

WLAN Security

4. Standard Authentication

(C) Herbert Haas 2006/4/1

Content

In this chapter a detailed overview about today's WLAN security problems and solutions are presented.

This subchapter provides an introduction into the 802.11 standard authentication methods.

Objective

After completing this chapter the following tasks could be solved:

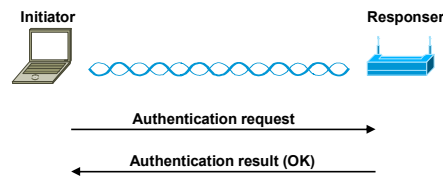
- Emphasize the basic vulnerabilities of WLAN
- Explain why WEP is insecure and give a mathematical example
- Compare WEP and TKIP/MIC
- Highlight the design flaws of the WLAN standard authentication
- Explain the design idea of 802.1x
- Compare EAP-TLS, LEAP, PEAP, EAP-TTLS, EAP-FAST with each other and emphasize important security features
- Explain the design concept of WPA and WPA2
- Implement a reliable 802.1x infrastructure over a WAN connection
- List important issues to be considered when choosing a VPN design
- Explain PSPF

802.11 Standard Authentication Methods



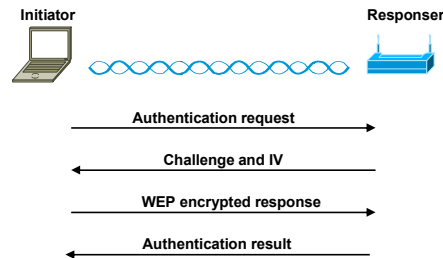
Open System Authentication

- Anyone is granted access
- Ideal for transient users
- Default method
- All frames sent in clear, even when WEP is enabled



Shared Key Authentication

- Relies on WEP algorithm
- Every user has same shared key—and same as AP
- Only client device authentication
- User is not authenticated (device theft critical)
- AP is not authenticated (!)
- Vulnerable...



Open System Authentication allows anyone to gain access to the WLAN. It is generally applicable where public access should be provided, for example in universities, airports, or hotels. The authentication process is realized using "management" frames with "authentication" as subtype. Specifically, the open system method is indicated using an algorithm identification field.

Shared Key Authentication uses the WEP algorithm to implement a four-step handshake procedure, provided that each user has the same shared key. Shared Key Authentication only enables client authentication but the client can never be sure whether the AP is a "rogue" AP. Furthermore, WEP is vulnerable, and hence this authentication process can be attacked.

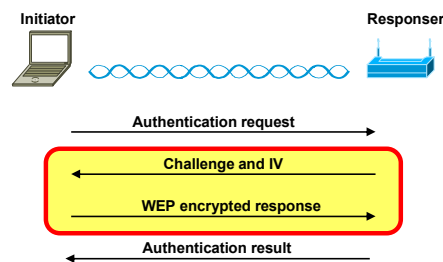
This four-step procedure requires WEP support from both sides. It is assumed that both sides possess the same shared key. The initiator sends an authentication request management frame indicating that it wish to use "shared key" authentication. The responder replies by sending an authentication management frame containing an 128 octets challenge text. This challenge text is generated by using the WEP pseudo-random number generator (PRNG) with the "shared secret" and a random IV. The initiator receives the challenge and the IV and sends a WEP-encrypted version of the challenge back to the responder, hereby using the shared secret and the IV. The responder decrypts the received frame and verifies the 32-bit CRC integrity check and that the challenge text matches that sent in the first message. In this case the authentication is successful and the responder completes the process by sending the authentication result. Optionally, the initiator and the responder switch roles and repeat the process to ensure mutual authentication. However, mutual authentication is seldom implemented. The value of the status code field is set to zero when successful, and to an error value if unsuccessful. The element identifier identifies that the challenge text is included. The length field identifies the length of the challenge text and is fixed at 128. The challenge text includes the random challenge string.

Besides WEP design flaws, the whole authentication is tied to the device identity, not the user's identity. That is, a stolen device can be abused to gain access to the WLAN.

Shared Key Authentication



- Attacker captures 2nd and 3rd authentication message and has
 - ♦ Plaintext P (the challenge)
 - ♦ Ciphertext C = RC4^K (P)
- The keystream is simply
 $S = C \oplus P$
- Other fields than the challenge are known a priori
 - ♦ Have always the same value in each authentication process
- Possessing S, an attacker can correctly respond to each challenge
- **Never use Shared Key Authentication !!!**



Never use Shared Key Authentication

An attacker could easily capture the 2nd and 3rd authentication messages and possesses a plaintext (the challenge) and the corresponding ciphertext. Remember that the keystream S can be easily calculated by XORing both messages.

Other fields (besides the challenge) are rather static and can be guessed—they have always the same values in each authentication process.

Having S, an attacker can easily authenticate to the network as he is able to correctly respond to each challenge sent by a responder.