Jurnal SimanteC
Vol 1, No 3 Desember 2010 ISSN 2088-2130

SERANGAN ARP DAN DHCP PADA JARINGAN IPV4 DAN IPV6

Husni

Laboratorium Sistem Terdistribusi
Jurusan Teknik Informatika, Fakultas Teknik, Universitas Trunojoyo Madura
Jl. Raya Telang PO. BOX 2 Kamal, Bangkalan, Madura 69192
E-Mail : husni@if.trunojoyo.ac.id

Abstrak

Protokol ARP berperan memetakan IP *address* ke MAC *address* dan sebaliknya pada komunikasi antar *node* di dalam jaringan lokal. DHCP bertugas memberikan IP *address* serta konfigurasi lain ke suatu *interface* jaringan. Berbagai bentuk serangan dapat terjadi terhadap dua protokol ini dimulai dari ARP *spoofing*. IPv6 memperbaiki banyak kelemahan pada IPv4 tetapi tidak terhadap ARP dan DHCP. IPv6 menggunakan protokol *Neighbor Discovery* dalam penanganan komunikasi jaringan lokal (*local link*). Hampir semua teknik serangan ARP dan DHCP pada IPv4 dapat diberlakukan pada IPv6, bahkan terbuka kemungkinan hadirnya jenis serangan baru. Teknik *SEcure Neighbor Discovery* (SEND) sebagai solusi yang diperkenalkan pada IPv6 sangat sulit diaplikasikan terutama terkait dengan pembangkitan alamat terkriptografi. Teknik pengamanan yang dilakukan terhadap jaringan IPv4 dapat diterapkan pada IPv6.

Kata kunci: *ARP, DHCP, Neighbor Discovery, IPv4, IPv6*

Abstract

ARP protocol has a role in IP address to the MAC address mapping and vice versa on the communication between nodes on the local network. DHCP assigns IP address and provides other configuration to a network interface. Various forms of attack can happen to these two protocols which are started by ARP spoofing. IPv6 fix many flaws in IPv4 but not to the ARP and DHCP. IPv6 uses the Neighbor Discovery protocol in the handling of communication local network (local link). Almost all the attack techniques in IPv4's ARP and DHCP can be applied to IPv6, even open the possibility of the presence of new types of attacks. The Secure Neighbor Discovery (SEND) technique as a solution that was introduced in IPv6 is very difficult to be applied primarily associated with the generation of cryptographed address. Security techniques which are performed on IPv4 networks can be applied to IPv6, too.

Key words: ARP, DHCP, Neighbor Discovery, IPv4, IPv6

PENDAHULUAN

Suatu komputer pada jaringan IP (*Internet Protocol*) atau Ethernet mempunyai dua alamat, yaitu alamat alamat *hardware* (fisik) atau *Media Access Controller* (MAC) *address*, dan alamat IP yang dikenal sebagai alamat logis atau *software* yang dapat ditentukan oleh pengguna [1]. Sebagai pelengkap untuk memudahkan pengguna, setiap node dapat diberi nama, biasanya memanfaatkan layanan *Domain Name System* (DNS).

Komunikasi antar *host* dalam suatu jaringan terjadi pada lapisan Data Link dari model OSI.

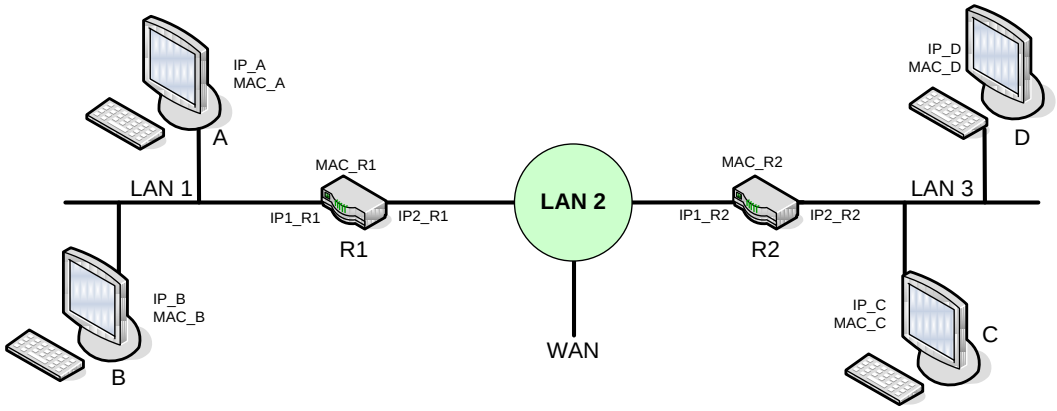
Hardware pada lapisan ini tidak memahami IP *address* dan hanya mengerti alamat fisik [4].

Komputer tersambung ke jaringan menggunakan suatu *interface card* yang mempunyai suatu alamat fisik unik bernama

MAC *address* dengan panjang 48-bit. Setiap kartu antarmuka mempunyai MAC *address* berbeda, tidak ada yang sama. Setiap pabrik pembuat kartu memperoleh nomor unik dari suatu otoritas sentral sepanjang 24 bit. Pabrik kemudian menentukan 24 bit nomor unik untuk setiap kartunya. Kedua nomor tersebut disatukan untuk menghasilkan MAC *address* lengkap. Keunikan ini diharapkan menjamin tidak terjadinya konflik MAC address pada suatu jaringan.

Naskah ini secara urut akan menguraikan cara kerja dari protokol ARP dalam memetakan IP *address* ke MAC *address* dan sebaliknya, protokol DHCP untuk memberikan IP *address* serta beberapa parameter konfigurasi jaringan kepada *client node*, baik dalam jaringan berbasis IPv4 maupun IPv6, berbagai kemungkinan serangan terhadap kedua protokol dan jaringan, serta teknik-teknik penyelesaiannya. Pertanyaan yang hendak dijawab adalah ”apakah serangan ARP dan DHCP yang berlaku di IPv4 dapat berjalan pada *Neighbor Discovery* di IPv6?”, ”jika jawabannya ’iya’, pendekatan apa yang dapat dilakukan untuk mengurangi atau mencegah serangan tersebut?”.

Terdapat cukup banyak perbedaan antara IPv4 dan IPv6. IPv4 adalah revisi ke-4 dari pengembangan *Internet Protocol*. IPv4 dan IPv6 berfungsi sebagai metode *internetworking* standard, baik pada tingkatan lokal maupun global [14]. Perbedaan yang dapat dirasakan langsung adalah alamat IPv4 ditulis dalam notasi desimal dan mempunyai panjang 32 bit sedangkan alamat IPv6 menggunakan notasi *hexadecimal* 128 bit. IPv4 diimplementasikan di setiap perangkat jaringan dan sistem operasi, sedangkan IPv6 masih dianggap sebagai solusi masa depan dunia jaringan komputer meskipun sudah mulai diimplementasikan oleh beberapa vendor besar. Tulisan ini akan memperlihatkan perbedaan kedua protokol dalam penanganan komunikasi antar node di dalam jaringan lokal dan pengaruhnya terhadap keamanan komunikasi tersebut.



