Detection and Prevention against ARP Poisoning Attack using Modified ICMP and Voting

Prerna Arote
ABV-Indian Institute of Information Technology and Managament, Gwalior, India
arote.prerna@gmail.com

Karam Veer Arya
ABV-Indian Institute of Information Technology and Managament, Gwalior, India
kvarya@iiitm.ac.in

Abstract— **Address Resolution Protocol (ARP) poisoning is the leading point for refined LAN attacks like denial-of-service (DOS) and Man-In-The-Middle (MITM). Weakpoint of ARP that is being Stateless, directly affects security standards of Network and specially Ethernet. In proposed mechanism of detection, initially traffic over the network is sniffed by Central Server (CS). Then, CS sends trap ICMP ping packet, analyze the response in terms of ICMP reply and successfully detects attacker. In order to prevent ARP poisoning over centralized system, voting process is used to elect legitimate CS. Validating and Correcting < IP, MAC > pair entries residing in hosts cache tables, CS successfully prevents ARP poisoning while maintaining performance of the system. Our technique is based on ICMP and Voting such mechanism with Backward Compatibility, Less Cost, Minimal Traffic and Easily Deployable is proposed to detect and prevent MITM based ARP poisoning which is effectual version overcoming weaknesses of ARP.**

Keywords- Address Resolution Protocol, Man-In-The-Middle, ARP poisoning, ICMP ping packet, MAC address, Attack.

Address Resolution Protocol (ARP) is specifically used to convert protocol address (IP) into hardware address (MAC). RFC 826 had already defined Address Resolution Protocol in 1982 [1]. ARP was designed for a more trusting world. While processing packets and passing them down the stack, the responsible factor for framing the packets is Data Link Layer (DLL). In LAN environment, the MAC Address scheme which is used to transfer data operates at DLL. While the Network Layer will have provided the IP address, there is a need to provide a physical address. A Threat is a potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. Therefore, chances to exploit a vulnerability caused by threat which is a possible danger. Internet Control Message Protocol (ICMP) is a network protocol useful in Internet Protocol (IP) for performing administration and managing network. ICMP is a key element of IP implementations. ICMP is also called as a control protocol, explaining the thing that, it not only carry application data, but also information about the status of the network by itself [4]. ICMP mainly used by Ping command for Echo Request and Echo Reply. The basic idea of presented architecture is to design such structure on more than 3 systems, for transmitting ARP and ICMP packets, or the central server plays an important role in proposed scheme. During the failure of any one of the systems, other system can work alone. Therefore, backward compatibilty is achieved with the original ARP structure. As well as, it is less costly and easy to use. Rest of the paper is organized as follows. Section 2 discusses context of ARP and ARP Spoofing. Section 3 covers the preventive approaches for ARP poisoning. Section 4 presents proposed scheme. Section 5 describes experimental setup, results and performance analysis. Finally we conclude in Section 6.

I. BACKGROUND

A. ARP

The Address Resolution Protocol (ARP) [1] is main factor used for discriminating MAC addresses of each other over the network. Resolving IP address into MAC address is the main task of Address Resolution Protocol. There are four types of messages that can be send through ARP, those are ARP Request, ARP Reply, RARP Request and RARP Reply. Working of ARP Messages is shown in Fig. 1.

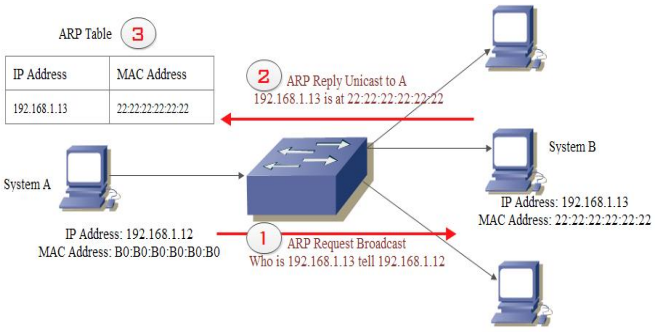

Fig. 1: ARP request by system A and ARP reply from system B

System A with IP address 192.168.1.12 broadcasts an ARP request to get MAC address of 192.168.1.13. Further, System B unicasts ARP reply to System A with it's 22:22:22:22:22:22 MAC address. Then, it get stored in ARP cache table.

