UNSAFE AT ANY SSID: Wireless Hotspot (In)Security

By Christopher Byrd – ISSA member, Saint Louis, USA Chapter

The author discusses common approaches used to secure wireless hotspots, pointing out their weaknesses, suggesting methods for secure hotspot deployment, and giving a glimpse into what the future of wireless hotspot security could look like.

Join the Discussion
Connect

Abstract

The popularity of wireless hotspots and their lack of security pose a challenge for companies and individuals. While a number of current solutions exist, they all have shortcomings. Methods to secure open wireless networks have been proposed, but they need vendor and customer support to succeed. This article will discuss common approaches used to secure wireless hotspots and their weaknesses. It includes suggestions for secure hotspot use, and a glimpse into what the future of wireless hotspot security could look like.

Since the opening of the 2011 Missouri legislative session, four Missouri legislators and a staffer have had their Facebook accounts hijacked after using the House's free open wireless hotspot available at the Missouri State Capitol building.[1] As an example, one state representative's Facebook status was updated to "I love lobbyist! [sic] All the free food and stuff you get. This job is awesome!" While the exact cause of the breach has not been confirmed, it is highly likely that this was the work of an attacker taking advantage of the insecure open wireless network.

The rise of hotspots

The increase in the number of hotspots has come from an overall rise in mobility. Increased use of consumer technology in business has led to a more mobile workforce. According to a Wi-Fi Alliance survey, 62% of respondents said they would look for Wi-Fi for their next cellphone.[2] Mobility brings with it a desire for connectivity and a corresponding explosion in the use of wireless hotspots. Once these hotspots were mostly limited to homes and coffee shops, but increasingly they are becoming a business requirement to support everything from mobile workers to vendors and customers.

To this end businesses deploy wireless hotspots for a variety of audiences. Some companies are providing hospitality to customers, others providing a vital link for suppliers to improve their service levels. A few municipalities have even deployed wireless networks that provide coverage for their residents and businesses.

Increased risks

Along with the *advantages* of wireless mobility come additional *risks*. As a side effect of deploying a new wireless hotspot network across a company's operations, the company in effect could be considered a nationwide or international

1 Berg, Rebecca (2011, February, 7). "Facebook accounts of four Mo. legislators are hacked," *St. Louis Post Dispatch*. Retrieved February 19, 2011 from http://www.stltoday.com/news/local/govt-and-politics/article_b6e40444-1414-563a-bf25-d28869ca4f0a.html.

2 "Wi-Fi® expands as the center of leading-edge technologies in 2011," Wi-Fi Alliance®. Retrieved February 19, 2011 from http://www.wi-fi.org/news_articles.php?f=media_news&news_id=1035.

Popular Wireless Auditing, Attack Tools

NAME	DESCRIPTION	URL
Aircrack-ng	Aircrack-ng is a collection of security tools for auditing wireless networks. Tools include sniffing, MitM, and encryption cracking.	http://www.aircrack-ng.org
Airpwn	Software for performing wireless injection attacks.	http://airpwn.sourceforge.net
Firesheep	Extension to the Firefox web browser for capturing user sessions to popular websites such as Facebook and Twitter.	http://codebutler.com/firesheep
Iodine	Tool for bypassing captive portals and other Internet filters by tunneling traffic through DNS.	http://code.kryo.se/iodine
Kismet	Kismet is a wireless audit tool for finding and sniffing wireless networks, and includes some intrusion detection features.	http://www.kismetwireless.net
sslstrip	Tool for demonstrating attacks against HTTPS-protected websites.	http://www.thoughtcrime.org/software/sslstrip
Wireshark	General purpose network analyzer for Windows and Unix. Wireshark includes extensive wireless sniffing and decoding support.	http://www.wireshark.org

Table 1: Popular Wireless Auditing, Attack Tools

Internet Service Provider (ISP). However, unlike a traditional ISP, customers may be effectively anonymous and not have agreements in advance governing acceptable use. Also unlike an ISP's customer base, consumers of wireless hotspots are usually mobile. Wireless guests are often only connected for minutes or hours, making them difficult to track down and rendering disconnects largely ineffective. When deploying a hotspot network, businesses have good reason to be concerned about misuse of those wireless networks. Open wireless hotspots can provide a safe haven for people seeking anonymity or just additional bandwidth. Hotspot clients can be infested with malware, be controlled by a botnet, and be used to send spam. If not properly controlled, responding to these incidents can become time consuming for information security departments.

