Avdelning för datavetenskap

David Andersson

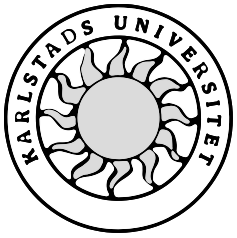
Teoretisk och praktisk genomgång av IPv6 och dess säkerhetsaspekter

Theoretical and practical review of IPv6 and its security aspects

Datavetenskap
C-uppsats

Datum/Termin: 12-06-07
Handledare: Hans Hedbom
Examinator: Donald F. Ross
Löpnummer: C2012:11

Karlstads universitet 651 88 Karlstad
Tfn 054-700 10 00 Fax 054-700 14 60
Information@kau.se www.kau.se



Computer Science

David Andersson

Teoretisk och praktisk genomgång av IPv6 och dess säkerhetsaspekter

---

Examensarbete, C-nivå

C2012:11

Teoretisk och praktisk genomgång av IPv6 och dess säkerhetsaspekter

David Andersson

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

David Andersson

Godkänd, 2012-06-07

Handledare: Hans Hedbom

Examinator: Donald F. Ross

Sammanfattning

Den här uppsatsen går teoretiskt och praktiskt igenom IPv6 för att skapa en förståelse för det nya protokollet. Uppsatsen beskriver utöver den teoretiska genomgången av protokollet även det praktiska arbete som ligger bakom implementationen som är tänkt att kunna ge såväl privatpersoner som mindre organisationer tillgång till en säker IPv6 lösning även om Internetleverantören endast kan erbjuda IPv4.

Arbetet är gjort på eget initiativ då behovet av kunskap kring IPv6 växer lavinartat och då privatpersoner troligen kommer att vara bland de sista som erbjuds IPv6 direkt från sin Internetleverantör.

Theoretical and practical review of IPv6 and its security aspects

Abstract

This paper takes a theoretical as well as a practical approach to what IPv6 is in order to create an understanding for the new protocol. In addition to the theoretical review of the protocol, this paper also describes the work that has been done in order to create an implementation that is supposed to bring secure IPv6 support to both private individuals and smaller organizations, even if the Internet service provider only offers IPv4.

This paper and implementation is done on a personal initiative, since the need for knowledge regarding IPv6 is rapidly increasing and private individuals probably will be among the last to be offered native IPv6 support from their Internet service providers.

Innehållsförteckning

Figurförteckning

Tabellförteckning

1 Inledning

I samband med insikten att de publika adresserna för IPv4 skulle börja ta slut påbörjades ett arbete med att ta fram en ny protokollspecifikation, IP Next Generation. Detta protokoll fick senare namnet IPv6.

Trots att IPv6 nu har funnits i över tio år så är mognaden och kännedomen kring protokollet fortfarande mycket låg och denna uppsats gör en ansats till att teoretiskt gå igenom de delar av IPv6 som behövs för att få en grundförståelse av protokollet och hur det kan samexistera med IPv4. Uppsatsen tar även upp vilka olika strategier som är att rekommendera då IPv6 skall införas.

Utöver protokollskillnader som IPv6 innebär så innebär det även en del förändringar när det gäller vilka tilltag som krävs för att få en god säkerhetsnivå. Uppsatsen gör även ett försök att adressera dessa frågeställningar.

Då den teoretiska genomgången av protokollet är gjord genomförs även en praktisk implementation vilket är den största delen av arbetet. Denna implementation är tänkt att kunna användas för mindre verksamheter som vill ha en möjlighet att få IPv6-stöd infört med en god grundkonfiguration där säkerheten hela tiden är i åtanke.

1.1 Disposition

Uppsatsen är uppdelad i sex kapitel utöver detta inledande kapitel.

Kapitel 1 är det inledande kapitlet och ger en orientering till uppsatsen och en förklaring till uppsatsens disposition.

Kapitel 2 går igenom bakgrunden bakom protokollet och förklarar vilket arbete som hittills har blivit gjort.

Kapitel 3 går igenom hur protokollet är uppbyggt samt vilka direkta skillnader som lätt kan urskiljas mellan IPv4 och IPv6. Utöver detta tar kapitlet upp vilka strategier som är att rekommendera vid ett IPv6-införande.

Kapitel 4 är det kapitel som går igenom säkerhetsaspekter och åtgärder som kan vidtas vad gäller IP i allmänhet och IPv6 i synnerhet.

Kapitel 5 går igenom vilka krav som ställs på designen av implementationen och varför dessa krav har ställts.

Kapitel 6 går praktiskt igenom hur installationen har genomförts och diskuterar även vilka beslut som tagits under vägen.

Kapitel 7 är det kapitel som sammanställer det arbete som gjorts och vilka lärdomar som dragits under arbetets gång.

2 Bakgrund

2.1 Historik IP

IP, eller Internet Protocol, är grundstommen i all kommunikation som idag sker på Internet och till stor del även den kommunikation som sker på lokala nätverk både hemma och på företag.

Idag använder vi oss främst av version 4 av IP, som definieras i RFC 791 [30] från år 1981. Detta var, oavsett vad namnet antyder, den första versionen där IP var ett eget protokoll. De funktioner som från och med den versionen hanteras av protokollet fanns dock tidigare [24], men var fram till separationen en del av TCP, eller Transmission Control Protocol. TCP hade dock genomgått flertalet revisioner sedan tidigare och kom i och med födseln av IP som ett eget protokoll upp i versionsnummer 4. I samband med detta valdes versionsnummer 4 även för IP.

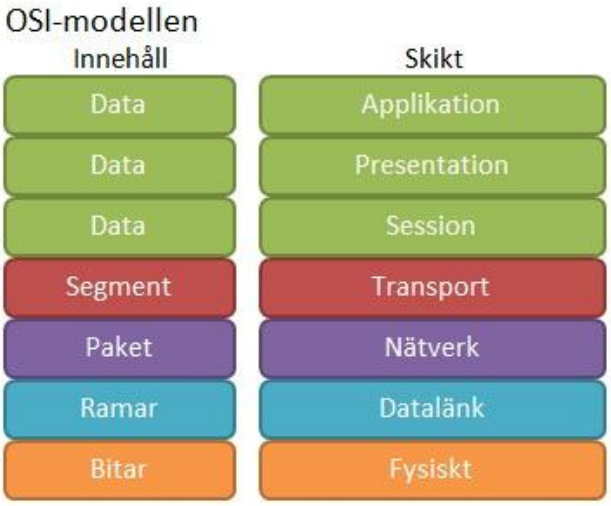
Under mitten av 1990-talet påbörjades arbetet med en ny version av IP. Den nya versionen kallades Internet Protocol Next Generation, men går numera under beteckningen IPv6. Anledningen till att versionsnumret för Internet Protocol Next Generation blev just version 6 var att version 5 redan hade reserverats för Internet Stream Protocol [15] . Detta protokoll arbetades fram redan under slutet av 1970-talet och var främst tänkt för att överföra ljud och rörlig bild över Internet. Denna version fick aldrig något genomslag utan användes endast sparsamt av några få företag.

IPv6 är alltså relativt nytt och stora delar skiljer sig från IPv4, av vilka kanske den tydligaste skillnaden rör de rent visuella aspekterna kring adresseringen och den stora adressutökning som protokollet möjliggör. Detta till trots så är strategierna som krävs för att skapa förutsättningar för en säker nätverksmiljö endast avvikande i ett mindre antal protokollspecifika aspekter. Fortfarande gäller de flesta grundläggande strategier som är aktuella för IPv4.

I och med detta så fokuserar uppsatsen på att skapa en förståelse kring IPv6 och både de säkerhetsaspekter som är unika för protokollet och gemensamma med IPv4. Med hjälp av denna kunskap skapas en implementation som med fördel kan användas för mindre verksamheter.

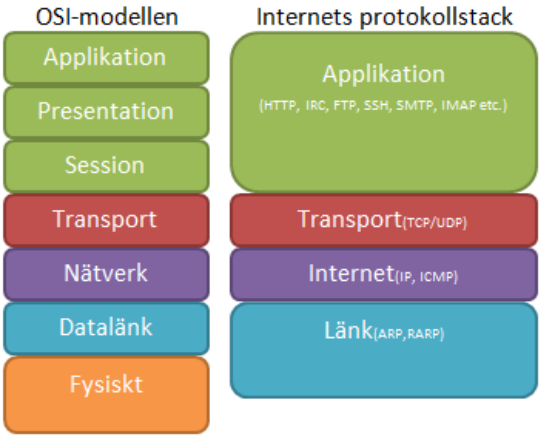
2.2 Skiktning av nätverkskommunikation

I likhet med OSI-modellen (se Figur 2.1), eller Open Systems Interconnection model, så gör TCP/IP ett försök att dela upp all nätverkskommunikation i olika lager. Tanken är att på så vis dela upp de moment som sker vid nätverkskommunikationen och endast tillåta relevanta operationer på sina respektive lager. Varje lager kan i sin tur endast direkt kommunicera med lagret direkt ovan eller under.



Figur 2.1 OSI-modellen

OSI-modellen och TCP/IP-modellen skiljer sig dock i det att OSI-modellen valt sju lager, medan TCP/IP-modellen blivit uppdelad i fyra lager. Enligt RFC 1122 [4] är dessa lager Applikation, Transport, Internet och Länk. Dessa fyra lager kan även till viss del översättas till OSI-modellen, men inte rakt av. Figuren nedan (se Figur 2.2) visar grovt hur översättningen mellan dessa modeller kan göras.



Figur 2.2 Jämförelse mellan OSI-modellen och Internets protokollstack

2.3 Relevanta RFC:er

Organisationen IETF (Internet Engineering Task Force) upprätthåller ett register med ett stort antal så kallade RFC:er (Request For Comments). En RFC är ett dokument som ofta, men inte alltid, används för att beskriva hur ett visst protokoll skall införas. Sedan tidigare har RFC 791 [30] nämnts som en viktig RFC för IP version 4. RFC 791 är dock huvudsakligen en revision av den till stor del liknande RFC 760. Tanken bakom dessa RFC:er är att oavsett vilken utvecklare som skriver en implementation av protokollet så skall alla implementationer kunna kommunicera med varandra, då de har följt samma standard när implementationen av protokollet har blivit skriven.

Ett antal av dessa RFC:er blir också till standarder, vilket är fallet med RFC 791. I de fall där de blir standarder får de även ett standardnummer. Detta standardnummer förändras inte även om de underliggande RFC:erna revideras. Dessa standarder får sedan även en status som appliceras, där den strängaste nivån är ”Required” medan den lösaste nivån är ”Elective”

Utöver dessa två RFC:er så är även RFC 1918 intressant för IPv4, då denna RFC behandlar tre segment av alla tillgängliga IP-adresser som klassas som privata. Detta innebär att dessa tre nätverkssegment endast får användas för lokala nätverk och att adresserna inte är tillgängliga på Internet.

På samma sätt finns det även RFC:er som gäller för IP version 6. Den som idag gäller och som antogs år 1998 är RFC 2460 [12]. I likhet med den RFC som finns för IPv4 beskriver RFC 4193 [21] vilka nätverkssegment som endast är avsedda för privata adresser.

Utöver dessa finns självfallet många fler RFC:er som är intressanta att läsa. Jag har nämnt RFC 1122 [4] som en sådan, vilket är en kravspecifikation för hur Internetvärdar skall dela upp kommunikationen i flera olika lager.

Även stödprotokollen för de olika versionerna av IP har RFC:er. Här kan exempelvis RFC 5494 [1] nämnas, som är den senaste versionen av RFC för ARP-protokollet, eller RFC 4884 [3] som beskriver ICMP för IPv4.

En enkel sökning på http://www.rfc-editor.org/index.html med sökorden ”Internet Protocol” ger i skrivande stund 644 träffar, så ett större antal RFC:er utöver redan nämnda finns tillgängliga för vidare förkovring.

2.4 Tidigare forskning

Då IP är en sådan viktig beståndsdel av all nätverkskommunikation av idag så har självklart mycket arbete kring berörda protokoll redan gjorts, även gällande säkerhetsfrågeställningar. Bland annat finns RFC 6274 [17] från IETF med en bedömning av säkerheten i IPv4 som på ett grundligt vis går igenom samtliga mekanismer som på något vis kan utnyttjas och på så vis ses som en säkerhetsrisk. Författaren som ligger bakom denna RFC heter Fernando Gont och han har även blivit tongivande inom säkerhetsforskning kring IPv6. Tyvärr har inte någon liknande bedömning kring säkerheten i IPv6 gjorts som motsvarar RFC 6274.

Kontinuerlig forskning kring säkerhetsaspekter i IPv6 sker dock fortlöpande och IETF har bland annat en arbetsgrupp för IPv6 maintenance (6man) som är en av IETF:s flera arbetsgrupper. Gruppens syfte är att underhålla och förbättra IPv6-protokollet och stor del av arbetet i gruppen rör säkerhetsaspekter i protokollet som kan förbättras. I skrivande stund arbetas det i gruppen bland annat med ”Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)” [18] och ”Security Implications of Predictable Fragment Identification Values” [19].

Utöver detta kan arbete som National Institute of Security Technologies nämnas som bland annat har producerat en publikation med referensnummer NIST SP800-119 [16] som behandlar vilka frågeställningar som är relevanta för en säker implementation av IPv6.

3 Uppbyggnad av IPv6

3.1 Kort beskrivning av IPv6

Införandet av Internet Protocol version 6 resulterade i ett stort antal förändringar från IPv4. Följande lista är tagen från boken ”Understanding IPv6 Second Edition” [10] av Davies och den sammanfattar mycket väl vad IPv6 faktiskt innebär.

- Nytt format på pakethuvud
- Stor adressrymd
- Stateful och stateless adresskonfiguration
- Krav på IPSec support
- Bättre support för högprioriterad leverans
- Nytt protokoll att interagera med nätverksgrannar
- Flexibilitet

En stor anledning till varför arbetet med IPv6 påbörjades i mitten av 1990-talet är att adressrymden för IPv4 redan då började ta slut. På gott och ont togs simultant specifikationen för NAT (Network Address Translation) fram. Detta har i sin tur resulterat i att den globala migreringen till IPv6 har dröjt[1], men intentionen bakom IPv6 var likväl densamma och har inneburit att adressrymden har vuxit markant i den nya versionen.

Adressrymden i IPv4 var 32 bitar, vilket genom binär matematik resulterar i att antalet globalt unika adresser teoretiskt begränsades till 2^{32} som kan uttryckas decimalt som 4,3 * 10^{9}. I praktiken är det till och med färre än så, då vissa nätverkssegment är reserverade för privat bruk och då varje nätverkssegment utnyttjar två adresser som inte kan adresseras av enskilda noder. Dessa adresser är nätverksadressen och broadcastadressen.

Adressrymden i IPv6 däremot är 128 bitar lång, vilket betyder att antalet globalt unika adresser är enormt. Med hjälp av samma binära matematik som i fallet med IPv4 kan man konstatera att detta ger en teoretisk möjlighet till cirka 3,4 * 10^{38} adresser. Samma resonemang gäller dock här, att ett antal adresser försvinner på grund av hur protokollet är designat.

Trots att det nya protokollet möjliggör så många fler adresser så är inte pakethuvudet för IPv6 motsvarande gånger större. På grund av protokollets design så möjliggörs all nödvändig information för involverade routers på blott dubbla storleken mot headerstorleken för IPv4.

[1] http://www.isoc.org/internet/issues/ipv6_faq.shtml

Även tilldelning av adresser sköts på ett nytt sätt. Redan då en nätverkskabel ansluts till ett nätverkskort kan detta nätverkskort kommunicera med grannar som finns på samma länk genom vad som kallas en link-local adress(se Kapitel 3.3.1.1). Den stora skillnaden i detta fall mellan IPv4 och IPv6 är att i IPv4 måste en förfrågan efter en dynamiskt tilldelad IP-adress från en DHCP-server tagit för lång tid varpå protokollet sedan genererar en lokal adress. I IPv6 däremot har nätverkskorten alltid en link-local IPv6adress, oavsett om nätverkskortet sedan skall tilldelas en global IPv6-adress eller inte.