Gambar 1. Contoh jaringan yang terdiri dari 3 LAN, dimana LAN 1 dan LAN 3 terdiri dari masing-masing 3 *node* dan LAN 2 merupakan penghubung antara LAN 1 dan LAN 2. Setiap *node* di dalam LAN mempunyai pasangan IP dan MAC *address* [4]

ARP dan DHCP Pada IPv4

Address Resolution Protocol

Address Resolution Protocol (ARP) didefinisikan di dalam RFC 826 [5]. Protokol ini bertugas memetakan IP *address* yang merupakan alamat pada lapisan Network ke MAC *address* pada lapisan Data Link. ARP bekerja dalam proses komunikasi *node-node* di dalam suatu *Local Area Network* (LAN). Gambar 1 memperlihatkan adanya 3 LAN yang saling terhubung melalui Router R1 dan R2. Protokol ARP pada komputer A hanya bertugas memetakan IP *address* ke MAC *address* atau sebaliknya dari *node-node* yang terdapat di dalam LAN 1, yaitu node A, B dan *interface* pertama router R1.

Kedua komputer yang terdapat pada LAN 1, yaitu A dan B, mempunyai pasangan IP *address* dan MAC *address*, masing-masing [IP_A, MAC_A] dan [IP_B, MAC_B].

Sedangkan pada LAN 3 terdapat komputer C dan D dengan informasi [IP_C, MAC_C] dan [IP_D, MAC_D]. Router R1 dan R2, masing-masing mempunyai 2 IP *address* meskipun setiap router hanya mempunyai satu MAC *address*. Router R1 menghubungkan LAN 1 dengan LAN 2, sedangkan router R2 menghubungkan LAN 2 dengan LAN 3.

Jika pengguna A akan mengirimkan paket ke pengguna B maka terjadi proses berikut. A melakukan *query* ke DNS (pada konfigurasi lokal atau server DNS), dan diperoleh IP *address* IP_B. *Host* A kemudian membuat *frame* data dengan IP_B sebagai nilai dari *field* tujuan dan melewatkannya ke lapisan IP untuk

ditransmisikan. Lapisan IP mengetahui bahwa alamat tujuan berada dalam jaringan yang sama. Tetapi A harus menemukan MAC *address* B. Untuk mendapatkan itu, A mem*broadcast* suatu *packet* menanyakan ”Siapa yang memiliki IP *address* IP_B?”. *Broadcast* ini akan sampai pada semua komputer dalam LAN 1. Hanya komputer B yang akan merespon dengan MAC *address*nya, MAC_B. Jadi ARP bekerja dengan pendekatan *request* dan *reply* ini [4].

Dalam komunikasinya, ARP memanfaatkan 4 pesan (*message*), yaitu [1]:

1. **ARP *Request***. Pesan ini digunakan untuk meminta MAC *address* dari suatu IP *address*. Pesan ini biasanya di*broadcast* ke semua *host* pada jaringan melalui alamat *broadcast* ethernet.
2. **ARP *Reply***. Jawaban dari ARP *Request*. Setiap host yang menerima ARP *Request* akan memeriksa *request* tersebut untuk mengetahui apakah dirinya adalah pemilik IP address yang ada di dalamnya, jika ‘iya’ maka harus memberikan jawaban berupa pesan ARP *Reply* yang salah satu *field*nya mengandung MAC *address* dari IP *address* yang diminta tadi.
3. **RARP** (*Reverse* ARP) ***Request***. Pesan ini meminta IP *address* dari suatu MAC *address*.
4. **RARP *reply***. Pesan ini merupakan jawaban dari RARP *Request*, memberikan IP *address* dari MAC *address* yang berasosiasi.

Host memelihara suatu *cache* ARP *Reply* untuk meminimalkan jumlah ARP *Request* yang di*broadcast*. Saat menerima suatu ARP *Reply* maka *host* melakukan *update* terhadap *cache* ini dengan asosiasi IP *address* ke MAC *address* baru [1]. Sehingga jika dalam periode singkat A ingin berkomunikasi dengan B, ia cukup merujuk ke *cache* ARP lokal, tidak perlu melakukan *broadcast* lagi.

Karena protokol ini bersifat *stateless* [1, 7], maka pada beberapa implementasi (di dalam sistem operasi) dimungkinkan terjadinya *update* terhadap entri di dalam *cache* ARP selama penggunaan. Node-node di dalam LAN dengan bebas dapat mengirimkan pesan ARP *Reply* ke node lain tanpa melihat apakah node tujuan telah mengirimkan pesan ARP *Request* sebelumnya. Ini merupakan titik kunci dari serangan ARP [2].

Keamanan ARP

Serangan terhadap ARP berbentuk ARP *poisoning*. Teknik ini menggunakan paket ARP *Request* dan *Reply* palsu untuk meng*update cache* ARP dari *node* target. *Node* target dibuat yakin bahwa MAC address dari *node* penyerang merupakan MAC address dari IP *address* tertentu yang diharapkan. Jadi, penyerang dapat mengawasi paket yang dikirim oleh *node* target ke tujuan asli karena paket tersebut sebenarnya terkirim ke *node* penyerang sebelum dilanjutkan ke penerima sesungguhnya [6]. Teknik ini dikenal juga sebagai ARP *spoofing* dan merupakan basis bagi serangan yang lebih kompleks seperti *sniffing*, *connection hijacking*, *connection spoofing*, dan *denial of service* [1].

Kondisi perlu dari serangan ARP adalah penyerang harus mendapatkan akses jaringan dan mengetahui informasi mengenai IP dan MAC *address* dari beberapa komputer di dalam jaringan. Secara garis besar skenario infeksi *cache* ARP adalah sebagai berikut [2]:

1. Katakanlah di dalam LAN 1 terdapat *node* lain bernama H yang akan melakukan serangan ARP terhadap A dan B. Informasi

IP *address* dan MAC *address* ketiga node tersebut adalah

A (IP = 10.0.0.2, MAC =

AA:AA:AA:AA:AA:AA)

B (IP = 10.0.0.3, MAC =

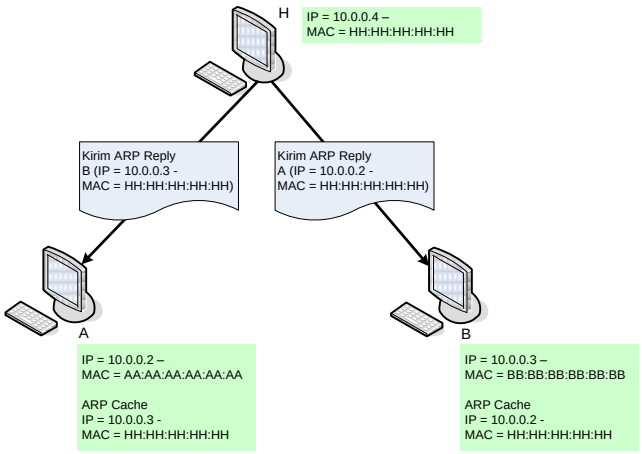
BB:BB:BB::BB:BB:BB)

H (IP = 10.0.0.4, MAC =

HH:HH:HH:HH:HH:HH)

2. H mengirimkan suatu pesan ARP *Reply* ke A mengatakan bahwa IP *address* 10.0.0.3 mempunyai MAC *address* HH:HH:HH:HH:HH:HH. Karena itu, tabel ARP dari A akan berupa IP = 10.0.0.3 – MAC = HH:HH:HH:HH:HH:HH.
3. H juga mengirimkan pesan ARP *Reply* ke B mengatakan bahwa IP *address* 10.0.0.2 mempunyai MAC *address* HH:HH:HH:HH:HH:HH. Tabel ARP di B akan di*update* dengan IP = 10.0.02 – MAC = HH:HH:HH:HH:HH:HH.

