilken tid på dagen det är, etc.). Medelfrekvensen och standardavvikelsen beräknas för varje typ av normalt beteende. När observerade värden faller utanför gränsvärdena så signaleras att ett intrång har skett. T.ex. kan ett intrång signaleras om en användare som aldrig tidigare har loggat in på nätverket före kl. 8.00 på morgonen skulle logga in sig kl. 7.00.

Den största fördelen med statistiska system är att de kan lära sig användarnas beteende. De kan också upptäcka komplexa attacker t.ex. sådana attacker som pågår under en längre tidsperiod.

Nackdelar med statistisk analys är att de gradvis kan bli tränade av inkräktare så att intrångshändelser till slut betraktas som normala. Sambandet mellan olika händelser kan missas på grund av att de statistiska mätningarna är okänsliga för händelsernas ordning [SUN 96]. Statistiska detektorer har också problem med att ta hand om förändringar av användares beteende i systemet t.ex. om en chef får göra sekreterarens uppgifter bara för att sekreteraren är sjuk.

Förutspå händelser
Den här metoden försöker förutsäga nästa händelse baserat på händelser som redan har inträffat. Vi kan därför ha en regel:

$$E1 - E2 \rightarrow (E3 = 80\ \%, E4 = 15\ \%, E5 = 5\ \%)$$

Detta betyder att om händelserna E1 och E2 har inträffat och där händelse E1 inträffade före E2, så är sannolikheten 80 % att nästa händelse är E3, 15 % sannolikhet att nästa händelse är E4 och 5 % sannolikhet att händelsen är E5. Ifall det saknas regler som beskriver intrångsscenarion, så kommer dessa intrång inte att upptäckas. Det kan man delvis lösa genom att flagga alla okända händelser som intrång (ökar sannolikheten för falsklarm) eller att flagga dem som *ej intrång* (ökar sannolikheten för missar). Vanligtvis flaggas en händelse som intrång om vänstersidan av regeln är sann men höger sidan är statistiskt sätt mycket avvikande från förutsägelsen. De finns flera fördelar med denna metod [SUN 96]:

1. Regelbaserade sekvensmönster kan upptäcka avvikande aktiviteter som är svåra att upptäcka med traditionella metoder.

2. System som bygger på denna modell är mycket lämpade för förändringar.

3. Det är lättare att upptäcka användare som försöker träna systemet under sin inlärningsperiod.

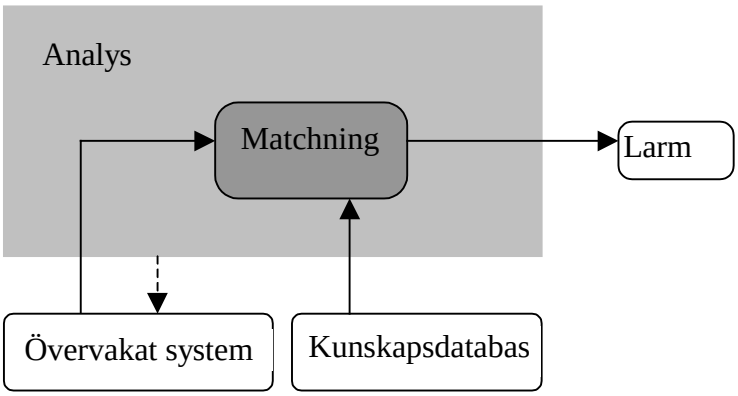
4. Med denna modell arbetar IDS nära realtid. Avvikande aktiviteter kan upptäckas och rapporteras inom några få sekunder efter att IDS fått operativsystemsloggarna.

Neurala nätverk
Idén med neurala nätverk är att träna det neurala nätverket att förutspå en användares nästa handling eller kommando, givet ett fönster av ’n’ stycken tidigare handlingar eller kommandon. Nätverket tränas på en representativ mängd av användare kommandon. Efter inlärningsperioden (träningsperioden) försöker nätverket att matcha aktuella kommandon med profilen från den aktuella användaren som redan finns i nätet. Några fördelar med neurala nätverk är: De klarar brusig data bra, de lyckas oberorende av statistiska antaganden om underliggande data och de är lättare att modifiera för nya användargrupper. Det finns också problem, ett för litet fönster kommer att resultera i falsklarm medan ett för stort fönster kommer att resultera i irrelevant data samt ökad sannolikhet för att ett intrång missas. Nättopologin bestäms bara efter ett stort antal försök och fel [SUN 96].

4.5.2. Missbruksdetektering

Idén, som missbruksdetektering bygger på, är att det går att representera otillåtet beteende (attacker) eller i vissa system tillåtet beteende, i form av signaturer (mönster). En missbruksdetektor letar efter händelser eller en mängd av händelser, som matchar de fördefinierade signaturerna och som finns lagrade i en kunskapsdatabas, se figur 9. Eftersom missbruksdetektorn letar efter signaturer kallas missbruks-detektering också för *signaturbaserad detektering*.

Den första och vanligaste principen är att missbruksdetektorn letar efter signaturer på kända attacker. Systemet vet alltså i förväg vad som är förbjudet och kan därför bara upptäcka kända attackmönster men missar attacker som använder sig av nya attackmetoder (snarlikt så som antivirusprogram fungerar). En missbruksdetektor som använder sig av denna princip sägs vara i förval-tillåt (default-permit) läge, d.v.s. den tillåter allt som inte är specificerat. Det är därför viktigt att man kontinuerligt uppgraderar systemet för nya attackmönster. En annan nackdel är att den kan missa varianter av en attack. En lösning på detta är att använda sig av tillståndsbaserade missbruksdetektorer, något som dock inte används kommersiellt i någon högre utsträckning. Fördelar med den här principen är att den är effektiv (snabb) och att det blir få falsklarm.



Figur 9. Missbruksdetektering

Den andra, mindre vanliga, principen är att endast tillåta sådant som är specificerat, s.k. förval-förbjud (default-deny). I stället för att specificera oönskat beteende, specificerar vi tillåtet beteende och rapporterar allt annat. Fördelen med att använda sig av denna princip är att vi kan upptäcka tidigare okända attacker. Den största nackdelen är att det ofta är mycket svårt att specificera alla tillåtna beteenden korrekt, särskilt för program som redan existerar. Det är stor risk att det blir många falsklarm.

Ett vanligt argument mot missbruksdetektering [LIN 99] är:

> ”Om du känner till en svaghet i ett system, varför kan du då inte rätta till det, istället för att försöka upptäcka när någon utnyttjar svagheten?”.

Svaret är att det inte alltid är så lätt att ta bort en svaghet utan att behöva göra fundamentala ändringar i systemets arkitektur. Ett systems konfiguration är sällan statisk. Nya komponenter installeras, nya användare läggs till osv. Det finns också en risk att ny sårbarhet läggs till när man gör ändringar i systemet.

Nedan går vi igenom några exempel på olika slags system som använder sig av missbruksdetektering.

Expertsystem

Ett expertsystem löser problem som i vanliga fall löses av mänskliga experter. För att kunna göra detta behöver systemet ha tillgång till en stor kunskapsbas (en mängd regler) [STA 00]. Expertsystem bygger sin analys på regler som skapats av experter som t.ex. systemadministratörer och säkerhetsspecialister. En regel består av två delar, en villkorsdel och en konsekvensdel. När villkorsdelen är uppfylld så aktiveras regeln.

Expertsystemen är utrustade med strategier och mekanismer för att bearbeta fakta om tillståndet i en given omgivning och härleda logiska slutsatser ur dessa fakta [LIN 99]. Fakta är ofta implementerade som attribut eller värden som

representerar omgivningens tillstånd i vilket expertsystemet verkar. Om t.ex. ett faktum är att *det ryker* och vi har en regel:

IF *det ryker* THEN *det brinner i närheten.*

Så är konsekvensen att det brinner i närheten. Expertsystemet kan ju även ha ytterligare en regel som är:

IF *det brinner i närheten* THEN *starta sprinkler*

Expertsystemet är alltså en process där utvärdering av fakta leder till en slutsats eller till ny fakta. En sådan process kallas för modus ponens, vilket säger att givet (P=> Q) och P kan vi härleda Q.

Modellbaserad intrångsdetektering

Genom att övervaka aktiviteter så är det möjligt för modellbaserade intrångsdetekteringssystem att hitta intrångsförsök som leder till ett visst intrångsscenario. Modellbaserade intrångssystem består av tre viktiga moduler:

- *Siaren* - använder de aktiva modellerna och scenariomodellerna för att försöka förutse vilket nästa drag kommer att bli i scenariot.
- *Planeraren* - översätter denna hypotes till ett format som visar hur beteendet skulle se ut i operativsystemsloggarna.
- *Tolken* - söker sedan efter den här datan i operativsystems-loggarna.

Systemet fortsätter att samla på sig mer och mer bevis mot ett intrångsförsök tills en tröskel passeras då det signalerar för intrångsförsök.

Eftersom planeraren och tolken vet vad de letar efter i varje steg så kan den stora mängden av brusdata i operativsystemsloggarna filtreras bort vilket leder till en utmärkt prestandaförbättring. Baserat på intrångsmodellen kan systemet förutse en inkräktares nästa drag. Dessa förutsägelser kan användas för att verifiera intrångshypoteser, för att kunna göra förebyggande åtgärder eller för att bestämma vilken data den ska leta efter i nästa steg. Det finns dock några nackdelar:

1. Intrångsscenariers mönster måste vara lätta att känna igen.
2. Det måste alltid finnas mönster i det beteendet som söks efter.
3. Mönstren får inte associeras med några normala beteenden.

4.6. Respons

När en IDS har fått information om händelser som skett och analysen av dessa visar symptom på ett intrång, ska IDS ge någon typ av respons. Det kan vara att rapportera resultat till en förutbestämd plats (passiv respons) eller göra ett automatiskt ingripande i systemet (aktiv respons).

4.6.1. Passiv respons

IDS med passiv respons förser användare av systemet med information och de förlitar sig på att användarna ska agera utifrån den information de fått. Många kommersiella IDS använder sig endast av passiv respons. Passiv respons kan vara larm eller meddelanden som IDS genererar när den har upptäckt ett intrång. Den vanligaste formen av alarm är varningar som skrivs ut på ett terminalfönster eller att det kommer upp ett fönster med ett varningsmeddelande på skärmen, ett så kallat popup-fönster. Vissa IDS skickar varningar eller larm via e-post. Detta är dock inget att rekommendera, eftersom angripare ofta rutinmässigt övervakar e-post, de kan till och med blockera meddelandet.

4.6.2. Aktiv respons

Aktiv IDS respons är en automatisk aktivitet som görs när en viss typ av intrång har upptäckts. Det finns tre kategorier av aktiv respons:

Samla mer information
Det kan innefatta att öka operativsystemets loggning av händelser eller att titta på alla paket i ett nätverk och inte bara på dem som ska till en viss port eller system. Att samla mer information hjälper till att avgöra om det verkligen har skett ett intrång eller inte och kan även understödja ett eventuellt åtal mot angriparen.

Ändra omgivningen

- Konfigurera om routrar och brandväggar för att förbjuda trafik från de IP-adresser som angreppet kommer ifrån.
- Avbryta sessionen genom att inflika ett TCP resetpaket i angriparens förbindelse till det drabbade systemet.
- I extrema fall, konfigurera om routrar och brandväggar för att kapa alla förbindelser som använder sig av ett visst nätverksinterface.

Slå tillbaka mot angriparen
Att attackera angriparen eller att aktivt försöka få information om angriparens värddator eller webbplats är en typ av aktiv respons. Detta är något som bör undvikas och definitivt inte ska ske automatiskt. Det är mycket troligt att det är olagligt och eftersom många angripare använder sig av falska nätverksadresser när de angriper system, så är risken också stor att någon oskyldig drabbas. Att slå

tillbaka kan dessutom medföra att angriparen blir provocerad och därför ökar sina ansträngningar och attacken eskalerar.

4.7. Var ska IDS placeras?

En fråga som kommer upp när man ska använda sig av IDS är var de bör placeras. Att installera IDS på alla värddatorer i företaget kan bli mycket tidskrävande, eftersom varje IDS måste installeras och konfigureras för varje specifik värddator. Därför ska de i första hand placeras på kritiska enheter som t.ex. servrar. De nätverksbaserade IDS sensorerna kan placeras på flera olika platser i nätverket, se figur 10.

Plats 1 (figur 10)
IDS utplacerad bakom extern brandvägg i nätverkets s.k. demilitariserade zon (DMZ).

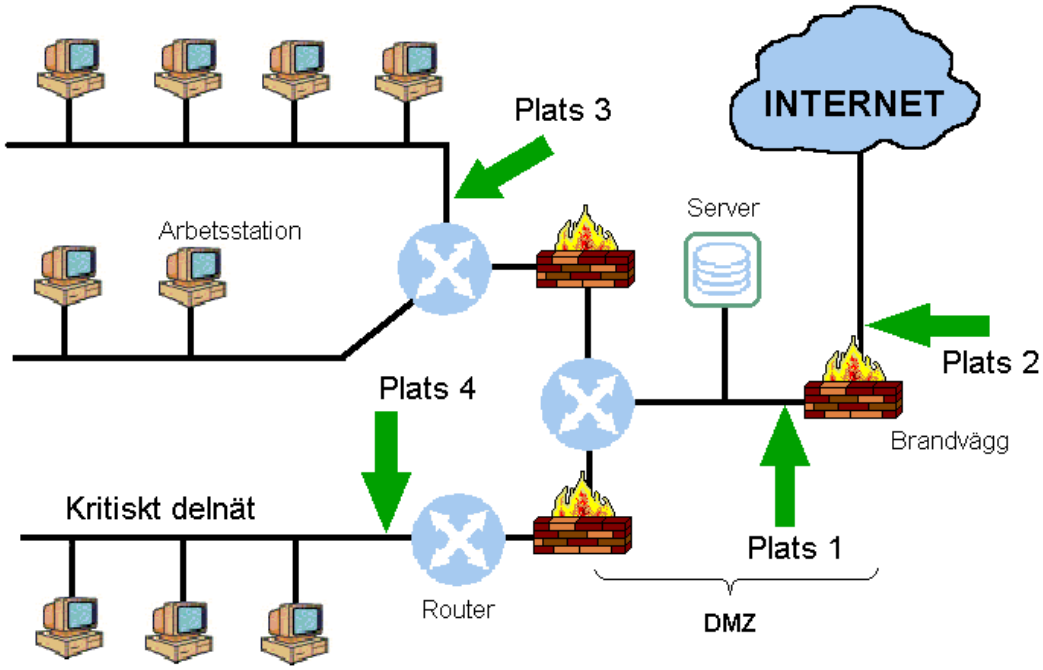
- Ser attacker som har sitt ursprung från utsidan och som penetrerar nätverkets yttre skydd (perimeter defense).
- Uppmärksammar problem med brandväggens policy eller utförande.
- Upptäcker attacker mot webb- eller ftp-server, som vanligtvis brukar finnas i den DMZ.
- Även om en attack (utifrån) inte upptäcks kan IDS ibland känna igen den utgående trafiken från den drabbade servern och kan på så sätt upptäcka ett intrång om trafiken plötsligt är avvikande från det normala.

Plats 2 (figur 10)
IDS utanför den externa brandväggen.

- Dokumenterar antalet attacker mot nätverket, som har sitt ursprung från Internet.
- Dokumenterar vilken typ av attack det är mot nätverket.

Plats 3 (figur 10)
IDS i ett större delnät:

- Övervakar en stormängd av nätverkstrafik vilket medför att möjligheten för att upptäcka attacker ökar
- Upptäcker betrodda användare som överskrider sina befogenheter.



