802.11, 802.1x, and Wireless Security

Wireless local area networks are increasingly deployed by businesses, government, and SOHO users because of the freedom wireless communications afford and the decreasing costs of the underlying technology. Current security mechanisms for maintaining the confidentiality, integrity, and availability of wireless communications are problematic, however. For example, although the 1997 IEEE 802.11 wireless standard specifies both an authentication service and encryption protocol, sources have demonstrated these to be severel...

GIAC Security Essentials Certification (GSEC)
Practical Assignment
Version 1.4

802.11, 802.1x, and Wireless Security

J. Philip Craiger

June 23, 2002

Abstract

Wireless local area networks are increasingly deployed by businesses, government, and SOHO users because of the freedom wireless communications afford and the decreasing costs of the underlying technology. Current security mechanisms for maintaining the confidentiality, integrity, and availability of wireless communications are problematic, however. For example, although the 1997 IEEE 802.11 wireless standard specifies both an authentication service and encryption protocol, sources have demonstrated these to be severely flawed, leaving wireless communications open to several types of attacks. Recent security standards, such as the IEEE 802.1x, intend to provide solutions to these security defects. However, sources have shown that even the new standards are flawed, allowing attackers to perpetrate both active as well as passive attacks.

This paper focuses on a description and analysis of the security standards described in the IEEE 802.11 and 802.1x standards, as well as some of the inherent problems with the security mechanisms defined in the standards. Recommendations for securing wireless networks are provided.

Introduction to IEEE 802.11 Standard

In 1997 the Institute of Electrical and Electronics Engineers (IEEE) Working Group for Wireless Standards passed the first standards for wireless communications in the United States. The standard, IEEE 802.11 (IEEE, 1997), provides a common standard that allows vendors to create wireless technologies that are interoperable.

WLANs are similar to wired LANs only communications among elements on the network is accomplished through wireless transmissions, typically radio waves, as opposed to the more common wired, physical connections. 802.11-based WLANs may run in one of two modes. A WLAN running in **infrastructure mode** (or Basic Service Set; BSS) is comprised of clients or 'stations,' i.e., computers with wireless network interface cards (NICs), and access points (APs). APs act as bridges between the wired and wireless networks. The second mode is the **ad-hoc mode** (or Independent Basic Service Set, IBSS) where clients communicate directly with other clients without an intervening AP (Nicholls & Lekkas, 2002).

A WLAN uses radio waves operating within the 2.4 GHz Industry, Scientific, and Medical (ISM) band. The three physical layers defined in the 802.11 standard include two spread-spectrum radio techniques and a diffuse infrared specification. 802.11 defines data rates of 1 Mbps and 2 Mbps via radio waves using **frequency hopping spread spectrum** (FHSS) or **direct sequence spread spectrum** (DSSS). 802.11b is an enhancement of 802.11 employing DSSS to achieve a maximum throughput of 11 Mbps (Nicholls & Lekkas, 2002).

802.11 Security Mechanisms

Security in IEEE 802.11 is provided by an authentication service composed of two types of authentication, and an encryption protocol. A number of sources have demonstrated that both the authentication service and encryption mechanism are inherently, and perhaps irreparably, flawed. Each security mechanisms, along with ancillary mechanisms often used to provide security, are described and analyzed below.

Wired Equivalent Privacy (WEP)

WEP is a link-layer security protocol that is specified, but not required, by the 802.11 standard. WEP is based on the RC4 stream cipher, a symmetric cipher where the same key is used for both encryption and decryption. RC4 is the most widely used stream cipher in software applications (Fluher, Mantin & Shamir, 2001). The term 'wired equivalent' denotes that the security provided by WEP is intended to be roughly equivalent to what one would expect in a wired LAN.

Wired LANs, of course, can be protected by numerous physical mechanisms, unlike wireless transmissions.

WEP was intended to enforce three security goals (Borisov, Goldberg, & Wagner, 2001):

1. Confidentiality, i.e., to prevent eavesdropping, through the use of encryption;
2. Access control, through the option to discard improperly encrypted packets and through authentication mechanisms; and
3. Data integrity, i.e., preventing tampering with transmissions through the use of a data checksum.

WEP Mechanics

The original 802.11 standard stipulates a 40-bit WEP key. Cryptographically stronger 104-bit keys implementations are provided by a number of WLAN vendors.

Figure 1 graphically illustrates WEP.

