secorvo
security consulting

IPv6

Secorvo White Paper

Die grundlegenden Funktionen, Bedrohungen und Maßnahmen

Version 1.3
Stand 11. Februar 2014

Dr. Safuat Hamdy

Secorvo Security Consulting GmbH
Ettlinger Straße 12-14
D-76137 Karlsruhe

Tel. +49 721 255171-0
Fax +49 721 255171-100

info@secorvo.de
www.secorvo.de

The Internet Is Full... Go Away!

Inhaltsübersicht

1 Einleitung

IPv6 ist das Netzwerkprotokoll, das als Nachfolger für das bisher genutzte IPv4 entwickelt wurde. Die Aufgabe des Internet-Protokolls (IP) besteht im Wesentlichen darin, Datenpakete von einem System über verschiedene Netzwerke hinweg zu einem Zielsystem zu vermitteln. Diese Vermittlung erfolgt anhand von IP-Adressen. Ohne eine gültige IP-Adresse kann ein System nicht am Internet teilnehmen.

Mit der Aufzehrung des IPv4-Adressraumes im Jahr 2012, in welchem IANA ihren letzten /8-Block vergeben hat, wurde ein wichtiger „Meilenstein“ auf dem Weg zur Einführung von IPv6 erreicht. Zwar verfügen die regionalen Internetregistrierungsstellen noch über freie Adressblöcke, jedoch werden auch diese Adressblöcke in absehbarer Zeit vergeben sein.[1] IPv6 wurde u. a. als Antwort auf die sich abzeichnende Adressknappheit entwickelt; die frühesten Anfänge gehen in die Jahre 1992 und 1993 zurück, eine Standardisierung durch die IETF erfolgte 1998 [RFC 2460]. Es folgten eine Testphase sowie eine Phase der Konsolidierung, die ihren vorläufigen Abschluss etwa im Jahr 2011 hatte.[2]

Vor diesem Zeitpunkt wurde IPv6 im Wesentlichen als Konstruktion aus dem Elfenbeinturm angesehen, und Verweise auf eine zukünftige Einführung von IPv6 wurden als rein akademische Betrachtung abgetan. Mit anderen Worten, IPv6 war nichts, was eine ernsthafte Betrachtung wert gewesen wäre. Doch auch danach wurde IPv6 weitgehend ignoriert – Hersteller und Provider verwiesen gegenseitig auf fehlende Unterstützung in Produkten und fehlende Angebote von IPv6-Diensten.

Gleichzeitig wurde die Adressknappheit bei IPv4 jedoch immer offensichtlicher. Verschiedene Provider wie auch verschiedene Hersteller haben in der Zwischenzeit erkannt, dass die Auseinandersetzung mit IPv6 nicht nur unvermeidlich ist, sondern eine günstige Ausgangsposition auf einem neu entstehenden Markt verschafft. In der Folge nahm das Angebot an IPv6-fähigen Netzwerkgeräten, an IPv6-Anbindungen sowie an über IPv6 angebotenen Inhalten im Internet zu. Einschlägige Interessengruppen wie die Internet Society haben darüber hinaus 2011 den World IPv6 Day und 2012 den World IPv6 Launch Day veranstaltet. Der IPv6 Day war im Wesentlichen ein Awareness-Event, bei dem es um den Nachweis ging, dass IPv6 auf globaler Ebene einsatzbereit ist; hierzu haben die teilnehmenden Organisationen einen Tag lang Ihre Dienste auch über IPv6 angeboten. Beim IPv6 Launch Day ging es darum, diese Angebote dauerhaft über IPv6 verfügbar zu machen.

Die Folge ist ein derzeit stärker als zuvor wachsender Markt an Geräten und Angeboten rund um IPv6.[3] So bieten verschiedene Provider IPv6 mittlerweile sogar für Endkunden an. Dies alles macht deutlich, dass IPv6 endgültig in der Praxis ankommt. Auch Unternehmen und Organisationen aus dem Kreis der Internet-Nutzer nähern sich dem Punkt, an dem eine ernsthafte Auseinandersetzung mit IPv6 unausweichlich ist. Es wird von der Professionalität und der Vorausschau des jeweiligen IT-Managements abhängen, ob sich dieser Punkt als Meilenstein einer betrieblichen Entwicklung oder als Klippe am Abgrund erweist.

---

1 Siehe beispielsweise die RIPE-Ankündigung vom 14.09.2012.

2 Als dafür sichtbaren Meilensteine können die RFCs 6204 *IPv6 CE Router Requirements* sowie 6434 *IPv6 Node Requirements* angesehen werden ebenso die Dokumente ripe-501 und dessen (derzeit gültiger) Nachfolger ripe-554 [RIPE 554].

3 Unter http://www.vyncke.com/ipv6status erhält man eine länderspezifische Übersicht über die derzeit über IPv6 angebotenen Dienste, vergebenen Präfixe usw. Zu den bekannteren Anbietern von allgemein populären Web-Inhalten über IPv6 gehören beispielsweise Bing, Google, Yahoo, Youtube, Wikipedia und Facebook.

Es besteht in Fachkreisen ein allgemeiner Konsens darüber, dass eine unternehmensweite Umstellung auf IPv6 bezüglich des zu erwartenden Aufwands in etwa mit der Euro-Einführung oder der Vorbereitung auf das Jahr-2000-Problem vergleichbar ist; ein Vergleich mit der Umstellung von Windows XP auf beispielsweise Windows 7 ist ebenfalls angemessen. Zwar ist kein Stichtag vorgesehen, zu dem die Umstellung global erfolgt, jedoch wird für jede Organisation der individuelle Punkt kommen, an dem die Umstellung unvermeidlich ist. Der Versuch, mit mehrstufigem NAT auszuweichen, wird die Umstellung nicht verhindern, sondern nur hinauszögern und verteuern.

Da der Aufbau der Datenpakete bei IPv6 vollkommen inkompatibel zu IPv4 ist und IPv6 einige neue Konzepte mitbringt, ist es in jedem Fall notwendig, vor einer Einführung ausreichend Fachkenntnis zu erwerben und Erfahrungen zu sammeln. Mit anderen Worten, die Hoffnung, IPv6 mit minimalem Aufwand etwa im Rahmen einer Wochenendaktion einzuführen, wird sehr wahrscheinlich an der Komplexität der Aufgabe zerschellen. Zur Erleichterung des Übergangs wurden verschiedene Mechanismen entwickelt, wie beispielsweise Dual-Stack-Betrieb, Tunnelung oder Protokollübersetzung. Dennoch bleibt der Übergang kein leichtes Unterfangen, sondern erfordert ein Mindestmaß an Vorbereitung und Planung, was seinerseits eine gewisse Kenntnis und Erfahrung voraussetzt.

Dieser Artikel zeigt die konkreten Sicherheitsrisiken und Bedrohungen auf, die sich bei der Nutzung von IPv6 ergeben, und zeigt verschiedene Maßnahmen, um diesen Bedrohungen zu begegnen. Gegenstand dieses Artikels sind die Grundfunktionen von IPv6. Weiterführende Mechanismen wie Übergangsmechanismen, Multicast, Mobile IPv6 oder Multihoming erfordern eine separate Beschreibung und Diskussion.

Auch IPsec wird in diesem Artikel nur am Rande behandelt, obwohl IPsec – ursprünglich – als integraler Bestandteil von IPv6 entworfen wurde. Das liegt daran, dass IPsec bereits zu IPv4 zurück portiert wurde und deswegen viel stärker als IPv6 etabliert ist. Tatsächlich ist an IPsec technisch gesehen nichts IPv6-Spezifisches, so dass IPsec hier nur als Mechanismus verstanden wird, der zur Absicherung von IPv6 herangezogen werden kann.

Dieser Artikel ist für ein White Paper relativ umfangreich, was daran liegt, dass IPv6 noch nicht weit verbreitet ist. Es ist daher damit zu rechnen, dass viele Leser erst über rudimentäre Kenntnisse über IPv6 verfügen. Mit Abschnitt 2 wird der Versuch unternommen, die für die folgenden Abschnitte relevanten Details einzuführen; Leser mit ausreichenden Kenntnissen der IPv6 Funktionalitäten mögen diesen Abschnitt überspringen.

IPv6 bringt zudem reichlich neue Terminologie mit sich, die teilweise auch für den IPv4-erfahrenen Administrator neu ist. Daher wurde am Ende dieses Artikels ein umfangreiches Glossar erstellt, in dem alle relevanten Begriffe und Akronyme kurz erklärt werden. Die für das Verständnis zentralen Begriffe werden nachstehend zusammengefasst.

1.1 Terminologie

Mit *Internet4* wird der Teil des gesamten Internets bezeichnet, der über IPv4 erreichbar ist; entsprechend bezeichnet *Internet6* den Teil des Internets, der über IPv6 erreichbar ist.

Gegenstand dieses Artikels ist IPv6, bei dem Datenpakete zwischen Nodes ausgetauscht werden. Ein IP-*Paket* besteht aus einem *Header* und – üblicherweise – einer *Nutzlast* (*Payload*). Die Nutzlast wird durch das Format des sogenannten *Upper-Layer Protocols* (*ULP*) bestimmt, typischerweise handelt es sich dabei um ICMP oder ein Transportschicht-Protokoll wie TCP, UDP oder SCTP; im Fall von Tunnelung kann es sich um ein Netzwerkprotokoll wie IPv6 selbst oder IPv4 handeln. Der Header des Upper-Layer Protocols wird als *Upper-Layer Header* bezeichnet.

Ein *Node* ist ein Gerät, das über ein oder mehrere Interfaces an ein oder mehrere Netzwerke angeschlossen ist. Ein *Router* ist ein Node, der Pakete weiterleitet, die nicht an ihn selbst gerichtet sind. Ein *Host* ist ein Node, der kein Router ist.

Ein *Interface* ist die Einrichtung eines Nodes, über das der Node an ein Netzwerk angeschlossen werden kann. Das Netzwerk wird in IPv6-Terminologie auch als *Link* bezeichnet. Aus der Sicht eines spezifischen Nodes, der an einen Link angeschlossen ist, werden alle weiteren Nodes, die an denselben Link angeschlossen sind, als *on-link* bezeichnet; diese Nodes werden auch als *Nachbarn* (*Neighbors*) bezeichnet. Nodes, die nicht on-link sind und nur über wenigstens einen Router erreichbar sind, werden auch als *off-link* bezeichnet.

Netze und die daran angeschlossenen Nodes formen eine *Site*, wenn sie – in einem geeigneten Sinn – einer gemeinsamen und zusammenhängenden Verwaltung unterstellt sind.

2 Motivation und grundlegende Eigenschaften von IPv6

In diesem Abschnitt werden die für diesen Artikel relevanten Konzepte von IPv6 in Kurzform vorgestellt. Detaillierte Beschreibungen finden sich in den einschlägigen RFCs. Eine relativ aktuelle und breite Darstellung der IPv6-Konzepte wurden von Silvia Hagen [Hagen 2009] sowie durch NIST [NIST SP800-119] gegeben; für Praktiker ist vor allem das Buch von Benedikt Stockebrand [Stockebrand 2010] von Nutzen.

2.1 Die IPv4-Problematik

Die Entwicklung von IPv6 wurde durch verschiedene Probleme und Erfahrungen mit IPv4 motiviert:

Aufzehrung des Adressraums
Der offensichtlichste Mangel von IPv4 besteht in der Knappheit an global routbaren Unicast-Adressen[4]. Diese Knappheit ist teilweise dem anfänglich kaum vorhersehbaren Wachstum des Internet und teilweise der unbedachten Vergabepraxis in den Anfangsjahren geschuldet, in denen großzügig Class-A und -B-Netze vergeben wurden. Viele dieser Zuweisungen haben sich im Nachhinein als sachlich nicht gerechtfertigt gezeigt. Die Architektur von IPv4 lässt keine einfachen Auswege aus dieser Situation zu: Eine Rückgabe ungenutzter Netzbereiche führt zu starker Fragmentierung des Adressraums (s. u.); ein Austausch von großen Netzblöcken gegen kleinere Netzblöcke führt zu signifikanten Ausfallzeiten, da die betroffenen Systeme neu adressiert (d. h. umnummeriert) werden müssen.

Dem steht eine zunehmende Zahl von Endgeräten gegenüber, die vernetzt werden sollen, angefangen von zahlreichen mobilen Geräten (Smartphones, Tablets usw.) einer wachsenden Nutzerbasis bis hin zum sprichwörtlichen Kühlschrank mit IP-Adresse und anderen integrierten Steuersystemen. Als Folge dessen wurde NAT dazu verwendet, um den knapper werdenden, verbleibenden Adressraum besser auf ein stark wachsendes Internet zu verteilen.

Network Address Translation (NAT)
NAT ist ein Netzwerkmechanismus, bei dem zunächst IP-Adressen auf andere IP-Adressen abgebildet werden. Hierbei sind verschiedene Formen möglich, auf die hier nicht näher eingegangen werden soll; hierfür wird auf RFC 3489 für eine Beschreibung verwiesen.

---

4 Landläufig auch als „öffentliche IP-Adressen“ bekannt.

Um das Internet einer breiten Nutzerbasis zugänglich zu machen, wird eine Form des sogenannten Restricted Cone NAT [RFC 3489] verwendet, um ganze Netzwerke – typischerweise private Netzwerke nach RFC 1918 – auf eine einzelne (global routbare) IP-Adresse abzubilden. NAT wird dabei typischerweise auf Firewalls durchgeführt, was zu der (falschen) Annahme verleiten kann, dass NAT originär ein Sicherheitsmechanismus wäre.

Aus Sicht der Sicherheit hat Cone NAT die interessante Eigenschaft, dass ein Angreifer den internen Zustand der NAT-Engine nicht kennt und daher nicht weiß, welche Ports zu welchen internen Systemen führen. Die Entscheidung darüber, ob eine Anfrage weitergeleitet oder verworfen wird, ist jedoch streng betrachtet eine Leistung der Firewall und nicht der NAT-Engine.

Bei jeder Form von NAT werden am NAT-Gateway die Adressen in den IP-Headern umgeschrieben. Bei der hier relevanten Form von NAT wird bei ausgehenden Paketen die Absendeadresse am NAT-Gateway auf die IP-Adresse des NAT-Gateways umgeschrieben. Bei eingehenden Paketen wird die Zieladresse auf eine Adresse im „internen“ Netz umgeschrieben, hierbei kann es sich um eine statisch konfigurierte Weiterleitung handeln, oder um eine dynamisch im Rahmen einer ausgehenden Verbindung eingerichtete Weiterleitung. Im letzteren Fall muss das NAT-Gateway Buch über die aktiven Verbindungen führen, wodurch das NAT-Gateway zum Flaschenhals oder gar zum Single Point of Failure werden kann.

Die verschiedenen Vor- und Nachteile von NAT werden beispielsweise in RFC 2993 diskutiert. Die sichtbarste Nebenwirkung von NAT ist die Aufgabe des Ende-zu-Ende-Prinzips. Eine weitere Nebenwirkung, die im praktischen Betrieb zu Schwierigkeiten führen kann, ist die Tatsache, dass die Adressen im IP-Header umgeschrieben werden. Dadurch scheitern Mechanismen wie IPsec, sobald die Integrität des IP-Headers überprüft wird, und auch Internet-Telefonie mit einer Signalisierung auf Grundlage von SIP oder H.323 schlägt an dieser Stelle zunächst fehl. Um beispielsweise SIP unter diesen Bedingungen noch betreiben zu können, müssen zusätzliche Gateways betrieben werden, welche die SIP-Pakete in geeigneter Weise modifizieren.

Andererseits sind die Systeme im per NAT abgebildeten Netzwerk von außen nicht mehr anhand der IP-Adresse voneinander zu unterscheiden, was einen gewissen Grad an Privatsphäre einbringt, siehe Abschnitt 3.5 für eine Diskussion von NAT im Zusammenhang mit Privatsphäre.

Aufgabe des Ende-zu-Ende-Prinzips
Eines der wesentlichen Designprinzipien des Internets liegt im Ende-zu-Ende-Prinzip, nach dem jeder Node im Internet unmittelbar Kontakt mit jedem anderen Node im Internet aufnehmen kann. NAT macht dieses Prinzip zunichte. Infolgedessen funktionieren Protokolle wie SIP, H.323 oder IPsec nicht ohne Hilfskonstruktionen, welche die Komplexität und die Fehleranfälligkeit davon abhängiger Systeme und Anwendungen noch weiter ansteigen lassen.

Fragmentierung des Adressraums
Ein weiteres Problem, das den Endnutzern eher verborgen bleibt, ist die starke Fragmentierung des IPv4-Adressraums. Auch dies ist der unbedachten Vergabe von IP-Adressblöcken in der Anfangszeit des Internet geschuldet. Eine Folge besteht darin, dass die Routingtabellen in der sogenannten *Default Free Zone*[5] mittlerweile eine Größe von grob 500.000 Einträgen haben. Für jedes einzelne IP-Paket und jedes Providernetz auf dem Weg vom Absender zum Empfänger muss der entsprechende Eintrag in der Routingtabelle

5 Landläufig auch als „Internet-Backbone“ bekannt.

wenigstens einmal ermittelt werden. Dieser Flaschenhals droht die Expansion des Internets mit einer rasant ansteigenden Zahl von angebundenen Geräten zu bremsen.

Weitere Aspekte
Daneben haben sich im Zuge der intensiven Nutzung von IPv4 weitere Probleme oder Verbesserungsmöglichkeiten herauskristallisiert, die nicht oder nicht ohne Weiteres zu beheben bzw. umzusetzen sind. So hat sich Broadcast-Übertragung im Wesentlichen für Angreifer als interessant und nützlich erwiesen, während legitime Nutzungsmöglichkeiten eher selten sind. Multicast dagegen ist genau wie Dienstgüte in den Ansätzen stecken geblieben, und Mobile IP wurde für IPv4 zwar spezifiziert, doch die IPv4-Adressarchitektur verhindert einen praktikablen Einsatz.

2.2 Unterschiede zwischen IPv6 und IPv4

In diesem Abschnitt werden die wesentlichen Unterschiede zwischen IPv6 und IPv4 beschrieben. Wie sich zeigt ist IPv6 weit mehr als nur IPv4 mit längeren Adressen – IPv6 bringt in vielen Punkten eine andere Architektur mit. Um IPv6 sicher zu betreiben, muss man als Administrator auch in den Kategorien von IPv6 denken – viele Rezepte aus der IPv4-Welt taugen unter IPv6 nicht mehr. Im Folgenden sind die wichtigsten Neuerungen von IPv6 im Vergleich zu IPv4 aufgeführt. Einzelne Punkte werden im Anschluss daran im Detail erklärt.

Größerer Adressraum
Der wohl am deutlichsten sichtbare Unterschied zu IPv4 ist die Größe des IPv6-Adressraums. Mit 128 Bit erlaubt IPv6 die Adressierung von deutlich mehr Nodes und Sites, als dies bei IPv4 mit 32 Bits der Fall war, siehe Abschnitt 2.3.8.

Einfachere Adressstruktur
Unicast-Subnetze sind bei IPv6 grundsätzlich einheitlich groß, nämlich /64-Netze; eine weitere Segmentierung dieser Subnetze ist nicht vorgesehen, da dies verschiedenen Architekturmerkmalen von IPv6 widerspricht. Aufgrund der einfachen Adressstruktur entfällt die Konfiguration von Netzmasken.

Einfache Adresskonfiguration
IPv6 bietet neben einer manuellen Konfiguration oder einer Konfiguration über DHCPv6 mit der Stateless Address Autoconfiguration (SLAAC) einen weiteren Mechanismus zur Vergabe von IPv6-Adressen an. Dieser Mechanismus beruht darauf, dass die MAC-Adressen der an einem Link angeschlossenen Systeme eindeutig sein müssen. Auf Grundlage der MAC-Adresse wird dann eine IPv6-Adresse gebildet, die zumindest an dem betreffenden Link ebenfalls eindeutig ist. Der Vorteil dieser Lösung besteht darin, dass unmittelbar auf IP-Layer kommuniziert werden kann, ohne dass Dienste wie DHCP zur Verfügung stehen müssen.

DHCPv6
Der Einsatz von DHCP ist dennoch möglich, sei es, um IPv6-Adressen wie bei IPv4 zu vergeben, und/oder um zusätzliche Konfigurationsinformation zu verteilen, wie beispielsweise die IP-Adresse eines DNS- oder NTP-Servers.

DNS
IPv6-Adressen werden im DNS durch AAAA-Einträge („Quad-A“) abgebildet; die Rückwärtsauflösung von IPv6-Adressen erfolgt mit Hilfe der Pseudo-Domain .ipv6.arpa. [RFC 3596].

Mehrere IPv6-Adressen pro Interface
Konzeptionell sieht IPv4 nur eine IP-Adresse pro Interface vor. Diese Beschränkung kann zwar in der Praxis über virtuelle Interfaces umgangen werden. Unter IPv6 wurde die Beschränkung aber nicht nur aufgehoben, hier ist es nun üblich, dass ein Interface mehrere IPv6-Adressen hat, nämlich wenigstens eine Link-Local-Adresse sowie eine oder mehrere andere Unicast-Adressen.

Einfachere Umnummerierung
Die Möglichkeit, dass einem einzelnen Interface mehrere IPv6-Adressen zugewiesen werden können, und die vereinfachte Adressstruktur ermöglicht es, Netze bei Bedarf *im laufenden Betrieb* relativ einfach und ohne direkten Eingriff an den betroffenen Systemen umzunummerieren. Die entsprechende Funktionalität wurde in ICMPv6 integriert.

Routenaggregation und effizientere Routingtabellen
Bei der Festlegung der Vergabestrategie für IPv6-Adressbereiche wurde darauf geachtet, dass IPv6-Adressen von vornherein so vergeben werden, dass topologisch benachbarte Adressbereiche routingtechnisch zusammengefasst werden können. Damit soll die Größe der Routingtabellen in der Default Free Zone trotz des enormen Adressraums übersichtlicher bleiben, als dies heute bei IPv4 der Fall ist. Dazu trägt bei, dass Netze mit relativ wenig Aufwand (im Vergleich zu IPv4) umnummeriert werden können, und dass die Vergabe von sogenannten Provider-Independent Adressen noch strikter gehandhabt wird als bei IPv4.

Schlankere Routingtabellen ermöglichen u. a. ein effizienteres Routing, was beispielsweise auch die Dienstgüte (Quality of Service, QoS) begünstigt. Darüber hinaus können Filterregeln viel einfacher formuliert werden. So erfordern beispielsweise länderspezifische Filterregeln mit IPv6 nur wenige Einträge im Vergleich zu IPv4, wo dies praktisch nur unter Rückgriff auf Geolocation-Dienste möglich ist.[6]

Einheitliches Steuerprotokoll
Mit IPv6 wurde auch ICMPv6 gegenüber dem ICMP von IPv4 stark erweitert. So wurden die Steuerfunktionen von ARP und RARP sowie von IGMP in Form von Neighbor Discovery bzw. Inverse Neighbor Discovery sowie Multicast Listener Discovery in ICMPv6 integriert. Die entsprechenden Funktionen wurden darüber hinaus weiter ausgebaut. Neue Steuerfunktionen, wie Router Discovery und Router Renumbering wurden ebenfalls in ICMPv6 integriert.

Einfacheres Header-Format
Der IPv6-Header ist im Vergleich zum IPv4-Header vereinfacht worden. Am auffälligsten ist die feste Header-Länge von 40 Bytes. Informationen zur Fragmentierung, die Checksumme oder Optionen (d. h. Felder für optionale Informationen) wurden aus dem Header entfernt. Der Grundgedanke hier war, eine effizientere Verarbeitung in Routern zu ermöglichen, was wiederum beispielsweise die Dienstgüte verbessert.

Extension Header
Optionen werden nun nicht mehr direkt im Header sondern in sogenannten Header-Erweiterungen (Extension Headers) untergebracht. Dadurch werden die von IPv4 her bekannten Größenbeschränkungen für Optionen beseitigt – solange das IP-Paket dafür Platz lässt, können Optionen im Prinzip deutlich länger und in beliebiger Zahl vorhanden sein. Darüber hinaus ermöglicht es die neue Architektur, auf einfache Weise neue Optionen einzuführen, ohne das Protokoll selbst zu verändern. Optionen werden von Routern nur noch selektiv ausgewertet, was einer effizienteren Verarbeitung zugute kommt.

Verbessertes Multicast und Abschaffung von Broadcast
Die Multicast-Funktionalität von IPv6 wurde stark ausgebaut und erweitert. Im Zuge dessen wurde bei IPv6 Broadcast abgeschafft; die Funktion einer Broadcast-Übertragung wird nun durch geeignetes Multicast erbracht.

Fragmentierung nur an der Quelle
Anders als bei IPv4 fragmentieren Router zu große Pakete bei IPv6 nicht mehr. Stattdessen wird grundsätzlich eine ICMPv6-Fehlermeldung an den Absender des Pakets

6 Siehe beispielsweise http://ipinfodb.com/

zurückgeschickt, falls ein Paket in der empfangenen Größe nicht weitergeleitet werden kann. Auch dies dient einer effizienteren Verarbeitung an Routern.

Größere minimale MTU und Path MTU
Die minimal garantierte MTU wurde für IPv6 auf 1280 Bytes (im Vergleich zu 576 Bytes bei IPv4) angehoben. Damit wird der Durchsatz bei der Übertragung großer Datenmengen gesteigert, da der Protokoll-Overhead verringert wird.

Da die Fragmentierung mit IPv6 nunmehr nur an der Quelle stattfindet, gewinnt die Feststellung der größten zulässigen MTU entlang der Route vom eigenen Node zum Ziel-Node (Path MTU) an Bedeutung.

Abschaffung von NAT
Der Hauptgrund für die Einrichtung von NAT bei IPv4 war die Knappheit an global routbaren Adressen. Der deutlich größere Adressraum von IPv6, die einfachere Adressstruktur sowie die strikteren Vergaberegeln für IPv6-Adressblöcke sollen sicherstellen, dass der ursprüngliche Grund für die Einführung von NAT entfällt. Dementsprechend war NAT für IPv6 zunächst überhaupt nicht vorgesehen – technisch gesehen ist eine Adresstranslation natürlich genau wie bei IPv4 machbar. Die Diskussion über den Sinn von NAT ist noch nicht beendet, dennoch lässt sich an dieser Stelle festhalten, dass der überwiegende Teil der „IPv6-Community“ NAT sehr skeptisch gegenübersteht. Darüber hinaus wirkt NAT – wenn auch marginal – zum Nachteil der Dienstgüte; die Abschaffung von NAT trägt also zur Verbesserung der Dienstgüte bei.

Mobile IPv6 (MIPv6)
Mit IPv6 wurde gleich die Grundlage für das Roaming mobiler Geräte geschaffen. Das Ziel ist es, zwischen verschiedenen Netzen umherwandern zu können, ohne dabei die Konnektivität auf IP-Ebene zu verlieren – selbst für bestehende Verbindungen, beispielsweise per TCP über IPv6. Aus Sicherheitssicht ist MIPv6 aufgrund des Risikos unautorisierter Umleitungen ausgesprochen problematisch, und da MIPv6 im Vergleich zu IPv6 noch relativ jung ist, ist dieses Gebiet noch nicht als stabil anzusehen. Mobile IP wurde zwar auch für IPv4 spezifiziert, ist aber schon allein aufgrund der Adressverhältnisse bei IPv4 praktisch unbrauchbar.

Quality of Service
Es wurden in den vorangegangenen Punkten bereits auf mehrere Verbesserungen in Bezug auf die Dienstgüte hingewiesen. Neben den bereits genannten Architekturmerkmalen von IPv6, die „nebenbei“ auch der Dienstgüte nützen, gibt es ein weiteres Merkmal, das gezielt die Verbesserung der Dienstgüte anspricht, nämlich das sogenannte Flow Label. Dies ist ein (neues) Feld im IPv6-Header, mit dem ein Datenfluss gekennzeichnet wird. Alle Pakete eines Flusses werden identisch behandelt. Mit dem Flow Label entfällt die Angabe von (wiederholten) Optionen in den Paketen des Flusses, die jeweils ausgewertet und verarbeitet werden müssten. Mit dem Flow Label werden somit im Prinzip Routingentscheidungen vereinfacht.

2.3 IPv6-Adressen

Die auffälligste sichtbare Neuerung von IPv6 gegenüber IPv4 ist die Länge und die Darstellung von IPv6-Adressen.

