Attack on WPA-PEAP

Neil Gentleman, Insoo Kwon, William Wong, Keith Kam
Electrical and Computer Engineering, University of British Columbia

Abstract – **This report provides an analysis of the vulnerabilities a wireless network system protected by WPA-PEAP. As the part of our analysis, our group attempted a variety of attacks on the UBC-Secure network which uses WPA-PEAP as its securing method. Our attacks include de-authentication of the UBC-Secure network's clients and configuring a rogue access point and radius server to recover the clients' username and password. The vulnerabilities discovered are demonstrated only for illustration purposes. However, these vulnerabilities can facilitate attackers to perform further actions and the possible solutions are examined in the report.**

Index Terms – **WPA, PEAP, security analysis, wireless network, computer network security**

I. Introduction

The wireless internet network at the University of British Columbia is in constant use, for academic research and personal communication over the internet. A reported 72% of students use it at least once a week [1].While wireless networks provide convenient access to the internet and are easy to use, problems associated with their security have been raised. Although UBC IT Services has created a secure option for students using WPA-PEAP, more diligence is required. Even without considering the communication that passes over the network, the information that users provide to login to the network is valuable. If stolen, usernames and passwords, in our case Campus Wide Login information, is the most important asset because CWL allows access to numerous systems such as Student Information System, The Library, WebCT, Blackboard Vista, UBC Wireless, UBC VPN, UBC Interchange, and more. This paper will evaluate the level of security provided by the use of WPA-PEAP. The evaluation encompasses the analysis and the methods of exploiting the vulnerabilities of WPA-PEAP used in UBC-Secure network system. Main exploits include de-authenticate clients on UBC-Secure network, and simulating UBC-Secure network working as a rogue router that snatches clients' CWL accounts and their passwords. Our project goal is to show the public that a so called "secure" network is not so secure if there are vulnerabilities left open within the system. In the next section, the infrastructure any given enterprise wireless network will be explained. Section III will explain our analysis of the system design as well as the core principles of computer security. Vulnerability of the system and the details of our attack will be highlighted in section IV. Lastly our report will wrap up with section V and VI, which are the results to our user survey and our proposed countermeasures respectively.

II. Enterprise Wireless Network Infrastructure

The most common securing method of an enterprise wireless network is WPA-PEAP. UBC's secure network uses this method to authenticate students, instructors and faculty members, and will be the target of our analysis. To provide wireless network coverage over an extended area, many overlapping access points (APs) with a common SSID are deployed. A client can communicate with any of these APs, each of which will delegate the authentication process to a Remote Authentication Dial In User Service (RADIUS) server; a central repository of user information.

III. Design Flaws

A. Principles of Designing Secure Systems

We analysed the following design principles for three most popular operating systems on the market today.

Fail-Safe Defaults:
Microsoft Windows, by default, rejects any certificate that is Not Verified for wireless authentication. OS X warns the user that the certificate is unsigned, but will still allow the user to continue if he/she so desired. Linux allows any certificates to be passed regardless of signed or unsigned.

Psychological Acceptability:
Microsoft Windows applies this design principle by hiding the security mechanism from the user. While on the other hand, OS X and Linux on the other hand does not hide the security mechanism from the user and the user is allowed to proceed if he/she so chose to.

Question Assumptions:
Linux operating system needs to question the assumptions that users are aware of the system they are connecting to –

there are no prompts for unsigned certificates. Slightly different, OS X informs the user if the certificate is unsigned and will then ask the user if they want to process or not. On the contrary, Windows doesn't question any assumptions and will automatically reject any unsigned certificates.

B. Confidentiality, Integrity, and Availability

The main focus of computer security policies is on three main goals, confidentiality, integrity and the availability of digital information [10]. We have discovered from this attack that all three core goals can be denied and violated.

If our attack is successful, we can attain the user's username and password and use that to attain their private information. This is in strict violation of the confidentiality goal, as the attacker has access to any digital information stored on that specific account. The breach of confidentiality goes hand in hand with the break of integrity, because the information being relayed from the user to the access point will be intercepted by our attack as a "man in the middle". Lastly, the concept of the data availability is broken due to our secondary attack of de-authorizing the user on the access point and forcing them to our rouge access point. In other words we are denying the users service to the non-rouge access point.

