Nubo Client Version 3.2
Security Target

Version 1.18

15 December 2023

Prepared by:

Nubo Software LTD.

8 Ariel Sharon St, 3rd Floor,
Or Yehuda, Israel

Revision History

Version	Date	Author	Detail
1.0	16 June 2021	Nubo	Initial submission.
1.1	01 April 2022	Nubo	Addressed lab comments.
1.2	08 April 2022	Nubo	Addressed lab comments.
1.3	15 February 2023	Nubo	Addressed lab comments.
1.4	26 February 2023	Nubo	Addressed lab comments.
1.5	03 April 2023	Nubo	Addressed lab comments.
1.6	19 April 2023	Nubo	Addressed lab comments.
1.7	20 April 2023	Nubo	Addressed lab comments.
1.8	24 May 2023	Nubo	Addressed lab comments.
1.9	31 May 2023	Nubo	Addressed lab comments.
1.10	27 July 2023	Acumen	Addressed TDs issued since previous check-in.
1.11	08 August 2023	Acumen	Addressed ORs.
1.12	29 August 2023	Acumen	Review
1.13	11 September 2023	Acumen	Review
1.14	28 September 2023	Acumen	Final Review
1.15	02 October 2023	Acumen	Added TD0779 for TLS and archived TD0588
1.16	20 October 2023	Acumen	1 Addressed QA mods.
1.17	08 December 2023	Acumen	Update based on ECR comments
1.18	15 December 2023	Acumen	Update based on ECR comments.

TABLE OF CONTENTS

LIST OF TABLES

ACRONYM LIST

AA ..... Assurance Activities
AES ..... Advanced Encryption Standard
API ..... Application Programming Interface
APK ..... Android Application Package
APP ..... Application
ASLR ..... Address Space Layout Randomization
CA ..... Certificate Authority
CAVP ..... Cryptographic Algorithm Validation Program
CC ..... Common Criteria
CM ..... Configuration Management
CMC ..... Certificate Management over CMS
CMS ..... Cryptographic Message Syntax
CN ..... Common Names
CTR ..... Counter Mode
CRL ..... Certificate Revocation List
DHE ..... Diffie-Hellman Ephemeral
DNS ..... Domain Name System
DRBG ..... Deterministic Random Bit Generator
ECDHE ..... Elliptic Curve Diffie-Hellman Ephemeral
EST ..... Enrollment over Secure Transport
FIPS ..... Federal Information Processing Standards
GCM ..... Galois Counter Mode
HMAC ..... Hash-based Message Authentication Code
HTTP ..... HyperText Transfer Protocol
HTTPS ..... HyperText Transfer Protocol Secure
IP ..... Internet Protocol
IT ..... Information Technology
MIME ..... Multi-purpose Internet Mail Extensions
NIAP ..... National Information Assurance Partnership
NIST ..... National Institute of Standards and Technology
OCSP ..... Online Certificate Status Protocol
OE ..... Operational Environment
OID ..... Object IDentifier
OS ..... Operating System
PII ..... Personally Identifiable Information
PP ..... Protection Profile
RBG ..... Random Bit Generator
RFC ..... Request For Comments
RNG ..... Random Number Generator
S/MIME ..... Secure/Multipurpose Internet Mail Extensions
SAN ..... Subject Alternative Name

SAR ..... Security Assurance Requirement
SFR ..... Security Functional Requirement
SHA ..... Secure Hash Algorithm
SoC ..... System on Chip
SP ..... Special Publication
ST ..... Security Target
TD ..... Technical Decision
TLS ..... Transport Layer Security
TOE ..... Target of Evaluation
TSF ..... TOE Security Functionality
TSS ..... TOE Summary Specification
VM ..... Virtual Machine
VMI ..... Virtual Mobile Infrastructure

1 Security Target Introduction

This document is a Common Criteria Security Target (ST) for Nubo Client Version 3.2 hereafter referred to as Nubo or the TOE. The ST is authored in accordance with Common Criteria Version 3.1 Revision 5.

The Security Target contains the following sections:

- Security Target Introduction (This section)
- Conformance Claims (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

1.1 Security Target and TOE Identification

1.1.1 Security Target Reference

ST Title - Nubo Client Version 3.2 Security Target

ST Version – Version 1.18

ST Date – 15 December 2023

1.1.2 TOE Reference

TOE Identification - Nubo Client Version 3.2

TOE Developer – Nubo Software LTD.

1.1.3 Keywords

Software, thin client, Virtual Mobile Infrastructure (VMI).

1.2 TOE Overview

The Target of Evaluation (TOE) is the Nubo Client Version 3.2. It is a thin client application installed and executed on an Android mobile device. The TOE establishes communications to a Virtual Mobile Infrastructure (VMI) platform (using a remote display protocol) and remotely displays the virtual apps that are running within the VMI platform. No output is displayed from other applications. The TOE only connects the mobile device to the virtual servers and is not responsible for the execution of the virtual apps.

With VMI, virtual applications execute on a user's behalf on VMI servers. No executable code associated with the virtual applications is downloaded to the user's device. Instead, the TOE displays the output from the virtual applications, and forwards input from the user to the virtual applications.

The TOE controls all communication between itself and the VMI environment. The TOE is only to be used with the Nubo Management Server and the Nubo Gateway. This ensures that all

communication occurs over a secure connection within a secure remote application infrastructure. All network connections are initiated by the TOE. Connection requests by a VMI server are not accepted.

Direct connection is established between the TOE and the Nubo Management Server. The Nubo Management Server processes user activation and login and communicates with the TOE and the Nubo Gateway. The Nubo Gateway implements the connection for executing the virtual applications. The traffic for the virtual applications (that are transmitted from the VMI platform to the Nubo Gateway) is sent over a single trusted channel between the Management Server and the TOE.

The user installs the TOE from the Google Play Store. The app store contains a generic version of the Android app which does not contain any user credentials or details. Initially, TOE user credentials are sent to the Management Server, the Management Server registers the TOE user, the user activates the TOE, and connects to the Nubo Management Server. Once registered, the user is required to authenticate itself to the Management server on successive sessions with the VMI environment.

1.3 TOE Description

The TOE is the thin client executing on mobile devices. It implements a user interface to virtual mobile applications executing on VMI servers. The TOE runs on Samsung Galaxy S10 devices with Android 12 operating system.

1.3.1 Evaluated

The TOE was tested on the following mobile device:

Table 1: Test System

Device Name	Chipset Vendor	SoC	Base Model Number	32 bits/64 bits
Galaxy S10	Qualcomm	Snapdragon 855	SM-G973	64 bits

There are no equivalency claims.