2.3.1 Schreibweise

Mit 128 Bits sind IPv6-Adressen viermal länger als IPv4-Adressen. Eine an die IPv4-Notation angelehnte Schreibweise von IPv6-Adressen würde demnach aus sechzehn durch Punkte getrennte Dezimalzahlen bestehen. Diese Schreibweise wäre so unhandlich, dass man sich

für eine andere Schreibweise entschieden hat. Eine IPv6-Adresse besteht aus acht durch Doppelpunkte getrennten Gruppen zu je vier hexadezimalen Ziffern [RFC 4291]. Beispiel:

```
2001:0db8:0000:0000:0000:cafe:0000:0000
```

Führende Nullen in einer Vierergruppe dürfen ausgelassen werden, Gruppen aus vier Nullen werden dabei zu jeweils einer einzelnen Null zusammengefasst:

```
2001:db8:0:0:0:cafe:0:0
```

Zwei oder mehr aufeinanderfolgende Blöcke aus Nullen dürfen zu `::` komprimiert werden. Um die Eindeutigkeit dieser Notation zu wahren, darf diese Verkürzung jedoch höchstens einmal angewendet werden, d. h. das vorstehende Beispiel kann als

```
2001:db8::cafe:0:0
```

oder als

```
2001:db8:0:0:0:cafe::
```

dargestellt werden, wobei die erste Darstellung kürzer ist als die zweite Darstellung. Nicht erlaubt ist hingegen die Darstellung `2001:db8::cafe::`, weil beispielsweise auch `2001:db8:0:0:cafe:0:0:0` dieselbe Kurzdarstellung hätte.

Eine besondere Schreibweise ergibt sich bei IPv6-Adressen mit eingebetteten IPv4-Adressen, deren letzte 32 Bits in der herkömmlichen „dotted quad“ Schreibweise von IPv4 angegeben werden dürfen. Die sogenannten IPv4-mapped Adressen haben die Form

```
::ffff:192.0.2.1
```

Die sogenannten IPv4-compatible Adressen der Form

```
::192.0.2.1
```

wurden mit RFC 4291 für obsolet erklärt und werden oft nicht mehr unterstützt. Diese Adressen sollten nicht verwendet werden, tauchen bei veralteten Implementierungen aber gelegentlich noch auf.

Wie sich herausstellt, erweist sich die Komplexität von IPv6-Adressen durchaus als relevant für verschiedene Sicherheitsaspekte, siehe Abschnitt 3.6.

2.3.2 Adresstypen

Bei IPv6-Adressen unterscheidet man zwischen den Adresstypen Unicast-, Anycast- und Multicast-Adresse. Broadcast-Adressen, wie von IPv4 her bekannt, existieren bei IPv6 nicht. Dafür wurde Multicast besser ausgebaut. So wird die herkömmliche Funktion der Broadcast-Adresse von der Link-Local All-Nodes Multicast-Adresse `ff02::1` übernommen. Grundsätzlich kann man diese Adresstypen folgendermaßen charakterisieren:

Unicast
Unicast dient der Kommunikation zwischen zwei Interfaces. Jede Unicast-Adresse wird genau einem Interface zugewiesen; eine Nachricht an eine Unicast-Adresse wird nur an das entsprechende Interface gesendet. Global Unicast-Adressen werden derzeit aus dem Block `2000::/3` vergeben; Unique Local Unicast hat das Routing-Präfix `fc00::/7`.

Multicast
Multicast dient der Kommunikation von einem Interface zu mehreren Interfaces, beispielsweise für Internetradio. Eine Multicast-Adresse kann beliebig vielen Interfaces zugewiesen werden; eine Nachricht an die Multicast-Adresse wird an alle entsprechenden Interfaces gesendet. Multicast-Adressen sind an dem Routing-Präfix `ff00::/8` zu erkennen.

Anycast
Anycast dient der Kommunikation zwischen zwei Interfaces. Anders als beim Unicast ist dem Initiator der Kommunikation nicht so sehr daran gelegen, mit welcher Gegenstelle er kommuniziert, sondern welche Eigenschaft die Gegenstelle hat. So könnten beispielsweise unterschiedliche DNS- oder NTP-Server ihre Dienste unter derselben, bekannten IP-Adresse anbieten; die Router in dem Netzwerk würden – dem Grundgedanken nach – den Verkehr zum nächstgelegenen „passenden" Server routen. Anycast-Adressen finden derzeit jedoch noch keine weite Verbreitung, hierzu wäre es erforderlich, Router regelmäßig über die Präsenz von Diensten zu informieren. Die dafür erforderliche Protokollinfrastruktur existiert jedoch noch nicht.

Jede Anycast-Adresse kann beliebig vielen Interfaces zugewiesen werden; eine Nachricht an die Anycast-Adresse wird an genau eines der entsprechenden Interfaces gesendet. Anders als Multicast-Adressen sind Anycast-Adressen von Unicast-Adressen nicht anhand ihres Routing-Präfix unterscheidbar.

Eine typische Anycast-Adresse ist die Subnet-Router Anycast-Adresse, die den Routern zugewiesen wird, die das Subnet-Präfix über Router Advertisements verteilen. Diese Anycast-Adressen haben die Form *präfix*`::/64` [RFC 4291].

2.3.3 Scopes

Bei IPv6-Adressen wird zwischen verschiedenen „Scopes", d. h. Gültigkeitsbereichen, innerhalb derer die betreffende Adresse routbar ist, unterschieden [RFC 4007]. Dies hat Auswirkungen auf Routing und Filterung.

Global Scope
Global Scope Adressen sind global routbar. Bis auf spezielle Adressbereiche ist grundsätzlich der gesamte Block `2000::/3` global routbar. Der Bereich von `4000::` bis `fc00::` ist zwar prinzipiell auch global routbar, wird von IANA aber derzeit reserviert gehalten und nicht vergeben.

Site-Local Scope und Unique Local Adressen
Site-Local Scope Adressen sind routbar, jedoch sollten diese Adressen an Border-Routern des jeweiligen Standorts abgewiesen werden. Diese Adressen entsprechen somit den IPv4-Adressen für den privaten Gebrauch nach RFC 1918. Site-Local Scope Adressen sind an ihrem Präfix zu erkennen. Zunächst wurde dafür das Routing-Präfix `fec0::/10` vorgesehen, Site-Local Scope Adressen wurden aber wegen konzeptioneller Schwierigkeiten wieder verworfen [RFC 3879] und sollten nicht verwendet werden.

Stattdessen wurden Unique Local Adressen (ULAs) eingeführt [RFC 4193], die am Routing-Präfix `fc00::/7` zu erkennen sind. Das Präfix einer ULA enthält ferner eine 40-Bit Global ID. Um eine der Schwierigkeiten von Site-Local Scope Adressen zu vermeiden, muss für die Global ID ein *zufälliger* Wert gewählt werden. Damit soll sichergestellt werden, dass es zu keinen Kollisionen zwischen ULA-Netzwerken kommt, etwa wenn mehrere Intranets zu einem übergreifenden Intranet vereinigt werden sollen, beispielsweise im Zuge einer Unternehmensfusion.[7]

Link-Local Scope
Link-Local Scope Adressen sind nicht routbar und haben nur an dem jeweiligen Netzwerk-

[7] Die Global ID 0 ist *keine* gute Wahl. Die Administratoren, die hierfür dennoch den Wert 0 auswählen, dürfen sich zu der Mehrarbeit beglückwünschen, sobald zwischen den betroffenen Netzen geroutet werden soll. Glücklicherweise bietet der IPv6-Mechanismus zur Umnummerierung einen Ausweg, der die Mehrarbeit auf ein erträgliches Maß begrenzt.

Link eine Bedeutung. Diese Adressen sind an dem Präfix `fe80::/10` zu erkennen. Da jeder Link dasselbe Link-Local-Präfix hat, ist bei Systemen mit mehreren Interfaces eine Angabe wie `fe80::cafe` mehrdeutig. Daher muss über einen *Zonenindex* spezifiziert werden, über welches Interface das Paket versendet werden soll. Dies erfolgt meist[8] über die Schreibweise *adresse%zonenindex* [RFC 4007], also beispielsweise

```
gondor ~ # ping6 -c1 fe80::20c:29ff:fe08:454b%eth1
PING fe80::20c:29ff:fe08:454b%eth1(fe80::20c:29ff:fe08:454b) 56 data bytes
64 bytes from fe80::20c:29ff:fe08:454b: icmp_seq=1 ttl=64 time=0.333 ms

--- fe80::20c:29ff:fe08:454b%eth1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.333/0.333/0.333/0.000 ms
```

Ohne Angabe des Zonenindex käme es bei der Anfrage zu einer Fehlermeldung. Die Pflicht zur Angabe des Zonenindex macht Link-Local-Adressen unhandlich und für den Routineeinsatz unbrauchbar, andererseits stellen Link-Local-Adressen einen guten Rettungsanker für Notfälle dar, wenn beispielsweise das Netz falsch konfiguriert wurde.

2.3.4 Übersicht der vorgeschriebenen Adressen

Jeder Node muss an jedem Interface die folgenden Adressen als eigene Adresse erkennen [RFC 4291]:

- Die Loopback-Adresse
- Eine Link-Local-Adresse des betreffenden Interfaces
- Die Link-Local All-Nodes Multicast-Adresse
- Alle zugewiesenen Unicast- oder Anycast-Adressen des betreffenden Interfaces
- Für jede zugewiesene Unicast- oder Anycast-Adresse die dazugehörige Link-Local Solicited-Node Multicast-Adresse
- Die Multicast-Adressen aller Gruppen, denen das betreffende Interface angehört

Router müssen zusätzlich die folgenden Adressen als eigene Adressen erkennen:

- Für jedes Interface, auf dem der Node als Router auftritt, die Subnet-Router Anycast-Adresse
- Alle weiteren Anycast-Adressen, für die der Router konfiguriert wurde
- Die Link-Local und Site-Local All-Routers Multicast-Adressen (`ff02::2` bzw. `ff05::2`)

2.3.5 Übersicht der bisher definierten Adressbereiche

Die folgende Tabelle zeigt eine Übersicht der wichtigsten, derzeit definierten Adressbereiche [RFC 6890]. Grau dargestellte Einträge gehören zu Adressbereichen, die ehemals verwendet und später zurückgegeben wurden, oder deren Nutzung nicht empfohlen wird. Kursiv gesetzte Einträge beziehen sich auf reservierte Präfixe für IPv4/IPv6-Tunneling und -Übergangsverfahren, Testnetze, experimentelle Protokolle etc.

8 In URIs wird das Zeichen „%“ bereits für die hexadezimale Zeichencodierung verwendet. Daher sollte es dort durch seinen eigenen Hex-Code „%25“ ersetzt werden [RFC 6874], also beispielsweise http://[fe80::20c:29ff:fead:49ac%25eth1]/.

Präfix	Bezeichnung	IPv4-Äquivalent
::/128	Unspezifizierte Adresse [RFC 4291]	0.0.0.0
::1/128	Loopback-Adresse [RFC 4291]	127.0.0.1
::ffff:0:0/96	*IPv4-Mapped* [RFC 4038]	
::/96	*IPv4-Compatible* [RFC 4291]	
fe80::/10	Link-Local [RFC 4291]	169.254.0.0/16
fec0::/10	Site-Local [RFC 3513, RFC 3879]	
fc00::/7	Unique Local [RFC 4193]	10.0.0.0/8 172.16.0.0/12 192.168.0.0/16
2001::/32	*Teredo* [RFC 4380]	
2001:2::/48	Benchmarking	198.18.0.0/15
2001:10::/28	*ORCHID* [RFC 4843]	
2002::/16	*6to4* [RFC 3056]	
2001:db8::/32	*Documentation* [RFC 3849]	192.0.2.0/24 198.51.100.0/24 203.0.113.0/24
2000::/3	Global Unicast	
3ffe::/16 5f00::/8	*6bone* [RFC 1897, RFC 2471, RFC 3701]	
ff00::/8	Multicast [RFC 4291]	224.0.0.0/4

2.3.6 Routbare Unicast-Adressen

Eine global routbare Unicast-Adresse folgt im Prinzip dem in Abbildung 1 gezeigten Muster. Die 128-Bit IPv6-Adresse teilt sich in ein 64-Bit Präfix[9] und eine 64-Bit Interface-ID. Das Präfix ist nochmals (an einer variablen Grenze) unterteilt in ein Routing-Präfix[10] und die Subnet-ID. Das Routing-Präfix wird der End-Site durch den ISP zugewiesen, die Vergabe der Subnet-ID liegt in der Verantwortung der End-Site. Bekommt eine End-Site ein Präfix der Länge 48 Bits, so verbleiben für die Subnet-ID 16 Bits; bekommt die End-Site ein Präfix der Länge 56, so verbleiben für die Subnet-ID 8 Bits, usw. Abbildung 1 zeigt eine Aufteilung des Präfix in 40 Bits für das Routing-Präfix sowie 24 Bits für die Subnet-ID; eine solche Zuteilung wäre für relativ große Sites typisch.

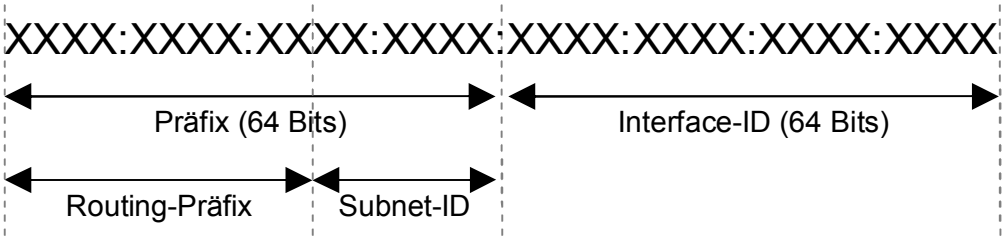

Abbildung 1: Schematische Darstellung der Elemente einer routbaren Unicast-Adresse

Die Länge des Präfix bzw. des Routing-Präfix wird in Anlehnung an CIDR-Notation als

9 Wird auch als Subnet-Prefix bezeichnet.

10 Wird auch als Global Routing-Präfix bezeichnet.

IPv6-Adresse/Präfixlänge

dargestellt. Ein typisches Subnetz sieht beispielsweise so aus:

`2001:db8:0:cafe::/64`

Das dazu passende Routing-Präfix könnte beispielsweise `2001:db8:0::/48` oder `2001:db8:0:ca00::/56` lauten.

2.3.7 Multicast-Adressen

Multicast-Adressen stammen grundsätzlich aus dem Adressblock `ff00::/8`. In ihrem Aufbau entsprechen Multicast-Adressen der Darstellung in Abbildung 2. In der Abbildung bezeichnen „F“ Flags und „S“ den Scope der Multicast Adresse (jeweils vier Bits).

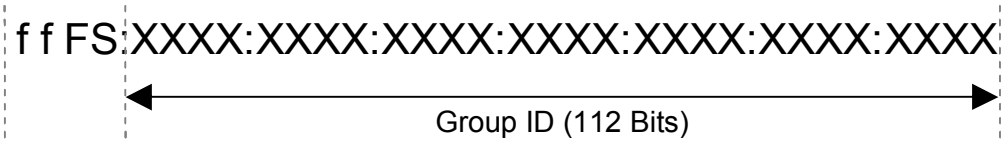

Abbildung 2: Schematische Darstellung der Elemente einer Multicast-Adresse

Von den vier möglichen Flags sind drei definiert, das vierte ist derzeit noch reserviert. Die Flags beschreiben die Art der Multicast-Adresse und wie der betreffende Multicast-Verkehr zu routen ist. Eine Beschreibung der Flags führt an dieser Stelle zu weit, hierfür wird auf Silvia Hagen [Hagen 2009] und Bendikt Stockebrand [Stockebrand 2010, Kapitel 18] verwiesen.

Für den Scope S sind die folgenden Werte definiert:

Wert	Scope	Erklärung
1	Interface Local	Multicast Loopback
2	Link Local	Nicht routbar, nur am betreffenden Link gültig
4	Aministration Local	Umfasst administrativ zusammengehörige Netzwerkgruppen innerhalb einer Site
5	Site Local	Umfasst alle Netze einer Site
8	Organization Local	Umfasst alle Netze einer Organisation
e	Global	Global routbar

Alle nicht in der Tabelle aufgeführten Scope-Werte sind derzeit noch reserviert.

Der Scope ist bei der Einrichtung von Filterregeln an Routern und Firewalls zu berücksichtigen.

RFC 2375 führt u. a. folgende bekannte Multicast-Adressen auf:

`ff02::1`	Link-local All Nodes
`ff02::2`	Link-local All Routers
`ff05::2`	Site-local All Routers
`ff05::101`	Site-local All NTP Servers
`ff05::1:3`	Site-local All DHCP Servers

Weitere Beispiele sind in RFC 2375 zu finden, die jeweils aktuelle und vollständige Liste ist bei IANA abrufbar.[11]

2.3.8 Der Adressraum

Einer der deutlichsten Unterschiede zwischen IPv6 und IPv4 ist wie bereits mehrfach erwähnt die Länge einer Adresse und somit die Größe des Adressraumes. Mit 128 Bit erreicht der IPv6-Adressraum eine unvorstellbare Größe. In der Literatur finden sich zahlreiche mehr oder weniger blumige Vergleiche zur Verdeutlichung der Adressfülle, etwa, dass jedem Quadratzentimeter der Erde oder jedem Sandkorn mehrere IPv6-Adressen zugewiesen werden können. Abgesehen von der schwachen Aussagekraft des Vergleichs (wer kann sich schon alle Sandkörner vorstellen) und der inhaltlichen Fragwürdigkeit der Aussage (die Vergabe einzelner IPv6-Adressen ist so nicht vorgesehen) werden die tatsächlichen Dimensionen nicht deutlich. Andere Vergleiche stellen dar, wie viele IP-Adressen jedem Erdbewohner zur Verfügung stünden; obwohl dieser Vergleich nicht weniger krude ist, kommt er der IPv6-Vergabephilosophie schon deutlich näher.

Würde der gesamte IPv6-Adressraum gleichmäßig auf alle derzeit etwa 7,1 Milliarden Erdenbürger verteilt, dann bekäme jeder Mensch grob 10^{29} Adressen, das entspricht in etwa 39644 /48-Subnetzen. Würde ab heute jeder Erdenbürger zu seinen Lebzeiten stattdessen lediglich *ein* /48-Subnetz mit jeweils etwa 10^{24} Adressen erhalten, dann wäre der IPv6-Adressraum (ohne Berücksichtigung der bereits vergebenen Präfixe) bei einer jährlichen Bevölkerungszunahme von konstant 1.2% (siehe http://www.prb.org/) erst etwa im Jahr 2900 erschöpft. Selbst wenn man sich lediglich aus dem Pool bedienen würde, den IANA derzeit zur Vergabe freigegeben hat (`2000::/3`), dann wäre dieser Pool erst etwa im Jahr 2725 erschöpft. Eine Vergabe von /40- Präfixen (also 256 /48-Präfixe) *an jeden Menschen nur aus diesem Pool* wäre immerhin noch 248 Jahre lang möglich. Diese Rechenbeispiele zeigen nicht nur die Dimension des IPv6-Adressraums, sondern auch, dass die Vergabe von /48-Präfixen an End-Sites (d. h. auch an Privathaushalte) durchaus gerechtfertigt ist.

Eine andere Darstellung der Adressfülle ergibt sich durch einen Vergleich mit IPv4, konkret durch den Vergleich der Abschätzungen für die konzeptionell maximale Anzahl von Sites in beiden Fällen. Bei IPv4 mit 32 Bit-Adressen stehen nach Abzug aller Sonderadressen etwa $3{,}67 \times 10^9$ global routbare Adressen zur Verfügung. Würde für jede dieser Adressen konsequent NAT von und auf Adressen nach RFC 1918 durchgeführt werden, so dass jeder Site lediglich eine einzige global routbare Adresse zugewiesen würde, dann entspräche diese Zahl von $3{,}67 \times 10^9$ zugleich der maximalen Anzahl aller Sites im Internet4.

Würde im Vergleich dazu jeder Site konsequent ein /48-Präfix aus dem Block `2000::/3` zugewiesen (wie durch RFC 3177 empfohlen), dann ergäbe sich die Zahl von etwa 35×10^{12} Sites im Internet6. Somit wäre das Internet6 in Bezug auf die maximale Anzahl der Sites etwa das 9600-fache des Internet4. Allerdings werden bei dieser Gegenüberstellung Erbsen mit Kürbissen verglichen, denn zum einen erhielte jede Site dadurch bei IPv6 *65536* global routbare *Netze*, während eine Site bei IPv4 nur *eine einzelne* global routbare *Adresse* bekäme. Zum anderen stehen bei IPv6 noch wenigstens fünf weitere Blöcke derselben Größe zur Verfügung, aus denen derzeit jedoch keine Adressen vergeben werden. Und schließlich entspricht die dem Vergleich zu Grunde gelegte hypothetische Vergabepraxis für IPv4 nicht der Realität, denn tatsächlich belegen sehr viele Sites im Internet4 ein Class-C-Netz oder größer.

---

11 IANA IPv6 Multicast Address Space Registry

Die Empfehlung zur konsequenten Vergabe von /48-Präfixen an End-Sites wurde mit RFC 6177 verworfen. Ausgehend von einer konsequenten Vergabe von /56-Präfixen im Internet6 aus dem Block `2000::/3` im Vergleich zu einer konsequenten Vergabe von Class-C-Netzen im Internet4 ergibt sich ein Größenverhältnis von etwa 625 Millionen zu 1; bei einer Vergabe aus dem gesamten IPv6-Unicast-Bereich `2000::/3` bis `fc00::/7` beträgt das Verhältnis etwa 4,36 Milliarden zu 1. Dies entspricht in etwa auch dem Verhältnis 2^{32} zu 1.

Damit entspricht der IPv6-Adressraum aus Sicht der Vergabepraxis einer Erweiterung von 32 Bits (IPv4-Adressen) auf 64 Bits (IPv6-Präfixe). Aus praktischer Sicht ist das sehr viel – aus konzeptioneller Sicht stellt sich die Frage, wieso eine viermal längere Adresse „nur“ einen so „kleinen“ Gewinn an Adressraum bringt, d. h. es stellt sich die Frage, was mit den restlichen 64 Bits passiert. Die Antwort liegt in der Struktur von IPv6-Adressen, genaugenommen in der Struktur der Unicast-Adressen. Die 128 Bits einer Unicast-Adresse werden aufgeteilt in ein 64-Bit-Präfix und eine 64-Bit-Interface-ID (siehe Abbildung 1). Diese Aufteilung ist fest und darf nicht verändert werden. Das Präfix teilt sich nochmals auf in das Routing-Präfix und die Subnet-ID. Das Routing-Präfix wird der End-Site vom ISP zugewiesen, die Zuweisung der Subnet-ID liegt in der Verantwortung der End-Site. Ein Subnetz bezeichnet hier ein /64-Subnetz, denn dies ist die kleinste Subnetz für Unicast-Adressen. Es ist zu beachten, dass /64-Subnetze *nicht weiter unterteilt werden dürfen*!

Warum IPv6-Adressen 128 Bits haben

Ursprünglich waren nur 64 Bits für IPv6-Adressen vorgesehen. Im Vergleich zu IPv4-Adressen führt dies bereits zu einer so große Erweiterung, dass man auf absehbare Zeit eigentlich genügend Adressen zur Verfügung gehabt hätte. Warum also 128 Bits?

Der Grund liegt darin, dass die Adressen genügend Raum bieten sollen, um topologische Informationen unterzubringen.[12] Dies ist notwendig, um das Routing einfach und die Routingtabellen übersichtlich zu halten. Zu Beginn dieses Abschnitts wurde das Gedankenexperiment gemacht, jedem Erdenbürger ein /48-Netz zu geben. Eine sequenzielle Vergabe ist jedoch routingtechnisch sehr ungünstig. Man stelle sich entsprechend vor, die in Deutschland üblichen Adressen mit Postleitzahl, Ortsname, Straßenname und Hausnummer würden durch eine Art Seriennummer ersetzt werden, die landesweit sequenziell nach Einrichtungszeitpunkt einer Adresse vergeben würden. Adressen in einem derartigen Schema wären zum Auffinden eines Ortes völlig ungeeignet. In Netzwerken verhält es sich genauso – eine ungeordnete Vergabe von Adressen (oder Präfixen) macht Routing praktisch unmöglich (und aus diesem Grund können Link-Layer-Adressen nicht zum Routing verwendet werden).

Um das Routing effizient zu gestalten, müssen Adressen hierarchisch strukturiert vergeben werden. Dazu ist es jedoch notwendig, für Informationen der verschiedenen Ebenen der Hierarchie (globale Region – Land – Provider usw.) genügen Raum in der Adresse (d. h. im Präfix) zu lassen. Darüber hinaus soll durch eine spärliche Vergabe von Präfixen Raum für zukünftige Entwicklungen vorgesehen werden. Dies wirkt zwar zunächst verschwenderisch, macht aber langfristig Erweiterungen möglich, ohne später größere Netzbereiche umnummerieren zu müssen.

12 Ein gute Darstellung dieses Sachverhalts sowie zur Vorgeschichte von IPv6 ist unter http://oversteer.bl.echidna.id.au/IPv6/misc/jaubert-ipv6.html zu finden

2.3.9 Interface Identifier

In den vorangegangenen Abschnitten war verschiedentlich vom sogenannten Interface Identifier die Rede. Dies ist ein 64-Bit langer Wert, der sich aus der Adresse des Interfaces auf Link-Ebene ergibt. Wird IPv6 beispielsweise über Ethernet oder WLAN betrieben, dann wird der Interface Identifier aus der betreffenden MAC-Adresse gebildet. Wie aus einer 48-Bit-MAC-Adresse ein 64-Bit-Identifier erzeugt wird, wurde durch IEEE definiert; das Ergebnis ist der sogenannte 64-Bit Extended Unique Identifier (EUI-64). Die Details hierzu sind auch in RFC 4291, Anhang A beschrieben. Neuere Windows-Versionen verwenden jedoch nicht den EUI-64 Identifier, sondern einen pseudozufälligen Wert.

Beispiel im Detail

Ein Interface mit der MAC-Adresse `00:0c:29:08:45:4b` bekommt den Interface Identifier `20c:29ff:fe98:454b`, d. h. zwischen die drei hochwertigen (linken) und die drei niederwertigen (rechten) Bytes werden die zwei Bytes `ff` und `fe` eingefügt, außerdem wird sogenannte Universal/Local-Bit (Bit 1 des höchstwertigen Bytes) invertiert.

Per SLAAC würde das Interface also die Link-Local-Adresse `fe80::20c:29ff:fe98:454b` erhalten, und zu dem Präfix `2001:db8:ca84:e2d5::/64` würde das Interface sich selbst die IPv6-Adresse `2001:db8:ca84:e2d5:20c:29ff:fe98:454b` zuweisen.

2.3.10 Aufteilung des Adressraums einer Site

2.3.10.1 Mehrere Subnetze

Bei der Aufteilung des Adressraums steht aufgrund der Erfahrungen mit IPv4 bei IPv6 der Gedanke der Routenaggregation im Vordergrund. Bekommt eine End-Site einen Adressblock, dann stellt sich die Frage, wie diese Netze vergeben werden. Eine sequenzielle Vergabe, wie bei IPv4 üblich, ist möglich, der Idee der Routenaggregation jedoch nicht zuträglich. Stattdessen sollte nach Topologie nummeriert werden, d. h. routing-technisch benachbarte Netze sollten auch bezüglich der Adressierung benachbart sein. RFC 3531 beschreibt eine Methodik zur Vergabe, bei der die Möglichkeit zur Routenaggregation möglichst lange gewahrt bleibt. Glücklicherweise sind Fehlentscheidungen an dieser Stelle mit IPv6 nicht so gravierend wie bei IPv4, da die Architektur von IPv6 ein relativ einfaches Umnummerieren von Netzen ermöglicht.

2.3.10.2 Verteilung innerhalb eines einzelnen Subnetzes

Für die Adressierung in einem einzelnen Unicast-Netz stehen 64 Bits zur Verfügung. Dies erscheint zunächst verschwenderisch groß. Einer der Grundgedanken dabei ist, dass niemals wieder Techniken wie NAT zur „Streckung“ des bestehenden Adressraumes verwendet werden sollten. Zum anderen benötigen einige Funktionen einen so großen Adressraum, um Kollisionen zwischen automatisch – teilweise mit einer Zufallskomponente – gewählten Adressen mit hoher Wahrscheinlichkeit auszuschließen. Zu diesen Funktionen gehören:

- Stateless Address Autoconfiguration (SLAAC) [RFC 4862]
- Cryptographically Generated Addresses (CGA) [RFC 3972]
- Hash-Based Addresses (HBA) [RFC 5535]
- Privacy Extensions [RFC 4941]

Von IPv4 ist die Zuordnung von Netzen nach Funktion bekannt, beispielsweise für Server, für Clients, für Peripheriegeräte, für Gäste usw. Darin werden einzelne Adressen für Systeme typischerweise sequenziell vergeben, also beispielsweise `192.0.2.1`, `192.0.2.2`, `192.0.2.3` usw. Es gibt keine eindeutigen Empfehlungen, wie ein Subnetz mit Adressen zu bevölkern ist, jedoch widerspricht es dem Geist von IPv6, auch hier zu einer sequenziellen Zählung zu greifen. Stattdessen wird eine mehr oder weniger zufällige Verteilung über den gesamten für das Subnetz zur Verfügung stehenden Adressraum vorgezogen. Eine derartige Verteilung bietet auch Vorteile bei der Verteidigung gegen Systemaufzählung (etwa mit einem Portscanner wie nmap), da ein Angreifer ohne direkten Zugriff auf den Link einen extrem großen Adressraum durchsuchen müsste, um alle Systeme aufzuzählen, siehe dazu auch Abschnitt 4.2.

