# ARP Spoofing Detection Algorithm Using ICMP Protocol

Gao Jinhua
School of Computer and Communication Engineering
University of Science and technology Beijing
Beijing 100083, China
gagybaby@163.com

Xia Kejian
School of Computer and Communication Engineering
University of Science and technology Beijing
Beijing 100083, China
bjxkj@vip.163.com

*Abstract*—**Today, there are an increasing number of attack technologies among which ARP spoofing attack is considered as one of the easiest but dangerous method in local area networks. This paper discusses ARP spoofing attack and some related works about it first. On these bases, the paper proposed an efficient algorithm based on ICMP protocol to detect malicious hosts that are performing ARP spoofing attack. The technique includes collecting and analyzing the ARP packets, and then injecting ICMP echo request packets to probe for malicious host according to its response packets. It won't disturb the activities of the hosts on the network. It can also detect the real address mappings during an attack.**

***Keywords- ARP cache; ARP protocol; ARP spoofing attack; ICMP protocol***

## I. INTRODUCTION

ARP is a protocol that dynamically maps a protocol address to a given hardware address (MAC address) [1]. When a host wants to communicate with another host whose hardware address it does not know, it broadcasts an ARP request for the hardware address associated with the protocol address of the destination. And only the host with corresponding protocol address sends a unicast reply to the sender with its < protocol address, hardware address,> pair. Obviously, ARP protocol plays a key role in local area network communication, but due to its own loopholes, it is often used as part of other serious attacks such as Man-in-the-Middle (MiM) attack, Denial of Service (DoS) attack. With a MiM attack, the attacker can sniff the traffic between two victim hosts. With a DoS attack, the attacker makes a victim host deny communicating with others. So ARP spoofing attack is becoming the most dangerous attack in the LAN. This paper proposes a comprehensive method to detect ARP spoofing. The proposed detection method generates an ICMP packet according to the received ARP packet, and then sends it to the network's host. It identifies a malicious host by collecting and analyzing the response packets. Although the method injects additional packets into the network, the performance of the network will not be affected, since a very small amount of packets are injected. At the same time, we will see the activities of the hosts are not disturbed.

The rest of the paper is organized as follows. In section 2 ARP spoofing attack and several serious ARP attacks are described. Section 3 provides an overview of currently available techniques to deal with ARP attacks. Section 4 discusses the proposed method. Section 5 describe the algorithm with natural language and C language And finally in section 6 conclusions are made.

## II. BACKGROUND

In order not to make the same request in the near future, every host maintains a table called ARP cache to store the address pairs learnt from the network. The host who issued the request will cache the address pair taken by the reply. There are probably two types of entries in an ARP cache: Static entries which remain in the ARP cache until the system reboots, Dynamic entries which only remain in the ARP cache for few minutes. Most operation systems allow creating a new entry by an ARP reply packet and all operation systems allow creating a new entry by an ARP request packet. All operation systems allow update an old entry by an ARP request or reply packet [2].

As ARP is a stateless protocol and its reply packets are not authenticated, all hosts blindly cache the ARP replies they receive from the network. This mechanism provides convenient for malicious host. Attackers send forging ARP packets to the victim periodically to perform ARP spoofing attack. In an ARP spoofing attack, the attacker sends ARP request or reply packets with fake <IP, MAC> mappings. For example, if a malicious host wants to sniff traffic sent from X to Y, he could send X an ARP packet with the address mapping <IP of Y, MAC of attacker>. Host X will cache the wrong address mapping and send data destined to Y to the attacker instead. If the attacker also wants to know the information sent from Y to X (MiM attack), the attacker just needs to send Y an ARP packet with address mapping <IP of X, MAC of attacker>. In order not to interrupt normal communication between host X and Y, the attacker need to enable IP packet routing to redirect the packet to the original destination host. If the attacker wants to perform a DoS attack, the attacker can poison the ARP cache of a host in the same way. Every packet the host sends is sent to the attacker. Once the attacker receives the packet, he simply drops it, and therefore, blocks the communication of the victim host.

Even a secure connection such as SSL or SSH is not automatically immune [3]. Although the application will warn the user that the certificate for this connection is invalid, many

978-1-4673-2907-1/13/$31.00 ©2013 IEEE

users choose to ignore the warning making the security measures perform practically no function. In addition, loopholes of some Internet Explorer make it possible to attack SSL connections even without a warning. This problem makes almost every internal communication vulnerable to the attack. Worse still, there are some tools that make it easy to perform ARP attacks even the attacker is not so experienced.

