IPv6におけるノードの発見とプロミスキャスモードの検出

Detection of promiscuous IPv6 nodes

佐々木　太一
Taichi Sasaki

太田　貴彦
Takahiko Ohta

佐内　大司
Daiji Sanai

キーワード
IPv6，IoT，プロミスキャス，ネットワーク盗聴，ハッキング，サイバー攻撃，セキュリティ

ローカルネットワーク内のIPv6(Internet Protocol Version 6)ノードを列挙し，そのIPv6ノードがプロミスキャスモードかどうかを判定できる技術を開発した。この技術は，IoT時代のネットワーク資産の管理や，ネットワーク盗聴やハッキングを行うプロミスキャスモードのPCなど悪意ある機器を検出可能とすることで，より高度なセキュリティ管理を実現できる。

We have developed a technology that can enumerate Internet Protocol Version 6 (IPv6) nodes on a local network and determine whether they are in promiscuous mode or not. This technology allows more advanced security management of network assets in the IoT era and enables the detection of malicious devices such as PCs in promiscuous mode used for network tapping and hacking.

1. 背景

IPv4(Internet Protocol Version 4)アドレスの枯渇問題によりIPv6の推進が始まってから既に20年以上が経過している。一般的に広く使われているWindows®はWindows®XPからIPv6を搭載した。そしてWindows Vista®以降のWindowsは，IPv6を優先的に利用する設定に変わった。つまりユーザーが意識しなくても自動的にIPv6が使われている。さらにスマートフォン，Wi-Fi，IoT機器などの加速的な普及によってIPv6はバックグラウンドでも確実に普及している。その結果，このIPv6の普及はネットワークの運用･管理において新しい以下の課題を生み出している。

(1) ネットワーク管理者やシステム管理者の意図しないIPv6の通信が行われている。
(2) セキュリティ監視ソフトウェアやネットワーク監視システムが，IPv6に対応していない。IPv6による攻撃を検知できない。
(3) ネットワーク管理者は，IPv6アドレスとPCとの関連を管理できていない。結果としてIPv6アドレスからPCを特定できない。
(4) 管理されていないIoT機器がサイバー攻撃の起点となる可能性がある。

このような背景からローカルネットワーク内のIPv6ノードを発見し調査する技術が求められている。

2. IPv6 ノードの検出における基本的な技術課題

ネットワークに接続しているノードを列挙するとき，従来のIPv4ネットワークであれば，ネットワークアドレスの範囲内のすべてのアドレスに対してアクセスを試みる手法が有効であった。例えば，192.168.1.1～192.168.1.254のネットワークセグメントであれば，254回のARP (Address Resolution Protocol) リクエストを送信することでノードの有無を確認できた。また，TCP (Transmission Control Protocol)やUDP (User Datagram Protocol)を使ってのアクセステストでもノードの列挙が可能であった。しかしIPv6のアドレス空間は，IPv4の32bitから128bitへと天文学的な数に拡大されている。このためIPv4の場合と同様な総当たりでの検索方法は不可能である。さらにIPv6ではブロードキャストおよびARPの廃止によって，従来のノードを探索する手法自体も使えなくなった。そこで，我々がまず解決しなければいけない課題は，ローカルネットワーク，すなわち管理下にある物理的ネットワーク上に接続しているIPv6ノードを列挙する手法の開発であった。IPv6では，ブロードキャストに代わって多くのマルチキャストアドレスが定義されている。我々の開発目的の1つとして，これらマルチキャストアドレスを活用してネットワーク上の未知のノードを列挙する手法を検討し開発することである。

3. IPv6 ノード検出のアプローチ

我々は，IPv6ノード検出として4つのアプローチを試みた。
(1)IPv6のPingを使用した検出方法
(2)RA (Router Advertisement)を使用した検出方法
(3)パラメータ不正のICMPv6(Internet Control Message Protocol for IPv6)パケットを使った方法
(4)MLD (Multicast Listener Discovery)を使った方法

