A REVIEW ON PWLAN IN CONTEXT OF WI-FI

Gajraj Singh*

ABSTRACT

This Paper is about PWLAN and as we know Mobile devices with integrated wireless local area network (WLAN) technology is becoming increasingly popular. Not only laptop computers are Wi-Fi (Wireless Fidelity) enabled, also mobile phones, gaming consoles and digital cameras are being equipped with wireless communication modules. Mobile Internet, mobile applications and services depend on connectivity to online platforms thus requiring instant Internet connection. While handsets can make use of third generation (3G) communication technology, most other devices cannot access these networks. Although there has been lately also a trend towards 3G, resp. Universal Mobile Telecommunication System (UMTS) communication interfaces being integrated into laptop computers and net books, 802.11, or Wireless LAN (WLAN) for short, has become fast and cheap enough to be integrated into everyday' things. While at home these devices are easily integrated into our home networks, they are almost useless when travelling, since open and free WLAN access points are rarely available. On the other hand, public WLAN hotspots and open network communities are also increasing in number and already cover large areas in the cities. Most train stations, airports, restaurants and hotels offer wireless access to the Internet through public hotspots. Unfortunately, public hotspots are not as secure as they could be. For home and enterprise wireless networks strong cryptographic mechanisms do exist, providing confidentiality, data integrity and mutual authentication, hence respecting the user's privacy. Wi-Fi Protected Access (WPA and WPA2), also known as 802.11i and Robust Secure Network (RSN), provide network access control based on shared secrets between the client and the network. WPA-PSK, the private mode of WPA and WPA2 restricts access to users knowing a common passphrase (the so called Pre-Shared Key, hence PSK), and the enterprise mode of WPA and WPA2 relies on 802.1X port based access control and the Extensible Authentication Protocol (EAP) for authentication.

Now that we have seen the basic concepts, protocols and standards of wireless LAN security and authentication, we concentrate on public hotspot environments. We identify the different requirements for home-, enterprise and public WLANs. We look at the current approach to authentication in public hotspot environments. The most widespread mechanism today is the Universal Access Method (UAM) in combination with Captive Portal Pages (also known as

WISPr). While this solution provides great flexibility for the different business models, it does not sufficiently protect the customer's privacy and is not the most usable solution.

Keywords:- *PWLAN, WLAN, Wi-Fi ,3G,UMTS,RSN,EAP,UAM,WISPr.*

*Research Scholar, NIMS University, Jaipur, Rajasthan, India.

1.1 PARTY WLAN

The most common solution that is used today by almost all hotspot operators is based on the WISPr recommendations. This involves having an open (unprotected) SSID broadcast, accepting all users without access control. The users are then redirected to an HTTPS secured CPP that lets the user enter the credentials. After the login process, all traffic towards the Internet is granted for that specific user. We have also mentioned several times that this solution does not provide any means of protecting the user before and after the login process. First and foremost, after the user is authenticated, there is no link layer encryption that protects the data traffic between the client and the access point. Not having a secured channel in a public environment can be seen equal to shouting out loud your name, your email passwords and Credit Card number at a party. Most of us would not do that. Unfortunately, using PWLAN without any further counter measures taken is such a Party WLAN.

Is it really that bad? While it is true that everything being transmitted without encryption can be heard by others, the question is what actually is transmitted. Most e-commerce web sites are additionally protected using secure SSL tunneling based on digital certificates. This ensures that all traffic to that specific web site is protected in an end-to-end manner. Hence, entering your Credit Card information in a secured eShop is actually safe, even without enabled channel encryption. On the other hand, email traffic is seldom protected additionally. Standard protocols such as POP3 (Post Office Protocol Version 3) and SMTP (Simple Mail Transfer Protocol) do not by default encrypt the application data. At the Black Hat 2007[1], a security conference also inviting hackers, Robert Graham showed in a live demo how he can get session IDs and authentication cookies out of the unencrypted traffic of an authorized hotspot user. He used the information to hack into the victims Gmail account (check[2] for details). The problem however, is not only that the attacker can read private emails. Often, people use the same (or a similar) password for several purposes and accounts. Furthermore, most otherwise secure websites with user accounts offer to send forgotten passwords via email. That way the attacker can access almost any online account of the victim. Last but not least, a private email account contains much information that can be used for social engineering [3].

People often think that it would take too much effort to actually hack a WLAN. This is not the case, because there is nothing to be hacked. Unencrypted traffic can easily be sniffed with free network monitoring tools such as Ethereal [4] or its successor WireShark [5] without the need of special hardware. Anyone without any special education or knowledge of network

security can easily start an attack. One cannot even consider just listening to open traffic passively as an attack. It is actually not illegal to do that, only the saving and further use of the data is prohibited by law and is therefore a criminal offense.

1.2 THE EVIL TWIN ATTACK

In the Section above we have argued that Credit Card information entered during eShopping typically is protected using HTTPS. The same is true for the credentials which are entered at the CPP for login in. Does this mean that we are also safe to use our Credit Card for authentication at public hotspots? Not so, because of the evil twin.

The evil twin is an attacker that sets up a rogue access point with the same SSID as your PWLAN operator. The attacker then provides you with a faked CPP on which the unsuspecting user will enter his Credit Card information, or subscription credentials. The evil twin just records the log in information and can then even let the victim connect to the Internet, so that he or she does not even recognize having been tricked by a MITM. The next thing will be a massive Credit Card bill or a massive subscription bill at the end of the month. Furthermore, since the victim is directly connected to the evil twin's computer, the attacker can further try to find security holes of the victim's system, e.g. browser exploits, which could allow him to place viruses, malware or directly access the victim's computer hard drive. Why does this work? If the signal of the rogue access point is stronger than the original one, due to a stronger transmission power or just by locating it closer to the victim, the original signal will get suppressed. The only hint that one has to decide whether the access point is legitimate is the SSID being broadcast – which can easily be faked. If one sees a wireless network with SSID "MOBILE", "t-mobile" or "free WLAN", one will connect to it, and if one was successfully connected to a network, the WLAN client will probably connect to it automatically. The next step is to open a browser window that will display the CPP. How can one decide whether this site is actually coming from the hotspot operator? One could look closely at the digital certificate and understand the mechanism behind it. However, most people will recognize the look and feel of their operator's CPP and will easily be tricked.