In the following section, we will discuss the details our findings regarding this attack on the three different operating systems we used which include, Windows, Mac OS X and Linux.

IV. Vulnerability

A. Authentication

WPA allows network administrators to choose from several authentication schemes (by way of Extensible Authentication Protocol, or EAP), but all fall into two simple categories. EAP-TLS, the most secure option, requires client certificates for authentication, and so is less acceptable to users. The more agreeable group uses password-based authentication. Many password-based schemes, like LEAP [3] are vulnerable if an attacker is in position to eavesdrop. PEAP, as used by ubcsecure, avoids this by wrapping the authentication handshake inside a TLS tunnel. Because of this, we cannot merely eavesdrop, but must fully impersonate an AP to control the endpoint of the tunnel.

In this first, outer stage, the RADIUS server identifies itself to the client using an X.509 certificate. There are two options for the certificate we will present: self-signed, or "genuine". A "genuine" certificate will be signed by a third-party Certificate Authority (CA), albeit not under the name of the organization that we are attempting to impersonate. Different operating systems respond differently to each of these options; the user may need to accept a self-signed certificate, or may refuse to connect entirely. Note that self-signed certificates were used in the attack; "genuine" certificates were only simulated by importing a new trusted root CA on our test clients. A real attack would need to purchase such a certificate, from Thawte or Verisign[1] [4].

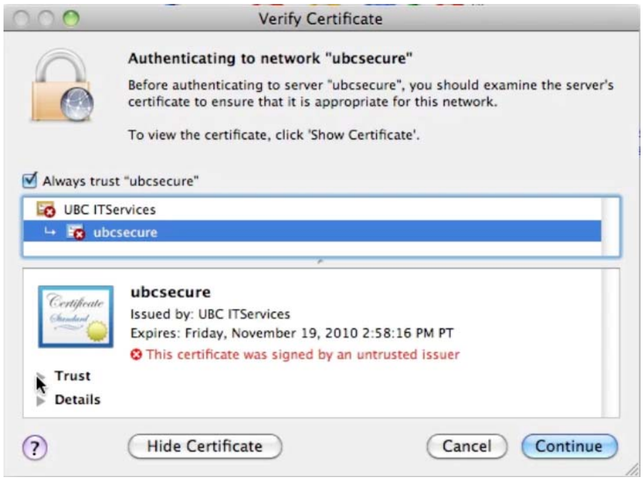
1. Windows

Windows clients will never accept a self-signed certificate for PEAP, and once connected for the first time will only accept certificates signed by the same CA as on the first connection. The error message that Windows provides after an unsuccessful attempt does not suggest that an attack is in progress, but a detail-free "contact your network administrator" message. Server validation is a further step that clients can be configured to perform, but is not configured by default, and UBC's setup guide does not include it [5]. With server validation, the client will check that the server's certificate was issued specifically to cover the wireless network (SSID) in question [6]. Assuming that a CA is doing their job properly, and will not issue two certificates with the same subject name, there is no way to get a successful connection from a fully configured windows client.

2. Mac OS X

Because of the difficulties presented by Windows clients (above), our attack focused on Mac clients. When presented with a self-signed certificate, OS X will prompt the user to "verify" the offered certificate. We believe that every client presented this message accepted the proffered certificate. iPhones present similarly-worded warnings, and succumb identically. A sample of what OS X shows when the user connects to network with a self-signed certificate is shown on the next page in Figure 1.

[1] Originally, we called this the $1500 attack, but as of the writing of this report, such a certificate can be had for $350, a significantly lower barrier.



Verify Certificate

Authenticating to network "ubcsecure"

Before authenticating to server "ubcsecure", you should examine the server's certificate to ensure that it is appropriate for this network.

To view the certificate, click 'Show Certificate'.