各手法について説明する。

3.1 IPv6のPingを使用した検出方法

このアプローチではIPv6ノードの有無を確認するためにIPv6のPing機能を使用する。一般的にPingは検査をしたいノードに対してPingリクエストを送信し，その応答によってノードの存在を確認する。しかし先に述べたように，未知のノードを探索する場合にIPv6のアドレス空間は広すぎるので，すべてのアドレスに1つずつPingを送信する手法は現実的に不可能である。そこでIPv6 Pingの宛先アドレスをマルチキャストグループアドレスに対して送信する。例えばオールノードマルチキャストのアドレスである「ff02::1」にして送信する。これによりローカルネットワークに接続しているすべてのIPv6ノードがPingリクエストに応答すると期待できる。しかし残念ながらこの方法では，検出できないノードが多数あった。理由は近年のセキュリティ意識の向上から，存在確認に応答すること自体にセキュリティ上のリスクがあると考えられ，Pingリクエストを受け付けないノードが多いのである。代表的な例としては，Windows10がIPv6 Pingには応答していない。

3.2 RA(Router Advertisement)を使用した検出方法

IPv6ネットワークにおけるグローバルアドレスやゲートウェイの設定は，ルータからのRAにより行われる。具体的にはルータからのRAパケットを受信した各ノードは，RAのPrefix値から新たにIPv6アドレスを算出する。それを重複確認してから利用する。この重複確認にはNS(Neighbor Solicitation)が使われる。この機能を利用して，ルータのふりをしてRAパケットを送信する。その応答のNSパケットを収集すれば，ノードのリストが生成できる。

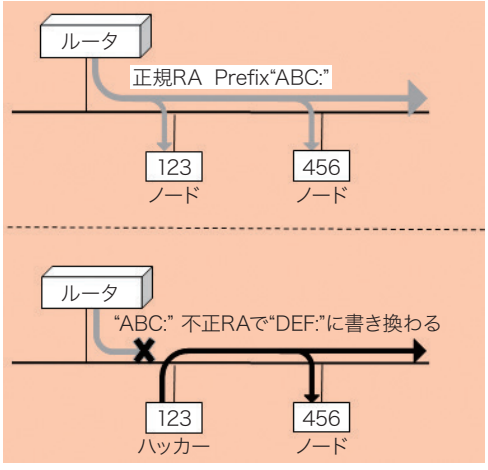

図1 RAによるPrefix書き換え

このRAは，ほとんどのIPv6機器で動作するので高い確率でノードを列挙できる。しかし，この方法はIPv6ネットワークのハッキング手法の1つでもあるので，不正な設定を強制的に行ってその反応を見るという本手法は，通信障害の原因になるなどリスクも多く実践には利用できない。

3.3 パラメータ不正のICMPv6パケットを送信する方法

この手法は，パラメータに異常があるパケットを送り付け，ノードがエラー応答することを期待する手法である。IPv6に関連する標準勧告文書のRFC1885の3.4項には，「パケットを処理するIPv6ノードが，完全にそのパケットを処理することが不可能であるIPv6ヘッダもしくは拡張ヘッダに含まれる問題のあるフィールドを見つけたのであれば，そのパケットは破棄しなければならず(MUST)，パケットの発信元にタイプと問題個所を示したICMPv6パラメータ不正メッセージを送信すべきである(SHOULD)。」[(1)]と定義されている。この定義に準じていれば，IPv6のヘッダに矛盾があるパケットを，オールノードマルチキャスト宛に送信したとき，パラメータ不正を通知するICMPv6が返信されてくることが期待できる。このエラー応答により，IPv6ノードを列挙できる可能性がある。実験の結果，カーネルレベルの応答においては，ほとんどのノードがパラメータ不正のICMPv6パケットに応答することが分かった。しかし，現在のOSは，セキュリティ対策としてファイアウォールを実装しているものが多い。このファイアウォールの機能によりパラメータ不正のパケットはカーネルに届くことなく破棄されている。つまり，不正パケットはカーネルに到達しないことがありノード検出の手法としては利用できない。

