ARP Poison Routing

Anurag Chugh and Ranjit Puri

Abstract—The development of the internet has threatened the privacy of any individual connected to it. ARP Poison Routing technique has only magnified this threat by providing a way to overcome the protection offered by switches.

The real utility of ARP Poison Routing lies in the fact that it can be used to perform packet sniffing over a switched network. Packet Sniffing enables a spy to eavesdrop on the data flowing on a network segment. By installing a sniffer program on his computer, he can sniff data flowing on his segment of the network. If data flowing on a particular segment of a remote network needs to be sniffed, the spy can do so by installing a trojan sniffer program on a machine on that segment and administer it remotely.

Network administrators used switches to deal with this problem and they were quite successful.

ARP Poison Routing has been instrumental in providing spies with a way to fool these switches and get access to sensitive data flowing over a network. ARP Poison Routing utilizes an inherent bug in the Address Resolution Protocol definition (RFC 826) to manipulate the ARP tables of remote computers. By doing so the remote computer can be fooled to send data packets to a destination other than the one the packets are intended for. Usually the packets are rerouted to the destination machine by the spy computer which copies it before retransmitting to the intended host. In this way the spy achieves his goal and at the same time avoids detection by re-manipulating the ARP table of the target computer again and again after certain intervals of time.

Incase the number of target computers are many, the spy may use ARP flooding to cause a switch to go into Hub mode thereby allowing the spy to sniff information flowing between not one but all pairs of hosts connected to that switch which are exchanging information.

This paper describes this technique and the how it can be neutralized.

Index Terms—Address Resolution Protocol, ARP Spoofing, MAC Flooding, RFC 826, Reverse Engineering, Packet Sniffing, Hacking.

ARP Poison Routing

Anurag Chugh and Ranjit Puri

***Abstract*—The development of the internet has threatened the privacy of any individual connected to it. ARP Poison Routing technique has only magnified this threat by providing a way to overcome the protection offered by switches. This paper describes this technique and the how it can be neutralized.**

***Index Terms*—Address Resolution Protocol, ARP Spoofing, MAC Flooding, RFC 826, Reverse Engineering, Packet Sniffing, Hacking.**

I. Introduction

PACKET SNIFFING enables an spy to eavesdrop on the data flowing on a network segment. By installing a sniffer program on his computer, he can sniff data flowing on his segment of the network. If data flowing on a particular segment of a remote network needs to be sniffed, the spy can do so by installing a trojan sniffer program on a machine on that segment and administer it remotely.

Network administrators used switches to deal with this problem and they were quite successful.

ARP Poison Routing has been instrumental in providing spies with a way to fool these switches and get access to sensitive data flowing over a network. ARP Poison Routing utilizes an inherent bug in the Address Resolution Protocol definition (RFC 826) to manipulate the ARP tables of remote computers. By doing so the remote computer can be fooled to send data packets to a destination other than the one the packets are intended for. Usually the packets are rerouted to the destination machine by the spy computer which copies it before retransmitting to the intended host. In this way the spy achieves his goal and at the same time avoids detection by re-manipulating the ARP table of the target computer again and again after certain intervals of time.

Incase the number of target computers are many, the spy may use ARP flooding to cause a switch to go into Hub mode thereby allowing the spy to sniff information flowing between not one but all pairs of hosts connected to that switch which are exchanging information.

In this paper we describe the technique, the underlying principles, implementation and prevention of ARP Poison Routing.

II. The Ethernet

Ethernet has been a relatively inexpensive, reasonably fast, and very popular LAN technology for several decades. Ethernet was developed at Xerox PARC in the beginning of 1972 and its specifications appeared in IEEE 802.3 in 1980. Ethernet specifications define low-level data transmission protocols and the technology needed to support them. In the ISO OSI model, Ethernet technology exists at the physical and data link layers (layers 1 and 2).

A low-level network technology, Ethernet supports IP and most other higher-level protocols. Traditional Ethernet supports data transfers at the rate of 10 Megabits per second (Mbps). Over time, as the performance needs of LANs have increased, related technologies like Fast Ethernet and Gigabit Ethernet have been developed that extend traditional Ethernet to 100 Mbps and 1000 Mbps speeds, respectively.

