Recent Issues with Wireless (802.11) Security

Recent Issues with Wireless (802.11) Security

1. Introduction

My goal for this project is to present recent information about developments within wireless security. When I say wireless in this paper I am referring to the 802.11 standard that is also known as Wi-Fi. I won't be mentioning topics that include Bluetooth, cell phones, RFID or other wireless categories.

Many media outlets have already covered the security problems with WEP, so those methods will only be mentioned to re-establish where wireless security was, to compare where it is today. It might have been thought that the end of WEP was the end of the security problems with wireless, but I will attempt to show that even as recent as November 2008, researchers are finding more problems with current standards.

The following sections will include some background information about how wireless security was done. Then the first recent issue I will talk about is the encryption method of TKIP within WPA and how the use of it creates an insecure environment. The second topic I will talk about is Jasager and how it exploits a problem with how some Wi-Fi software allows authentication with access points. Following that is a section describing some solutions for the security issues, and ways to protect a wireless connection from the different attacks.

2. Background

This is a brief overview of open wireless, WEP, WPA and their vulnerabilities. There are other issues with wireless but these are the main issues that have been shown in many articles, books, and websites.

2.1 Open Wireless

Open Wireless is an unencrypted wireless broadcast between computers to an access point. When a computer connects with an access point that is unencrypted all data is sent in the clear and can be read by anyone who is within range of the broadcast. Tools are available for an attacker to pick up these broadcasts and other tools such as Wireshark can read the actual packets down to the Hex level that are being broadcasted.

Beauchesne et al. (2007) mentions a program called Airpwn that will do packet injection. The interesting thing about Airpwn is that an attacker does not need to have administrator access to the access point or need to go through the trouble of setting up a Man-in-the-Middle Attack.

> "Airpwn works by sniffing one or more wireless networks, looking for user-supplied patterns of data sent from a client … to the access point (AP). If a pattern is detected, Airpwn injects a packet back to the client with user-supplied data that appears to come from the AP."

Once the user's computer finally does receive a packet from the access point, it disregards it "allowing Airpwn to control the server-side of the communication". Further discussion about Man-in-the-Middle attacks is in section 3.2 and packet injection is in section 3.2.3.

Connecting to an open wireless access point is not a good idea because of the security hazards it presents, which is why IEEE created WEP.

2.2 WEP

WEP stands for Wired Equivalency Policy, which is a simple method to try to keep the data from being read in the open. Its intention was not to ensure encryption, but to try to emulate a wired network. I don't think I can explain the weakness any better than by quoting Beauchesne et al. (2007).

> "WEP uses both the RC4 stream cipher—the same cipher used in Secure Sockets Layer (SSL) – and then an integrity check that is CRC-32. […] A 128-bit WEP key uses a 104-bit key size with the 24-bit initialization vector [(IV)][…]. The use of a stream cipher means that the same traffic key should not be used twice, and the 24-bit initialization vector is supposed to ensure that this never happens. Unfortunately, on busy networks, a 24-bit initialization vector

> is not long enough to prevent the key from being used twice, which allows a patient attacker the opportunity to eventually crack the WEP key."

According to Berghel and Uecker (2004), even with disabling the broadcast of an SSID (Service Set Identifier) and enabling WEP, the attacker can monitor current clients' traffic to find the weak IV and crack WEP with a tool called AirSnort. Beauchesne et al. (2007) give examples on the use of an alternate tool called Aircrack-ng because AirSnort was abandoned in 2004, and Aircrack-ng is being maintained by a community of people interested in seeing further development in creating tools to show the insecurities of wireless networks.

The SSID is what identifies one wireless network from another. Some Administrators try to hide the SSID to try to keep potential attackers out. It may keep the casual user looking for an internet connection out, but Johnny Cache and Vincent Liu (2007), state that disabling the SSID will not stop an attacker from using a passive scan on the network and discovering the SSID of the network. This is an example of security by obscurity, by which a system is trying to be secure by hiding the insecurities of the network. In general security by obscurity is a bad idea because hiding the security holes may keep attackers out for a time, but if the information is valuable enough, someone will try to find a way to get at it.