Detta leder också till nästa stora skillnad. Till skillnad från IPv4 som oftast endast har en IP-adress per nätverkskort så har varje nätverkskort flertalet IPv6-adresser. Oftast har en värd följande IPv6-adresser:

- Loopback (::1, jämför med 127.0.0.1)
- Link-local adress (en FE80::-adress per anslutet nätverkskort)
- Minst en officiell, globalt åtkomlig adress per anslutet nätverkskort
- Ett antal multicast-adresser beroende på roller

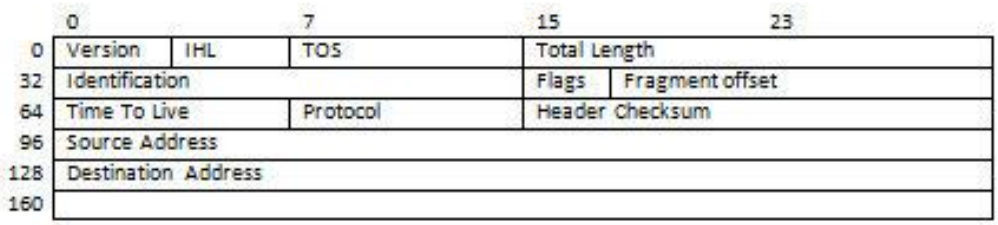
3.2 Protokollskillnader mellan IPv4 och IPv6

3.2.1 Inkompatibilitet

Namnet till trots så är det stora skillnader mellan IPv4 och IPv6 och de båda protokollen är inte kompatibla med varandra. Detta innebär också att för att kunna kommunicera över både IPv4 och IPv6 så krävs att båda protokollstackarna är installerade på noden i fråga.

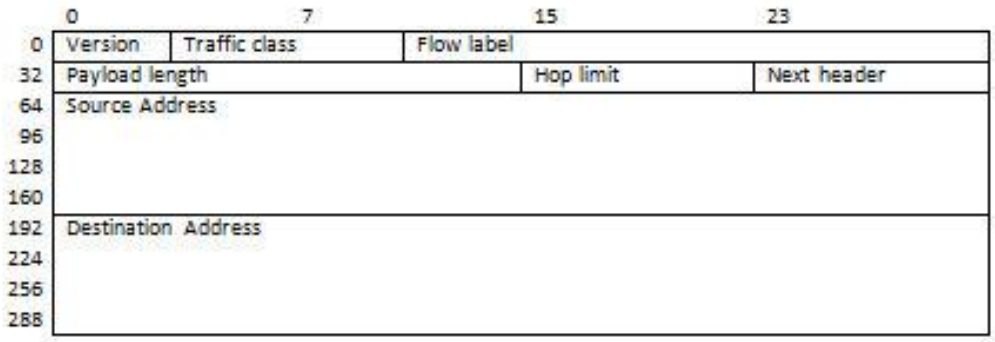
3.2.2 Pakethuvud i IP

I IPv4 så är storleken av själva pakethuvudet av en variabel karaktär. I sitt minimala utförande så är pakethuvudet 160 bitar långt, eller 20 bytes (se Figur 3.1) och i sitt maximala utförande är det 480 bitar långt, eller 60 bytes. Detta inkluderar då en källadress på 32 bitar och en destinationsadress på 32 bitar.



Figur 3.1 Pakethuvud i IPv4

Pakethuvudet i IPv6 däremot är av en fast karaktär på 320 bitar, eller 40 bytes (se Figur 3.2). Detta trots att både källadressen och destinationsadressen är 128 bitar lång.

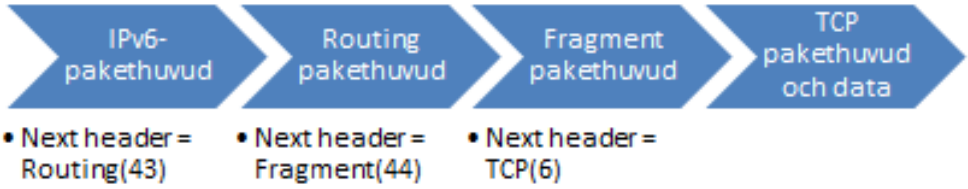


Figur 3.2 Pakethuvud i IPv6

Anledningen till att pakethuvudet i IPv6 är av en fixerad storlek är att pakethuvudet endast innehåller den information som behövs för att transportera vidare paketet. På så vis kommer endast själva informationen slutanvändaren till del.

Detta har möjliggjorts genom införandet av ”Next Header”-värdet. Detta värde är 8 bitar långt och kan antingen vara en så kallad ”Extension Header” eller innehålla den data (exempelvis TCP eller UDP) som skall sändas.

I en kedja av en eller flera Extension Headers kan man således specificera flertalet av de alternativ som fanns tillgängliga i IPv4, såväl i det minimala 20 byte långa pakethuvudet eller med ytterligare alternativ tillagda (se Figur 3.3 [12]).



Figur 3.3 Exempel på IPv6-paket med flera Extension Headers

I tabellen nedan (se Tabell 3.1) görs ett försök till att sammanfatta likheterna och skillnaderna i IPv4 och IPv6 på ett överskådligt vis (se RFC 791 [30] och RFC 2460 [12]).

IPv4	IPv6	Förklaring
Version	Version	Samma funktion i IPv4 och i IPv6
Header length	-	Saknas i IPv6, pakethuvudet är alltid 40 bytes
Type Of Service	Traffic Class	Samma funktion i IPv4 och i IPv6
-	Flow label	Nytt fält för att märka ett flöde i IPv6-paket
Total Length	Payload Length	Samma funktion i IPv4 och i IPv6
Identification	-	Saknas i IPv6 eftersom fragmentering hanteras via EH.
Flags	-	Saknas i IPv6 eftersom fragmentering hanteras via EH.
Fragment offset		Saknas i IPv6 eftersom fragmentering hanteras via EH.
Time To Live	Hop Limit	Samma funktion i IPv4 och i IPv6
Protocol Number	Next Header	Samma funktion i IPv4 och i IPv6
Header checksum	-	Saknas i IPv6. Felkontroll sker i andra lager av OSI-modellen
Source Address	Source Address	Samma funktion i IPv4 och i IPv6
Destination Address	Destination Address	Samma funktion i IPv4 och i IPv6
Options	-	Saknas i IPv6
Padding	-	Saknas i IPv6

Tabell 3.1 Skillnader mellan IPv4-värden och IPv6-värden

3.3 Adressering

Adresseringen skiljer sig markant mellan de två olika protokollen. I IPv4 är adresserna 32 bitar långa [20] och skrivs ut enligt en så kallad ”dotted-decimal-form”. Denna form består av fyra delar där samtliga delar är uppbyggda av åtta bitar som noteras i decimal form. Vardera av dessa grupper av åtta bitar kallas för en oktett. En giltig IP-adress har alltså fyra grupper av åtta bitar, eller fyra oktetter. 213.12.116.199. kan alltså vara ett giltigt exempel på en IPv4-adress.

För att notera adresser och nätverk i IP så använder man en notering som bygger på att man först skriver nätverksadressen och sedan hur många av adressens bitar som är låsta. Detta innebär att en fast IPv4-adress kan skrivas 213.12.116.199/32, då samtliga 32 bitar i adressen är låsta. Om man istället skriver 213.12.116.0/24 så innebär det att de första 24 bitarna är låsta, medan de sista 8 bitarna är variabla. I det här fallet skulle det innebära att nätverket 213.12.116.0/24 skulle ha möjlighet att husera 254 värdar. Den här noteringen kallas för CIDR (Classless Interdomain Routing).

IPv6 använder samma notering, men då adresserna är 128 bitar långa så innebär detta att man får använda ett annat antal bitar för att beskriva adresserna.

Då adresserna är så långa delas de upp i 8 så kallade 16-bit boundaries. Dessa 8 grupper av 16 bitar är sedan omvandlade till hexadecimala värden och är separerade av kolon. En giltig IPv6-adress kan alltså se ut 2001:0db8:0022:0000:0000:29ff:0000:009a.

I och med att adresserna i IPv6 är så långa har ett behov för förkortningsregler vuxit fram och följande två förkortningsregler finns [10].

1. **Initiala nollvärden**. Där värdet i de hexadecimala grupperna har inledande nollor kan dessa nollor uteslutas. Adressen ovan skulle med denna förkortning istället se ut 2001:db8:22:0:0:29ff: 0:9a.
2. **Grupper av nollvärden**. Där grupper av nollvärden finns kan värdena helt uteslutas och istället ersättas av ::. Denna förkortning kan endast användas en gång per adress. Adressen ovan skulle med denna förkortningsregel istället se ut 2001:db8:22::29ff: 0:9a

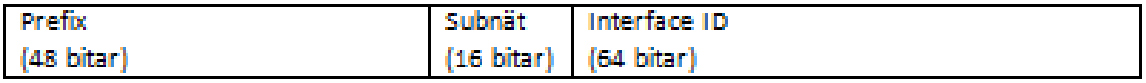
3.3.1 Adresstyper

Det finns flera olika typer av IP-adresser inom IPv6. Dessa adresstyper kan återfinnas i RFC 4291 [13] och i detta sammanhang bör följande adresstyper särskilt nämnas:

- Unicast-adresser
- Multicast-adresser
- Anycast-adresser

Värt att notera är att det i IPv6 inte finns några broadcast-adresser. Detta sköts istället med hjälp av multicastadresser. För att underlätta detta finns också ett stort antal fördefinierade multicastadresser.

Gemensamt för alla typer av adresser är att de följer en designstruktur som dikterar att de första 48 bitarna i adressen är ett globalt routing-prefix, de nästkommande 16 bitarna är ett subnät-ID och de sista 64 bitarna är ett anslutnings-ID (se Figur 3.4).



Figur 3.4 Adressuppdelning

Dessa adresstyper kan i sin del delas upp i ytterligare undergrupperingar.

3.3.1.1 Unicastadresser

En unicastadress används för att adressera en specifik nätverksanslutning och kan på så vis jämföras med en vanlig IPv4-adress.

Det finns flertalet olika typer av unicastadresser definierade. Bland dessa kan nämnas följande sex som är tagna ur RFC 3513 [11] och RFC 4193 [21]:

- Global
- Link-local
- Site-local
- Unique local
- Special
- Transition

De två typer av unicast-adresser som främst kommer att beröras är globala adresser och link-local adresser.

Globala unicast-adresser kan enklast jämföras med en publik IPv4-adress. Detta innebär att adresserna är globalt åtkomliga och kan adresseras på Internet. I RFC 3587 [14] definieras att samtliga globala unicast-adresser börjar med de att de tre första bitarna är satta till 001. Detta innebär i sin tur att det finns ett globalt routingprefix på 45 bitar som kan delas ut till

företag och organisationer. Att de tre första bitarna är låsta innebär att man enkelt kan se om adressen är global, då de alltid börjar på 2 (första fyra bitarna 0010) eller 3 (första fyra bitarna 0011).

Link-local adresser fanns även i IPv4 och dessa definierades i RFC 3927 [8]. Dessa adresser placerades på 169.254.0.0/16-nätverket. För att få en sådan adress tilldelad måste dock anslutningen ha försökt att få en vanlig adress tilldelad via DHCP. Först när tidsgränsen för detta överskridits kunde en link-local adress tilldelas. I IPv6 har samtliga anslutningar en link-local adress som tilldelas så fort anslutningen initieras. Dessa adresser är inte globalt åtkomliga och fungerar endast på anslutningsbasis. En nod med två anslutningar har alltså två link-local adresser. Link local adresser i IPv6 börjar alltid med de första tio bitarna satta till 1111 1110 10 och de följande 54 bitarna satta till noll. Dessa adresser börjar alltså alltid med FE80.

3.3.1.2 Multicastadresser

Multicastadresser fanns även i IPv4, men adresserna har fått en utökad roll och används mer frekvent i IPv6. Dessa adresser befolkas dynamiskt genom medlemskap, vilket innebär att en multicastadress kan innehålla flera IPv6-noder och en IPv6-nod kan när som helst begära medlemskap i gruppen eller lämna gruppen.

Gemensamt för samtliga multicastadresser i IPv6 är att de börjar med de första åtta bitarna satta till ett. Samtliga multicastadresser börjar således med FF.

3.3.2 Adresstilldelning

Interface-delen av en IPv6-adress är den del av en IP-adress som i IPv4 kallades för host ID eller node ID. I IPv4 var denna del av variabel storlek, beroende på vilken storlek man hade valt på nätmasken. I IPv6 är storleken på Interface ID alltid 64 bitar stor.

Denna del är den del av unicast-adressen som definierar en enskild värd på nätverket och kan konfigureras manuellt eller automatiskt med hjälp av antingen en stateful eller stateless konfigurationsmetod.

Både EUI-64 och Temporary Addresses är så kallade stateless konfigurationsmetoder. Samlingsnamnet för dessa båda adresstilldelningsmetoder är Stateless Address Autoconfiguration (SLAAC). I tillägg till detta kan man välja att använda en Dynamic Host Configuration Protocol (DHCP) v6-server för att tilldela ytterligare konfigurationsdetaljer. Om DHCPv6-servern i tillägg till dessa konfigurationsdetaljer även används för att generera värdet på Interface ID så kallas denna metod istället för stateful.

3.3.2.1 EUI-64

I RFC 4291 [13] definierades möjligheten att automatiskt tilldela Interface ID med hjälp av Extended Unique Identifier (EUI) -64-adressen på nätverkskortet. Nätverkskort av typen ethernet har så kallade MAC-adresser som kan översättas till EUI-64-adresser. Då EUI-64-adressen är av en fast karaktär så blir då även Interface ID-delen statisk oavsett vilket globalt routing-prefix och subnät-ID som används.

3.3.2.2 Temporary addresses

RFC 4941 [28] med titeln ”Privacy Extensions for Stateless Address Autoconfiguration in IPv6” definierade ytterligare en möjlighet att tilldela Interface ID och innebär att Interface ID istället tilldelas baserat på resultatet av en algoritm då detta kan generera viss anonymitet. Då Interface ID beräknas med hjälp av värdet från en statisk EUI-64-adress innebär detta att en nod alltid kan spåras med hjälp detta värde.

De steg som används för att räkna ut Interface ID om man följer RFC 4941 är som följer:

1. Läs in det historiska värde som finns lagrat.
2. Beräkna en MD5-hash av det inlästa värdet.
3. Spara de sista 64 bitarna av den uträknade hashen som det nya historiska värdet. Detta skall användas nästa gång en tillfällig adress skall tilldelas.
4. Använd de första 64 bitarna av den uträknade hashen. Den sjunde biten sätts till 0 vilket indikerar att adressen hanteras lokalt.
5. Resultatet av ovan listade steg tilldelas nu Interface ID och gäller tills dess att giltighetstiden för den tillfälliga adressen har gått ut och en ny adress skall tilldelas.

3.3.2.3 DHCPv6

I likhet med IPv4 så kan man även välja att tilldela adresser med hjälp av DHCP. Detta kräver dock en DHCPv6-server.

Denna metod kan användas både som stateful och stateless. Då man endast använder servern för att tilldela konfigurationsdetaljer som exempelvis vilken namnserver som skall användas så kallas det för en stateless metod.

Om DHCPv6-servern dessutom genererar den specifika adressen som skall användas för Interface ID och således skapar en tabell med information om vilka adresser som för tillfället tilldelats av servern ifråga så är den istället en stateful metod.

