



WICED Studio



WICED™ - Enterprise Security User Guide

Doc. No.: 002-22776 Rev. **

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

Contents

| | |
|---|----------|
| About This Document..... | 3 |
| Purpose and Audience | 3 |
| Scope | 3 |
| Acronyms and Abbreviations | 3 |
| IoT Resources and Technical Support | 3 |
| 1 Prerequisites..... | 4 |
| 2 Connecting WICED to Enterprise Security Network..... | 5 |
| 2.1 Certificate and Key Installation | 5 |
| 2.2 Creating a Build Target..... | 5 |
| 2.3 Details of Enterprise Security Commands | 6 |
| 2.4 Executing Enterprise Security Commands | 7 |
| 3 Supported Platforms and Enterprise Security Types..... | 8 |
| 3.1 Supported Platforms..... | 8 |
| 3.2 Supported Enterprise Security Types | 8 |
| Document Revision History | 9 |

About This Document

This document explains how to use the enterprise security features supported in Cypress Wireless Internet Connectivity for Embedded Devices (WICED™; pronounced “wick-ed”) Wi-Fi devices. This document also lists the various enterprise security features supported in WICED and the supported WICED platforms.

Purpose and Audience

This document is intended for software developers who use WICED SDK to create applications that connect to enterprise security network. The document assumes that you are familiar with Wi-Fi router configurations for RADIUS server and know how to generate self-signed certificates.

Scope

This document uses BCM943364WCD1 as the reference platform to explain the details. The document assumes that the enterprise security network already exist. Setting up of enterprise security network is not in the scope of this document.

Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use.

For a comprehensive list of acronyms and other terms used in Cypress documents, go to www.cypress.com/glossary.

IoT Resources and Technical Support

Cypress provides a wealth of data at www.cypress.com/internet-things-iot to help you to select the right IoT device for your design, and quickly and effectively integrate the device into your design. Cypress provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates. Customers can acquire technical documentation and software from the Cypress Support Community website (community.cypress.com).

1 Prerequisites

The following are required to connect a WICED device to an enterprise network:

- A computer with at least one USB port to connect the WICED Evaluation Board and run the WICED SDK
- Serial communication program, such as PuTTY, installed in the computer.
- An enterprise network infrastructure
- RADIUS (Authentication) Server - Ubuntu 16.04 LTS machine with:
- Free-Radius Version 2.2.9 as an EAP-Server - Free-Radius configured to support EAP-TLS, EAP-TTLS, PEAPv0-MSCHAPv2
- Self-signed certificates
- Wi-Fi Routers
- Routers available in the market, which supports enterprise security (802.1X) configurations
- Supplicant
- WICED Wi-Fi device (For demonstration purpose, BCM943364WCD1 is used as a supplicant device)

2 Connecting WICED to Enterprise Security Network

This section explains the step-by-step procedure to connect a WICED device to an enterprise security network.

2.1 Certificate and Key Installation

To install certificates and security keys on WICED, replace the `certificate.h` file located in `Wiced-SDK\libraries\utilities\command_console\wifi\` with the appropriate certificates in `.h` format. For instance, replace Root CA certificate, Client/User certificate, and Keys, as appropriate.

2.2 Creating a Build Target

Console application supports the commands that enable a device to connect to and leave an enterprise security network. The build target for the WICED test console application is constructed from several build components. [Table 2-1](#) lists the components used for the test console application.

| Component | Available Options |
|------------------|-------------------|
| Application Name | test.console |
| RTOS | ThreadX |
| Network Stack | NetX, NetX_Duo |
| Platform | BCM943364WCD1 |
| Interface | SDIO, SPI |
| Build type | release |

Table 2-1. Components for an Example Test Console Application Build Target

A sample test console build target:

```
test.console-BCM943364WCD1 download_apps download run
```

See the *WICED Quick-start Guide* for a complete description on how to build an application and download the firmware image to WICED device.

Note: The purpose of the console application is to demonstrate WICED features by issuing certain commands in the console. If there is a memory constraint in the reference platform, tune `console.mk` as per the comments given in `console.mk` to ensure enough memory is available to run enterprise security commands.

For BCM943364WCD1 platform, the sample tunable parameters shown in [Figure 2-1](#) will help in reducing the overall code memory usage to free enough memory required to run enterprise security commands

```

$ git diff apps/test/console/console.mk
diff --git a/apps/test/console/console.mk b/apps/test/console/console.mk
index 58efb87..657a7c1 100644
--- a/apps/test/console/console.mk
+++ b/apps/test/console/console.mk
@@ -123,7 +123,7 @@ $(NAME)_INCLUDES := .
 #by uncommenting below line.
 ifeq ($(PLATFORM),$(filter $(PLATFORM), BCM943364WCD1 BCM94343WWCD1 BCM94343
 # Disable components not needed due to application size limitation on this pl
-CONSOLE_NO_P2P ?=1
+CONSOLE_NO_P2P :=1
 CONSOLE_DISABLE_TRACEX_COMMANDS := 1

$(NAME)_DEFINES += CONSOLE_DISABLE_TRACEX_COMMANDS
@@ -133,8 +133,8 @@ $(NAME)_DEFINES += CONSOLE_DISABLE_MAILINFO_COMMANDS
#GLOBAL_DEFINES += WICED_DISABLE_TLS

ifeq ($(WICED_DISABLE_COMMON_PKT_POOL),1)
-GLOBAL_DEFINES += TX_PACKET_POOL_SIZE=14 \
-                RX_PACKET_POOL_SIZE=12 \
+GLOBAL_DEFINES += TX_PACKET_POOL_SIZE=10 \
+                RX_PACKET_POOL_SIZE=10 \
                WICED_TCP_TX_DEPTH_QUEUE=10 \
                WICED_TCP_WINDOW_SIZE=131072
else #WICED_DISABLE_COMMON_PKT_POOL
@@ -146,14 +146,14 @@ else #WICED_DISABLE_COMMON_PKT_POOL
ifeq ($(CONSOLE_NO_P2P),1)
GLOBAL_DEFINES += WICED_USE_COMMON_PKT_POOL \

