t6x / **reaver-wps-fork-t6x**

No description, website, or topics provided.

273 commits | **7** branches | **7** releases | **14** contributors

Branch: **master** ▾ | New pull request | Create new file | Upload files | Find File | Clone or download ▾

rofl0r print message when we get EAP_FAILURE Latest commit 37de8c8 on Jun 21

docs	replace gzipped manpage with unzipped one	2 years ago
src	print message when we get EAP_FAILURE	2 months ago
tools	logfilter: fix typo	2 years ago
.gitignore	update .gitignore	2 years ago
README.md	wash: show crack progress with -p option (#268)	5 months ago

README.md

Overview

Reaver implements a **brute force attack** against **Wifi Protected Setup** (WPS) registrar **PINs** in order to recover **WPA/WPA2 passphrases**, as described in Brute forcing Wi-Fi Protected Setup When poor design meets poor implementation. by **Stefan Viehböck**.
Reaver has been designed to be a robust and practical attack against **Wi-Fi Protected Setup (WPS)** registrar PINs in order to **recover WPA/WPA2 passphrases** and has been tested against a wide variety of access points and WPS implementations.
Depending on the target's Access Point (AP), to recover the plain text WPA/WPA2 passphrase the **average** amount of time for the transitional **online brute force** method is **between 4-10 hours**. In practice, it will generally take half this time to guess the correct WPS pin and recover the passphrase. When using the **offline attack**, **if** the AP is vulnerable, it may take only a matter of **seconds to minutes**.

The first version of **reaver-wps** (reaver 1.0) was created by **Craig Heffner** in 2011.
reaver-wps-fork-t6x version **1.6.x** is a **community forked version** which includes **various bug fixes**, **new features** and additional attack method (such as the **offline Pixie Dust** attack).

- The original Reaver (version 1.0 to 1.4) can be found in google code archives.
- The discontinued reaver-wps-fork-t6x community edition, reaver version 1.5.3, which includes the Pixie Dust attack, is now the old-master branch from this repository.
- The latest revison of reaver-wps-fork-t6x community edition is the master branch from this repository.
  Reaver versioning was updated to **1.6.x** in order to identify the new cycle.
  All stable relases since the first beta version of reaver 1.6 can be downloaded from our Releases page.
- More information about the Pixie Dust attack (including **which APs are vulnerable**) can be found in pixiewps repository, pixie dust thread (in Kali forum) & Dominique Bongard's full disclosure

Requirements

Build-time dependencies

- libpcap-dev

- build-essential

Runtime-time dependencies

- pixiewps (optional, required for pixiedust attack)
- aircrack-ng (optional, though recommended)

Example

```
sudo apt -y install build-essential libpcap-dev aircrack-ng pixiewps
```

The example uses Kali Linux as the Operating System (OS) as `pixiewps` *is included.*

You **must** already have Wiire's Pixiewps installed to perform a pixie dust attack, latest version can be found in its official github repository.

Setup

Download

```
git clone https://github.com/t6x/reaver-wps-fork-t6x
```

or

```
wget https://github.com/t6x/reaver-wps-fork-t6x/archive/master.zip && unzip master.zip
```