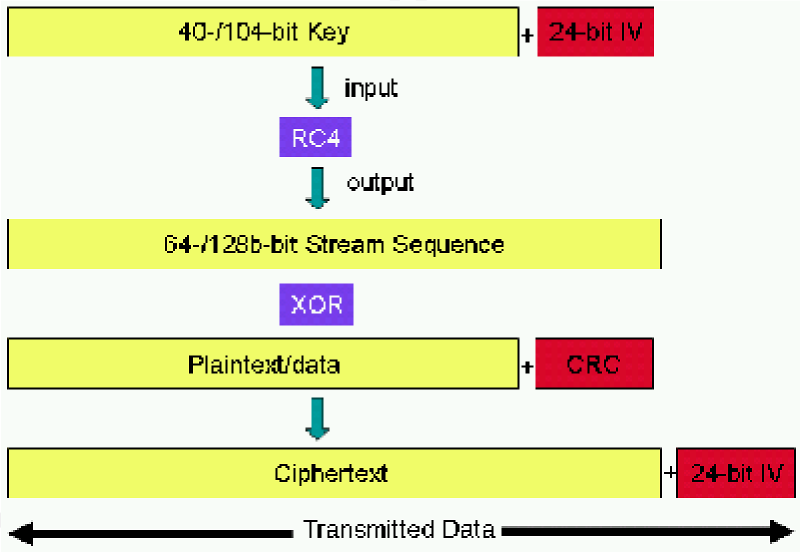


Figure 1. WEP Illustrated [adapted from Loeb, 2001].

WEP functions as follows:

1. A secret key (either 40- or 104-bits) is concatenated with a 24-bit initialization vector (IV) resulting in a 64- or 128-bit key. An IV is added to the secret key in each packet to ensure that each packet has a different RC4 key (given that the secret key doesn't change frequently)

2. The key from (1) is input into the RC4 PRNG (pseudorandom number generator), resulting in pseudorandom keystream of the same length as the initial key (i.e., either 64 or 128 bits).
3. The plaintext (data) is run through an integrity checking algorithm resulting in a checksum. This checksum (the CRC in Figure 1) is concatenated onto the plaintext so that the integrity of the information may be checked by the decrypting party.
4. The data vector, i.e., data + checksum vector from step (3), is encrypted by doing a bitwise XOR with the keystream from step (2) above, which results in the ciphertext.
5. The IV is appended to the ciphertext and the result is transmitted via wireless.

Note that the 802.11 standard does not specify any type of key management, meaning that vendors are free to implement key management as they like. In practice, key management is handled manually by systems administrator and/or users.

Problems with WEP

Several WEP flaws have been widely documented and disseminated. Each of these flaws allows passive or active attacks on wireless transmissions, either allowing attackers to decrypt information or inject forged information into the transmissions. Each attack depends upon the ability of the attacker to monitor 2.6 GHz radio frequencies and translate the 802.11 physical layers into human readable form. This is fairly easy to accomplish, requiring a laptop or handheld (e.g., Compaq IPAQ), a wireless NIC capable of running in promiscuous mode, and freely available sniffer software capable of translating the packets into human readable form (e.g., ethereal: www.ethereal.org). Several WEP flaws are described below.

IV Collisions

One WEP flaw is based upon what is called an 'IV collision.' An IV collision simply means that an IV is reused at some point in a wireless transmission. Recall that an IV is added to the secret key in each packet to ensure that each packet has a different RC4 key, given that the secret key doesn't change frequently. A well-known problem with stream ciphers is that two packets encrypted with the **same IV** can be easily decrypted (Borisov, Goldberg, & Wagner, 2001). The equations below demonstrate algebraically how the attack works.

$$C_1 = P_1 \otimes RC4(v,k)$$
$$C_2 = P_2 \otimes RC4(v,k)$$
$$C_1 \otimes C_2 = (P_1 \otimes RC4(v,k)) \otimes (P_2 \otimes RC4(v,k))$$
$$C_1 \otimes C_2 = P_1 \otimes P_2$$

The first and second equations show that two ciphertexts (C_1 and C_2) are calculated by XORing the plaintext (P_1 and P_2) and the same keystream *RC4*(*v*, *k*), where *v* is the IV and *k* is the secret key. A little algebraic manipulation demonstrates that two ciphertexts that use the same keystream (i.e., *RC4*(*v,k*)) cancel the keystream, resulting in the XOR of the plaintexts: $P_1 \otimes P_2$ (Borisov, Goldberg, & Wagner, 2001).

Attackers have several avenues for partitioning the two XORed plaintexts. One way is through the use of a **known plaintext attack**. If an attacker can get a victim to send known plaintext, such as spam or through an email, then it is fairly trivial to recover the unknown part of the XORed plaintext message (Stark, 2001). Also, the probability that an attacker is able to infer plaintext in a message is fairly good given that IP traffic is structured in a well-known manner, e.g., there is consistent information in the TCP and UDP headers across packets. Failing this, some understanding of the statistical nature of repetition of letters in an alphabet and some common sense may led to an attacker's partitioning of the two plaintext messages.

IV collisions are all but ensured by several factors. First, the 24-bit IV keyspace is not large enough to ensure against collisions for any reasonable length of time. An AP that sends 1500 byte packets at 11Mbps will exhaust the keyspace of IVs in as little as five hours (Borisov, Goldberg, & Wagner, 2001).