Today, networks based on this specification have been deployed in very large numbers and at very large scales. The interconnection of these networks on an international level has given rise to the Internet. The Internet has made possible for any computer connected to it to exchange data with any other computer connected to regardless of their physical separation on earth.

For a data packet to be sent from a particular computer to another computer on the Internet (or local Intranet), it needs to be routed by special devices (or computers). The purpose of these devices is to know where the destination computer for a data packet can be found and to forward that data packet on the right way so it can reach its destination.

III. Creating and Interconnecting Networks

As said above, special devices are used to connect two or more networks or segments together and route data from one network to the others. These devices have many ports to which computers (or other devices like themselves) are connected to form a network (or segment). These devices then facilitate the flow of data from one computer connected to their port to any other computer or device connected to any other of their ports. These devices are described as follows:

A. Hubs

Hubs classify as layer 1 devices in the ISO OSI model.

Hubs do not read any of the data passing through them and are not aware of their source or destination. Essentially, a hub simply receives incoming packets, possibly amplifies the electrical signal, and broadcasts these packets out to all devices connected to it - including the one that originally sent the packet.

There are three kinds of hubs:

Passive hubs or concentrator: These do not amplify the electrical signal of incoming packets before broadcasting them out to the network.

Active hubs or multiport repeater: These perform amplification before broadcasting packets out to the network.

Intelligent hubs: These add extra features to an active hub that are of particular importance to businesses. An intelligent hub typically is stackable (built in such a way that multiple units can be placed one on top of the other to conserve space). It also typically includes remote management capabilities via SNMP and virtual LAN (VLAN) support.

Hubs remain a very popular device for small networks because of their low cost. All computers connected to the same hub (or stack of hubs) from a single segment of the network.

B. Switches

Although there are layer 3 to layer 7 switches on the market, most of the Ethernet switches today are OSI layer 2 devices. When a data packet comes into a switch, it will analyze the data and direct it to the specific destination only. Thus, a switch can effectively control the traffic flow of the network and ensure independence of each individual port. Layer 2 switches use network card's MAC (media access control) address to identify the destination of a data packet. When a switch is turn on, it will search through the entire network and memorize each station's MAC address on the filtering address table. All computers connected by a layer 2 switch belong to the same subnet. Thus, they share the same first 3 numbers in the IP address. These devices are a bit expensive as compared to hubs but unlike hubs, these allow for the control of network traffic.

C. Routers

A router is generally an OSI layer 3 device. Routers use IP address instead of MAC address for the destination of a data packet. Therefore, it is hardware independent and gives network designer more flexibility. A router can divide network into different subnets, thus making Internet sharing possible. Nevertheless, because routers use software method to control the network traffic, they are generally slower than switches. Therefore, the use of routers for most business is limited to Internet connection sharing through NAT routers. In a wide area network (WAN), routers are necessary for connections between different networking standards such as the connection between Ethernet and ATM. These devices are the most expensive among the three network devices discussed so far.

IV. Address Resolution Protocol (ARP)

The world is a jungle in general, and the networking game contributes many animals. At nearly every layer of network architecture there are several potential protocols that could be used. The Ethernet allows all of these protocols to coexist on a single cable by means of a type field in the Ethernet packet header. However, the Ethernet requires 48 bit addresses on the physical cable, yet most protocol addresses are not 48 bits long, nor do they necessarily have any relationship to the 48 bit Ethernet address of the hardware. For example, CHAOS addresses are 16 bits and IP addresses are 32.bits. A protocol was needed to dynamically distribute the correspondences between a <protocol, address> pair and a 48 bit Ethernet address (also known as Media Access Control Address).

The development of the Address Resolution Protocol (RFC 826) solved the above problem and provided with an efficient way to resolve IP Addresses to MAC addresses.

