NTNU

Kunnskap for en bedre verden

IDS og IPS og VPN

Helge Hafting, Institutt for datateknologi og informatikk (IDI), NTNU
Lærestoffet er utviklet for faget «INFT1007 Datakommunikasjon»

Resymé: I denne leksjonen ser vi på systemer for å oppdage og avverge angrep over nett. (Intrusion detection) Deretter ser vi på virtuelle private nett, en mekanisme for sikker kommunikasjon via usikre forbindelser.

Innhold

12 IDS og IPS og VPN

12.1 IDS og IPS

IDS[1] er systemer som oppdager datainnbrudd og varsler oss på noe vis.

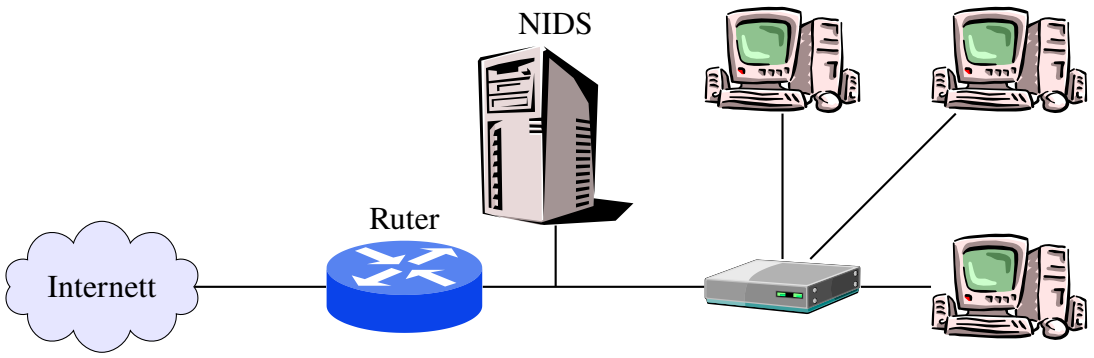
IPS[2] er systemer som i tillegg til å oppdage innbrudd, også gjør noe aktivt for å blokkere innbruddet.

12.1.1 NIDS

Network intrusion detection system oppdager innbrudd og innbruddsforsøk i nettverket. Et NIDS består av programvare som installeres på en maskin et sted i nettverket. NIDS-programvaren settes opp til å fange opp og analysere all trafikk som passerer nettverkskortet i maskinen. For at dette skal ha noen verdi er man avhengig av at den interessante

[1] Intrusion detection system
[2] Intrusion prevention system

trafikken passerer NIDS-maskinens nettverkskort. Det vanligste scenarioet er at man vil overvåke trafikk som går på tvers av ulike nettverk, og man vil da plassere NIDS-maskinen i tilknytning til rutere i nettverket.



Figur 12.1: NIDS som overvåker all trafikk på vei inn og ut av organisasjonens nettverk

I figur 12.1 er NIDS-maskinen koblet til kabelen mellom grenseruteren og svitsjen i et lite nettverk ved hjelp av en såkalt «passive tap[3]». Dette er en enkel innretning som gjør at NIDS-maskinen mottar all trafikken på kabelen som tappes.

Alternativt kan NIDS-maskinen være tilknyttet en svitsj som dupliserer trafikk fra andre porter ut på NIDS-maskinens svitsjport. Denne metoden har et problem. En svitsj har gjerne samme båndbredde på alle portene. Hvis det går mye trafikk gjennom svitsjen, er det simpelthen ikke nok båndbredde på én port til å duplisere alle de andre. Dermed går NIDS glipp av informasjon når belastningen er høy. Planlagte innbrudd forsøkes gjerne skjult ved å kjøre et belastningsangrep samtidig. Svitsjen kan forsåvidt begrense den totale båndbredden slik at NIDS får med seg alt, men det kan fort bli et problem i seg selv.

Et tredje alternativ er å kjøre et NIDS på selve ruteren, noe som til en viss grad er populært med rutere basert på vanlige Pcer med Linux, FreeBSD eller tilsvarende.

All trafikk som passerer NIDS-maskinen vil nå bli analysert i henhold til regler i programvaren. Programvaren vil for eksempel kunne kjenne igjen port- og ping-scanning, forsøk på misbruk av kjente sikkerhetshull og lignende. Regeldatabasen kan inneholde regler fra leverandøren av programvaren, og du kan legge inn dine egne. En del NIDS-produkter lærer seg også trafikkmønstrene i nettverket og kan detektere unormaliteter i forhold til det den har lært. Et eksempel kan være et tilfelle hvor det plutselig blir mye trafikk mot TCP-port 80 på en maskin hvor dette aldri har vært tilfelle før.

[3] På `http://en.wikipedia.org/wiki/Network_tap` finner du informasjon om hvordan en passive tap fungerer.

Hva kan et IDS gjøre?

Når NIDS-programvaren har identifisert at et gitt stykke trafikk (en enkelt pakke eller fler pakker sett i sammenheng) stemmer overens med en av innslagene i regeldatabasen kan programvaren varsle om dette på ulike måter.

Minimum er å logge hendelsen. Om den skulle vise seg å være del av et større angrep, kan vi finne detaljene i loggen senere.

Programvaren kan også varsle oss. Den kan f.eks. sende e-post og sms til de som er ansvarlige for nettverket. For ikke å bli oversvømt med henvendelser fra NIDS-programvaren, er det viktig å kunne skille mellom trivielle og viktige hendelser. Dette krever litt innkjøring med lokale tilpasninger, samt at NIDS-programvaren selv som regel skiller i alvorlighetsgrad på de ulike reglene.

Om en overvåker store systemer, går det an å ha sanntidsdisplayer som viser hva som skjer rundt om kring. For alvorlige hendelser som krever rask reaksjon, kan man ha alarmer. Dette er neppe aktuelt i mindre bedrifter, men nyttig for cyberforsvaret, PST og lignende.

12.1.2 IPS for nettverk

Det er mulig å få et NIDS til iverksette handlinger på egen hånd ut fra hendelser som registreres. Dermed blir NIDS til et IPS. Dette foregår ved at programvaren sender beskjed til brannmurer, rutere eller svitsjer om å legge til aksessregler for stanse det IPS-programvaren mener er et problem.