1.3.2 Software Requirements

Operating system: Android 12.0 with Linux kernel 4.14.

1.3.3 Hardware Requirements

A Samsung Galaxy S10 device (Qualcomm Snapdragon 855, 64 bits).

1.4 Physical Boundary

The physical boundary of the TOE is:

1) the thin client Android application (TOE): Nubo Client Version 3.2 and
2) the security guidance documentation. The security guidance is identified as follows:
   - *Nubo Client Version 3.2 Guidance Document*, October 2023

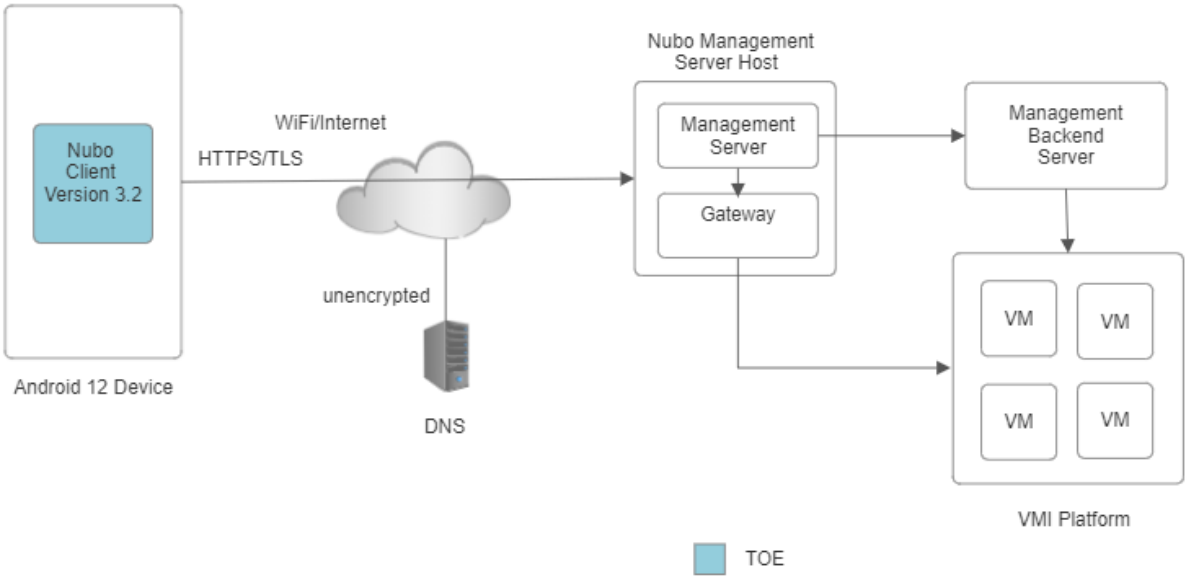
The TOE is the Nubo client software application version 3.2 and is packaged with the following libraries:

- com.afollestad.material-dialogs, v0.9.4.2
- Material Components for Android, v1.2.1
- fr.bmartel:http-endec, v1.04
- io.jsonwebtoken:jjwt, v0.9.1
- libphonenumber, v8.12.17
- com.karumi:dexter, v6.2.2
- Androidx.multidex, v2.0.1
- Androidx.preference:preference, v1.1.1
- Androidx.appcompat:appcompat, v1.2.0
- Androdx.constraintlayout:constraintlayout, v2.0.4
- Androidx.biometric:biometric, v1.1.0
- Gstremer, android-universal-1.20.3
- Conscrypt, v2.5.2
- BoringSSL, No specific version, source is dated April 2, 2021
- Firebase messaging, v21.0.1
- Gson, v2.8.6
- Guava, v30.1-android
- commons-net, v20030805.205232.

The BoringSSL library provides the cryptographic library. This library is NIST CAVP certified. Details of the certification can be found in Table 2.

The TOE's operational environment is illustrated in Figure 1. The TOE is the client App and is as defined in this ST and highlighted in the figure. Other components constitute the Nubo thin client infrastructure required for the full operation of the TOE.

Figure 1: TOE Operational Environment



The following external components are required by the TOE:

- Android 12 OS installed on a supported platform host (refer to Table 1).
- Platform connection to the Internet – either using WiFi or Cellular wireless service.
- DNS Server – access to a Domain Name System Server to resolve FQDN to IP addresses. Access is automatic if connected to a WiFi or Cellular wireless service.
- Nubo Management Server Host – a frontend server host running Nubo Management Server and Nubo Gateway. TOE users are instructed in the ancillary guidance document that the URL of this host is https://cc.nubo.co.
- Nubo Management Server – a frontend software server running on the Nubo Server Management Host that processes TOE user activation and login and communicates with the Nubo Gateway, the Management Backend Server, and the TOE.
- Nubo Gateway – a frontend application running on the Nubo Management Server Host that implements the connection for executing the virtual applications and communicates with the Nubo Management Server.
- Management Backend Server – a backend server that supports databases and other supporting services for the Nubo Management Server Host and the VMI Platform.
- VMI platform – A host supporting multiple virtual machines. At least one VM is required.

The Nubo Server is configured with information about the Nubo Gateway, Management Backend Server, VMI platforms, and VMs. Therefore, the configuration of the backend environment is not required by the TOE.

1.5 Logical Boundary

The TOE implements the security functions and security mechanisms identified in the following Sections.

1.5.1 Cryptographic Support

The TOE implements cryptographic functions for DRBG, key establishment, TLS and HTTPS protocols, and X.509 certificate validation. The TOE implements TLS using BoringSSL Library which in turn implements cryptographic functions using the BoringCrypto Library.

The CAVP certificates for the algorithms implemented by the TOE are included in the following table.