```

Figure 2-1. Console.mk make file

2.3 Details of Enterprise Security Commands

Following are the enterprise security commands that can be used to connect WICED to an enterprise security network:

- `join_ent` – Associates with an Enterprise Network

Syntax:

```

join_ent <ssid> <EAP-Protocol> <User Name> <Password> <Tunnel Auth Type> <Phase2
Protocol Type> <Client-Cert> <Wi-Fi security type>

```

Where,

`ssid` – SSID string of the enterprise network

`EAP-Protocol` – Security protocol used. Valid values: `peap`, `eap_tis`, `eap_ttls`

`User Name` – User name string of the supplicant

`Password` – Password string of the supplicant

`Tunnel Auth Type` – Tunnel authentication type. This argument is valid only when `<EAP-Protocol>` is `eap_ttls`. Valid value: `eap`

`Phase2 Protocol Type` – Phase 2 protocol type used in case of tunneled authentication. This argument is valid only when `<EAP-Protocol>` is `eap_ttls`. Valid value: `mschap2`

`Client-Cert` – Indicates if client certificate needs to be used during Phase1 authentication of EAP-TTLS. This argument is valid only when `<EAP-Protocol>` is `eap_ttls`. Valid value: `client-cert`

`Wi-Fi security type` – Represents Wi-Fi security types (for example, `wpa2`, `wpa2_tkip`, `wpa`, `wpa_tkip`, `wpa2_ftb`)

Usage:

To connect enterprise network with PEAP (PEAPv0 – MSCHAPv2) security:

```

join_ent wiced_ssid peap username password wpa2

```

To connect enterprise network with EAP TLS security:

```

join_ent wiced_ssid eap_tls username password wpa2

```

To connect Enterprise Network with EAP TTLS security:

With client certificate:

```
join_ent wiced_ssid eap_ttls username password eap mschapv2 client-cert wpa2
```

Without client certificate:

```
join_ent wiced_ssid eap_ttls username password eap mschapv2 wpa2
```

- **leave_ent** – Disassociates from an enterprise network

Syntax:

```
leave_ent
```

2.4 Executing Enterprise Security Commands

After successfully downloading the console application using one of the build targets explained in [Creating a Build Target](#), open PuTTY (or any other serial console application) in the development machine where DUT is connected and issue the enterprise security console commands.

Consider that DUT is being connected to an enterprise network with EAP TLS as enterprise security type:

```
join_ent wiced_ssid eap_tls username password wpa2
```

The following log messages are displayed on the serial console:

Console app

```
> join_ent WICED_SSID eap_tls username password wpa2
Joining : WICED_SSID
Successfully joined : WICED_SSID
Obtaining IPv4 address via DHCP
DHCP CLIENT hostname WICED IP
IPv4 network ready IP: 192.168.1.15
Setting IPv6 link-local address
IPv6 network ready IP: FE80:0000:0000:0000:02A0:50FF:FE46:8506
Successfully retrieved the session information 0
```

3 Supported Platforms and Enterprise Security Types

3.1 Supported Platforms

The following are the WICED platforms that support enterprise security features:

- BCM943907WCD1
- BCM943340WCD1
- BCM943364WCD1
- BCM943438WCD1
- BCM94343WWCD1
- BCM94343WWCD2
- BCM943362WCD1

3.2 Supported Enterprise Security Types

The following enterprise security types are supported in WICED:

- PEAPv0 – MSCHAPv2 as inner authentication method
- EAP-TLS
- EAP-TTLS – EAP-MSCHAPv2 as inner authentication method

Document Revision History

Document Title: WICED™ - Enterprise Security User Guide

Document Number: 002-22776

| Revision | ECN | Issue Date | Description of Change |
|----------|---------|------------|-----------------------|
| ** | 6049558 | 01/29/2018 | Initial release |

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at [Cypress Locations](#).

Products

| | |
|-------------------------------|--|
| Arm® Cortex® Microcontrollers | cypress.com/arm |
| Automotive | cypress.com/automotive |
| Clocks & Buffers | cypress.com/clocks |
| Interface | cypress.com/interface |
| Internet of Things | cypress.com/iot |
| Memory | cypress.com/memory |
| Microcontrollers | cypress.com/mcu |
| PSoC | cypress.com/psoc |
| Power Management ICs | cypress.com/pmic |
| Touch Sensing | cypress.com/touch |
| USB Controllers | cypress.com/usb |
| Wireless Connectivity | cypress.com/wireless |

PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6 MCU](#)

Cypress Developer Community

[Forums](#) | [WICED IOT Forums](#) | [Projects](#) | [Videos](#) | [Blogs](#)
| [Training](#) | [Components](#)

Technical Support

cypress.com/support



Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.