Netstumbler is one tool that can be put in passive mode and view if there is a network in range, displaying the MAC address of the access point but it won't display a SSID of the network. So an attacker can know that there is an access point within range, but more information is required before they can do anything. The Linux tool Kismet will listen to the legitimate client traffic and when the client joins the network it sends the SSID frame in a packet that is in plaintext, which Kismet will discover (Cache and Liu 2007).

But what if no new users connect to the access point? The best way to capture the authentication packets is to kick the user off and wait for them to re-authenticate to the access point. The only thing an attacker would need to do is send a de-authentication packet to the user that looks like it came from the access point. Cache and Liu (2007) say that a de-authentication "attack is effective regardless of the type of security the AP is using."

WEP has some fundamental flaws with how the connections are made "secure". With more and more methods of breaking WEP, it's hard to rationalize using WEP instead of better encryption standards. Hal Berghel and Jacob Uecker (2005) say it best; "The question isn't whether WEP can be broken, but how long it takes."

2.3 WPA

WPA is a certification protocol, not an encryption standard. When the IEEE noticed that WEP had a vulnerability, they started a group (called "i") to design another form of protection, this is also known as 802.11i. The IEEE is the Institute of Electrical and Electronics Engineers which is a non-profit organization and one of their responsibilities is to create standards for many technologies within the electronics field.

WPA or Wi-Fi Protected Access was released in 2003 (Ben Adida 2007). The first sets of standards were created to work on, at that time, current hardware. These new standards were CCMP and TKIP. CCMP is Counter Mode with Cipher Block Chaining Message Authentication Code Protocol, and uses AES for its encryption standard (Johnny Cache and Vincent Liu 2007). TKIP will be talked about in section 3.1.

One issue with WPA is that it allows for a Pre-Shared Key (PSK) to be under 20 characters. A PSK is a key or pass phrase that is shared by the access point and the user machine, which is used to verify access permission. Cache and Liu (2007) explain that in WPA the PSK and the SSID are computed to create a pairwise master key (PMK) and this is what is stored on the computer. This makes it more difficult for attackers to pre-create a hash file of all the possible PSKs that will work on all access points. According to Berghel and Uecker (2005), even after a hashing operation of the PSK that is done 4,096 times; a tool called coWPAtty can perform a dictionary attack on the authentication frames of the connection to recover the PSK. CoWPAtty will also allow the attacker to create a hash file with dictionary words for a specific SSID. Creating a pre-created hash file will speed up the time it takes for an attacker to crack the access point that is using a dictionary word as its PSK.

In general a PSK is not a problem within a home environment, but in a corporate setting with hundreds of computers and multiple locations this becomes a problem. For the PSK to remain secure, the System Administrator would need to install the PSK in every wireless device individually. It would not be reasonable to think that the PSK would remain secure by sending it out in an E-mail to all the employees for them to enter. The solution to this is to implement a RADIUS server, which will be discussed in section 4.4.

3. Recent Security issues

These next sections explain some other methods that researchers have recently discovered or created to show other aspects where wireless is weak. It's good that they expose these techniques, because even though it could allow for malicious individuals or groups to break the wireless, it helps the developers who create the standards work on better methods to protect future versions of the standard.

3.1 *TKIP within WPA*

A paper released by Martin Beck and Erik Tews (2008) exposes a flaw within TKIP while within WPA. TKIP stands for Temporal Key Integrity Protocol. TKIP was created to allow older hardware to use a better form of encryption than WEP and to keep the wireless connection more secure (Berghel 2005).

With older hardware in mind, TKIP still uses some features that WEP used, such as the stream cipher, a 4-byte CRC checksum, a RC4 stream cipher, and every packet sends a 32bit CRC32 checksum (called ICV). Using an attack method that was used with WEP, called the ChopChop attack, and knowing the IPv4 address of the network, the attacker can decrypt an ARP (Address Resolution Protocol) request. Johnny Cache and Vincent Liu (2007), state:

> "ChopChop works by systematically modifying an encrypted packet one byte at a time and replaying it to the AP. By monitoring if the AP accepts the modified packet, ChopChop can slowly decrypt any packet protected by WEP [in this case WPA], regardless of key or key size."

