Analysis on Manipulation of the MAC Address and Consequent Security Threats

Kyungroul Lee
R&BD Center for Security and Safety Industries
SCH University, 646 Eupnae, Shinchang
Asan, 336-745 Korea
carpedm@sch.ac.kr

Hyeungjun Yeuk
R&BD Center for Security and Safety Industries
SCH University, 646 Eupnae, Shinchang
Asan, 336-745 Korea
goodyug@sch.ac.kr

Kangbin Yim*
Dept. of Information Security Engineering
SCH University, 646 Eupnae, Shinchang
Asan, 336-745 Korea
yim@sch.ac.kr

Suhyun Kim
IoT Security Research Center
SCH University, 646 Eupnae, Shinchang
Asan, 336-745 Korea
kimsh@sch.ac.kr

ABSTRACT

In this paper, we introduce manipulation methods on MAC addresses and consequent security threats. The Ethernet MAC address is known unchangeable, and so is highly considered as platform-unique information. For this reason, various services are researched to use the MAC address. This kind of services are organized with the MAC address as platform ID or password, and a diverse range of security threats are caused when the MAC address is manipulated. Therefore, we research on manipulation methods for MAC addresses at different levels on a computing platform and highlight the security threats resulted from modification of the MAC address. In this paper, we introduce different methods causing a MACd of spo kinofing attack, which are unknown to be general approaches. This means it is difficult to detect the falsification and the result is crucial in most MAC address-based services.

CCS Concepts

•Security and privacy → Malicious design modifications;

Keywords

MAC address; Hardware Unique Information;
Security Threats; Countermeasure

*Corresponding author: +82-10-8958-9080,
Director of R&BD Center for Security and Safety Industries,
Member of IoT Security Research Center

MIST'16 October 28 2016, Vienna, Austria

DOI: http://dx.doi.org/10.1145/2995959.2995975

1. INTRODUCTION

The MAC address plays an important role as a unique identifier in communication networks. As originally designed, it has been utilized in order to provide the uniqueness of the endpoint in network communication, in which it should be only differentiable one within a reachable communicating segment. As additional applications, various services also use the MAC address for device authentication information or the encryption/decryption key because the MAC address is known to be immutable. These services that utilize the MAC address as a sensitive information are configured based on the misunderstood immutability of the MAC address. Therefore, verification on its safety against existing vulnerabilities is highly required.

Typical examples caused by these vulnerabilities include MITM (Man-In-The-Middle) and DoS (Denial of Service) attacks that have been attempted as a result of ARP (Address Resolution Protocol) spoofing. These attacks do not modify the real MAC address, and they modify the MAC addresses in the cache instead. The problem of ARP cache-based MAC modification can be easily resolved by tracking the cache changes. However, it is really difficult to detect modification of the MAC address if the attacks change the real MAC address fabricated in the hardware platform.

So, it is urgent to research for the possibility of hardware based MAC address modification. Because of this reason, we survey services utilizing MAC addresses as sensitive information and introduce how to extract and modify the MAC addresses. Finally, we highlight potential security threats that can be resulted from manipulating the MAC address.

2. RELATED WORKS

In this section, we describe MAC address-based services and extraction methods for the MAC address in order to show security threats resulting from the modification of a MAC address.

2.1 MAC address-based services

The MAC address in MAC address-based services is utilized as a mean of, for example, identifier for device authenti-

cation, tag for access control, seed for encryption/decryption key, or oracle for preventing ARP spoofing.

2.1.1 Device authentication and access control

The MAC address is used as an oracle to permit accesses in device authentication and access control services. After a user registers the MAC address of the device as the oracle for a service, the service is allowed to use only when the candidate MAC address of the userâĂŹs device is equal to the registered MAC address during the user requests a service. This feature is often provided in the network switches. The administrator collects the MAC addresses of the devices for legitimate users and registers them into the switch. When an approaching device tries to access into the network, the switch authorizes only the devices whose MAC address is already registered into the switch. The device whose MAC address was not registered cannot use the network.

A typical example of such a system is CiscoâĂŹs MAB (MAC Authentication Bypass)[4]. Similarly, some universities deployed the services registering the MAC addresses of devices to manage access control for the network within the universities, for instance, Yale University[10], Tufts University[8], University of Washington[9], and The University of Melbourne[7]. These services are like Cisco's, which enable only MAC address registered hosts to use the in-campus or campus-scale networks[11].

2.1.2 Encryption/decryption key

Utilization of an encryption/decryption key is classified by ensuring the confidentiality of media and protecting privacy information.

