# ARQIT NetworkSecure™
# Standalone (Test Mode) Adaptor v3.3.0
# Installation Guide

For integration testing with devices and applications using the ETSI 014 interface

| | | |
|---|---|---|
| Version | : | 1.0 |
| Classification | : | PUBLIC |
| Export Rating | : | NOT EXPORT CONTROLLED |
| Status | : | Issued |
| Date | : | 07/02/2024 |

# Version Control

| Version | Status | Date changed | Change summary | Authored by |
|---|---|---|---|---|
| 1.0 | Issued | 07/02/2024 | Install guide for v3.3.0 of Standalone Adaptor | Ravi Patel |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Contents

# Introduction

## Overview

The Arqit NetworkSecure Adaptor is a software application that interfaces with network devices e.g. firewalls or applications (collectively referred to ‘Devices’ in the rest of the document – that support the ETSI 014 specification for external key retrieval.

Connecting Devices to NetworkSecure Adaptors enables out-of-band symmetric keys to be requested on-demand that can be used to secure communications between Devices e.g. a point-to-point IPSec VPN link between firewalls.

Adding additional key material at both ends of the tunnel outside of the IPSec key agreement protocols (such as IKE v2 which are believed to be susceptible to retrospective attack by quantum computers) adds protection against attack by quantum computers in the future; breaking the key agreement protocol cannot reveal the additional key material ensuring data passing through the tunnel remains secure.

## Purpose and scope

This document introduces the high-level architecture of Arqit’s NetworkSecure Adaptor configured in ‘Standalone’ or ‘Test Mode’ (shortened to **‘Standalone Adaptor’** in the document), Version 3.3.0, and describes the steps required to configure and deploy the Standalone Adaptor to provide fixed symmetric keys to Devices using the ETSI 014 interface for purposes of integration testing only.

It does not include a detailed description of the steps required to configure the Device to make key requests to the Standalone Adaptor using the ETSI REST API. It is recommended to refer to the specific Device vendor documentation for configuration instructions.

## Variable Notation

In script and command line examples, variables that should be replaced are indicated by the <VariableName> notation; when replacing the variables, the <> should *not* be included.

For example, in the command line

```
sudo ./setup.sh deploy -x -i <sae_id> -u <user_to_run_service_as> -t
-l '<passphrase>'
```

the following should be entered

```
sudo ./setup.sh deploy -x -i MySAEID -u standalone-adaptor-user -t
-l 'testmode'
```

Where MySAEID and standalone-adaptor-user and ‘testmode’ are the values to be used.

## Further reading

The ETSI 014 specification is a comprehensive description of the interface used between the network firewall gateways (SAEs) and the Standalone Adaptor.

The Standalone Adaptors replace the KME devices shown in this specification, allowing key agreement over a classical network.

***https://www.etsi.org/deliver/etsi_gs/QKD/001_099/014/01.01.01_60/gs_qkd014v010101p.pdf***

# Prerequisites

## Introduction

Before starting to install, please ensure the following pre-requisites are available.

## Checklist

| Item | Description |
|---|---|
| **Secure Application Entity (SAE) licenses** | A minimum of two Devices (SAEs) e.g. firewall gateways are required to make use of the Standalone Adaptor functionality and each Device must be registered with any required vendor licenses. |
| **SAE firmware updates** | Devices (SAEs) must be running firmware that generates and responds to ETSI 014 REST API calls made to Arqit's Standalone Adaptor. |
| **Standalone Adaptor virtual machines** | A dedicated VM (AWS EC2 instance or Azure Virtual Machine) is required to host each Standalone Adaptor. |

| | |
|---|---|
| **Certificates** | The ETSI 014 specification requires mutual TLS authentication between a network device (SAE) and its key provider (KME).<br><br>Two certificates are required to enable mutual authentication between a Standalone Adaptor and its corresponding SAE e.g. firewall gateway.<br><br>• The setup procedure provides the ability to generate self-signed certificates that can be used for integration testing. |
| **Adaptor Virtual Machine image** | Once you have purchased the Standalone Adaptor product, access is provided to the Standalone Adaptor Amazon Machine Image (AMI) that can be used to create the cloud VMs required for performing ETSI 014 integration testing with Devices. |

# Architectural overview

The Standalone Adaptor is deployed for purposes of integration testing with Devices that support the ETSI 014 interface. This solution provides a quick and easy setup enabling OEM network appliance vendors or application developers to request fixed 'dummy' symmetric keys (256 bits) to test the ETSI 014 REST interface and related API calls.

