Brute forcing Wi-Fi Protected Setup

When poor design meets poor implementation.
Introduction

"Wi-Fi Protected Setup™ is an optional certification program from the Wi-Fi Alliance that is designed to ease the task of setting up and configuring security on wireless local area networks. Introduced by the Wi-Fi Alliance in early 2007, the program provides an industry-wide set of network setup solutions for homes and small office (SOHO) environments.

Wi-Fi Protected Setup enables typical users who possess little understanding of traditional Wi-Fi configuration and security settings to automatically configure new wireless networks, add new devices and enable security. More than 200 products have been Wi-Fi CERTIFIED™ for Wi-Fi Protected Setup since the program was launched (sic!) in January 2007."

The Wi-Fi Simple Configuration Specification (WSC) is the underlying technology for the Wi-Fi Protected Setup certification.

Almost all major vendors (including Cisco/Linksys, Netgear, D-Link, Belkin, Buffalo, ZyXEL and Technicolor) have WPS-certified devices, other vendors (eg. TP-Link) ship devices with WPS-support which are not WPS-certified.

WPS is activated by default on all devices I had access to.

Although WPS is marketed as being a secure way of configuring a wireless device, there are design and implementation flaws which enable an attacker to gain access to an otherwise sufficiently secured wireless network.

Configuration Options Overview

WPS supports out-of-band configuration over Ethernet/UPnP (also NFC is mentioned in the specification) or in-band configuration over IEEE 802.11/EAP. Only in-band configuration will be covered in this paper.

Terminology

- The enrollee is a new device that does not have the settings for the wireless network.
- The registrar provides wireless settings to the enrollee.
- The access point provides normal wireless network hosting and also proxies messages between the enrollee and the registrar.

---

2. [http://download.microsoft.com/download/a/f/f/af777e5-7dcd-4800-8a0a-b18336565f5b/WCN-Netspec.doc](http://download.microsoft.com/download/a/f/f/af777e5-7dcd-4800-8a0a-b18336565f5b/WCN-Netspec.doc)
Push-Button-Connect ("PBC")

The user has to push a button, either an actual or virtual one, on both the Access Point and the new wireless client device. PBC on the AP will only be active until authentication has succeeded or timeout after two minutes.

This Option is called wps_pbc in wpa_cli\(^3\) (text-based frontend program for interacting with wpa_supplicant).

![Figure 1: activated "virtual Push Button" (Windows acts as enrollee) (Windows 7)](image1)

![Figure 2: Description of PBC option (Linksys WRT320N User Manual)](image2)

PIN

Internal Registrar

The user has to enter the PIN of the Wi-Fi adapter into the web interface of the access point. The PIN can either be printed on the label of the adapter or generated by software.

This option is called wps_pin in wpa_cli.

![Figure 3: Description of PIN internal Registrar option (Linksys WRT320N User Manual)](image3)

![Figure 4: PIN field – Router is Registrar (Linksys WRT320N Web Interface)](image4)

---

\(^3\) [http://hostap.epitest.fi/wpa_supplicant/](http://hostap.epitest.fi/wpa_supplicant/)
External Registrar
The user has to enter the PIN of the access point into a form on the client device (e.g. computer).

This option is called **wps_reg** in wpa_cli.

**Method #3**
Use this method if your client device asks for the Router’s PIN number.

1. Enter the PIN number listed on this screen. (It is also listed on the label on the bottom of the Router.)
2. After the client device has been configured, click OK. Then refer back to your client device or its documentation for further instructions.

Figure 5: Description of PIN external Registrar option (Linksys WRT320N User Manual)

Figure 6: Windows Connect Now Wizard acting as a Registrar (Windows 7)

Figure 7: Label with WPS PIN on the back of a D-Link router

**Design Flaw #1**

<table>
<thead>
<tr>
<th>Option / Authentication</th>
<th>Physical Access</th>
<th>Web Interface</th>
<th>PIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push-button-connect</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PIN – Internal Registrar</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PIN – External Registrar</td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

WPS Options and which kind of authentication they actually use.

As the External Registrar option does not require any kind of authentication apart from providing the PIN, it is potentially vulnerable to brute force attacks.
## Authentication (PIN – External Registrar)\(^4\)

<table>
<thead>
<tr>
<th>IEEE 802.11</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppliant → AP</td>
<td>Authentication Request</td>
</tr>
<tr>
<td>Suppliant ↔ AP</td>
<td>Authentication Response</td>
</tr>
<tr>
<td>Suppliant → AP</td>
<td>Association Request</td>
</tr>
<tr>
<td>Suppliant ↔ AP</td>
<td>Association Response</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IEEE 802.11/EAP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppliant → AP</td>
<td>EAPOL-Start</td>
</tr>
<tr>
<td>Suppliant ↔ AP</td>
<td>EAP-Request Identity</td>
</tr>
<tr>
<td>Suppliant → AP</td>
<td>EAP-Response Identity (Identity: &quot;WFA-SimpleConfig-Registrar-1-0&quot;)</td>
</tr>
</tbody>
</table>

