UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE MATEMATICHE,
FISICHE E NATURALI

Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche

Protocollo IP versione 6: vulnerabilità e attacchi

RELATORE
Prof. Marco Cremonini

TESI DI LAUREA DI
Davide Gerhard
Matr. 707705

Anno Accademico 2011-2012

Ai miei genitori,
che mi hanno sostenuto
in tutti questi anni.

Indice

Convenzioni

Nel documento verranno usate le seguenti convenzioni:

- si userà il termine *indirizzo di collegamento* per identificare l'indirizzo di livello 2 dello stack ISO/OSI. Nel caso si usasse una rete Ethernet quest'ultimo equivarrà all'indirizzo MAC;
- verranno usati i termini nodo, host e router come specificato nel documento RFC 2460 [28];
- in caso ci fossero nomi il cui significato è stato definito nella documentazione originale, si prenda per esempio il nome dei messaggi ICMPv6, questi verrano mantenuti tali quali e non verrà fatta alcuna traduzione;
- nell'elaborato si userà frequentemente la notazione IPv6 per indicare il protocollo IP versione 6 e IPv4 per indicare la corrispettiva versione 4. Tale convenzione è valida anche per tutti gli altri protocolli, come per esempio ICMPv6.

Introduzione

Il protocollo IP versione 6 si pone come successore della versione 4 tutt'ora implementata in Internet. L'obiettivo primario dello sviluppo fu la necessità di ampliare lo spazio di indirizzamento che, a causa dell'aumento esponenziale degli utenti e dei dispositivi ad esso collegati, si stava esaurendo. In questi ultimi anni, vista l'imminente fine degli indirizzi IPv4, è stata presa in seria considerazione la migrazione su larga scala ad IPv6. Diverse organizzazioni e diversi enti nazionali hanno già aggiornato le proprie infrastrutture ma è stato deciso di non raggiungere gli utenti finali o le piccole/medie imprese, salvo alcuni rari casi. Questa politica, se da un lato confina le problematiche della nuova implementazione a dipartimenti specializzati, dall'altro ne limita l'espansione ed una maggiore verifica da parte di un pubblico più ampio.
Visto quindi l'interesse dell'argomento e la necessità di rendere obbligatorio IPv6 anche per gli utenti finali nell'immediato futuro, ci si è posti l'obiettivo di analizzare da un lato le specifiche del protocollo IPv6, mettendone in luce le vulnerabilità, e dall'altro le possibili ricadute in fatto di sicurezza derivanti da implementazioni non corrette o comunque non ancora sufficientemente sviluppate per confrontarsi con IPv4. Si ritiene infatti che una maggiore conoscenza dello stesso ed una corretta implementazione possano minimizzare in maniera sostanziale gli attacchi che verranno qui presentati. Si vorrebbero quindi evitare quelle conseguenze, il più delle volte non derivabili direttamente dalla volontà degli amministratori di rete, che una migrazione frettolosa potrebbe comportare sugli aspetti di sicurezza.
Oltre a tale aspetto prettamente divulgativo sulla sicurezza del protocollo IP versione 6, l'elaborato si prefigge l'obiettivo di investigare quali siano i nuovi vettori di attacco oggi utilizzabili nelle reti IPv6 e quali siano le vulnerabilità presenti all'interno delle specifiche. Un inseme così completo delle vulnerabilità e degli attacchi ad IPv6 non è stato trovato in letteratura e quindi si presume che questo sia il primo documento di tale genere. Se questo è in parte dovuto alla modalità con cui si divulgano gli attacchi e le vulnerabilità, basata su documenti singoli e specifici, dall'altra si ha che l'ambito della sicurezza di IPv6 non è stato molto approfondito nell'ultimo decennio. Solo in questi ultimi anni ne è stata riconsiderata l'importanza viste soprattutto l'espansione e la priorità che avrà nel prossimo futuro. Ad oggi i maggiori ricercatori nell'ambito degli attacchi ad IPv6 sono soltanto due e i tool disponibili sono ben lungi dall'essere alla pari con quelli di IPv4, come si vedrà nel testo.

Nel primo capitolo si introdurranno brevemente le specifiche IPv6 inquadrando così l'ambiente in cui verranno esposti gli argomenti dei capitoli successivi.
Si inizierà col presentare le motivazioni e quali sono i limiti riscontrati durante l'espansione di IPv4 su scala mondiale. In particolare la necessità di uno spazio di indirizzamento maggiore a quello di IPv4 e una maggiore efficenza nella fase di routing.
Successivamente si passerà ad analizzare le specificità del header IPv6, per esempio l'introduzione di una dimensione fissa, e l'aggiunta degli Extension Header come soluzione alla dimensione dinamica delle opzioni di IPv4.
Altre novità introdotte nel protocollo IPv6 sono il supporto trasparente al Multihoming e la nuova versione di ICMP che, a differenza di IPv4, ha il compito di gestire e ricercare gli indirizzi nella rete. Si vedrà infatti quali sono i compiti di ICMPv6, come per esempio la modalità di autoconfigurazione (SLAAC) e la sostituzione del protocollo ARP, e quali sono le regole per una corretta politica di filtraggio atta a vincolare i messaggi ICMPv6 all'interno della rete.
Per facilitare la transizione al nuovo protocollo o per ovviare alla mancanza di connettività IPv6 si sono definite due modalità. La prima introdotta è stata la possibilità di gestire simultaneamente sia connessioni IPv4 sia IPv6 ed è stata chiamata Dual-Stack. La seconda prevede l'introduzione di un tunnel che incapsuli il traffico IPv6 da un host all'altro nel caso in cui i nodi intermedi non abbiano implementato IPv6; analogamente un tunnel può trasportare il traffico IPv4 tra due host non ancora aggiornati a IPv6, attraverso una rete intermedia già adeguata al nuovo protocollo. Tuttavia queste modalità di tunneling, come si vedrà, sono portatrici di diverse vulnerabilità, visto che un tunnel potrà essere usato come canale di evasione dei sistemi perimetrali.
In conclusione di capitolo verranno presentati quegli aspetti, sviluppati negli anni, atti a difendere il protocollo da eventuali attacchi o per proteggere l'utente da un eventuale tracciamento.

Nel secondo capitolo si affronteranno in dettaglio le tematiche di sicurezza e le vulnerabilità derivabili direttamente dalle specifiche.
Si iniziarà col presentare quali siano gli aspetti di maggior interesse nella sicurezza di IPv6. Per esempio la mancanza di apparati atti ad ispezionare correttamente i pacchetti IPv6 e la complessità introdotta dalla modalità Dual-Stack e dagli Extension Header. Si presenterà, inoltre, un grafico in cui si mostra l'andamento negli ultimi anni delle vulnerabilità sulle implementazioni IPv6. Come si potrà notare questo è in accordo con la riflessione proposta all'inizio dell'introduzione nella quale si sottolineava l'aumento da una parte delle vulnerabilità e dall'altra del maggiore interesse verso IPv6.
Il primo argomento che verrà affrontato sarà il Multihoming e verranno presentate le vulnerabilità che affliggono tale modalità, per esempio DoS e re-routing dei pacchetti, e quali siano le raccomandazioni per minimizzare l'impatto di tali attacchi.
Un aspetto molto importante in IPv6 è l'uso del Multicast visto che sostituisce completamente il broadcast presente in IPv4. Verrà fatta un'analisi sui tipi di indirizzo disponibili e come questi vengano suddivisi per *scope*. In particolare, si faranno esem-

pi su quali siano le possibili richieste attuabili con tali indirizzi e come possano essere usate per effettuare l'enumerazione dei servizi UDP nella rete locale.
Altri aspetti introdotti in IPv6 sono il campo di Quality of Service, *Flow Label*, e l'allargamento del campo *Traffic Class* già definito in IPv4. Si vedrà come questi possano favorire attacchi DoS o come possano essere usati come *covert channel*.
Si presenteranno in seguito le vulnerabilità e i cambiamenti introdotti con il protocollo DHCPv6 in cui non sarà più presente la possibilità di esaurire lo spazio di indirizzamento disponibile ma si potrà incorrere in attacchi Denial of Service o di impersonificazione.
Conclusa tale spiegazione ci si addentrerà nel secondo argomento più importante di IPv6: i tunnel. Verranno discussi i cinque principali tunnel: 6over4, 6to4 e 6rd, ISATAP, Teredo e i Tunnel Broker. Verranno evidenziati gli aspetti di sicurezza e vulnerabilità presenti in ciascuno di essi. In particolare si darà maggiore attenzione ai quei protocolli, come Teredo, che permettono una facile implementazione anche in architetture NAT e che quindi facilitino l'evasione delle politiche perimetrali.
In conclusione di capitolo si affronterà un aspetto tutt'ora privo di soluzioni: la gestione dello Spam su protocollo IPv6. In particolare si disquisirà sui limiti delle odierne metodologie, in primis DNSBL, e attraverso dei grafici pubblicati dal RIPE nel 2010 si vedranno quali siano i volumi di traffico generati.

Il terzo capitolo presenterà gli attacchi perpetrabili al protocollo IPv6 che sfruttano non solo le vulnerabilità viste nel capitolo precedente ma anche le implementazioni non corrette. In particolare si affronteranno quegli attacchi che portano ad una condizione di Denial of Service o di man-in-the-middle.
Il capitolo si aprirà con una breve discussione sulle opportunità d'attacco che il nuovo protocollo pone all'eventuale utente malevolo. Si vedrà come uno degli aspetti più importanti sia la mancanza di un corretto controllo degli accessi e del traffico IPv6 da parte dei dispositivi e degli amministratori e come molti apparati di sicurezza in realtà lo supportino solo in parte e non sempre correttamente. In aggiunta agli attacchi proposti bisogna sempre considerare le singole vulnerabilità presenti negli stack dei vari Sistemi Operativi ed essendo molto specifici non verrano trattati in questa tesi.
Il primo attacco proposto è quello relativo al campo Hop Limit. Si vedrà come questo possa essere usato, in parte come già avviene per IPv4, per rilevare la tipologia del Sistema Operativo o per rilevarne la posizione all'interno dell'infrastruttura. Questi risultati permettono ad un eventuale attaccante di avere delle informazioni di base sulla vittima e quindi focalizzare i propri sforzi verso una direzione piuttosto che un'altra.
L'argomento successivo è l'uso della frammentazione per effettuare attacchi di idle-scanning e di DoS. Verranno presentati quali Sistemi Operativi siano vulnerabili a questo tipo di problematica e in cosa differisca la frammentazione in IPv6 rispetto a quella in IPv4. Ad esempio possiamo trovare i problemi derivanti dall'*overlapping* e dai frammenti atomici. Saranno proposti degli esempi sul funzionamento dell'attacco idle-scanning e come questo sia visualizzabile con il comando tcpdump. Altro aspetto

che verrà presentato riguarda la possibilità, attraverso la frammentazione di un messaggio su più pacchetti, di nascondere gli eventuali Extension Header o le eventuali informazioni in esso contenute.
L'attacco seguente, chiamato Type 0 Routing Header, è molto simile a quello presente in IPv4 e permette di specificare una serie di nodi intermedi attraverso i quali il pacchetto dovrà transitare. Questo, come avviene già in IPv4, può essere usato per evadere protezioni perimetrali o per verificare i percorsi attivi che un pacchetto può percorrere. Verrà presentata anche una tabella in cui saranno rimarcati i Sistemi Operativi che supportano ancora tale opzione, essendo stata resa obsoleta alcuni anni fa. Si vedranno inoltre delle tecniche atte ad effettuare il Network Discovery, il controllo dei filtri ingress sui firewall, l'evasione delle regole imposte dai sistemi di sicurezza perimetrale e come sia possibile, con l'uso del messaggio Type 0 Routing, effettuare un attacco DoS attraverso l'effetto ping-pong.
Concluso tale paragrafo si inizieranno ad investigare gli attacchi al protocollo di Neighbor Discovery. Si vedrà come funziona in dettaglio il protocollo per la risoluzione degli indirizzi di livello 2 e come sia possibile recuperare i parametri della rete locale attraverso il messaggio di Neighbor Advertisement. Come avviene già per IPv4 anche il processo di risoluzione si avvale di una cache, chiamata *Neighbor Cache*, che sarà oggetto del primo attacco presentato in questo paragrafo. In particolare si vedrà come attuare l'attacco di *Neighbor Cache Poisoning*, la pubblicazione di un indirizzo speciale di link-layer e la possbilità, ove i Sistemi Operativi mal gestiscano la Neighbor Cache, di esaurire la cache e quindi portare il sistema in una situazione di blocco.
Il paragrafo successivo tratterà della modalità SLAAC (autoconfigurazione) e vedremo in dettaglio come funzioni e come sia possibile perpetrare i seguenti attacchi: disabilitazione del router esistente, uso del messaggio *Duplicate Address Detection* (DAD) per effettuare un attacco Denial of Service, pubblicazione di parametri di rete diversi da quelli predefiniti, Router Advertisement flooding e abilitazione della modalità Dual-Stack, che permette quindi una più facile scansione del nodo.
Il paragrafo seguente presenterà l'attacco facente uso del messaggio Router Advertisement e come questa vulnerabilità sia stata affrontata dai produttori di dispositivi dedicati alla sicurezza. Tale difesa è stata chiamata RA-Guard e nella forma più semplice funziona pressapoco come il *DHCP Snooping* previsto in IPv4. Verranno quindi illustrati gli attacchi atti ad evadere tale difesa. Le tecniche presentate si basano su un uso ingegnoso della frammentazione e della possibilità infinita di espandere la catena degli Extension Header.
Il penultimo paragrafo sarà dedicato ad uno degli aspetti più innovativi degli ultimi due anni: la ricerca degli indirizzi IPv6 attivi. Lo spazio di indirizzamento di IPv6 è pari a 2^{128} e lo spazio di una sotto rete è pari a 2^{64}, ben lontano dal 2^{32} di IPv4. Questo implica che la ricerca esaustiva presente in IPv4 diventi non praticabile in IPv6. Si vedranno, quindi, delle tecniche che permettono ad un eventuale attaccante di individuare, con una buona approssimazione, i nodi attivi nella rete in cui si trova. In particolare verranno affrontate tecniche dedite alla ricerca degli indirizzi nella rete locale, piuttosto che

in Internet. Tutti questi metodi, come si vedrà nel testo, saranno usati in maniera aggregata permettendo così di ottenere risultati molto accurati. Saranno presentati anche dati statistici sull'efficacia di tali tecniche.
In conclusione di capitolo verranno presentati alcuni attacchi che per loro natura non hanno trovato collocazione nei paragrafi precedenti. Troviamo i messaggi di *Node Information Query/Response* che ci permettono di ricavare alcune informazioni dai nodi che hanno attiva tale funzione. Si vedrà infine la possibilità di un uso malevolo del messaggio ICMPv6 Redirect e come sia possibile sfruttare il protocollo *Multicast Listener Discovery* per un attacco Denial of Service.

Nel quarto ed ultimo capitolo verranno presentati brevemente gli aspetti di ricerca che sono stati affrontati solo in parte nell'elaborato e che andrebbero ulteriormente approfonditi. Si presenteranno una pubblicazione inerente lo sviluppo e l'evoluzione del Malware in reti IPv6 e in reti miste e la necessità di miglioramenti nelle tecniche di ricerca dell'indirizzo. Si vedrà anche come sia necessario un ulteriore approfondimento sugli aspetti di IPv6 Mobile e su come siano necessari ulteriori tool per verificare la sicurezza di una rete IPv6.

Al termine della lettura di questa tesi, il lettore dovrebbe essersi reso conto di come il protocollo IPv6 non sia per niente più sicuro della controparte IPv4, come ben spesso si vuol far credere; che molte delle vulnerabilità devono ancora essere scoperte e che una corretta conoscenza e progettazione dell'infrastruttura può evitare gran parte degli errori futuri e soprattutto quelli già visti con IPv4.

Prerequisito
Nell'elaborato si dà per scontata la conoscenza del protocollo IPv4 e delle vulnerabilità ad esso associate.

Capitolo 1

Introduzione all'IPv6

Il protocollo IPv6 nasce dalla necessità di supplire alla carenza di indirizzi IPv4 causata dall'aumento considerevole di apparati connessi alla rete Internet durante gli ultimi decenni. Si pensi che IPv4 era stato progettato come evoluzione di un progetto militare e si pensava che l'utilizzo sarebbe rimasto confinato all'interno di università e centri di ricerca e che non avrebbe mai sconfinato nella vita di tutti i giorni, attraverso, per esempio, i telefoni cellulari e le applicazioni di domotica. All'epoca si pensò, che 32 bit fossero più che sufficienti per descrivere l'identificativo di ogni nodo e che potesse supplire ai futuri decenni di espansione che il progetto avrebbe avuto.
Ben presto si comprese che l'esaurimento di tale spazio di indirizzamento non era poi così lontano come ci si era prefigurati e si pensò quindi di procedere con la definizione di uno nuovo standard, successivamente chiamato IPv6, che supplisse sia a questa problematica sia ai problemi, che all'aumentare della dimensione, IPv4 aveva presentato. Si passò quindi da un indirizzamento lungo 32 bit ad uno lungo 128 bit che permise di avere una spazio di indirizzamento davvero grande, si pensi che tale spazio è pari a circa 2×10^{28} indirizzi per ogni abitante della Terra. Inoltre, si decise di semplificare il processo di routing, di migliorare la gestione del Quality of Service (QoS), di evitare l'uso del NAT[1] e di eliminare l'indirizzo di broadcast e usare al suo posto indirizzi multicast.
In questo capitolo verranno introdotti tutti questi ed altri aspetti presenti negli RFC[2] inerenti ad IPv6. In particolare, il primo paragrafo tratterà della nascità di IPv6 e quali siano i documenti che ne definiscono le specifiche. Il secondo verterà sul header IPv6 e sulle sue possibili estensioni. Il terzo sull'indirizzamento, il quarto sul Multihoming, il quinto su ICMPv6 e sulla modalità di assegnazione dell'indirizzo, il quinto sulle differenze tra IPv4 e IPv6, il sesto sulle modalità dual stack, il settimo sui tunnel e l'ottavo sulle modalità di sicurezza introdotte da IPv6. Per concludere si esporranno alcune considerazioni relative alla sicurezza di IPv6 basate sulla semplice analisi degli aspetti precedentemente illustrati.

[1] Network address translation
[2] Request for Comments - `http://www.ietf.org/rfc.html`

1.1 Da IPv4 a IPv6

Per comprendere la necessità di IPv6 bisogna pensare che IPv4 è stato progettato oltre 30 anni fa quando i computer erano relegati solo a piccoli gruppi di persone (principalmente università ed enti di ricerca) e quando l'infrastruttura era di piccole dimensioni; non si pensò, quindi, che Internet potesse svilupparsi come oggigiorno possiamo tutti constatare.
Per ovviare a tale esaurimento e privi di un'alternativa immediata, si svilupparono negli anni diverse tecniche: si decise di imporre distinzione tra indirizzi pubblici e privati (RFC 1918), di definire una tecnica per cui gli indirizzi privati potessero accedere ad internet (NAT[3]) e un modo per gestire gli indirizzi senza definirne la classe (CIDR[4]). Nel frattempo si capì che tale soluzione era solo temporanea e quindi si decise di creare un gruppo di lavoro che potesse supplire sia a tali restrizioni sia ad altre carenze riscontrate in fase di routing e di aumento del traffico IP. A questo proposito IETF[5], a metà degli anni novanta, decise di formare un gruppo di lavoro per la definizione del nuovo protocollo. Nel 1996 venne pubblicata la prima proposta del nuovo protocollo (RFC 1883), che poi divenne definitiva con la pubblicazione del RFC 2460 [28]. A questa ne seguirono molte altre che ne definirono gli aspetti implementativi, di gestione degli indirizzi, della parte ICMPv6 e delle procedure di routing. Tra le tante novità introdotte con il nuovo protocollo troviamo: il nuovo spazio di indirizzamento, header a dimensione fissa, l'autoconfigurazione degli indirizzi, l'integrazione di IPsec nel protocollo, una migliore gestione del QoS, un insieme di protocolli per la gestione della mobilità, una migliore gestione durante la fase di trasmissione del pacchetto ed eliminazione della distinzione tra IP privati e pubblici. Quest'ultimo aspetto, in particolare, responsabilizza nuovamente i nodi estremi (end-point), i quali saranno i soli a poter frammentare un pacchetto o a gestire gli header estesi.
Si inizierà col delineare la struttura del header per poi approfondire man mano gli aspetti principali che ci serviranno per comprendere meglio i tipi di attacco e di vulnerabilità che andremo ad analizzare nei prossimi capitoli.

1.2 Header IPv6

La figura 1.1 presenta la struttura a dimensione fissa (40 byte) dell'header di IPv6. Si optò per una dimensione fissa, a differenza di quella usata in IPv4, perchè negli anni si notò che tale variabilità aggiungeva notevole complessità agli algoritmi presenti nei router e che col crescere della velocità, che in quegli anni stava già iniziando a manifestarsi, tale complessità si sarebbe ben presto tramutata in un problema rilevante limitando l'ampiezza di banda trasmissibile.

[3]Network Address Translation - RFC 3022
[4]Classless inter-domain routing - RFC 4632
[5]Internet Engineering Task Force

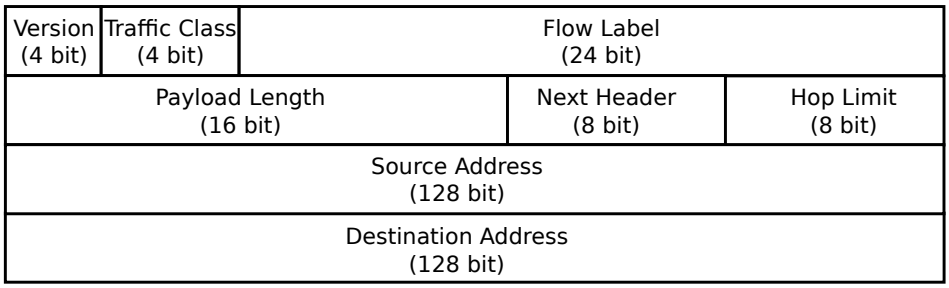


Figura 1.1: Header IPv6

Vediamo in breve i campi presenti e la loro funzione.

- **Version** (4 bit): Identifica la versione del protocollo (in binario 6 e' espresso con il valore 0110).

- **Traffic Class** (8 bit): Identifica la classe di traffico così come è identificata nel campo TOS dell'IPv4. L'RFC 2474 specifica i diversi valori che possono essere usati.

- **Flow Label** (20 bit): Identifica un flusso di traffico e viene usato nelle procedure di QoS[6].

- **Payload Length** (16 bit): Identifica la lunghezza dei dati presenti dopo l'header IPv6. Il valore è rappresentato senza segno quindi la lunghezza massima del pacchetto potrà essere di $65535 + 40 = 65575$ byte. Questo limite può essere esteso usando l'opzione Jumbogram Hop-by-Hop.

- **Next Header** (8 bit): Contiene il codice identificativo del protocollo successivo. Quest'ultimo potrà essere un header esteso o il codice del protocollo usato a livello superiore (per esempio TCP, UDP).

- **Hop Limit** (8 bit): Identifica il numero di router massimi che il pacchetto potrà attraversare, come specificato nella seconda versione del campo TTL di IPv4. Può essere anche usato come filtro aggiuntivo per attacchi provenienti da reti non fidate, per esempio da Internet (GTSM[7]).

- **Source Address** (128 bit): Identifica l'IP unicast del pacchetto sorgente.

- **Destination Address** (128 bit): Identifica l'IP di destinazione. Può essere unicast, multicast o anycast. Vedremo meglio in seguito cosa rappresentino questi valori.

[6]Quality of Service
[7]Generalized TTL Security Mechanism

1.2.1 Extension Header

Parte peculiare di IPv6 è la possibilità di aggiungere delle estensioni al header originale. Queste possono essere usate per aggiungere un livello di sicurezza attraverso IPsec o per instradare il pacchetto attraverso uno specifico percorso; in IPv4 questa procedura è chiamata Source Routing.

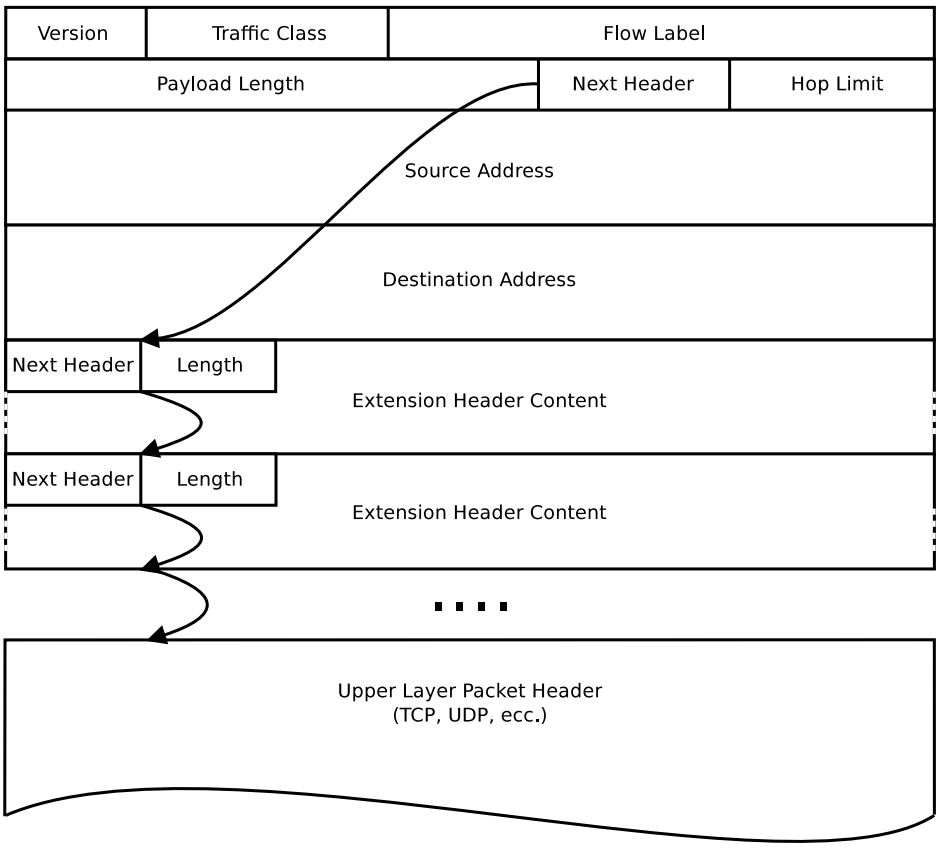


Figura 1.2: IPv6 Extension Header

In figura 1.2 vediamo come sia possibile innestare i diversi header e come questi vengano elaborati in successione per poi raggiungere il protocollo di livello superiore, per esempio TCP o UDP. Possiamo, quindi, avere estensioni concatenate l'una all'altra e che ci permetto di definire molteplici configurazioni attraverso l'uso di un solo pacchetto. Tale caratteristica usa il campo *Next Header* per identificare quale sia il protocollo successivo e il campo *Length* per definirne la lunghezza. Di norma tali estensioni vengono ispezionate soltanto dal ricevente e quindi non si ha un degradamento delle prestazioni nei nodi intermedi mantenendo valida l'ottimizzazione ottenuta attraverso l'uso di una dimensione statica del header. Unica eccezione avviene con l'estensione Hop-by-Hop la quale dev'essere ispezionata da ogni nodo intermedio così da permetterne un corretto instradamento. La definizione dei diversi tipi è già presente nella

prima specifica pubblica di IPv6 (RFC 2460) e si compone di sette estensioni; vediamo in breve quali essi siano e quale sia la loro funzione.

- **Hop-by-Hop Option** (valore 0): Permette di definire delle opzioni che devono essere esaminate da ogni nodo. Permette, per esempio, di usare l'opzione Jumbogram che consente il trasporto di pacchetti di dimensione maggiore dello standard. In IPv6 il pacchetto minimo è pari a 1280 byte ma è strettamente consigliato l'uso di un pacchetto con lunghezza pari a 1500 byte.

- **Routing Header** (valore 43): Permette al trasmettitore di definire il percorso che il pacchetto deve obbligatoriamente seguire. Il tipo 0 è stato identificato come portatore di diverse vulnerabilità e quindi è stato eliminato dall'IETF (RFC 5095). Il tipo 2 (usato dalla parte mobile di IPv6) è stato mantenuto.

- **Fragment Header** (valore 44): Viene usato dal nodo sorgente per poter inviare pacchetti più grandi del Path Maximum Transmission Unit (PMTU) al nodo destinatario. Ogni frammento deve avere lo stesso identificativo e lo stesso identico indirizzo sorgente e di destinazione. Può essere usato soltanto dal nodo sorgente e non dai nodi intermedi. Bisogna sempre tenere presente che, a differenza di IPv4, in IPv6 non è concesso al router di frammentare un pacchetto.

- **Authentication Header** (valore 51): Permette di assicurare l'integrità e la correttezza dell'indirizzo di origine di un pacchetto IP. Questo tipo di sistema permette la protezione da attacchi di tipo replay. È una delle due modalità di funzionamento di IPsec integrato in IPv6.

- **Encapsulating Security Payload Header** (valore 50): Questa estensione porta all'interno di IPv6 l'altra modalità operativa di IPsec. Permette di autenticare l'origine, assicurarsi dell'integrità del pacchetto e di crittografare i dati (confidenzialità). Normalmente chiamato ESP.

- **Destination Options Header** (valore 60): Header che viene usato per trasportare delle opzioni che devono essere analizzate soltanto dal nodo destinatario.

- **Mobility Header** (valore 135): Viene usato soltanto da un'infrastruttura di tipo mobile.

1.3 Indirizzi IPv6

Come detto in precedenza uno dei motivi fondamentali per cui è nato il nuovo protocollo è stata la necessità di avere maggiori indirizzi a disposizione. Vediamo brevemente la struttura base dell'indirizzo IPv6 e come esso venga suddiviso. Una trattazione completa è presente nel documento RFC 4191.
In figura 1.3 è rappresentata la divisione a blocchi dell'indirizzo IPv6 in cui si identificano tre parti:

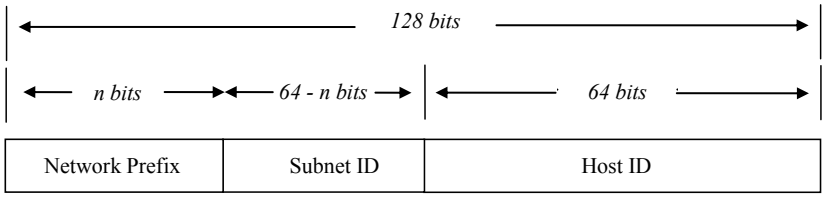


Figura 1.3: Indirizzo IPv6

- **Network Prefix** (lunghezza n): identifica il sito e/o l'ISP di appartenenza.
- **Subnet ID** (lunghezza 64-n): identifica una sotto rete.
- **Host ID** (64 bit): identifica un'interfaccia di rete.