The Standalone Adaptor provides test keys and is **not suitable for production environments** that require secure i.e. quantum safe keys for enhanced security. It additionally does not provide the policy and management benefits that Arqit's QuantumCloud™ and NetworkSecure Adaptor delivers.

**IMPORTANT:** Running Test mode in a production environment is not supported.

The following two network deployment scenarios are supported.

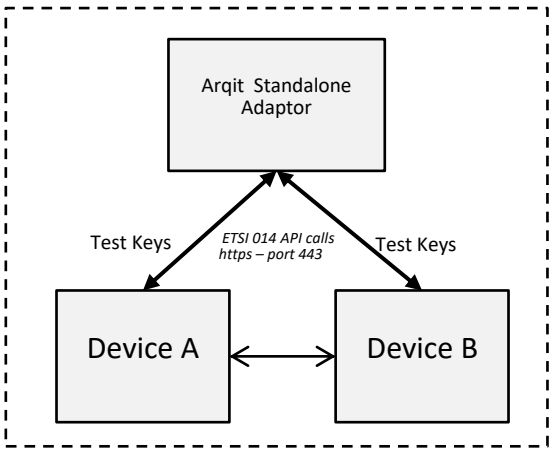


Figure 1 – Single Standalone Adaptor providing test keys to two devices locally

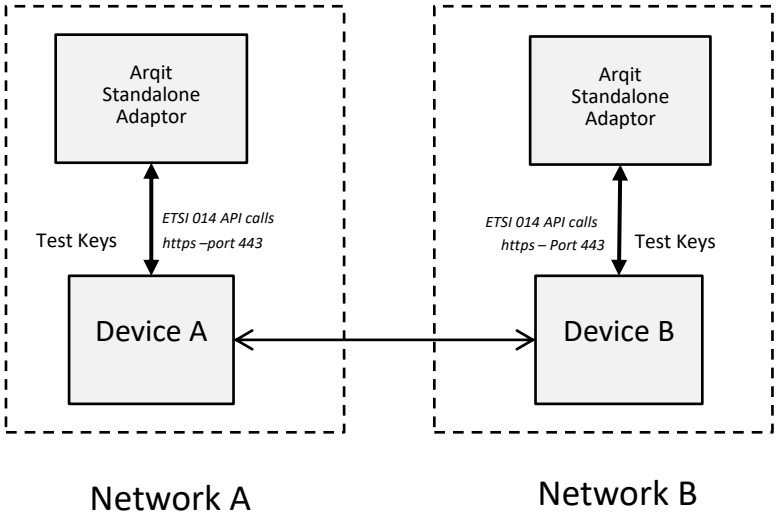


Figure 2 – Two Standalone Adaptors providing test keys to connected devices

In Figure 2, Standalone Adaptors are providing the same test key to its corresponding locally connected Device. Each Device/Standalone Adaptor pair can be located in different networks to simulate a 'real' network communication link between Devices.

**NOTE**:

- The test keys generated by the Standalone Adaptor(s) are based on a passphrase entered by the user at the time of deployment of the Adaptor. The same symmetric key is generated for each ETSI 'Get Key' and ETSI 'Get Key with ID' request based on the unique passphrase entered in the Standalone Adaptor.
- In both deployment scenarios, the appropriate network configuration needs to be put in place in order for both Device A and B to communicate with each other over the network to establish data communication sessions using the Standalone Adaptor test keys.

**IMPORTANT** – In Scenario 2, **the same passphrase** needs to be entered in each of the Standalone Adaptors to generate identical test keys.

