Deliverable D3.8
5G-PPP Security Enablers Documentation (v2.0)
Enabler: Device Identifier Privacy

Project name	5G Enablers for Network and System Security and Resilience	
Short name	5G-ENSURE	
Grant agreement	671562	
Call	H2020-ICT-2014-2	
Delivery date	31.08.2017	
Dissemination Level:	Public	
Lead beneficiary	NEC	Felix Klaedtke, felix.klaedtke@neclab.eu
Authors	UOXF: Piers O'Hanlon	

Document Version	Date	Change(s)	Author(s)
0.1	28.07.2016	Created template	Felix Klaedtke
0.1.1	28.07.2016	Initial version	Piers O'Hanlon
0.2	02.09.2016	Removed comments and updated log	Piers O'Hanlon
0.3	16.09.2016	Updated after peer review comments	Piers O'Hanlon
0.4		Replaced phone icon in Fig 1	
2.0	24.07.2017	Updated for R2	Piers O'Hanlon
2.0.1	28.07.2017	Updated template	Piers O'Hanlon
2.0.2	03.08.2017	Added extra config option	Piers O'Hanlon

Foreword

5G-ENSURE belongs to the first group of EU-funded projects which collaboratively develop 5G under the umbrella of the 5G Infrastructure Public Private Partnership (5G-PPP) in the Horizon 2020 Programme. The overall goal of 5G-ENSURE is to deliver strategic impact across technology and business enablement, standardisation and vision for a secure, resilient and viable 5G network. The project covers research & innovation - from technical solutions (5G security architecture and testbed with 5G security enablers) to market validation and stakeholders engagement - spanning various application domains.

This is the manual for the Device Identifier Privacy Enabler which has been specified in D3.6. It provides for enhanced network attachment privacy and reduced attachment latency for devices utilising non-3GPP access to 5G networks.

Disclaimer

Copyright notice

Contents

1 Introduction

The Device Identifier Privacy (DIP) enabler provides privacy enhanced network attachment, offering protection against unauthorized device tracking and device location disclosure. The main focus is to offer improved privacy protection of device identifiers, on IP-based networks, for access to 5G services via non-3GPP access mechanisms, such as via Generic Access Network (GAN) [1], or to operator services. It also provides enhanced device identity privacy for access to third party Internet-based resources when utilising authentication schemes (e.g. EAP-AKA [2] for WiFi). These privacy mechanisms are built upon the Detection of Network Attachment (DNA) [3] protocol which provides for reduced handover latency when moving and reattaching between points of attachment.

2 Installation and Administration Guide

This section covers the system requirements, and describes how the enabler is installed, configured, and administered.

2.1 System Requirements

The system requirements for this enabler are that the target client device should be running Linux with a WiFi network interface. The client device runs the enabler which implements a privacy enhanced version of the DNA protocol in conjunction with a DHCP [4] client. The WiFi interface needs to connect via a direct, or simulated link, to a DHCP server (e.g. ISC-DHCP [5]) either running on a separate physical machine or, as outlined in the testing environment section 4.1, in a separate LXC [6] container or Virtual machine.

2.2 Enabler Installation

The enabler may be installed via the project's Artifactory based repository, by installing the binary package. Given the machine is configured to utilise Artifactory it may be installed as follows:

```
$ sudo apt install dhcpcd-dip-2.XX
```

The enabler may also be built and installed from the source code package. To install it from source, first obtain the compressed source tar file, and then follow the steps below:

```
$ tar xf dhcpcd-dip.2.XX.tgz
$ cd dhcpcd-dip.2.XX
$ ./configure
$ sudo make install
```

2.3 Enabler Configuration

This enabler provides for enhanced location privacy for a device that roams from one Internet-based network to the next, primarily in the context of non-3GPP 5G access. It provides for enhanced privacy with respect to the leakage of identifiers for previous points of attachment. Specifically, we introduce four new mechanisms to protect the privacy of the device:

- **Randomised ordering (**`opt: dna_random`**)**
  - This mechanism provides for randomised delivery of candidate link layer addresses in the DNA reachability tests, so as to enhance the location privacy with respect to the location and time-based analysis of the device's movement patterns.
- **Dummy addresses (**`opt: dna_random`**)**
  - This mechanism allows for the introduction of dummy addresses into the DNA reachability tests to enhance location privacy, with respect to location identification and time-based analysis of the device's movement patterns.
  - With the R2 release, we introduce a new option that provides for automated the choice of the number of dummy addresses.
- **Pre-analysis phase of the address privacy metrics (**`opt: dna_preanalysis`**)**
  - This mechanism allows for the set of existing candidate link layer (MAC) addresses for DNA to be analysed before use to ascertain their privacy metrics with release R2. Specifically, we provide for the option to check and filter geolocatable candidate MAC address pairs.

