Mathematical and Computer Modelling 58 (2013) 254–260

Contents lists available at SciVerse ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Design and implementation of an efficient defense mechanism against ARP spoofing attacks using AES and RSA

Seungpyo Hong[a], Myeungjin Oh[b], Sangjun Lee[a,*]

[a] School of Computing, Soongsil University, Seoul, Republic of Korea
[b] Security R&D Center, Samsung Electronics Co., Suwon, Republic of Korea

ARTICLE INFO

Article history:
Received 29 July 2011
Received in revised form 26 July 2012
Accepted 9 August 2012

Keywords:
Address Resolution Protocol (ARP)
Spoofing
Network
Security

ABSTRACT

The Address Resolution Protocol (ARP) is used to resolve the MAC address of a host given its IP address. ARP is stateless, as there is no authentication when exchanging a MAC address between hosts. Hacking methods using ARP spoofing are being continuously abused in various ways, and there have been many prior studies of the prevention of such attacks. However, prevention requires the modification of the basic network protocol or expensive additional equipment, so it is hard to apply these methods to the current network. In this paper, we examine the limits of prior research into ARP spoofing prevention. In addition, we suggest a defense mechanism that does not require changes to the network protocol or expensive equipment. Our system automatically renews the reliable MAC address information to the ARP table as a static type to protect users from ARP spoofing.

1. Introduction

Internet usage has increased with the rapid development of the network, in particular its convenience, accessibility, and marketability [1]. However, this heavy dependency on the internet may create havoc if it is exposed to security vulnerabilities. Security [2–4] is one of the most important issues in various areas, such as networking, databases, and financial services. There is an urgent need for a system that can protect against Address Resolution Protocol (ARP) spoofing, which has recently been occurring at an increasing rate due to the security vulnerabilities of the ARP.

It is 30 years since the weakness of ARP was first discovered [5]. However, ARP spoofing is still widely used for system damage [6]. This proves that, even though many protection systems have been developed, they are still not sufficiently effective overall. There are several ways to prevent ARP spoofing; however, they require expensive equipment, and thus only selected facilities can operate these systems, while others are simply exposed to hacking without any protection or recovery systems.

Most protection methods suggested by prior researchers require additional hardware or modification of the current ARP. These are practically impossible, and they are difficult to apply due to the use of expensive equipment. The method of maintaining the ARP table using static ARP cache entries to stop manipulation without additional equipment is well known. Static ARP cache entries cannot be manipulated by ARP reply packets, so ARP spoofing is prevented. However, this method requires manual work by administrators to input every host's address, so it is practically impossible to perform for large network environments or a dynamic host configuration protocol (DHCP) environment. In our paper, we suggest a method that can automatically renew trustworthy information in the ARP table to protect users from ARP spoofing attacks. This eliminates the weakness of the ARP by automatically maintaining an ARP table of every host in the local network. In this paper, we extend our previous work [7] and provide more intensive experimental results and analysis that were not reported.

[*] Corresponding author. Tel.: +82 2 820 0672.
E-mail addresses: hsp8405@naver.com (S. Hong), myeongjin.oh@samsung.com (M. Oh), sangjun@ssu.ac.kr (S. Lee).