3.4 MLD(Multicast Listener Discovery)を使った方法

IPv6ネットワークは，マルチキャストを多用するプロトコルである。ルータは自分の配下にあるノードが，どのマルチキャストグループのパケットを必要としているのかを管理することで，セグメントを超えたマルチキャストを実現している。このルータによるマルチキャストノードの管理に使われるのがMLDである。本手法では，ルータが配下のノードに対してマルチキャストグループを問い合わせるMLQ(Multicast Listener Query)を使った。MLQの宛先をオールノードマルチキャストにすることで，配下のノードは，自身が参加しているマルチキャストグループ情報を記したMLR(Multicast Listener Report)をマルチキャスト宛に返信してくる。この応答を収集することで，マルチキャストを受信しようとするノードの列挙ができる。このマルチキャストの管理は，IPv6の基盤となる仕組みのため，ほぼすべてのIPv6ノードが検知可能である。

我々は，これらの4つの手法を実験した結果，システム障害の可能性がなく，セキュリティシステムの影響を受けずにIPv6ノードが検知できるMLDを採択することにした。

4. IPv6 でのプロミスキャスモードの検出

前述の手法でIPv6ノードを列挙できれば，その中でネットワーク盗聴しているものを検出できる可能性がある。ネッ

トワーク盗聴はNIC（Network Interface Card）に対して，宛先にかかわらずすべてのパケットを受信することができるプロミスキャスモードという特別なモードに設定し，行われている。そのため，NICがプロミスキャスモードになっているノードの検出とネットワーク盗聴を行っているノードの検出とはほぼ同等と考えられる。まず，IPv6の検出手法の前に，既に公表しているIPv4におけるプロミスキャスモードの検出手法を解説する。

ネットワークを盗聴するとき，盗聴者はパケットを一切送信することなくこれを実行できる。このため，ネットワーク盗聴の発見は難しい。盗聴を発見するためには，盗聴しているPCが，意図せず自動的に応答してしまう何らかのパケットを送付しなければならない。通常のPCは自分が受け取るべきパケットだけを受信し，他人宛など不要なパケットをすべて捨てている。それに対して，ネットワーク盗聴を行っているPCの場合，すなわちプロミスキャスモードで動作しているPCでは，NICのフィルタリングを無効にして，すべてのパケットを受け取っている。つまり，プロミスキャスモードのPCと通常のモードのPCでは，カーネルに渡されるパケットの種類に差がある。プロミスキャスモードの場合だけカーネルに渡されるパケットが存在している。このプロミスキャスモードの時だけカーネルに渡されるパケットを活用する。

例えば，アドレス解決のような基本的なリクエストを，宛先MAC（Media Access Control）アドレスに送付したとき，プロミスキャスモードのPCだけが，自動で応答すると期待できる。しかしながら，宛先MACアドレスを他のノードアドレスに書き換えただけではプロミスキャスモードのPCが応答することはなかった。これは，カーネルのソフトウェア部分でもアドレスのフィルタリングを行っているためである。しかし，我々はこのソフトウェア上のフィルタとNICのフィルタには精度の差があることを発見した。NICのフィルタでは，厳密にMACアドレス6byteすべてを検証しているのに対して，カーネル上のソフトウェアフィルタでは，先頭の2byteや重要な1bitしか検証しない場合が多い。このフィルタ精度の差つまり，「ソフトウェアフィルタは通過するが，NICのフィルタは通過しないMACアドレス」が存在し，それを利用することでプロミスキャスモードの検知が可能になるのである。IPv4においては，ARPパケットの宛先MACアドレスを偽ブロードキャストアドレスに変形することで，プロミスキャスモードの検出に成功した。[2]

