SU-WiFi's Security Hole from Fake WiFi and Detection Technique Based on DHCP Log

Sirak Kaewjamnong*
Chalong Viriyatham*
Pongsakorn Choksomngam*

Abstract

Nowadays, many organizations utilize wireless LAN to provide access to the Internet and enable flexible workplaces. Users can connect their smartphones or laptop computers to the organization's network. Over the last few years, the growing of wireless system presents risks of wireless security. One of the most challenging risks is from the fake access points (fake AP). Undetected fake AP can harm the network in many ways. For example, unauthorized user can use fake AP to gain permission from an authorized user and gets access to the network. Silpakorn University provides more than 250 WiFi hotspots across three campuses to serve a number of users. This paper focuses on the security hole in Silpakorn WiFi (SU-WiFi) and authorization system from fake WiFis, and presents the implementation of the client side's detection technique that combines the "white list" scheme and the use of DHCP log from the university DHCP server to detect fake APs from the client side.

Keywords: fake access point, rogue access point, AP, fake AP detection, rogue AP detection

Introduction

Currently, wireless network has become an essential requirement for everyone to access the Internet because it is convenient to connect and easy to be installed. It is scalable and cost effective as well. An organization or a company just provides a number of access points (AP) in working area for clients. As wireless networks are being popular, unauthorized access points become a security issue. An unauthorized AP is an AP that is set up into the network without authorization from the network authority. Unauthorized APs may be divided into two types, for the reason of surfing the Internet without permission, and sniffing packets from other users for negative proposes (Kao et al. 2014). It can lead to many security threats such as data interception, misbehaving clients, or wireless phishing (Phifer, 2010). An unauthorized AP can be called the fake AP or the rogue AP (Kao et al. 2014).

The fake APs that use their original MAC address can be detected by the "white list" technique (Kao et al. 2011). However, attacker can easily break the "white list" scheme by changing MAC address of that AP. A number of tool or program for forging MAC address can be

* ภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์ ม.ศิลปากร
Dept. of Computing Fac. of Science Silpakorn University email: Kaewjamnong_s@silpakorn.edu

downloaded from the Internet. A fake AP with fully forging SSID and MAC address of the legitimate AP is very hard to detect from both network administrators and legitimate users. The attacker can lure the clients to connect to the fake AP and launches several types of attack on the connecting clients, or the network system. The attacker can use passive attacks which the attack does not affect normal behavior of the network such as sniffing user packets. The passive attacks may lead to the active attacks such as stealing the client's permission to launch further attacks as the legitimate client behalf.

The legitimate users connect to the fake APs because they do not know which AP is fake or legitimate. They risk themselves and the entire network because of lacking information. Our work is to study the security issue from fake APs in Silpakorn WiFi (SU-WiFi) system, and proposes technique that can be used to detect fake APs from the client side.

The rest of this paper is organized as follows. Literature review is given in section 2. SU-WiFi and its security issues are stated in section 3. Section 4 presents our approach to detect fake APs. Then, experimental results are discussed in section 5. In the last section, conclusion is presented.

Literature Reviews

There are many of fake AP detection techniques. Traditional approach uses the concept of the "white list" by checking MAC address of the AP from the trusted list. This approach can easily be overcame. Some AP detections rely on network enumeration tools such as NetStumbler (NetStumbler, 2016) running on a laptop carried by an IT personal to detect rogue APs. This approach is time-consuming and unreliable.

There are a number of tools available for fake APs detection e.g. Mojo wireless manager (Mojo, 2016), Airtights WIPS (Airtights WIPS, 2016), Air Defense (Air Defense, 2016) provide a complete software and hardware system for enterprise networks. They use a combination of radio frequency sensors and an IDS/IPS server to monitor network events. However, they are not an economic option for networks with limited budget.

Ma et al. (Ma, Teymorian, and Cheng, 2007) proposed techniques to detect rogue APs by providing rogue AP detection engine, preemption engine and probing engine as small plugins. This solution may appropriate to the network administrators to monitor but it does not work at the client side.