La notazione IPv6 è rappresentata da caratteri esadecimali raggruppati a blocchi di 16 bit (4 cifre esadecimali) separati dal carattere ':'. Nel caso fossero presenti blocchi di quattro 0 si può ricorrere all'omissione degli stessi. Come si è visto dalla figura precedente, la dimensione dei primi due campi è variabile ed è dipendente dalla tipologia di indirizzo usato. Vediamo quindi quali siano le tipologie di indirizzo messe a disposizione dal protocollo IPv6.

Tipologia Indirizzo	**Indirizzo IPv6**	**Uso**
Indirizzo IPv4	::FFFF/96	Prefisso per gli indirizzi IPv4 all'interno di IPv6
Loopback	::1/128	Indirizzo di loopback (definito nel RFC 2460)
Global unicast	2000::/3	Indirizzi globali unicast e anycast allocati (definiti nel RFC 4291)
Global unicast	4000::/2 - FC00::/9	Indirizzi globali unicast e anycast (non allocati)
Nonroutable	2001:DB8::/32	Non instradabili. Ad uso documentativo (RFC 3849)
6Bone	3FFE::/16	Deprecato (RFC 3701)
Link-local unicast	FE80::/10	Spazio di indirizzamento per il collegamento locale
Riservato	FEC0::/10	Deprecato. Usato una volta come indirizzo site-local unicast e anycast
Local IPv6 address	FC00::/7	Spazio di indirizzamento usato nella rete locale unicast e anycast
Multicast	FF00::/8	Spazio di indirizzamento multicast

Tabella 1.1: Tipi di indirizzo IPv6

La tabella 1.1 espone quali siano questi tipi e quali caratteristiche di prefisso e di notazione abbiano. In particolare, vediamo i tre gruppi di indirizzi definiti dal protocollo.

- **Unicast Addresses**: Indirizzo che identifica un'interfaccia di un nodo.

- **Multicast Address**: Identifica gli indirizzi di multicast. Si tenga presente che in IPv6 l'indirizzo di broadcast è stato soppiantato dal multicast ed è stato eliminato il protocollo ARP[8] per l'individuazione dell'indirizzo MAC e al suo posto è stata implementata la procedura di Neighbor Discovery (ND). Per la ricerca del router, invece, viene usato il Router Discovery (RD). Un esempio del loro funzionamento verrà illustrato in seguito.

- **Anycast Addresses**: Indirizzo che permette di identificare diverse interfacce di uno o più nodi.

Come si è visto non è più presente alcuna distinzione tra indirizzi pubblici e privati, presente invece in IPv4. Così facendo ogni interfaccia sarà dotata di uno o più indirizzi pubblici (multi-homed) che in base alle regole configurate sui sistemi di sicurezza perimetrale sarà più o meno raggiungibile da Internet. Oltre a questi, sono stati definiti alcuni indirizzi ad uso speciale che dovranno essere presenti solo nella rete locale, quindi non propagati, e che avranno il compito di comunicare tra interfacce di host diversi prive di indirizzo pubblico o usati durante la fase di gestione della rete. Vediamo i più importanti.

- **Interface-local**: Indirizzo di loopback. Definito con l'indirizzo ::1.

- **Link-local**: indirizzo presente in ogni interfaccia e che permette funzioni amministrative nella rete locale (neighbor e router discovery). Questo indirizzo non dev'essere mai inviato in Internet e deve sempre rimanere confinato nella rete locale.

- **Unique local unicast**: Indirizzo usato all'interno di un'azienda o di un'università. L'uso di tale indirizzo non è molto diffuso.

- **Global**: indirizzo ad uso globale; unico su tutta la rete Internet.

- **Embedded IPv4 Unicast**: Indirizzo IPv6 che incorpora un'indirizzo IPv4 al suo interno. Non molto usato e deprecato dal RFC 4291.

- **Unspecified Address**: Indirizzo composto da tutti zeri (::) e normalmente usato come indirizzo sorgente durante una richiesta DHCPv6.

[8] Address Resolution Protocol

1.4 Multihoming

Una delle novità introdotte da IPv6 è sicuramente il pieno supporto al Multihoming. Questo significa avere la possibilità di usare più di una connessione verso Internet. Come visibile in figura 1.4 ogni host avrà più di un indirizzo per interfaccia o potrà avere

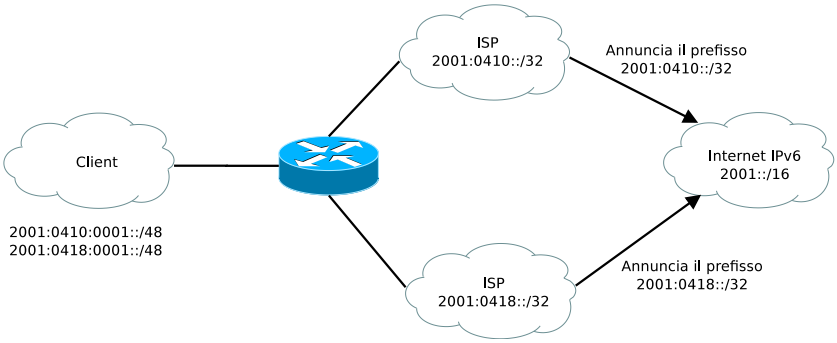


Figura 1.4: Esempio di infrastruttura Multihoming

due interfacce simultaneamente connesse ciascuna con un indirizzo diverso. Un'altra possibilità è nell'avere una singola connessione verso il router locale il quale avrà più di una connessione verso Internet. Come si può immaginare il Multihoming è molto utile, per esempio permette una maggiore resistenza a problemi riguardanti il singolo operatore, ma potrebbe avere un notevole impatto sull'architettura Internet, in particolare sulla grandezza delle tabelle di routing e forwarding. Di conseguenza, il Multihoming dovrà soddisfare i seguenti obiettivi:

- l'implementazione Multihoming dovrà essere trasparente ai protocolli di livello superiore. In caso contrario, la soluzione sarebbe pari a sostituire manualmente gli indirizzi e l'ISP principale;

- evitare l'aumento esponenziale delle tabelle globali di routing e di forwarding.

Da questi obiettivi sono nate diverse proposte le quali sono state raccolte nel documento RFC 4177. Nel tempo, la soluzione che ha preso maggiormente piede è quella che introduce la divisione dell'indirizzo IPv6 in una parte chiamate *identifier* e una chiamata *locator*. L'idea di base è che i protocolli di livello superiore usino l'identificatore mentre i router usino la parte di localizzazione. Per realizzare la gestione dinamica del legame tra i due valori si è dovuto definire un nuovo protocollo, chiamato Site Multihoming by IPv6 Intermediation (SHIM6) e definito nel documento RFC 5533.

Type (8 bit)	Code (8 bit)	Checksum (16 bit)
Message Body (n bit)		

Tabella 1.2: Formato del messaggio ICMPv6

1.5 ICMPv6

Le specifiche IPv6 ridefinisco l'ICMP[9] di IPv4 con un insieme di aggiunte e di modifiche. Il nuovo protocollo, definito nel RFC 4443 e chiamato ICMPv6, ingloba la parte di errore e di diagnostica della rete, già presente in IPv4, e ne estende le potenzialità aggiungendo le funzioni di amministrazione, come *Neighbor Discovery* (ND) e *MTU discovery*. Il primo, definito nel RFC 4861, verrà discusso nel prossimo paragrafo, mentre quì di seguito vedremo la struttura del messaggio ICMPv6 e le differenze con la versione di IPv4. Per ultimo, vedremo delle raccomandazioni per una gestione corretta di ICMPv6 nelle regole di firewalling, perchè essendo quest'ultimo attore principale nel funzionamento della rete non è possibile pensare di bloccarlo con le stesse modalità di IPv4.

1.5.1 Messaggio ICMPv6

Essendo una parte fondamentale nel funzionamento del protocollo IPv6 non è possibile esonerasi da una visione completa della struttura del protocollo ICMPv6. L'intento, quindi, è quello di presentare al lettore i fondamenti del protocollo ICMPv6 nelle sue parti: formato del messaggio, gestione e codici di errore e di diagnostica. Iniziamo col vedere il formato del messaggio. Questo sarà accodato all'header IPv6 ed agli eventuali Extension Header e avrà come codice identificativo (Next Header) il valore 58.
Il messaggio, illustrato nella tabella 1.2, avrà i seguenti campi:

- **Type**: indica il tipo di messaggio. I valori usabili sono definiti nelle tabelle 1.3 e 1.4.
- **Code**: dipende dal tipo di messaggio. È usato per creare una maggiore granularità nella tipologia del messaggio.
- **Checksum**: usato per determinare se il messaggio ICMPv6 e una parte dell'header IPv6 siano stati danneggiati durante la trasmissione. Si tenga presente che l'header IPv6 non ha nessun meccanismo per rilevare l'alterazione dello stesso e/o del payload e tale controllo viene demandato ai protocolli di livello superiore.
- **Message Body**: di lunghezza variabile e dipendente dal tipo e dal codice del messaggio. I messaggi ICMPv6 sono raggruppati in due classi: i messaggi di errore, tabella 1.3, e i messaggi informativi, tabella 1.4.

[9] Internet Control Message Protocol

Tipo	Descrizione	Codice
1	Destination Unreachable	0 = Nessun instradamento per la destinazione 1 = Comunicazione con la destinazione proibita 2 = Oltre lo *scope* dell'indirizzo sorgente 3 = Indirizzo irraggiungibile 4 = Porta irraggiungibile 5 = Indirizzo sorgente fallito; politica ingress/egress 6 = Instradamento rifiutato per la destinazione
2	Packet Too Big	Settato a zero dall'autore e ignorato dal ricevitore
3	Time Exceeded	0 = Superato il limite di nodi intermedi transitati 1 = Tempo di riassemblamento della frammentazione superato
4	Parameter Problem	0 = Errore in qualche campo del header 1 = Valore del campo Next Header non riconosciuto
100 e 101	Sperimentazioni Private	RFC 4443
127	Riservato per espansioni future	RFC 4443

Tabella 1.3: Tipi di errore ICMPv6 e relativi codici

1.5.2 Differenze tra ICMPv4 e ICMPv6

Come già accennato, esistono molteplici differenze tra le specifiche di ICMP di IPv6 e IPv4. In particolare, le più importanti sono:

- **Next Header Value**: il codice che identifica il messaggio passa da 1 di IPv4 a 58 di IPv6;

- **Neighbor Discovery (ND) sostituisce ARP**: nuova funzionalità che sostituisce il protocollo ARP e ne estende le potenzialità. Permette, in pratica, di migliorare la consegna dei messaggi in caso ci siano fallimenti da parte del router o di interfacce che cambiano il loro indirizzo di rete. Evita quindi i problemi presenti con l'uso della cache ARP. Per il suo funzionamento usa l'indirizzo di link-local. Verrà approfondito nel prossimo paragrafo;

Tipo	Descrizione	Codice
128 129	Echo Request Eco Replay	RFC 4443. Usato dal comando ping
130 131 132	Multicast Listener Query Multicast Listener Report Multicast Listener Done	RFC 2710. Usato per la gestione del gruppo multicast
133 134 135 136 137	Router Solicitation Router Advertisement Neighbor Solicitation Neighbor Advertisement Redirect Message	RFC 4861. Usato per la fase di Neighbor Discovery e di auto-configurazione
200 e 201	Sperimentazioni Private	RFC 4443
255	Riservato per espansioni future	RFC 4443

Tabella 1.4: Tipi di informazione ICMPv6 e relativi codici

- **eliminazione della frammentazione "in transito"**: viene eliminata la possibilità da parte dei router di frammentare i pacchetti;
- **Multicast Listener Discovery (MLD)**: insieme di tre messaggi equivalenti a quelli presenti in IGMP nella versione 2 di IPv4. Usati per gestire l'appartenenza di sotto reti ad un gruppo multicast.

1.5.3 Neighbor Discovery

Neighbor Discovery, descritto nel documento RFC 4861, è un processo per cui un nodo IPv6 può richiedere importanti informazioni sull'indirizzo di rete nel segmento in cui è collegato. In questo modo, ND rimpiazza a tutti gli effetti il protocollo ARP di IPv4. In questa sezione vedremo quali siano le funzionalità di tale processo e quali siano i messaggi che vengono usati per portarlo a compimento.
Vediamo, quindi, quali sono gli scopi del Neighbor Discovery Protocol (NDP) usato da tutti i nodi, siano essi host o router, in un segmento di rete:

- per l'autoconfigurazione (SLAAC) dell'indirizzo IPv6. Discusso nel prossimo paragrafo;
- per determinare il prefisso della rete e altre configurazioni;
- per determinare la presenza di indirizzi duplicati;

- per determinare l'indirizzo di rete di livello due (in Ethernet è chiamato MAC[10]);
- per determinare i router che possono inviare i propri pacchetti oltre il segmento di rete a cui appartengono;
- per tener traccia di quali vicini siano disponibili e quali non lo siano più. (Neighbor Unreachability Detection - NUD);
- per determinare i cambiamenti dell'indirizzo di livello due.

Per portare a termine tali operazioni il processo usa cinque differenti messaggi di ICMPv6. Vediamoli in breve.

- **Router Solicitation (RS)**: Quando un'interfaccia viene abilitata, il sistema invia un messaggio RS al router chiedendogli di generare immediatamente un messaggio RA e non attendere il prossimo istante in cui era previsto l'invio automatico.
- **Router Advertisement (RA)**: Il router avverte della sua presenza nella rete locale in modo periodico o in risposta ad un messaggio RS. Il messaggio contiene il prefisso e l'indirizzo, il valore di hop suggerito e il valore massimo di trasmissione supportato (MTU).
- **Neighbor Solicitation (NS)**: Il nodo invia un messaggio NS per determinare l'indirizzo di un vicino o per determinare se tale vicino è ancora raggiungibile attraverso l'indirizzo di collegamento precedente.
- **Neighbor Advertisement (NA)**: Risposta ad un messaggio NS. Il nodo può anche inviare un messaggio NA per avvisare che il suo indirizzo di collegamento è cambiato.
- **Redirect Message**: Usato dai router per informare i nodi che esiste un tragitto migliore per raggiungere una destinazione.

Vediamo ora come questi pacchetti siano usati per risolvere un indirizzo IPv6 trovando il corrispondente indirizzo MAC di un nodo presente nella stessa rete. Si tenga presente che in questo esempio verrà usato un segmento di tipo Ethernet[11].
La figura 1.5 rappresenta tale procedura.
Il nodo A manda un messaggio NS ad un indirizzo multicast con il quale richiede l'indirizzo di livello due associato all'indirizzo IPv6 2001:DB8::1234:5678:BBBB. Il nodo a cui appartiene tale indirizzo, in questo caso il nodo B, risponde con un messaggio NA il quale al suo interno avrà l'indirizzo MAC corrispondente. Si noti che l'indirizzo destinatario del primo messaggio è stato costruito prendendo i 24 bit inferiori dell'indirizzo MAC del destinatario e aggiunti in coda all'indirizzo FF02::1:FFxx:xxxx.

[10] Media Access Control
[11] `http://www.ieee802.org/3/`

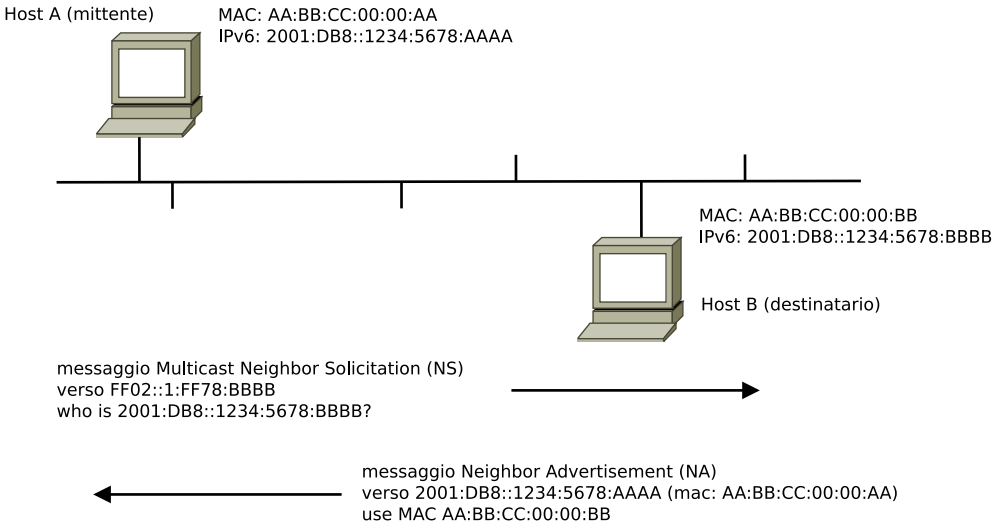


Figura 1.5: Esempio della procedura di Neighbor Discovery

1.5.4 Assegnazione di un indirizzo IPv6

Uno dei maggiori cambiamenti introdotti nel nuovo standard IPv6 è la possibilità da parte del client di autoconfigurarsi. Quest'ultimo si aggiunge alle due modalità già presenti in IPv4: manuale e DHCP. Definita col nome *Stateless Address Autoconfiguration* (SLAAC) permette al host di definirsi il proprio indirizzo IP e tutti gli altri parametri in modo automatico e quindi senza la necessità di un server centralizzato che gestisca tali assegnazioni. Naturalmente questo dovrà essere in accordo con le informazioni presenti nel segmento di rete in cui si trova il nodo. Inoltre, le specifiche prevedono un metodo chiamato *renumbering* per cui è possibile rinumerare tutti gli IP di una rete. Nel seguente paragrafo tratteremo soltanto il metodo SLAAC e tralasceremo le modalità manuale e DHCPv6 perchè molto simili a quelle definite in IPv4.

1.5.4.1 Autoconfiguratione

L'autoconfigurazione, documentata nel RFC 4862, permette essenzialmente di avere un sistema di rete plug-and-play. In sostanza, permette all'host, come accennato in precedenza, di generarsi il suo indirizzo in combinazione ad informazioni locali e informazioni pubblicizzate dal router. L'indirizzo è normalmente così costruito: il nodo riceve il prefisso della sotto rete dal router o dai router a cui aggiunge un identificativo di interfaccia, in caso di rete basata su Ethernet questo identificativo sarà l'indirizzo MAC. Nel caso non sia presente nessun router nel segmento di rete in cui è posizionato il nodo, quest'ultimo non potrà fare altro che generare un indirizzo di tipo link-local (per il prefisso si veda la tabella 1.1) e comunicare soltanto con gli altri nodi presenti in quel segmento di rete. Ad ogni modo, questo indirizzo sarà sempre presente anche in caso il nodo abbia a disposizione un indirizzo globale perchè sarà usato durante le

procedure di gestione della rete locale.
Vediamo brevemente quali siano gli stati che un indirizzo IPv6 può assumere.

- **Tentative Address**: Un indirizzo perchè sia definito unico deve essere prima verificato. Un indirizzo incerto non sarà considerato assegnato ad un'interfaccia e non potrà ricevere nessun pacchetto se non quelli di tipo ND relativi a un DAD[12].
- **Preferred Address**: Indirizzo assegnato ad un'interfaccia e usabile dai protocolli di livello superiore.
- **Valid Address**: Indirizzo preferito o deprecato. Può essere usato come indirizzo sorgente o di destinazione di un pacchetto ed i router si aspettano che il nodo avente tale indirizzo possa ricevere correttamente il pacchetto.
- **Invalid Address**: Indirizzo che non è assegnato ad un interfaccia. Un indirizzo diventa invalido alla scadenza del relativo tempo di validità impostato precedentemente.

Un esempio di esecuzione del protocollo SLAAC può essere visto in figura 1.6.

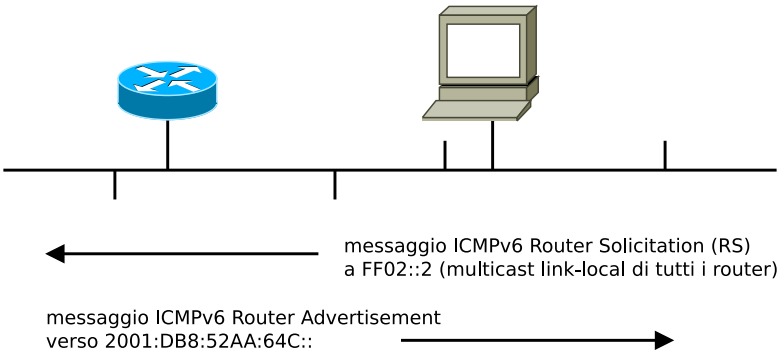


Figura 1.6: Esempio di Stateless Address Autoconfiguration (SLAAC)

1.5.5 Raccomandazioni sulla gestione di ICMPv6

I filtri per ICMPv6 proposti nella tabella 1.5 sono basati sul RFC 4890[13]. Queste raccomandazioni permettono la propagazione dei messaggi ICMPv6 mantenendo le funzionalità della rete ma bloccando i messaggi potenzialmente dannosi. Infatti, molti messaggi ICMPv6 dovrebbero essere unicamente usati in un contesto locale e quindi solo con indirizzi link-local; possiamo quindi identificare tali connessioni usando dei filtri basati sull'indirizzo sorgente e di destinazione e sulla tipologia di ICMPv6 trasmesso. L'RFC 4890 segue proprio tale politica, classificando tali messaggi in base al

[12]Duplicate Address Detection
[13]Recommendations for Filtering ICMPv6 Messages in Firewalls

loro contesto: comunicazioni end-to-end (traffico che transita attraverso il firewall) e comunicazioni locali (traffico generato con l'indirizzo di link-local).
Configuriamo, quindi, le ACL[14] permettendo solo le regole definite nella seguente tabella e bloccando tutti quei messaggi che non siano elencati, siano essi messaggi ICMPv6 sperimentali o non definiti.

	da non bloccare		**dovrebbe essere bloccato**	
Tipo di Messaggio	**In transito**	**Locale**	**In transito**	**Locale**
Gestione della connessione: *permettere connessioni non locali quando sono associate a connessioni autorizzate*				
Destination Unreachable (1)	•	•		
Packet Too Big (2)	•	•		
Time Exceeded (3) - codice 0	•	•		
Parameter Problem (4) - codice 1,2	•	•		
Controllo della connettività: *permettere o non permettere connessioni locali basate sulla topologia*				
Echo Request (128)	•	•		
Echo Response (129)	•	•		
Configurazione dell'indirizzo e del router predefinito: *permettere solo traffico link-local*				
Router Solicitation (133)		•		
Router Advertisement (134)		•		
Neighbor Solicitation (135)		•		
Neighbor Advertisement (136)		•		
Inverse Neighbor Discovery Solicitation (141)		•		
Inverse Neighbor Discovery Advertisement (142)		•		
Ricezione della notifica dall'indirizzo multicast di link-local: *permettere solo traffico da link-local*				
Listener Query (130)		•		
Listener Report (131)		•		
Listener Done (132)		•		
Listener Report v2 (143)		•		
Invio della notifica SEND: *permettere solo traffico da link-local*				
Certification Path Solicitation (148)		•		
Certification Path Advertisement (149)		•		

[14]Lista di controllo degli accessi - Access control list

	da non bloccare		**dovrebbe essere bloccato**	
Tipo di Messaggio	**In transito**	**Locale**	**In transito**	**Locale**
Multicast Router Discovery: *permettere solo traffico da link-local*				
Multicast Router Advertisement (151)		•		
Multicast Router Solicitation (152)		•		
Multicast Router Termination (153)		•		
Messaggi di errore: *permettere connessioni non locali quando sono associate a connessioni autorizzate*				
Time Exceeded (3) - codice 1			•	•
Parameter Problem (4) - codice 0			•	•
IPv6 mobile: *permettere traffico non local verso gli endpoint predefiniti*				
Home Agent Address Discovery Request (144)			•	
Home Agent Address Discovery Reply (145)			•	
Mobile Prefix Solicitation (146)			•	
Mobile Prefix Advertisement (147)			•	

Tabella 1.5: Filtri raccomandati per ICMPv6

1.6 Differenze tra IPv4 e IPv6

Vediamo ora di riassumere quali siano le differenze tra il protocollo IPv4 e il nuovo protocollo IPv6. Tali aspetti sono raffigurati nella tabella 1.6.

1.7 Dual Stack e Tunnel

Visto l'importanza e la pervasività di IPv4 nell'epoca in cui viviamo non è pensabile sostituire tale protocollo in modo istantaneo e indolore; si è deciso quindi di definire due modalità di transizione. La prima, attraverso la cosiddetta implementazione *dual stack* che permette di avere sulla stessa interfaccia sia indirizzi IPv4 sia indirizzi IPv6. Questo permette, a chi è fornito di connettività IPv6, di operare con entrambi i protocolli e di poter quindi accedere ad entrambe le reti. La seconda modalità, predisposta per coloro che non hanno a disposizione connettività IPv6, è chiamata *tunneling* e permette all'utente di realizzare un canale verso un server provvisto di connetività IPv6 il quale incapsulerà il traffico e lo trasmetterà all'utente finale. Attraverso tale modalità possiamo provare il protocollo IPv6 anche nella situazione in cui il provider non lo

Proprietà	IPv4	IPv6
lunghezza indirizzo	32 bit	128 bit
lunghezza indirizzo network	da 8 a 30 bit	64 bit
lunghezza header	da 20 a 60 byte	40 byte
numero di estensioni del header	limitato dal basso numero di opzioni IPv4	illimitato attraverso l'Extension Header di IPv6
frammentazione	ammesso sia a livello di mittente sia nei nodi intermedi	solo a livello di mittente
protocolli di controllo	insieme di diversi protocolli: ARP, ICMP e altri	ICMPv6
MTU minimo	576 byte	1280 byte
ricerca del MTU	opzionale	strettamente raccomandata
assegnamento dell'indirizzo	normalmente un indirizzo per host	normalmente più indirizzi per host
tipi di indirizzo	usa unicast, multicast e broadcast	broadcast non è più usato; si usa unicast, multicast e anycast
configurazione dell'indirizzo	manuale o attraverso DHCP	SLAAC, DHCP o manuale

Tabella 1.6: Differenze tra IPv4 e IPv6

supporti direttamente sulla sua infrastruttura. Di contro, questo può generare problemi di sicurezza ed eludere filtri perimetrali.
Ad oggi, tutti i più recenti Sistemi Operativi implementano entrambe le modalità, dando così la possibilità all'utente di selezionare quella più attinente alle sue necessità.
In particolare, si è di molto sviluppato l'uso del tunneling[15] in quelle aree in cui ci fosse la necessità di scavalcare le restrizioni imposte dagli Internet Service Provider o dalle organizzazioni, di attivare servizi raggiungibili dal mondo IPv6, per esempio web server o mail server, o più semplicemente di avere la possibilità di provare il nuovo protocollo. Questo però, il più delle volte, genera problemi di sicurezza lasciando l'utente senza alcuna protezione da attacchi esterni e permettendo quindi l'accesso indiscriminato a tutti i servizi attivi nel sistema. Tale argomento verrà trattato nei capitoli successivi; per il momento ci limitiamo soltanto nell'esporre i diversi tipi di tunnel a disposizione dell'utente e quali siano le modalità per riconoscerli.

[15] si vedano per esempio: Hurricane (`http://tunnelbroker.net`) e SixXS (`http://www.sixxs.net`)

Nella tabella 1.7 vediamo appunto i nomi dei tunnel e quali siano i loro prefissi.

Tipologia dell'indirizzo	Indirizzo IPv6	Uso
6to4	2002::/16	Il traffico IPv6 viene incapsulato in IPv4 con il protocollo 41 (RFC 3056)
Teredo	2001:0000::/32	Teredo (RFC 4380)
ISATAP	fe80::200:5efe: + ipv4	Intra-Site Automatic Tunnel Addressing Protocol (RFC 5241)

Tabella 1.7: Tunnel IPv6

Come definito nella tabella si ha che ogni tunnel, oltre ad avere uno specifico formato e pacchetto, è identificato univocamente da un prefisso e quindi può essere facilmente bloccato. Vediamoli brevemente.

- **6to4**. Protocollo specificato nel RFC 3056. Permette di incapsulare il protocollo IPv6 all'interno del payload di IPv4 (protocollo 41). È la versione più diffusa in Internet visto che permette facilmente ad una sottorete o ad un host di poter comunicare in internet con IPv6. Essendo identificato dal protocollo 41 possiamo filtrarlo attraverso una semplice ACL IPv4.
- **Teredo**. Protocollo basato su UDP e ha la possibilità di funzionare su reti NAT. Di default è attivo su Windows Vista e Windows 7 ma può essere implementato anche su Sistemi Operativi *nix.
- **ISATAP**. Permette di generare un indirizzo IPv6 locale partendo da un indirizzo IPv4 e di avere un meccanismo di Neighbor Discovery al di sopra di IPv4. Poco usato.

1.8 Specifiche di sicurezza definite in IPv6

1.8.1 Indirizzi privati

Gli indirizzi privati possono essere usati dalle applicazioni client per inibire il tracciamento dell'utente, in particolare per proteggere le connessioni verso Internet e quindi fuori dal proprio segmento di rete. Dalle specifiche del documento RFC 4291, le interfacce Ethernet che usano SLAAC genereranno un indirizzo unico basato sullo standard IEEE EUI-64. Questo, che può sembrare a prima vista un valore aggiunto, si scontra con la necessità di mantenere un certo grado di privacy della persona e/o dell'impresa. A questo scopo è stato definito il documento RFC 4941[16] che specifica una procedura

[16]Privacy Extensions for Stateless Address Autoconfiguration in IPv6

per generare e cambiare temporaneamente l'indirizzo. I requisiti più importanti che si devono rispettare durante la generazione di tale indirizzo sono la totale imprevedibilità del valore e la bassa probabilità di collisione con le altre interfacce già presenti in Internet.
La procedura introdotta dall'RFC funziona nel seguente modo:

1. ottenere l'identificativo dell'interfaccia che si vuole usare senza le modifiche apportate da questo algoritmo;
2. applicare una funzione hash al valore appena ricavato concatenato ad un valore precedentemente salvato o un valore casuale lungo 64 bit;
3. usare l'output della funzione di hash per selezionare l'identificativo dell'interfaccia e aggiornare il valore storico;
4. eseguire la procedura Duplicate Address Detection (DAD);
5. configurare il tempo di validità dell'indirizzo e unirsi al gruppo multicast corrispondente a quel valore di interfaccia;
6. continuare ad usare il precedente identificativo per le vecchie connessioni;
7. ripetere tale procedura ogni volta che ci si collega ad una nuova rete o nel caso scadesse il tempo di validità dell'indirizzo.

