

# Open Secure Wireless

Christopher Byrd

May 5<sup>th</sup>, 2010

## Abstract

Current open wireless networks expose users and operators to unnecessary risk. Open Secure Wireless (OSW) networks are possible today using the existing EAP-TLS standard and minor modifications to authentication servers. Current wireless supplicants work with these modified authentication servers with specific configurations, and with minor modifications would provide a complete solution.

## Background

Imagine if you had to enter a password or other credentials just to connect to a secure (HTTPS) web site. It would be a difficult proposition, as you would need credentials to get to the site, but couldn't get to the site to register or find out the credentials without already having credentials. This would likely have dramatically reduced the adoption of SSL/TLS, as it would be unwieldy to use.

Although client authentication for secure public web sites is not required and is not in widespread use, it is currently the only available option for secure public wireless networks. Public wireless providers therefore have to choose between offering unsecure open wireless and requiring guests to authenticate **before** they can access the network.

## Current Solutions

### Open Unsecure Wireless

Open, unsecured wireless providers should be concerned with potential liability from unknown clients, such as becoming a hotspot for spammers. Further, many providers do not like the thought of potential inappropriate use of their networks brought about by the anonymity of clients.

Clients themselves face a number of risks accessing open, unsecure wireless networks. Attacks against wireless clients is an active area of research, and attacks such as session hijacking, sensitive data sniffing, and other man in the middle attacks are increasing. For example, evilgrade attacks allow an attacker to compromise systems through the automatic update functionality of clients, often before a secure VPN connection can be established. Even secure web sites using SSL rely in part on a secure connection, as unsecure wireless can allow attackers to use tools like SSLsniff and SSLstrip to subvert unsuspecting users.

### Open Wireless with Captive Portal

Many hotspot operators currently have users connect to an open wireless network, but disallow any Internet traffic until users authenticate with a web site that they are automatically redirected to, called a 'captive portal'. However, because the lower layers of the connection are not secured, such captive portals are often easily bypassed.

One method that attackers use to bypass captive portals is through traffic tunneling. Freely available programs are available that can encapsulate traffic in common protocols such as ICMP and DNS, which are often allowed through firewalls either directly or indirectly before authentication is complete.

Attackers can also spoof legitimate clients to bypass the captive portal. Captive portals that identify clients by MAC or IP address can be easily bypassed by changing the attacker's addresses to that of an already authenticated client.

Of course, the clients are at just as much risk regardless of the use of a captive portal. Compromised clients can also be used to bypass captive portals, and attacks can disclose registration details including credit card numbers for sites that require payment.

### **WPA-PSK**

Some hotspot operators, primarily businesses operating guest networks, use encrypted Wireless Protected Access (WPA) with a Pre-Shared Key (PSK) rather than using open wireless for guest use. As mentioned before, this is similar to needing a password just to get to a HTTPS web site. Operators must find a way to communicate this password to users before they connect, usually done through signage or giving it to employees to share. There are also a number of other issues with the use of pre-shared keys, including the need for users to configure their client with the proper settings, opening up the possibility of mistyped passwords and technical support calls.

But there are also fundamental problems with using pre-shared keys for guest networks. A shared secret password, one that you give out freely, results in shared security. Once an attacker has the shared secret key, they can capture the connection sequence of clients as they connect to get their session keys. An attacker doesn't even have to wait for new clients to connect, as connected clients can be forced to disconnect and reconnect using a deauthentication attack.

### **WPA-Enterprise**

The use of WPA Enterprise mode, while the most secure option available, has many pitfalls for hotspot operators. Primary among these is the need to communicate the credentials to the user before the user connects. This is occasionally done in companies that will give guests individualized credentials and instructions on how to connect to the network. However, guests are left to configure their systems, which can lead to incorrect configuration, permission issues, and mistyped passwords which result in helpdesk calls.

### **Open Secure Wireless Solution**

Another alternative is possible. EAP-TLS defines a method of creating a secure connection between the authenticator and client, and the secure transmission of keying material over this connection. As currently implemented, it is commonly used for client and server authentication in WPA-Enterprise networks.

The current standard for EAP-TLS is defined in RFC 5216. According to RFC5216:

The `certificate_request` message is included when the server desires the peer to authenticate itself via public key. While the EAP server SHOULD require peer authentication, this is not mandatory, since there are circumstances in which peer authentication will not be needed (e.g., emergency services, as described in [UNAUTH]), or where the peer will authenticate via some other means.

Put simply, EAP-TLS allows for a secure connection, without requiring for the client to authenticate. This is the same as how TLS is used for secure HTTPS web connections, where client authentication is optional and rarely required.

Just like TLS used for secure web servers, EAP-TLS provides for server cert validation. This allows clients to verify that they are connecting to the proper service, and not an attacker's system. If properly implemented, it would also not require any prior configuration and be easy for the end user, while still providing the security benefits.

### **Businesses**

For businesses, an Open Secure Wireless solution would enable operation of guest networks that are both convenient and secure. Because connections between clients and the network are secure, captive portals

can be used for client authentication without their current limitations. Guest network traffic would be encrypted, and server certificate validation would prevent evil twin wireless attacks.

