INTEGRITY Enterprise OS – Archon Edition Security Target

Acumen Security, LLC.

Table Of Contents

Revision History

Version	Date	Description
1.0	2/20/2022	Initial Release
1.1	4/5/2022	Addresses validator ECRs
1.2	4/26/2022	Addressed validator comments

1 Security Target Introduction

1.1 Security Target and TOE Reference

This section provides information needed to identify and control this ST and its TOE.

Category	Identifier
ST Title	INTEGRITY Enterprise OS – Archon Edition Security Target
ST Version	1.2
ST Date	4/26/2022
ST Author	Acumen Security, LLC.
TOE Identifier	INTEGRITY Enterprise OS – Archon Edition
TOE Software Version	Archon Edition 1.0
TOE Vendor	Archon Secure LLC
TOE Developer	Archon Secure LLC
Key Words	General Purpose, Operating System

Table 1 TOE/ST Identification

1.2 TOE Overview

The TOE is the INTEGRITY Enterprise OS – Archon Edition, which provides a secure computing environment for mobile platforms. The TOE provides end users with the ability to install their own custom user software in a high security sandbox, while maintaining a secure operating system enclave logically isolated from the end user's application.

1.2.1 TOE Product Type

The TOE type is a general-purpose operating system. It satisfies all the criterion to meet the Protection Profile for General Purpose Operating Systems v4.2.1 [OSPP].

1.3 TOE Description

1.3.1 Evaluated Configuration

The TOE is the INTEGRITY Enterprise OS – Archon Edition, which provides secure computing on the Archon ZV secure mobility platform. This includes the bootloader and all code that executes from device power on until the full OS is loaded, and runs only on the secure mobile platforms below:

Platform Name	CPU	HDD	RAM
Archon ZV 5400	Intel Core i5-8365U (Whiskey Lake microarchitecture)	250 GB	8 GB or 16GB

Table 2 Platform Identification

The TOE also requires the operational environment to provide the following to support its security functions:

- For administration of the TOE, one authorized administration server and one management server, which may be combined onto one platform.
- For certificate management, at least one CA/revocation server.

1.3.2 Physical Boundaries

Because the TOE is an operating system, it does not have physical boundaries. The TOE runs on a physical hardware platform provided by the OE.

1.3.3 Logical Scope of the TOE

The TOE implements the following security functional requirements from [OSPP] as listed below:

1.3.3.1 Audit Data Generation (FAU)

The TOE audits the following events and details:

- Audit all administrative functions.
- Audit all security-relevant functions of the OS.
- Audit the causing user, calling process, and specific error messages for any logged events.

1.3.3.2 Cryptographic Support (FCS)

The TOE includes the INTEGRITY Crypto Library v1.0 (ICL) running on SWIC Operating System 1.0 on INTEGRITY-IoT-2020.24 RTOS Microkernel. Functions implemented with ICL are in service of all cryptographic functionality required by the SFRs. The TOE supports the following cryptographic functions:

SFR	Cryptographic Algorithm	Operating Env.	Modes & Key Sizes	CAVP
FCS_CKM.1	*ECC KeyGen in accordance with FIPS 186-4 Appendix B.4*	SWIC Operating System 1.0 on INTEGRITY-IoT-2020.24 RTOS Microkernel	NIST Curves P-256, P-384, P-521	C1871
FCS_CKM.2	*Elliptic Curve key establishment in accordance with NIST SP 800-56A*	SWIC Operating System 1.0 on INTEGRITY-IoT-2020.24 RTOS Microkernel	NIST Curves P-384	C1871
FCS_COP.1(1)	*AES-XTS in accordance with NIST SP 800-38E*	SWIC Operating System 1.0 on INTEGRITY-IoT-2020.24 RTOS Microkernel	256-bit	C1871
	AES-GCM in accordance with NIST SP 800-38D	SWIC Operating System 1.0 on INTEGRITY-IoT-2020.24 RTOS Microkernel	256-bit	C1871
FCS_COP.1(2)	*SHA-1, SHA-256, SHA-384, SHA-512 in accordance with FIPS Pub 180-4*	SWIC Operating System 1.0 on INTEGRITY-IoT-2020.24 RTOS Microkernel	160 bits for SHA-1, 256 bits for SHA-256; 384 bits for SHA-384; 512 bits for SHA-512	C1871
FCS_COP.1(3)	*ECDSA SigGen and SigVer in accordance with FIPS Pub 186-4 Section 5*	SWIC Operating System 1.0 on INTEGRITY-IoT-2020.24 RTOS Microkernel	NIST curve P-384	C1871
	RSA SigGen and SigVer in accordance with FIPS Pub 186-4	SWIC Operating System 1.0 on INTEGRITY-IoT-2020.24 RTOS Microkernel	2048-bit	C1871
FCS_COP.1(4)	*SHA-1, SHA-256, SHA-384, SHA-512 in accordance with FIPS Pub 198-1 and FIPS Pub 180-4*	SWIC Operating System 1.0 on INTEGRITY-IoT-2020.24 RTOS Microkernel	For SHA-256, a 256-bit key size and message size. For SHA-384, a 384-bit key size and message size. For SHA-512, a 512-bit key size and message size.	C1871
FCS_RBG_EXT.1	*HMAC_DRBG in accordance with NIST SP 800-90A*	SWIC Operating System 1.0 on INTEGRITY-IoT-2020.24 RTOS Microkernel	Random number generation for all cryptography	C1871

Table 3 Cryptographic Parameters

1.3.3.3 User Data Protection (FDP)

All data on the disk, including the OS files and all user data, are automatically encrypted. This includes all Protection Profile-defined Sensitive Data, including:

- User application private keys, secrets, and key material.
- User account login information.

- Certificates and keys used for trusted path establishment, trusted channel establishment, and trusted update verification.

1.3.3.4 Identification and Authentication (FIA)

The TOE implements user identification and authentication, including authentication failure limiting, at all administrative interfaces. No more than three consecutive unsuccessful authentication attempts on any given power cycle. The TOE requires that the administrator successfully authenticate prior to performing any management or configuration functions.