☑ Always trust "ubcsecure"

UBC ITServices
↳ ubcsecure

ubcsecure
Issued by: UBC ITServices
Expires: Friday, November 19, 2010 2:58:16 PM PT
This certificate was signed by an untrusted issuer

Figure 1. Certificate Verification on OSX

3. Linux (ConnectionManager)

OS X and Windows both ship with a pre-configured set of CAs to trust. ConnectionManager doesn't have this facility, and so self-signed certificates are accepted on an equal footing with "genuine" certificates. The ability to subject name of the server's certificate, as offered by Windows, is also missing, though it would not make any difference since self-signed certificates can claim anything.

Once the client has accepted our self-signed certificate, the inner handshake begins.

Simplified MSCHAP handshake flow:

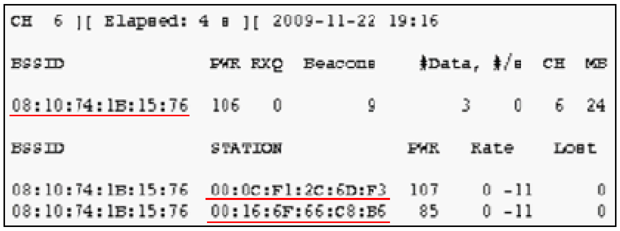
=> server challenge
<= client responds by encrypting the nonce, using a hash of their password as a key
<= client challenge
=> server responds by encrypting the nonce, using a hash of the client's password as a key

This scheme provides mutual authentication, using the user's password as a shared secret. Because the user's password is not known to us ahead of time, we cannot respond appropriately to the client's challenge. Due to the problems with the way the encryption is designed, though, their password can be found using their response to our challenge. Doing this the hard way, we'd need to do three DES keyspace searches, and then reverse a MD4 hash. DES keyspace searches are practical for corporate or governmental attackers, but beyond the scope of our project. MD4 is no longer regarded as a cryptographically strong hash function, but that is not to say reversible. Instead, because the last two bytes of the hash are trivially available, a dictionary attack like the one we mount only has to consider 2^{-16}th as many possible passwords. This weakness has been long known, [7] and tools exist to generate the dictionaries required [8]. Since our goal was only to determine the possibility of the attack, attempts to recover user's credentials were not exhaustive. A sample 10-minute session in the SUB yielded 5 "successful" connections. (Compared to survey of the area, 5 connections was more than the number of Apple computers seen, or connections expected.) Of that sample, 20% of passwords followed the pattern <dictionary word><numerals>, which are trivially crackable with a wordlist. A more determined attacker would further extend a wordlist to include common letter-number substitutions, etc. Note that unless the user changes their password, an attacker can always decide to try more possible passwords at a later date.

B. De-Authentication

The de-authentication attack will take a supporting role of our project. This attack sends de-authentication packets to clients who are accessed to associated access points [2]. The following sections provide an insight of how this attack works and its result as well.

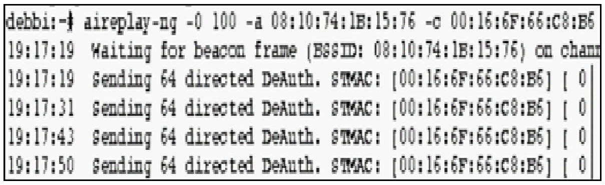
The initial step of de-authentication attack begins with a packet capturing tool that intercepts and logs traffic passing over a network (see Figure 2). This step is necessary that it identifies MAC addresses of access points and their clients. From the Figure, MAC addresses of the access point (BSSID) and the associated clients (STATION) are exposed.



```
CH  6 ][ Elapsed: 4 s ][ 2009-11-22 19:16

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB

08:10:74:1B:15:76  106   0        9        3    0   6  24

BSSID              STATION            PWR   Rate    Lost

08:10:74:1B:15:76  00:0C:F1:2C:6D:F3  107    0 -11     0
08:10:74:1B:15:76  00:16:6F:66:C8:B6   85    0 -11     0
```