Vi kan for eksempel tenke oss at en maskin driver ping-scanning av nettverket. Når IPS-programvaren registrerer denne hendelsen som et treff mot et innslag i sin regeldatabase kan programmet sende beskjed til en ruter om at den må stoppe all trafikk til og fra IP-adressen som bedriver scanningen. Etter en time kan IPS-programmet be ruteren om å fjerne aksessfilteret. Regelen kan f.eks. være «ping mot flere av våre adresser fra samme avsender innenfor et kort tidsrom»

Hvis IPS oppdager tvilsom aktivitet fra en maskin på det interne nettverket, kan IPS gi beskjed til svitsjen om å koble fra den aktuelle porten. Slik kan vi isolere en maskin som sprer virus eller skaper problemer ved å kjøre uønskede dhcp-tjenester.

Da et IPS er avhengig av å kommunisere med annen nettelektronikk for å fungere er dette lettest å gjennomføre om man kjøper all involvert nettelektronikk fra samme leverandør. Cisco har for eksempel en serie integrerte NIDS-enheter, Cisco IPS 4200 Series Sensors[4], som er i stand til å kommunisere med annet Cisco utstyr.

[4] Cisco IDS 4200 Sensors, se `http://www.cisco.com/en/US/products/hw/vpndevc/ps4077/`

Faremomenter med IPS

Ukritisk bruk av IPS er farlig. Et IPS som blokkerer IP-adresser kan lures med falske fra-adresser, dermed brukes firmaets naive IPS som en del av et dos[5]-angrep.

Et eksempel er å portscanne et nettverk, men med fra-adresser som tilhører firmaets tjenermaskiner. Hvis IPS blokkerer disse, får hverken interne eller eksterne klienter kontakt med tjenermaskinene. Det har også vært eksempler på IPS som har blokkert sin egen ruter, og dermed koblet hele firmaet fra nettet.

Nå kan vi unngå spoofing av firmaets egne adresser, siden disse ikke skal kunne være fra-adresser på pakker som kommer utenfra. Men i stedet kan angriperen benytte fra-adressene til populære eksterne tjenester. Slik kan et IPS lures til å blokkere en hvilken som helst ekstern tjeneste. Eller det kan lures til å stenge viktige kunder ute, noe vi ikke nødvendigvis merker selv.

Blokkering av adresser har mer for seg når vi oppdager en type angrep hvor angriper har behov for å motta svar. Da kan ikke angriper bruke falske adresser. Innloggingsforsøk er et slikt eksempel.

IPS kan heller ikke stoppe alle slags angrep. Hvis IPS f.eks. oppdager at nedlasting fra webtjeneren tusendobles, så kan det tyde på et DOS-angrep. Men det er ikke sikkert IPS kan vite hvilke forespørsler som er «ekte» og hvilke som bare sendes oss for å bruke opp båndbredden. Det er i hvert fall ikke aktuelt å blokkere alle forbindelser mot webtjeneren, for da har angriper oppnådd det de ville — de har fått oss av nettet.

12.2 Introduksjon til VPN

12.2.1 VPN for å gi fjerntilgang til bedriftens ressurser

De fleste organisasjoner tilbyr tjenester til sine brukere hvor adgang gis basert på at de befinner seg i organisasjonens nettverk. Dette innebærer at du må sitte i lokalnettverket for å kunne kontakte tjenestene. Dette være seg e-post, delte filområder, webbaserte tjenester og så videre. Dersom organisasjonen er knyttet til Internett kan man benytte seg av brannmurer for å hindre adgang til tjenestene for de som befinner seg utenfor lokalnettverket. Hvis organisasjonen har avdelinger som ligger geografisk spredt kan man leie dedikerte linjer for å knytte avdelingene sammen til ett stort nettverk.

Selv om leide linjer er blitt rimeligere med årene, blant annet som følge av inntoget av MPLS[6], er kostnadene mye høyere enn tilkobling og kommunikasjon via Internett. Så

[5] DOS: Denial Of Service

[6] MPLS — Multi Protocol Label Switching. Gir blant annet mulighet for å lage mange virtuelle dedikerte linjer over samme fysiske infrastruktur. For mer informasjon se `http://en.wikipedia.org/wiki/Multiprotocol_Label_Switching`

hvorfor ikke la tjenestene være tilgjengelige via Internett — hvorfor benytte brannmurer for å stoppe denne kommunikasjonen? De enkelte tjenestene krever jo uansett at brukerne må autentisere seg med brukernavn og passord — er ikke dette nok?

Man må vurdere dette ut i fra praktiske og sikkerhetsmessige hensyn. Ved å åpne en tjeneste for hele Internett gir dette en rekke muligheter for alle tilknyttet Internett. Alle kan utnytte eventuelle kjente sikkerhetshull eller feilkonfigurasjoner i tjenerprogramvaren. Dette være seg enten målrettet cracking eller virus/ormer/trojanere. Når trafikk passerer nettverk utenfor vår kontroll, har andre mulighet til å avlytte kommunikasjonen og potensielt få ut informasjon dersom trafikk sendes ukryptert. Gjennom avlytting, gjetting, brute force, keylogging eller andre metoder kan uvedkommende skaffe seg gyldige brukernavn og passord for å logge seg på tjenestene. Utenforstående kan også se på hvilke data som passerer gjennom nettet.

Med problemstillinger som disse i bakhodet, velger de fleste organisasjoner å begrense adgangen til sine tjenester. Det enkleste og sikreste er å hindre all adgang fra Internett, men det er ikke uvanlig å gi spesifikke IP-adresser/nett adgang til en eller flere tjenester i organisasjonens nettverk. Det kan diskuteres hvorvidt dette er heldig, men det er i alle fall bedre enn å gi adgang til tjenestene for hele Internett.

Løsningen med å gi adgang til enkelte IP-adresser/nett har flere svakheter. Sikkerhetsmessig er det ikke en god løsning, og for mange organisasjoner/tjenester er dette derfor uaktuelt. I tillegg er det i praksis vanskelig å vedlikeholde slike lister, dersom man har mange ulike tjenester og de som har behov for tjenestene befinner seg i mange ulike IP-nett. Mens et avdelingskontor kanskje har et fast IP-område, vil dette ikke være tilfelle for medarbeidere som vil jobbe hjemme eller er på reise.

En bedre løsning på problemstillingen med å tilby tjenester sikkert over Internett er å bygge et *Virtuelt Privat Nettverk (VPN)* basert på krypteringsteknologi.

For de fleste organisasjoner er det også slik at tjenestene som tilbys i de interne nettverkene kun er tilgjengelig med private IP-adresser, slik som 192.168.0.50 eller 10.0.3.201. Disse vil ikke være tilgjengelig over Internett med mindre vi allokerer offentlige IP-adresser til tjenestene eller skriver om adresser (NAT) til offentlige IP-adresser i brannmur eller ruter. For tjenester som kun skal være tilgjengelig for interne brukere vil det mest hensiktsmessige være å la tjenestene kun være tilgjengelig utenfra via VPN. Som vi skal se kan tjenester tilgjengelig på private IP-adresser altså aksesseres dersom brukeren er tilknyttet med VPN.