# Locating Standalone Adaptors within the network

The Standalone Adaptors do not require any internet connectivity to function, nor do the Adaptors need to communicate with each other in the deployment scenario outlined in Figure 2. They therefore can be deployed in a private network (assigned private IPv4 addresses), and in the same subnet as the locally connected Device e.g. firewall.

The Standalone Adaptor and connected Device communicate with each other over the ETSI 014 REST interface using https over port 443 (http over TLS).

# Adaptor as a service

The Standalone Adaptor runs as a Linux *systemd* service. This enables the Adaptor to automatically start and run as a background service in the context of a specified user. The user must exist on the system (see how to create a User). The setup.sh script will create a new folder structure in the Linux file system including the `networksecure-adaptor.service` *systemd* service file in the following folder:

```
/etc/systemd/system/networksecure-adaptor.service
```

## Adaptor folder structure

```
/opt
    /arqit
        /networksecure-adaptor
            certs/
                ...
            logs/
                adaptor.log
            versions/
                3.3.0/
            certs.sh
            deviceMetadata.json
            failedAudits.json
            monitorProperties.json
            policy.json
```

If the Adaptor service fails, *systemd* will attempt to restart the Adaptor three (3) times (once every 10 seconds) before generating an error message. Additionally, OS re-boots will automatically re-start the Adaptor service.

## Useful Systemd commands

- `sudo systemctl status networksecure-adaptor.service`
  - get status of service
- *`sudo journalctl -u networksecure-adaptor.service`*
  - get systemd service logs. This command enables informational messages generated by the Standalone Adaptor to be displayed in the

console, including any errors that may be encountered during the normal operations of the Adaptor once it has been started by the systemd service.

- `sudo systemctl stop networksecure-adaptor.service`
  - stop the Standalone Adaptor service if running
- `sudo systemctl start networksecure-adaptor.service`
  - start the Standalone Adaptor if not already running
- `sudo systemctl restart networksecure-adaptor.service`
  - re-start the Standalone Adaptor

### Systemd Security Hardening features

*Systemd* natively provides system hardening capabilities to enhance the security of deployed Adaptors. Arqit recommended security settings are enabled by default and outlined in Appendix A.

**NOTE**: systemd security hardening measures can be enabled/disabled depending on your specific system security policies and requirements. More information can be found at https://www.freedesktop.org/software/systemd/man/systemd.exec.html.

**Do not modify** the 'ExecStart' line in the `networksecure-adaptor.service` file.

# Standalone Adaptor Setup and Configuration

This section runs through the steps to setup and configure a Standalone Adaptor.

The following sections describe the workflow.

- Certificate creation and installation
- Standalone Adaptor configuration
- Device configuration
- Standalone Adaptor Setup
- Run the Standalone Adaptor
- Stop the Standalone Adaptor

It is recommended that you examine the configuration parameters (Appendix A) before beginning the setup to ensure you understand what the parameters mean and where they are located.

## Pre-reading

### Certificate creation and installation

The Standalone Adaptor uses self-signed certificates that are created using the provided setup.sh script.

The script will create a dummy CA with CA root certificate, plus test certificates for the Standalone Adaptor and the associated Device e.g. a firewall gateway.

### The SAE_ID – format for unique identifier of SAE (Device)

The ETSI 014 standard supported by the Standalone Adaptor limits the passing of a single standardised parameter (the SAE_ID of the Device at the “remote end” of the communication link e.g. a VPN tunnel) when requesting a key.

The SAE_ID for each Device must conform to the following format:

**<Hostname>::<Adaptor FQDN>**

where ***<Hostname>*** is the identifier for the locally connected Device:

E.g. ‘firewall-xxx’; ‘router-xxx’; ‘app-xxx’

where ***<Adaptor FQDN>*** is the Fully Qualified Domain Name of the Standalone Adaptor. (*A private IPv4 address of the Standalone Adaptor can be used here for convenience as no internet connectivity is required by the Adaptor in Test mode)*

**Note**: The Device Hostname and the Adaptor FQDN must not contain the character ":" (colon).

The SAE_ID should be configured to this compound value in the Device's management console and configured when registering the Adaptor during Adaptor Setup.