Again, all the hacker needs to start this kind of attack is a notebook computer with a WLAN interface and dedicated software tools such as AirSnarf [6] to fake the access point. He then only has to redirect all traffic coming from the victim to a faked CPP, to let the victim enter his credentials or Credit Card information. Because it is so easy to perform this relatively simple attack, not only attackers with criminal intentions are doing it, also less skilled computer enthusiasts are trying it out, just out of curiosity. Until now, no hacker has yet been

prosecuted for acting as the evil twin. This is because the victims may never realize that they were using an illegitimate hotspot, and once the evil twin shuts down his rogue access point, it is almost impossible to find him.

1.3 OTHER ATTACKS

The above two attacks on wireless LANs are only two of many others. However, with respect to public wireless LANs, these are those that customers really should be afraid of, since they can easilybe performed without the victims noticing the attack. On the other hand, as we shall see below, these attacks can be prevented by adapting to the 802.1X standard for authentication. One attack that looks very similar to the evil twin attack from the customer's point of view is called session hijacking. Here, the attacker waits until an authenticated user leaves the hotspot without explicitly logging off. The network will wait another minute or so (depending on the timeout value) for further traffic coming from the client. During this timeout period, the attacker can easily pretend to be the authenticated client by spoofing the MAC- and the IP address of the authenticated client. This information is sent unprotected within every data packet. The access controller has no other means of telling whether such spoofed traffic is coming from the previously authenticated customer, or from an attacker. The attacker can then use the customer's session to access PWLAN. This attack is not that dangerous, since the customer's credentials are not compromised as in the evil twin attack. Nevertheless, the customer will get charged for the traffic being generated by the attacker. Another attack that is possible in wireless LANs is the Denial of Service attack (DoS). As the name suggests, the attacker tries to overload the system with a large number of generated messages in order to prevent other users from accessing the network. To be more specific, an unauthenticated user at a public hotspot who is not authenticated will still get an IP address assigned. This is necessary since without a valid IP address, the browser would not be able to access the CPP. An attacker can exploit this by requesting new IP addresses from the DHCP (Dynamic Host Control Protocol) server with faked MAC addresses. By faking the MAC address, the DHCP server cannot decide if the new requests belong to the same sender and thus cannot block further requests. The attacker continues to request addresses until the whole IP range is used up. The server is then no longer able to assign new IP addresses to legitimate customers. This specific DoS attack is called DHCP poisoning. DoS attacks are not that dangerous in public wireless LAN networks since no sensitive data of a customer is in danger. However, the above described DoS attack would not be possible in 802.1X enabled networks. Client machines only get an IP address assigned after having authenticated the

customer. On the other hand, it is never possible to completely protect against DoS attacks in wireless networks, because any radio source can be used to jam the communication channel by generating noise within the frequency band in use.

Of course there are lots of other attacks possible in wireless LANs. It is a never ending game between network engineers and the hacker community. However, most of the attacks are against the network, not the users. The attackers want to get into the network of a company in order to steal sensitive data that could be of interest to others. In public wireless LANs, the only entity to be protected is the customer. With 802.1X authentication using strong EAP methods that provide mutual authentication, the situation would be a lot better from the viewpoint of the customer.

1.4 WHY VPN IS NOT THE ULTIMATE ANSWER

Public WLAN operators are aware of the dangers that arise when not providing link layer encryption such as WPA or WPA2. Their answer is to use a Virtual Private Network (VPN) which provides confidentiality, data integrity and authenticity. VPN is a software solution that establishes a secure tunnel between two networks. A public wireless LAN user can install a VPN software client and then connect to a VPN server. This will encrypt all traffic between the client and the VPN server. On the one hand, this is a good solution to overcome the missing channel encryption. Once the VPN connection is established, the user must not fear anymore that his traffic is being sniffed by an attacker. He can then securely send and receive mails, do online shopping and chat with friends. However, the VPN solution is not the ultimate answer, for several reasons. First of all, just promoting the use of VPN software by customers – as WISPr recommends it – puts the responsibility of secure networking with the customer. Hotspot users must be aware that they are sitting ducks without VPN's protection. Then they have to know that there is something such as VPN that can be used to securely browse the Internet. However, just knowing it does not secure anything. The customer then has to find and download a VPN client, install and configure it correctly. This is not an easy task that just any using a computer user can handle. Furthermore, a VPN server that can be trusted must be known in order to connect to it. Nowadays, most large companies provide a corporate VPN server that can be used by employees in order to connect to the enterprise's network. What about the others? They can either choose a freely available VPN server, or pay for a commercial VPN service. Free VPN servers are not easy to find, and must be questioned whether they can be trusted, since any further traffic will go through this server. As you can see, all this is way too much for a Starbucks customer wanting to check

his emails. PWLAN operators often support their customers in the process by providing a VPN client that can be downloaded directly from the CPP and a VPN server that can be used for secure Internet browsing. Nevertheless, most people are already struggling with the authentication process itself and are so happy once they can access the Internet that they just forget to care about their security. In my opinion, it should be the operator's responsibility to provide secure networking, not the users'.

1.4.1 Link Layer still unprotected

There are other, more technical issues with the VPN solution. First of all, VPN can only go as far down in the OSI layering model as to the network layer. This leaves the link layer unprotected, allowing a variety of attacks to be performed. Whether VPN is used or not, each client gets an IP address assigned before authentication is performed. This allows the aforementioned DHCP poisoning attack, as well as so-called domain casting [7] attacks which allow tunneling Internet traffic through the DNS service. There are other, more severe attacks possible that could shut down the network completely.