The TOE supports the use of X.509v3 certificates, including revocation and validity checking. The administrator may choose which certificate is used for any given trusted path or trusted channel.

1.3.3.5 Security Management (FMT)

The TOE permits authorized and authenticated administrators to perform the following management functions:

- Set the inactivity timeout.
- Configure trusted paths and channels.
- Configure the networking parameters.
- Configure automatic updates.
- Management of user accounts.

1.3.3.6 Protection of the TSF (FPT)

The TOE implements protection of the kernel, audit logs and functions, and credential repositories. The TOE implements Address Space Layout Randomization and Stack-Based Buffer Overflow protection. The TOE performs self-tests of the cryptographic functions prior to operation and implements security checking prior to installing updates.

1.3.3.7 Trusted Paths and Channels (FTP)

The TOE provides a TLS trusted communication path to both administrators and trusted IT entities that protects the channel data from modification or compromise.

1.4 Excluded Functionality

The following interfaces are not included as part of the evaluated configuration:

Functions	Exclusion discussion
None	All TOE functions are evaluated as part of the Common Criteria Evaluation.

Table 4 Excluded Functionality

1.5 TOE Documentation

The following documents are available in PDF format from the NIAP website:

Documentation	Version	Date
INTEGRITY Enterprise OS – Archon Edition Security Target	1.2	4/26/22
INTEGRITY Enterprise OS – Archon Edition Common Criteria Administrative Guidance	1.2	4/27/22

Table 5 TOE Documentation

2 Conformance Claims

2.1 CC Conformance

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017: Part 3 extended

2.2 Protection Profile Conformance

This TOE is conformant to:

- Protection Profile for General Purpose Operating Systems, version 4.2.1, dated 2019-04-22 [OSPP].

2.3 Conformance Rationale

This Security Target provides exact conformance to [OSPP]. The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile performing only operations defined there.

2.3.1 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date that are applicable to [OSPP] have been addressed. The following table identifies all applicable TDs:

Identifier	Applicable	Exclusion Rationale
TD0600: Conformance claim sections updated to allow for MOD_VPNC_V2.3	No	Conformance to MOD_VPNC_V2.3 is not claimed in this ST
TD0578 – SHA-1 is no longer mandatory	Yes	
TD0525 – Updates to Certificate Revocation (FIA_X509_EXT.1)	Yes	
TD0501 - Cryptographic selections and updates for OS PP	Yes	
TD0496 – GPOS PP adds allow-with statement for VPN Client v2.1	No	VPN Client functionality is not claimed in this ST
TD0493 – X.509v3 Certificates when using digital signatures for Boot Integrity	No	X.509 certificates are not used with digital signatures for boot integrity
TD0463 – Clarification for FPT_TUD_EXT	Yes	
TD0441 – Updated TLS Ciphersuites for OS PP	Yes	
TD0386 – Platform-Provided Verification of Update	Yes	
TD0365 – FCS_CKM_EXT.4 Selections	Yes	

Table 6 NIAP Technical Decisions

3 Security Problem Definition

The security problem definition has been taken from [OSPP] and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any organizational security policies that the TOE is expected to enforce.

3.1 Threats

The following threats are drawn directly from the [OSPP]:

ID	Threat
T.NETWORK_ATTACK	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with applications and services running on or part of the OS with the intent of compromise. Engagement may consist of altering existing legitimate communications.
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 applications and services that are running on or part of the OS.
T.LOCAL_ATTACK	An attacker may compromise applications running on the OS. The compromised application may provide maliciously formatted input to the OS through a variety of channels including unprivileged system calls and messaging via the file system.
T.LIMITED_PHYSICAL_ACCESS	An attacker may attempt to access data on the OS while having a limited amount of time with the physical device.

Table 7 Threats

3.2 Assumptions

The following assumptions are drawn directly from the [OSPP]:

ID	Assumption
A.PLATFORM	The OS relies upon a trustworthy computing platform for its execution. This underlying platform is out of scope of this PP.
A.PROPER_USER	The user of the OS is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy. At the same time, malicious software could act as the user, so requirements which confine malicious subjects are still in scope.
A.PROPER_ADMIN	The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

Table 8 Assumptions

3.3 Organizational Security Policies

No explicit organizational security policies are defined by [OSPP].

4 Security Objectives

The security objectives for the TOE have been taken from [OSPP] and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the TOE

The following security objectives for the TOE describe how the TOE will respond to the THREATS, based on the ASSUMPTIONS.

ID	Objective for the TOE
O.ACCOUNTABILITY	Conformant OSes ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the operating system and discover its cause. Gathering event information and immediately transmitting it to another system can also enable incident response in the event of system compromise.
O.INTEGRITY	Conformant OSes ensure the integrity of their update packages. OSes are seldom if ever shipped without errors, and the ability to deploy patches and updates with integrity is critical to enterprise network security. Conformant OSes provide execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems.
O.MANAGEMENT	To facilitate management by users and the enterprise, conformant OSes 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 and application execution control.
O.PROTECTED_STORAGE	To address the issue of loss of confidentiality of credentials in the event of loss of physical control of the storage medium, conformant OSes provide data-at-rest protection for credentials. Conformant OSes also provide access controls which allow users to keep their files private from other users of the same system.
O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant OSes provide mechanisms to create trusted channels for CSP and sensitive data. Both CSP and sensitive data should not be exposed outside of the platform.

Table 9 Objectives for the TOE

4.2 Security Objectives for the Operational Environment

The following security objectives for the operational environment assist the TOE in correctly providing its security functionality. These track with the assumptions about the environment.

ID	Objectives for the Operational Environment
OE.PLATFORM	The OS relies on being installed on trusted hardware.
OE.PROPER_USER	The user of the OS is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy. Standard user accounts are provisioned in accordance with the least privilege model. Users requiring higher levels of access should have a separate account dedicated for that use.

ID	Objectives for the Operational Environment
OE.PROPER_ADMIN	The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

Table 10 Objectives for the Operational Environment

5 Security Requirements

This section identifies the Security Functional Requirements for the TOE. 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, dated: April 2017 and all international interpretations.