Figur 10. Placering av IDS i ett nätverk

Plats 4 (figur 10)
IDS i kritiska delnät.

- Upptäcker attacker som är riktade mot kritiska system och resurser
- Tillåter att övervakningen fokuseras på begränsade resurser som anses vara av stort värde.

4.8. Önskvärda egenskaper för IDS

Realistiska förväntningar på IDS:

- De ska upptäcka kända attacker inom rimlig tid.
- Genom att föra bra loggar som kan analyseras ska IDS hjälpa dig att förstå vad som verkligen sker i ditt nätverk. Med denna kunskap som bakgrund, ska du sedan kunna besluta vilka olika typer av säkerhetsmekanismer du behöver i ditt nätverk, utreda vad som har hänt vid intrånget och när intrånget skedde samt ge bevis som håller vid en eventuell rättegång.
- De ska kunna följa försök att överbelasta nödvändiga resurser.
- Den ska vara pålitlig, tyst (osynlig) när den övervakar system och vid alarm ska den inte sända i klartext.
- Larma eller aktivera en förutbestämd åtgärd.
- De ska vara svårhackade.

- IDS ska påverka det övriga systemet så lite som möjligt, d.v.s. så lite overhead som möjligt.

- Det ska kunna anpassas för ändringar i systemet och ändras med tiden

- Det ska kunna omkonfigureras dynamiskt, så att man inte behöver starta om systemet. Det ska fungera även om någon komponent är ur funktion.

- Vid uppstart ska IDS kunna återställa sig till det tillstånd den hade tidigare och kunna fortsätta opåverkat.

Orealistiska förväntningar på IDS:

- Upptäcka hittills okända attacker.

- Att de ska ha förmågan att kunna reagera mot attacker och på något förebyggande sätt skydda systemet. Detta är farlig mark eftersom det betyder att falsklarm kan trigga reaktioner som stänger ner det egna nätverket, tjänster eller kanske stänger ner ett nätverk som har blivit ditsatt.

- Att de automatiskt ska skydda mot all slags hot.

- Att de ska upptäcka attacker som är utspridda under en lång period.

- Att de ska vara helautomatiska. Det kommer inom den närmsta framtiden krävas hjälp av en professionell som känner till omgivningen och systemets natur för att kunna avgöra om larm är falsklarm (misstag, buggar, harmlös nyfikenhet) och vilka som är verkliga attacker.

- Att de ska ersätta andra säkerhetsmekanismer.

5. Olika typer av attacker

De flesta attacker är riktade mot system som inte är uppdaterade vilket gör att det finns kända säkerhetshål på dessa system. Attackerna kan delas in i olika grupper: *skanningattacker, denial of service attacker* och *penetrationsattacker*. Dessa attacker kan startas lokalt från den attackerade datorn eller på distans med hjälp av ett nätverk.

5.1. Skanningattack

Skanningattacker är en av de vanligaste attackerna som dessutom är laglig. Vid en skanningattack söker inkräktaren igenom nätverket eller systemet genom att sända olika typer av paket. Genom att använda sig av svaren på dessa paket, kan inkräktaren lära sig mycket om karakteristik och sårbarhet. En skanningattack penetrerar eller skadar inte systemet, men de är en förberedande åtgärd innan en attack sätts in mot systemet. Vid en skanningattack kan följande information fås om systemet.

- Nätverkets topologi.
- Vilken typ av nätverkstrafik som tillåts igenom brandväggen.
- Vilka värddatorer som är aktiva i nätverket.
- Vilka operativsystem dessa värddatorer använder sig av.
- Vilken mjukvara servrarna använder sig av.
- Vilket versionsnummer det är på den mjukvara som hittas på respektive server.

Sårbarhetsskanning är en speciell typ av skanning, den letar efter vissa kända säkerhetshål på en värddator. Resultat av attacken är en lista med värddatorer (IP-adresser) som är känsliga för en specifik attack [BAC 01]. Med den här informationen kan alltså en knäckare identifiera vilka system som kan penetreras med en specifik attack.

5.2. Denial of Service (DoS)

DoS attacker används för att slöa ner eller krascha nätverk, system eller tjänster. DoS attacker är bland de svåraste attackerna att skydda sig mot, de är tyvärr också bland de vanligaste. Det finns två huvud typer av DoS attacker:

- DoS attack som utnyttjar fel i systemets mjukvara, för att få processorfel eller att förbruka systemets resurser. Ett exempel på en sådan här attack är ”ping of death”. I denna attack sänds ett antal för stora pingpaket till vissa Windows-

system, dessa Windowssystem kan inte hantera för stora pingpaket och kommer då att krascha.

- DoS attack som överbelastar systemet med information, skickar helt enkelt mer information än vad det mottagande systemet klarar av att hantera. I de fall där hackaren inte kan sända tillräckligt mycket information för att överbelasta systemet, kan hackaren manipulera med nätverksuppkopplingen till systemet och på detta vis stoppa åtkomsten till systemet. Detta kan t.ex. göras genom att sända ett ICMP-meddelande direkt till den värddator som ska stängas ner, med innehållet att sluta sända IP-datagram över hela eller delar av nätverket [HAR 99].

Distribuerad DoS (DDoS) attack är en delmängd till DoS attacker. DDoS attacker innebär överbelastning av systemet där knäckaren använder sig av många datorer i attacken. Knäckaren kan t.ex. broadcasta[4] ett ping med förfalskad avsändaradress så att ett stort antal maskiner svarar till offret som har den förfalskade adressen, en sk SMURF attack.

5.3. Penetrationsattacker

Penetrationsattacker involverar obehörigt förvärv eller ändring av systemets privilegier, resurser eller data. En penetrationsattack kan ge kontroll över systemet, genom att utnyttja mjukvarufel i systemet. Penetrationsattacker är oftast de man förknippar med intrång. Det finns många olika typer av penetrationsattacker, de vanligaste är att [BAC 01]:

- Användare skaffar sig rooträttighet - en lokal användare på en värddator får en total kontroll över värddatorn.

- Inkräktare skaffar användarkonto - en inkräktare på nätverket får kontroll över ett användarkonto på värddatorn.

- Inkräktare skaffar sig rooträttighet - en inkräktare på nätverket får kontroll över värddatorn.

- Inkräktare får läsrättigheter - en inkräktare på nätverket skaffar sig möjlighet att läsa datafiler utan ägarens tillåtelse.

- Inkräktare får skrivrättigheter – en inkräktare på nätverket skaffar sig möjlighet att skriva till datafiler utan ägarens tillstånd.

[4] Nätverkstrafik som skickas från en nod till alla andra noder i nätverket.

6. Vilseledande system

”In war, truth is so precious, it must
be protected by a bodyguard of lies”
Winston Churchill

Vilseledande system kan uppehålla och avleda en angripares ansträngningar så att angriparen fokuserar på fel system. Detta tar bort belastningen från de värdefulla datorerna och ger säkerhetspersonalen mer tid att reagera med motåtgärder, kartlägga angreppet och samla information om angriparen.

6.1. Tekniker för att vilseleda

Rent tekniskt är det inte svårt att vilseleda en inkräktare i ett datorsystem men om inkräktaren är insatt i systemets uppbyggnad är det stor risk att inkräktaren kommer underfund med att hon/han har blivit vilseledd. Eftersom ett vilseledande system ska fungera mot olika typer av attacker blir systemet komplext. För att ge läsaren en bild av varför det blir komplext ska vi gå igenom några olika tekniker för att vilseleda en inkräktare. Dessa tekniker har vi delat in i två olika grupper visa respektive dölja information. Mycket av teorin nedan är hämtat från Fred Cohen’s hemsida [COH 98].

6.1.1. Visa information

Ökad synlighet

Att använda sig av ökade synlighetsattribut i en dator som t.ex. indikerar tillgång till webben eller åtkomst till ett nät kan vara ett vilseledande system på enklaste nivån. Handlingen att bara besöka en dator med hög synlighet som det tillsynes inte finns någon annan anledning till än ren nyfikenhet, kan ge viktig information om besökaren (d.v.s. vem är besökaren, vad är meningen med besöket, vilken ansträngning läggs ner, när sker denna aktivitet o.s.v.). Kunniga undersökare kan utifrån sådan information upptäcka identiteter och dra slutsatser om möjliga motiv och metoder av obehörig aktivitet.

Visa falsk styrka

Att visa falsk styrka, precis som katter gör när de visar ragg, går ut på att få angriparen att tro att systemet är bättre skyddat än vad det i verkligheten är. Genom att slumpmässigt slänga upp en ruta med följande information, *vi har upptäckt ett intrång och försöker nu att spåra dig, vi kommer även att vidta rättsliga åtgärder mot dig,* kommer en inkräktare att tänka sig för innan denne fortsätter med attacken medan de anställda vet om att detta meddelandet kommer slumpmässigt och kan därför ignorera det.

Exponera säkerhet

Genom att visa att systemet använder sig av kryptering, t.ex. RSA, DES, eller engångslösenord, kan man visa för inkräktaren att det inte är någon mening att fortsätta med intrånget.

Avskräckning

Genom att visa att du inte tänker acceptera ett intrång och att du till varje pris kommer att försöka få fast förövaren kan attacker undvikas. Avskräckning byggs upp under ett längre tidsperspektiv genom att alltid anmäla varje intrång till polisen och dra förövaren inför rätta om det är möjligt.

Lockbete

Metoden att använda sig av lockbeten går ut på att avleda en angripare från de viktiga systemen och locka angriparen till ett system med en begränsad och säker omgivning. I denna omgivning uppehåller systemet angriparen och det finns möjlighet till övervakning av angriparens aktiviteter. För att göra lockbetets omgivning trovärdig så innehåller systemen information, som kan tyckas vara värdefull för en angripare men som i själva verket bara är fabricerad. Lockbetet kan vara en fysisk kopia av ett system men där all viktig och känslig information saknas.

Ljuga

Avsikten med att ljuga är att övertyga inkräktaren att han/hon aldrig kommer att veta om den information som ges är korrekt eller inte. Att övertyga en angripare, som inte utvärderar det hon/han ser, fungerar bra. För att ljuga övertygande för en inkräktare som gör en noggrannare undersökning, krävs det att lögnen kompletteras med andra lögner.

Genom att använda sig av list, knep och lögner går det att lura inkräktare till att tro att det angripna systemet är oskyddat dvs att det är dålig kontroll av systemet och att det finns liten risk för upptäckt.

Falsk och planterad information

Idén med falsk och planterad information bygger på att inkräktaren måste ha en viss mängd av information innan attacken mot systemet inleds. Genom att ge ut felaktig information kommer inkräktaren att göra sitt intrångsförsök med felaktig information. Vilket gör det svårare att lyckas med intrångsförsöket och systemansvariga får lättare att upptäcka intrånget.

För att lyckas med falsk och planterad information finns det några saker som de ansvariga måste tänka igenom mycket noga. Dessa är: vilken påhittad information som passar i de olika situationer som uppstår, att informationen stämmer under en längre tid och att det medför en risk för interna missförstånd. Information som inte är tillräckligt genomarbetat fungerar mot vissa enklare attacker men för större organisationer och mot mer sofistikerade inkräktare behövs väl genomarbetat material.

6.1.2. Dölja information

En i mängden

Precis som att en enskild gnu kan försvinna i en flock och på så sätt vara mer skyddad så går den här metoden ut på att gömma en maskin bland tillsynes flera identiska maskiner, man använder sig av en så kallad flermålsfilosofi. Om en bankrånare står inför tre likadana kassaskåp varav två är fällor, vilket kassaskåp ska bankrånaren välja?

Kamouflering

Kamouflering görs genom att skapa en konstgjord omgivning där en tillämpning ser ut att vara en annan. Detta görs för att det ska bli svårare att veta vilken tillämpning som används. Det går t.ex. att kamouflera ett UNIX system genom att använda sig av en DOS promt. Ett annat exempel är att använda sig av ett användargränssnitt som liknar en viss sorts e-post program men som egentligen är ett annat e-post program. På detta sätt kan man att lura inkräktaren till att använda attacker som baseras på buggar i vissa program.

Gömma information

Genom att visa en omgivning som en inkräktare förväntar sig finns det möjlighet att gömma vissa funktioner i omgivningen eftersom inkräktaren ”känner igen sig”. Detta kan t.ex. göras på följande sätt: i Word under menyn File är sista valet Exit. Under Exit och ”utanför menyn” kan det finnas ett meny val som inte visas men är klickbart. När vi gömmer något i en synlig bild finns det möjligheter att knäcka detta genom uttömmande testning, eftersom vi har en ändlig mängd. Det är viktigt att denna mängd görs tillräckligt stor så att det finns tillräckligt många olika alternativ att testa sig igenom. Detta är samma typ av problem som uppstår med lösenord.

Filer som innehåller viktig systeminformation eller information om användare bör förvaras i filer och bibliotek med namn som inte kan förknippas med innehållet, t.ex. löneinformation bör inte sparas i fil med namnet löner. Detta för att minska intresset för undersökning av dessa filer av en inkräktare. Lösenordsfiler ska vara krypterade och åtkomstskyddade.

Osynlighet

Denna metod går ut på att det finns maskiner som inte är synliga i nätverket men som inte desto mindre är mottagliga för data i det lokala nätverket. Deras huvudsyfte är att avslöja brott eller felaktig användning av systemet men de kan även användas för att göra psykologiska profiler på användare och inkräktare. Eftersom maskinen är osynlig, är all data som kommer till maskinen antingen felaktig nätverksdata eller data från aktiviteter som sonderar nätverket. Sondering av ett nätverk är, nästan utan undantag, alltid en avsiktlig handling. Du kan jämföra det med en tjuv som går på en parkeringsplats och försöker öppna alla bildörrar i avseende att stjäla en bil eller något inne i bilen.

Rörliga mål

Efter en tid blir det oundvikligt att användare kommer att upptäcka datorer som används för nätverks- och datasäkerhet. När en användare har upptäckt en sådan dator kommer den datorns effektivitet som säkerhetsverktyg att minska väsentligt. En ”rörlig maskin” kan rekonfigureras eller i vissa fall rekonfigurera sig själv för att undvika att bli identifierad. Rekonfigurering görs huvudsaklingen via två element:

- Skifta nätverksadresser, d.v.s. att logiskt skifta adresserna inom nätverket.
- Skifta tjänster, med vilket man menar att man ändrar de tjänster som är aktiva och därmed det som är synligt för användarna inom nätverket.

Den viktigaste aspekten i sådana här vilseledande system är den psykologiska inverkan det får på angripare. Angripare kommer kanske aldrig att få veta vilka datorer och tjänster som de når, vilka som är riktiga och vilka som är falska [SCG 00]. ”Rörliga maskiner” är också värdefulla genom att de representerar ett proaktivt snarare än reaktivt synsätt. De ger en slags rökridå oavsett om någon faktiskt initierar en attack.

En före detta avhoppad KGB agent har utvecklat säkerhetslösningen *Variable Cyber Coordinates-system* där systemet växlar IP-adresser oftare än en gång per sekund vilket ska göra dem osynliga för alla utom legitima användare [LÖV 01].

6.2. Vad är ett vilseledande system?

När vi nu gått igenom vilka tekniker det finns för att vilseleda är det dags för att ge vår definition på vad ett vilseledande system är.

Definition av vilseledande system

Ett *vilseledande system* är ett system som använder sig av en eller flera komponenter vars syfte är att förvilla eller vilseleda en angripare.