Table 2: CAVP Certificates

Algorithm	Standard	Modes Supported	CAVP Certificate #
Cryptographic Asymmetric Key Generation (FCS_CKM.1/AK)			
RSA KeyGen	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	2048 bits	#A1109

Algorithm	Standard	Modes Supported	CAVP Certificate #
ECDSA KeyGen	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	Curves P-256 and P-384	#A1109
Cryptographic Key Establishment (FCS_CKM.2)			
ECDHE Key Establishment	NIST SP 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	Curves P-256 and P-384	#A1109
Cryptographic Operation – Hashing (FCS_COP.1/Hash)			
SHA2-256	FIPS Pub 180-4	Digest size 256 bits	#A1109
SHA2-384	FIPS Pub 180-4	Digest size 384 bits	#A1109
Cryptographic Operation – Keyed-Hash Message Authentication (FCS_COP.1/KeyedHash)			
HMAC-SHA2-256	FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'	Key size 256 bits, block size 512 bits, digest size 256 bits	#A1109
HMAC-SHA2-384	FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'	Key size 384 bits, block size 512 bits, digest size 384 bits	#A1109
Cryptographic Operation – Signing (FCS_COP.1/Sig)			
RSA Digital Signature Algorithm (rDSA)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.	2048-bits	#A1109
Cryptographic Operation - Encryption/Decryption (FCS_COP.1/SKC)			
AES-GCM	NIST SP 800-38D	256 bits	#A1109
Random Bit Generation from Application (FCS_RBG_EXT.2)			
CTR_DRBG	NIST SP 800-90A	AES-256	#A1109

1.5.2 User Data Protection

The TOE stores sensitive user data (such as the user's full name and email address) encrypted in local files. These files are private to the TOE. The TOE also stores unencrypted cache files of graphical resources that are fetched from the server, which are pre-defined as non-sensitive data.

The TOE can access physical resources on the mobile device but does not store locally any data fetched from a physical resource.

1.5.3 Identification and Authentication

The identification of a user is comprised of the email address of the user and a unique client activation code, which identifies the specific TOE installation on a specific device. Authentication of the user of the TOE to the Nubo Management Server is one factor.

1.5.4 Security Management

The TOE does not have default credentials. The user selects the credentials when registering to the Management Server. The TOE uses the platform mechanism for storing configuration settings.

When the TOE is installed for the first time, it is not recognized by the remote system and must be activated prior to becoming operational. The user sends an activation request to which the Management Server responds either by a Client Activation Key or by a rejection of the activation. If activation is successful, the TOE saves the Client Activation Key in the Android keystore.

Once installed, the TOE may be upgraded and patches may be obtained from the Google Play Store if an appropriate upgrade is available.

1.5.5 Privacy

Personal Identifiable Information (PII) collected during the activation is transmitted to the Management Server over a trusted channel. The PII is a password created by the user. User consent is required before transmitting the information to the server.

1.5.6 Protection of the TSF

The Android platform provides protection of the TSF data. The platform protection mechanisms include checks that the TOE is properly signed and protection of the TOE and TOE Data from access by other apps. Secure delivery of the TOE and updates is accomplished though the delivery of the TOE and updates from the Google Play Store.

1.5.7 Trusted Channels

The TOE establishes a TLS 1.2 connection for all communications with the Management Server. The channel is used for identification, configuration, authentication, receiving remote display data from the virtual apps, and sending user input data to the servers and to the virtual apps.

1.6 Product Functionality not Included in the Scope of the Evaluation

- Fingerprint authentication functionality has been excluded from the evaluation.
- The evaluated configuration only supports the default Nubo Management Server Host (https://cc.nubo.co).

2 Conformance Claims

2.1 Common Criteria Conformance

- *Common Criteria for Information Technology Security Evaluations Part 2: Security functional components*, Version 3.1, Revision 5, April 2017: Part 2 extended.
- *Common Criteria for Information Technology Security Evaluations Part 2: Security assurance components*, Version 3.1, Revision 5, April 2017: Part 3 conformant.

2.2 Protection Profile Conformance

This Security Target claims exact conformance to the *Protection Profile for Application Software*, Version 1.4, October 7, 2021 [PP_APP].

This Security Target claims exact conformance to the *Functional Package for Transport Layer Security (TLS)*, Version 1.1, March 1, 2019 [PKG_TLS].

2.3 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date that are applicable to [PP_APP] are addressed in the table below.

Table 3: Technical Decisions for PP_APP

TD #[1],[2]	Summary	Applicable	Rationale/Notes
TD0798	Static Memory Mapping Exceptions	Yes	Applies to TSS & Test AA.
TD0780	FIA_X509_EXT.1 Test 4 Clarification	Yes	Applies to Test only.
TD0756	Update for platform-provided full disk encryption	Yes	Applies to Test only.
TD0747	Configuration Storage Option for Android	Yes	Applies to Test only.
TD0743	FTP_DIT_EXT.1.1 Selection exclusivity	Yes	
TD0736	Number of elements for iterations of FCS_HTTPS_EXT.1	No	The ST does not include FCS_HTTPS_EXT.1/Server.
TD0719	ECD for PP APP V1.3 and 1.4	Yes	
TD0717	Format changes for PP_APP_V1.4	Yes	
TD0664	Testing activity for FPT_TUD_EXT.2.2	Yes	Applies to Test only.
TD0650	Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	No	The TOE does not implement PP-Module for VPN Clients
TD0628	Addition of Container Image to Package Format	Yes	

[1] The listed TDs have been modified per TD0717 (TD0659 and TD0626 were archived).
[2] The listed TDs have been modified per TD0743 (TD0655 was archived).

All NIAP Technical Decisions (TDs) issued to date that are applicable to [PKG_TLS] are addressed in the Table below.

Table 4: Technical Decisions for PKG_TLS

TD #	Summary	Applicable	Rationale/Notes
TD0779	Updated Session Resumption Support in TLS package V1.1	No	The TOE does not implement TLS Server
TD0770	TLSS.2 connection with no client cert	No	The TOE does not implement TLS Server
TD0739	PKG_TLS_V1.1 has 2 different publication dates	Yes	
TD0726	Corrections to (D) TLSS SFRs in TLS 1.1 FP	No	The TOE does not implement TLS Server
TD0513	CA Certificate loading	Yes	Applies to Test only.
TD0499	Testing with pinned certificates	Yes	Applies to Test only.
TD0469	Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	No	The TOE does not implement TLS Server
TD0442	Updated TLS Ciphersuites for TLS Package	Yes	

3 Security Problem Definition

The SPD as defined in [PP_APP] is fully applicable. [PKG_TLS] does not introduce additional SPD elements.

3.1 Threats

T.NETWORK_ATTACK

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter communications between the application software and other endpoints in order to compromise it.

T.NETWORK_EAVESDROP

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the application and other endpoints.

T.LOCAL_ATTACK

An attacker can act through unprivileged software on the same computing platform on which the application executes. Attackers may provide maliciously formatted input to the application in the form of files or other local communications.

T.PHYSICAL_ACCESS

An attacker may try to access sensitive data at rest.

3.2 Assumptions

A.PLATFORM

The TOE relies upon a trustworthy computing platform with a reliable time clock for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.

A.PROPER_USER

The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.

A.PROPER_ADMIN

The administrator of the application software is not careless, willfully negligent or hostile, and administers the software in compliance with the applied enterprise security policy.

3.3 Organizational Security Policies

There are no Organizational Security Policies defined in [PP_APP].