Second, some wireless NICs reinitialize IVs to 0 each time a card is initialized and increments by 1 for each packet (Stubblefield, Ioannidis, & Rubin, 2001). This means that transmission begins with a known and repeating IV, resulting in the opportunity for more IV collisions or allowing attackers to guess the IV.

Third, WEP security is based on the assumption that secret keys are changed on a frequent basis. In reality, secret keys are **not** changed on a frequent basis, largely because it's a manual process and time consuming: the key must be distributed and inputted into each user's software, as well as the AP. And it is unlikely that anyone would change keys every five hours (i.e., before a collision could occur). Based on these factors it is almost a certainty that collisions occur frequently.

Other Attacks

The IV collision problem was described in detail because it is one of the best documented attacks and a fairly understandable problem. There are several other well documented attacks, including:

- RC4s weak key scheduling, allowing keys to be guessed based on the first few packets transmitted (Fluher, Mantin, & Shamir, 2001); and,
- Linearity of the integrity check value algorithm that produces the CRC data fingerprint allows an attacker to flip the bits on encrypted data to determine how the CRC value changes, which may provide clues as to the underlying plaintext. (Arbaugh, 2001)

Perhaps the easiest attack on a wireless network may be facilitated through **social engineering**. Social engineering is when an attacker attempts to spoof his/her identity, e.g., by pretending to be someone who is authorized to have access to network information, and tricks a user out of that information, such as the user's username and password (Cole, 2002). Kevin Mitnick, perhaps the world's most notorious 'hacker,' claimed that over 90% of his break-ins were accomplished through social engineering (Granger, 2001). Because key management is manual, and users have access to this information, breaking into a wireless network would be easier accomplished through social engineering than through the means cited above.

Authentication and Association

A client must authenticate and establish an association with an AP prior to transmitting data. An association is simply a binding between the client and an AP. The 802.11 standard provides for two types of authentication. **Open systems authentication** is a requirement of the standard and is the default for most APs (i.e., the default out of the box). Open authentication allows any clients to associate with an AP as long as the SSIDs match. **Shared-key authentication** controls access to the WLAN through a shared-key, and is used to provide more stringent access to network resources. WEP must be enabled for shared-key authentication, because the key used for WEP is the same as that used for authentication.

Figure 2 graphically illustrates the shared-key authentication process.

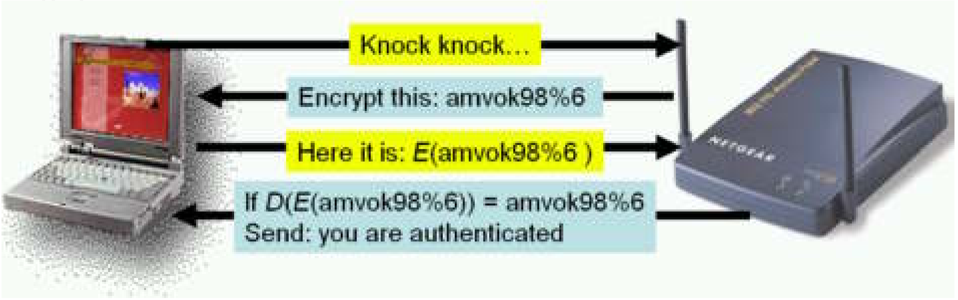

Knock knock…
Encrypt this: amvok98%6
Here it is: E(amvok98%6)
If D(E(amvok98%6)) = amvok98%6
Send: you are authenticated
NETGEAR

Figure 2. Shared-Key Authentication

The process of shared-key authentication is as follows:

- A client attempts to associate with an AP [“Knock knock…” →].
- The AP replies with a string of random text as a challenge [← “amvok98%6”].
- The client uses its shared key to encrypt the text, then sends the encrypted text back to the AP [*E*(amvok98%6) →].
- AP decrypts the encrypted text using its shared-key [*D*(E(amvok98%6))]. If the decrypted text is the same as the original text then the client is authenticated. This only occurs, of course, when the shared-keys are the same.

Problems with Authentication

Clearly there are serious problems with open authentication. Anyone who knows the SSID can gain network access. In fact, it is simple to discover an APs SSID because each AP transmits beacon frames that contain its SSID in the clear (i.e., they are human readable; Arbaugh, Shankar, & Wan, 2001). Unless you enjoy sharing network resources with anyone and everyone, open authentication should not be used. Shared-key authentication is more secure of course, but still problematic. Because all clients use the same key it is not possible to authenticate (and track) an individual user. Also, the key used for WEP is the same used as the shared-key for authentication. Therefore, by stealing one key an attacker kills the proverbial two-birds, and now has access to authentication methods as well as encryption and decryption. Again, a little social engineering would likely reveal an APs SSID.