Figure 2. MAC addresses of Access Point

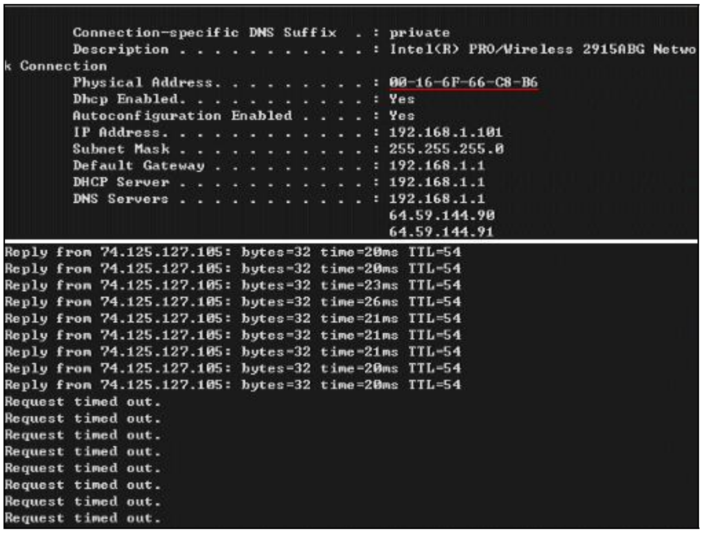
Then, using the MAC addresses of the access points and their clients the program from the attacker's computer sends de-authentication packets to the clients pretending the packets are from the access points (see Figure 3).



```
debbi:~# aireplay-ng -0 100 -a 08:10:74:1B:15:76 -c 00:16:6F:66:C8:B6
19:17:19  Waiting for beacon frame (BSSID: 08:10:74:1B:15:76) on chann
19:17:19  Sending 64 directed DeAuth. STMAC: [00:16:6F:66:C8:B6] [ 0|
19:17:31  Sending 64 directed DeAuth. STMAC: [00:16:6F:66:C8:B6] [ 0|
19:17:43  Sending 64 directed DeAuth. STMAC: [00:16:6F:66:C8:B6] [ 0|
19:17:50  Sending 64 directed DeAuth. STMAC: [00:16:6F:66:C8:B6] [ 0|
```