4 Security Objectives

This section states the security objectives for the TOE and the security objectives for the operational environment of the TOE.

4.1 Security objectives for the TOE

The Security Objectives as defined in [PP_APP] are fully applicable. [PKG_TLS] does not introduce additional Security Objectives.

O.INTEGRITY

Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom, if ever, shipped without errors. The ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security.

Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options.

O.QUALITY

To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs.

O.MANAGEMENT

To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the

deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII.

O.PROTECTED_STORAGE

To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data.

O.PROTECTED_COMMS

To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application.

4.2 Security Objectives for the Operational Environment

The Security Objectives for the environment as defined in [PP_APP] are fully applicable. [PKG_TLS] does not introduce additional Security Objectives for the environment.

OE.PLATFORM

The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.

OE.PROPER_USER

The user of the application software is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.

OE.PROPER_ADMIN

The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

4.3 Security Objectives Rationale

The security objectives rationale is as in [PP_APP] and [PKG_TLS]. It is not reproduced herein.

5 Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the following Protection Profile (PP) and Functional Package:

- *Protection Profile for Application Software*, Version 1.4, October 7, 2021
- *Functional Package for Transport Layer Security (TLS)*, Version 1.1, March 1, 2019

As a result, any selection, assignment, or refinement operations already performed by that PP or the Package on the claimed SFRs are not identified here (i.e., they are not formatted in accordance with the conventions specified in Section 5.2 of this ST). Formatting conventions are only applied on SFR text that was chosen at the ST author's discretion and are defined in section 5.2

5.1 Extended Requirements

All the extended requirements in this ST have been drawn from the [PP_APP] or [PKG_TLS]. This document identifies the extended SFRs; since they have not been redefined in this ST, the [PP_APP] or [PKG_TLS] should be consulted for more information regarding these extensions to CC Parts 2 and 3.

- FCS_CKM_EXT.1 Cryptographic Key Generation Services
- FCS_HTTPS_EXT.1/Client HTTPS Protocol
- FCS_RBG_EXT.1 Random Bit Generation Services
- FCS_RBG_EXT.2 Random Bit Generation from Application
- FCS_STO_EXT.1 Storage of Credentials
- FCS_TLS_EXT.1 TLS Protocol
- FCS_TLSC_EXT.1 TLS Client Protocol
- FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension
- FDP_DAR_EXT.1 Encryption Of Sensitive Application Data
- FDP_DEC_EXT.1 Access to Platform Resources
- FDP_NET_EXT.1 Network Communications
- FIA_X509_EXT.1 X.509 Certificate Validation
- FIA_X509_EXT.2 X.509 Certificate Authentication
- FMT_CFG_EXT.1 Secure by Default Configuration
- FMT_MEC_EXT.1 Supported Configuration Mechanism
- FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information
- FPT_AEX_EXT.1 Anti-Exploitation Capabilities
- FPT_API_EXT.1 Use of Supported Services and APIs
- FPT_IDV_EXT.1 Software Identification and Versions
- FPT_LIB_EXT.1 Use of Third Party Libraries
- FPT_TUD_EXT.1 Integrity for Installation and Update
- FPT_TUD_EXT.2 Integrity for Installation and Update
- FTP_DIT_EXT.1 Protection of Data in Transit

5.2 Conventions

The CC defines operations on security requirements. The font conventions listed below state the conventions used in this ST to identify the operations completed in this Security Target. Whenever a specific formatting is required by [PP_APP] or [PKG_TLS], that formatting is used instead of the convention below.

- Refinement: Any added text is indicated with **bold font** and any removed text is ~~overstricken~~.
- Iteration: Indicated by a slash followed by an identifier, e.g., FCS_HTTPS_EXT.1/Client.
- Assignment: indicated in underlined text
- Selection: *indicated in italics*
- Assignments within selections: *indicated in italics and underlined text*

5.3 TOE Security Functional Requirements

The Security Functional Requirements included in this section are derived from Part 2 of the *Common Criteria for Information Technology Security Evaluation*, Version 3.1, Revision 5, with the extended functional components as defined in [PP_APP] and [PKG_TLS]. The Security Functional Requirements applicable to the TOE are summarized below.

Table 5: Security Functional Requirements

SFR	Description
FCS_CKM_EXT.1[3]	Cryptographic Key Generation Services
FCS_CKM.1/AK	Cryptographic Asymmetric Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_COP.1/Hash	Cryptographic Operation – Hashing
FCS_COP.1/KeyedHash	Cryptographic Operation – Keyed-Hash Message Authentication
FCS_COP.1/Sig	Cryptographic Operation – Signing
FCS_COP.1/SKC	Cryptographic Operation - Encryption/Decryption
FCS_HTTPS_EXT.1/Client	HTTPS Protocol
FCS_RBG_EXT.1	Random Bit Generation Services
FCS_RBG_EXT.2	Random Bit Generation from Application
FCS_STO_EXT.1	Storage of Credentials
FCS_TLS_EXT.1	TLS Protocol
FCS_TLSC_EXT.1	TLS Client Protocol
FCS_TLSC_EXT.5	TLS Client Support for Supported Groups Extension
FDP_DAR_EXT.1	Encryption Of Sensitive Application Data

[3] Applied TD0717 (renaming FCS_CKM.1).

SFR	Description
FDP_DEC_EXT.1	Access to Platform Resources
FDP_NET_EXT.1	Network Communications
FIA_X509_EXT.1	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FMT_CFG_EXT.1	Secure by Default Configuration
FMT_MEC_EXT.1	Supported Configuration Mechanism
FMT_SMF.1	Specification of Management Functions
FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information
FPT_AEX_EXT.1	Anti-Exploitation Capabilities
FPT_API_EXT.1	Use of Supported Services and APIs
FPT_IDV_EXT.1	Software Identification and Versions
FPT_LIB_EXT.1	Use of Third Party Libraries
FPT_TUD_EXT.1	Integrity for Installation and Update
FPT_TUD_EXT.2	Integrity for Installation and Update
FTP_DIT_EXT.1	Protection of Data in Transit

5.3.1 Cryptographic Support (FCS)

FCS_CKM_EXT.1[4] Cryptographic Key Generation Services

FCS_CKM_EXT.1.1[5] The application shall [

- *implement asymmetric key generation*

].

FCS_CKM.1/AK Cryptographic Asymmetric Key Generation

FCS_CKM.1.1/AK[6] The application shall [

- *implement functionality*

] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- *[RSA schemes] using cryptographic key sizes of [2048-bit or greater] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3],*

[4] Applied TD0717 (PP_APP) (renaming FCS_CKM.1).
[5] Applied TD0717 (PP_APP) (renaming FCS_CKM.1).
[6] Applied TD0717 (PP_APP).