5.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with [*bracketed italicized*] text; Assignments within selections are indicated with [*bracketed italicized underlined*] text.
- Refinement: additions or modifications are indicated with **bold** text; removals are indicated with ~~strikethrough~~ text.
- Selection: Indicated with [bracketed underlined] text; Selections within selections are indicated with [bracketed multiple underlines]. Selections within selections within selections are indicated with [bracketed dash-dot underlines]. Selections within selections within selections within selections are indicated with [bracketed dash-dot-dot underlines].
- Iteration: Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3); and/or using one of : "/OS" identifier to indicate the PP or EP from which the SFR was taken.
- **Where operations were completed in the PP itself, the formatting used in the PP has been retained.**

Extended SFRs are identified by having a label 'EXT' after the requirement name. Formatting conventions outside of operations matches the formatting specified within the PP or EP.

5.2 Security Functional Requirements

5.2.1 Audit Data Generation (FAU)

5.2.1.1 FAU_GEN.1 Audit Data Generation (Refined)

FAU_GEN.1.1 The **OS** shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;
b) All auditable events for the [*not specified*] level of audit; and
c) [
   - *Authentication events (Success/Failure);*
   - *Use of privileged/special rights events (Successful and unsuccessful security, audit, and configuration changes);*
   - *Privilege or role escalation events (Success/Failure);*
   - *[*
     - *System reboot, restart, and shutdown events (Success/Failure),*

   ]

].

FAU_GEN.1.2 The **OS** shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*no other audit relevant information*]

.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_CKM.1 Cryptographic Key Generation (Refined)

FCS_CKM.1.1 The **OS** shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- ***ECC schemes using "NIST curves" P-256, P-384 and [P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 ,***

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards]~~.

5.2.2.2 FCS_CKM.2 Cryptographic Key Establishment (Refined)

FCS_CKM.2.1 The OS 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 Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"****,***

] ~~that meets the following: [assignment: list of standards]~~ .

5.2.2.3 FCS_CKM_EXT.4 Cryptographic Key Destruction

FCS_CKM_EXT.4.1 The OS shall destroy cryptographic keys and key material in accordance with a specified cryptographic key destruction method [

- *For volatile memory, the destruction shall be executed by a [*
  - *removal of power to the memory,*

  ],

- *For non-volatile memory that consists of [*
  - *destruction of all key encrypting keys protecting the target key according to FCS_CKM_EXT.4.1, where none of the KEKs protecting the target key are derived*
  - *the invocation of an interface provided by the underlying platform that [*
    - *logically addresses the storage location of the key and performs a [single] overwrite consisting of [zeroes, a new value of a key of the same size],*

  ]

] .]

FCS_CKM_EXT.4.2 The OS shall destroy all keys and key material when no longer needed.

5.2.2.4 FCS_COP.1(1) Cryptographic Operation - Encryption/Decryption (Refined)

FCS_COP.1.1(1) The **OS** shall perform [*encryption/decryption services for data*] in accordance with a specified cryptographic algorithm [

- ***AES-XTS (as defined in NIST SP 800-38E) ,***

] **and [**

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

] and cryptographic key sizes [128-bit, 256-bit] ~~that meet the following: [assignment: list of standards]~~.

5.2.2.5 FCS_COP.1(2) Cryptographic Operation - Hashing (Refined)

FCS_COP.1.1(2) The **OS** shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [

- *SHA-256,*
- *SHA-384,*
- *SHA-512,*

]] **and message digest sizes [**

- ***256 bits,***
- ***384 bits,***
- ***512 bits,***

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

5.2.2.6 FCS_COP.1(3) Cryptographic Operation - Signing (Refined)

FCS_COP.1.1(3) The **OS** 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 4,***
- ***ECDSA schemes using "NIST curves" P-256, P-384 and [no other curves] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5***

] ~~and cryptographic key sizes [assignment: cryptographic algorithm] that meet the following: [assignment: list of standards]~~.

5.2.2.7 FCS_COP.1(4) Cryptographic Operation - Keyed-Hash Message Authentication (Refined)

FCS_COP.1.1(4) The **OS** shall perform [*keyed-hash message authentication services*] in accordance with a specified cryptographic algorithm [***SHA-256, SHA-384, SHA-512***] **with key sizes [256 bits, 384 bits, 512 bits] and message digest** sizes [*256 bits, 384 bits, 512 bits*] that meet the following: [*FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard*].

5.2.2.8 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1 The OS shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [

- *HMAC_DRBG (any),*

] .

FCS_RBG_EXT.1.2 The deterministic RBG used by the OS shall be seeded by an entropy source that accumulates entropy from a [

- *platform-based noise source*

] with a minimum of [

- *256 bits*

] 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.

5.2.2.9 FCS_STO_EXT.1 Storage of Sensitive Data

FCS_STO_EXT.1.1 The OS shall implement functionality to encrypt sensitive data stored in non-volatile storage and provide interfaces to applications to invoke this functionality.

5.2.2.10 FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1 The OS shall implement TLS 1.2 (RFC 5246) supporting the following cipher suites: [

- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,*

].

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

FCS_TLSC_EXT.1.3 The OS shall only establish a trusted channel if the peer certificate is valid.

5.2.2.11 FCS_TLSC_EXT.2 TLS Client Protocol (SELECTION BASED)

FCS_TLSC_EXT.2.1 The OS shall present the Supported Groups Extension in the Client Hello with the following supported groups: [*secp384r1*].

5.2.2.12 FCS_TLSC_EXT.4 TLS Client Protocol (OPTIONAL)

FCS_TLSC_EXT.4.1 The OS shall support mutual authentication using X.509v3 certificates.

5.2.3 User Data Protection (FDP)

5.2.3.1 FDP_ACF_EXT.1 Access Controls for Protecting User Data

FDP_ACF_EXT.1.1 The OS shall implement access controls which can prohibit unprivileged users from accessing files and directories owned by other users.

5.2.3.2 FDP_IFC_EXT.1 Information flow control (OPTIONAL)

FDP_IFC_EXT.1.1 The OS shall [

- *provide an interface which allows a VPN client to protect all IP traffic using IPsec,*

] with the exception of IP traffic required to establish the VPN connection and [*no other traffic*].