V. Address Resolution Process

Figure 1 shows how an ARP data packet is encapsulated before it is sent on to the wire. The Address Resolution process takes place according to the following algorithm (incase of IP over Ethernet):

1) A routing protocol determines the next-hop IP address.
2) Is the next hop MAC for the IP in question in the ARP table?
   a) YES: create an 802.x packet using the destination MAC and send.
   b) NO:
      i) Generate Ether packet with:

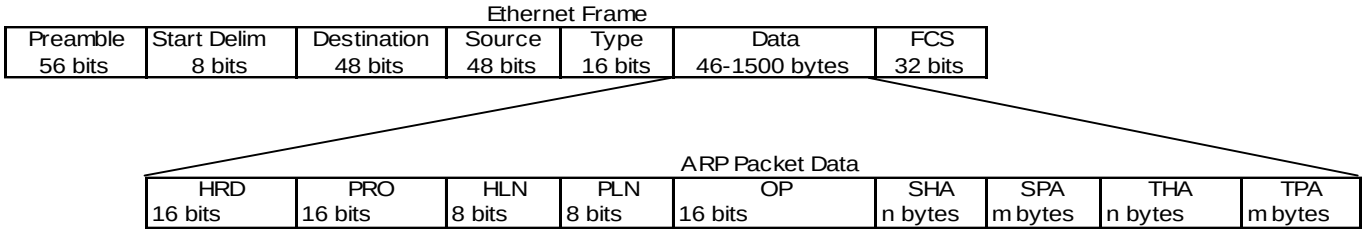

Fig. 1 ARP Packet Data Encapsulated in an Ethernet Frame

Destination = Unset

Source = Local MAC Address

Type = ARP

HRD = Ethernet

PRO = IP

HLN = 6

PLN = 4

OP = ARP-Request

SHA = Local MAC Address

SPA = Local IP Address

THA = ff:ff:ff:ff:ff:ff (Broadcast)

TPA = Remote IP Address

ii) Broadcast ARP packet to the wire

3) Next-hop host will update it's ARP table with the requestor's IP address and MAC, then respond with packet set to:

Destination = Unset

Source = Local MAC Address

Type = ARP

HRD = Local MAC Address

PRO = IP

HLN = 6

PLN = 4

OP = ARP-Reply

SHA = Local MAC Address

SPA = Local IP Address

THA = Requestor's MAC Address

TPA = Requestor's IP Address

4) Any other host will update its IP-ARP table with the requestor's info, and then drop the packet.

Table 1 shows the details of each field in an ARP data packet encapsulated in an Ethernet Frame.

The following point can be observed from the above algorithm:

"If host A needs to modify host B's ARP table only, it can do so by sending an ARP request packet, whose Ethernet Destination field is set to the MAC address of host B. Host B will modify the corresponding entry in its ARP table as per the data contained in the ARP Data Packet (Step 4 of algorithm above)."

This is the basis of ARP Poison Routing.

Ethernet Frame

Preamble:	10101010…
Start Delimiter:	10101011
Destination:	IEEE MAC Address of the receipient if Unicast otherwise set to ff:ff:ff:ff:ff:ff if the packet is to be Broadcast.
Source:	IEEE MAC Address of the sender of the packet.
Type:	Protocol Type (Set to 0x0806 for ARP and 0x8035 for RARP)

ARP Packet Data

HRD	Hardware address space (Set to 0x1 for Ethernet)
PRO	Protocol address space (Set to 0x0800 for IP)
HLN	Hardware address length (Set to 0x6 for Ethernet MAC Addresses)
PLN	Protocol address length (Set to 0x4 for IP Addresses)
OP	Operation Code (See Table Below)
SHA	Requestor Hardware Address (MAC Address in case of Ethernet) (Sender of this Packet)
SPA	Requestor Protocol Address (IP Address in case of IP) (Sender of this Packet)
THA	Remote Hardware Address (set to ff:ff:ff:ff:ff:ff in case of broadcast on Ethernet)
TPA	Remote Protocol Address (Remote IP address in case of IP)

OP Codes

Number	Operation
0x1	ARP-Request
0x2	ARP-Reply
0x3	RARP-Request
0x4	RARP-Reply
0x5	DRARP-Request
0x6	DRARP-Reply

Table 1 Fields in ARP data packet encapsulated in an Ethernet Frame

VI. ARP POISON ROUTING (APR)

Consider a small network in which three hosts (A,B & C) are connected to a switch as shown in Figure 2.

When Host A, B & C send out data to each other on the network, the switch processes the headers of the Ethernet frames which are passing through it from the different ports and forms its Switch Route Table. After the table has been formed and/or its entries updated, the switch scans each incoming Ethernet and forwards it out only to that port to which the host having the same MAC address as the

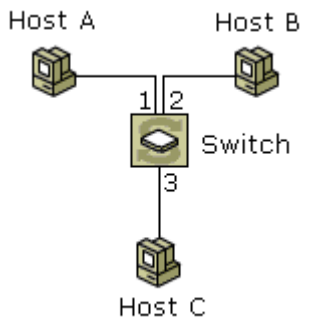

Fig. 2

destination in the Ethernet frame is connected. This helps in controlling the traffic on a network.