0895-7177/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2012.08.008

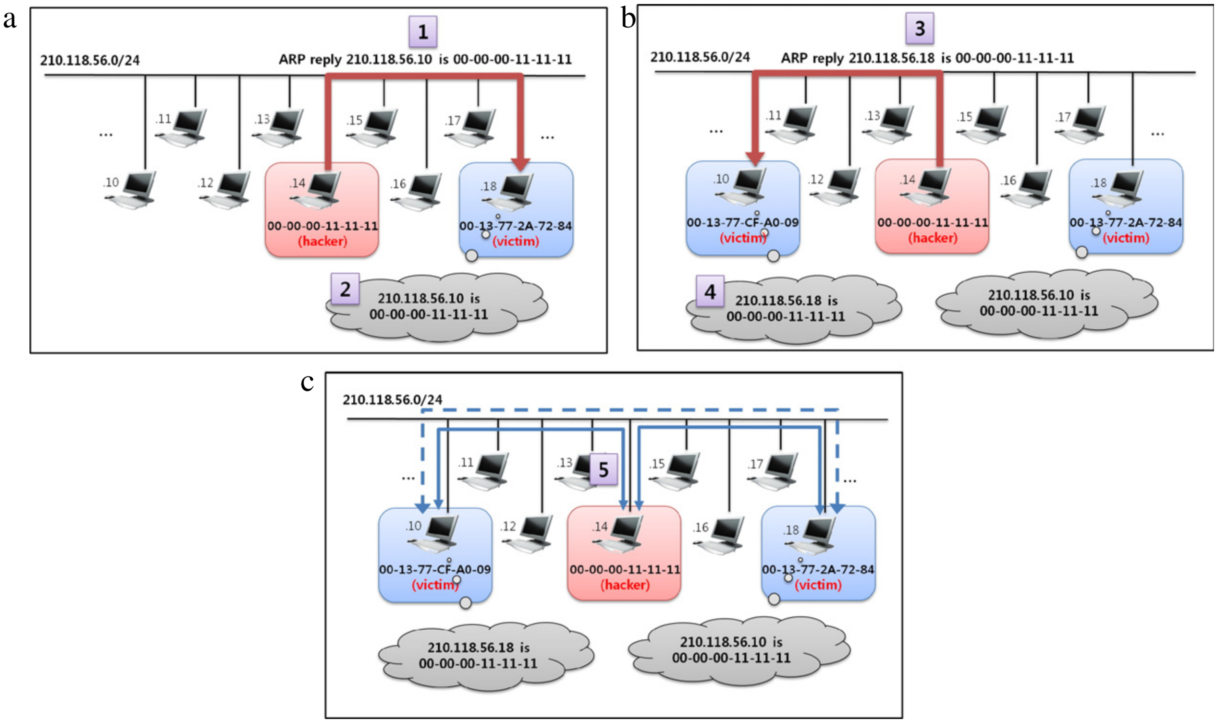


Fig. 1. ARP spoofing attack [9]. ① In (a), the hacker attacks one host (.18). ② Host .18 has the wrong ARP table. ③ In (b), the hacker attacks another host (.10). ④ Host .10 also has the wrong ARP table. ⑤ The hacker sniffs all packets between the two victims (.10 and .18).

The remainder of this paper is organized as follows. Section 2 discusses the context of ARP spoofing studies and prior related work. Section 3 covers the explanation of fixed ARP table maintenance. Section 4 describes the experimental results and analysis of our system and Section 5 presents our conclusions.

2. Background and related work

2.1. ARP spoofing

Before transmitting some data, a host broadcasts the ARP request if there is no MAC address in its ARP table that matches the target IP address. If there is a host with the requested IP in the same subnet, the host sends an ARP reply packet to the requesting host. There is no authentication in this process, so anyone can make a malformed ARP reply packet to send to other hosts, as shown in Fig. 1. The common host believes the ARP reply packet to be authentic, and the host's ARP table is easily manipulated by this reply. If a hacker modifies a normal user's MAC address to theirs, data transmission between normal users is exposed to the hacker without the normal users being aware. This is called a Man in the Middle (MITM) attack [8].

2.2. Related work

Typically, in prior research, three approaches have been used to protect the vulnerability of the ARP. Gouda and Chin-Tser [10] suggest a protocol that can solve uncertified ARP table renewal from an ARP reply. While this can overcome the problems of ARP, it is practically impossible to modify the current protocol. Ramachandran and Nandi [11] suggested effective search methods for ARP spoofing attacks. However, search methods do not solve the problem, and the management of such passive methods requires effort on the part of network administrators. Pansa and Chomsiri [12] presented a method of installing a new DHCP server for use as a MAC-IP database center. This research is similar, in context, to ours, but they suggested a new DHCP for MAC address transmission between each host. However, DHCP is also widely used and, in practice, hard to fix, so this method cannot be applied easily in a real-life situation.