1.8.2 IPsec

In questo paragrafo accenneremo brevemente al protocollo IPsec integrato in IPv6, definito negli RFC 4301, 4302, 4303, 4306, 4307, 4308 e 4835, e ai suoi usi per mitigare alcune problematiche di sicurezza del protocollo. Come avviene già per IPv4, IPsec implementa due distinte forme di crittografia: la prima che fornisce l'autenticazione (AH = Autentication Header) e la seconda che fornisce la riservatezza (ESP = Encapsulating Security Payload). Quest'ultimo si suddivide in due modalità distinte: il *tunnel mode* che permette l'incapsulamento e cifratura di un pacchetto IP e la seconda, chiamata *transport mode*, che permette soltanto la cifratura del payload e non quindi del header IP. Queste forme possono essere usate, dietro l'esistenza pregressa di un'infrastruttura di sicurezza a supporto del protocollo, per autenticare e/o cifrare il protocollo OSPF versione 3 o la fase di Neighbor Discovery o nella gestione delle procedure inerenti alla parte mobile di IPv6.

1.8.3 SEND

Il protocollo SEcure Neighbor Discovery (SEND), definito nel RFC 3971, è stato specificato come alternativa all'uso di IPsec nelle funzionalità di rete interne ad IPv6, in particolare quelle di Neighbor Discovery.
Le idee alla base di SEND sono le seguenti:

- usare la procedura CGA[17]. Permette di assicurarsi che il mittente di un messaggio ND sia l'effettivo proprietario di quell'indirizzo;
- aggiungere l'opzione RSA Signature al messaggio ICMPv6 per proteggere tutti i messaggi associati a ND ed a RD;
- definire nell'host una *trust anchors* che corrisponda al certification path del router per poter adottare quest'ultimo come gateway predefinito;
- aggiungere le opzioni Nonce e Timestamp al messaggio ICMPv6 per ovviare ad un attacco di tipo replay.

In definitiva SEND definisce nuove opzioni per trasportare la firma digitale basate su un'infrastruttura PKI[18]; usa un hash della chiave pubblica per generare gli indirizzi; i router sono certificati attraverso l'uso di certificati X.509 e una trust anchor presente nel host e tutti i messaggi SEND sono firmati.

1.9 Prime considerazioni

Concludendo questo capitolo vediamo quali siano gli aspetti positivi del nuovo protocollo e invece quali possano essere le problematiche introdotte. La parte inerente alla sicurezza verrà sviluppata in modo approfondito nei prossimi capitoli. Partiamo col constatare che il nuovo spazio di indirizzamento, pari a 2^{128}, elimina quasi definitivamente la pratica del *port-scanning* della rete locale. A conferma di tale ipotesi basti pensare che per una classe C di IPv4 servono circa 4 minuti (1 host/sec) per completare il port-scanning mentre per il protocollo IPv6, usando 64 bit per definire una classe, si richiedono circa 584 Miliardi di anni. Ovviamente il tempo necessario per eseguire tale procedura su singolo host non varia da un protocollo all'altro. Altra peculiarità del nuovo protocollo è l'esistenza dell'indirizzo link-local che ogni interfaccia dovrà avere anche nel caso il segmento di rete non disponga di indirizzi globali. Questo potrebbe rendere il nodo vulnerabile visto che il più delle volte l'utente e in particolare il Sistema Operativo non considerano a sufficienza il traffico proveniente da IPv6. Negli ultimi tempi questa tendenza stà lentamente mutando anche se ad oggi molti sistemi rimangono ancora privi di software efficace alla protezione dal nuovo protocollo.
Come già visto più volte in questo capitolo, gli host vengono nuovamente responsabilizzati. Questo, se da un lato semplifica il compito dei router, dall'altro gli impone una maggiore attenzione al traffico ricevuto e generato. A questo si aggiunge l'eliminazione degli indirizzi privati e pubblici che quindi, nella visione IPv4, annichilisce il senso introdotto con la suddivisione degli indirizzi. In IPv6 la sicurezza verrà attuata soltanto a livello firewall. Quest'ultimo dovrà gestire in modo corretto il traffico ICMPv6, che

[17] Cryptographically Generated Address
[18] Public key infrastructure

come detto più volte non può essere bloccato a priori, e dovrà controllare il traffico in ingresso e in uscita dei nodi aventi indirizzo globale. Inoltre, tale oggetto dovrà prendersi carico di ispezionare i possibili Extension Header che a differenza di IPv4, il cui header aveva dimensione massima di 60 byte, in IPv6 non è imposto nessun limite. Sicuramente nel caso peggiore questo non è un beneficio in termini di elaborazione, ma essendo questa una situazione abbastanza rara, non può essere definita come uno svantaggio effettivo. Un pregio, invece, è la possibilità di bloccare a priori dei servizi non necessari o non graditi. Questo potrà essere effettuato applicando delle regole in base all'indirizzo di provenienza/destinazione o del tipo di protocollo trasportato, come per i tunnel di tipo 6to4.
Altro importante cambiamento introdotto, e più volte discusso in questo capitolo, è l'eliminazione del protocollo ARP che soffriva di problematiche relative al caching (avvelenamento e lentezza nell'aggiornamento). Con la nuova versione e l'introduzione dei sistemi SEND e IPsec è possibile garantirne l'invulnerabilità da attacchi di tipo *man-in-the-middle* e non avere le problematiche di indirizzi vecchi e non più validi.
Per concludere questo excursus sulle proprietà del protocollo IPv6 non possiamo fare a meno di menzionare una singolarità, se così può essere chiamata, relativa alla generazione del indirizzo in modalità SLAAC: nel caso cambiassimo interfaccia, l'indirizzo IP sarà diverso. Questo in particolare è vero per interfacce di tipo Ethernet in cui la parte inferiore dell'indirizzo è costruito a partire dall'indirizzo MAC aggiungendo a metà dello stesso il valore FF:FE.

Capitolo 2

Sicurezza e Vulnerabilità

Questo capitolo fornirà una panoramica sulle vulnerabilità intrinseche al protocollo IPv6 e quali siano, nel caso ci fossero, le soluzioni adottabili per mitigarne l'uso malevolo. Consapevoli del fatto che la migrazione ad IPv6 è inevitabile, lo spazio di indirizzamento di IPv4, come più volte detto, è esaurito, e si dovrà quindi procedere con una maggiore comprensione del funzionamento e delle limitazioni imposte dalle attuali infrastrutture. Quest'ultimo, come ben precisato nel documento di Babiker, Nikolova e Chittimaneni [6], è un elemento che limita ulteriormente la disponibilità e la presenza di IPv6 nelle reti aziendali. Come tutti i nuovi protocolli, IPv6 ha quindi bisogno di essere implementato e testato in ogni sua componente e come sempre accade questo processo necessita di tempo. Nel caso di IPv6, questo periodo può essere notevolmente diminuito grazie alle conoscenze pregresse con IPv4 e con una maggiore consapevolezza dei problemi di sicurezza che negli anni sono stati evidenziati. Vedremo che alcune delle problematiche sono analoghe a quelle presenti in IPv4 ed altre, invece, sono intrinsecamente legate al protocollo IPv6.
Vediamo in breve quali siano gli aspetti implementativi che possano generare problematiche di sicurezza:

- vulnerabilità delle implementazioni e della gestione del protocollo all'interno dei Sistemi Operativi. Per esempio la vulnerabilità CVE-2010-1843[1] che provoca un attacco DoS nel Sistema Operativo Apple Mac OS;

- possibilità di avere nodi attivi con IPv6 in infrastrutture aventi solo protocollo IPv4 e quindi non autorizzate;

- complessità delle nuove opzioni aggiunte con IPv6 e la possibilità di duplice convivenza tra i due protocolli (IPv4 con IPv6);

- immaturità delle componenti adibite al controllo perimetrale e di sicurezza del traffico;

[1] `http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-1843`

- immaturità dei dispositivi infrastrutturali, prive di alcune funzionalità base o limitati a livello software e quindi non performanti come per IPv4;
- vi è già un uso cospicuo del protocollo IPv6 per scopi malevoli.

Prima di addentrarci negli aspetti specifici di IPv6 si ha la necessità di precisare un aspetto che, anche se può apparire scontato, è di notevole importanza durante la lettura di queste pagine. Il protocollo IPv6 non è intrinsecamente più sicuro di IPv4 ma al più mette a disposizione alcune funzionalità atte a diminuire alcuni tipi di attacchi. Di conseguenza la sicurezza delle implementazioni andrà di pari passo con le implementazioni del nuovo protocollo nelle infrastrutture e quelle dei Sistemi Operativi. Questa tendenza può essere notata nell'istogramma 2.1 in cui si vede come le vulnerabilità riportate siano in notevole crescita, in particolare dovuto al maggior impiego di IPv6 nelle infrastrutture pubbliche (operatori telefonici e mobili). Si noterà anche che nei primi tre mesi del 2012 si ha avuto un notevole aumento delle vulnerabilità rispetto agli anni precedenti, sintomo che la comunità di ricerca stà prendendo in seria considerazione il protocollo IPv6.

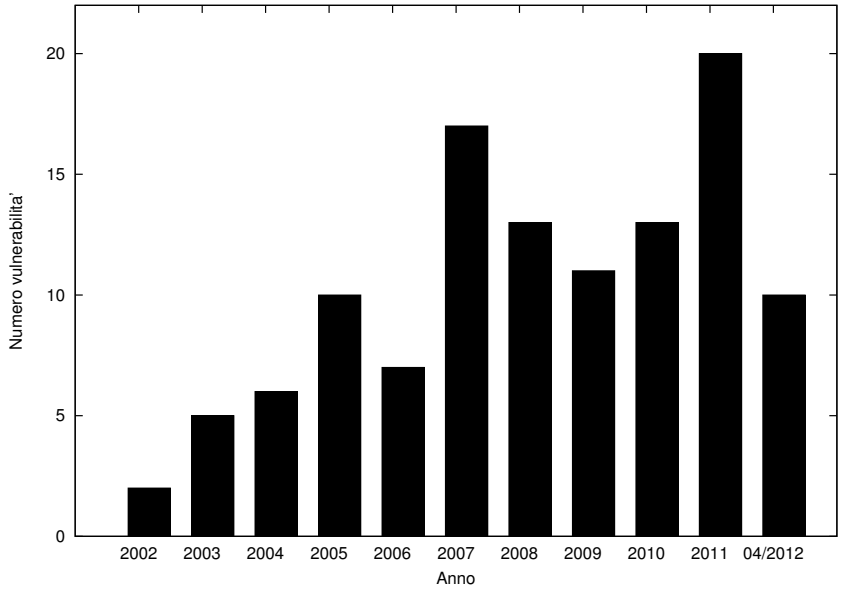


Figura 2.1: Vulnerabilità CVE riguardanti IPv6

Come si vedrà nei successivi paragrafi e nel prossimo capitolo, IPv6 privo di SEND o IPsec non risolve nessuna delle vulnerabilità presenti a livello 2 della pila ISO/OSI. Per esempio attacchi di tipo sniffing, traffic flooding, man-in-the-middle, rogue devices o ARP table overflow sono ancora attuabili, anche se alcuni con tecniche diverse. Possiamo infatti usare in modo malevolo i messaggi di Neighbor Advertisement (NA), Router Advertisement (RA) e Duplicate Address Detection (DAD) per perpetrare attacchi simili a quelli già conosciuti in IPv4. Si ha quindi che una buona gestione del livello 2 come già avviene per IPv4, per esempio con l'uso delle vlan e del protocollo 802.1x,

può mitigare enormemente le vulnerabilità di livello 3. Per quest'ultimo, come abbiamo già accennato all'inizio del paragrafo, c'è la necessità che i produttori di dispositivi di rete si attrezzino a sviluppare le funzionalità in modo equivalente, nella forma, a quelle presenti nel mondo IPv4. Questo, se da un lato è un ottima opportunità commerciale per i produttori, dall'altro il più delle volte si traduce in un ulteriore onere a carico delle imprese che, in tempo di crisi, non giova certamente ad incentivare la transizione. Una nota particolare va agli apparati firewall e IDS/IPS i quali, ad oggi, supportano solo in parte l'enorme complessità introdotta da IPv6. In particolare, come vedremo, la complessità della concatenazione degli Extension Header e all'uso della frammentazione. A questo riguardo il NIST[2] ha pubblicato un documento [30] in cui identifica le caratteristiche che un dispositivo IPv6 deve avere per processare correttamente il nuovo protocollo e aiuta quindi le imprese a scegliere correttamente gli apparati.
In questo capitolo si tratterà delle vulnerabilità derivanti dall'uso della funzionalità Multihoming, della priorità nella selezione dell'indirizzo e delle vulnerabilità presenti nell'uso del multicast. Nel quarto paragrafo si vedranno le vulnerabilità presenti nei servizi accessibili direttamente dagli indirizzi IPv6, nel quinto della gestione del QoS, nel sesto del DHCPv6 e infine delle problematiche introdotte dall'uso dei Tunnel. Per concludere il capitolo si evidenzieranno le debolezze delle difese odierne dalle e-mail di Spam provenienti da server di posta IPv6.

2.1 Multihoming

In questo paragrafo vediamo una delle funzionalità aggiuntive di IPv6 rispetto a IPv4: il Multihoming. Come già definito nel precedente capitolo, tale aspetto ci permette di avere in simultanea più di una connessione per accedere a delle risorse in Internet. Per esempio, l'utente potrebbe avere contemporaneamente più di un'interfaccia di rete attiva nello stesso istante, cablata e wireless, e ciascuna avente più di un router definito. Se da un lato introduce una maggiore affidabilità nella trasmissione dall'altro complica notevolmente l'infrastruttura e potrebbe, in alcune condizioni, portare a problemi di sicurezza.
Il documento RFC 4218 [72] presenta una panoramica delle vulnerabilità che affliggono la componente Multihoming. In particolare, tratta delle ripercussioni che tale implementazione potrà avere nella rete: denial of service e re-routing dei pacchetti verso una destinazione non desiderata (*black hole*[3]). Inoltre precisa che i pacchetti di trasporto privi di connessione, come UDP, presentano maggiori problemi di sicurezza in confronto ai rispettivi protocolli orientati alla connessione e che, nel caso si usassero identificativi e indicatori di locazione disgiunti, la sicurezza si basa sui primi e non sui secondi. In aggiunta, per minimizzare i possibili attacchi al servizio Multihoming, bisogna porre molta attenzione sia al servizio DNS, per esempio usando DNSSEC, sia

[2]National Institute of Standards and Technology
[3]un luogo nella rete in cui il traffico in ingresso viene silenziosamente scartato

ai protocolli di routing. Nel caso di OSPF è possibile usare IPsec. In conclusione si suggerisce l'uso di filtri di indirizzi *spoofed*, tali da eliminare gli indirizzi non validi all'interno della rete IPv6 (RFC 2827 e RFC 3704).
Altre raccomandazioni derivano dal protocollo SHIM6. Vediamole brevemente:

- usare indirizzi basati sul hash (RFC 5535 [7]) così da poterne provare l'identità ed evitare attacchi di reindirizzamento;
- attuare il documento RFC 5534 [5] che permette di identificare attacchi di tipo flooding provenienti da entità di terze parti;
- prima che il destinatario crei uno stato bisogna che ci si accerti che la comunicazione sia bidirezionale. Questo obbliga l'attaccante a rivelare il proprio indirizzo;
- usare messaggi comprensivi di identificativo di contesto tali da evitare attacchi di tipo replay e quindi prevenire attacchi off-path;
- usare il precedente identificativo in ogni messaggio riguardante il protocollo SHIM6.

2.2 Priorità nella selezione dell'indirizzo

In IPv4 ogni interfaccia fisica ha normalmente, se tralasciamo le possibilità di alias offerte dal singolo Sistema Operativo, un solo indirizzo unicast che potrà o meno essere globalmente visibile. In antinomia, l'interfaccia IPv6 potrà avere molteplici indirizzi: un indirizzo link-local, un indirizzo locale unico e un indirizzo globale raggiungibile da Internet. Nel caso si usassero le tecniche di renumerazione e di Multihoming si potrebbero avere più indirizzi dello stesso tipo. Si ha quindi, che già con le possibilità offerte da IPv6 la tabella di routing può diventare decisamente complessa. Introducendo anche la possibilità di far convivere nella stessa interfaccia i due protocolli, la selezione dell'indirizzo sarà notevolmente complicata e sarà quindi necessario definire alcune regole. A questo riguardo è stato pubblicato il documento RFC 3484 [31] in cui si specificano le regole, basate sull'indirizzo sorgente e di destinazione, con le quali bisogna preferire un instradamento piuttosto che un altro. Non ci soffermeremo sul descrivere tali regole, non essendo l'obiettivo di questo testo, ma passeremo direttamente alle implicazioni di sicurezza derivanti da tali scelte. La tabella 2.1 è solo un esempio di come possano essere definite tali regole. Un corretto uso della procedura di selezione dell'indirizzo, o meglio uno scorretto uso di tale procedura, può procurare un notevole problema di sicurezza, in particolare nella disponibilità e integrità dei dati. Una selezione impropria potrà risultare in una indisponibilità della risorsa richiesta o ad un instradamento poco efficiente. In un'ottica da attaccante questi aspetti diventano molto importanti. Si pensi al caso in cui un attaccante riesca, compromettendo o modificando le politiche di selezione, a redirigere il traffico verso un punto da lui prestabilito

Prefisso	Precedenza	Etichetta	Uso
::1/128	50	0	loopback
::/0	40	1	predefinito (incluso l'indirizzo IPv6 nativo)
2002::/16	30	2	6to4
::/96	20	3	indirizzi IPv4 compatibili
::ffff:0:0/96	10	4	indirizzi IPv4 mappati all'interno di IPv6

Tabella 2.1: Politiche di selezione dell'indirizzo IPv6

causando così un attacco di tipo man-in-the-middle o semplicemente un attacco DoS. In aggiunta a questo tipo di attacchi, è possibile usare tali politiche per evidenziare informazioni private. Per esempio l'attaccante potrebbe effettuare una richiesta verso un host vittima con un indirizzo sorgente differente dalla rete e osservare l'indirizzo all'interno del pacchetto di risposta. Questo permetterebbe di estrarre diverse informazioni riguardanti gli indirizzi usati dall'obiettivo preso in considerazione.
Un'ultima interessante considerazione sulla priorità delle richieste è come i Sistemi Operativi POSIX gestiscano le richieste DNS attraverso la funzione getaddrinfo[4]. Tale funzione predilige le richieste IPv6 e nel caso queste non fossero disponibili effettuerà una richiesta IPv4. Questa politica, definita nel file */etc/gai.conf*, si ripercuote in tutte le applicazioni che facciano richieste DNS, per esempio il browser. Quest'ultimo, in particolare, potrebbe essere un buon vettore di attacco nel caso in cui l'attaccante forgiasse una risposta DNS di tipo AAAA avente un indirizzo da lui designato. In questo modo la vittima sarebbe rediretta verso un sito diverso da quello richiesto e quindi si potrebbero perpetrare diversi tipi attacco.

2.3 Multicast

2.3.1 Indirizzi

Iniziamo il paragrafo, forse uno dei più importanti del capitolo per via dei risvolti pratici, col presentare gli indirizzi multicast definiti dal protocollo IPv6 e quali siano gli usi ad esso associati. Nel primo capitolo si è semplicemente definito come indirizzo multicast un indirizzo la cui parte più significativa era identificata da otto bit posti a 1 (FF in esadecimale) e che tali indirizzi erano usati dal protocollo ICMPv6, per esempio dalla procedura di Neighbor Discovery e da quella di Router Discovery. Si vedrà come tali indirizzi non siano limitati solo a tale uso ma che negli anni siano stati impiegati per un vasto insieme di applicazioni: groupware, distribuzione di contenuti multimediali, ricerca, routing, replicazione di database, grid computing e distribuzione di informazioni real-time. Per ottimizzare ulteriormente la distribuzione dei dati è stato introdotto il concetto di *scope* il quale può essere implementato sia a livello di singola interfaccia sia

[4]documentata nel RFC 2553

a livello globale (Internet). Un esempio dell'uso di indirizzi multicast associati ad uno scope è sicuramente la distribuzione di contenuti multimediali ad alta definizione a più utenti contemporaneamente. In IPv6 è stata inoltre introdotta una nuova versione del protocollo di Multicast Listener Discovery (MLD), che in versione 1 era già presente in IPv4. La versione 2 permette la gestione, l'unione e l'uscita dal gruppo multicast attraverso due nuovi messaggi ICMPv6:

- Multicast Listener Query (tipo 130);
- versione 2 del Multicast Listener Report (tipo 143).

Questi messaggi dovranno essere inviati con indirizzo sorgente di tipo link-local e dovranno avere i seguenti parametri configurati: Hop Limit uguale ad 1 e l'opzione Router Alert nell'header Hop-by-Hop. Veniamo ora alla principale novità introdotta. A differenza di MLDv1 per IPv4, la nuova versione dà la possibilità all'interfaccia di valutare, per ogni indirizzo multicast, quali indirizzi sorgenti debbano essere accettati e quali debbano essere rifiutati. Tale caratteristica è documentata dal RFC 3569[5] col nome di Source Specific Multicast.
In figura 2.2 sono rappresentati i campi che compongono l'indirizzo multicast IPv6.

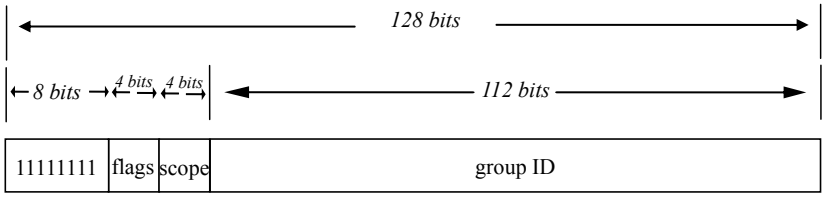


Figura 2.2: Formato dell'indirizzo multicast IPv6

Come definito all'inizio di questo paragrafo l'indirizzo multicast è identificato dai primi otto bit a 1. Il campo flag ha funzionalità bivalente: la prima, di identificare se un indirizzo è di tipo multicast predefinito o se è un indirizzo temporaneo, cioè non definito in modo permanente dalle specifiche; la seconda, di specificare se l'indirizzo temporaneo è autorizzato e se ha un prefisso unicast correttamente specificato al suo interno, individuando così un punto di rendezvous.
I valori ammessi dal campo flag sono i seguenti (per una completa disamina si vedano gli RFC 3306 e 3956):

bit	**descrizione**
0 0 0 0	indirizzo multicast definito
0 0 0 1	indirizzo transitorio senza un prefisso unicast incorporato
0 0 1 1	indirizzo transitorio con un prefisso unicast incorporato
0 1 1 1	indirizzo transitorio con un prefisso unicast incorporato e un rendezvous point

[5]RFC 3569: An Overview of Source-Specific Multicast (SSM)

Il campo scope, obbligatorio, limita la ricezione del pacchetto a soli alcuni dei gruppi. I valori ammessi sono i seguenti (RFC 4291):

valore	scope
1	Interface Local
2	Link Local
4	Admin. Local
5	Site Local
8	Organization Local
E	Global

Per concludere, il Group ID specifica l'insieme dei nodi membri di un gruppo multicast. Si potrà avere, per esempio, che l'insieme dei server NTP[6] sono raggiungibili dai seguenti indirizzi:

FF02::101	tutti i server NTP con scope uguale a Link Local
FF04::101	tutti i server NTP con scope uguale a Admin Local
FF05::101	tutti i server NTP con scope uguale a Site Local
FF08::101	tutti i server NTP con scope uguale a Organization Local
FF0E::101	tutti i server NTP con scope uguale a Global

Una lista completa degli indirizzi IPv6 multicast categorizzati per scope può essere reperita nel documento RFC 2375 [51] ed eventuali aggiornamenti saranno pubblicati alla pagina del IANA[7].

2.4 Servizi

Come si è visto l'indirizzo multicast permette l'accesso diretto a servizi presenti sulla rete senza conoscerne a priori l'indirizzo link-local o site-local. Questo, se da un lato ha il grosso vantaggio di trovare facilmente i servizi UDP, dall'altro ne permette una facile identificazione anche a chi non è autorizzato o ha intenzioni malevoli.
Vediamo un breve elenco degli indirizzi multicast più significativi (si sostituisca il carattere *x* con lo scope desiderato):

[6] Network Time Protocol

[7] `http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml`

Indirizzo	Descrizione
ff02::1	tutti i nodi del segmento locale
ff02::2	tutti i router del segmento locale
ff02::5	tutti gli SPF router di OSPFv3
ff02::6	tutti i DR router di OSPFv3
ff02::9	router RIP
ff02::a	router EIGRP
ff02::d	router PIM
ff02::16	report MLDv2 (definito nel RFC 3810)
ff02::1:2	tutti i server e i relay DHCPv6 (definito nel RFC 3315)
ff05::1:3	tutti i server DHCPv6 (definito nel RFC 3315)
ff0*x*::fb	richiesta multicast DNS
ff0*x*::101	server NTP
ff0*x*::108	server NIS
ff0*x*::114	definito per eventuali esperimenti

Prendiamo per esempio l'indirizzo multicast ff02::fb che ci permette di effettuare una richiesta multicast DNS (mDNS) e quindi senza essere a conoscenza a priori di un indirizzo link-local o globale di un server DNS.

```
$ dig -p5353 google.com @ff02::fb A

;; ANSWER SECTION:
google.com. 3600 IN A 173.194.35.4

;; SERVER: fe80::xxxx:xxxx:xxxx:xxxx%2#5353(ff02::fb)
```

Un alternativa all'uso specifico di un indirizzo multicast per un dato servizio è l'uso dell'indirizzo generico ff02::1. Infatti, a differenza dei servizi UDP IPv4 i quali non erano in ascolto sull'indirizzo di broadcast, i servizi UDP IPv6 sono in ascolto su tutti gli indirizzi (::) presenti nel sistema, rispondo quindi alle richieste con indirizzo di destinazione multicast. Questa caratteristica è valida per tutti i servizi UDP e quindi si potrebbe effettuare la precedente richiesta nella seguente maniera.

```
$ dig google.com @ff02::1 A

;; ANSWER SECTION:
google.com. 3600 IN A 173.194.35.4

;; SERVER: fe80::xxxx:xxxx:xxxx:xxxx%2#53(ff02::1)
```

Si avrà che tale richiesta raggiungerà tutti i server DNS presenti nella rete locale ma con la semplicità di aver inviato un solo pacchetto. Automatizzando la procedura sarà possibile effettuare l'enumerazione dei servizi UDP presenti sulla rete. Questa possibilità non è certamente valida per i protocolli orientati alla connessione, come può essere

TCP.
Dopo aver visto come sia possibile identificare i servizi a protocollo non connesso vediamo ora come sia possibile, sempre usando gli indirizzi multicast, rilevare i nodi presenti nella rete. Un esempio semplice di tale procedura in un sistema Linux è la seguente.

```
# ping6 -c 3 -I eth0 ff02::1 >/dev/null 2>&1
# ip neigh | grep '^fe80'
fe80::211:43ff:fexx:xxxx dev eth0 lladdr 00:11:43:xx:xx:xx REACHABLE
fe80::21e:c9ff:fexx:xxxx dev eth0 lladdr 00:1e:c9:xx:xx:xx REACHABLE
fe80::218:8bff:fexx:xxxx dev eth0 lladdr 00:18:8b:xx:xx:xx REACHABLE
[...]
```

In pratica, si invia una richiesta ICMPv6 di tipo echo request (*ping6*) all'indirizzo multicast corrispondente a tutti i nodi e specificando quale interfaccia dovrà essere usata (*eth0*). A questo punto controlliamo la tabella dei "vicini" (*ip neigh*) nei quali verrà visualizzato quali siano gli indirizzi link-local e i rispettivi indirizzi MAC delle interfacce presenti nella rete e che rispondano a questo tipo di richiesta. Non è sicuramente una modalità molto comoda ma ci da un'idea sugli host e i router presenti sul segmento di rete in cui ci troviamo.
Queste ed altre tecniche per l'enumerazione dei nodi verranno approfondite nel prossimo capitolo.

2.4.1 Considerazioni sulle vulnerabilità

Nel precedente paragrafo abbiamo visto come sia facile recuperare informazioni dal segmento di rete locale usando semplicemente indirizzi definiti nelle specifiche e come questi possano essere suddivisi in base al valore del campo scope. Un'altra vulnerabilità presente nelle specifiche è la possibilità, da parte di un attaccante, di amplificare il consumo di banda o di esaurire le risorse disponibili ad una vittima. Questo attacco, chiamato *reflector attack*, permette all'attaccante di forgiare ed inviare un pacchetto multicast avente come indirizzo sorgente quello della vittima cosichè quest'ultimo riceva tutte le risposte provenienti dai nodi presenti nella rete.
Altre vulnerabilità specifiche del protocollo MLD includono: Denial of Service (DoS), traffico non voluto e la possibilità di retrocedere dalla versione 2 alla versione 1. Un semplice sistema per evitare queste problematiche è forzare i valori di Hop e di scope di tipo unicast. Altre vulnerabilità, presentate nel documento RFC 5294 [78], sono presenti nel Protocol Independent Multicast (PIM) il quale può essere usato per inviare traffico non voluto al fine di sostituire il router predefinito. Tale problematica può essere evitata con l'uso di IPsec nei messaggi PIM.

2.4.2 Aspetti di sicurezza irrisolti

Nel primo capitolo si è parlato di come sia possibile proteggere il protocollo IPv6 da attacchi o falsificazioni di messaggi adibiti alla gestione della rete. Il modo più naturale per attuare questa difesa in IPv6 è sicuramente l'uso di IPsec essendo quest'ultimo già integrato nello standard. Per contro, IPsec soffre di una grave lacuna nel supporto degli indirizzi multicast: semplicemente non li supporta, o meglio, li supporta nel sol caso il mittente del gruppo sia unico.
In particolare, esistono due aspetti ancora irrisolti nella gestione sicura del multicast e riguardano IPsec e la gestione delle chiavi. IPsec, nelle sue versioni ESP e AH, è stato progettato principalmente per indirizzi di tipo unicast sebbene funzioni, in linea di principio, anche con indirizzi multicast ma ad oggi non esiste nessuna specifica a riguardo. Per esempio, cosa succederebbe ad una associazione multicast quando i membri si uniscono o lasciano il gruppo?
La gestione delle chiavi per IPsec, fornita dal protocollo IKE, è intrinsecamente legata ad un protocollo a due entità. Alternative a questo protocollo per la gestione di un gruppo di chiavi sono state più volte proposte ma non è mai stato concordato uno standard. PIM, dal canto suo, suggerisce di usare IPsec con chiavi manuali ma, come si può ben intuire, questo inficia la scalabilità dell'infrastruttura e ne pone grosse limitazioni.

