ARPを利用したローカルエリアネットワークにおける不正接続の排除

松谷健史
慶應義塾大学 環境情報学部

内部情報漏洩を防ぐためにローカルエリアネットワークに接続できるＰＣを制限し、不正なネットワーク接続を排除する手段の提案。ＡＲＰを用いることにより既存のネットワークレイアウトを変更せずに実現する。

The exclusion of malicious access within LAN using ARP

Takeshi MATSUYA
Keio University, Faculty of Environmental Information

I suggest how to exclude malicious access within Local Area Network in order to limit PCs which can access LAN.The above can be accomplished without changing the network systems by using ARP.

１．はじめに

近年のネットワーク通信技術の普及により、様々なコンピュータが容易にネットワークを利用できる環境が整えられている。

その一方で、不正なコンピュータでもネットワーク機器に接続さえすれば通信できる。

社会的背景としても、不特定多数の人によるネットワーク利用を妨げたいという要望が高まっている。例えば大手インターネットプロバイダーの顧客情報流出事件以降、個人情報流出を防ぐの為のネットワーク管理が注目されるようになり、また2005年4月1日から施工される個人情報保護法[※1]により一定以上の顧客情報を持つ事業者は、個人情報漏洩対策を含む情報管理が責務となる為、セキュリティ管理がより重要視されてきている。

本論文は、企業内部ネットワークにおいてコンピュータ接続をARPを用いて制限するセキュリティ手段について考察、実験したものである。

このような目的の既存技術としては、VLAN認証やIEEE802.1xがあるが、ネットワーク上のHUBや無線LANアクセスポイントをこれらの技術に対応させたものに変えなければならない。

本論文では既存のHUBや無線LANアクセスポイントを活用した上で、不正なネットワークアクセスを排除する方法について２章で提案し、３章でこれを解決する本手法の実装、４章で実験を行い、５章でまとめとする。

２．不正アクセス排除の手法

2.1 TCP/IP通信の基本

TCP/IPを用いた通信では、通信の相手先を指定するためにＩＰアドレスを用いる。しかし、実際のローカルエリア内での通信においては個々のLANカードに埋め込まれているMACアドレスを併用しいている。

図1に、例として192.168.1.100のIPアドレスを持つPCが192.168.1.1のIPアドレスを持つサーバーと最初の通信を開始するまでの過程を示す。

(1) PCが192.168.1.1を持つノードのMACアドレスを調べる為に ARP Requestのブロードキャストパケット（一斉同報）をネットワーク全体に送信する。
(2) 該当するIPアドレスを持つサーバーが、自分のMACアドレスを告知するARP Reply(ユニキャスト)をPCに対して返信する。PCはARPキャッシュに192.168.1.1のIPアドレスがAA:AA:AA:01:A0:01であることを学習する。
(3) PCよりIPデータパケットを送信する (送信先はMACアドレスAA:AA:AA:01:A0:01 IPアドレス192.168.1.1)
(4) サーバーよりIPデータパケットが送信される。

[※1] http://www.soumu.go.jp/gyoukan/kanri/031219_1.html

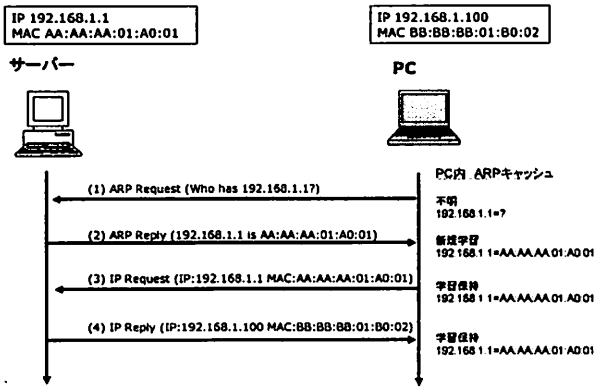

図１：TCP/IP通信例

2.2 通信の排除（妨害）

2.1では、IPアドレスからMACアドレスを調べて通信を開始するまでの過程を確認した。ここで重要な点としては下記の２点が挙げられる。

(1) 相手のMACアドレスがわからないと通信が開始できない。

(2) IPアドレスとMACアドレスの対応表(ARPキャッシュ)の作成に、ARP Replyパケット内のMAC,とIP情報が用いられている。

以上の点からネットワーク上に存在しないMACアドレス埋め込み偽装したARP Replyパケットをターゲットに送信する事により、正常な通信をできなくする事ができると推測される。