3.3.2.4 Manuell tilldelning

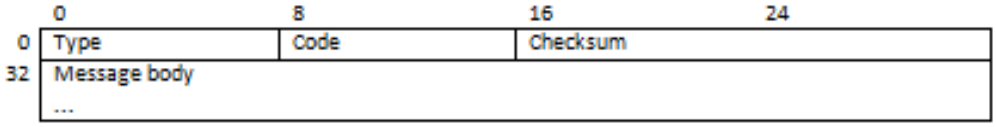
IPv6-adresser kan också i likhet med IPv4-adresser tilldelas manuellt för att få en statisk tilldelning. Detta kan exempelvis göras för att underlätta adressering och för att kunna dra nytta av de förkortningsregler som tillåts och används främst till nätverksutrustningar eller servrar.

3.4 ICMPv6

ICMPv6 är en viktig del av kommunikationen över IPv6 som definieras i RFC 4443 [7] och är i likhet med motsvarande protokoll för IPv4 en viktig del för kontroll och felsökning av nätverket.

Till skillnad från IPv4 så krävs dock ICMP i IPv6 för att nätverkskommunikationen överhuvudtaget skall fungera, då Neighbor Discovery är en del av ICMPv6 och ARP-protokollet är ett eget protokoll i IPv4.

ICMPv6-meddelanden är uppdelade i två typkategorier, felmeddelanden och informationsmeddelanden. De båda kategorierna separeras genom den mest signifikanta biten i Type-fältet. Är den satt till 0, vilket innebär att det totala värdet är mellan 0 och 127, så är meddelandet i fråga ett felmeddelande. Är biten istället satt till 1, vilket placerar det totala värdet mellan 128 och 255, så är meddelandet ett informationsmeddelande.



Figur 3.5 ICMPv6 paket

3.4.1 Neighbor Discovery-protokollet

Neighbor Discovery-protokollet innehåller meddelanden med typfältet satt till värden 133-137 och är en viktig del av ICMPv6, då de styr hur nätverket befolkas och hur trafiken etableras. Hela Neighborhood Discovery-protokollet definieras i RFC 2461 [27].

3.4.1.1 Router Solicitation (ICMPv6 typ 133)

Meddelande av typen Router Solicitation är ett meddelande som sänds ut då en nod blir aktiv på en länk. Detta meddelande sänds ut från den nod som vill börja kommunicera istället för att vänta på ett meddelande av typen Router Advertisement från någon router på länken.

Detta meddelande sänds sedan ut till alla noder som lyssnar på multicast-adressen FF02::2, vilket är multicast-adressen för samtliga routrar på den lokala länken.

```
Type: Router Solicitation (133)
Code: 0
Checksum: 0x23d1 [correct]
Reserved: 00000000
⊟ ICMPv6 Option (Source link-layer address : a0:88:b4:8c:da:24)
    Type: Source link-layer address (1)
    Length: 1 (8 bytes)
    Link-layer address: IntelCor_8c:da:24 (a0:88:b4:8c:da:24)
```

Figur 3.6 Exempel på Router Solicitation-meddelande

3.4.1.2 Router Advertisement (ICMPv6 typ 134)

Som nämnt ovan så finns även meddelande av typen Router Advertisement. Dessa meddelanden kan vara antingen Solicited (ett direkt svar på ett Router Solicitation-meddelande) eller Unsolicited.

Dessa meddelanden innehåller grundläggande information om de routrar som finns tillgängliga på den lokala länken och hur dessa är konfigurerade. I detta ingår också information om hur den efterfrågande nodens adress skall sättas. De flaggor som styr detta är Managed Address Configuration-flaggan och Other Configuration-flaggan och används då adresskonfigurationer skall tilldelas med hjälp av DHCPv6.

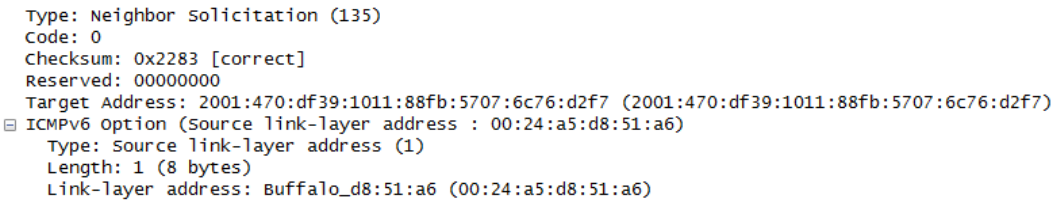
```
Type: Router Advertisement (134)
Code: 0
Checksum: 0x4319 [correct]
Cur hop limit: 64
⊟ Flags: 0xc0
    1... .... = Managed address configuration: Set
    .1.. .... = Other configuration: Set
    ..0. .... = Home Agent: Not set
    ...0 0... = Prf (Default Router Preference): Medium (0)
    .... .0.. = Proxy: Not set
    .... ..0. = Reserved: 0
Router lifetime (s): 1800
Reachable time (ms): 0
Retrans timer (ms): 0
⊟ ICMPv6 Option (Source link-layer address : 00:24:a5:d8:51:a6)
    Type: Source link-layer address (1)
    Length: 1 (8 bytes)
    Link-layer address: Buffalo_d8:51:a6 (00:24:a5:d8:51:a6)
```

Figur 3.7 Exempel på Router Advertisement-meddelande

3.4.1.3 Neighborhood Solicitation (ICMPv6 typ 135)

Neighborhood Solicitation är ett meddelande som används för att kartlägga det lokala nätverket och kan närmast jämföras med ARP i IPv4. Meddelande av typen Neighborhood Solicitation sänds ut för att översätta IPv6-adresser till MAC-adresser då nätverket är av typen Ethernet.

I likhet med Router Solicitation-meddelanden så sänds en förfrågan ut till alla noder som lyssnar på en särskild multicast-adress, i det här fallet FF02::1 vilket är adressen till samtliga noder på den lokala länken.

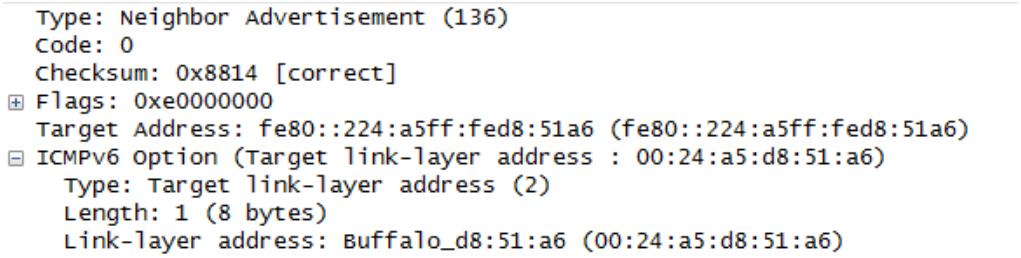


Figur 3.8 Exempel på Neighborhood Solicitation-meddelande

3.4.1.4 Neighborhood Advertisement (ICMPv6 typ 136)

Meddelanden av typen Neighborhood Advertisement sänds ut som ett direkt svar till den nod som har skickat ut ett meddelande av typen Neighborhood Solicitation så att denne kan kartlägga MAC-adresserna till de noder som finns tillgängliga på samma länk.

Neighborhood Advertisement-meddelanden kan dock också skickas ut till alla noder som lyssnar på multicast-adressen FF02::1 vilket är adressen för alla noder på samma länk. Dessa meddelanden skickas ut då en förändring har skett hos den sändande noden.



Figur 3.9 Exempel på Neighborhood Advertisement-meddelande

3.4.1.5 Redirect (ICMPv6 typ 137)

Redirect är meddelanden som skickas av en router för att informera en nod om att den kan nå en önskad nod genom att använda en annan första-punkt på dess väg.

Dessa meddelanden nyttjar endast unicast-adresser och går endast från en router till avsändande nod.

3.5 Strategier för införande av IPv6

Då IPv6 skall införas i ett nätverk som är av en något större natur finns det primärt två olika strategier för införande av IPv6. Dessa två strategier nämns bland annat i artiklarna av Juniper Networks [23] och av Cisco [9] och bygger på vilken del av infrastrukturen som man först skall välja att göra tillgängligt över IPv6. Vilken strategi man faktiskt väljer kan påverkas av såväl internt som externt ställda krav.

3.5.1 Kärna till kant

Den första strategin kallas kärna till kant och kan i ord uttryckas som att införandet börjar från de innersta delarna av nätverket för att sedan expandera stödet för IPv6 mot ytterkanten av nätverket.

Detta innebär att arbetet med att införa IPv6-stöd påbörjas på den fundamentala nätverksutrustningen så som routrar, switchar och brandväggar först för att sedan steg för steg införa stöd på utrustning längre ut, till exempel applikationsservrar eller klientdatorer.

Denna strategi är den som är lättast att införa förutsatt att stöd för IPv6 finns i den nätverksutrustning som är i kärnan av nätverket. Anledningarna till varför denna strategi kan ses som enklast är många, bland annat då man i ett tidigt skede kommer att upptäcka kravbilder och säkerhetsaspekter och kan göra dessa förändringar utan att det påverkar några publikt åtkomliga tjänster eller slutanvändares upplevelse på nätverket vilket försvårar förändringsbarheten i nätverket. Då denna strategi också i ett tidigt skede involverar den personal som skall tillhandahålla stöd för slutanvändare ger det också dem en möjlighet att bekanta sig med teknologin långt innan den når slutanvändarna vilket gör verksamheten bättre rustad för att klara av de operationella frågor som slutresultatet av ett IPv6-införande innebär.

3.5.2 Kant till kärna

Den andra strategin kallas kant till kärna och bygger på ett motsatt tillvägagångssätt. Denna strategi kan väljas då krav ställs från exempelvis en leverantör eller samarbetspartner på att stöd för IPv6 skall finnas på publikt exponerade tjänster, så som mail eller webb.

Denna strategi bygger oftast på ett stort användande av tunnlingstjänster, då all trafik behöver tunnlas genom den befintliga IPv4-infrastrukturen om den skall färdas längre in i nätverket.

En kant till kärna-strategi kan också innebära svårigheter då de som skall hantera kommunikationen oftast inte har fått erforderlig tid att bekanta sig med den nya teknologin och större risker uppstår för längre tider av otillgänglighet.

Utöver denna uppenbara risk innebär det vid vidare expandering till det övriga nätverket att stor del av säkerhetsdesignen redan är klarlagd och att förändringar i kärnan av nätverket riskerar att innebära stora konsekvenser ur såväl ett tidsperspektiv som ett säkerhetsperspektiv.

4 Säkerhet och säkerhetsmekanismer i IP

Darrin Miller arbetar som ingenjör på Ciscos säkerhetsavdelning och har uttalat sig i boken IPv6 Security [22] kring skillnaden i säkerhet från IPv4 och IPv6.

”IPv6 makes some things better, other things worse, and most things are just different, but no more or less secure”.

Denna mening sammanfattar mycket väl hur man bör förhålla sig till säkerhetsaspekterna av IPv6. Så länge man fortsätter att upprätthålla en god hållning till säkerhet överlag så kommer även IPv6 att hållas relativt säker.

Frågeställningarna är många, men till stor del är svårigheterna kring säkerhetsfrågor liknande i sin natur mellan de två olika versionerna. Säkerhetsfrågor som rör de övriga nivåerna i OSI-modellen kommer dessutom att vara desamma oavsett version av IP. En sårbarhet i en webb-server kommer finnas där oavsett om den lyssnar på IPv4 eller IPv6 och en nätverkskabel kan avlyssnas oavsett om trafiken som går över den går över IPv4 eller över IPv6.

Detta till trots så uppstår dock några nya frågeställningar som är specifika för IPv6 och ett antal frågeställningar som är specifika för IPv4 upphör att vara relevanta då man väljer att helt frångå detta protokoll.

4.1 Gemensamma frågeställningar för IPv4 och IPv6

4.1.1 Brandväggskonfiguration

I och med att man väljer att införa IPv6 så kommer samtliga noder att ha adresser som är publikt åtkomstbara. Detta ställer än större krav på att brandväggskonfigurationen är korrekt och att all trafik som inte uttryckligen skall nå en publik tjänst skall nekas.

Att alla noder är publikt åtkomstbara innebär inte en svaghet i sig, men då det under lång tid har använts privata icke-åtkomstbara adresser har det också glömts bort hur vital en brandvägg är i en nätverksinfrastruktur.

Att börja med att regelmässigt neka all ingående trafik som inte uttryckligen skall tillåtas är ett mycket gott första steg och ger en god grundsäkerhet.

I tillägg till detta kan ett införande av IPv6 vara ett gott tillfälle att införa lokala brandväggar på varje nod och inte bara tillförlita sig på den yttre brandvägg som hanterar trafik till och från Internet. Även för denna brandvägg gäller att standardregeln skall vara att neka all trafik som inte är uttryckligen godkänd.

Att börja med en dylik strategi innebär en avvägning mellan funktion och säkerhet, men ur ett säkerhetsperspektiv är det absolut nödvändigt att denna strategi införlivas i all utrustning.

4.1.2 Nätverksövervakning

Det finns flertalet tredjepartsmjukvaror som används i syfte att övervaka och rapportera eventuella avvikelser i nätverkstrafiken, såväl i IPv4 som i IPv6. Modeller och metoder som är gemensamma för IPv4 och IPv6 är så kallade Intrusion Detection Systems (IDS) eller Intrusion Prevention Systems (IPS) [22].

I tillägg finns även vissa specifika programvaror som endast används för att övervaka delar av nätverkstrafiken som endast är giltig för IPv6. Ett sådant exempel är det öppna verktyget NDPmon[2], som endast övervakar nätverkstrafiken och noterar trafik och avvikelser som är av typen NDP. Som en direkt jämförelse mot IPv4 kan ARPwatch[3] nämnas, vilken endast noterar trafik och avvikelser som är av typen ARP.

4.2 Inbyggd säkerhet

Som tidigare nämnts så finns det krav på att IPsec skall stödjas i varje utrustning som kan hantera IPv6 [26]. Detta innebär att all utrustning som använder IPv6 så som brandväggar, switchar, servrar och klienter fullt ut skall stödja användandet av IPsec.

IPsec ger stöd till att både autentisera och kryptera trafik och då detta sker så långt ner i nivåerna på OSI-modellen så innebär detta också att oavsett om applikationen i fråga har valt att kryptera nätverkstrafiken eller inte så kommer all data vid införande av IPsec-kryptering att vara krypterad.

Tyvärr ställs dock endast kravet att varje IPv6-produkt skall stödja IPsec, men man kan fortfarande välja när man gör en faktiskt implementation av IPv6 att inte dra nytta av dessa fördelar som IPsec innebär.

[2] http://ndpmon.sourceforge.net/
[3] http://www.securityfocus.com/tools/142

4.3 Tunnlingsprotokoll

Att använda sig av tunnlingsprotokoll kan vara ett sätt att möjliggöra IPv6-åtkomst trots att stöd inte tillhandahålls från Internetleverantören.

Det kan också vara ett illvilligt sätt att maskera otillåten trafik eller vara ett sätt att antingen kontrollerat eller okontrollerat skapa en öppen väg in till nätverket som kanske inte upptäcks av nätverksgrupper eller säkerhetsavdelningen.

Många organisationer använder sig idag av så kallade intrångsdetekteringssystem. Dessa kan antingen endast detektera intrång alternativt aktivt avbryta ett pågående intrång.

Denna utrustning kräver dock hög prestanda och kan lätt generera ett stort antal falsklarm om de är felaktigt konfigurerade. Detta leder till att dylik utrustning oftast endast övervakar redan kända protokoll, som TCP eller UDP.

Vad de dock oftast inte övervakar är exempelvis protokoll 41 [6], som är ett protokoll som möjliggör tunnling av IPv6-trafik över IPv4.

Detta har redan upptäckts av virusmakare [5] och virus har skapats med detta protokoll som kommunikationskanal för att undgå upptäckt.

Än värre kanske möjligheten är att en tunnel kan etableras från insidan för att sedan helt kringgå de säkerhetsmekanismer som redan finns på plats.