2.5 QoS

Il protocollo IPv6 introduce diverse novità nel campo del Quality of Service di livello 3. Viene introdotto un nuovo campo chiamato *Flow Label* e viene ridefinito il campo *Traffic Class*. Ci si deve, quindi, preoccupare e assicurare che tali campi non vengano usati per analizzare il tipo di traffico o usati impropriamente come vettori di nuove forme di attacco. Si tenga presente che i campi Traffic Class e Flow Label presenti nel header statico di IPv6 non sono protetti nè da IPsec ESP nè da IPsec AH. Questo per via che tali informazioni devono essere disponibili ai nodi intermedi per essere alterate in base alle condizioni del traffico. Sebbene il documento RFC 3697 specifichi che il campo Flow Label non possa essere modificato, AH ha mantenuto la possibilità prevista in origine. Essendo quindi tali campi facilmente modificabili non si dovrebbe predisporre nessuna regola di sicurezza basata sul valore di quest'ultimi. Infatti, un utente malevolo potrebbe modificarli e violare le regole imposte. Inoltre tali campi potrebbero essere usati da un attaccante per ottenere un servizio privilegiato, a maggior priorità, o nel caso generasse molto traffico con priorità maggiore, per produrre un attacco Denial of Service nei confronti del traffico legittimo. Si pensi alla priorità del traffico voce (VoIP). Nel caso in cui l'utente malintenzionato riuscisse a carpire i valori di QoS e le politiche ad esso associate, e come abbiamo visto questa operazione non è complessa essendo questi disponibili a qualsiasi osservatore, potrebbe con estrema facilità impedire sia il traffico con lo stesso livello di priorità sia con priorità inferiore. Non essendoci modo per ovviare a tale problema, le uniche soluzioni sono quelle di

imporre le politiche di priorità a determinati nodi e di fissare un valore massimo, di banda per esempio, usabile da quel valore di QoS. Quest'ultimo, di contro, non eviterà il blocco del servizio associato a quel determinato valore ma permetterà solamente di non saturare l'intera banda disponibile. Altro aspetto che denota la necessità di visionare i campi di QoS è l'uso di protocolli di segnalazione delle priorità e di prenotazione delle risorse, per esempio RSVP, i quali, di norma, non funzionano strettamente tra gli host ma precludono l'ispezione del contenuto dei pacchetti nei nodi intermedi. Ultimo, ma non per questo meno importante, è l'uso del campo Flow Label come *covert channel* permettendo all'eventuale attaccante di trasportare informazioni senza indurre in sospetto i sistemi si sicurezza perimetrale.

2.6 DHCPv6

Dynamic Host Configuration Protocol versione 6 (DHCPv6) è la controparte IPv6 della versione IPv4 e fornisce un servizio di assegnamento degli indirizzi e di altri parametri utili al funzionamento del host. Infatti, a differenza della modalità SLAAC, il servizio DHCPv6 può inviare gli indirizzi unicast dei server DNS ed NTP. Questi, come abbiamo visto nel paragrafo dedicato al Multicast, possono essere anche usati interrogando appositi indirizzi multicast, sempre che il servizio sia presente nella rete locale. Ci permette, inoltre, di gestire l'aggiornamento automatico del campo DNS relativo all'indirizzo IPv6 appena rilasciato (dynamic DNS update). Questo aspetto è molto importante in infrastrutture in cui vi è la necessità di contattare direttamente i nodi senza un servizio di multicast dedicato. Si tenga presente che l'indirizzo IPv6 è notevolmente più lungo di IPv4 e quindi una sua eventuale memorizzazione mentale è assai più complessa. Inoltre, essendo l'assegnazione dell'indirizzo basata su *lease*, si avrà che al termine dell'assegnamento dell'indirizzo quest'ultimo potrà essere cambiato e quindi non si avrà un identificativo perennemente uguale. Tale problema di privacy, in ambito pubblico, è stato più volte discusso, anche in ambiti governativi, e diverse soluzioni alternative all'uso del servizio DHCPv6 sono state proposte ed implementate. Una delle ultime, la quale sembra avere ricevuto diversi consensi positivi, è stata proposta da Fernando Gont[8] in cui presenta un nuovo algoritmo di generazione della parte relativa al Host ID dell'indirizzo.
Molte organizzazioni hanno da sempre preferito l'uso di una gerarchia di assegnazione degli indirizzi e quindi si presuppone che tale scelta verrà adottata anche per IPv6. Il futuro è d'obbligo, perchè ad oggi solo pochi Sistemi Operativi supportano nativamente un client DHCPv6 e quindi si è obbligati a ripiegare sulla modalità SLAAC, l'unica resa obbligatoria dallo standard IPv6. A differenza di quest'ultima, il server DHCPv6 permette sia di avere sempre a disposizione i dati relativi alle assegnazioni sia di poterne definire una struttura divisa per area. Questo, se da un lato semplifica e aumenta

[8] `http://tools.ietf.org/id/draft-gont-6man-stable-privacy-addresses-00.txt`

la possibilità di controllo, dall'altro proroga le vulnerabilità già presenti nella versione per IPv4. Si avrà, il più delle volte, un indirizzamento contiguo permettendo una facile identificazione dei nodi presenti nella rete. Basterà trovare un indirizzo e poi sarà facile risalire a tutti gli altri vanificando la difficoltà di enumerazione imposta dal numero elevato di valori che può assumere il campo Host ID. A tale vulnerabilità alcuni produttori hanno introdotto la possibilità di generazione random degli indirizzi.
Altro punto di vulnerabilità, come accade per tutti i servizi erogati da un unico punto, è la possibilità di attacchi DoS. Per esempio, l'attaccante potrebbe usare i messaggi SOLICIT o RENEW per esaurire le risorse a disposizione del servizio e di conseguenza interrompere il servizio erogato. Altri possibili attacchi potrebbero essere:

- un server malevolo potrebbe inviare risposte a richieste DHCPv6 distribuendo valori o parametri diversi da quelli imposti dall'organizzazione;
- un alto volume di richieste provenienti da clienti malevoli potrebbero causare un Denial of Service;
- l'intercettazione dei pacchetti DHCPv6 atti a comprendere i servizi disponibili nel segmento di rete.

L'unica vulnerabilità che la versione IPv6 non ha importato da IPv4 è quella di non soffrire dell'attacco causato dall'esaurimento degli indirizzi di rete, essendo quest'ultimi pari a 2^{64}.

2.7 Tunnel

I tunnel, come abbiamo già intravisto nel primo capitolo, permettono una più facile transizione al nuovo protocollo in quelle infrastrutture prive di connettività IPv6 verso Internet. In particolare, permettono di incapsulare il traffico IPv6 all'interno di IPv4 o in futuro, di incapsulare IPv4 all'interno del payload di IPv6. Si potranno, quindi, avere situazioni simili a quella proposta in figura 2.3 in cui tra i router delle due aree è presente solo connettività IPv4. Per questo esempio potrebbe essere usato il protocollo 41 in cui IPv6 viene incapsulato in un pacchetto IPv4. Nel caso la situazione fosse inversa si userebbe il Next Header uguale a 4.
Altre soluzioni possibili per il trasporto di IPv6 sono: il Generic Routing Encapsulation (GRE[9]), IPSec ESP[10], 6to4, 6rd, Teredo e altri che verranno discussi qui di seguito. Nelle prossime pagine verranno presentate le problematiche e le vulnerabilità che affliggono i tunnel, siano essi usati per scopi leciti o meno leciti. In particolare ci si concentrerà su quegli usi che permettono di evadere le regole imposte dalla sicurezza perimetrale.

[9]RFC 2784 - usato, per esempio, all'interno dell'infrastruttura di Google [6]
[10]RFC 4303

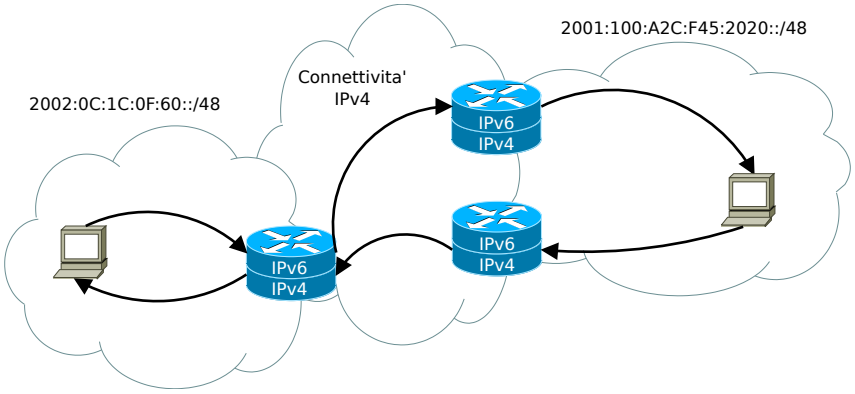


Figura 2.3: Esempio di Tunnel IPv6

2.7.1 Sicurezza

Introduciamo questa sezione con un breve elenco in cui definiamo gli elementi da tenere in considerazione durante l'implementazione o la difesa dei tunnel IPv6. Gli aspetti su cui ci si concentrerà sono i seguenti:

- Tunnel endpoint
- Ispezione del traffico
- Controllo degli accessi
- Terminazione

Iniziamo con l'analizzare l'importanza dei tunnel endpoint. Questi, essendo il punto di ingresso e di uscita del traffico incapsulato, sono anche i punti più vulnerabili della catena e quelli che, normalmente, subiscono più attacchi. Per questo motivo, si dovrebbe attuare una politica atta ad ispezionare il traffico passante e imporre le stesse politiche già previste per il protocollo IPv4. Si dovranno quindi prevedere dei filtri basati sull'indirizzo sorgente e destinazione, un meccanismo di protezione e ispezione di eventuali virus, malware e di eventuali proxy a livello applicativo e sistemi di protezione delle intrusioni di pari efficacia a quelli presenti in IPv4. In particolare è fondamentale l'ispezione della catena degli Extension Header e la normalizzazione dei valori all'interno del header IPv6. Non è certamente un lavoro di facile implementazione con i dispositivi oggi presenti e molta strada dovrà essere fatta, come vedremo meglio nel prossimo capitolo, in cui si presenteranno alcuni attacchi che sfruttano proprio la mancanza di un corretto controllo degli header. Applicare tali politiche, sia nella parte di ingresso che di uscita del tunnel, proteggerebbe sia l'eventuale provider che fornisce il servizio sia l'infrastruttura del cliente. Nel caso in cui il tunnel fosse realizzato con IPsec e quindi si avrebbe una protezione end-to-end, si dovrebbero applicare regole tali da autorizzare correttamente le estremità del tunnel. In aggiunta a queste politiche si consiglia

l'adozione del Unicast Reverse Path Forwarding (Unicast RPF[11]), un filtro dinamico implementato alle estremità del tunnel. Vediamone un esempio. Un utente malintenzionato potrebbe inviare o riflettere del traffico alle estremità del tunnel ed inviarlo ad un router presente all'interno della rete. Quest'ultimo lo invierà alla corretta destinazione perchè presume che tali pacchetti siano legittimi e quindi il traffico raggiungerà la destinazione voluta dall'attaccante senza che alcun sistema di protezione sia stato allertato. Tale minaccia potrà essere individuata e ridotta utilizzando le regole Unicast RPF e filtrando correttamente i pacchetti ad entrambe le estremità. Altra minaccia di notevole importanza è la possibilità, per un attaccante, di intercettare e comprendere il traffico all'interno del tunnel. Un rimedio possibile consta nell'uso di IPsec o di GRE. Tuttavia, l'identificazione del header IPv4 come un pacchetto valido, e quindi in accordo con le regole di sicurezza precedentemente configurate, non equivale necessariamente ad affermare che il pacchetto incapsulato sia altrettanto legittimo. Infatti, a seconda del dispositivo di sicurezza usato, può essere realmente complesso verificare che il contenuto del pacchetto incapsulato verifichi le regole imposte per il pacchetto esterno. Per questo motivo, un utente malintenzionato o semplicemente con l'intenzione di aggirare le politiche potrebbe utilizzare il tunnel per nascondere il traffico ai sistemi di deep packet inspection o di firewall.
Le Access Control List (ACL) dei router, normalmente presenti nelle infrastrutture odierne, e quindi ottimizzate per il protocollo IPv4 e non per IPv6, dovrebbero essere sempre in grado di bloccare l'accesso alla rete da specifici indirizzi IPv6 o da interi blocchi. Un esempio di politica potrebbe essere quella di impedire gli indirizzi, definiti dallo IANA, corrispondenti al protocollo 6to4. Queste ACL potrebbero, infatti, non essere in grado di riconoscere correttamente il campo Next Header, i messaggi ICMPv6 o di processare correttamente i valori di porta presenti in TCP e UDP. Il motivo è presto detto. L'intestazione IPv6 e gli indirizzi non sono visibili direttamente dal router di livello 3, essendo questi parte integrante del payload del pacchetto, d'altronde è proprio questo l'intento funzionale del concetto di tunnel. Inoltre bisogna essere consapevoli, come vedremo, che un tunnel IPv6 può essere creato o distrutto senza che venga allertato alcun sistema di sicurezza e che quindi non ci sia nessun avviso per l'amministratore di rete. In definitiva sarà strettamente necessario implementare delle strategie efficienti atte a rilevare e filtrare correttamente eventuale traffico IPv6 incapsulato. In caso contrario si avrà che i tunnel IPv6 diventano a tutti gli effetti una minaccia molto seria, capaci di generare una backdoor all'interno dell'infrastruttura. Un esempio è rappresentato dalla figura 2.4. I protocolli come quelli illustrati in figura sono facilmente identificabili dai dispositivi periferici, come router e firewall, anche se quest'ultimi non hanno le capacità di ispezionare il traffico incapsulato e quindi non possono applicare le corrette politiche al traffico IPv6. Bisognerebbe, quindi, imporre delle ACL IPv4 tali da bloccare i protocolli di tunneling noti e le porte predefinite e garantire che tale traffico passi attraverso un apparato capace di ispezionare correttamente il traffico IPv6

[11]RFC 3704 - `https://tools.ietf.org/rfc/rfc3704.txt`

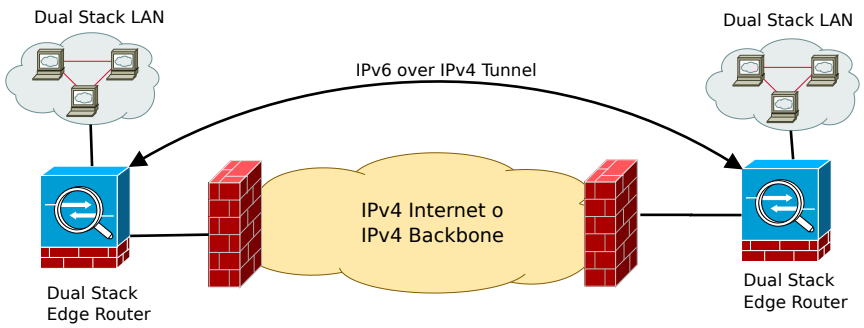


Figura 2.4: Tunnel 6to4 trasparente all'infrastruttura di sicurezza IPv4

incapsulato. Molti firewall odierni hanno la capacità di bloccare il protocollo IPv4 numero 41 e così facendo si eviterebbe l'uso dei tunnel 6over4, ISATAP e 6to4 da parte degli utenti. Negli anni, come vedremo, si sono sviluppati altri tipi di protocollo tali da supplire al funzionamento dei tunnel all'interno di reti in cui fosse presente l'architettura NAT. Uno di questi è chiamato Teredo e sfrutta per il proprio funzionamento il protocollo non connesso UDP. Infatti, anche se i firewall IPv4 di livello 3 filtrassero la porta predefinita (3544) non si avrebbe la certezza di impedire che tale protocollo venga usato. Si sà quanto sia facile cambiare il valore di porta e usare, per esempio, la porta 53. Una tecnica più efficiente è l'uso di firewall di livello applicativo o sistemi di Intrusion e Prevention (IDP) in grado di identificare gli elementi distintivi del protocollo e quindi bloccarli indipendentemente dalla porta o dalla destinazione usata. Per ovviare anche questo "pericolo" si è introdotto l'uso di tunnel incapsulati all'interno di HTTP o di tunnel cifrati, per esempio SSL/TLS, IPsec o OpenVPN. Se per il primo possiamo ancora usare il metodo di ispezione del traffico per il secondo non c'è alcuna possibilità, se non quella di interrompere il tunnel in qualche punto del tragitto per ispezionarlo o possederne la chiave di cifratura.
Nei prossimi paragrafi verrano presi in considerazione le vulnerabilità e gli aspetti di sicurezza dei seguenti tunnel:

- 6over4: tunnel da host verso un router o da router verso un host;
- 6to4 e 6rd: tunnel da router verso un altro router;
- ISATAP: tunnel interni ad un organizzazione;
- Teredo: tunnel incapsulato all'interno del protocollo UDP e che quindi permette di ovviare al problema del NAT;
- Tunnel Brokers: si usa un server esterno all'organizzazione per automatizzare le procedure di creazione e gestione del tunnel.

2.7.2 6over4

Il protocollo 6over4, definito nel RFC 2529, è un meccanismo semplice per effettuare la transizione usando gli indirizzi multicast di IPv4 come strato virtuale di trasporto. Infatti il più delle volte viene chiamato *IPv4 multicast tunneling* o *Virtual Ethernet*. Considerazioni sulla sicurezza.

- In aggiunta agli attacchi verso IPv6, sono possibili anche attacchi verso IPv4 e sono quindi necessarie politiche di controllo su questo protocollo.
- I router di confine devono applicare il normale filtro di ingresso e uscita degli indirizzi IPv4 e bloccare pacchetti con indirizzo IPv4 unicast con protocollo 41 da fonti sconosciute. Inoltre i router di confine dovrebbero rifiutare i pacchetti IPv4 multicast non legati alle interfacce in cui tale tunnel è attivo.
- I pacchetti IPv6 non incapsulati e aventi hop uguale a 255 non devono essere trattati come se fossero stati generati localmente.
- Nel caso si volesse usare IPsec si consiglia di usarlo nel dominio IPv6 e non in quello IPv4.

2.7.3 6to4 e 6rd

I protocolli 6to4 e 6rd sono meccanismi che permettono di trasportare IPv6 in modalità site-to-site o da un sito verso una rete IPv6 esistente. È uno dei protocolli maggiormente implementati perchè permette con semplicità di trasportare intere reti IPv6 in pacchetti IPv4. Lo svantaggio è la necessità di avere indirizzi IPv4 pubblici fissi nel tempo o, nel qual caso fossero dinamici, la necessità di introdurre una procedura di aggiornamento dei valori di endpoint. Inoltre, come si è già visto in precedenza, il protocollo è facilmente identificabile e di conseguenza bloccabile. Oltre a questi aspetti, negli anni sono stati pubblicati due documenti inerenti la sicurezza dei tunnel 6to4. Il primo, RFC 3056 [11], in cui si ha una panoramica generale della sicurezza del tunnel. Il secondo, RFC 3964 [79], in cui si ha una disamina accurata e completa delle vulnerabilità presenti. Vediamo quali siano tali problematiche.

- I router 6to4 non hanno la possibilità di sapere se un relay è legittimo o meno.
- L'implementazione dei controlli di sicurezza nei router o nei relay del protocollo 6to4 è parzialmente o erroneamente implementata.
- L'architettura 6to4 può essere usata per partecipare ad attacchi DoS, reflected DoS o per nascondere altri attacchi.
- I relay 6to4 possono essere soggetti ad abusi amministrativi.

Una soluzione unica a questi problemi non esiste ma sono state delineate alcune linee guida. Vediamole.

- Nel qual caso si volesse usare IPsec si consiglia di usarlo all'interno di IPv6 e non di IPv4.

- Se lo spoofing dell'indirizzo sorgente di IPv6 è un problema, controllare che i prefissi di IPv4 e l'incapsulamento di IPv6 rispettino le necessarie eccezioni per i router di relay.

- Molti attacchi di spoofing e di DoS derivano dal fatto che i router accettano traffico proveniente da qualsiasi altro router, relay 6to4 o nodo IPv6. Per evitare questo pericolo si dovrebbero applicare delle regole IPv4 sia in ingresso sia in uscita tali da bloccare i nodi non voluti.

- Ogni prefisso 6to4 deve corrispondere ad un indirizzo IPv4 unicast globale. L'indirizzo IPv4 non deve essere privato, di loopback, non specificato, multicast, broadcast, DHCP locale o riservato.

- L'indirizzo IPv6 non deve essere IPv4 compatibile, mappato IPv4, di loopback, non specificato, link-local, site-local o multicast.

Uno dei rischi maggiori, come già visto nell'introduzione, è rappresentato dall'uso di 6to4 tra gli host interni ad una organizzazione e Internet, aggirando di fatto le politiche di sicurezza. Un utente malintenzionato potrebbe realizzare un relay 6to4 avente, per esempio, un indirizzo IP anycast 192.88.99.1 con cui si inserisce fra due estremità del tunnel, realizzando di fatto un attacco man-in-the-middle. Tale attacco è sempre possibile per via che i tunnel di tipo 6to4 sono privi di autenticazione se non per la conoscenza reciproca dell'indirizzo IP. Due sono i meccanismi di protezione possibili per il protocollo 6to4; il primo, l'uso del Unicast RPF nei router di relay e il secondo, l'uso di ACL applicate all'interfaccia del tunnel dei relay.
A differenza di 6to4 il protocollo 6rd ha numerosi vantaggi ed è quindi meno afflitto da vulnerabilità. Innanzitutto, il cliente finale di un tunnel 6rd deve accettare solo pacchetti provenienti da un numero ristretto di router o relay 6rd. Inoltre, essendo le estremità del tunnel appartenenti alle stesse organizzazioni, bisogna solo accertarsi che il traffico provenga solo da indirizzi interni all'infrastruttura.

2.7.4 ISATAP

Il protocollo ISATAP permette il collegamento di nodi IPv6, isolati e attivi all'interno di un network avente solo connettività IPv4, in maniera completamente automatica, creando un tunnel IPv6-in-IPv4. Inoltre, ISATAP implementa IPv6 senza la dipendenza del multicast IPv4, come invece accadeva con il protocollo 6over4. Di contro vi è la necessità di una configurazione più complessa. Un semplice schema è visibile in figura 2.5. Vediamo ora quali siano le problematiche di sicurezza derivanti dall'uso del protocollo ISATAP.

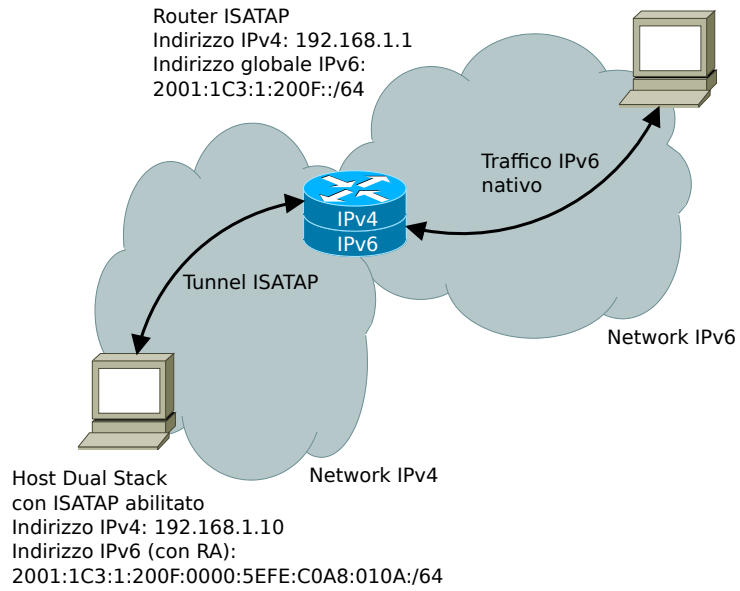


Figura 2.5: Tunnel IPv6 su IPv4 con ISATAP

- ISATAP non presume il funzionamento del multicast IPv6, e quindi non prevede che la procedura di autoconfigurazione e di Neighbor Discovery funzioni, ma conosce a priori la corrispondenza tra indirizzi IPv6 e IPv4 (link layer). Di conseguenza gli amministratori dovranno configurare staticamente l'indirizzo del router predefinito. Così facendo molte delle considerazioni valide per l'autoconfigurazione di IPv6 non sono valide in ISATAP.

- L'uso di IPsec è uguale a quello già presentato per 6to4. È bene usare IPsec all'interno di IPv6.

- Le considerazioni riguardanti le politiche di filtraggio del protocollo 41 rimangono invariate anche per ISATAP. I firewall dovrebbero filtrare correttamente le sorgenti e le destinazioni dei pacchetti aventi protocollo 41.

- I client ISATAP sono vulnerabili ad attacchi provenienti dall'interno dell'infrastruttura. L'attaccante può, in qualsiasi momento, immedesimarsi in un router valido. Per mitigare tale rischio si consiglia di assicurarsi che la tabella *potential router list* (PTR) sia accurata e protetta da eventuali cambiamenti non desiderati.

Altri pericoli possono sorgere nel caso in cui l'attaccante abbia la possibilità di cambiare il record A del server DNS. Per esempio, se cambiasse isatap.example.com e lo dirigesse verso un proprio indirizzo IP, trasformandosi quindi in un router/relay, avrebbe la possibilità di comunicare con tutti gli host abilitati all'uso del protocollo ISATAP presenti nella rete. Si consiglia, quindi, anche nelle organizzazioni che non avessero abilitato il protocollo ISATAP, di configurare staticamente l'indirizzo isatap.miaorganizzazione.it con l'indirizzo 127.0.0.1 così da evitare una sua futura abilitazione e controllare periodicamente che il valore non sia stato modificato. Altri rischi

derivanti dall'uso del protocollo ISATAP sono le possibili iniezioni di traffico malevolo nell'interfaccia del tunnel del router/relay. Una contro misura efficace a questo tipo di attacco è l'uso di appropriate ACL e filtri di tipo Unicast RPF all'interno del router ISATAP. Per concludere, una buona regola generale, come già precisata nei precedenti paragrafi, è la definizione di una regola che blocchi il protocollo 41 verso Internet.

2.7.5 Teredo

Il protocollo Teredo, sviluppato da Microsoft e definito nel RFC 4380[12], supplisce a tutte le limitazione che i precedenti tunnel hanno introdotto. Il protocollo 6over4 ha la necessità del multicast IPv4, 6to4 e 6rd necessitano di un indirizzo IPv4 statico e ISATAP, supplisce ai problemi precedenti, ma non supporta il NAT. Un esempio del suo funzionamento è proposto in figura 2.6.

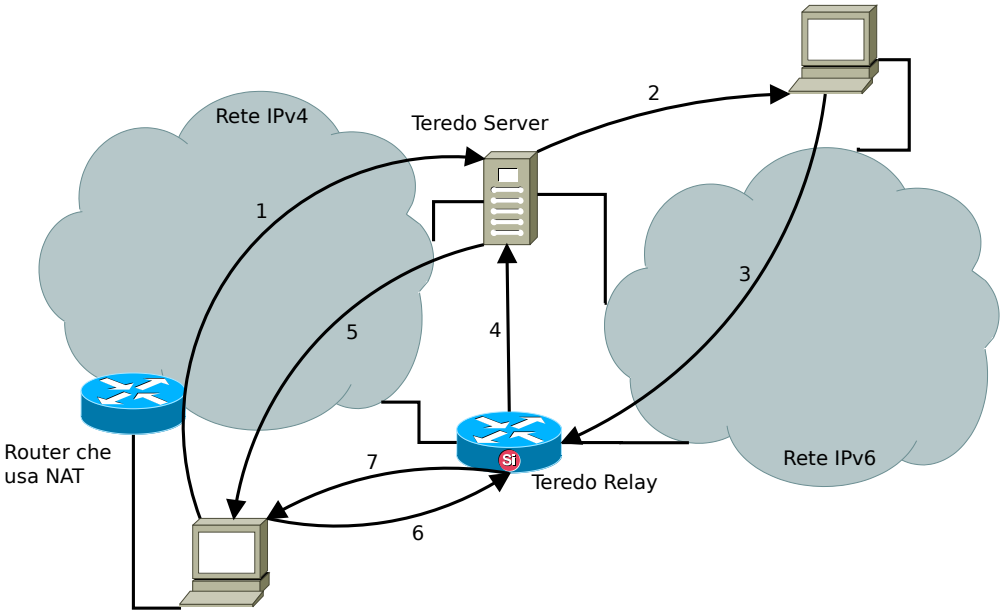


Figura 2.6: Tunnel IPv6 su IPv4 con il protocollo Teredo

La figura 2.7, invece, ci mostra gli header coinvolti nella trasmissione di un canale TCP all'interno del tunnel Teredo. Si può notare fin da subito la complessità introdotta e la difficoltà che i dispositivi di tipo deep inspection dovranno affrontare durante l'analisi del traffico.
Vediamo ora quali siano gli aspetti di sicurezza e le vulnerabilità introdotte dal protocollo Teredo. In particolare, verranno usati come riferimenti alcuni documenti del IETF in versione Draft e il documento RFC 6169 [58].

- Teredo permette l'uso end-to-end di un canale criptato con IPsec, impedendo di fatto la possibilità di ispezionarne il contenuto.

[12]Teredo: Tunneling IPv6 over UDP through Network Address Translation (NAT)

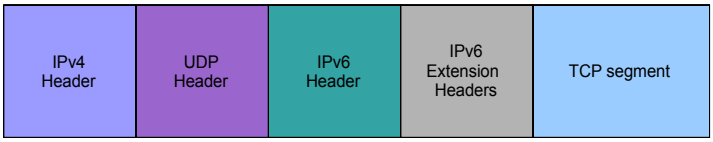


Figura 2.7: Pacchetto inviato attraverso un tunnel Teredo

- Non esiste un metodo efficace per bloccare Teredo e il traffico da esso generato. La comunicazione iniziale verso il server avviene sulla porta UDP 3544 che, come per tutti i servizi, può essere facilmente cambiata.

- Il traffico generato da un Tunnel Teredo non ha un elemento facilmente distinguibile. Negli anni sono stati proposti diversi controlli basati sull'euristica ma quest'ultima non è di facile applicazione e non garantisce una corretta difesa.

- Teredo richiede che almeno una porta UDP non sia filtrata dai dispositivi perimetrali.

- Gli indirizzi IPv6 usati da Teredo sono facilmente riconoscibili e distinguibili dagli indirizzi IPv6 nativi. Infatti, Teredo non prevede l'uso di tecniche come l'indirizzamento privato o il CGA.

- Un utente malintenzionato potrebbe realizzare un server o un relay Teredo contraffatto ed effettuare un attacco di tipo man-in-the-middle. Difendersi da tali attacchi non è sempre semplice. In particolare perchè il più delle volte i dispositivi non sono controllati direttamente dall'organizzazione che stà subendo l'attacco. Una soluzione a questo problema non è ancora presente.

- Una parte estranea all'architettura può impadronirsi di un server Teredo ed effettuare attacchi DoS inviando informazioni errate ai client. Per evitare tali attacchi si consiglia l'uso della mutua autenticazione tra client e server.

- I relay Teredo devono essere provvisti di metodi tali da proteggersi da traffico proveniente da mittenti anonimi i quali potrebbero far parte di attacchi DoS distribuiti.