- *[ECC schemes] using ["NIST curves" P-384 and [P-256]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4]*

].

FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1 The application shall [*implement functionality*] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- *[Elliptic curve-based key establishment schemes] that meets the following: [NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"]*

].

FCS_COP.1/Hash Cryptographic Operation – Hashing

FCS_COP.1.1/Hash[7] The application shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm [

- *SHA-256,*
- *SHA-384*

] and message digest sizes [

- *256,*
- *384*

] bits that meet the following: [FIPS Pub 180-4].

FCS_COP.1/KeyedHash Cryptographic Operation - Keyed-Hash Message Authentication

FCS_COP.1.1/KeyedHash[8] The application shall perform [keyed-hash message authentication] in accordance with a specified cryptographic algorithm [

- *HMAC-SHA-256*
- *HMAC-SHA-384*

] and [

- *no other algorithm*

] with key sizes [256, 384] and message digest sizes [*256, 384*] and [*no other size*] bits that meet the following: [FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'].

[7] Applied TD0717.
[8] Applied TD0717.

FCS_COP.1/Sig Cryptographic Operation – Signing

FCS_COP.1.1/Sig[9] The application shall perform [cryptographic signature services (generation and verification)] in accordance with a specified cryptographic algorithm [

- *RSA schemes using cryptographic key sizes of [2048-bit or greater] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5].*

].

FCS_COP.1/SKC Cryptographic Operation - Encryption/Decryption

FCS_COP.1.1/SKC[10] The application shall perform [encryption/decryption] in accordance with a specified cryptographic algorithm [

- *AES-GCM (as defined in NIST SP 800-38D) mode*

] and cryptographic key sizes [*256-bit*].

FCS_HTTPS_EXT.1/Client HTTPS Protocol

FCS_HTTPS_EXT.1.1/Client The application shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2/Client The application shall implement HTTPS using TLS as defined in the Functional Package for TLS.

FCS_HTTPS_EXT.1.3/Client The application shall [*not establish the application-initiated connection*] if the peer certificate is deemed invalid.

FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1 The application shall [

- *implement DRBG functionality*

] for its cryptographic operations.

FCS_RBG_EXT.2 Random Bit Generation from Application

FCS_RBG_EXT.2.1 The application shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [*CTR_DRBG (AES)*].

FCS_RBG_EXT.2.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [

- *no other noise source*

] with a minimum of [

- *256 bits*

[9] Applied TD0717.
[10] Applied TD0717.

] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1 The application shall [

- *invoke the functionality provided by the platform to securely store* [encryption key used to encrypt/decrypt sensitive User credentials (User full name, User email address) in non-volatile memory, Client Activation Key]

] to non-volatile memory.

FCS_TLS_EXT.1 TLS Protocol

FCS_TLS_EXT.1.1 The product shall implement [

- *TLS as a client*

].

FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1[11] The product shall implement TLS 1.2 (RFC 5246) and [*no earlier TLS versions*] as a client that supports the cipher suites [

- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*

] and also supports functionality for [

- *none*

].

FCS_TLSC_EXT.1.2 The product shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3 The product shall not establish a trusted channel if the server certificate is invalid [

- *with no exceptions*

].

FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension

FCS_TLSC_EXT.5.1 The product shall present the Supported Groups Extension in the Client Hello with the supported groups [

- *secp384r1*

].

[11] Applied TD0442.

5.3.2 User Data Protection (FDP)

FDP_DAR_EXT.1 Encryption of Sensitive Application Data

FDP_DAR_EXT.1.1 The application shall [

- *leverage platform-provided functionality to encrypt sensitive data*

] in non-volatile memory.

FDP_DEC_EXT.1 Access to Platform Resources

FDP_DEC_EXT.1.1 The application shall restrict its access to [

- *network connectivity,*
- *camera,*
- *microphone,*
- *location services,*
- *Bluetooth,*

].

FDP_DEC_EXT.1.2 The application shall restrict its access to [

- *no sensitive information repositories*

].

FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1 The application shall restrict network communication to [

- *user-initiated communication for [*
  - *activation of the TOE with the Management Server running on the Management Server Host,*
  - *user authentication with the Management Server],*

].

5.3.3 Identification and Authentication (FIA)

FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1 The application shall [*implement functionality*] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.

- The application shall validate that any CA certificate includes caSigning purpose in the key usage field.
- The application shall validate the revocation status of the certificate using [*CRL as specified in RFC 5280 Section 6.3*].
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the EKU field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
    - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
    - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

FIA_X509_EXT.1.2 The application shall treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [*HTTPS, TLS*].

FIA_X509_EXT.2.2 When the application cannot establish a connection to determine the validity of a certificate, the application shall [*not accept the certificate*].

5.3.4 Security Management (FMT)

FMT_CFG_EXT.1 Secure by Default Configuration

FMT_CFG_EXT.1.1 The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

FMT_CFG_EXT.1.2 The application shall be configured by default with file permissions which protect the application binaries and data files from modification by normal unprivileged users.

FMT_MEC_EXT.1 Supported Configuration Mechanism

FMT_MEC_EXT.1.1 The application shall [*invoke the mechanisms recommended by the platform vendor for storing and setting configuration options*].

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions [

- *[Downloading and installing a TOE upgrade]*

].

5.3.5 Privacy

FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information

FPR_ANO_EXT.1.1 The application shall [

- *require user approval before executing [Activation of the TOE with the Management Server]*

].

5.3.6 Protection of the TSF (FPT)

FPT_AEX_EXT.1 Anti-Exploitation Capabilities

FPT_AEX_EXT.1.1 The application shall not request to map memory at an explicit address except for [no exceptions].

FPT_AEX_EXT.1.2 The application shall [

- *not allocate any memory region with both write and execute permissions*

].

FPT_AEX_EXT.1.3 The application shall be compatible with security features provided by the platform vendor.

FPT_AEX_EXT.1.4 The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

FPT_AEX_EXT.1.5 The application shall be built with stack-based buffer overflow protection enabled.

FPT_API_EXT.1 Use of Supported Services and APIs

FPT_API_EXT.1.1 The application shall use only documented platform APIs.

FPT_IDV_EXT.1 Software Identification and Versions

FPT_IDV_EXT.1.1 The application shall be versioned with [*[Version number consisting of major.minor.optional build number]*].

Application Note: the TOE is identified by the major.minor fields. Some versions of the TOE will include an optional build number.