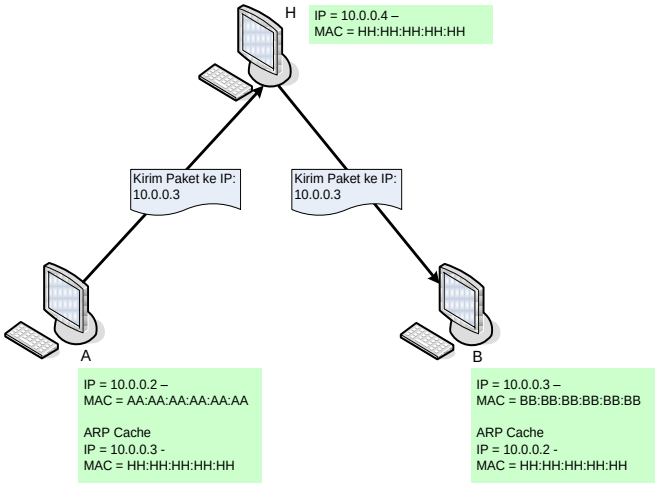
   Proses infeksi *cache* ARP dari node A dan B diperlihatkan pada gambar 2.



Gambar 2. Proses infeksi terhadap cache ARP A dan B oleh H

4. Pada saat A ingin mengirim suatu pesan ke B, karena MAC *address* B dalam tabel ARP A adalah HH:HH:HH:HH:HH:HH maka A akan mengirimkan ke H, bukan ke B. H menerima pesan ini, memrosesnya dan kemudian meneruskan ke B.
5. Jika B mengirimkan suatu pesan ke A, terjadi proses seperti sebelumnya.
6. Karenanya, H bertindak sebagai *man-in-the-middle* untuk menerima dan meneruskan pesan-pesan antara A dan B. H dapat mengubah pesan sebelum mengirimnya ke mesin tujuan.

   Gambar 3 memperlihatkan fungsi H sebagai *man-in-the-middle*.



Gambar 3. Node H bertindak sebagai *man-in-the-middle* dalam penyampaian pesan pesan antara A dan B

Agar *cache* ARP pada suatu *node* tidak terinfeksi (*poisoned*) salah satunya dengan membuat *cache* tersebut statis [1]. Jika *cache* ARP dibuat statis maka *node* tidak akan memroses suatu ARP *Reply*, tidak seperti *cache* ARP dinamis sebelumnya. Ini tidak praktis bagi jaringan besar karena asosiasi IP *address* ke MAC *address* yang tepat dari setiap *node* harus sudah disediakan di dalam *cache* dari setiap *node* sebelum itu dibuat statis. Jika satu *node* berubah MAC addressnya misalnya karena penggantian NIC maka *cache* ARP pada semua *node* perlu di*update* secara manual.

Selain melalui teknik entri *cache* statis, beberapa teknik yang umum digunakan untuk mendeteksi dan mengurangi adanya serangan ARP adalah

1. **Secure ARP** (S-ARP) [8].

   Protokol S-ARP menambahkan suatu skema integritas dan otentikasi terhadap pesan ARP *Reply* untuk mencegah datangnya serangan ARP *poisoning*. Karena S-ARP dibangun di atas ARP maka spesifikasinya, seperti pertukaran pesan, *timeout*, *cache*, mengikuti *standard* dalam

ARP [5]. Dalam rangka memelihara kompatibilitas dengan ARP, suatu *header* tambahan disisipkan pada ujung pesan protokol *standard* untuk membawa informasi otentikasi. Dengan cara ini, pesan dari *node* yang telah menerapkan S-ARP juga dapat diproses oleh *node* yang tidak menjalankan S-ARP, meskipun dalam suatu LAN yang *secure* ARP semua *node* sebaiknya memasang S-ARP.

Node-node yang menjalankan protokol S-ARP tidak akan menerima pesan-pesan yang tidak terotentikasi kecuali ditentukan di dalam daftar *node* yang telah dikenal. Kebalikannya, *node* yang menjalankan protokol ARP klasik akan dapat menerima pesan yang terotentikasi. Suatu LAN campuran tidak direkomendasikan dalam suatu lingkungan produktif karena bagian yang menjalankan ARP tradisional masih menjadi subyek dari ARP *poisoning*. Lebih jauh, daftar yang *node* yang tidak menjalankan S-ARP harus diberikan kepada setiap *secured node* yang harus berkomunikasi dengan *unsecured node*. Interoperabilitas dengan protokol ARP *insecure* diberikan hanya bagi kejadian luar biasa dan sebaiknya selalu dihindari. Ini ditujukan untuk digunakan hanya selama fase transisi menuju suatu LAN yang S-ARP *enabled* secara penuh.

Protokol S-ARP merupakan solusi permanen bagi ARP *spoofing* tetapi kelemahan utamanya adalah keharusan melakukan perubahan terhadap *stack* jaringan dari semua *node*. Tentu sangat tidak *scalable* jika harus meng*upgrade stack* semua sistem operasi apalagi pada perangkat keras dengan ruang *memory* sangat terbatas seperti switch dan router. S-ARP menggunakan enskripsi *Digital Signature Algorithm* (DSA). Karena itu dibutuhkan *overhead* tambahan untuk melakukan kalkulasi kriptografi meskipun pengusulnya mengklaim bahwa *overhead* tersebut tidak signifikan [7].

Kelemahan dari S-ARP dapat diperbaiki melalui implementasi protokol *Secure Unicast* ARP (S-UARP). Mekanisme pada S-UARP menangani pemetaan IP *address* ke MAC *address* dari semua *node* di dalam jaringan memanfaatkan satu layanan terpusat, sebuah DHCP+ server, yaitu server *standard* yang ditambahkan beberapa fitur pengamanan khusus. Detail dari teknik ini dapat dilihat di dalam [4].

2. *Patch* Berbasis Kernel

Teknik *patch* seperti Anticap [9] dan Antidote [10] mencoba untuk melindungi node dari ARP *spoofing* pada suatu tingkatan *node*. Anticap tidak membolehkan *update cache* ARP dengan suatu ARP *Reply* yang membawa MAC *address* berbeda daripada yang telah ada di dalam *cache*. Ini mematikan fitur ARP *Reply* otomatis yang legal, berlawanan dengan spesifikasi protokol ARP [5]. Sedangkan Antidote, saat menerima suatu ARP *Reply* dengan MAC *address* yang berbeda dengan *cache* sebelumnya mencoba untuk memeriksa apakah MAC sebelumnya tersebut masih aktif. Jika MAC *address* yang berada di *cache* tersebut masih aktif maka *update* ditolak dan MAC *address* yang "legal" tersebut ditambahkan ke daftar *banned addresses*.