Several researches used round-trip time to determine that the given AP is legitimate or not (Kao et al. 2014) (Han et al. 2011). In this approach, the rogue AP is detected because it relays traffic from the actual AP and relaying traffic needs one more hop to reach the destination. However, there are many reasons that can cause delay such as the interference or high traffic-loaded environment. Thus, this scheme is not accurate.

Arackaparambil (Arackaparambil et al., 2010) presented a clock skew approach which extracts Timing Synchronization Function (TFS) timestamp from beacon frame and compares the beacon frame timestamp that is generated by the AP with an arrival time of the frame at the client device. This technique is not robust due to the variation of the transmission delay.

The strength of signal is also be used to detect fake APs. Kim (Kim et al., 2012) proposed the client-side detection by using the deviations of the two AP's received signal strength. However, the authors assume that the fake APs relays traffic to the legitimate AP, which is not always the case.

Yan (Yang, Song and Gu, 2012) proposed user side evil twin detection technique. This technique presents two algorithm, Training Mean Matching (TMM) and Hop Differentiating Technique (HDT) by using Inter-packet Arrival Time (IAT) to detect Evil Twin AP. However, this scheme cannot work for all kinds of man-in-the-middle attack.

Alotaibi (Alotaibi and Elleothy, 2015) proposed the technique that takes advantage of the characteristics of physical layer field as a fingerprint to identify and detected rogue APs. The authors claimed that the detection accomplished 100 percent accuracy to determine rogue APs in a lightly traffic environment.

Some researches require the modification of 802.11 protocols and standards. Cross and Takahashi introduced a "Secure Open Wireless Access" protocol (Cross and Takahashi, 2011). This protocol uses the SSL protocol to distribute certificates that certified by a trusted CA. Bauer (Bauer, Gonzales and McCoy, 2008) proposed an authentication method using the Extensible Authentication Protocol (EAP) to verify the trusted AP. However, techniques that require protocol modification may not be possible to deploy in the real world.

Security Issue in SU-WiFi from Fake APs

Silpakorn WiFi network provides more than 250 hot-spots in Sanamchantra campus. The role to detect fake APs is currently relied on network administrators. In the authorization process, ordinary users have no role to prevent themselves from fake APs as shown in figure 1. To access SU-WiFi, a user connects his or her device to any AP which uses SU-WiFi SSID. His machine obtains its IP address from the university DHCP server. After that, the user gets log-in notification screen or uses web browser to open authentication page. User name and password will be verified via secure channel. After that, authorized users can use their log-in session until it expires at 6.00 pm., their log-in session can also be extended.

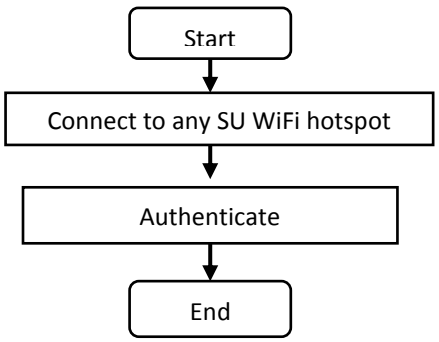

Figure 1 Connection processes

Veridian E-Journal, Science and Technology Silpakorn University
Volume 3 Number 5 September– October2016 (ISSN 2408 – 1248)

SU-WiFi requires authorization because of two main reasons. Firstly, it prevents the network to be used from unauthorized users. Secondly, by the Computer-Related Crime Act B.E. 2550, network administrators have to keep activity logs of all users for at least 90 days. Thus, all computing devices e.g. desktop computers, laptop computers or smartphones need to be verified user permission before accessing to the Internet. There is no limitation in term of the number of devices per user nor the number of user per each device's MAC address. Users are not required to register their devices or MAC address to the system.

Moreover, even though there are a number of APs in Silpakorn network, they are not covering all areas such as in different floors in the same building, or in different rooms in the same floor. The university has also provided a number of wire connectors in classrooms and office workplaces across the campus. Thus, local network administrators can add their own AP to the network and they are free to use any SSID they prefer including SU-WiFi SSID. Just plug an AP to a wire connection, local network administrators can use that AP as the campus hotspot and extend the university's wireless infrastructure. They can also connect their APs to any existing APs without wire connection as well. The university does not have the strict policy to limit connection from private APs yet because users who connect to that unauthorized AP, eventually have to authenticate themselves via the university authentication system. However, this presents a serious security hole in the network.