Also note that an AP authenticates a user, but a user does not and cannot authenticate an AP. That is, there is no two-way authentication. If a rogue AP is placed on a WLAN, it can be a launch pad for denial-of-service attacks through the ‘hijacking’ of the clients of legitimate users.

Service Set Identification (SSID)

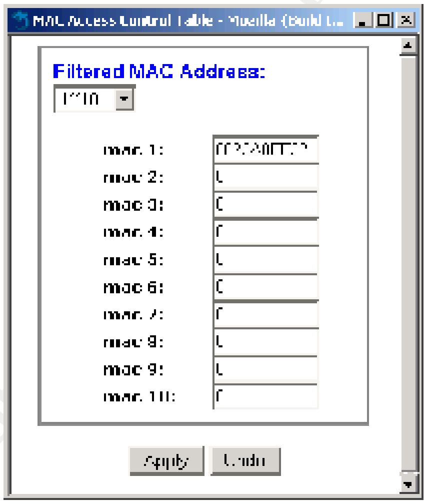
A mechanism that some have used as a security mechanism, but which was not intended to be as such, is the **service set identification** (SSID). SSIDs were originally intended to logically segment a network into subsystems. The SSID is simply a network name that must be specified and matched by both AP and clients in order for the client to associate with the AP. Some clients use the SSID in the same capacity as a shared secret key, and as a rudimentary form of access control. (Note: The author was guilty of this before becoming more security conscious.)

The 'secrecy' of the SSID is a myth because, as described above, APs transmit beacon frames that contain its SSID in the clear (Arbaugh, Shankar, & Wan, 2001). One needs only a computer with a wireless NIC running in promiscuous mode, and 'sniffer' software (e.g., ethereal) to capture this information. If the AP is running in open authentication mode then an intruder may access the wireless network by simply changing her SSID to that which she just discovered. Also, given that all (authenticated) clients know or have access to the SSID, then a bit of social engineering is all that is needed to acquire the SSID from a client.

MAC Address Filtering

Several vendors added access control lists (ACL), implemented through MAC address filtering, to increase security. MAC address filtering amounts to allowing predetermined clients with specific MAC addresses to authenticate and associate.

Figure 3 is a screen capture of the web management interface for MAC address filtering for a Linksys router/AP on the authors WLAN.



MAC Access Control Table - Mozilla {Build I...

Filtered MAC Address: 1~10

mac 1: [illegible]
mac 2: 0
mac 3: 0
mac 4: 0
mac 5: 0
mac 6: 0
mac 7: 0
mac 8: 0
mac 9: 0
mac 10: 0

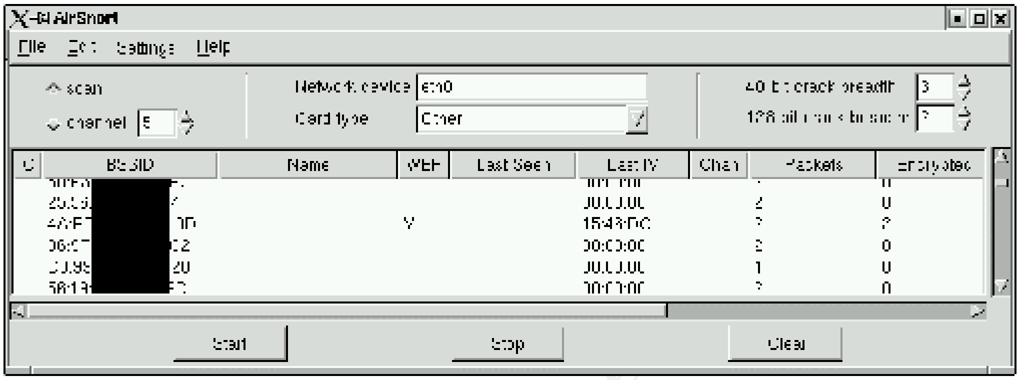
Figure 3. MAC Address Filtering [Note: MAC address shown is a fictitious address created by the author].

The addition of MAC address filtering increases security, however it is not a perfect solution given that MAC addresses can be spoofed (i.e., forged; Cole, 2002). Also, the process of manually maintaining a list of all MAC addresses can be time consuming and error prone. Therefore MAC address filtering is probably best left for only small and fairly static networks.

"Sniffing" Tools

There are several software tools, widely and freely available over the Internet, that allow attackers to sniff (i.e., listen in on) wireless transmissions. Some of these have the ability to break WEP if provided a sufficient number of encrypted packets. Airsnort (airsnort.shmoo.com), WEPcrack (wepcrack.sourceforge.net), and ethereal (www.ethereal.com) are well known sniffing tools, with the former two providing WEP decrypting capabilities.

Figure 4 is a screen capture of AirSnort running under Linux on the author's laptop computer.