- Il servizio Teredo non deve scavalcare o disattivare in nessun modo le regole dei filtri di ingresso o di qualsiasi altro servizio di firewall presente nella rete. Tale restrizione è già stata più volte enunciata durante la trattazione dei tunnel ma essendo Teredo un protocollo progettato per ovviare a tutte le precedenti difese bisognerebbe essere molto più cauti nell'applicazione e nelle precauzioni. Si potrebbe imporre l'eliminazione di tutte quelle proprietà non valide o dal comportamento pericoloso che sono incluse in IPv6, per esempio l'header di tipo 0. L'attacco che usa questo tipo di vettore verrà visto nel prossimo capitolo.

- I servizi che sono attivi sull'indirizzo IPv6 del tunnel Teredo possono diventare obiettivi di attacchi raggiungibili da Internet. Questa vulnerabilità può essere mitigata in tre modi. In primo luogo, è possibile limitare alcuni servizi ed accettare solo il traffico proveniente dalla rete locale, ad esempio filtrando gli indirizzi diversi da quelli link-local. Secondo, il firewall locale dovrebbe avere le stesse politiche presenti sul firewall perimetrale. Terzo, si consiglia l'uso di IPsec per proteggere le informazioni dei clienti dai nodi intermedi.

- Una semplice verifica basata sul campo *nonce* descritta nel RFC 4380 fornisce una prima difesa contro attacchi nei quali si cerca di falsificare il server Teredo. In pratica, tale procedura può essere aggirata solo nel caso l'attaccante abbia a disposizione sia il traffico client sia il traffico server.

- Se il client e il server condividono un segreto e concordano un algoritmo di autenticazione, il tunnel Teredo ha una maggiore protezione dagli attacchi. Uno dei modi più probabili con cui l'attaccante possa recuperare il segreto condiviso è catturare il traffico ed effettuare una ricerca della chiave con un approccio di tipo dizionario, in maniera offline. Per ovviare a questo semplice attacco basta che il segreto condiviso tra le due parti contenga una complessità e una imprevedibilità tale che l'attaccante non possa trovarlo nel dizionario e che scoraggi un attacco di tipo bruteforce.

- Un attacco più sofisticato del precedente può essere perpetrato a comunicazioni non protette in una rete NAT. Una procedura con cui potrebbe essere attuato è la seguente:

    1. il client (vittima) prepara un messaggio RS, includendo l'autenticazione;
    2. l'attaccante intercetta la sollecitazione e in qualche modo riesce ad impedirgli di raggiungere il server. Un semplice modo per evitare che tale pacchetto arrivi al server è la realizzazione di un piccolo attacco DoS verso lo stesso;
    3. l'attaccante sostituisce l'indirizzo IPv4 e la porta UDP di origine del pacchetto con un proprio indirizzo e porta e invia tale richiesta modificata al server;
    4. il server riceve il messaggio dall'indirizzo IPv4 dell'attaccante e ne controlla l'autenticazione. Nel caso fosse tutto corretto prepara l'annuncio del router e la firma e lo invia all'utente malintenzionato;
    5. l'attaccante riceve il messaggio e avendo preventivamente salvato i dati della richiesta RS, invia il messaggio al client con i nuovi parametri;
    6. il client riceve tale messaggio modificato e lo usa per configurarsi l'indirizzo e instaurare il tunnel. A questo punto l'attaccate potrà eseguire un attacco DoS o un attacco man-in-the-middle.

- Altri attacchi potrebbero falsificare un relay Teredo. Un utente malintenzionato potrebbe tentare di falsificare l'identità di un altro host IPv6 o di porsi tra un nodo e l'altro. Tali attacchi possono essere perpetrati convincendo il client Teredo che i pacchetti legati ad un determinato host debbano essere inoltrati verso l'indirizzo IPv4 dell'attaccante piuttosto che direttamente al nodo corretto. Questi attacchi sono possibili perchè non esiste nessuna relazione evidente tra l'indirizzo IPv4 di un relay e l'indirizzo IPv6 nativo usato. Un client Teredo non potrà prendere nessuna decisione basata su tali informazioni perchè tali dati non sono presenti nè nel pacchetto IPv4 nè nel header UDP. Il relay è identificato direttamente da un test di connettività a livello IPv6. Per ovviare a questi tipi di attacchi l'unica soluzione è l'uso di IPsec.

- Un client Teredo mantiene una cache di peer utilizzati di recente. È possibile provocare un sovraccarico di questa lista.

- È possibile effettuare un attacco DoS contro la procedura di local peer discovery bubbles nel caso un attaccante riesca ad inviare molti di tali pacchetti verso un client Teredo. L'attacco può essere mitigato con l'uso di filtri, restringendo tale procedura al solo collegamento locale o semplicemente disattivando la procedura.

- Un attaccante può tentare di sopraffare o bloccare un server Teredo. Per ovviare a questo inconveniente è consigliata l'implementazione di un server di backup. Questò però causerà la rienumerazione degli indirizzi assegnati. Un simile attacco verso un relay può portare ad una situazione di irraggiungibilità del servizio Teredo e quindi dell'intera rete IPv6 servita. Inoltre, l'attaccante può anche provare a sopraffare la tabella di stato di un relay Teredo. Bisognerebbe, quindi, essere molto selettivi sui client che il relay debba servire.

- Esistono tre classi di possibili attacchi DoS che tentano di iniettare traffico da punti in cui non è previsto. Il primo utilizza un server Teredo come punto di riflessione in un attacco di Denial of Service. Il secondo utilizza un server Teredo per effettuare un attaco DoS contro i nodi IPv6. Il terzo utilizza i relay Teredo per effettuare l'attacco contro i nodi IPv4. Aggiungendo appropriati filtri si può mitigare efficacemente questi attacchi.

- Un utente malintenzionato può utilizzare in due modi distinti un server Teredo come riflettore in un attacco DoS verso un nodo IPv4. Il primo, costruendosi un messaggio RS ed inviandolo ad un server Teredo con indirizzo sorgente IPv4 della vittima. Così facendo il server Teredo invierà la risposta al nodo bersaglio. Il secondo, costruendosi un messaggio da inviare al server Teredo avente indirizzo IPv6 con il prefisso usato dal server Teredo, con l'indirizzo del server selezionato, con l'indirizzo IPv4 della vittima e con una porta UDP casuale. Come nel caso precedente, per mitigare tali attacchi basta applicare opportuni filtri.

- Un utente malintenzionato può utilizzare un server Teredo per lanciare un attacco DoS contro IPv6 arbitrari. L'attaccante definisce un pacchetto con l'indirizzo IPv6 che si vuole attaccare e lo invia all'indirizzo IPv4 e alla porta UDP del server Teredo. A questo punto il server Teredo lo trasmetterà tale quale all'obiettivo IPv6. Il server dovrebbe verificare se l'indirizzo IPv4 e IPv6 sono coerenti e bisognerebbe cercare di applicare delle regole in ingresso basate sugli indirizzi IPv6.

- Un attaccante dotato di connettività IPv6 potrebbe utilizzare un relay Teredo per un attacco DoS contro una destinazione IPv4. L'utente malintenzionato potrà costruirsi un pacchetto usando il prefisso supportato dal relay Teredo, impostando il flag "cone" ad uno, l'indirizzo IPv4 della vittima ed una porta UDP arbitraria. In primo luogo, il relay non dovrebbe permette ad un aggressore di utilizzare indirizzi IPv4 multicast, broadcast o indirizzi privati. In secondo luogo, bisognerebbe filtrare gli indirizzi IPv6 in ingresso ed usare un firewall stateful.

- Molti dispositivi NPD[13] non hanno la possibilità di controllare il traffico UDP di IPv6.

Un meccanismo di protezione efficace contro le minacce appena citate è quello di disattivare Teredo da tutti i computer della propria organizzazione. Se i sistemi informatici dell'organizzazione sono parte di un dominio Microsoft AD questi non avranno abilitato in modo predefinito Teredo. In aggiunta, un'organizzazione non dovrebbe usare Teredo se non nel caso in cui il nodo avesse configurato un firewall capace di elaborare le richieste IPv6. Una strategia alternativa per impedire l'uso di Teredo è l'uso di IPv6 in dual-stack.
Nel caso l'organizzazione non usi tale protocollo si consiglia di filtrarlo bloccando tutti gli indirizzi provenienti da 2001::/32, porta 3544 UDP, e usare un dispositivo con capacità di deep inspection e capace di individuare il protocollo anche nel qual caso non usi la porta predefinita. Si consiglia, inoltre, di usare una politica in cui si definiscono le sole porte accettate e permesse e automaticamente si blocchino le rimanenti. Per concludere, nel caso si necessiti del protocollo Teredo si consiglia l'installazione di un proprio server all'interno dell'infrastruttura.

2.7.6 Tunnel Broker

I sevizi di Tunnel Broker IPv6 forniscono la connetività dual stack IPv4/IPv6 in una rete IPv4 e permettono di avere connettività IPv6 senza la complessità amministrativa che si avrebbe, per sempio, con il protocollo 6to4. Il suo uso è destinato a reti di piccole dimensioni o a singoli host e si richiede che sia già disponibile e configurato un server broker. Tale infrastruttura potrebbe essere vista come un ISP[14] IPv6 virtuale.

[13] NIST USGv6 Network Protection Device
[14] Internet Service Provider

Informazioni più complete riguardo i diversi tipi di protocolli di Tunnel Broker sono disponibili nel documento RFC 3053 (*IPv6 Tunnel Broker*). La sicurezza dei Tunnel Broker dev'essere considerata in tutti gli aspetti e iterazioni che ne compongono l'infrastruttura: client, server dei tunnel e DNS. Inoltre, la sicurezza e le vulnerabilità sono dipendenti dai protocolli usati e da come questi interagiscono. Vediamo alcune delle problematiche derivate dall'uso dei Tunnel Broker.

- Per i clienti HTTP si consiglia l'uso della crittografia SSL/TLS per proteggere il nome utente e la password durante la fase di autenticazione tra i client e il server. Quando si inviano informazioni riservate, un canale crittografato deve essere utilizzato. Inoltre si consiglia di scegliere quei protocolli che garantiscono la maggiore sicurezza e che non abbiano documenti di crittoanalisi che ne delineano una falla. Si consiglia anche di fornire direttamente una lista dei parametri del tunnel e non renderla scaricabile automaticamente dal client.

- SNMPv1 e SNMPv2 sono protocolli in chiaro e quindi possono fornire informazioni private ad utenti malevoli. SNMPv3 fornisce uno strato crittografico ma non è molto supportato dagli apparati. Si consiglia quindi, nel caso SNMPv3 non fosse disponibile, di incapsulare tali informazioni all'interno di IPsec o di un tunnel SSH[15].

- Per gli aggiornamenti DNS bisognerebbe usare il protocollo definito dal documento RFC 3007 (Secure Domain Name System (DNS) Dynamic Update) o proteggere gli automatismi con protocolli SSH o IPsec.

- Nel caso un host si disconetta da Internet e il relativo indirizzo IPv4 venisse riassegnato ad un client diverso, il server del Tunnel Broker continuerà ad inviare il traffico IPv6 al vecchio indirizzo IPv4.

- Ogni server del Tunnel Broker mantiene lo stato di ogni connessione verso il client. Se usato in modo malevolo, questo comportamento potrebbe portare all'esaurimento delle risorse o ad altri tipi di attacco DoS.

2.7.7 Possibilità d'utilizzo

In conclusione di questo paragrafo vediamo quali siano gli usi, più o meno leciti, dei tunnel. Partiamo dall'obiettivo primario con cui sono stati definiti: una modalità alternativa alla migrazione dal protocollo IPv4 a IPv6, in particolare per quei siti in cui il provider non fornisce direttamente connettività IPv6. Questa particolarità è stata più volte usata sia da privati sia da compagnie in cui si iniziava a sperimentare il nuovo protocollo e a prevederne una migrazione. Per quest'ultima, un buon esempio è riportato nel documento che presenta la migrazione all'interno dell'infrastruttura di Google [6].

[15] Secure shell

Negli anni sono nate parecchie iniziative, per esempio SixXS[16] e Hurricane[17], le quali mettevano a disposizione tunnel broker capaci di connettere host o intere reti a Internet su protocollo IPv6.
Ovviamente tali protocolli possono essere usati per scopi diversi da quelli di migrazione o testing. Per esempio, possono essere usati con il protocollo torrent per ovviare ai filtri imposti dalle organizzazioni o dagli ISP nazionali. Possono essere usati per raggiungere nodi all'interno di reti private, per esempio infrastrutture come Fastweb. Ma possono anche essere usati per ovviare ai limiti imposti per legge dai provider, per esempio per raggiungere siti web bloccati dai filtri nazionali. Possono, inoltre, essere usati per creare delle reti di botnet tali da evitare i controlli normalmente presenti nel protocollo IPv4. In effetti, il problema di sicurezza maggiore di tutti i tunnel è che espongono direttamente la macchina host senza nessun controllo preventivo, come potrebbe essere un sistema perimetrale installato e gestito da un organizzazione.

2.8 Spam

Come abbiamo visto nei paragrafi precedenti i tunnel e più in generale il protocollo IPv6 può essere sfruttato, come già avviene per IPv4, come mezzo di diffusione di malware, di vulnerabilità e di spam. In particolare, quest'ultimo utilizzo presenta una specificità non presente negli altri casi. In IPv4 uno dei metodi più adottati dai server mail è l'uso delle DNSBL[18] in cui si elencano indirizzi o un insieme di indirizzi IP su cui è stato intercettato l'invio di mail di spam. L'uso delle Blacklist in IPv6 è ancora in discussione per via del grande numero di indirizzi bloccati nel caso si introducesse nel filtro una classe (minimo /64) di indirizzi. A questo proposito il RIPE, nel 2010, diffuse alcuni documenti, derivanti dai rilevamenti effettuati sui propri server, in cui ritraeva lo stato di diffusione dello spam nella rete IPv6.
In particolare il primo diagramma circolare 2.8 ci permette di ricavare le seguenti informazioni (i dati presentati escludono i messaggi già bloccati con tecniche di blacklist o greylist):

- nel totale delle mail ricevute il 14% proviene dal protocollo IPv6;
- guardando solo le mail ricevute con IPv4, il 31% è stato classificato come spam;
- guardando solo le mail ricevute con IPv6, il 3.5% è stato classificato come spam.

Nel secondo diagramma 2.9 vediamo la componente IPv6 delle e-mail totali di spam ricevute. Come si può notare soltanto l'1.89% è spam derivante da connessioni IPv6.

[16] `http://www.sixxs.net`
[17] `http://ipv6.he.net`
[18] Domain Name System Blacklists - RFC 5782

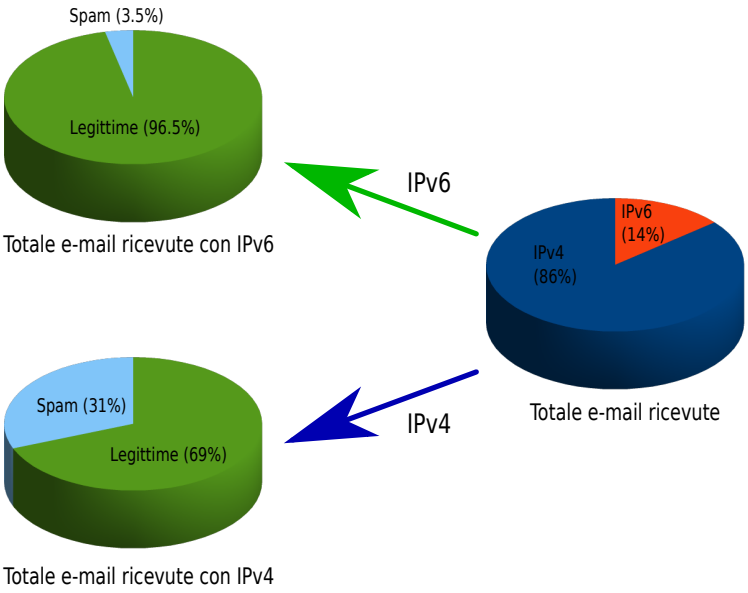


Figura 2.8: Numero di e-mail di spam ricevute in una settimana

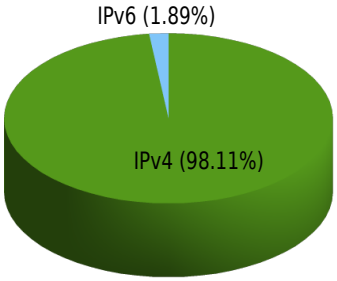


Figura 2.9: Numero totale di e-mail di spam ricevute

Questi dati sono comunque derivanti da statistiche effettuate nel 2010 ed in una sola organizzazione. Un aumento consistente di tali percentuali è sicuramente avvenuto, soprattutto in quanto i grandi provider di posta, per esempio Google, hanno effettuato la migrazione dei server di posta a IPv6 e quindi dando maggiore spazio al nuovo protocollo.
Come si aveva accennato prima dell'introduzione dei diagrammi i problemi che affliggono il controllo dello spam nei server di posta IPv6 sono ancora molti. In particolare, il sistema DNSBL, che insieme al sistema di reputazione degli indirizzi IP bloccano da soli circa il 90% dello spam, ha ancora i seguenti aspetti irrisolti:

- non è ancora chiaro un piano di migrazione delle DNSBL e delle Reputation List verso IPv6;

- non è ancora chiaro un modello di efficienza di questi sistemi a lista con l'indirizzamento IPv6;

- non è ancora chiaro come sia possibile usare questo modello di filtro IP in modo distribuito.

L'unico servizio ad oggi che fornisce DNSBL per IPv6 è il progetto Virbl[19]. Nell'attesa che una nuova soluzione o un nuovo approccio venga proposto bisognerà appoggiarsi sugli algoritmi euristici usati nel valutare la tipologia del messaggio.

[19] `http://virbl.bit.nl`

Capitolo 3

Attacchi al protocollo

Il protocollo IPv6, come si è visto, presenta diversi aspetti che un attaccante può sfruttare per i suoi scopi malevoli. Questo, in parte, è imputabile al periodo in cui IPv6 è stato pensato e definito. A metà degli anni novanta molti degli attacchi oggigiorno conosciuti per IPv4 non erano presenti e si dava per scontato che le reti mantenute dalla stessa organizzazione fossero sicure. Si pensò ugualmente di includere nello standard il protocollo IPsec così da facilitarne l'adozione ed eventualmente usarlo per proteggere i protocolli di gestione della rete. Tale implementazione, però, preclude diverse problematiche che hanno portato negli anni a ricercare soluzioni alternative. Ad oggi sono stati pubblicati diversi documenti atti ad arginare le vulnerabilità riscontrate negli anni. Uno dei protocolli più importanti è SEND che però mantiene una complessità tale da scoraggiare molti produttori e sistemisti ad implementarlo. Queste ed altre mancanze verranno illustrate nel prossimo paragrafo, in cui si elencheranno le opportunità che il protocollo IPv6 mette a disposizione di un eventuale attaccante. Conclusa tale sezione si vedranno quali siano, ad oggi, gli attacchi più diffusi e più interessanti. Fra questi, si disquisirà sugli attacchi derivanti dal campo Hop Limit, dalla frammentazione dei pacchetti, dal Routing Header di tipo 0, dall'abuso del protocollo di Neighbor Discovery e dalla procedura SLAAC. Per concludere si vedranno l'attacco al sistema di difesa RA-Guard e le nuove tecniche per la ricerca degli indirizzi IPv6.

3.1 Opportunità d'attacco

Le opportunità introdotte dal nuovo protocollo di livello 3 sono molteplici e non sempre derivano dallo stesso, ma il più delle volte possono derivare da una non corretta implementazione o semplicemente da una mancanza di controllo. Caso eclatante di questa lacuna è stato il servizio di posta Gmail, il quale non prendeva nota nei log degli accessi provenienti da indirizzi IPv6. Questo permetteva ad un eventuale attaccante di accedere "anonimamente" ad una casella di posta Gmail solamente con l'uso del protocollo IPv6. A questo esempio si aggiunge la mancanza o la limitazione implementativa di protocolli come netflow/sflow o snmp e che quindi limitano il corretto controllo del

traffico presente sulla rete. Il più delle volte, inoltre, non sono presenti circuiti hardware dedicati per gestire i pacchetti IPv6, questo a discapito sia della velocità sia della possibilità di controllo del traffico al crescere del throughput. Si dovrà, quindi, prevedere che nei prossimi anni ci sarà un sostanzioso investimento nella gestione hardware del protocollo e nelle nuove infrastrutture che dovranno supportarlo. In particolar modo, come si è visto nel capitolo precedente, i dispositivi maggiormente interessati a questa transizione saranno gli apparati di firewall e di deep inspection i quali dovranno supplire al grande numero di caratteristiche e di possibilità presenti nel protocollo IPv6. Questa lacuna, per il momento, avvantaggia considerevolmente le possibilità di attacco in una rete IPv6 e, come vedremo in seguito, permette di dirottare con facilità il traffico verso un utente malevolo. Alle carenze di tipo infrastrutturale si aggiungono quelle derivanti dalle implementazioni presenti nei Sistemi Operativi. Infatti, non possiamo definire IPv6 come un protocollo più sicuro di IPv4 solo rifacendoci agli standard ma và visto come un protocollo ancora poco usato e quindi intrinsecamente legato ai bug. Per concludere, i più recenti Sistemi Operativi hanno attivo in modo predefinito il protocollo IPv6 ed è quindi facile per un attaccante fingersi un router IPv6 ed effettuare diversi attacchi senza che l'utente se ne renda conto.

3.2 Hop Limit

Il campo Hop Limit, presente nel header IPv6 può essere usato per carpire alcune informazioni sulla posizione o sul Sistema Operativo usato dal nodo. Vediamo in dettaglio quali possano essere tali usi:

- rilevare il Sistema Operativo usato dal host;
- rilevare un dispositivo remoto;
- localizzare un nodo all'interno della rete o dell'infrastruttura;
- evadere i sistemi di Network Intrusion Detection Systems (NIDS).

3.2.1 Rilevazione di un dispositivo remoto

Così come avviene per IPv4 possiamo, attraverso la rilevazione del valore di Hop Limit, capire la tipologia del Sistema Operativo usato dal mittente e quanti router siano presenti fra di esso e l'attaccante. In particolare, per rilevare il Sistema Operativo si preclude la conoscenza del valore predefinito impostato da ogni sistema. Nel caso si normalizzasse tale valore, l'attaccante non potrebbe rilevare la posizione all'interno dell'infrastruttura e il Sistema Operativo.
Un esempio dei pacchetti ricevuti dall'attaccante può essere il seguente:

pacchetto da un server FTP 2001:db8::1 con Hop Limit uguale a 60
pacchetto da un server HTTP 2001:db8:::2 con Hop Limit uguale a 124

Da queste informazioni possiamo desumere che:

- il server FTP ha impostato il valore di Hop Limit a 64 e che quindi tra l'attaccante e il mittente ci sono 4 router;
- il server HTTP ha impostato il valore di Hop Limit a 128 e che quindi tra l'attaccante e il mittente ci sono 4 router.

Possiamo quindi dedurre che i due server stiano nella stessa rete ma abbiamo Sistemi Operativi diversi.

3.2.2 Localizzazione di un nodo

Essendo i valori di Hop Limit potenze di due, possiamo, in molte situazioni, ricavarci il valore iniziale e quindi desumere quanti nodi intermedi ha attraversato il pacchetto. Questo può essere molto utile per comprendere la posizione del nodo nella rete, in particolar modo nel caso si usasse un algoritmo di triangolazione che, ricevendo più valori di Hop Limit da zone differenti della rete, ricostruisce una buona approssimazione della posizione del nodo selezionato.
Un esempio è rappresentato dalla figura 3.1 in cui è rappresentato il nodo F in base agli Hop Limit presenti nella tabella 3.1.

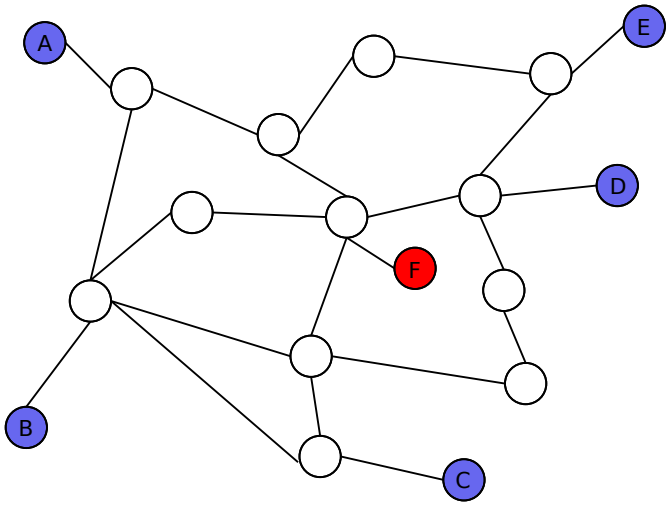


Figura 3.1: Grafo risultante dalla triangolazione degli Hop Limit

3.3 Frammentazione

La frammentazione è uno di quegli aspetti che nella definizione del nuovo protocollo è stato cambiato. In particolare, come si è visto nel primo capitolo, non è più permesso ai nodi intermedi di effettuare la frammentazione ma tale procedura potrà essere effettuata

Sorgente	Hop Limit	Distanza dalla sorgente al nodo F
A	61	4
B	61	4
C	61	4
D	62	3

Tabella 3.1: Valori di Hop Limit e relativa distanza da F

solo dagli host che instaurano la connessione. Inoltre con l'aggiunta degli Extension Header è stata eliminata la possibilità di frammentare l'header di IPv6 e la funzionalità di frammentazione è stata spostata nel header esteso avente valore 44. La figura 3.2 ne visualizza i campi.

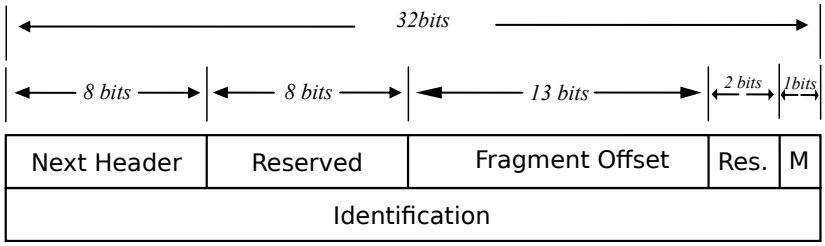


Figura 3.2: Fragmentation Extension Header

Vediamo brevemente il loro significato.

- **Fragmentation Offset**: Offset dei dati che seguono il pacchetto. Il valore è relativo all'inizio della frammentazione del pacchetto originale.
- **M**: Bit che identifica se sono presenti ancora frammenti o meno.
- **Identification**: Insieme all'indirizzo sorgente e all'indirizzo destinatario identificano i frammenti che corrispondono allo stesso flusso.

Un esempio legittimo dell'uso della frammentazione da parte del nodo sorgente è il seguente:

```
$ ping6 -s 1800 2004::1
PING 2004::1(2004::1) 1800 data bytes
1808 bytes from 2004::1: icmp_seq=1 ttl=64 time=0.973 ms

--- 2004::1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.973/0.973/0.973/0.000 ms
```

e i pacchetti avranno le seguenti informazioni:

```
27.232273 IP6 2004::5e26:aff:fe33:7063 > 2004::1: frag (0|1448)
  ICMP6, echo request, seq 1, length 1448
27.232314 IP6 2004::5e26:aff:fe33:7063 > 2004::1: frag (1448|360)
27.233133 IP6 2004::1 > 2004::5e26:aff:fe33:7063: frag (0|1232)
  ICMP6, echo reply, seq 1, length 1232
27.233187 IP6 2004::1 > 2004::5e26:aff:fe33:7063: frag (1232|576)
```

3.3.1 Attacchi

La tabella 3.2 ci mette in evidenza le modalità con cui ogni Sistema Operativo genera gli ID di frammentazione. Vediamo come le versioni di Linux 3.0.0-15, di Solaris 10 e di Windows 7 Home Premium siano facilmente predicibili e quindi vulnerabili ad attacchi già noti nell'universo IPv4. Per esempio, idle-scanning e DoS. Gli acronimi GC e PDC hanno rispettivamente il seguente significato: Global Counter e Per-Destination Counter.

Sistema Operativo	**Algoritmo**
FreeBSD 9.0	Random
NetBSD 5.1	Random
OpenBSD-current	Random (basato su SKIPJACK)
Linux 3.0.0-15	Predicibile (GC iniz. a 0, incr. di +1)
Linux-current	Non predicibile (PDC iniz. ad un valore random)
Solaris 10	Predicibile (PDC, init. to 0)
Windows 7 Home Prem.	Predicibile (GC, iniz. a 0, incr. di +2)

Tabella 3.2: Algoritmi di generazione degli ID di frammentazione suddivisi per Sistema Operativo

Vediamo ora quali siano le problematiche introdotto dalla frammentazione.
Iniziamo col vedere quelle condivise con il protocollo IPv4:

- le operazioni stateful con un protocollo privo di stato possono portare ad eventuali esaurimenti della memoria del kernel se il buffer per ricostruire i frammenti non è propriamente gestito;
- se i valori sono predicibili è possibile effettuare un port scanning di tipo "stealth".

Altri, invece, sono strettamente legati alla nuova implementazione:

- il campo Identification è molto più grande di quello presente in IPv4 e quindi si ha una minore possibilità di collisione;

- l'*overlapping* dei frammenti è stato recentemente proibito (RFC 5722) ma è ancora permesso da molti Sistemi Operativi. Tale possibilità, nel caso fosse implementata, può essere usata come sistema di evasione dalla protezione RA-Guard, come si vedrà meglio in seguito;

- frammenti atomici. Le specifiche IPv6 permettono di avere pacchetti contenenti un Fragment Header anche se questi non sono realmente frammentati in più parti (*Fragment Offset=0, More Fragment=0*). Di norma, questi pacchetti si generano dalla ricezione di un messaggio ICMPv6 di tipo "Packet Too Big" che pubblicizza un "Next-Hop MTU" inferiore a 1280 byte. Molti Sistemi Operativi trattano tale pacchetto come se fosse parte di un flusso frammentato. Si avrà quindi che l'attaccante potrà usare il frammento atomico per sfruttare gli attacchi già descritti per i pacchetti frammentati. La tabella 3.3 illustra quali siano i Sistemi Operativi affetti da tale problema. Come si può vedere sono solo in tre a supportarli correttamente e a trattarli come pacchetti non facenti parte di un messaggio frammentato.