B. ARP Cache Poisoning

In a LAN environment, when a Host A needs to know the MAC for a particular IP address, it broadcast an ARP Request asking for MAC Address. The system with the IP address will

2375-5822/15 $31.00 © 2015 IEEE
DOI 10.1109/CINE.2015.34

unicast reply to host A on its MAC Address. Host A then stores the < IP, MAC > pair in it's ARP Cache Table. ARP does not support any authentication and thus can be easily spoofed. A simple script using linux to perform MITM based on this attack:

```
#!/bin/bash
echo1 /proc/sys/net/ipv4/ip_forward
 arpspoof -i eth0 –t  192.168.1.32 192.168.1.1 &
  arpspoof -i eth0 -t 192.168.1.1 192.168.1.32 &
```

When Victim broadcast an ARP Request for gateway. The Attacker replies with ARP Reply packets and effectively poison the victims ARP Cache. Thus, the attacker becomes MITM between gateway and victim by:

- poison the victim so that gateways IP address gets mapped with attackers MAC address and
- poison the gateway so that victims IP address gets associated with attackers MAC address and
- forwarding the packets the attacker receives to victim/gateway.

Now the attacker is MITM between victim and gateway [2], [3].

II. RELATED WORK

There are bunch of solutions proposed in the literature to detect, and prevent such ARP related attacks. The schemes can be broadly classified as:

Solution based on patches and tools: Technique called Antidote which is used for checking host is alive or not and unicasting ARP packets in the network, is proposed by Teterin [5]. Anticap [6] which is linux based kernel patch works only in static environment, and is constrained to be used by a very few operating systems. Cisco switches allows to drop ARP packets with incorrect <IP, MAC> entry but it has very high cost [7]. Hou et al. [8] invented the solution for detecting MITM using snort is presented where, the MITM attacks are detected based on static <IP, MAC> mapping. But, it does not provide a solution to MITM between two hosts for which static entry is not saved.

Cryptographic Solutions: T-ARP using cryptographic solutions [9] has been proven that improved value of executing ARP security over existing protocols by two orders of magnitude. But, the solution lacks in, maintaining public key of each host, upgradation of network stack to configure all host, processing overhead of sign generation, verification and key management. Hybrid cryptographic solution using a mixed combination of digital signatures and one time passwords [10]. Bruschi et al. [11], presented S-ARP using concept of Public/Private key pairs, digital certificate and Authoritative Key Distributor (AKD). However, it has a high computational cost.

Modified Protocol based Solutions: Stateful ARP concept, where main focus on extending the existing the standard ARP protocol by changing the ARP cache being a stateful from a stateless one [12]. Drawbacks of this system are, no support of gratuitous request or reply and modification of protocol leads to complex problems. Issac and Mohammed presented S-UARP [13] new protocol, where ARP messages were played by DHCP server. It reduces broadcast congestion occurred within network during process. It failed to detect malicious node in the network. Moreover, it requires to upgrade the DHCP Server. Still, incremental deployment is tough.

Techniques Proposing Architecture: An architecture by S. Y. Nam [14], which efficiently mitigates ARP poisoning based MITM using puzzle based system, easily deployable and require no manual configuration. However, the significant disadvantage is to maintain fairness among different nodes where computational power of machines is more diverse. Detection using ICMP is presented by Jinhua and Kejian [15] where, malicious hosts are detected which are performing ARP spoofing attack. However, no solution provided to prevent the attack and there is a need to support backward compatibility.

Our proposed solution will overcome these drawbacks of previous approaches in terms of cost, compatibility, traffic, and deployability.