○ scan
○ channel 5

Network device eth0
Card type Other

40 bit crack breadth 3
128 bit crack breadth 2

C	BSSID	Name	WEP	Last Seen	Last IV	Chan	Packets	Encrypted
	00:E0[redacted]F:				00:00:00		1	0
	20:30[redacted]7				00:00:00		2	0
	4A:F7[redacted]0D		Y		15:43:DC		?	?
	06:C7[redacted]C2				00:00:00		2	0
	00:99[redacted]20				00:00:00		1	0
	58:13[redacted]FC				00:00:00		?	0

Figure 4 AirSnort Running on a Laptop

AirSnort works by setting the wireless NIC into capture (promiscuous) mode. Note it has the ability to capture SSIDs (sanitized for privacy), whether WEP is enabled, the last IV transmitted, the number of packets sent, encrypted packets, and so on.

Stubblefield, Ioannidis, & Rubin (August, 2001) described an experiment conducted using simple hardware and the aforementioned software (ethereal) to recover the 128 bit secret key with a passive attack, demonstrating that the WEP implements RC4 IVs improperly.

Summary of 802.11 Security Mechanisms

Given the tremendous growth in WLAN usage, and the weakness of current security protocols, new and better security mechanisms are required to protect wireless transmissions. One of these is the relatively new IEEE 802.1x standard.

802.1x: Port-Based Network Access Control

The IEEE 802.11 Working Group passed the 802.1x standard in 2001 to improve upon the security specified in the original 802.11 standard (IEEE, 2001). 802.1x was intended to provide strong authentication, access control, and key management (Mishra & Arbaugh, 2001), and allow WLANs to scale by allowing centralized authentication of wireless users or stations (Geier, 2002; Roshan, 2001).

802.1x is based upon an existing authentication protocol known as the **Extensible Authentication Protocol** (EAP) which in itself is an extension of PPP (point-to-point protocol). 802.1x isn't tied to any particular networking scheme, but serves as the basis for defining a means for authenticating users to the physical network, regardless of the underlying network protocols. Thus, 802.1x maps EAP to the physical medium, regardless of whether it is Ethernet, Token Ring or wireless LAN. It also supports multiple authentication methods, including token cards, Kerberos, one-time passwords, certificates, and public key authentication (Geier, 2002; Roshan, 2001).

802.1x Mechanics

802.1x authentication has three main components: A client; an authenticator (here an AP); and an authentication server. The authentication server is usually a Remote Authentication Dial-In User Service (RADIUS) server, although RADIUS is not specifically required by the standard.

Figure 5 graphically depicts the authentication process.

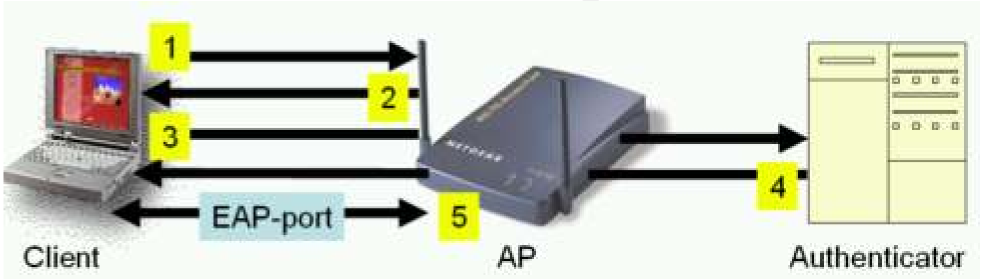


Figure 5. 802.1x Authentication Process

802.1x authentication occurs as follows:

1. The client sends a request for authentication to the AP.
2. The AP replies with a request that the client provide identification, and blocks all other traffic, such as HTTP, DHCP, and POP3 packets, until the AP can verify the client's identity using the authentication server.
3. The client sends a response containing the identity to the authentication server. (Recall that the type of identification is not specified in the protocol, but rather left up to the vendors, so authentication could be of any form).
4. The authentication server receives the request and uses an appropriate authentication algorithm to verify the client's identity. If the user can be identified, an accept message is sent to the AP, otherwise a reject message is sent.
5. If the authentication server accepts the client, then the AP will transition the client's port to an authorized state and forward additional traffic.

802.1x and Dynamic Key Management

Note that the 802.1x standard provides for authentication only. The standard does not specify the specific types of authentication or any type of encryption. Nevertheless, as of June 2002 several vendors offer proprietary versions of dynamic key management using 802.1x as a delivery mechanism. Through dynamic key exchange the authentication server can return session keys to the AP along with the accept message (Geier, 2002).

Figure 6 graphically illustrates the use of session keys in the interaction.