1.4.2 VPN Latency

Moreover, establishing a VPN connection takes time. Even if it "only" takes several seconds, this will become an issue when the user is mobile and wants to connect to another access point. As soon as the user moves out of the range of an access point through which his VPN connection is running, the connection will be aborted. He then has to first authenticate at the new hotspot, and then re-establish the VPN connection. This makes latency critical applications such as Voice-over IP (VoIP) infeasible. On the other hand, 802.1 X authentications provides fast-handoff and fast re-authentication mechanisms which allow changing from one access point to another in less than a second. Latency is not only an issue with VPN when establishing a new connection. Since VPN encryption, depending on the protocol in use, is relatively resource intensive. VPN clients are typically software based, hence require a good share of CPU and RAM for encrypting the traffic. This results in increased round trip times that are affecting time critical applications. Furthermore, if the VPN server is not located at the operator, additional hops are introduced depending on the geographic location of the server. Again, this may further increase latency.

1.4.3 VPN Scalability

An important issue with VPN is scalability. If all hotspot customers were to use the operator's VPN server, the server would quickly become a performance bottleneck and a single point of failure. Surprisingly, the VPN server that Swisscom offers to its customers is rarely used. On the one hand, typical hotspot users today still are managers that travel around

the world using PWLAN to connect to their corporate network. They do that by directly connecting to their company's VPN server. On the other hand, an almost idling PWLAN VPN server at Swisscom shows that private users not having the possibility to use their corporate VPN server either do not know, or do not care about their security.

As already mentioned, VPN uses up resources, not only on the client side. The server has to open and maintain a session for each user separately. This dramatically restricts the number of concurrent sessions, hence limiting the number of customers that can use the VPN server to secure their network connection. If the server is down due to whatever reason, customers can no longer securely browse the Internet.

1.5 802.1X AUTHENTICATION AT PUBLIC HOTSPOTS

According to Trustive's report on PWLAN usage in 2007, 45% of hotspot customers were CEOs, Chairmen, Managers and Sales representatives. Swisscom states that over 90% of their PWLAN customers are business people. However, the ongoing trend towards more and more mobile devices with Wi-Fi capabilities will affect this usage pattern. Private users will increasingly look for available access points, and will also be willing to pay for PWLAN Internet access. Since they will not primarily connect to a corporate network using VPN, transparent, automated logins, security and privacy concerns will become important operator selection criteria. At the beginning of this chapter, we have seen that current WISPr-style solutions are not providing the required security. No link encryption, vulnerability to the evil twin attack, uncomfortable login using a browser, all this clearly shows that the WISPr recommendation is no longer the appropriate solution. On the other hand, 802.1X port based authentication using the EAP framework and RADIUS messaging, solves all these problems. However, 802.1X is not primarily intended for public WLANs and thus the different requirements must be addressed. At the time the first devices supporting the new WPA standard became available, several experts stated that the upcoming 802.11i (WPA2, RSN) standard with its different EAP flavors will be the perfect solution also for hotspot operators. Microsoft states in the context of introducing the Windows Provisioning Service (WPS, will be discussed shortly), that: *"As lack of security is a growing concern, the wireless industry in general is in agreement that* ***public Wi-Fi hotspots need to be secured as well, using technologies such as 802.1x*** *and Wireless Provisioning Services."* [8] George Ou, a former ZD Net journalist, IT consultant and network security specialist, said in early 2005: *"A recent story on "Evil twin" Wi-Fi networks that spoof legitimate hotspots or corporate networks* ***makes it clear that all public hotspots should immediately implement 802.1X and PEAP***

authentication. *Currently, with most Wi-Fi hotspots, there is no simple way to tell whether or not you are using a legitimate hotspot. If you don't think this is a big deal — since you're probably using VPN anyway –think again!"* [9] Also the Wi-Fi Alliance writes in its early WISPr recommendation, that:

"The recently introduced IEEE 802.1x standard provides a protocol for authentication and port-based access control supporting enhanced access security, but has not been widely deployed in public access environments". These statements clearly show that 802.1X is also applicable to public hotspot environments, not only large enterprise networks. The problem in the beginning was that in order to support the new WPA2 standard, the hardware had to be upgraded. This includes the operator that had to update its access points, and the customers, required to have WPA2 compatible network interfaces in their devices. Naturally, it takes time until the standard is widely available. However, almost five years after the standard was ratified, every wireless network card must support the standards in order to get certified by the Wi-Fi Alliance. Have the hotspot operators also updated their networks? Unfortunately, the answer is no, with only a few exceptions:

T-Mobile, a mobile phone subsidiary of Deutsche Telekom AG, announced in October 2004 [10] that they upgraded their networks in the US to the new standard:

"The company […] said today that it has officially completed the upgrades to support 802.1X in the access points at its 4700 domestic locations, one year after announcing plans to do so." [10] However, in order to authenticate at T-Mobile's hotspots using the new authentication method, the customers need to download client software, called the *T-Mobile Hotspot Manager*. We will see later, why usually client software is needed, despite the fact that almost all operating systems have pre-installed a client that is able to connect to 802.1X access points according to the standard. Another hotspot operator that took the challenge of upgrading his network is Swisscom. Swisscom jointly developed the EAP-SIM authentication method, which is used nowadays in combination with the *Unlimited Data Manager* client and dedicated UMTS/3G/PWLAN hardware. Unfortunately, EAP-SIM only works with devices that include SIM card readers. Such devices are gaining popularity, since more and more notebook computers and netbooks are equipped with UMTS/3G connectivity, thus include such a SIM card reader. Nevertheless, most current devices do not support EAP-SIM. Non-SIM based 802.1X authentication on the other hand, is not yet possible in Swisscom PWLANs.

1.5.1 Reasons for not adapting to 802.1X

The two examples of a successful implementation of the 802.1X standard show that it is possible to deploy the enhanced security standard in public wireless LANs. Unfortunately, almost all other hotspots have not adapted yet. What are the reasons for sticking to the insecure Universal Access Method using Captive Portal Pages? One simple reason is that PWLAN customers do not ask for the additional security, hence there is no need to upgrade. This has also to do with the fact that still almost all users of PWLAN are business customers, who are used to use VPN software. With 802.1X, they would still need VPN software – not primarily to secure their communication channel, but to connect to the corporate network. Entering Credit Card numbers into a CPP is a process which frequent business users get used to. Another option for corporate customers is to use services like iPass, which will let users log in using a third party connection manager (the *iPass Connect* client), which provides a one-click solution based on iPass' proprietary protocols and servers, connecting through PWLAN.