Ensuring confidentiality of media: If a user buys a sort of media, such as image, music, video, and file, the service provider will try to verify the consumer to ensure its confidentiality and has to apply a security feature to access the media only available to the authorized user. A previous study [2] proposes an encryption method for data using the MAC address as a key. This approach encrypts and decrypts data by XOR with the MAC address using a generic algorithm. In detail, the target-image is encrypted based on the MAC address, so if a device has the correct MAC address, the encrypted image is decrypted to the original image. Otherwise, if a device does not have one same to the registered MAC address, the encrypted image is decrypted to an unrecognizable image. Therefore, the proposed idea assures confidentiality and safe because the data is decrypted ordinarily only in the device that has the registered MAC address.

Privacy protection: Cookies stores important information related to the identification and privacy information. Nevertheless, an efficient method for protecting cookies has not been proposed to date, as such there is a risk of exposure of privacy and important information. In a previous study [12], the authors propose an encryption/decryption method based on cookies and the MAC address as the unique identifier using a key ring structure. This approach does not directly use the MAC address as the encryption/decryption key. Actually, the MAC address is utilized for protecting the encryption/decryption key, so cookies are able to decrypt only in the host whose MAC address is equal to the registered one. Therefore, this approach is safe due to its ability to protect privacy and identification-related important information.

2.1.3 Prevention of ARP spoofing

Prevention of ARP spoofing is classified by verification of the mapping relationship with IP and MAC address, IP address assignment using MAC address, and ARP cache deletion.

Verification of the mapping relationship between the IP address and MAC address: ARP spoofing is such that an attacker receives all translated packet from or to the victim by sending the ARP reply packet to the victim in order to change the MAC address stored in the cache. Through this, the attacker can attempt an MITM attack or DoS attack. The reasons behind this are that the ARP protocol does not require authentication and IP address and MAC address are not related each other. Therefore, if the IP address is associated with the MAC address, it is possible to prevent the ARP spoofing attack.

A previous study [3] verifies the mapping relationship between the IP address and MAC address by comparing the original IP address and MAC address to the ARP request and reply IP address and the MAC address after capturing the ARP packet. If the test results that the two addresses are the same, it means the attack did not occur successfully, otherwise the attack would succeed, as such it is possible to detect an ARP spoofing attack. Similarly, in [13], the proposed method collects the transferring packet using winpcap library, and then an ARP spoofing attack is detected by comparing the original IP address and MAC address with IP address and MAC address when ARP response packet is transmitted from collected packets.

IP address assignment using MAC address: The above method for verification of mapping relationship is not directly related to the IP address and MAC address; practically, it is an indirect method. In a previous report [1], the authors propose a prevention method for ARP spoofing by associating the IP address and MAC address directly. This approach generates seed by adding the MAC address, and the IP address is assigned by XOR operation with seed and MAC address. When the ARP packet is received, the receiver generates seed by XOR operation with received IP address and MAC address, and finally, value of sum is generated by adding the MAC address. Therefore, if the extracted value of the sum is equal to the seed, the ARP spoofing attack is not achieved. Otherwise, an ARP spoofing attack is attempted, and as such it is possible to prevent the attack.

ARP cache deletion: One of the biggest causes of ARP spoofing attack is due to the presence of an invalid IP address and MAC address pairs in the ARP cache. For this reason, this attack can fail to detect incorrect IP and MAC address pairs. Alternately, it is possible to prevent the attack by ensuring the use of the correct IP and MAC address pairs. One of the solutions, the deletion method, is thoroughly researched. This method deletes the ARP cache in order to clear the correct IP and MAC address pairs in the ARP cache [5], possible to clear using JAVA code. Therefore, when a user executes the command, malicious entries are deleted in ARP cache, as such allowing this method to prevent ARP spoofing attack.

Above all, services are provided based on the MAC address and it is possible to apply because the MAC address is known as permanently unchangeable information. Hence, if a MAC address is able to change, these services are exposed to security threat, and as such we research modification methods of MAC addresses and verify them in this

```
LONG Result = RegOpenKeyEx(HKEY_LOCAL_MACHINE,
    "System¶¶CurrentControlSet¶¶Control¶¶Class¶¶{4D36E972-E325-11CE-BFC1-08002bE10318}¶¶0001",
    NULL,
    KEY_READ,
    &hkResult);

if(Result == ERROR_SUCCESS)
{
    RegQueryValueEx(hkResult, "NetworkAddress", NULL, &dwType, (LPBYTE)Data, &dwSize);
}
```