12.2.2 VPN for å skjule hvor man er

Søker man litt på nett, finner man mange VPN-tilbydere. Bruker man et slikt VPN for å koble seg opp mot servere på nettet, vil serveren tro at vi kontakter den fra vpn-

tilbyderens nettverk heller enn fra vårt eget nettverk. Serveren får ikke vite hvilken IP-adresse du har, den ser bare VPN-tilbyderens adresse.

Dette kan brukes til å skjule hvor man er i verden. Tjenestetilbydere har til tider ulike priser i ulike land. Dette fordi prisnivået er ulikt rundt om kring i verden, og de prøver å maksimere fortjeneste. Det er vel kjent at Norge er et høykostland, så med en norsk IP-adresse, risikerer man høye priser. Ved å bruke et VPN i et lavkostland, kan man dermed få billigere Netflix og lignende tjenester.

Konflikter mellom land, kan være en annen grunn til å skjule hvor man er.

Vær oppmerksom på at selv om et VPN skjuler deg for omverdenen, så vet VPN-tilbyderen utmerket godt hvor du er, det begrenser hvor hemmelig dette blir.

12.3 Bakgrunn

Vi kan definere hva et VPN er på to ulike måter. Den ene er løsningsorientert og den andre er teknologisk. Dersom vi tenker løsningsorientert, er et VPN noe som gir oss sikker tilgang til tjenester over et åpent nettverk. Den teknologiske tolkningen av begrepet er et privat nettverk vi lager oss i programvare over en fysisk infrastruktur. Med privat menes at kun de som skal ha adgang har adgang til nettverket. Dette innebærer at uvedkommende ikke kan bruke vårt virtuelle nettverk, og heller ikke har innsyn i trafikken i nettverket. Dette er et «ekte» VPN — det er med andre ord et virtuelt privat nettverk. Løsningsorientert produkter kan sies å gi oss de samme gevinstene som et «ekte» VPN.

Disse to måtene å tenke på gir seg utslag i ulike metoder for å designe VPN-løsninger. Et VPN laget i henhold til den teknologiske tankegangen er bundet til å lage et faktisk virtuelt nettverk. VPN-produkter laget i henhold til tanken om å tilby sikker tilgang til tjenester over et åpent nettverk kan potensielt gjøre dette på mange måter. Som vi skal se om litt, har begge metodene sine fordeler og ulemper.

12.3.1 Hvorfor VPN

Før vi ser nærmere på hvordan et VPN fungerer, kan vi summere opp de viktigste årsakene til at VPN-løsninger tas i bruk:

- Rimelig og fleksibelt alternativ til dedikerte linjer for å knytte sammen geografisk spredte deler av organisasjonen
- I praksis det eneste gode alternativet for tilgang til interne tjenester for hjemmekontor, medarbeidere på reise og lignende.
- Gir adgang til tjenester som kun er tilgjengelig på private (lokale) IP-adresser. Eksempelvis 10.0.0.50, 192.168.2.12

- Transport av andre lag-3 protokoller enn IP over en IP-infrastruktur. Eksempelvis IPX og Appletalk. For de fleste er dette en mindre aktuell problemstilling.
- For den enkelte: Sikre seg mot andres innsyn når en befinner seg i fremmede nettverk og ikke ønsker å bli avlyttet.

12.3.2 Begreper

Vi skiller mellom to former for VPN, *klient-VPN* og *nettverk-nettverk-VPN*. Klient-VPN vil si å knytte enkelt-PC-er til et internt nettverk. Det vil altså si tilknytning fra et hjemmekontor eller når vi er ute og reiser. Klienten (brukerens PC) kobler seg til en nettverkskomponent som kalles *VPN-konsentrator*. En VPN-konsentrator er enten en selvstendig boks (datamaskin) eller funksjonalitet i brannmur eller router.

Nettverk-nettverk-VPN brukes for å knytte sammen to i utgangspunktet fjerntliggende nettverk. Dette gjøres som regel med programvare i brannmurer eller rutere som binder de to nettverkene sammen.

12.3.3 VPN og kompleksitet

Virtuelle private nettverk er et av de mer komplekse områdene innenfor nettverksteknologi. Dessverre må man ofte legge relativt mye ressurser i å etablere fungerende VPN-løsninger. Dette har flere årsaker:

- Mange ulike teknologier.

  For å etablere et VPN kan man velge mellom mange ulike produkter.

- Støtte for ulike teknologier avhengig av hvilke produkter du har.

  Hvilke VPN-løsninger du kan ta i bruk avhenger av hvilke produkter du har. Dette være seg nettelektronikk (rutere, brannmurer, VPN-konsentratorer, …) og operativsystem på vanlige datamaskiner. Mange organisasjoner opererer med flere VPN-teknologier for ulikt bruk. Et eksempel på dette er at mange bruker en VPN-teknologi for å etablere nettverk-nettverk-VPN, mens klient-VPN løses med annen programvare.

- VPN-teknologien er under utvikling.

  Selv om VPN-løsninger har eksistert relativt lenge (utviklet seg på 90-tallet), har mange av løsningene elementer som er under utvikling.

- En spade er ikke nødvendigvis en spade.

  I motsetning til f.eks HTTP og SMTP hvor implementasjoner fra ulike leverandører snakker fint sammen, er det motsatte ofte tilfelle med VPN-protokoller. Dette gjelder for eksempel en av de mest brukte VPN-protokollene, IPSec.

- VPN og VPN-teknologiene er i seg selv komplekse.

  I likhet med krypto-teknologi generelt er VPN av natur relativt komplekst. Enkelte VPN-løsninger er svært generaliserte, da spesielt IPSec-baserte, og det er derfor mye å sette seg inn i.

12.4 VPN-løsninger

Når man skal konstruere et VPN kan man velge mellom mange ulike protokoller. For enkelte av protokollene varierer implementasjonen fra de ulike produsentene så mye at det kan være vanskelig å få dem til å fungere sammen. I enkelte tilfeller er ulikhetene så store at det rett og slett ikke er mulig.