The attacker can work as the university hotspot by installing a fake AP into the campus network, and luring any victim to connect to that AP. When the victim connects to that fake AP, the victim will receive authentication page from the actual authentication system that forwarded by the fake AP. After that, the victim authenticates himself. However, the system matches MAC address of the attacker's machine to the victim's privilege rather than the actual MAC address of the victim's machine. Then, both the victim and the attacker can access to the Internet. The attacker uses the victim permission, and the victim gets forwarded traffic from the attacker. Finally, the attacker can use man-in-the-middle attack for eavesdropping messages of the victims, or use the victim permission to do anything he wants without worrying to be identified.

Even though the authentication page that the user got from the university authentication system also shows the assigned IP address of the machine as an example in figure 2, "IP of your machine is 172.27.225.159". However, only a few users will check that the assigned IP from the authentication page is the same as the machine's current IP address before logging in. Thus, user is easy to be lured. In the eavesdropping attack, the attacker receives data stream of the victim before forwarding to the actual destinations, and sensitive information can be leaked. The attacker can record and analyze data packets. He can also perform manipulation attack by changing and retransmitting data to and from the victim's machine. In addition, the attacker can use phishing technique by providing fake university log-in page and steal user name/password of the victim. Moreover, the attacker uses the victim permission to access the network, all activity logs will be recorded as the victim transactions.

Figure 2 SU Log-in Page

The attacker can install the fake AP in several ways. He can use device such as a laptop computer that has both wire and wireless interfaces. By plugging the wire interface to the campus provided connector, attacker can use the wireless interface as the hotspot. There are several areas in the university that allow users to plug their machines to the wire connectors and get IP address from DHCP server. The attacker can also use only wireless interface to connect to the actual hotspot and work as the hotspot at the same time. There are a number of tools, or operating system's built in commands that can provide service for this.

To simulate the attack, the authors use an Ubuntu laptop computer with a wire and a wireless interface as shown in figure 3. The wire interface is plugged into the university wire network. We do not authorize ourselves to the campus log-in system and can only access to the university's internal network. The wireless interface is then used as the AP and broadcast itself using SU-WiFi SSID. The victim connects to the fake AP without noticing he is connecting to the fake AP. Then, the victim obtains the university log-in page from the university authentication system which is forwarded by the fake AP. After authentication process, both victim and the attacker can access to the Internet as the victim privilege. The attacker can use the victim session until it expires even though the victim has left the fake hotspot long before.

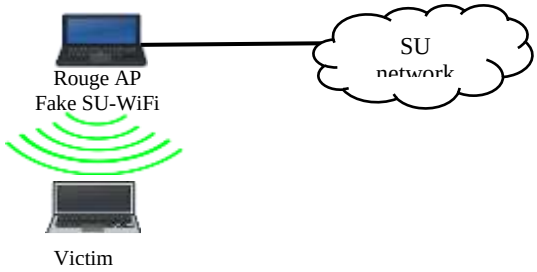

Figure 3 Fake AP installation with wire and wireless interface

We also simulate the attack without wire connection as shown in figure 4. In this simulation, We use an Ubuntu laptop with two wireless interfaces. One is used to connect to the actual SU-WiFi hotspot and the other works as the fake AP. It works in the same way as the previous simulation with wire connection. Both simulations do not require any external tools or programs. They only used internal commands that come with Ubuntu desktop operating system version 14.04.