Locate the shell

```
cd reaver-wps-fork-t6x*
cd src
```

Compile

```
./configure
make
```

Install

```
sudo make install
```

Reaver Usage

```
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

Required Arguments:
        -i, --interface=<wlan>          Name of the monitor-mode interface to use
        -b, --bssid=<mac>               BSSID of the target AP

Optional Arguments:
        -m, --mac=<mac>                 MAC of the host system
        -e, --essid=<ssid>              ESSID of the target AP
        -c, --channel=<channel>         Set the 802.11 channel for the interface (implies -f)
        -s, --session=<file>            Restore a previous session file
        -C, --exec=<command>            Execute the supplied command upon successful pin recovery
        -f, --fixed                     Disable channel hopping
        -5, --5ghz                      Use 5GHz 802.11 channels
        -v, --verbose                   Display non-critical warnings (-vv or -vvv for more)
        -q, --quiet                     Only display critical messages
        -h, --help                      Show help

Advanced Options:
        -p, --pin=<wps pin>             Use the specified pin (may be arbitrary string or 4/8 digit WPS
pin)
        -d, --delay=<seconds>           Set the delay between pin attempts [1]
        -l, --lock-delay=<seconds>      Set the time to wait if the AP locks WPS pin attempts [60]
        -g, --max-attempts=<num>        Quit after num pin attempts
        -x, --fail-wait=<seconds>       Set the time to sleep after 10 unexpected failures [0]
        -r, --recurring-delay=<x:y>     Sleep for y seconds every x pin attempts
        -t, --timeout=<seconds>         Set the receive timeout period [10]
        -T, --m57-timeout=<seconds>     Set the M5/M7 timeout period [0.40]
        -A, --no-associate              Do not associate with the AP (association must be done by
another application)
        -N, --no-nacks                  Do not send NACK messages when out of order packets are
received
        -S, --dh-small                  Use small DH keys to improve crack speed
        -L, --ignore-locks              Ignore locked state reported by the target AP
        -E, --eap-terminate             Terminate each WPS session with an EAP FAIL packet
        -J, --timeout-is-nack           Treat timeout as NACK (DIR-300/320)
        -F, --ignore-fcs                Ignore frame checksum errors
        -w, --win7                      Mimic a Windows 7 registrar [False]
        -K, --pixie-dust                Run pixiedust attack
        -Z                              Run pixiedust attack

Example:
        reaver -i wlan0mon -b 00:90:4C:C1:AC:21 -vv
```

Options description and examples of use can be found in the Readme from Craig Heffner. Here comes a description of the new options introduced since then:

-K or -Z // --pixie-dust

The `-K` and `-Z` option perform the offline attack, Pixie Dust *(`pixiewps`)*, by automatically passing the **PKE**, **PKR**, **E-Hash1**, **E-Hash2**, **E-Nonce** and **Authkey** variables. `pixiewps` will then try to attack **Ralink**, **Broadcom** and **Realtek** detected chipset. **Special note**: If you are attacking a **Realtek AP**, **do NOT** use small DH Keys (`-S`) option. User will have to execute reaver with the cracked PIN (option -p) to get the WPA pass-phrase. This is a temporary solution and an option to do a full attack will be implemented soon

-p with arbitrary string // --pin=

See our wiki: Introducing a new way to crack WPS: Option p with an Arbitrary String

Wash Usage

```
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner

Required Arguments:
        -i, --interface=<iface>              Interface to capture packets on
        -f, --file [FILE1 FILE2 FILE3 ...]   Read packets from capture files

Optional Arguments:
        -c, --channel=<num>                  Channel to listen on [auto]
        -n, --probes=<num>                   Maximum number of probes to send to each AP in scan mode
[15]
        -F, --ignore-fcs                     Ignore frame checksum errors
        -2, --2ghz                           Use 2.4GHz 802.11 channels
        -5, --5ghz                           Use 5GHz 802.11 channels
        -s, --scan                           Use scan mode
        -u, --survey                         Use survey mode [default]
        -a, --all                            Show all APs, even those without WPS
        -j, --json                           print extended WPS info as json
        -p, --progress                       Show percentage of crack progress
        -h, --help                           Show help

Example:
        wash -i wlan0mon
```

A detailed description of the options with concrete syntax examples can be found in Craig Heffner's wash readme. About the new options and features:

-a // --all

The option `-a` of Wash will list all access points, including those without WPS enabled.

-j // --json

The extended WPS information (serial, model...) from the AP probe answer will be printed in the terminal (in json format)

"Vendor" column

Wash now displays the manufacturer of the wifi chipset from the Acces Points in order to know if they are vulnerable to pixie dust attack.

Stdout can be piped

Notice that wash output can be piped into other commands. For more information see the wiki article Everything about the new options from wash

Acknowledgements

Contribution

Creator of reaver-wps-fork-t6x "community edition": `t6x`

Main developer since version 1.6b: `rofl0r`

Modifications made by: `t6_x` , `DataHead` , `Soxrok2212` , `Wiire` , `AAnarchYY` , `kib0rg` , `KokoSoft` , `rofl0r` , `horrorho` , `binarymaster` , `Ǹotaz`

Some ideas made by: `nuroo` , `kcdtv`

Bug fixes made by: `alxchk` , `USUARIONUEVO` , `ldm314` , `vk496` , `falsovsky` , `rofl0r` , `xhebox`

Special Thanks

- `Soxrok2212` for all work done to help in the development of tools
- `Wiire` for developing Pixiewps
- `Craig Heffner` for creating Reaver and for the creation of default pin generators (D-Link, Belkin) - http://www.devttys0.com/
- `Dominique Bongard` for discovering the Pixie Dust attack.