If in the incoming frame has destination MAC address as ff:ff:ff:ff:ff:ff , then the switch forwards the packet to all the ports since the frame is a broadcast frame.

Consider a situation in which host A wants to transmit a frame to host B. Let the Switch Route Table of the switch be as shown in Table 2.

Port	MAC Address
1	00:01:aa:bb:cc:01
2	00:01:aa:bb:cc:02
3	00:01:aa:bb:cc:03

Table 2 Switch Route Table

To transmit a packet to host B, host A needs to know the MAC address of host B. To know this, it will transmit an ARP request packet (with TPA set as the IP address of Host B) encapsulated in a broadcast Ethernet frame to the network through port 1 of the switch. When the switch checks the destination of this frame, it forwards the frame to all ports. Host B will then reply with the ARP-Reply packet to host A informing it of the former's MAC address. This will add an entry in Host A ARP Table. Next host A will send the data in Ethernet frames to host B with source address as MAC address of A and destination address as MAC address of B. The switch will check the destination address of the packet and will forward the packet only to port 2.

Now if we replace the switch in Figure 1 with a hub, the data packet sent by host A to host B will forwarded to all three ports by the hub. This means that the Network Interface Controller (NIC) of host C will also receive the packet meant for host B. The NIC of host C will check the destination MAC address of the incoming packet with its own MAC Address and will reject the packet on finding that the two addresses do not match. If the host C has a sniffer program installed on it, the NIC of this host will be in promiscuous mode in which the NIC will perform no checking of the incoming packets and all the incoming packets will be available to the higher network layers on host C's network stack. So in effect, if a hub is used, host C will be able to eavesdrop on the data passing from host A to host B and back. But in case of a switch, this is not possible.

When a switch is installed, the sniffer on host C will only be able to see data coming and going out of its own NIC and any broadcast packets that maybe sent out (like ARP) by host A and B. Usually, these broadcast packets are of no use if host C is a computer used by a spy as these packets do not contain any critical data that host A wants to transmit to host B. Such data packet will not be visible to host C due to the presence of the switch.

One way to overcome this problem is by performing MAC flooding. By sending incorrect ARP replies to a switch at an extremely rapid rate, the switch's port/MAC table will overflow. Results vary by brand, but some switches will revert to broadcast mode (turn temporarily into hubs) at this point. By doing this, host C can sniff all data flowing throughout the network. In case of large networks, this may cause a drop in throughput.

Another more sophisticated way to do this is by poisoning the ARP table of the hosts whose data packets need to be sniffed. This can be illustrated as follows:

On an un-poisoned network, the ARP tables of host A, B & C would be as follows:

Host A (IP: 202.134.111.1 MAC:00:01:aa:bb:cc:01

IP	MAC
202.134.111.2	00:01:aa:bb:cc:02
202.134.111.3	00:01:aa:bb:cc:03

Host B (IP: 202.134.111.2 MAC:00:01:aa:bb:cc:02

IP	MAC
202.134.111.1	00:01:aa:bb:cc:01
202.134.111.3	00:01:aa:bb:cc:03

Host C (IP: 202.134.111.3 MAC:00:01:aa:bb:cc:03

IP	MAC
202.134.111.1	00:01:aa:bb:cc:01
202.134.111.2	00:01:aa:bb:cc:02

Table 3. ARP Table of the hosts A, B &C

The Routing Table of the switch will be same as in Table 2. ARP is a stateless protocol that does not require authentication so a simple ARP Reply packet sent to host A & B by host C will force an update in their ARP tables.