2.3 不正アクセスの発見

次に不正アクセスを試みようとするPCの発見方法について考える。

同一ネットワーク内において、ブロードキャスト(一斉同報)パケットはどこの場所にいても受信することができる。

また、どのようなPCでもネットワーク通信を開始する為には相手のMACアドレスを知る必要がある。その動的解決手法としてはARP Requestを使ったブロードキャストパケットを用いるのが一般的である。

以上の事から、同一ネットワーク上に流れるARP Requestのパケットを監視する事で、これから通信を試みようとするPCの存在とそのMACアドレスを知ることができる。

この時、MACアドレスから不正なPCであるかどうかを知る手段としては、今回は事前に用意しておいたMACアドレスによるアクセス許可リストと照合するものとする。その為、ネットワーク内に存在する正当なネットワークノードすべて(PC、ファイル・プリントサーバー、ルーター、VoIP G/W等)において予めMACアドレスを調べて、アクセス許可リストに登録しておく必要がある。

厳密な管理が必要でなければ、平常時においてネットワーク上に流れるARP Requestパケットを監視して送信者のMACアドレスをアクセス許可リストに登録していく事も考えられる。この場合、全てのノードからARP Requestを送信させる為に、電源を入れなおすか、又はメモリーに学習されたARPキャッシュが破棄されるまでの一定時間監視しておくよう注意が必要である。

2.4 不正アクセス排除

不正アクセスを検知して排除する機能を持つPCを以後、保護PCと呼ぶ。実際に不正アクセスを検知して、通信を妨害するのには下記の仕組みを提案する。(図2)

(1) 不正PCが192.168.1.1を持つノードのMACアドレスを調べるARP Requestを送信。このブロードキャストパケットは、正規のサーバーとともに保護PCにも受信される。

(2) 192.168.1.1のアドレスを持つサーバーが、不正PCに自分のMACアドレスをARP Replyパケットにて返信する。この時点で不正PCのARPキャッシュに正規のMACアドレスが学習される。

(3) (1)により保護PCが、許可されていないMACアドレスを持つ不正PCのアクセスを検知して、ネットワーク上に存在しないMACアドレスを情報として持つ偽装ARP Replyを不正PCに送信する。このパケットを受け取る事により不正PCのARPキャッシュにMACアドレスが学習される。

(4) 不正PCがARPキャッシュ内にある偽装されたMACアドレスを用いて、IPパケットのユニキャスト送信を行う。しかし、これは実存しないMACアドレスの為、IPデータパケットがサーバーまで到達する事ができない。

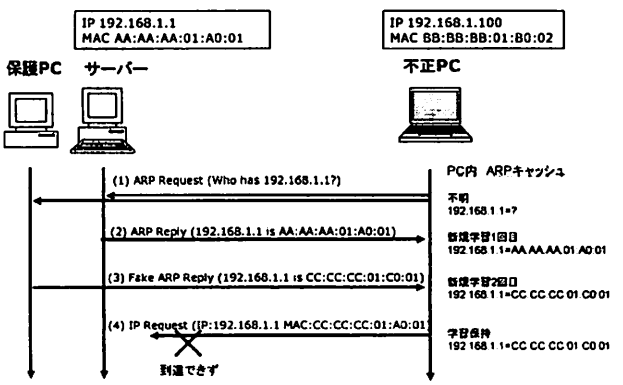

図２：不正アクセス排除の仕組み

3．実装

保護PC上で動作するプログラムは、表１のプラットフォームにて実装する。動作は以下のように行った。

(1) ネットワーク上に流れるブロードキャストパケットを取得できるようにする為、LANインターフェースをプロミスカスモード(Promiscuous Mode)に設定する。
(2) ARP Requestパケットのみを監視の対象として取得する。
(3) ARP Requestパケットの送信元MACアドレスが、許可リストに登録されているMACアドレス中にあれば、何もせずに引き続き次のARP Requestパケットを監視する。
(4) 偽装用のARP Replyパケットを生成する。この時、偽装するMACアドレスのベンダーコードにはネットワーク内で存在しないプライベートMACアドレスである00-01-01を今回は割り当てる。
(5) 生成したパケットをネットワーク上に一度だけ送信する。

OS	Linux
開発ツール	GCC

表１：実装環境

4．実験

4.1 実験の目的

2.4で不正アクセス排除の方法に関して述べた。しかし、ここでいくつかの疑問点が生じる。

第一に、一回のARP Requestに対して続けて二回のARP Replyが返信された時、ARPキャッシュが更新されるかという点。

第二に、偽装されたARPキャッシュ情報の保持に関する点。ARPキャッシュは一般的に数分間程度は保持されるが、全く通信できないMACアドレスに対して作成されたARPキャッシュがどの程度の時間保持されるかという点である。