# Setting up the Standalone Adaptor

## Certificate creation

The dummy certificates required for each Standalone Adaptor and connected Device are created as part of the Adaptor Setup process by passing *'-x'* as a parameter when running the `setup.sh deploy` command.

## Configure Standalone Adaptor Configuration Files

The Standalone Adaptor is designed to be deployed and setup with minimal configuration.

The following instructions outline a minimal configuration.

Ensure that this configuration is completed for **each Adaptor**.

**Steps**

1. Open the `setup.sh` script
2. Update the SAE ID parameter according to Appendix A or alternatively this can be passed via the command line as outlined in Standalone Adaptor Setup
3. Save the file
4. (Optional) If required, the default passwords for the certificates (required for mutual authentication between Adaptor and Device as per ETSI 014 standard) can be changed i.e. CA_PASSWORD, CA_P12_PASSWORD, ADAPTOR_P12_PASSWORD and CLIENT_P12_PASSWORD can be changed in `/opt/adaptor-runtime-secrets.conf` according to the descriptions in Appendix A
5. Save the file

## Create a runtime User for the Standalone Adaptor service (OPTIONAL)

The Standalone Adaptor has a pre-configured default Linux user (*netsecure-test*) with sudo privileges. For enhanced security, it is recommended (but not essential for testing scenarios) to create a 'non-privileged' user for running the NetworkSecure Adaptor as a daemon service on the Linux system. This limits the attack surface and minimises the risk of an attacker who has compromised this specific user account from gaining access to root/sudo privileges. The following command should be run to create this user :

**Steps:**

1. Run the following command:

```
sudo adduser <username>
```

## Device Configuration

See vendor documentation for Device specific configuration to enable key requests to be made over the network to the Standalone Adaptor using the ETSI 014 API.

## Standalone Adaptor Setup

Once configuration is completed, the Standalone Adaptor can be deployed using the `setup.sh` shell script provided.

Note: Only users with root or sudo privileges (including the default 'netsecure-test' user) on the Linux system can modify the `setup.sh` script or execute the script to deploy the Adaptor.

**Steps**

1. Run the following command below with the arguments:
    - '-x' - generates dummy/PoC certs
    - '-i' - the SAE ID value (SAE ID must conform to this format)
    - '-u' - the user that the Adaptor runs as – see here. If not provided, this defaults to the 'netsecure-test' user.
    - '-t' - configures the Adaptor to run in Standalone – Test mode
    - '-l' - the passphrase required for test key generation

**Steps…. continued**

- Alternatively, the '-i' argument can be configured directly in the 'Variables' section within the setup.sh script - Appendix A
- The default directory where the config.json and adaptor-run-time secrets.conf files are stored is '/opt/'

**NOTE**:

- Dummy/PoC certs generated by the Adaptor (in the 'opt/arqit/networksecure-adaptor/certs' folder) need to be uploaded to its connected Device(s) - both client.p12 and ca.crt certificates. The default CA_PASSWORD, CA_P12_PASSWORD, ADAPTOR_P12_PASSWORD and CLIENT_P12_PASSWORD can be changed if required in the adaptor-runtime-secrets.conf file, before executing the deploy command below. IMPORTANT: Devices connecting to the Standalone Adaptor may perform validation of the Adaptor's server name by checking the SAN and/or CN value in the Adaptor's certificate. In these cases, the CN/SAN value of the Standalone Adaptor test certificate (*www.example.com*) needs to be configured in the Device.
- For Scenario 1, the SAE ID configured on the Standalone Adaptor is that of Device A or Device B
- For Scenario 2, the same passphrase needs to be entered in each of the Standalone Adaptors
- For a given passphrase, the same symmetric key (256 bits) is generated by an Adaptor for all subsequent ETSI 014 key requests. Passphrase value - special characters should be escaped using the backslash '\' character e.g. 'pas\$345\\'
- To change the symmetric key generated by an Adaptor, the command below needs to be executed again with a different passphrase value

```
sudo ./setup.sh deploy -x -i <sae_id> -u <user_to_run_service_as>
-t -l '<passphrase>'
```

# Running the Standalone Adaptor

Once the Standalone Adaptor has been deployed, the *systemd* service will automatically start it.