Figure 1: Part of modification code using the registry

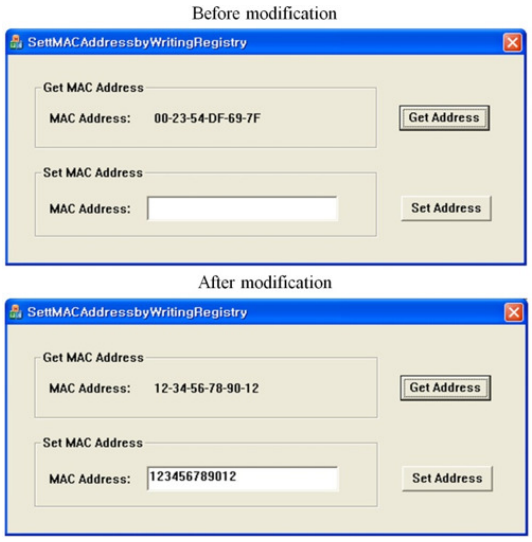

Figure 2: Modification result by using the registry

paper. In addition, we also highlight security threats to the above services when the MAC address is changed. Extraction of the MAC address can be carried out in several ways: reading the registry database, calling one of the dedicated APIs, polling registers in the I/O controller of the NIC, and communicating with EEPROM in the NIC. The operational environment for each implementation includes Microsoft Windows XP service pack 3, Intel I/O controller hub 7 (ICH7) family, and Realtek RTL8169/8110 family. Each code was programmed on Microsoft visual studio 2005 with Winddk 3790.1830[6].

2.2 Modification methods for platform-unique information (MAC address)

Modification of a MAC address can be carried out in several ways such as writing the registry database, hooking the dedicated APIs, trapping registers in the I/O controller of the NIC, and communicating with EEPROM in the NIC. Experimentation and implementation environment are the same as the extraction methods.

2.2.1 Writing the registry database

Modification method using writing is classified by writing the registry database of wanted MAC address directly and hooking function which reads the registry. Writing the registry database calls the RegSetValueEx function. Figure 1 shows a part of the implemented code and Figure 2 shows the result.

2.2.2 Hooking the dedicated APIs

Modification method using hooking is enabled by hooking dedicated functions. They are UuidCreate, UuidCreateSequential, NetWkstaTransportEnum, GetAdaptersInfo, GetIfTable and Netbios.

```
g_pOrgFunc = GetProcAddress(GetModuleHandle("Iphlpapi.dll"), "GetAdaptersInfo");

DWORD dwBufLen = sizeof(IP_ADAPTER_INFO);

DWORD dwStatus = GetAdaptersInfo(&AI, &dwBufLen);

if(dwStatus == ERROR_SUCCESS)
{
    InlineHook((int)NewGetAdaptersInfo, (DWORD)g_pOrgFunc, origbyte);
}


memcpy(pAdapterInfo, &AI, *pOutBufLen);

pAdapterInfo->Address[0] = 0x12;
pAdapterInfo->Address[1] = 0x34;
pAdapterInfo->Address[2] = 0x56;
pAdapterInfo->Address[3] = 0x78;
pAdapterInfo->Address[4] = 0x90;
pAdapterInfo->Address[5] = 0x12;

return ERROR_SUCCESS;
```

Figure 3: Part of modification code using hooking

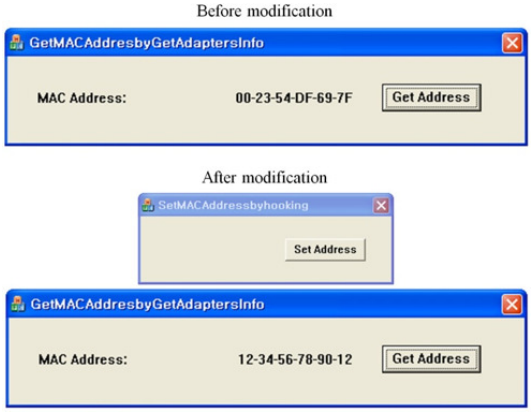

Figure 4: Modification result using hooking of GetAdaptersInfo function

In this paper, we chose GetAdaptersInfo function to perform hooking. Well known hooking methods include IAT, EAT, inline hooking, etc. and we chose to change the MAC address by inline hooking. Figure 3 shows part of the implemented code and Figure 4 shows the result of the implementation.

2.2.3 Trapping the I/O registers

The PC platform prepares extra controllers for I/O devices and the O/S extracts information from them through corresponding controllers. I/O devices are usually connected to the PCI bus, therefore, you can extract the MAC address by reading the MAC address-related register after obtaining the base address of the NIC. This means that this method has to access a specific memory or I/O address directly. If attackers want to change the MAC address in the register in the case of the Intel processor, they can place a debug trap on the dedicated memory or I/O address and pre-defined trap handler will be called if anyone accesses the register, then the attacker can replace the MAC address with one they want.