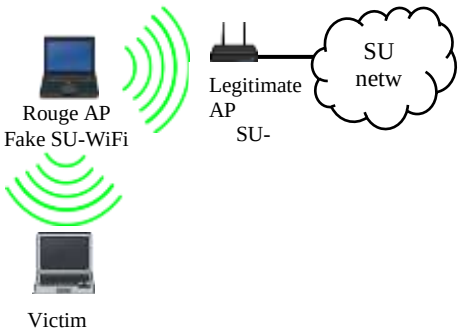

Figure 4 Fake AP installation with only wireless interfaces

Our Solution and Implementation

In this section, we present our solution to detect fake APs from the client side. We implemented an Android application for verifying SU-WiFi. The clients have to install this application on their devices. Our implementation is the combination of the “white list” technique and DHCP log information. Even though the “white list” scheme is easy to break, it can be used to detect fake APs from amateur attackers. For the DHCP log, almost all devices that connect to the SU-WiFi obtain its IP address from the university DHCP server and their requests must be kept in DHCP log file. There is an exception in a few branches e.g. the Department of Computing that has its own DHCP server. So, it can assume that almost all devices will get IP address from the central DHCP servers. In contrast, devices those connect to the fake AP getting their IP address from the fake AP rather than from the university DHCP server. Thus, its DHCP request will not be presented in the DHCP log because only the legitimate APs will forward DHCP request from the client to DHCP server. We can use this to identify what type of AP the device is connecting to.

In our implementation, there are two types of server that the client will connect to. The central DHCP server that provide IP address for the client, and the verifying server. Both servers can be on the same machine or separate machines but have permission to access the DHCP log. We divided the checking program into two sub-programs that can work individually. The first sub-program uses the “white list” policy. However, the first subprogram is an option, and can be omitted by the client.

In the implementation of the first sub-program, the client uses the existing “white list” to verify MAC address of the candidate AP. Because the “white list” is rarely changed, it can be added as a part of the program that the client can use to check before connecting to the target AP. However, if the client wants to update the list before connect, the client can

also request for the last up-to-date “white list” from the verifying server. The clients can use 3G/4G or any external connection (if present) on their device to request the last up-to-date “white list” from the verifying server. In addition, the clients may even connect to the unidentified AP and request the last “white list” from the verifying server because Silpakorn network allows local connections within the network without authentication. So, the client can connect to any local server including the verifying server. The verifying server only uses secure connection to reply any request. Thus, the client can trust that it got the last “white list” from the actual verifying server. With this scheme, the client can connect to the AP that only has MAC address in the “white list”, and then proceeds the next stage to verify the AP. However, after connecting to the AP, the client will get the authentication session from the university authentication system. We recommend the client to wait for verifying in the next stage before authenticating. Figure 5 shows the flowchart of the first sub-program.

.

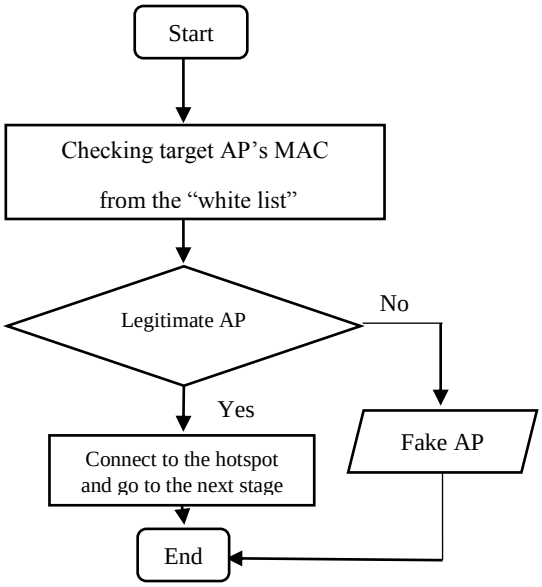

Figure 5 Flowchart of the “white list” scheme sub-program

Even though the connected AP uses MAC address as in the “white list”, it is possible that MAC address is fake. The second sub-program has an intension to verify that the connecting AP is the true AP. After the client has connected to the AP, the client will automatically send DHCP request to the system through that AP. By then, the central DHCP server sends DHCP reply and this transaction is kept in log file as shown in figure 6 (a). This is a normal process for every client to obtain IP address.

In case of the fake APs, the attacker has to work at the middle of the connection as proxy by getting and forwarding client’s packets. Thus, the client will get IP address from the fake AP rather than from the central DHCP server. In addition, the IP address that the client got may not be in the same subnetwork as in normal operation. However, the client can connect to the verifying server even though it is connecting to the fake AP. The client then connects to the verifying server via secure connection that no one can intervene. Then the client sends its

MAC address and assigned IP address to the verifying server. The verifying server checks from DHCP log that MAC and IP address are matched and presented in the DHCP log, and send the result back to the client as shown in figure 6(b). If the client connects to the legitimate AP, its MAC and IP address must be in the DHCP log and matched. Then, the client can log-in the authentication process to access the Internet.