Sistemi Operativi	Supporto ai frammenti atomici	Supporto corretto
FreeBSD 8.0	No	No
FreeBSD 8.2	Sì	No
FreeBSD 9.0	Sì	No
Linux 3.0.0-15	Sì	Sì
NetBSD 5.1	No	No
OpenBSD-current	Sì	Sì
Solaris 11	Sì	Sì
Windows Vista (build 6000)	Sì	No
Windows 7 Home Premium	Sì	No

Tabella 3.3: Supporto dei frammenti atomici in alcuni Sistemi Operativi [43]

3.3.2 IPv6 idle scanning

Come si è visto nel precedente paragrafo, la frammentazione con incremento predicibile potrebbe essere usata per effettuare una scansione anonima verso un nodo prescelto. Per tale scopo verrà usato un nodo definito col nome di *zombie*. Vediamo un esempio in cui i pacchetti, catturati con il comando *tcpdump*, sono generati dal comando ping6, hanno dimensione pari a 1800 byte e hanno l'identificativo di frammentazione incrementato di 1.

```
1. IP6 (hlim 64, next-header Fragment (44) payload length: 1456)
   2004::5e26:aff:fe33:7063 > 2004::1: frag (0x0000007a:0|1448)
   ICMP6, echo request, length 1448, seq 1
```

```
2. IP6 (hlim 64, next-header Fragment (44) payload length: 368)
     2004::5e26:aff:fe33:7063 > 2004::1: frag (0x0000007a:1448|360)
3. IP6 (hlim 64, next-header Fragment (44) payload length: 1240)
     2004::1 > 2004::5e26:aff:fe33:7063: frag (0x4973fb3d:0|1232)
     ICMP6, echo reply, length 1232, seq 1
4. IP6 (hlim 64, next-header Fragment (44) payload length: 584)
     2004::1 > 2004::5e26:aff:fe33:7063: frag (0x4973fb3d:1232|576)
5. IP6 (hlim 64, next-header Fragment (44) payload length: 1456)
     2004::5e26:aff:fe33:7063 > 2004::1: frag (0x0000007b:0|1448)
     ICMP6, echo request, length 1448, seq 2
6. IP6 (hlim 64, next-header Fragment (44) payload length: 368)
     2004::5e26:aff:fe33:7063 > 2004::1: frag (0x0000007b:1448|360)
7. IP6 (hlim 64, next-header Fragment (44) payload length: 1240)
     2004::1 > 2004::5e26:aff:fe33:7063: frag (0x2b4d7741:0|1232)
     ICMP6, echo reply, length 1232, seq 2
8. IP6 (hlim 64, next-header Fragment (44) payload length: 584)
     2004::1 > 2004::5e26:aff:fe33:7063: frag (0x2b4d7741:1232|576)
```

Come possiamo osservare dai valori in grassetto, i pacchetti successivi generano un valore incrementale che permette di effettuare la procedura di *idle scan* illustrata in figura 3.3.

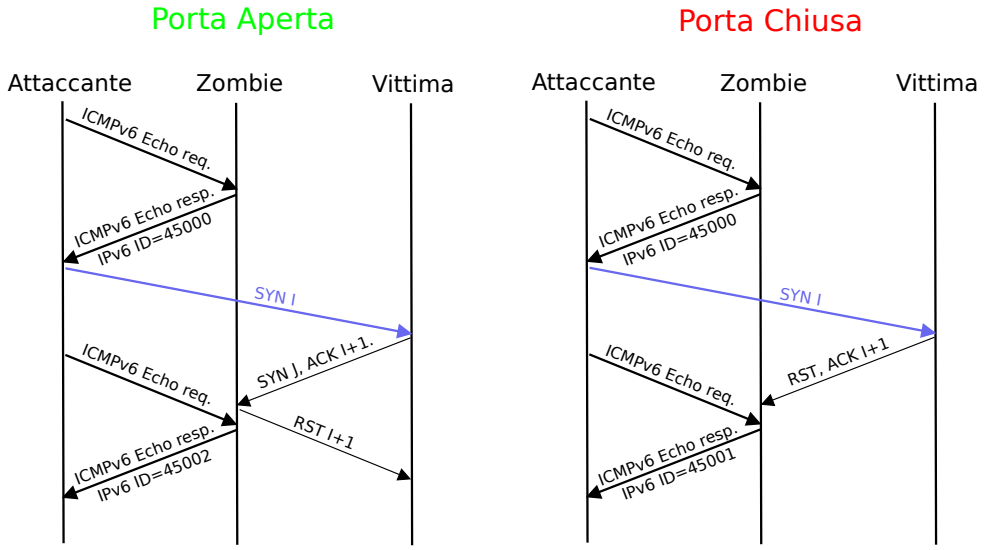


Figura 3.3: Procedura di idle scan

Così facendo l'attaccante potrà enumerare le porte della vittima senza che quest'ultima riceva alcun pacchetto dal suddetto host.

3.3.3 Risvolti pratici

La specifica di IPv6 (RFC 2460) non impone nessun vincolo al numero di estensioni usabili. In particolare, è possibile accodare più estensioni dello stesso tipo visto che gli

stack IPv6 dei Sistemi Operativi lo supportano nativamente. Questo permette di creare pacchetti intrinsecamente complessi e di conseguenza un'elevata difficoltà nel processare ed applicare le corrette regole di filtraggio. Questo aspetto, se da un lato complica la vita ai produttori di apparati di sicurezza perimetrale, dall'altro facilità la possibilità di attacchi. Un esempio interessante di un pacchetto avente più header è rappresentato dalla figura 3.4, nel quale ritroviamo due Destination Options Header e un Fragment Header.

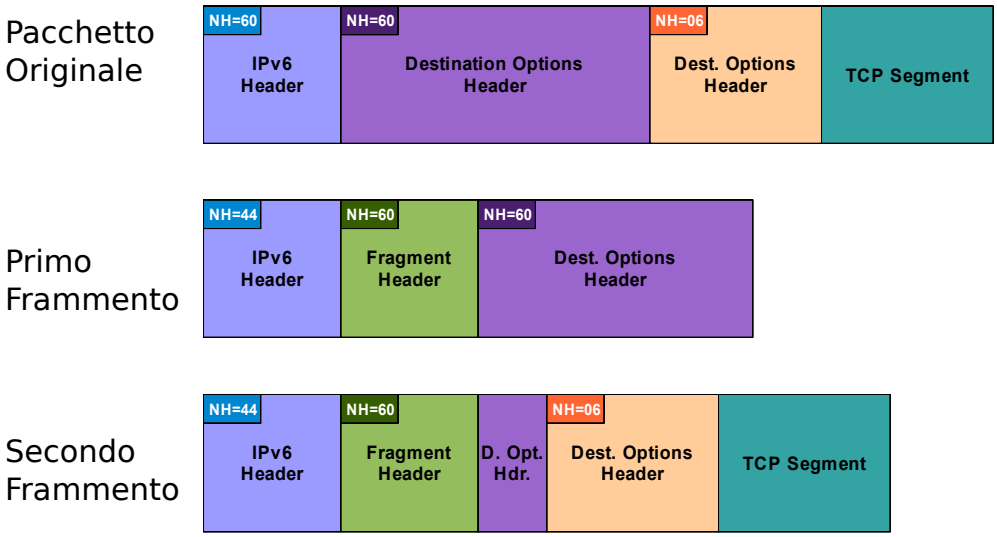


Figura 3.4: Pacchetto con due Destination Option Header e frammentazione

Così facendo, potremmo evadere il controllo sul secondo pacchetto frammentato e quindi aggirare le regole imposte dal firewall. Possibili contromisure sono l'uso di un firewall di tipo stateful che riassembla i pacchetti frammentati prima di procedere con l'analisi. In aggiunta, si potrebbe pensare di imporre delle regole sull'uso degli Extension Header, per esempio:

- bloccare i pacchetti che usano sia la frammentazione sia altre estensioni;
- bloccare i pacchetti con un numero di estensioni superiori a 5.

3.4 Type 0 Routing Header

Nel primo capitolo abbiamo definito l'header esteso di tipo Routing un header che ci permette di dichiarare, come avviene già per *IPv4 lose source routing option*, una sequenza di indirizzi nei quali il pacchetto è obbligato a transitare prima di giungere a destinazione. Questo paragrafo vuole illustrare come questo header possa essere usato per l'attacco Type 0 Routing Header e il motivo per cui tale caratteristica sia stata resa obsoleta dal documento RFC 5095 [1].
Quando il Routing Header è usato per trasmettere un pacchetto da una sorgente ad

una destinazione, la destinazione definita nell'intestazione IPv6 non è quella finale ma soltanto l'indirizzo del prossimo nodo intermedio che deve attraversare. Potremmo usarlo, quindi, per evadere le restrizioni imposte dai sistemi perimetrali e accedere indisturbati ai sistemi interni all'infrastruttura. Questa problematica era già emersa in IPv4 ed è riapparsa tale quale in IPv6. Questo è il primo motivo per cui tale funzionalità è stata resa obsoleta. Il secondo, è la vulnerabilità chiamata *Type 0 routing header* che vedremo qui di seguito.

3.4.1 Routing Extension Header

3.4.1.1 Header

Il Routing Header è definito con il valore di Next Header uguale a 43 e nel caso il campo Type sia uguale a 0 l'estensione verrà chiamata Type 0 Routing Header.
L'estensione avrà i campi presenti nella tabella 3.4 e saranno così definiti.

Next Header	Length	Routing type=0	Segment Left
Reserved			
Address 1			
Address 2			
...			
...			
...			
Address N			

Tabella 3.4: Type 0 Routing Extension Header

- **Next Header** (8 bit): Identifica l'estensione successiva.
- **Length** (8 bit): Numero intero che definisce la lunghezza degli indirizzi in byte (8 byte per N indirizzi). In particolare, per il Type 0 il valore è lungo il doppio del numero degli indirizzi presenti nel header.
- **Routing Type**: Nel nostro caso 0. Sono possibili anche i seguenti valori:
  - tipo 1: definito da Nimrod, vecchio progetto sviluppato dal DARPA[1];
  - tipo 2: usato da MIPv6 e analizzato solo dallo stack MIPv6. Definito per permettere di filtrare il Type 0 Routing Header;
- **Segment Left** (8 bit): Numero di nodi che devono essere ancora visitati prima di giungere a destinazione.
- **Addresses** (128 bit): Indirizzi numerati da 1 a N.

[1]Defense Advanced Research Projects Agency

3.4.1.2 Funzionamento

Vediamo ora come l'header venga processato dai nodi intermedi. La figura 3.5 ne è un esempio.

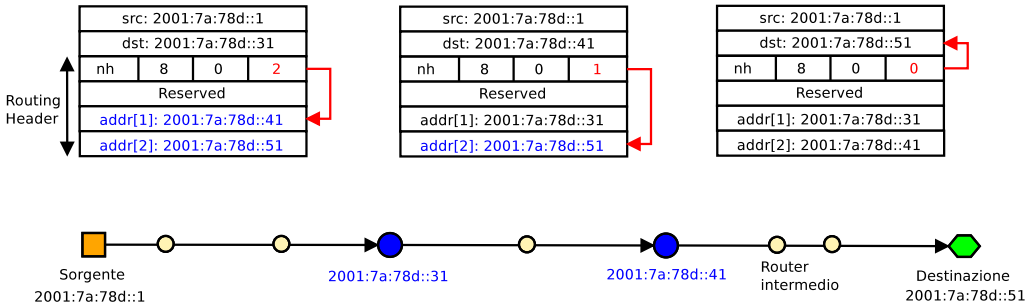


Figura 3.5: Esempio d'uso del Type 0 Routing Extension Header

In sostanza, si avrà che se il campo Segment Left non è zero, e quindi ci sono ancora nodi intermedi da visitare, gli indirizzi sono quelli presenti nel Extension Header. In caso contrario, cioè nel caso in cui si avesse raggiunto il valore 0, il nodo in cui ci si trova è la destinazione finale del pacchetto. Si avrà, quindi, che ad ogni passaggio in un nodo il valore Segment Left verrà decrementato e verrà tolto un indirizzo dall'estensione. Nel 2007 due ricercatori, P. Biondi e A. Ebalard [36], hanno pubblicato una tabella 3.5 nella quale si faceva chiarezza su quali Sistemi Operativi implementassero tale estensione e come essa venga gestita.

OS	Host	Router	Disattivabile?
Linux 2.6	bloccato	processato	no
FreeBSD 6.2	processato	processato	no
NetBSD 3.1	processato	processato	no
OpenBSD 4.0	processato	processato	no
MacOS X	processato	processato	no
Cisco IOS	n/a	processato	sì
Cisco PIX	n/a	bloccato	n/a
Juniper RTR	n/a	processato	no
Netscreen FW	n/a	bloccato	n/a
Windows XP SP2	bloccato	n/a	n/a
Windows VISTA	bloccato	n/a	n/a

Tabella 3.5: Supporto dei Sistemi Operativi del Type 0 Routing Header

3.4.2 Attacchi

L'abuso di tale estensione ci permette di effettuare i seguenti attacchi:

- Network Discovery
- Controllo dei filtri ingress
- Evasione delle regole di firewall
- Attacchi DoS

Vediamoli ora uno per uno.

3.4.2.1 Network Discovery

Vediamo come sia possibile ricavare alcune informazioni riguardo la rete con l'uso del Type 0 Routing. Nei seguenti esempi verrà usato il tool *scapy*[2] che ci permette di creare e manipolare i pacchetti IPv6 a nostro piacimento.

Traceroute
Usando il source Type 0 Routing Header possiamo forzare il traceroute per uno o più punti intermedi così da permetterci di investigare sia eventuali problematiche sia eventuali politiche di routing. Di seguito vediamo il codice con cui possiamo generare tale richiesta e, nel grafo 3.6, quale sia il percorso intrapreso.

```
>>> waypoint = "2001:301:0:8002:203:47ff:fea5:3085"
>>> target = "2001:5f9:4:7:2e0:81ff:fe52:9a6b"
>>> traceroute6(waypoint, minttl=15 ,maxttl=34,
      l4=IPv6ExtHdrRouting(addresses=[target])/
      ICMPv6EchoRequest(data=RandString(7)))
   2001:301:0:8002:203:47ff:fea5:3085        :IER
15 2001:319:2000:5000::92                      3
16 2001:301:0:1c04:230:13ff:feae:5b            3
17 2001:301:0:4800::7800:1                     3
18 2001:301:0:8002:203:47ff:fea5:3085          3
19 2001:301:0:2::6800:1                        3
20 2001:301:0:1c04:20e:39ff:fee3:3400          3
21 2001:301:133::1dec:0                        3
22 2001:301:901:7::18                          3
23 2001:301:0:1800::2914:1                     3
24 2001:319:2000:3002::21                      3
25 2001:319:0:6000::19                         3
26 2001:319:0:2000::cd                         3
27 2001:519:0:2000::196                        3
28 2001:519:0:5000::1e                         3
29 2001:5f9:0:1::3:2                           3
30 2001:5f9:0:1::5:2                           3
31 2001:5f9:0:1::f:1                           3
32 2001:5f9:0:1::14:2                          3
```

[2] http://www.secdev.org/projects/scapy/

```
33  2001:5f9:4:7:2e0:81ff:fe52:9a6b                129
34  2001:5f9:4:7:2e0:81ff:fe52:9a6b                129
(<Traceroute: ICMP:0 UDP:0 TCP:0 Other:20>,
<Unanswered: ICMP:0 UDP:0 TCP:0 Other:0>)
```

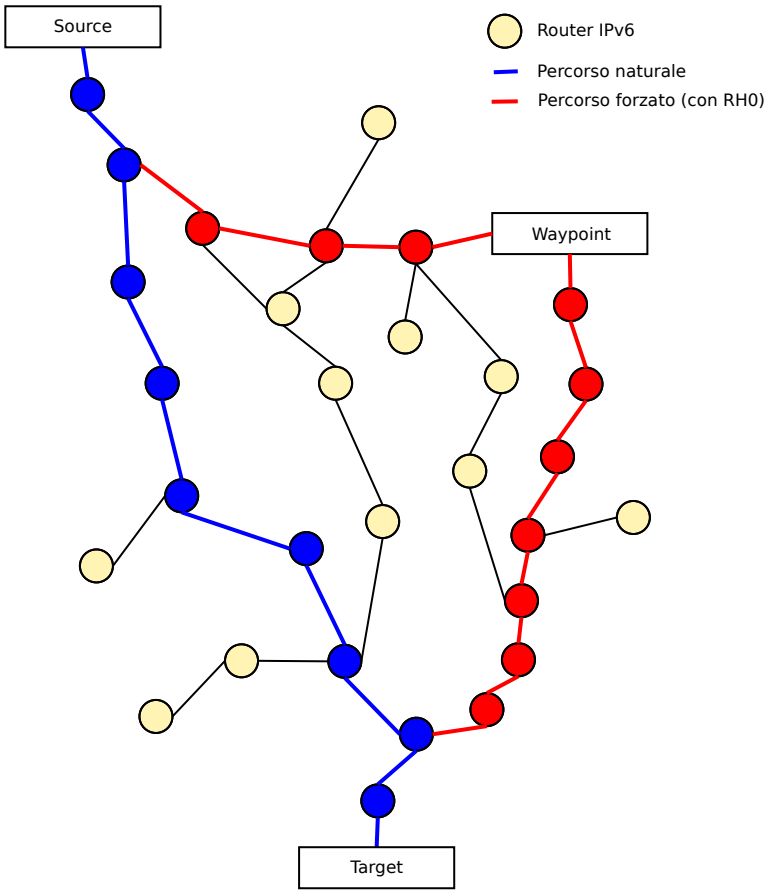


Figura 3.6: Percorso Traceroute

3.4.2.2 Controllo dei filtri ingress

Con l'uso dell'estensione Routing possiamo verificare se esistono dei particolari filtri imposti dall'operatore. Procediamo nella seguente maniera:

1. Troviamo un client che supporti il Type 0 Routing Header.
2. Inviamo un pacchetto "boomerang" con la richiesta che vogliamo verificare.
3. Se il pacchetto ritorna a noi, l'infrastruttura non applica quel tipo di filtro.

Usando scapy, la procedura si traduce in:

```
>>> source="2001:301:1111::1"
>>> dest="2002:301:f2d::3"
>>> sr1(IPv6(src=source  dst=dest)/
    IPv6ExtHdrRouting(addresses=[us])/
    ICMPv6EchoRequest())
```

3.4.2.3 Evasione delle regole di Firewall

Partiamo da alcune considerazioni:

- molti sistemi hanno attivo, in maniera predefinita, il supporto al Routing Header;
- non tutti i firewall processano in maniera corretta il Type 0 Routing Header;
- la granularità dei filtri dipende da molti aspetti (OS, politiche, regole, vincoli infrastrutturali).

Vediamo quindi quali siano i due possibili attacchi:

- possiamo usare il Type 0 Routing Header per nascondere del traffico dal controllo dei sistemi perimetrali;
- possiamo raggiungere host interni all'infrastruttura sfruttando nodi visibili da internet. Si veda la figura 3.7.

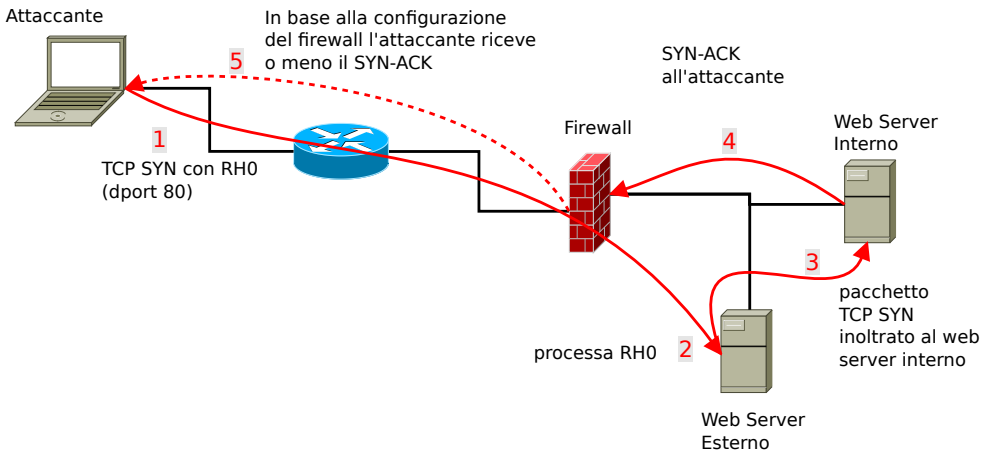


Figura 3.7: Esempio di attacco verso i servizi interni all'infrastruttura

3.4.2.4 DoS

Un aspetto molto importante da tenere sempre presente nella progettazione di un nuovo sistema è l'effetto DoS o DDoS permettendo quindi all'attaccante di bloccare il servizio erogato dal nodo prescelto. Nel 2007 fu pubblicato un avviso di sicurezza[3] che permetteva, attraverso l'uso di un pacchetto Type 0 Routing Header, di bloccare un sistema avente Cisco IOS. Essendo uno dei sistemi più usati per la gestione di Internet ci si rese subito conto dell'impatto che un simile attacco poteva avere sul funzionamento della stessa. Ci si convinse, inoltre, che la sicurezza degli stack IPv6 presenti nei vari Sistemi Operativi non era ancora adeguata e che quindi necessitasse di ulteriore sviluppo.

Effetto Ping Pong
Un altro tipo di attacco DoS, non derivante da vulnerabilità specifiche della piattaforma, è l'uso di un pacchetto Type 0 avente una lunghissima sequenza di host intermedi del tipo

$$A \rightarrow B \rightarrow A \rightarrow B \rightarrow A \rightarrow ....$$

provocando un consumo di traffico e di risorse nocivo per l'attività di rete. In pratica, si cerca di mantenere il più a lungo possibile la permanenza del pacchetto nella rete facendolo rimbalzare tra i due host. Un semplice esempio di tale attacco, sempre con l'ausilio di scapy, è il seguente:

```
>>> addr1 = "2001:4830:ff:12ea::2"
>>> addr2 = "2001:360:1:10::2"
>>> zz=time.time();
    a=sr1(IPv6(dst=addr2, hlim=255)/
    IPv6ExtHdrRouting(addresses=[addr1, addr2]*43)/
    ICMPv6EchoRequest(data="staythere"), verbose=0, timeout=80);
    print "%.2f seconds" % (time.time() - zz)
32.29 seconds
```

In una rete con 4 Mbit/s di banda in upload si avrà che l'esecuzione impiegherà circa 32 secondi e avrà prodotto circa 16 MByte di traffico aggiuntivo. Inoltre, se il traffico fosse transitato in un tunnel, per esempio 6to4, questo tipo di attacco avrebbe avuto ripercussioni anche sul traffico IPv4 sottostante.

3.4.2.5 Conclusione

In conclusione possiamo affermare che:

[3] http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20070124-IOS-IPv6

- l'estensione Type 0 Routing Header non ha molta utilità pratica se non per gli attacchi visti precedentemente;
- gli effetti collaterali per l'infrastruttura sono il più delle volte importanti;
- gli ideatori di IPv6 non avevano compreso le problematiche di sicurezza di IPv4;
- gli sviluppatori non hanno da subito preso in considerazione le problematiche già presenti in IP4.

Sarebbe quindi consigliato disattivare o filtrare in maniera corretta i pacchetti inerenti al Routing Header, qualora fosse ancora attivo. Nel caso si necessiti di tale funzionalità si consiglia l'uso del tipo 2 su infrastruttura Mobile IPv6.

3.5 Neighbor Discovery

3.5.1 Risoluzione dell'indirizzo in IPv6

In IPv6, come si è appreso dal primo capitolo, è stato sostituito il meccanismo ARP presente in IPv4 con il protocollo di Neighbor Discovery che, con l'uso di due specifici pacchetti ICMPv6, permette di scoprire l'indirizzo MAC del nodo ricercato. I pacchetti implicati nella ricerca dell'indirizzo di livello 2 sono Neighbor Solicitation (NS) e Neighbor Advertisement (NA) e un esempio del loro utilizzo è il seguente:

1. L'host A invia un pacchetto NS con la seguente richiesta: *Chi ha l'indirizzo IPv6 corrispondente a 2001:db8::1?*
2. L'host B, proprietario di quell'indirizzo, risponde con un pacchetto NA con le seguenti informazioni: *Sono io l'indirizzo IPv6 e il mio indirizzo MAC è 06:09:12:cf:db:55*.
3. L'host A salva l'informazione ricevuta nella Neighbor Cache per un certo periodo. Nello stesso modo in cui avveniva per la cache ARP di IPv4.
4. A questo punto l'host A può inviare i pacchetti al host B.

Un esempio di tale procedura è visibile dall'output del comando *tcpdump* appena si invia un pacchetto di echo request.

```
$ ping6 2004::1

42.086657 2004::20c:29ff:fe49:ebdd > ff02::1:ff00:1: icmp6: neighborsol:
    who has 2004::1(src lladdr: 00:0c:29:49:eb:dd) (len 32, hlim 255)
42.087654 2004::1 > 2004::20c:29ff:fe49:ebdd: icmp6: neighbor adv:
    tgt is 2004::1(RSO)(tgt lladdr: 00:0c:29:c0:97:ae) (len 32, hlim 255)
42.089147 2004::20c:29ff:fe49:ebdd > 2004::1: icmp6: echo request
  (len 16, hlim 64)
```

```
42.089415 2004::1 > 2004::20c:29ff:fe49:ebdd: icmp6: echo reply
  (len 16, hlim 64)
```

3.5.1.1 Messaggio di Neighbor Solicitation

Il primo messaggio che andiamo ad analizzare è il Neighbor Solicitation che, come abbiamo visto, ci permette di richiedere l'indirizzo MAC di un nodo presente nella rete. È identificato dal campo Type uguale a 135, Code uguale a 0 e l'unica opzione ammessa è Source Link-layer address. La tabella 3.6 ne rappresenta la struttura.

Type (8 bit)	Code (8 bit)	Checksum (16 bit)
Reserved (32 bit)		
Target Address (N bit)		
Options (N bit)		

Tabella 3.6: Struttura del messaggio Neighbor Solicitation

Il Target Address è l'indirizzo IPv6 del richiedente e non può essere di tipo multicast.

3.5.1.2 Messaggio di Neighbor Advertisement

La risposta al messaggio di Neighbor Solicitation è il messaggio Neighbor Advertisement, rappresentato in tabella 3.7. Il messaggio conterrà il valore link-layer corrispondente all'indirizzo IPv6 richiesto. È definito dal campo Type uguale a 136, dal valore Code uguale a 0 e l'unica opzione ammessa è Target Link-layer address.

Type (8 bit)		Code (8 bit)	Checksum (16 bit)
R	S	O	Reserved (29 bit)
Target Address (N bit)			
Options (N bit)			

Tabella 3.7: Struttura del messaggio Neighbor Advertisement

Vediamo brevemente i campi non già definiti durante la trattazione di ICMPv6.

- **R**: Router flag. Se impostato ad 1 il mittente è un router. È usato da Neighbor Unreachability Detection per rilevare se un router è diventato un host.

- **S**: Solicited flag. Se impostato ad 1 indica che l'annuncio è in risposta ad un messaggio di Neighbor Solicitation dall'indirizzo di destinazione.

- **O**: Override flag. Se impostato ad 1 indica che il valore presente nel messaggio dovrebbe sovrascrivere il valore presente nella cache.

- **Target Address**: In caso di annuncio sollecitato da un messaggio NS il campo contiene il valore Target Address del messaggio di richiesta. In caso non sia stato richiesto, contiene l'indirizzo link-layer di colui che l'ha cambiato. Non dev'essere un indirizzo multicast.

3.5.1.3 Opzione per l'indirizzo Link-Layer

Vediamo ora come sia strutturata l'opzione nella quale viene definito l'indirizzo link-layer. La tabella 3.8 definisce i campi presenti all'interno dell'opzione.

Type (8 bit)	Length (8 bit)	... Link-layer Address (N bit) ...

Tabella 3.8: Struttura dell'opzione Link-Layer Address

I campi sono così definiti:

- **Type**: Identifica il tipo di indirizzo ed è così specificato:
  - 1 : per l'indirizzo Source Link-layer;
  - 2 : per l'indirizzo Target Link-layer.

- **Length**: Lunghezza complessiva dell'opzione espressa in unità di 8 ottetti. Il valore 0 non è valido e il pacchetto dev'essere immediatamente scartato. Per esempio, il valore 1 identifica un indirizzo IEEE 802.

- **Link-Layer Address**: Indirizzo di livello 2.

3.5.1.4 Neighbor Cache

Ogni host mantiene una propria cache in cui avrà a disposizione, per un lasso di tempo, gli indirizzi link-layer ricevuti attraverso messaggi NA. Ogni voce, composta dall'indirizzo IPv6 e dall'indirizzo link-layer, potrà assumere i seguenti stati:

Stato	Semantica
Incomplete	la procedura di Address Resolution è in esecuzione
Reachable	il vicino è raggiungibile
Stale	non si è a conoscenza se il vicino è raggiungibile
Delay	non si è a conoscenza se il vicino è raggiungibile (in attesa di indicazioni)
Probe	non si è a conoscenza se il vicino è raggiungibile (NS inviato)

Un esempio del contenuto della tabella di Neighbor Cache in un sistema FreeBSD è il sequente:

```
% ndp -a

Neighbor                        Linklayer Address  Netif Expire    S F
2004:1::f8dd:347d:8fd8:1d2c     0:c:29:49:eb:e7    em1   permanent R
fe80::20c:29ff:fec0:97b8%em1    0:c:29:c0:97:b8    em1   23h48m16s S R
2004:1::20c:29ff:fe49:ebe7      0:c:29:49:eb:e7    em1   permanent R
fe80::20c:29ff:fe49:ebe7%em1    0:c:29:49:eb:e7    em1   permanent R
2004::1                         0:c:29:c0:97:ae    em0   23h49m27s S R
2004::20c:29ff:fe49:ebdd        0:c:29:49:eb:dd    em0   permanent R
fe80::20c:29ff:fe49:ebdd%em0    0:c:29:49:eb:dd    em0   permanent R
fe80::20c:29ff:fec0:97ae%em0    0:c:29:c0:97:ae    em0   23h48m16s S R
2004::d13e:2428:bae7:5605       0:c:29:49:eb:dd    em0   permanent R
```

3.5.2 Attacchi

3.5.2.1 Neighbor Cache Poisoning

Come avviene già ora con IPv4 è possibile perpetrare un attacco di avvelenamento della cache che ogni host mantiene.
L'attaccante rimane in ascolto di una richiesta di Neighbor Solicitation proveniente dal nodo da attaccare. Appena la riceve, risponde con un messaggio di Neighbor Advertisement contenente un indirizzo link-layer. Questo attacco può portare a due risvolti. Il primo, può essere sfruttato per provocare un attacco DoS semplicemente usando un indirizzo di link-layer fasullo e non presente sulla rete. Nel secondo, ritorniamo l'indirizzo di link-layer uguale a quello presente nel computer dell'attaccante. In questo modo si ha perpetrato un attacco man-in-the-middle come quello visibile in figura 3.8.