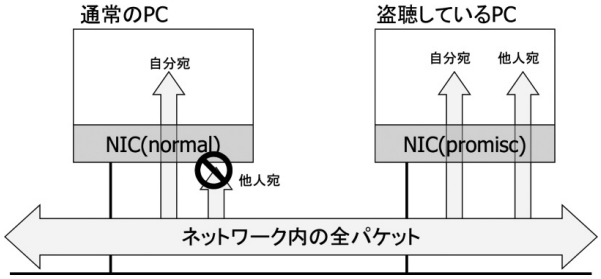

図2　ネットワーク盗聴とNICのフィルタリング

では，本題のIPv6におけるプロミスキャスモードの検知について考える。

IPv6のプロミスキャス検出においても検知方法の要は，宛先MACアドレスの選択にある。IPv6にはブロードキャストアドレスが存在しないため，マルチキャストアドレスを基にMACアドレスを変形する方法を検討した。IPv6には，全ノードリンクローカルマルチキャストという，アドレスグループが存在する。これは，基本的にすべてのノードが受信するアドレスグループであり，実質的にIPv4のブロードキャストアドレスと同等の役割を担っている。そしてこのIPv6のマルチキャストアドレスに対応するマルチキャスト用のMACアドレスが存在していることからこのマルチキャスト用のMACアドレスを変形して利用する。次に，使用する上位プロトコルの選択である。IPv4ではARPを使用したが，IPv6にはARPが存在しない。Pingでの代替を検討したが，Windows10においてPingの応答を得られないことが判明し，別のプロトコルを検討した。

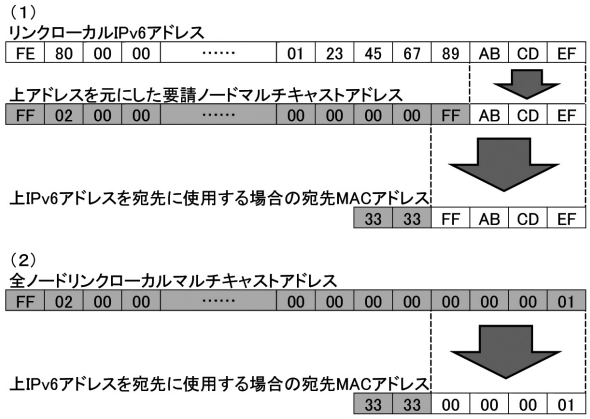

図3　IPv6マルチキャストアドレスとMACアドレス

代替のプロトコルに求められる条件は下記の2つである。

(1)すべてのノードが必ず応答するプロトコル

(2)1つのノードに対して，2つ以上のパケットを必要としない。1つのパケットで応答が得られる。

この条件を基に，様々なプロトコルを比較した結果，ICMPv6の中から，RA（Router Advertisement），MLQ（Multicast Listener Query），NS（Neighbor Solicitation）の3つが有力な候補として挙がった。

4.1　RA(Router Advertisement)

前述のようにRAはIPv6アドレスの自動設定に使用されるプロトコルである。ルータが下位セグメントに存在するノードへ，Prefix情報を通知するために使用される。通知されたPrefix情報を基に作成されたアドレスをノードは使用する。そして，ノードが新たなアドレスを使用する場合，重複確認のためにNSパケットを送信する。この仕組みを利用する。まず，検知する側が作成した偽のPrefix情報でRA通知を送信する。この時，宛先MACアドレスを本来受信しないはずのマルチキャストグループのMACアドレスに変形しておく。通常のノードはこのパケットを受け取らないため，それ以降何も起こることはない。対照的に，プロミスキャスモードのノードはこれを受信し，指定のPrefixに基づ

いたアドレスの使用手続きとして重複確認のNSパケットを送信してしまう。検知する側はNSパケットを監視する。偽のPrefix情報に基づいたIPv6アドレスが使用されようとした場合，それを送信したノードはプロミスキャスモードであるといえる。しかし，この手法では不正なPrefixを使用するので，ネットワークが破壊される恐れがある。これは，ハッカーの攻撃手法と同じであり，RAを使用する方法は好ましくない。