De ulike produktene kan grovt deles inn i følgende fire kategorier:

- IPSec: IPSec er et sett sikkerhetsprotokoller som mange VPN-løsninger baserer seg på. De ulike løsningene tilbyr ulik funksjonalitet og er i mange tilfeller ikke kompatible med hverandre.
- Tunneler med virtuelle nettverkskort og SSL/TLS: Disse løsningene bruker etablert teknologi, slik som virtuelle nettverkskort og SSL/TLS, på en ny måte for å etablere VPN.
- Andre tunneleringsløsninger: Mange klassiske VPN-løsninger, som PPTP og L2TP / IPSec i Windows, baserer seg på ulike former for tunneler.
- Webbasert VPN: Ikke et VPN i teknisk forstand, men baserer seg på bruk av kryptert HTTP (HTTPS) for kommunikasjon med en webtjener som tilbyr proxytjeneste mot tjenestene vi ønsker å nå.

De tre øverste kategoriene representerer «ekte» VPN, hvilket vil si at det konstrueres en tunnel mellom to punkter i nettverket. Dette innebærer at de to kommuniserende partene etablerer en «forbindelse» — en tunnel, hvor trafikk kan sendes som om de to partene var knyttet direkte sammen med en nettverkskabel. Vi ser nærmere på dette når vi tar for oss de ulike teknologiene.

12.4.1 IPSec

IPSec (av IP Security) er et sett protokoller utviklet av IETF[7] for å forestå sikker kommunikasjon på IP-laget i TCP/IP. IPSec er i utgangspunktet utviklet for å være en del

[7] IETF — Internet Engineering Task Force. Åpen, internasjonal organisasjon som arbeider med utvikling av Internett-standarder. Se `http://www.ietf.org/`

av IPv6, men er også mye brukt sammen med IPv4. IPSec er av de mest utbredte teknologiene for å lage VPN. Formålet med IPSec er å tilby en generisk mekanisme for å transportere IP-pakker på en sikker måte over en åpen IP-infrastruktur (f.eks. Internett).

IPSec tilbyr blant annet følgende funksjonalitet til mottakeren av en IP-pakke:

Autentisering Verifisering av at pakken kommer fra riktig avsender.

Integritet Forsikring om at pakken ikke er blitt endret under transport.

Konfidensialitet Kryptering av innholdet.

IPSec fungerer ved at to maskiner som skal kommunisere sammen konfigureres med hvilke egenskaper de ønsker å ha på ulik trafikk som skal passere dem i mellom. Dette betyr at du kan velge ulik type beskyttelse for ulik type trafikk (ulike kombinasjoner av mottaker-IP-adresser og UDP/TCP-porter). I konfigurasjonen ligger også de krypteringsnøklene som skal benyttes. Konfigurasjonen kan gjøres manuelt, eller den kan delvis automatiseres ved bruk av *The Internet Key Exchange Protocol (IKE)*. IKE kan hjelpe oss med utveksling av krypteringsnøkler m.m.

I figur 12.2 ser vi hvordan bruk av IPSec påvirker IP-pakkene som sendes ut av en maskin. Man kan velge mellom transport- og tunnel-modus, samt at pakkene kan beskyttes enten av *Authentication Header (AH)* eller *Encapsulating Security Payload (ESP)*. Ved bruk av AH kan man autentisere avsenderen og oppnå integritet gjennom at det lages en sjekksum av innholdet i pakken. Ved bruk av ESP krypteres i tillegg nyttelasten i pakken.

IP-pakke

IP-hode	Nyttelast

IPSec transport-modus

IP-hode	IPSec-hode	Nyttelast (kryptert hvis ESP)

IPSec tunnel-modus

Nytt IP-hode	IPSec-hode	Nyttelast (kryptert hvis ESP):	
		Gammelt IP-hode	Nyttelast

Figur 12.2: Konseptuell fremstilling av hvordan IPSec påvirker IP-pakkene

Transport-modus brukes for kommunikasjon direkte mellom to kommuniserende parter. Tunnel-modus brukes mellom rutere/VPN-konsentratorer for å knytte sammen nettverk, eller mellom den ene kommuniserende parten og en ruter/VPN-konsentrator. Sistnevnte er typisk tilfelle for hjemmekontorer eller medarbeidere på reise. Når vi snakker om IPSec og VPN er det tunnel-modus vi tenker på.

Vi kan tenke oss at en organisasjon har to kontorer, A og B, som ligger geografisk spredt fra hverandre. På hvert av kontorene står det en ruter som knytter lokalnettverket til Internett. Mellom de to ruterne er det etablert en IPSec-tunnel hvor ESP benyttes. Vi kan så tenke oss at en PC på kontor A sender en IP-pakke til en PC på kontor B. Når denne pakken kommer til ruter A vil hele pakken krypteres og legges inn i en ny IP-pakke. Mottaker-adressen i den nye IP-pakken er IP-adressen ruter B har mot Internett. Når pakken kommer frem til ruter B dekrypteres innholdet og den originale IP-pakken sendes ut til den endelige mottakeren.

Den endelige løsningen?

IPSec har blitt utpekt som «den endelige løsningen» på sikkerhet i datakommunikasjon, men så har ikke skjedd. Dette skyldes flere forhold:

- Kompleksitet: IPSec er komplekst, både for leverandørene som skal implementere protokollene og for de som skal ta teknologien i bruk
- Kompatibilitet: IPSec er en generisk løsning, så den som skal implementere IPSec står ovenfor mange valg, noe som gjør at løsninger fra ulike leverandører i stor grad blir inkompatible
- Uvanlige protokoller og porter: IPSec bruker IP-protokoll 50 (ESP), IP-protokoll 51 (AH) og UDP-port 500 (IKE). AH-protokollen er i utgangspunktet ikke mulig å NAT-e. Disse forholdene skaper ofte problemer med at trafikken ikke slipper igjennom brannmurer og rutere. Noen av disse problemene, slik som NAT-ing av AH, er det mulig å løse, men det er opp til den enkelte leverandør.

Dette har ført til at for klient-VPN brukes i all hovedsak proprietære IPSec-implementasjoner fra for eksempel CheckPoint og Cisco, hvor man kjøper VPN-konsentrator fra leverandøren og får levert med en klient som må installeres på PC-ene. Dette altså i motsetning til å bruke en generisk IPSec-implementasjon som følger med operativsystemet. Mye av det samme er forsåvidt også tilfelle for nettverk-nettverk-VPN, da komponentene som binder sammen nettverkene typisk er produkter fra de samme leverandørene, slik som CheckPoint og Cisco.