For guests using these hotspots, the security risks are even worse. Connecting to an insecure wireless hotspot can be analogous to connecting a projector to your laptop in that you are broadcasting everything you do. Anyone nearby with the right tools can capture and view your session. Attackers can also leverage insecure wireless connections to compromise a guest's computer.

Controlling access with captive portals

Many companies employ captive portal technology to control access. A captive portal redirects user traffic to a webpage where the user may be asked to agree to a terms of service, authenticate, and possibly pay for service before being allowed access to the Internet. Captive portal technology is especially popular with hotspots in retail settings. While a captive portal does provide a barrier to unauthorized access, some can be bypassed by attackers.

Bypassing captive portals

There are two common methods of bypassing captive portals. The first is by tunneling traffic over an otherwise required network protocol such as Domain Name System (DNS) or Internet Control Message Protocol (ICMP). For example, the tool *Iodine* (see Table 1 for tools list) is an open source application that enables unauthorized users to bypass some captive portals by encapsulating traffic into DNS requests. If hotspot operators blocked DNS, then legitimate clients would not be able to resolve names during the authorization process. It is also possible to bypass some captive portals by sniffing for an authorized user's Media Access Control (MAC) address and Internet Protocol (IP) address and configuring an unauthorized system to the same settings, thereby assuming the legitimate system's identity. This can be especially problematic for hotspot providers when an unauthorized user gains free access, and for users of hotspots that charge by time, because they may be billed based on an attacker's activity.

Connecting to an insecure wireless hotspot can be analogous to connecting a projector to your laptop in that you are broadcasting everything you do.

Attacks against wireless

All it takes to capture or view wireless traffic on an unencrypted wireless network is a regular off-the-shelf wireless card installed in a laptop or plugged in with Universal Serial Bus (USB). Most wireless cards sold support *monitor mode*, which allows them to listen to all wireless traffic. Often an attacker will start by scanning for wireless networks, configuring the wireless card to automatically hop through all channels. Once a target is identified, the wireless card is then configured to capture all traffic on the given channel. Although wireless cards can only receive one channel at a time,

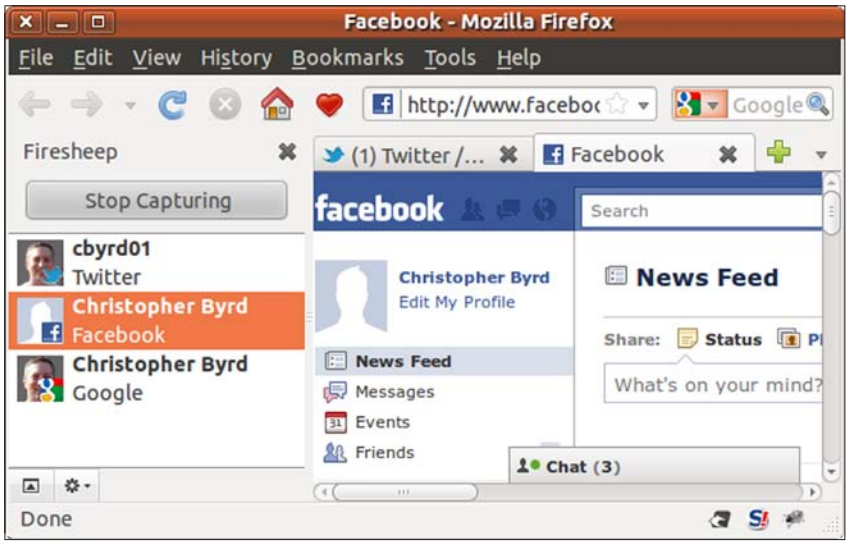

Figure 1 – Firesheep capturing wireless session

similar to tuning a radio to a particular station, an attacker can attach multiple cards to capture several channels simultaneously. Wireless traffic sniffing is easy using operating systems like Linux and Mac OS X, where there are plenty of tools such as *Wireshark* (see Table 1) to capture and display this traffic. Until recently it required specialized hardware to utilize monitor mode on Windows, but Microsoft introduced a new driver model called Network Driver Interface Specification (NDIS) 6 that supports it, starting with Windows Vista.

Once an attacker is able to capture the wireless traffic, he can go beyond just viewing by parsing content and extracting useful data. If someone captures a username and password for a site over an unencrypted hotspot, he can simply log in as the victimized user. Many sites secure the login page using Hypertext Transfer Protocol Secure (HTTPS). However, even these sites can be targeted through a technique called sidejacking, which exploits a flaw in how some sites handle session tracking.