Vilseledande system har använts inom datasäkerhet och då främst i forskningssyfte sedan slutet på 80-talet. *Honeypots* och *ironboxs* användes (och används) för att se hur knäckare gick tillväga när de försökte hacka sig in i system, ta reda på vilka mål de hade och vilka verktyg de utnyttjade.

6.2.1. Honeypot

Maskiner som används som lockbete brukar allmänt kallas för honeypots men även andra uttryck förekommer. Barry Schlossberg, säkerhetsansvarig på sNET,

använder sig t.ex. av uttrycket *varuautomat* (Vending machine) för en värddator som används som lockbete [SCG 00].

Jeff Forristal, säkerhetskonsult på Neohapsis i Chicago [FOR 00], definierar en honeypot på följande sätt:

Definition på Honeypot
”En honeypots uppgift är att efterlikna systemets servrar medan man larmar och loggar inkräktares aktiviteter.”

Honeypots är ett komplement till traditionella IDS och brandväggar. De innehåller känsliga monitorer och händelseloggar, som upptäcker accesser och samlar information om angriparens aktiviteter.

Det finns två typer av honeypots: hårdvarubaserade och mjukvarubaserade. Hårdvarubaserade honeypots är servrar, omkopplare (switchar) och routrar. De har gjorts attraktiva med vanliga kända felkonfigurationer. Dock har några hårfina ändringar gjorts i operativsystemet så att knäckaren inte ska kunna ta över honeypoten.

De mjukvarubaserade honeypots är program som efterliknar riktiga servrar. Eftersom en inkräktare bara leker med en mjukvaruattrapp är det ingen risk att inkräktaren får kontroll över den riktiga hårdvaran. Även om knäckaren listar ut att det är en mjukvaruattrapp, bör ”buren” som programmet körs i, vara så säkert eller isolerat att det inte går att knäcka ”buren”. Att skapa en mjukvarubaserad honeypot som kan lura en sofistikerad knäckare är dock mycket svårt.

I princip kan vilket system som helst ”göras om” till en honeypot. En honeypot kan vara en maskin installerad med ett äldre och ej uppdaterat operativsystem med ett eller flera kända säkerhetshål. En standard IDS som är nätverksbaserad kan sedan användas för att logga hackningar som är riktade mot maskinen. När maskinen är knäckt så fortsätter IDS att logga knäckarens fortsatta aktiviteter. Något man bör vara medveten om är att angriparen då kan göra attacker ifrån maskinen, där avsändareadressen är förfalskad och som IDS kan missa.

Istället för att använda termen honeypot föredrar Douglas Moran, vice vd för forskning och utveckling på Recourse Technologies, att använda sig av de två termerna *deception server* och *deception host* för de två grundläggande kategorierna av *honeypots*[MOR 00].

Deception servers

En *deception server* efterliknar en värd på nätverkstjänstsnivån, den har hand om samspelet på protokollnivå men tillhandahåller inget riktigt innehåll. En deception server kan användas för att skapa ett helt nätverksutseende med en blandning av olika plattformar.

Deception servers är mycket användbara för identifiering, när en angripare söker av ett nätverk. De är också mycket användbara för att upptäcka detaljer i nya attacker där inledningen är nyckeldraget men eftersom de stoppar åtkomst med en gång, så ger de nästan ingen information av angriparens intentioner.

Deception hosts

En *deception host* är en normal värddator med fabricerat innehåll och med några extra instrument för att kunna övervaka aktiviteter. Angriparen kan bli en lokal användare eller få tillträde till innehållet genom vanliga nätverkstjänster.

Deception hosts ger ett mycket större djup av vilseledning i jämförelse med deception servers. De låter attackeraren avslöja mer av sina trick, allt från hur de försöker få privilegier till metoder för att: förstöra loggfiler, installera bakdörrar, trojanska hästar m.m. Deception hosts kräver närmare övervakning och mer uppmärksamhet på var de placeras i nätverket. En angripares mål kan vara att få tillgång till mer cpu-tid, resurser eller mer bandbredd från nätverket.

6.2.1.1. Fördelar med honeypot

En ärlig användare ska normalt aldrig tillträda en honeypot och därför, när man upptäcker att någon har accessat en honeypot, så är det med stor sannolikhet en angripare [ITL 99]. Detta gör att de är särskilt bra på att upptäcka okända attacker som har slunkit förbi andra system. Den tid en angripare lägger på en honeypot är tid som administratören kan lägga på att spåra angriparen och förbättra försvaret. En honeypot kan upptäcka attacker både från interna och externa angripare. Redan vid första accessen till en honeypot kan larmet gå, medan IDS har problem att avgöra när ett intrång har skett. IDS behöver mer information än honeypot för att besluta om det är ett intrång eller inte.

Genom att studera inkräktaren i en honeypot kan man lära sig mera om olika intrångsmetoder och kan på så sätt förbättra hotmodeller och systemskydd. Det är även enklare att övervaka en angripare inne i en honeypot. Det gör inget om inkräktaren förstör inställningarna i honeypoten eftersom den inte påverkar funktionaliteten på det övriga systemet. Inställningarna i en honeypot bör vara utformade på ett sådant sätt att det är enkelt att göra en ny inladdning av programvara.

Honeypot presenteras ofta som lätthackade. En av de vanligaste saker knäckarna gör är att skanna Internet efter tjänster som har kända fel i sig. Honeypot kan konfigureras upp till att liknande system som med lätthet kan knäckas. Dessa system larmar om det är någon som försöker knäcka dem, t.ex. kan en POP3 tjänst rapportera vilken version av mjukvara som används. Knäckarna kopplar upp sig till port 110 och får versionsinformation, om POP3-tjänsten, från honeypotdemonen. Efter detta kollar angriparna vilket script som ska användas för att knäcka den specificerade mjukvaran.

6.2.1.2. Nackdelar med honeypot

Det finns även några nackdelar med att använda sig av honeypots. Om en honeypot blir knäckt så kan den användas som en språngbräda för fortsatta attacker. Med honeypots ökar komplexiteten i systemet och honeypots måste underhållas precis som all annan nätverksutrustning. Hur ska man annars få veta om loggningen är korrekt eller att en honeypot inte blivit knäckt?

Det är bara en tidsfråga innan en sofistikerad inkräktare kommer att skaffa sig rooträttigheter i en honeypot. Därför är det viktigt att logga och kontrollera alla aktiviteter som sker i honeypoten. All trafik som sänds från honeypoten till Internet är med största sannolikhet information som sänds av en inkräktare. En av de första åtgärder en inkräktare vidtar, för att dölja intrånget, är att byta ut loggfilen, därför är det viktigt att placera en loggfil under en katalog där inkräktaren inte letar efter loggfiler t.ex. /var/tmp/.conf eller på en annan dator [HON2 00].

För att använda sig av honeypotsystem krävs det att den systemansvarige är mycket kunnig, så att han kan utreda hur intrånget har skett. Systemansvariga måste vara tillräckligt kunniga för att kunna dra korrekta slutsatser. Dessa slutsatser ska sedan användas för att säkra upp det ”riktiga systemet” mot liknande attacker.

Mike Fratto[5] anser att honeypots inte tillför någon fundamental ökning av säkerheten i ett nätverk, snarare är det så att de komplicerar det hela [FRA 00]. Han menar att nu har du ytterligare en eller flera servrar som måste säkras, så att de inte används som språngbräda för vidare attacker. När en angripare har tillgång till ett system så är angriparen ett steg närmare till ditt nätverk. Kom ihåg att det inte är de kända attackerna som lägger ditt nätverk i händerna på en angripare.

Honeypots och liknande teknologier bör därför användas med försiktighet och endast av organisationer med mycket tekniskt kunnig personal, som är villiga att experimentera med spjutspetsteknologier [BAC 01].

6.2.1.3. Vad ska man tänka på när man bygger en honeypot?

En honeypot ska se attraktiv ut och den ska se ut att vara lätt att knäcka men utan att det får verka allt för lätt. I så fall kan knäckaren lätt identifiera den som en honeypot och istället angripa andra servrar i nätverket.

Hårdvarubaserade honeypotar bör vara isolerade och endast köra honeypot mjukvara. Även om en honeypot är kopplad till det interna nätverket så bör den inte kommunicera med någonting annat i nätverket.

Kärnorna bör vara skrivskyddade eller så fort någon försöker ändra i kärnan ska maskinen boota om sig så att ändringarna inte får någon effekt.

[5]Mike Fratto är chefsredaktör på Syracuse University Real-World lab. Hans ansvarsområde täcker security, remote access, VPN, enterprise network management, och UPS's

De vanligaste operativsystemen som knäckare angriper är äldre Linux (som Red Hat 5.0), Solaris 2.6 och Microsoft IIS 4 [HON2 00]. Därför kan det vara en god idé att använda sig av något eller flera av dessa operativsystem som honeypot.

Oskyldiga användare ska inte av misstag kunna tillträda en honeypot och skulle de göra det så får det inte resultera i någon skada. Man bör t.ex. inte konfigurera en honeypot så att användare kan accessa den med sina riktiga användarnamn och lösenord.

För det första så exponerar man då känslig information om användarna till angriparen. I vissa fall kan det dock vara nödvändigt att använda sig av riktiga användarnamn för att honeypoten ska verka trovärdig t.ex. mot insiders.

För det andra, om en användare oavsiktligt får tillgång till information från en honeypot, så kan användaren ta viktiga beslut som baseras på felaktig data.

6.2.2. Ironbox

Maskiner med honeypotegenskaper men med den skillnaden att de inte ska attrahera angripare utan att de istället jobbar i det dolda tillsammans med traditionella IDS kallas för *ironbox* eller *Internet-fälla*. Brighton University har definierat en ironbox enligt:

Definition av Ironbox
En *Ironbox* är en speciell omgivning som satts upp för att uppehålla en knäckare tillräckligt länge för att kunna bli spårad. Den kan innehålla en modifierad shell som förhindrar oförutsägbara steg av knäckaren och lockande filer för att få knäckaren att stanna kvar. [BRI 01]

När ett IDS upptäcker angripare så överför den angriparna, utan att angriparen märker det, till en ironbox. Väl inne i ironboxen hålls angriparna i en simulerad miljö där de inte kan göra någon skada och där vartenda steg som angriparen tar noteras.

Ett annat namn som ibland förekommer i litteraturen för ironbox är padded cell men en padded cell är enligt Brighton University:

Definition av padded cell
En *padded cell* är ett program som begränsar en lokal användare till en väl definierad delmängd av värdsystemets möjligheter (t.ex. ”rsh” på USG Unix). Där placerar man en lokal användare (luser) så att de inte kan förstöra något.

Notera att detta är skillnad från ironbox eftersom en padded cell är offentlig (öppen) och inte avsedd att upprätthålla säkerhet lika mycket som att skydda andra från konsekvenser från en lokal användares naivitet.

6.2.3. Kameleontmaskin

En kameleontmaskin är en ”riktig” maskin som använder sig av flermålsfilosofin för att skydda sig mot intrång [SCG 00]. Genom att det finns ett antal attrapper (kan ses som honeypotar) som liknar kameleontmaskinen är sannolikheten mindre att angriparen går på kameleontmaskinen vid första försöket. När en angripare accessar en attrapp vet man att det är med stor sannolikhet ett intrång och man kan då öka beredskapen mot intrång genom att t.ex. larma säkerhetspersonal samt att öka loggningen på kameleontmaskinen och i nätverket. Precis som med att använda sig av honeypotar, finns det en risk att användare av misstag accessar kameleontmaskinens attrapper.

6.2.4. Honeynet

I vår analysdel använder vi oss av termen *honeynet*, som vi har hämtat från *The Honeynet Project* [HON 01], och har därför valt att ge den en definition.

Definition av honeynet

Ett honeynet är ett nät eller en del av ett nät som använder sig av vilseledning och där inga produktiva maskiner finns och vars syfte är att uppehålla och studera inkräktare.

Ett honeynet är alltså ett nät som enbart består av honeypots och/eller ironboxar och där inga ”riktiga” maskiner finns förutom kanske en loggserver. Det finns alltså inga normala användare i systemet. Maskinerna i ett honeynet kan vara både mjukvarubaserade som hårdvarubaserade. De maskiner som används i The Honeynet Project, är dock alla hårdvarubaserade. Alla maskiner som finns i ett honeynet övervakas noga av både NIDS och HIDS. Precis som med honeypots är alla som accessar honeynet troligen inkräktare och all trafik som sker i ett honeynet är misstänkt.

6.3. Vad bör man tänka på när man bygger ett vilseledande system?

En av de viktigaste sakerna att ta hänsyn till vid konstruktion av ett vilseledande system är hur kontrollen av en knäckt honeypot eller ironbox bedrivs, så att inte dessa används som plattform för fortsatta attacker över Internet eller LAN. För att klara detta måste det vara en bra kontroll över dataflödet så att det finns möjlighet att upptäcka när ett system har blivit knäckt. Den stora utmaningen ligger i hur dataflödet kan kontrolleras utan att angriparen blir misstänksam. När ett system har knäckts så är ofta en av knäckarens första åtgärder att begära en Internetförbindelse för att ladda ner olika programverktyg (toolkits), sätta upp en IRC (Internet Relay Chat) förbindelse, etc.

Om syftet med det vilseledande systemet är att lära sig vad knäckaren gör efter att denne fått kontroll över systemet, måste kanske angriparen tillåtas att begära Internetförbindelser, för att inte angriparen ska bli misstänksam. Det svåra är att ge knäckarna möjligheter att exekvera det de vill, men samtidigt inte tillåta dem att använda det knäckta systemet för att angripa andra system.

Är målet att bara studera eller uppehålla knäckaren så länge som det tar för denne att knäcka en honeypot bör förbindelsen genast avbrytas vid upptäckt att en honeypot har knäckts.

En brandvägg, som all trafik måste passera för att nå det vilseledande systemet, kan användas för att hålla räkningen på antalet förbindelser till utsidan som initierats från en honeypot. När en honeypot nått en viss gräns på antal förbindelser till utsidan kan brandväggen blockera resten. Detta ger angriparna möjlighet att exekvera det de vill samtidigt som det hindrar att det knäckta systemet får agera plattform för skanningattacker och DoS attacker.

I Honeynet-projektet har de kommit fram till att en honeypot kan tillåtas att ha mellan fem och tio stycken förbindelser till utsidan om det sköts automatiskt [HON 01]. Det brukar räcka för att inte göra angriparen misstänksam.

Har organisationen råd med att ha anställda som övervakar systemet 24 timmar om dygnet så kan man tillåta obegränsat med utgående förbindelser. Om en DoS-attack identifieras så kan personen som övervakar systemet avbryta attacken.

Genom att placera en router mellan brandväggen och det vilseledande systemet kan man uppnå två saker.

En router döljer brandväggen. När en honeypot har blivit knäckt, så kommer angriparna att finna en router mellan den och utsidan. Detta skapar en mer realistisk omgivning och gör det svårare att upptäcka brandväggen.

Det andra är att routern agerar som en andra kontrollenhet. Routern kompletterar brandväggen och ser till att knäckta honeypotar inte används för att angripa system på utsidan.