The problem is that people do not ask for the additional security, because they are not aware of the current situation. If one would demonstrate to a regular PWLAN guest what actually can happen, he or she would be so scared that he would decide not to use a public hotspot anymore, even if the networks had been upgraded in the meantime. What we are saying here is that security should be the responsibility of the operator. Why are so few private people using PWLAN? Often current price plans are too costly for sporadic usage. Prices, however, are beginning to drop, since there are operators already offering flat-rate based price plans for a few Dollars per month (e.g. Boingo [11]). Aside from the cost factor, there are other reasons why people seldom use PLWAN. Either they do not know that it exists, or because they do not want to use their Credit Card account to be charged for PWLAN usage. People are uncertain how much it will actually cost them, even if prices are clearly communicated. There is just too much mistrust against PWLAN, not only because of the bad news stories associated with public networks. Another important reason why PWLAN operators have not yet adopted the 802.1X standard is the problem of correctly configuring the devices. Although modern Wi-Fi devices supporting 802.1X authentication have integrated 802.1X clients pre-installed in the operating system, they still require careful configuration. Due to the diversity of different operating systems and devices, there is no common configuration that suits all. The consequence is that individual configuration procedures, profiles or client software has to be developed. In the introduction we already mentioned that security is always a trade-off between the level of security on the one hand, and usability and cost on the

other. The current Universal Access Method (UAM) is a solution that clearly goes for usability and lower cost, thus compromising the security level. However, we will see that an 802.1X implementation does not cost much more, and in terms of usability can be even better than the UAM method.

1.6 USER EXPERIENCES

Let us now look at the different user experiences of customers authenticating at public hotspots. Supposed Alice is a Swisscom customer with an existing Natel® (the cellular phone of Swisscom) subscription. She wants to check her email, listen to a newly released music album on iTunes and finally buy and download it, everything using her Apple iPhone [12]. We have chosen the Apple iPhone platform since it is the best representing the growing market of mobile devices with Wi-Fi capabilities. It also stands for the new kind of the ultra mobile customer that will use PWLAN in the near future, using bandwidth demanding applications. The iPhone (from firmware 2.0 on) supports the full 802.11i standard, including WPA and WPA2, both in private and enterprise mode. The experience would be similar using a laptop computer with a WLAN network interface, independent of the operating system.

1.6.1 Captive Portal Page User Experience

Let us first consider the current situation with a CPP. Alice opens the network GUI on her iPhone to check weather public WLAN is available. She will then see the access point with SSID “MOBILE” and tries to connect. The connection will be established without any further user intervention since no credentials have to be entered. Alice now thinks that she is connected since the WLAN indicator of her iPhone indicates a working connection. However, Alice is not yet authenticated, thus does not have Internet access yet. Unaware of that, she will open the email client and try to check her mails. The attempts will time out without any hint why it is not working. A simple message, such as “cannot connect to the server” will be displayed. The same will happen, if she opens iTunes. However, since Alice is vaguely familiar with network protocols, she wants to find out what the problem is, and tries to open the Google search. Once her browser (Safari) opens and tries to connect to the search page, the Captive Portal Page of Swisscom will appear. Now Alice realizes that she first has to log in. She enters her mobile subscription number and receives an SMS message shortly after. She enters the one-time-password that she received into the CPP which will grant her access to the Internet. She can check her mails and listen to the new music album. Alice will be billed directly through her monthly subscription bill. The next time Alice walks into the range of a PWLAN access point, the iPhone will automatically connect to the SSID

"MOBILE". However, authentication will not be performed in an automated manner. This time, Alice does not want to use PWLAN because she just wants to check her emails using UMTS/3G connectivity. Unfortunately, the iPhone thinks that a Wi-Fi connection is available (since it is connected to the open access point) and disables UMTS/3G. Alice will not be able to check her emails, since she would first have to open Safari, request a new OTP and manually enter these credentials. The iPhone is stuck in a so-called "hotspot black hole". Alice has to manually disable Wi-Fi connectivity temporarily to use 3G when she is in the range of a "known" hotspot.

1.6.2 Smart Client User Experience

There are several applications available that act as the UAM Smart Client as described in the WISPr recommendation. Such applications are available for a diversity of operating systems, including the iPhone OS. Example applications available in the Apple App-Store are AutoWi-Fi [13] and Devicescape Easy-Wi-Fi [14]. Such client software allows automatic authentication based on username and password credentials which you can get from your operator, if supported. For example, Swisscom customers can use their DSL subscription credentials to log into Swisscom PWLAN hotspots. Once the application has been downloaded and installed, the credentials have to be entered only once. When Alice comes into the range of a PWLAN access point, she will again fall into the hotspot black hole, if she was once connected to the access point before. However, instead of having to open the browser and manually enter her credentials, she can now just open the third party application which then automatically authenticates her using the previously entered credentials using the WISPr XML protocol. This solution however, still requires Alice to explicitly launch the third party application, since there is no possibility with the current iPhones to allow applications to be running in the background. If it were possible run such applications in the background, the application would be able to detect that a known hotspot is in range and then automatically authenticate. Other operating systems, e.g. Microsoft's Windows XP allow such software, thus enabling true seamless authentication without any user intervention. While this solution can be really comfortable for the user, there is still the issue that no link layer encryption is enabled. If Alice wants to securely browse the Internet and check her private emails, she still would have to configure and run a VPN client. This involves clicking, or "tabbing", another few unnecessary buttons. Furthermore, Alice has no clue if she actually is connected to a legitimate access point.