Poison ARP packets must use sniffer source MAC address (MAC address of host C) as source in Ethernet Frame. These ARP packets should contain the following data:

Poison packet for host A:

Destination in Ethernet frame: MAC address of A
Source in Ethernet frame: MAC address of C
OP=ARP-Reply
SHA=MAC address of host C (00:01:aa:bb:cc:03)
SPA=IP address of host B (202.134.111.02)
THA=MAC address of host A (00:01:aa:bb:cc:01)
TPA=IP address of host A (202.134.111.01)

Poison packet for host B:
Destination in Ethernet frame: MAC address of B
Source in Ethernet frame: MAC address of C
OP=ARP-Reply
SHA=MAC address of host C (00:01:aa:bb:cc:03)
SPA=IP address of host A (202.134.111.01)
THA=MAC address of host B (00:01:aa:bb:cc:02)
TPA=IP address of host B (202.134.111.02)

This will update the ARP tables of hosts A and B. After poisoning, the data sent by host A destined for host B (and vice versa) will be sent to host C as they all will contain the MAC address of host C in the destination field of Ethernet frame thereby causing the switch to forward all such packets to host C on port 3. The sniffer software on host C will then be able to capture all data sent by A for B and data sent by B for A. The sniffer program on host C must also perform one more important; it must re-send all incoming packets meant for A coming from B to host A (and vice-versa) to prevent the higher layers of the network stack of hosts A and B from detecting a timeout in the connection between the two hosts A and B. This will cause the flow of data between A and B to stop and thereby terminating connection between them. Host C must also update the ARP tables of hosts A & B by sending out poison packets after regular intervals of time to keep the ARP tables of these hosts updated incase of a flush due to a timeout between A and B. Hence the host A ↔ host B communication will flow through host C thereby bypassing the main feature of the switch.

This process can also be carried out if host A was a gateway router connecting the network to the Internet.

Lets consider a complex case in which APR can be used.

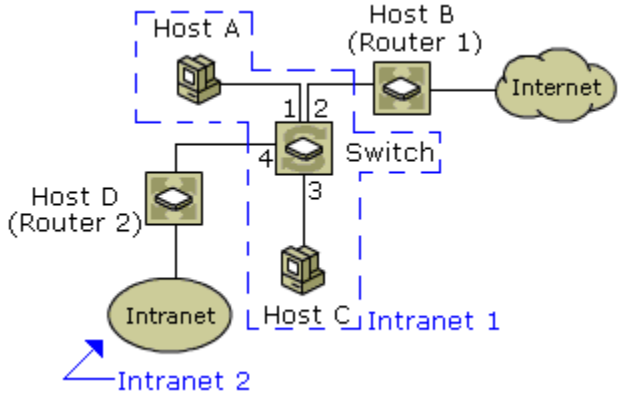

Fig. 3 Complex Network Topology

Consider an organization which has two Intranets physically separated by some geographic distance. Let these be named Intranet 1 and Intranet 2. These Intranets are connected together by a router (host D: router 2). Intranet 1 is connected to Internet by use of another router (host B: router 1). The core of Intranet 1 is the switch which is used to connect the hosts A, B, C and D. All traffic from Intranet 2 must pass through this switch to reach any host on Intranet 1 or the Internet. Let router 2 be default gateway for host A and let Router 1 be default gateway for router 2.

Now consider host C to be under the control of a spy who has installed a sniffer program with APR capabilities on it. The following cases then arise:

Case 1:
If host C poisons the pair <host A, router 2>
Then the traffic that can be intercepted by sniffer on host C is as follows:

- host A ↔ router 2
- host A ↔ hosts on Intranet 2
- host A → hosts on Internet

Case 2:
If host C poisons the pairs <host A, router 2> and <router 1, router 2>
Then the traffic that can be intercepted by sniffer on host C is as follows:

- host A ↔ router 2
- host A ↔ hosts on Intranet 2
- host A → hosts on Internet
- hosts on Intranet 2 ↔ hosts on Internet
- hosts on Intranet 2 ↔ router 1
- router 1 ↔ router 2

Case 3:
If host C poisons the pairs <host A, Router 2>, <Router 1, Router 2> and <host A, Router 1>
Then the traffic that can be intercepted by sniffer on host C is as follows:

- host A ↔ router 2
- host A ↔ hosts on Intranet 2
- host A → hosts on Internet
- hosts on Intranet 2 ↔ hosts on Internet
- hosts on Intranet 2 ↔ router 1
- router 1 ↔ router 2
- host A ← hosts on Internet

In this way APR can be used to intercept traffic on a switched network.