Implementasjoner av IPSec

Det finnes mange implementasjoner av IPSec, hvor noen av de mest utbredte er:

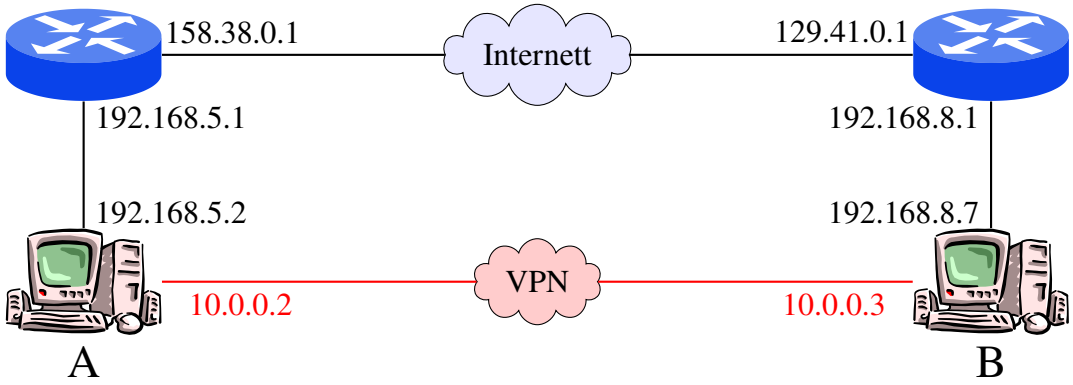
- CheckPoint VPN-1: CheckPoint leverer brannmurer med integrert VPN-konsentrator. Dette produktet heter VPN-1 og har en medfølgende klient som kan installeres på brukernes PC-er. Produktet er ganske utbredt, noe som blant annet skyldes at det er veldig funksjonsrikt. VPN-1 tilbyr mange ulike former for autentisering ut over

det som er spesifisert i IPSec (krypteringsertifikater), blant annet vanlig brukernavn/passord og RSA SecurID (brikke tilsvarende det de fleste har til nettbanken). VPN-1 har også bra funksjonalitet for policy based access control, det vil si at de som administrerer VPN-konsentratoren kan stille krav til PC-en din for at du skal få logge deg inn. Det kan være for eksempel at du må ha skrivebeskyttet skjermsparer, oppdatert antivirus-program og oppdatert operativsystem.

- Cisco VPN: Cisco leverer både egne VPN-konsentratorer og tilsvarende funksjonalitet til ruterne sine. Cisco VPN er mye brukt, blant annet fordi mange organisasjoner har en stor base av Cisco-basert nettverksutstyr. Cisco har erstattet denne produktlinjen med en ny serie som kalles Cisco AnyConnect VPN, som vi ser på litt senere.
- Microsoft Windows: Alle Windows-versjoner fra og med Windows 2000 er levert med IPSec. Implementasjonen er mest brukt sammen med protokollen L2TP, som vi ser på litt senere.

12.4.2 Tunneler med virtuelle nettverkskort og TLS/SSL

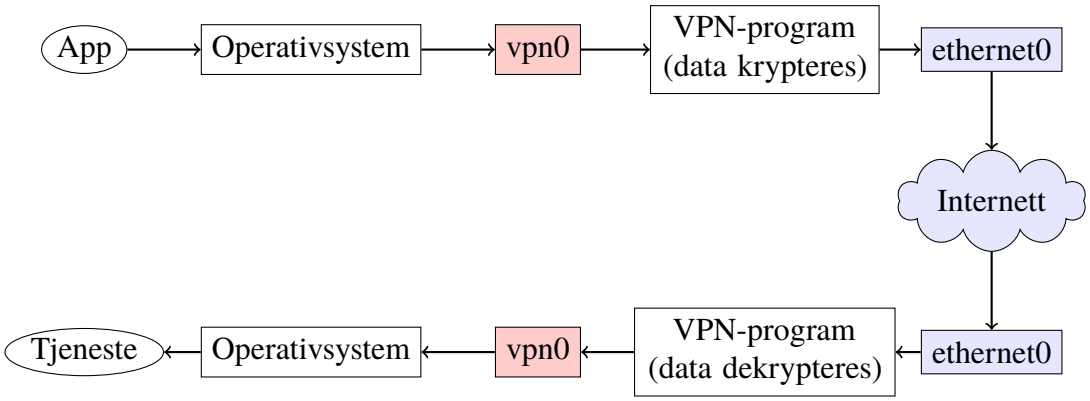
En metode for å konstruere VPN som på vei oppover i popularitet, er å konstruere et nettverk basert på krypterte tunneler og virtuelle nettverksgrensesnitt. Dette ser vi et eksempel på i figur 12.3.



Figur 12.3: VPN konstruert ved hjelp av virtuelle nettverksgrensesnitt («nettkort»). Det sorte nettet eksisterer i form av kabler, det røde er virtuelt og finnes bare som programvare.

I figur 12.3 er de to maskinene A og B tilknyttet to ulike nettverk. De to maskinene kan ikke kommunisere direkte med hverandre, fordi ip-adresser i serien 192.168.x.x ikke rutes på Internettet. Om de hadde hatt offentlige adresser slik som ruterne har, kunne de kommunisert direkte. Men det ville ikke nødvendigvis vært trygt, fordi Internettet kan avlyttes.

For å kunne kommunisere trygt er det etablert et VPN mellom maskin A og maskin B. VPNet består av en kryptert forbindelse, hvilket betyr at VPN-programvare på maskinene har kontaktet en VPN-tjeneste. De kan bruke en intern vpn-tjeneste implementert i ruterne, eller en ekstern tjeneste implementert noe sted på Internettet. De har autentisert seg mot denne tjenesten, og er klar til å forestå kommunikasjon. Hver av de to maskinene får da et virtuelt nettverksort vi kan kalle «vpn0». Dersom operativsystemet på maskin A sender data til «vpn0» vil disse dataene krypteres, sendes via nettverket og komme frem til maskin B hvor dataene dekrypteres og kommer ut av nettverkskortet «vpn0». Fysisk transporteres naturligvis dataene ut og inn på de fysiske nettverkskortene i maskinene, men operativsystemet kan sende og motta data til og fra det virtuelle nettverkskortet som om det var et fysisk nettverkskort. Vi kan behandle «vpn0» på de to maskinene som nettverkskort knyttet direkte sammen med en (virtuell) nettverkskabel.



Figur 12.4: Dataflyt i et VPN basert på virtuelle nettverksgrensesnitt. Svar på forespørsler tar motsatt vei tilbake.

Som vi ser i figur 12.4, kan operativsystemet rute IP-pakker til nettverkskortet vpn0. Dette fører til at pakkene krypteres, og det krypterte innholdet legges i nye IP-pakker som sendes ut via ethernet0 med mottaker-IP-adresse satt til IP-adressen til maskinen i den andre enden av VPN-forbindelsen. I den andre enden reverseres prosessen.