Ponera att tunnlingsprotokoll inte är nekade vid den yttre brandväggen och att en dator på nätverket av okänd anledning har etablerat en IPv6-över-IPv4-tunnel. Detta innebär att den datorn i sin tur har en publik IPv6-adress som kan nås genom denna tunnel.

Ponera vidare att denna även utger sig för att vara en router och således skickar ut RA-meddelanden till andra noder på samma länk. På detta vis kan sedan även dessa noder på nätverket bli publikt tillgängliga och även de helt kringgå befintliga säkerhetsmekanismer.

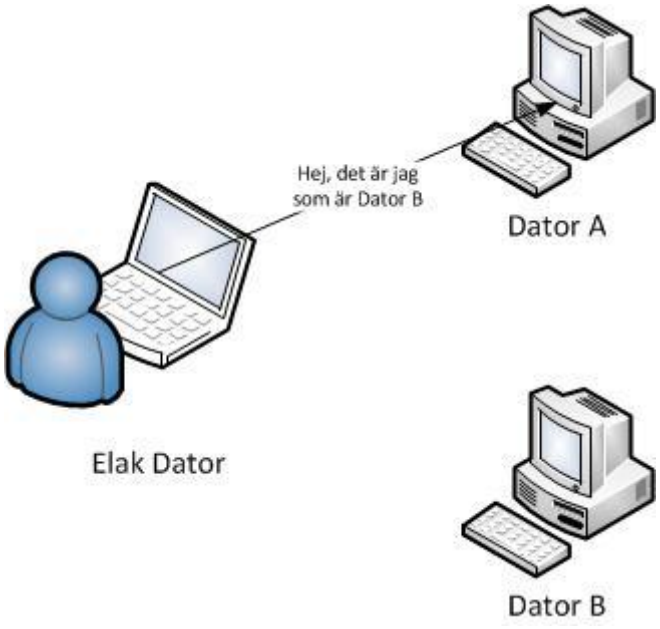
Dessa tillvägagångssätt är tyvärr helt möjliga och det finns flertalet instruktionsvideor tillgängliga på Internet som i detalj beskriver hur man skall gå tillväga.

Tunnlingsprotokoll kan således vara mycket nyttiga, men kan också vara en stor risk om de inte kontrolleras på tillbörligt vis.

4.4 SEND

Neighbor Discovery-protokollet (se Kapitel 3.4.1) är som tidigare diskuterat en kritisk del av IPv6 och behövs för att funktionaliteten skall kunna upprätthållas. Med hjälp av dessa bygger man bland annat upp en topologi över vilka enheter som är direkt anslutna till varandra.

Dessa meddelanden kan med enkelhet förfalskas, vilket kan leda till att information som är ämnad för en dator skickas direkt till en helt annan utan att slutanvändaren känner till detta. Används detta till att förfalska en nätverksnod som är ansvarig för att vidarebefordra trafik, såsom en router, så innebär detta att samtlig trafik som skall lämna nätverket först går genom den illasinnade enhet som har utgett sig för att vara denna nätverksnod (se Figur 4.1).



Figur 4.1 Förfalskad Neighbor Discovery

För att lösa denna problemställning har två nya typer av ICMPv6-meddelanden införts och är utökningar av NDP. Dessa heter CPS (Certification Path Solicitation) (ICMPv6 148) och CPA (Certification Path Advertisement) (ICMPv6 149) och är de ICMPv6-meddelanden som används då Secure Neighbor Discovery är infört. CPA är den del som ersätter Neighbor Advertisement och CPS är den del som ersätter Neighbor Solicitation.

Införandet av SEND [2] (Secure Neighbor Discovery) innebär att samtliga meddelanden som skickas är kryptografiskt säkrade med hjälp av RSA-nyckelpar och att Interface ID-delen av adressen genereras kryptografiskt.

SEND-förfarandet följer förenklat följande steg

1. Generera/erhåll ett RSA-nyckelpar med en publik och en privat nyckel

2. Generera en hash av den publika nyckeln och använd denna hash för att generera en Cryptographically Generated Address (CGA) som används till Interface ID
3. Tillsammans med prefixet på de första 64 bitarna genererar detta en komplett IPv6-adress
4. Sedan skapas ett Neighbor Discovery-meddelande där man skickar med adresser som gäller för noden i fråga och sänder med den publika nyckeln och en signatur som genererats av den privata nyckeln och meddelandet ifråga.

När motstående nod sedan tar emot Neighbor Discovery-meddelandet så verifieras adresserna med hjälp av den publika nyckeln och signaturen som sänts tillsammans med meddelandet.

SEND löser ett antal säkerhetsaspekter av NDP och beskrivs i korthet i Tabell 4.1:

Säkerhetsaspekter	**Lösningsmetoder**
Förfalskande av ND/NS	I och med kravet på en RSA-signatur och CGA-alternativ så kan inte en adress förfalskas
Attacker på RS/RA	SEND kräver att alla noder som skickar Router Advertisements inkluderar en RSA-signatur och bevis på behörighet vilket förhindrar möjligheten att anta rollen som router
Attacker baserade på återspelning av giltig trafik	SEND inkluderar ett alternativ att skicka med en tidsstämpel och både solicititation- och advertisement-meddelanden måste ha med detta värde

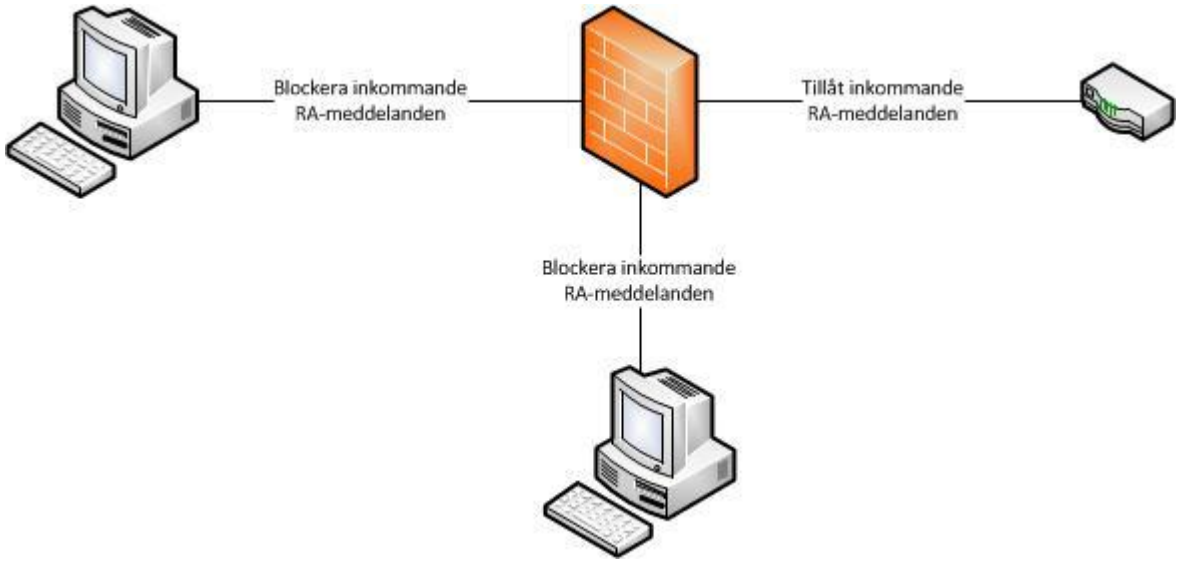
Tabell 4.1 Säkerhetsaspekter motverkade av SEND

Tyvärr är det i nuläget endast nätverksutrustning som har ett välfungerande stöd för SEND, medan det endast finns ett fåtal implementationer som kan användas av värdoperativsystem så som Linux och Windows.

4.5 RA-guard

Som ett något enklare alternativ till SEND kan routrar skyddas med hjälp av RA-guard som definierades i RFC6105 [25] . Detta bygger dock på att samtliga ändnoder som skall skyddas av RA-guard behöver skicka all trafik direkt genom en enhet som kan filtrera trafik på det andra lagret i OSI-modellen (se Figur 2.1) och att ändnoden inte kan kommunicera med någon annan nod än denna enhet.

På så vis kan denna enhet blockera Router Advertisements från enheter som inte är godkända routrar.



Figur 4.2 Teorin bakom RA-guard

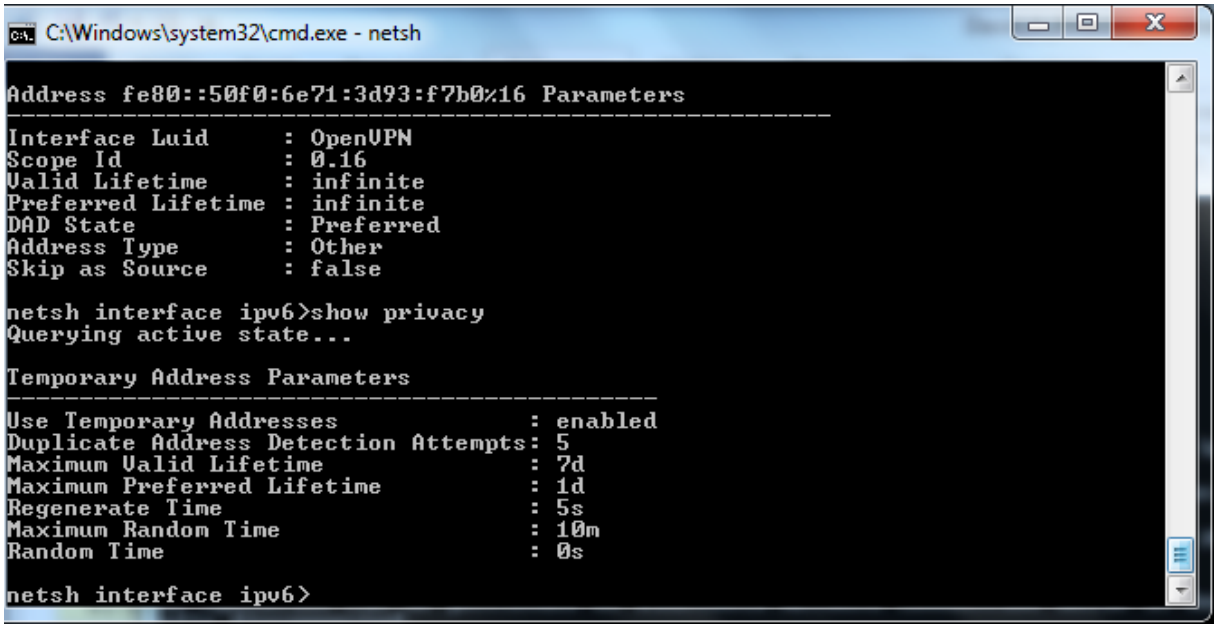
4.6 Personlig Integritet

Integritetsfrågor var i början av IPv6-införanden en stor fråga då de automatiska tilldelningarna till största delen baserades på EUI-64 och varje ansluten IPv6-nod då fick samma värde på Interface ID, vilket innebar att riktad reklam och övriga spårningsmekanismer underlättades.

Oavsett vilket prefix och vilket subnät som används (de första 64 bitarna i Figur 3.4), så kommer alltid de sista 64 bitarna ha samma värde. Med hjälp av detta försvinner stor del av anonymiteten som Internet i övrigt bidrar med. Detta underlättar kartläggning av en enskild värds beteendemönster på Internet.

Denna problemställning var något som som bidrog till RFC 4941 [28] som diskuterades i 3.3.2.2. Genom införandet av Privacy Extensions skapades Interface ID som varierade över tid och som på så vis inte kunde användas för att kartlägga en användares beteendemönster.

Idag stödjer de flesta operativsystem med IPv6-kompatibilitet möjligheten att använda sig av Privacy Extensions och från och med Windows Vista och Windows Server 2008 används denna adresstilldelning som standard för IPv6. Status på inställningen av Privacy Extensions kan enkelt visas i operativsystemet.



Figur 4.3 Status för Privacy Extensions i Windows 7

4.7 Okunskap

I likhet med införandet av andra nya teknologier innebär införandet av IPv6 att kännedomen kring teknologins styrkor och svagheter saknas bland de som är tänkta att hantera och administrera teknologin i fråga.

I detta fall innebär det ofta att de som först får ta del av IPv6 är individer som tillhör nätverksgrupperingen på en organisation och som ofta är självlärda på området. I bästa fall innebär det också att grupperingen har ett stort fokus på säkerhetsaspekter. Grupperingen kan dock lika gärna ha ett huvudfokus på funktionalitet och endast försöka tillämpa de säkerhetsmekanismer som varit giltiga för teknologins föregångare, i detta fall IPv4.

Då några säkerhetsmekanismer som är aktuella i IPv4 inte längre är giltiga för IPv6 och några av dessa dessutom kan förstöra funktionen helt och hållet för IPv6 kan detta resultera i att säkerhetsmekanismer tas bort godtyckligt tills funktionen är återställd. I värsta fall nöjer sig administratörerna vid detta och väljer att inte återställa de säkerhetsmekanismer som inte haft någon inverkan på funktionen.

5 Design till implementation av IPV6-infrastruktur

5.1 Grundläggande designspecifikation

Implementationen av IPv6-infrastrukturen baserar sig på tredjepartsmjukvaran OpenWRT som kan installeras på ett större antal routers för hemmabruk, bland annat produkten Buffalo Linkstation WZR-HP-G300NH vilken är den som kommer att användas i detta exempel. OpenWRT finns tillgängligt för nedladdning på http://www.openwrt.org/. I Bilaga A Installationsförfarande diskuteras dessa val mer ingående.

Implementationen ifråga kommer att vara en så kallad Dual Stack-implementation, vilket innebär att den kommer att supportera både IPv4 och IPv6.

För att få en implementation som är så verklig som möjligt har jag begärt ut ett globalt adresserbart nätverksspann för IPv6. Detta kommer dock att tunnlas ut över IPv4 via tunnlingsmekanismen 6in4. 6in4 och liknande tunnlingsmekanismer skulle i en skarp IPv6-implementation ha blockerats (se Kapitel 4.3) för att undvika att otillåten extern åtkomst möjliggörs. Dock kommer den yttre brandväggen att använda sig av just en sådan tunnlingsmekanism, då IPv6 inte tillhandahålls av Internetleverantören.