3. Our approach

3.1. System outline

Our system consists of a MAC-Agent and a Client-Agent. The MAC-Agent makes a reliable ARP table and sends the data to the Client-Agent, which prevents the host from using ARP. Instead, the Client-Agent receives the reliable ARP table information from the MAC-Agent and updates the ARP table information as static type.

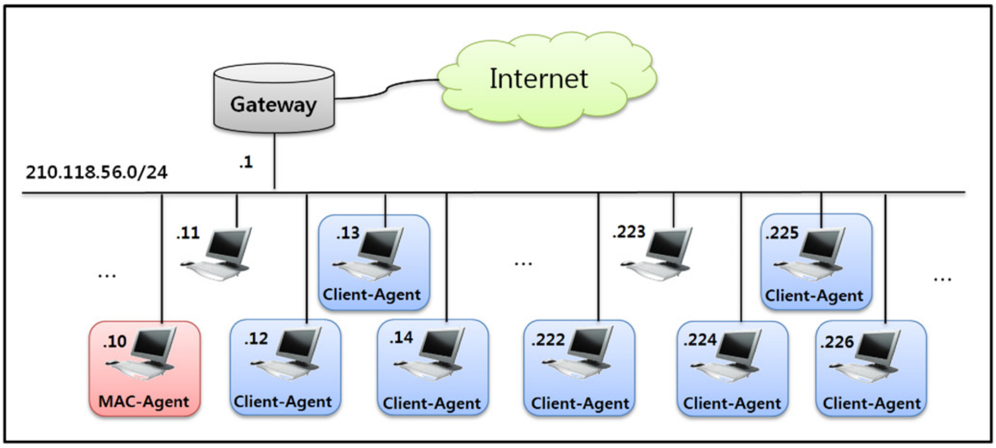


Fig. 2. System outline.

Table 1
Client-Agent actions on ARP packets.

Event	Action
Inbound ARP request	Block the ARP packet
Outbound ARP request	Block the ARP packet after informing the user-level application
Inbound ARP reply	Block the ARP packet
Outbound ARP reply	This event does not occur

To operate our system, a single MAC-Agent must be installed in the subnet, and the hosts in the same subnet have to install a Client-Agent, as shown in Fig. 2. Even though some hosts have not installed the Client-Agent, they can use the Internet because our system does not modify the network protocol. However, our system does not protect hosts that have not installed the Client-Agent from an ARP spoofing attack. Thus, users who want protection from ARP spoofing need to install the Client-Agent.

3.2. *Client-Agent*

The Client-Agent consists of a user-level application and device driver [13] and is developed under the Windows operating system. The user-level application stores the ARP table information sent from the MAC-Agent. If a host transmits the data to an IP address that is not listed on the host's ARP table, the application searches reliable address lists. If it is a reliable IP, the corresponding MAC address is renewed in the ARP table as a static type.

The device driver operates on a NIC. It analyzes the inbound and outbound packets and filters the ARP request and reply packets. Normally, the network library [14] is performed on the same layer of TCP/IP, and is thus able to analyze the packet but unable to filter it. In this system, we restrict the packets that can be modified, such as ARP replies and requests, as shown in Table 1, to protect against ARP spoofing.

An ARP request occurs when there is no IP address in the ARP table of the same subnet during packet transmission. Inbound ARP requests are filtered, whereas outbound ARP requests are notified to the application through interruptions and application searches for the corresponding address, as mentioned above. Inbound ARP replies are filtered to protect the ARP table, and outbound ARP replies do not occur because the inbound ARP request is cut off.

The Client-Agent operates as shown in Fig. 3. The details are as follows.