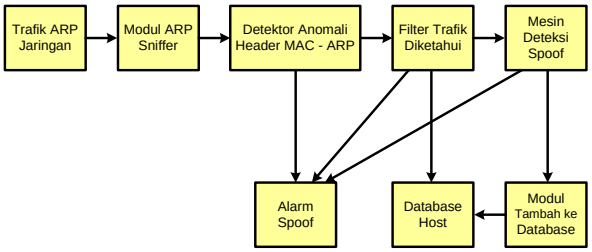
Dua teknik *patch* ini bersandar pada fakta bahwa entri ARP dalam *cache* harus terlegitimasi. Ini menciptakan *race condition* antara penyerang dan korban. Jika penyerang berhasil meletakkan entri ARP palsunya ke dalam *cache* sebelum *node* sesungguhnya, maka MAC *address* yang sesungguhnya akan diblok. Jadi, pembelajaran yang salah dapat menyebabkan *tool* ini gagal dalam mendeteksi ARP *spoofing* [7].

3. Deteksi Pasif

Dalam deteksi pasif dilakukan *sniffing request* atau respon ARP pada jaringan dan membangun suatu database pemetaan MAC *address* ke IP *address*. Jika ditemukan suatu perubahan pemetaan dalam suatu trafik ARP maka dimunculkan suatu alarm dan menyatakan bahwa sedang terjadi suatu serangan ARP *spoofing*. Tool yang paling popular dalam kelompok ini adalah ARPWatch [7].

4. Deteksi Aktif

Ramachandran [7] mengusulkan sistem deteksi aktif yang sekaligus juga dapat melakukan pekerjaan yang dilakukan oleh administrator jaringan pada teknik deteksi pasif seperti mengabaikan *update cache* ARP dengan MAC *address* penyerang. Gambar 4 memperlihatkan garis besar dari model yang diusulkan beserta interaksi antar modul di dalamnya.



Gambar 4. Interaksi antar modul dalam mekanisme deteksi Aktif

Dynamic Host Configuration Protocol

Dynamic Host Configuration Protcol (DHCP) adalah protokol Internet yang bertugas memberikan informasi TCP/IP secara otomatis kepada komputer dan perangkat jaringan lain yang menggunakan protokol TCP/IP. Protokol ini merupakan pengembangan dari protokol manajemen IP jaringan BOOTP (Bootstrap Protocol) dengan menambahkan kemampuan alokasi otomatis alamat-alamat jaringan yang *reusable* dan pilihan konfigurasi tambahan.

DHCP menyediakan dua fungsi utama. Pertama adalah sebagai *persistent storage* (media penyimpanan menetap) dari parameter-parameter jaringan bagi *client*. Model *persistent storage* dari DHCP menyatakan bahwa layanan DHCP menyimpan suatu entri *key-value* setiap *client*, dimana *key* merupakan pengenal unik dan *value* mengandung parameter konfigurasi bagi *client*. Pengenal unik *client* dapat berupa suatu nomor subnet IP dan pengenal unik di dalam subnet tersebut. Kedua adalah mengalokasikan alamat jaringan client secara temporer atau permanen. Mekanisme alokasi menjamin tidak terjadi realokasi terhadap alamat yang telah diberikan ke suatu client dan mencoba untuk memberikan alamat jaringan sama setiap kali client meminta suatu alamat. Ini berarti bahwa DHCP tidak akan memberikan alamat IP *address* yang sama untuk lebih dari satu *node* pada waktu yang sama. Bahkan setelah *node* di*reboot*, DHCP tetap memelihara konfigurasi dari node [12].

DHCP mendukung tiga mekanisme alokasi IP address, yaitu:

1. Metode alokasi otomatis yang memberikan IP *address* permanen kepada setiap *client*.
2. Metode alokasi dinamis yang memberikan IP *address* kepada *client* untuk periode waktu terbatas atau sampai *client* secara eksplisit melepas alamat tersebut.
3. Metode alokasi manual yang memungkinkan pengelola jaringan memberikan suatu IP *address* kepada *client*.

Suatu jaringan dapat menggunakan satu atau lebih mekanisme ini, tergantung pada kebijakan dari pengelola jaringan.

Proses alokasi IP memanfaatkan DHCP dimulai oleh *client* dengan mengirimkan suatu permintaan *broadcast* bernama pesan DHCPDISCOVER yang berisi MAC *address* client tersebut ke jaringan dan mencari DHCP *server*. Setelah menerima pesan DHCPDISCOVER, server menentukan suatu alamat yang tepat (jika ada) untuk diberikan kepada *client* sesuai dengan ketersediaan dan kumpulan kebijakan pemanfaatan yang diatur pada *server*. Kemudian *server* secara temporer mencadangkan alamat tersebut untuk client dan mengirim balik (mengembalikan) pesan DHCPOFFER kepada *client*. DHCPOFFER mengandung informasi IP *address* dan seting TCP/IP lain yang dapat digunakan oleh *client* untuk berkomunikasi pada jaringan. *Client* kemudian mengirimkan pesan DHCPREQUEST, sehingga server mengetahui bahwa *client* memang bermaksud untuk menggunakan IP *address* tersebut. *Server* mengirimkan pesan DHCPACK, mengkonfirmasikan bahwa *client* telah diberikan pinjaman pada terhadap alamat tersebut selama periode waktu yang ditentukan oleh *server*. Drooms [12] membuat suatu diagram *timeline* dari pertukaran pesan antara *client* dan *server* DHCP pada saat alokasi alamat jaringan baru.

Keamanan DHCP

DHCP telah menjadi layanan kritis pada banyak lembaga atau perusahaan, namun keamanan *server* ini masih sangat sering dilewatkan dalam penanganan keamanan jaringan. Jika tidak terdapat pemrosesan otentikasi selama pertukaran pesan DHCP antara *client* dan *server*, maka *server* DHCP tidak mengetahui apakah *client* yang meminta *address* merupakan *client* yang sah di dalam jaringan, dan *client* tidak mengetahui apakah *server* DHCP yang memberikannya *address* merupakan *server* yang sah. Kemungkinan hadirnya *client* dan *server* nakal pada jaringan dapat menyebabkan berbagai jenis masalah. Sebagai contoh, *client* dapat menjadi subyek serangan *Denial of Service* (DoS) melalui penggunaan *server* DHCP gadungan, atau terjadinya kesalahan konfigurasi.

Ancaman terhadap DHCP datang dari *node-node* di dalam jaringan yang sama. Serangan yang khusus untuk client DHCP kemungkinan berupa pembuatan *server* palsu yang bertujuan menyediakan informasi konfigurasi salah kepada *client*. Ancaman lain bagi *client* adalah saat *server* secara tidak sengaja salah dalam konfigurasi dan memberikan informasi tidak tepat kepada *client*. Ancaman yang khusus untuk *server* DHCP adalah berupa suatu *invalid client* yang berpura-pura sebagai *client* sungguhan. Motivasinya mungkin untuk ”mencuri layanan”, atau untuk mengelak *auditing* karena alasan tidak baik. Ancaman umum terhadap *client* dan *server* adalah serangan DoS sumber daya. Serangan ini biasanya menghabiskan alamat-alamat valid, menguras *bandwidth* CPU atau jaringan, dan hadir saat terdapat sumber daya yang di*share* [13].