## III. Related Works

One simple but effective way to prevent ARP attacks is using static entries in the ARP cache. The drawbacks of this solution are its low scalability. It does not work well in dynamic environment and it would be a really heavy work for the network administrator to deploy and update these tables throughout the network especially when the network is big.

Gouda et al.[4] proposed an architecture to resolve IP addresses into MAC addresses over an Ethernet. This solution is not practical because it requires changing the ARP protocol implementation of every host with this new address resolution protocol. Furthermore, the secure server represents a single point of failure in the network, and becomes an obvious target for DoS attacks.

Secure ARP protocol (S-ARP) [5] is a backward compatible extension to ARP. This solution involves cryptography to authenticate the origin of ARP packets. To implement this solution in a LAN, every host has to be modified to use S-ARP instead of ARP. This is not scalable to update a stack across all available operating systems. Another disadvantage of this method is that it has the additional overhead of cryptographic calculations as S-ARP uses Digital Signature Algorithm (DSA).

Anticap [6] is a kernel patch for UNIX-based operating systems. It prevents ARP poisoning attacks by rejecting ARP updates that contain a different MAC address from the current table entry for the same IP address. This solution works only in static environment, and is available for a limited number of operating systems.

Some high-end Cisco switches have a new feature which allows the switch to drop ARP packets with invalid <IP, MAC)>address bindings [7]. One disadvantage of this feature is its high cost, another one is that it might not be able to verify some ARP packets on all switches in the VLAN.

## IV. The Proposed Algorithm for Detecting ARP Spoofing

The following sections describe the details of the algorithm we propose. Firstly, we construct an experiment simulation network, and practice our technique based on this network.

### *A. Experiment Simulation Network*

The composition of the experiment is listed here (see Fig. 1):

- VMware workstation Ver8.0.3.
- Installed two Window XP systems on VMware.
- Installed Red Hat9 system on VMware.
- Colasoft Packet Builder Ver1.0 [8]: running on suspicious host to send ARP packets to the victim.
- Program used to capture ARP and ICMP packet, analyze packet and send trap ICMP ping packets based on WinPcap technique [9].

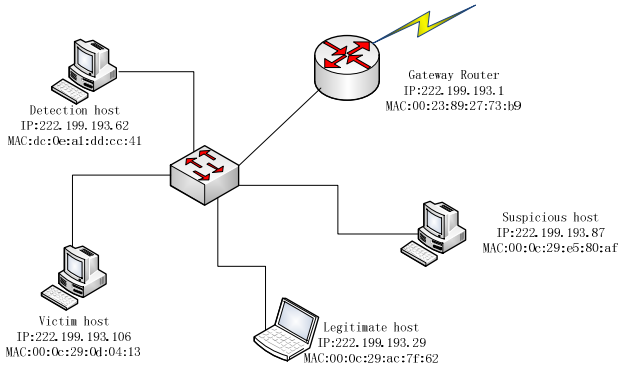


Figure 1 The Experiment Simulation Network Structure

Fig. 1 shows the topology of the experiment network. Suspicious host (IP:222.199.193.87) will send ARP packets to victim host (IP:222.199.193.106), detection host (IP:222.199.193.62) will capture the packets and send trap ICMP ping packet to the suspicious host, and make a judgment on the suspicious host according to the responding packets.

In this paper, we mainly use two types of packet: ARP packet and ICMP ping packet. The headers are as follows:

```
/*structure of Ethernet header*/
    struct  ETHER_HEADER
    {
        u_char dmac[6];
        u_char smac[6];
        u_short type;
     };
/*structure of ARP header*/
    struct ARP_HEADER
    {
        unsigned short arp_hdw_type;
        unsigned short arp_pro_type;
        unsigned char arp_mac_len;
        unsigned char arp_pro_len;
        unsigned short arp_opt_type;
        unsigned char arp_src_mac[6];
        unsigned char arp_src_ip[4];
        unsigned char arp_dst_mac[6];
        unsigned char arp_dst_ip[4];
    }
/*structure of ICMP header*/
    struct ICMP_HEADER
    {
         byte i_type;
         byte i_code;
         ushort i_cksum;
         ushort i_id;
         ushort i_seq;
    };
```