1.6.3 802.1X User Experience

802.1X promises to provide the same seamless experience as with Smart Clients, while also protecting the network and the user with mutual authentication and strong encryption mechanisms. Let us now consider the situation that Alice would experience if the operator offers 802.1X authentication. However, we now have to consider two separate cases. The first being Alice with her device already configured and the second when Alice tries to connect to a new hotspot for the first time. Let us start with the former scenario, where Alice has a configured iPhone. When Alice walks into the range of a known hotspot (i.e. one of an operator for which she has configured her device), the iPhone will automatically connected to the hotspot within seconds. No third party application has to be launched, no captive portal page has to be opened, and no password has to be entered. This is seamless authentication, secure and without user intervention. As simple and easy as this works for frequent customers with configured devices, as tedious it can get when new customers try to connect for the first time. This issue is the topic of the next Sections.

1.7 PROVISIONING 802.1X CLIENTS

If Alice has never been connected to a PWLAN hotspot with her iPhone before, she first has to configure her device (this holds for all devices, not only the iPhone). A correct configuration of the device is a problem with 802.1X. Incorrectly configured devices will not be able to connect or be still vulnerable to the evil twin attack. This information originates from Joshua Wright, who presented these aspects of PEAP provisioning at the ShmooCon in 2008 [15].

1.7.1 Important Client Settings

There are several issues that must be considered when configuring the supplicants on the devices. While default configurations can work (in case of the iPhone the connection would successfully be established using the default settings), others, such as MS Windows XP, will not connect. However, all devices need some important settings in order to perform true mutual authentication. The problem comes from the fact that 802.1X in combination with an EAP method that supports mutual authentication is typically based on server certificates that need to be verified. The server certificate is used to authenticate the network to be genuine, i.e. guarantees that the access point is legitimate. Based on the certificate, a TLS tunnel will be established between the server and the client that allows secure authentication of the user. If the server provides a certificate that can be verified by the operating system's pre-installed root certification authorities (root CA), then the user will not even be asked to check the

certificate, since it will automatically be verified and accepted. On the other hand, if the certificate were not verifiable with root CAs of the OS, then the user would be asked whether the certificate should be accepted. However, how can the user tell if the certificate is valid? While common SSL certificates used for HTTPS contain a fully qualified domain name (FQDM) in the common name (CN) field, which has to match the URL of the HTTPS server, there is no such requirement for certificates used for network access. This means that an attacker could use a certificate that is either accepted by another root CA of the OS, or provide a private certificate that must be verified by the user. Either way, the common name in the certificate is not checked against the server addresses to be used for authentication. However, this check is essential for true mutual authentication, hence must be enabled in the supplicant. Furthermore, a list of legitimate authentication servers must be defined. Last but not least, as in HTTPS, letting the user decide whether or not to accept untrusted certificates asks too much of the user. Most users will not understand the exception and just accept the certificate. This would allow an attacker to perform the dangerous evil twin attack. Consequently, such certificate exceptions must be disabled. Let me summarize what settings need to be configured in order to avoid the evil twin attack completely. Every supplicant has to be configured in such a way that the operator explicitly defines the following settings:

- **Server certificates must be verified-** Without this property set, mutual authentication is disabled.
- **The root CA that must be used to validate the server certificate must be specified-** If this setting is left unspecified, any root CA can potentially be used to validate server certificates, thus enabling the evil twin to provide his own certificate.
- **Deactivate user exceptions for unknown certificates-** If such exceptions are allowed, the evil twin could use any certificate, and the user must decide if it should be accepted. This decision cannot be made by the naive user.
- **Define a list of Common Names (CN) of RADIUS servers that must match the CN of the certificate-** This ensures that only legitimate authentication servers are allowed. Since the CN of the certificate is checked against the list, an attacker can no longer provide a certificate with an arbitrary CN.

Only if these settings are made, true mutual authentication is performed, and the evil twin attack is no longer possible. All this may seem a bit confusing, complicated and cumbersome. This is true to some extent, and is also one of the reasons why PWLAN operators are not going for 802.1X authentication. Letting the customers configure their devices themselves by

providing detailed tutorials would not be acceptable. On the one hand this would ask too much of the customer and would be prone to user errors, and on the other hand this would drastically increase help desk calls. The only way to solve the problem is to find individual solutions for each device. Nevertheless, once the configuration is done, all advantages of 802.1X will shine.

At this point we want to closely look at the correct client configuration for Microsoft Windows XP's built-in supplicant. We choose the Protected EAP (PEAPv0 with MS-CHAPv2) as the EAP method, because PEAP is the most common EAP method and is supported by all Wi-Fi certified products, hence the perfect candidate for public hotspot access.

1.7.1.1 Manually configuring Windows XP for PEAP

Microsoft Windows XP is still one of the most wide spread operating systems on the market. A public hotspot operator must support devices running this operating system. However, the WPA2 standard which uses advanced cryptography (AES) for network authentication is only available since Service Pack 2. This is why we consider MS Windows XP with Service Pack 2 or later installed. Older XP versions prior to SP2 would require an update patch[16]. The client settings for wireless networks can be opened by right clicking on the *wireless status indicator* located in the taskbar, then selecting *View available networks* and finally clicking *Change advanced settings*. On the second tab called *Wireless Networks* select the network with the PWLAN operator's SSID (typically "MOBILE" for Swisscom PWLAN) and click *Properties*.

The second tab called *Authentication.*

In order to enable PEAP authentication, the EAP type has to be selected accordingly (1) Clicking on the *Properties* button opens the *Protected EAP Properties* window. Here we can set all properties as described further above. The first check mark will (2) enable server authentication (set by default), (3) is restricting the authentication server to specific addresses and (4) selects the root CA to be used for certificate verification. Finally, (5) deactivates certificate exceptions which would allow the user to accept unknown certificates.

As you can see, these settings cannot be made by the naive user. A missing check mark in the *Trusted Root Certification Authorities* list would result in unsuccessful connection attempts, which would abort without an error message. The customer would either call the support hotline (which can cost the operator up to CHF 60. - per case), or he would give up trying and never use PWLAN again. Hence, the only way to correctly setup the client is to let the operator do the configuration. Naturally, the operator cannot manually configure every single

device requesting network access. The question is how can these important settings be made without having direct access to the device? As it turns out, this is the most serious problem with 802.1X in hotspot environments. Nevertheless, there are solutions to this problem which we now will look at in detail.