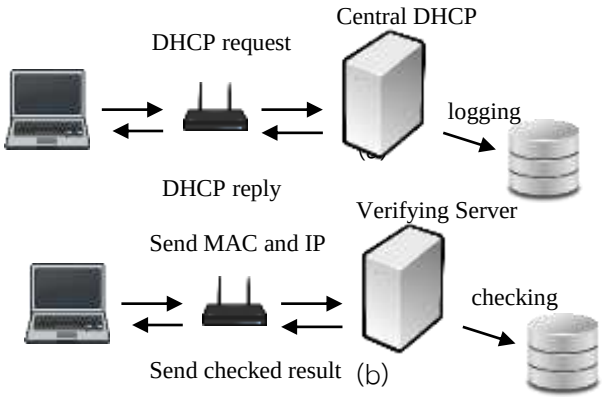

Figure 6 DHCP log checking architecture

Experimental Results

The experimental evaluations focus on seeing how effective our solution is in four different locations, Department of Computing, Faculty of Science, Faculty of Pharmacy, and Silpakorn Central Library, and doing the experiments to measure and analyze average time to verify. In the experiments, we simulated two types of fake AP. The first type we used a laptop with a wire interface connecting to the wire network and a wireless interface as AP. The second type of fake AP, we used the laptop with two wireless interfaces. One interface connected to the actual SU-WiFi and the second interface worked as the fake AP. We simulated attacker machine using a Dell laptop computer with Intel core i5 processor and Ubuntu 14.04 Desktop operating system. This machine has one wire and one wireless built-in interface. We also used USB wireless adapter for the test with only wireless interfaces.

We split our experiment into two different tests, the “white list” scheme, and the DHCP check. Table 1 presents experimental results of the “white list” scheme. As you can see from table 1, our solution can detect fake APs because we did not forge MAC address of the fake AP. However, the results also show false positive of the legitimate AP in the Department of Computing. It classes the legitimate AP as the fake AP. This results from the reason that our “white list” is not up-to-date. The “white list” policy requires co-operation from both central administrator and local administrators. If the local administrator adds a new AP without informing the central administrator, MAC address of the new AP will not be in the “white list”. The new AP was installed by the Department of Computing administrator, but has not been reported to the central administrators yet. This causes false positive in our experiment. In addition, in the tests from Faculty of Science and Central Library, all devices connecting to wire network have to have assigned static IP and have no DHCP server. Thus, without addressing information, we cannot plug our machine to the wire network in both areas.

Table 1 Results from the “White list” scheme

	Fake AP		Legitimate AP
	Wire&wireless	Only wireless	
Department Of Computing	Detect	Detect	Detect as fake
Faculty of Science	N/A	Detect	Pass
Faculty of Pharmacy	Detect	Detect	Pass
Central Library	N/A	Detect	Pass

In the second experiment, we measured time usage for verifying APs from DHCP log. We compared verifying time in three cases, fake AP with wire and wireless interface, fake AP with only wire interfaces, and time to verify the legitimate AP. Table 2 shows the experimental results. The results use the average time from at least 5 tests in each condition. There is not applicable result from the test with wire and wireless interface in Faculty of Science and Central Library because both places use static IP address for wire connections.

Table 2 Results from the DHCP log scheme

	Fake AP				Legitimate AP	
	Wire&wireless		Only wireless			
	Time (ms)	Verify result	Time (ms)	Verify result	Time (ms)	Verify result
Department of Computing	59.3	✓	32.0	✓	33.0	✘
Faculty of Science	N/A	N/A	216.3	✓	46.3	✓
Faculty of Pharmacy	40.2	✓	58.0	✓	295.4	✓
Central Library	N/A	N/A	114.0	✓	49.3	✓

The results show that our implementation can detect fake APs in all testing areas. It can also verify the legitimate AP in three out of four testing areas. However, it cannot verify the legitimate AP in the Department of Computing and classes this legitimate AP as fake. This dues to that the Department of Computing has its own DHCP server and its DHCP log is not shared.