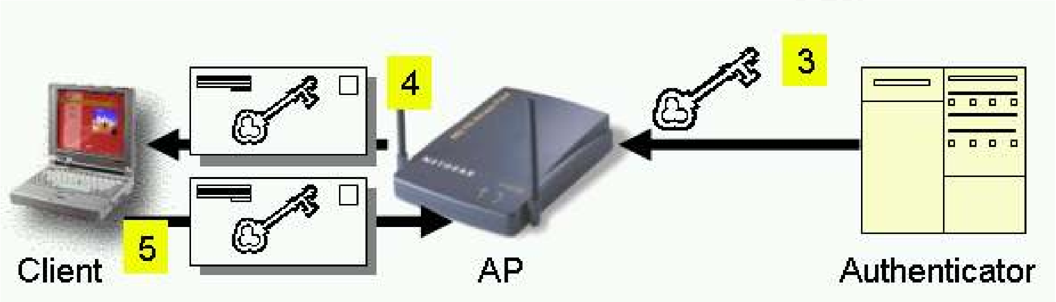


Figure 6. 802.1x using Dynamic Session Keys

In step 3, rather than returning a simple accept or reject message, the authenticator returns both the results of authentication plus a session key. The AP uses the session keys from the authentication server to sign and encrypt a message that is forwarded to the client after sending the success message (step 4). The client can then use contents of the key message to define appropriate encryption keys (step 5 and thereafter).

The mechanism for dynamic key management provides a more secure mechanism than the manual maintenance of keys. The 802.1x mechanism allows clients - through the use of dynamic key management -- to automatically change encryption keys as often as necessary to minimize the possibility of a passive attack.

Problems with 802.1x

Alas, the 802.1x protocol is not foolproof. Cisco published “Cisco Security Advisory: Catalyst 5000 Series 802.1x Vulnerability” in April of 2001 regarding a problem with its own 802.1x implementation (Cisco, 2001). Researchers at the University of Maryland found that 802.1x is susceptible to session hijacking as well as man-in-the-middle attacks (Mishra & Arbaugh, 2001; Connolly, 2002; Schwartz, 2002).

Session hijacking is when an attacker takes over an existing session, meaning the attacker is relying on an existing authenticated connection to acquire access to network resources (Cole, 2002).

Figure 7 graphically illustrates session hijacking.

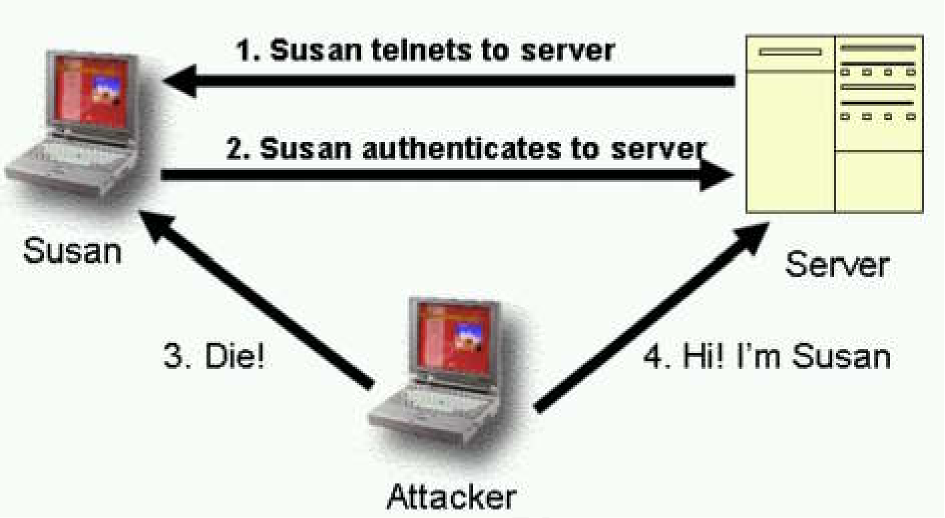


Figure 7. Session Hijacking [adapted from Cole, 2002]

Figure 7 shows that the attacker waits until Susan (a valid user) authenticates, then kills or blocks Susan's connection (e.g., through various forms of denial of service attacks, see Cole, 2002), and subsequently pretends to be Susan. This requires that the attacker spoof the authenticated user's IP address in order to maintain the connection.

Similarly, Mishra & Arbaugh (2001) explained session hijacking in the context of an authenticated 802.1x connection as follows:

1. The hacker waits for someone to authenticate successfully.
2. The attacker sends a "disassociate" (quit) message, spoofing it to make it look like it came from the AP.
3. The client thinks they have been kicked off, but the AP thinks the client is still out there.
4. As long as transmissions are not encrypted the attacker can start using that connection up until the next time out (~1 hour).

WEP 2

This discussion would not be complete without a discussion of WEP2. Although not passed formally (yet) as a standard, the fact that WEP2 was conceived acknowledges problems with the initial encryption protocol, WEP.

