Re-Breaking Wireless Protected Setup

Leonardo Maccari
DISI - University of Trento, Italy
Email: leonardo.maccari@unitn.it

Matteo Rosi
Phoops S.R.L. Firenze, Italy
Email: matteo.rosi@phoops.it

***Abstract*—Wireless Protected Setup (WPS) is a recent authentication mechanism introduced by the Wi-Fi Alliance in order to simplify the cumbersome authentication procedure specified by previous Wi-Fi standards. It reduces the authentication to a simple exchange of a token that, in the most simple configuration, is a numerical PIN. It has been shown that using this simple configuration it is possible to reveal the PIN with an on-line brute force attack . The attack can be mitigated introducing increasing delays between two consecutive authentication attempts. This paper has a double focus, the first is to describe and explain the existing attack, since documentation on this topic is scarce in scientific literature. The second is the introduction of an extension of the attack that can be successfully run even when the access point implements the suggested countermeasure.**

I. Introduction

The introduction of IEEE 802.11i solved many security problems that were present in the previous versions of 802.11. The WPA2-Personal (*Wireless Protected Access* with shared key) left to the attacker only the chance of off-line brute force attacks. To avoid that risk the user must configure his terminal with a strong shared key, which is unpractical in certain situations (i.e. keyboard-less devices or even smartphones).

To reduce this burden the WPS (Wireless Protected Setup) protocol has been recently proposed by the Wi-Fi alliance [1]. WPS introduces a new negotiation that can be carried on with out-of-band channels such as near field communications or the exchange of crypto tokens via USB. Alternatively an in-line verification of a numeric PIN between the AP and the client can be used. We focus on this last option, since it is the only one for which the specifications have been completed and it is currently implemented in many access points (APs).

WPS exchange is performed only the first time that a client connects to an AP: on their first contact the two entities will have to demonstrate each other the knowledge of a numeric PIN which can be generated on the fly or can be statically configured in the AP. In the first case the AP must be equipped with a LCD screen, it will generate the PIN, the user will read it from the screen and will enter it in its terminal. In the second case, the PIN is generally pre-configured and printed on the AP. Once the PIN has been verified the AP will move to the client all the configuration parameters (essentially the WPA key and name of the network) and WPA authentication can take place. From that moment on the client will use standard WPA authentication when re-entering the same network. WPS proves to be useful for printers, scanners and keyboard-less devices and is also part of the Wi-Fi direct specification, a Wi-Fi alliance protocol imagined to allow direct connection between devices, primarily smartphones. Wi-Fi direct is supported by the latest mobile operative systems.

WPS has been cracked in December 2011 with an on-line brute force attack, showing that the static PIN can be revealed with less than 11.000 attempts. If an AP is not equipped with a LCD screen the only possible countermeasure (other than switching the feature off) is to increase the timeout between two authentication sessions after a certain number of failures.

The attack is perfectly feasible and there are open source tools available to carry it on. We will describe it and we will improve it, showing that if the attacker is able to intercept the first authentication of a client, the brute force can be transformed in an *oracle attack* which can invalidate the countermeasures implemented by the vendors.

II. Description of the attacks

In figure 1 is reported the handshake used by WPS when a static PIN is used. In this case, using WPS notation, the client takes the role of the *Enrollee* while the AP is called *Registrar*. This choice of names is due to the fact that WPS is a peer-to-peer protocol, so both parties can play the role of a Registrar, in the case of static PIN it is the client. The PIN is an 8-digit number with the last one used as a checksum, it is split in two halves P_1 and P_2. WPS packets are sent after the standard 802.11 authentication and association plus an EAP Identity Request/Response exchange, summarized in fig. 1 with phase 0. Phase 1 continues with the exchange of public keys used for the Diffie Hellman algorithm. The generated symmetric session key K is used to encrypt critical data in order to avoid an off-line brute force attack by a passive attacker. The field named *Auth* is an authentication token produced using K and an HMAC function on the current and previous packet. In phase 2 the AP sends two hashes (E-Hash1 and E-Hash2) to the client. They are HMAC computed on a number of fields, but mainly depend on: P_1 and P_2 respectively, K, and two random numbers E-S1 and E-S2 respectively. E-S1 and E-S2 are generated by the AP and unknown to the client in this phase. The client in phase 3 mirrors this packet and sends two HMAC R-Hash1 and R-Hash2 generated using P_1 and P_2 respectively, K, and two random numbers R-S1 and R-S2 respectively. It also sends R-S1 encrypted with K. The AP is now able to use R-S1 to recompute R-Hash1 and verify that the client knows P_1. The AP now partially trusts the client while

Financed by Provincia di Trento under *The Trentino programme of research, training and mobility of post-doctoral researchers, incoming Post-docs 2010* CALL 1, PCOFUND-GA-2008-226070