5.2.4 Identification and Authentication (FIA)

5.2.4.1 FIA_AFL.1 Authentication failure handling (Refined)

FIA_AFL.1.1 The **OS** shall detect when [

- *[3],*

] unsuccessful authentication attempts occur related to **events with** [

- ***authentication based on username and password,***

] .

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts for an account has been **met**, the **OS** shall: [***[shut down]***].

5.2.4.2 FIA_UAU.5 Multiple Authentication Mechanisms (Refined)

FIA_UAU.5.1 The **OS** shall provide the following authentication mechanisms [

- ***authentication based on username and password,***

] to support user authentication.

FIA_UAU.5.2 The **OS** shall authenticate any user's claimed identity according to the [*constant time matching of usernames and SHA-384 constant time matching of password hashes stored in the underlying filesytem*].

5.2.4.3 FIA_X509_EXT.1 X.509 Certificate Validation[1]

FIA_X509_EXT.1.1 The OS 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 OS shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field
- The OS shall validate the revocation status of the certificate using [OCSP as specified in RFC 6960, CRL as specified in RFC 5759]
- The OS shall validate the extendedKeyUsage 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 extendedKeyUsage 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.
    - (conditional) 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.

[1] This SFR has been modified by TD0525

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

5.2.4.4 FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The OS shall use X.509v3 certificates as defined by RFC 5280 to support authentication for TLS and [no other protocols] connections.

5.2.5 Security Management (FMT)

5.2.5.1 FMT_MOF_EXT.1 Management of security functions behavior

FMT_MOF_EXT.1.1 The OS shall restrict the ability to perform the function indicated in the "Administrator" column in FMT_SMF_EXT.1.1 to the administrator.

5.2.5.2 FMT_SMF_EXT.1 Specification of Management Functions

FMT_SMF_EXT.1.1 The OS shall be capable of performing the following management functions:

Management Function	**Administrator**	**User**
Enable/disable [session timeout]	X	-
Configure [session] inactivity timeout	X	-
Configure local audit storage capacity	-	-
Configure minimum password length	-	-
Configure minimum number of special characters in password	-	-
Configure minimum number of numeric characters in password	-	-
Configure minimum number of uppercase characters in password	-	-
Configure minimum number of lowercase characters in password	-	-
Configure lockout policy for unsuccessful authentication attempts through [timeouts between attempts]	X	-
Configure host-based firewall	-	-
Configure name/address of directory server with which to bind	-	-
Configure name/address of remote management server from which to receive management settings	X	-
Configure name/address of audit/logging server to which to send audit/logging records	X	-
Configure audit rules	-	-
Configure name/address of network time server	-	-
Enable/disable automatic software update	X	-
Configure WiFi interface	-	-
Enable/disable Bluetooth interface	-	-
Enable/disable [*no other external interfaces*]	-	-
[*Create or Modify user accounts*	X	-
Change the password of the currently-authenticated user	X	X
Configure network interface settings	X	X
Set system Time-Zone	X	X
Perform Factory Reset]	X	X

Table 11 Management Functions

5.2.6 Protection of the TSF (FPT)

5.2.6.1 FPT_ACF_EXT.1 Access controls

FPT_ACF_EXT.1.1 The OS shall implement access controls which prohibit unprivileged users from modifying:

- Kernel and its drivers/modules
- Security audit logs
- Shared libraries
- System executables
- System configuration files
- [*no other objects*]

FPT_ACF_EXT.1.2 The OS shall implement access controls which prohibit unprivileged users from reading:

- Security audit logs
- System-wide credential repositories
- [*no other objects*].

5.2.6.2 FPT_ASLR_EXT.1 Address Space Layout Randomization

FPT_ASLR_EXT.1.1 The OS shall always randomize process address space memory locations with [8] bits of entropy except for [*no explicit exceptions*].

5.2.6.3 FPT_SBOP_EXT.1 Stack Buffer Overflow Protection

FPT_SBOP_EXT.1.1 The OS shall [employ stack-based buffer overflow protections].

5.2.6.4 FPT_TST_EXT.1 Boot Integrity

FPT_TST_EXT.1.1 The OS shall verify the integrity of the bootchain up through the OS kernel and [

- no other executable code

] prior to its execution through the use of [

- a hardware-protected hash

] .

5.2.6.5 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1[2] The OS shall provide the ability to check for updates to the OS software itself and shall use a digital signature scheme specified in FCS_COP.1(3) to validate the authenticity of the response.

FPT_TUD_EXT.1.2 The OS shall [cryptographically verify] updates to itself using a digital signature prior to installation using schemes specified in FCS_COP.1(3).

[2] This SFR has been modified by TD0463

5.2.6.6 FPT_TUD_EXT.2 Trusted Update for Application Software

FPT_TUD_EXT.2.1 [3]The OS shall provide the ability to check for updates to application software and shall use a digital signature scheme specified in FCS_COP.1(3) to validate the authenticity of the response.

FPT_TUD_EXT.2.2 The OS shall cryptographically verify the integrity of updates to applications using a digital signature specified by FCS_COP.1(3) prior to installation.

5.2.7 TOE Access (FTA)

5.2.7.1 FTA_TAB.1 Default TOE access banners (OPTIONAL)

FTA_TAB.1.1 Before establishing a user session, the **OS** shall display an advisory warning message regarding unauthorized use of the OS.

5.2.8 Trusted Paths and Channels (FTP)

5.2.8.1 FTP_ITC_EXT.1 Trusted channel communication

FTP_ITC_EXT.1.1 The OS shall use [

- TLS as conforming to FCS_TLSC_EXT.1,

] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: [*administration server*] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

5.2.8.2 FTP_TRP.1 Trusted Path

FTP_TRP.1.1 The **OS** shall provide a communication path between itself and [*local*] users that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from [modification, disclosure].

FTP_TRP.1.2 The **OS** shall permit [*local users*] to initiate communication via the trusted path.

FTP_TRP.1.3 The **OS** shall require use of the trusted path for [[*all remote administrative actions*]].

5.3 TOE SFR Dependencies Rationale for SFRs

[OSPP] contain all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PPs have been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from [OSPP] which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the table below.