Intermediate driver part

① Extracts only the ARP packet by scanning the header part of the packet received from NIC.

② Saves the extracted data in a packet queue.

③ ④ ⑤ If interrupt occurs, the data in the packet queue are transmitted to the application.

User application part

⑥ Synchronizes the data from the driver by saving in a packet queue.

⑦ Analyzes the packet queue data in order and examines the data with ARP table information from the MAC-Agent. If there is no problem, data are registered on the real ARP table as static type.

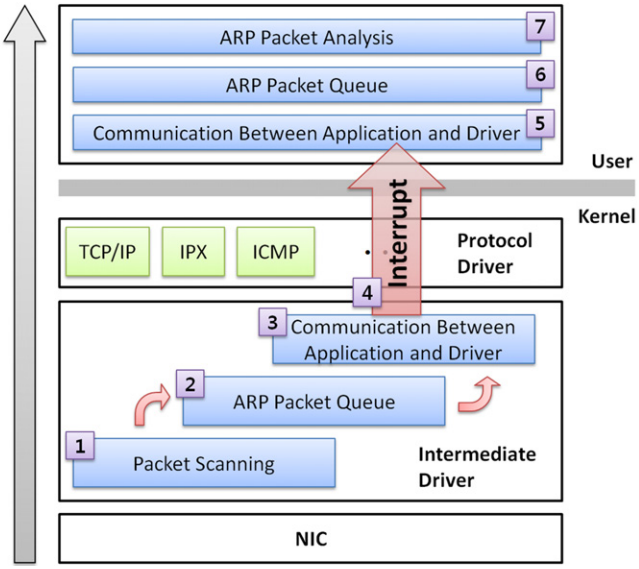


Fig. 3. Client-Agent operation.

3.3. MAC-Agent

While the Client-Agent protects the ARP table of every host, the gateway ARP table also needs to be protected because it can be modified by ARP replies without user certification, and hackers can "spoof" the data through the gateway. However, gateway ARP tables that are operated in Firmware cannot be fixed using static types, so the gateway ARP table must be constantly checked for modifications.

To check whether the ARP table is modified, the MAC-Agent broadcasts the reliable address developed by the Client-Agent through an ARP request. At this time, the MAC-Agent compares the MAC address transferred from the gateway and the known MAC address to check whether it has been modified.

The MAC-Agent checks whether the ARP table has been modified at random intervals, with a maximum period of 30 s. This implies that hackers can "spoof" the packet from outside for a maximum of 30 s. As Client-Agent users never reply to an ARP request, the only way that hackers can figure out a MAC address is through social technology. Thus, this maximum of 30 s only becomes a danger if both the IP and MAC address are exposed.

3.4. Data encryption

The biggest problem with ARP is that there is no authentication when exchanging a MAC address between nodes. All ARP packets can be exposed by a hacker, so our system does not use ARP. We exchange MAC addresses on the application layer instead of the data link layer in order to add the authentication process to the exchange of MAC information. Fig. 4 shows the exchange process of MAC information between the MAC-Agent and certain Client-Agents.

In our system, AES [15] and RSA [16] encryption are used to protect the protocol. First, when the Client-Agent connects to the MAC-Agent, the MAC-Agent distributes the RSA public key. The Client-Agent then generates an AES key, encrypts it using the public key, and sends the AES key to the MAC-Agent. After the AES key exchange, ARP table information can be safely sent and received using the AES key.

4. Experiments and evaluation

4.1. Experiments

We verified our system in two ways based on whether the hacker's target is a normal host or a gateway, because the defense mechanism is different in each case. For the experiment, we used a gateway server, 10 PCs as Client-Agents, one PC as a MAC-Agent, and one PC for the hacker. Windows XP and Windows 7 operating systems were used and ARP Spoof [17] was employed as a hacking tool.

Scenario 1: *target—normal host*

Fig. 5 shows the host's ARP table before being attacked by an ARP spoofing attack.