Pada prakteknya, serangan terhadap DHCP menyerupai teknik serangan ARP dengan sedikit modifikasi. Beberapa jenis serangan yang umum terjadi serta solusinya banyak diimplementasikan pada perangkat jaringan yang bekerja pada lapisan Data Link adalah [3]

1. Pembuatan *server* DHCP palsu, seperti disebutkan di atas. Salah satu solusi terhadap serangan ini adalah pemasangan DHCP *snooping* pada switch yang menjadi sentral bagi *client* untuk terhubung ke jaringan. Respon DHCP yang diterima dari *port* yang ”dipercaya” akan diproses sedangkan yang diterima dari *port* “tidak jelas” akan dibuang, sehingga menghindarkan *client* DHCP mendapatkan IP *address* dari *server* yang tidak resmi.
2. *Man-in-the-Middle*, seperti pada serangan ARP. Beberapa vendor perangkat switch melindungi serangan ini dengan memasukkan fitur ARP *detection* yang menggunakan entri DHCP *snooping* statis dan dinamis untuk mendeteksi paket-paket ARP *invalid* dan mengabaikannya.
3. IP/MAC *Spoofing*.

   Serangan ini umumnya berupa *spoofing* terhadap alamat MAC, IP atau kombinasi keduanya. Penyerang mengirimkan paket dengan alamat asal palsu untuk mengakses jaringan atau mendapatkan beberapa *privilege* yang tergantung pada alamat IP atau MAC. Metode ini juga digunakan dalam serangan DoS. Perlindungan terhadap serangan ini adalah dengan menyediakan fitur IP *filtering*. Fitur ini dapat diterapkan pada suatu *port*, sehingga switch dapat menyaring paket pada *port* tersebut dengan mencocokkan alamat asal paket dengan entri DHCP *snooping* statis dan dinamis, dan paket yang tidak memenuhi kualifikasi diabaikan. Fitur ini sekaligus dapat menghindarkan konflik alamat.
4. Pembanjiran (*flooding*) Paket DHCP. Jika penyerang mengirimkan sejumlah besar permintaan DHCP ke suatu *server* DHCP maka semua IP *address* pada server akan diserahkan. Ini mengakibatkan banyak *client* tidak memperoleh IP address. Jika terdapat switch dengan DHCP *snooping* antara penyerang dan server maka switch dan *server* tersebut dapat sangat terbebani ketika memroses paket DHCP.

   Salah satu solusi untuk masalah ini adalah memberikan batasan jumlah paket DHCP pada switch, server DHCP atau keduanya. Jika diimplementasikan pada switch, tindakan lanjutan yang cukup efektif adalah mematikan *port* yang menghubungkan switch dengan sumber serangan.

Salah satu bentuk perlawanan terhadap DoS adalah dengan hanya menyediakan sejumlah terbatas IP *address*. Namun peluang datangnya serangan tetap ada terutama berupa kehadiran *client* atau *server* DHCP palsu dengan berbagai kombinasi serangan. Pengamanan lebih lanjut dapat menggunakan mekanisme otentikasi mengikuti dasar yang terdapat dalam RFC-3118 [13].

Tujuan otentikasi terhadap pesan DHCP adalah untuk melindungi jaringan dari gangguan *node* nakal dan membangun asosiasi yang aman antara *client* dan *server* DHCP. Dalam rangka validasi pesan DHCP, penerima memeriksa MAC (*message authentication code*) yang terdapat di dalam pesan DHCP yang datang. Jika nilai MAC yang diterima tidak cocok dengan nilai MAC yang dihitung, penerima menolak pesan DHCP yang mengikutinya. Pada saat perhitungan nilai MAC, pengirim atau penerima menggunakan *keyed-hashing for message authentication* (HMAC).

Ju dan Han [11] membuat diagram *timeline* dari pertukaran pesan DHCP yang berbeda dengan *timeline* yang diusulkan Drooms [12]. Pada satu jaringan boleh terdapat lebih dari satu *server* DHCP termasuk yang tidak dilengkapi fitur otentikasi. Beberapa *server* DHCP dapat mengirimkan pesan DHCPOFFER, tetapi IP *address* untuk *client* hanya akan dialokasikan oleh *server* DHCP yang otentik setelah mengikuti prosedur otentikasi. Detail dari rancangan dan implementasi teknik ini dapat dilihat di dalam [11].

Keamanan *Neighbor Discovery* IPv6

Neighbor Discovery

Pada IPv6, bagian *interface identifier* (ID) dari alamat lapisan *Network* IPv6 (IP *address*) secara langsung diturunkan dari alamat lapisan *Data Link* (MAC *address*). Alamat IPv6 lengkap tersebut kemudian digunakan untuk berkomunikasi pada tingkatan global [18]. Pembuatan IP *address* secara otomatis ini dinamakan *stateless autoconfiguration* [16]. Fitur ini dapat meniadakan fungsi dari layanan DHCPv6 [20], meskipun masih dapat digunakan jika diperlukan penetapan IP *address* yang terpusat. Tidak ada perbedaan mekanisme kerja antara DHCP untuk IPv4 dan DHCPv6. *Node-node* di dalam jaringan IPv6 menggunakan *Neighbor Discovery Protocol* (NDP) [21] untuk saling berinteraksi melalui pesan-pesan ICMPv6 [22]. Ini sekaligus menggantikan peran dari protokol ARP, *Router Discovery*, dan *Redirection* pada IPv4 dan menawarkan beberapa fungsi baru [19]. NDP memungkinkan node IPv6 menemukan alamat lapisan *Data link node* lain dan router pada *local link* (dalam LAN), mendeteksi saat *local node unreachable* (node dalam LAN tidak dapat diakses), menyelesaikan alamat ganda dan oleh router digunakan untuk memberitahu adanya *route* baru kepada *node* (*redirect*) [17].

NDP mendefinisikan 5 pesan yang digunakan selama menjalankan peran *discovery* di atas [15,19]:

1. *Router Solicitation* (RS), dikirimkan oleh *node* yang ingin mengkonfigurasi dirinya, menanyakan informasi mengenai router dan prefiks *on-link* (prefiks dari IPv6 untuk LAN dimana node berada).
2. *Router Advertisement* (RA), dikirimkan oleh router secara periodik atau sebagai respon terhadap *Router Solicitation*, mengandung alamat *default gateway*, validitas router, daftar prefiks IPv6 yang ditangani oleh router tersebut, MTU, opsi-opsi *Mobile* IPv6 dan lain-lain.
3. *Neighbor Solicitaion* (NS), dikirimkan oleh suatu *node* untuk mendapatkan informasi tetangga (*node* lain di dalam LAN) mencakup alamat fisik (MAC), *reachability* atau *availability* dari suatu alamat selama *Duplicate Address Detection* (DAD).
4. *Neighbor Advertisement* (NA), dikirimkan secara periodik oleh suatu *node* atau sebagai jawaban terhadap *Neighbor Solicitation* dalam rangka mengabarkan alamat fisik dari suatu *interface*nya.
5. *Redirect*, digunakan oleh router untuk mengumumkan route yang lebih baik kepada *node-node* di dalam *link-local*.