Assurance Class	**Components**	**Components Description**
Development (ADV)	ADV_FSP.1	Basic Functional Specification
Guidance Documents (AGD)	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative Procedures
Life cycle Support (ALC)	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage
	ALC_TSU_EXT.1	Timely Security Updates

[3] This SFR has been modified by TD0463

Assurance Class	Components	Components Description
Security Target evaluation (ASE)	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_TSS.1	TOE summary specification
Tests (ATE)	ATE_IND.1	Independent Testing – Conformance
Vulnerability Assessment (AVA)	AVA_VAN.1	Vulnerability Assessment

Table 12 Security Assurance Requirements

5.5 Rationale for Security Requirements

A mapping of the Security Functional Requirements to the Security Objectives for the TOE can be found in Section 4.1 of the [OSPP].

The Security Assurance Requirements were chosen because they are required by the [OSPP], and the ST claims exact conformance to the [OSPP].

5.6 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Vendor to satisfy the assurance requirements. The table below lists the details.

Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1 ALC_CMS.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ALC_TSU_EXT.1	The vendor will provide timely security updates to the TOE, and the ST describes how updates are made available to the TOE & TOE administrator.
ATE_IND.1	Vendor will provide the TOE for testing.

Component	How the SAR will be met
AVA_VAN.1	Vendor will provide the TOE for testing.

Table 13 TOE Security Assurance Measures

6 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td>FAU_GEN.1</td><td>
<p>Auditing on the TOE is managed by an application-layer OS function. Application-layer functions and virtual machines on the TOE report auditable events to this application. Received events are written to the local Audit MFI entry. When instructed by an authorized administration server over its trusted channel, the TOE establishes a trusted channel to the designated audit server and uploads its current audit log to the audit server via this channel. The audit server is then capable of decoding the log into a human-readable text file.</p>
<p>The list of auditable events are as follows:</p>
<table>
<tr><th>Event Name</th><th>Process/VAS</th><th>User</th><th>Description</th></tr>
<tr><td>Reboot</td><td>Audit</td><td>System</td><td>Indicates boot of TOE and start of audit services.</td></tr>
<tr><td>Start Shutdown</td><td>Power</td><td>System</td><td>Indicates start of system shutdown. UserData indicates whether the system will reboot or power off.</td></tr>
<tr><td>AuditShutdown</td><td>Audit</td><td>System</td><td>Indicates the audit function is stopping due to the system being shut down.</td></tr>
<tr><td>ChangeBootPassword</td><td>Admin</td><td>Primary User</td><td>Indicates an attempt to change the system boot password. UserData indicates whether the attempt succeeded or failed.</td></tr>
<tr><td>Login</td><td>AuditProxy or GPOS</td><td>Indicated User</td><td>Indicates an attempt to login to the system shell. UserData indicates the user in question and whether the attempt succeeded or failed.</td></tr>
<tr><td>Logout</td><td>AuditProxy or GPOS</td><td>System</td><td>Indicates the logout of a user from the system shell. UserData indicates the user in question.</td></tr>
<tr><td>UserAdd</td><td>AuditProxy or GPOS</td><td>Indicated User</td><td>Indicates an attempt to create a new user account. UserData indicates the username of the new account, the user attempting to create the account, and whether the attempt succeeded or failed.</td></tr>
<tr><td>UserDel</td><td>AuditProxy or GPOS</td><td>Indicated User</td><td>Indicates an attempt to delete a user account. UserData indicates the username of the account, the user attempting to delete the account, and whether the attempt succeeded or failed.</td></tr>
<tr><td>Passwd</td><td>AuditProxy or GPOS</td><td>Indicated User</td><td>Indicates change of account password by the account owner. UserData indicates the username of the account.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>SystemUpdate</td><td>AuditProxy</td><td>OTA or OTA_TLS</td><td>Indicates an attempt to update the system software or configuration. UserData indicates the availability of an update and the progress of installing an update.</td></tr>
<tr><td>Upload Audit Logs</td><td>AuditProxy</td><td>OTA</td><td>Indicates an attempt to upload the audit logs to the administration server.</td></tr>
<tr><td>Trusted Channel</td><td>AuditProxy</td><td>OTA_TLS or VM names</td><td>Indicates status of administration server trusted channel. UserData indicates the current status, including attempts to establish the channel, successful establishment of the channel, or failure to establish the channel.</td></tr>
</table>
Each event is timestamped based on when the event is logged. Each event contains an ID indicating the type of the event and a data payload containing additional information. The process that originated the event is predefined based on the event ID. The identity of the user that caused the event is either contained within the event’s data payload or is predefined based on the event ID. If the action corresponding to the event has a success or failure outcome, this outcome is conveyed in the event’s data payload.

The process is specified as “AuditProxy” when the OS performs a service on behalf of a virtual AddressSpace (e.g. adding a user account), as opposed to an action being performed by the OS (e.g. start shutdown).
</td></tr>
<tr><td>FCS_CKM.1
FCS_CKM.2
FCS_CKM_EXT.4
FCS_COP.1.1(1)
FCS_COP.1.1(2)
FCS_COP.1.1(3)
FCS_COP.1.1(4)
FCS_RBG_EXT.1</td><td>The TOE implements a cryptographic library. The library is the INTEGRITY Crypto Library, which provides cryptographic services for the INTEGRITY Enterprise OS – Archon Edition and runs in the SWIC Operating System 1.0 on INTEGRITY-IoT-2020.24 RTOS microkernel on Intel Core i5-8365U (Whiskey Lake) operational environment which part of the TOE.

No persistent asymmetric keys are generated on the TOE. All persistent public and private keys used with X.509v3 certificates (as part of TLS trusted channels to other IT devices) are imported onto the TOE by the administrator during provisioning or update of the TOE. Where ECDSA or RSA is used for a TLS connection, the temporary public and private keys used by this algorithm are generated in accordance with FIPS PUB 186-4. The TOE implements ECC key generation as specified in FIPS 186-4. ECC curves P-256, P-384, and P-521 are supported

The TOE performs Elliptic curve-based key establishment for each trusted channel that uses TLS.