So according to Beck and Tews (2008), when the Attacker captures a packet "he truncates the packet by one byte, guesses [the last byte of the trailing checksum], corrects the checksum and sends the packet to the access point to find out if the guess was correct". "Packets with an incorrect checksum are silently discarded", but if the checksum is correct then "the access point will generate an error message". The attacker is then able to slowly increment the values of the checksum.

The difference between TKIP and WEP is that TKIP has some countermeasures in place to try to keep this from happening. In WEP the Attacker could increment the guessed checksum fairly quickly, but with TKIP the Attacker needs to wait 60 seconds before trying again, because of the countermeasures. Then to know the exact sender and receiver IP address the attacker uses last 12 bytes of the packet to compare with more guesses made by the attacker (Beck and Tews 2008).

The attacker manipulating the ARP traffic could cause problems when the attacker responds to the ARP request and sends a fake ARP reply with an incorrect address to the client, which could then re-route some traffic to a hazardous address. This is an example of ARP poisoning or ARP spoofing. ARP is a protocol used by many routers and Ethernet devices to find the hardware address of a device when it only knows the IP address. There are other protocols that may be used, but because of how simple ARP is, it is fairly common to see.

Even though the complete TKIP PSK is not recovered, this does show that partial data manipulation is possible and could potentially lead to a full PSK recovery, similar to the way that WEP was broken (Beck and Tews 2008).

3.2 *Jasager*

Jasager is a piece of software that is written to run on an OpenWrt device called a Fon. A Fon router is a small device that was originally created to provide wireless internet access. OpenWrt is an operating system for small hardware such as the Linksys WRT-54G wireless router and others including the Fon. OpenWrt provided a way for developers to write software without having to worry about the firmware. The reason the Fon was used is because of its small size and ability to be inconspicuous if left in the open.

Figure 1

Figure 1 shows the small size of the Fon Router that I was able to use for my tests with Jasager.

According to the program's author, Digi Ninja (2008), Jasager means "Yes Man" in German. When a computer stores a wireless network SSID and settings are configured in a way (such as "connect to these networks even if SSID is not broadcasting" within Windows), the computer will send out a beacon broadcast asking a wireless access point within range using that beacon if it belongs to that SSID. When Jasager receives that beacon broadcast, it will respond to the beacon that it is the owner of that SSID. This makes the user computer think that the Fon/Jasager is the desired access point. The DHCP and DNS settings for the users computer are controlled by the Attackers computer.

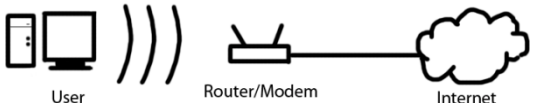

Figure 2

Figure 2 is showing the standard method of connecting to a wireless access point and to the internet. The user's traffic is getting sent wirelessly to the router/modem which is also an access point. This does not show RADIUS servers, which will be talked about in section 4.4, or other switches and routers that may be in place at a corporate environment, but the idea is basically the same.

I was able to purchase a Fon online and the process of installing Jasager took about 2 hours. I then configured my laptop with internet connection sharing and hooked the Fon to the laptop and, created a test network with several computers. With the operating systems Windows XP, Windows 7 beta and Mac OS X I found the quickest way for the clients to connect through Jasager was for the clients to try to connect to an access point they have not connected to before. Only with Windows XP did the setting "connect to these networks even if SSID is not broadcasting" turned on allow for the computer to connect to Jasager. The other operating systems only connected to it, if the access point was new and didn't have any settings stored.

When some computers try to access a SSID that it knows about in range, it doesn't always send out a beacon request, which makes it difficult for an Attacker to force the user to connect to Jasager.

Once there is at least one connected user then the next part of the attacks can take place.