Ett vilseledande system kräver konstant underhåll och vaksamhet. För Att analysera 30 minuter som en knäckare lagt på en knäckt honeypot får man räkna med 30 till 40 timmars dataanalys [HON2 00]. IDS-signaturer blir omoderna, konfigurationsfiler blir förvanskade, brandväggar behöver uppdateras, diskar blir fulla. Man bör konstant testa och uppdatera systemet för att försäkra sig om att systemet fungerar som det ska. Man får aldrig underskatta motståndarna. Knäckare utvecklar hela tiden metoder för att undgå upptäckt, som anti-IDS-tekniker eller kryptering. Dug Song, säkerhetsarkitekt på Arbor Networks och medlem i *The Honeynet Project*[6], har utvecklat ett verktyg som kallas för ”Fragrouter” som är designat för att kringgå intrångsdetekteringssystem [SON 99]. Det fragmenterar paket i ett unikt mönster vilket gör det svårt för IDS att upptäcka signaturer. Det finns även andra liknande system som försöker kringgå IDS. De flesta IDS kan dock upptäcka dessa tekniker men det kan finnas några nya som är helt okända.

6.3.1. Var ska insamlad data sparas?

För att upptäcka intrång och lära sig hur de har gått till behövs så mycket information som möjligt. Det får bli en avvägning av hur mycket overhead man vill tillåta i systemet. Insamlad data som ska analyseras ska inte lagras lokalt i en honeypot. För information som lagras lokalt finns det en potentiell risk att de upptäcks av knäckaren och därmed varnar denne för att systemet är en honeypot. Den lagrade datan kan också förstöras eller försvinna. Därför bör datainformation lagras på ett avlägset system.

Genom att sammanställa information som hämtats från flera olika nivåer i systemet så går det att skapa sig en helhetsbild av läget. Brandväggen loggar all data som går till och från en honeypot. Det finns då möjlighet att upptäcka bakdörrar och det talar också om att ett system har knäckts. En nätverksbaserad IDS fångar all nätverkstrafik och hostbaserad IDS fångar alla användare och systemaktiviteter i ett system. Denna information av aktiviteter lagras på en avlägsen loggserver.

För UNIX-system och de flesta nätverksenheter görs detta helt enkelt genom att addera ett entry för en avlägsen loggserver i konfigurationsfilen. Ett standard-förfarande för knäckare är att disabla syslogd. Detta betyder att det inte längre finns någon mer kontinuerlig loggning men man har i alla fall informationen om hur knäckarna fick tillgång till maskinen. Sofistikerade knäckare kommer att försöka att ta kontroll över loggservern för att dölja sina spår. En loggserver är i allmänhet ett mycket mer skyddat system och därför kommer det att krävas mer avancerade tekniker för att knäcka den, tekniker som man kan lära sig genom att logga knäckaren under tiden.

[6] The Honeynet Project behandlas i avsnitt 7.2

7. Existerande vilseledande system

Det finns vilseledande system ute på Internet som används i forskningssyfte för att studera knäckare och deras metoder. Det finns även några få produkter som använder sig av vilseledning, de flesta av dem är kommersiella.

7.1. An Evening with Berferd …

Ett känt fall där vilseledande system använts är när Bill Cheswick 1991, som då var anställd på AT&T, konstruerade en mjukvarubaserad ironbox (eller ”chroot Jail” som Bill kallade det) för att studera en knäckare som angrep AT&T i USA [CHE 91]. En slutsats Bill drog var att om en knäckare får tag i en loginmaskin, så är det stor risk att hon/han blir root förr eller senare. Det finns många program som innehåller buggar och som körs med rooträttigheter, vilket ger en knäckare stora möjligheter att få rooträttigheter i systemet. En annan slutsats Bill drog var att det inte är värt besväret att sätta upp en ”chroot Jail”, även om det var intressant och lärorikt. Det är för svårt att få den korrekt och säker. Bättre är att ha en skrotmaskin med riktiga säkerhetshål och en maskin som övervakar trafiken på samma Ethernet som skrotmaskinen

7.2. The Honeynet Project

Honeynet Project är ett projekt med 30 medlemmar och vars syfte är att studera ”svarthattarna”[7] för att lära sig vilka verktyg de använder, vilken taktik de har och vilket deras motiv är [HON 00]. Genom att lära sig hur svarthattarna gått tillväga kan organisationer bättre förstå vilka hot som finns och hur de ska skydda sig.

Honeynet är ett nätverkssystem, det är placerat bakom en brandvägg som fångar och kontrollerar all ingående och utgående trafik. Den uppfångade informationen analyseras sedan för att lära sig hur knäckarna gått tillväga m.m. Alla system som placeras i ett honeynet är standard systemprodukter. Det är verkliga system och applikationer, precis som de du finner på Internet. Systemen kan vara olika typer av honeypots t.ex. Solaris, Linux, Windows NT, Cisco router, Alteon switch etc. Det här skapar en realistisk omgivning och genom att använda sig av olika system med olika tjänster (typ Linux DNS server, Windows NT server, Solaris FTP server) har projektmedlemmarna kunnat lära sig vilka olika verktyg och taktiker som används mot olika mål. Ingenting simuleras eller är gjort för att minska säkerheten i systemen.

Honynet är inget system som används i produktionen och därför är all trafik som lämnar eller kommer till Honeynet misstänkt. All förbindelse som initieras från utsidan av Honeynet till nätverket är troligen någon typ av skanningattack eller annan illasinnad aktivitet. All förbindelse som initieras från Honeynet till utsidan är en indikation på att ett system har blivit knäckt.

[7] Svarthattarna är författarnas översättning av blackhats, dvs knäckare.

7.3. Produkter som finns ute på marknaden

Sedan ett par år tillbaka finns det ett fåtal produkter ute på marknaden som använder sig av vilseledning. The Deception Tool Kit (DTK), sNET, ManTrap, Specter och Cybercop Sting är några sådana. Nedan beskriver vi kort dessa produkter utifrån den information vi fått från deras hemsidor. Vi har alltså inte själva lagt ned någon värdering om deras beskrivningar är riktiga eller inte.

7.3.1. DTK

DTK är ett verktygsset som Fred Cohen [8]& Associates har skapat, idén med DTK är att vilseleda en angripare så att angriparen tror att systemet, som använder sig av DTK, har ett stort antal kända säkerhetshål (sårbarheter) [COH 98]. Fred menar att port 365 ska användas för att visa att systemet använder sig av vilseledning som skydd. Detta talar då om för eventuella inkräktare att det inte är någon idé att försöka hacka sig in i det här systemet, eftersom det är skyddat med DTK eller något annat vilseledande system. Om det blir vanligt att använda sig av vilseledning så kommer angripare som vill undvika vilseledande system att först kolla port 365 och sedan istället söka upp ett annat system som inte använder sig av vilseledning. Om det då är många som använder sig av port 365 så kommer angriparna att förbruka tid på att leta upp system som inte använder sig av vilseledning. Några kommer säkert att använda sig av port 365 fast de inte har ett vilseledande system medan andra som har vilseledande system inte kommer att använda sig av port 365 för att t.ex. kunna studera knäckare som undviker vilseledande system. Detta kommer att bidra till att angriparna blir osäkra på hur systemen är skyddade.

DTKs vilseledning är programmerbar men är begränsad till att producera output i respons till angriparens input på ett sådant sätt att den ska simulera ett systems beteende som är sårbart för angriparens metoder. Enligt Cohen medför detta ett antal intressanta sidoeffekter:

- Det ökar angriparens arbetsbelastning eftersom de inte på ett lätt sätt kan avgöra vilka angreppsförsök som lyckas och vilka som misslyckas. En attack mot en falsk lösenordsfil förbrukar tid och ansträngningar för angriparen.

- Det tillåter oss att följa en angripares påbörjade försök och svara med motåtgärder innan de hittar ett säkerhetshål som vi är sårbara för.

7.3.2. sNET

sNET är en distribuerad nätverkslösning som sköter ett vilseledande system. Det är ett mycket sofistikerat system med komplex funktionalitet. Det innehåller ett antal komponenter så som: nätverksbaserade och hostbaserade IDS, honeypots,

[8] Fred Cohen är medlem i den tekniska staben på Sandia National Laboratories och chef för Fred Cohen & Associates i Livermore California, en grupp som är specialiserade på informationsskydd.

loggservrar, en kontrollenhet m.m. All kommunikation som sker mellan komponenterna använder sig av ett eget protokoll, sNet Protocol. I detta protokoll är det inbyggt anti-spoofing, anti-replay och autentiseringsmekanismer mellan enheterna.

Det är ett kommersiellt system och riktar sig i huvudsak mot stora organisationer. De som vill veta mer kan besöka sNET:s hemsida på www.snetcorp.com.

7.3.3. ManTrap

ManTrap och ManHunt är produkter som skapats av Recourse Technologies [REC 01]. ”ManTrap-burar” kan konfigureras för att likna värddatorer som för närvarande finns i nätverket genom att ha liknande namn, speglade filsystem, sekventiella IP-nummer och liknande servertjänster som finns i nätverket. En ManTrap bur kan härma ett annat system eller konfigureras för att se lite mer sårbart ut än de andra systemen i nätverket, vilket kan vara lockande för en angripare. För att skydda sig mot externa angripare kan ManTrap finnas på många olika sätt inom DMZ. Den kan t.ex. likna en publik FTP-, mail- eller webb-server.

ManTrap presenterar ett riktigt system för angriparen, komplett med vanlig data. Till skillnad från en simulerad omgivning som lätt kan avslöjas av erfarna angripare, är ManTrap byggd på en server med ett äkta operativsystem komplett med förväntade beteenden, applikationer och företagsinformation.

ManTrap loggar relevanta händelser som tangenttryckningar, processanrop och filaccesser. Loggfilerna kan lagras lokalt på värddatorn eller ännu hellre på en avlägsen värddator. För att garantera integriteten av loggfilerna använder sig ManTrap av digital signering av loggfilerna med hårdvara.

ManTrap kan konfigureras för att larma. Larmmeddelande baserade på specifika klasser av händelser. När en inkräktare accessar ManTrap så larmas genast systemadministratörer. De får kontrollen och låter dem avgöra vad de ska göra. ManTrap har få falsklarm eftersom all trafik riktad till en ManTrap bur kan anses misstänkt.

Det kan finnas fyra stycken ManTrap system på en värddator. ManTrap är den enda tillgängliga produkt som använder vilseledande värddatorteknologi. Andra tillgängliga produkter använder sig bara av ett fåtal virtuella klient-server nätverksstackar. ManTrap använder en hel Solaris-omgivning som tillåter i princip alla applikationer att exekvera i buren.

Mantrap installerar hundratals av användare och kontonamn, e-post-filer, och systemfiler. Den simulerar också en aktiv e-post-server genom att vid slumpmässiga intervall skapa nya e-postmeddelanden.

ManTrap kan även kombineras med ManHunt som spårar ursprungskällan för angreppet.

ManHunt
ManHunt använder sig av avvikelsedetektering i realtid, den fångar allt som är utanför normala eller förväntade protokoll. Den använder sig också av statistisk analys av händelser för positiv identifiering eller prioritering av potentiella attacker, vilket minskar falsklarmen.

ManHunt kan automatiskt spåra källan till attacken genom kedjan av ISPs (Internet Service Providers) genom hela Internet, även attacker med förfalskade IP-adresser. På några minuter kan den skära av attacken från dess källa och vara i full tjänst.

ManTrap och ManHunt är kommersiella produkter och kan användas av både mindre och större organisationer. Mer information finns på Recourse hemsida: www.recourse.com.

7.3.4. Specter

Specter är en honeypot som är tillverkat av NETSEC i Schweitz. Specter kan köras på Microsoft Windows 2000 eller NT. Den kan simulera 11 stycken olika operativsystem, fem olika nätverkstjänster (FTP, Telnet, SMTP, Finger och http) och sju protokollfällor t.ex. DNS, IMAP, SUN-RPC och POP3. Specter kan också simulera det trojanska programmet Netbus [SPE 01].

Det går även att ställa in vilken nivå av säkerhetskonfiguration man vill att systemet ska simulera:

Open - Systemet uppför sig som ett dåligt konfigurerat system vad det gäller säkerhet.

Secure - Systemet uppför sig som ett välkonfigurerat system vad det gäller säkerhet.

Failing - Systemet uppför sig som en maskin med flera olika hårdvaru- och mjukvaruproblem.

Strange - Systemet uppför sig oförutsägbart, med avsikten att inkräktaren ska undra vad det är som händer.

Aggressive - Systemet kommunicerar så länge som det är nödvändigt för att samla information om inkräktaren, sedan avslöjar den sin riktiga identitet innan den bryter förbindelsen.

Vid varje uppkoppling mot en simulerad tjänst, så utför Specter i det tysta en Finger och/eller portskanning mot inkräktarens maskin. Har angriparen inte dolt sina spår så kan det ge värdefull information som kan användas om juridiska åtgärder blir aktuella.

Den som vill veta mer kan gå till Specters hemsida: www.specter.com

7.3.5. CyberCop Sting

CyberCop Sting tillverkas av PGP som är en del i Network Associates Inc. (NAI). CyberCop Sting är en honeypot som ska lura potentiella angripare medan den spårar ursprunget av den misstänkta aktiviteten i nätverket [PGP 01]. CyberCop Sting skapar ett virtuellt nätverk av routrar och arbetsstationer samt övervakar all trafik till dessa virtuella värdar. Den kan simulera ett nätverk av Windows NT 4.0 servrar, Solaris 2.6 servrar och Cisco IOS 11.2 routrar. Varje virtuell nätverksenhet har en riktig IP-adress och kan sända och ta emot riktiga paket från och till den större nätverksomgivningen. Varje virtuell nätverksnod kan köra simulerade demoner som t.ex. finger, telnet, SMTP och FTP.

CyberCop Sting finns att laddas ner gratis på prov i 30 dagar. Vi gjorde detta men kunde dock inte att testa CyberCop Sting eftersom det inte ingick någon instruktionsfil (vilket det skulle göra enligt andra textfiler), trots att vi ringde, mejlade och faxade till NAI fick vi inte tag på några instruktioner.

8. Analys I

8.1. Inledning

I avsnitt 2.1.1 har vi tagit upp hur svårt det är att mäta säkerhet och att det inte finns någon generell metod för detta. För att kunna svara på vår problemformulering ”Kan säkerheten ökas med hjälp av vilseledande system och i så fall hur?” så har vi valt att föra en diskussion om detta. I detta kapitel analyserar vi några exempel på vilseledande system och i kapitel 9 analyserar vi hur testning av vilseledande system skulle kunna visa om de ökar säkerheten.

Hur ett vilseledande system är uppbyggt beror på ett antal faktorer som resurser (pengar), tid, storlek, vilka komponenter ska ingå, att det följer säkerhetspolicyn, vad ska det skydda mot etc. Ju större det är, ju mer komplext blir det och ju mer tid kommer det att krävas för att underhålla det.

För att göra det enkelt att följa med i vårt resonemang kommer vi att välja enkla konfigurationer med få applikationer och komponenter. Liknande resonemang kan göras där fler enheter och applikationer används.