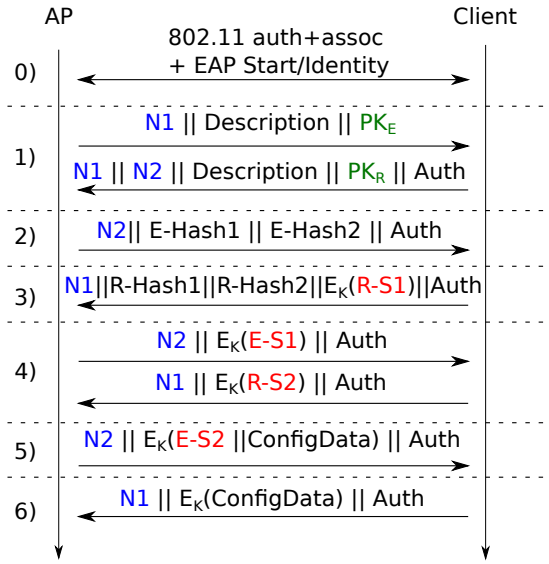

Fig. 1. The WPS authentication procedure, blue labels represent nonces, green labels represent public keys, red labels represent generated symmetric keys. The exchange has been divided in its main phases. $E_k()$ indicates encryption using the generated session key K.

the client has received no proof. The AP in phase 4 reveals E-S1 to the client, which can verify E-Hash1. Now also the client has the proof that the AP knows P_1. With the same principle then the client reveals R-S2 and, after a check on P_2 also the AP reveals E-S2. At the end of a correct session the AP moves encrypted configuration parameters on the client.

The rationale of the choice of splitting the PIN in two is probably due to the intention of avoiding *oracle attacks* in which the attacker is able start an authentication and collect enough information to brute force the secret key before it has to show to the other peer the knowledge of the shared secret himself. This *poor-man zero-knowledge proof* led to a perfectly feasible on-line brute force attack as described and implemented in [2]. The attacker initializes $P'_1 = 0000$ and $P'_2 = 000$ and performs the protocol up to phase 2, then computes R-Hash1 and R-Hash2 using P'_1 and P'_2 as the PIN. It waits for an answer from the AP, in case of a NAK it will increment P'_1 and start again. In case of an ACK message then $P_1 = P'_1$ and P_1 is revealed. The running authentication will probably fail since R-Hash2 has been generated using $P'_2 = 000$ but now the attacker can continue the brute force in phase 4.

A brute force attack on a 7 digit pin would require an average of $10^7/2$ attempts. Instead, the brute force can be split in two attacks with an average of $10^4/2 + 10^3/2$ attempts that require a few hours. To mitigate this attack vendors introduced a timeout after each authentication failure to make the attack too long to be practically feasible [3]. This comes at the cost of exposing the AP to a denial of service attack, since the attacker can trigger false authentications using spoofed MAC addresses to force the AP to deny any legitimate client.

In this case, we found out that the attack can be easily inverted if the attacker is able to interfere with the first WPS authentication of a client and impersonate the AP (since there is no control on the public key of the AP). The attacker initializes $P'_1 = 0000$ and $P'_2 = 000$ and sets its MAC address to the same one of the AP. When the client performs phase 0 there will be a race condition with the real AP to be the first to answer. The attacker can win the race condition using a very short contention window in 802.11 MAC layer (open source drivers support this feature). In phase 2 the attacker will use P'_1 and P'_2 to generate E-Hash1 and E-Hash2, that the client can not yet verify. Thus, the attacker is able to get to phase 3 and receive R-Hash1 and R-S1 from the client. The attacker can then perform an off-line brute force on R-Hash1 and can thus obtain P_1. The brute force attacks requires at most 10^4 attempts so it is perfectly feasible. The following phase 4 will fail, since E-Hash1 has been computed with $P'_1 = 0000$ and the client will answer with an error message. One error in an authentication procedure is perfectly tolerable by the software drivers and by the users, so the procedure will start again. The attacker will impersonate the AP again, set $P'_1 = P_1$ and correctly complete phase 4. In phase 4 the attacker will receive R-S2 and perform another off-line brute force attack against the remaining 3 digit of P_2. The attacker now can successfully authenticate itself with the real AP and receive a correct WPA key to enter the network. In case this key is shared by all the clients he will also be able to decrypt the traffic of other clients, after forcing them to a new WPA authentication.

We have verified the feasibility of the attack (without implementing the race-condition code) using a modified version of the widely used Linux-based hostAPd software. The modified code is available on www.pervacy.eu.

III. CONCLUSIONS

WPS is a fairly new authentication algorithm supported by many access points, vulnerable to an on-line brute force attack. As a partial fix, vendors implemented timers that can make the attack unpractical at the price of exposing the AP to a DoS attack. We have shown that if the attacker is able to impersonate the AP when a client authenticates for the first time he can obtain the PIN with just two runs of the protocol. Since the attack is against a client, the timers have no influence over it while they still leave the possibility of a DoS. The only effective countermeasure is to switch WPS with static PIN off. We are now evaluating the possibility of an attacker to trigger a client WPS re-authentication in order to perform the attack even when there is no new client joining the network.

REFERENCES

[1] "Wi-fi CERTIFIED wi-fi protected setup: Easing the user experience for home and small office wi-fi networks." [Online]. Available: https://www.wi-fi.org/knowledge-center/white-papers

[2] "Wi-fi protected setup pin brute force vulnerability." [Online]. Available: http://sviehb.wordpress.com/2011/12/27/wi-fi-protected-setup-pin-brute-force-vulnerability/

[3] "Cisco security response: Wi-fi protected setup PIN brute force vulnerability." [Online]. Available: http://tools.cisco.com/security/center/content/CiscoSecurityResponse/cisco-sr-20120111-wps