Hvilken trafikk er det som sendes ut på vpn0, og hvilken sendes ut på ethernet0? Dette bestemmes av maskinens rutetabell. At VPN-tunnelen fremstår som et vanlig nettverkskort gjør at vi ikke trenger å behandle den spesielt — vi kan behandle trafikk til og fra VPNet på samme måte som om trafikk som skal gå direkte via et fysisk nettverkskort.

Vi kan tenke oss at maskin B er en ruter som har tilknytning til nettverket 192.168.8.0/24. Dersom vi på maskin A ønsker at trafikk til dette nettverket skal gå via VPN-tunnellen sørger vi for å legge til en rute via VPN-tunnelen til dette nettverket i maskin As rutetabell. Dette er illustrert i tabell 12.1 på neste side.

Tabell 12.1: Tenkt rutetabell i maskin A

Adresse	**Nettmaske**	**Neste Hopp**	**Nettkort**
192.168.5.0	255.255.255.0	–	ethernet0
10.0.0.0	255.255.255.252	–	vpn0
192.168.8.0	255.255.255.0	10.0.0.2	vpn0
0.0.0.0	0.0.0.0	192.168.5.1	ethernet0

Lokale ip-pakker til 192.168.5.x sendes ut på det lokale nettet. Pakker til 10.0.0.x sendes til vpn-programvaren, som krypterer og sender pakkene til vpn-tjenesten så de etterhvert kommer til maskin B. Pakker til 192.168.8.x sendes til ruteren 10.0.0.2 (maskin B på vpn), så maskin A kan dermed snakke med andre maskiner i dette nettet ved å gå via maskin B. Pakker til alle andre adresser rutes via den lokale ruteren 192.168.5.1.

Kryptering med TLS/SSL

De to partene i eksemplet kan lage en tunnel som er helt åpen, for eksempel ved å bruke Point-to-Point Protocol, PPP. Innenfor vårt scope med sikre VPN er dette imidlertid uaktuelt. Tunnelen dem i mellom må altså opprettes med en kryptert protokoll, og protokollen som brukes er TLS.

Transport Layer Security, TLS (`http://en.wikipedia.org/wiki/Transport_Layer_Security`), tidligere kjent under navnet Secure Sockets Layer, SSL, er et sett protokoller for å sikre kommunikasjon mellom to parter. Vi bruker navnene TLS og SSL litt om hverandre i denne leksjonen, og for alle praktiske formål kan vi si at de er det samme (TLS 1.0 er praktisk talt det samme som SSL 3.0). TLS brukes for å sikre en forbindelse mellom to prosesser, hvor de to prosessene typisk går på hver sin maskin to ulike steder på Internett. Med TLS er det mulig å bruke mange ulike krypteringsalgoritmer hvor de nyeste naturlig nok er de sikreste. Når to prosesser initierer TLS-kommunikasjon blir de enige om hvilke algoritmer som skal brukes til de ulike formålene (nøkkelutveksling, autentisering, symmetrisk kryptering og sjekksumgenerering). De bestemmer seg for «minste felles multiplum» for hvert formål, altså den beste algoritmen som begge parter støtter. Med oppdatert programvare kan vi derfor regne kommunikasjonen som sikker.

TCP eller UDP

Kommunikasjon mellom partene må over Internett forgå med protokoller som lar seg transportere. Det vil si at trafikken må gå inne i IP-pakker. Inne i IP-pakkene må TCP eller UDP brukes til å transportere den krypterte protokollen (TLS). TCP er forbindelsesorientert og håndterer feil, mens UDP er forbindelsesløs og håndterer ikke feil. Hvilken

av disse protokollene er egnet til å formålet?

Dersom vi bruker UDP vil det gi mindre overhead og altså være raskere. Inne i tunnelen vil jo likevel TCP brukes for de forbindelsene som trenger feilhåndtering, eller hva? Hva skjer egentlig dersom en pakke forsvinner og vi har brukt UDP? Da vil mottakeren av pakken få problemer med å sette sammen SSL-strømmen og den må derfor ha en mekanisme for å håndtere denne type feil. OpenVPN som vi ser på nedenfor bruker denne metoden.

Bruk av TCP gir oss feilhåndtering og VPN-programvaren trenger derfor ikke tenke på dette. Dette er imidlertid mindre effektivt og er problematisk for applikasjoner som sender trafikk gjennom tunnelen og som benytter seg av UDP sine ytelsesfordeler. Microsoft har valgt denne løsningen i sin implementasjon av denne type VPN.

Som vi skal se litt senere så er det laget en protokoll som erstatter TLS: DTLS, som håndterer feilsituasjonene som OpenVPN selv håndterer.

OpenVPN

OpenVPN (`http://openvpn.sourceforge.net/`) var den første skikkelige VPN-løsningen basert på virtuelle nettverkskort og SSL/TLS. Programvaren kjører på de fleste plattformer, inkludert Windows, MacOS X, Linux og FreeBSD. Klientprogrammer fins også for android og Iphone. Slik kan man gi også telefoner og nettbrett tilgang til å bruke sikre nett, uansett hvor de er i verden.

OpenVPN består av et program som kjører på to maskiner for å opprette en tunnel seg i mellom. Trafikk i tunnelen transporteres i UDP-pakker hvor innholdet er kryptert med SSL. Hver av de to maskinene får etter etablering av tunnelen et virtuelt nettverkskort hvor trafikk sendes inn og ut på samme måte som et fysisk nettverkskort.

Vi får følgende protokollstack dersom vi surfer gjennom VPN-et: HTTP $\Rightarrow$ TCP $\Rightarrow$ IP $\Rightarrow$ SSL $\Rightarrow$ UDP $\Rightarrow$ IP.

OpenVPN brukes gjerne for å gi tilgang til ressurser plassert i sikre nettverk, som filtjenere, intranettressurser, epost og lignende.

Av annen nyttig funksjonalitet kan det nevnes at OpenVPN kan benyttes både for å lage tradisjonelle VPN hvor trafikk rutes mellom endepunktene, samt for å lage et virtuelt Ethernet. Sistnevnte betyr at også ikke-IP-baserte protokoller (IPX, Appletalk, …) vil fungere fint over VPNet, samt at broadcast-trafikk transporteres på samme måte som i et fysisk Ethernet. Det virtuelle Ethernettet kan godt være en del av et fysisk Ethernet. I tillegg støtter OpenVPN kommunikasjon over TCP som et alternativ til UDP, og støtter å sende trafikk via socks5-proxy og http-proxy. Dette gjør at man kan opprettet et VPN hvor man kan komme seg gjennom restriktive brannmurer.

