Technical Science

UDC 004.056.53

**Rvach Dmytro**

*Student of the Faculty of Informatics*

*and Computer Science of the*

*National Technical University of Ukraine*

*"Igor Sikorsky Kyiv Polytechnic Institute"*

# WIRELESS ACCESS POINTS VULNERABILITIES AND PROTECTION METHODS FOR THEM

***Summary.*** *Presently we have widely common technologies for wireless access to global networks which allow us to exchange different (sometimes even valuable) information. That's why the question of them and their errors, problems, vulnerabilities is very actual and important in our times. Because all their problems sometimes may became a big or even fatal problem for usual user. In this article will be described and shown: vulnerabilities in such technologies as WPS and WPA, possibilities of their exploitation for gaining access or for the bypassing of the available protection provided by the access system, and also methods for protection your own wireless networks and your personal information from illegal access by using these vulnerabilities.*

***Key words:*** *information security, wireless network, Wi-Fi, wireless access point, vulnerability, data transferring.*

Scientific and technological progress moves presently so quickly that we don't even think about that how our daily technologies exactly work. And we can't even imagine what serious problems and threats we can get and also what unexpected negative consequences will be next if we don't to follow some simple rules for safe using and security of our wireless access points.

Let's imagine that you have a device which can adopt commands. Different users can connect to this device and send different commands. But you should separate users who can control this device.

The first variant is checking them according to password. User must transfer certain password and then he can send new commands. And device will perform only those instructions which have correct password. This method has one trouble because password is sent "as is" and everyone can catch it then uses it.

For preventing similar situation it's necessary to use double-side encoding algorithm where our password will be represented as private key. However, it won't fix a problem because if anyone steals password then it will be available to decode all data were transferred and also encode own data which will be look like as from user. Passwords are designed for humans, and humans aren't practice 256 bytes in every symbol.

And what are the other options? We will create lengthy (lengthy enough that it couldn't be cracked) random string at first connection. Then we will encrypt all data with help of this "alias" for our password and also sometimes alter this string.

The first two transmissions are called "handshake" phase when we tell server about us at first, by presentation correct password. And then server responds by random string which we will use for encoding and sending any data. Malefactors will need very high computing capabilities for "brute forcing" this password. This necessity is existed because random keys may have any length and have all 256 codes.

Malefactors' task anyway comes down to interception handshake because it contains temporal key or password. In general, this is quite long task. Instead, this option we have Wi-Fi Protected Setup (WPS) technology. This technology allows don't use password and just press button for directly connection to your network. Hackers can use it as method of bypassing password protection.

WPS allows connect to wireless access point only by 8-digit (PIN). But it has some error which allows use only 4 of them for connection. This is about 10000 attempts of "brute forcing". And then you will get connection to access point and its password too. Malefactors can send 15-60 requests per second and in few hours they will get expected result [2]. Some devices have limited attempts after which WPS will be turned-off for some period. But such access points are very rare.

In case of switched-on WPS your password will be cracked. If you really require WPS technology, you should switch-on it only when you want to connect.

Ten years ago, we also had another vulnerable technology which called WEP and which is very rare now because we have WPA.

Wireless Fidelity Protected Access (WPA) technology is a new generation which much succeeded than WEP. It has qualitatively another protection level than WEP. Password length is varies from 8 to 63 bytes.

WPA is also differed from WEP by separately encoding data for each user. After handshake, temporal key (PTK) is used for data encryption only for this user. If someone penetrated into your network by using your handshake, he won't be able to read data of other users because he also will need their handshakes.

WPA (2) also can use two authentication modes: PSK or Enterprise. PSK (Personal) has only one password for all users. But if you have big corporation, you will get some problems. Let's imagine that you decide that one of your employees couldn't get a network access anymore. In this situation, you should setup new password and tell it to other. Enterprise has various passwords which are gathered together on RADIUS server.

So again, malefactors need to intercept 4 initial packets which send to each other by user and access point for connection establishment. These packets are "handshake". Encrypt data is start transferring after them.

After interception, hackers can save them and then start offline-attack by trying to find the initial password by generation different combinations for PMK

> PTK > MIC and matching last one to that which was actually transferred [1]. Malefactors usually use airodump-ng tool for interception:

*airodump-ng monitoring_interface -c chanel_number --bssid attacked_bssid -w file_to_save*

This command saves packets in libpcap format file which is existed in many libraries on all Operating Systems.

Let's say that attacked network BSSID is AA:BB:CC:DD:EE:FF and it's on channel 1. Then malefactor use:

*airodump-ng mon0 -c 1 --bssid AA:BB:CC:DD:EE:FF –w handshake.cap [5]*

He also can make this process much faster by disconnecting available clients and force them to transmit authentication data again. He can perform this action by the aireplay-ng tool:

*aireplay-ng mon1 -0 5 –a AA:BB:CC:DD:EE:FF -c 11:22:33:44:55:66*

In previous command 11:22:33:44:55:66 is a client's MAC address [4].