### *B. Architecture*

As shown in Fig. 2, we adopt a modularized approach and divide our ARP spoofing detection into the following modules:

- **ARP Packet Sniffer Module**: This module sniffs all ARP packets from the Ethernet.
- **Invalid Packet Detection Module**: This module classifies the ARP packets into valid and invalid packets in two steps. If there are any invalid packets turning up, it will be guaranteed that an ARP attack is occurring. All the new IP-MAC mappings are sent to the next module for in-depth analysis. We will discuss this module in Section C.
- **ARP Spoofing Detection Module**: This is the main detection module. We feed the new valid packets into it as input. The details of the module will be discussed in Section D.
- **IP-MAC Mapping Database**: IP-MAC mappings proved to be valid will be added into the database.
- **Response Module**: This module is used to alert the administrator the happening of ARP spoofing attack, automatically create and send repaired ARP packets to the victim, and deny the hosts that identified as malicious hosts by creating and sending ARP packets with random MAC address to the attacker.

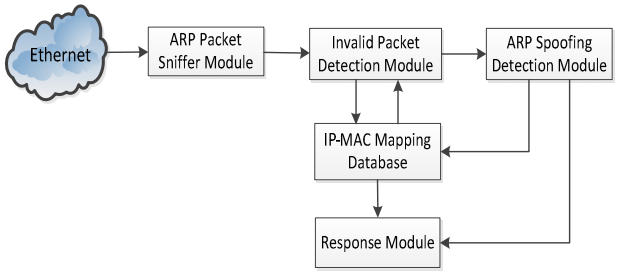


Figure 2 ARP spoofing detection algorithm architecture.

### *C. Invalid Packet Detection Module*

Packets we get from ARP Packet Sniffer Module will go through two steps in this module and finally be classified into two types: valid and invalid packets. Invalid packets with contradictions will be discarded and the program will turn to Response Module to alert an administrator. Valid packets will be sent to the next module.

**Step1: Detection of Fake Packets with Conflicting Hardware Addresses**

This step is used to do a cross-layer detection. If the source and/or destination MAC address in the Ethernet header are not the same with that in the ARP header, we will see this packet as invalid, and guarantee there are ARP spoofing attacks happening. Fig. 3 shows two invalid packets we capture from the network. These packets turn out to be valid will be sent to Step2 to check out whether the address map has an entry in the database.

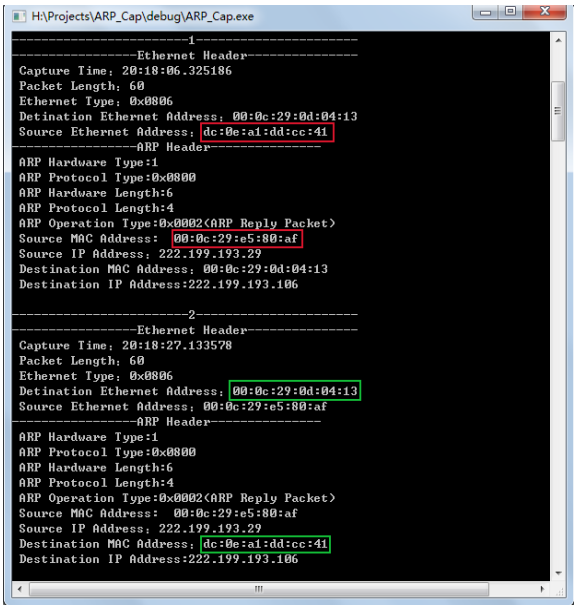


Figure 3 Fake packets with conflicting hardware addresses

**Step2: Detection of Fake Packets with Inconsistent IP-MAC Mappings**

This step is used to filter the valid packet we have got in the last step by comparing IP-MAC address mappings of the valid ARP packet with that in the IP-MAC mapping database. It will drop all packets coherent with the database entries. If there are any conflicts, it can be assured that there are ARP spoofing attack occurring, and then it turns to Response Module for help. All newly seen address pairs are passed on to the ARP Spoof Detection Module. Below are the details about the module.

### *D. ARP Spoofing Detection Module*

The ARP Spoofing Detection Module is the heart of the whole architecture which applies our detection algorithm to detect ARP spoofing. It works based on two rules as follows.