### *Consumers*

For consumers, Open Secure Wireless would provide assurance that you are connecting to a trusted network, and that both the system and transmitted data is secure from eavesdroppers.

### *Wireless ISPs*

For Wireless ISPs and other hotspot operators, Open Secure Wireless would prove a method to offer secure payment and registration. It would prevent bypass of captive portal systems. Like HTTPS, once consumers have learned to look for secure networks, Wireless ISPs would benefit through enhanced customer satisfaction.

### *Current state*

Lab testing has been successful. By modifying the open source authenticator Hostapd, I have been able to connect to a wireless network using EAP-TLS without client authentication from both commercial and open source supplicants. When a connection is made to a modified EAP-TLS server that does not require client authentication, the server does not send a `certificate_request`, and moves directly to the SUCCESS on the EAP state machine.

### *Supplicant Support*

In testing, all wireless supplicants required a client certificate to be configured before the supplicant would connect to the network, even though the certificate is never used. EAP-TLS works with all tested supplicants as long as a candidate client certificate is available. It is apparent that the wireless supplicants were written to require client authentication, although this behavior is not required by RFC 5216. For the open source `wpa_supplicant`, the following code performs the client certificate check:

```
wpa_supplicant-0.6.9/src/eap_peer/eap_tls.c
if (config == NULL ||
    ((sm->init_phase2 ? config->private_key2 : config->private_key)
     == NULL &&
     (sm->init_phase2 ? config->engine2 : config->engine) == 0)) {
    wpa_printf(MSG_INFO, "EAP-TLS: Private key not configured");
    return NULL;
}
```

If the above code is removed, `wpa_supplicant` is able to connect successfully without any client certificate configured. Although source code for commercial supplicants are not available, from their behavior it is possible to surmise that they have a similar conditional check before attempting to connect to an EAP-TLS network.

### *Windows*

Microsoft Windows supplicants tested included Windows XP, Windows Vista, and Windows 7. All versions were able to connect to an Open Secure Wireless network when a candidate client certificate was configured. To support unauthenticated EAP-TLS as defined in RFC 5216, the check for client certificate should be moved to occur only when a `certificate_request` is received. In addition to moving the check, it would improve end user experience if the default behavior was changed to attempt to connect before prompting for authentication credentials, prompting only if credentials are necessary.

## Macintosh

Apple Macintosh 10.5 was tested, and the results were similar to Windows. As long as a candidate certificate was installed, a user can select and authenticate to an open secure wireless network. Similar to Windows, an authentication prompt is displayed before the connection is attempted.

## Linux

The open source wpa\_supplicant was tested on Ubuntu Linux. This supplicant is used on a number of popular Linux distributions. By default wpa\_supplicant checks for a candidate client certificate, but that can be changed in the source before compilation as mentioned previously.

## Server Support

### Hostapd

As mentioned previously, I modified the open source Hostapd authenticator to not require client authentication for EAP-TLS. This simple value change caused EAP-TLS to behave similar to EAP-PEAPv0.

```
src/eap_server/eap_tls.c
68c68
< if (eap_server_tls_ssl_init(sm, &data->ssl, 0)) {
---
> if (eap_server_tls_ssl_init(sm, &data->ssl, 1)) {
```

### Microsoft Network Policy Server

The Network Policy Server component of Windows 2008 sends a certificate\_request and performs client certificate validation. Providing a configuration option for Open Secure Wireless would enable it to perform this function.

## Access Point (Authenticator) Support

No changes to access points would be required to support Open Secure Wireless, as EAPOL is designed to make EAP types transparent to the authenticator. This is an important point, as access points are widely deployed and may be difficult to update.

## Future Work

### Certificate Validation

Current wireless protocols EAP-TLS and EAP-PEAP do not perform path validation. For a wireless configuration, either the specific server name must be specified, or any certificate signed by one or more valid Certificate Authorities can be used. For HTTPS applications, path validation validates that the CN or SubAltName on the presented server certificate matches the DNS name of the site that the user is connecting to.

One possible solution may be to change the default behavior of EAP-TLS clients to match the SSID to the CN or SubAltName on the certificate. This would allow current Certificate Authority DNS-based validation to be used. One limitation to this approach is the 32 character limit for SSIDs, but this appears sufficient for most cases.

### Wireless Network Intention

There is currently no way in wireless standards for a wireless provider to communicate the intent of a wireless network. Without such a method, physical signage or other methods are needed for a guest to know whether a network was intentionally or accidentally left open.

It might be possible to communicate the intent using an Information Element (IE). Although this is not a specific concern with using EAP-TLS without client certificates, it would be useful if wireless supplicants indicated the intention of a network to help guide the user.

## **Conclusion**

Open Secure Wireless networks based on current EAP-TLS standards are possible today using modified authentication servers and unmodified clients with candidate client certificates that will not be used or validated. Modifications to wireless supplications would remove the requirement that a client certificate exist, and provide for a better user experience. The use of TLS in HTTPS web sites has been the foundation for Internet commerce, and it is possible that Open Secure Wireless networks would benefit businesses, consumers, and wireless internet providers by making open wireless trustworthy.