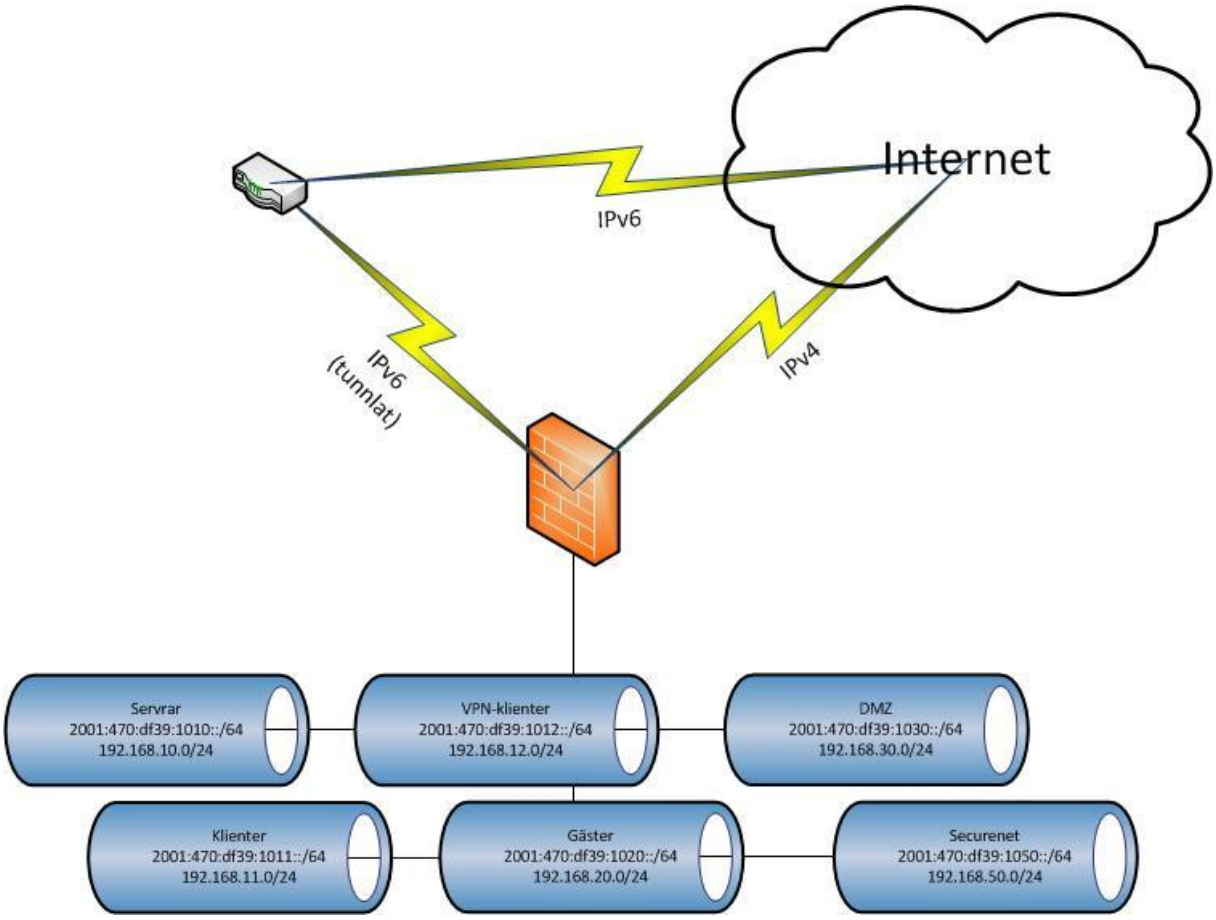
Samtliga klienter på de interna subnäten kommer att ha ursprungligt stöd för både IPv4-adresser och IPv6-adresser, medan Internet-åtkomsten endast kommer ha ursprungligt stöd för IPv4 och IPv6 tunnlas över denna IPv4-adress. Ett alternativ hade varit att använda Unique Local IPv6-adresser, men för att kunna verifiera implementationen även från ett WAN-gränssnitt så valdes ändå detta tillvägagångssätt.

Då det IPv6-nätverksspann som tilldelats innehåller giltiga och fullt routningsbara IPv6-adresser kommer alltså samtliga klienter på de interna subnäten att vara routningsbara via IPv6 vilket ställer höga krav på brandväggskonfigurationen.

5.2 Nätverksadressering

Det IPv6-spann som jag har blivit tilldelad är ett routat /48-nät som kommer att delas upp i flertalet /64-nätverk som kommer att användas av de interna klienterna. Kommunikationen mellan brandväggen och leverantören av tunneln (http://www.tunnelbroker.net) sköts över ett länknät som tilldelats av leverantören. Det /48-nät som har tilldelats och som kommer att delas upp ytterligare är 2001:470:df39::/48. Länknätet som sköter kommunikationen mellan brandväggen och leverantören är 2001:470:27:aa5::/64

I Figur 5.1 följer en nätverksskiss som illustrerar vilka nätverk som är anslutna till routern samt hur kommunikationen sker över både IPv4 och IPv6.



Figur 5.1 Designspecifikation och nätverksskiss

Som jag tidigare har nämnt så tillåter IPv6 vissa förkortningsregler i adresseringen. Med detta i åtanke och med rådande rekommendationer från NIST SP800-119 [16] har nätverksadresser skapats som skall vara så lättlästa som möjligt där det första fyra-bitarsvärdet är låst till värdet 1. Detta innebär att samtliga värden efter denna etta måste skrivas ut, även om värdet är 0.

I denna implementation kommer sex subnät att användas vilka är uppdelade enligt vilken funktion klienterna på respektive subnät skall ha. De subnät som finns tillgängliga är

Subnät	IPv4-spann	IPv6-spann
Servrar	192.168.10.0/24	2001:470:df39:1010::/64
Klienter	192.168.11.0/24	2001:470:df39:1011::/64
KlientVPN	192.168.12.0/24	2001:470:df39:1012::/64
Gäster	192.168.20.0/24	2001:470:df39:1020::/64
DMZ	192.168.30.0/24	2001:470:df39:1030::/64
Endast Internetåtkomst	192.168.50.0/24	2001:470:df39:1050::/64

Tabell 5.1 Nätverkstabell

5.3 Kommunikationsbegränsningar

Samtliga av dessa nätverkssegment kommer att ha regler där standardinställningarna kommer att tillåta såväl inkommande som utgående trafik mellan respektive nätverkssegment och brandväggen. I tillägg skall trafik som vidarebefordras från respektive nätverkssegment tillåtas ut på Internet. Trafik som skall vidarebefordras mellan de olika zonerna bestäms individuellt från zon till zon.

Däremot kommer samtlig trafik att nekas från Internet till brandväggen, med ett fåtal undantag som bland annat styr publicering av webserver och mailserver samt de ICMPv6-meddelanden som krävs för att upprätthålla funktionaliteten.

Brandväggsreglerna kommer att styra både IPv4- och IPv6-trafik, vilket bland annat syns tydligt vid kommunikation med servrar på DMZ.

5.4 Kravspecifikation

Ovan nämnda designspecifikationer har även omvandlats till en kravlista vilken här synliggörs med hjälp av en tabell (se Tabell 5.2) som i detalj listar vilka krav som skall uppfyllas av installationen.

Krav
1. IPv4 och IPv6 skall samexistera och samma brandväggsregler skall gälla för båda protokollen
2. Installationen skall så långt som möjligt likna ett skarpt införande av IPv6, vilket bland annat innebär globalt åtkomstbara IPv6-adresser
3. Den interna adresseringen skall vara logiskt åtföljd mellan IPv4 och IPv6
4. Installationen skall möjliggöra extern publicering av tjänster för både IPv4 och IPv6
5. Kommunikationen mellan zonerna skall införas med principen ”lägst möjliga åtkomstnivå”
6. Installationen skall i samtliga moment vara utförd med stor hänsyn tagen till säkerhetsaspekter och där det är tillämpligt skall rådande branschstandarder tillämpas

Tabell 5.2 Kravspecifikation

6 Implementation av IPv6-infrastruktur

Installationen av OpenWRT är väldigt enkel och man kan använda webbgränssnittet för att uppgradera firmware. Den firmware som valts för installation finns tillgänglig för nedladdning på http://downloads.openwrt.org/backfire/10.03.1/ar71xx/.

Installationen är automatiserad och skapar en filstruktur som liknar de flesta Linux-installationer som idag finns tillgängliga.

Hela installationsförfarandet finns beskrivet i Bilaga A Installationsförfarande, medan detta kapitel endast berör de moment som har en direkt anknytning till införande av IPv6.

6.1 IPv6-stöd

För att kunna använda IPv6 i brandväggen så har följande paket lagts till

```
$ opkg install ip ip6tables kmod-sit kmod-iptunnel6 radvd 6in4 luci-
proto-6x4 dibbler-relay
```

Utöver dessa paket som tillhandahåller grundläggande IPv6-funktionalitet så har även ett antal stödpaket lagts till, vilka används för att i större grad kunna diagnosticera och felsöka eventuella problem i IPv6-kommunikationen.

```
$ opkg install iputils-traceroute6
```

6.2 Nätverkskonfiguration

Nätverket som sätts upp konfigureras med fyra olika subnät kopplade till de fyra portarna på routern samt två extra subnät till vpn-nätverket och det trådlösa gästnätverket.

Utpekade IP-segment är tagna från specifikationen och tabellen Tabell 5.1. De fysiska portarna på brandväggen representerar sedan logiska nätverkssegment (se Tabell 6.1) .

Port	IPv4-spann	IPv6-spann
Port 1	192.168.50.0/24	2001:470:df39:1050::/64
Port 2	192.168.30.0/24	2001:470:df39:1030::/64
Port 3	192.168.11.0/24	2001:470:df39:1011::/64
Port 4	192.168.10.0/24	2001:470:df39:1010::/64

Tabell 6.1 Portkonfiguration

Konfigurationen finns lagrad i filen /etc/config/network och varje nätverksgränssnitt konfigureras med följande segment.

```
config 'interface' 'servers'
        option 'ifname' 'eth0.1'
        option 'type' 'bridge'
        option 'proto' 'static'
        option 'ip6addr' '2001:470:df39:1010::1/64'
        option 'ipaddr' '192.168.10.1'
        option 'netmask' '255.255.255.0'
```

Den konfigurationsrad som läggs till för att stödja IPv6 heter ip6addr och specificerar vilken adress gränssnittet i fråga skall tilldelas, men även vilken nätmask som skall användas. Då rekommendationen är att varje nätverk skall nyttja en 64-bitars nätmask så har detta använts även i denna implementation.

6.3 Adresstilldelning

För att adresstilldelningen för IPv6 skall fungera så har paketen radvd och dibbler-relay lagts till, vilka sköter olika delar av adresstilldelningen.

6.3.1 Radvd

Radvd är den implementationen som används i OpenWRT för att skicka Router Advertisements och Router Solicitations. Paketet konfigureras med filen /etc/config/radvd.

Radvd har konfigurerats för både clients, clientvpn, guests, servers och securenet. Med hjälp av detta så behöver inte en default gateway anges, utan routern skickar själv ut information om vilken möjlighet som finns att kommunicera med övriga IPv6-noder.

För varje gränssnitt som skall skicka Router Advertisements så konfigureras dessa med en definition enligt nedan. Varje rad definierar sedan konfigurationsparametrar för respektive gränssnitt.

```
config 'interface'
        option 'AdvSendAdvert' '1'
        option 'ignore' '0'
        option 'interface' 'clients'
        option 'IgnoreIfMissing' '1'
        option 'AdvManagedFlag' '1'
        option 'AdvOtherConfigFlag' '1'
        option 'AdvSourceLLAddress' '1'
        option 'AdvDefaultPreference' 'medium'
```

6.3.2 Dibbler-relay

Då brandväggen inte har någon DHCP-server installerad, utan den är installerad som en roll på servern kil-s002 så krävs det att samtliga begäranden av DHCPv6-adresser

vidarebefordras till en DHCPv6-server. Denna vidarebefordring skes av en så kallad DHCPv6-relay och krävs för varje nätverkssegment där en IPv6-adress skall tilldelas över DHCP.

Vad detta innebär är att brandväggen lyssnar på multicastadressen ff02::1:2 som är adressen för samtliga dhcp-servrar och dhcp-relays på det lokala nätverkssegmentet. Sedan vidarebefordras dhcp-begäran till Windows-servern kil-s002 för tilldelning av adress.

Detta innebär att ett gränssnitt måste definieras som en server, enligt nedan.

```
iface br-servers {
        server multicast
}
```

Utöver denna konfiguration behöver även varje gränssnitt som skall tilldelas adresser via DHCP definieras, enligt nedan.

```
iface br-clients {
        client multicast yes
        interface-id 10
}
```

6.4 Brandväggskonfiguration

I OpenWRT finns en konfigurationsfil för den inbyggda brandväggen under /etc/config/firewall. Denna konfigurationsfil skapar sedan ett antal regler som begränsar kommunikationsmöjligheterna med hjälp av Iptables [29].

Då Iptables är det som ligger i botten så kan även direkta Iptables-kommandon användas av brandväggen och matas då in i filen /etc/firewall.user.

I filen /etc/config/firewall finns ett antal konfigurationsdefinitioner som kan användas för att styra brandväggens funktionalitet. Först kan definitionen ”zone” nämnas, vilken definieras enligt nedan.

```
config zone
        option name             servers
        option network         'servers'
        option input            ACCEPT
        option output           ACCEPT
        option forward          REJECT
```

Denna definition knyter nätverksgränssnittet servers till zonen servers, där standardreglerna definierar att ingående och utgående trafik direkt till den zonen tillåts, medan trafik som skall vidarebefordras till och från andra zoner inte tillåts. Liknande zon-definitioner har sedan skapats för alla gränssnitt.

Vidare har även vidarebefordringsregler definierats för varje zon. Exempelvis tillåts klienter att komma åt Internet via den zon som sköter tunnlingen mellan IPv4 och IPv6 genom definitionen nedan.

```
config forwarding
        option src              clients
        option dest             henet
        option family           ipv6
```

Liknande definitioner har sedan skapats för såväl IPv4 som IPv6 mellan de olika interna zonerna som tillåts kommunicera med varandra.

Utöver dessa definieras sedan specifika åtkomstregler för varje tjänst som skall tillåtas. Nedan visas definitionen för regeln som tillåter åtkomst för webb-trafik till den servern som skall vara publikt åtkomlig.

```
config 'rule'
        option 'src'            '*'
        option 'dest'           'dmz'
        option 'proto'          'tcp'
        option 'dest_port'      '80'
        option 'target'         'ACCEPT'
        option 'family'         'ipv6'
```

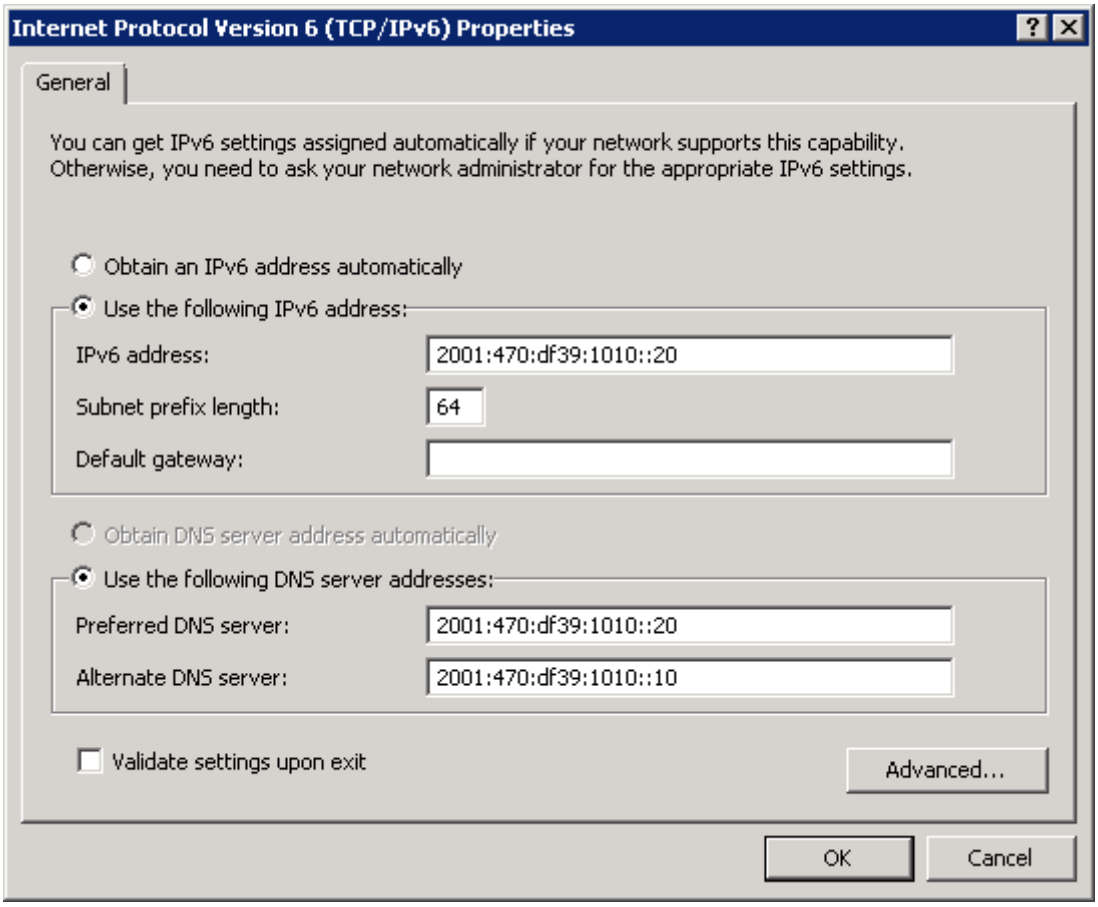
Med hjälp av dessa tre konfigurationsdefinitioner skapas sedan ett komplett regelset vilket används för att evaluera vilken slags trafik som skall tillåtas.