The enabler is based upon the widely used dhcpcd [7] DHCP client and utilises the same configuration file (/etc/dhcpcd.conf). For the purposes of this enabler three new configuration options are provided with their arguments as applicable:

- **dna** **<no args>**
  - Enable Detection of Network Attachment (DNA) functionality
- **dna_random** **<integer>**
  - **1** Enable randomised ordering of DNA requests
- **dna_dummy** **<integer>**
  - **0-N** Number of dummy addresses
  - **-1** Automatic choice of number of dummy addresses
- **dna_preanalysis** **<integer>**
  - **1** Enable use of geolocation based pre-filtering of DNA requests
- **dna_geourl** **<string:URL>**
  - Specify URL to be used for geolocation pre-filtering of DNA requests

Once installed the client may be configured by editing the configuration file (/etc/dhcpcd.conf).

2.4 Troubleshooting

The enabler should normally run without user intervention as explained in the user guide section. However, if there are problems, then the enabler may be run in the foreground with console based debug logging enabled:

```
$ sudo /usr/sbin/dhcpcd -B –d [interface_name]
```

3 User and Programmer Guide

The enabler is not programmable.

3.1 User Guide

The detailed use of dhcpcd is well documented in its manual pages [7], but we will cover the aspects relevant to the operation of the enabler here. The service is configured as described in section 2 and typically it is invoked as a system service, though it also may be started directly on the command line, both procedures are outlined in the following sections.

3.1.1 Systemctl based management

With suitable system and service management tools such as `systemd`, or `system V` init scripts, the client may be launched via the system management controls.

To start the client, using `systemctl`, run the following command:

```
$ sudo systemctl start dhcpcd.service
```

To stop the client, using `systemctl`, run the following command:

```
$ sudo systemctl stop dhcpcd.service
```

3.1.2 Command line based management

The client may be launched directly via the command line, where upon it will start and attempt to acquire a DHCP lease, once acquired it will continue to run in the background.

To start the client, run the following command:

```
$ sudo /usr/sbin/dhcpcd [interface_name]
```

To stop the client, run the following command:

```
$ sudo /usr/sbin/dhcpcd –x [interface_name]
```

To stop the client, and release its DHCP leases, run the following command:

```
$ sudo /usr/sbin/dhcpcd –k [interface_name]
```

3.2 Programmer Guide

This section is not applicable as the enabler is not programmable.

4 Unit Testing

This section provides an overview of the virtualised testing environment, and number of tests that may be run to assess whether the tool is performing correctly.

4.1 Testing environment

For testing in a virtual environment, as shown in Figure 1, a number virtual WiFi devices may be instantiated using Linux's IEEE802.11 WiFi radio simulator (`mac80211_hwsim`) [8]. The client, with one simulated WiFi interface (i.e. wlan0), resides in one LXC container, whilst the server resides in a second LXC container. The server container includes multiple simulated WiFi interfaces (e.g. wlan1-4) which can be used to simulate movement by turning them on and off in a sequence using the `rfkill` [9] tool for enabling and disabling wireless devices. The mobility paths are shown to illustrate the different potential paths a client might take.