When viewing the configuration page of Jasager, black-listing or white-listing SSIDs are a few different options that Jasager allows. White-listing is useful if the Attacker only wants to obtain users that are trying to connect to the "Coffee_Shop" SSID. While black-listing is to attain anyone but the people trying to connect to that SSID. The Attacker can also filter out MAC addresses. When a user computer has connected, there is a table that lists the 'date', 'SSID', 'IP', and 'MAC addresses'.

Darren Kitchen (2008) mentioned that because of the small size of the Fon, it would be possible to create a battery pack, and maybe attach another Fon to the Jasager/Fon to forward any intercepted traffic to another location. Another option would be to connect the Jasager/Fon to a Cell Phone and do the same. Properly hidden these options would create a small, portable, and automated MITM device.

3.2.1 Man-in-the-Middle Attacks

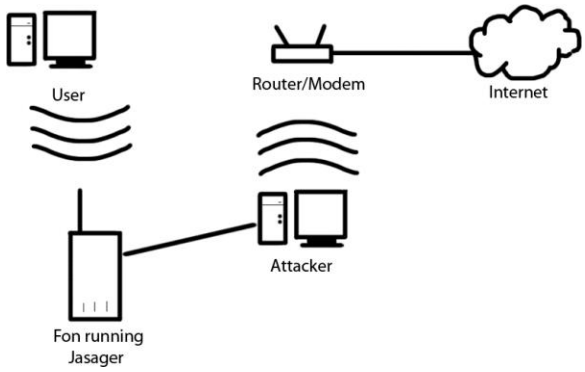

Figure 3

Figure 3 is showing how the Fon running Jasager re-maps the network to push the user's traffic through the Attacker's computer. The attacker then forwards the traffic to the correct location such as the router/modem. In a public setting the attacker will most likely be connected wirelessly, but can also be hardwired to the router. This is an example of how a Man-in-the-Middle Attack is created.

A MITM attack is what Jasager was designed to create. The Fon router is then configured to bridged mode and connected to the wired NIC (Network Interface Card) on the Attackers Computer. The Wi-Fi NIC on the Attackers computer is set to internet connection sharing, connected to the actual access point, this will allow for the internet traffic to be sent through the Attacker's Computer and out to the Internet. While the user's traffic is going though the Attacker's Computer it is possible to capture and/or manipulate the traffic.

The following sections describe a few examples for what is possible in a situation where a MITM attack is taking place.

3.2.2 Session hijacking

Session hijacking is the process of stealing someone else's Internet session. According to Ben Adida (2007), "the server assigns the browser a token, and the browser sends this token back to that specific server on every subsequent request". So a session is a static token or cookie that gets sent back to a server every time after authentication, which is where the server gives the user the token. Adida (2007) also states:

> "A static token sent over a plaintext channel is obviously insecure: a network eavesdropper can easily read this token and replay it to tap into the victim's session, effectively impersonating the user for the length of the session. Interestingly, web sessions have not evolved much since the first days of web cookies: they remain quite vulnerable to eavesdropping".

Web sessions are used as an alternative to SSL (Secure Socket Layer) to verify the user. Because of the encryption hardware overhead and the extra bandwidth used the cost for using SSL is high.

Darren Kitchen (2008), showed how a tool called Ferret looks at the connection traffic, for cookies coming across the wire. Another program called Hamster that piggybacks off Ferret offers a point and click option to hijack a session. A link is displayed for a site that it has captured. One site that is known to be vulnerable is the non https (https is a SSL connection over http) G-mail web page. With the Hamster and Ferret toolset, a user connects to G-mail, a link gets displayed on the attacker's Hamster page that allows them to click on the link. Once the link is clicked, the attacker is automatically brought to G-mail as if they had logged onto G-mail

themselves with the user's login information. From there the attacker can read, send, or delete anything that they want on that account. When a session is hijacked almost anything can be done when they are impersonating the user.

Hamster and Ferret were created by security researchers Robert Graham and David Maynor (2009) at Errata Security and released it during a DefCon Conference August 5th 2007. Defcon is a hacker/security researcher conference that is held every year in Las Vegas.