Figure 3. Sending De-authentication Packets

This results the client’s computers to reconnect to the access points; however, as the de-authentication packets are sent continuously, they will not be able to reconnect, and clients will eventually try to find other access points (see Figure 4).



```
        Connection-specific DNS Suffix  . : private
        Description . . . . . . . . . . . : Intel(R) PRO/Wireless 2915ABG Netwo
k Connection
        Physical Address. . . . . . . . . : 00-16-6F-66-C8-B6
        Dhcp Enabled. . . . . . . . . . . : Yes
        Autoconfiguration Enabled . . . . : Yes
        IP Address. . . . . . . . . . . . : 192.168.1.101
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.1.1
        DHCP Server . . . . . . . . . . . : 192.168.1.1
        DNS Servers . . . . . . . . . . . : 192.168.1.1
                                            64.59.144.90
                                            64.59.144.91
Reply from 74.125.127.105: bytes=32 time=20ms TTL=54
Reply from 74.125.127.105: bytes=32 time=20ms TTL=54
Reply from 74.125.127.105: bytes=32 time=23ms TTL=54
Reply from 74.125.127.105: bytes=32 time=26ms TTL=54
Reply from 74.125.127.105: bytes=32 time=21ms TTL=54
Reply from 74.125.127.105: bytes=32 time=21ms TTL=54
Reply from 74.125.127.105: bytes=32 time=21ms TTL=54
Reply from 74.125.127.105: bytes=32 time=20ms TTL=54
Reply from 74.125.127.105: bytes=32 time=20ms TTL=54
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

Figure 4. End User Loses Current Connection

To facilitate the attack, attackers can choose specific targets instead of broadcasting the de-authenticate packets. Because different vendors use a different prefix of MAC addresses, disconnecting a particular brand of network adapters is feasible.

V. User Survey

During the duration of our project we conducted a user survey regarding the UBC wireless network and the general knowledge of the public about certificates. The questions we asked the public and the answers to the questions were tabulated below.

1.) Do you ever check what network you're connected to while at UBC?
2.) Are you content with the speed that you connect to the internet via UBC Network?
3.) Do you ever connect to the regular UBC network when the signal for UBC Secure is poor?
4.) Is security something you think about when joining any wireless network?
5.) Do you know what certificates are?
6.) Do you know notice what the details of the certificate are? Or do you just press OK/Continue when given a certificate?

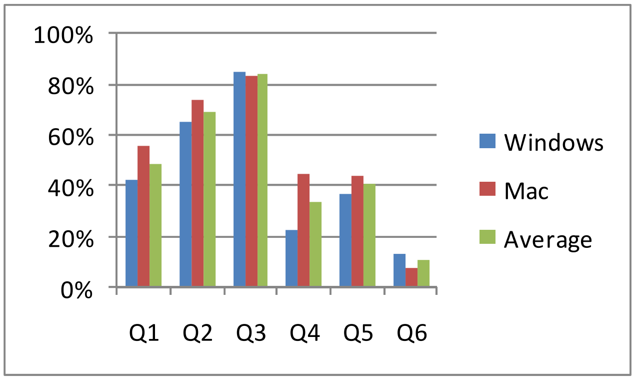


Figure 5. User Survey Results

The sample size of this survey was 25 people for each Windows and Mac operating system. From the data we saw, many users using both operating system would connect to the regular WEP UBC wireless network if the WPA secure network is down or have poor signal. Since launching a WEP attack is much easier than attacking the WPA network, we learn from the survey that we have another possible attack to add to our already successful attack on the secure network. Another good statistic we found from the user survey is the fact that many users “claim” to understand what certificates are but only a miniscule percentage of people notice the details of the certificate. From this statistic, we learned that to some extent the use of a fake “non-trusted” certificate will be sufficient and users will still accept our certificate regardless.

VI. Counter Measures

A. Hardware

The sample size of this survey was 25 people for each Windows and Mac operating system. From the data we saw, many users using both operating system would connect to the regular WEP UBC wireless network if the WPA secure network is down or have poor signal. Since launching a WEP attack is much easier than attacking the WPA network, we learn from the survey that we have another possible attack to add to our already successful attack on the secure network. Another good statistic we found from the user survey is the fact that many users “claim” to understand what certificates are but only a miniscule percentage of people notice the details of the certificate. From this statistic, we learned that to some extent the use of a fake “non-trusted” certificate will be sufficient and users will still accept our certificate regardless

The intrusion detection system uses smart sensors connected to a server appliance with a proprietary application to monitor wireless network traffic. Most WIPS scans for access points in the coverage area and compares the MAC address of the AP with a predefined list that is expected to be in the wireless system. [9] The WIPS identifies rogue access points that is using the same SSID, notifies the administrator with the information as a threat and sends an interference signal to prevent clients from associating with the rogue access point.

Unfortunately, the solution requires a lot of money to implement. The administrator essentially has to build a second wireless network in addition to the wireless network that he wishes to protect. For every access point in the wireless system, a smart sensor will have to be installed with equal strength to cover the same area. For a small company, the cost may not be justifiable. And for a large company or in this case, the University of British Columbia, implementing a WIPS on top of its existing wireless infrastructure would cost a lot of money.

Even if a company chooses to install a WIPS, there is no way to control what the client does outside of the company's coverage area. If an attacker decides to run this attack at an airport and the client has their computer configured to automatically connect to the network, he will still be vulnerable. The system only prevents a large scale attack inside the company or the coverage area but does not prevent the execution of the attack itself.