1.7.2 Automated Provisioning of the Supplicants

Unfortunately, there is no universal mechanism available that allows the automated provisioning of supplicants (i.e. automatic configuration of the client device). This means, that individual solutions have to be found for the different systems, devices and operating systems. In the context of this thesis, we will analyze two prominent platforms, namely MS Windows XP and Apple's iPhone.

1.7.2.1 Provisioning in MS Windows XP

In enterprise environments, administrators are in full control of all devices and have the possibility to automatically provision the supplicants using Microsoft Active Directory. MS Active Directory allows administrators to define Group Policies that can be pushed to the clients (only to Windows XP/Vista clients running the Windows Zero Configuration (WZC) service). These policies can also include the settings for true mutual authentication for PEAP as described in the last Section. Unfortunately, this solution is not applicable in hotspot environments. Or is it?

1.7.2.1.1 Windows Provisioning Service

In 2003, Microsoft teamed up with several hotspot operators (including Swisscom) and promised a sound solution for PEAP provisioning at public hotspots. The service is called *Wireless Provisioning Service (WPS)* and was introduced in October 2004. WPS is installed on all Windows XP systems with Service Pack 2 or later natively. With WPS, Microsoft introduces a so-called provisioning server into the hotspot operator's network architecture. This server can be seen as a normal HTTPS web server that contains profiles that will automatically be downloaded before the authentication process. However, these profiles contain more than just a correct PEAP configuration. WPS also helps hotspot operators in the process of subscribing new users. Based on the information contained in the profile, a Windows styled sign-up wizard will pop up that allows the unregistered user to enter his name and address, billing information, Credit Card number and let him select different subscription options.

The operator has full control of all texts that are displayed, can brand the wizard with his own logo and define different subscription plans. Microsoft provides a tool which supports the operator in the wizard design process and generates the XML based profiles to be stored on

the provisioning server. The tool is called *WPS Authoring Tool* and can be downloaded from Microsoft [17]. In order to see how WPS works from a protocol viewpoint, let us consider the case where a new hotspot customer visits the hotspot for the first time. The process is performed in three phases.

1. The customer walks into the range of an access point of the hotspot operator. Windows XP will show the broadcasted SSID and let the customer connect. The client authenticates using a guest account. This is necessary, since the customer has no own credentials yet. The client automatically connects using no username and no password. The authentication server sends the URL of the provisioning server inside a PEAP-TLV message (PEAP-TLV is a mechanism of PEAP that allows arbitrary data to be exchanged between the client and the server inside a normal PEAP message). Additionally, the client gets a temporary IP address assigned. The client connects to the provisioning server and downloads the XML-files (the profile).
2. Based on these files, the WPS Sign-up wizard will be loaded which prompts the user for identity and other information needed to set up a new account. Once the user has completed the wizard, the information is sent back to the provisioning server. The server checks the payment options, sets up a new account and sends an update to the AAA server. The provisioning server generates a new XML-file which contains the correct PEAP settings, including the username and a password.
3. Once the client has received the configuration file, it automatically disconnects from the access point, and re-establishes a new connection, now using the username and the password contained in the configuration file. The authentication server informs the access gateway of the successful authentication and assigns a new IP to the client.

It has to be noted here that in the above diagram several functional entities are omitted. Important missing entities are the Access Controller, the Access Point itself, a DHCP server as well as a Microsoft Active Directory server. The Active Directory server is used to manage the accounts and provision the clients. Other LDAP based servers can be used instead if they support the dynamic creation of new accounts. The AAA server in a WPS scenario is typically a Microsoft IAS (Internet Authentication Service) which is integrated in current Windows Server solutions Of course there would be a lot more to tell about WPS. However, this would be useless, since Microsoft decided to no longer support the WPS service in current and future OSs (Windows Vista and Windows 7).

The main reason for no longer supporting WPS was because the PWLAN operators would have had to invest too much money into Microsoft‘s products and licenses. While this would have enabled seamless and secure provisioning of devices, this solution would have been restricted to devices running Microsoft’s OSs.

1.7.2.2 Software Client Requirements

You might ask why we presented the WPS solution despite the fact that it is no longer supported by Microsoft. The reason is the following. With WPS, Microsoft provided a solution to several issues of 802.1X in the environment of public hotspots. It shows very clearly, what problems need to be solved in order to allow secure and seamless authentication in public WLANs. As already mentioned, not only correct configured devices are an issue, also the user interaction with new customers is a problem. 802.1X tries to avoid CPPs, but how can new users set up a new accounts if no browser is involved at all? How can existing customers renew their subscription or change their password? Microsoft’s WPS shows how it should be done: using dedicated client software. The good news is that when Microsoft stopped their WPS efforts, they published a new APIs that allows developers to use the native wireless services and functions of the Windows OS. The API is called the *Native Wi-Fi API* and replaces the WPS functionality. It is available on Windows Vista, as well on future Windows 7 versions. Microsoft has also down-ported a subset of the API to Windows XP with Service Pack 2 and Hotfix KB918997, as well as Windows XP with Service Pack 3 [18]. However, a detailed look of the new Native Wi-Fi API is beyond the scope of this thesis. Instead, we will identify the issues that need to be addressed when developing a dedicated software client for 802.1X authentication in pubic hotspot environments. Among others, the most important design requirements are:

- Promoting the secure 802.1X authentication, deployment of the software client
- Interaction with new users, account management for existing customers
- Correct configuration of the devices / supplicant
- Cost control
- Credentials Management

Let us go through each listed item in the next sub-Sections.

1.7.2.2.1 Promoting Secure Authentication, deploying the Client Software

An important question is how customers are being informed about the availability of the secure 802.1X login and how they receive instructions on how to download and install possible client software. PWLAN customers are used to connecting to an open, unprotected

SSID and being redirected to a CCP. If 802.1X would be available, a separate SSID would be required. However, how can a customer decide which SSID he should connect to? One option is to hide the 802.1X SSID and only make it accessible for clients with pre-configured devices, i.e. the software client installed. A new user (one without any configuration or software client) would still connect to the open SSID and get redirected to the CPP. He then can use any of the traditional (less secure) authentication methods that he is used to. Additionally, the CPP can be used to promote the use of secure communication and offer a direct download for the client software. This approach is different from WPS, where the client software is already pre-installed, i.e. integrated in the OS. Naturally, it is not possible to have operator dependent clients pre-installed in the OS.