**Note**: Auto-start of the Standalone Adaptor can be prevented by providing the '-m' argument in the command during the Adaptor Setup stage:

```
sudo setup.sh deploy -m
```

# Stop the Standalone Adaptor

**Steps**

1. Run the following `systemd` command:

```
sudo systemctl stop networksecure-adaptor.service
```

# Uninstall the Standalone Adaptor

**Steps**

1. To uninstall the Standalone Adaptor using the shell script, run the command below.

```
sudo ./setup.sh uninstall
```

# Upgrade Standalone Adaptor to access QuantumCloud features

Contact enquiries@arqit.uk and visit Arqit's website to learn more about the benefits of Arqit's Symmetric Key Agreement Platform QuantumCloud and the NetworkSecure Adaptor; which provides quantum safe symmetric keys over the ETSI 014 interface to protect point-to-point VPN links between network firewall gateways. An annual subscription license is required to upgrade the Standalone

Adaptor to the full-featured NetworkSecure Adaptor and benefit from QuantumCloud features.

# Appendix A – Configuration parameters

## setup.sh

This section replicates the contents of the `setup.sh` file which documents the parameters required for deployment and running of the Standalone Adaptor.

```
#!/usr/bin/env bash

ADAPTOR_VERSION="3.3.0-x"

set -eou pipefail


###################################
#            VARIABLES            #
###################################

# required. Valid format <firewall-id>::<adaptor-address>
SAE_ID=''

# (optional) only needed if creating certs for demo/PoC or in Test Mode
CA_CN='arqit.network.adaptor'
ADAPTOR_CN='www.example.com'
CLIENT_CN='www.example.com'

###################################
#              INPUTS             #
###################################
ACTION="${1:-help}"  # possible inputs: deploy, upgrade (not applicable
for Test Mode), uninstall, help


……..
```

The `setup.sh help` command will provide a list of the arguments and default values:

-i <SAE ID in the format '<firewall-id>::<adaptor-address>'> *(note: hostname can be used instead of firewall-id')*

-u <User to run systemd service as. Defaults to logged in user>

[-m : Disable auto-start of the service]"

[-x : Create certificates]"

# config.json

This section replicates the contents of the `config.json.txt` file which documents the parameters in `config.json` (default file location after deployment is *'/opt'* ).

```
// This file documents the config.json file.
// This is not possible inline as JSON does not support comments
{
  // Adaptor hosting port for serving keys via ETSI protocol
  "HOST_PORT_API": 443,

// The domain for the QuantumCloud instance to be used - not required for
// Test Mode
  "QC_DOMAIN": "uk.quantum.cloud",

// The region of the QuantumCloud instance - not required for Test Mode
  "REGION": "uk",

"CERTIFICATES":{
    // Absolute path to the Network Adaptor signing certificate (a .p12
// file). The default setting is "/opt/arqit/networksecure-
// adaptor/certs/adaptor.p12"
  "ADAPTOR_FILEPATH": /opt/arqit/networksecure-adaptor/certs/adaptor.p12",

// Absolute path to the CA verification certificate installed
    // into a .p12 file. The default setting is "/opt/arqit/networksecure-
// adaptor/certs/ca.p12"
  "CA_FILEPATH": "/opt/arqit/networksecure-adaptor/certs/ca.p12"
  },
  // Device properties for billing purposes - not required for Test Mode
    "DEVICE_BILLING_INFO": {
        "FIREWALL_VENDOR": "Some Vendor",
        "FIREWALL_MODEL": "ABC-1234",
        "DEVICE_TIER": 1
  },
  // Recommended timeouts - not required for Test mode
    "TIMEOUTS": {
  // The timeout when contacting QuantumCloud for authentication. Must be
between 1000 and 60000.
    "AUTHENTICATION_API_TIMEOUT_MS": 1000,
  // The timeout when contacting QuantumCloud for bilocation key peering.
Must be between 1000 and 60000.
    "DSCC_API_TIMEOUT_MS": 1000
  }
}
```

