WEP deprecated

PB/INFO/071 (10/06)

WEP, WPA or Other?

The JANET Wireless Advisory Group (WAG) strongly advises the replacement of WEP techniques with WPA or WPA2/802.11i-based security, or alternatives such as encrypted VPN and self-securing protocols like SSH, as soon as possible.

The emergence of new attack techniques against the WEP encryption standard, outlined below, lead WAG to reduce its assessment of the utility of WEP. While WEP was previously considered to have some value for real-time attack deterrence despite known security weaknesses, the new capacity to compromise WEP-protected networks from tiny samples (even single packets) in real time means that WEP can no longer be relied upon to protect the content of wireless transmissions.

WEP – Early Attacks

WEP's reuse of initialisation vectors1 opened it to dictionary attack for the recovery of the WEP key, since it effectively meant that the secret portion of a 64 bit key was only 40 bits (and a 128 bit key only 104 bits). A 40 bit keyspace made offline brute force attacks feasible with conventional hardware. The Fluhrer-Mantin-Shamir cryptanalysis2 further simplified this attack and, as implemented in tools such as AirSnort, the requirement to break a WEP key was reduced to only a few thousand packets with the same recycled IV – representing a sampled set of a few million packets. Despite wide publicity given to these attacks, standard advice was that frequent rekeying coupled with hardware that filtered known weak IVs was sufficient to evade attacks based on large packet samples.

Developments in attack techniques based on more active probing of the wireless infrastructure were equally successful. Building on simple reinjection attacks from tools such as Aireplay, in 2003 Anton Rager's WEPWedgie3 tool exploited the weakness of shared key authentication to allow packet injection to the wireless network, using spoofed deauthentication to generate the required key exchange with a valid client. Also released in 2003, the chopchop4 tool decrypts individual packets by replaying a modified version of them a byte at time and monitoring whether the access point accepts them as valid (i.e. by rebroadcasting them). At most, 256 test packets are required to guess each byte of the keystream by this method.

Taken as a whole, these attacks revealed WEP as fundamentally flawed, but frequent rekeying (avoiding the typically large datasets required to attack it) and taking a few precautions (avoiding shared key WEP and filtering known weak IVs) were believed sufficient to deter attacks in real time.

WEP – The Final Nail

A new paper by Bittau, Handley and Lackey5 entitled ‘The Final Nail in WEP’s Coffin’ was published in the summer of 2006. The fragmentation attack it describes provides a means to attack WEP-protected networks without actually recovering the key itself. The technique therefore does not require a large dataset of captured traffic to operate upon, and can potentially be launched given only a single sniffed packet. This eliminates the time and access overheads of data capture and enables real-time attacks.

The fragmentation attack relies on knowing a couple of bytes of the original plaintext. Since the headers of 802.11 packets are constant for given packet types and those types can be identified by packet length, in practice the first few bytes of every packet are known. Given known plaintext and an encrypted packet, a few bytes of the keystream used to encrypt it can be recovered. This keying material can be recycled by the attacker to encrypt an equivalent amount of his own data. Since only a few bytes could be inserted by this approach (insufficient even for an 802.11 header) it was not considered a problem. However, the contribution of this paper lies in its analysis of the interaction of WEP with the MAC layer of the 802.11 specification. The 802.11 MAC layer can reassemble up to 16 x 4 byte fragmentary packets encrypted under the same keystream. The attacker can create these fragments as described above and so has a means to inject 64 bytes of data into the network.

Given this capability it is possible to decrypt intercepted packets. The attacker uses the fragment reassembly of the 802.11 MAC layer to prepend a new destination MAC address onto the packet, persuading the access point to forward the plaintext to a device the attacker controls on the wired network behind the access point. With a little more investigation to determine the next hop router’s address on the attacked network, the prepended data can be crafted to arrange for the plaintext to be routed out to an arbitrary address on the Internet.

The same approach also enables the recovery of complete keystreams in seconds. By sending a fragmented packet of known plaintext and observing the reassembled and re-encrypted packet broadcast by the AP, a longer sample of keystream can be extracted. Repeating this with increasingly larger fragments encoded with the larger keystream samples recovered finally captures the entire 1500 byte keystream. This process can either be repeated to form a complete IV dictionary, or used in a variant of the chopchop technique referred to above to recover the plaintext of an arbitrary captured packet without having it forwarded to a co-operating host.

Finally, this technique can be used to greatly optimise most of the early attack tools, by using injected traffic to elicit the ideal traffic types required for their attacks to work efficiently.

Footnotes

1. As pointed out by Borisov, Goldberg and Wagner. Intercepting mobile communications: The insecurity of 802.11. In MOBICOM 2001, Rome, Italy, July 2001.

2. Fluhrer, Mantin and Shamir. Weaknesses in the key scheduling algorithm of RC4. In Eighth Annual Workshop on Selected Areas in Cryptography, Toronto, Canada, August 2001.

3. http://sourceforge.net/projects/wepwedgie/ [1]

4. http://www.netstumbler.org/ [2]

5. http://tapir.cs.ucl.ac.uk/bittau-wep.pdf [3]

Links
[1] http://sourceforge.net/projects/wepwedgie/
[2] http://www.netstumbler.org/
[3] http://tapir.cs.ucl.ac.uk/bittau-wep.pdf