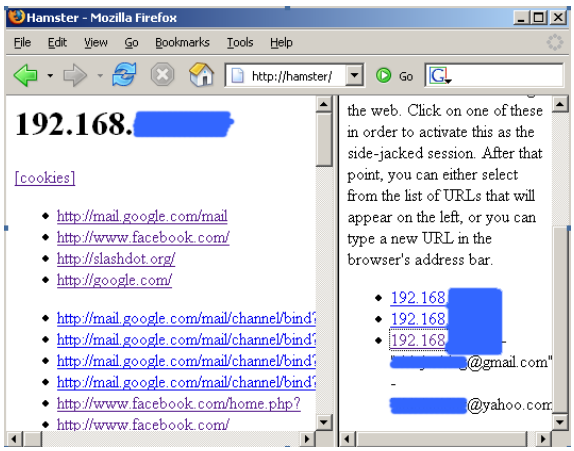

Figure 4

Figure 4 shows the web browser connecting to Hamster on the Attackers computer. The right part of the window shows the different clients that have connected to through Ferret. Once one of those has been clicked, the left hand side shows the cookies that have been captured. Clicking on one of those links will begin to hijack the connection.

Another method, that is more difficult, but allows for more freedom in capturing the sessions, is to use Wireshark to capture the traffic and a Firefox Add-on called Add N Edit Cookies to do what the name implies (Graham and Maynor 2009). These tools are free to use, and Wireshark is very powerful.

During the traffic capture, the first thing the Attacker is going to do is to apply a filter of the traffic for "http" within Wireshark. They can then look through the traffic to see if there is anything interesting such as finding a host for "mail.google.com". Once something has been found another filter can be added to look for all the http traffic to mail.google.com. This filter is "http.host == 'mail.google.com'". The Attacker will find the POST and find a section that mentions a cookie. In the example in Figure 5, the cookie has been truncated; the reason for this is that a cookie only has a limited size. But with Wireshark it's not a problem because the Attacker can then follow the TCP stream and view the entire cookie. Using this information, they can enter it into the Add N Edit Cookies add-on and hijack the session by traveling to the same pages the user did (Graham and Maynor 2009).

This process using Wireshark is not an easy one, because it seems that Google and other site use multiple cookies for authentication and other items, it's hard to know exactly which one does what. During my tests, I was able use Hamster and Ferret to hijack a G-mail session that I had connected to on another computer. But when I tried to use the Firefox add-on it was difficult for me to figure out exactly which cookie(s) was the correct one(s) that Google uses to verify users.

Figure 5 is showing Wireshark with the http filter for mail.google.com in place. Also marked in red is where you would find the Host, marked in green is the truncated Cookie, and marked in blue is the Post that the Attacker looks for a Cookie.

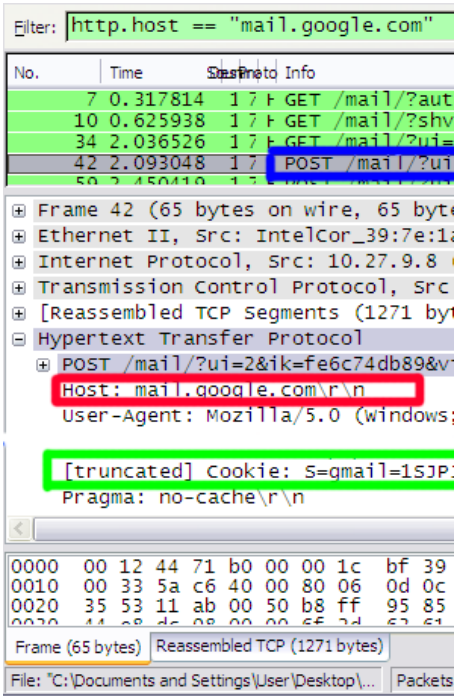

Figure 5

3.2.3 Packet Injection

Packet injection is the process of capturing packets, analyzing them and either substituting packets with custom packets or modifying packets to produce a desired result. It can be something as simple as whenever a request for an image is made, that image address is replaced with something different. Another more serious example of this is whenever a computer requests a website such as http://www.wellsfargo.com, the DNS record for this site could be replaced with an IP for a fake Wells Fargo page that looks identical. The unsuspecting user could enter their username and password, and that data could be recorded by the Attacker or by the fake site. This is an example of phishing.