It can be seen from the experimental results that the average times are quit fluctuating. Time to verify fake AP with wire connection is about 50 ms whereas the average time to verify fake AP with only wireless interfaces is at around 32 to 216 ms. It is the same trend when we verified legitimate APs at around 33 to 295 ms. This may result from that

wireless network is shared medium, and a number of users may use the same AP at the same time.

To avoid false positive results, our solution requires good management and co-operation between the university network administrators and local network administrators. Important information such as list of MAC address of the legitimate devices or log files should be reported or shared. In addition, strict policy such as static IP for wire network can reduce the risk from amateur attackers. Ultimately, the authentication process should be fixed to avoid any possibility of the authentication hijacking.

Conclusion

In this paper we present security issue in SU-WiFi system due to fake APs. Currently the role to detect fake APs is based on only the university administration. The clients should protect themselves as well. We have proposed the client side solution to detect fake APs using the “white list” scheme and information from DHCP logs to verify that user is connecting to the legitimate AP. Even though our solution cannot detect fake APs in every case e.g. the case that the DHCP log is not shared or MAC address of the legitimate AP is not in the list. However, it can verify Fake APs in areas that managed by the same administration. In addition, our solution is simple and practical. In the future, we will implement the detection system with additional techniques for more accurate detection.

REFERENCES

K.F. Kao, W. C. Chen, J. C. Chang and H. T. Chu. (2014). “An accurate fake access point detection method based on deviation of beacon time interval.” **IEEE Eighth International Conference on Software Security and Reliability-Companion (SERE-C)**, San Francisco, CA, USA.

L.Phifer. (2010). "Top ten Wi-Fi security threats." **ESecurity Planet**, March 8, 2010, http://www.esecurityplanet.com/views/article.php/3869221/Top-Ten-WiFi-Security-Threats.htm, Retrieved 10/3/2016

K.F. Kao, T.H. Yeo, W.S. Yong, and H.H. Chen. (2011) "A location-aware rogue AP detection system based on wireless packet sniffing of sensor Aps." **2011 ACM Symposium on Applied Computing (SAC '11)** , pp.32-36.

Netstumbler, http://www.netstumbler.com/, Retrieved 13/3/2016.

Mojo, (2016), http://www.mojonetworks.com/products/mojo-features, Retrieved 13/3/2016.

Airtights WIPS, (2016), http://www.bluecubesecurity.com/airtight-wips/, Retrieved 14/3/2016.

Air Defense, (2016), https://www.zebra.com/us/en/products/software/wlan-systems/wlan-management-and-security-software/airdefense-wids-wips.html, Retrieved 20/3/2016.

L.Ma, A.Y. Teymorian, and X.Cheng, (2007), "RAP: protecting commodity Wi-Fi networks from rogue access points." **The Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness & Workshops Article No. 21**, NY, USA.

H. Han, B. Sheng, C. Tan, Q. Li and S. Lu, (2011), "A timeing-based scheme for rogue AP detection." **Parallel and Distributed Systems, IEEE transaction vol.22 no.11**, pp. 1912-1925.

C. Arackaparambil, S. Bratus, A. Shubina, and D, Kotz, (2010) "On the reliability of wireless fingerprinting using clock skews." **Third ACM Conference on Wireless Network Security (WiSec'10)**.

T.Kim, H.Park, H. Jung, and H.Lee, (2012), "Online detection of fake access points using received signal strengths," **75th IEEE Vehicular Technology Conference (VTC Spring 2012)**.

C.Yang, M.Song, and G.Gu, (2012), "Active user-side evil twin access point detection using statistical techniques." **IEEE Transactions on Information Forensics and Security archive Volume 7 Issue 5**, IEEE Press Piscataway, NJ, USA, pp 1638-1651.

B. Alotaibi and K. Elleothy, (2015), "A passive Fingerprint Technique to Detect Fake Access Points." **IEEE Wireless Telecommunications Symposium**, New York, April 15-17, 2015.

T. Cross and T. Takahashi, (2011), "Secure open wireless access." **Black Hat USA**.

K.Bauer, H. Gonzales, and D. McCoy, (2008), "Mitigating Evil Twin Attacks in 802.11." **1st IEEE International Workshop on Information and Data Assurance (WIDA 2008)**, Austin, TX, USA.