III. THE PROPOSED SOLUTION

In this paper, we propose a new solution overcoming weakness of ARP cache poisoning. As already discussed, this mechanism provides backward compatibility with existing ARP, less cost and complexity in the network that is because we do not use the cryptographic techniques. In design of current system, we have used 4 systems connected over wired LAN. Victim will maintain 2 tables that is primary cache and secondary cache table. Central Server uses only secondary cache table. Algorithm 1 is mainly used for detection of ARP poisoning attack using Ettercap and SSLstrip. Algorithm for ARP poisoning prevention is divided into 3 modules. Where, first type includes client side implementation which is shown in Algorithm 3, then CS implementation which is shown in Algorithm 4, and finally CS antidote implementation as shown in Algorithm 5.

A. Algorithm for Detection and Prevention

The proposed scheme has two main modules detection and prevention. Modules describing detection using ARPspoof, ICMP and prevention using voting are discussed below.

Algorithm 1: Man-in-the-Middle based ARP poisoning attack over LAN: A current algorithm deals with 3 systems Victim, Attacker, and Gateway over the network connected through the wired connection. Attacker uses Ettercap and SSLstrip tools to play the role of man in the middle like an intruder. Algorithm forwards IP tables and ports using

SSLstrip. Then, it does redirecting of ARP traffic by listening on port 80. With the help of Ettercap, ARP poisoning is done between Victim and Gateway using arpspoof command. So, that all the traffic is routed to an Attacker. Then, python script of sslstrip is run on port 80 to listen for the packets flowing between the host over network. It captures all kind of traffic including arp request and arp response packets flowing among hosts. In next Algorithm, ARP poisoning detection using ICMP protocol is explained which is prevented with 3 modules of Prevention Algorithm.

Algorithm 1 Man-in-the-Middle based ARP poisoning attack over LAN

MITM_ARPspoof ETTERCAP();
Input: Victim, Attacker, Gateway, Ettercap, SSLstrip;
Enable IP forwarding;
ip forward=1;
Redirecting HTTP traffic to SSLstrip by setting iptables;
iptables -t nat -A PREROUTING -p tcp - -destination-port 80 -j REDIRECT –to-port < listenPort >;
Do arpspoof between the target and gateway;
arpspoof -i eth0 -t < targetIP > < gatewayIP >;
Run SSLstrip;
sslstrip.py -l < listenPort >;
Run Ettercap;
ettercap -T -q -M arp:remote -i eth0 /< TargetIP > // < GatewayIP > / -p remote-browser;

Algorithm 2: ARP Poisoning Detection using ICMP: Here, ICMP protocol, which is used by CS for detection and it is able to find out whether host is either malicious or legitimate. Main role is played by CS. Because, CS does sniffing of the packets flowing towards victim from all other hosts in the network. . We have considered only one host that is Attacker here.Trap ICMP ping packet used to identify, the identifier and sequence number field. In Algorithm 2, sequence number and identifier having value 0 indicates success. If attacker machine is not willing to harm or change cache table contents by overwriting the cache table contents, then CS nominates Attacker machine as legitimate otherwise, malicious.

Algorithm 2 Algorithm for ARP Poisoning Detection

ARP_Poisoning_Detection();
Input: Victim, CS, Attacker, Gateway;
for Central_Server **do**
Do monitoring the traffic of ARP packets flowing towards Victim from ALL_Hosts;
Maintain a secondary long term ARP cache;
Capture those packet;
for All_Host **do**
Send TRAP ICMP ping packet;
Check replies with Identifier and Sequence Number;
if *Identifier*=0 and *Sequence No_*= 0 **then**
Check for source IP of ARP and ICMP packet header;
if < IP, MAC > pair matched **then**
Nominate it as Legitimate;
else
Nominate it as Malicious host;
trying to do ARP spoofing attack;
end if
else
ARP Poisoning Detected;
end if
end for
end for

Algorithm 3: Client Side ARP Poisoning Prevention: is an client side implementation over centralized system which consist of a private or local network, where client wants to check that it's cache table entry is poisoned or not. Algorithm 3 is used when any new node joins the network and want to communicate to CS to get correct < IP, MAC > pair. It sends voting request to all other systems present over the network. Then, collect replies from Voting Cognizant hosts by waiting for random interval of time 0 to 100 msec. For each MAC collected, new node finds out polling score. The MAC address for which it gets votes more than 50 percent, will be accepted[14]. Primary and Secondary cache are updated with new MAC address.