FPT_LIB_EXT.1 Use of Third Party Libraries

FPT_LIB_EXT.1.1 The application shall be packaged with only [

- com.afollestad.material-dialogs, v0.9.4.2
- Material Components for Android, v1.2.1
- fr.bmartel:http-endec, v1.04
- io.jsonwebtoken:jjwt, v0.9.1
- libphonenumber, v8.12.17
- com.karumi:dexter, v6.2.2
- Androidx.multidex, v2.0.1
- Androidx.preference:preference, v1.1.1
- Androidx.appcompat:appcompat, v1.2.0
- Androdx.constraintlayout:constraintlayout, v2.0.4
- Androidx.biometric:biometric, v1.1.0
- Gstremer, android-universal-1.20.3
- Conscrypt, v2.5.2
- BoringSSL, No specific version, source is dated April 2, 2021
- Firebase messaging, v21.0.1
- Gson, v2.8.6
- Guava, v30.1-android
- commons-net, v20030805.205232.

].

FPT_TUD_EXT.1 Integrity for Installation and Update

FPT_TUD_EXT.1.1 The application shall [*leverage the platform*] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2 The application shall [*provide the ability*] to query the current version of the application software.

FPT_TUD_EXT.1.3 The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.4 Application updates shall be digitally signed such that its platform can cryptographically verify them prior to installation.

FPT_TUD_EXT.1.5 The application is distributed [*as an additional software package to the platform OS*].

FPT_TUD_EXT.2 Integrity for Installation and Update

FPT_TUD_EXT.2.1[12] The application shall be distributed using [*the format of the platform-supported package manager*].

[12] Applied TD0628.

FPT_TUD_EXT.2.2 The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.

FPT_TUD_EXT.2.3 The application installation package shall be digitally signed such that its platform can cryptographically verify them prior to installation.

5.3.7 Trusted Path/Channel (FTP)

FTP_DIT_EXT.1 Protection of Data in Transit

FTP_DIT_EXT.1.1[13] The application shall [

- *encrypt all transmitted [data] with [TLS as a client as defined in the Functional Package for TLS for [providing the transport layer for the HTTPS client connections]]*,
- *invoke platform-provided functionality to encrypt all transmitted data with [HTTPS] for [connections to the Nubo Management Server] ,*

] between itself and another trusted IT product.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are stated in [PP_APP] and [PKG_TLS]. They are summarized below.

Table 6: Security Assurances

Assurance Classes	Assurance Component	Description
ASE: Security Target Evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives
	ASE_REQ.1	Security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
ADV: Development	ADV_FSP.1	Basic Functional Specification
AGD: Guidance Documentation	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative Procedures
ALC: Life-cycle Support	ALC_CMC.1	Labeling of the TOE

[13] Applied TD0743.

Assurance Classes	Assurance Component	Description
	ALC_CMS.1	TOE CM Coverage
	ALC_TSU_EXT.1	Timely Security Updates
ATE: Tests	ATE_IND.1	Independent Testing – Conformance
AVA: Vulnerability Assessment	AVA_VAN.1	Vulnerability Survey

As a functional package, the TLS Package does not define its own SARs. The expectation is that all SARs required by the [PP_APP] will apply to the entire TOE, including the portions addressed by the TLS Package. Consequently, the evaluation activities specified in the [PP_APP] apply to the entire TOE evaluation, including any changes made to them by subsequent NIAP Technical Decisions as summarized in section 2.3 above.

6 TOE Summary Specification

This table below describes how the Security Functional Requirements stated for the TOE and the extended security assurance requirement ALC_TSU_EXT.1 (Timely Security Updates) are met by the TOE.

Table 7: SFR Rationale

SFR	Rationale
FCS_CKM_EXT.1 **FCS_CKM.1/AK** **FCS_CKM.2**	The TOE generates 2048-bit RSA keys for the use in digital signatures. The keys are generated using the BoringCrypto modules of the BoringSSL libraries. The CAVP details are given in Section 1.5.1 The TOE implements ECDSA Key Generation, Signature Generation, and Signature Verification as part of TLS trusted channel establishment. NIST curves P-256 and P-384 are supported. The TOE implements key establishment using the BoringSSL Library as part of TLS session establishment. The key establishment scheme used is Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). The CAVP details of the BoringCrypto routines used by the BoringSSL library are given in Sect. 1.5.1
FCS_COP.1/SKC **FCS_COP.1/Hash** **FCS_COP.1/Sig** **FCS_COP.1/KeyedHash**	The TOE encrypts and decrypts all network data using TLS v1.2. The following algorithms are implemented: • AES-GCM 256-bit encryption and decryption, • SHA-256 and SHA-384 Hash function used in association with digital signature computation, • HMAC-SHA-256 and HMAC-SHA-384 keyed-hash message authentication, • Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) key establishment, and • 2048 bit RSA for signature computation. CAVP certificate numbers for the cryptographic algorithms implemented by the TOE are given in Sect. 1.5.1
FCS_HTTPS_EXT.1/Client	The TOE uses platform-provided APIs (javax.net.ssl.HttpsURLConnection) to implement HTTPS, and sets up an SSL Socket factory which uses the implemented TLS functionality. The TLS implementation used for HTTPS is in full accordance with RFC2818.
FCS_RBG_EXT.1 **FCS_RBG_EXT.2**	The TOE implements DRBG functionality using the BoringCrypto modules of the BoringSSL Library. The RNG algorithm that used is AES-256 CTR_DRBG. The TOE seeds its DRBG using 256-bits of data from /dev/random, thus ensuring at least 256-bits of entropy. The CAVP details are given in Sect. 1.5.1 Additional information related to entropy functionality of the TOE can be reviewed in the Entropy Assessment Report (EAR) provided as an ancillary document.