2.2.4 Communicating with EEPROM

One extraction method involves communication with EEPROM directly accessing the EEPROM inside the NIC to ex-

```
WRITE_PORT_UCHAR(((PUCHAR)PCI_BaseAddress+0x50), EECS_UP_EESK_UP_EEDI_DOWN_EEDO_DOWN );
delay = RtlConvertLongToLargeInteger(DELAY_BASE*TSKH );
KeDelayExecutionThread(KernelMode, FALSE, &delay);

WRITE_PORT_UCHAR(((PUCHAR)PCI_BaseAddress+0x50), EECS_UP_EESK_DOWN_EEDI_DOWN_EEDO_DOWN );
delay = RtlConvertLongToLargeInteger(DELAY_BASE*TSKL_TDIS );
KeDelayExecutionThread(KernelMode, FALSE, &delay);

WRITE_PORT_UCHAR(((PUCHAR)PCI_BaseAddress+0x50), EECS_UP_EESK_DOWN_EEDI_UP_EEDO_DOWN );
delay = RtlConvertLongToLargeInteger(DELAY_BASE*TDIS );
KeDelayExecutionThread(KernelMode, FALSE, &delay);

WRITE_PORT_UCHAR(((PUCHAR)PCI_BaseAddress+0x50), EECS_UP_EESK_UP_EEDI_UP_EEDO_DOWN );
delay = RtlConvertLongToLargeInteger(DELAY_BASE*TSKH );
KeDelayExecutionThread(KernelMode, FALSE, &delay);

WRITE_PORT_UCHAR(((PUCHAR)PCI_BaseAddress+0x50), EECS_UP_EESK_DOWN_EEDI_DOWN_EEDO_DOWN );
delay = RtlConvertLongToLargeInteger(DELAY_BASE*TSKL );
KeDelayExecutionThread(KernelMode, FALSE, &delay);
```

Figure 5: Part of modification code using communicating with EEPROM

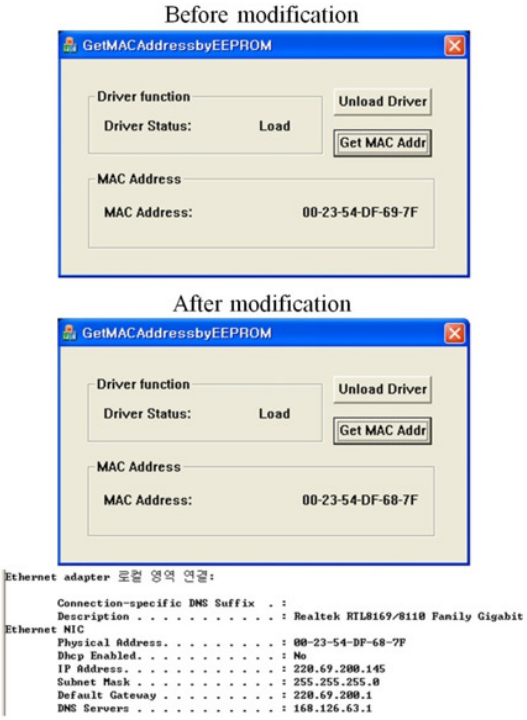

Figure 6: Modification result using communicating with EEPROM

tract the MAC address. Other methods explained are able to modify the shadow MAC address according to the practical implementations, but this method modifies the EEPROM itself, which is the source of the MAC address. Therefore, it causes permanent modification and so serious problems are able to be caused. Eventually, if the MAC address is changed by this method, the modified MAC address is extracted using all of the above extraction methods. In the case of selected NIC devices, it is possible to write into the EEPROM so an attacker can change the MAC address to whichever they want. Figure 5 shows a part of implemented code and Figure 6 shows the implemented result.

Therefore, in this paper, we verify falsification possibility of the MAC address according to above modification methods which are related extraction methods such as reading the registry database, calling the dedicated APIs, polling the I/O registers, and communicating with EEPROM. In conclusion, we verify that the MAC address is modified permanently by communicating with EEPROM so various MAC address-based services are neutralized by analyzed modification methods.

3. SECURITY THREATS CAUSED BY MODIFICATION OF MAC ADDRESS

Through the above researched results, we verified the MAC address is neither unique nor permanent information. Therefore, all MAC address-based services may cause problem to be exposed to security threats so we draw the threats by changing the MAC address in this paper.