**Rule 1**: The NIC of a host only accepts packets with its own hardware address, broadcast address, and subscribed multicast addresses. The network layer only accepts IP packets addressed to its IP address and will drop the other packets silently. For example, there is a host with MAC address X and IP address Y, it would accept packet with destination MAC address X and destination IP address Z as the destination MAC address matches, but still discard the packet as the destination IP address doesn't match, without sending any error messages back to the source host.

**Rule 2:** Hosts with enabled IP packet routing will forward the packet to the destination host. All legitimate hosts in the network do not enable IP packet routing and will response back after it receives an ICMP echo request packet.

Based on the two rules, we can verify the ARP packets we've got (no matter they are request or response packets) whether they are real or fake packets. We will divide this module into two steps as follows:

**Step1: Generation of trap ICMP echo request packet**

A ping packet is an ICMP packet usually used to detect the connectivity between two hosts. When a host A wants to ping another host B in a network, it will sent B an ICMP echo request packet and wait for an ICMP echo reply packet sent by B. It is worth mentioning that the identifier and sequence number fields of the echo reply packet are set to the same with the received echo request packet in order to facilitate the sender to match the echo reply with the request packets.

When the detection host gets an ARP packet with source MAC address X and source IP address Y which does not have an entry in the mapping database, we will regard the source addresses <MAC address = MAC-X, IP address = IP-Y> as the addresses of the suspicious host. And then it will construct a trap ICMP ping packet with them as the destination addresses. The value of the source addresses is set to the addresses of the detection host (222.199.193.62, DC: 0E: A1: DD: CC: 41). Table 1 shows the value of the main fields of the trap ICMP ping packet. When an ICMP packet as constructed below is sent to the source of the ARP packet, the host's response will be based on Rule1 and Rule2.

TABLE I. TRAP ICMP PING PACKET

| **Ethernet Header** | |
|---|---|
| Destination MAC address = | MAC-X |
| Source MAC address = | DC: 0E: A1: DD: CC: 41 |
| Ethernet Type = | 0x0800 |
| **IP Header** | |
| Destination IP address = | IP-Y |
| Source IP address = | 222.199.193.62 |
| **ICMP Header** | |
| Type = | 8 (echo request) |
| Code = | 0 |
| Identifier = | - |
| Sequence Number = | - |

**Step2: Identification of the suspicious hosts**

In this step, we discuss the process to identify malicious hosts among the suspicious hosts. Based on Rule1 and Rule2, we can classify the suspicious host into three types: (1) malicious host without enabled IP packet routing; (2) malicious host with enabled IP packet routing: (3) legitimate host. Next, we will describe these three different situations one by one. We assume that the suspicious host has sent an ARP request packet showed in Fig. 4, and the algorithm will send an ICMP packet constructed using the source MAC and IP address advertised in the ARP packet automatically.

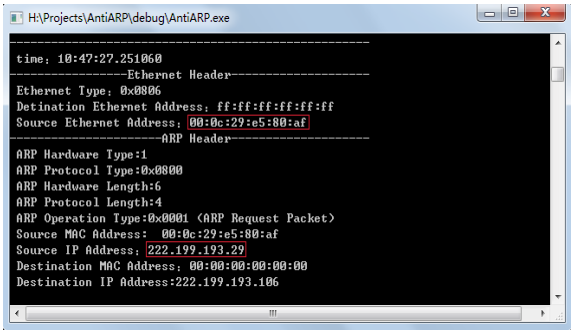


Figure 4 ARP packet sent by a malicious host

*1) Malicious Host without enabled IP packet routing*

Fig. 5 shows the details of the ICMP packet. When the echo request packet arrives at a malicious host without enable IP packet routing, it will be passed on to the IP layer. The IP layer of the host will find that the destination IP address (222.199.193.29) is not addressed to its IP address (222.199.193.87), and will drop the rest of the packet silently. Therefore, the detection host would not receive any responding messages. So the previously received ARP packet will be seen as a forged one. The program will alert an administrator to fix this problem.