B. Change of Credentials Used for Authentication

Many WPA-PEAP secured wireless networks have their RADIUS server tied to the Active Directory Server of their domain since this attack will reveal the credential a client uses to authenticate, there is significant risk in the assets in question.

In our attempts to recover credentials used to authenticate with the UBC Secure network, we were able to retrieve the username and password of clients who accepted our certificate. The same username and password is used with the Campus Wide Login (CWL) system for UBC. This login system allows users to access the student service center, make course changes and access student resources. With the same credential as the UBC Secure wireless network, the attacker will have access to personal information and is able to change them if they wish.

The countermeasure for this vulnerability would be to have a separate username and password for authentication with the wireless network after which the user will have to enter their Active Directory password if they were to access different network resources. The attacker will be unable to access network resources using this attack as the credential retrieved will only give them access to the network but not to the resources protected by the Active Directory security policies.

In addition, the wireless network can use EAP-TLS instead of PEAP-MSCHAPv2 for authentication; this would render the attack useless although this change requires the client computer to install a certificate to validate the server for authentication.

C. Education of Users

It is the end user who decides which wireless network to he/she desires to connect to, so educating the user to not connect to suspicious networks is very important. By teaching the users to only connect to WPA networks with trusted certificates it will reduce the chance of connecting to rogue access points and giving out personal information. Furthermore, to understand the details of certificates and how they work can significantly reduce the chance of the user being attacked.

Going the extra mile, users can ensure their private information would not be stolen or modified if they create a secure enough password as a deterrent. The CWL password only requires the user to have a minimum of 8 characters of which there must be one number and one letter. Passwords that only meet the minimum requirement can easily be decoded. By bringing it up a notch and making the minimum requirement harder such as adding at least one special character to the users password, the time needed to decode the password becomes significantly longer. Another strong point users should know is that making a password that does not contain words from a dictionary will almost make your password invulnerable to wordlist and dictionary attacks. Simply by using these two methods of using special characters and using passwords that aren’t in a dictionary, the time needed to brute force the password will increase exponentially and the wordlist of the attacker would need to increase significantly in size

VII. Conclusion

The consequences of having an unsecure wireless network are quite devastating. In our case, CWL credentials were attained through this attack, and with those credentials we had access to bountiful amounts of private information, such as home address, social insurance numbers, health implications, etc. Not only that we had the ability to modify student’s courses, apply for loans and have unrestricted access to the user’s interchange email account. This was only applied to a school campus, but if it was an international corporation, there could have been severe loss of extremely sensitive or private information. We have discussed a few design principles which were violated and the core fundamentals that were broken regarding WPA-PEAP wireless networks. As for the user, to protect yourself and your personal information by extending your knowledge on this topic by applying some of the tips we have provided. From this report we urge the owners of these types of wireless networks to apply some of the countermeasures listed above to reduce the chance of a computer security breach.

VIII. References

[1] "Student Awareness Survey". April 2008. [Online]. Available: http://www.it.ubc.ca/__shared/assets/Student_Awareness_Survey_2008_-_Brief_Summary2733.pdf

[2] "De-authentication". September 2009. [Online]. Available: http://www.aircrack-ng.org/doku.php?id=deauthentication

[3] “Common Vulnerabilities and Exposures”. November 2009. [Online]. Available: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1096

[4] “Wireless LAN Server Certificates”. November 2009. [Online]. Available: http://www.verisign.com/ssl/buy-ssl-certificates/specialized-ssl-certificates/wireless-lan-security/index.html

[5] “UBC Information Technology – Microsoft Windows XP WPA Setup”. November 2009. [Online]. Available: http://www.it.ubc.ca/internet/wireless/wpa/wirelesswpawinxp.html

[6] “Microsoft Support: How to configure PEAPv0 to reduce potential risks against man-in-the-middle attacks and against password-based attacks when you use authentication servers in Windows Vista or in Windows Server 2008”. November 2009. [Online]. Available: http://support.microsoft.com/kb/941123

[7] B. Schneier, Mudge and D. Wagner. "Cryptanalysis of Microsoft’s PPTP Authentication Extensions (MS-CHAPv2)". October 19, 1999. [Online]. Available: http://www.schneier.com/paper-pptpv2.pdf.

[8] “Asleap – exploiting Cisco leap”. November 2009. [Online]. Available: http://asleap.sourceforge.net/

[9] “Wireless Intrusion Prevention System”. November 2009. [Online]. Available: http://en.wikipedia.org/wiki/Wireless_intrusion_prevention_system

[10] Beznosov, Konstantin, “Introduction to Computer Security”, unpublished.