

The Emperor Has No Cloak - WEP Cloaking Exposed

Deepak Gupta and Vivek Ramachandran
Security Research Team (Amit, Gopi, Pravin)

1. Summary:

WEP Cloaking is a recently proposed anti-WEP-cracking technique that is claiming to be the savior of legacy WLAN devices still relying on WEP encryption. The WEP Cloaking mechanism is meant to be used in Wireless Intrusion Prevention Systems (WIPS) to protect WEP encrypted networks. The WEP Cloaking technique sends spoofed WEP encrypted packets a.k.a. "chaff" into the air. These packets are specially crafted to try and confuse WEP cracking tools which subsequently would fail to crack the WEP key. In course of our talk, we will demonstrate that WEP Cloaking is no panacea; it can at best delay WEP key cracking by a few seconds. We will discuss 3 techniques: Visual Inspection, Sequence number + IV filtering, and Active Frame Replay to reliably beat WEP Cloaking. We also plan to release new tools and patches for existing ones to incorporate these techniques.

2. History of WEP Cracking

In 2001, Fluhrer, Mantin and Shamir, in their celebrated paper "Weaknesses in the Key Scheduling Algorithm of RC4", proved that WEP could be cracked using statistical attacks. A few years later, in 2004, Korek released a couple of more statistical attacks, making WEP key cracking even faster. Soon after, many WEP cracking tools were released into the public domain, making WEP cracking an absolute Script-Kiddie job. WEP cracking tools work by collecting WEP encrypted packets over the air, then run them through these statistical attack filters and try to converge to the authorized network's key. It is important to note here that not all WEP encrypted packets aid in cracking the key. Only those which contain "Weak IVs" help in the cracking process. An IV is the Initialization Vector which is transmitted in the clear with each WEP encrypted packet and is used along with the WEP key to decrypt the packet. A Weak IV is an IV which satisfies one or more of the FMS and Korek statistical attack conditions. Due to the easy availability of WEP cracking tools and their widespread usage by the hacker community, WEP was officially declared dead and the industry started adopting the WPA/WPA2 security mechanisms to stay protected against Wireless snoopers.

3. What is WEP Cloaking?

WEP Cloaking is an anti-WEP-cracking technique which aims to breathe life back into WEP and make it safe for use again. WEP Cloaking works by injecting "Chaff" packets into the air. These Chaff packets are spoofed WEP encrypted packets which are generated using a set of predetermined or entirely random keys. The MAC header would be spoofed to use addresses of the Access Points and Clients of the authorized network that the technique is intended to protect. The chaff packets thus get homogeneously mixed with authorized network's packets and it is difficult to tell them apart by glancing at a packet trace. It is important to note here that WEP Cloaking does not address other WEP vulnerabilities such as Message Modification, Replay attacks, Shared authentication flaws, Packet decoding using ICV etc.

4. Why WEP Cracking tools fail in presence of WEP Cloaking?

WEP Cracking tools “trust” the packets they see over the air. They assume that the captured packets are not “corrupted” in anyway. WEP Cloaking in principle simply attacks this notion of trust. It injects Chaff packets generated with a set of random keys to introduce “noise” into the network. Current WEP Cracking tools blindly use all packets they capture as input to the statistical attack filters and thus are misled by these Chaff packets. The Chaff packets may be specially crafted to only include Weak IV packets in order to cause a wrong bias in the cracking logic of these tools, thus causing these tools to either converge to a wrong key or give up after trying for a long time. We will demonstrate this behavior in the demo section.

5. How can WEP Cloaking be beaten?

To beat WEP Cloaking cracking tools need a way to separate the chaff packets from those that are transmitted by the actual devices. We will use Aircrack as a benchmark for our discussion as it is one of the most popular and widely used tools for WEP cracking today. We will discuss the following three techniques which will aid cracking in presence of Chaff:

5.1) Guiding Aircrack with Manual Inputs via Visual Inspection:

WEP cracking is a byte by byte process. Once the first byte of the key has been guessed, we move on to cracking the next byte of the key using the guessed value for the first byte. Thus all guessed key bytes are used in guessing the next key byte. Aircrack uses the same logic. For every byte that it guesses, Aircrack prints the votes in favor of the possibilities for that byte and chooses the “guessed byte” as the one with the highest vote. In presence of Chaffing we will demonstrate that the votes for each possibility for the byte in question show an abnormal bias towards some values. We use this anomaly to our advantage by modifying Aircrack to support user interaction while cracking each byte. The modified Aircrack prints all the possibilities for a byte and asks the user which possibility he would like to use as the “guessed key byte”. Our experiments demonstrate that apart from the case where the Chaffer uses a very large number of absolutely random keys, the modified Aircrack is able to zero down on the network key. For handling the case of large number of random keys we will use Aircrack in conjunction with the next two techniques.