2.3.11 Adressauswahl

Ein weiterer signifikanter Unterschied zu IPv4 besteht darin, dass einem Interface mehrere Adressen aus unterschiedlichen Scopes zugewiesen werden können. Im Betrieb kann sich bei einer Anfrage dynamisch ergeben, welche Quelladresse und welche Zieladresse für die Anfrage verwendet wird.

Beispiel: Mehrere Systeme an einem Link haben eine gemeinsames ULA-Präfix sowie ein gemeinsames global routbares Präfix. Jedes System kann für eine Kommunikation mit einem anderen System auf demselben Link also jeweils auf wenigstens drei Adressen (Link-Local, Unique Local und Global) zurückgreifen, und zwar sowohl für die Quelle als auch für das Ziel. Welche davon wird jeweils ausgewählt?

RFC 6724 beschreibt die Grundsätze, nach der Quell- und Zieladresse ausgewählt werden. Die genaue Darstellung ist nicht Gegenstand dieses Artikels, zum groben Verständnis reicht es zu wissen, dass die Scopes von Quell- und Zieladresse zueinander passen müssen, und dass möglichst effiziente Routen ausgewählt werden. Dazu gehört, dass möglichst kleine Scopes und möglichst große Präfixübereinstimmung gewählt werden. Die Entscheidung kann durch eine Policy-Tabelle beeinflusst werden. Diese Tabelle ist üblicherweise vorkonfiguriert und braucht für gewöhnlich nicht angepasst zu werden. Der Algorithmus aus RFC 6724 legt nahe, dass das Routing *vor* der Adressauswahl stattfindet, es wird aber auch angemerkt, dass in bestimmten Fällen ein Routing *nach* der Adressauswahl sinnvoll sein könnte.

2.3.12 Adresszustände und Gültigkeitsdauer

Im Rahmen der Adresskonfiguration kann eine Unicast- oder Anycast-Adresse verschiedene Zustände einnehmen:

Tentative (vorläufig)
Eine Adresse ist vorläufig, wenn bei der Zuweisung einer Adresse (Unicast oder Anycast) im Rahmen der Duplicate Address Detection noch nicht abschließend festgestellt wurde, dass die Adresse nicht von einem anderen Interface am selben Link beansprucht wird. Eine vorläufige Adresse darf nicht als Absenderadresse verwendet werden, und bis auf bestimmte Nachrichten der Neighbor Discovery darf das betreffende Interface auch kein Paket mit der vorläufigen Adresse annehmen. Üblicherweise ist jede Adresse zunächst vorläufig, die einem Interface zugewiesen wird; die Duplicate Address Detection wird automatisch durchlaufen. Verläuft die Duplicate Address Detection nicht erfolgreich, dann kann die zugewiesene Adresse nicht verwendet werden und alle weiteren Schritte sind manuell durchzuführen.

Valid (gültig)
Verläuft die Duplicate Address Detection für eine Adresse erfolgreich, dann wird diese

Adresse gültig. Eine gültige Adresse kann als Absenderadresse verwendet werden, und Pakete mit gültiger Zieladresse müssen vom Empfänger verarbeitet werden. Eine gültige Adresse kann entweder bevorzugt oder abgelaufen sein, siehe nachstehend.

Preferred (bevorzugt)
Eine Adresse ist solange bevorzugt, wie dies durch die entsprechende Gültigkeitsdauer (preferred lifetime) angezeigt wird. Eine bevorzugte Adresse kann uneingeschränkt verwendet werden.

Deprecated (abgelaufen)
Nach Ablauf der Gültigkeitsdauer als bevorzugte Adresse kann die Adresse nach wie vor gültig sein, beispielsweise für bestehende Verbindungen. Dieser Zustand bleibt solange erhalten, bis auch die Gesamtgültigkeitsdauer (valid lifetime) abgelaufen ist.

Invalid (ungültig)
Nach Ablauf ihrer Gesamtgültigkeitsdauer wird eine Adresse ungültig und darf nicht mehr verwendet werden, d. h. sie darf weder als Absenderadresse verwendet werden, noch dürfen Pakete mit dieser Zieladresse verarbeitet werden.

Die Zuweisung der Gültigkeitsdauern kann manuell erfolgen, wie nachstehendes Beispiel zeigt:

```
arnor ~ # ip -6 addr add 2001:db8::cafe/64 dev eth1 valid_lft 20 preferred_lft 10
arnor ~ # ip -6 addr show dev eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 2001:db8::cafe/64 scope global dynamic
       valid_lft 16sec preferred_lft 6sec
    inet6 fe80::20c:29ff:fe08:454b/64 scope link
       valid_lft forever preferred_lft forever
```

und etwas später

```
arnor ~ # ip -6 addr show dev eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 2001:db8::cafe/64 scope global deprecated dynamic
       valid_lft 8sec preferred_lft 0sec
    inet6 fe80::20c:29ff:fe08:454b/64 scope link
       valid_lft forever preferred_lft forever
```

Der übliche Fall im praktischen Betrieb wird aber sein, dass Adressen und Gültigkeitsdauern automatisch im Rahmen eines Router Advertisements zugewiesen werden. Die Parameter für die preferred lifetime und die valid lifetime sind für jedes Präfix separat im Router Advertisement enthalten. In dem Beispiel aus Abbildung 6 in Abschnitt 2.6.2 wird für die preferred und die valid lifetime ein Wert von 4 bzw. 24 Stunden gewählt.

Die valid lifetime ist im Rahmen einer Umnummerierung eines Netzwerks kritisch. Es ist zwar möglich, den Wert auf „forever“ zu setzen (Link-Local-Adressen haben üblicherweise eine unbegrenzte Gültigkeit, wie aus dem vorstehenden Beispiel ersichtlich ist), somit bleibt ein Präfix (mitsamt allen daraus gebildeten Adressen) bis auf Weiteres unbegrenzt gültig, davon wird jedoch abgeraten, da dies im Zusammenhang mit Umnummerierung zu Problemen führen kann [RFC 4861, Abschnitt 12].

2.4 Der IPv6-Header und seine Erweiterungen

In der IPv6-Architektur ändert sich im Vergleich zu IPv4 ein wichtiges Element, nämlich der IPv6-Header mit seinen Erweiterungen. Dieses Element erweist sich durchaus als sicherheitsrelevant. Eine detaillierte Darstellung des IPv6-Headers ist in RFC 2460 oder in der Literatur [Hagen 2009] gegeben. Der IPv6-Header ist im Vergleich zum IPv4-Header vereinfacht worden. Die wesentlichen Änderungen betreffen optionale Informationen: Optionen wurden aus dem Header gestrichen, diese finden sich nun in gesonderten Header-

Erweiterungen (Extension Header). Dies hat gleich mehrere Vorteile, die der Effizienz der Verarbeitung und der Erweiterbarkeit zugute kommen:

- Der Header hat nun eine feste Größe.
- Eine Checksumme ist nicht mehr vorgesehen. Dies wird damit begründet, dass Checksummen bereits jeweils auf Verbindungsebene und auf Transportebene berechnet werden; eine Checksumme auf Netzwerkebene liefert dann keinen Mehrwert.
- Optionen werden aus dem Header verbannt, belegen dort keinen Platz mehr und müssen dort – vor allem von Routern – nicht verarbeitet werden
- Optionen brauchen nicht auf einem beschränkten Platz untergebracht zu werden.
- Neue Optionen können eingeführt werden, ohne den bestehenden Header zu verändern oder bestehende Optionen zu „verbiegen“.

Die neue Flexibilität führt jedoch auch dazu, dass Filterregeln komplexer werden. Die folgende Tabelle verdeutlicht die Unterschiede zwischen IPv4 und IPv6:

IPv4 Header-Element	IPv6 Header-Element
Version (4)	*Version* (6)
Header Length	—
Type Of Service	*Traffic Class*
—	*Flow Label*
Total Length	*Payload Length*
Identification	in den Fragment Header verschoben
Flags	in den Fragment Header verschoben
Fragment Offset	in den Fragment Header verschoben
Time To Live	*Hop Limit*
Protocol	*Next Header*
Checksum	—
Source Address	*Source Address*
Destination Address	*Destination Address*
Options	In Header-Erweiterungen verschoben

Die nachstehende Abbildung zeigt eine Übersicht über den IPv6-Header. Wie der Abbildung zu entnehmen ist, sind in dem Header nur noch relativ wenige Informationen untergebracht. Auch Traffic Class und Flow Label hätte man in einer Header-Erweiterung unterbringen können. Allerdings dienen diese Informationen der Verbesserung der Dienstgüte und sollten daher möglichst effizient ausgewertet werden können. Eine Unterbringung in einer Header-Erweiterung wäre hierzu kontraproduktiv.

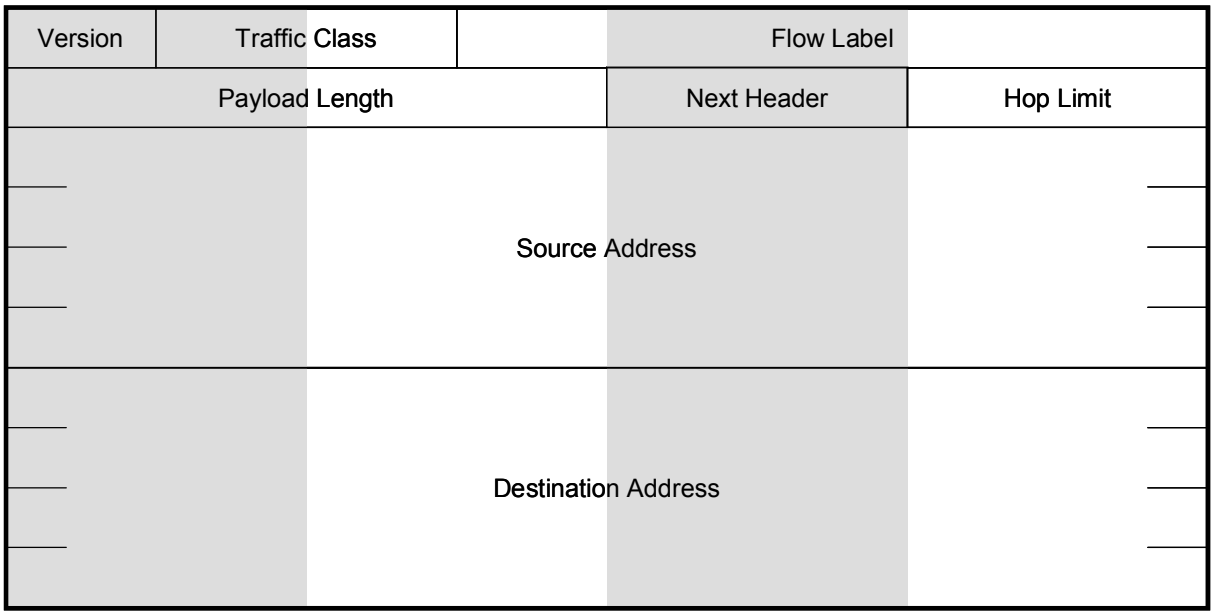

Abbildung 3: Der IPv6-Header

2.4.1 Header-Erweiterungen

Header-Erweiterungen werden durch eine Nummer identifiziert, die sich in das bestehende System der Protokollnummern einfügt, die aktuell gültige Liste ist bei IANA abrufbar.[13] RFC 2460 definiert die nachfolgend beschriebenen Header-Erweiterungen.

Es wird dabei auch der jeweils maximale *sinnvolle* Platzbedarf angegeben. Für die Kalkulation des Platzbedarfs ist zu beachten, dass jede Header-Erweiterung ein Vielfaches von 8 Bytes belegen muss [RFC 2460, Abschnitt 4], der Platzbedarf ist daher sowohl in Bytes als auch in Worten zu je 8 Bytes angegeben. Ungenutzter Platz muss mit Padding gefüllt werden, es wurde dabei zugrunde gelegt, dass kein unnötiges Padding eingesetzt wird, und dass nur nach derzeit gültigen Standards definierte Optionen und Options-Typen verwendet werden.

2.4.1.1 Hop-by-Hop Options Header

Wenn diese Erweiterung verwendet wird, dann muss sie an erster Stelle nach dem IPv6-Header stehen. Wenn ein IP-Paket diesen Header enthält, dann muss die Information darin von jedem Router auf dem Weg zum Ziel verarbeitet werden. Die einzelnen Optionen in dieser Header-Erweiterung sind variabel, die Codierung erfolgt nach dem sogenannten TLV-Schema, bei dem Typ, Länge und Wert nacheinander angegeben werden. Verschiedene Optionen müssen gemäß Definition an bestimmten Bytegrenzen innerhalb der der Header-Erweiterung ausgerichtet sein. Zu diesem Zweck gibt es die Padding Typen Pad1 und PadN, die hierfür je nach Bedarf zu nutzen sind.

Die Hop-by-Hop Header-Erweiterung wird derzeit für die folgenden Zwecke verwendet.

Jumbogramme
Sollen Jumbogramme [RFC 2675] übertragen werden, dann dient diese Erweiterung dazu, die Länge der Payload aufzunehmen. In diesem Fall muss die Länge der Payload im

13 IANA Protocol Numbers

eigentlichen IP-Header auf 0 gesetzt werden. Zudem schließt die Verwendung von Jumbogrammen Fragmentierung aus.

Die Länge dieser Option beträgt sechs Bytes und belegt einschließlich Padding 1 Wort.

Router Alerts
Router Alerts [RFC 2711] werden eingesetzt, um Router auf eine relevante Payload in einem IP-Paket hinzuweisen, das nicht an ihn selbst gerichtet ist. Derzeit werden Router Alerts in den folgenden Situationen verwendet:

- Wenn ein Node einer Multicast-Gruppe beitreten oder diese wieder verlassen möchte, dann muss der Node eine ICMPv6 Multicast Listener Discovery-Nachricht schicken. Im Fall von MLDv1 wird diese Nachricht beispielsweise an die entsprechende Multicast-Adresse der Gruppe verschickt. Der Router muss dies jedoch zur Kenntnis nehmen, um die Multicast-Routingtabellen entsprechend zu ändern.
- Wenn im Rahmen von QoS Bandbreite reserviert werden soll, dann muss hierzu eine RSVP-Nachricht verschickt werden. Auch in diesem Fall muss der Router die Nachricht zur Kenntnis nehmen, um die entsprechende Reservierung durchzusetzen.

Die Länge eines Router Alerts beträgt 4 Bytes und belegt somit (einschließlich minimalem Padding) 1 Wort. Jedes IP-Paket darf höchstens einen Router Alert enthalten.

2.4.1.2 Routing Header

Diese Header-Erweiterung muss von allen Routern auf dem Weg zum Ziel verarbeitet werden. Er dient der Beschreibung einer Route, die ein Paket nehmen soll. Es werden die folgenden Typen unterschieden:

- Typ 0 – Source Routing: Dies entspricht dem aus IPv4 bekannten Source Routing, die Sicherheitsprobleme sind dieselben wie dort, weswegen dieser Routing-Typ wieder abgeschafft wurde [RFC 5095].
- Typ 1 – Nimrod Routing: Über diesen Typ, der einem früheren Forschungsprojekt entstammt, „… ist wenig bekannt und noch weniger spezifiziert…“ [Potyraj 2007] und er wird laut IANA als überholt (deprecated) geführt.[14]
- Typ 2 – Mobile IPv6: Mit Hilfe dieses Routing Headers werden IP-Pakete an die Home Address eines mobilen Nodes versendet [RFC 6275].
- Typ 3 – RPL: Routing in verlustbehafteten Netzwerken [RFC 6554]. Dieser Header ist aus Sicherheitssicht eine Reinkarnation des Typs 0, allerdings schreibt RFC 6554 vor, dass dieser Header nur innerhalb eines definierten, unter einheitlicher Administration stehenden RPL-Bereichs verwendet werden darf und dass IP-Pakete mit einem Routing Header dieses Typs an den Grenzen eines solchen Bereichs verworfen werden müssen.

Die Länge der Routing Header hängt vom Typ ab, Routing Header Typ 0 und Typ 3 könnten prinzipiell bis zu 4086 Bytes einnehmen; Legt man das „übliche“ Hop Limit von 64 zugrunde, dann kommt man auf 1010 Bytes, die 127 Wörter belegen. Der Routing Header Typ 2 hat eine feste Länge von 24 Bytes und belegt 3 Wörter.

2.4.1.3 Fragment Header

Dieser Header-Erweiterung wird für jedes IP-Paket verwendet, das lediglich das Fragment eines ursprünglich größeren Pakets darstellt. Die Verwendung ist an die von IPv4 angelehnt.

---

14 IANA Internet Protocol Version 6 (IPv6) Parameters

Ein Fragment Header hat stets die Länge von 8 Bytes und belegt ein Wort.

2.4.1.4 Destination Options Header

Diese Header-Erweiterung kann im Gegensatz zu allen anderen Erweiterungen zweimal verwendet werden. Steht dieser Header vor dem Routing Header, dann ist er auch von Routern entlang des Wegs, den das Paket nimmt, auszuwerten, ansonsten nur vom Ziel-Node. Genau wie bei der Hop-by-Hop Header-Erweiterung sind die einzelnen Optionen in dieser Header-Erweiterung variabel, die Codierung erfolgt auch hier nach dem TLV-Schema.

Dieser Header wird beispielsweise zur Angabe der folgenden Optionen verwendet:

Tunnel Encapsulation Limit
Werden Tunnelmechanismen in einer Weise eingesetzt, bei der es zu verschachtelter Tunnelung kommen kann, dann kann mit dieser Option die Schachtelungstiefe begrenzt werden [RFC 2473].

Diese Option ist 3 Bytes lang und belegt ein Wort.

Mobile IPv6 Home Address
Ist ein mobiler Node nicht am Heimatort, dann wird diese Option dazu verwendet, um einen Empfänger über die Home Address des absendenden mobilen Nodes zu informieren [RFC 6275].

Diese Option ist 18 Bytes lang und belegt 3 Wörter.

RFC 4302 und RFC 4303 definieren darüber hinaus noch die in den nachfolgenden Abschnitten beschriebenen, für IPsec relevanten Header-Erweiterungen.

2.4.1.5 Authentication Header

Diese Header-Erweiterung wird zur Authentisierung von IP-Paketen bei der Nutzung von IPsec verwendet.

Die Länge dieser Header-Erweiterung beträgt 12 Bytes plus Länge der verwendeten Checksumme. Üblich sind verkürzte Checksummen von 12 Bytes Länge, denkbar ist aber eine Länge von 64 Bytes (etwa bei der unverkürzten Verwendung von HMAC-SHA-512), so dass in der Praxis eine Länge von maximal 76 Bytes, d. h 10 Wörtern anzunehmen ist.

2.4.1.6 Encapsulating Security Payload Header

Diese Header-Erweiterung wird zur Verschlüsselung und zur optionalen Authentisierung der Nutzlast von IP-Paketen im Zusammenhang mit IPsec verwendet und unterscheidet sich insofern von allen anderen Header-Erweiterungen, als dass diese Header-Erweiterung bereits die (verschlüsselte) Payload des IP-Pakets enthält.

2.4.1.7 Mobility Header

Der Moblity Header wurde im Zusammenhang mit Mobile IPv6 in RFC 6275 definiert. Dieser Header wird eingesetzt, um die Bindung zwischen sogenannter Care-Of-Adresse und Home-Adresse zu erzeugen und zu verwalten. Mobile IPv6 ist nicht Gegenstand dieses Artikels, daher wird auf die Details nicht weiter eingegangen. Je nach Typ und Packung der Optionen kann dieser Header bis zu 16 Wörter belegen.

2.4.1.8 Verkettung mehrerer Header-Erweiterungen

Header-Erweiterungen werden miteinander verkettet. Jede Header-Erweiterung enthält ein Feld *Next Header* (NH), das darüber Auskunft gibt, was nach dieser Erweiterung folgt. Wenn keine Erweiterung mehr kommt, dann folgt üblicherweise die Payload, also beispielsweise ein TCP-Paket, wie in Abbildung 4 für drei denkbare Fälle dargestellt. Als NH-Eintrag für die nachfolgende Payload wird in der letzen Header-Erweiterung die Protokoll-Nummer des verwendeten Upper-Layer Protocols verwendet.

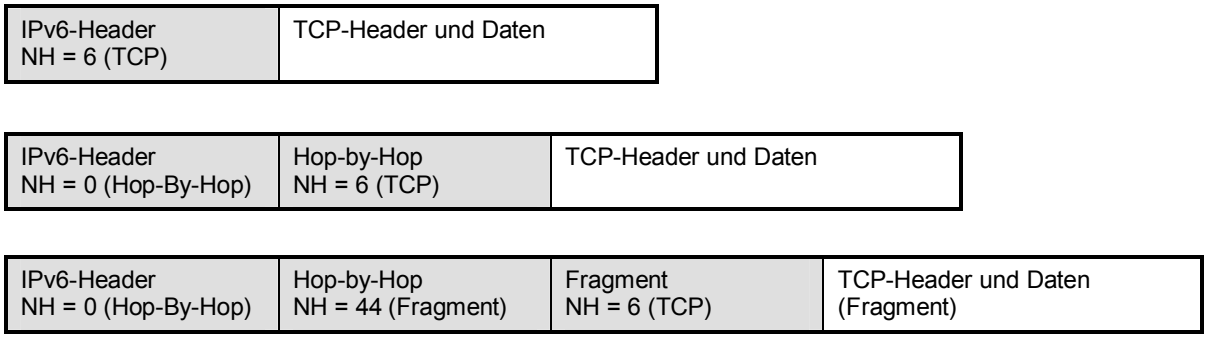

Abbildung 4: Verkettung von Header-Erweiterungen

Jede Header-Erweiterung (mit Ausnahme des Destination Options Header) darf höchstens einmal in einem IP-Paket verwendet werden. Header-Erweiterungen sind optional und werden nur dann verwendet, wenn die darin enthaltenen Informationen übermittelt werden müssen. Wird mehr als eine Header-Erweiterung verwendet, dann sollte die folgende Reihenfolge eingehalten werden:

- IPv6-Header
- Hop-By-Hop-Options Header[15]
- Destination Options Header (für den Empfänger des IP-Pakets[16])
- Routing Header
- Fragment Header
- Authentication Header
- Encapsulating Security Payload Header
- Destination Options Header (für den finalen Empfänger des IP-Pakets[16])
- Mobility Header
- Upper-Layer Header

Für den Fall von Kapselung oder Tunnelung kann anstelle des Upper-Layer Headers auch ein IPv6-Header kommen, dem seinerseits Header-Erweiterungen folgen können – diese

---

15 Wenn der Hop-by-Hop Option Header verwendet wird, dann **muss** dieser zwingend an erster Stelle nach dem eingentlichen IPv6-Header kommen.

16 Die Destination Options gelten für den Empfänger, dessen Adresse im Destination Address-Feld des IP Headers steht. Wenn ein Routing Header vorhanden ist, dann ist der finale Empfänger jedoch nicht identisch mit dem ursprünglichen Empfänger. Die Destination Options *vor* dem Routing Header gelten für jeden Node, dessen Adresse im Destination Address-Feld des IP Headers auftaucht; die Destination Options *nach* dem Routing Header gelten nur für den finalen Empfänger.

Header gelten als Nutzlast des kapselnden IP-Pakets und werden bei der Zählung der Header-Erweiterungen des kapselnden IP-Pakets nicht berücksichtigt.

Header-Erweiterungen müssen in der Reihenfolge ausgewertet werden, in der sie im IP-Paket erscheinen – ein Node darf beispielsweise nicht nach bestimmten Erweiterungen suchen und diese zuerst auswerten. Erweiterungen, die von Routern ausgewertet werden sollen, müssen zuvorderst stehen. Es erscheint jedoch plausibel, dass ein Firewall-Router gegen diese Regel ignoriert, um effizienter zu einer Filterentscheidung zu kommen.

IP-Pakete, deren Header durch exzessives Padding, die Nutzung unbekannter Optionen oder (ohnehin nicht standardkonforme[17]) mehrfache Verwendung von Header-Erweiterungen die MTU überschreiten, oder die so ungünstig fragmentiert sind, dass die für eine Filterentscheidung notwendigen Informationen nicht im ersten Fragment enthalten sind, sollten daher verworfen werden.

2.5 Fragmentierung

Fragmentierung funktioniert bei IPv6 im Prinzip genauso wie bei IPv4. Die wesentlichen Unterschiede bestehen darin, dass die Informationen zur (De-)Fragmentierung in einer eigenen Header-Erweiterung untergebracht sind und dass grundsätzlich nur der Absender fragmentiert. Dies hat verschiedene Konsequenzen:

- Überschreitet auf der Strecke von Absender zu Empfänger die Länge eines Pakets die MTU auf einem Streckenabschnitt, dann wird – anders als bei IPv4 – grundsätzlich das inkriminierte Paket verworfen und dem Empfänger eine entsprechende ICMP-Fehlermeldung zurückgeschickt. Das bedeutet, dass derartige Fehlermeldungen an Firewalls o. ä. nicht gefiltert werden dürfen, um nicht die Übertragungsfunktionalität grundlegend zu gefährden.
- Da die Fragmentierung durch den Absender vorgenommen wird, findet keine mehrfache Fragmentierung statt, d. h. es gibt keinen legitimen Grund für mehr als eine Fragment Header-Erweiterung pro Paket.

2.6 ICMPv6

ICMP hat unter IPv6 eine deutlich größere Bedeutung, als dies bei IPv4 der Fall war. So sind die Funktionen von ARP und RARP in ICMPv6 aufgegangen, ein ARPv6 existiert nicht. Für den praktischen Einsatz bedeutet dies, dass ICMPv6 nicht pauschal gefiltert werden kann, wie das etwa bei ICMPv4 üblich ist. Auch die Funktion von IGMP wurde in ICMPv6 aufgenommen.

2.6.1 Neighbor Discovery

Die Aufgaben der Neighbor Discovery umfassen die folgenden Punkte:

- Router Discovery
- Prefix Discovery
- Parameter Discovery
- Automatische Adresskonfiguration
- Adressauflösung (Neighbor Discovery und Inverse Neighbor Discovery)

[17] Mit Ausnahme der geschilderten zweifachen Verwendung des Destination Options Headers.

- Bestimmung des Next-Hop
- Neighbor Unreachability Detection (NUD)
- Duplicate Address Detection (DAD)
- Redirect

Das Neighbor Discovery Protocol (NDP) wird in RFC 4861 beschrieben, Stateless Address Autoconfiguration (SLAAC) wird in RFC 4862 beschrieben; Inverse Neighbor Discovery ist in RFC 3122 beschrieben.

Für die Neighbor Discovery wurden ursprünglich fünf ICMPv6-Pakettypen definiert; für die Inverse Neighbor Discovery wurden zwei weitere Pakettypen definiert. Die entsprechenden ICMPv6-Nachrichten dürfen am Link nicht gefiltert werden, andernfalls kann keine Kommunikation über IPv6 stattfinden. Andererseits haben die entsprechenden Nachrichten nur link-local eine Bedeutung – Administratoren müssen also dafür sorgen, dass NDP-Nachrichten nicht geroutet werden. In RFC 4861 wurde zu diesem Zweck festgelegt, dass NDP-Nachrichten das maximale Hop Limit von 255 haben müssen; NDP-Nachrichten mit einem geringeren Hop Limit müssen verworfen werden. Damit wird sichergestellt, dass von „extern“ eingebrachte NDP-Nachrichten keinen Schaden anrichten können.

Beispielszenario

Im Folgenden werden einige Beispiele zur Illustration verwendet. Auf den Nodes der Beispielszenarien ist ein Linux-Betriebssystem installiert. Die Interface-Adressen der beteiligten Hosts sind in der nachstehenden Tabelle aufgeführt.

Name	MAC-Adresse	IPv6-Adresse
Gondor	`00:0c:29:51:ee:69`	`2001:db8:ca84:e2d5:16b1:d70d:6a2b:bdc6`
Arnor	`00:0c:29:08:45:4b`	`2001:db8:ca84:e2d5:928a:701e:50b:3f2a`

2.6.2 Router Discovery

Router Discovery spielt eine Rolle bei der Bestimmung von Routern, bei der Verteilung von Präfix-Information und bei der Bestimmung, ob eine Adresse on-link oder off-link ist. Ein Router kann die Hosts über ein Router Advertisement auch anweisen, Adresskonfiguration oder andere Konfigurationsparameter über DHCPv6 zu beziehen. Als Erweiterung der Router Discovery schlägt RFC 6106 auch die Verbreitung von DNS-relevanten Konfigurationsparametern vor.