3.1 Device Authentication and Access Control

Approaches for device authentication and access control register MAC address of host and requested service is authorized to only host which has registered MAC address. This approach assumes that MAC address in unchangeable so if MAC address is changed, these services are neutralized. In case of router, only host which has registered MAC address can access network. Nevertheless, if attacker changes his or her MAC address to registered MAC address, he or she can access network even though attacker is not authorized user. For this reason, when attacker preempts network, the user who has to get service normally is not provided requesting service so a problem arises that service is not functional.

3.2 Encryption/decryption key

Utilizing MAC address for way as encryption/decryption key causes problem by simply extracting the MAC address. Because MAC address is easy to obtain information so if attacker has cipher text and MAC address, he or she is able to decrypt it. For this reason, important information does not ensure confidentiality. Moreover, MAC address is also able to change; hence, unauthorized user can get service legitimately if the user changes his or her MAC address to userâĂŹs MAC address.

3.3 Prevention of ARP spoofing

In order to prevent ARP spoofing, there is an approach by verification of mapping relationship with IP address and MAC address. For this reason, if an attacker changes MAC address, incorrect mapping relationship changes correct mapping relationship so this approach does not prevent APR spoofing any more. Another approach for IP address assignment using MAC address also has a problem when MAC address is changed. Because attacker can assign correct IP address after that, the approach will not detect and prevent ARP spoofing attack any more. Other approach for ARP cache deletion also has vulnerability. The reason is that MAC address is changed permanently. Therefore, this approach does not prevent ARP spoofing attack because changed MAC address is not malicious entry so it must not delete in cache.

4. CONCLUSIONS

Commonly, the MAC address is known as unchangeable information so various MAC address-based services are proposed. Nevertheless, if the MAC address is changed, these services cause critical security threats. Therefore, in this paper, we surveyed MAC address-based services and analyzed extraction methods for the MAC address and their corresponding modification methods. As a result, O/S, I/O controller and EEPROM steps are able to change the MAC

address. Especially, if the source MAC address inside EEPROM is changed, high levels such as I/O controller level, O/S level, and application level extracts modified MAC address as well as the modification method changes MAC address permanently. Hence, this problem is serious security vulnerability. For this reason, we verified that the MAC address is not unchangeable information any more through experiment and all MAC address-based services are not provided normally by neutralization. In this respect, novel countermeasures are needed.

5. ACKNOWLEDGMENTS

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-R0992-15-1006) supervised by the IITP (Institute for Information & communications Technology Promotion).

6. REFERENCES

[1] M. Ahmed and Z. M. Hazza. A novel algorithm to prevent man in the middle attack in LAN environment. In *Proceedings of the 2010 Spring Simulation Multiconference, SpringSim 2010, Orlando, Florida, USA, April 11-15, 2010*, page 106, 2010.

[2] M. A. F. Al-Husainy. MAC address as a key for data encryption. *CoRR*, abs/1311.3821, 2013.

[3] M. Ataullah and N. Chauhan. An efficient and secure solution for the problems of arp cache poisoning attacks. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 6(8):989 – 996, 2012.

[4] Cisco. *MAC Authentication Bypass Deployment Guide*. Cisco Press, November 2011.

[5] F. Fayyaz and H. Rasheed. Using JPCAP to prevent man-in-the-middle attacks in a local area network environment. *IEEE potentials*, 31(4):35–37, 2012.

[6] K. Lee, K. Lee, J. Byun, S. Lee, H. Ahn, and K. Yim. Extraction of platform-unique information as an identifier. *JoWUA*, 3(4):85–99, 2012.

[7] U. of Melbourne. MAC address registration. May 2014.

[8] U. of Tufts. Network device registration. May 2014.

[9] U. of Washington. Register for CSE network access (wired or wireless). May 2014.

[10] U. of Yale. Netreg (device registration). May 2014.

[11] Y. Watanabe, M. Otani, H. Eto, K. Watanabe, and S. Tadaki. A MAC address based authentication system applicable to campus-scale network. In *15th Asia-Pacific Network Operations and Management Symposium, APNOMS 2013, Hiroshima, Japan, September 25-27, 2013*, pages 1–3, 2013.

[12] H. Wu, W. Chen, and Z. Ren. Securing coockies with a mac address encrypted key ring. In *2nd International Conference on Network Security Wireless Communications and Trusted Computing (NSWCTC)*, pages 62–65, 2010.

[13] W. Xing, Y. Zhao, and T. Li. Research on the defense against arp spoofing attacks based on winpcap. In *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on*, volume 1, pages 762–765. IEEE, 2010.