### IEEE 802.11/EAP Expanded Type, Vendor ID: WFA (0x372A), Vendor Type: SimpleConfig (0x01)

| M1 | Enrollee → Registrar | N1 || Description || PK_A | Diffie-Hellman Key Exchange |
| M2 | Enrollee ↔ Registrar | N1 || N2 || Description || PK_A || Authenticator |
| M3 | Enrollee → Registrar | N2 || E-Hash1 || E-Hash2 || Authenticator |
| M4 | Enrollee ↔ Registrar | N1 || R-Hash1 || R-Hash2 || E_{keyWrapKey}(R-S1) || Authenticator |

\( E_{keyWrapKey} = \text{Stuff encrypted with KeyWrapKey (AES-CBC)} \)

| M5 | Enrollee → Registrar | N2 || E_{keyWrapKey}(E-S1) || Authenticator | prove possession of 1\(^{st}\) half of PIN |
| M6 | Enrollee ↔ Registrar | N1 || E_{keyWrapKey}(R-S2) || Authenticator | prove possession of 2\(^{nd}\) half of PIN |
| M7 | Enrollee → Registrar | N2 || E_{keyWrapKey}(E-S2) || ConfigData || Authenticator | prove possession of 2\(^{nd}\) half of PIN, send AP configuration |
| M8 | Enrollee ↔ Registrar | N1 || E_{keyWrapKey}(ConfigData) || Authenticator | set AP configuration |

---

\( PK_A = \text{Diffie-Hellman Public Key Enrollee} \)

\( PK_L = \text{Diffie-Hellman Public Key Registrar} \)

\( Authkey \) and KeyWrapKey are derived from the Diffie-Hellman shared key.

\( Authenticator = \text{HMAC}_{\text{Authkey}}(\text{last message} || \text{current message}) \)

---

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(^{st}) half of PIN</td>
<td>checksum</td>
<td>2(^{nd}) half of PIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If the WPS-authentication fails at some point, the AP will send an EAP-NACK message.

---

\(^4\) based on [http://download.microsoft.com/download/a/f/7/af7777e5-7dcd-4800-8a0a-b18336565f5b/WCN-Netspec.doc](http://download.microsoft.com/download/a/f/7/af7777e5-7dcd-4800-8a0a-b18336565f5b/WCN-Netspec.doc)
**Design flaw #2**

An attacker can derive information about the correctness of parts the PIN from the AP’s responses.

- If the attacker receives an EAP-NACK message after sending M4, he knows that the 1st half of the PIN was incorrect.
- If the attacker receives an EAP-NACK message after sending M6, he knows that the 2nd half of the PIN was incorrect.

This form of authentication dramatically decreases the maximum possible authentication attempts needed from $10^8$ ($=100.000.000$) to $10^4 + 10^4$ ($=20.000$).

As the 8th digit of the PIN is always a checksum of digit one to digit seven, there are at most $10^3 + 10^3$ ($=11.000$) attempts needed to find the correct PIN.

**Brute Force Methodology**

![Flowchart](image)

Figure 8: Flowchart showing how an optimized brute force attack works
Brute Force Implementation
A proof-of-concept brute force tool was implemented in Python. It uses the Scapy\(^5\) Library for decoding, generating, sending and receiving packets. This tool was used on several routers made by different vendors.

Sample output
sniffer started

trying 0000000
attempt took 0.95 seconds
trying 0081000
attempt took 1.28 seconds
trying 0082000
attempt took 1.03 seconds

<snip>

trying 18660005
attempt took 1.08 seconds
trying 18670004 # found 1st half of PIN
attempt took 1.09 seconds
trying 18670011
attempt took 1.08 seconds
trying 18670028
attempt took 1.17 seconds
trying 18670035
attempt took 1.12 seconds

<snip>

trying 18674071
attempt took 1.15 seconds
trying 18674088
attempt took 1.11 seconds

trying 18674095 # found 2nd half of PIN
E-52:
0000 16 F6 82 CA A8 24 7E 98 85 4C BD A6 BE D9 14 50
......$-..L......P
SSID:
0000 74 70 2D 74 65 73 74
tp-test
MAC:
0000 F4 EC 38 CF AC 2C
..8..
Auth Type:
0000 00 20
.
Encryption Type:
0000 00 08
.
Network Key:
0000 72 65 61 6C 6C 79 5F 72 65 61 6C 6C 79 5F 6C 6F really_really_lo
0010 6E 67 77 70 70 61 5F 70 61 73 73 70 68 72 61 73 ng_wpa_passphrases
0020 65 5F 67 6F 6F 64 56 6C 6F 67 73 72 61 63 e_good_luck_crac
0030 68 69 66 6E 67 5F 74 68 69 73 5F 6F 6E 65 king_this_one
Key Wrap Algorithm:
0000 76 3C 7A 87 0A 7D F7 E5
<\z..>
Results