3.5.2.2 Pubblicazione di un indirizzo speciale di link-layer

Si potrebbe pensare di inviare un messaggio Neighbor Advertisement (si ricorda che può essere inviato senza la necessità di un messaggio NS precedente, per esempio nel caso di aggiornamento dell'indirizzo link-layer) avente un indirizzo di link-layer non rappresentante un singolo nodo. In Ethernet, per esempio, potremmo usare i seguenti indirizzi:

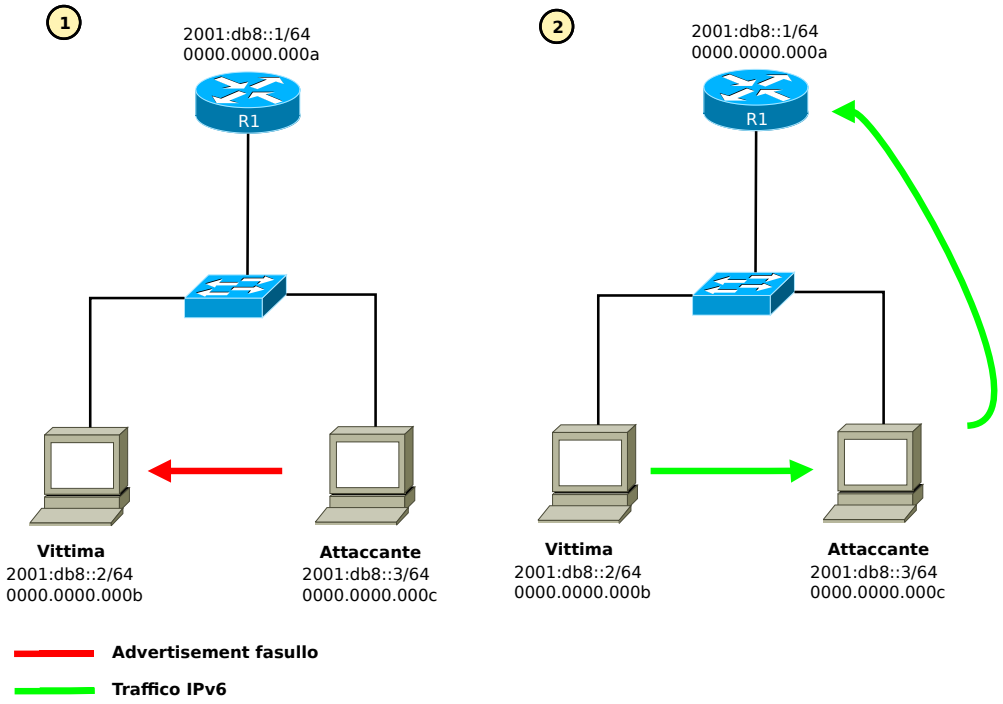


Figura 3.8: Attaco man-in-the-middle con l'uso del messaggio NA

- ff:ff:ff:ff:ff:ff : indirizzo broadcast;
- 33:33:00:00:01 : indirizzo multicast.

Nel caso il nodo fosse un router, l'invio di un messaggio di Neighbor Solicitation produrrebbe un forwarding loop.

3.5.2.3 Neighbor Cache Overflow

Alcune implementazioni dello stack IPv6 non impongono alcun limite al numero di valori che possono essere inseriti nella Neighbor Cache. Questa lacuna può essere usata da un attaccante per perpetrare un attacco DoS, attraverso i seguenti passi:

- l'invio di un numero molto elevato di Neighbor Solicitation con un indirizzo Source link-layer;
- per ogni pacchetto ricevuto, la vittima inserirà nella cache i nuovi valori;
- se i pacchetti in ricezione sono aggiunti più velocemente di quelli cancellati si avrà che l'attacco esaurirà velocemente la memoria messa a disposizione per questa struttura.

3.6 SLAAC

In questo paragrafo vedremo quali siano i possibili attacchi nel caso in cui la rete abbia abilitata l'autoconfigurazione. Si ricorda che l'implementazione di SLAAC, non la sua attivazione, è obbligatoria per tutti i sistemi che utilizzino IPv6.

3.6.1 Protocollo

Iniziamo col vedere quali siano gli stati in cui si può giungere durante la fase di autoconfigurazione, rappresentati nel diagramma illustrato in figura 3.9.

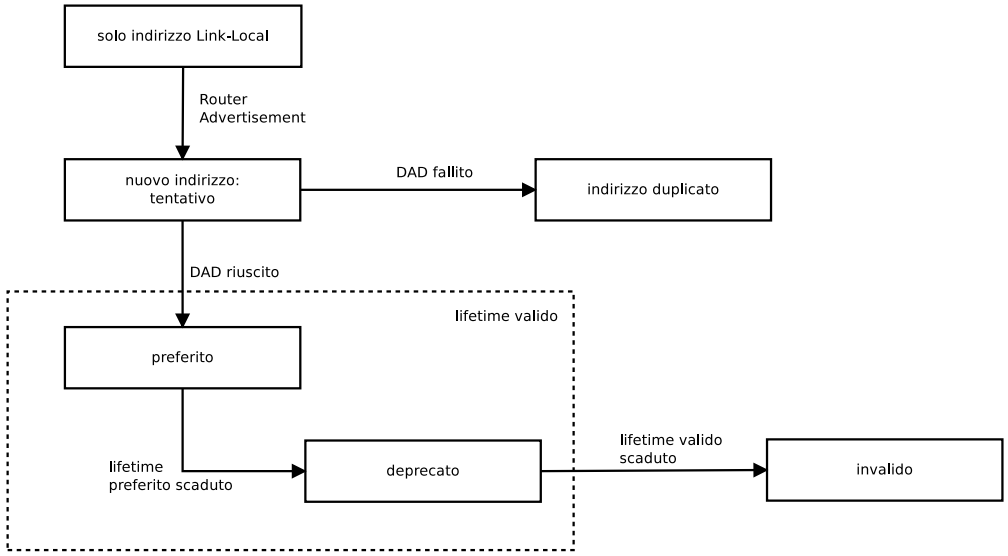


Figura 3.9: Diagramma a stati del protocollo SLAAC

Un nodo che si collega ad una nuova rete si trova nella situazione di definire un nuovo indirizzo IPv6 partendo dal solo indirizzo link-local. In breve, il protocollo SLAAC, funziona nella seguente maniera:

1. l'host configura l'indirizzo link-local dell'interfaccia prescelta. Normalmente questo passo viene effettuato all'attivazione dell'interfaccia;

2. l'host controlla che l'indirizzo sia unico attraverso il messaggio Duplicate Address Detection (DAD);

3. invia un messaggio di Router Solicitation;

4. quando un messaggio di Router Advertisement viene ricevuto, l'host si configura l'indirizzo con il prefisso contenuto nel messaggio;

5. controlla se l'indirizzo appena configurato è unico sulla rete locale (procedura DAD);

6. se l'indirizzo è unico nella sottorete verrà mantenuto fino alla scadenza del valore di *lifetime*.

3.6.1.1 Router Solicitation

Vediamo ora come sia strutturato il messaggio di Router Solicitation e quale sia la sua utilità. Come già accennato, tale messaggio viene usato durante la fase di auto-configurazione dell'indirizzo. In particolare, viene inviato per effettuare una richiesta immediata e senza attendere il prossimo messaggio RA che normalmente viene inviato periodicamente dal router.
Il messaggio è definito all'interno di ICMPv6 con il tipo 133 e il codice 0 ed è strutturato nella seguente maniera.

Type (8 bit)	Code (8 bit)	Checksum (16 bit)
Reserved		
Options ...		

L'unica opzione ammessa all'interno del messaggio è l'indirizzo Source Link-Local.

3.6.1.2 Router Advertisement

Il messaggio Router Advertisement è uno dei messaggi più importanti di IPv6. Come più volte si è definito, RA serve ad annunciare i parametri di configurazione della rete locale. Vedremo, alla conclusione di questo paragrafo, come tale messaggio possa essere sfruttato per dirottare il traffico o per causare seri problemi all'infrastruttura.
Il messaggio Router Advertisement è definito, all'interno di ICMPv6, con il tipo 134 e il codice 0 ed è così composto.

Type (8 bit)	Code (8 bit)					Checksum (16 bit)
Cur. Hop Limit	M	O	H	Prf	Res.	Router Lifetime
Reachable Time						
Retrans Timer						
Options ...						

Le opzioni che possono essere incluse all'interno del messaggio Router Advertisement sono le seguenti:

- Indirizzo Source Link-Local

- Informazioni sul prefisso
- MTU
- Router Information
- DNS server ricorsivo

Un'opzione importante, in particolare per possibili attacchi, è il Router Information che permette di definire specifici instradamenti aventi priorità diverse.

3.6.2 Attacchi

Vediamo ora quali siano i possibili attacchi perpetrabili in una rete in cui non è attivo alcun sistema di RA-Guard. Tale sistema verrà illustrato nel paragrafo 3.7 in cui verranno presentate le problematiche, le soluzioni e i possibili attacchi a tale difesa.

- **Disabilitare un router esistente**
  Possiamo impersonare un router esistente ed inviare un messaggio di RA con il Router Lifetime uguale a 0. Come risultato si avrà che la vittima cancellerà il router dalla tabella di routing.

- **Usare DAD per effettuare un Denial of Service**
  Possiamo sfruttare la procedura di Duplicate Address Detection per impedire ad uno o a qualsiasi host di configurarsi. L'attacco è molto semplice: mi metto in ascolto di tutti i messaggi di Neighbor Solicitation con l'indirizzo sorgente configurato sull'indirizzo IPv6 imprecisato (::). Quando verrà ricevuto un messaggio l'attaccante risponderà con un messaggio di Neighbor Advertisement. In questo modo l'host che stava effettuando la procedura di ND si accorgerà che l'indirizzo non è unico e quindi non potrà usarlo. Iterando questa procedura si produrrà un DoS tale da impedire a tutti i nodi di collegarsi alla rete.

- **Pubblicazione di parametri di rete maligni**
  Un attaccante potrebbe pubblicare, all'interno di messaggi RA, valori di rete non veritieri e che quindi potrebbero portare ad una degradazione della rete o ad un completo Denial of Service. Per esempio, potrebbe pubblicare un valore di MTU basso così da comportare un aumento degli header o pubblicare un valore molto basso di Current Hop Limit così da far sì che i pacchetti vengano scartati nei router intermedi.

- **Router Advertisements flooding**
  Nel caso si inviasse un gran numero di messaggi Router Advertisement casuali gli host sarebbero costretti ad aggiornare continuamente le informazioni di rete e di conseguenza si potrebbe incorrere in un consumo eccessivo di risorse e, nel peggiore dei casi, ad un blocco del sistema.

- **Dual Stack**
  Si può pensare di propagare un messaggio Router Advertisement così da indurre i sistemi aventi la modalità SLAAC attiva ad autoconfigurarsi. In questo modo si avrebbe la possibilità di effettuare una enumerazione dei servizi in ascolto sull'indirizzo IPv6 e normalmente privi di difese adeguate.

3.7 RA-Guard

In questo paragrafo verranno presentati, da prima, un metodo con cui è possibile fingersi il router predefinito della rete e in seguito come sia stato sviluppato un sistema per difendersi da tale attacco, chiamato RA-Guard. In conclusione vedremo quali siano le tecniche che possono invece aggirare questo nuovo tipo di protezione.

3.7.1 Attacco

Ogni host che ha abilitato la modalità di autoconfigurazione cambierà il suo stato alla ricezione di un messaggio di Router Advertisement. Quest'ultimo, nel caso non siano presenti particolari filtri, potrà essere inviato da qualsiasi altro nodo sulla rete e sarà ritenuto valido. Questo, come si può ben immaginare, permette ad un utente malevolo di generare un Router Advertisement falso tale da provocare un DoS o una redirezione del traffico della vittima verso l'attaccante, diventando quindi il router predefinito.
Un modo semplice per attuare tale attacco, oltre alla configurazione della parte di forwarding, è l'uso iterato del seguente codice prodotto con scapy.

```
# scapy
>>> a = IPv6()
>>> a.dst = "ff02::1"
>>> b = ICMPv6ND_RA()
>>> c = ICMPv6NDOptSrcLLAddr()
>>> # set local mac address
>>> c.lladdr = "00:50:56:24:3b:c0"
>>> d = ICMPv6NDOptMTU()
>>> e = ICMPv6NDOptPrefixInfo()
>>> e.prefixlen = 64
>>> e.prefix = "cc5f::"
>>> send(a/b/c/d/e)
```

Il pacchetto inviato sarà il seguente.

```
# tcpdump -i eth0 ip6

listening on eth0, link-type EN10MB (Ethernet),
  capture size 96 bytes
09:49:47.976530 IP6 fe80::239:20ff:fe52:7666 > ff02::1:
  ICMP6, router advertisement, length 64
```

Inviando periodicamente questo messaggio o inviandolo in risposta a un messaggio di Router Solicitation avremo che il nodo si autoconfigurerà con il prefisso specificato all'interno del messaggio (cc5f::) e il router predefinito sarà uguale all'indirizzo link-local corrispondente al mac address dell'attaccante. Così facendo, tutto il traffico destinato ad indirizzi diversi da quello della rete locale sarà inviato all'attaccante che poi lo propagherà al router corretto.
Per automatizzare l'intera procedura possiamo usare il seguente tool presente all'interno della suite thc-ipv6[4].

```
# fake_router6 eth0 2001:db8:dead:beef::/64
```

Con il comando *fake_router6* tutti i nodi con l'autoconfigurazione attiva riceveranno un messaggio di Router Advertisement con il prefisso specificato e l'attaccante comparirà nella tabella di routing come router predefinito e avente massima priorità (valore 01).

3.7.2 Difesa

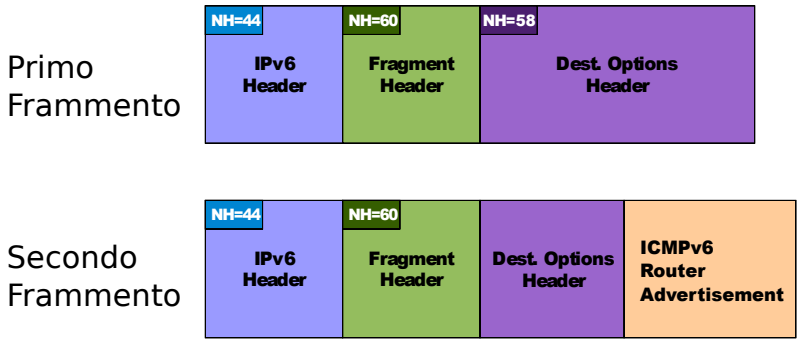
In alternativa al già citato SEND, il documento RFC 6105 [62] specifica delle regole atte ad eliminare gli attacchi sopra citati, in particolare quelli definiti nel RFC 6104 [16]. L'insieme di tali protezioni è chiamato RA-Guard. Il concetto alla base di tale documento è molto simile a quello già presente in IPv4, chiamato *DHCP snooping*, e già implementato in molti switch. Tale protezione prevede di filtrare i messaggi ICMPv6 di tipo Router Advertisement a livello 2 in base a differenti criteri. Uno dei criteri più semplici da adottare è permettere l'invio del traffico contenete messaggi RA dalle sole porte fisiche aventi un router fidato. Questo, ovviamente, richiede che il device sia in grado di identificare correttamente i messaggi ICMPv6 di tipo Router Advertisement ed essendo un apparato di livello 2 e vista la complessità introdotta con gli Extension Header non è sicuramente facile implementare un algoritmo atto all'ispezione completa del pacchetto.

3.7.3 Evasione da RA-Guard

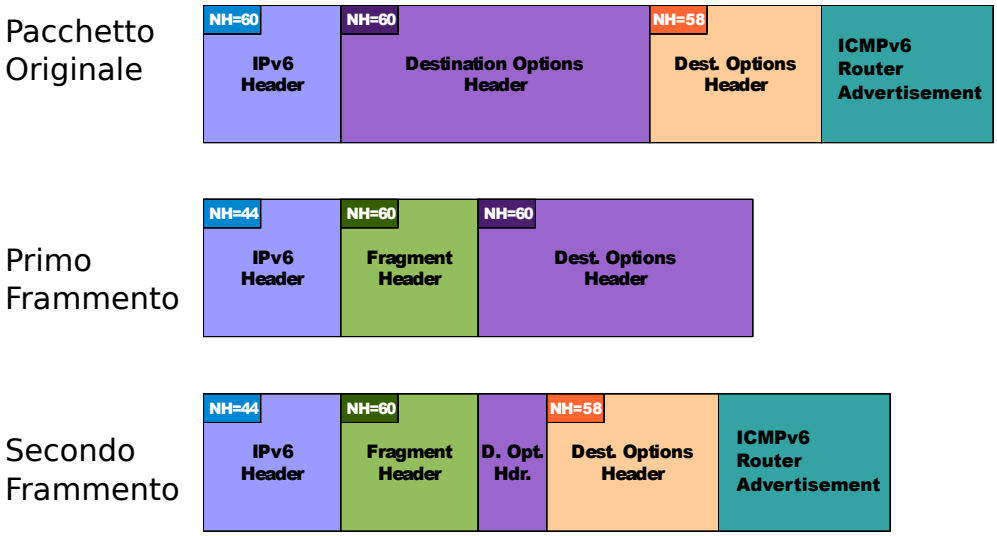
Le implementazione più semplici di RA-Guard (es. Cisco) cercano di identificare il messaggio ICMPv6 di tipo Router Advertisement controllando solo il campo Next Header del header IPv6, piuttosto che controllare la presenza di più Extension Header. Questo, come si può immaginare, è facilmente aggirabile aggiungendo un header prima del header ICMPv6. Non essendoci alcuna controindicazione e visto che tutte le implementazioni lo supportano, si ha che RA-Guard può essere facilmente raggirato. Un esempio del pacchetto di tale attacco è visibile nella seguente figura.

[4] http://www.thc.org/thc-ipv6

Questo tipo di attacco può essere semplicemente eliminato controllando tutta la catena di header presenti nel pacchetto. Ovviamente questo ne diminuirà la velocità di processamento.
A fronte di questa modifica si pensò di raggirare la protezione usando un attacco che facesse uso della frammentazione IPv6. L'idea di base è di frammentare un messaggio ICMPv6 RA in almeno due frammenti tale da non permettere all'apparato di livello 2 di riassemblare il pacchetto e quindi di applicare i dovuti filtri. Un esempio di come potrebbero essere i pacchetti frammentati è il seguente.



Si tenga presente che la lunghezza del Destination Options Header ("Hdr Ext Len") è presente solo nel primo frammento ma non nel secondo. Di conseguenza è impossibile per l'apparato che analizza il secondo frammento trovare l'header ICMPv6 non conoscendone la posizione.
Si potrebbe inoltre far leva sull'uso del Fragmentation Header e del Destination Option Header nascondendo ulteriormente il tipo e il contenuto nel messaggio ICMPv6 che stà inviando. Uno switch potrebbe accorgersi, attraverso il campo Next Header uguale a 58 presente nel primo frammento, che si stà inviando un messaggio ICMPv6. Si potrebbe, quindi, usare due Destination Options Header in due frammenti. Di seguito un'immagine rappresentativa dell'attacco.



In questa variante il campo Next Header del primo frammento avrà valore pari a 60 e quindi sarà impossibile per l'apparato di livello 2 identificare che si stia mandando un messaggio ICMPv6. Per il secondo frammento vale lo stesso ragionamento posto per l'attacco precedente. Lo switch non potrà analizzarlo perchè non ne conosce la dimensione.
Inoltre, tali tecniche potrebbero essere usate per eludere software adibiti alla sorveglianza dei messaggi di Neighbor Discovery (es. NDPMon [5]). Questi attacchi sono stati presentati nel documento *IPv6 Router Advertisement Guard (RA-Guard) Evasion* [39] da Fernando Gont il quale, in seguito, ha proposto alcuni rimedi a tali attacchi. Una delle idee prospettate da Gont per eliminare tali attacchi è vietare l'uso degli Extension Header con i messaggi di Neighbor Discovery. Per concludere questo argomento, Marc Heuse propose, in aggiunta a quelle precedenti, l'uso dei frammenti di overlapping. Questa tecnica non è più realizzabile sui moderni Sistemi Operativi, come già precisato, ma potrebbe essere usata su sistemi embedded in cui l'aggiornamento del kernel è molto raro.

3.8 Discovery

Uno degli ambiti più attivi negli ultimi anni è quello relativo alle tecniche atte a ricavare gli indirizzi attivi nella rete locale, così come già accade per IPv4. Come abbiamo più volte visto, lo spazio di indirizzamento di una classe ha dimensione pari a 2^{64} e quindi diventa impraticabile effettuare la classica scansione per ogni singolo indirizzo. A tale proposito nel 2005, Marc Heuse affermò che tale problema era irrisolvibile con le tecniche conosciute[6]. A distanza di anni, riuscì a trovare diverse tecniche con cui

[5] `http://ndpmon.sourceforge.net`
[6] "Remote alive scans (ping scans) as we know them are unfeasible on IPv6" - Marc Heuse

è possibile recuperare tali indirizzi e quindi individuare i diversi host attivi nella rete. In questo paragrafo saranno presentate queste ed altre tecniche le quali, aggregandole, permettono di ottenere una lista molto prossima agli host effettivamente presenti nella rete.

3.8.1 Tecniche di ricerca in rete locale

Vediamo quali siano le tecniche possibili per effettuare una ricerca degli host all'interno della rete locale.

- **Analizzatore di traffico**: potremmo analizzare il traffico multicast e individuare i pacchetti di Neighbor Discovery.

- **Ping**: è il metodo più semplice per individuare alcuni host che rispondono a tale richiesta. La tecnica consiste nell'inviare un pacchetto ICMPv6 con un messaggio di echo request all'indirizzo link-local multicast (ff02::1). Alla ricezione dei messaggi di echo reply si avranno gli indirizzi link-local degli host e si avrà quindi la certezza che tali nodo siano attivi. Questa modalità è prevista dal protocollo IPv6 e normalmente ottiene buoni risultati.

- **Invalid Extension Header**: si invia un pacchetto ICMPv6 con un Invalid Extension Header avente come indirizzo di destinazione l'indirizzo multicast di link-local corrispondente a tutti i nodi. Ogni nodo attivo replicherà a tale richiesta con un pacchetto ICMPv6 contenente un parametro di errore. Così facendo si potrà identificare quali siano i nodi attivi.

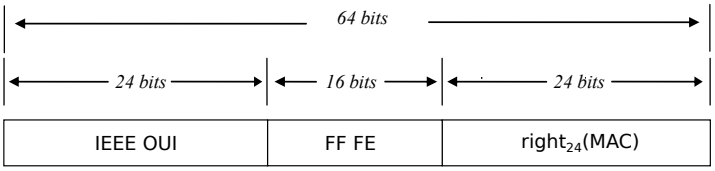
- **Autoconfigurazione**: si sfrutta il supporto SLAAC dei client per individuare la loro presenza in rete. In particolare, la ricerca consterà delle seguenti parti:

    1. invio un messaggio RA con un prefisso random, il valore Autoconfig a 1, Lifetime uguale ad 1 e una bassa priorità;
    2. il sistema che avrà abilitato l'autoconfigurazione definirà il proprio indirizzo ed entrerà nella fase di DAD;
    3. l'attaccante catturerà il pacchetto DAD e convertirà l'indirizzo globale in un indirizzo link-local;
    4. per concludere, l'indirizzo link-local verrà verificato con una richiesta echo request o con la procedura di Neighbor Discovery.

Normalmente i risultati di tali tecniche sono usati in maniera aggregata così da ottenere una visione il più possibile vicina alla realtà.

3.8.1.1 EUI-64

Come abbiamo visto nel primo capito gli indirizzi, durante la procedura SLAAC su Ethernet, usano il formato EUI-64 per configurarsi. Questo, apparentemente, potrebbe sembrare impossibile da predire e quindi renderebbe, come più volte affermato, impossibile una ricerca esaustiva a partire dal prefisso ricavabile, per esempio, dall'analisi del traffico di rete e dai pacchetti di Router Advertisement. Vediamo brevemente come si compone la parte che identifica il nodo.



Vediamo che la parte sinistra dell'indirizzo è costituita dal codice del produttore ed essendo questo di dominio pubblico è facilmente ricavabile in base al tipo di computer o delle schede di rete usate nel segmento analizzato. Rimane, quindi, che la parte variabile e non predicibile è soltanto quella di destra, lunga 24 bit. Così facendo si avrebbe che la ricerca per ogni IEEE OUI corrisponde a 2^{24} richieste e quindi una ricerca esaustiva degli indirizzi diventa una tecnica praticabile.

3.8.2 Ricerca degli host in Internet

Nel seguente paragrafo illustreremo quali siano le tecniche che permettono di ricercare gli indirizzi IPv6 esposti in Internet e quindi facenti uso di indirizzi globali. L'insieme delle tecniche, illustrate nella figura 3.10, permettono di avere un'ottima approssimazione, circa 66%, su una rete di 2000 host.

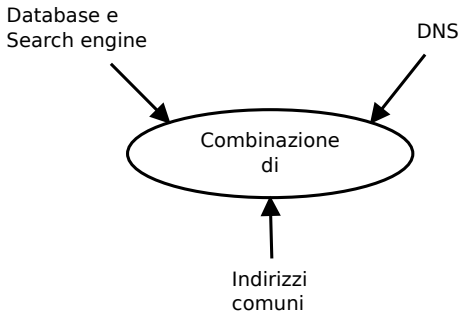


Figura 3.10: Tecniche di identificazione remota

Vediamo quindi quali siano le caratteristiche di ogni tecnica.

- **Search Engine**: possiamo usare i motori di ricerca per trovare informazioni sulle configurazioni e sugli indirizzi implementati.
- **DNS**: attraverso l'uso di un attacco a dizionario o bruteforce possiamo individuare l'indirizzo IPv6 degli host presenti.

- **Indirizzi comuni**: si potrebbe effettuare una ricerca basata su indirizzi che vengono usati per la loro semplicità mnemonica o sulle abitudini usuali degli amministratori di rete. Vediamoli in base alla tipologia di appartenenza.
  - Autoconfigurazione: vediamo quali siano le possibilità in caso si usasse SLAAC per configurare l'host:
    - indirizzo MAC: come abbiamo visto, con l'uso dell'indirizzo MAC, la ricerca si circoscrive ad un campione di 2^{24} per ciascun valore di vendor ID;
    - opzione Privacy: non esistendo una funzione predeterminata per il calcolo ci è impossibile derivarne uno schema fisso;
    - random: come per il precedente.
  - Configurati a mano:
    - modello: normalmente la persona addetta all'assegnazione degli indirizzi IPv6 usa, per facilità, un motivo ben distinguibile nell'assegnazione degli indirizzi. Per esempio:
      - ::1, ::2, ::3
      - ::porta
      - ::1:porta, ::2:porta
      - ::porta:1, ::porta:2
    - random: ci è impossibile predeterminare un andamento dell'indirizzo.
  - DHCPv6: normalmente l'assegnazione degli indirizzi DHCPv6 è sequenziale. Quindi, trovato un indirizzo si può facilmente cercare gli altri. Un esempio di indirizzi spesso usati nei server DHCPv6 sono i seguenti:
    - ::1000-2000
    - ::100-200
    - ::1:0-1000
    - ::1:1000-2000

In figura 3.11 vediamo la distribuzione, relativa all'anno 2010, rilevata con il comando *alive* della suite thc-ipv6. Si noti come la parte configurata con indirizzi facili e predicibili sia in netta maggioranza.
In conclusione, tali tecniche permettono di trovare circa il 90-95% dei server presenti nelle classi scansionate.

3.8.3 Record PTR

In questo paragrafo vedremo una recente tecnica con cui è possibile identificare indirizzi IPv6 validi sfruttando la richiesta PTR ed una gestione non corretta degli errori presente in alcuni server DNS (es. BIND e NSD). Un'idea della potenzialità di tale

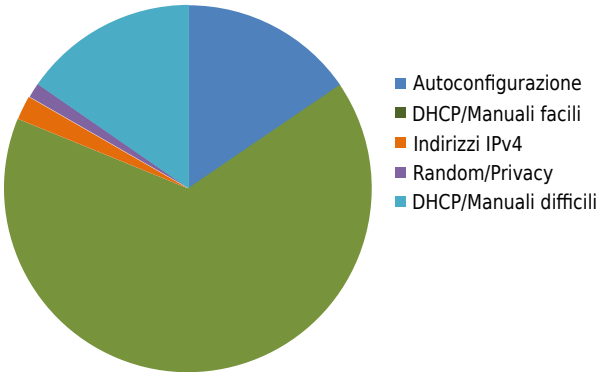


Figura 3.11: Distribuzione degli Indirizzi IPv6 (da Marc Heuse [49])

tecnica è il seguente esempio: permette di recuperare 737 valori PTR da una rete /48 con solo 14785 richieste. In pratica, la ricerca si basa sulla differenza di errore ricevuto nel caso il campo PTR fosse valido o meno. In particolare, nel caso esistesse un valore PTR nel sotto-albero successivo, il server risponderebbe con un messaggio *NOERROR* mentre nel caso non ci fosse alcun indirizzo nel sotto-albero successivo, il server DNS risponderebbe con un messaggio *NXDOMAIN*. La figura 3.12 è una rappresentazione grafica di tale spiegazione.

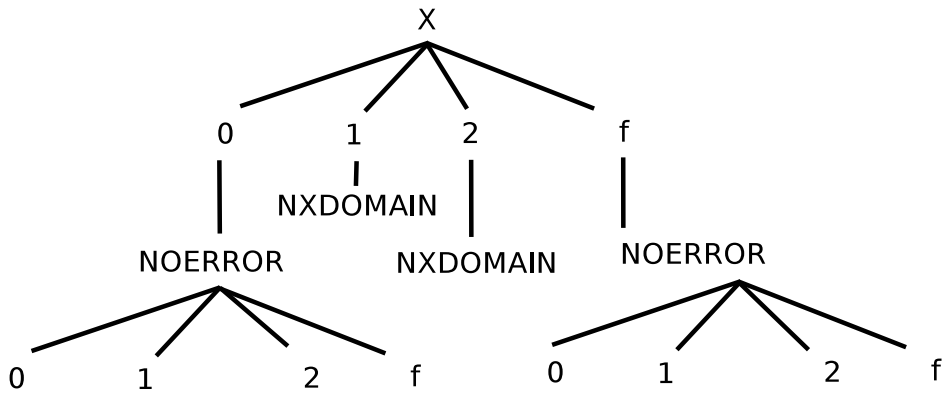


Figura 3.12: Albero di richieste PTR

Vediamo un semplice esempio così da chiarirne il funzionamento.
Assumiamo di avere un blocco 2001:DB8::/32 in cui si avrà una sotto rete avente prefisso 2001:DB8:80::/48 e dotata di un reverse DNS e la zona è definita come 0.8.0.0.8.b.d.0.1.0.0.2.ip6.arpa. In seguito la zona sarà definita con la lettera X. Iniziamo con interrogare il server DNS con i seguenti valori: 0.X, 1.X, 2.X fino a f.X. Molte delle risposte ritorneranno il codice NXDOMAIN che ci avvisa che il valore richiesto non esiste. In quei server vulnerabili, tale errore ci avvisa che oltre all'indirizzo richiesto non è presente nessun altro indirizzo di livello inferiore. Supponiamo che nel caso di 0.X e f.X il server non ritorni NXDOMAIN bensì il valore NOERROR. Questo esplicita che esistono, nell'albero DNS, valori successivi a quello richiesto. Iterando

tale algoritmo è facile scoprire tutti gli indirizzi registrati nel server DNS. Questo comportamento agevola notevolmente la ricerca degli indirizzi IPv6 di un'organizzazione e mediamente non richiede molto tempo. Si suppone, ovviamente, che l'organizzazione in questione abbia inserito gli indirizzi IPv6 all'interno del server DNS.