WPS solves the problem of connecting to the protected hotspot without having valid credentials by providing guest credentials (no username and no password). However, common supplicants of the different operating systems might act differently. For example, pre-WPS Windows OSs will try to connect to the protected access point using the Windows-Logon credentials. This kind of behavior stems from the corporate world, where the Windows Logon often is the same as the network access credentials. However, whatever credentials are being sent by non-configured supplicants, a hotspot authenticator could accept any username and password pair in order to establish a connection and allow assigning an IP address to the client. As with normal CCPs, such un-authenticated clients could then be redirected to a CPP where the customer can setup a new account and download the client software or network profile. Of course, customers providing credentials that belong to an existing account are automatically authenticated and access is granted without redirection to a CPP.

1.7.2.2.2 Interaction with new users, account management for existing customers

The interaction with new users is already covered the last Section. However, suppose a customer has successfully downloaded the client and gets automatically connected to the secure hotspot each time he walks into the range of a known hotspot. How can he or she be informed about the current price plan of the hotspot? What if his or her subscription period is over and must be renewed? As in WPS, since the client is now installed on the end-device, it would be straightforward to integrate this functionality into the client software. User interaction can be handled comfortably for the customer through a well designed GUI. However, this would require the client to interact with a server, thus introducing additional complexity, compared to a client-only solution. An alternative would be to provide a web link to a web application where the customer can log in and manage his account in the web

browser. Since this also requires additional server capabilities and would require the customer to log in a second time, the former solution is preferable with respect to usability.

1.7.2.2.3 Correct configuration of the devices

The configuration of the device (i.e. the supplicant) is essential to secure networking, as discussed further above. Provisioning these essential settings to the client is the primary intention of the client software. Moreover, once the configuration has been stored on the device, the software client could be removed if the connection profile is kept. The device would still be able to connect to the hotspot because authentication and connection establishment is performed by the OS. Furthermore, connection profiles (i.e. client configuration) can be stored on the network. The client can be set up to periodically update the settings, enabling the hotspot operator to make changes in the network architecture without bothering the customer.

1.7.2.2.4 Cost Control

One important issue with seamless authentication in public hotspot is cost control. In a true 802.1X enabled network, the client gets automatically connected as soon as he or she walks into the range of a hotspot. However, current price plans are typically based on connection time or data volume. Automated authentication implies that the user might be unaware of the fact that he is connected. This can be dangerous, because nowadays almost every application will try to access the Internet and check for updates. Application updates can be several hundreds of megabytes large and would cost the customer a fortune, depending on his subscription plan. Automated updates running in the background or traffic being generated without the user's knowledge can be reduced to some degree by the operator maintaining a blacklist of known update server URLs and ports. Nevertheless, this is no 100% solution since blocking too many ports and addresses could also restrict normal browsing and other applications which the customer wants to use. The problem might be less severe if we look a bit into the future of PWLAN. Nowadays, customers pay for PWLAN usage based on time or data volume. However, the trend clearly goes towards flat-rates. If customers can use PWLAN as long as they want to with almost no limits, they would not have to worry about updates running in the background. On the other hand, the PWLAN operator still would want to minimize such traffic, since it uses up valuable channel capacity.

1.7.2.2.5 Credentials Management

Last but not least, it has to be mentioned that the storage of the customer's credentials can be problematic with respect to security. Some operators have security policies which currently do not allow user passwords for PWLAN access to be stored on the client. Actually,

password based credentials are problematic anyway. Every system is only as secure as the weakest segment in the whole chain. Both passwords are human friendly and thus too weak, or they are strong passwords which the user cannot remember. Letting the user decide on the password is the worst thing that can be done because such passwords can easily be compromised using dictionary attacks and social engineering. On the other hand, strong enough passwords (randomly generated, including symbols, upper- and lowercase letters and 12 to 14 characters long) are cumbersome to be entered and are prone to be written down. On the other hand, if the credentials can be managed by a trusted client developed by the operator, then they can be stored safely. The problem of human unfriendly passwords also gets alleviated if they have to be entered only once. We have now seen what problems need to be addressed when 802.1X with EAP, independent of the specific EAP method, is used for public hotspot authentication. All these issues can be solved using a dedicated software client. Using Windows's Native Wi-Fi API, such client software can be kept relatively simple, small in size and be individually customized by the operator at the same time. Nevertheless, such client device. This would at least include Microsoft's OSs, Apple OS X, Apple's iPhone, Nokia devices, Windows Mobile based devices, and maybe the Android platform. This is the challenge that an operator has to face when considering to adapt to 802.1X authentication. It is actually the main reason why 802.1X is still not common for public hotspots. The effort seems too high and too costly. However, as we will see in the next Section, individual solutions can be very simple.

1.7.2.3 Apple iPhone Provisioning

The Apple iPhone represents the new kind of mobile devices which will get very popular in the next years. It was the first device that allowed true mobile Internet browsing and is currently the largest platform today for mobile applications offering over 20'000 different applications (as of 11.2.2009). iPhone users regularly check their emails, download Google maps, watch Youtube videos, buy online music, twitter and upload photos and soon also videos to community web portals. These applications are bandwidth intensive and thus users would welcome seamless and easy to use PWLAN integration. We will now present a solution which does not use a dedicated software client to be installed on the device. Instead, we make use of so-called iPhone profiles [19][20]. iPhone profiles are XML based configuration files that are intended for enterprise administrators to deploy the employees' iPhones in a centrally managed manner. Such profiles can include security policies, VPN connection settings and configuration settings to be defined and deployed using the *iPhone Configuration Utility* which is freely available from Apple [19]. iPhone profiles can be