# networksecure-adaptor.service (systemd config file)

This section replicates the contents of the `networksecure-adaptor.service` file which is a systemd configuration file created by the setup.sh script when deploying the adaptor. This file is created in the '*/etc/systemd/system/*' directory.

```
[Unit]
Description=Run Arqit NetworkSecure Adaptor
After=network-online.target
StartLimitIntervalSec=60

[Service]
# DO NOT MODIFY

ExecStart=/usr/bin/bash /opt/arqit/networksecure-adaptor/versions/2.0.0/adaptor.sh start -
d /opt/arqit/networksecure-adaptor -v 2.0.0 -c /opt/config.json
EnvironmentFile=/opt/adaptor-runtime-secrets.conf

# RESILIENCY

Type=simple
Restart=on-failure
RestartSec=10
StartLimitBurst=3

# SECURITY

# Run service as specific user
User=networksecure-adaptor-user

# Prevent service from obtaining new privileges
NoNewPrivileges=yes

# Turn off physical device access
PrivateDevices=yes
DevicePolicy=closed

# Set specific system folders as read-only
ProtectSystem=yes

# Set home directory as read-only
ProtectHome=read-only

# Set Linux Control Groups as read-only
ProtectControlGroups=yes

# Deny explicit module loading
ProtectKernelModules=yes

# Set kernel variables as read-only
ProtectKernelTunables=yes

# Restrict access to Linux namespace functionality
RestrictNamespaces=yes

# Restrict access to realtime task scheduling policies
RestrictRealtime=yes

# Restrict user privilege escalation
RestrictSUIDSGID=yes

# Deny personality system call
LockPersonality=yes

# Allow adaptor service to bind to privileged ports
AmbientCapabilities=CAP_NET_BIND_SERVICE


[Install]
WantedBy=multi-user.target
```

# adaptor-runtime-secrets.conf

This section replicates the contents of the `adaptor-runtime-secrets.conf` file (default location is *'/opt'* ) which is where the secrets required to start the Standalone Adaptor are located.

Password values - special characters should be escaped using the backslash '\' character e.g. 'pas\$345\\'

```
# Passphrase to unlock the dummy CA root private key file (optional –
only needed if creating certs for Demo/PoC or Test Mode. Password must
be enclosed in '')
CA_PASSWORD='password'

# Passphrase to unlock the PKCS12 key store containing the CA
certificates public key (required - for production deployments,
replace the default value 'password' (used for Demo/PoC certs or Test
Mode) with the password used in production. Password must be enclosed in
'')
CA_P12_PASSWORD='password'

# Passphrase to unlock the PKCS12 key store containing the adaptor
certificates public and private keys (required -for production
deployments, replace default value 'password' (used for Demo/PoC or Test
Mode) with the password used in production. Password must be enclosed in
'')
ADAPTOR_P12_PASSWORD='password'

# Passphrase to unlock the PKCS12 key store containing the firewalls
certificates public and private keys (optional – only needed if
creating certs for Demo/PoC or Test Mode. Password must be enclosed in
'')
CLIENT_P12_PASSWORD='password'

# Username to connect to MQTT broker (optional. Password must be
enclosed in '')
MQTT_BROKER_USERNAME=''

# Password to connect to MQTT broker (optional. Password must be
enclosed in '')
MQTT_BROKER_PASSWORD=''
```

# Appendix C: Common errors

## Standalone Adaptor Setup

- Missing or Invalid parameter values that cause the setup to fail.

  Command line or setup.sh

  - Incorrectly formatted SAE ID
  - Missing ‘-l’ parameter and associated passphrase value
  - Missing ‘-x’ parameter to create dummy certificates
  - Missing ‘-t’ parameter to start Adaptor in test mode

## Standalone Adaptor execution

- Missing or invalid parameter values that prevent starting of the Standalone Adaptor.

  adaptor-runtime-secrets.conf

  - CA_P12_PASSWORD
  - ADAPTOR_P12_PASSWORD
  - CA_PASSWORD
  - CLIENT_P12_PASSWORD