6.5 Serverkonfiguration

Samtliga ingående servrar har konfigurerats med en statisk IPv6-adress som hör till respektive nätverkssegment. De ingående Windows-servrar som används ligger på det interna serversegmentet och den ingående Linux-servern som används ligger på DMZ-segmentet.

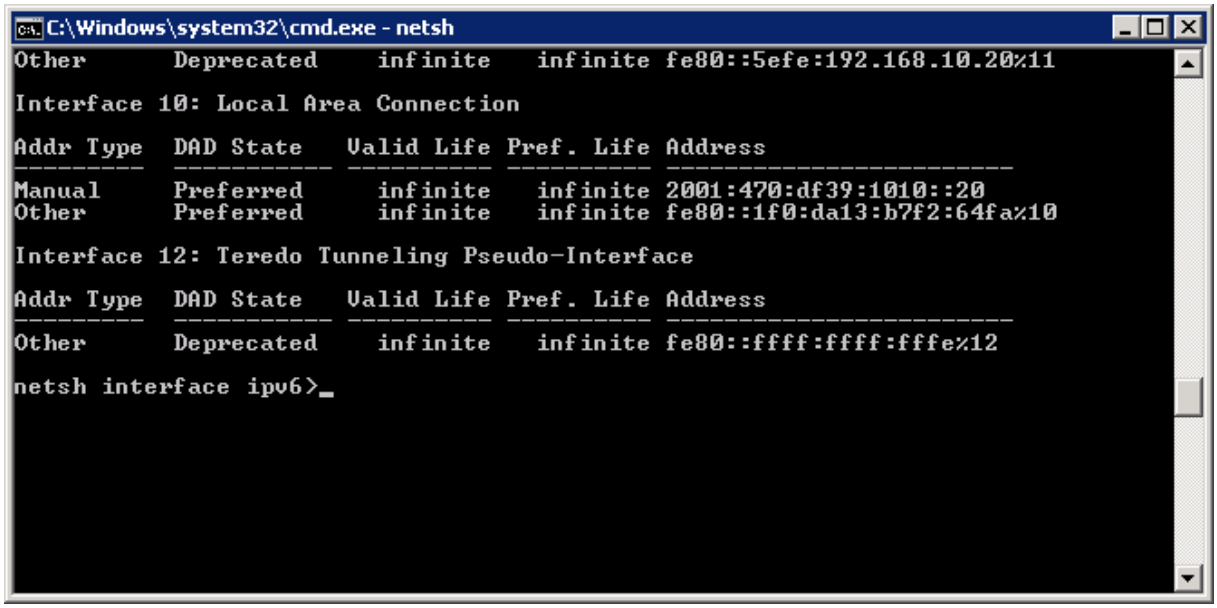
6.5.1 Windows-servrar

Servern med adressen kil-s002.hogbacken.local har nätverkskonfigurationen satt enligt Figur 6.1. Värdet på Default Gateway sätts med hjälp av Router Advertisement från brandväggen.



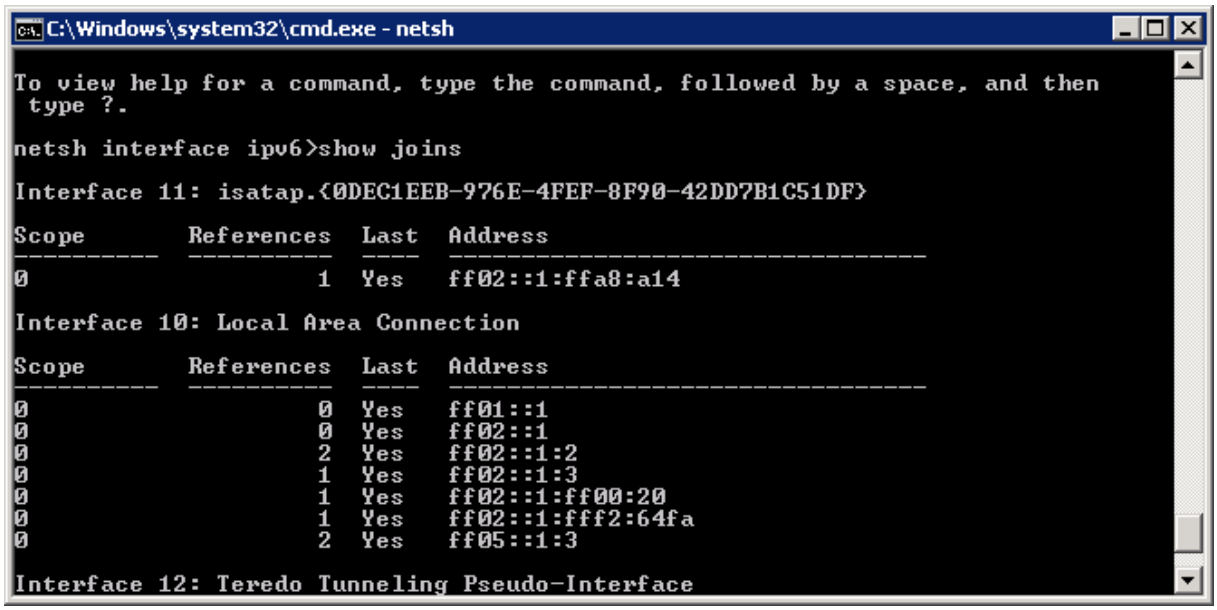
Figur 6.1 Statisk nätverkskonfiguration av intern server

Utöver den statiskt tilldelade adressen så har servern även en Link-Local-adress, enligt figuren nedan.



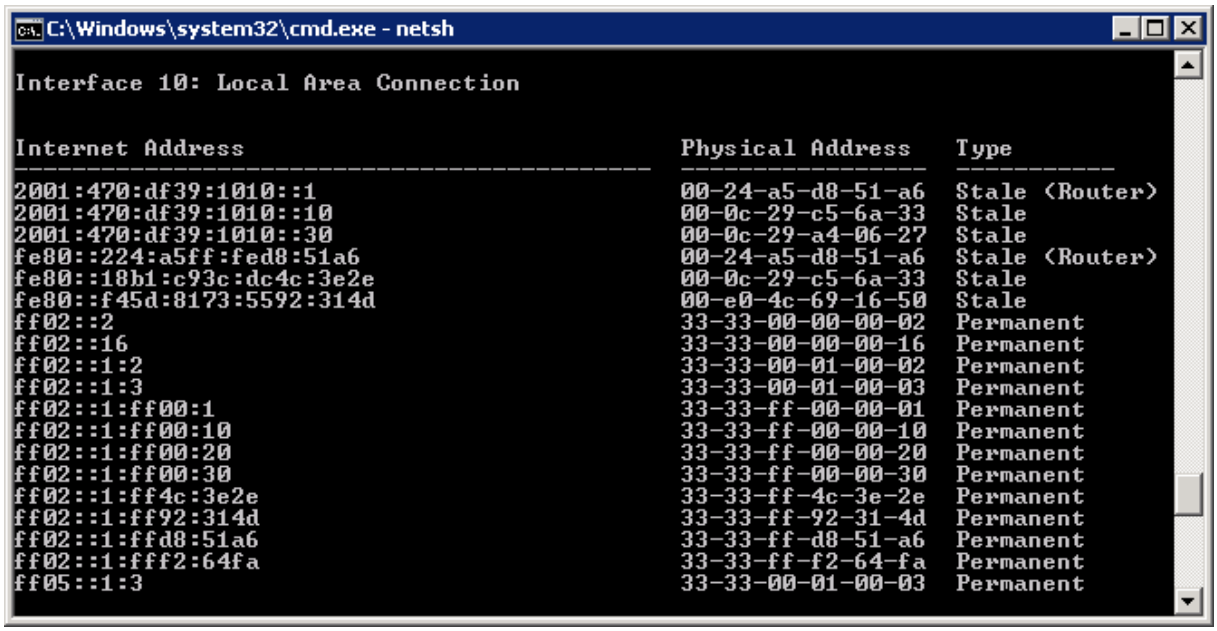
Figur 6.2 Lokala IPv6-adresser på server

Som tidigare nämnts så lyssnar serverroller på multicast-adresser, som är specifika för respektive roll. Servern i fråga är bland annat en DHCP-server vilket syns i figuren nedan då den lyssnar på både ff02::1:2 och ff05:1:3



Figur 6.3 Multicast-adresser som server lyssnar på

Vilka grannar som finns på det lokala nätverkssegmentet visas i figuren nedan



Figur 6.4 Närliggande noder till server

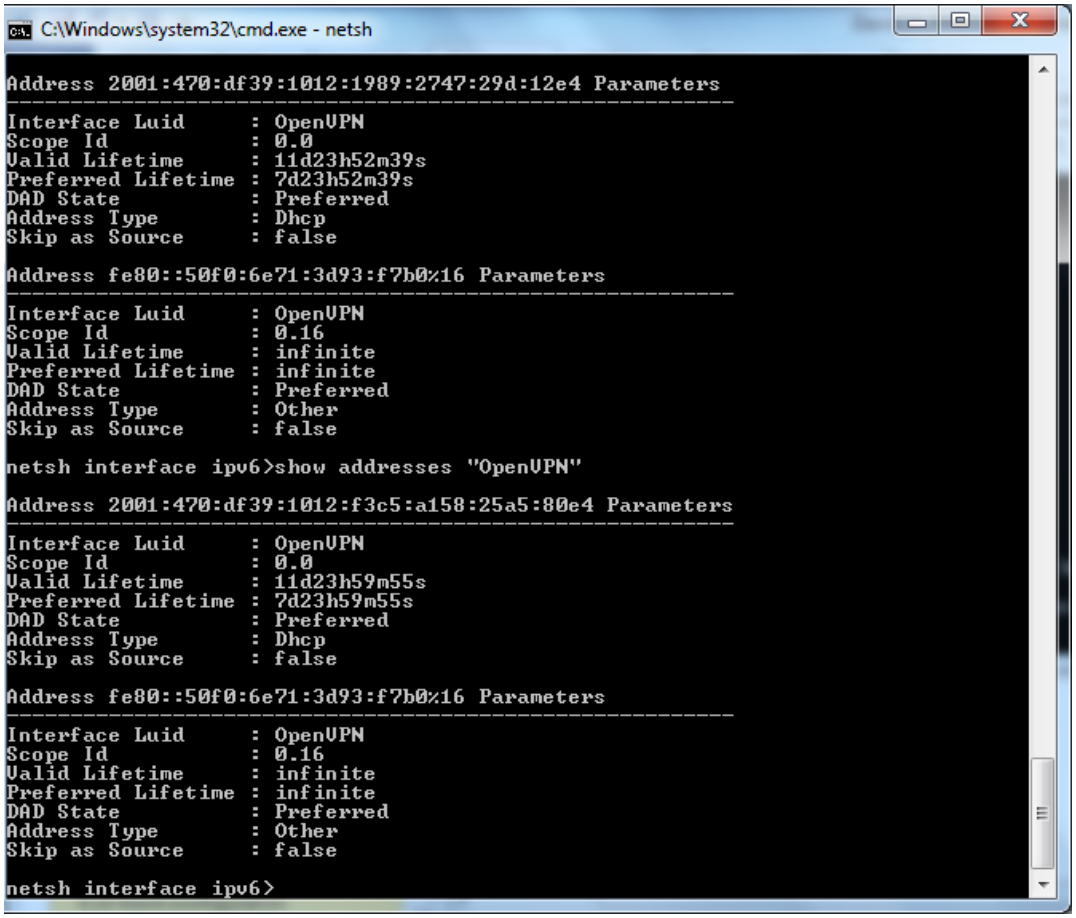
6.5.2 Linux-servrar

Servern som bland annat svarar på adressen ipv6.davidandersson.se har en statisk IPv6-adress satt. Detta är den server som konfigureras att vara publikt åtkomlig via brandväggsregler på portarna för bland annat http och smtp. Nätverkskonfigurationen är lagrad i /etc/networking/interfaces och ser ut enligt nedan.

```
# This file describes the network interfaces available on your
system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
    post-up /usr/sbin/ethtool -s eth0 wol g
    post-down /usr/sbin/ethtool -s eth0 wol g
    address 192.168.30.2
    netmask 255.255.255.0
    gateway 192.168.30.1
    dns-nameservers 8.8.8.8

iface eth0 inet6 static
    address 2001:470:df39:1030::2
    netmask 64
    gateway 2001:470:df39:1030::1
```

6.6 Klient-konfiguration

De ingående klienterna använder både IPv4 och IPv6 och får adresserna tilldelade via DHCP i båda fall. Trots att klienten är inställd att använda Privacy Extensions (se Kapitel 4.6) så får klienten endast en IP-adress, vilket beror på att den tilldelas av just en DHCPv6-server. Detta påverkar ändå inte integritetsaspekten i särskilt stor utsträckning, då en ny adress tilldelas när den gamla har överskridit en viss tidsperiod och visas i figuren nedan.



Figur 6.5 Lokala IPv6-adresser på klient

7 Sammanfattning och resultat

Huvudtanken med uppsatsen har varit att genomgående försöka skapa en grundförståelse och grundförutsättningar för att på ett säkert och medvetet vis införa IPv6. Slutresultatet skulle sedan kunna användas för att stegvis bygga upp mottagarens förståelse så att personen med hjälp av den nyvunna kunskapen kan återskapa den implementation som satts upp. Oavsett om samma hårdvara eller mjukvara används skall mottagaren alltså kunna upprätta ett nätverk som tillhandahåller stöd för både IPv4 och IPv6.

Under arbetet har flertalet frågeställningar och problem uppstått, bland annat gällande den nya kravbilden för ICMPv6-protokollet i IPv6. Tidigare har ICMP endast varit ett stödprotokoll som till stor del kunnat åsidosättas medan det i IPv6 är ett grundläggande krav för att kommunikationen skall fungera. Utöver detta har dock det största problemet berott på det ursprungliga valet av hårdvara som var ämnat att användas i implementationen, då denna gav ett undermåligt stöd för IPv6 och således var tvungen att bytas ut.

7.1 Resultat

Arbetet har resulterat i en fullt fungerande IPv6-implementation där fokus hela tiden har legat på säkerhet. Att tonvikten har legat på säkerheten har lett till att varje större konfigurationsförändring som har orsakat en ny revision av implementationen har säkerhetsvaliderats med hjälp av en checklista. Denna checklista är beskriven i Bilaga B Säkerhetsvalidering och resultatet av den sista valideringen återges i sin helhet i Bilaga B.2 Resultat efter sista konfiguration.

Resultatet har dessutom lyckats uppfylla samtliga kravspecifikationer som satts upp i Tabell 5.2, bland annat med avseende på samexistens av IPv4 och IPv6 samt de krav som funnits på en produktionslik miljö. Utöver detta har rådande rekommendationer följts med avseende på säkerhetshöjande mekanismer från såväl tillverkare som oberoende parter. Däremot har ett avsteg gjorts från rekommendationen att tunnlingsprotokoll inte skall tillåtas, då brandväggen själv kräver denna tunnlingsmekanism. Anledningen till detta är att ingen av de större svenska Internetleverantörerna i nuläget tillhandahåller IPv6 till privatkunder.

7.2 Vidare arbete

För att vidare skapa bättre förutsättningar för en säker miljö bör införandet av SEND (se Kapitel 4.4) utredas och vilka möjligheter som finns för att införa detta i miljön. Utöver detta kan nätverkssäkerheten ytterligare förbättras med hjälp av införandet av intrångsdetekteringssystem på både nätverksnivå och värdnivå.