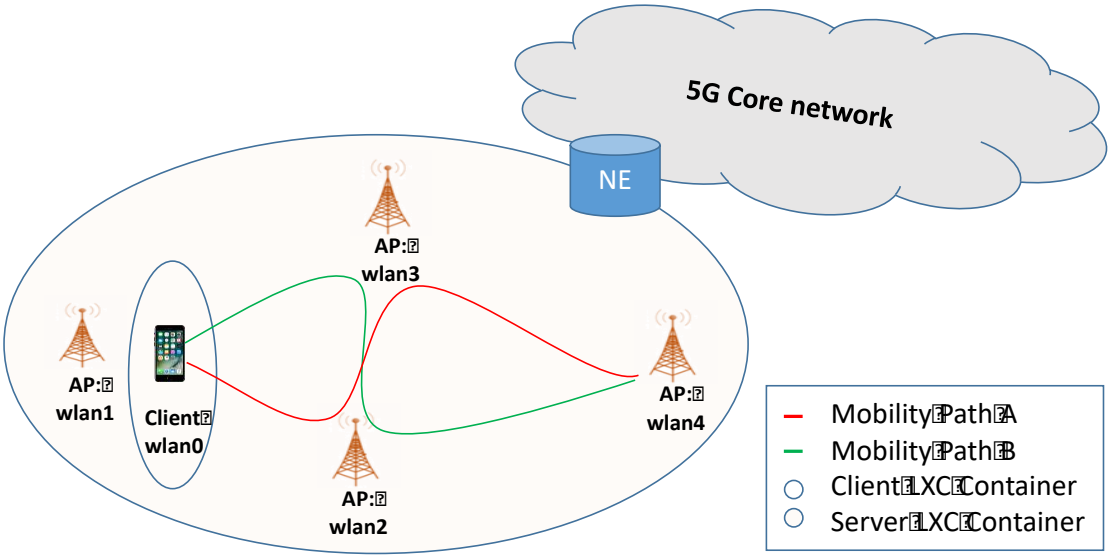

Figure 1 Test network configuration

4.2 Unit Tests

4.2.1 Unit Test 1

The dhcpcd source distribution includes a basic test suite. The test suite firstly builds the test executable, which allows one to verify that the test binary has been correctly built. Secondly the test binary allows for the confirmation of the correct operation of the security mechanisms:

```
$ tar xf dhcpcd-dip.X.XX.tgz
$ cd dhcpcd-dip.X.XX
$ ./configure
$ make test
```

The build and tests should complete with no errors reported.

4.2.2 Unit Test 2

The dhcpcd tool also has a run-time test mode which allows for testing of the tools functionality in obtaining a DHCP lease. The test mode will attempt to obtain an Internet address from a DHCP server, then perform Duplicate Address Detection (DAD), after which it exits and report the results of the interaction. The test mode is invoked by starting the tool using the –T option (and optionally the –d debug mode enabled):

```
$ ./dhcpcd -T –d [interface_name]
```

Here is an example run with WiFi interface `wlan0` attached to WiFi network ‘`ssidtest1`’ – with debug mode enabled:

```
$ ./dhcpcd -T -d wlan0
dhcpcd-6.11.1-DIP-v2.0 starting
Using Config file: /usr/local/etc/dhcpcd.conf
wlan0: IPv6 kernel autoconf disabled
DUID 00:01:00:01:1f:07:b4:4e:08:00:27:75:f4:19
wlan0: IAID 00:00:00:00
wlan0: delaying IPv6 router solicitation for 0.8 seconds
wlan0: DNA Option Present
wlan0: No DNA leases found
wlan0: delaying IPv4 for 0.9 seconds
wlan0: soliciting an IPv6 router
wlan0: sending Router Solicitation
wlan0: reading lease `/var/db/dhcpcd-wlan0-ssidwlan.lease'
wlan0: discarding expired lease
wlan0: soliciting a DHCP lease
wlan0: sending DISCOVER (xid 0x6450b71a), next in 4.9 seconds
wlan0: offered 10.10.1.19 from 10.10.1.1
wlan0: executing `/usr/local/libexec/dhcpcd-run-hooks' TEST
interface='wlan0'
pid='11970'
reason='TEST'
ifcarrier='up'
ifflags='4163'
ifmtu='1500'
ifssid='ssidwlan'
ifwireless='1'
new_broadcast_address='10.10.1.255'
new_dhcp_lease_time='6000'
new_dhcp_message_type='2'
new_dhcp_server_identifier='10.10.1.1'
new_ip_address='10.10.1.19'
new_network_number='10.10.1.0'
new_subnet_cidr='24'
new_subnet_mask='255.255.255.0'
dhcpcd exited
```

5 Acknowledgements

We would like to acknowledge Roy Marples, the main developer of the dhcpcd software.

6 Abbreviations

5G-PPP	5G Infrastructure Public Private Partnership
DAD	Duplicate Address Detection
DNA	Detection of Network Attachment
DHCP	Dynamic Host Configuration Protocol
GAN	Generic Access Network
LXC	Linux Containers

7 References

[1] 3GPP, “TS44.318 Generic Access Network (GAN); Mobile GAN interface layer 3 specification (Release 13),” 3GPP, 2016.

[2] J. Arkko and H. Haverinen, “RFC4187 Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA),” IETF, 2006.

[3] B. Aboba, J. Carlson and S. Cheshire, “Detecting Network Attachment in IPv4 (DNAv4). RFC 4436.,” IETF, 2006.

[4] R. Droms, “Dynamic Host Configuration Protocol. RFC 2131 (Draft Standard),” IETF, 1997.

[5] I. S. C. (ISC), “ISC DHCP Server,” [Online]. Available: https://www.isc.org/downloads/dhcp/.

[6] “Linux LXC Containers,” [Online]. Available: https://linuxcontainers.org/.

[7] R. Marples and e. a. , “dhcpcd - DHCP client,” [Online]. Available: http://roy.marples.name/projects/dhcpcd/.

[8] J. Malinen, “Linux 802.11 radio software simulator (hwsim),” [Online]. Available: http://www.linuxwireless.org/en/users/Drivers/mac80211_hwsim/.

[9] J. Berg and M. Holtmann, “rfkill tool - to query the state of the rfkill switches,” [Online]. Available: http://www.linuxwireless.org/en/users/Documentation/rfkill/.