Sidejacking

After login, websites track web users through the use of session cookies, unique strings sent from the website to web browsers on login. The browsers then include these session cookies on each subsequent connection to establish user identity. If web requests after the login page are not encrypted with HTTPS, an attacker can capture this session cookie and use it to gain access to the website as the "authorized" user. This is sidejacking. *Firesheep* (see Table 1) is a new plug-in for the popular web browser Firefox that automates the process of sidejacking.

Figure 1 is an example of Firesheep in use against an unencrypted wireless hotspot. In the interface there is a single button to start capturing credentials. Once capturing, Firesheep identifies session cookies that can be used and exploited and displays them as a list of users and sites in a sidebar. Once a session shows up in the list, breaking into that session is as simple as clicking on which one the attacker wants to use. By default Firesheep can allow an attacker to easily access a victim's Facebook, Twitter, webmail, and other accounts, and it is possible to add support for additional sites as desired.

At least in part due to the popularity of Firesheep, some additional websites have begun adding full-time HTTPS as an option for their users. Unfortunately even for sites that are normally encrypted, there are still ways to attack these sites through Man-in-the-Middle (MitM) attacks. There are two ways to launch a MitM attack against open wireless networks. The first is by using raw packet injection to send responses to the client faster than the legitimate site responds. This allows an attacker to send his own responses to requests made from the client. For example, using a tool like *airpwn* (see Table 1) an attacker can listen for requests to HTTP sites and send responses to the client, allowing the attacker to control the content of the requested page. This can be used for everything from pranks such as replacing images on webpages, to injecting JavaScript code running in a user's browser. The second more popular method is called *evil twin attacks*, whereby the attacker runs a rogue hotspot with the same name as the target network. The attacker takes advantage of the selection process where clients typically select an access point with the strongest signal. Attackers can have the advantage of being physically closer to the clients or transmitting at higher power levels than the legitimate access points, which generally do not transmit at a higher power than necessary to avoid interference.

Once clients are connected to the evil twin access point, the attacker typically forwards their traffic on to the original access point, but as they are in the middle of the connection they can modify both requests and responses. For example, by using a tool called *sslstrip* (see Table 1) an attacker can rewrite traffic to the client side of a connection, changing HTTPS sites to HTTP, while connecting with HTTPS to the site itself. This requires that the user not notice that a site is being delivered HTTP instead of HTTPS, but allows the attacker to capture interaction with a site that would have otherwise been encrypted. To help fool the user, sslstrip can replace the site's icon to an image of a lock, which people often associate with a secure connection. Other tools such as *evilgrade* take advantage of flaws in update tools by waiting for requests to update popular software and sending attacker tools as upgrades instead.

Current wireless protection methods

To protect against these and other attacks, some hotspots use Wireless Protected Access (WPA) or WPA2 encryption with a Pre-Shared Key (PSK).[3] WPA2-PSK is an encryption and key

3 Benton, Kevin (2010, April 18). "The Evolution of 802.11 Wireless Security." Retrieved February 19, 2011, from http://itffroc.org/pubs/benton_wireless.pdf.

derivation method popular in home wireless networks. This approach requires some type of out-of-band communication of the shared key, which is often accomplished in business with signage or in-person communication. However, the use of WPA2-PSK in hotspot networks suffers from a fatal flaw. Because the encryption of the network is derived from a key that is shared by necessity, an attacker who has the key can decrypt traffic on the wireless network, including traffic to and from other clients. Having a known shared key also facilitates evil twin attacks where the imposter access point uses the same shared key as the original.

Because of these challenges some hotspots, most notably those at conferences and trade shows, have begun offering wireless security based on WPA2-Enterprise. In enterprise mode, also known as 802.11i or Robust Security Networks (RSN), a client uses individual credentials to authenticate to the network. The types of credentials depend on the Extensible Authentication Protocol (EAP) type. The most popular is the use of Protected EAP, Microsoft Challenge Handshake Authentication Protocol (EAP-PEAPv0/MS-CHAPv2.) This protocol employs a simple username and password for authentication. As with WPA2-PSK, it is still necessary to find a way to communicate credentials to users before they connect, and it can be difficult to support due to increased time for configuration.

Consumers and business users alike have been turning to the use of Virtual Private Network (VPN) solutions over wireless networks to help stop wireless attacks. A number of consumer-oriented VPN services are now available for users concerned about these attacks. While VPNs do provide an added level of security, there are two viable attack methods against them. The first is to take advantage of the window of time between when the client connects to the wireless network and when the VPN is established. During this time the attacker can use the methods previously mentioned to compromise the client. Once compromised the attacker can continue control after VPN establishment. The second is a more brute force method. By employing MitM techniques the attacker can prevent the client from connecting to the VPN server. When faced with a failure to connect, users may opt to continue to use the insecure wireless network anyway.