WEP2 was created to be backwards compatible with WEP. Two major additions in WEP2 are the enforcement of 128-bit keys, and mandatory support for KerberosV (Adoba, 2001). Unfortunately, these changes do not eliminate the flaws that exist in WEP (Adoba, 2001; Stark, 2001):

1. Although the mandatory key length has been increased to 128 bits, it fails to prevent IV replay exploits and still permits IV key reuse.
2. Known plaintext attacks are still possible.
3. Mandatory KerberosV support opens WEP2 to dictionary-based attacks.
4. WEP2 is vulnerable to denial of service attacks because reassociate and disassociate messages are not secure.
5. Clients can roam to a rogue AP, which could completely compromise them because beacon messages are not authenticated (that is, two-way authentication still doesn't exist).

As it stands currently, it appears that WEP2 is as broken as the original version.

Wireless Security Recommendations

Strong wireless security mechanisms are lacking as of mid-2002. It is not unreasonable to argue that a cause is that vendor's zealous attempt to meet demand for wireless products as quickly as possible, combined with the fact that creating good security is **difficult**: it takes time to develop, test, and implement good solutions.

The biggest threat to the security of WLANs is the failure to use **any** form of security. Most APs ship with a default SSID that is easily guessed or listed on the Internet, and with WEP disabled. This makes it easier for the naïve user to set up a WLAN, but at the cost of security.

Below is a compilation of recommendations to increase the security of WLANs. It is best to remember that **security is best implemented in layers**. An attacker may get through the first few layers, but most attackers, except the most diligent, will stop before they compromise your system.

1. Enable WEP, however imperfect. Sources indicate that on average only 30% of APs use WEP (Phifer, 2002).
2. Change WEP keys frequently.
3. Change the default SSID to something difficult to guess (e.g., a long and random sequence of characters).
4. Use dynamic session keys if implemented in your product.
5. Use shared-key authentication in preference to open authentication.
6. Consider using an 802.1x implementation (e.g., Windows XP)
7. Use MAC address filtering if available.
8. Use a virtual private network (VPN) if available.

9. Track computer inventory to ensure wireless NICs remain in employee hands (Phifer, 2002).
10. Block the MACs of wireless NICs of lost or stolen wireless NICs (Phifer, 2002).
11. Password protect AP management interfaces, just as you would on any perimeter router or firewall. (Phifer, 2002)
12. Use anti-virus and personal firewall software to keep the wireless client clean, preventing back-channels (Phifer, 2002).
13. By combining firewall defense with IPsec, SSH, or SSL, you can better prevent wireless eavesdropping and block access by unauthenticated clients (Phifer, 2002).
14. Treat all systems that are connected via 802.11 as external (Stubblefield, Ioannidis, & Rubin, 2001).
15. Place APs outside of firewalls (Stubblefield, Ioannidis, & Rubin, 2001).
16. Assume that anyone within physical range [~300 yards] can communicate on the network as a valid user. (Stubblefield, Ioannidis, & Rubin, 2001).

Until improved security mechanisms are in place, users of WLANs who are concerned with security have two options:

1. make the best of what is available, or
2. quit using wireless altogether.

References

Aboba, B. "WEP2 Security Analysis: IEEE 802.11-00/253" URL. http://www.drizzle.com/~aboba/IEEE/11-01-253r0-I-WEP2SecurityAnalysis.ppt, 2001.

Arbaugh., W. An inductive chosen plaintext attack against WEP/WEP2. IEEE Document 802.11-01/230, May, 2001.

Arbaugh, W., Shankar, N., & Wan, J. Your 802.11 Wireless Network has No Clothes. Department of Computer Science University of Maryland College Park, Maryland, March, 2001.

Borisov, N. Goldberg, I. & Wagner, D. Intercepting Mobile Communications: The Insecurity of 802.11. http://www.isaac.cs.berkeley.edu/isaac/wep-draft.pdf, August, 2001.

Cisco Systems. "Cisco Security Advisory: Catalyst 5000 Series 802.1x Vulnerability." April 2001. URL: http://www.cisco.com/warp/public/707/cat5k-8021x-vuln-pub.shtml. (1 May 2002)

Cole, E. (2002). Hackers Beware. Boston, MA: New Riders

Connolly, P.J. “The trouble with 802.1x.” InfoWorld. 8 March 2002. URL: http://www.infoworld.com/articles/fe/xml/02/03/11/020311fe8021x.xml. (20 May 2002).

Conover, J. “Anatomy of IEEE 802.11b Wireless” 7 August 2001. URL: http://www.networkcomputing.com/1115/1115ws2.html. (7 June 2002).

Fluher, S., Mantin, I., & Shamir, A. (2001). Weaknesses in the Key Scheduling Algorithm of RC4. http://downloads.securityfocus.com/library/rc4_ksaproc.pdf

Geier, J. “802.1X Offers Authentication and Key Management.” 802.11 Planet. 7 May, 2002. URL: http://www.80211-planet.com/tutorials/print/0,,10724_1041171,00.html. (4 June, 2002)

Geier, J. “802.11 WEP: Concepts and Vulnerability.” 802.11 Planet. 20 June 2002.
URL: http://www.80211-planet.com/tutorials/article/0,,10724_1368661,00.html. 22 June 2002.