En annan aspekt av säkerhet är tillgänglighet och för att dessutom skapa ett nätverk med hög tillgänglighet bör installationen dubbleras och konfigureras därefter för att minska den eventuella verksamhetspåverkan som ett hårdvarubortfall skulle innebära.

7.3 Slutsats

Sammantaget kan konstateras att uppsatsens syfte har uppfyllts, då man med grundläggande nätverkskompetens kan utöka sin kunskap inom IPv6-området med hjälp av uppsatsen.

Den tydligaste slutsatsen som kan dras av det löpande arbetet är ändå att säkerheten inom all datorkommunikation främst bygger på kunskap och kännedom om de tekniker som används snarare än inbyggd säkerhet i respektive teknik. Tydligast framgår detta om man sätter IPv4 och IPv6 i en större kontext och ser på vilka hotbilder som finns oberoende av vald teknik.

En teoretisk beskrivning av de grundläggande deltekniker som berörs av IPv6 har gjorts i Kapitel 3 och med hjälp av detta kapitel och Kapitel 4 så får läsaren en god översikt av vad både IPv6 innebär rent tekniskt, men även vilka säkerhetsmekanismer som är relevanta och specifika för IPv6.

Vidare får läsaren en möjlighet att rent praktiskt prova på att arbeta med IPv6 om Kapitel 6 följs och en motsvarande implementation införs. Resonemanget har ju hela tiden byggt på att lösningen skall kunna användas hos privatpersoner eller mindre organisationer.

Uppsatsområdet känns idag mycket relevant och min personliga förhoppning är att det inom en snar framtid fokuseras mer på detta, såväl inom statliga verk och myndigheter som inom den privata marknaden. Och självklart även hos gemene man.

Faktum kvarstår att den här kunskapen kommer att efterfrågas i allt större takt och för att undvika ytterligare en millenium-hysteri så är det av yttersta vikt att den här kunskapen kommer fler till del.

Referenser

[1] J. Arkko och C. Pignataro. *IANA Allocation Guidelines for the Address Resolution Protocol (ARP).* IETF (2009) http://tools.ietf.org/html/rfc5494 Besökt 7 juni 2012

[2] J. Arkko, J. Kempf, B. Zill och P. Nikander. *SEcure Neighbor Discovery.* IETF (2005) http://tools.ietf.org/html/rfc3971 Besökt 7 juni 2012

[3] R. Bonica, D. Gan, D. Tappan och C. Pignataro. *Extended ICMP to Support Multi-Part Messages.* IETF (2007) http://tools.ietf.org/html/rfc4884 Besökt 7 juni 2012.

[4] R. Braden. *Requirements for Internet Hosts – Communication Layers.* IETF (1989): http://tools.ietf.org/html/rfc1122 Besökt 18 maj 2012

[5] K. Chittimaneni, M. Kaeo och E. Vyncke. *Operational Security Considerations for IPv6 Networks.* IETF (2012) http://tools.ietf.org/html/draft-vyncke-opsec-v6-00 Besökt 7 juni 2012.

[6] A. Conta och S. Deering. *Generic Packet Tunneling in IPv6 Specification.* IETF (1998) http://tools.ietf.org/html/rfc2473 Besökt 7 juni 2012

[7] A. Conta, S. Deering och M. Gupta. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.* IETF (2006) http://tools.ietf.org/html/rfc4443 Besökt 18 maj 2012

[8] S. Cheshire, B. Aboba och E. Guttman. *Dynamic Configuration of IPv4 Link-Local Addresses.* IETF (2011) http://tools.ietf.org/html/rfc3927 Besökt 18 maj 2012

[9] Cisco. *IPv6 Deployment Strategies*. Cisco. 4th edition (2002) http://www.cu.ipv6tf.org/pdf/ipv6dswp.pdf Besökt 18 maj 2012

[10] Joseph Davies. *Understanding IPv6.* Microsoft Press, 2nd edition, 2008.

[11] S. Deering och R. Hinden. *Internet Protocol Version 6 (IPv6) Addressing Architecture* IETF (2003) http://tools.ietf.org/html/rfc3513 Besökt 18 maj 2012

[12] S. Deering och R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*. IETF (1998) http://tools.ietf.org/html/rfc2460 Besökt 18 maj 2012

[13] S. Deering och R. Hinden. *IP Version 6 Addressing Architecture.* IETF (2006) http://tools.ietf.org/html/rfc4291 Besökt 18 maj 2012

[14] S. Deering, R. Hinden och E. Nordmark. *IPv6 Global Unicast Address Format.* IETF (2003) http://tools.ietf.org/html/rfc3587 Besökt 18 maj 2012

[15] L. Delgrossi och L. Berger. *Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+*. IETF (1995) http://tools.ietf.org/html/rfc1819 Besökt 18 maj 2012

[16] S. Frankel, R. Gravenman, J. Pearce och M. Rooks. *Guidelines for the Secure Deployment of IPv6. NIST (2010)* http://csrc.nist.gov/publications/nistpubs/800-119/sp800-119.pdf Besökt 18 maj 2012

[17] F. Gont. *Security Assessment of the Internet Protocol Version 4*. IETF (2011) http://tools.ietf.org/html/rfc6274 Besökt 18 maj 2012

[18] F. Gont. *Implementation Advice for IPv6 Router Advertisement Guard(RA-Guard).* IETF (2012) http://tools.ietf.org/id/draft-gont-v6ops-ra-guard-implementation-01.txt Besökt 18 maj 2012

[19] F. Gont. S*ecurity Implications of Predictable Fragment Identification Values.* IETF (2012) http://tools.ietf.org/id/draft-gont-6man-predictable-fragment-id-01.txt Besökt 18 maj 2012

[20] Gary Govanius. *TCP/IP 24sju.* Pagina förlags AB, 2000.

[21] R. Hinden och B. Haberman. *Unique Local IPv6 Unicast Addresses.* IETF (2005) http://tools.ietf.org/html/rfc4193 Besökt 18 maj 2012

[22] Scott Hogg och Eric Vyncke. *IPv6 Security.* Cisco Press, 2008.

[23] Juniper Networks. *Deploying IPv6 : Issues and Strategies.* Juniper Networks (2009) http://www.iec.org/newsletter/february09_1/juniperwhitepaper.pdf Besökt 18 maj 2012

[24] Charles M. Kozierok. *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference.* No Starch Press, 2005.

[25] E. Levy-Abegnoli, G. Van de Velde, C. Popoviciu och J. Mohacsi. *IPv6 Router Advertisement Guard.* IETF (2011) http://tools.ietf.org/html/rfc6105 Besökt 18 maj 2012

[26] Qing Li, Tatuya Jinmei och Keiichi Shima. *IPv6 Advanced Protocols Implementation.* Morgan Kaufmann, 2007.

[27] T. Narten, E. Nordmark och W. Simpson. *Neighbor Discovery for IP Version 6 (IPv6).* IETF (1998) http://tools.ietf.org/html/rfc2461 Besökt 18 maj 2012

[28] T. Narten, R. Draves och S. Krishnan. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6.* IETF (2007) http://tools.ietf.org/html/rfc4941 Besökt 18 maj 2012

[29] Netfilter. *The Netfilter.org Project.* http://iptables.org/ Besökt 7 juni 2012.

[30] J. Postel. *INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION* . IETF (1981) http://tools.ietf.org/html/rfc791 Besökt 18 maj 2012

A Installationsförfarande

Versionsinformation

Löpnr	Kortfattad beskrivning/förändring	Datum	Utfört av
1	Dokumentet skapat	2012-03-31	David A

1 Bakgrund

Följande dokument beskriver installationsförfarandet för kil-n001.hogbacken.local.

2 Hårdvara

Hårdvaran som ligger till grund för kil-n001 är en Buffalo Linkstation WZR-HP-G300NH[4], då det är en av de hårdvaror som rekommenderas av OpenWRT.

3 Operativsystem

Underliggande operativsystem som valts är OpenWRT 10.03.1 då denna har stöd för både VLAN, multipla SSID, VPN och tillpassning med övrig intressant mjukvara.

4 Anpassning av OpenWRT

Installationen av OpenWRT är väldigt enkel och man kan använda webbgränssnittet för att uppgradera firmware. I webbkatalogen http://downloads.openwrt.org/backfire/10.03.1/ar71xx/ finns den firmware tillgänglig som valts för installationen.

4.1 Stöd för USB

För att kunna installera mer mjukvara än vad som begränsas av det interna minnet så skall ett USB-minne anslutas. För att detta skall kunna göras måste ett antal paket installeras.

```
$ opkg update
$ opkg install kmod-usb-uhci kmod-usb-storage kmod-usb-storage-
extras block-extroot block-mount block-hotplug kmod-fs-vfat kmod-fs-
ext2 kmod-fs-ext3 fdisk e2fsprogs
```

För att kunna använda multipla usb-enheter har en usb-hubb anslutits. För att OpenWRT skall klara av att hantera flera usb-enheter måste en förändring göras i filen /etc/modules.d/60-usb-storage. Följande rad skall läggas till längst ner i filen.

```
max_scsi_luns=8
```

Då detta är gjort krävs en omstart.

```
$ reboot
```

När omstarten är färdig kan konfigurationen kontrolleras med hjälp av följande kommando.

```
$ dmesg
```

Förbered sedan usb-minnet genom att skapa en ext3-partition.

```
$ fdisk /dev/sda
```

Förbered därefter denna partition med följande kommando.

```
$ mkfs.ext3 /dev/sda1
```

[4] http://wiki.openwrt.org/toh/buffalo/wzr-hp-g300h

Sedan behöver det befintliga systemet kopieras över till usb-enheten.

```
$ mkdir /mnt/sda1
$ mount /dev/sda1 /mnt/sda1
$ tar -C /overlay -cvf - . | tar -C /mnt/sda1 -xf -
```

Uppdatera filen /etc/config/fstab så att den ser ut enligt nedan.

```
config global automount
        option from_fstab 1
        option anon_mount 1

config global autoswap
        option from_fstab 1
        option anon_swap 0

config mount
        option target   /mnt/sda1
        option device   /dev/sda1
        option fstype   ext3
        option options  rw,sync
        option enabled  1
        option enabled_fsck 0
        option is_rootfs        1
```

4.2 Konfigurationer

4.2.1 Nätverk

Nätverket som sätts upp konfigureras med fyra olika subnät kopplade till de fyra portarna på switchen samt två extra subnät till vpn-nätverket och det trådlösa gästnätverket.
Konfigurationsfilen /etc/config/network ser ut som följer.

Port 1 -> Securenet (192.168.50.0/24)
Port 2-> DMZ (192.168.30.0/24)
Port 3->Clients (192.168.11.0/24)
Port 4->Servers (192.168.10.0/24)

Konfigurationen finns lagrad i /etc/config/network.

```
config 'interface' 'loopback'
        option 'ifname' 'lo'
        option 'proto' 'static'
        option 'ipaddr' '127.0.0.1'
        option 'netmask' '255.0.0.0'

config 'interface' 'securenet'
        option 'ifname' 'eth0.4'
        option 'type' 'bridge'
        option 'proto' 'static'
        option 'ip6addr' '2001:470:df39:1050::1/64'
        option 'ipaddr' '192.168.50.1'
        option 'netmask' '255.255.255.0'

config 'interface' 'dmz'
        option 'ifname' 'eth0.3'
        option 'type' 'bridge'
        option 'proto' 'static'
        option 'ip6addr' '2001:470:df39:1030::1/64'
        option 'ipaddr' '192.168.30.1'
        option 'netmask' '255.255.255.0'

config 'interface' 'clients'
        option 'ifname' 'eth0.2'
        option 'type' 'bridge'
        option 'proto' 'static'
        option 'ip6addr' '2001:470:df39:1011::1/64'
        option 'ipaddr' '192.168.11.1'
        option 'netmask' '255.255.255.0'

config 'interface' 'servers'
        option 'ifname' 'eth0.1'
        option 'type' 'bridge'
        option 'proto' 'static'
        option 'ip6addr' '2001:470:df39:1010::1/64'
        option 'ipaddr' '192.168.10.1'
        option 'netmask' '255.255.255.0'

config 'interface' 'guests'
        option 'proto' 'static'
        option 'type' 'bridge'
        option 'ip6addr' '2001:470:df39:1020::1/64'
        option 'ipaddr' '192.168.20.1'
        option 'netmask' '255.255.255.0'

config 'interface' 'clientvpn'
        option 'ifname' 'tap0'
        option 'type' 'bridge'
```

```
        option 'proto' 'static'
        option 'ip6addr' '2001:470:df39:1012::1/64'
        option 'ipaddr' '192.168.12.1'
        option 'netmask' '255.255.255.0'

config 'interface' 'wan'
        option 'ifname' 'eth1'
        option 'proto' 'dhcp'

config 'switch'
        option 'name' 'switch0'
        option 'reset' '1'
        option 'enable' '1'
        option 'enable_vlan' '1'

config 'switch_vlan' 'eth0_1'
        option 'device' 'switch0'
        option 'vlan' '1'
        option 'ports' '0 5t'

config 'switch_vlan' 'eth0_2'
        option 'device' 'switch0'
        option 'vlan' '2'
        option 'ports' '1 5t'

config 'switch_vlan' 'eth0_3'
        option 'device' 'switch0'
        option 'vlan' '3'
        option 'ports' '2 5t'

config 'switch_vlan' 'eth0_4'
        option 'device' 'switch0'
        option 'vlan' '4'
        option 'ports' '3 5t'

config 'interface' 'henet'
        option 'proto' '6in4'
        option 'peeraddr' '216.66.80.90'
        option 'ip6addr' '2001:470:27:aa5::2/64'
        option 'tunnelid' '**********'
        option 'username' '**********'
        option 'password' '**********'
```

4.2.2 Trådlöst nätverk

Det finns två trådlösa nätverk tillgängliga, hbclients och hbguest. hbclients är bryggat med klientnätverket och hbguest med gästnätverket.

Konfigurationen finns lagrad i /etc/config/wireless.

```
config wifi-device  radio0
        option type     mac80211
        option channel  11
        option macaddr  00:24:a5:d8:51:a6
        option hwmode   11ng
        option htmode   HT20
        list ht_capab   SHORT-GI-40
        list ht_capab   DSSS_CCK-40
        # REMOVE THIS LINE TO ENABLE WIFI:
        # option disabled 1

config wifi-iface
        option device   radio0
        option network  clients
        option mode     ap
        option ssid     hbclients
        option encryption psk2
        option key      *********

config wifi-iface
        option device   radio0
        option network  guests
        option mode     ap
        option ssid     hbguests
        option encryption psk2
        option key      **********
```

4.2.3 Brandvägg

Brandväggens konfiguration ligger lagrad i /etc/config/firewall.

I filen /etc/firewall.user finns extra regler som skrivs direkt till iptables.

4.2.4 System

I filen /etc/config/system regleras både namn och tidszon för routern.

```
config 'system'
        option 'zonename' 'Europe/Stockholm'
        option 'timezone' 'CET-1CEST,M3.5.0,M10.5.0/3'
        option 'hostname' 'kil-n001.hogbacken.local'
```

4.2.5 Slå av dnsmasq och telnet

Slå av dnsmasq med tanke på att både en dhcp-server och en dns-server används utanför brandväggen. Slå även av telnet.

```
§ /etc/init.d/dnsmasq stop
§ /etc/init.d/dnsmasq disable
§ /etc/init.d/telnet stop
§ /etc/init.d/telnet disable
```

4.2.6 Lokal hosts-fil

Uppdatera filen /etc/hosts med information om kil-s001, kil-s002 och kil-s003.

```
192.168.10.10       kil-s001.hogbacken.local
```

```
192.168.10.20       kil-s002.hogbacken.local
192.168.10.30       kil-s003.hogbacken.local
127.0.0.1           localhost
```

4.3 IPv6-stöd

Konfigurationen för adresser som är gjorda i 4.2.1 förutsätter att följande kapitel redan är genomgånget.