Algorithm 3 Algorithm of ARP Poisoning Prevention on Client Side

Client_Side_Prevention();
Input: New Node, CS, Node already present in the Network, Victim;
if newly joined node **then**
for each node present in network **do**
Broadcast voting request
end for
Collect replies from voting cognizant hosts with < IP, MAC > of CS
Wait for random time of 0 to 100 msec
for each MAC **do**
Find polling score
if (*pollingscore* > *0.5N*) **then**
Accept MAC as correct MAC of CS
Send ARP request to this MAC as CS's MAC
Collect < IP, MAC > reply from CS
Update primary cache and secondary cache of victim
end if
end for
end if
Do monitoring of ARP cache;

```
if change in primary cache then
   Check entry in secondary table;
   if < IP, MAC > pair present in Secondary Table then
      Goto Step "ChangeinARPcache" ;
       else
      Goto Step "Broadcastvotingrequest" ;
   end if
else
   Goto Step "Change inARPcache" ;
end if
```

Algorithm 4: CS Side ARP Poisoning Prevention: is one of the module for preventing ARP poisoning. Here, in CS side prevention, CS already having long term cache table. If any request for MAC given IP comes to CS, it will go for searching into seconadary cache. If entry found then reply containing pair is sent to Victim. Otherwise, request is broadcasted for IP and received MAC is stored into secondary cache.

Algorithm 4 Algorithm of ARP Poisoning Prevention on CS Side

```
CS_Side _Prevention();
Input: CS, Request for MAC given IP;
CS has long term cache table;
if req for MAC given IP then
   Search < IP, MAC > pair in secondary cache;
    if entry_found then
      Send reply to Victim;
   else
      Broadcast request for IP;
      Store received MAC in secondary cache table;
   end if
end if
```

Algorithm 5: (*CS Antidote*) ARP Poisoning Prevention: describes Antidote Solution for CS. It can be used only when cache table of CS get poisoned. This can rarely happen with our approach. If happened then, CS will check it's cache table contents if change found with incorrect pair, it will send 50 unicast request of ARP to earlier MAC and check for replies. If more than 1 replies then earlier < IP, MAC > pair is stored else primary and secondary cache table is updated with new < IP, MAC > pair.

Algorithm 3 Algorithm of ARP Poisoning Prevention CS_Antidote

```
CS_Antidote_Prevention();
Input: Secondary Cache without Requested IP_MAC entry;
if change_found then
   Check for pair in secondary cache;
   if incorrect_pair then
      Send 50 unicast request of ARP to earlier MAC;
       Check for replies;
      if reply > 1 then
         Store earlier < IP, MAC > pair;
         Goto step "ifchangefound";
      else
         Update primary and secondary cache table with new
         < IP, MAC > pair;
         Goto step "ifchangefound";
      end if
   else
      Update pair in primary cache table;
      end if
end if
```

Antidote approach using 50 unicast request is used to make sure the remote host would have replied back with atleast 1 ARP reply.

Result of MITM attack under this approach:

- Attack on CS where already present entry:

If there is any attempt by an attacker on CS, the CS will send 50 ARP request unicast packet to the earlier MAC of <IP, MAC> pair. CS will get response from victim if the host is alive. In this situation, new request is removed by CS and continue with earlier <IP, MAC> pair. Thus, it results in unsuccessful ARP poisoning.

- Attack on CS where no earlier entry:

If there is CS without being assigned an IP address and Attacker tries any attempt of attack on it. CS will accept that entry. So, all remaining nodes also honor that binding. It is quite similar to Attacker possessing the IP addresses. However, following the rules Attacker accepting it as a gateway can't poison any other IP address. It get limited to Attacker possessing multiple IP addresses. So, MITM is impossible here

IV. EXPERIMENTS AND EVALUATION

A. Experimental Setup

To implement the attack, we have taken a real network environment consisting of 4 systems. Oracle Virtual Box is setup using NAT gateway, where all 4 systems have backtrack operating system installed. Each system is having 2GB RAM. All are connected over wired network. Port forwarding is enabled. Following IP addresses and MAC addresses of the machines present in the network.