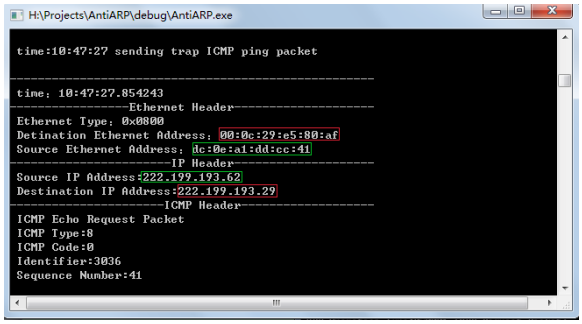


Figure 5 Trap ICMP ping packet sent by the detection host

*2) Malicious Host with enabled IP packet routing*

The malicious host enables IP packet routing to forward the received packets to the original destination, so it can sniff the traffic between two victims without interrupt the communication between them and the two victims will not notice his existence. When the malicious host with enabled IP packet routing receives the trap ICMP ping packet showed in Fig. 5 and the IP layer of the host find the destination IP address (222.199.193.29) does not match its IP address (222.199.193.87), it will forward the received packet to the victim host as a router does. So host with the destination IP address will receives this ICMP echo request packet and responds back with an echo reply packet.

In order to validate this situation, the detection host must sniff the ICMP echo reply packet that has identical identifier and sequence number with the echo request packet. Fig. 6 shows the echo reply packet. We can see that it has the same identifier and sequence number with the echo request packet showed in Fig. 6. And then we will compare the source MAC address in the reply packet with that in the ARP packet.

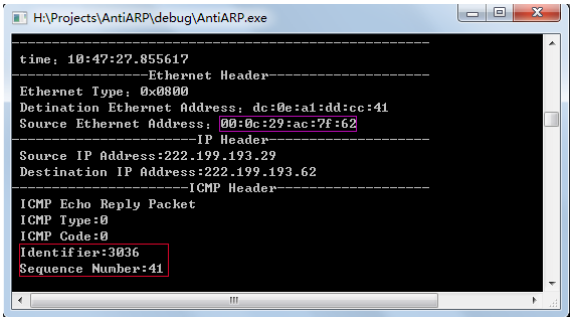


Figure 6.ICMP echo reply packet sent by the true host 222.199.193.29

From Fig. 4-6, we know that they have different source MAC addresses corresponding to the same IP address. So we can be sure that the ARP packet is a fake one and the suspicious host is not authentic. We will raise a spoofing alarm to alert an administrator.

Note that not only we have detected ARP spoofing attack but also have detected the IP and MAC address mapping of the true host on the network, as only the true host will reply back with echo reply packet.

*3) Legitimate Hosts*

If the suspicious host is a legitimate host, then < IP-Y, MAC-X > must be correct address mapping <222.199.193.87, 00:0c:29:e5:80:af>. Our algorithm will captures this address mapping, and then send back a trap ICMP ping packet. From Fig. 7 we can see the suspicious host receives the trap ICMP ping packet successfully, and respond back with an ICMP echo reply packet with exchanged source and destination addresses. The detection host will receive this reply packet within a timeout time, and recognize that this host is a legitimate one. The program will add this entry into database as a legitimate address mapping.

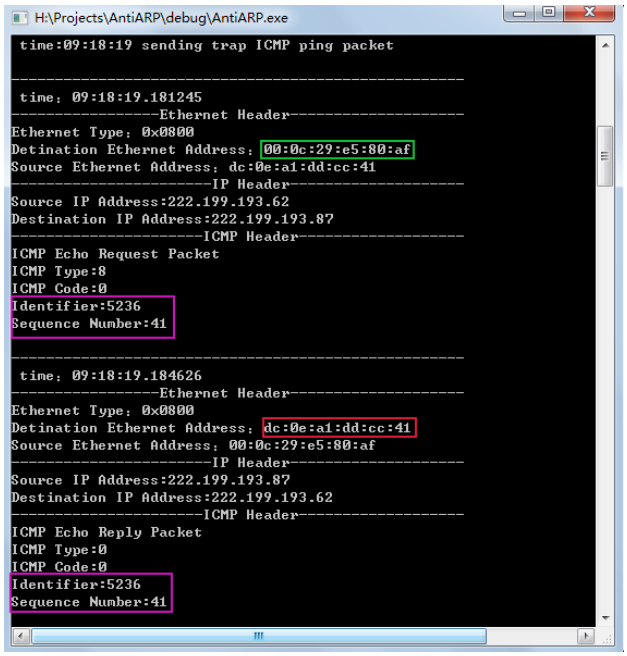


Figure 7 ICMP packet sent to and from a legitimate host

ICMP routing redirect message [10] is a legitimate protocol feature used to notify a source of traffic about its suboptimal use of routing. When the malicious host with enabled IP packet routing receives packet addressed to other hosts, it will forward the packet, and at the same time, it will send the source host a redirect for host packet. Consequently, if the malicious host is performing man in the middle attack, there will be a large number of ICMP redirect packets on the network. Administrator should pay attention to this abnormal traffic. By analyzing the packet, we could get the true address pair of the malicious host from the Ethernet header and IP header. And then we could corrupt its ARP cache, using ARP cache poisoning.