Granger, S. “Social Engineering Fundamentals, Part I: Hacker Tactics.” December 2001. URL: http://online.securityfocus.com/infocus/1527. (17 June 2002).

IEEE. “IEEE 802.11-1997 (ISO/IEC 802-11: 1997). “LAN MAN Standards of the IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer(PHY) specification. IEEE Standard 802.11, 1997 Edition.

IEEE. “IEEE 802.11-1999 (ISO/IEC 8802-11: 1999). “Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.” URL: http://standards.ieee.org/reading/ieee/std/lanman/802.11-1999.pdf

IEEE. “IEEE 802.1x-2001 (ISO/IEC 802-1x: 2001), Part 11: Wireless LAN Medium Access Local and metropolitan area networks: Port-Based Network Access Control.” URL: http://standards.ieee.org/getieee802/download/802.1X-2001.pdf

Loeb, L. “What's up with WEP?” April 2001. http://www-106.ibm.com/developerworks/library/s-wep/. (4 June 2002)

Mishra, A. , & Arbaugh, W. (2002). An initial security analysis of the 802.1x standard. http://www.cs.umd.edu/~waa/1x.pdf

Nicholls, R., & Lekkas, P. (2002). Wireless Security: Models, Threats, and Solutions.

Petertz, M. "A Very Funky 802.1x Security Solution. 802.11 Planet. 31 January 2002. URL: http://www.80211-planet.com/news/print/0,,1481_965961,00.html. (13 May 2002).

Phifer, L. "Improving WLAN security" 802.11 Planet. 11 January 2002. URL: http://www.80211-planet.com/tutorials/print/0,,10724_953651,00.html (7 May 2002).

Roshan, P. "802.1X authenticates 802.11 wireless." NetworkWorldFusion. 24 November 2001. URL: http://www.nwfusion.com/news/tech/2001/0924tech.html (14 May 2002).

Schwartz, E. "Researchers crack new wireless security spec." InfoWorld. 14 February 2002. URL: http://www.infoworld.com/articles/hn/xml/02/02/14/020214hnwifispec.xml. (12 June 2002).

Stark, T. "WEP2, Credibility Zero." 2001. URL: http://www.dnai.com/~thomst/wireless003.html. (29 April 2002).

Stubblefield, A., Ioannidis, J., & Rubin,A.D., (August, 2001). Using the Fluher, Mantin and Shamir Attack to Break WEP. AT&T Labs Technical Report TD-4ZCPZZ.

Last Updated: September 20th, 2010

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SOS: SANS October Singapore 2010	Singapore, Singapore	Oct 04, 2010 - Oct 11, 2010	Live Event
EU Process Control and SCADA Security Summit 2010	London, United Kingdom	Oct 07, 2010 - Oct 14, 2010	Live Event
SANS Gulf Region 2010	Dubai, United Arab Emirates	Oct 09, 2010 - Oct 21, 2010	Live Event
SANS Geneva Security Essentials at HEG Fall 2010	Geneva, Switzerland	Oct 11, 2010 - Oct 16, 2010	Live Event
SANS App Sec India 2010	Bangalore, India	Oct 18, 2010 - Oct 22, 2010	Live Event
CyberSecurity Malaysia SEC 401 Onsite	Kuala Lumpur, Malaysia	Oct 18, 2010 - Oct 23, 2010	Live Event
SANS Chicago 2010	Skokie, IL	Oct 25, 2010 - Oct 30, 2010	Live Event
SANS San Francisco 2010	San Francisco, CA	Nov 05, 2010 - Nov 12, 2010	Live Event
SANS San Antonio 2010	San Antonio, TX	Nov 13, 2010 - Nov 20, 2010	Live Event
SANS Tokyo 2010 Autumn	Tokyo, Japan	Nov 15, 2010 - Nov 20, 2010	Live Event
SANS Geneva CISSP at HEG Fall 2010	Geneva, Switzerland	Nov 15, 2010 - Nov 20, 2010	Live Event
SANS Sydney 2010	Sydney, Australia	Nov 15, 2010 - Nov 20, 2010	Live Event
SANS London 2010	London, United Kingdom	Nov 27, 2010 - Dec 06, 2010	Live Event
WhatWorks in Incident Detection and Log Management Summit 2010	Washington, DC	Dec 08, 2010 - Dec 09, 2010	Live Event
SANS Cyber Defense Initiative East 2010	Washington, DC	Dec 10, 2010 - Dec 17, 2010	Live Event
SANS WhatWorks: PCI Compliance in Information Security Summit 2010	OnlineNV	Sep 22, 2010 - Sep 29, 2010	Live Event
SANS OnDemand	Books & MP3s Only	Anytime	Self Paced