Beauchesne et al (2007) mentions a framework called Netfilter, which is the packet filtering framework that is built into the Linux Operating System kernel. Its original intention was to provide a way for firewalls and other networking software to operate. Python code can be written to use the framework to modify the packets on the fly. The first example was to change the text of the Google homepage to be other characters. The second example was to exploit a vulnerability in Skype and not allow the Skype client to communicate with the server.

Using the iptable command in Linux the Attacker will push the traffic through the custom Python program using a technique called transparent proxying (Beauchesne et. al. 2007). Transparent proxying was created to allow for redirection of a connection without the user being aware. It comes in handy when traffic needs to be rerouted to a web cache in a situation where a web site needs to reduce bandwidth.

4. Solutions

This is a brief overview of some of the possible solutions to ensure data security. Some of the solutions permit the user to make changes to their networking habits without too much difficulty and other solutions require more technical knowledge that a user may need help with by a System Administrator or another form of technical support, would need to enable and configure.

4.1 Tunnels

A Virtual Private Network is one example of a tunnel, in which a VPN is a network that allows for a secure connection to another computer, creating a network, without the need for the other computer being physically connected to the same network. Cache and Liu (2007) state:

> "VPNs work by encapsulating a lower-layer protocol inside a higher layer, for example Ethernet over TCP… By embedding a lower-layer protocol in a higher layer you can have the most of the security associated with link layer encryption, along with the convenience of encrypting at a higher layer in the protocol stack."

A program called Hamachi will allow for a quick and easy VNP to be created and joined using Windows, OS X, and/or Linux. It will allow for anyone to create a network that is protected by a unique network name and a semi-unlimited character password. Once other clients have attached themselves to the network then all communication that goes on between the users is encrypted in multiple ways to ensure that the only computers that know what's really going on are the ones that are supposed to know (LogMeIn 2009).

Figure 6 is showing the GUI that is included with the windows version of Hamachi and shows the different computers that are connected to the VPN network. If a user wanted to get a file from one of these connected computers, on a Windows machine they would just try to connect to them in the same way that they normally would; within the network neighborhood and a computer called 8183_test would show up.

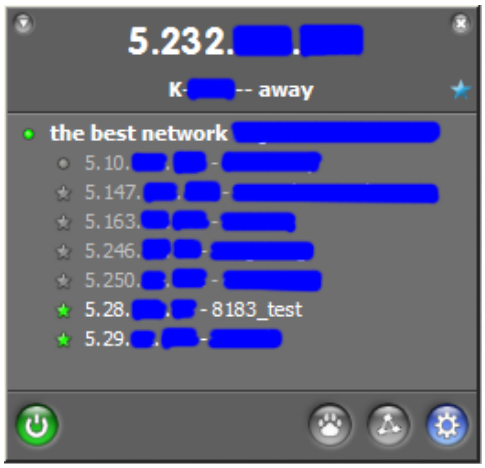

Figure 6

Even if the wireless information is sent in the open, a secure VPN can allow the user to use another computer on the VPN to proxy the traffic through. Anyone sniffing the traffic will not be able to decrypt the data being transferred on the wireless.

Another example of a tunnel would be to connect to another computer to do the browsing with a VNC, or Virtual Network Computing. This allows for a user to control a remote computer, with only keyboard and mouse actions being transmitted one way, and a picture of what's happening on the remote computer's screen being sent back to the user. One benefit of this would be to keep e-mail or other private documents only at a trusted location, with only a graphical representation of the text being sent back to the user.

UltraVNC or TightVNC are examples of free VNC clients and server applications. These require for a user to have a desktop computer that is on running the VNC server software, and for them to know their public IP address, and open the port on the router and forward it to the computer. Doing all those configurations may be difficult for the average user. PC Anywhere is not free but allows a user to install the software, and it will keep track and walk the user through the more technical settings. Citrix makes products that will enable remote and virtual computing systems so the user doesn't need to own a computer on the other end, they provide a virtual computer for a user to connect to.