Für die Router Discovery sendet ein Router in regelmäßigen Abständen ein sogenanntes Router Advertisement aus. Diese Nachricht wird an die Link-Local All-Nodes Multicast-Adresse (`ff02::1`) gesendet.

Will ein Host nicht auf ein Router Advertisement warten, dann sendet er eine sogenannte Router Solicitation. Der Router kann darauf mit einer Multicast-Nachricht an alle oder mit einer Unicast-Nachricht an den fragenden Node antworten.

Beispiel im Detail

Beim Bootvorgang wird am Host Arnor das Interface eth1 initialisiert. Dazu wird zunächst die Duplicate Address Detection (siehe unten) für die Link-Local-Adresse (`fe80::20c:29ff:fe08:454b`) durchlaufen. Nach erfolgreicher Initialisierung versendet der Host eine Router Solicitation wie in Abbildung 5 dargestellt.

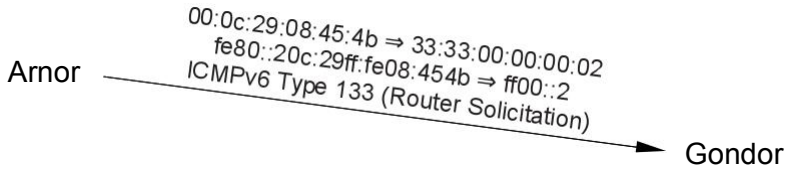

Abbildung 5: Router Solicitation

In unserem Fall antwortet der Router an die Link-Local All-Nodes Multicast-Adresse wie in Abbildung 6 gezeigt. Tatsächlich enthält das Router Advertisement noch weitere Informationen, wie beispielsweise die Link-Layer-Adresse des Routers.

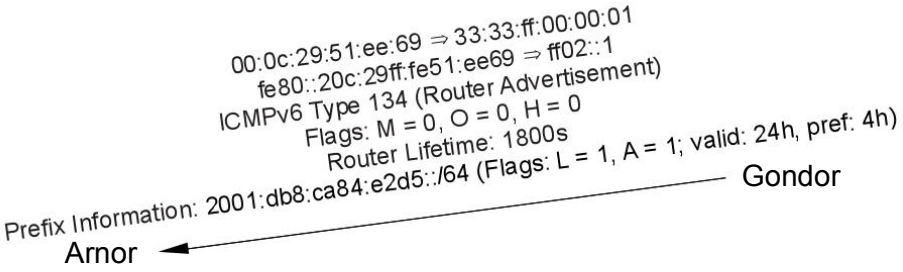

Abbildung 6: Router Advertisement

Die Flags im Router Advertisement sagen aus, dass keine weiteren Informationen über DHCP bezogen werden können (M = 0 und O = 0), und dass der Router kein sogenannter Home Agent ist (H = 0).[18] Ein Router Advertisement kann (im Prinzip) beliebig viele Präfixe enthalten. Für jedes Präfix werden Flags und Parameter mitgegeben. In diesem Fall wird das Präfix `2001:db8:ca84:e2d5::/64` verteilt, und es gelten alle Nodes mit diesem Präfix als on-link (L = 1), d. h. Pakete an Adressen mit diesem Präfix können direkt auf dem Link verschickt werden und brauchen nicht über den Router zu gehen. Außerdem darf das Präfix für SLAAC verwendet werden (A = 1). Wenn das Interface eth1 auf Host Arnor für SLAAC konfiguriert ist, dann wird dem Interface eth1 neben der Link-Local-Adresse noch die Adresse `2001:db8:ca84:e2d5:20c:29ff:fe08:454b` zugewiesen, wie die nachstehende Ausgabe zeigt:

```
arnor ~ # ip -6 addr show dev eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 2001:db8:ca84:e2d5:20c:29ff:fe08:454b/64 scope global dynamic
       valid_lft 86392sec preferred_lft 14392sec
    inet6 fe80::20c:29ff:fe08:454b/64 scope link
       valid_lft forever preferred_lft forever
```

Da die Router Lifetime größer als Null ist, wird der Router als Defaultrouter eingetragen:

```
arnor ~ # ip -6 route show
2001:db8:ca84:e2d5::/64 dev eth1  proto kernel  metric 256  expires 86341sec
fe80::/64 dev eth1  proto kernel  metric 256
ff00::/8 dev eth1  metric 256
default via fe80::20c:29ff:fe51:ee69 dev eth1  proto kernel  metric 1024  expires
1736sec hoplimit 64
```

2.6.3 Prefix Discovery

Prefix Discovery wird im Rahmen der Router Solicitation und Router Advertisement durchgeführt, siehe Abschnitt 2.6.2.

---

18 Home Agents sind Bestandteil der Mobile IPv6 Architektur und haben im Kontext des Routings für mobile Hosts eine Bedeutung.

2.6.4 Stateless Address Autoconfiguration (SLAAC)

Stateless Address Autoconfiguration wird im Rahmen der Duplicate Address Detection (DAD, siehe Abschnitt 2.6.8) mit anschließender Prefix Discovery durchgeführt.

2.6.5 Adress-Auflösung

Ein Node, der ein Paket an einen Neighbor auf dem Link senden möchte, muss dessen Link-Layer-Adresse kennen. Die Auflösung von IPv6-Adressen zu Link-Adressen (im Fall von Ethernet oder WLAN also MAC-Adressen) erfolgt mit Hilfe der Neighbor Discovery. Zur Auflösung einer Unicast-Adresse sendet der fragende Node eine Neighbor Solicitation an die Solicited-Node Multicast-Adresse, die der fraglichen Unicast-Adresse entspricht. Die Neighbor Solicitation enthält bereits die Link-Layer-Adresse des anfragenden Nodes. Der Ziel-Node antwortet dann per Unicast mit einem Neighbor Advertisement, das seine eigene Link-Layer-Adresse enthält.

Beispiel im Detail

Von Host Arnor (`2001:db8:ca84:e2d5:928a:701e:050b:3f2a`) soll ein Paket an den Router Gondor (`2001:db8:ca84:e2d5:16b1:d70d:6a2b:bdc6`) gesendet werden. Im Neighbor Cache auf Arnor findet sich kein Eintrag für die Ziel-Adresse. Es muss also eine Adressauflösung durchgeführt werden. Abbildung 7 zeigt eine für diesen Fall typische Neighbor Solicitation. Diese wird an die Solicited-Node Multicast-Adresse `ff02::1:ff2b:bdc6` gesendet, die auf Link-Layer an die entsprechende Ethernet-Multicast-Adresse `33:33:ff:2b:bd:c6` adressiert wird. Die Neighbor Solicitation enthält dabei die Link-Layer-Adresse des fragenden Nodes (Source Link-Layer Address Option).

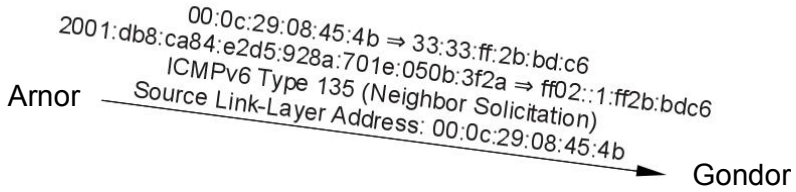

Abbildung 7: Neighbor Solicitation zur Adressauflösung

Abbildung 8 zeigt das typische Neighbor Advertisement als Antwort auf die Neighbor Solicitation. Da der fragende Node bereits seine Link-Layer-Adresse in der Neighbor Solicitation mitgeliefert hat, kann das antwortende Neighbor Advertisement per Unicast an den fragenden Node zurück gesendet werden.

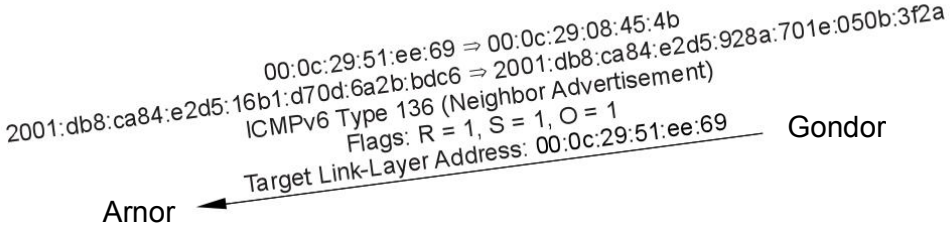

Abbildung 8: Neighbor Advertisement zur Adressauflösung

Über die zusätzliche Flags teilt der antwortende Node mit, dass er ein Router ist (R = 1), dass das Neighbor Advertisement die Antwort auf eine Neighbor Solicitation ist (S = 1), und

dass eventuell vorhandene Daten im Neighbor Cache durch dieses Neighbor Advertisement aktualisiert werden sollen (O = 1).

Der Neighbor Cache enthält nach der Anfrage die Adresse des Ziel-Nodes:

```
arnor ~ # ip -6 neigh show
fe80::20c:29ff:fe51:ee69 dev eth1 lladdr 00:0c:29:51:ee:69 router STALE
2001:db8:ca84:e2d5:16b1:d70d:6a2b:bdc6 dev eth1 lladdr 00:0c:29:51:ee:69 router
REACHABLE
```

2.6.6 Bestimmung des Next Hop

Die Bestimmung des geeigneten Next Hop über Redirect-Nachrichten ist im Wesentlichen dann relevant, wenn sich mehrere Router in einem Netz mit Endgeräten befinden. Wenn ein Host ein Paket an einen anderen Node senden möchte, der off-link ist, dann muss das Paket über einen Router verschickt werden. Aber über welchen? Wenn der Host das Paket an einen bestimmten Router verschickt, während tatsächlich ein anderer Router für Pakete an die betreffende Zieladresse geeigneter wäre, dann informiert der zunächst verwendete Router den Host darüber, dass für Pakete an diese Zieladresse der andere Router zuständig ist. Diese Information erfolgt über eine entsprechende ICMP-Redirect-Nachricht an den sendenden Host. Dieser Vorgang ist prinzipiell auch Bestandteil der IPv4-Spezifikation, wurde aber praktisch nie für legitime Zwecke genutzt, zumal Redirect-Nachrichten prädestiniert für Man-in-the-Middle-Angriffe sind. Es wird sich noch zeigen, wie sich die praktische Relevanz und die Nutzung von Redirects bei IPv6 entwickelt. Da IPv6-Redirects Teil der Neighbor Discovery sind und als solche nur link-local wirken, muss ein Angreifer zumindest Zugang zu dem entsprechenden LAN haben.

2.6.7 Neighbor Unreachability Detection (NUD)

Neighbor Unreachability Detection ist ein Mechanismus, um festzustellen, ob die Kommunikation *zu* einem Interface an einem Nachbar-Node funktioniert. Damit soll vermieden werden, dass Pakete an ein nicht erreichbares Interface gesendet werden. Dies kann beispielsweise passieren, wenn der Nachbar-Node oder dessen Interface ausfällt. Der Zustand, ob ein Nachbar erreichbar ist, wird mit dem entsprechenden Eintrag im Neighbor Cache[19] abgelegt. Eine kurze Erläuterung der möglichen Zustände von Einträgen im Neighbor Cache ist nachstehend gegeben, die genauen Details finden sich in Abschnitt 7.3 sowie Anhang C aus RFC 4861.

Incomplete
In diesem Zustand wird gerade eine Adressauflösung durchgeführt: eine Neighbor Solicitation wurde gesendet, aber das dazugehörige Neighbor Advertisement wurde noch nicht empfangen. Wird die Antwort empfangen, dann geht der Eintrag in den Zustand *Reachable* über, andernfalls wird eine ICMP-Fehlermeldung (Destination Unreachable) zurückgegeben.

Reachable
Die Erreichbarkeit des entsprechenden Nodes wurde bestätigt. Erfolgt innerhalb einer bestimmten Zeitspanne seit der letzten Bestätigung der Erreichbarkeit des Nodes keine erneute Bestätigung, dann geht der Eintrag in den Zustand *Stale* über. Die Bestätigung der Erreichbarkeit kann entweder über Neighbor Discovery erfolgen, oder über Hinweise aus dem Upper-Layer Protocol, dass eine Verbindung aktiv ist.

Stale
Die Erreichbarkeitsbestätigung ist abgelaufen. Der Eintrag verbleibt in diesem Zustand, bis

19 Der Neighbor Cache ist das IPv6-Äquivalent zu dem ARP-Cache bei IPv4.

ein Paket an den entsprechenden Nachbar-Node gesendet werden soll; in diesem Fall geht der Eintrag in den Zustand *Delay* über.

Delay
In diesem Zustand wird geprüft, ob eine Erreichbarkeitsbestätigung aus den Protokoll-Zuständen des Upper-Layer Protocols gefolgert werden kann. Wenn dies innerhalb einer bestimmten Zeit nicht möglich ist, dann wird die Neighbor Unreachability Detection durch Neighbor Discovery ausgelöst, und der Eintrag geht in den Zustand *Probe* über.

Probe
Innerhalb eines festgelegten Zeitraums werden Neighbor Solicitations verschickt, bis ein entsprechendes Neighbor Advertisement empfangen wird; nach Ablauf der Zeit wird der -Eintrag aus dem Neighbor Cache entfernt und ggf. eine erneute Adressauflösung eingeleitet.

Die Information, ob ein Interface erreichbar ist, kann durch ein Ereignis auf der Transportschicht gewonnen werden, beispielsweise wenn bei einer Kommunikation über TCP ein Acknowledgement an den lokalen Node zurück gesendet wird. Steht keine solche Information zur Verfügung, dann wird auf Neighbor Discovery zurückgegriffen.

Die Neighbor Unreachability Detection wird genau wie Adressauflösung oder Duplicate Address Detection über ein Paar aus Neighbor Solicitation und Neighbor Advertisement durchgeführt. Im Unterschied zu diesen Fällen ist die MAC-Adresse der beteiligten Nodes bekannt und soll durch die NUD lediglich bestätigt werden. Genaugenommen soll bestätigt werden, dass die Übertragung *vom lokalen Host zum entfernten Host* (Forward-Path Confirmation) funktioniert [RFC 4861, Abschnitt 7.3.1]. Daher dürfen unaufgeforderte (unsolicited) Neighbor Advertisements *nicht* für die NUD herangezogen werden.

Beispiel im Detail

Node Arnor kommuniziert über UDP mit dem Router Gondor. UDP liefert keine Informationen darüber, ob der Adressat eines Pakets das Paket tatsächlich erhalten hat. Die Einträge in den Neighbor Caches gehen nach einer Zeit in den Zustand *Stale* über.

```
arnor ~ # ip -6 neigh show
fe80::20c:29ff:fe51:ee69 dev eth1 lladdr 00:0c:29:51:ee:69 router STALE
```

Beim Versand des nächsten UDP-Pakets wird die Neighbor Unreachability Detection mittels Neighbor Discovery eingeleitet. Hierzu sendet beispielsweise Arnor die in Abbildung 9 dargestellte Neighbor Solicitation an Gondor. Anders als bei Adressauflösung oder Duplicate Address Detection wird die Neighbor Solicitation per *Unicast* verschickt.

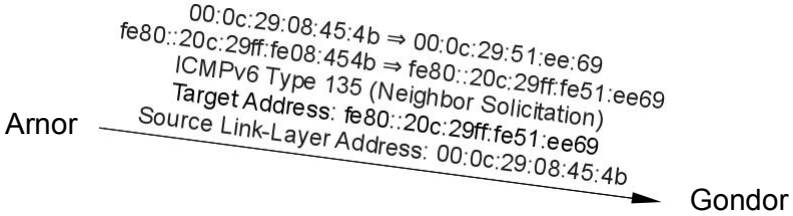

Abbildung 9: Neighbor Solicitation für Neighbor Unreachability Detection

Abbildung 10 zeigt die entsprechende Antwort. Die Flags zeigen an, dass Gondor ein Router ist (R = 1) und dass das Advertisement die Antwort auf eine Solicitation ist (S = 1).

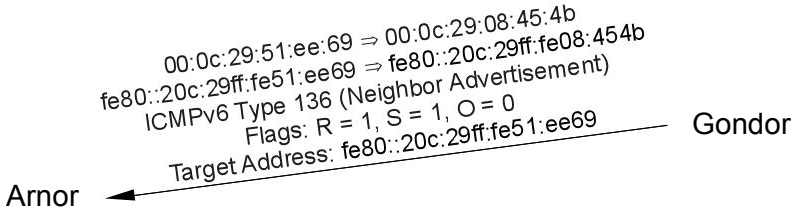

Abbildung 10: Neighbor Advertisement für Neighbor Unreachability Detection

Durch das Neighbor Advertisement wechselt der Eintrag für die Adresse `fe80::20c:29ff:fe51:ee69` im Neighbor Cache auf Arnor in den Zustand *Reachable* über, wie nachstehend gezeigt ist; der Eintrag selbst wird jedoch nicht verändert (O = 0).

```
arnor ~ # ip -6 neigh show
fe80::20c:29ff:fe51:ee69 dev eth1 lladdr 00:0c:29:51:ee:69 router REACHABLE
```

2.6.8 Duplicate Address Detection (DAD)

RFC 4862 schreibt vor, dass vor jeder Zuweisung von Unicast- oder Anycast-Adressen an ein Interface die sogenannte Duplicate Address Detection durchzuführen ist, um zu verhindern, dass mehrere Interfaces auf demselben Link dieselbe IP-Adresse verwenden. Hierfür sendet der Node über das entsprechende Interface eine Neighbor Solicitation an alle Interfaces auf dem Link. Wenn nicht nach einer kurzen Zeit eine Antwort in Form eines widersprechenden Neighbor Advertisements erfolgt, dann wird davon ausgegangen, dass die entsprechende Adresse nicht von einem anderen Interface verwendet wird.

Beispiel im Detail

Auf dem Node Arnor soll dem Interface eth1 die Unicast-Adresse `2001:db8:ca84:e2d5:16b1:d70d:6a2b:bdc6` zugewiesen werden. Zunächst muss das Interface als Listener für die Multicast-Gruppe Link-Local All-Nodes (`ff02::1`) registriert werden. Dies ist erforderlich, da IPv6 kein Broadcast kennt, eventuelle Antworten auf die nun folgende Neighbor Solicitation aber verarbeitet werden müssen. Außerdem muss das Interface als Listener für die der vorläufigen Unicast-Adresse entsprechenden Solicited-Node Multicast-Adresse (`ff02::1:ff2b:bdc6`) eingetragen werden.

Im Rahmen der DAD ist diese Adresse zunächst vorläufig (tentative), d. h. diese Adresse kann vorerst nicht verwendet werden – der Node darf diese Adresse nicht als Quelladresse verwenden und darf Pakete mit dieser Zieladresse nicht annehmen.

Für die DAD wird die Neighbor Solicitation folgendermaßen genutzt (siehe auch Abbildung 11): Als IP-Quelladresse wird die unspezifizierte Adresse `::` und als IP-Zieladresse die der fraglichen Unicast-Adresse entsprechende Solicited-Node Multicast-Adresse verwendet. Auf Data Link-Ebene wird als Quelladresse die MAC-Adresse des Interfaces auf Arnor (`00:0c:29:08:45:4b`) verwendet, während als Zieladresse die der Solicited-Node Multicast-Adresse zugeordnete Ethernet-Multicast-Adresse verwendet wird (`33:33:ff:2b:bd:c6`).

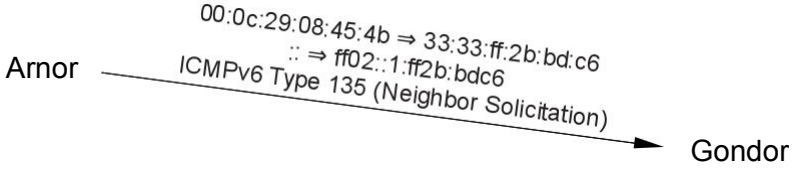

Abbildung 11: Neighbor Solicitation für Duplicate Address Detection

Wird einem Interface eine Unicast-Adresse zugewiesen, dann muss das Interface auch Listener der entsprechenden Solicited-Node Multicast-Adresse sein. Somit würde ein Knoten auf demselben Link, der die fragliche Unicast-Adresse bereits belegt, diese Nachricht empfangen und sofort ein widersprechendes Neighbor Advertisement verschicken, siehe Abbildung 12. Hierbei verwendet der widersprechende Node als IP-Quelladresse die bereits von ihm beanspruchte Unicast-Adresse, während als IP-Zieladresse die Link-Local All-Nodes Multicast-Adresse verwendet. Auf Data Link-Ebene verwendet der widersprechende Node als Quelladdresse die eigene MAC-Adresse (`00:0c:29:51:ee:69`) und als Zieladresse die der Link-Local All-Nodes Multicast-Adresse entsprechende Ethernet Multicast-Adresse (`33:33:00:00:00:01`).

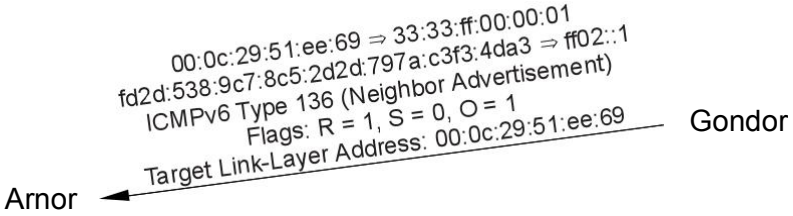

Abbildung 12: Neighbor Advertisement für Duplicate Address Detection

Erhält ein Node im Rahmen der DAD ein Neighbor Advertisement, dann darf er die entsprechende Unicast-Adresse nicht zuweisen. In diesem Fall sähe eine Statusabfrage etwa so aus:

```
arnor ~ # ip -6 addr show dev eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 2001:db8:ca84:e2d5:16b1:d70d:6a2b:bdc6/64 scope global tentative
dadfailed
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe08:454b/64 scope link
       valid_lft forever preferred_lft forever
```

2.7 Path MTU Discovery

Analog zu IPv4 bietet IPv6 die Möglichkeit der Path MTU Discovery. Im Gegensatz zu IPv4 hat Path MTU Discovery bei IPv6 einen deutlich höheren Stellenwert, da Pakete bei IPv6, die schon auf dem Weg vom Absender zum Empfänger sind, grundsätzlich nicht fragmentiert werden dürfen. Stattdessen passiert Folgendes [RFC 1981]: Erreicht ein Paket einen Steckenabschnitt, dessen MTU kleiner ist als die Größe des inkriminierten Pakets, dann wird das Paket verworfen und die ICMP-Fehlermeldung Packet Too Big mit der maximal für den Weitertransport möglichen MTU an den Absender geschickt. Der ursprüngliche Absender kann das Paket dann selbst fragmentieren oder auf einer höheren Protokollschicht den Versand kleinerer Pakete veranlassen. Dieser Prozess kann sich auf dem Weg vom Absender zum Empfänger prinzipiell beliebig oft wiederholen. Für Echtzeit-Anwendungen wie Multimedia-Streaming sollte die Path MTU daher bereits im Vorweg bestimmt werden.

Die Details und die Wirkung auf höhere Protokollschichten der Path MTU Discovery sind in RFC 1981 beschrieben.

3 Bedrohungen und grundsätzliche Gegenmaßnahmen

In diesem Anschnitt werden verschiedene Bedrohungen gegen IPv6 oder durch IPv6 oder im Zusammenhang mit IPv6 dargestellt und diskutiert, siehe auch die aus [ISi-LANA, Abschnitt 7] anwendbaren Punkte.

3.1 Neighbor Discovery

Die Neighbor Discovery umfasst zahlreiche Funktionen, von denen die Kommunikation über IPv6 abhängt. Die Neighbor Discovery ist a priori ungeschützt, insbesondere sind ND-Nachrichten weder authentisiert, noch gegen Manipulation geschützt. So kann ein Angreifer durch ND-Spoofing sowohl Man-in-the-Middle-Angriffe als auch Denial-of-Service-Angriffe durchführen. Eine Analyse von Bedrohungen gegen die Neighbor Discovery allgemein wurde bereits in RFC 3756 durchgeführt; RFC 6104 geht im Speziellen auf Router Advertisements ein. Fernando Gont hat die Neighbor Discovery eingehend analysiert und dabei Informationen aus verschiedenen Quellen zusammengetragen [Gont 2013a] und konkrete Hinweise zur Ausnutzung von Schwachstellen und deren Konsequenzen gegeben [Gont 2012a].

3.1.1 Bedrohungen

Im Einzelnen sind die folgenden Angriffe auf die Neighbor Discovery möglich, wenn der Angreifer Zugang zum Link hat:

- Ein Angreifer kann Nodes mit ND-Nachrichten (und anderen ICMPv6-Nachrichten) fluten. Dadurch könnte die Netzwerkbandbreite oder die Rechenleistung der angegriffenen Nodes ausgelastet werden. Durch fehlerhafte Implementierungen kann es sogar zu Abstürzen kommen. Wenn die Implementierung beispielsweise nicht die Anzahl der Einträge im Neighbor Cache begrenzt, kann durch einen einfach durchzuführenden Angriff der Betriebssystemspeicher komplett belegt werden. Beispiele für anfällige Systeme sind FreeBSD 9.0 und NetBSD 5.1 [Gont 2013a].

  Eine spezielle Variante davon, die auch off-link leicht durchgeführt werden kann, besteht in der Anfrage nach Nodes mit nicht-existierenden Adressen, beispielsweise durch Aufzählung eines Netzwerks mit einem Portscanner, siehe Abschnitt 4.2. Der Router, der für das entsprechende Präfix zuständig ist, muss die entsprechenden Neighbor Solicitations versenden und auf die entsprechenden Neighbor Advertisements warten. Bei einer Flutung mit Anfragen nach nicht-existieren Adressen könnte der dafür vorgesehene Speicher komplett belegt werden, so dass legitime Anfragen nach bestehenden Nodes nicht mehr verarbeitet werden können.

- Ein Angreifer kann gefälschte Router Advertisements verschicken. Zum Zweck eines Man-in-the-Middle-Angriffs kann der Angreifer seinen eigenen Node zum bevorzugten Router für ein oder mehrere bestehende Präfixe erklären. Zum Zweck eines Denial-of-Service-Angriffs wird ein nicht existierender Node zum Router erklärt.

- Ein Angreifer kann zum Zweck eines Denial-of-Service-Angriffs über gefälschte Router Advertisements bestehende Präfixe ungültig machen oder neue Präfixe verbreiten. Hierbei würden die entsprechenden Präfixe von allen Nodes am Link (fälschlicherweise) als on-link betrachtet werden. Anstelle einer Weiterleitung an einen Router würde für Adressen mit den betroffenen Präfixen eine (vergebliche) Neighbor Discovery durchgeführt.

- Ein Angreifer kann über gefälschte Router Advertisements falsche Parameter verbreiten. Denkbar ist beispielsweise die Verbreitung eines so niedrigen Hop Limits, dass Kommunikation im Extremfall nur noch link-local stattfinden kann. Denkbar ist beispielsweise auch die Verbreitung von gefälschten Router Advertisements mit gesetztem M-Flag, wodurch die Hosts am Link angewiesen werden, ihre Adresse von einem (nicht existierenden) DHCPv6-Server zu beziehen.

- Ein Angreifer kann einen Node während der Duplicate Address Detection davon abhalten, ein Interface mit einer Adresse zu konfigurieren. Hierbei wird zum Zweck eines Denial-of-

Service-Angriffs jede entsprechende Neighbor Solicitation (d. h. mit Quelladresse `::` und einer Link-Local Solicited-Node Multicast-Adresse als Zieladresse) mit einem widersprechenden Neighbor Advertisement beantwortet.

- Ein Angreifer kann zum Zweck eines Denial-of-Service-Angriffs einen Node im Rahmen der Neighbor Unreachability Detection davon überzeugen, dass ein Node erreichbar ist, der eigentlich nicht mehr erreichbar ist. Der angegriffene Node geht dann fälschlicherweise davon aus, dass der betroffene Node (beispielsweise ein Router) Pakete korrekt entgegennimmt und verarbeitet bzw. weiterleitet.
- Ein Angreifer kann während der Adressauflösung gefälschte Neighbor Advertisements verschicken, bei denen die im Neighbor Advertisement angegebene Link-Layer-Adresse gefälscht ist. Dieser Angriff entspricht dem „klassischen" ARP-Spoofing bei IPv4. Zum Zweck eines Man-in-the-Middle-Angriffs könnte der Angreifer die eigene Link-Layer-Adresse verwenden, zum Zweck eines Denial-of-Service-Angriffs könnte der Angreifer auf eine nicht existierende Link-Layer-Adresse verweisen.