8.2. Honeypot i det produktiva nätet

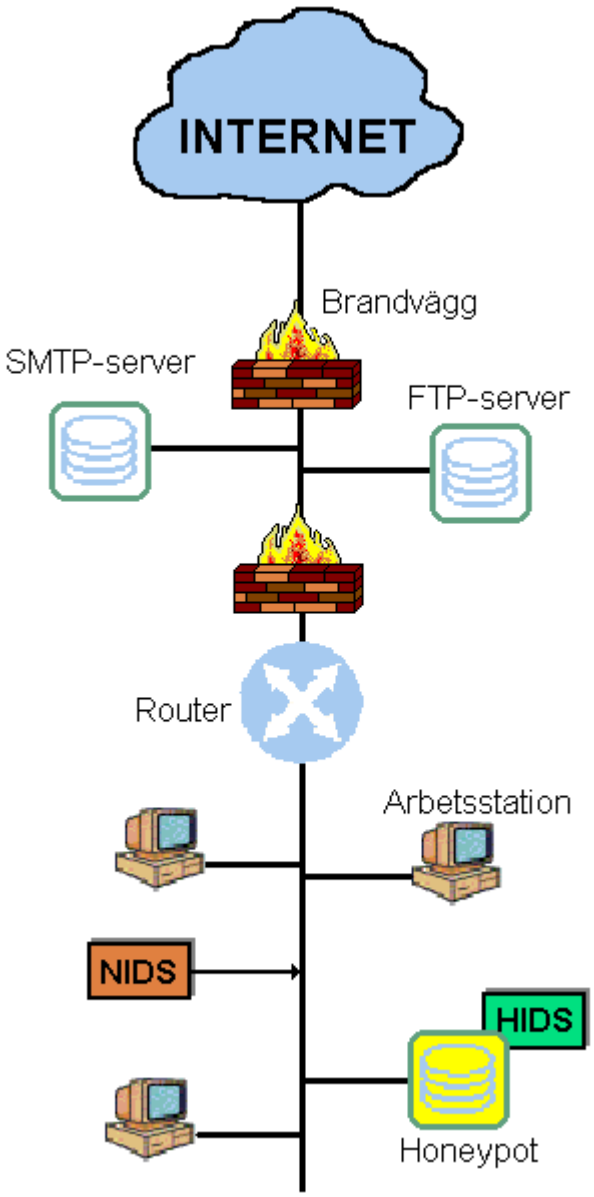
I figur 11 illustreras hur en organisations LAN skulle kunna se ut som använder sig av en honeypot. Mot Internet skyddas det av två stycken brandväggar. Mellan de båda brandväggarna i den demilitariserade zonen finns det två stycken servrar, en SMTP-server och en FTP-server. I det produktiva nätet finns det några arbetsstationer och en honeypot. Brandväggen närmast Internet tillhandahåller två tjänster, FTP på port 21 och SMTP på port 25 (e-post-meddelande).

8.2.1. Vad upptäcker brandväggen?

Det finns flera anledningar till att använda sig av en brandvägg i ett nätverk. En brandvägg är ett bra skalskydd och beroende på vilken modell av brandvägg som används, har dessa olika kapacitet att analysera trafiken genom sig i båda riktningarna (se kap 3).

I en brandvägg har vi möjlighet att kontrollera vilka IP-adresser som används. Denna möjlighet har vi med både inkommande och utgående paket. När en inkräktare har lyckats med en penetrationsattack finns det möjligheter för inkräktaren att föra ut information.

Om inkräktaren accessar honeypoten och försöker sända ut information genom brandväggen, så har brandväggen möjlighet att upptäcka detta, eftersom brandväggen kontrollerar avsändarens IP-adress.



Figur 11 Honeypot i LAN

Genom att en brandvägg har möjlighet att analysera datapaketen som sänds genom sig innebär detta att vi har möjlighet att upptäcka, om någon sänder information från honeypoten till Internet via brandväggen, genom att analysera informationen i datapaketen. Om informationen i honeypoten är utformad på ett sådant sätt att brandväggen kan avgöra om informationen sänds från honeypoten innebär detta att vi kommer att upptäcka en inkräktare som använder sig av en godkänd användares användarkonto. Vi kommer även att upptäcka insiders som försöker sända ut information från honeypoten.

Om inkräktaren har ändrat på informationen i honeypoten på ett sådant vis att informationen är godkänd enligt säkerhetspolicyn, finns det en risk att brandväggen missar information som sänds från honeypoten.

Genom att använda sig av en brandvägg i ett vilseledande system finns det möjlighet till att analysera hur inkräktaren har tagit sig in i systemet och på detta sätt täta säkerhets luckor.

8.2.2. Vad upptäcker IDS?

Det kommer alltid att finnas inkräktare som klarar av att passera brandväggen och som kommer in i systemet utan att vi förstår hur de kommer in. Det är därför viktigt att försöka hitta inkräktarna på något sätt. IDS är det traditionella sättet att upptäcka sådana intrång.

De värdbaserade IDS som är på SMTP-servern respektive FTP-servern kan bara upptäcka attacker riktade mot respektive server. Upptäcker de en attack så kan de skicka larmmeddelanden till säkerhetspersonal. De kan också göra en mer aktiv respons som att meddela brandväggarna att de ska blockera trafik som kommer från angreppets avsändaradress.

8.2.3. Vad upptäcker en honeypot?

Det finns ännu inga IDS som upptäcker alla intrång. Genom att använda sig av honeypot finns det möjlighet att upptäcka intrång som IDS missar.

Genom att placera en honeypot i det produktiva nätet, se figur 11, kan inkräktare som lyckats ta sig förbi brandväggarna eller insiders som snokar runt i det egna nätet upptäckas om de accessar honeypoten. För att inkräktare inte ska bli misstänksamma bör honeypoten efterlikna något som kan förväntas finnas i nätet. I det här fallet kan honeypoten efterlikna t.ex. en arbetsstation eller en NT-server.

Frågan är om en honeypot ökar säkerheten i det här fallet eller om det räcker med brandväggar och IDS som skydd? Som vi nämnt i kapitel 6.2.1 så är en honeypot en passiv enhet och bara upptäcker intrång som är riktade mot den själv. Om honeypoten är tillräckligt lockande, gärna lite mer än övriga enheter i samma nät, så kan vi förutsätta att den kan öka skyddet om följande gäller:

- Om IDS inte upptäcker intrånget eller om IDS upptäcker intrånget betydligt senare. - Om en IDS upptäcker samma intrång ungefär lika snabbt som honeypoten, så har ju inte honeypoten tillfört något. Eftersom en honeypot inte har några falsklarm kan den upptäcka intrång väldigt fort. Det räcker ju att någon försöker accesssa den.

- Vi kan analysera intrånget och därefter uppdatera vårt system. - Klarar vi inte av att analysera och uppdatera systemet så har vi ju inte ökat skyddet.

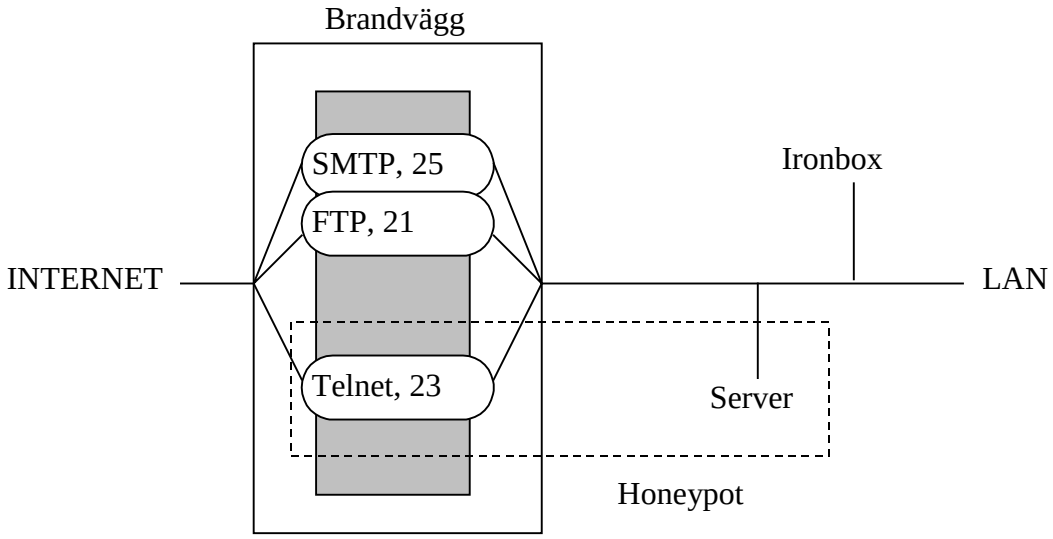
- Om angriparen inte lyckas knäcka honeypoten. - Om angriparen lyckas knäcka honeypoten så kanske vi kan dra lärdom av det i framtiden men i nutid kan angriparen göra fortsatta attacker mot ”riktiga maskiner”, med honeypoten som bas. Vi anser då att vi snarare har minskat säkerheten eftersom vi anser att vad som sker i nutid är betydligt viktigare än vad som sker i framtiden.

Om det är ett system med många användare och flera servrar så kan det vara svårt att få honeypoten att verka mer lockande än andra enheter. Honeypoten kan ju fortfarande ge ett ökat skydd för systemet men det kanske är för lite i förhållande till vad det kostar i tid och pengar för att underhålla honeypoten. För att attrahera angripare bättre kan vi ha ett lockbete redan i den första brandväggen.

8.3. Honeypot i brandvägg

I figur 12 har vi byggt ut brandväggen med tjänsterna Telnet, port 23, och FTP, port 21, detta för att visa hur vi kan använda oss av en Honeypot tillsammans med en brandvägg i ett vilseledande system. Vi har också satt in en ironbox för att förtydliga skillnaden mellan en ironbox och en honeypot.

Telnet är i detta fall ett lockbete till honeypoten. Om någon på utsidan försöker anropa Telnet svarar Telnet precis som vanligt, men den har ingen koppling till det inre systemet utan endast till en server dvs. om någon loggar in på Telnet kommer personen att endast kunna utforska servern och inte övriga delar av LAN. Denna server har en konstgjord struktur som ska vara intressant att utforska för en inkräktare. Lockbetet i brandväggen och servern arbetar tillsammans och ska betraktas som en honeypot. Anledningen till att använda sig av lockbete i brandväggen är att en inkräktare som har skannat brandväggen på utsidan ser att brandväggen tillåter tre tjänster, SMTP, FTP och Telnet. I och med att det finns vissa sårbarheter med Telnet, finns det en möjlighet att inkräktaren försöker ta sig in via dessa tjänster.



Figur 12 Brandvägg med honeypot

En av nackdelarna med Telnet är att den sänder användarnamn och lösenord i klartext. När Telnet används som lockbete är det tänkbart att en inkräktare avlyssnar trafiken till denna demon. För att upptäcka om det är någon som avlyssnar trafiken, kan den säkerhetsansvarige logga in sig på Telnet hemifrån

några gånger i månaderna, detta är naturligtvis något som kan ske automatiskt. De användarnamn och lösenord som den säkerhetsansvarige använder sig av bör endast användas till Telnet. Inkräktaren har då möjlighet att spela in användarnamn och lösenord och använda dessa för att komma in i systemet.

När någon försöker accessa Telnet med ett användarnamn som tillhör honeypoten kommer inkräktaren att kopplas upp mot den server som ingår i honeypoten. Denna server visar en trovärdig omgivning för inkräktaren. Säkerhetspersonal har nu möjlighet att studera hur inkräktaren försöker få rooträttigheter och på detta sätt lära sig mer om hur en knäckare arbetar och vad som behövs ändras för att få det korrekta systemet säkrare.

Genom att ha lockbete redan vid brandväggen ökar vi sannolikheten att en inkräktare försöker attackera lockbetet och inte en av de riktiga tjänsterna. I de fall en inkräktare ändå skulle attackera en av de korrekta tjänsterna och vi upptäcker detta med IDS, har vi möjlighet att förflytta inkräktaren till ironboxen.

8.3.1. Analys av ironbox

Att automatiskt flytta över en inkräktare som försöker knäcka t.ex. vår SMTP-server, till en ironbox är inte oproblematiskt. Eftersom vi inte kan vara säkra på hur långt inkräktaren hade kommit innan hon/han upptäcktes så kan inkräktaren bli misstänksam om information, som tidigare gick att få tag på, inte längre är tillgänglig. För att inkräktaren inte ska bli misstänksam är det viktigt att ironboxen har samma information som angriparen har fått från den riktiga servern. Om detta sköts automatiskt finns det risk att information som inkräktaren inte har fått tillgång till följer med till ironboxen.

Det bästa är kanske att stänga ner SMTP-server tills säkerhetspersonal har analyserat intrånget och satt upp ironboxen baserat på analysen. Nästa gång samma inkräktare (från samma avsändaradress) försöker accessa servern så överför brandväggen direkt inkräktaren till ironboxen samtidigt som säkerhetspersonal larmas.

Om en attack med samma signatur som tidigare, fast med en annan avsändaradress, initieras mot servern så kan vi anta att det är samma angripare. Vi får då ta ställning till om angriparen ska föras till ironboxen direkt eller om vi ska konfigurera om ironboxen innan vi gör detta.

I en ironbox ställs det högre krav på att informationen är trovärdig jämfört med en honeypot eftersom den ska stämma överens med den information som inkräktaren har kommit över innan vi har upptäckt denne. För att en ironbox ska fungera bra anser vi att säkerhetspersonal ska interagera med ironboxen under en attack. Säkerhetspersonal kan då på plats avgöra om viss information ska lämnas till angriparen eller inte.

8.3.2. Nackdelar med lockbete i brandvägg

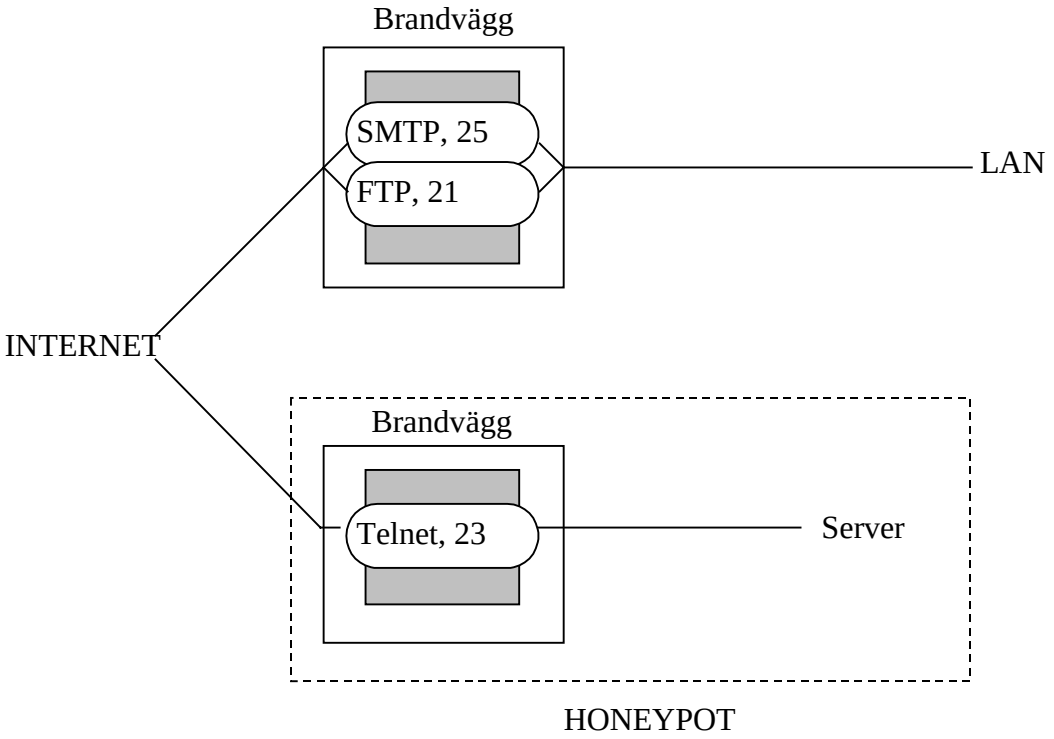
Att använda sig av lockbete i brandväggen medför en ökad risk att vi lockar till oss angripare som annars aldrig hade angripit oss. Många knäckare använder sig av script som skannar Internet efter maskiner som har trojanska hästar (t.ex. Netbus) eller maskiner som har tjänster med kända sårbarheter (t.ex. Telnet). En knäckare som skannar Internet efter Telnettjänsten skulle aldrig accessat vårt system om vi inte hade haft en honeypot som efterliknar Telnet. Så frågan är om vi ökar säkerheten genom att locka till sig angipare som vi sedan kan studera?