Proses pertama yang dilakukan agar node dapat berkomunikasi dalam jaringan adalah konfigurasi antarmuka jaringan melalui mekanisme *autoconfiguration*. Proses ini dimulai dengan pembangkitan otomatis suatu

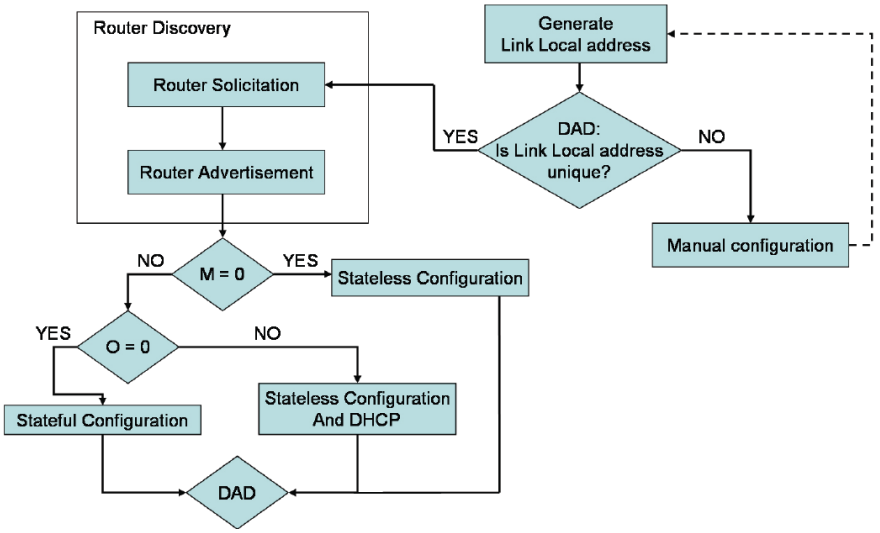
alamat IPv6 link local. Karena diperbolehkan juga konfigurasi alamat lapisan data link secara manual maka ada kemungkinan terjadi duplikasi alamat *link local* pada link jaringan yang sama. Kasus duplikasi ini akan ditangani melalui DAD. Diagram dari proses ini diperlihatkan pada gambar 5 [15].

Selanjutnya adalah proses resolusi alamat tetangga, yaitu menemukan MAC *address* dari *node* yang mempunyai IP *address* tertentu sebagaimana yang dilakukan ARP pada IPv4.

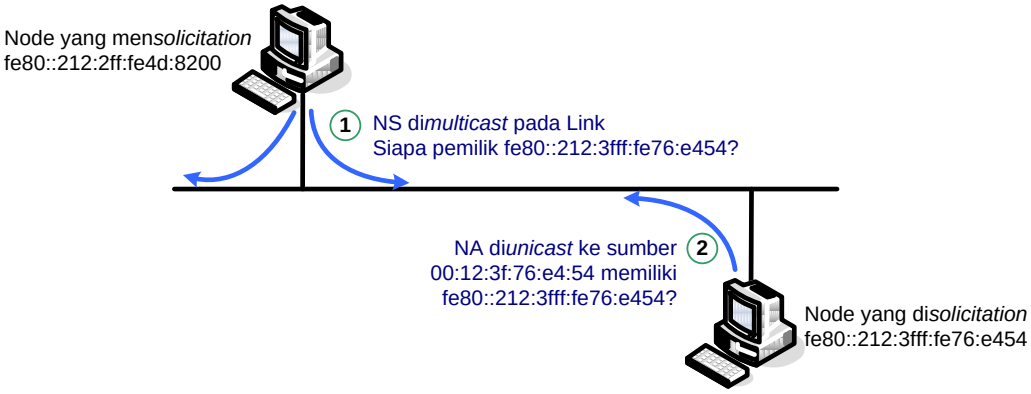
Mekanisme ini dinamakan *Neighbor Discovery* dengan tahap utama adalah [15,17,19]:

1. *Node* yang memerlukan MAC *address* tetangganya mengirimkan pesan NS ke alamat *multicast* dari jaringan lokal (*local-link*).
2. *Node* target "*listening*" pada alamat *multicast*. Begitu menerima *solicitation*, *host* ini membalas dengan pesan NA.

Operasi *standard* ini diperlihatkan pada gambar 6. Pada tahap lanjut pesan yang terlibat dapat dilindungi dengan IPSec *Authentication Header* (AH) [17].



Gambar 5. Proses konfigurasi *host* otomatis pada IPv6



Gambar 6. Proses dasar *Neighbor Discovery*

Agar *node* IPv6 dapat berkomunikasi dengan *node* pada jaringan lain maka dijalankan mekanisme *router discovery* [15], berupa:

1. Host yang baru dikonfigurasi melalui *autoconfiguration* mengirimkan pesan RS ke semua *router* pada *local link* menggunakan alamat *multicast* yang sesuai spefisikasi pengalaman IPv6.
2. Semua router memberikan respon berupa suatu pesan RA. Pesan ini mengandung beberapa informasi penting mengetani *routing* seperti alamat lapisan *data link* router, *lifetime*nya dan prefiks jaringan.
3. Begitu menerima RA, *host* meng*update field* terkait dari daftar *defaut router*nya, *cache neighbor* dan daftar prefiks.

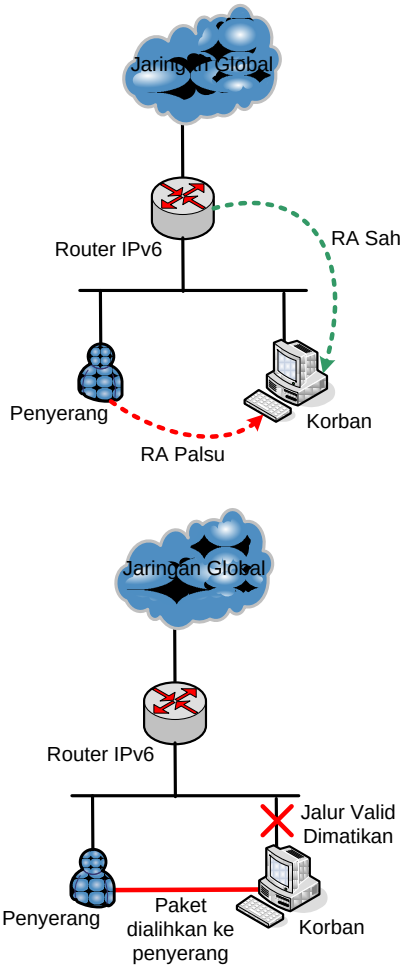
Keamanan

Neighbor discovery, *router discovery* dan *auto configuration* merupakan mekanisme yang memudahkan konfigurasi jaringan IPv6. Namun, fleksibilitas dan keamanan sifatnya saling berlawanan [18]. Setidaknya ada 3 tipe tipe serangan terhadap NDP [19], yaitu:

1. *Redirect*. Penyerang mengalihkan paket yang ditujukan ke suatu penerima ke *node* lain dalam jaringan.
2. DoS. Penyerang mencegah komunikasi antara *node* yang diserang dengan semua *node* lain atau suatu alamat tujuan tertentu.
3. *Flooding* DoS. Penyerang mengalihkan trafik *node* lain ke *node* korban.