  Als Spezialfall hiervon könnte auch die Link-Layer Broadcast-Adresse oder eine Link-Layer Multicast-Adresse verwendet werden.
- Ein Angreifer kann zum Zweck eines Denial-of-Service-Angriffs gefälschte Neighbor Advertisements im Namen eines existierenden Routers verschicken, bei denen im Neighbor Advertisement das Router-Flag nicht gesetzt ist. Dadurch würde der Router von den Hosts am Link nicht mehr als Router betrachtet werden und alle Routen über den betreffenden Router würden aus den Routingtabellen der Hosts entfernt werden.
- Ein Angreifer kann gefälschte Redirect-Nachrichten verschicken. Zum Zweck eines Man-in-the-Middle-Angriffs könnte der Angreifer auf den eigenen Node verweisen, zum Zweck eines Denial-of-Service-Angriffs könnte der Angreifer einen nicht existierenden Node verweisen.

Noch weitere, hier nicht aufgeführte Angriffe sind unter bestimmten Umständen denkbar. Diese hängen beispielsweise von der Qualität der jeweiligen Implementierungen ab. Ein Node, dessen IPv6-Implementierung lediglich älteren RFCs folgt, könnte gegen Angriffe verwundbar sein, denen durch Maßnahmen in neueren RFCs begegnet wird.

3.1.2 Denkbare Maßnahmen

Prinzipiell kommen zum Schutz der Neighbor Discovery die in den folgenden Abschnitten betrachteten Gegenmaßnahmen in Betracht.

3.1.2.1 IPsec

Als Bestandteil von ICMPv6 kann jede Nachricht der Neighbor Discovery prinzipiell mit IPsec gesichert werden.

Kritikpunkte

Die vorgesehene Nutzung von IPsec für die Neighbor Discovery leidet unter einem Henne/Ei-Problem. Das IPsec-Schlüsselmanagement erfolgt über das Protokoll IKE. Dies wird jedoch über UDP betrieben, d. h. für eine Nutzung müssen bereits IP-Adressen konfiguriert sein. Aus prinzipieller Sicht spräche zwar nichts gegen die Nutzung von IKE über ICMP, diese Idee wurde jedoch bisher noch nicht aufgegriffen. Die einzige Möglichkeit besteht dann in manuellem Schlüsselmanagement. Dies skaliert jedoch nicht und kommt aufgrund des Betriebsaufwands allenfalls für Umgebungen mit höchstem Schutzbedarf in

Frage. Stattdessen könnte man dann aber auch gleich die IP-Adressen sowie die Einträge für den Neighbor Cache manuell konfigurieren, siehe Abschnitt 3.1.2.3.

Eine Kompromisslösung könnte so aussehen, dass Neighbor Discovery link-local ungesichert abläuft und IKE anschließend am Link zum Einsatz kommt, so dass die Neighbor Discovery für alle größeren Scopes IPsec-gesichert abläuft.

3.1.2.2 SEND und CGA

Secure Neighbor Discovery (SEND) [RFC 3971] und Cryptographically Generated Addresses (CGA) [RFC 3972] stellen eine Alternative zu IPsec bei der Absicherung der Neighbor Discovery dar. Dabei werden die entsprechenden ICMP-Pakete um RSA-Signaturen und CGA-Optionen als zusätzliche neue NDP-Optionen in TLV-Codierung erweitert. Im Groben soll Folgendes erreicht werden.

Nachweis der Legitimität der Absenderadresse
Im Rahmen der Neighbor und Router Discovery müssen Nodes durch die hierfür eingeführte CGA-Option nachweisen, dass sie legitime Eigentümer der Absenderadresse sind. In diesem Zusammenhang werden Interface Identifier auf kryptografische Weise aus einem öffentlichen RSA-Schlüssel erzeugt. Die Überprüfung erfolgt mittels einer dazugehörigen Signatur, die in der entsprechenden CGA-Option hinterlegt wird.

Authentisierung von Neighbor Discovery Nachrichten
Die Secure Neighbor Discovery umfasst auch die Authentisierung Neighbor Discovery Nachrichten. Hierzu werden die öffentlichen Schlüssel verwendet, die in der CGA-Option hinterlegt sind, für die Signatur wird seinerseits eine Signatur-Option vorgesehen.

Verteilung von Zertifikatsketten
Die Signatur mit dem öffentlichen Schlüssel authentisiert nur, dass ein Node eine bestimmte IPv6-Adresse tatsächlich besitzt, für die Router Discovery oder Redirects ist dies jedoch noch nicht ausreichend. Hier sollte eine Authentisierung gegen einen (in geeignetem Sinn) allgemein bekannten Vertrauensanker stattfinden. SEND sieht hierzu die Verwendung von speziellen X.509-Zertifikaten vor. Um die Größe regulärer Router Discovery-Nachrichten so gering wie möglich zu halten, werden Zertifikatspfade mit Hilfe separater Certification Path Solicitations und Advertisements angefordert bzw. verteilt. Hierbei handelt es sich um für SEND entwickelte ICMPv6-Nachrichten, die zur Familie der Neighbor Discovery gehören.

Kritikpunkte

SEND ist bisher noch nicht weit verbreitet und erfordert zudem eine dafür nutzbare Public-Key Infrastruktur. SEND verhindert wirksam verschiedene Angriffe gegen die Neighbor Discovery, insbesondere verhindert SEND, dass ein Angreifer einen anderen Host am Link auf Grundlage der IP-Adresse imitiert. SEND verhindert jedoch nicht, dass Angreifer sich als legitime Hosts ausgeben – Zertifikatsbasierte Authentisierung ist nur für Router vorgesehen.

Darüber hinaus eröffnet SEND neue Angriffsvektoren gegen einzelne Nodes. Hier ist vor allem festzuhalten, dass bei SEND-Nachrichten stets die Signatur überprüft werden muss, was zu einem gewissen Rechenaufwand für die empfangenden Nodes führt. Eine Flut von SEND-Nachrichten könnte einen Node überlasten und somit zu einem Ausfall führen.

Ein weiterer Kritikpunkt liegt darin, dass die Verfahren RSA und SHA-1 in RFC 3971 fest vorgegeben sind.[20] Dies wurde bewusst so gewählt, um die aus Optionen und Auswahlmöglichkeiten folgende Komplexität zu vermeiden und Interoperabilität sicherzustellen. Jedoch wird der Wechsel eines der Verfahren einen neuen RFC erfordern und könnte dann schließlich doch zu Problemen der Interoperabilität führen.

Insgesamt machen SEND und CGA einen wenig durchdachten Eindruck. Zweifelhaft ist beispielsweise das vorgesehene Verteilungsmodell von zertifikatsbasierten Ankern: neben der lokalen Verteilung wird tatsächlich eine globale PKI (sic!) vorgeschlagen. Auch sind einige Details der Bildung der kryptografischen Adressen fragwürdig. Es bleibt abzuwarten, wie sich SEND und CGA weiterentwickeln, und ob diese Funktionen sich in gängigen Betriebssystemen etablieren.

3.1.2.3 Statische Einträge im Neighbor Cache und in Routingtabellen

Eine weitere Möglichkeit zur Vermeidung der genannten Angriffe bestünde in der statischen Konfiguration der IP- und Link-Layer-Adressen sowie der Routingtabellen aller Hosts am Link (vgl. [ISi-LANA, Punkt 7.2.7 A]).

Kritikpunkte

Sieht man einmal davon ab, dass diese Lösung ebenfalls nicht skaliert und daher vom Betriebsaufwand nur für Umgebungen mit höchsten Ansprüchen an die Sicherheit in Frage kommt, ist hierbei penibel darauf zu achten, dass die so konfigurierten Nodes tatsächlich nicht durch im Netz auftretende Neighbor Discovery Pakete beeinflusst werden können.

3.1.2.4 Überwachung und Alarmierung

Neben der reinen Prävention sind auch Maßnahmen der Überwachung denkbar. So kann ein Sensor an einem Link angeschlossen werden, der den Verkehr auf verdächtiges Verhalten wie beispielsweise NDP-Fluten und andere Anomalien untersucht. Beispiele für diesen Ansatz sind die Tools RAguard, NDPmon, RAfixd und RAmond. Diese Produkte haben ihren Fokus in der Mehrzahl auf gefälschten Router Advertisements. Die Reaktionsmöglichkeiten gehen von reiner Alarmierung bis hin zur Behebung. So verbreitet beispielsweise RAfixd auf erkannte illegitime Router Advertisements sofort Korrektur-Advertisments.

Kritikpunkte

Es wurde festgestellt, dass die genannten Produkte ihrerseits anfällig gegen Angriffe im Zusammenhang mit Fragmentierung und Header-Erweiterungen sind. Diese werden in den Abschnitten 3.3 und 3.4 diskutiert.

3.1.2.5 FCFS SAVI

Das First-Come-First-Served (FCFS) Source Address Validation Improvement (SAVI) [RFC 6620] ist eine Technik, die wie die Methoden des vorangegangenen Abschnittes auf dem beabsichtigten Zusammenhang zwischen Link-Layer und Network-Layer beruht. FCFS SAVI ist eine Maßnahme mit dem spezifischen Zweck, Adressfälschungen am Link zu

[20] Sowohl SHA-1 als auch die in RFC 3971 festgeschriebene Padding-Variatne für die RSA-Signatur gelten nach heutigem Stand der Technik nicht mehr als sicherste Verfahren ihrer Klasse und sollten unter diesem Gesichtspunkt eigentlich ersetzt werden.

erkennen und zu behandeln. FCFS SAVI ist somit komplementär zu Maßnahmen wie RAguard. Anders als RAguard ist FCFS SAVI nicht anfällig gegen Angriffe mittels Header-Erweiterungen.

FCFS SAVI arbeitet mit allen Methoden der Adresszuweisung zusammen, etwa SLAAC oder DHCPv6. Die Nutzung von FCFS SAVI erfordert, dass die Switche die dafür notwendige Funktionen aufweisen. Dies können beispielsweise alle Edge-Switche sein. Die Funktion von FCFS SAVI beruht auf einer Zuordnung (Bindung) von IPv6-Adressen zu Switch-Ports.[21] Auf den Switches wird dabei hinterlegt, welche IP-Adressen für die jeweiligen Ports gültig sind.

Die Hinterlegung erfolgt nach dem Prinzip, dass eine IP-Adresse, die zuerst an einem Port beobachtet wurde, an diesen Port gebunden wird (daher die Bezeichnung FCFS). Änderungen dieser Zuordnung sind nicht ohne Weiteres, sondern nur unter bestimmten Bedingungen möglich. Eine Fälschung der hintelegten Adressen an anderen Ports ist dann praktisch nicht mehr möglich. Durch Begrenzung der Anzahl der IP-Adressen, die auf einen Port gebunden werden können, werden Denial-of-Service-Angriffe gegen den Switch verhindert. Ebenso werden Denial-of-Service-Angriffe im großen Maßstab gegen andere Nodes unterbunden, welche die Duplicate Address Detection durchlaufen oder eine Adressauflösung betreiben.

Um eine gute Skalierbarkeit zu gewährleisten, werden die Bindungen nicht zentral hinterlegt. Stattdessen werden die Bindungen zu einem Port an dem entsprechenden Switch hinterlegt – jeder Switch „kümmert“ sich somit nur um die „eigenen“ Bindungen. Die SAVI-Switche bilden einen Perimeter, dessen Inneres als vertrauenswürdig gilt. Im Einzelnen wird an den SAVI-Switchen zwischen vertrauenswürdigen (trusted) und validierenden (validating) Ports unterschieden. Nur Verkehr, der an validierenden Ports eingeht, wird überprüft. Ports, an denen vertrauenswürdige Geräte wie Router und Server angeschlossen sind, werden dem Grundgedanken als vertrauenswürdig konfiguriert, während Ports, an denen beispielsweise Anwender-Endgeräte angeschlossen sind, als validierend konfiguriert werden.

Der Einsatz von nicht-SAVI-Switches ist in einer SAVI-Architektur möglich. Es muss jedoch penibel darauf geachtet werden, dass durch einen solchen Switch nicht die Perimetergrenze überbrückt wird.

Kritikpunkte

Die für FCFS SAVI notwendige Funktionalität muss auf wenigstens einem Switch implementiert sein. Aus Gründen der Skalierbarkeit und der Performanz sollten aber nach Möglichkeit alle Edge-Switche die entsprechenden Funktionen aufweisen. Derartige Switche sind derzeit noch vergleichsweise teuer; die Verbreitung von SAVI-Switchen dürfte auf absehbare Zeit daher eher gering bleiben.

Obwohl FCFS SAVI zu den wirksamsten Maßnahmen gegen Angriffe auf die Neighbor Discovery gehört, lässt sich ein großer Teil dieser Angriffe zuverlässig mit der als Nächstes vorgestellten Methode begegnen.

3.1.2.6 Kleine Netze und Schutz auf Link-Layer

Da Angriffe gegen die Neighbor Discovery nur link-local funktionieren, ist eine Aufteilung in kleine Subnetze ein sehr effektives Mittel, um derartige Angriffe einzudämmen. Im Extremfall könnte jedes Gerät einem eigenen Subnetz zugewiesen werden, obgleich dies allgemein

[21] Eine Zuordnung aufgrund von MAC-Adressen ist nicht vorgesehen, da MAC-Adressen gefälscht werden können.

weder notwendig noch ratsam ist. Eine angemessene Gruppierung der Systeme in Subnetzen nach Funktion, Vertrauenswürdigkeit und Kritizität wird in den meisten Fällen ausreichend sein.

In vielen Unternehmen und Organisationen ist es zudem zunehmend üblich, den Netzwerk-Zugang auf Link-Layer zu abzusichern, beispielsweise mit 802.1X. Über diese Maßnahme kann gesteuert werden, welche Systeme in welchem Subnetz untergebracht werden. Kommt dabei Link-Layer-Verschlüsselung zum Einsatz, dann kann diese Zuordnung bei korrekter Konfiguration und Implementierung praktisch nicht umgangen werden.

3.1.2.7 Schutz gegen Off-link-Angriffe

Zum Schutz gegen Off-link-Angriffe, wie im ersten Punkt aus Abschnitt 3.1 beschrieben, sollte ein Router Anfragen an nicht-exisitierende (routbare) Adressen für ein On-link-Präfix von vornherein verwerfen. Dies erfordert jedoch, dass in den Filterregeln des Routers die routbaren Adressen aller On-link-Nodes aufgeführt und ggf. regelmäßig aktualisiert werden. Hilfreich ist es zudem, wenn bestehende Einträge im Neighbor Cache des Routers in den Zuständen *Reachable* oder *Stale* (siehe Abschnitt 2.6.7) nicht von Anfragen nach „unbekannten“ Nodes verdrängt werden, die zunächst den Zustand *Incomplete* erhalten.

3.2 Weitere ICMP-Funktionen

Neben der Neighbor Discovery bietet ICMP noch weitere Angriffspunkte:

Path MTU Discovery
RFC 1981 beschreibt zwei prinzipielle Angriffe gegen die PMTU Discovery, nämlich die fälschliche Angabe einer zu kleinen MTU, die effektiv zu einer Drosselung des Verkehrs vom Absender zum Empfänger führt, sowie die fälschliche Angabe einer zu großen MTU, die zu Ausfällen von Paketen führt. RFC 1981 weist jedoch darauf hin, dass im Rahmen der Path MTU Discovery ein Node die Abschätzung der Path MTU aufgrund einer (gefälschten) Packet Too Big Fehlermeldung nicht erhöhen darf, so dass der zweite genannte Angriff eigentlich nicht erfolgreich sein sollte. Mit IPv6 darf die MTU 1280 Bytes nicht unterschreiten, so dass auch die Verringerung der Path MTU in den meisten Fällen keinen effektiven Angriff darstellt.

3.3 Header-Erweiterungen

3.3.1 Bedrohungen

Im Wesentlichen sind im Zusammenhang mit Header-Erweiterungen die folgenden Angriffsvektoren denkbar:

- Header-Erweiterungen können dazu verwendet werden, um IP-Pakete künstlich aufzublähen. Dies kann vor allem in Kombination mit Fragmentierung dazu verwendet werden, um filterrelevante Informationen aus dem ersten Fragment in ein nachfolgendes Fragment zu verschieben. In diesem Fall könnte möglicherweise eine Fehlentscheidung getroffen werden, weil die Information im ersten Fragment nicht enthalten ist, oder Filtermechanismus wird komplexer, weil das Paket vor einer Entscheidung zunächst rekonstruiert werden muss.
- Unbekannte Header-Erweiterungen oder unbekannte Optionen (im Fall von Hop-by-Hop Options oder Destination Options) könnten dazu verwendet werden, um Schwachstellen in der Implementierung auszunutzen. Konkret könnte ein Node zum Absturz oder in einen

undefinierten Zustand gebracht werden; im Fall von Pufferüberläufen ist auch die Übernahme des betroffenen Nodes denkbar.

- Dasselbe gilt für die unzulässige Verwendung oder fehlerhafte Codierung bekannter Header-Erweiterungen, beispielsweise die mehrfache Verwendung von Headern oder Header-Optionen.

3.3.1.1 Unbekannte Header-Erweiterungen oder Optionen

Unbekannte Header-Erweiterungen oder unbekannte Optionen können dazu verwendet werden, um eine (ggf. auch standardkonforme) Reaktion des empfangenden Nodes zu provozieren. Ein gutes Beispiel hierfür ist das Tool alive6 der THC-IPv6-Suite. Ein Node, der nicht auf Echo-Request Nachrichten (Ping) an die Link-Local All-Nodes Multicast-Adresse reagiert, wird seine IP-Adresse auf diese Weise nicht preisgeben. Dies kann jedoch unterlaufen werden, da im Fall eines unbekannten Headers oder einer unbekannten Option eine ICMPv6-Fehlermeldung versendet wird. Zwar wird in RFC 4443 der Versand von Fehlermeldungen untersagt, wenn das ursächliche IP-Paket eine Multicast-Quelladdresse hatte, der Fall von undefinierten Header-Parametern wird jedoch ausdrücklich davon ausgenommen.

3.3.1.2 Header-Erweiterungen und Fragmentierung

Die Header-Erweiterungen Hob-by-Hop Options und Destination Options haben eine variable Länge und können prinzipiell bis zu 2048 Bytes lang werden. Wenn allein für die Header-Erweiterung(en) schon die MTU überschritten wird, dann führt dies zu einer Fragmentierung, durch die der Upper-Layer Header der Nutzlast nicht mehr im ersten Fragment sichtbar ist, d. h. eine Prüfung gegen übliche Firewall-Filterregeln erfordert ggf. eine Rekonstruktion des ursprünglichen IP-Pakets. Dies ist jedoch aufwändig und widerspricht dem Grundsatz, dass die Rekonstruktion aus Fragmenten erst am Ziel-Node stattzufinden hat. Siehe Abschnitt 3.4 für weitere Details.

3.3.2 Denkbare Maßnahmen

Betrachtet man die existierenden Header-Erweiterungen, dann kann man zu dem Schluss kommen, dass in „gewöhnlichen“ Nutzungsszenarien fast alle Header-Erweiterungen verworfen werden können:

- Hop-by-Hop Options Header spielen derzeit nur für Router Alerts und Jumbogramme eine Rolle. Router Alerts werden derzeit nur für RSVP und Multicast Listener Discovery eingesetzt und dürften mit Ausnahme der Neighbor Discovery in den meisten betrieblichen Nutzungsszenarien irrelevant sein. Jumbogramme finden nur auf speziellen Netzen mit einer MTU über 65535 eine sinnvolle Anwendung.
- Routing Header werden nur für Mobile IPv6 (Typ 2) und für RLP (Typ 3) eingesetzt und spielen im gewöhnlichen Betrieb keine Rolle.
- Destination Options Header kommen derzeit nur bei Mobile IPv6 oder bei Tunnel-verschachtelungen zum Tragen und spielen im gewöhnlichen Betrieb ebenfalls keine Rolle.
- Der Authentication Header sowie die Encapsulated Security Payload hat nur im Zusammenhang mit IPsec eine Bedeutung; an den Übergängen von und zu Netzbereichen, zu denen IPsec-Verkehr nicht vorgesehen ist, können Pakete mit diesen Header-Erweiterungen aussortiert werden.

- Der Mobility Header hat nur für Mobile IPv6 eine Bedeutung und spielt in gewöhnlichen Szenarien keine Rolle.

Als einzig durchgängig legitime, im üblichen Betreib zu erwartende Header-Erweiterungen verbleiben der Hop-by-Hop Options Header für Link-Local Multicast Listener Discovery sowie der Fragmentation Header, wobei auch hier noch Einschränkungen möglich sind.

Darüber hinaus gibt es bei vielen ICMPv6-Nachrichtentypen derzeit gar kein legitimes Einsatzszenario für Header-Erweiterungen einschließlich Fragmentierung, hier ist vor allem die Neighbor Discovery genannt. Seit Neuestem ist tatsächlich vorgesehen, dass fragmentierte Nachrichten der Neighbor Discovery verworfen werden [RFC 6980]. Darüber hinaus dürfen Header-Erweiterungen nicht so genutzt werden, dass der Upper-Layer Header in ein nachfolgendes Fragment verschoben wird – im Fall der Fragmentierung muss der Upper-Layer Header noch vollständig im ersten Fragment liegen [RFC 7112].

3.4 Fragmentierung

Bei der Fragmentierung von IPv6-Paketen spielen teilweise dieselben Probleme eine Rolle, die auch schon bei IPv4 aufgetreten sind. Unglücklicherweise hat RFC 2460 bei der Definition von IPv6 überlappende Fragmente zugelassen, obwohl dies von IPv4 her bereits als problematisch bekannt war. Überlappende Fragmente können beispielsweise dafür verwendet werden, Filterentscheidungen an Firewalls zu unterlaufen. Hier wurde erst mit RFC 5722 Abhilfe geschaffen – überlappende Fragmente sind nicht mehr zulässig und Pakete mit überlappenden Fragmenten müssen stillschweigend verworfen werden.

3.4.1 Bedrohungen

Die bereits von IPv4 bekannten Angriffsvektoren sind:

- Denial of Service gegen den Ziel-Node: Für die Wiederherstellung eines Pakets aus Fragmenten muss der Ziel-Node bis zur vollständigen Wiederherstellung oder bis zu einem Time Out ausreichend Speicher für das Paket vorhalten. Die Gesamtdauer der Vorhaltung kann durch den Versand von Kleinstfragmenten erheblich in die Länge gezogen werden. Durch massenhaften Versand von Paketfragmenten kann der Ziel-Node dazu gezwungen werden, seinen gesamten hierfür vorgesehenen Speicher zu blockieren.
- Ein Paket, das einen Fragment-Header enthält, tatsächlich aber nicht fragmentiert ist, wird als *atomares Fragment* bezeichnet. Die Behandlung atomarer Fragmente ist nicht eindeutig geregelt und könnte je nach Implementierung zu Schwierigkeiten führen [RFC 6946]. Der Kern des Problems besteht darin, dass eine Implementierung anfällig für einen Denial-of-Service-Angriff ist, wenn atomare Fragmente genauso behandelt werden, wie „gewöhnliche“ Fragmente. Wenn die verwendeten Fragment-IDs nicht unvorhersagbar sind, dann wäre es konkret möglich, atomare Fragmente dazu einzusetzen, um Überlappungen mit (legitimen) Fragmenten eines anderen Pakets zu provozieren und somit ein stillschweigendes Verwerfen des legitimen fragmentierten Pakets herbeizuführen.

Spezifisch für IPv6 sind die folgenden Aspekte:

- Ein großes Sicherheitsproblem stellen überlappende Fragmente dar, da hiermit Filterrichtlinien umgangen werden können. Nach RFC 2460 sind überlappende Fragmente bei IPv6 zulässig; erst mit RFC 5722 wurden überlappende Fragmente ausnahmslos für unzulässig erklärt und sollten verworfen werden. Hier könnten sich Systeme mit einer älteren Implementierung jedoch trotzdem als anfällig erweisen.

- Darüber hinaus könnte Fragmentierung – trotz des Verbots überlappender Fragmente – in Kombination mit Header-Erweiterungen verwendet werden, um Filterentscheidungen an Firewalls zu umgehen oder zumindest aufwändig zu gestalten. Bei IPv6 leisten hierzu die Erweiterungs-Header für die Hop-by-Hop Options und Destination Options Vorschub. Diese beiden Erweiterungs-Header haben eine variable Länge und lassen sich leicht so aufblähen, dass beispielsweise der Upper-Layer Header in das zweite, dritte oder ein noch weiter nachfolgendes Fragment verschoben wird. Die vollständige Inspektion eines solchen Pakets erfordert die Rekonstruktion aus allen Fragmenten. Diese Rekonstruktion bindet Ressourcen, so dass ein Angreifer dies für einen Denial-of-Service-Angriff nutzen könnte.

  Fernando Gont hat Mechanismen zur Überwachung der Neighbor Discovery wie u. a. RAguard (siehe Abschnitt 3.1.2.4) untersucht und kam zu der Feststellung, dass alle derartigen Mechanismen zum Zeitpunkt der Untersuchung durch die kombinierte Verwendung von Fragmentierung und extensiven Header-Erweiterungen umgangen werden können [Gont 2011b].[22]

  Um den Missbrauch von Fragmentierung einzuschränken, wurde mit RFC 6980 die Verwendung von fragmentierten NDP-Nachrichten untersagt, und zusätzlich wurde in RFC 7112 festgelegt, dass im Fall der Fragmentierung der Upper-Layer Header vollständig im ersten Fragment liegen muss. Es ist davon auszugehen, dass diese Vorschriften erst nach und nach implementiert werden wird; in der Zwischenzeit sollten die gegannten Anforderungen mit dazu geeigneten Firewallregeln durchgesetzt werden.

3.4.2 Denkbare Maßnahmen

IPv6 schreibt vor, dass Pakete nur vom Absender fragmentiert werden dürfen – mehrfache Fragmentierung kann dadurch ausgeschlossen werden, d. h. Pakete mit mehr als einem Fragment-Header können verworfen werden.

Header		Bytes	Wörter
IPv6-Header		40	5
Hop-by-Hop Options	Router Alert	4	1
Destination Options	Tunnel Encapsulation Limit	3	1
	Home Address	18	3
Routing	Mobility	24	3
Fragmentation		8	1
Authentication		≤ 76	≤ 10
Mobility		≤ 64	≤ 8
Upper-Layer Header	TCP	≤ 60	< 8
	UDP	8	1
	SCTP (nur Common Header)	12	2
	ICMP	4	1

Tabelle 1: Maximal sinnvoller Platzbedarf verschiedener Header und Header-Erweiterungen

22 Siehe auch *IPv6 NIDS evasion and improvements in IPv6 fragmentation/reassembly* sowie *RA guard evasion technique*

Grundsätzlich könnten an einer Firewall zwei Strategien zum Umgang mit Fragmenten zur Anwendung kommen: Entweder Rekonstruktion an der Firewall mit anschließender Inspektion des gesamten Pakets oder Inspektion nur des ersten Fragments. Casimir Potyraj nennt als jeweilige Vor- und Nachteile einer Rekonstruktion an der Firewall u. a. [Potyraj 2007]:

Vorteile

- Deep Packet Inspection möglich
- Filterentscheidung wird anhand des vollständigen Pakets ermittelt
- Interne Nodes sind vor etwaigen Denial-of-Service-Angriffen durch unvollständige fragmentierte Pakete geschützt

Nachteile

- Mögliche Denial-of-Service-Angriffe richten sich direkt und konzentriert gegen die Firewall selbst
- Gesteigerte Performance-Erwartungen an die Firewall
- Ggf. müssen inspizierte Pakete zur Weiterleitung erneut fragmentiert werden

Darüber hinaus werden verschiedene Empfehlungen angegeben, die vor allem Fragmentierung in Kombination mit Tunnelung betreffen.

Falls die Firewall nicht gleichzeitig Deep Packet Inspection betreiben soll, dann ist die Filterung ohne Rekonstruktion vorzuziehen, da

- ansonsten eine rekonstruierende Firewall selbst Ziel von Denial-of-Service-Angriffen werden kann,
- die meisten Header-Erweiterungen im gewöhnlichen Betrieb keine Rolle spielen und a priori ausgefiltert werden können,
- selbst „exotische“ Kombinationen von Header-Erweiterungen nebst Upper-Layer Header mühelos innerhalb der minimal garantierten MTU untergebracht werden können (siehe die Tabelle oben), sofern kein unnötiger Gebrauch von Padding gemacht wird.

Daher ist davon auszugehen, dass alle für die Filterentscheidung relevanten Informationen im ersten Fragment untergebracht sein können – Pakete, bei denen dies nicht der Fall ist, können verworfen werden.