OpenVPN access server

OpenVPN er gratis programvare. Å ta den i bruk, krever at man setter seg inn i hvordan det virker. For å komme fort i gang finnes produktet «openvpn access server», som ikke er gratis. Det har en del brukervennlige muligheter, som gjør det lett å komme i gang i stor skala. Når man har installert tjenesten, blir klientoppsett og programvare gjort tilgjengelig på en nettside. Dermed kan klientmaskiner surfe innom denne siden, og laste ned programvare med ferdig oppsett for akkurat ditt VPN.

Se `http://openvpn.net/index.php/access-server/overview.html`

Alt man kan gjøre med «access server» kan gjøres med gratisprogrammet også. Så det blir en avveiing; hva koster mest, lisensen eller arbeidstiden du trenger på å sette opp ting selv?

Openvpn nettverk til nettverk

Openvpn kan også brukes for å koble sammen hele nettverk, altså nettverk-til-nettverk-VPN. Dette kan f.eks. brukes for å koble sammen nettverkene til en bedrift som har kontorer i ulike byer. For et eksempel på slike oppsett, se:

`http://www.smallnetbuilder.com/other/security/security-howto/30353-how-to-set-up-a-site-to-site-vpn-with-openvpn`

Microsoft Secure Socket Tunnelling Protocol (SSTP)

Microsoft har definert protokollen Secure Socket Tunnelling Protocol, SSTP (`http://en.wikipedia.org/wiki/Secure_Socket_Tunneling_Protocol`). Med SSTP etableres VPN-et ved at klienten kontakter serveren på TCP-port 443 (porten avsatt til HTTP over SSL, HTTPS). Inne i SSL-forbindelsen som så etableres, tunnelerer klienten HTTP-pakker som inneholder en Point-to-Point Protocol (PPP)-tunnel, hvor selve nyttelasten går inni. Vi får altså følgende protokollstack dersom vi surfer gjennom VPN-et: HTTP $\Rightarrow$ TCP $\Rightarrow$ IP $\Rightarrow$ PPP $\Rightarrow$ HTTP $\Rightarrow$ SSL $\Rightarrow$ TCP $\Rightarrow$ IP.

SSTP følger med Windows Vista SP1, Windows Server 2008 og Windows 7. Protokollen er ment å erstatte VPN-protokollene PPTP og L2TP som følger med også tidligere Windows-versjoner. SSTP adresserer sikkerhetssvakhetene til PPTP og problemene med å traversere brannmurer som både PPTP og L2TP sliter med. Når det gjelder førstnevnte så brukes god kryptering (SSL) og når det gjelder sistnevnte så bruker SSTP altså en HTTPS-forbindelse på standard HTTPS-port (TCP-port 443), noe som slipper igjennom de aller fleste brannmurer.

Cisco AnyConnect VPN

Cisco har kommet med en erstatter for sitt «gamle» IPSec-baserte VPN (Cisco VPN), som de kaller Cisco AnyConnect VPN. AnyConnect baserer seg på protokollen Datagram Transport Layer Security, DTLS[8]. Med bruken av DTLS får AnyConnect UDP sine ytelsesfordeler samtidig som sikkerheten er ivaretatt. Det er ikke definert noen velkjent port for DTLS, men for å lett slippe igjennom brannmurer kan man for eksempel velge å bruke UDP-port 53, som er satt av til DNS og dermed ofte åpen.

Vi får følgende protokollstack dersom vi surfer gjennom VPN-et: HTTP ⇒ TCP ⇒ IP ⇒ DTLS ⇒ UDP ⇒ IP

OpenSSH VPN

Programvaren OpenSSH (`http://en.wikipedia.org/wiki/OpenSSH`) er som du kanskje er kjent med en SSH-klient, altså et program for å opprette SSH-forbindelser mellom maskiner. Fra og med versjon 4.3 av OpenSSH har du mulighet til å lage et VPN basert på virtuelle nettverkskort inne i SSH-forbindelsen ved hjelp av opsjonen «-w». Dette fungerer greit og har den fordelen at dersom du har en fungerende SSH-tjener med ønsket innloggingsmekanisme for brukerne, så er det lett å etablere et VPN. Minuset er at løsningen mangler mange av de «kjekke funksjonene» som de spesialiserte VPN-løsningene har, for eksempel for å dytte konfigurasjon ut til klienten når den kobler seg opp.

Vi får følgende protokollstack dersom vi surfer gjennom VPN-et: HTTP ⇒ TCP ⇒ IP ⇒ SSH ⇒ TCP ⇒ IP

12.4.3 Andre tunneleringsløsninger

Foruten VPN-løsninger basert på IPSec og løsninger basert på virtuelle nettverkskort og TLS/SSL, tar vi her for oss noen andre vanlig brukte tunneleringsløsninger laget etter andre prinsipper.

Point-to-Point Tunnelling Protocol (PPTP)

Point-to-Point Tunnelling Protocol, PPTP (`http://en.wikipedia.org/wiki/PPTP`) er et navnet på et sett protokoller spesifisert i RFC 2637 (`http://tools.ietf.org/html/rfc2637`). PPTP er i all hovedsak basert på arbeid nedlagt av Microsoft og er et populært VPN-alternativ fordi den dermed støttes av alle Windows-versjoner. PPTP er lett å ta i bruk fra Windows, men fordi det benyttes uvanlige protokoller/porter oppstår det ofte problemer når trafikken skal passere brannmurer. Spesifikt

[8] `http://en.wikipedia.org/wiki/Datagram_Transport_Layer_Security`

bruker PPTP protokollen Generic Routing Encapsulation (GRE) som har IP protokolltype 47, samt at kommunikasjon også foregår på TCP-port 1723.

Dersom man ønsker et sikkert VPN bør man styre unna PPTP, da den har flere klare svakheter. Blant annet er krypteringsnøklene en funksjon av brukerens passord, hvilket gjør dem lette å knekke. I Windows er PPTP gjort overflødig av først L2TP (fom. Windows 2000) og senere SSTP (fom. Windows Vista SP1).

Layer 2 Tunnelling Protocol (L2TP)

Layer 2 Tunnelling Protocol, L2TP (`http://en.wikipedia.org/wiki/Layer_2_Tunneling_Protocol`)er en tunneleringsprotokoll spesifisert i RFC 2661 (`http://tools.ietf.org/html/rfc2661`). L2TP er en populær VPN-protokoll fordi den støttes av alle Windows-versjoner fra og med Windows 2000. Fordi L2TP i seg selv ikke har noen sikkerhetsfunksjoner som kryptering og autentisering benyttes protokollen vanligvis sammen med IPSec, en kombinasjon som kalles L2TP/IPSec. Det er relativt lett å sette opp dette i Windows, men i likhet med PPTP møter man ofte problemer fordi L2TP/IPSec bruker uvanlige porter – UDP-port 500 (IKE) og UDP-port 1701 (L2TP).