If the host PC is attacked by ARP Spoof under a normal network environment, the ARP table of the host is easily modified, as shown in Fig. 6. After the hacking, the host transmits data via the hacker to another host that has an IP address between .232 and .241. Thus, there can be data manipulation and sniffing by the hacker during this data transmission.

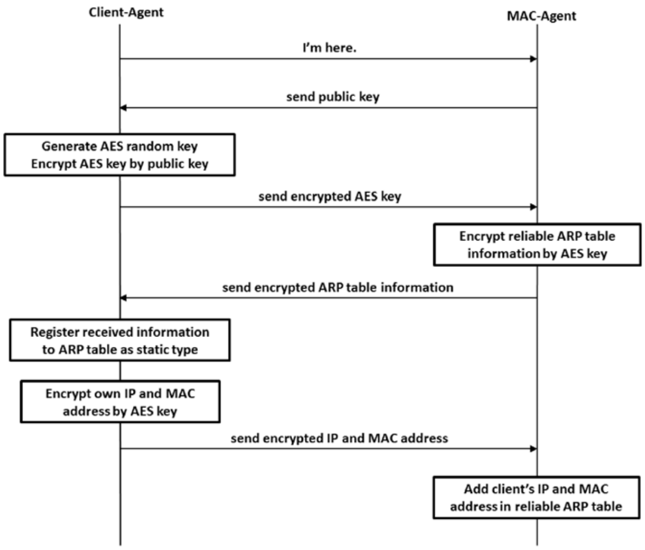
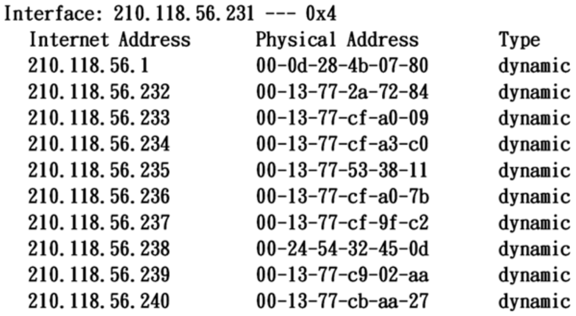
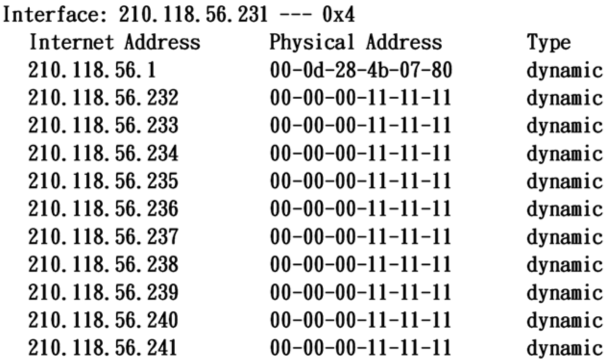

Client-Agent
MAC-Agent
I'm here.
send public key
Generate AES random key
Encrypt AES key by public key
send encrypted AES key
Encrypt reliable ARP table
information by AES key
send encrypted ARP table information
Register received information
to ARP table as static type
Encrypt own IP and MAC
address by AES key
send encrypted IP and MAC address
Add client's IP and MAC
address in reliable ARP table

Fig. 4. Sequence of encryption key distribution.



```
Interface: 210.118.56.231 --- 0x4
  Internet Address      Physical Address      Type
  210.118.56.1          00-0d-28-4b-07-80     dynamic
  210.118.56.232        00-13-77-2a-72-84     dynamic
  210.118.56.233        00-13-77-cf-a0-09     dynamic
  210.118.56.234        00-13-77-cf-a3-c0     dynamic
  210.118.56.235        00-13-77-53-38-11     dynamic
  210.118.56.236        00-13-77-cf-a0-7b     dynamic
  210.118.56.237        00-13-77-cf-9f-c2     dynamic
  210.118.56.238        00-24-54-32-45-0d     dynamic
  210.118.56.239        00-13-77-c9-02-aa     dynamic
  210.118.56.240        00-13-77-cb-aa-27     dynamic
```

Fig. 5. Normal ARP table.