Serangan di atas juga dapat dikelompokkan menjadi serangan terkait *routing* atau *non-routing*. Serangan *non-routing* hanya dilakukan terhadap fungsi *neighbor discovery* murni. Penyerang dapat menyebabkan kegagalan proses DAD pada *host* yang sedang melakukan konfigurasi otomatis dengan melakukan *spoofing* terhadap NA. Ini mengakibatkan *host* tersebut tidak akan mengakui (menggunakan) IP *address* yang diperolehnya. Host demikian mungkin tidak akan pernah dapat mendapatkan alamat IPv6 yang valid. Pengiriman *advertisement* yang dipalsukan juga dapat menimpa *neighbor cache* pada *node*. Kemudian, beberapa paket dapat terkirim ke tujuan yang salah sehingga dapat menyebabkan terjadinya DoS.

Pada saat suatu layanan atau *host* tidak memberikan respon, mekanisme *Neighbor Unreachability Detection* (NUD) dijalankan untuk memeriksa *reachability* dari *node*. Pesan NS dikirim ke *node* tersebut dan diharapkan menerima kembalian NA. Penyerang tetap mengirimkan NA palsu. Jika proses NUD gagal maka trafik pada lapisan lebih tinggi dihentikan. Tetapi karena mekanisme menganggap *host* masih *reachable* maka yang terjadi adalah *delay* tanpa batas dari proses deteksi keberadaan layanan.



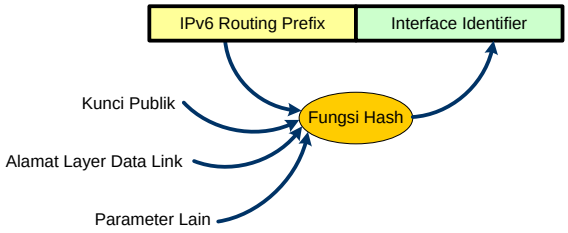
Gambar 7. Pengalihan trafik dengan RA yang dipalsukan

Serangan *routing* berkaitan dengan mekanisme *router discovery*. Jika penyerang mengirimkan suatu RA yang telah diubah (palsu) maka *host* akan menjadikan penyerang sebagai *default router*. Jadi, penyerang dapat memindahkan trafik dari *host* atau menjadikan dirinya sebagai *man-in-the-middle* seperti diperlihatkan pada gambar 7. Jika penyerang

kemudian meneruskan trafik ke router yang sah, ini menjadi transparan bagi *host*. Jika suatu *invalid prefix* dimasukkan ke dalam *advertisement* maka akan digunakan oleh *host* untuk *autoconfiguration*. Akibatnya suatu *invalid source address* akan selalu digunakan dan paket-paket yang dikirim oleh *host* yang salah konfigurasi tersebut dapat tidak pernah mencapai tujuan.

Jika prefiks palsu merupakan suatu prefiks yang valid tetapi dari jaringan atau *link* lain maka *host* akan percaya prefiks ini adalah *on-link* dan tidak akan pernah mengirim paket untuk prefiks ini ke *default router*, dan prefiks ini akan menjadi *unreachable* bagi node. RA palsu demikian juga dapat berupa perubahan pesan RA asli dari router yang sah seperti mematikan *autoconfiguration* bagi *host* atau menetapkan suatu *hop limit* terlalu kecil bagi *routing* paket. *Default router* dapat pula dimatikan dengan menjalankan DoS klasik pada *default router* dan memalsukan *advertisement* sehingga *node-node* percaya semua tujuan adalah *on-link* atau dengan mengubah RA dengan suatu *router lifetime* bernilai nol. Terakhir, penyerang dapat mengirimkan pesan *Redirect* yang dipalsukan dengan suatu *source address* sah untuk mengirimkan paket ke suatu tujuan misalnya ke alamat tertentu *data link* pada *link local* [19].

Salah satu solusi yang disediakan IPv6 dalam menghadapi serangan di atas adalah melalui *SEcure Neighbor Discovery* (SEND) [24]. Sekumpulan opsi baru digunakan dalam rangka memproteksi pesan-pesan *Neighbor Discovery*. Di samping opsi-opsi tersebut, solusi ini mengenalkan beberapa komponen baru dari arsitektur *neighbor discovery*.



Gambar 8. Pembangkitan alamat kriptografis

Komponen pertama adalah *certificate path* (jalur sertifikat). Setiap *host* pada *link* harus dikonfigurasi dengan suatu *anchor* ke router yang mempunyai jalur sertifikat sebelum memilihnya sebagai *default router*. Pesan *Certificate Path Solicitation* dan *Advertisement* digunakan untuk menemukan jalur-jalur tersebut. Teknik *Cryptographically Generated Addresses* (CGA) digunakan untuk memastikan bahwa pengirim pesan *neighbor discovery* merupakan pemilik alamat yang diklaim. Sebelum mengklaim suatu alamat, semua host membangkitkan suatu pasangan kunci publik-privat. Opsi baru bernama CGA digunakan untuk membawa kunci publik dan parameter yang berasosiasi dengannya. Gambar 8 memperlihatkan bagaimana alamat tersebut dibangkitkan.

Opsi RSA *Signature* digunakan untuk mengamankan semua pesan *Neighbor* dan *Router Discovery*. Opsi ini melindungi integritas dari pesan dan mengotentikasi identitas dari pengirimnya. Otoritas dari kunci publik dibangun dengan proses delegasi otorisasi menggunakan sertifikat atau melalui mekanisme bukti kepemilikan alamat menggunakan CGA atau dengan keduanya.

Serangan ulangan dicegah melalui opsi Timestamp dan Nonce. Opsi Timestamp menyediakan proteksi ulangan tanpa status yang dibangun sebelumnya atau tanpa *sequence number* (nomor urut). Ketika pesan digunakan dalam pasangan *solicitation-advertisement* maka digunakan opsi Nonce.

Infrastruktur yang ditawarkan oleh SEND dapat dikatakan sangat sulit untuk diimplementasikan terutama karena penggunaan sertifikat. Pemanfaatan CGA akan menyebabkan waku *treatment* terhadap pesan *Neighbor Discovery* menjadi lebih lama. Ini dapat menghadirkan beberapa bahaya lain seperti masalah konfigurasi pada lingkungan Mobile IPv6 [24]. Perubahan terhadap sistem operasi yang beragam juga tidak mudah dilakukan.

KESIMPULAN

Dari uraian mengenai mekanisme *node discovery* di dalam LAN di atas, baik pada jaringan IPv4 maupun IPv6, menggunakan protokol ARP ataupun *Neighbor Discovery*, ada beberapa hal yang dapat disimpulkan mengenai keamanan ARP dan DHCP pada dua protokol IP tersebut.