Future solutions

There have been multiple attempts to come up with solutions. In a 2007 article for ZDNet,[4] George Ou suggested that hotspots authenticate with EAP-PEAPv0/MS-CHAPv2 using a well-known (or even blank) username and password combination. EAP-PEAP uses the public/private key encryption-based protocol Transport Layer Security (TLS) to establish a secure channel with the server. Once authentication is complete EAP-PEAP provides a secure method for the delivery of a per-session key. Because each individual connection is secured with its own key, it would not be possible to decrypt another user's session. This is true even for someone knowing the credentials.

While his method is more secure than the alternatives, it still requires the user to know credentials before connecting. Another possible solution exists that would secure communications without requiring any authentication using current wireless standards. Internet Engineering Task Force (IETF) Request For Comments (RFC) document RFC 5216, "The EAP-TLS Authentication Protocol"[5] defines EAP-TLS, which is similar technology to that used to secure HTTPS websites and many other services. This RFC says, in part:

> "While the EAP server SHOULD require peer authentica-

4 Ou, George (2007, July, 18). "A secure Wireless LAN hotspot for anonymous users," ZDNet. Retrieved February 19, 2011, from http://www.zdnet.com/blog/ou/a-secure-wireless-lan-hotspot-for-anonymous-users/587.

5 Simon, D., Aboba, B., Hurst, R. (2008, March). "The EAP-TLS Authentication Protocol." Retrieved February 19, 2011 from http://tools.ietf.org/search/rfc5216.

tion, this is not mandatory, since there are circumstances in which peer authentication will not be needed."

This solution, which I refer to as Open Secure Wireless, allows the establishment of an individual secure connection without the need for a client to know credentials beforehand. Unfortunately, most if not all popular authentication servers and wireless supplicants treat the client authentication as mandatory. However, I have been able to modify the source of *hostapd*, an open source authentication server, and configure Windows and other popular clients to connect to it successfully. Without code changes Windows and other supplicants believe that they have to have a certificate before attempting to connect, although that certificate is never used in the connection. I have published further information on my blog[6] about the specifics of how vendor support for EAP-TLS without client authentication may provide a solution to many hotspot security issues. This is a possible, but not currently practical, solution for providing secure open wireless access. With vendor support this could become a reality using current standards.

There are also other efforts underway to address the security of wireless hotspots. The standards body that developed the foundational wireless communication protocols, the Institute of Electrical and Electronics Engineers (IEEE), may be close to ratification of 802.11u, a complex amendment to 802.11 that would add a significant number of features for devices that support the new standard. These potentially include enhancements to network discovery, selection, and information transfer functions before a client connects to a wireless network. There is some information that appears to indicate 802.11u will provide support for RSN networks that do not require client credentials, but this work has not been completed and IEEE working drafts and standards are not made public.

6 Byrd, Christopher (2010, May, 19). "Open Secure Wireless.," RioSec. Retrieved February 19, 2011, from http://riosec.com/open-secure-wireless.

Conclusion

The use of wireless hotspots continues to rise, driven by increased mobility. Unfortunately, it is difficult using existing protocols to protect the confidentiality, integrity, and availability of wireless hotspots using today's technology. Open secure wireless is a technical possibility, and other standards are already under way to make this a reality.

Without a secure alternative, consumers of hotspot services should treat each hotspot as a hostile, compromised network. Before connecting to these networks, users should follow the usual advice of keeping systems up to date with patches, making sure antivirus software is installed and working, and using a host-based firewall. If you have a VPN service available, make sure it is connected before doing anything over insecure wireless.

For businesses offering hotspot services, there are no good solutions for securing hotspots today. In the meantime consider using a combination of the technologies discussed to protect public wireless networks. Companies should demand that their vendors deliver more secure solutions using existing standards, and work with standards bodies to develop new security protocols. This will lead to improved security for wireless users in the future.

About the Author

Christopher Byrd is an Information Security Architect residing in Saint Louis, Missouri. He has earned the Bachelor of Science in IT-Security from Western Governor's University, is a Certified Information Systems Security Professional (CISSP), GIAC Certified Incident Handler, and is GIAC Assessing Wireless Networks (GAWN) certified. He is the author of the RioSec security blog at http://www.riosec.com and can be contacted at chris@riosec.com.