3.5 Privatsphäre

Die Rückkehr des Ende-zu-Ende-Prinzips und die Beseitigung von NAT bei IPv6 haben zu einigen Bedenken hinsichtlich der Privatsphäre der Nutzer geführt. Gleichzeitig wurden jedoch mit IPv6 die sogenannten Privacy Extensions (PEX) eingeführt. In diesem Abschnitt wird für drei unterschiedliche Szenarien die Tauglichkeit von NAT im Vergleich mit Privacy Extensions zur Wahrung der Privatsphäre diskutiert, vgl. auch [BVA 2013, Abschnitt 8.5]. Dazu ist zunächst zu definieren, was eigentlich Wahrung der Privatsphäre bedeutet.

- Das erste Ziel kann darin bestehen, als Host einer End-Site nicht ausfindig gemacht werden zu können, d. h. ein Dienstanbieter (beispielsweise eine Nachrichtenseite) soll die Nutzungen des Dienstes zu verschiedenen Zeiten von einer bestimmten Site aus nicht miteinander korrelieren können. Mit anderen Worten, es soll nicht möglich sein, verschiedene Kommunikationsvorgänge einer Site zuzuordnen.
- Das zweite Ziel kann darin bestehen, als Host in einer Gruppe anderer Hosts innerhalb einer Site nicht aufgespürt werden zu können, d. h. ein Dienstanbieter mag zwar die Zugehörigkeit eines Nutzers zu einer Site nachverfolgen, aber nicht, welcher Host innerhalb der Site den betreffenden Dienst verwendet. Mit anderen Worten, es soll nicht möglich sein, zwischen den Hosts einer Site zu differenzieren.

- Das dritte Ziel kann darin bestehen, als einzelner *mobiler* roaming Host nicht verfolgt werden zu können.

Für die Diskussion der nachfolgend diskutierten Szenarien sei zunächst auf einige Aspekte der Adressvergabe bei IPv6 im Vergleich zu IPv4 hingewiesen:

- End-Sites bekommen üblicherweise ein festes IPv6-Präfix vom ISP zugewiesen, beispielsweise ein /48- oder ein /56-Präfix. Es ist davon auszugehen, dass allgemein bekannt sein wird, welche ISPs Präfixe aus welchem Adressbereich und welcher Größe an End-Sites vergeben. *Somit kann davon ausgegangen werden, dass das Präfix die End-Site eines Nutzers identifiziert*. Daran würde auch NAT nichts ändern.
- Selbst bei einer Vergabe von dynamischen IPv6-Präfixen an eine Site durch einen ISP wird eine dauerhafte Korrelation von Zugriffen allein auf Grundlage der IP-Adresse nur bedingt verhindert – bei der Verwendung von SLAAC zur Vergabe von IPv6-Adressen kann ein Host über den quasi-eindeutigen Interface-Identifier verfolgt werden. Dasselbe gilt, wenn der Interface-Identifier auf andere Weise statisch vergeben wird.
- Durch die Vergabe von dynamischen IPv4-Adressen an einen Node durch einen ISP wird dagegen eine dauerhafte Korrelation von Zugriffen allein auf Grundlage der IP-Adresse im Allgemeinen verhindert.

3.5.1 Szenario 1 – Heimnutzer

In diesem Szenario betrachten wir einen Nutzer, beispielsweise einen Heimnutzer, der von seinem ISP ein /56-Präfix zugewiesen bekommt.

Wird dem Nutzer über einen längeren Zeitraum dasselbe Präfix zugewiesen, dann kann die Site des Nutzers innerhalb dieses Zeitraums verfolgt werden. Die Verwendung von Privacy Extensions verschleiert zwar den einzelnen Host, aber nicht die Site. Selbst eine Verwendung von wechselnden Subnet-Ids bringt keinen Gewinn an Privatsphäre, da die Site nach wie vor am Präfix zu erkennen ist. Der Einsatz von NAT hilft an dieser Stelle überhaupt nicht weiter.

Werden Präfix (vom ISP) und Subnet-ID (vom Nutzer) dynamisch vergeben, reicht das allein jedoch auch nicht aus. Wenn die Interface-Ids statisch vergeben werden, etwa über SLAAC oder pseudozufällig über DHCPv6, dann verraten die Interface-Ids die Hosts innerhalb der Site und damit auch die Site an sich.

Um eine dauerhafte Korrelation von Zugriffen allein auf Basis der IP-Adresse zu verhindern, müssen sowohl das Präfix als auch die Interface-ID und möglichst auch die Subnet-ID dynamisch vergeben werden.

3.5.2 Szenario 2 – Nutzer einer Site

In diesem Szenario betrachten wir eine Site mit statischem Präfix und einer signifikanten Anzahl von Nutzern, zwischen denen von außen eine Differenzierung nicht möglich sein soll.

In diesem Szenario helfen Privacy Extensions genauso wie NAT oder die Verwendung eines Proxys, das Aufspüren eines bestimmten Hosts zu vermeiden. Bei Verwendung der Privacy Extensions bleibt das Ende-zu-Ende-Prinzip gewahrt. Im Fall von NAT würde das Ende-zu-Ende-Prinzip wieder aufgegeben. Bei Verwendung eines Proxys bliebe das Ende-zu-Ende-Prinzip erhalten und dennoch würden ausgehende Verbindungen von einer einzigen IP-Adresse kommend erscheinen. Da man auf dem Proxy auch gleich Content-Filterung vornehmen kann, ist die Verwendung von Proxys gegenüber NAT oder Privacy Extensions im Unternehmensumfeld in der Regel die bessere Wahl.

3.5.3 Szenario 3 – Mobile roaming Hosts

Dieses Szenario hat Ähnlichkeit zu Szenario 1 mit dynamischer Vergabe des Präfixes. Hierbei wird ein einzelner mobiler Host betrachtet, der sich von Netz zu Netz bewegt (Roaming).

Hosts, deren Interfaces über SLAAC konfiguriert werden, bekommen stets dieselbe Interface-ID, nämlich den EUI-64-Identifier, der aus der MAC-Adresse gebildet wird. Ein roaming Host kann über die Interface-ID über die Netzgrenzen hinweg verfolgt werden. Neben den „üblichen" Bedenken in Bezug auf die Wahrung der Privatsphäre kommt als potenzielle Bedrohung hinzu, dass hierüber ein räumliches Bewegungsprofil des Nutzers erstellt werden kann. Die Privacy Extensions wurden dafür entworfen, um in genau diesem Szenario Abhilfe zu schaffen, indem regelmäßig eine zufällige temporäre Interface ID gewählt wird.

An dieser Stelle würde auch NAT ggf. helfen, wenn alle Betreiber der besuchten Netze dies konsequent durchführen würden. Hierbei würde jedoch die Verantwortung zur Wahrung der Privatsphäre in die Hände der einzelnen Betreiber gelegt. Daher ist es vorzuziehen, durch Verwendung von Privacy Extensions eigenverantwortlich die eigene Privatsphäre durchzusetzen und Privacy Extensions zu nutzen.

3.5.4 Fazit zur Privatsphäre

Die Privatsphäre wird durch IPv6 nicht mehr oder weniger bedroht, als durch IPv4. Mechanismen wie Cone-NAT oder Proxys können hier in einigen Szenarien zur Wahrung der Privatsphäre beitragen, ebenso die dynamische Vergabe von Präfixen. Jedoch sollte man dabei Folgendes bedenken: Die IP-Adresse ist nur eine Möglichkeit, Nutzer zu verfolgen. Auf Anwendungsebene hinterlassen vor allem Browser weitere Spuren, die zur Verfolgung von Nutzern geeignet sind. Wer gezielt Anonymität im Netz sucht, sollte unabhängig vom genutzten Netzwerkprotokoll einen entsprechenden Dienst in Anspruch nehmen.

3.6 IP-Adressen und deren Schreibweise

Die Komplexität von IPv6-Adressen ergibt sich nicht nur aus deren Länge, sondern auch aus den verschiedenen Schreibweisen, mit eine IPv6-Adresse dargestellt werden kann. Dies erschwert sowohl die Nutzung durch Anwender als auch die automatisierte Verarbeitung.

3.6.1 Bedrohungen

Die Komplexität von IPv6-Adressen kann sich an den folgenen Stellen bemerkbar machen:

- Eine typische IPv6-Adresse sieht aus wie im nachstehenden Beispiel:

  `2001:db8:2e5a:7d04:20c:29ff:fee8:1c84`

  Während eine typische IPv4-Adresse noch einigermaßen leicht zu merken ist, stellt eine typische IPv6-Adresse eine noch größere Herausforderung an die Merkfähigkeit des Anwenders dar als etwa ein gutes Passwort. Die Angewohnheit, IP-Adressen anstelle von Domain-Namen zu verwenden, ist mit IPv6 daher deutlich fehleranfälliger als bei IPv4. Darüber hinaus trennt ein Punkt optisch stärker als ein Doppelpunkt eine Folge von Zeichen und Ziffern. Angreifer könnten dies u. U. ausnutzen und Verwirrung stiften, etwa durch Verwendung lexikalisch ähnlicher Adressen, beispielsweise `2001:db8::12:345` anstelle von `2001:db8::123:45` .

- Auch die Schreibweise von IPv6-Adressen mit Port-Nummer ist ggf. problematisch, beispielsweise in URIs. Leider wird nach RFC 3986 der Doppelpunkt bereits zur

Abtrennung von IP-Adressen und Portnummern in URIs verwendet. Ist die Angabe einer Portnummer optional, dann ist `2001:db8::cafe:affe:80` nicht eindeutig. RFC 5952 empfiehlt die aus RFC 3986 bereits bekannte Schreibweise mit eckigen Klammern, etwa `[2001:db8::cafe:affe]:80` . Eine ähnliche Syntax wird für die Secure Shell (SSH) und verschiedene Kommandozeilen-Tools verwendet. In typischen Kommandozeilen-Shells haben eckige Klammern jedoch bereits eine syntaktische Bedeutung, so dass zusätzlich Quotierung notwendig ist. All dies verbessert weder die Lesbarkeit, noch macht es die automatisierte Verarbeitung einfacher. Auch dies ist ein Grund, verstärkt Domain-Namen anstelle von IP-Adressen einzusetzen.

- Bei der Schreibweise einer Adresse können unterschiedliche, aber äquivalente – oder zumindest von gängigen Implementierungen gleich behandelte – Formen auftreten:
  - Führende Nullen können weggelassen werden. Denkbar ist es auch, führende Nullen *hinzuzufügen*, die Adressen `2001:db8::1:1001` und `2001:db8::1:01001` unterscheiden sich nur lexikalisch. Obwohl die zweite Schreibweise nicht RFC-konform ist, ist die Bedeutung eigentlich klar – allerdings wäre hier auch ein Schreibfehler plausibel, beispielsweise `2001:db8::1:0100:1` .
  - Die Kompression von aufeinanderfolgenden Null-Blöcken zu `::` kann ggf. unterschiedlich gehandhabt werden.
  - Die hexadezimalen Ziffern „a“, „b“, „c“, „d“, „e“ und „f“ können in Klein- oder Großschreibung auftreten.
  - Eingebettete IPv4-Adressen können in verschiedenen Schreibweisen auftreten: `::ffff:192.168.0.1` ist identisch zu `::ffff:c0a8:1` .

  Diese Vielfalt kann Probleme bei der Verarbeitung oder beim manuellen Abgleich bereiten. Es können sich beispielsweise Probleme im Zusammenhang mit digitalen Signaturen und Zertifikaten ergeben. Auch bei der Auswertung von Protokolldaten können unterschiedliche Schreibweisen zu Schwierigkeiten bei der Korrelation von Ereignissen führen. Daher ist es für die Verarbeitung wichtig, Adressen zuvor zu *kanonisieren*. Die textuelle Darstellung von IPv6-Adressen sollte einheitlich erfolgen.
- Johannes Endres hat eine Untersuchung gängiger Implementierungen in verschiedenen Sprachen durchgeführt [Endres 2012]. Das Ergebnis war, dass zum Zeitpunkt der Untersuchung keine Implementierung zuverlässig gültige von ungültigen IPv6-Adressen unterscheiden konnte, wobei jedoch deutliche Unterschiede im Grad der Abweichungen existieren.
- Wenig durchdachte User Interfaces stellen einen gewöhnlichen Benutzer vor große intellektuelle Herausforderungen. So ist beispielsweise die gutmeinende Vorgehensweise, acht einzelne Textfelder für die Eingabe einer IPv6-Adresse vorzusehen, kontraproduktiv, insbesondere, wenn der Nutzer `::` in Adressen wie beispielsweise `2001:db8::cafe:affe` zunächst gedanklich selbst expandieren muss [Endres 2012].

3.6.2 Denkbare Maßnahmen

Zur eindeutigen Darstellung enthält RFC 5952 die folgenden Empfehlungen:

- Führende Nullen eines Blocks sollten grundsätzlich vollständig eliminiert werden (bis auf 0000, was zu 0 wird).
- Die Kompression `::` sollte grundsätzlich für zwei oder mehr aufeinanderfolgende Null-Blöcke angewendet werden, jedoch nicht für einen einzelnen Null-Block; `2001:db8:0:cafe:5:6:7:8` sollte also nicht zu `2001:db8::cafe:5:6:7:8` verkürzt werden.

- Die Kompression `::` sollte bestmöglich verwendet werden; die Adresse `2001:db8:0:0:0:0:cafe:affe` sollte somit als `2001:db8::cafe:affe` und nicht etwa als `2001:db8:0:0::cafe:affe` oder `2001:db8::0:0:cafe:affe` dargestellt werden.
- Bei mehr als einer Möglichkeit für die Verwendung von `::` sollte die Möglichkeit gewählt werden, die zu der kürzesten Schreibweise führt, bei „Gleichstand“ sollte der erste Folge von Null-Blöcken komprimiert werden.
- Für die Darstellung der hexadezimalen Ziffern „a“ bis „f“ sollten grundsätzlich Kleinbuchstaben verwendet werden.

Bei der Entwicklung von Anwendungen gilt

- Es sollte auf eine verlässliche Validierung von IPv6-Adressen geachtet werden.
- Bei der Gestaltung von User Interfaces sollte darauf geachtet werden, dass mit Copy & Paste gearbeitet werden kann.

Alle genannten Punkte sprechen für einen verstärkten Einsatz von DNS-Namen anstelle von IPv6-Adressen sowie für die Verwendung eines Tool-gestützten Adressmanagement.

4 Weitere Sicherheitsaspekte

In diesem Abschnitt werden sicherheitsrelevante Aspekte diskutiert, die durch IPv6 in einem anderen Licht erscheinen, aber keine Bedrohung gegen IPv6 darstellen.

4.1 Angriffserkennung und Angriffsbehandlung

4.1.1 Über IPv4 und IPv6 verteilte Angriffe

Für die Erkennung von Angriffen durch Signatur- oder Verhaltensanalyse und für die Analyse des Umfangs und Schadens eines erfolgreichen Angriffs werden maßgeblich die Ereignisprotokolle eines oder mehrerer Systeme herangezogen. Wenn eine Site über IPv4 und IPv6 erreichbar ist, dann müssen zur Angriffserkennung und -analyse die kombinierten Ereignisprotokolle für IPv4 und IPv6 herangezogen werden.

Eine weitere Schwierigkeit ergibt sich aus den verschiedenen Schreibweisen, mit denen eine IPv6-Adresse dargestellt werden kann, siehe Abschnitt 3.6. Wenn verschiedene Anwendungen IPv6-Adressen auf unterschiedliche Arten protokollieren, dann ist ohne Kanonisierung der Darstellung keine Korrelation verschiedener Ereignisse möglich. In der Folge könnten Angriffsmuster übersehen werden.

4.1.2 Sperrung von IP-Adressen

Ein typisches Reaktionsschema auf Brute-Force-Angriffe oder Portscans und ähnliche „Belästigungen“ besteht darin, die IP-Adresse der Quelle temporär zu sperren oder einer Verbindungslimitierung zu unterwerfen. Mit IPv6 wird dieses Schema jedoch obsolet, da sich ein Angreifer mühelos eine neue Adresse innerhalb seines Netzes oder seiner Site beziehen kann. Jeder Angriff bekommt somit potenziell Züge von Distributed Brute-Force-Angriffen.

Die Sperrung von ganzen Präfixen mag hier zwar naheliegen, jedoch sollte bedacht werden, dass ein Angreifer aus einem öffentlichen Netz heraus agieren könnte, etwa aus dem Netz eines Mobilfunkbetreibers. Eine automatisierte Sperrung des Präfixes führt in diesem Fall zum Ausschluss zahlreicher legitimer Nutzer und letzten Endes potenziell zu einem Denial of Service gegen sich selbst. *Die Sperrung von IPv6-Adressen oder Präfixen als Reaktion auf*

Angriffe ist bei IPv6 nicht sinnvoll. Zur Abwehr solcher Angriffe sind hier andere, nicht auf IP-Adressen beruhende Mechanismen notwendig.

4.2 Netzwerkabtastung

Die IPv6-Adressarchitektur hat eine aus Sicherheitssicht eine interessante Wirkung auf die Möglichkeit einer Netzwerkabtastung, beispielsweise mit einem Portscanner wie nmap. In IPv4-Netzen können mit Hilfe eines Portscanners Netze auf aktive Hosts überprüft werden, indem der Netzbereich aufgezählt wird und an jede zum Netz gehörige Adresse ein oder mehrere Testpakete geschickt werden. Dies ist selbst für die größten IPv4-Netze noch praktikabel. Da ein IPv6-Netzwerk dagegen jedoch grundsätzlich mindestens 2^{64} Adressen umfasst, verbietet sich diese Vorgehensweise für IPv6-Netze schon auf den ersten Blick. Eine detaillierte Betrachtung hierzu wurde in RFC 5157 vorgenommen, die durch Fernando Gont verfeinert wurden [Gont 2011a, Gont 2012b].

Unter Umständen ist jedoch die Aufzählung aller *wahrscheinlich* in Frage kommenden IPv6-Adressen bzw. Interface-Ids praktikabel. Im Internet6 wurden bereits die folgenen Bildungsmuster für die Interface-ID beobachtet:

- Eingebettete IPv4-Adressen, wie beispielsweise `2001:db8::ac10:1846` für `172.16.24.70`
- Portnummer eines charakteristischen Dienstes, wie beispielsweise `2001:db8::80` für einen Webserver oder `2001:db8::53` für einen DNS-Server
- Andere einfache Muster, wie beispielsweise sequenzielle Vergabe wie für IPv4 typisch.
- Mnemonische Ids, wie beispielsweise in `2001:db8::cafe`, `2001:db8::affe`, `2001:db8::f00d` usw.[23]
- EUI-64-Identifier
- Pseudozufällige Interface-Ids (mit Privacy Extensions, über DHCPv6 oder von Hand)

Wird der letzte Fall konsequent angewendet, dann ist die Aufzählung der Systeme in den entsprechenden Netzen jenseits jeder Praktikabilität. Im Fall von EUI-64 Interface-Ids ist die Aufzählung aufwändig, aber nicht unmöglich, da diese Interface-Ids typischerweise aus 48-Bit-MAC-Adressen gebildet werden. Diese werden nicht vollkommen zufällig vergeben, sondern die oberen 24 Bits kennzeichnen u. a. den Hersteller der Netzwerkschnittstelle und sind relativ vorhersagbar.[24] Es verbleiben die unteren 24 Bits, deren Aufzählung so aufwändig ist, wie die eines Class-A-Netzes unter IPv4. Somit entspricht der Aufwand einer Abtastung dem Scan einiger Class-A-Netze – aus praktischer Sicht ist dies prinzipiell noch machbar. Alle anderen beobachteten Typen von Interface-ID lassen sich mehr oder weniger leicht aufzählen.

David Malone hat zur Verteilung verschieden konstruierter Adressen im Jahr 2008 eine Studie veröffentlicht [Malone 2008]. Hierfür wurden die über einen Zeitraum von vier Jahren beobachteten Adressen untersucht. Die Studie zeigt, dass die sogenannten „wordy“ Adressen mit mnemonischen Begriffen seinerzeit nicht verbreitet waren. Dagegen bildeten SLAAC-Adressen, die aus dem EUI-64-Identifier erzeugt wurden, mit etwa 40% den Großteil

---

23 Der Leser mag raten, welcher – reale – Host sich hinter `2a03:2880:2110:cf01:face:b00c:0:9` verbirgt.

24 Mit zunehmender Virtualisierung von Server-Systemen wird die Vorhersage besonders einfach. Aber auch bei realer Hardware werden in vielen Unternehmen für die Mehrzahl der Rechner nur wenige Baureihen von einigen bekannten Herstellern eingesetzt.

der beobachteten Adressen. Mit der zunehmenden Verbreitung von Windows 7 dürfte dieser Anteil gesunken sein, da unter Windows 7 standardmäßig nicht der EUI-64-Identifier verwendet wird. Die seinerzeit am zweithäufigsten beobachtete Form von Adressen war mit etwa 25% aus IPv4-Adressen gebildet, davon überwiegend 6to4-Adressen, gefolgt von etwa 15% der IPv6-Adressen, bei denen alle bis auf das niederwertigste Byte Null waren.

Ähnliche Beobachtungen wurden in jüngeren Untersuchungen durch Marc Heuse und Fernando Gont gemacht – etwa zwei Drittel der im Internet6 beobachtbaren Server-Adressen hat eine Interface-ID, die einem der ersten drei oben aufgeführten Typen entspricht [Heuse 2013]. Eine nähere Untersuchung hat darüber hinaus gezeigt, dass etwa drei Viertel der untersuchten Netze ein System mit den Interface Ids …`::1` oder …`::2` enthalten. Auf diesem Weg kann beispielsweise schnell getestet werden, ob ein Netz überhaupt für eine nähere Untersuchung interessant sein könnte.

Da DNS mit IPv6 deutlich systematischer eingesetzt werden dürfte, ist zu erwarten, dass über geschickte DNS-Anfragen auch Systeme mit pseudozufälligen Interface-Ids aufgefunden werden können. Das bedeutet, dass mit Verschleierung der IPv6-Adresse zwar die Existenz eines Systems vor der Allgemeinheit verborgen werden kann, dies ersetzt jedoch nicht eine angemessene Härtung der im Internet erreichbaren Systeme.

Soll eine Netzwerkabtastung „on-link“ erfolgen, dann ergeben sich noch weitere Möglichkeiten. Dies ist insofern von Bedeutung, als dass nicht nur ein Angreifer ein Interesse an der Aufzählung aller potenziellen Angriffsziele hat, sondern auch ein Administrator hat ein (legitimes) Interesse daran zu wissen, welche Systeme in den Netzen betrieben werden, die unter seiner Verantwortung stehen.

Bei direktem Zugriff auf den Link werden beispielsweise mit einem Ping an die Link-Local All-Nodes Multicast-Adresse (`ff02::1`) in vielen Fällen alle am Link angeschlossenen Systeme gefunden. Nun muss ein Node zwar Nachrichten an `ff02::1` verarbeiten, er muss aber nicht auf ein derartiges Ping antworten; über eine einfache Filterregel könnten Echo Requests an `ff02::1` am Node verworfen werden. Die meisten IPv6-Implementierungen beanstanden jedoch fehlerhafte IPv6-Pakete, etwa mit fehlerhaften Hop-By-Hop Header-Erweiterungen. RFC 4443 untersagt zwar den Versand von ICMPv6-Fehlermeldungen für Pakete, die an eine Multicast-Adresse gesendet wurden, für den Fall von Parameterproblemen wird jedoch ausdrücklich eine Ausnahme gemacht. Die entsprechende ICMPv6-Fehlermeldung verrät dann die Adresse des Nodes. Auf diese Weise kann man relativ einfach ohne sequenzielle Aufzählung eine Liste aller aktiven Systeme im Netz erhalten. Diese Vorgehensweise wird wie bereits in Abschnitt 3.3 erwähnt beispielsweise vom Tool alive6 aus der THC-IPv6-Suite oder scan6 aus dem IPv6-Toolkit angewendet.

Selbst hier sind Restriktionen über geeignete Firewallregeln denkbar, die auch diesen Mechanismus unbrauchbar machen. In diesem Fall bleibt nur noch ein Mittel der Aufzählung, nämlich der Versand fingierter DAD-Nachrichten an die entsprechenden Solicited-Node Multicast-Adressen, denn diese müssen von allen standardkonformen Nodes obligatorisch beantwortet werden. Der Nachteil dieser Methode ist, dass auch hier alle in Frage kommenden Solicited-Node Multicast-Adressen aufgezählt werden müssen, insgesamt wiederum 2^{24}. Dieser Aufwand ist zwar on-link relativ leicht zu bewältigen, aber nicht vernachlässigbar.

Als Administrator eines Netzes kann man sich jedoch auch zunutze machen, dass kein Node ohne Neighbor Discovery mit anderen Nodes kommunizieren kann; eine entsprechende Überwachung der ND-Pakete an geeigneter Stelle kann aufzeigen, welche Nodes sich im Netz befinden oder ob sich unerwartete Nodes im Netz befinden.

4.3 Privacy Extensions in Unternehmensnetzen

Gelegentlich tauchen in Gesprächen, in Vorträgen oder in schriftlichen Beiträgen Bedenken gegen die Verwendung von Privacy Extensions in Unternehmensnetzen auf. Die Quelle der Bedenken ist ein – zumindest subjektiv wahrgenommener – Kontrollverlust, etwa für die Nachvollziehbarkeit der Aktivitäten im Netz. Dem kann durch geeignete Überwachung und Protokollierung bei der Zugangskontrolle entgegengewirkt werden, was allerdings einvernehmliche Regelungen mit dem Datenschutz und/oder mit der Mitarbeitervertretung erfordern kann. Ob die Bedenken gegen Privacy Extensions gerechtfertigt sind, hängt von den konkreten Einsatzszenarien und Anforderungen ab. Sollen Privacy Extensions unterbunden werden, so kann dies technisch durch die Unterbindung von SLAAC für (global) routbare Adressen erreicht werden, indem die entsprechenden Flags in den Router Advertisements gesetzt werden; zumindest automatisch vergebene Privacy Extensions werden damit verhindert.

Derzeit wird an den sogenannten Stable Privacy Extensions gearbeitet [Rafiee 2013], bei denen die Interface ID in Abhängigkeit vom Präfix bestimmt wird, so dass in unterschiedlichen Netzen unterschiedliche Interface Ids verwendet werden, innerhalb eines Netzes aber stets dieselbe Interface ID verwendet wird. Diese Vorgehensweise hat eine Wirkung auf alle drei Szenarien aus Abschnitt 3.5; eine abschließende Beurteilung steht noch aus.

4.4 Wilde IP-Adressen

IP-Adressen können unter IPv6 derzeit auf eine oder mehrere der folgenden Arten bezogen werden: SLAAC, DHCPv6, Privacy Extensions, SEND mit CGA oder manuelle Konfiguration. Im Vergleich dazu bietet IPv4 nur DHCP und manuelle Konfiguration der IP-Adresse. Bis auf DHCPv6 kann der Zustand (belegt, verfügbar, reserviert etc.) einer Adresse aus Sicht eines übergeordneten Managements nicht administriert und nicht oder nur sehr aufwändig ermittelt werden. Zustandslose Adressvergabe ist dabei zumindest auf Link-Local-Ebene ausdrücklich erwünscht.

Wird eine zustandsbehaftete, gemanagte Adressvergabe durch DHCPv6 gewünscht, dann enthalten die Router Advertisements die entsprechenden Flags für alle in Frage kommenden Präfixe. Eine konforme IPv6-Implementierung wird die IP-Adressen für die betreffenden Präfixe dann über DHCPv6 beziehen. Das hält eine privilegierte Anwendung auf dem Node jedoch nicht davon ab, weitere, „wilde“ Adressen für das Interface zu konfigurieren, beispielsweise in Manier der Privacy Extensions. Auch der Einsatz von SEND und CGA ändert daran nichts. Neighbor Discovery funktioniert in diesem Fall wie gehabt, und was der Router mit Paketen macht, die „wilde“ IP-Adressen enthalten, wird durch die IPv6-Spezifikation nicht festgelegt.

Wenn die rigorose Durchsetzung einer Richtlinie zur Adressvergabe gewünscht wird, dann muss beispielsweise der Router des entsprechenden Netzes dies mit entsprechenden Filterregeln durchsetzen. Alternativ wäre es auch denkbar, einen Sensor im Netz zu platzieren, der MAC- und IPv6-Adressen mit Zugriffslisten abgleicht und bei einer Abweichung davon eine Alarmierung auslöst. Ein derartiger Sensor kann dabei noch weitere, ähnliche Aufgaben übernehmen, siehe Abschnitt 3.1.2.4.