Authentication attempt duration
One authentication attempt usually took between 0.5 and 3 seconds to complete. It was observed that the calculation of the Diffie-Hellman Shared Key (needs to be done before generating M3) on the AP took a big part of the authentication time. This can be speeded up by choosing a very small DH Secret Number, thus generating a very small DH Public Key and making Shared Key calculation on the AP’s side easier.

Implementation Flaws
Some vendors did not implement any kind of blocking mechanism to prevent brute force attacks. This allows an attacker to try all possible PIN combinations in less than four hours (at 1.3 seconds/attempt).

On average an attack will succeed in half the time.

The Netgear device has lock down functionality implemented, but the lock down phases are not long enough to make an attack impractical. In this case an attack will on average succeed in less than a day (timing data can be found on the next page).

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Device Name</th>
<th>HW-Version</th>
<th>FW-Version</th>
<th>Lock down</th>
<th>WPS-certified</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-Link</td>
<td>DIR-655</td>
<td>A4 (Web Interface)</td>
<td>1.35</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Linksys</td>
<td>WRT320</td>
<td>1.0</td>
<td>1.0.04</td>
<td>?[^6]</td>
<td>Yes</td>
</tr>
<tr>
<td>Netgear</td>
<td>WGR614v10</td>
<td>?</td>
<td>1.0.2.26</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>TP-Link</td>
<td>TL-WR1043ND</td>
<td>1.8</td>
<td>V1_110429</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Firmware versions are up-to-date as of 18.10.2011.

In rare cases devices started to send malformed messages or their web interface and routing did not work properly anymore. A reboot was needed to solve the problem. This might be evidence of some kind of corruption, but was not investigated further.

[^6]: WPS-functionality always stopped to work somewhere between 2 and 150 failed authentication attempts. The functionality did not even return after several hours. I would consider this a bug in the firmware which causes a DoS rather than lock-down functionality.
Mitigations

End users
Deactivate WPS. This may not always be possible.

Vendors
Introduce sufficiently long lock-down periods in order to make an attack impractical. Of course this requires a new firmware release.

<table>
<thead>
<tr>
<th>Attempts before lock</th>
<th>Lock down time</th>
<th>Attempts per minute</th>
<th>Maximum attack time</th>
<th>Maximum attack time</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>11000</td>
<td>0 minutes</td>
<td>46.15</td>
<td>3.97 hours</td>
<td>0.17 days</td>
<td>no lock down</td>
</tr>
<tr>
<td>7</td>
<td>1 minutes</td>
<td>3.20</td>
<td>43.65 hours</td>
<td>1.82 days</td>
<td>Netgear WGR614v10</td>
</tr>
<tr>
<td>3</td>
<td>1 minutes</td>
<td>2.82</td>
<td>65.08 hours</td>
<td>2.71 days</td>
<td>Requirement for WSC 2.0</td>
</tr>
<tr>
<td>15</td>
<td>60 minutes</td>
<td>0.25</td>
<td>737.31 hours</td>
<td>30.72 days</td>
<td>Lock down configurations making brute force less practical</td>
</tr>
<tr>
<td>10</td>
<td>60 minutes</td>
<td>0.17</td>
<td>1103.97 hours</td>
<td>46.00 days</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>60 minutes</td>
<td>0.08</td>
<td>2203.97 hours</td>
<td>91.83 days</td>
<td></td>
</tr>
</tbody>
</table>

Assumed time per attempt: 1.3 seconds

Considering that an AP typically runs for several months, a determined attacker might still be able to successfully attack a WPS-enabled AP. This attack is low-cost and has a high success guarantee compared to cracking WPA/WPA2-PSK.

Conclusion
As nearly all major router/AP vendors have WPS-certified devices and WPS – PIN (External Registrar) is mandatory for certification, it is expected that a lot of devices are vulnerable to this kind of attack.

Having a sufficiently long lock-down period is most likely not a requirement for certification. However it might be a requirement in the (new) WSC Specification Version 2\(^6\). I contacted the Wi-Fi Alliance about this – they have yet to respond.

Collaboration with vendors will be necessary for identifying all vulnerable devices. It is up to the vendors to implement mitigations and release new firmware.

Affected end-users will have to be informed about this vulnerability and advised to disable WPS or update their firmware to a more secure version (if available).

\(^7\) No consistent lock down pattern was found. However on average about 4.20 authentication attempts per minute were possible.