SFR	Rationale
FCS_STO_EXT.1	All User credentials (User full name, User email address, User job title, Nubo Management Server URL) are encrypted by the platform with AES-CBC using a 256-bit key. The key is stored in the Android private keystore. The encrypted User credentials are stored in /data/data/package/shared_prefs/ with the MODE_PRIVATE flag set. The TOE stores the Client Activation Key received from the Nubo Management Server in the Android keystore. The Client Activation Key is issued to the TOE by the Management Server upon successful activation of the TOE by the user and stored by the TOE in the platform keystore for future use. The Client Activation Key is used to ensure that only an approved TOE can authenticate, and to prevent use of stolen or guessed passwords in other devices.
FCS_TLS_EXT.1 **FCS_TLSC_EXT.1**	The TOE implements TLS v1.2 by using Conscrypt and BoringSSL libraries included in the TOE. No other TLS versions are supported. Only the following cipher suite is supported: • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289. The TOE uses the Conscrypt and BoringSSL libraries for all certificate validations. BoringSSL supports Common Name (CN) and Subject Alternative Name (SAN) (DNS and IP address) as reference identifiers. The TOE supports the use of wildcards in X.509 reference identifiers (CN and SAN). The TOE utilizes certificate pinning. Connections to the Nubo Management Server are pinned to certificates built into the TOE application.
FCS_TLSC_EXT.5	The implemented TLS library implements the following Elliptic Curve extension. No configuration is required by the user: • secp384r1
FDP_DAR_EXT.1	The TOE processes the following sensitive data: User full name, User email address, User job title, and Nubo Management Server URL. When stored in the non-volatile memory, all sensitive data is stored by the TOE in /data/data/package/shared_prefs/ with the MODE_PRIVATE flag set and encrypted by the platform using AES-CBC using a 256-bit key. When transmitted over a network to the Management Server, all sensitive data is protected by HTTPS/TLS.
FDP_DEC_EXT.1	The TOE can access the following platform resources: network connectivity, camera, microphone, location services, and Bluetooth. This access aims to support the user's access of applications in the Nubo VMI. No sensitive information repositories are accessed.
FDP_NET_EXT.1	The TOE opens connections to the Nubo Management Server. No incoming connections are accepted. The connection is opened for the following functions: • Activation of the TOE with the Management Server and • User authentication with the Management Server.

SFR	Rationale
FIA_X509_EXT.1	All certificate validations are implemented using the Conscrypt and BoringSSL libraries. The libraries also validate X.509v3 certificates and check their revocation status. The TOE checks the validity of all imported CA certificates by checking for the presence of the basicConstraints extension and that the CA flag is set to TRUE as the TOE imports the certificate into the TOE's Trust Anchor Database. If the TOE detects an absence of the basicConstraints extension or the CA flag, the TOE imports the certificate as a user public key and adds it to the keystore (instead of the Trust Anchor Database). The TOE also checks for the presence of the basicConstraints extension and CA flag in each CA certificate presented in a peer server's certificate chain. Similarly, the TOE verifies the extendedKeyUsage Server Authentication purpose during certificate validation. The TOE's certificate validation algorithm examines each certificate in the path starting with the peer's certificate: • It first checks the validity of that certificate (e.g., the certificate has not expired, the certificate is valid, whether the certificate contains the appropriate X.509 extensions [e.g., the CA flag in the basic constraints extension for a CA certificate, or that a server certificate contains the Server Authentication purpose in the extendedKeyUsage field]). • The algorithm then verifies each certificate in the chain (applying the same rules but also ensuring that the Issuer of each certificate matches the Subject in the next rung "up" in the chain and that the chain ends in a self-signed certificate present in either the TOE's trusted anchor database or matches a specified Root CA). • Finally, the TOE checks the revocation status of all certificates in the chain. • CRL checking as specified in RFC 5280 Section 6.3 revocation checking will be attempted on certificates that have listed distribution points.
FIA_X509_EXT.2	The TOE uses X.509v3 certificates as defined by RFC 5280 for server authentication for HTTPS/TLS connections. Certificates for the Nubo Management Server are built into the TOE application. When revocation status cannot be determined, certificates are not accepted.
FMT_CFG_EXT.1	There are no default credentials within the TOE. Users of the TOE must activate with the Nubo Management Server before any applications in the Nubo VMI can be accessed.
FMT_MEC_EXT.1	The TOE stores configuration data in /data/data/package/shared_prefs/ with the MODE_PRIVATE flag set. Configuration data stored by the TOE is the URL of the Nubo Management Server and user creation data. The default Nubo Management Server URL is built in the TOE.
FMT_SMF.1	The TOE implements one management function: upgrading the TOE. The TOE is distributed through the Google Play store. Once installed and activated, the user may download upgrades to the TOE from the Google

SFR	Rationale
	Play Store. All upgrades are protected by the digital signature computed using the Nubo software signature key. The platform verifies the signature. If the signature verification fails, the Android OS will reject the upgrade.
FPR_ANO_EXT.1	PII collected from the user is transmitted to the Management Server during the TOE activation. The information is collected from the user using a GUI which seeks for user's consent for transmitting the information to the Management Server. The GUI also informs the user of the PII being transmitted over a secure connection. The user may supply additional PII when interacting with applications in the Nubo VMI, but the applications are not part of the TOE and the TOE is not aware of the semantics of any application data communicated between the TOE and the VMI Server. The TOE transmits this data but is unaware of the nature of the data.
FPT_AEX_EXT.1	The TOE is a Java application. Memory mapping and permissions on memory regions are not accessible to Java applications. Native libraries are incorporated into the TOE; none of those map code to specific locations or include memory regions with both write and execute permissions. Native libraries are compiled with the "-fstack-protector-all" and "-pie" flags.
FPT_API_EXT.1	The following Android Java APIs are used by the TOE: • android.security.keystore.KeyGenParameterSpec • android.security.keystore.KeyProperties • java.security.InvalidAlgorithmParameterException • java.security.InvalidKeyException • java.security.Key • java.security.KeyManagementException • java.security.KeyStore • java.security.KeyStoreException • java.security.MessageDigest • java.security.NoSuchAlgorithmException • java.security.NoSuchProviderException • java.security.PublicKey • java.security.SecureRandom • java.security.UnrecoverableKeyException • java.security.cert.Certificate • java.security.cert.CertificateException • java.security.cert.CertificateFactory • java.security.spec.InvalidParameterSpecException • javax.crypto.BadPaddingException • javax.crypto.Cipher • javax.crypto.IllegalBlockSizeException • javax.crypto.KeyGenerator • javax.crypto.Mac • javax.crypto.NoSuchPaddingException • javax.crypto.SecretKey • javax.crypto.SecretKeyFactory