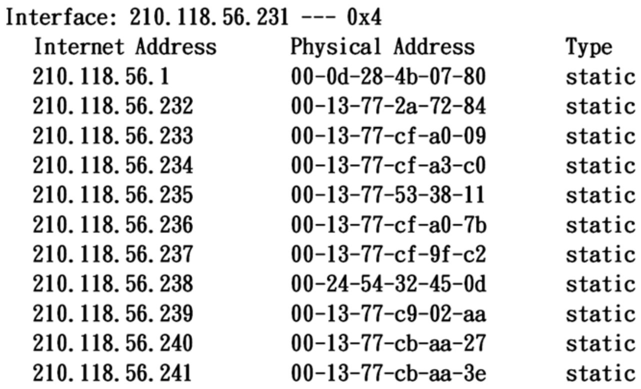
```
Interface: 210.118.56.231 --- 0x4
  Internet Address      Physical Address      Type
  210.118.56.1          00-0d-28-4b-07-80     dynamic
  210.118.56.232        00-00-00-11-11-11     dynamic
  210.118.56.233        00-00-00-11-11-11     dynamic
  210.118.56.234        00-00-00-11-11-11     dynamic
  210.118.56.235        00-00-00-11-11-11     dynamic
  210.118.56.236        00-00-00-11-11-11     dynamic
  210.118.56.237        00-00-00-11-11-11     dynamic
  210.118.56.238        00-00-00-11-11-11     dynamic
  210.118.56.239        00-00-00-11-11-11     dynamic
  210.118.56.240        00-00-00-11-11-11     dynamic
  210.118.56.241        00-00-00-11-11-11     dynamic
```

Fig. 6. Ruined ARP table.

In our system, the Client-Agent saves the received MAC address in the host's ARP table as a static type, as shown in Fig. 7. ARP request and reply packets are blocked by the device driver, and thus the ARP spoofing attack becomes ineffective.

Scenario 2: *target—gateway*

While every host is protected by the Client-Agent, the gateway server itself is vulnerable against ARP spoofing attacks. In this experiment, we verified that when the gateway server is exposed to an ARP spoofing attack, the MAC-Agent sends a message to all Client-Agents that the network is under attack.



```
Interface: 210.118.56.231 --- 0x4
  Internet Address      Physical Address      Type
  210.118.56.1          00-0d-28-4b-07-80     static
  210.118.56.232        00-13-77-2a-72-84     static
  210.118.56.233        00-13-77-cf-a0-09     static
  210.118.56.234        00-13-77-cf-a3-c0     static
  210.118.56.235        00-13-77-53-38-11     static
  210.118.56.236        00-13-77-cf-a0-7b     static
  210.118.56.237        00-13-77-cf-9f-c2     static
  210.118.56.238        00-24-54-32-45-0d     static
  210.118.56.239        00-13-77-c9-02-aa     static
  210.118.56.240        00-13-77-cb-aa-27     static
  210.118.56.241        00-13-77-cb-aa-3e     static
```

Fig. 7. Secure ARP table.

Table 2
Comparison between our system and prior defense mechanisms.

Defense mechanism	Supports DHCP	No additional equipment	Practicality
Static MAC entries	**X**	**O**	**X**
L3 switch & router	**O**	**X**	**O**
Encryption ARP packet (new ARP)	**O**	**X**	**X**
ARP spoofing attack monitoring tool	**O**	**X**	**O**
Our system	**O**	**O**	**O**