Malefactors got packets with MIC and authorization data. Now it's necessary to "brute force" password by comparing which one will be suitable for those packets.

"Brute forcing" requires a lot of computational capabilities, but it's possible just for passing packets through the wordlist with help of usual CPU. The most common way of "brute force" is aircrack-ng tool which use only the CPU, but totally maintains multithreading:

*aircrack-ng –w /path/to/your/wordlist.txt handshake.cap [3]*

Intel i7 GHz CPU cans "brute force" about 4350 passwords per second with help aircrack-ng tool. And we can calculate which time it will be required for complete search of all imaginable 8-digits combinations:

$$\frac{10^8}{4350 * 3600} = 6{,}386\ hours.$$

There are explanations of the formula from above:

— $10^8$ is an amount of potential combinations; it's determined as $potential\ combinations^{string\ length}$. For example, for the 8-unit

password which include only Latin symbols in a lower case it will be 26 8 ≈ 208 827 064 576 combinations. This fact shows that length of the password more important, than probable number of symbols in it;

— 4350 — is an amount of comparing passwords in a second ("brute force" speed);

— 3600 — is coefficient for converting the result to "per second" (minute * hour).

Graphic chips have different architecture than CPU. If in case of CPU we use about 2, 4, 6 kernels (excluding server solutions), then in case of GPU it's about 2,5 thousand separate kernels. It's needed to assist operations with large arrays which are represented by matrixes during graphics processing. Similar operations are also needed in cryptography. That's why GPU can be used for computation of hashes or for "mining" in case with cryptocurrencies.

Considering that fact the hashcat is more global tool and that's why it accepts only own file format, which has *.hccap extension, instead of the standard libpcap file.

The process of transformation from *.cap to *.hccap does in such way:

— removing from file all packets, exclude handshake;

— transforming the resulting file into *.hccap format.

These actions will be performed by next steps:

*wpaclean cleanHandshake.cap handshake.cap*

*aircrack-ng cleanHandshake.cap -J hashcat [3]*

The resulting hashcat.hccap is passed to hashcat/oclhashcat/cudahashcat (malefactors can choose one of them according with their GPU which they want to use for this action).

Malefactors can start search by dictionary in this way:

*hashcat -m2500 cleanHandshake.hccap /path/to/wordlist.txt*

In previous command -m2500 is an option for execute WPA-handshakes cracking [6].

The possibility of computation hashes by GPU lets us know as far as everything is unsafe and how fast our short and "hard" passwords will be cracked. It doesn't belong only for WPA and wireless technologies in general, but for web resources too.

Malefactors also have the third solution for similar actions which called pyrit. Pyrit is an open source instrument for "brute forcing" WPA passwords with help CPU and GPU together. It was created on Python and it has interpretation for different CPUs and for two vendors of GPUs, but it works only with WPA passwords. Hashcat developers explain that performance of dual using CPU and GPU is minimum and it's so hard to realize.

We can't be sure at reliability of digital passwords which less than 21 symbols long or alphabetic passwords which less than 16 symbols long. The variation between 12-digit numerical and literal passwords is duration of "brute forcing" process which will be for a half an hour more for a literal password. That's why anyone of us isn't insured from illegal using our wireless access points and stealing a lot of our personal data. But there are several methods which can minimize chances of similar unexpected situations.

Here are some of them which will be more effective against many vectors of wireless attacks:

— use only WPA2-PSK-CCMP long (more than 12 characters) and complicated passwords (their complexity can be checked by yourself on some special web resources), and it will take thousands years to "brute force" it;
— never use vulnerable WPS technology;
— never tell your password to the unchecked persons and/or don't place it in open access;
— periodically (as often as possible) cardinally change your password for a new one which doesn't similar to previous (this rule will exclude a possibility of "brute forcing" your password by the systems which have

very high computing capabilities allowing finding even difficult passwords for a rather small period).

Such easy ways will help you to protect as much as possible your wireless access point and also your personal data which presently are very expensive and desired by many third parties.

## References

1. "Exposing WPA2 security protocol vulnerabilities" in Int. J. Information and Computer Security, 2014, 6, 93-107.
2. Viehbock, Stefan (26 December 2011). "Brute forcing Wi-Fi Protected Setup" Retrieved from https://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf.
3. Aircrack-ng documentation http://www.aircrack-ng.org Retrieved from http://www.aircrack-ng.org/doku.php?id=aircrack-ng.
4. Aireplay-ng documentation http://www.aircrack-ng.org Retrieved from http://www.aircrack-ng.org/doku.php?id=aireplay-ng.
5. Airodump-ng documentation http://www.aircrack-ng.org Retrieved from http://www.aircrack-ng.org/doku.php?id=airodump-ng.
6. Hashcat documentation https://hashcat.net Retrieved from https://hashcat.net/wiki/doku.php?id=hashcat.