All keys or key materials that are loaded from disk into volatile memory are destroyed upon removal of power to the memory (i.e., upon system shutdown or reboot). This consists of:
• All keys loaded or generated during TLS connections:
o For TLS, the client private key loaded from the MFI entry, the symmetric AES session key, and any asymmetric keys listed by FCS_CKM.2.

The following keys or key materials are stored in non-volatile memory and are managed as specified:
• The client private keys for TLS are stored in on-disk MFI entries. Each entry is encrypted with a DEK using AES-XTS. Each entry’s DEK is stored in the encrypted MFI. A client private key is destroyed upon the destruction of its MFI entry DEK as described previously (making the MFI entry containing the client private key impossible to decrypt).</td></tr>
</table>

Requirement	TSS Description
	No configurations or circumstances would cause the TOE to not conform to this key destruction requirement. The TOE uses AES with the following modes and key sizes: • XTS with 256-bit keys for local encryption • GCM with 128-bit and 256-bit keys for TLS encryption. The TOE uses ECDSA with SHA-256, SHA-384, and SHA-512 and RSA with 2048 bit keys for certificate signature checking during TLS. The TOE uses HMAC-SHA-256 and HMAC-SHA-384 for message authentication during TLS. The TOE uses SHA-512 to derive the MasterKey from the TPMSalt and the system boot password. The TOE uses SHA-384 in password hashing, and in validation of the update packages. Each entry in the MFI contains a SHA-384 hash of the entry’s contents. Before loading the INTEGRITY Enterprise OS – Archon Edition monolith, the TOE checks that the SHA-384 hash of the monolith matches the value stored in the MFI. This ensures that the monolith has not been modified. During provisioning or update, the TOE performs two checks. First, the TOE verifies the digital signature on the update candidate using ECDSA SigVer with a key size of 384 bits over the P-384 curve, with a hash size of 384 bits. If the signature is valid, the TOE decrypts the update candidate and then checks the integrity of all downloaded files by comparing the SHA-384 hash of the downloaded file against the value stored in its header. This protects against the modification of update files and provides assurance that the update is legitimate and from the vendor. The TOE uses HMAC-SHA-512 in the HMAC_DRBG. The TOE uses a HMAC_DBRG(SHA-512) to generate pseudorandom bits for cryptographic operations. The DRBG is seeded with 1024-bits from the Intel RDSEED third-party entropy source and 512 bits from the hardware TPM (used only as a mixing value). The 1024-bits of data from RDSEED are assumed to have at least 512-bits of entropy.
FCS_STO_EXT.1	The TOE may contain the following “sensitive data”: • User application private keys • User login information • The administration/update/audit client private key. All of the above “sensitive data” are automatically encrypted with AES-XTS with 256-bit keys. No user intervention is required to protect them. Each item of the above “sensitive data” may only be read and used by appropriate built-in system applications. The items CANNOT be read by the user through any interface on the TOE.
FCS_TLSC_EXT.1	For the administration server trusted channel, TLS is implemented by a TOE Application using the INTEGRITY Crypto Library (ICL). Only TLS 1.2 is supported. The secp384r1 elliptic curve is supported for communications with the administration server. Only the following cipher suites are supported: • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 Administrators may configure the TLS client’s expected reference identifier by setting the FQDN of the destination server. For example, “audit.example.com”. The TOE supports only DNS name / SAN-DNS, CommonName, and FQDN. IP addresses are not supported. When establishing a TLS trusted channel to an administration server, the TOE checks the server’s identity according to the rules specified in RFC 2818 section 3.1:

Requirement	TSS Description
	• If the server certificate contains a subjectAltName extension, the TOE checks for a SAN-DNS that matches the expected DNS name of the server. If none are found the TLS connection fails. • If the server certificate does NOT contain a subjectAltName extension, the TOE checks if the server certificate's Common Name matches the expected DNS name of the server. If it does not the TLS connection fails. In matching DNS names, the TOE follows the rules specified in RFC 6125. Specifically, wildcards may only be used in the leftmost label of a domain name. "Certificate pinning" is not supported. The TOE supports the use of OCSP and CRLs and queries the OCSP/CRL server specified in the presented certificate. If no OCSP/CRL server is provided, the TOE takes the action defined in FIA_X509_EXT.2. For all trusted channels using TLS, the TLS implementation on detection that the peer certificate is not valid, sends an appropriate Alert message to the server as defined in RFC 5246 and terminates the TLS connection. As the TLS connection is not successfully established, the trusted channel is not established. The TOE supports the secp384r1 group extension. This is by default; no configuration is available.
FCS_TLSC_EXT.2	The TOE only supports the Supported Groups Extension containing the group secp384r1. This behavior is performed by default.
FCS_TLSC_EXT.4	For all TLS, the TOE presents its client certificate associated with the connection to the remote server. The administrator chooses which certificate the TOE will use for each connection during provisioning or update of the TOE.
FDP_ACF_EXT.1	The TOE filesystem's root directory is owned by the "root" user. The root directory contains a "home" directory for each user whose name matches their username and is owned by the user. The root directory also contains a special "sys" directory containing system files and is owned by the "root" user. After fully booting the TOE, a user may login to the system shell with their username and password. Regular users can only list files and directories that they own, create or delete files inside a directory that they own, and read from or write to files that they own. The "root" user may list, create, delete, read, and write any file regardless of ownership.
FDP_IFC_EXT.1	The TOE does not provide a VPN client directly. However, the TOE provides common API interfaces through which the user may install a VPN client of their choice. When the VPN client is enabled, the client will configure the rules under which traffic is handled (i.e., to or from certain addresses, or of a certain traffic type, or all traffic, etc.). The configuration of the IPsec SAs is outside the control of the OS other than provide interfaces to the network stack to implement the specific routing requirements of the client.
FIA_AFL.1	After boot of the TOE, the user may login to the system shell with their username and password. Entering an incorrect username and password combination increments the AFL counter. After three unsuccessful attempts, the TOE forcibly powers down. The AFL counter is reset to zero only upon full power off of the TOE.
FIA_UAU.5	On system boot, the bootloader prompts the user for the system boot password. This password must be between 0 and 64 characters long and may only contain the UTF-8 characters with codes between 32 (space) and 126 (~). This password is used in combination with the TPMSalt to derive the 512-bit MasterKey (consisting of the 256-bit MfiKey and OpalKey). Without the correct boot password being entered the TOE CANNOT decrypt the MFI (Master File Index) and proceed with booting the INTEGRITY Enterprise OS – Archon Edition. Entering an incorrect boot password three times in a row causes the TOE to fully power off. Powering off the TOE resets the AFL counter.