Genom att studera en knäckare som accessar vår Telnet-honeypot kan vi lära oss hur vi bättre ska skydda vårt nät men vi har samtidigt utsatt oss själva för risken att angriparen knäcker vår honeypot. Om vår honeypot blir knäckt så har angriparen en plattform inne i vårt LAN varifrån angriparen kan angripa andra enheter i LAN. Enheter som kanske innehåller viktig och känslig information. Vi har i detta fall, genom att använda oss av en Telnet-honeypot, snarare minskat än ökat säkerheten i vårt nätverk.

8.4. Honeypot avskiljt från det produktiva delnätet

Som vi nämnt ovan finns det en nackdel med att använda sig av en honeypot som arbetar både i brandväggen och i LAN eftersom det finns en risk att inkräktaren knäcker honeypoten utan säkerhetsansvarigas vetskap. När honeypoten är knäckt ligger hela LAN öppet för vidare attacker, eftersom vi har lett in inkräktaren genom brandväggen. Detta skulle kunna lösas genom att använda sig av två parallella brandväggar, se figur 13. Dessa brandväggar har samma IP-adress men de lyssnar på olika portnummer. Det är viktigt att dessa brandväggar kan kontrollera om det är fritt fram att sända eller om den andre brandväggen redan sänder.

Genom att använda sig av två brandväggar får vi ett separat system för vilseledning. Om en inkräktare attackerar Telnet och kommer in i systemet, så har inkräktaren endast kommit till en server som ingår i honeypoten. Om inkräktaren knäcker servern finns det inga möjligheter att fortsätta intrånget in i LAN. Inkräktaren har fortfarande möjlighet att använda honeypoten som en plattform för vidare attacker ut på Internet.



Figur 13 Vilseledande system med parallella brandväggar

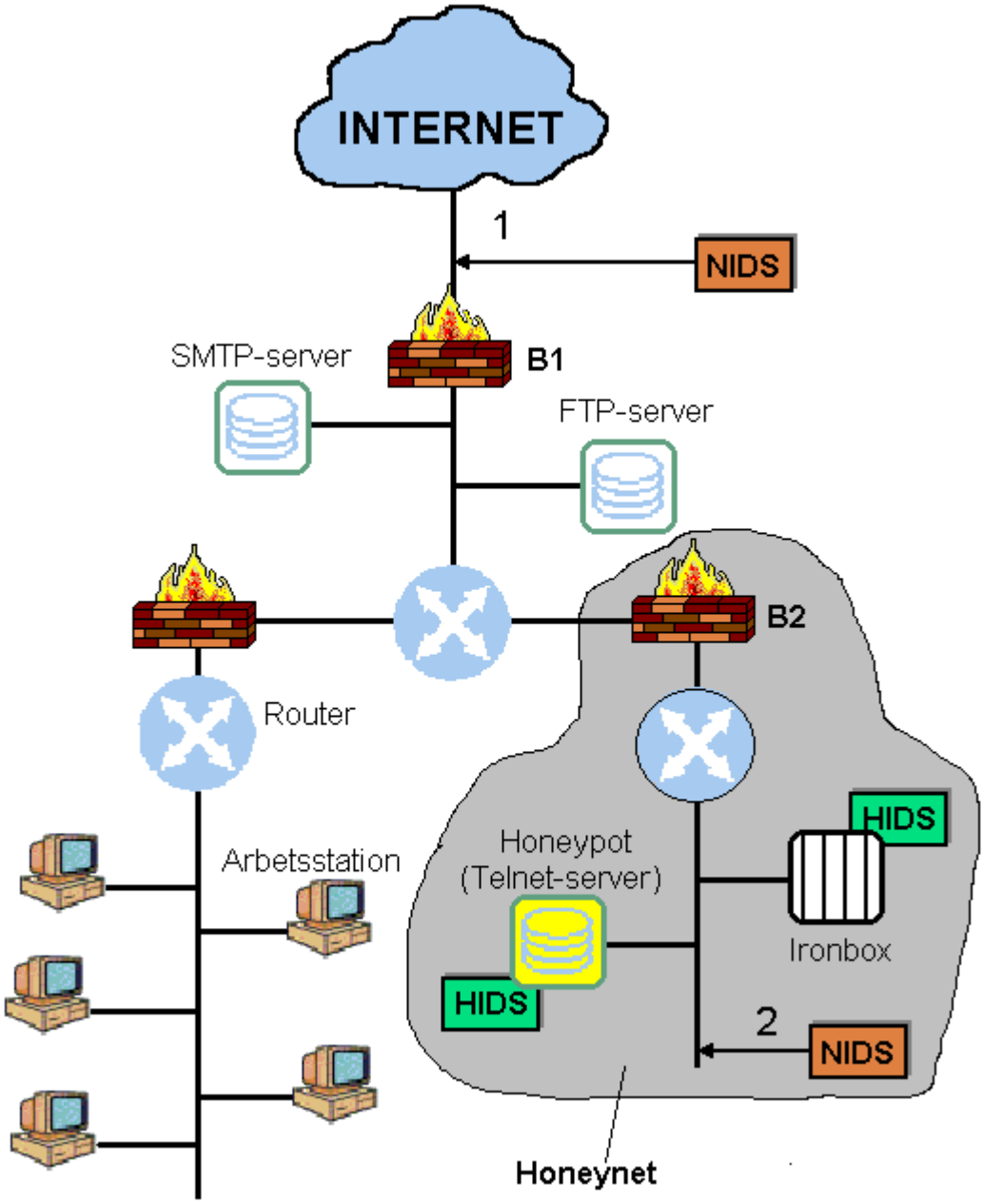
En stor fördel med att dela upp systemet på detta sätt är att det riktiga systemet inte kommer att få in inkräktaren i sitt system, när personen triggar Telnet. Om den systemansvarige upptäcker att en inkräktare har fått rooträttigheter finns möjligheten att stänga ner hela honeypoten och boota om enheten. Detta kan göras utan att det riktiga systemet berörs.

8.4.1. Tekniken med honeynet

I figur 14 visar vi hur det vilseledande systemet i figur 13 mer konkret skulle kunna vara uppbyggt. Brandvägg (B1) är konfigurerad med de riktiga tjänsterna FTP och SMTP samt med tjänsten Telnet som lockbete, se figur 13. Mellan Internet och den första brandväggen (B1) ligger en NIDS (1). Den kan upptäcka DoS-attacker och portskanning attacker. Upptäcker den en skanning attack så talar den om för brandväggen vilken avsändaradress portskanningen utgår ifrån. Brandväggen vet sedan att trafik som kommer från den adressen troligen är från en angripare. När brandväggen upptäcker trafik som kommer från potentiella angripare kan den vidarebefordra sådan trafiken till en ironbox, samtidigt som den larmar säkerhetspersonal.

I ironboxen kan den potentiella angriparen noga övervakas medan säkerhetspersonal under tiden försöker verifiera om det är en angripare eller inte. Det kan ju vara så att angriparen har spoofat en ärlig användares adress under

skanningattacken och sedan har den intet ont anande användaren helt ärligt försökt accessat nätverket och därmed blivit misstänkt.



Figur 14 Vilseledande system

För att inte utsätta det produktiva nätet i onödan har ironbox placerats i ett Honeynet d.v.s. ett delnät där det inte finns några produktiva maskiner. Om en angripare som ovetandes förts över till en ironbox skulle få rooträttigheter över ironboxen så gör detta inte så mycket, eftersom ironboxen inte innehåller någon viktig information. Angriparen kan ju fortsätta och göra attacker mot andra enheter i delnätet med ironboxen som plattform och det är precis detta man vill.

Säkerhetspersonal kan då lära sig hur angriparen går tillväga eftersom angriparens aktiviteter hela tiden noga övervakas och loggas.

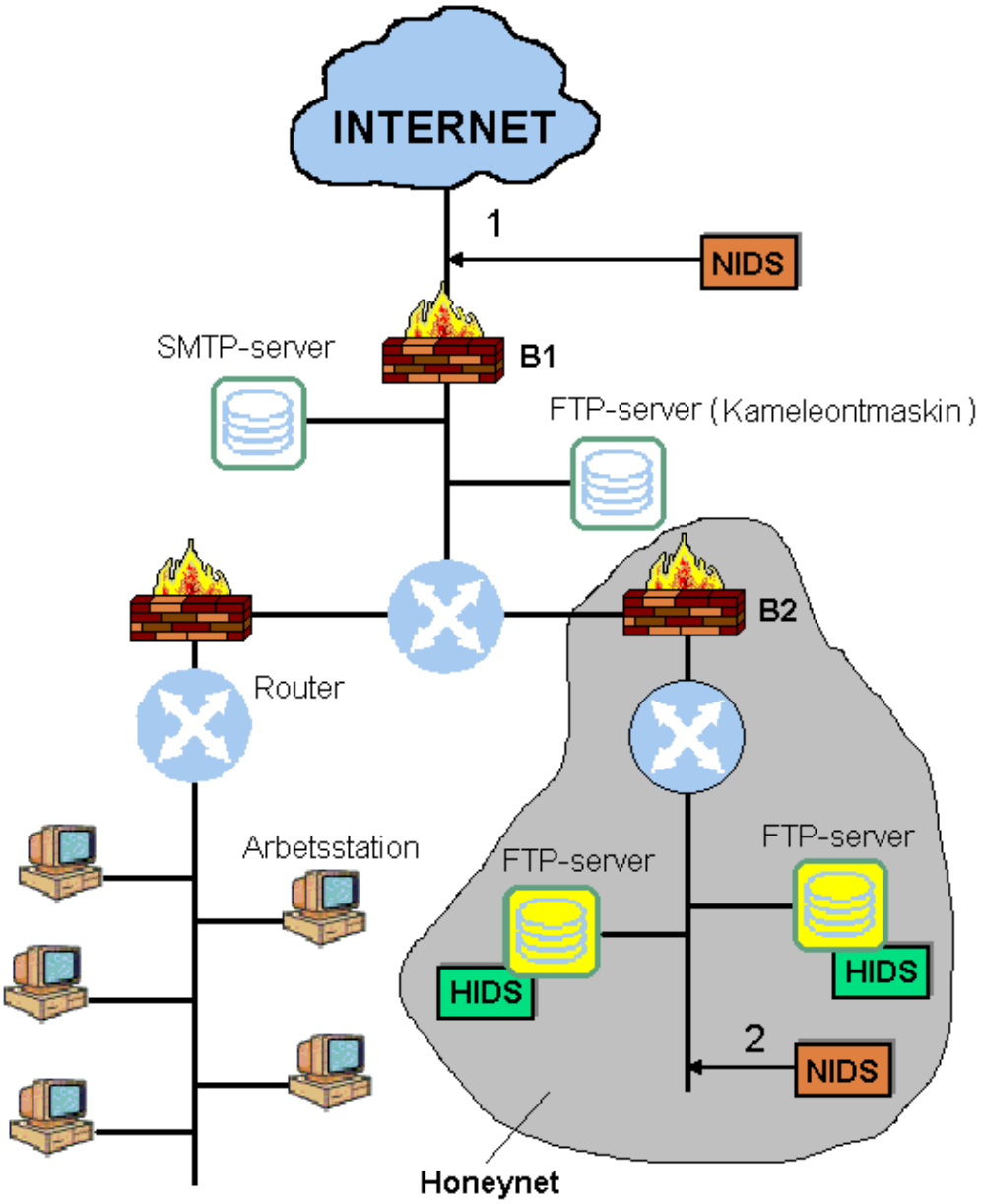
En fördel med att använda sig av ett honeynet är att det nästan inte ger någon prestandamässig overhead på det produktiva nätet eftersom det är avskiljt från detta. Det gör heller inte så mycket om andra enheter i ett honeynet knäcks, eftersom dessa ändå inte innehåller någon viktig eller riktig information. Det är betydligt allvarligare om angriparen med det knäckta systemet som bas försöker angripa system som ligger utanför honeynet. Som skydd mot detta har vi brandvägg (B2). Eftersom honeynet inte är något produktivt nät så är all trafik som initieras från det misstänkt. Brandvägg (B2) upptäcker all sådan utgående trafik och kan blockera denna om så önskas.

Den enda trafik som lämnar honeynet är kodade meddelande till säkerhetspersonal, möjligtvis också loggdata som ska sparas på en server i en annan del av systemet.

8.4.2. Tekniken med kameleontmaskiner

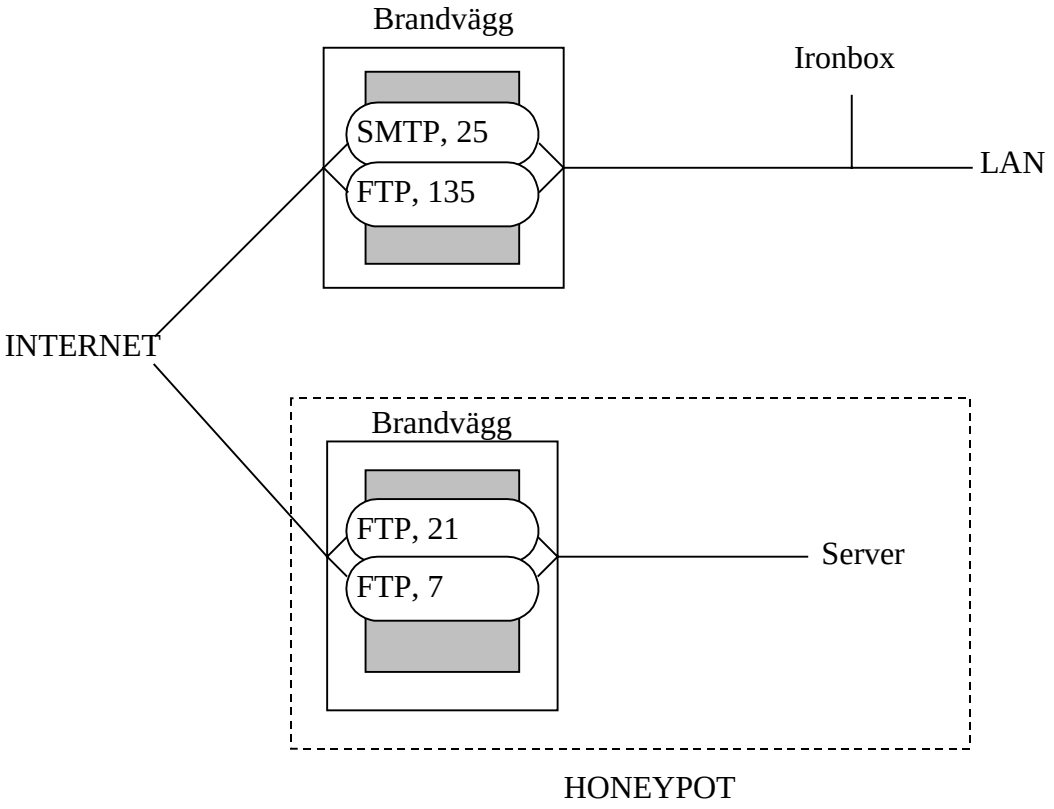
För att öka skyddet för FTP servern i figur 15 låter vi den vara en så kallad kameleontmaskin. Det innebär att det finns en eller flera attrapper som efterliknar FTP-servrar. I det här fallet har vi två honeypotar som gör det och som är placerade i honeynet, se figur 16.