IPv6 membawa banyak perbaikan terhadap IPv4, namun tidak ada mekanisme pengamanan melekat yang ditambahkan ke dalam IPv6 terkait dengan serangan ARP dan DHCP. Berbagai bentuk serangan menyerupai ARP dan DHCP tetap memiliki peluang besar. S*tateless autoconfiguration* sangat memudahkan konfigurasi *interface* jaringan, akan tetapi pesan-pesan yang dipertukarkan dapat dipalsukan oleh penyerang bahkan dapat menyebabkan penolakan akses terhadap suatu perangkat, misalnya penggagalan dari mekanisme DAD. Solusi terhadap masalah ini adalah melalui implementasi konsep port ”terpercaya” pada perangkat jaringan yang beroperasi pada lapisan Data Link model OSI seperti switch sebagai diberlakukan pada IPv4 [23]. Mekanisme SEND dapat digunakan untuk mengamankan pesan-pesan dalam NDP namun sangat sulit diimplementasikan dan akan memperlambat proses komunikasi antar node karena melibatkan CGA.

Sampai saat ini masih sulit memperoleh tool yang dapat membantu mendeteksi atau menghentikan penyalahgunaan DHCP, *autoconfiguration* atau *Neighbor Discovery* dalam IPv6. Sebenarnya pesan-pesan ini dapat disaring pada *router* atau *firewall* seperti terhadap pesan ICMP tetapi karena sebagian besar serangan ARP dan DHCP bersifat lokal maka fitur proteksi tersebut seolah tanpa arti. Pada IPv4 terdapat *tool* ARPWatch yang sangat membantu memonitor serangan ARP. Tanpa kemampuan untuk mendeteksi penyalahgunaan pesan-pesan ND dan mengamankan transportasinya maka pendekatan terbaik masih terbatas pada pemanfaatan entri *neighbor* statis, terutama pada lingkungan kritis. Pendekatan terakhir ini menjadi sulit pada jaringan skala besar.

Intinya, tidak ada peningkatan pengamanan terhadap serangan ARP dan DHCP pada IPv6. Karena itu, teknik pengamanan yang dapat dilakukan juga masih sama sebagaimana diterapkan pada jaringan berbasis IPv4. Perbedaan yang mendasar yang harus diperhatikan hanya pada arsitektur dan cara kerja dari protokol yang lebih baru tersebut, misalnya panjang IP address dan mekanisme operasional dari protokol yang digunakan.

DAFTAR PUSTAKA

[1] Fewer, S. (2007) : **ARP Poisoning – An Investigation Into Spoofing the Address Resolution Protoco**l. [online] http://www.harmonysecurity.com /files/HS-P004 _ARPPoisoning.pdf

[2] Thuc N.D. dkk. (2006) : A Software Solution for Defending Against Man-in-the-Middle Attacks on WLAN. [online] http://www.utdallas.edu /~htv041000/files/MiMA.pdf

[3] H3C (2008) : DHCP Security Features Technology White Paper. Hangzhou H3C Technologies. [online] http://www.h3c.com/portal/res /200802 /03/20080203_320314_DHCP Security Features Technology White Paper(V1.00)_333753_57_0.pdf

[4] Issac B. (2009) : Secure ARP and Secure DHCP Protocols to Mitigate Security Attacks. *International Journal of Network Security*, Vol. 8 No. 1, PP.102-113. [online] http://ijns.femto.com.tw/contents/ijns-v8-n2/ijns-v8-n2.html

[5] Peterson, D.C. (1982) : Ethernet Address Resolution Protocol, RFC-826. [online] http://www.faqs.org/rfcs/rfc826.html.

[6] Nachreiner, C (2003) : Anatomy of an ARP Poisoning Attack. [online] http://www. Watch-guard.com/infocenter/editorial/135324.asp

[7] Ramachandran V. dan Nandi S. (2004) : Detecting ARP Spoofing: An Active Techniques. [online] http://www.springerlink.com/index /l421371736251342.pdf

[8] Bruschi B., Ornaghi, A., dan Rosti E. (2003) : S-ARP: a Secure Address Resolution Protocol, 1*9th Annual Computer Security Applications Conference*. [online] http://www.acsac.org/2003/papers/111.pdf

[9] Barnaba, M. (2003) : Anticap. [online] http://cvs.antifork.org/cvsweb.cgi /anticap, 2003

[10] Teterin (2003) : Antidote [online] http://online.securityfocus.com/archive /1/299929

[11] Ju, H.I, dan Han, J.W. (2005) : DHCP Message Authentication with an Effective Key Management, *Proceedings of World Academy of Science, Engineering and Technology* Vol. 8. [online] http://www.waset.org /pwaset/v8/v8-25.pdf

[12] Droms, R. (1997) : Dynamis Host Configuration Protocol, RFC-2131. [online] http://www.ietf.org/rfc/rfc2131.txt

[13] Droms, R., dan Arbaugh, W. (2001): Authentication for DHCP messages, RFC-3118. [online] http://tools.ietf.org/html/rfc3118

[14] Wikipedia (2008) : IPv4. [online] http://en.wikipedia.org/wiki/IPv4

[15] Hines, A. (2004) : Neighbor Discovery in IPv6. [online] http://wwwcs.uni-paderborn.de/cs/ag-madh/WWW/Teaching/2004SS /AlgInternet /Submissions/17-neighbour-discovery-protocol-in-IPv6.pdf.

[16] Thomson, S., dan Narten, T. (1998) : IPv6 Stateless Address Autoconfiguration, RFC-2462. [online] http://www.ietf.org/rfc/rfc2462.txt

[17] Arkko J., dkk. (2002) : Securing IPv6 Neighbor and Router Discovery. [online] http://www.tml.tkk.fi/~pnr/publications/ WiSe2002-Arkko.pdf

[18] Majstor, F. (2003) : Does IPv6 Protocol Solve All Security Problems of IPv4?, *Information Security Solutons Europe*, Vienna Austria. [online] http://www.6journal.org/archive/00000183/01/IPv6_security_paper.pdf

[19] Beck, F. (2007) : Monitoring the Neighbor Discovery Protocol, *The 2nd International Workshop on IPv6 Today – Technology and Deployment*, Guadeloupe.

[20] Droms, R. (2003) : Dynamic Host Configuration Protocol for IPv6 (DHCPv6), RFC-3315. [online] http://tools.ietf.org/html/rfc3315T.

[21] Narten, T. dkk. (2007) : Neighbor Discovery for IP version 6 (IPv6), RFC–4861. [online] http://tools.ietf.org/html/rfc4861

[22] Conta, A., Deering, S., dan Gupta, M. (2006) : Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC-4443. [online] http://tools.ietf.org/html/rfc4443

[23] Convery, S., dan Miller, D. (2004) : IPv6 and IPv4 Threat Comparison and Best-Practice Evaluation (v1.0). [online] http://www.seanconvery.com/v6-v4-threats.pdf

[24] Arkko, J., dkk. ((2005) :SEcure Neighbor Discovery (SEND), RFC-3971. [Online] Available: http://www.ietf.org/rfc/rfc397