System	IP Address	MAC Address
Attacker Machine	10.0.0.3	08:00:27:26:3b:f8
Victim Machine	10.0.0.2	08:00:27:85:2b:5f
Gateway	10.0.0.1	08:00:27:3b:05:b1
Central Server	10.0.0.4	08:00:27:00:70:32

B. Result

Results of ARP Poisoning detection with setting as shown in subsection 5.1 are discussed here.

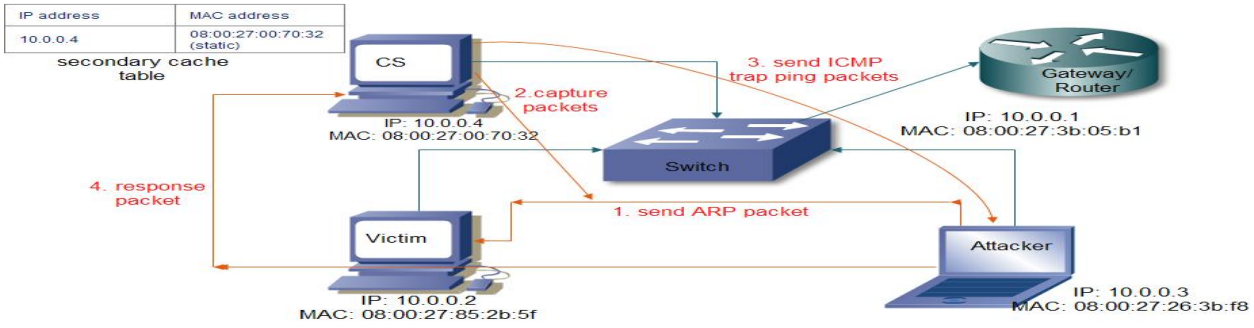

Fig. 2: Details of ARP Poisoning Detection

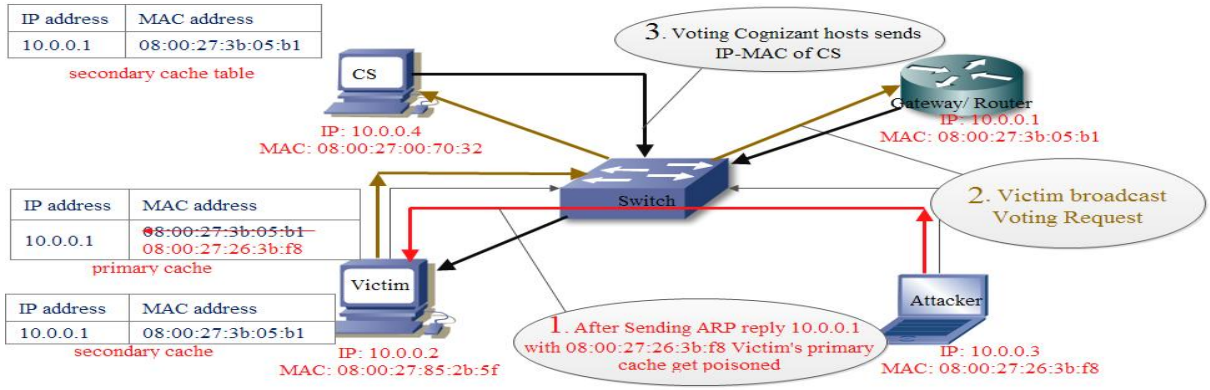

Fig. 3: Phase 1 of ARP Poisoning Prevention

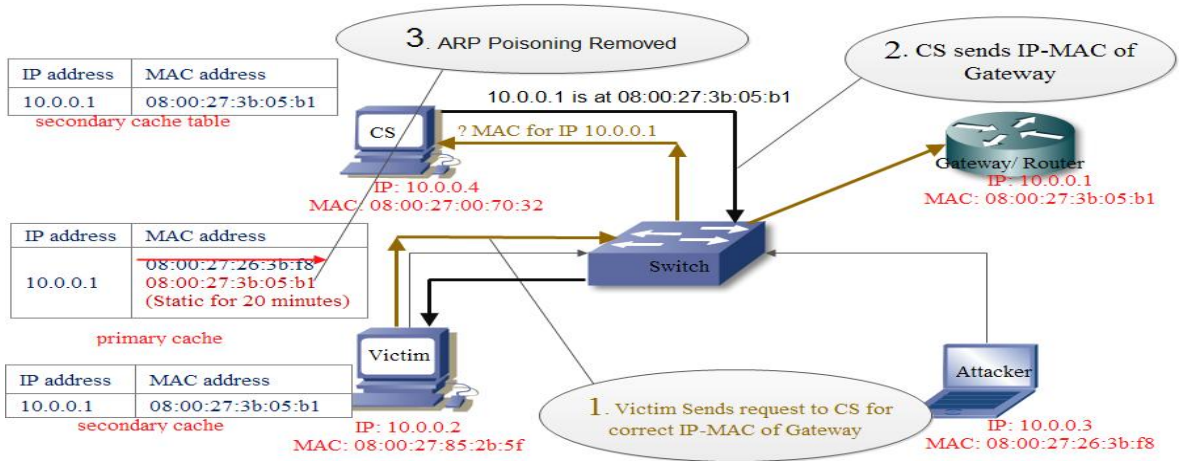

Fig. 4: Phase 2 of ARP Poisoning Prevention

Fig. 2. Shows detection of ARP poisoning which illustrates, first Attacker sending ARP packet to Victim, then CS captures those packets. CS sends ICMP trap ping packet to Attacker then Attacker sends response packets. Attacker with IP address 10.0.0.3 and Gateway with IP address 10.0.0.1 are poisoned with same MAC address 08:00:27:26:3b:f8. Thus, ARP poisoning is detected in Fig. 5.

Fig. 3. shows first module of prevention where, initially case after sending ARP reply to Victim then Victim broadcasts voting request. Voting cognizant hosts sends < IP, MAC > of CS and Fig. 4. illustrates second module of prevention where, Victim sends request to CS for correct < IP, MAC > of gateway then CS sends < IP, MAC > of gateway.