5.2) Sequence Number and IV based passive filtering:

Chaff frames are essentially spoofed and do not belong to the authorized network traffic. Thus a filter which can detect spoofed packets in an 802.11 network can essentially detect these Chaff packets. Sequence Number based analysis is a well established way of detecting spoofed packets. Because the sequence number space is small and rewinds quite often, we also use IV based analysis for detecting spoofed packets. The logic behind both these techniques is that when we compare the trend of the values of the sequence numbers and IVs of an authorized network device with the Chaffer generated traffic, we will clearly see a visible difference. Using this, we can reliably separate the trace in question. Our experiments demonstrate that using Sequence Number + IV trend analysis as a pre-processor, WEP cracking tools such as Aircrack are easily able to crack the key in presence of Chaffing. We have found

this technique to work well even in presence of a Chaffer that uses a large number of randomly changing keys.

5.3) Active Frame Replay based filtering:

The idea behind this technique is that when a authorized network device receives a WEP encrypted packet addressed to it, it tries to decrypt it and checks whether the decryption succeeded by testing the decrypted packet against the Integrity Check Value (ICV) inside the packet. If the ICV does not match, the packet is discarded. If it matches the packet is accepted. Only a packet encrypted with the authorized network key will pass this test. This fact can be used to filter out Chaff packets from the captured packet stream, since these packets when received by an authorized network device will be discarded as the ICV check will fail. Whenever a Weak IV packet is encountered while cracking the key we will selectively replay these packets to an authorized AP by spoofing the source MAC of an authorized client and changing the destination MAC as broadcast or a chosen multicast address. If we see a packet being replayed back to the same multicast address then it can be safely inferred that the packet in question is legitimate and not Chaff. Our experiments demonstrate that by using this technique 100% Chaff separation is easily possible. Note that replaying all packets is not needed; only those packets that potentially influence the decision making of the cracking logic need to be replayed.

We will, in our presentation condense the above 3 techniques together and talk about the design of a Chaff resistant Aircrack.

6. Final Verdict on WEP Cloaking

WEP is dead and any attempt to revive it, will meet a similar fate. Our techniques prove beyond doubt that WEP cloaking can be reliably and consistently beaten no matter what the complexity of the Chaffing Engine.

References:

1. Vendor aims to 'cloak' WEP
<http://www.networkworld.com/news/2007/032907-air-defense-wep-wireless-devices.html?page=1>
2. The TJX breach using Wireless
<http://www.emailthis.clickability.com/et/emailThis?clickMap=viewThis&etMailToID=2131419424>
3. RC4 stream Cipher basics
<http://en.wikipedia.org/wiki/RC4>
4. Wired Equivalent Privacy (WEP)
http://en.wikipedia.org/wiki/Wired_Equivalent_Privacy
5. Weaknesses in the Key Scheduling Algorithm of RC4, Selected Areas in Cryptography, 2001 - Fluhrer, Mantin and Shamir
http://www.wisdom.weizmann.ac.il/~itsik/RC4/Papers/Rc4_ksa.ps
6. Korek's post on Netstumbler
<http://www.netstumbler.org/showpost.php?p=89036>
7. WEP Dead Again: Part 1 – Infocus, Securityfocus.com
<http://www.securityfocus.com/infocus/1814>
8. WEP Dead Again: Part 2 – Infocus, Securityfocus.com
<http://www.securityfocus.com/infocus/1824>

9. Aircrack-ng : WEP Cracker
<http://www.aircrack-ng.org/>
10. Aircsnort : WEP Cracker
<http://airsnort.shmoo.com/>
11. Pcap2air : Packet replay tool
<http://www.802.11mercenary.net/pcap2air/>
12. Chop-Chop : Packet decoder using WEP ICV flaw
<http://www.netstumbler.org/showthread.php?t=12489>
13. Intercepting Mobile Communications: The Insecurity of 802.11 – N.Borisov
<http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf>
14. Your 802.11 Wireless Network has No Clothes – William Arbaugh
<http://www.cs.umd.edu/~waa/wireless.pdf>
15. Detecting Detectors: Layer 2 Wireless Intrusion Analysis – Joshua Wright
<http://home.jwu.edu/jwright/papers/l2-wlan-ids.pdf>
16. Detecting WLAN MAC address spoofing – Joshua Wright
<http://home.jwu.edu/jwright/papers/wlan-mac-spoof.pdf>
17. WPA/WPA2 the replacement for WEP
<http://en.wikipedia.org/wiki/WPA2>
18. AirDefense Perpetuates Flawed Protocols – Joshua Wright
<http://edge.arubanetworks.com/blog/2007/04/airdefense-perpetuates-flawed-protocols>