4.2. Evaluation

Practically, it would be difficult to fix the current protocol, because the ARP is already spread around the world. Therefore, the following four methods are normally used to protect against ARP spoofing. The first method is to fix the MAC address in the ARP table; the second method is to monitor whether ARP packets have been modified. Thirdly, hardware equipment can be used to fix the MAC address assigned to a port, and the final method is to code the ARP packet. However, these methods also have limitations. The first method must be performed manually, and is thus inadequate for an environment where users frequently change location or for large-scale network environments. The second method uses an ARP monitoring tool [18] and, as it requires manual work by administrators, is not suitable for large-scale networks. The third and fourth methods are usually used by corporations or facilities where security is essential. They purchase L3 switch equipment from CISCO, MS, or 3Com and receive NAC (Network Access Control) or NAP (Network Access Protocol). However, L3 switches are very expensive tools, so using them solely for blocking ARP spoofing seems like a waste of money. Table 2 shows that our proposed system is practical and superior to other prior defense mechanisms in several aspects.

5. Conclusion

Numerous methods have been developed to prevent ARP spoofing attacks. However, no single system is popular due to difficulties in their practical application to the current network or monetary issues.

In this paper, we examined the protection of users from ARP spoofing attacks. Our suggested method is based on the host environment and does not require protocol modification or any additional equipment. It requires a physically separated PC with a MAC-Agent, but it is a light application, and thus the suggested system can be widely used under the current network environment.

Acknowledgment

This research was supported by a National Research Foundation of Korea Grant funded by the Korean Government (2012000565).

References

[1] S. Garfinkel, Web Security and Commerce, O'Reilly & Associates, Cambridge, 1997.
[2] M. Han, D. Li, T. Jeong, Adaptive security model in real-time intrusion detection environment, Information-International Interdisciplinary Journal 14 (4) (2011) 1373–1384.
[3] H.U. Khan, Crime tracking e-security system: using wireless technology and database, Information-International Interdisciplinary Journal 15 (2) (2012) 679–688.
[4] J. Ryoo, E. Park, Internet security readiness: the influence of internet usage level and awareness on internet security readiness capital, skill, and actual uptake/use of infrastructure, Journal of Computing Science and Engineering 5 (1) (2011) 33–50.

[5] Gibson Research Corporation, ARP Cache Poisoning, 2005. http://www.grc.com/nat/arp.htm.
[6] http://www.ahnlab.com/kr/site/securitycenter/asec.
[7] S. Hong, M. Oh, S. Lee, S. Lee, Efficient technique for preventing ARP spoofing attacks using reliable ARP table, Journal of KIISE: Computing Practices and Letters 17 (1) (2011) 26–30.
[8] R. Wagner, Address resolution protocol spoofing and man in-the-middle attacks, 2001. http://rr.sans.org/threats/address.php.
[9] Y. Liu, K. Dong, L. Dong, B. Li, Research of the ARP spoofing principle and a defensive algorithm, WSEAS Transactions on Communications 7 (5) (2008) 516–520.
[10] G. Gouda, H. Chin-Tser, A secure address resolution protocol, Computer Networks 1 (41) (2003) 57–71.
[11] V. Ramachandran, S. Nandi, Detecting ARP spoofing: an active technique, Lecture Notes in Computer Science 3803 (2005) 239–250.
[12] D. Pansa, T. Chomsiri, Architecture and protocols for secure LAN by using a software-level certificate and cancellation of ARP protocol, in: Proceedings of International Conference on Convergence and Hybrid Information Technology, vol. 2, 2008, pp. 21–26.
[13] http://en.wikipedia.org/wiki/Network_Driver_Interface_Specification.
[14] W. Stevens, R. Wright, TCP/IP Illustrated (vol. 2): The Implementation, Addison-Wesley, Boston, MA, 1995.
[15] J. Daemen, V. Rijmen, The Design of Rijndael: AES—The Advanced Encryption Standard, Springer-Verlag, Berlin, Germany, 2002.
[16] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM 21 (2) (1978) 120–126.
[17] http://arpspoof.sourceforge.net.
[18] http://www.mynitor.com/2010/02/13/14-useful-arp-monitoring-tools.