Fig. 5: ARP Poisoning Detected

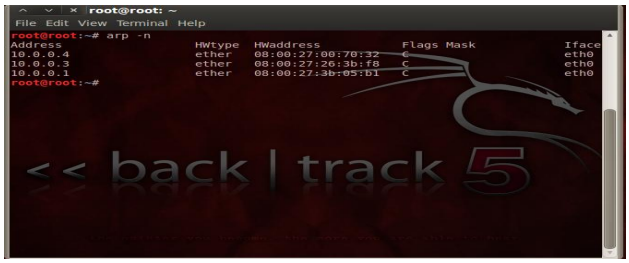

Fig. 6: ARP Poisoning Prevented

Fig. 6 shows the result of ARP poisoning prevention. Because, all hosts are now having their respective IP and MAC addresses that is 10.0.0.3 at 08:00:27:26:3b:f8, 10.0.0.1 at 08:00:27:3b:05:b1 and 10.0.0.4 at 08:00:27:00:70:32. Thus, ARP poisoning is removed.

C. Performance Analysis

In the network, in general if N nodes are present. Then, for each ARP packet having new < IP, MAC > pair, client needs to verify it with respect to it's secondary cache table. Just to see out, if match found or not. Complexity for such step will be O(logn). In case, if the match didn't find, then to send voting request, to get CS' s <IP, MAC> pair and then to send again request to CS to get correct pair. CS has to check against seconadary cache table. The Complexity for such step will be O(logn).

But, one more possibility that CS is also unable to find correct match, again he needs to send broadcast request and send reply back to Victim. It requires complexity of order 1, that is O(1). CS upadtes it' s cache with O(logn). Same thing can happens with Victim, when it needs to update it' s secondary cache, with the complexity O(logn). So, if we go for Worst Case complexity, then it will be O(logn).