4.5 Implementierungen und Produkte

IPv6 wurde zwar bereits vor 15 Jahren zum ersten Mal spezifiziert, jedoch lange Zeit von den Herstellern weitgehend ignoriert. Erst seit jüngerer Zeit bemühen sich viele Hersteller ernsthaft, IPv6 in ihre Produkte zu integrieren. Da IPv6 auch heute noch gelegentlich

Nachbesserungen erfährt, ist zu erwarten, dass diese Nachbesserungen erst mit Verzögerungen in existierende Produkte einfließen. Daher ist vor allem bei der Beschaffung von Hardware darauf zu achten, dass alle sicherheitskritischen RFCs angemessen umgesetzt wurden, beispielsweise die RFCs zu Routing Header 0 [RFC 5095] oder zu überlappenden Fragmenten [RFC 5722]. Um diesem Problem zu begegnen hat RIPE ein Dokument herausgegeben [RIPE 554], bei dem für verschiedene Gerätetypen präzise Anforderungen gestellt werden, welche RFCs mindestens erfüllt sein müssen. Einen weitergehenden Prüfkatalog wurde unter Federführung des Bundesverwaltungsamts für die öffentliche Verwaltung erarbeitet [BVA 2013], der in den meisten Fällen im Wesentlichen von beliebigen Organisationen übernommen werden kann.

Darüber hinaus sollte berücksichtigt werden, dass eine vom Hersteller angegebene IPv6-Unterstützung u. U. noch nicht alle Aspekte eines Produktes umfasst. So existieren nach Untersuchungen der Firma ERNW noch zahlreiche Firewalls namhafter Hersteller, die zwar IPv6-Verkehr regeln können, deren Management-Oberflächen, etwa über SSH, HTTP oder SNMP, jedoch nicht unter IPv6 ansprechbar sind.[25] Ebenso verhält es sich noch oft mit Protokollen wie RADIUS, NTP, Syslog oder Routing-Protokollen. Dies ist bestenfalls kurios; gefährlich wird es, wenn ein Malware- oder Spam-Schutz oder eine UTM-Lösung zwar beispielsweise Pattern-Updates auch über IPv6 beziehen kann, aber aktiv nur den IPv4-Verkehr inspiziert, den IPv6-Verkehr jedoch unkontrolliert passieren lässt. Bei der Beschaffung von derartigen Produkten sollte unbedingt darauf geachtet werden, dass die relevanten Funktionen auch für IPv6 uneingeschränkt zur Verfügung stehen.

5 Maßnahmen

Dieser Abschnitt zeigt verschiedene Maßnahmen auf, die zu einem sicheren Betrieb von IPv6 beitragen können. Dies ist nicht als Checkliste zu verstehen, die die eigene eingehende Beschäftigung mit IPv6 überflüssig macht, sondern als Liste von Punkten, über die sich die jeweiligen Betriebsverantwortlichen Gedanken machen sollten. Punkte, die nicht IPv6-spezifisch sind oder im Kontext von IPv6 keine Besonderheit darstellen, werden hier nicht aufgeführt, beispielsweise die Härtung der beteiligten Systeme. IPsec wird hier ebenfalls nicht näher betrachtet, weil die Überlegungen zum sicheren Einsatz von IPsec weitgehend unabhängig von IPv6 sind.

Weitere konkrete Maßnahmenvorschläge finden sich beispielsweise in den BSI-Standards zur Internet-Sicherheit [ISi-LANA, Kapitel 7] oder beispielsweise Abschnitt 8.6 im Migrationsleitfaden des Bundesverwaltungsamtes [BVA 2013], wenig Konkretes erhält man dagegen aus dem BSI-Leitfaden zu IPv6 [ISi-L-IPv6]. Das IPv6 Security Audit/Assurance Program der ISACA [ISACA 2012] orientiert sich stark an allgemeinen ISMS-Erwägungen und bietet auf technischer Ebene überraschend wenig IPv6-Spezifisches.[26]

5.1 Organisatorische Maßnahmen

Genau wie in der übrigen IT leisten auch bei IPv6 geeignete organisatorische Maßnahmen den größten Beitrag zum sicheren Betrieb. Hierzu zählen die nachstehend aufgeführten Maßnahmen, die den jeweils relevanten Management-Gebieten nach ISO 27001 Annex A zugeordnet wurden. Die Maßnahmen dieses Annexes stellen im Informationssicherheits-

---

25 *Overview of the Real-World Capabilities of Major Commercial Security Products*, Workshop auf dem IPv6 Security Summit 2013.

26 Und das Wenige, was dort zu finden ist, ist teilweise falsch oder veraltet, wie beispielsweise Forderungen nach totaler Filterung von ICMP zeigt [ISACA 2012, Punkt 7.2.1.6].

management – unabhängig von IPv6 – ohnehin Best Practice dar. Die hier genannten Punkte sollten insbesondere im Zuge einer Einführung von IPv6 besonders sorgfältig befolgt werden.

5.1.1 Umgang mit IPv6 (A 12.1.1)

Betriebssysteme wie Linux, FreeBSD oder Windows 7 bieten die für IPv6 notwendigen Funktionen an und haben bereits in der Grundkonfiguration IPv6 aktiviert. Selbst wenn die Verantwortlichen eines IT-Betriebes sich dazu entscheiden, IPv6 zunächst nicht einzusetzen, ist Untätigkeit fehl am Platz, da die IPv6-Tunnelmechanismen es u. U. ermöglichen, aus dem Internet6 ungehindert auf interne Netze einer Organisation zuzugreifen. Es ist daher notwendig, eine strategische Management-Entscheidung zum Einsatz von IPv6 zu treffen und diese dann konsequent umzusetzen. Dies bedeutet beispielsweise, dass die entsprechenden Funktionen und Dienste in den Netzen deaktiviert werden, in denen noch kein IPv6 verwendet werden soll.

Für Netze, in denen IPv6 verwendet werden soll, sollte frühzeitig entschieden werden, wie IP-Adressen zugewiesen werden, also ob SLAAC verwendet werden oder DHCPv6 zum Einsatz kommen soll. Im Fall des Einsatzes von SLAAC sollte entschieden werden, wie mit Privacy Extensions verfahren werden soll. Weitere Punkte, zu denen eine explizite Entscheidung vorliegen sollte, betreffen den Einsatz und die Auswahl von Übergangsmechanismen sowie die Nutzung von Multicast oder MIPv6.

5.1.2 Aus- und Weiterbildung (A 8.2.2)

Aus den bisherigen Ausführungen sollte an dieser Stelle bereits klar geworden sein, dass sich IPv6 in vielen Dingen von IPv4 unterscheidet, wobei es sich dabei nicht nur Details handelt, sondern um neue Architekturelemente. IPv6 ist nicht einfach nur IPv4 mit längeren Adressen – Unkenntnis der Verantwortlichen über IPv6-Mechanismen und deren Wirkung gilt allgemein als eine der größten Bedrohungen beim Einsatz dieses Netzwerkprotokolls. Die Verantwortlichen einer Organisation sollten daher rechtzeitig für eine praxisgerechte und der Aufgabe angemessene Aus- und Weiterbildung zumindest einiger Schlüsselpersonen sorgen, um IPv6 möglichst störungsfrei und sicher einführen zu können. In dem Migrationsleitfaden des Bundesverwaltungsamtes wird empfohlen [BVA 2013, Abschnitt 5.3], zunächst allgemeine und hersteller-unabhängige Grundlagenschulungen durchzuführen und darauf mit Schulungen zu konkreten Produkten sowie zur IPv6-Sicherheit aufzubauen.

5.1.3 Einrichtung von Testnetzen (A 10.1.4)

Die Einrichtung von Testnetzen dient sowohl dem Erwerb von Erfahrung im Umgang mit IPv6, als auch der Erprobung und Abnahme von neuen Netzwerkkomponenten (siehe Abschnitt 5.1.5). In bestimmten Fällen können Testnetze auch zur Aufklärung von Störungen verwendet werden.

5.1.4 Überprüfung von Dienstleistern (A 10.2.2)

Werden die Dienste von Dritten im Kontext von IPv6 in Anspruch genommen, sollte sichergestellt werden, dass diese Dienstleister über die notwendige Kompetenz verfügen. Darüber hinaus sollten Dienstleistungen im Bereich von IPv6 in der Anfangsphase besonders sorgfältig überwacht werden.

5.1.5 Kriterien zur Systemabnahme (A 10.3.2)

Auch wenn die Standards zu IPv6 eine gewisse Stabilität erreicht haben, wird sich auch IPv6 ständig weiterentwickeln, wobei Bestehendes teilweise verworfen oder durch Neues ergänzt oder ersetzt werden wird. Hier gilt es, die aktuellen Entwicklungen im Auge zu behalten und einen Abgleich gegen die geltenden Best-Practice Empfehlungen vorzunehmen. Hervorzuheben ist hier beispielsweise RIPE-554 [RIPE 554]. In diesem Dokument wird je nach Gerätetyp festgelegt, welche Mindestanforderungen die IPv6-Implementierung eines Geräts erfüllen sollte. Diese Anforderungen sollten in die eigenen Kriterien zur Systemabnahme einfließen.

Darüber hinaus hat das Bundesverwaltungsamt kürzlich einen Migrationsleitfaden veröffentlicht [BVA 2013]. Die Zielgruppe dieses Leitfadens sind zwar Behörden und die öffentliche Verwaltung, der Leitfaden ist größtenteils jedoch allgemein gültig und dürfte für viele Administratoren und IT-Verantwortliche in den Unternehmen von großem Nutzen sein.

5.1.6 Neustrukturierung und Konfiguration der Netze (A 10.6.1)

Die Einführung von IPv6 bietet eine gute Gelegenheit, die Beschränkungen von IPv4 hinter sich zu lassen und große Netzwerke in angemessener Weise in kleinere Netzwerke und Vertrauenszonen zu unterteilen. Hierbei sollten Systeme nach Schutzbedarf und administrativer Verantwortlichkeit gruppiert in eigenen Netzen untergebracht werden.

Systeme in Netzen, die nicht mit IPv6 betrieben werden sollen, sollten so konfiguriert werden, dass IPv6 deaktiviert wird, dass keine IPv6-spezifischen Dienste angeboten oder in Anspruch genommen werden und dass keine IPv6-Tunnel (vgl. Abschnitt 5.2.1.4) etabliert werden können. System, die mit IPv6 betrieben werden sollen, sollten ebenfalls so konfiguriert werden, dass keine unerwünschten IPv6-Tunnel aufgebaut werden.

Sofern IPv6-Tunnel verwendet werden sollen, sollten die entsprechenden Tunnelausgänge nicht automatisch, sondern manuell eingerichtet werden. Tunnelverkehr ist im Zweifelsfall nicht vertrauenswürdig und sollte ebenso behandelt werden, wie anderer Verkehr vom und zum Internet.

5.1.7 Überwachung der Systemnutzung (A 10.10.2, A 10.10.5, A 13.2.2)

Vor allem in der Anfangsphase sollte der Netzverkehr überwacht werden, um erstens eine Routine für den Betrieb von IPv6-Netzen zu entwickeln und um zweitens Anomalien erkennen zu können. Eine bloße Protokollierung auf Vorrat reicht hierzu nicht aus, stattdessen sollte insbesondere in der Anfangsphase eine regelmäßige Auswertung der relevanten Protokolldaten erfolgen. Für die Auswertung von Ereignisprotokollen sollten angemessene Werkzeuge verwendet werden. Die Erkenntnisse sollten dafür verwendet werden, um die Angemessenheit von Maßnahmen zu überprüfen und ggf. Änderungen zu planen. Die Aufzeichnung von Ereignissen sollte in angemessener Detailtiefe erfolgen, wobei jedoch rechtliche Anforderungen beispielsweise aus dem Datenschutz nicht verletzt werden dürfen.

5.1.8 IPv6-Adressen am Benutzerinterface (A 12.2.1)

Bei der Entwicklung von Anwendungen sollte darauf geachtet werden, dass IPv6-Adressen korrekt validiert werden. Nach Möglichkeit sollte die Komplexität von IPv6-Adressen vor Endanwendern verborgen bleiben, etwa durch ausschließliche Verwendung von DNS-Namen.

5.1.9 Nachvollziehbarkeit und Überprüfung von Änderungen (A 12.5.2)

Der Betrieb von IPv6 wird vor allem in der Anfangsphase zahlreiche betriebsspezifische Anpassungen an der Netzwerkkonfiguration oder an Anwendungen erfordern. Die daraus resultierenden Änderungen sollten nicht „wild“ durchgeführt werden, sondern gesteuert vorgenommen werden. Das bedeutet vor allem, dass die Änderungen dokumentiert und vor der produktiven Aktivierung angemessen erprobt werden. Hierbei ist darauf zu achten, dass alle relevanten Bereiche berücksichtigt werden, beispielsweise Adressformate, DNS usw. Im Rahmen eines Informationssicherheitsmanagements ist diese Vorgehensweise zwar grundsätzlich selbstverständlich; im Rahmen einer IPv6-Einführung sollte hier aber besonders sorgfältig vorgegangen werden.

5.1.10 Umgang mit Schwachstellen und Vorfällen (A 12.6.1, A 13.2.1)

Es ist zu erwarten, dass die mit IPv6 betriebenen Systeme und Anwendungen noch nicht denselben Reifegrad aufweisen, wie er für IPv4 heutzutage vorausgesetzt werden kann. Es ist wahrscheinlich, dass diese Systeme und Anwendungen Schwachstellen aufweisen, für deren Behandlung sich noch keine Routine bei den Netzwerkverantwortlichen sowie bei den Herstellern entwickelt hat. Es sollte ein enger Kontakt mit den Herstellern und Produktlieferanten gepflegt werden, um bei Bekanntwerden von Schwachstellen adäquat reagieren zu können. Es sollten beispielsweise Vorkehrungen getroffen werden, um gefährdete oder kompromittierte Systeme unkompliziert in separate (Quarantäne-)Netze zu verbringen, und es sollte ausreichend Kompetenz für den Umgang mit IPv6-relevanten Sicherheitssystemen wie Firewalls aufgebaut werden, um kurzfristig angemessene Schutzmaßnahmen ergreifen zu können.

5.2 Technische Maßnahmen

5.2.1 Filterung

Für Filterregeln gelten auf Anwendungsebene grundsätzlich dieselben Prinzipien wie von IPv4 her bekannt. Die IPv6-spezifischen Besonderheiten werden nachstehend aufgezählt.

5.2.1.1 ICMPv6

ICMPv6 darf nicht pauschal gefiltert werden. Eine Empfehlung zur zielgerichteten Filterung von ICMPv6 wird in RFC 4890 gegeben.

- Neighbor und Router Discovery sowie Redirects sollten ausschließlich link-local zugelassen werden und dürfen nicht geroutet werden. Systeme mit RFC-konformer IPv6-Implementierung verwerfen unzulässige Nachrichten schon beim Eingang, eine dedizierte Filterung ist daher nicht notwendig. Es sollte jedoch geprüft werden, ob die jeweiligen Systeme in diesem Sinne RFC-konform sind, siehe Abschnitt 5.2.4. Wird SEND nicht genutzt, dann sollten die SEND-spezifischen ICMP-Nachrichten gefiltert werden.

  Fragmentierte NDP-Nachrichten müssen seit Neuestem verworfen werden [RFC 6980]. Da derzeit noch keine Implementierung dies umsetzt, sollte hier mit entsprechenden Regeln gefiltert werden. Denkbar ist auch der radikale Ansatz, Nachrichten der Neighbor Discovery nur dann zu verarbeiten, wenn sie überhaupt keine Header-Erweiterungen enthalten.

- Multicast Listener Discovery muss link-lokal zugelassen werden und sollte darüber hinaus gefiltert werden, sofern keine Multicast-Anwendungen betrieben oder genutzt werden.

- Fehlermeldungen sollten zustandsabhängig zugelassen werden. Im Rahmen der PMTU Discovery müssen Packet Too Big Fehlermeldungen auf jeden Fall zugelassen werden.
- Über die Zulässigkeit von Ping gibt es divergierende Ansichten. Als sinnvoller Kompromiss könnte Ping nur link-local oder mit einer Beschränkung auf Unicast-Absendeadressen zugelassen werden; eine zusätzliche Beschränkung der Übertragungsrate ist in jedem Fall sinnvoll.
- Wird kein Mobile IPv6 eingesetzt, dann sollten die entsprechenden ICMP-Nachrichten gefiltert werden.
- Alle weiteren ICMP-Nachrichten sollten gefiltert werden, sofern diese nicht für das zu schützende Netzwerk spezifisch in explizit definierter Weise erforderlich sind.

Grundsätzlich gilt, dass Stürme von ICMP-Paketen nicht normal sind, so dass für ICMP prinzipiell die Übertragungsrate auf ein sinnvolles Maß beschränkt werden sollte.

5.2.1.2 Ungültige Adressbereiche

Präfixe für ungültige Adressbereiche sollten ausgefiltert werden, insbesondere die folgenden:

- Documentation – `2001:db8::/32`
- Benchmarking – `2001:2::/48`
- Orchid – `2001:10::/28`
- Site Local – `fec0::/10`
- IPv4-mapped – `::ffff:0:0/96`
- IPv4-embedded – `::0.0/96`

Darüber hinaus sollten die folgenden Präfixe ausgehend ausgefiltert werden, sofern kein spezifischer Anwendungsfall für einen der betreffenden Tunnelmechanismen vorliegt:

- Teredo – `2001::/32`
- 6to4 – `2002::/16`

An Site-Grenzen sollten die folgenden Präfixe ausgefiltert werden:

- Unique Local Addresses – `fc00::/7`

Darüber hinaus sollte Multicast (`ff00::/8`) wenigstens an Site-Grenzen gefiltert werden. Wird Multicast Site-überschreitend genutzt, sollten nur entsprechende Scopes zugelassen werden (`ff08::/8` oder `ff0e::/8`). Sind keine Multicast-Anwendungen im Einsatz, dann sollte Multicast außer für die Neighbor Discovery pauschal gefiltert werden.

Grundsätzlich werden global routbare Präfixe derzeit nur aus dem Block `2000::/3` vergeben, d. h. Verkehr aus anderen Blöcken (ggf. mit Ausnahme von Multicast) kann am Übergang zum Internet6 abgewiesen werden.

Auch innerhalb des Blocks `2000::/3` sind ebenfalls viele Teilblöcke noch nicht vergeben. Die Ansichten darüber, ob Verkehr nur aus den jeweils aktuell gültigen Belegungen zugelassen werden soll, gehen auseinander. Der hier vertretene Standpunkt dazu lautet, dass der Sicherheitsgewinn vernachlässigbar ist – ob ein Angreifer aus einem unbelegtem Block oder aus einem belegten Block heraus agiert, spielt für die Netzwerksicherheit keine Rolle. Sollte die Erfolgsaussicht für einen Angriff welcher Art auch immer tatsächlich von diesem Unterschied abhängen, dann sind das Sicherheitskonzept oder die ergriffenen Sicherheitsmaßnahmen nicht stimmig. Darüber hinaus erhöht jeder denkbare Mechanismus

zur Aktualisierung der jeweils gültigen Belegung lediglich die Komplexität der beteiligten Prozesse und schafft neue und konkrete Möglichkeiten zur Betriebsbeeinträchtigung.

Als Kompromisslösung ist denkbar, Verkehr nur von und zu den Adressbereichen zuzulassen, die durch IANA den regionalen Registrierungsstellen zugewiesen wurde.[27] Diese Zuweisungen ändern sich relativ selten und sind an Anzahl noch recht überschaubar.

5.2.1.3 Anti-Spoofing – Reverse Path Forwarding (RPF)

Reverse Path Forwarding (RPF) ist eine Maßnahme zur Verhinderung von Adressfälschung (Spoofing) [RFC 2827, RFC 3704]. RPF ist keine IPv6-spzifische Maßnahme, sondern ist auch unter IPv4 bereits bekannt gewesen. Der Kern der Maßnahme besteht darin, dass an einem Router prinzipiell bekannt ist, welche Pakete mit bestimmten Absendeadressen über welche Routen zu dem Router gelangen können. Ist beispielsweise ein Netz mit dem Präfix `2001:db8:0:cafe::/64` an einem Port des Routers angeschlossen, dann kann – ohne Berücksichtigung von Multihoming – kein legitimes Paket mit einer entsprechenden Absendeadresse über einen anderen Port zu dem Router gelangen. Im Fall von Multihoming ist RPF nur eingeschränkt möglich [RFC 3704]. Der Einsatz von RPF ist Best Practice und sollte nach Möglichkeit stets verwendet werden, insbesondere wenn kein Multihoming zum Einsatz kommt.

5.2.1.4 Tunnelprotokolle

Automatisch konfigurierte Tunnel haben den Zweck, den Zugang zu IPv6-Netzen über IPv4-Infrastrukturen mit möglichst wenig administrativem Aufwand zu erreichen. Es kommen hierfür verschiedene Mechanismen zum Einsatz:

- 6over4, 6to4 und ISATAP nutzen Transportprotokoll 41, um IPv6-Pakete in IPv4-Pakete einzubetten.
- Denkbar ist auch eine Nutzung von Transportprotokoll 47 (GRE).
- Teredo nutzt UDP Port 3544, um IPv6-Pakete in UDP-Pakete einzubetten.

Generell sollten keine automatisch konfigurierten Tunnel genutzt werden, um keine unkontrollierten Zugänge zum eigenen Netzwerk zu schaffen. Dem entsprechend sollten die Transportprotokolle 41 und 47 sowie UDP-Protokoll 3544 an Netzwerkübergängen gefiltert werden. Den üblichen Best-Practice-Empfehlungen folgend, nach denen zulässige Protokolle und Dienste explizit zugelassen werden und alles andere gefiltert wird (White-Listing), ergibt sich eine Filterung auf natürliche Weise.

Ein Angreifer könnte jedoch auch ein modifiziertes Arrangement wählen, bei dem unübliche und harmlos erscheinende Protokollnummern und -Ports verwendet werden, beispielsweise die Nutzung von UDP Ports 53 (DNS) oder 123 (NTP) für Teredo – hiergegen hilft nur Deep Packet Inspection und eine rigorose Beschränkung des ausgehenden IPv4-Verkehrs.

5.2.1.5 Header-Erweiterungen

In Abschnitt 3.3 wurde bereits ausgeführt, dass für typische Betriebsszenarien die meisten Header-Erweiterungen mit den folgenden beiden Ausnahmen gefiltert werden können:

- Hop-by-Hop Options dürfen für Nachrichten der Multicast Listener Discovery nicht gefiltert werden. Jedoch sollten diese Nachrichten bei exzessivem Padding verworfen werden.

---

27 Siehe IANA IPv6 Global Unicast Address Assignments

- Einfache Fragment Options dürfen grundsätzlich nicht gefiltert werden, mehrfache Fragment Options sollten immer gefilter werden.

Alle anderen Header-Erweiterungen sollten gefiltert werden, sofern kein spezifischer Anwendungszweck vorliegt. Im Fall von Hop-by-Hop oder Destination Options sollten die entsprechenden Pakete verworfen werden, wenn exzessives Padding angewendet wurde.

Pakete der Neighbor und Router Discovery mit Header-Erweiterungen jeglicher Art sollten verworfen werden.

5.2.1.6 Fragmentierung

Fragmentierte Pakete stellen a priori kein Sicherheitsproblem dar. Es sollte jedoch darauf geachtet werden, dass insbesondere Firewalls und Gateways gemäß RFC 5722 Pakete mit überlappenden Fragmenten verwerfen.

In Verbindung mit Header-Erweiterungen sollten Pakete verworfen werden, deren initiales Fragment durch exzessives Padding der Erweiterungen den Upper-Layer Header nicht enthält.

Pakete der Neighbor und Router Discovery müssen grundsätzlich verworfen werden, wenn sie fragmentiert sind [RFC 6980]. Falls SEND eingesetzt wird, dann sind Certification Path Advertisements hiervon ausgenommen, da die hierfür nötigen X.509-Zertfikate in vielen Fällen legitim die MTU-Grenze überschreiten.

5.2.2 Sichere Neighbor Discovery

Die Möglichkeiten zur sicheren Neighbor Discovery und deren Vor- und Nachteile wurden bereits in Abschnitt 3.1 diskutiert. Aus praktischen Erwägungen heraus hergibt sich, dass eine Überwachung und Plausibilitätsprüfung der Neighbor und Router Discovery in Verbindung mit einer Aufteilung großer Netze in kleinere Netze (siehe Abschnitt 5.1.6) am effektivsten ist. Bei der Überwachung der Neighbor Discovery mit Tools wie Raguard und vergleichbaren Ansätzen sollte darauf geachtet werden, dass die jeweilige Implementierung nicht anfällig für einen Angriff über Fragmentierung in Verbindung mit Header-Erweiterungen ist.

5.2.3 Node-Konfiguration

Für den sicheren Betrieb von IPv6 spielt auch die Konfiguration der einzelnen Nodes eine Rolle, angefangen von der Entscheidung, ob ein Node überhaupt über IPv6 ansprechbar sein soll, über die Nutzung von SLAAC und Privacy Extensions bis hin zum Feintuning von IPv6-spezifischen Parametern. Die Konfiguration dieser Einstellungen ist für jedes Betriebssystem spezifisch geregelt, und nicht jeder relevante Parameter kann an jedem Betriebssystem individuell konfiguriert werden. Die relevanten Parameter umfassen u. a.:

- Aktivierung oder Deaktivierung von IPv6
- Aktivierung oder Deaktivierung von Tunnelschnittstellen (speziell bei Microsoft Windows)
- Nutzung von Privacy Extensions
- Nutzung von SLAAC
- Beachtung von Redirects
- Beachtung von Router Advertisements
- Verschiedene Timing Parameter

Im Rahmen einer Härtung der Betriebssystemplattform sollten die jeweiligen Einstellungen für die genutzten Systeme bestimmt werden, die zur Umsetzung der beabsichtigten Sicherheitsrichtlinie beitragen.

5.2.4 Technische Prüfungen

Es ist zu erwarten, dass die Routine und die Erfahrung im Umgang mit IPv6 in der Anfangszeit einer Umstellung noch nicht so ausgeprägt ist, wie beispielsweise im Hinblick auf IPv4. Gleichzeitig sind auch die Hersteller in Bezug auf IPv6 noch nicht so „trittsicher“ wie bei IPv4, was sich beispielsweise dadurch äußern kann, dass Vorgaben geltender RFCs nicht vollständig oder nicht korrekt umgesetzt werden. Hier gilt es für den Betreiber, in einem Testnetz das Verhalten der jeweiligen Komponenten näher zu untersuchen. Darüber hinaus werden Organisationen und Unternehmen, die ein Informationssicherheitsmanagement etabliert haben, regelmäßig technische Audits durchführen, um die technische Umsetzung der Sicherheitsrichtlinien zu überprüfen.

Zur technischen Prüfung von IPv6-spezifischen Details bewähren sich vor allem die Tools der THC-IPv6-Suite[28] von Marc Heuse, das IPv6-Toolkit[29] von Fernando Gont sowie Scapy[30] von Philippe Biondi. Die THC-IPv6-Suite enthält Tools, mit denen die Anfälligkeit für bestimmte bekannte Angriffe gezeigt werden kann, während das IPv6-Toolkit eher auf Audits ausgerichtet ist [Gont 2012a]. Die Nutzung einer jeden der beiden Werkzeugsammlungen kann zu Ausfällen im Netz führen. Scapy ist ein Werkzeug für Erzeugung und Versand beliebiger Pakete, das in Python geschrieben ist; Scapy-Scripte sind Python-Scripte, denen der Funktionsumfang von Scapy zur Verfügung steht. In Verbindung mit Werkzeugen wie Wireshark oder TCPdump kann die Reaktion im Netz auf derart beliebig gestaltete Pakete geprüft werden.