VII. Limitations of APR

1. ARP Poisoning will work only inside your Broadcast Domain so you can't use this technique to redirect traffic between hosts on different subnets or VLANs.
2. The Sniffer must be able to reroute packets to the correct destination or the hosts will not be able to communicate.
3. The Sniffer must know where to route packets so it needs to know IP-MAC mappings of interested hosts before the poison process.
4. APR will slow down network performance because the Sniffer must process packets that it normally doesn't see.

5. ARP Poisoning does not insert a new entry in ARP table, it can only update an existing one. The mapping must already be present in the table in order to manipulate it.

VIII. Implementations

The following tools are available for implementing APR:

1. ARPoison:
   ARPoison is a command-line tool for UNIX which creates spoofed ARP packets. Users can specify the source and destination IP/MAC addresses. http://web.syr.edu/~sabuer/arpoison/
2. Cain & Abel:
   A password retrieval/sniffer utility which implements APR.
   http://www.oxid.it
3. Ettercap:
   Ettercap is a powerful program UNIX program employing a easy to use text-mode GUI. All operations are automated, and the target computers are chosen from a scrollable list of hosts detected on the LAN. Ettercap can perform four methods of sniffing: IP, MAC, ARP, and Public ARP. It also automates the following procedures:

   a. Injecting characters into connections
   b. Sniffing encrypted SSH sessions
   c. Password collection
   d. OS fingerprinting
   e. Connection killing

   http://ettercap.sourceforge.net

4. Parasite:
   Another tool for linux environment.
   http://www.thehackerschoice.com/releases.php

IX. Defenses

The best defence against APR is to enable MAC binding on the switch. This is a feature usually found on high quality switches which does not allow the MAC address associated with a port to change once it is set. Legitimate MAC changes could be performed by the network admin on a per-case basis. Another defence is the use of static routes. ARP tables can have static (non-changing) entries, so poisoned ARP replies would be ignored. This approach is not practical on anything but small home LANs, consequently where APR is not a large concern. Also of note is the behaviour of static routes under Windows. Tests have found that Windows would still accept poisoned ARP replies and use dynamic routes instead of static routes, nullifying any effect of using static routes under Windows.

Aside from these two methods, the only other defence available is detection. Arpwatch is a free UNIX program which listens for ARP replies on a network. It will build a table of IP/MAC associations and store them in a file. When the MAC address associated with an IP changes (referred to as a flip-flop), an email is sent to an administrator. Tests showed that running Parasite on a network caused a flood of flip-flops, leaving the MAC of the attacker present in Arpwatch's emails. Ettercap caused several flip flops, but would be difficult to detect on a DHCP-enabled network where flip flops occur at regular intervals. MAC cloning can be detected by using RARP (Reverse ARP : It is used to find the IP address of a remote machine if its MAC Address is known). RARP requests the IP address of a known MAC address. Sending a RARP request for all MAC addresses on a network could determine if any computer is performing cloning, if multiple replies are received for a single MAC address.

Many methods exist for detecting machines in promiscuous mode. These can be found in the Sniffing FAQ, at http://www.robertgraham.com/pubs/sniffing-faq.html. It is important to remember that operating systems have their own TCP/IP stacks, and Ethernet cards have their own drivers, each with their own quirks. Even different versions of the same operating system have variations in behaviour. Solaris is unique in its treatment of ARP packets. Solaris only accepts ARP updates after a timeout period. To poison the table of a Solaris host, an attacker would have to execute Denial of Service (DoS) on the second target machine in order to avoid a race condition after the timeout period. This DoS may be detected if the network has an Intrusion Detection System in place.

X. Conclusion

APR is one of several vulnerabilities which exist in modern networking protocols, which allow a knowledgeable individual free reign over a network. Appropriate defences must be deployed by an network administrator to prevent a spy from compromising his organisation's network and eavesdropping on the data flowing through it.

References

[1] Plummer David C. *RFC 826: An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48-bit Ethernet Address for Transmission on Ethernet Hardware*, Network Working Group, November 1982

[2] http://www.oxid.it : Website of the creators of Cain & Abel

[3] Fleck, Bob & Dimov, Jordon *Wireless Access Points and ARP Poisoning*, Cigital Inc. http://www.cigital.com

[4] Whalen, Sean *An Introduction to ARP Spoofing*, April 2001, Revision 1

[5] Finlayson Ross, Mann Timothy, Mogul Jeffrey, Theimer Marvin *RFC 903: A Reverse Address Resolution Protocol*, Network Working Group, June 1984