V. EXPERIMENTS AND EVALUATION

In this work, a new approach for detection and prevention against ARP poisoning attack is proposed. Our solution is fully based on the ICMP and Voting over centralized system. In the proposed strategy, MITM attack over Secure Socket Layer is implemented with the help of SSLstrip and Ettercap, further Central Server does monitor and analyze the traffic flowing between Victim and Attacker and detects ARP poisoning attempt. Then Victim electing CS to prevent ARP poisoning attack. Efficient algorithms for the detection and prevention of ARP poisoning attack has several advantages over other previous approaches, like Backward Compatibility, as if central server fails then remaining systems will not be affected. It also requires less cost because of few systems in local network, usage of open source tools, and minimal traffic as all packets over the network are stored into the database. No change in ARP structure as here ARP protocol is not modified and it is also easily deployable. By doing this way, with the help of voting concept to elect CS, the possibility of Attacker pretending itself as CS is also removed. At present, the scheme can only using Voting for prevention. In other words, in case of prevention without modifying structure, ICMP can be used for preventing ARP poisoning with large number of nodes and there exists a need to protect the CS from attack of IP exhaust.

REFERENCES

[1] Plummer: An Ethernet address resolution protocol. RFC 826 (1982)
[2] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-Middle Attack to the HTTPS Protocol," Security & Privacy, IEEE, vol. 7, no. 1, 2009, pp. 78-81.
[3] S. M. Bellovin, "Security problems in the TCP/IP proto-col suite," in Special Interest Group on Data Communi-cation (ACM SIGCOMM'89), Volume 19, pp. 32-48.
[4] ICMP REDIRECT MESSAGES, Available at URL: http://www.embeddedlinux.org.cn/linux-net/0596002556/understandlniCHP-31-SECT-6.html.
[5] I. Teterin, "Antidote," 2002, Available at URL: http://online.securityfocus.com/archive/1/299929.
[6] M. Barnaba, "Anticap," 2003, Available at URL: http://cvs.antifork.org/cvsweb.cgi/anticap.
[7] C. Headquarters, "Cisco Security Appliance Command Line Configuration Guide," 2005.
[8] X. Hou, Z. Jiang, and X. Tian, "The Detection and Prevention for ARP Spoofing based on SNORT," in Proc. of IEEE International Conference on Computer Application and System Modeling (ICCASM'10), vol. 5, pp.V5-137.
[9] W. Lootah, W. Enck, and P. McDaniel, "TARP: Ticket-based Address Resolution Protocol," in IEEE Computer Society, vol. 51, no. 15, 2007, pp. 4322-4337.
[10] V. Goyal and R. Tripathy, "An Efficient Solution to the Arp Cache Poisoning Problem," in Information Security and Privacy, Springer, 2005, pp. 40-51.
[11] D. Bruschi, A. Ornaghi, and E. Rosti, "S-ARP: A Secure Address Resolution Protocol," in Proc. of Ninteenth An-nual IEEE Computer Security Applications Conference (ICSAC'03), 2003, pp. 66-74.
[12] Z. Trabelsi and W. El-Hajj, "Preventing ARP Attacks using a Fuzzy-Based Stateful ARP Cache," in Proc. of IEEE International Conference on Communications, (ICC '07), 2007, pp. 1355-1360.
[13] B. Issac and L. A. Mohammed, "Secure unicast address resolution protocol (S-UARP) by extending DHCP," in Proc. of Networks, 2005. Jointly held with the Proc. of IEEE Seventh Malaysia International Conference on Communication, vol. 1, 2005, pp.6-pp.
[14] S. Y. Nam, S. Jurayev, S. S. Kim, K. Choi, and G. S. Choi, "Mitigating ARP poisoning-based man-in-the-middle attacks in wired or wireless LAN," EURASIP Journal on Wireless Communications and Networking, vol. 2012, no. 1, 2012, pp. 1-17.
[15] G. Jinhua and X. Kejian, "ARP spoofing detection al-gorithm using ICMP protocol," in IEEE International Conference on Computer Communication and Informat-ics (ICCCI'13) 2013, pp. 1-6.