Requirement	TSS Description
	After fully booting the TOE, the user may login to the system shell with their username and password. The login information for each account is stored in an on-disk encrypted MFI entry. This information includes a plaintext username, a random 32-byte salt value generated during creation of the account, and the PBKDF2-HMAC-SHA384 hash of the account's password and salt. After a user enters their username and password into the system shell, the INTEGRITY Enterprise OS – Archon Edition finds the requested account if it exists and generates the hash from the entered password. If the account exists and the password hashes match, the user is successfully logged in. A logged in user may change their account password from the system shell using the "passwd" command. This password must be between 1 and 64 characters long and may only contain the UTF-8 characters with codes between 32 (space) and 126 (~). If the account exists and the provided password is correct the TOE grants access to the system shell with that account. Otherwise, the TOE denies all access to the system shell.
FIA_X509_EXT.1	The TOE implements TLS functionality in an INTEGRITY Application using the INTEGRITY Crypto Library (ICL). TLS X.509 certificates are validated according to the algorithm specified in RFC 5280 Section 6 et seq. The certificates of the Trust Anchor Database are chosen by the administrator during provisioning or automatic update of the TOE. Verification (including revocation checking) of a certificate occurs when the certificate is first loaded into the TOE, when the certificate is loaded for first-use, and when a peer certificate is presented during an authentication step. Certificate revocation "checking of a server certificate (including it's complete CA chain) is done via either the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, or the checking of Certificate Revocation Lists (CRL) as specified in RFC 5759. This occurs immediately after receipt of the server certificate and before key exchange. The implementation first queries the certificate's first four listed OCSP responders in order. If the OCSP responder does not assert that the certificate is valid, it is rejected. If the certificate does not specify any OCSP responders, the TOE falls-back to CRLs, and downloads and checks the CRLs from the certificate's first four listed CRL Distribution Points in order. If the CRL indicates that the certificate has been revoked, the certificate is rejected. If no CRL can be retrieved, the certificate is rejected. Furthermore, the implementation is configured to reject any peer certificates that meet any of the following conditions: • Any CA certificate in the certificate path is missing the basicConstraints extension or does not have the CA flag set to TRUE in this extension. • A server certificate does not have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field. • A certificate used to sign an OCSP response does not have the OCSP Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field. • A certificate used to sign a CRL does not have the CRL-signing bit set.
FIA_X509_EXT.2	X.509v3 certificates are used during TLS mutual authentication between the TOE and a remote server. This applies to the TLS trusted channel to an authorized administration server.
FMT_MOF_EXT.1	The following management functions are supported by the TOE and are restricted to administrators. These functions may be configured by the administrator during provisioning or automatic update of the TOE. Both activities can only be performed by an authorized administrator with physical access to the unprovisioned TOE for the former, or with administrative access to the authorized administration and/or logging server for the latter: • Set the session inactivity timeout (in minutes). Setting the timeout value to zero disables the session inactivity timeout. • Set the address of the administration server from which to receive software updates and management settings from. • Set the address(es) of network time server(s). • Enable/disable automatic software updates.

Requirement	TSS Description
	• Create a new user account and set its initial password. • Delete a user account.
FMT_SMF_EXT.1	The following management functions are supported by the TOE and are restricted to administrators. These functions may be configured by the administrator during provisioning or automatic update of the TOE. Both activities can only be performed by an authorized administrator with physical access to the unprovisioned TOE for the former, or with administrative access to the authorized administration server for the latter: • Set the session inactivity timeout (in minutes). Setting the timeout value to zero disables the session inactivity timeout. • Set the address of the administration server from which to receive software updates and management settings from. • Enable/disable automatic software updates. The following management functions are supported by the TOE and are restricted to administrators. These functions are only available via local access to the TOE: • Create a new user account and set its initial password. • Delete a user account. The following management functions are supported by the TOE and are available to the user. These functions are only available via local access to the TOE: • Change the password for the current user account. • Configure the IPv4, IPv6, DNS, and DHCP settings of the TSF. • Set the system Time Zone. • Factory reset the TOE.
FPT_ACF_EXT.1	All TOE executable code, security audit logs, INTEGRITY Applications, and system configuration files on the TOE are stored in the underlying file system, which are not accessible to non-administrative users. All audit logs and credential repositories on the TOE are stored in the underlying file system, which are not accessible to non-administrative users. Audit logs and credential repositories cannot be read by the user through any interface on the TOE.
FPT_ASLR_EXT.1	The TOE initializes each Integrity Application VAS (Virtual Address Space) with a random address offset value. This value contains 8 bits of entropy from the RDRAND platform-based noise source.
FPT_SBOP_EXT.1	INTEGRITY Applications on the TOE use “stack guards” for stack-based buffer overflow protection. On system boot, a random 64-bit guard value is generated using the RDRAND platform-based noise source. On function entrance, the callee pushes the guard value onto the stack. Right before exit from the function, the callee pops the guard value from the stack and compares it to its initial value. If it has changed, a buffer overflow is detected, and the application self-terminates. The TOE is developed and reviewed under the DO-178B Level A processes which include stack analysis. For reference, please refer to DO-178B section 6.3.4-f (Accuracy and consistency). The purpose of using this process is to maintain a high quality, well reviewed, simple code base for a microkernel which does not require the additional CPU and memory overhead of (among other things) a runtime stack guard. In addition, the TOE does not make use of dynamic memory allocation on the stack or on the heap. All use of recursion is provably bounded (verified by analysis and testing). Only the minimal core OS functionality is included in the kernel address space. All other functionality is implemented in protected virtual address spaces (which do include full stack protections).
FPT_TST_EXT.1	After power on, the hardware platform firmware loads the digitally signed SWIC bootloader. The digital signature of the signed bootloader is verified using ECDSA over NIST curve P-384. The SWIC bootloader then attempts to obtain the 256-bit TPMSalt from the hardware system