downloaded from a (secured) web site accessed directly with the iPhone, or they can be mailed to the iPhone. Furthermore, profiles are also supported by Apple's iPod Touch devices. The profile settings in which we are interested with respect to 802.1X authentication are the Wi-Fi settings. A native Mac OS X version is available which offers further deployment options software has to be developed for each and every operating system. All important settings can be set in iPhone profiles. First and foremost, the SSID of the hotspot and the encryption type can be set (1), the according EAP method be selected (2) and a username for outer- (3) and inner (4) authentication can be defined. The most important settings are related to trust, i.e. the certification authority (5) that will verify the server's certificate (can be included in the profile) and the common names of the authentication servers (6). Finally, trust exceptions must be deactivated (7) as an important security requirement. In order to ensure that the profile is legitimate it must be signed using a digital certificate and a private key. Depending on the certificate's origin, the profile will be trusted by the iPhone's pre-installed root CAs. As mentioned above, iPhone profiles can be published on a secure web server where users can download it (e.g. on the CPP) or they can be directly mailed to the phone. When a new profile has been downloaded, the user is prompted whether it should be accepted. Once the profile is accepted and installed, the customer can connect to the hotspot simply by hitting the "connect" button in the network settings. The user can then enter the password, which will be stored for future connection attempts. The iPhone will perform true mutual 802.1X authentication based on PEAPv0 with username and password credentials. The next time the user walks into the range of a known hotspot, the iPhone will automatically connect to it, without any user interaction. This iPhone provisioning based on profiles has not yet been proven to be working as described. Actually, there are several issues that still need to be solved. One of them being that it is required to explicitly enter a username in the profile. Of course, this is not what hotspot operators would want to do, since the best solution would be to just have one common profile for all iPhone customers, not a separate one for each user. On the other hand, if iPhone profiles could be generated dynamically for each new customer, then the profile could also include the customer's password. This would be very convenient for the user and would allow using very strong passwords.

However, these are details that must be addressed within a proof of concept in a next step. Furthermore, this solution does not address the problem of interacting with the customer (e.g. account and cost management). One possible solution would be to still redirect the customer after authentication to a CPP where he or she can see account information. Another approach

would be to send an SMS to the customer stating that he currently is connected to a hotspot and what the current prices are. Moreover, the iPhone (and most other cellular phones) do support so-called flash-SMS and network status messages. The former is similar to a normal SMS, however it will be directly displayed to the user without having him or her to open the SMS inbox. Network status messages are typically displayed in the status bar of the device. Such short message based interaction with the customer would be also applicable for devices other than an iPhone. Customers connecting to a hotspot with a laptop computer could also receive an SMS indicating the connection status and current prices. Last but not least, customer interaction could of course be done though a dedicated native iPhone application for account management. Despite the fact not all issues with iPhone profiles in hotspot environments are solved, this example shows that it can be relatively simple to find individual, device specific solutions to the provisioning problem. This is why we presented my idea. Actually, one day after writing the above paragraphs, Apple presented the new firmware 3.0 for the iPhone and the iPod Touch (on 17. March 2009). It will be available for customers in summer 2009. Most interestingly, Apple now does support EAP-SIM, which allows the iPhone (not the iPod Touch) to seamlessly log on to hotspots (those that support EAP-SIM) using the credentials stored on the SIM card. Furthermore, Apple introduced Wi-Fi auto login and on-demand VPN. At the time of writing, it cannot be said what exactly this is and how it works. Supposedly it is some kind of automatic profile loading mechanism that allows the automatic provisioning of the iPhone when walking into the range of a hotspot. Another guess is that the browser acts as the traditional WISPr XML Smart Client, making the SMS one-time password procedure transparent to the customer. However, if the update will include automated 802.1X provisioning (resp. profile loading) is still unclear at the time of writing.

REFERENCES

[1] Black Hat - Briefings and Trainings., [Online] http://www.blackhat.com/.

[2] George Ou - ZD Net., Hamster plus Hotspot equals Web 2.0 meltdown! [Online] 2007. http://blogs.zdnet.com/Ou/?p=651.

[3] Sarah Granger., Social engineering reloaded. [Online] http://www.securityfocus.com/infocus/1860/2.

[4] Ethereal - Network Protocol Analyzer., http://ethereal.com/. [Online]

[5] Wireshark - The successor of Ethereal., http://www.wireshark.org/. [Online]

[6] AirSnarf - A rogue AP setup utility by The Shmoo Group., http://airsnarf.shmoo.com/. [Online]

[7] George Ou - ZD Net: "What exactly is this new DNS hack you keep hearing about?"., [Online] 5. 8 2004. http://blogs.zdnet.com/BTL/?p=280.

[8] Microsoft MSDN., Securing Public Wi-Fi Hotspots. [Online] http://msdn.microsoft.com/ en- us/library/ms818947.aspx.

[9] George Ou - ZD Net., You use my hotspot, I'll use your credit card. [Online] http://blogs.zdnet.com/Ou/?p=30.

[10] Wi-Fi Planet - News Oct 2004., T-Mobile's Hotspots Are Secured. [Online] http://www.wi-fiplanet.com/news/article.php/3417281.

[11] Boingo - Global Wi-Fi Roaming., [Online] http://www.boingo.com/.

[12] Apple iPhone., [Online] http://www.apple.com/iphone/.

[13] AutoWi-Fi Lite., [Online] http://subzero.eu/autowifi/AutoWiFi/Welcome.html.

[14] Devicescape Software Inc., Easy Wi-Fi. [Online] http://www.devicescape.com/.

[15] ShmooCon - The Hacker Convention (2008)., [Online] http://www.shmoocon.org /2008/.

[16] Microsoft Corporation., "WPA2 Update for Windows XP (KB893357)."

[17] Wireless Provisioning Services (WPS) Authoring Tool. [Online] Microsoft Download Center.

[18] Mircosoft Native Wi-Fi API (Microsoft Developer Network)., [Online] http://msdn.microsoft.com/en-us/library/ms706556(VS.85).aspx.

[19] Apple., iPhone Configuration Utility. [Online] http://www.apple.com/support/iphone/enterprise/.

[20] Apple iPhone Enterprise Deployment Guide (Fourth Edition), [Online] http://manuals.info.apple.com /en_US /Enterprise_Deployment_Guide.pdf.