以上の点は、OSに依存するところである。実際に保護用PCを用意し、いくつかの代表的OSを用いて実験をした。

4.2 実験環境

実験環境は３個のSwitching HUBをカスケード接続し、それぞれのHUBに不正PC(不正なアクセスを試みる)、サーバー、保護PC(不正アクセスを排除する)を接続するように用意し、不正PCのOSは毎回変えられるように準備した。(図３)

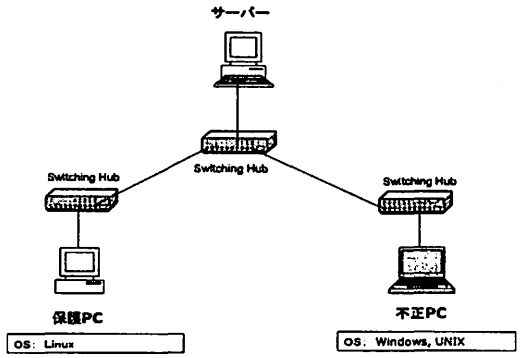

図３：構成図

4.3 不正通信排除の実験

不正PCよりサーバーに対して行われた通信が保護PCによって排除されるかを確認する。手順は下記の通り。

(1) 通信を開始する前に不正PC上からサーバーのARPキャッシュを記憶していない状態とする。(不正PCを電源投入直後とするか、arp -dコマンドによってARPキャッシュを削除する)
(2) 不正PCからサーバーに対してpingコマンドを行う(図４)。この時、偽装パケット受信による無応答時間が計測できるようにSTOP指示を与えるまでコマンドを終了しないように設定する。
(3) (2)の結果から、通信が排除されたかどうか、および、１つの偽装パケットでどの程度の時間、排除が有効であるかを不正PCのOSを変えながら調査する。(表２)

```
ping -t -w 1000 10.0.20.11

Pinging 10.0.20.11 with 32 bytes of data:

Reply from 10.0.20.11: bytes=32 time=16ms TTL=255   —— 正規のARPキャッシュが有効
Request timed out.                                  —— 偽装ARPキャッシュが有効
Request timed out.
Request timed out.
Request timed out.
    .
    .
    .
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Reply from 10.0.20.11: bytes=32 time=1ms TTL=255    —— 偽装ARPキャッシュが破棄され、
Reply from 10.0.20.11: bytes=32 time<1ms TTL=255    — 新たなARP Requestが再発行された
Reply from 10.0.20.11: bytes=32 time<1ms TTL=255
Reply from 10.0.20.11: bytes=32 time<1ms TTL=255
```

図４：不正PCよりサーバーへの通信と排除

Operating System	通信排除の有効性	偽装ARPキャッシュの保持時間
Windows Millenium Edition	あり	8:29 (509秒)
Windows XP Professional SP2 v.2149 (RC2)	あり	4:02 (242秒)
RedHat Linux 9 Kernel 2.4.20-8	あり	1:01 (61秒)
Solaris 9 (Ultra SPARC) (SunOS5.9 Generic_117171-05)	あり	26:52 (1612秒)
NetBSD 1.6.2	あり	13:12 (792秒)
FreeBSD 5.2.1	あり	19:45 (1185秒)

表２：OSによる排除が有効な時間の違い

4.4 不正通信排除の実験結果考察

4.3の実験および図４、表２の実験結果より、偽装ARP Replyによる通信の排除は主要OSに関して大変有効である事がわかった。

図４の結果にあるように、一番最初のpingによる通信が正しくできたのは正規のサーバーよりのARP Replyを最初に受け取ったからである。

保護PCより偽装ARP Replyを受けた以降は、数分以上の間、保護PCとサーバー間が通信不能に陥る事を確認できた。これは動的に記憶された偽装ARPキャッシュが一定期間を経過すると消去される為であり、この保持時間は表２よりそれぞれのOSのARPプロトコルの実装により大きく異なるといえる。

このことは、不正PCを排除し続ける為には偽装ARPキャッシュの保持が失われる前に定期的に偽装ARP Replyを送信つづける必要があることをあらわす。

4.5 ARPキャッシュ強制追加の実験

4.4の実験結果考察により通信中の他のネットワークノードのARPキャッシュ改竄が容易である事がわかった。

次にこの事から、ARPキャッシュに記録されておらず未通信のノードに関するARPキャッシュ情報でさえも自由に新規追加出来るのではないかと推測した。この事を検証する為に下記の手順の実験を行った。

(1) 不正PC上からARPキャッシュが記録されていない事を確認する。(arp -aコマンド)
(2) 保護PCから不正PCに対して、存在しない同一ネットワークIPアドレスおよびMACアドレスを持つ偽装ARP Replyパケットを送信する。
(3) 不正PC上において、(2)で送信したARPキャッシュが新規追加されているかを確認する。
(4) 以上の事を、不正PCのOSを変えながら調査する。また、不正PCの変わりに一部ルーターに置き換えた。（表３）