Requirement	TSS Description
	environment's TPM 2.0 module. This will only be successful if the TPM's PCRs (Platform Configuration Registers) have the correct values. The bootloader then prompts the user for the "system boot password." The bootloader generates the 512-bit MasterKey (consisting of the 256-bit MfiKey and 256-bit OpalKey) by combining the TPMSalt and boot password using PBKDF2-HMAC-SHA512. The bootloader then uses the MfiKey to decrypt the on-disk AES-GCM-256 encrypted MFI (Master File Index). The bootloader checks the integrity of the decrypted MFI by validating its GCM value. The entire INTEGRITY Enterprise OS – Archon Edition including the kernel is stored as an on-disk AES-XTS encrypted MFI entry. The bootloader decrypts the INTEGRITY Enterprise OS – Archon Edition using its 256-bit DEK stored in the MFI. The bootloader checks the integrity of the decrypted INTEGRITY Enterprise OS – Archon Edition by computing its SHA-384 hash and comparing this value to the hash stored in the MFI. After confirmation of the integrity of the INTEGRITY Enterprise OS – Archon Edition, the bootloader loads the INTEGRITY Enterprise OS – Archon Edition's INTEGRITY Separation Kernel. By this sequence of derivations and checks, the TOE confirms that the correct password has been entered and that none of the operational code has been modified in any way. Because critical information for this process is stored in the hardware-provided Trusted Platform Module, the TOE's firmware integrity is verified using a hash that is hardware protected.
FPT_TUD_EXT.1	After establishing the trusted channel to an authorized administration server, the TOE requests a system update from the server over this channel. In doing so, the TOE sends the server non-sensitive but identifying information from its MFI (Master File Index). The administration server may respond with a PCF (Provisioning Command File) directing the TOE to the URI locations of additional PCF or OTA (Over-the-air Update) files to download and update with. The TOE establishes a trusted channel with each required admin server and downloads the required PCF or OTA files over this channel. if a PCF or OTA file is encrypted (using AES-GCM-256) the decryption key is generated by taking a salted SHA-384 hash of the file's password. This password is provided by the PCF file which directed the TOE to install the encrypted PCF or OTA file. The TOE checks the integrity of all downloaded OTA and PCF files by comparing the SHA-384 hash of the downloaded file against the value stored in its header. This protects against the modification of update files. Each downloaded OTA file contains one or more MFI entry updates. When an MFI entry update is installed the existing MFI entry is completely overwritten (including the entry's DEK and hash). Before installing certain MFI entry updates, the TOE checks that the entry update has been digitally signed with the Vendor's ECDSA secp384r1 private key. The Vendor's public key used to verify this signature is installed as part of the SWIC bootloader during provisioning or automatic update of the TOE. If the new entry is not signed, the TOE will not install it. The list of MFI entries that require signature by the Vendor are as follows: • The MFI entry containing the SWIC bootloader (including the vendor public key). • The MFI entry containing the INTEGRITY Enterprise OS – Archon Edition. • The MFI entries containing an INTEGRITY Enterprise OS – Archon Edition Virtual Machine.
FPT_TUD_EXT.2	The TOE provides applications the ability to check for updates from within the TOE's own update channels. These applications are updated with and by the TOE's normal update process and in the same manner. The TOE provides applications the ability to check for updates from within the TOE's own update channels. These applications are updated with and by the TOE's normal update process and in the same manner.
FTA_TAB.1	During provisioning or automatic update of the TOE, the administrator may choose to select a background image that will appear onscreen whenever the user boots the TOE. The administrator

Requirement	TSS Description
	may choose to include an advisory warning message regarding unauthorized use of the TOE in this image.
FTP_ITC_EXT.1	The TOE can establish the following distinct trusted channels: • A trusted channel between the TOE and an authorized administration server. For the trusted channel between the TOE and an administration server, the server is authenticated through TLS mutual authentication with X.509 certificates. The confidentiality and integrity of this channel is secured through the TLS 1.2 protocol.
FTP_TRP.1	An authorized administrator may perform remote administrative actions of the TOE via the automatic update trusted path. An authorized administrator with administrative access to the TOE's configured administration server may perform the following remote administrative actions: • Upload and view the TOE's current audit records. On successful establishment of the trusted channel between the TOE and the administration server, the administration server may instruct the TOE to upload its current audit logs to a designated audit server. This communication is secured by the administration server and audit server trusted channels. • Create an update for the TOE that modifies one or more of the management parameters listed as "restricted to administrators" and "may be configured by the administrator during provisioning or automatic update of the TOE" in FMT_MOF_EXT.1/FMT_SMF_EXT.1.1. On successful establishment of the trusted channel between the TOE and the administration server, the TOE automatically requests a system update from the server via this channel (detailed in FPT_TUD_EXT.1). This communication is secured by the administration server trusted channel. The above actions cannot be performed by any other means. The TSF always initiates communication via the trusted path (TLS).
ALC_TSU.1	The vendor will address any reported vulnerabilities within 90 days of reporting. The following web site can be used to securely report security issues: https://archonidt.atlassian.net/servicedesk/customer/user/login?destination=portals. An account for the primary Point of Contact (PoC) for each customer is established when systems are first acquired and communicated to the PoC. Additional user accounts can be set up upon request from the PoC via the web site or by users contacting the vendor directly. This process is applicable for any component of the TOE.

Table 14 TOE Summary Specification

7 References

[CC_PART1]	Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-001
[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-002
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-003
[CEM]	Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated September 2012, version 3.1, Revision 5, CCMB-2017-04-004
[800-38A]	NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001
[800-56A]	NIST Special Publication 800-56A Rev 2, May 2013
[800-56B]	NIST Special Publication 800-56B Recommendation for Pair-Wise, August 2009
[800-38A]	[NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001
[800-38D]	NIST Special Publication 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.

Table 15 Annex A: References

--- End of Document ---