## V. THE DETECTION ALGORITHM FOR ARP SPOOFING

The algorithm is described by natural language and C language as follows:

```
if(ntohs(ether_protocol->ether_type)==0x0806)
                              /*if the packet is an ARP packet*/
ARPattackDetection()
{
    ArpPacketAnalysis();

    if (arp_dst_mac!= dmac ||arp_src_mac != smac)
    {
          Droppacket();
          ResponseModule();
    }

    QueryMapping DB();

    if (have an entry)
    {
          if(IP-MAC pair of the packet is coherent with the
                                                    DB entry)
          refresh the entry();

          else if(IP-MAC pair of the packet is not coherent
                                                with the DB entry)
          {
                Droppacket();
                Response Module();
          }
    }
    else if (don't have an entry)
    {
          SendICMPPacket();

          if(receives ICMP reply packet with identical
                      identifier and sequence number)
          {
              if(arp_src_mac==icmp_src_mac&&
                                  arp_src_ip==icmp_src_ip)
                  SetNewEntry();

              else if(arp_src_mac!=icmp_src_mac&&
                                  arp_src_ip==icmp_src_ip)
              {
                    Get address pair of the true host;
                    Droppacket();
                    ResponseModule();
               }
          }
          else if(no ICMP reply packet with identical
                      identifier and sequence number)
          {
                Droppacket();
                ResponseModule();
          }
     }
}
```

## VI. CONCLUSIONS

In this paper we study the theory of ARP spoofing attack and various existing techniques proposed to defend against this attack, and then, we proposed a comprehensive method to deal with ARP spoofing problem. Firstly, it does a cross layer control to examine the consistency of the source and destination address in Ethernet header and ARP header. Secondly, it compares the address mappings in valid ARP packets with those in the database. Finally, all new ARP packets will be sent to ARP Spoof Detection Module to be re-verified. As the method we proposed to probe the authenticity of every ARP packet is very active, the time delay between capturing the packets and detecting spoofing attack is minimum. We send one trap ICMP ping packet for each newly seen ARP packet on the network and then infer its authenticity according to the responding to our packet. In the meantime, the IP-MAC mapping database has stored a large proportion of address mappings on the LAN, so the network overhead due to our packet injection is fairly minimal and the performance of the network will not be influenced. In addition, our method can also detect correct IP-MAC address mappings of both the true host and the malicious host during an actual attack.

## REFERENCES

[1] Stevens, R.: TCP/IP Illustrated: vol. 1 (2001)

[2] Zouheir Trabelsi and Khaled Shuaib. Spoofed ARP Packets Detection in Switched LAN Networks. J. Filipe and M.S. Obaidat (Eds.): ICETE 2006, CCIS 9, pp. 81–91, 2008.

[3] T. Demuth and A. Leitner. ARP spoofing and poisoning: Traffic tricks. Linux Magazine, 56: 26–31,July2005

[4] M. Gouda and C.-T. Huang. A secure address resolution protocol. Computer Networks, 41(1):57–71, Jan. 2003.

[5] D. Bruschi, A. Ornaghi, and E. Rosti. S-ARP: A secure address resolution protocol. In Proceedings of the19th Annual Computer Security Applications Conference (ACSAC '03), Dec. 2003.

[6] M.Barnaba.anticap.<http://www.antifork.org/viewcvs/trunk/anticap>.(Last accessed April 17, 2006).

[7] Cisco Systems. Configuring Dynamic ARP Inspection, chapter 39, pages 39:1–39:22. 2006. Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide, Release12.2SX.

[8] Packet Builder. http://www.colasoft.com/packet_builder

[9] WINPCAP: http://www.winpcap.org/docs/docs_412/html/main.html

[10] ICMP_REDIRECT_Messages. http://www.embeddedlinux.org.cn/linux_net/0596002556/understandlni-CHP-31-SECT-6.html