Not all VNC programs are known for their strong passwords but using VNC along with a VPN would create one of the best ways to keep data secure on not only a wireless network but any network. By only passing the keyboard and mouse movements and receiving images through an encrypted channel, it would be fairly difficult for anyone to find out what the user was doing.

4.2 WPA with AES Encryption

WPA using the AES encryption is also known as WPA2. This is the second set of standards produced after WPA that is replacing TKIP with AES (Cache and Liu 2007). Maurer (2003), states that for a "128-bit AES algorithm" to be broken, it would take about $1.5x10^{14}$ years.

> "The AES algorithm is based on simple mathematical transformations whose inverses are difficult to compute without the key. The algorithm has 4 basic transformations that are repeated 10, 12, or 14 times, depending on what key size is being used. Repeating the transformations multiple times helps to ensure that breaking the algorithm will be more difficult to compute than trying every single key. Currently, is it believed that no simplification of the transformations will allow a shortcut to break the AES algorithm. This belief is held because the transforms are simple and allow thorough analysis."

So the math behind AES is fairly simple and there have yet to be any issues with the possibility of breaking the algorithm.

According to Committee on National Security Systems (2003) a policy released in June 2003 states that the NSA allows for 128-AES to be used to protect information at the SECRET level, but TOP SECRET requires 192-AES or 256-AES. This allows home and corporate users to trust the security behind this technology.

Most new Wi-Fi cards have the ability to connect to new access points using WPA2 which support AES as the encryption standard. Older laptops that have a built in Wi-Fi cards should be replaced or upgraded by purchasing a PCMCIA or USB Wi-Fi adapter that supports the AES standard and newer access points should be deployed that also support AES.

Another thing to consider when choosing to use AES, it is only as strong as the passphrase being

used. For true security a completely random passphrase with upper and lower letter, numbers and, if possible, symbols should be used. It may be harder to remember, or to deploy in a corporate environment but this is where a RADIUS server comes in handy. RADIUS will be talked about in section 4.4.

A good place to receive a pseudo-randomly generated key is from on security researcher Steve Gibson's (2009) site at https://www.grc.com/passwords.htm. This key generator provides a decent amount of protection against people intercepting what the key values are, by implementing SSL and other features to keep the key unique to every refresh of the page. Further information about the key generator can be found at the link.

4.3 Watch Access Point Names

When a computer user is trying to access a wireless network that is not completely known to them, the user should watch to ensure that the SSID they are connected to is not one that is at another location such as at their house or one they connect to at their work. If a user has a home wireless network SSID is called "Or4ng34421", the user should not see an SSID at the airport called "Or4ng34421", if they do then this could be an indication that there is a Jasager enabled access point around in that location.

Another option would be to disable the setting within Windows to automatically connect to non-preferred networks within range. This will help minimize the risk, but not stop it completely.

The last way to watch for Jasager, I found a solution written by Ryan Pfleghaar (2009). He talks about a method that can help remind users that Jasager might be nearby.

> "By creating an access point in your preferred networks list titled 'Jasager Detected' and pushing it to the highest priority you will now know when a Jasager device is being used. If you ever scan to see what networks are available and see the access point 'Jasager Detected' then it's a good idea to check your email at home instead of the coffee shop or airport."

Jasager is hard to protect against because it requires the user to always remember the possibility of Jasager being in use at any public location. It might be possible to write some software that can help remind a user that Jasager might be in an area, but so far I am unaware of anything that will do this.

Owners of coffee shops or other public Wi-Fi hotspots should also be aware or the possibility of a rogue access point. If they want to remain free of potential legal issues, it would be a good idea for them to do periodical checks for Jasager type devices, both physically and using software.