Operating System 又は Router	ネットワークからの強制ARPキャッシュ追加
Windows Millenium Edition	有効
Windows XP Professional SP2 v.2149 (RC2)	有効
RedHat Linux 9 Kernel 2.4.20-8	有効
Solaris 9 (Ultra SPARC) (SunOS5.9 Generic_117171-05)	有効
NetBSD 1.6.2	有効
FreeBSD 5.2.1	有効
YAMAHA RTX1000 Rev.8.01.15	有効
CISCO 1605 IOS 12.2	有効

表３：OSによる強制ARP追加の有効性

4.6 ARPキャッシュ強制追加の実験結果

4.5の実験および表３の実験結果より、ARP ReplyによるARPキャッシュの強制追加は容易である事が判明した。この事から、ここで挙げたOSのARP実装においてはタイミングや送信相手関わらずARP Replyパケット情報は常に正しいものとして受け入れられ、ARPキャッシュへ記憶するものと考えられる。

５．まとめ

今回の実験を通して、ローカルエリアネットワーク上の同一セグメント内であれば不正ネットワーク接続を排除できることが確認できた。

不正PCの発見にはARP Requestのブロードキャストパケットを監視することが有効であり、不正PCの接続排除には偽装したARP Replyパケットを送信する事が極めて有効であった。

一般的に通信先のIPアドレスからMACアドレス情報を得る為にはARP Requestを送信し、相手からのARP Replyパケットより ARP情報を得ると考えられているがARP Request要求を行っていないのにも関わらず第三者からのARP Replyパケットをも受け取ってしまう。

その為、偽装したARP Replyパケットを送信する事により存在しないMAC情報を含むARPキャッシュを記憶させ、特定の相手と通信が出来ない状態にする事ができる。

実際の運用においてはいくつかの検討が考えられる。

(1) ブロードキャストが到達できない複数のネット

ワークセグメントが存在するケースへの適応。
(2) パケットロスによってブロードキャストが取得できないケース。
(3) 接続を一度排除したPCを再度、許可して通信ができるようにする手段の必要性。
(4) ブロードキャストネットワークアドレス宛へのARP Requestが行われた時の対処。

(1)は、保護PCにいくつかのLANアダプターカードを備え、それぞれのネットワークに接続する事で対処が可能となる。
(2)は、ブロードキャストパケットがサーバーにのみ到達して、保護PCには到達できないという場合に問題となる。今回の実験中ではこのような状況は確認できなかったが、理論上では想定できる事でもある。このようなケースには対応できない。。
(3)は、一度不正PCと認識したものを正規のPCとして許可しなおす処理を保護PC上で必要とする。この場合、不正とみなされたPCには偽装ARPパケットが記憶されているので、過去送信した偽装ARPに対応した正規のARPキャッシュを再送しなおすなどの実装が考えられる。
(4)は、例えば不正PC上でIPネットワークのブロードキャストアドレスへpingを発行する等(例: ping 192.168.1.255)。これは同一ネットワークに接続している機器の存在をまとめて調べる場合に利用される(実行後、arp -aにて一覧確認する)。ブロードキャストパケットを受け取ったノードは一斉に不正PCに対してARP Replyを返し、ARPキャッシュが正しい状態で埋め尽くされてしまい、さらにこれらの返答をしたノードのIPアドレスも判明できない。その為、今回の手法では対処できない。

また、より強固な排除をする為に不正PCを発見した場合、偽装ARP Replyを不正PCにのみ送信するのではなく、同時に重要なサーバーや不正PCが通信しようとした相手に送信し、不正PCとの通信を排除する手法も有効であると考えられる。

今回の実験は、個人情報流出保護対策の一環として不正接続の排除方法に関して提案・実装・実験したものである。

しかしながら、その過程においてARPの実装にセキュリティ上の問題を確認した。ARPキャッシュを自由に書き加え出来る現状において、ローカルエリアネットワーク内の通信においては、いつでも好きなノードを好きな場所へ強制的に通信させるように仕向けたり、または、排除できることをあらわしている。

この事から現状のARPを主体とした通信を行うローカルエリアネットワークは危険を抱えているともいえる。

参考文献

[1] 笠野英松,マルチメディア通信研究会: "インターネットRFC辞典", アスキー出版局
[2] IETF URL: http://www.ietf.org/
[3] CISCO社 URL: http://www.cisco.com/
[4] YAMAHA社 URL: http://www.rtpro.yamaha.co.jp/