Darüber hinaus bietet auch der populäre Portscanner nmap verschiedene Scripte, die jedoch bei Weitem nicht an den Umfang der IPv6-Suite oder des IPv6-Toolkits heranreichen. Wer jedoch nur nmap installiert hat und schnell einen Überblick über die Systeme mit aktivem IPv6 haben möchte (analog zu alive6 oder scan6), kann hierfür auch die entsprechenden nmap-Skripte verwenden.[31]

---

28 http://www.thc.org/thc-ipv6/

29 http://www.si6networks.com/tools/ipv6toolkit/

30 http://www.secdev.org/projects/scapy/

31 Beispielsweise mit `nmap -6 --script=targets-ipv6-multicast-invalid-dst -sP`

6 Glossar

802.1X	IEEE-Standard für Netzwerkzugangsschutz.
Adressauflösung	Siehe *Address Resolution*
Adress-Autokonfiguration	Siehe *Stateless Address Autoconfiguration*
Adresse	Ein Bezeichner, der ein oder mehrere Interfaces auf IP-Ebene identifiziert.
AH	Authentication Header
All-Nodes Multicast-Adresse	Multicast-Adresse zum Versand von Nachrichten an alle Nodes in einem Scope, beispielsweise Link-Local. Die Link-Local All-Nodes Multicast-Adresse lautet `ff02::1`.
Address Resolution	Zuordnung einer Network-Layer-Adresse zu einer Link-Layer-Adresse, also beispielsweise die Zuordnung einer IP-Adresse zu einer MAC-Adresse. Bei IPv4 wird die Zuordnung zwischen IP-Adresse und MAC-Adresse über *ARP* vorgenommen; bei IPv6 wird dies mit *NDP* durchgeführt.
Address Resolution Protocol	Steuerprotokoll zur *Adressauflösung* mit IPv4 auf Ethernet.
All-Routers Multicast-Adresse	Multicast-Adresse zum Versand von Nachrichten an alle Router in einem Scope, beispielsweise Link-Local, die Link-Local All-Routers Multicast-Adresse lautet `ff02::2`.
Anycast	Kommunikation mit genau einem von mehreren Interfaces mit derselben Adresse. Denkbare Einsatzszenarien sind Dienste wie DNS oder NTP, die von mehreren Servern unter einer gemeinsamen, bekannten IP-Adresse aus bedient werden; in diesen Szenarien wären die Router dafür verantwortlich, die Anfragen an den „besten“ unter mehreren verfügbaren Servern weiterzuleiten.
Anycast-Adresse	Eine Adresse für mehrere Interfaces, die typischerweise zu unterschiedlichen Nodes gehören. Ein Paket, das per *Anycast* gesendet wird, wird an genau eines der Interfaces ausgeliefert, dass durch die Anycast-Adresse identifiziert wird. Bei IPv6 sind Anycast-Adressen von gewöhnlichen Unicast-Adressen nicht zu unterscheiden.
ARP	*Address Resolution Protocol*
CGA	*Cryptographically Generated Address*
CIDR	*Classless Inter-Domain Routing*
Classless Inter-Domain Routing	Klassenlose Einteilung eines Adressraums in Subnetze. Die Einteilung wird durch die CIDR-Notation *Adresse/Präfixlänge* kenntlich gemacht; bei IPv4 beispielsweise `172.16.0.0/16`, bei IPv6 beispielsweise `fe80::/10`.
Cryptographically Generated Address	IP-Adresse, bei welcher der Interface-Identifier auf kryptografische Weise gebildet wird. CGA wird im Rahmen der *Secure Neighbor Discovery* verwendet, um Schutz gegen gewisse Denial-of-Service-Angriffe im Rahmen der *Neighbor Discovery* zu bieten.
DAD	*Duplicate Address Detection*
Default Free Zone	Teil des Internets, in dem keine Default-Routen an Upstream-Router existieren.

Deprecated Address	Unicast- oder Anycast-Adresse nach Ablauf der Gültigkeitsdauer preferred lifetime.
DHCP	*Dynamic Host Configuration Protocol*
DNS	*Domain Name Service*
Duplicate Address Detection	Verfahren um sicherzustellen, dass im Rahmen der Stateless Automatic Address Configuration eine IP-Adresse nur einem Interface zugeordnet wird.
ESP	*Encapsulating Security Payload*
EUI	*Extended Unique Identifier*
EUI-64	64-Bit-Identifier, der nach Vorgaben der IEEE konstruiert wurde.
FCFS SAVI	First-Come First-Served SAVI; Methode zur Erkennung von Adressfälschungen am Link, wobei Adressen nach dem First-Come-First-Serve-Prinzip an Switch-Ports gebunden wird.
FIPS	Federal Information Processing Standard
Global Unique Address	Global routbare Adresse; derzeit jede Adresse aus dem Bereich `2000::/3` mit Ausnahme der für spezielle Zwecke reservierten Bereiche.
GRE	Generic Routing Encapsulation
GUA	*Global Unique Address*
H.323	Protokollstandard der ITU-T zur audiovisuellen Kommunikation, u. a. für Internet-Telefonie
HMAC	Hashed Message Authentication Code
HMAC-SHA-512	HMAC unter Verwendung von SHA-512
Host	Ein Node, der kein Router ist, d. h. der keine Pakete weiterleitet.
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICMP	*Internet Control Message Protocol*
IEEE	Institute of Electrical and Electronics Engineers
ID	Identifier
IGMP	*Internet Group Management Protocol*
IKE	Internet Key Exchange Protocol. Die aktuelle Version 2 des Protokolls (IKEv2) ist in RFC 5996 spezifiziert.
IND	*Inverse Neighbor Discovery*
Interface	Schnittstelle, über die ein Node an einen Link angeschlossen wird.
Internet4	Der Teil des Internets, der über IPv4 erreichbar ist.
Internet6	Der Teil des Internets, der über IPv6 erreichbar ist.
Internet Control Message Protocol	Steuerprotokoll für IP. ICMP-Nachrichten werden in IP-Paketen gekapselt. Für IPv6 ist ICMPv6 das einzige Steuerprotokoll, bei IPv4 ist ICMP eines von mehreren Steuerprotokollen.
Internet Group Management Protocol	Steuerprotokoll zur Verwaltung von Multicast-Routen. Mit IPv6 ist dieses Protokoll als *MLD* in *ICMP* aufgegangen.
Intra-Site Automatic Tunnel Addressing Protocol	Tunnelprotokoll, bei dem ein IPv6-Tunnel automatisch über IPv4 aufgebaut wird.

Invalid Address	Unicast- oder Anycast-Adresse nach Ablauf der Gültigkeitsdauer valid lifetime.
Inverse Address Resolution	Zuordnung einer Link-Layer-Adresse zu einer Network-Layer-Adresse, also beispielsweise die Zuordnung einer MAC-Adresse zu einer IP-Adresse.
Inverse Adressauflösung	Siehe *Inverse Address Resolution*
Inverse Neighbor Discovery	Auflösung von *Link-Layer-Adressen* zu IP-Adressen.
IP	Internet Protocol
IPsec	IP Security
ISATAP	*Intra-Site Automatic Tunnel Addressing Protocol*
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
Jumbogramm	IPv6-Paket mit einer Payload von mehr als 65535 Bytes.
LAN	Local Area Network
Link	Kommunikationsmedium, über das Nodes auf Link-Layer miteinander kommunizieren können.
Link MTU	Siehe *Maximum Transmission Unit*.
Link-Layer Address	Adresse eines Interfaces auf Link-Layer, beispielsweise die MAC-Adresse bei Ethernet oder WLAN.
Link-Local Address	Adresse mit Link-Scope, die dafür verwendet werden kann um alle Nachbarn am selben Link zu erreichen. Jedes Interface muss eine Link-Local Address haben.
LLA	*Link-Local Address*
MAC	Media Access Control
Maximum Transmission Unit	Maximal zulässige Paketgröße auf Link-Ebene.
MLD	Siehe *Multicast Listener Discovery*
MSF	Siehe *Multicast Source Filter*
MTU	Siehe *Maximum Transmission Unit*
Multicast	Multicast dient der Verteilung von Inhalten an mehrere Interfaces, ohne den Inhalt separat an jedes Interface individuell zu versenden. Anwendungsszenarien liegen vor allem in der Verbreitung von Multimedia-Inhalten, also beispielsweise Internetradio usw.
Multicast-Adresse	Adresse, die mehreren Interfaces zugewiesen wird.
Multicast Listener Discovery	Protokoll zur Verwaltung von Multicast-Routen. Dieses Protokoll ist IPv6-spezifisch und löst das alte *IGMP* ab. MLD wird über ICMPv6 übertragen.
Multicast Source Filter	Funktion in MLDv2, um Multicast-Verkehr aus bestimmten Quellen zu filtern.
Nachbar	Siehe *Neighbor*
NA	*Neighbor Advertisement*

NAT	*Network Address Translation*
ND	*Neighbor Discovery*
NDP	*Neighbor Discovery Protocol*
Neighbor	Der Neighbor eines Nodes ist ein jeder Node, der über einen Link direkt mit diesem verbunden ist.
Neighbor Advertisement	*NDP*-Nachricht als Antwort auf eine *Neighbor Solicitation*.
Neighbor Cache	IPv6-Äquivalent zum IPv4-ARP-Cache.
Neighbor Discovery	Im engeren Sinn Methoden zur *DAD*, zur *NUD* und zur *Adressauflösung*; im weiteren Sinn Zusammenfassung aller Methoden zur Bestimmung von Nachbarn und Routern sowie der dazugehörigen Parameter.
Neighbor Discovery Protocol	Protokoll zum Austausch von link-relevanten Nachrichten, wie *Router Discovery* und *Neighbor Discovery*. NDP-Nachrichten sind Bestandteil von ICMPv6 und dürfen am *Link* typischerweise nicht gefiltert werden.
Neighbor Solicitation	NDP-Nachricht zum Zweck der *DAD*, der *NUD* oder der *Adressauflösung*.
Neighbor Unreachability Detection	Verfahren zur Feststellung, ob *Neighbors* am Link noch erreichbar sind.
Network Address Translation	Verfahren zur Abbildung von IP-Adressen auf andere IP-Adressen. Die Technik wird vor allem dazu verwendet, um den IPv4-Adressraum künstlich auszudehnen und damit der Adressknappheit entgegenzuwirken.
Network Time Protocol	Protokoll zur Synchronisierung der lokalen Uhrzeit auf einem System mit der Uhrzeit auf einem Referenzsystem.
Next-Hop Determination	Bestimmung des nächsten Nodes. Next-Hop Determination erfolgt über ICMP-Redirect-Nachrichten und ist im Wesentlichen dann relevant, wenn mehr als ein Router an einem Link angeschlossen ist.
Node	Ein System, das IP-Pakete senden und empfangen kann.
NS	*Neighbor Solicitation*
NTP	*Network Time Protocol*
NUD	*Neighbor Unreachability Detection*
Off-link	Ein Node ist off-link zu einem anderen Node, wenn nicht beide Nodes an denselben Link angeschlossen sind, d. h. wenn Pakete von einem Node zu dem anderen Node von einem Router weitergeleitet werden müssen.
On-link	Ein Node ist on-link zu einem anderen Node, wenn sie an denselben Link angeschlossen sind, d. h. wenn sie Nachbarn sind.
ORCHID	Overlay Routable Cryptographic Hash Identifiers
Parameter Discovery	Mechanismus, um Hosts an einem Link über verschiedene gültige relevante Parameter wie MTU oder Hop Limit zu informieren. Diese Informationen werden über *Router Advertisements* verteilt.
Path MTU	Mechanismus zur Bestimmung der MTU entlang des gesamten Pfades von einem Absender zu einem Empfänger.
Path MTU Discovery	Verfahren zur Bestimmung der Path MTU

PEX	*Privacy Extensions*
Präfix	Der höchstwertige (linke) Teil einer IPv6-Adresse. Die Präfixlänge wird durch die entsprechende CIDR-Notation angegeben. Das Präfix von `2001:db8:7a29:307:29ff:fe4a:30c:4209/56` lautet beispielsweise `2001:db8:7a29:300::/56`.
Prefix Discovery	Mechanismus, um Hosts an einem Link über die nutzbaren Präfixe am Link zu informieren. Diese Informationen werden über *Router Advertisements* verteilt.
Preferred Address	Unicast- oder Anycast-Adresse, deren Eindeutigkeitsprüfung am Link „vor Kurzem“ erfolgreich abgeschlossen wurde.
Privacy Extensions	Mechanismus zur regelmäßig wiederholten Vergabe jeweils zufälliger Interface Ids zur Wahrung der Privatsphäre des Nutzers.
RA	*Router Advertisement*
RADIUS	Remote Authentication Dial-In User Service
RARP	*Reverse Address Resolution Protocol*
Redirect	ICMP-Nachricht, mit der ein Router einen Host über einen besseren First Hop informiert.
Reverse Address Resolution Protocol	Steuerprotokoll zur *inversen Adressauflösung*.
Reverse Path Forwarding	Maßnahme zur Verhinderung von Adressfälschung.
Router	Ein Router ist ein Node, der Pakete weiterleitet, die nicht an an ihn selbst gerichtet sind.
Router Advertisment	Ein Router Advertisement ist eine Nachricht der *NDP*-Familie, mit der ein Router verschiedene Netzwerkparameter am Link verbreitet, hier vor allem Informationen über seine Existenz und über die geltenden Präfixe. Router Advertisements werden von Routern automatisch in regelmäßigen Abständen verschickt, können aber auch durch eine *Router Solicitation* von anderen Nodes am Link ausgelöst werden.
Router Discovery	Mechanismus, um Hosts an einem Link über die Anwesenheit eines Routers zu informieren. Diese Informationen werden über *Router Advertisements* verteilt.
Router Solicitation	Eine Router Solicitation ist eine Nachricht der *NDP*-Familie, mit der ein Node (typischerweise ein Host) um ein *Router Advertisement* bittet.
RPF	*Reverse Path Forwarding*
RPL	Routing Protocol for Low-Power and Lossy Networks
RS	*Router Solicitation*
RSA	Public Key Kryptoverfahren nach Rivest, Shamir und Adleman.
RSVP	Resource ReSerVation Protocol
SAVI	*Source Address Validation Improvement*
SCTP	Stream Control Transmission Protocol
Secure Neighbor Discovery	Verfahren zur kryptografischen Absicherung der Neighbor Discovery, das von IPsec unabhängig ist.
SEND	*Secure Neighbor Discovery*

SHA-512	Secure Hash Algorithm mit 512 Bits Ausgabe nach FIPS 180-4.
SIP	Session Initialization Protocol
SLAAC	*Stateless Address Autoconfiguration*
SNMP	Simple Network Management Protocol
SSH	Secure Shell
Solicited-Node Multicast-Adresse	Die Multicast-Adresse, an die im Rahmen der Adressauflösung und der NUD Neighbor Solicitations gesendet werden. Die Solicited-Node Multicast-Adresse hat die Form `ff02::1:ffxx:xxxx`, wobei die letzten drei Bytes der Unicast-Adresse entnommen werden, zu der die Multicast-Adresse gebildet wird.
Source Address Validation Improvement	Methode zur Erkennung von Adressfälschungen
Stateless Address Autoconfiguration	Mechanismus zur zustandslosen Konfiguration von IPv6-Adressen an einem Interface. Hierzu wird aus einem Präfix und dem *EUI-64*-Identifier eine mutmaßlich eindeutige Adresse gebildet, deren tatsächliche Eindeutigkeit – und damit Nutzbarkeit – im Rahmen der *DAD* überprüft wird.
TCP	Transmission Control Protocol
Tentative Address	Unicast- oder Anycast-Adresse, deren Eindeutigkeitsprüfung am Link noch nicht erfolgreich abgeschlossen wurde.
THC	The Hacker's Choice
TLV-Codierung	Type-Length-Value-Codierung für Optionen in den Header-Erweiterungen Hop-By-Hop Options und Destination Options. Hierbei zeigt das erste Byte den Typ der Option und das zweite Byte die Länge der Option an; dem folgt dann der eigentliche Inhalt der Option. Auch für NDP-Optionen wird eine Type-Length-Value-Codierung genutzt, allerdings wird die Länge dort in Vielfachen von 8 Bytes angegeben anstatt in Bytes.
UDP	User Datagram Protocol
ULA	*Unique Local Address*
ULP	*Upper-Layer Protocol*
Unicast-Adresse	IP-Adresse, die einem Interface eindeutig zugeordnet ist.
Unique Local Address	Lokal routbare IP-Adresse, die jedoch nicht über Site-Grenzen hinweg geroutet werden darf.
Unspezifizierte Adresse	Die Adresse `::`, die lediglich im Rahmen der *DAD* als Absendeadresse verwendet werden darf.
Upper-Layer header	Header des Upper-Layer Protocol
Upper-Layer Protocol	Protokoll, das in einem oder mehreren IP-Paketen transportiert wird, üblicherweise eine ICMP-Nachricht, ein TCP-Segment oder ein UDP-Datagram; für spezielle Anwendungen aber auch andere.
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTM	Unified Threat Management

Vorläufige Adresse	Siehe *Tentative Address*
WLAN	Wireless Local Area Network
X.509	ITU-T Standard, der Datenformate für Public-Key Zertifikate, Attributzertifikate und Sperrlisten spezifiziert.

7 Literatur

[BVA 2013] C. Schmoll et al.: *IPv6 – Migrationsleitfaden für die öffentliche Verwaltung*. Bundesverwaltungsamt, 2013.

[Endres 2012] J. Endres: *IPv6 im (G)UI – Was ist eine IPv6-Adresse?* Vortragsfolien zum IPv6-Kongress 2012.

[Gont 2011a] F. Gont: *Hacking IPv6 Networks*. Vortragsfolien zu DEEPSEC 2011

[Gont 2011b] F. Gont: *IPv6 Router Advertisement Guard (RA-Guard) Evasion*, IETF Internet Draft, Juni 2011.

[Gont 2012a] F. Gont: *Security/Robustness Assessment of IPv6 Neighbor Discovery Implemenations*. SI6 Networks, November 2012.

[Gont 2012b] F. Gont: *Network Reconnaissance in IPv6 Networks*. IETF Internet Draft, Dezember 2012.

[Gont 2013a] F. Gont: *Security Assessment of Neighbor Discovery (ND) for IPv6*. IETF Internet Draft, Januar 2013.

[Gont 2013b] F. Gont: *Security Implications of IPv6 Options of Type 10xxxxxx*. IETF Internet Draft, Januar 2013.

[Hagen 2009] S. Hagen: *IPv6 – Grundlagen, Funktionalität, Integration*. 2. Auflage, Sunny Edition, 2009

[Heuse 2013] M. Heuse: *Pentesting IPv6 Networks*. Vortragsfolien zu einem Workshop im Rahmen des IPv6 Summit, Troopers 2013

[IEEE 2012] IEEE Standards Association: *Guidelines for 64-bit Global Identifier (EUI-64™)*. IEEE, 2012.

[ISACA 2012] J. Kalwerisky, *IPv6 Security Audit/Assurance Program*. ISACA, 2012.

[ISi-L-IPv6] ISi-Projektgruppe: *Leitfaden für eine sichere IPv6-Netzwerkarchitektur (ISi-L-IPv6) – Version 1.1*. In *BSI-Leitlinie zur Internet-Sicherheit (ISi-L)*, Bundesamt für Sicherheit in der Informationstechnik, 2012.

[ISi-LANA] ISi-Projektgruppe: *Sichere Anbindung von lokalen Netzen an das Internet (ISi-LANA) – Version 2.0*. In *BSI-Standards zur Internet-Sicherheit (ISi-S)*, Bundesamt für Sicherheit in der Informationstechnik, 2012.

[Malone 2008] D. Malone: *Observations of IPv6 Addresses*, Proceedings of *Passive and Active Measurement Conference – PAM 2008*. LNCS 4979, Springer Verlag 2008

[NIST SP 800-119] S. Frankel et al.: *Guidelines for the Secure Deployment of IPv6*. 2010, NIST Special Publication 800-119

[Potyraj 2007] C. A. Potyraj: *Firewall Design Considerations for IPv6*. National Security Agency Report # I733-041R-2007, 2007.

[Rafiee 2013] H. Rafiee, C. Meinel: *Router Advertisement based privacy extensions in IPv6 autoconfiguration*. Juni 2013.

[RFC 1918] Y. Rekhter et al.: *Address Allocation for Private Internets*. IETF Best Current Practice, Februar 1996.

[RFC 1981] J. McCann, S. Deering, J. Mogul: *Path MTU Discovery for IP version 6*. IETF Draft Standard, August 1996.

[RFC 2375] R. Hinden, S. Deering: *IPv6 Multicast Address Assignments*. IETF, Juli 1998.

[RFC 2460] S. Deering, R. Hinden: *Internet Protocol, Version 6 (IPv6) Specification*. IETF Draft Standard, Dezember 1998.

[RFC 2464] M. Crawford: *Transmission of IPv6 Packets over Ethernet Networks*. IETF Proposed standard, Dezember 1998.

[RFC 2473] A. Conta, S. Deering: *Generic Packet Tunneling in IPv6 Specification*. IETF, Proposed Standard, Dezember 1998.

[RFC 2675] D. Borman, S. Deering, R. Hinden: *IPv6 Jumbograms*. IETF Proposed Standard, August 1999.

[RFC 2711] C. Partridge, A. Jackson: *IPv6 Router Alert Option*. IETF Proposed Standard, Oktober 1999.

[RFC 2827] P. Ferguson, D. Senie: *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. IETF Best Current Practice, Mai 2000.

[RFC 2894] M. Crawford: *Router Renumbering for IPv6*. IETF Proposed Standard, August 2000.

[RFC 2993] T. Hain: *Architectural Implications of NAT*. IETF, November 2000.

[RFC 3122] A. Conta: *Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification*. IETF Proposed Standard, Juni 2001.

[RFC 3177] IAB, IESG: *IAB/IESG Recommendations on IPv6 Address Allocations to Sites*. IETF, März 2001. Obsoleted by RFC 6177.

[RFC 3315] R. Droms (Ed.) et al.: *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*. IETF Proposed Standard, Juli 2003.

[RFC 3489] J. Rosenberg et al.: *STUN – Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*. IETF Proposed Standard, März 2003. Obsoleted by RFC 5389.

[RFC 3513] R. Hinden, S. Deering: *Internet Protocol Version 6 (IPv6) Addressing Architecture*. IETF Proposed Standard, April 2003. Obsoleted by RFC 4291.

[RFC 3531] M. Blanchet: *A Flexible Method for Managing the Assignment of Bits of an IPv6 Address Block*. IETF, April 2003.

[RFC 3587] R. Hinden, S. Deering, E. Nordmark: *IPv6 Global Unicast Address Format*. IETF, August 2003.

[RFC 3596] S. Thomson et al.: *DNS Extensions to Support IP Version 6*. IETF Draft Standard, Oktober 2003.

[RFC 3704] F. Baker, P. Savola: *Ingress Filtering for Multihomed Networks*. IETF Best Current Practice, April 2004.

[RFC 3756] P. Nikander (Ed.), J. Kempf, E. Nordmark: *IPv6 Neighbor Discovery (ND) Trust Models and Threats*. IETF, Mai 2004.

[RFC 3849] G. Huston, A. Lord, P. Smith: *IPv6 Address Prefix Reserved for Documentation*. IETF, Juli 2004.

[RFC 3879] C. Huitema, B. Carpenter: *Deprecating Site Local Addresses*. IETF Proposed Standard, September 2004.

[RFC 3971] J. Arkko (Ed.): *SEcure Neighbor Discovery (SEND)*. IETF Proposed Standard, März 2005.

[RFC 3972] T. Aura: *Cryptographically Generated Addresses (CGA)*. IETF Proposed Standard, März 2005.

[RFC 3986] T. Berners-Lee, R. Fielding, L. Masinter: *Uniform Resource Identifier (URI): Generic Syntax*. IETF Internet Standard, Januar 2005.

[RFC 4007] S. Deering et al.: *IPv6 Scoped Address Architecture*. IETF Proposed Standard, März 2005.

[RFC 4038] M-K. Shin et al.: *Application Aspects of IPv6 Transition*. IETF, März 2005.

[RFC 4192] F. Baker, E. Lear, R. Droms: *Procedures for Renumbering an IPv6 Network without a Flag Day*. IETF, September 2005.

[RFC 4193] R. Hinden, B. Haberman: *Unique Local IPv6 Unicast Addresses*. IETF Proposed Standard, Oktober 2005.

[RFC 4291] R. Hinden, S. Deering: *IP Version 6 Addressing Architecture*. IETF Draft Standard, Februar 2006.

[RFC 4302] S. Kent: *IP Authentication Header*. IETF Proposed Standard, Dezember 2005.

[RFC 4303] S. Kent: *IP Encapsulating Payload (ESP)*. IETF Proposed Standard, Dezember 2005.

[RFC 4380] C. Huitema: *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)*. IETF Proposed Standard, Februar 2006.

[RFC 4429] N. Moore, *Optimistic Duplicate Address Detection (DAD) for IPv6*. IETF Proposed Standard, April 2006.

[RFC 4443] A. Conta, S. Deering, M. Gupta (Ed.): *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*. IETF Draft Standard, März 2006.

[RFC 4843] P. Nikander, J. Laganier, F. Dupont: *An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID)* . IETF, April 2007.

[RFC 4861] T. Narten, E. Nordmark, W. Simpson, H. Soliman: *Neighbor Discovery for IP Version 6 (IPv6)*. IETF Draft Standard, September 2007

[RFC 4862] S. Thomson, T. Narten, T. Jinmei: *IPv6 Stateless Address Autoconfiguration*. IETF Draft Standard, September 2007.

[RFC 4864] G. Van de Velde et al.: *Local Network Protection for IPv6*. IETF, Mai 2007.

[RFC 4890] E. Davies, J. Mohacsi: *Recommendations for Filtering ICMPv6 Messages in Firewalls*. IETF, Mai 2007.

[RFC 4941] T. Narten, R. Draves, S. Krishnan: *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*. IETF Draft Standard, September 2007.

[RFC 4942] E. Davies, S. Krishnan, P. Savola: *IPv6 Transition/Coexistence Security Considerations*. IETF, 2007.

[RFC 4943] S. Roy, A. Durand, J. Paugh: *IPv6 Neighbor Discovery On-Link Assumption Considered Harmful*. IETF, September 2007.

[RFC 5095] J. Abley, P. Savola, G. Neville-Neil: *Deprecation of Type 0 Routing Headers in IPv6*. IETF Proposed Standard, Dezember 2007.

[RFC 5157] T. Chown: *IPv6 Implications for Network Scanning*. IETF, März 2008.

[RFC 5375] G. Van de Velde et al.: *IPv6 Unicast Address Assignment Considerations*. IETF, Dezember 2008.

[RFC 5535] M. Bagnulo: *Hash-Based Addresses (HBA)*. IETF Proposed Standard, Juni 2009.

[RFC 5722] S. Krishnan: *Handling of Overlapping IPv6 Fragments*. IETF Proposed Standard, Dezember 2009.

[RFC 5942] H. Singh, W. Beebee, E. Nordmark: *IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes*. IETF Proposed Standard, Juli 2010.

[RFC 5952] S. Kawamura, M. Kawashima: *A Recommendation for IPv6 Address Text Representation*. IETF Proposed Standard, August 2010.

[RFC 5996] C. Kaufman et al.: *Internet Key Exchange Protocol Version 2 (IKEv2)*. IETF Proposed Standard, September 2010.

[RFC 6104] T. Chown, S. Venaas: *Rogue IPv6 Router Advertisement Problem Statement*. IETF, Februar 2011.

[RFC 6105] E. Levy-Abegnoli et al.: *IPv6 Router Advertisement Guard*. IETF, Februar 2011.

[RFC 6106] J. Jeong et al: *IPv6 Router Advertisement Options for DNS Configuration*. IETF Proposed Standard, November 2010.

[RFC 6177] T. Narten, G. Huston, L. Roberts: *IPv6 Address Assignment to End Sites*. IETF Best Current Practice, März 2011.

[RFC 6275] C. Perkins (Ed.), D. Johnson, J. Arkko: *Mobility Support in IPv6*. IETF Proposed Standard, Juli 2011.

[RFC 6434] E. Jankiewicz, J. Loughney, T. Narten: *IPv6 Node Requirements*. IETF, Dezember 2011.

[RFC 6437] S. Amante et al.: *IPv6 Flow Label Specification*. IETF Proposed Standard, November 2011.

[RFC 6554] J. Hui et al.: *An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)*. IETF Proposed Standard, März 2012.

[RFC 6564] S. Krishnan et al.: *A Uniform Format for IPv6 Extension Headers*. IETF Proposed Standard, April 2012.

[RFC 6620] E. Nordmark, M. Bagnulo, E. Levy-Abegnoli: *FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses*. IETF Proposed Standard, Mai 2012.

[RFC 6724] D. Thaler et al.: *Default Address Selection for Internet Protocol Version 6 (IPv6)*. IETF Proposed Standard, April 2012.

[RFC 6874] B. Carpenter, S. Cheshire, R. Hinden: *Representing IPv6 Zone Identifiers in Address Literals and Uniform Resource Identifiers*. IETF Proposed Standard, Februar 2013.

[RFC 6890] M. Cotton et al.: *Special-Purpose IP Address Registries*. IETF Best Current Practice, April 2013.

[RFC 6946] F. Gont: *Processing of IPv6 "atomic" fragments*. IETF Proposed Standard, Mai 2012.

[RFC 6959] D. McPherson, F. Baker, J. Halpern: *Source Address Validation Improvement (SAVI) Threat Scope*. IETF, Mai 2013.

[RFC 6980] F. Gont: *Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery*. IETF Proposed Standard, August 2013.

[RFC 7112] F. Gont, V. Manral, R. Bonica: *Implications of Oversized IPv6 Header Chains*. IETF Proposed Standard, Januar 2014.

[RIPE 554] M. Käo, J. Žorž, S. Steffan: *Requirements for IPv6 in ICT Equipment*, RIPE-554, 2012.

[Stockebrand 2010] B. Stockebrand: *IPv6 in Practice – A Unixer's Guide to the Next Generation Internet*. Springer Verlag, 2010.

[Ziring 2006] N. Ziring: *Router Security Configuration Guide Supplement – Security for IPv6 Routers*. NSA Report Number: I33-002R-06, 2006.