4.4 RADIUS Authentication

RADIUS stands for Remote Access Dial-In User Service. This provides authentication, authorization, and accounting to a network resource. RADIUS is additional software that maintains a record of who is accessing (if they are authorized) the wireless network and can easily disallow a specific user without needing to change a PSK (Lockhart 2006).

> "RADIUS was originally defined to solve the username/password database problem when using [other protocols]… As far as wireless security is concerned, RADIUS is really just a crunchy old protocol being used to transport EAP packets from an access point to an authentication server…" (Cache and Liu 2007).

The RADIUS server has the ability to maintain the authentication to the wireless network instead of having the access point do the authentication.

> Cache and Liu state: "… when a user plugs into an Ethernet port protected by 802.1X, the only thing the user is allowed to send at first are packets related to authentication… Once a user successfully authenticates, she is allowed to transmit normal Ethernet data packets"

When a user is trying to connect through wireless this same example is can be used. The 802.1X is a standard created by IEEE that is using a similar protocol that is maintained by another

organization. Their intention was to bring that protocol's authentication method to everyday users (Cache Liu 2007).

In a corporate environment, RADIUS allows for System Administrators to control multiple access points with the way people authenticate to them. A typical RADIUS will prompt the user with a dialog box for a username and password (Cache and Liu 2007). This removes the burden of having to create and remember a random 128 bit key that a System Administrator will need to install on all the wireless access points and other wireless devices.

Another benefit to using RADIUS, is that it allows for a CA (Certificate Authority) to create certificates to use. Once both the RADIUS server and the user computer have the trusted certificates, it grants the user computers to validate the RADIUS server's certificate (Cache and Liu 2007).

5. Conclusion

Wireless networks make mobile computing extremely useful and handy to the everyday user, but without the correct type of authentication and security measures taken, they can create many problems that can lead to compromised data. New vulnerabilities are found every day, so it is impossible to say that the solutions presented will maintain their usefulness, but I hope that by writing this report people can know what to look out for and how they might be able to protect themselves.

To recap some of the vulnerabilities with Wi-Fi, the first is a general idea to not use WEP anymore. There are more and more problems being found that break WEP faster than the previous method. WPA is only as strong as the PSK if TKIP must be used because of older hardware limitations. WPA with AES encryption should be used with a random 64 character string in a home environment. Within a corporate environment, a RADIUS server should be implemented to reduce the chance of a PSK being leaked, and reduce the effort in maintaining a PSK in all the wireless devices. When anyone is at a public location accessing the internet, watching the name of the access point and planning ahead by using a VPN program such as Hamachi to tunnel traffic through will keep the users data from eyes that are not supposed to see the data.

IEEE and other organizations are continually working on new and better techniques to ensure that data will not be compromised. There are also security researchers and hackers that are continually trying to find methods that will break these techniques. While security researchers are doing it to help the industry as a whole, some hackers tend to do it for selfish reasons.

These security issues here were only dealing with wireless and to bypass most of these problems, a solution would be to plug the computer in to a network port, connected directly to a switch. Limiting the broadcast range of the network will also help control who has access to the network. Creating separate VLANs for unauthenticated clients will help reduce the exposure of the network resources available to an attacker. Unfortunately it is impossible to make a system 100% safe, but in the end, all it takes is one user to leave a post-it-note on their computer with their username and password on it to unravel even the most secure systems. So to end this I'd like to include a XKCD (2009) comic, in Figure 7, that somewhat explains the weakest link in computer security.

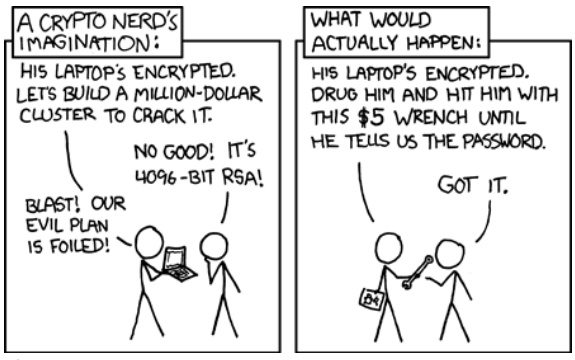

Figure 7