4.2 MLQ(Multicast Listener Query)

MLQは，正確にはMLD（Multicast Listener Discovery）メッセージで使用されるタイプの1つであり，対象ノードの参加マルチキャストグループを通知させるために使用される。MLQを受け取ったノードは，ネットワークに自身が参加しているマルチキャストグループをMLR（Multicast Listener Report）で通知する。これを利用し，宛先MACアドレスを変形したMLQを送信し，ネットワーク上のMLRを監視することでプロミスキャスモードの検出を行う。この方法はプロトコルの正常な動きのため，ネットワークに支障をきたす心配はない。しかし，MLRはマルチキャスト宛に返信される。そしてどのMLQに対応するMLRなのか識別ができない。別のMLQへの応答を受信して誤検知してしまう可能性がある。MLQはOSが自動で使用する頻度も高いため，この方法も断念せざるを得ない。

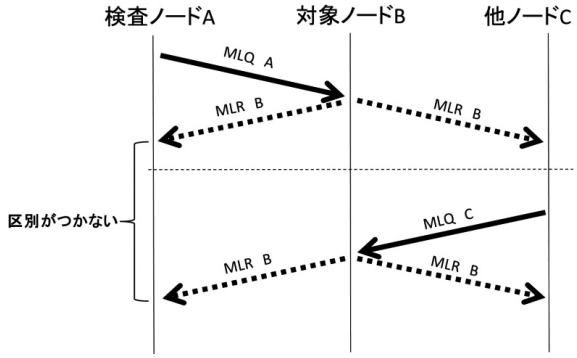

図4　MLRのマルチキャスト応答の識別

4.3 NS(Neighbor Solicitation)

4.2 MLQでの課題解決のため，NSを利用することを検討した。

NSはIPv6からのMACアドレスの解決および，アドレスの重複確認に利用される。NSパケットを受け取ったノードは，自身のIPv6アドレスと照合し，一致すればNICに割り当てられているMACアドレスをNA（Neighbor Advertisement）で通知する。このNSとNAの仕組みを利用する。宛先MACアドレスを変形したNSパケットで，既に列挙されているIPv6アドレスの重複確認を行う。盗聴状態にないノードは，この変形したNSパケットを受信しない。しかし，盗聴状態にあるノードはこれに応答する。つまり，ネットワーク上のNAによるレスポンスを監視することでプロミスキャスモードの検出ができる。このメッセージは上記2つと異なり，対象ノードのIPv6アドレスが分かっていなければパケットを作成することができない。しかし，先に記したノード発見のアプローチによって，事前にノードをリストアップしておくことが可能である。また，応答であるNAはNSの送信元に対してユニキャストで返信されるので，これを受信できれば盗聴を疑われるノードの検出が可能である。そしてこの方法は，ネットワークへの悪影響の心配がない。ただし，ユニキャスト送信の前にアドレス解決の処理を正しく行う必要がある。

5. 今後の課題

我々は，MLDを利用することで未知のIPv6ノードを列挙する手法を見出した。また，宛先MACアドレスを変形したNSパケットを使うことで，IPv6のプロミスキャス検出を実現できることを確認した。

今後，5Gネットワークが本格的に普及すると，制御システムにおいてもWi-Fiなどの無線ネットワークが標準的に利用されるようになるだろう。無線ネットワークの盗聴の手法は，有線LANのものとは違いその発見についてはまだ解決できていない。それ以外にも新しいネットワークのセキュリティにおいては新たな課題が次々と顕在化してくるであろう。我々はこれらのいち早い解決を目指して研究開発を進めていく。

＜参考文献＞
(1) RFC1885 3.4 Parameter Problem Message
(2) 日経ネットワークセキュリティ 2002 Vol.2
「社内の盗聴者を見つけ出す」，pp.116-125

＜著者所属＞
佐々木　太一　アズビル株式会社
IT開発本部開発1部
太田　貴彦　アズビル株式会社
ビルシステムカンパニー開発本部開発2部
佐内　大司　アズビル株式会社
人事部付