Brukt sammen med IPSec er L2TP en sikker løsning rent protokollmessig, men fordi IPSec er en del av kjernekoden i operativsystemet vil en potensiell sikkerhetsfeil her kunne få dramatiske konsekvenser.

12.4.4 «Uekte VPN»

Som nevnt innledningsvis tilbyr noen produkter de samme fordelene som et «ekte» VPN, uten å lage et faktisk virtuelt privat nettverk. Vi skal se på to slike teknologier som er vanlig brukt.

Webbasert VPN

Stadig flere ønsker å kunne jobbe mot interne tjenester når de er hjemme eller på reise og etterspørselen etter klient-VPN har derfor vært økende og dette er et marked som fremdeles er i stor vekst. For denne type bruk er det essensielt at løsningen er enkel å installere og bruke på klientmaskinene, samtidig som kommunikasjonen må kunne passere NAT-rutere, brannmurer og lignende uten problemer.

Siden alle PC-er har en nettleser med TLS-støtte, har mange produsenter av VPN-produkter laget løsninger som utnytter mulighetene dette gir. Et slikt VPN kan vi kalle en HTTPS-proxy. Produsentene kaller det gjerne SSL-VPN, men vi velger å ikke bruke denne betegnelsen for å unngå sammenblanding med de øvrige VPN-løsningene vi tar for oss.

Å bruke et webbasert VPN innebærer at du kobler deg til en VPN-server ved å taste inn VPN-serverens navn i URL-feltet i nettleseren, f.eks. https://vpn.mittfirma.no/. På websiden som kommer opp taster du inn din autentiseringsinformasjon (brukernavn og passord, RSA SecurID-token eller tilsvarende). Etter å ha logget deg inn får du adgang til organisasjonens webbaserte tjenester.

Fordi mange tjenester ikke er webbaserte, har produsentene av denne type VPN-produkter laget små Java-applets eller ActiveX-komponenter som lastes i nettleseren. Denne lille applikasjonen tilsvarer en ordinær VPN-klient og lar applikasjoner på maskinen sende trafikk via sesjonen VPN-klienten oppretter med VPN-tjeneren. Dette gir mulighet for å bruke f.eks din vanlige e-postklient til å lese e-post på organisasjonens eposttjener.

Fordelene med denne type VPN-løsninger er at de i utgangspunktet fungerer uten installasjon av ekstra programvare, samt at de ikke har problemer med NAT og brannmurer. De gir oss også mulighet til å styre aksess til ulike tjenester på brukernivå. Ulempene er at for applikasjoner som ikke er web-baserte må nettleseren laste ned en liten applikasjon, og både applikasjonen og (i de fleste tilfeller) nettleseren må være åpne så lenge man ønsker å bruke VPNet. Manglende java/activex-støtte i nettleseren, samt restriksjoner på hva en java-applet/activeX-applikasjon skal få lov å gjøre på maskinen kan også å by på problemer.

Denne type VPN har sine begrensninger, men fordi det er lett å ta i bruk opplever leverandørene en vekst i etterspørselen.

SSH tunnelering (port forwarding), SSH+PPP

Protokollen SSH (Secure Shell) brukes i de fleste tilfeller som en kryptert variant av telnet. SSH har imidlertid flere muligheter, slik som filoverføring og tunnelering av TCP-trafikk (*port forwarding*). Sistnevnte kan benyttes for å oppnå VPN-type funksjonalitet hvor man tunnelerer trafikk gjennom SSH-sesjonen. Dette fungerer ved at klienten etter å koblet seg til en SSH-tjener oppretter en tunnel fra 127.0.0.1:portnummer til annen.maskin:portnummer. Dersom man på klientmaskinen tar kontakt med 127.0.0.1:portnummer vil trafikken tunneleres gjennom SSH-sesjonen til SSH-tjeneren, hvor kontakt etableres med annen.maskin:portnummer. For annen.maskin:portnummer vil kommunikasjonen fremstå som å komme fra SSH-tjeneren.

Teknikken med tunnelering av TCP-trafikk gjennom SSH blir ofte brukt når vi har et enkelt behov for å kommunisere med en tjeneste vi ikke når direkte, men kan nå via en SSH-tjener vi har tilgang til.

Dersom man ønsker en fullverdig tunnel gjennom SSH-sesjonen kan man velge å tunnelere PPP-trafikk. Dette fordrer at man bruker en PPP-tjener på tjeneren og en PPP-klient på klientmaskinen. Mange «hjemmelagede» VPN-løsninger baserer seg på akkurat dette.

SSH-implementasjonen OpenSSH, som er den vanligst brukte i UNIX/Linux/Mac OS X, har som vi så litt lenger opp også mulighet for å opprette et fullverdig VPN på egen hånd.

12.5 Oppsummering av VPN

Internett er en rimelig kommunikasjonskanal hvor vi kan knytte oss til fra nær sagt hvor som helst. Med VPN-teknologi (Virtuelt Privat Nettverk) kan vi bruke Internett for å gi aksess til tjenester vi ønsker å gi begrenset aksess til. Med slike løsninger kan vi sikre oss mot uautorisert bruk, samt være sikre på at informasjon ikke kommer på avveie gjennom bruk av kryptering. Det gir oss også mulighet til å tilby adgang over Internett til tjenester som bare er tilgjengelig på private IP-adresser.

Det finnes mange ulike VPN-løsninger med ulike styrker og svakheter. Et «ekte» VPN fungerer ved at man lager en virtuell nettverkskabel gjennom det åpne nettverket, dette gjelder f.eks. IPSec-baserte løsninger. Enkelte andre løsninger gir oss de samme fordelene, men fungerer annerledes. Dette gjelder for eksempel webbaserte VPN (såkalte SSL-VPN) og tunnelering av trafikk (port forwarding) med SSH.

Når man skal etablere et VPN kan man velge mellom mange produkter, men man må forsikre seg om at løsningen som velges fungerer med det utstyret man har, samt at løsningen må fungere under de forutsetningene man har (NAT, restriktive brannmurer, installasjon og vedlikehold av klienter, …)

12.6 Oppsummering

Vi har sett på IDS (Intrusion detection) for nettverk og enkeltmaskiner. Vi har også sett på IPS, som stanser en del angrep automatisk. Til slutt har vi sett på VPN som krypterer forbindelser over usikre nett (som Internettet).