I det här fallet har vi tänkt att attrapperna ska vara helt identiska med den riktiga servern förutom att de inte innehåller samma information, vi har alltså inte lagt in några svagheter i attrapperna. Genom att ha tre stycken nästan identiska servrar så minskas sannolikheten att angriparna går på den riktiga och vi har därmed ökat säkerheten mot angripare som inte är medvetna om vilken port som är den riktiga. Skyddet mot insiders kan dock anses vara oförändrat eftersom vi anser det troligt att en insider vet eller lätt kan ta reda på, vilken FTP-port som är riktig. I figur 15 illustreras vilka tjänster som brandväggen har.



Figur 15 Kameleontmaskin

Förutom de risker som det finns generellt med honeypots så finns det ytterligare en risk med att använda sig av kameleontmaskiner. Vad händer om en angripare t.ex. knäcker en av kameleontmaskinens attrapper och en normal användare av misstag försöker accessa attrappen?



Figur 16 Brandvägg med kameleontportar.

När användaren försöker logga in sig på attrappen så tror användaren att det är den riktiga FTP-servern och kommer därför ge sitt användarnamn och lösenord. Angriparen kommer således att få tag på både ett giltigt användarnamn med tillhörande lösenord. Dessa kan angriparen sedan använda för att leta upp och accessa den riktiga FTP-servern. Angriparen behöver inte använda sig av den knäckta attrappen som bas. Även om vi får reda på att attrappen har blivit knäckt så kanske vi inte får reda på att angripare har kommit över ett riktigt användarnamn med tillhörande lösenord till vår riktiga FTP-server.

Det finns alltså en fara i att använda sig av attrappen som efterliknar riktiga befintliga servrar i systemet. Om vi använder oss av många FTP-attrapper, hundra stycken, så minskas sannolikheten att en angripare accessar den riktiga FTP-servern. En angripare som fått tag på riktiga användarnamn och lösenord och som skannar brandväggen för att leta upp den riktiga FTP-servern kan då upptäckas av NIDS(1), se figur 15. Om vi spelar in de användarnamn och lösenord som angriparen använder för att försöka ta sig in i FTP-attrapperna kan vi testa dem mot den riktiga servern. Kommer vi då in i den riktiga FTP-servern så vet vi att angriparen har kommit över riktiga användarnamn med lösenord. Det vi då får göra är att göra dessa lösenorden ogiltiga. Vi bör också försöka analysera hur angriparen knäckt en attrapp så att vi kan säkra upp vår riktiga FTP-server. Om angriparen knäckt en attrapp så kan även den riktiga maskinen

knäckas. Det kan i värsta fall vara nödvändigt att stänga ner servern medan vi försöker hitta ett skydd mot intrånget.

För att undvika att användare av misstag accessar attrapper så bör portnummren vara valda på ett sådan sätt att det är svårt att av misstag, t.ex. genom att slå en siffra fel, att accessa attrapper istället för den riktiga maskinen.

Att använda sig av kameleontmaskiner innebär med stor sannolikhet att angriparen förstår eller misstänker att det är ett vilseledande system han har accessat. Helst bör man aldrig lämna några ledtrådar till angriparna men i det här fallet kan det göra angriparen osäker på att den information som fås är riktig eller inte.

8.5. Diskussion

I detta avsnitt diskuterar vi några aspekter som vi tycker är viktiga att få med i vår rapport.

8.5.1. Tänkt målgrupp för vilseledande system

En målgrupp för vilseledande system är större företag. Det gäller att företaget har en kunnig och tillräckligt stor säkerhetsavdelning, som har möjlighet att underhålla och utveckla systemet med information under en stor del av dygnet. Företag som inte ligger i spjutspetsen var det gäller den senaste vidareutbildningen bör fundera på om dem har tillräcklig kompetens inom företaget, för att underhålla systemet. Det krävs även goda kunskaper att analysera vad inkräktare gör och vilka mot åtgärder som bör vidtagas mot dem.

En annan målgrupp kan vara utvecklings- och forskningsavdelningar som använder systemet till att utbilda sig i hur en knäckare går tillväga vid intrångs-försök. Om systemet endast används för utbildning bör det vara ett helt fristående vilseledande system som används för vilseledning.

8.5.2. Visa port 365

I kapitel 7.3.1 tar vi upp att Fred Cohen anser att port 365 ska användas för att visa att systemet använder sig av vilseledning. Detta ska då tala om för eventuella inkräktare att det inte är någon idé att försöka hacka sig in i det här systemet, eftersom det är skyddat med t.ex. DTK. Ett argument vi har mot detta är att om någon hittar en sårbarhet i DTK så kan denna person enkelt leta upp system som använder sig av DTK och utnyttja sårbarheten. Vi anser att man ska ge så lite ledtrådar som möjligt om hur ens system är skyddat.

8.5.3. Installation och administration

En färdig produkt borde inte var svår att installera och konfigurera upp till ett fungerande system. Detta under förutsättning att produkten har de tekniska lösningar som användaren behöver. Att tillverka ett eget vilseledande system kan

var allt från enkelt till väldigt komplicerat, beroende på nivån av vilseledning som ska uppnås. Ett enkelt system kan bestå av en dator med ett äldre operativsystem med en IDS medan ett större system endast begränsas av fantasin och den egna kompetensen.

För att administrera ett vilseledande system behövs kunskaper i att analysera data och veta vilka motåtgärder som ska vidtagas vid attacker mot systemet. Under den tid som systemet inte blir attackerat bör användas för att kontrollera innehållet i systemet, eftersom innehållet måste vara aktuellt. Den administrativa personalen måste även kolla om någon har gjort något intrång som inte har upptäckts.

8.5.4. Fördelar med vilseledande system

En fördel med vilseledande system är att de ökar det totala skyddet mot intrång. Det görs genom att skapa en intressant omgivning av lockbeten för inkräktaren. Inkräktaren kommer att lockas att attackera ett lockbete och på detta sätt ledas förbi de korrekta tjänsterna. Ser vi ett vilseledande system ur denna synvinkel ökas skyddet.

Vilseledande system ökar sannolikheten att upptäcka nya former och typer av attacker. Anledningen till att nya attacker kommer att upptäckas är att en honeypot i ett vilseledande system har till uppgift att logga vad som händer på dessa enheter och allt som sänds från honeypoten kan ses som tecken på intrång.

Ett vilseledande system bör inte påverka de normala användarna, eftersom de vilseledande komponenterna är passiva enheter. Om en användare triggar tjänster eller accessar ett lockbete eller beter sig på ett sätt som inte stämmer överens med säkerhetspolicyn kommer systemet att logga användaren.

8.5.5. Nackdelar med vilseledande system

När en knäckare har upptäckts i ett vilseledande system loggas inkräktarens beteende. För att kunna dra nytta av detta måste säkerhetspersonalen vara tillräckligt kunnig för att klara av att analysera loggarna. Eftersom jorden är rund är det möjligt att attackerna kommer när som helst på dygnet och därför bör det finnas säkerhetspersonal dygnet runt.

Ett säkerhetssystem bör vara så enkelt att implementera och underhålla som möjligt för att minska risken för buggar och mänskliga misstag i systemet. Ett vilseledande system ökar komplexiteten på säkerhetssystemet eftersom det är flera komplexa enheter som ska arbeta tillsammans. De vilseledande komponenterna kräver även mer underhåll en andra säkerhetsmekanismer.

Eftersom vi inte kan avgöra exakt när en honeypot eller ironbox har blivit knäckt, bör dessa installeras på fristående enheter. Detta för att minska störningar på övriga komponenter i systemet när dessa enheter bootas om. Att installera honeypot och ironbox på fristående enheter ökar kostnaden för hårdvara i systemet.

Det finns idag inte några möjligheter att automatiskt kontrollera om en honeypot eller en ironbox har blivit knäckt. Utan detta måste skötas manuellt av säkerhets-personal.

8.5.6. Overhead

Den extra kommunikations-overhead (loggdata och meddelanden till säkerhets-personal) som fås med ett vilseledande system är relativt liten, eftersom de vilseledande komponenterna är passiva enheter. Om de vilseledande enheterna är installerade på egna maskiner (vilket rekommenderas) så ger de inte heller någon prestandamässig overhead på de ”riktiga” maskinerna utan ökar endast overhead lokalt.

8.5.7. Omogen marknad

Fastän vilseledande system har använts i mer än 10 år, så har vi bara hittat fem produkter, varav tre/fyra är kommersiella produkter som kostar från 10 000 kr och uppåt. Ingen av dessa produkter har mer än fyra år på nacken. Vi anser att marknaden är relativt omogen ännu.

9. Analys II

I det här kapitlet diskuterar vi hur test av vilseledande system kan göras för att få en uppfattning om huruvida säkerheten ökar för ett system som använder sig av vilseledning. Målet med ett sådant test är att försöka få fram om antal lyckade intrång minskar per tidsenhet (kalendertid) i ett vilseledande system kontra ett system som bara använder sig av brandväggar och IDS som skydd.

9.1. Test av honeynet mot Internet

Ett sätt att testa vilseledande system skulle vara att sätta upp ett antal system som använder sig av vilseledning (honeynet) och lika många likvärdiga system, fast utan vilseledning, som skulle användas som referensgrupp. Alla enheter i systemen ska övervakas med värddatorbaserade IDS för att vi ska kunna analysera om intrång har skett. Systemen hade man sedan kunnat ha anslutna till Internet under en längre tidsperiod. Med jämna mellanrum (varje dag) skulle man sedan gå igenom alla system för att se vilka system som hade fått intrång. Skulle det då visa sig att de vilseledande systemen hade markant färre lyckade intrång så skulle det vara en indikation på att vilseledande system faktiskt ökar skyddet.

En nackdel med en sådan undersökning är att man skulle behöva testa ett ganska stort antal system under en längre testperiod för att få ett riktigt statistiskt underlag. Det är inte heller säkert att man upptäcker alla intrång eftersom det förutsätter att alla intrång kan detekteras - något som är svårt/omöjligt att göra.

En sådan undersökning tar ju inte heller hänsyn till vilken kunskap, erfarenhet, resurser m.m. som angriparna har. Undersökningen skulle ju också bara visa hur systemet fungerar som skydd mot outsiders och inte mot insiders. Som vi nämnt tidigare så finns det undersökningar som visar att de allra flesta intrång sker av insiders. Därför hade det varit mer intressant att använda systemet i en mer realistisk miljö där man även hade kunnat studera hur vanliga användare påverkas.

9.2. Test i verklig miljö

Även om ett vilseledande system testas ute på Internet, så ger detta inte någon information om det ökar möjligheten att hitta insiders. För att få en uppfattning av detta skulle man kunna sätta ut honeypots i en eller flera organisationers LAN. Det ska inte vara företag med allt för få användare, utan ju fler användare desto bättre. Det är naturligtvis också viktigt att inga användare förutom säkerhetspersonal får veta att det finns honeypots i nätet.

För att få en mer heltäckande undersökning så jämförs honeypoten med flera olika nätverksbaserade IDS som använder sig av olika metoder. Det kan ju vara så att vissa typer av IDS upptäcker ett intrång som honeypoten upptäcker medan andra typer av IDS missar detta intrång.

Om en potentiell angripare upptäcker honeypoten vid t.ex. portskanning av LAN, men som inte går vidare och försöker knäcka honeypoten så visar det kanske bara att någon är nyfiken men som egentligen inte har något ont uppsåt. Om det däremot är någon som accessar honeypoten och försöker knäcka den så har ett intrångsförsök upptäckts. Frågan är då om någon av de nätverksbaserade IDS har upptäckt intrånget eller inte.

Genom att föra statistik på alla intrång, som upptäckts och som är riktade mot honeypoten, kan man se om honeypoten lyckats upptäcka flest intrång. Har honeypoten upptäckt flest intrång så kan man dra den slutsatsen att honeypoten faktiskt ökar säkerheten i detta fall och att det är troligt att den generellt ökar säkerheten. Om det däremot är en eller flera IDS som tillsammans upptäcker lika många eller fler intrångsförsök mot honeypoten än vad honeypoten själv har upptäckt, så vet vi inte om säkerheten ökar vid användning av honeypot. Det beror på att vi inte vet om honeypoten fyllt sitt syfte och lockat till sig angripare, bort från de riktiga maskinerna. Har honeypoten fyllt sitt syfte, så har naturligtvis säkerheten ökat, men det är som sagt inget vi vet.

Vilken typ av honeypot ska då användas? Ska den efterlikna en enhet som redan finns i LAN eller ska det var en ny enhet?

Om honeypoten är en avbild av något som redan finns i det specifika företagets LAN finns det en risk att ärliga användare av misstag accessar honeypoten. Därför får man gå igenom loggarna noga så att sådana fall inte räknas som intrångsförsök, något som inte borde vara alltför komplicerat eftersom en ärlig användare inte försöker dölja sina spår.

Om det inte finns en FTP-server i LAN finns det en risk att undersökningen blir missvisande om honeypoten efterliknar en FTP-server. Man kanske då upptäcker intrång som aldrig hade initierats annars. Istället för att visa om säkerheten ökar vid användning av honeypot har man visat att det finns användare som har lockats att bryta mot säkerhetspolicyn och därmed blivit insiders. Det är kanske intressant att få veta men är inte syftet med undersökningen.

Därför anser vi att honeypoten helst bör efterlikna en enhet som redan finns i LAN vid en sådan undersökning.

9.3. Test i sluten miljö

Ett tredje alternativ kunde vara att utföra experiment liknande de som Chalmers utförde som vi nämnde i kapitel 2. Deltagarnas uppgift skulle kunna vara: Gör ett intrång i en maskin! Deltagarna skulle naturligtvis inte få veta att det var ett vilseledande system de angrep. Fördelar med ett sådant test är: Att deltagarna gör noggranna anteckningar på varje steg de tar som vi sedan kan analysera och dra slutsatser ifrån, vi vet kunskapsnivån på inkräktarna och vi missar inga intrång. Nackdelar är att det bara visar om säkerheten ökar för det specifika systemet och inte generellt.

10. Resultat och slutsatser

I detta kapitel redovisas de slutsatser som vi har dragit av vårt examensarbete. Syftet med detta examensarbete var att definiera vad vilseledande system är för något och att svara på vår problemformulering: Kan säkerheten ökas med hjälp av vilseledande system och i så fall hur?

En slutsats vi har dragit är att med kunnig personal och med ett väl genomtänkt vilseledande system så kan säkerheten ökas. Det är dock inte trivialt att sätta upp ett sådant system och ett felkonstruerat system kan istället leda till sämre säkerhet.

Vi har kommit fram till att genom att använda sig av ett honeynet så kan säkerheten ökas på lång sikt, eftersom vi har möjlighet att analysera knäckares beteende. Det ökar även skyddet i realtid på så sätt att angripare förbrukar tid på lockbeten. En förutsättning är dock att angripare verkligen lockas till honeynet, detta är något som vi anser går att utföra.

Genom att testa vilseledande system på sådant vis som vi gick igenom i kapitel 9 går det att få en indikation på om vilseledande system ökar säkerheten.

En honeypot kan bara upptäcka de intrångsförsök som är riktad mot den. Detta måste man vara medveten om! Bara för att det finns en eller flera honeypotar i nätverket så hindrar det inte en knäckare från att attackera en riktig server i stället för en falsk. Vår slutsats är därför: i första hand bör man säkra upp system med traditionella skydd som brandväggar, IDS, och kanske intrångstoleranta system etc. innan man använder sig av vilseledande system.