SFR	Rationale
	• javax.crypto.spec.IvParameterSpec • javax.crypto.spec.SecretKeySpec • javax.net.ssl.HttpsURLConnection • javax.net.ssl.SSLContext • javax.net.ssl.SSLPeerUnverifiedException • javax.net.ssl.SSLSession • javax.net.ssl.SSLSocket • javax.net.ssl.SSLSocketFactory • javax.security.auth.x500.X500Principal
FPT_IDV_EXT.1	The TOE version number can be displayed through the app settings and in the platform settings. The app version is also visible in Google Play. The format for the version is "3.X(.Y)" where 3 is the major version, X is the minor version and Y is an optional build number. The version number of the evaluated TOE is 3.2.
FPT_LIB_EXT.1	Only the following libraries are packaged with the TOE: • com.afollestad.material-dialogs, v0.9.4.2 • Material Components for Android, v1.2.1 • fr.bmartel:http-endec, v1.04 • io.jsonwebtoken:jjwt, v0.9.1 • libphonenumber, v8.12.17 • com.karumi:dexter, v6.2.2 • Androidx.multidex, v2.0.1 • Androidx.preference:preference, v1.1.1 • Androidx.appcompat:appcompat, v1.2.0 • Androdx.constraintlayout:constraintlayout, v2.0.4 • Androidx.biometric:biometric, v1.1.0 • Gstremer, android-universal-1.20.3 • Conscrypt, v2.5.2 • BoringSSL, No specific version, source is dated April 2, 2021 • Firebase messaging, v21.0.1 • Gson, v2.8.6 • Guava, v30.1-android • commons-net, v20030805.205232
FPT_TUD_EXT.1	The initial TOE installation, updates, and patches to the TOE are distributed by the Google Play. Each release is packaged in APK format and signed with a Nubo Software certificate. In the initial TOE installation, the user installs the app signed by Nubo. In each case of upgrading a package, the platform checks and validates the signature of the package. In case the user downloads an APK not signed by Nubo, or fails signature validation, the platform will disallow the installation of the package. The user may query the current version of the TOE by entering the Settings screen, which is available in the menu in all TOE screens.

SFR	Rationale
	The TOE does not download, modify, replace or update its own binary code.
FPT_TUD_EXT.2	The TOE is packaged in the Android application package (APK) format. When the application is deleted, the platform automatically deletes all its related files and settings that are stored in the application working directory. The TOE does not store any file outside of the application working directory.
FTP_DIT_EXT.1	The TOE communicates with the Nubo Management Server acting an HTTPS client. All external communications are protected by HTTPS/TLS: • For HTTPS, the TOE implements HTTPS using the platform-provided APIs (javax.net.ssl.HttpsURLConnection), and sets an SSL Socket factory that uses the implemented TLS functionality described in FCS_TLSC_EXT.1 • For TLS, the TOE creates an SSL Socket factory that uses the implemented TLS functionality described in FCS_TLSC_EXT.1, and creates the socket connection only with that factory.
ALC_TSU_EXT.1	The secure update process is as follows: 1. Nubo re-creates the software APK with a new version number. 2. An authorized Nubo engineer signs the APK using the app private key. The private key is protected by a password and it's only available to the authorized engineer. 3. Nubo uploads the updated APK to the Google Play Store and creates a new release for all the users. 4. Nubo updates its Management Server with a minimum version, which is the version of the security update. When a user with an old version of the TOE tries to use it, the TOE communicates with the Management Server and notifies the user that it cannot run without updating the TOE version. The TOE redirects the user to the Google Play Store to update the TOE. Given that the TOE is an Android app, the platform notifies the user when an update changes security permissions. The maximum time window between a public disclosure of a vulnerability and the upload of a security update to the Google Play is 14 days. The public availability of the update will be after the upload has been approved and distributed by the app store. The approval usually takes a few days. There may be vulnerabilities in the libraries or other third party components which are not in control of Nubo. In these cases, the issuance of the updates is up to the developer of the library of the third party component, and the response time by Nubo is from the making available of the patches to the vulnerabilities. Security issues may be securely reported to Nubo by anyone. Once a report has been received with the relevant information, a ticket is opened in the service portal at https://nubo.sysaidit.com/. Alternatively, issues may be reported by email to an address support@nubosoftware.com.

SFR	Rationale
	After such a report is submitted to Nubo, the following process starts: 1. A security engineer is assigned to the issue in the issues database. 2. The engineer verifies and reproduces the issue in a test environment. 3. Priority is assigned to the issue. 4. A party responsible for correcting the issue is nominated. 5. The fix is found and applied to the SW modules. 6. Integrating the fixed SW modules to a new release of the TOE.

7 Rationale

7.1 Conformance Claim Rationale

This Security Target includes the [PP_APP] and [PKG_TLS] Security Problem Definition, Security Objectives, and Security Assurance Requirements. The Security Target does not add, remove, or modify any of these items. Security Functional Requirements have been reproduced with the Protection Profile operations completed. All selections, assignments, and refinements made on the claimed Security Functional Requirements have been performed in a manner that is consistent with what is permitted by the [PP_APP] and [PKG_TLS]. The proper set of selection-based requirements have been claimed based on the selections made in the mandatory requirements. Consequently, the claims made by this Security Target are sufficient to address the TOE's security problem.

7.2 TOE Security Assurance Requirements Rationale

The rationale for choosing these security assurance requirements is that they define a minimum security baseline that is based on the anticipated threat level of the attacker, the security of the Operational Environment in which the TOE is deployed, and the relative value of the TOE itself. The assurance activities throughout the PP are used to provide tailored guidance on the specific expectations for completing the security assurance requirements.

7.3 TOE Security Functional Requirements Rationale

Table 8: Security Functions vs. Requirements Mapping

	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Privacy	Protection of the TSF	Trusted Channel
FCS_CKM_EXT.1	✓						
FCS_CKM.1/AK	✓						
FCS_CKM.2	✓						
FCS_COP.1/Hash	✓						
FCS_COP.1/KeyedHash	✓						

	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Privacy	Protection of the TSF	Trusted Channel
FCS_COP.1/Sig	✓						
FCS_COP.1/SKC	✓						
FCS_HTTPS_EXT.1/Client	✓						
FCS_RBG_EXT.1	✓						
FCS_RBG_EXT.2	✓						
FCS_STO_EXT.1	✓						
FCS_TLS_EXT.1	✓						
FCS_TLSC_EXT.1	✓						
FCS_TLSC_EXT.5	✓						
FDP_DAR_EXT.1		✓					
FDP_DEC_EXT.1		✓					
FDP_NET_EXT.1		✓					
FIA_X509_EXT.1			✓				
FIA_X509_EXT.2			✓				
FMT_CFG_EXT.1				✓			
FMT_MEC_EXT.1				✓			
FMT_SMF.1				✓			
FPR_ANO_EXT.1					✓		
FPT_AEX_EXT.1						✓	
FPT_API_EXT.1						✓	

	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Privacy	Protection of the TSF	Trusted Channel
FPT_IDV_EXT.1						✓	
FPT_LIB_EXT.1						✓	
FPT_TUD_EXT.1						✓	
FPT_TUD_EXT.2						✓	
FTP_DIT_EXT.1							✓