4.3.1 Nödvändiga paket

För att kunna använda IPv6 i brandväggen så har följande paket lagts till:

```
$ opkg install ip ip6tables kmod-sit kmod-iptunnel6 radvd iputils-
traceroute6 dibbler-relay
```

4.3.2 Radvd

Radvd är den implementation som används i OpenWRT för att skicka Router Advertisements och Router Solicitations. Paketet konfigureras med filen /etc/config/radvd.

För varje gränssnitt som skall skicka Router Advertisements så konfigureras dessa med en definition enligt nedan.Varje rad definierar sedan konfigurationsparametrar för respektive gränssnitt.

Då en extern DHCPv6-server skall användas är både AdvOtherConfigFlag och AdvManagedFlag satt till 1 för dessa interfaces.

```
config 'interface'
        option 'AdvSendAdvert' '1'
        option 'ignore' '0'
        option 'interface' 'clients'
        option 'IgnoreIfMissing' '1'
        option 'AdvManagedFlag' '1'
        option 'AdvOtherConfigFlag' '1'
        option 'AdvSourceLLAddress' '1'
        option 'AdvDefaultPreference' 'medium'
```

4.3.3 Dibbler-relay

För att vidaresända DHCPv6-förfrågningar till DHCPv6-servern som är installerad på en intern server konfigureras paketet Dibbler-relay för varje nätverkssegment som skall få IPv6-adresser tilldelade över DHCP.

Vad detta innebär är att brandväggen lyssnar på multicastadressen ff02::1:2 som är adressen för samtliga dhcp-servrar och dhcp-relays på det lokala nätverkssegmentet. Sedan vidarebefordras dhcp-begäran till Windows-servern kil-s002 för tilldelning av adress.

Detta innebär att ett gränssnitt måste definieras som en server, enligt nedan.

```
iface br-servers {
        server multicast
}
```

Utöver denna konfiguration behöver även varje gränssnitt som skall tilldelas adresser via DHCP definieras, enligt nedan.

```
iface br-clients {
        client multicast yes
        interface-id 10
}
```

4.4 Tillagda paket

4.4.1 Dhcp-forwarder

Paketet läggs till med följande kommando:

```
$ opkg install dhcp-forwarder
```

I filen /etc/dhcp-fwd.conf ställ in följande under ”Interface settings”.

```
if      br-guests       true    false   true
if      br-clients      true    false   true
if      br-vpnbridge   true    false   true
if      br-securenet   true    false   true
if      br-servers      false   true    true
```

Längre ner under Server settings, ställ in följande:

```
server ip       192.168.10.20
```

För att se till att dhcp-forwarder startas vid boot, kör följande kommando

```
§ /etc/init.d/dhcp-fwd enable
§ /etc/init.d/dhcp-fwd start
```

4.4.2 Openvpn

Paketet läggs till med följande kommando:

```
$ opkg install openvpn openvpn-easy-rsa
```

Redigera filen /etc/easy-rsa/vars och uppdatera följande variabler:

```
export KEY_COUNTRY="SE"
export KEY_PROVINCE="Vaermland"
export KEY_CITY="Kil"
export KEY_ORG="********"
export KEY_EMAIL=info@**********.se
```

Generera sedan nycklar.

```
§ build-ca
§ build-dh
§ build-key-server kil-n001
§ build-key <<client_name>>
```

Konfigurera sedan serverns konfiguration i /etc/config/openvpn. Under den instans som skall konfigureras, se till att följande variabler är satta:

```
    option enable 1
    option mode server
    option tls_server 1
    option port 1194
    option proto udp
    option dev tap0
    option ca /etc/easy-rsa/keys/ca.crt
    option cert /etc/easy-rsa/keys/kil-n001.crt
```

```
    option key /etc/easy-rsa/keys/kil-n001.key
    option dh /etc/easy-rsa/keys/dh1024.pem
    list push "redirect-gateway def1"
    option client_to_client 1
    option keepalive "10 120"
    option comp_lzo 1
    option max_clients 10
    option persist_key 1
    option persist_tun 1
    option status /var/log/openvpn-status.log
    option verb 3
    option auth_user_pass_verify "/etc/yubi/otp_auth.sh via-file"
    option script_security 3
```

Skapa sedan filen /etc/init.d/vpn_bridge och fyll den med följande innehåll:

```
#!/bin/sh /etc/rc.common
#/etc/init.d/openvpn
START=35
STOP=65

start() {
        openvpn --mktun --dev tap0
        ifconfig tap0 0.0.0.0 promisc up
}

stop() {
        ifconfig tap0 down
}
```

Utöver detta behöver filerna /etc/yubi/otp_auth.sh och /etc/yubi/yubi_user.conf skapas. Otp_auth.sh skall se ut enligt nedan (notera att ID skall bytas till ett giltigt ID hos Yubico).

```
#!/bin/sh
 USERS=`cat /etc/yubi/yubi_user.conf`
 ID=1234

 vpn_verify() {
     if [[ ! $1 ]] || [[ ! $2 ]]; then
         #echo "No username or password: $*"
         exit 1
     fi

     ## it can also be done with grep or sed
     for i in $USERS; do
        if [ `echo "$1:$2" |grep "$i"` ]; then
                if wget -O -
"http://api.yubico.com/wsapi/verify?id=$ID&otp=$2" 2> /dev/null |
grep status=OK > /dev/null
                then
                        exit 0
                fi
                exit 1
        fi
     done
 }

 if [[ ! $1 ]] || [[ ! -e $1 ]]; then
     # echo "No file"
     exit 1
 fi
```

```
## $1 is file name which contains
## passed username and password
vpn_verify `cat $1`
exit 1
```

Filen yubi_user.conf skall fyllas med den publika delen av de genererade OTP-strängarna och skall konfigureras med följande struktur:

```
User1:abcdefghijkl
User2:abcdefghijkl
```

Se sedan till att openvpn och den nyligen skapade vpn_bridge körs då routern startas.

```
§ /etc/init.d/vpn_bridge enable
§ /etc/init.d/vpn_bridge start
§ /etc/init.d/openvpn enable
§ /etc/init.d/openvpn start
```

4.4.3 Openssh-sftp-server

Paketet läggs till med följande kommando:

```
$ opkg install openssh-sftp-server
```

4.4.4 Tcpdump-mini

Paketet läggs till med följande kommando:

```
$ opkg install tcpdump-mini
```

4.4.5 Etherwake

Paketet läggs till med följande kommando:

```
$ opkg install etherwake
```

4.4.6 Ipsec-tunnel

För att etablera en site-to-site-tunnel över IPsec behöver följande paket installeras:

```
$ opkg install ipsec-tools kmod-crypto-authenc kmod-ipsec kmod-
ipsec4 ip openssl-util
```

När dessa paket är installerade behöver script läggas till i mappen /etc/init.d samt /etc/racoon/. Dessa script används för att etablera tunnlarna samt för att konfigurera brandväggen så att den tillåter trafik över dessa tunnlar. Den senaste versionen finns tillgänglig på http://wiki.openwrt.org/doc/howto/vpn.ipsec.basics.

Varje tunnel som ska etableras konfigureras sedan i filen /etc/config/racoon. Konfigurationsfilen börjar med följande konfiguration som är gemensam för servern.

```
config 'racoon'
        option 'foreground' '0'
        option 'zone' 'vpn'
        option 'debug' '0'
        list   'listen' 'wan'
```

Efter det kommer specifik konfiguration för varje tunnel.

```
config 'tunnel' 'tunnel_ett'
        option 'enabled' '1'
        option 'remote' 'tunnelett.homeip.net'
        option 'pre shared key' 'presharedkey'
        option 'exchange_mode' 'aggressive'
        option 'my_identifier' 'myidentifier'
        list   'p1_proposal' 'pre_g2_3des_md5'
        list   'sainfo' 'tunnel_ett_sa'

config 'sainfo' 'tunnel_ett_sa'
        option 'local_subnet' '192.168.0.0/18'
        option 'remote_subnet' '192.168.64.0/18'
        option 'p2_proposal' 'g2_3des_md5'

config 'p1_proposal' 'pre_g2_3des_md5'
        option 'encryption_algorithm' '3des'
        option 'lifetime' '28800'
        option 'hash_algorithm' 'md5'
        option 'authentication_method' 'pre_shared_key'
        option 'dh_group' '2'

config 'p2_proposal' 'g2_3des_md5'
        option 'pfs_group' '2'
        option 'lifetime' '3600'
        option 'encryption_algorithm' '3des'
        option 'authentication_algorithm' 'hmac_md5'
```

4.4.7 Ddns-scripts

Paketet läggs till med följande kommando:

```
$ opkg install ddns-scripts luci-app-ddns
```

Se till att variablerna i konfigurationsfilen /etc/config/ddns är satta enligt nedan:

```
config service "hogbacken"
        option enabled          "1"

        option service_name     "dyndns.org"
        option domain           "localdns.homeip.net"
        option username         "*********"
        option password         "*********"

        option ip_source        "network"
        option ip_network       "wan"


        option force_interval   "72"
        option force_unit       "hours"
        option check_interval   "10"
        option check_unit       "minutes"
```

B Säkerhetsvalidering

En säkerhetsvalidering av implementationen har skett efter varje förändring av konfigurationer. Detta vid såväl konfigurationsförändringar av nätverksdesign, tillagda tjänster eller nya åtkomstregler vad gäller brandväggsåtkomst.

Säkerhetsvalideringen har följt en checklista med ett antal steg och då ett avsteg från förväntat resultat påträffats har detta åtgärdats, varpå säkerhetsvalideringen på nytt inletts från steg 1.

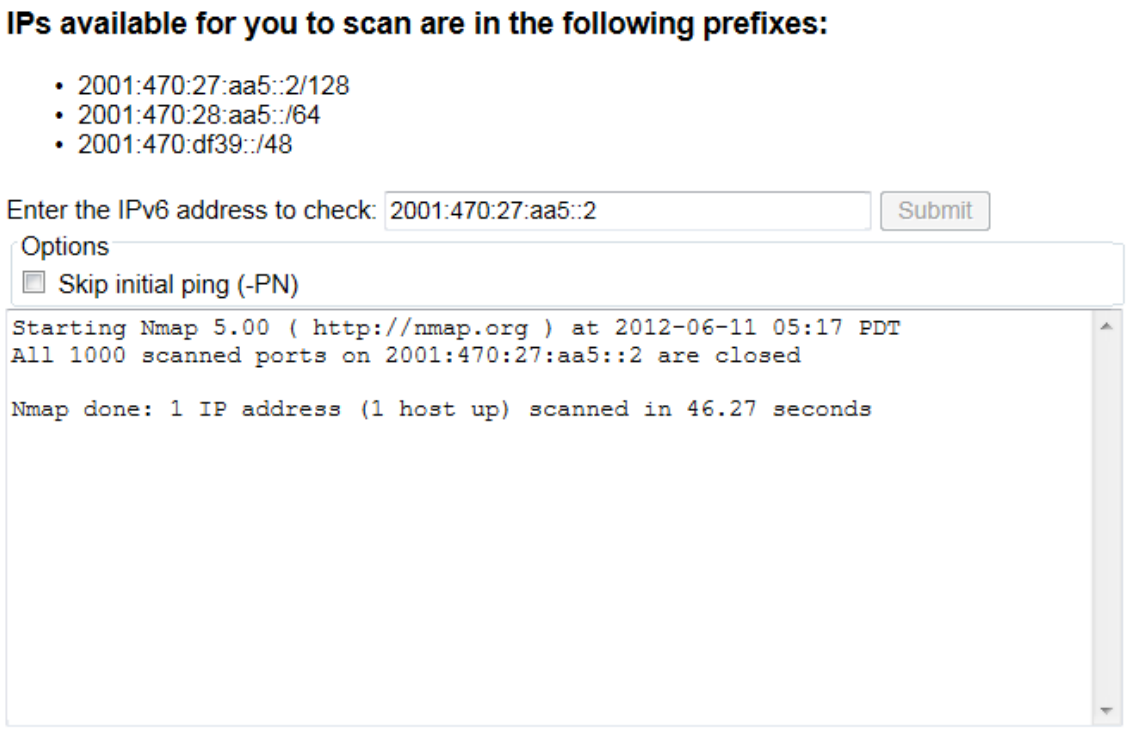
B.1 Checklista

Valideringskontroll	Förväntat resultat	OK/Ej Ok
1. Säkerställ att routern inte exponerar några tjänster mot Internet.	Inga tjänster exponerade, verifieras med portscanning mot routerns externa IPv6-adress.	
2. Säkerställ att routern endast exponerar godkända tjänster på interna nätverk.	Klienter, servrar och vpn skall komma åt allt på router. Övriga subnät skall endast kunna komma åt router över ICMPv6.	
3. Säkerställ att endast specifika tjänster är åtkomliga på publicerad server.	Endast SMTP och HTTP tillgängliga över IPv6, verifieras med portscanning mot publicerad servers externa IPv6-adress	
4. Säkerställ att samtlig mjukvara på routern är uppdaterad till senaste version.	Kör kommandot ”opkg upgrade” med genomförd uppgradering.	
5. Säkerställ att samtlig mjukvara på publicerad server är uppdaterad till senaste verision.	Kör kommandont ”apt-get update; apt-get upgrade” med genomförd uppgradering.	
6. Säkerställ att kommunikation fungerar över samtliga nätverk.	Klienter får DHCPv6-adress tilldelad, kan kommunicera mot interna resurser över IPv6 och kan kommunicera mot Internet över IPv6, verifieras exempelvis mot http://www.kame.net/	

B.2 Resultat efter sista konfiguration

Valideringskontroll	Förväntat resultat	OK/Ej Ok
1. Säkerställ att routern inte exponerar några tjänster mot Internet.	Inga tjänster exponerade, verifieras med portscanning mot routerns externa IPv6-adress.	OK, se B.2.1
2. Säkerställ att routern endast exponerar godkända tjänster på interna nätverk.	Klienter, servrar och vpn skall komma åt allt på router. Övriga subnät skall endast kunna komma åt router över ICMPv6.	OK.
3. Säkerställ att endast specifika tjänster är åtkomliga på publicerad server.	Endast SMTP och HTTP tillgängliga över IPv6, verifieras med portscanning mot publicerad servers externa IPv6-adress	OK, se B.2.2
4. Säkerställ att samtlig mjukvara på routern är uppdaterad till senaste version.	Kör kommandot ”opkg upgrade” med genomförd uppgradering.	OK.
5. Säkerställ att samtlig mjukvara på publicerad server är uppdaterad till senaste verision.	Kör kommandont ”apt-get update; apt-get upgrade” med genomförd uppgradering.	OK.
6. Säkerställ att kommunikation fungerar över samtliga nätverk.	Klienter får DHCPv6-adress tilldelad, kan kommunicera mot interna resurser över IPv6 och kan kommunicera mot Internet över IPv6, verifieras exempelvis mot http://www.kame.net/	OK.

B.2.1 Extern portscanning mot router



B.2.2 Extern portscanning mot publicerad server

IPs available for you to scan are in the following prefixes:

- 2001:470:27:aa5::2/128
- 2001:470:28:aa5::/64
- 2001:470:df39::/48

Enter the IPv6 address to check: 2001:470:df39:1030::2 Submit

Options

Skip initial ping (-PN)

```
Starting Nmap 5.00 ( http://nmap.org ) at 2012-06-11 05:22 PDT
Interesting ports on mail.ictsecurity.se (2001:470:df39:1030::2):
Not shown: 998 closed ports
PORT   STATE    SERVICE
25/tcp filtered smtp
80/tcp filtered http

Nmap done: 1 IP address (1 host up) scanned in 8.14 seconds
```