Ett välfungerande vilseledande system som använder sig av både ironboxar och honeypots, kräver personal som är anställd för att underhålla systemet dygnet runt, d.v.s. det passar mest för större organisationer. Normala användare i systemet påverkas endast av ett vilseledande system om de av misstag triggar felaktiga tjänster.

Vi har också undersökt vilka tekniker det finns för att vilseleda samt vilka fördelar och nackdelar det finns med dessa tekniker. Vi har funnit att det finns flera olika tekniker för att vilseleda inkräktare. Det finns fördelar och nackdelar med dem alla. Vi anser dock att tekniken med lockbeten är grundstommen i de vilseledande systemen eftersom honeypots är den vanligaste och som vi ser det, den naturligaste komponenten i ett vilseledande system.

En målsättning vi hade när vi startad vårt examensarbete var att sätta upp ett färdigt system som använder sig av en eller flera av dessa tekniker. Det visade sig tyvärr att det var svårt att få tag på ett vilseledande system. En orsak var att Ericsson inte hade några avtal med dessa företag vilket innebar att det skulle bli för långa leveranstider.

Eftersom det endast finns ett fåtal företag som säljer dessa produkter så har vi dragit den slutsatsen att marknaden ännu är omogen för vilseledande system.

Vi ser att en naturlig fortsättning på detta examensarbete skulle vara att köpa in någon färdig produkt och sätta upp ett vilseledande system, dels för att få kunskap om hur det är att administrera, installera samt för att testa om det ökar säkerheten. En undersökning av hur lätt/svårt det är att föra över en angripare till en ironbox utan att angriparen blir misstänksam skulle också vara intressant. Ett vilseledande system som använder sig av tekniken med rörliga mål kunde också vara intressant att undersöka.

11. Etiska synpunkter på vilseledning

En av anledningarna till att säkerhetsansvariga har avhållit sig från att använda sig av förvillande system kan vara att det anses omoraliskt att vilseleda och ljuga, även om man talar osanning för någon som ljuger för att komma in i systemet. En inkräktare har aldrig tvekat att fara med osanning för att komma in i ett system. Med tanke på detta och om man inte vilseleder eller lämnar information som kan skada inkräktaren eller dess utrustning bör man kunna acceptera att använda sig av att ljuga och vilseleda. Olika människor har olika uppfattningar om vad som är acceptabelt att ljuga om. Vissa kan t.ex. tänka sig en vit lögn för att komma ur en knipa, andra kan tänka sig att inte lämna alla uppgifter till skattemyndigheten men kan inte tänka sig att ta en godisbit för fem kronor i en butik trots att skattefusket kan vara värt hundratals kronor.

I ett vilseledande system finns det möjlighet att gömma en trojansk häst någonstans i systemet. Om vi antar att den trojanska hästen är placerad på ett sådant sätt i det vilseledande systemet att det endast är inkräktare, som klarar av att forcera en viss nivå av säkerhetsfunktioner, som ovetande kan komma åt och ladda ner den. I ett sådant här fall har inkräktaren gjort en aktiv handling för att komma in i systemet. Om den trojanska hästen förstör något på inkräktarens dator t.ex. raderar hårddisken, så anser vi att det är tveksamt om detta är försvarbart eftersom vi gör en direkt handling mot inkräktaren och på detta vis förstör information för inkräktaren.

Om den trojanska hästen endast skickar ut information om vad inkräktaren gör för något kan detta verka ok. Detta beteende skulle t.ex. kunna jämföras med vissa säkerhetssystem som går att installera på sin bil. Dessa säkerhetssystem har en sändare som aktiveras när ägaren av bilen har upptäckt att bilen är stulen. Bilen går sedan att pejlas in med olika radiomaster och kan på detta vis lokaliseras.

Att aktivt samla information, med en trojansk häst, om en angripare kan uppfattas som integritetskränkande eftersom det är svårt att selektera den information som sänds ut av det trojanska programmet. Det borde däremot inte finnas några integritetsproblem om vi aktivt spårar attackeraren bakåt efter den väg inkräktaren har tagit sig fram till oss, eftersom en spårning av denna typ inte genererar någon information som inkräktaren inte vet om att den sänds ut från hans/hennes dator.

Det är viktigt att de motåtgärder som görs mot en inkräktare inte uppfattas som kränkande eftersom det kan medföra en risk att attackerna eskalerar i styrka. Det är alltså viktigt att undvika en upptrappning av det elektroniska våldet.

En skanningattack är enligt svensk lag inte förbjudet, men det är en handling som är en förberedelse för attack. Vi bortser i denna diskussion från sökmotorer t.ex. Altavista, GoTo, Google, m.m. Dessa sökmotorer söker efter information på olika portar vilket kan uppfattas som ett skanningförsök. Eftersom ett skanningförsök

med hög sannolikhet görs för att förbereda en attack kan det vara försvarsbart att göra oss osynliga för attackerarens IP-adress. Det finns dock vissa frågetecken om detta. Det ena är ifall attackeraren använder sig av falsk avsändaradress. I detta fall drabbas en tredje part som inte kan t.ex. läsa vår hemsida. Det andra är att vi har svårt att veta om det är en sökmotor eller inte som kontaktar oss.

I en del av den litteratur vi har studerat kommer frågan upp om vad lagen säger om använda sig av vilseledning för att få bevis mot en inkräktare. Vi uppfattar den svenska lagen på ett sådant sätt att informationen som fås av ett vilseledande system kan användas mot en eventuell inkräktare i en rättegång. Vi rekommenderar dock berörda personer att göra en egen bedömning med hjälp av en datajurist.

Det finns en risk att inkräktare som har lagt ner stor ansträngning och arbetsresurser på att penetrera ett system blir väldigt besvikna om de upptäcker att de har blivit lurade. En tänkbar reaktion är att de ökar sina ansträngningar för att gör livet surt för de som har vilselett dem.

12. Referenser och litteraturförteckning

[AMO 99] Amoroso E, *Intrusion Detection An Introduction …*, Intrusion.Net Books,1999

[AND 80] Andersson J, *Computer Security Threat Monitoring and Surveillance,* Häfte från instutitionen för datasäkerhet på Chalmers

[BAC 01] Bace R & Mell P, *NIST Special Publication on Intrusion Detection Systems,* URL:csrc.nist.gov/publications/drafts/idsdraft.pdf, 14 Mars 2001

[BRI 01] Brighton University, URL://burks.bton.ac.uk/burks/foldoc/, 26 Mars 2001

[CHA 95] Chapman B och Zwicky E, Building Internet Firewalls, O´Reilly & Associates, Inc.

[CHE 91] Cheswick B, *An Evening with Berferd In Which a Cracker Is Lured, Endured, and Studied,* URL:www.bpfh.net/simes/computing/Bill_Cheswick_berferd.ps,15 Februari 2001

[COH 98] Cohen F, *A note on the Role of Deception in Information Protection,* URL:www.all.net/journal/ntb/deception.html, 2 Februari 2001

[DAH 01] Dahlin N, *Mobil IP har problem*, Datateknik 3.0, Nr 2, s 27, Februari 2001.

[FOR 00] Forristal J, *LURING KILLER BEES WITH HONEY*, Nework Computing, vol 11, issue 16, 21 Augusti 2000, s 100-103

[FRA 00] Fratto M, *Honeypots: Security Means Sticky Business*, URL:www.networkcomputing.com/columnists/1106colmike.html,10 April 2001

[HAR 99] Harris B & Hunt R, *TCP/IP security threats and attack methods*, Computer Communications, Nr 22, s 885-897, 1999.

[HON 00] *The Honeynet Project,* URL:project.honeynet.org, 21 Mars 2001

[HON 01] *Know Your Enemy: Honeynets,* URL:project.honeynet.org/papers/honeynet, 24 April 2001

[HON2 00] *To Build A Honeypot,* URL:project.honeynet.org/papers/honeypot, 12 Februari 2001

[ITL 99] *Acquiring and Deploying Intrusion Detection Systems,* URL: www.nist.gov/itl/lab/bulletns/nov99.htm, 13 Februari 2001

[ITS 91] *Information Technology Security Evaluation Criteria,* URL:www.itsec.gov.uk/docs/pdfs/formal/ITSEC.PDF, 28 Februari 2001

[JON 95] Jonsson E, *A Quantitive Approachh to Computer Security from a Dependability Perspective,* Chalmers Reproservice, 1995

[KOC 92] Kock J, *Operation Fortitude: The Backbone of Deception,* URL: www- cgsc.army.mil./milrev/index.htm , 2 Februari 2001

[LIN 99] Lindqvist U, *On the Fundamentals of Analysis and Detection of Computer Misuse*, Chalmers Reproservice, 1999.

[LÖV 01] Lövgren M, *Avhoppad KGB-man bakom innovativ säkerhetsteknik,* URL: computersweden.idg.se/text/010526-cs1, 28 maj 2001

[MOR 00] Moran D, *Using deception,* Information Security Bulletin, vol 5, issue 8, s27-34, Oktober 2000.

[PGP 01] PGP, *Cyber Cop Sting,* URL:www.pgp.com/products/freeware/agreement-win.asp, 16 April 2001

[REC 01] Recourse, *ManHunt*, URL:www.recourse.com/news/press/releases/r102400.html, 22 Mars 2001.

[SCG 00] Schlossberg B, *Using deception,* Information Security Bulletin, vol 5, issue 8, s13-26, Oktober 2000.

[SCH 00] Schultz E, *Using deception,* Information Security Bulletin, vol 5, issue 8, s3, Oktober 2000.

[SCHN 01] Schneider E, *Measurements of System Security*, URL:www.acsac.org/measurement/position-papers/schneider.pdf, 14 Maj 2001

[SLA 95] Slatalla M & Quittner J, *Hacker –berättelsen om gänget som knäckte koden*, Nordsteds Förlag AB, 1995.

[SON 99] Song D, *Nidsbench-A Network Intrusion Detection System Test Suite* URL: www.monkey.org/~dugsong/talks/nidsbench-slides.ps.gz , 20 Maj 2001.

[SPA 01] Spafford E & Zamboni D, *Intrusion detection using autonomous agents*, URL: www.elsevier.nl/gej-ng/10/15/22/49/30/25/article.pdf, 4 April 2001

[SPE 01] URL:www.specter.com

[STA 00] Stallings W, *Network security essentials*, Prentice-Hall, New Jersey, 2000

[STI 95] Stinson D, *Cryptography Theory and Practice*, CRC Press LLC, 1995

[SUN 96] Sundaram A, *An Introduduction to Intrusion Detection*, URL:www.acm.org/crossroads/xrds2-4/intrus.html, 28 Februari 2001

[THO 00] Thoresson A, *Ingen brandvägg helt säker*, Ny Teknik, 15 November 2000

[ZON 01] *Password Protection*, URL:www.zonelabs.com/pressroom/news/newsletters/February2001/article_tax_AdSub.html, 23 Februari 2001

13. Ordlista och index

Mycket material i detta avsnitt har hämtats från Brighton University
http://burks.bton.ac.uk/burks/foldoc/
och från Svenska datatermgruppen http://www.nada.kth.se/dataterm/

Agent:
En intrångsdetekteringskomponent som periodiskt samlar data från informationskällan. Ibland analyserar och ordnar den data. En agent som bara samlar data är detsamma som en sensor.

Analyserare:
Intrångsdetekteringskomponent som analyserar data, som samlats in av sensorn, och som letar efter tecken på obehörig och oönskad aktivitet eller efter händelser som kan vara av intresse för säkerhetsadministratörer.

Audit log:
En logg av systemhändelser och aktiviteter, som genereras av operativsystemet.

Broadcast:
Nätverkstrafik som skickas från en nod till alla andra noder i nätverket.

Cracker:
se knäckare.

Extern angripare:
En angripare som inte har tillgång till någon kunskap om företagets interna struktur.

Hackare:
Person som är mycket skicklig i att göra egna program eller ändra i andras program, ofta med inslag av perfektionism.

Honeynet:
Ett honeynet är ett nät eller en del av ett nät som använder sig av vilseledning och där inga produktiva maskiner finns och vars syfte är att uppehålla och studera inkräktare.

Honeypot:
En honeypots uppgift är att efterlikna systemets servrar, medan man larmar och loggar inkräktares aktiviteter.

Host:
Värd eller värddator.

Informationskälla:
Den obearbetade informationen som ett intrångsdetekteringssystem använder för att detektera obehörig eller oönskad aktivitet. Typiska informationskällor är nätverkspaket, operativsystemsloggar, applikations loggar och system loggar.

Insider:
En person som har eller har haft tillgång till företagets interna struktur.

Intern angripare:
En intern angripare är t.ex. en anställd (insider) som missbrukar eller överskrider sina befogenheter.

Intrångsdetektering:
Är en process som innefattar att övervaka händelser som sker i ett datorsystem eller ett nätverkssystem, analysera händelserna efter tecken på intrång och reagera med motåtgärder mot eventuella intrång.

Ironbox:
En Ironbox är en speciell omgivning som satts upp för att uppehålla en knäckare tillräckligt länge för att kunna bli spårad. Den kan innehålla en modifierad shell som förhindrar oförutsägbara steg av knäckaren och lockande filer för att få knäckaren att stanna kvar.

Jail:
Se Ironbox.

Knäckare:
Person som försöker forcera datorsystems säkerhetsspärrar i syfte att sabotera programvara eller att komma åt, utnyttja eller eventuellt förändra skyddad information.

Outsider:
En person som inte har någon kunskap om företagets interna struktur.

Padded cell:
Ett program som begränsar en lokal användare till en noggrann begränsad delmängd av värdsystemets möjligheter (t ex ”rsh” på USG Unix). Där placerar man en lokal användare (luser) så att de inte kan förstöra något. Notera att detta är skillnad från ironbox eftersom den är offentlig (öppen) och inte riktad till att upprätthålla säkerhet lika mycket som att skydda andra från konsekvenser från en lokal användares naivitet.

Packetfiltreringssystem:
Är en brandväggs typ som vidarebefordrar paket mellan datorer i det lokala nätverket och externa nätverk.

Penetrationsattacker:
Penetrationsattacker involverar obehörigt förvärv eller ändring av systemets privilegier, resurser eller data.

Proxyserver:
Server som utgör gränsen mellan två datornät, t.ex. mellan ett internt och ett externt datornät, och som vidarebefordrar trafiken dem emellan

Router:
Dator som väljer väg för och vidarebefordrar data i ett datornät

Screaning router:
Se paketfiltreringssystem.

Sensor:
En intrångsdetekteringskomponent som periodiskt samlar data från informationskällan. Även känd som en agent (I många befintliga IDS är sensorn och analyseraren delar i samma komponent).

Skanning:
Skanning innebär att inkräktaren söker igenom nätverket eller systemet i syfte att kartlägga det.

Social ingenjörskonst:
Går ut på att ljuga sig till information för att kunna penetrera ett system.

Spoofad attack:
En attack med falsk avsändaradress

Systemlogg:
En logg av systemhändelser och systemaktiviteter som genereras av en systemprocess. Abstraktionsnivån är högre i en systemlogg än i en audit log.

14. Förkortningar

CIA	Confidentiality Integrity Availability
DDoS	Distributed Denial of Service
DES	Data Encryption Standard
DMZ	Demilitarised Zone
DOS	Disk Operating System
DoS	Denial of Service
FTP	File Transfer Protocol
HIDS	Hostbased Intrusion Detection System
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
IRC	Internet Relay Chat
ITSEC	Information Technology Security Evaluation Creteria
ISP	Internet Service Provider
LAN	Local Area Network
NIDS	Networkbased Intrusion Detection System
RSA	Rivest, Shamir och Adelman. Uppfinnarna till "public-key" kryptosystem, 1977
TCP	Transport Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network