3.9 Ulteriori attacchi

Per concludere il capitolo dedicato ai possibili attacchi sfruttando il protocollo IPv6 analizziamo alcune strategie che per loro natura non sono state incluse nei paragrafi precedenti. Verranno presentati i messaggi Node Information Query/Response, l'ICMPv6 Redirection e l'attacco al Multicast Listener Discovery.

3.9.1 Node Information Query/Response

Il documento RFC 4620 [22] specifica un protocollo sperimentale per la richiesta di informazioni di rete in un ambiente privo di server. Per esempio, possono essere richieste le seguenti informazioni: l'hostname, il nome con il dominio e gli indirizzi IPv4 e IPv6 abilitati nel nodo. Tale protocollo, anche se definito dagli stessi autori sperimentale, è abilitato nello stack IPv6 KAME[7].
Nei sistemi BSD è presente una chiave (*net.inet6.icmp6.nodeinfo*) che definisce il tipo di risposta che il sistema dovrà assumere alla ricezione di una richiesta di Node Information. Vediamo quali siano tali valori:

- 0: nessuna risposta in caso di richiesta;
- 1: rispondi solo alle interrogazioni del campo FQDN;
- 2: rispondi alle richieste degli indirizzi IP;
- 3: rispondi a tutte le richieste.

In OpenBSD il valore predefinito è uguale a 1 mentre in FreeBSD è uguale a 3. Vediamo ora alcuni esempi di richiesta.

Richiesta del hostname

```
$ ping6 -w ff02::1%vic0

PING6(72=40+8+24 bytes) fe80::20c:29ff:feaf:194e%vic0
  --> ff02::1%vic0

41 bytes from fe80::20c:29ff:feaf:194e%vic0: openbsd46.my.domain.
30 bytes from fe80::20c:29ff:fe49:ebdd%vic0: freebsd
```

[7] http://www.kame.net

```
41 bytes from fe80::20c:29ff:feaf:194e%vic0: openbsd46.my.domain.
30 bytes from fe80::20c:29ff:fe49:ebdd%vic0: freebsd
41 bytes from fe80::20c:29ff:feaf:194e%vic0: openbsd46.my.domain.
30 bytes from fe80::20c:29ff:fe49:ebdd%vic0: freebsd
--- ff02::1%vic0 ping6 statistics ---
3 packets transmitted, 3 packets received, +3 duplicates,
  0.0% packet loss
```

Effettuando la richiesta sull'indirizzo multicast corrispondente a tutti i nodi, ogni nodo risponde alla richiesta riportando il proprio hostname Si noti il carattere % che separa l'indirizzo dall'interfaccia. Nelle richieste multicast o link-local è obbligatorio specificare l'interfaccia da usare per l'invio del messaggio.

Richiesta degli indirizzi

```
$ ping6 -a Aacgls ff02::1%vic0

PING6(72=40+8+24 bytes) fe80::20c:29ff:feaf:194e%vic0
  --> ff02::1%vic0

76 bytes from fe80::20c:29ff:fe49:ebdd%vic0:
fe80::20c:29ff:fe49:ebdd(TTL=infty)
::1(TTL=infty) fe80::1(TTL=infty)

76 bytes from fe80::20c:29ff:fe49:ebdd%vic0:
fe80::20c:29ff:fe49:ebdd(TTL=infty)
::1(TTL=infty) fe80::1(TTL=infty)

76 bytes from fe80::20c:29ff:fe49:ebdd%vic0:
fe80::20c:29ff:fe49:ebdd(TTL=infty)
::1(TTL=infty)
fe80::1(TTL=infty)

--- ff02::1%vic0 ping6 statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
```

In questo esempio abbiamo visto come ogni nodo riporta il proprio indirizzo link-local effettuando una richiesta sempre sull'indirizzo multicast corrispondente a tutti i nodi presenti sulla rete.

3.9.2 ICMPv6 Redirect

Il messaggio ICMPv6 Redirect viene usato per reindirizzare automaticamente il traffico di un host su un router diverso o per informare gli host che la destinazione è diventata un vicino. Il funzionamento è molto semplice: quando un first-hop router scopre un instradamento migliore per una specifica destinazione invia un messaggio di reindirizzamento al nodo sorgente indicandogli di usare il nuovo router. A questo punto il nodo

sorgente aggiorna la tabella di routing con la nuova destinazione. L'attacco è altrettanto semplice ed è uguale a quello presente in IPv4. Un malintenzionato può forgiare un messaggio ICMPv6 Redirect e redirigere il traffico verso se stesso o verso un router da lui controllato. Si potrebbe per esempio annunciare reti remote (es. Paypal) verso un router controllato dall'attaccante.
Una semplice procedura per l'attacco potrebbe essere la seguente.

1. L'attaccante invia un messaggio ICMPv6 echo request alla vittima con indirizzo sorgente uguale alla destinazione (D1) che si vuole redirigere.
2. La vittima invierà un messaggio echo reply a D1 usando il first-hop router (R1).
3. A questo punto l'attaccante invia un messaggio ICMPv6 Redirect alla vittima con indirizzo sorgente R1, come payload il messaggio echo reply e indicandogli, come nuovo indirizzo per la destinazione D1, l'indirizzo dell'attaccante.
4. La vittima aggiornerà la tabella di routing con il nuovo valore.

Tale possibilità è attiva in maniera predefinita sia sui sistemi *BSD sia su Linux.

3.9.3 Multicast Listener Discovery

Come abbiamo visto in precedenza, il Multicast Listener Discovery (MLD) è un protocollo che permette ai nodi di informare il router locale a quali gruppi siano iscritti. Il router, a sua volta, usa tali informazioni per decidere quali pacchetti deve inviare nel segmento locale. Vediamo come funziona.
La figura 3.13 ci illustra i pacchetti inviati durante una richiesta DNS multicast.

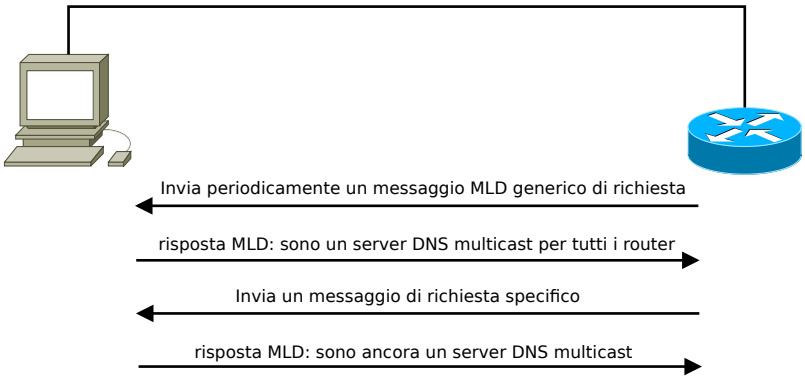


Figura 3.13: Traffico DNS multicast

Il primo problema che si pone per compiere l'attacco è quello di diventare il router MLD che effettua le richieste. La figura 3.14 ci illustra i passaggi con cui è possibile portare a termine questa richiesta.
L'ultimo problema che l'attaccante si deve porre per prolungare l'attacco è far sì che il nuovo router rimanga il predefinito per la rete e quindi senza la necessità di inviare

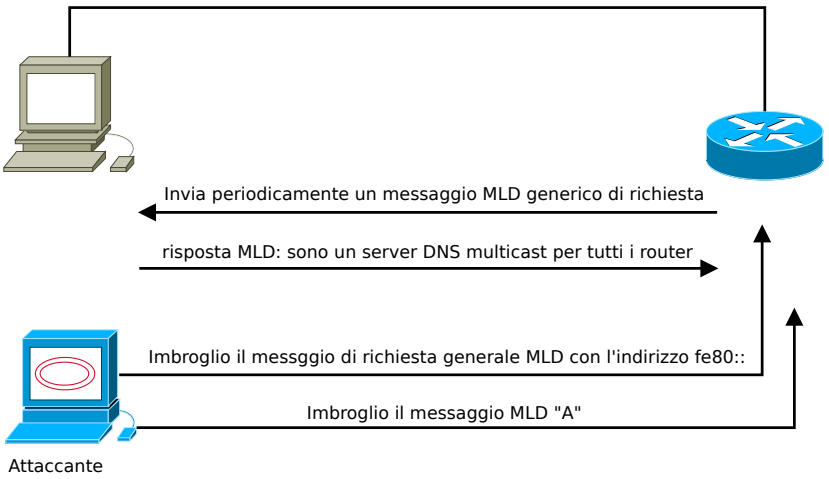


Figura 3.14: Attacco al router MLD predefinito

periodicamente un messaggio MLD. Una soluzione al problema è usare l'indirizzo di livello 2 multicast che si riferisce a tutti i router. La figura 3.15 ben rappresenta la successione dei messaggi inviati.

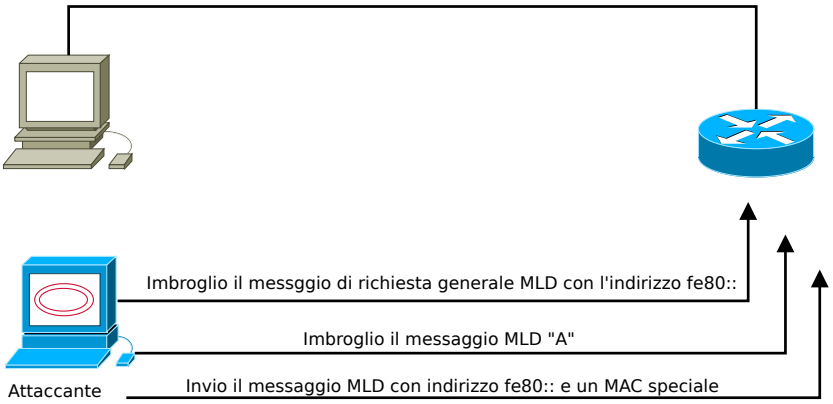


Figura 3.15: Mantenimento della posizione predefinita

A questo punto avremo ottenuto una rete in cui gli host non hanno più accesso corretto alla rete multicast.

Capitolo 4

Direzioni future di ricerca

Oltre agli aspetti di sicurezza e alle vulnerabilità discusse in questa tesi, ci sono una serie di fattori, tecnici e non tecnici, che influenzeranno notevolmente le implementazioni IPv6 e i relativi attacchi. Vediamo quali siano tali elementi e quali siano gli aspetti su cui la ricerca dovrà soffermarsi maggiormente in futuro.

4.1 Implementazioni immature

Come abbiamo visto all'inizio del secondo capitolo, il numero di vulnerabilità relative alle implementazioni IPv6 è in notevole aumento. In gran parte perchè in passato non si è effettuata molta ricerca sulle implementazioni IPv6 e solo ora, visto l'aumento considerevole dell'uso di IPv6 e delle possibilità di attacco, si è iniziato a prenderlo in considerazione. Tale sintomo propende per un aumento considerevole dei problemi di sicurezza nei prossimi anni ed è per questo che non bisogna considerare le implementazioni odierne più sicure di IPv4. Chiaramente, nel breve periodo, il protocollo IPv6 è destinato a diventare l'anello debole dell'infrastruttura, soprattutto per quelle implementazioni facenti uso della modalità Dual-Stack. Si ha quindi la necessità di formare correttamente gli addetti così da diminuire considerevolmente l'impatto che un'eventuale vulnerabilità avrà sull'infrastruttura.

4.2 Mancanza di formazione specifica

Si stima che a livello Mondiale circa 20 milioni di Ingegneri e Sistemisti abbiano bisogno di formazione specifica sul nuovo protocollo IPv6. Per quelli che hanno già ora qualche conoscenza di IPv6, il più delle volte non è paragonabile a quella di IPv4. Questo è dovuto, principalmente, alla bassa implementazione dello stesso, in particolare in quelle aree dedicate ai tecnici. Poichè a tali Ingegneri verrà richiesto, molto probabilmente, di implementare IPv6 senza o con una minima conoscenza, si rischia di incorrere in importanti ricadute nella sicurezza, come in parte avvenne per IPv4. Per evitare tali

inconvenienti si dovrebbe prevedere sia un'area di sperimentazione circoscritta, atta ad acquisire il necessario know-how, sia prevedere un piano di formazione e di implementazione del protocollo nella parte produttiva dell'organizzazione. In aggiunta al buon senso e alla formazione, negli anni sono stati prodotti diversi best-practice per l'implementazione sicura di IPv6. Tali documenti, sviluppati sia da organismi normativi come IETF sia da organizzazioni governative come NIST e CPNI, sono ancora in fase di sviluppo ma rimangono degli ottimi riferimenti per un'implementazione corretta del protocollo IPv6.

4.3 Supporto limitato nei security assessment tool

Strettamente correlato alla mancanza di ricerca nel campo IPv6 è il numero di tool che permetto di valutare la sicurezza dello IPv6. Come abbiamo visto in questa tesi sono pochi in confronto a quelli disponibili in IPv4. Per esempio, in questa tesi si è usata la suite thc-ipv6, scapy e nmap. Il risultato di questa lacuna è che alcuni amministratori addetti alla sicurezza di rete possono trovarsi nella situazione in cui ci sia un senso di sicurezza fuorviato dai pochi tool e dal poco supporto.

4.4 Supporto limitato nei dispositivi di sicurezza

I dispositivi di sicurezza, quali firewall e IDS di rete, di solito offrono meno supporto per i protocolli IPv6 che per le controparti IPv4. Questo può tradursi in una diminuzione sia in termini di funzionalità sia in prestazioni. Ad esempio, un dispositivo di sicurezza può implementare il deep-inspection per IPv4 e non per IPv6 o può implementare alcune caratteristiche in entrambi i protocolli ma il supporto IPv4 sarà implementato in hardware mentre quello IPv6 in software. Chiaramente, questa disparità di caratteristiche tra IPv4 e IPv6 potrebbe portare a ridurre la sicurezza del nuovo protocollo e può impedire l'attuazione delle politiche già presenti in IPv4.

4.5 Ricerca degli host

Una delle sfide maggiori per i prossimi anni è quella relativa alle tecniche atte all'individuazione completa degli host presenti in rete. Alcune proposte sono state discusse nel capitolo 3, le quali danno un ottimo risultato ma non pari a quello che si raggiunge oggi con il protocollo IPv4. Questo è sicuramente legato alla mancanza di ricerca ed è quindi un buon spunto per investigare ulteriormente questo aspetto.

4.6 Malware

Oggigiorno, il Malware è diventato una grave minaccia per le reti di computer. In particolare i nuovi malware, si veda Stuxnet, hanno preso in considerazione obiettivi specifici e di tipo strategico. Se una volta potevano essere usati per divertimento o per dimostrazione, oggi sono diventati una propria economia in mano ad organizzazioni non lecite il cui unico scopo è il lucro. Ci si ritrova, quindi, a poter acquistare sia il software pronto all'uso sia una rete di macchine, chiamate *botnet*, pronte ad effettuare l'attacco voluto. Viste le difficoltà odierne di controllo sull'infrastruttura IPv6, i creatori di malware si stanno sempre più appoggiando al nuovo protocollo, in particolare con l'uso di tunnel. Così facendo, il più delle volte, possono aggirare i controlli di sicurezza che man mano si stanno implementando in IPv4. Un documento interessante, pubblicato da Zulkiflee M., Faizal M.A., Mohd Fairuz I. O., Nur Azman A. e Shahrin S. [91], ritrae proprio come l'evoluzione delle tecniche di diffusione stia abbracciando IPv6. In particolare, il worm Nimda prevede nativamente la possibilità, a partire da reti IPv4, di infettare reti IPv6.
Si prevede quindi una notevole evoluzione dei Malware verso l'universo IPv6 nel quale molte politiche di monitoraggio e di analisi non sono presenti.

4.7 IPv6 Mobile

Uno degli aspetti che è stato poco considerato in questa tesi è la parte inerente al IPv6 Mobile. In particolare, si è solo accennato ad alcune differenze tra la versione standard e la versione mobile. Sarebbe quindi interessante approfondire gli aspetti di sicurezza ed eventuali vulnerabilità e attacchi presenti all'interno di tali specifiche. Negli anni sono stati pubblicati diversi documenti (RFC) relativi all'implementazione di IPv6 Mobile con IPsec e alla sua sicurezza ma non si trovano in letteratura documenti di eventuali tecniche di attacco.

Conclusione

Come si è potuto leggere nei precedenti capitoli, la ricerca sul protocollo IPv6 e le sue implementazioni sono soltanto all'inizio e quindi si presume che, in un futuro neanche tanto lontano, alcune delle problematiche ivi proposte saranno risolte. In particolare, alcuni degli attacchi non saranno certamente realizzabili con tale semplicità. Ad ogni modo, si è convinti che questo documento possa servire ad una maggiore comprensione del protocollo IPv6 ed in particolare sfatare quell'alone di eccessiva sicurezza che lo ricopre tra gli addetti ai lavori.
Gli attacchi precedentemente proposti, per esempio quelli di Denial of Service o di dirottamento, dovrebbero sensibilizzare i lettori durante una futura implementazione ed in particolare aumentare la richiesta, verso i produttori, di caratteristiche dedite alla protezione perimetrale del protocollo IPv6.

Elenco delle figure

Bibliografia

[1] J. Abley, P. Savola, and G. Neville-Neil. Deprecation of Type 0 Routing Headers in IPv6. RFC 5095, Internet Engineering Task Force, December 2007.

[2] James Woodyatt e Yiu L. Lee Alain Durand, Ralph Droms. Dual-stack lite broadband deployments following ipv4 exhaustion. *draft-ietf-softwire-dual-stack-lite-11*, May 2011.

[3] J. Amoss and D. Minoli. *Handbook of IPv4 to IPv6 transition: methodologies for institutional and corporate networks*. CRC Press, 2008.

[4] J. Arkko and A. Keranen. Experiences from an IPv6-Only Network. RFC 6586, Internet Engineering Task Force, April 2012.

[5] J. Arkko and I. van Beijnum. Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming. RFC 5534, Internet Engineering Task Force, June 2009.

[6] H. Babiker, I. Nikolova, and K.K. Chittimaneni. Deploying ipv6 in the google enterprise network. lessons learned. 2011.

[7] M. Bagnulo. Hash-Based Addresses (HBA). RFC 5535, Internet Engineering Task Force, June 2009.

[8] C. Bao, C. Huitema, M. Bagnulo, M. Boucadair, and X. Li. IPv6 Addressing of IPv4/IPv6 Translators. RFC 6052, Internet Engineering Task Force, October 2010.

[9] M. Blanchet and F. Parent. IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP). RFC 5572, Internet Engineering Task Force, February 2010.

[10] J. Bound, Y. Pouffary, S. Klynsma, T. Chown, and D. Green. IPv6 Enterprise Network Analysis - IP Layer 3 Focus. RFC 4852, Internet Engineering Task Force, April 2007.

[11] B. Carpenter and K. Moore. Connection of IPv6 Domains via IPv4 Clouds. RFC 3056, Internet Engineering Task Force, February 2001.

[12] A.R. Choudhary. In-depth analysis of ipv6 security posture. In *Collaborative Computing: Networking, Applications and Worksharing, 2009. CollaborateCom 2009. 5th International Conference on*, pages 1–7. IEEE, 2009.

[13] A.R. Choudhary and A. Sekelsky. Securing ipv6 network infrastructure: A new security model. In *Technologies for Homeland Security (HST), 2010 IEEE International Conference on*, pages 500 –506, November 2010.

[14] A.R. Choudhary and A. Sekelsky. Securing ipv6 network infrastructure: A new security model. In *Technologies for Homeland Security (HST), 2010 IEEE International Conference on*, pages 500–506. IEEE, 2010.

[15] T. Chown. IPv6 Implications for Network Scanning. RFC 5157, Internet Engineering Task Force, March 2008.

[16] T. Chown and S. Venaas. Rogue IPv6 Router Advertisement Problem Statement. RFC 6104, Internet Engineering Task Force, February 2011.

[17] M. Colajanni, L. Dal Zotto, M. Marchetti, and M. Messori. Defeating nids evasion in mobile ipv6 networks. In *Proc. of the 12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM 2011), Lucca, Italy, June*, 2011.

[18] A. Conta and S. Deering. Generic Packet Tunneling in IPv6 Specification. RFC 2473, Internet Engineering Task Force, December 1998.

[19] A. Conta, S. Deering, and M. Gupta. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. RFC 4443, Internet Engineering Task Force, March 2006.

[20] S. Convery and D. Miller. Ipv6 and ipv4 threat comparison and best-practice evaluation (v1. 0). *Presentation at the 17th NANOG*, 2004.

[21] M. Crawford. Transmission of IPv6 Packets over Ethernet Networks. RFC 2464, Internet Engineering Task Force, December 1998.

[22] M. Crawford and B. Haberman. IPv6 Node Information Queries. RFC 4620, Internet Engineering Task Force, August 2006.

[23] J. Damas and F. Neves. Preventing Use of Recursive Nameservers in Reflector Attacks. RFC 5358, Internet Engineering Task Force, October 2008.

[24] E. Davies, S. Krishnan, and P. Savola. Ipv6 transition/co-existence security considerations. *draft-ietf-v6ops-security-overview-06 (work in progress)*, 2006.

[25] E. Davies, S. Krishnan, and P. Savola. IPv6 Transition/Co-existence Security Considerations. RFC 4942, Internet Engineering Task Force, September 2007.

[26] E. Davies and J. Mohacsi. Recommendations for Filtering ICMPv6 Messages in Firewalls. RFC 4890, Internet Engineering Task Force, May 2007.

[27] S. Deering, W. Fenner, and B. Haberman. Multicast Listener Discovery (MLD) for IPv6. RFC 2710, Internet Engineering Task Force, October 1999.

[28] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, Internet Engineering Task Force, December 1998.

[29] R. Despres. IPv6 Rapid Deployment on IPv4 Infrastructures (6rd). RFC 5569, Internet Engineering Task Force, January 2010.

[30] Sheila Frankel Doug Montgomery, Stephen Nightingale and Mark Carson. *A Profile for IPv6 in the U.S. Government*. NIST - National Institute of Standards and Technology, July 2008.

[31] R. Draves. Default Address Selection for Internet Protocol version 6 (IPv6). RFC 3484, Internet Engineering Task Force, February 2003.

[32] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315, Internet Engineering Task Force, July 2003.

[33] A. Durand, P. Fasano, I. Guardini, and D. Lento. IPv6 Tunnel Broker. RFC 3053, Internet Engineering Task Force, January 2001.

[34] A. Durand, J. Ihren, and P. Savola. Operational Considerations and Issues with IPv6 DNS. RFC 4472, Internet Engineering Task Force, April 2006.

[35] Marco d'Itri. Multicast ipv6. IPv6 Task Force Italia, 2005.

[36] P.B.A. EBALARD. Ipv6 routing header security., 2007.

[37] Congxiao Bao Fred Baker, Xing Li and Kevin Yin. Framework for ipv4/ipv6 translation. *draft-ietf-behave-v6v4-framework-10*, February 2011.

[38] Fernando Gont. Hacking ipv6 networks. June 2011.

[39] Fernando Gont. Ipv6 router advertisement guard (ra-guard) evasion. *draft-gont-v6ops-ra-guard-evasion-00*, May 2011.

[40] Fernando Gont. A method for generating stable privacy-enhanced addresses with ipv6 stateless address autoconfiguration (slaac). *draft-gont-6man-stable-privacy-addresses-00*, December 2011.

[41] Fernando Gont. Results of a security assessment of the internet protocol version 6 (ipv6). November 2011.

[42] Fernando Gont. *Security Implications of the Use of IPv6 Extension Headers with IPv6 Neighbor Discovery*, May 2011.

[43] Fernando Gont. Recent advances in ipv6 security. Hackito Ergo Sum, April 2012.

[44] Fernando Gont. Security assessment of the ipv6 flow label. *draft-gont-6man-flowlabel-security-02*, January 2012.

[45] Fernando Gont. Security implications of the use of ipv6 extension headers with ipv6 neighbor discovery. *draft-gont-6man-nd-extension-headers-02*, January 2012.

[46] B. Haberman. Allocation Guidelines for IPv6 Multicast Addresses. RFC 3307, Internet Engineering Task Force, August 2002.

[47] B. Haberman and D. Thaler. Unicast-Prefix-based IPv6 Multicast Addresses. RFC 3306, Internet Engineering Task Force, August 2002.

[48] J. Hagino and K. Yamamoto. An IPv6-to-IPv4 Transport Relay Translator. RFC 3142, Internet Engineering Task Force, June 2001.

[49] Marc Heuse. Recent advances in ipv6 insecurities. December 2010.

[50] N. Hilliard. A discard prefix for ipv6. *draft-ietf-v6ops-ipv6-discard-prefix-00*, October 2011.

[51] R. Hinden and S. Deering. IPv6 Multicast Address Assignments. RFC 2375, Internet Engineering Task Force, July 1998.

[52] R. Hinden and B. Haberman. Unique Local IPv6 Unicast Addresses. RFC 4193, Internet Engineering Task Force, October 2005.

[53] S. Hogg and E. Vyncke. *IPv6 security*. Cisco Systems, 2009.

[54] G. Huston. Architectural Approaches to Multi-homing for IPv6. RFC 4177, Internet Engineering Task Force, September 2005.

[55] IAB and IESG. IAB/IESG Recommendations on IPv6 Address Allocations to Sites. RFC 3177, Internet Engineering Task Force, September 2001.

[56] Merike Kaeo. Ipv6 security technology paper. Technical report, North American IPv6 Task Force, July 2006.

[57] George Kargiotakis. Security considerations for a brave new (ipv6) world. November 2011.

[58] S. Krishnan, D. Thaler, and J. Hoagland. Security Concerns with IP Tunneling. RFC 6169, Internet Engineering Task Force, April 2011.

[59] Vassilis Merekoulias Krzysztof Cabaj et al. Security issues in autonomic ipv6 networks, jan 2011.

[60] T. Lancaster. Ipv6 & ipv4 threat review with dual-stack considerations. *COMP6009: Individual Research Project, University of Southampton, Department of Electronics and Computer Science, UK*, 2006.

[61] Clement Lecigne and George V. Neville-Neil. Walking through freebsd ipv6 stack. August 2006.

[62] E. Levy-Abegnoli, G. Van de Velde, C. Popoviciu, and J. Mohacsi. IPv6 Router Advertisement Guard. RFC 6105, Internet Engineering Task Force, February 2011.

[63] T. Lin, W. Liu, H. Duan, and D. Sun. Ipv6 traffic hijack test system and defense tools using dnssec. In *Internet Technology and Applications (iTAP), 2011 International Conference on*, pages 1–5. IEEE, 2011.

[64] D. Malone. Observations of ipv6 addresses. *Passive and Active Network Measurement*, pages 21–30, 2008.

[65] David Malone. Observations of ipv6 addresses. In *Proceedings of the 9th international conference on Passive and active network measurement*, PAM'08, pages 21–30, Berlin, Heidelberg, 2008. Springer-Verlag.

[66] J. McCann, S. Deering, and J. Mogul. Path MTU Discovery for IP version 6. RFC 1981, Internet Engineering Task Force, August 1996.

[67] Mirco Marchetti Michele Colajanni, Luca Dal Zotto and Michele Messori. *Defeating NIDS evasion in Mobile IPv6 networks*, 2011.

[68] D. Minoli and J. Kouns. *Security in an IPv6 environment*. Taylor and Francis, 2008.

[69] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). RFC 2461, Internet Engineering Task Force, December 1998.

[70] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 4861, Internet Engineering Task Force, September 2007.

[71] P. Nikander, J. Kempf, and E. Nordmark. IPv6 Neighbor Discovery (ND) Trust Models and Threats. RFC 3756, Internet Engineering Task Force, May 2004.

[72] E. Nordmark and T. Li. Threats Relating to IPv6 Multihoming Solutions. RFC 4218, Internet Engineering Task Force, October 2005.

[73] PineApp. Ipv6 - the battle against botnet. APNIC.

[74] Casimir A. Potyraj. *Firewall Design Considerations for IPv6*. National Security Agency, 2007.

[75] J. Rajahalme, A. Conta, B. Carpenter, and S. Deering. IPv6 Flow Label Specification. RFC 3697, Internet Engineering Task Force, March 2004.

[76] Mario Reale. Sicurezza con ipv6. In GARR, editor, *GARR WS9*, June 2009.

[77] D. Thaler S. Krishnan and J. Hoagland. Security concerns with ip tunneling. *draft-ietf-v6ops-tunnel-security-concerns-04*, 2010.

[78] P. Savola and J. Lingard. Host Threats to Protocol Independent Multicast (PIM). RFC 5294, Internet Engineering Task Force, August 2008.

[79] P. Savola and C. Patel. Security Considerations for 6to4. RFC 3964, Internet Engineering Task Force, December 2004.

[80] John Pearce Sheila Frankel, Richard Graveman and Mark Rooks. *Guidelines for the Secure Deployment of IPv6*. NIST - National Institute of Standards and Technology, December 2010.

[81] Aditya K. Sood. Botnets and browser - brothers in the ghost shell. In *BruCON 2011*. Michigan State University, September 2011.

[82] Network Sorcery. Rfc sourcebook, March 2011.

[83] F. Templin, T. Gleeson, and D. Thaler. Intra-Site Automatic Tunnel Addressing Protocol (ISATAP). RFC 5214, Internet Engineering Task Force, March 2008.

[84] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. RFC 2462, Internet Engineering Task Force, December 1998.

[85] S. Thomson, T. Narten, and T. Jinmei. IPv6 Stateless Address Autoconfiguration. RFC 4862, Internet Engineering Task Force, September 2007.

[86] Arrigo Triulzi. *Intrusion Detection Systems and IPv6*, 2003.

[87] US-CERT. *Malware Tunneling in IPv6*, May 2005.

[88] R. Vida and L. Costa. Multicast Listener Discovery Version 2 (MLDv2) for IPv6. RFC 3810, Internet Engineering Task Force, June 2004.

[89] I. Gashinsky W. Kumari and J. Jaeggli. Operational neighbor discovery problems. *draft-gashinsky-v6ops-v6nd-problems-00*, 2011.

[90] M. Wadhwa and M. Khari. Prevention algorithm against the vulnerability of type 0 routing header in ipv6. In *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*, pages 616–620. IEEE, 2011.

[91] Mohd Fairuz I. O. Zulkiflee M., Faizal M. A., Nur Azman A., and Shahrin S. Behavioral analysis on ipv4 malware in both ipv4 and ipv6 network environment. *Internation Journal of Computer Science and Information Security*, 9:10–15, February 2011.

This document is released under the Creative Commons license
(Attribution-ShareAlike 3.0).
Click on the image below for more details.