Security Target

Ziperase Drive Erasure Software v3.0.2

ST Version: 1.0

Date: July 17, 2024

Prepared for:

Ziperase Ltd.

Prepared by:

TERON LABS

www.teronlabs.com

Revision History

Version	Date	Author(s)	Description of Change
1.0	July 17, 2024	Teron Labs	Released version

Contents

List of Tables

References

[CCPart1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017

[CCPart2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017

[CCPart3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017

[STD-MAN] Standalone Erasure with Ziperase, Version 2.1.4: 2022-08-24

1 Security Target Introduction

This section is the introduction to the Security Target (ST) identifying and describing the Target of Evaluation (TOE).

The TOE is a drive erasure software by Ziperase Ltd. It is used for erasing disk drives in a secure manner according to a set of standard erasure algorithms. There are three possible configurations of the TOE: standalone configuration, network configuration and appliance configuration. In a standalone configuration, the TOE is booted into the target host from a specifically generated Universal Serial Bus (USB) drive. In the network configuration, the TOE is booted into the target host from a workstation across a network. In an appliance configuration, the TOE is booted from the TOE software uploaded to the host PC.

Once the TOE boots up, it bypasses the operating system of the host which is to be erased and allows the operator to select an erasure algorithm which is used for erasing the drive(s) of the target host. In the appliance configuration, where the software is installed on a host computer, the TOE may also be used for erasing attached drives through the USB bus or a Host Bus Adapter (HBA).

The Security Target Introduction consists of the identification of the ST and the TOE in Sect. 1.1 and of the TOE Overview given in Sect. 1.2.

1.1 ST and TOE Reference

The Security Target and the Target of Evaluation are identified as follows:

Security Target Title	Security Target Ziperase Drive Erasure Software v3.0.2
Security Target Version	1.0
Security Target Date	July 17, 2024
TOE Title	Ziperase Drive Erasure Software v3.0.2
TOE Software	Ziperase Drive Erasure Software v3.0.2 distributed as – ziperease-array-3.0.2.iso for the TOE in appliance configuration, – ziperase-command_center-3.0.2.iso for the TOE in network configuration, and – ziperease-core-3.0.2-x86_64.iso for the TOE in standalone configuration.
TOE Hardware	N/A
TOE Security Guidance	Common Criteria Guidance Supplement v1.0, Ziperase Drive Erasure Software v3.0.2

1.2 TOE Overview

TOE Overview summarizes the use and security functions of the TOE and states the physical and logical scope of the TOE.

The TOE Overview commences with an introduction of the use case for the TOE in Sect. 1.2.1. The is followed by the statement of the TOE Type in Sect. 1.2.2. The physical and logical scope of the TOE are defined in Sect. 1.2.3 and 1.2.4, respectively. The hardware, software and firmware items required by the TOE, but which are not parts of the TOE are identified in Sect. 1.2.5.

1.2.1 Usage and Major Security Features of the TOE

The TOE is a secure drive erasure tool developed by Ziperase Ltd.. It may be used in a standalone configuration, network configuration or appliance configuration. In each configuration, the TOE is used for booting up a target host which contains or is attached to a drive that needs to be erased in a secure manner.

Once the target host boots up with the TOE, the TOE software implements a minimalistic Graphical User Interface (GUI) which allows the operator to select the drive to be erased, the erasure algorithm as well as set other parameters for the erasure. Once the operator proceeds with the erasure, the TOE erases the drive of the target host in accordance with the selected algorithm and reports the findings to the operator.

In the appliance configuration, the TOE is executed on top of the operating system of the Host PC. In standalone and network configurations, the TOE includes a minimalistic operating system which bypasses the operating system of the host PC.

The Ziperase Drive Erasure Software v3.0.2 implements a rich set of erasure standards suitable for different drive types. The complete listing is available in Sect. 4.2 of [STD-MAN]. The exact same erasure algorithms are also available when the TOE is operated in each configuration. Many are legacy standards which may only be required for specific, rare use cases.

The erasure algorithms included in the certified configuration of the TOE are the following:

- Aperiodic Random Overwrite Method
- CESG CPA - Higher
- DoD 5220.22-M
- DoD 5220.22-ECE
- NIST 800-88 Clear
- NIST 800-88 Purge
- HMG Infosec Standard 5, Higher
- HMG Infosec Standard 5, Lower

The TOE interfaces with the drives through the Application Programming Interface (API) of the drive. Each standard for a drive defines an API for the low-level manipulation of the content of the drive. The firmware of the drive implements that API. The TOE implements the erasure standards and verifies the outcome of the erasure using the drive API.

Each TOE configuration executes similar software. In the standalone configuration, the TOE and the configuration files are stored on a USB drive from which the TOE boots up. In the appliance configuration, the TOE and the configuration files are stored on the Host PC. In the network configuration, the TOE and the configuration files are accessed from a Ziperase Command Center across a network. The network connection required for the network configuration may be any local area wireless network connection.

Network connectivity and the Ziperase Command Center are not parts of the TOE. When booting up from a USB token, the bootable USB token must be generated with the Ziperase Boot Media Creator v2.1.4 or newer. The Boot Media Creator is not part of the TOE. In appliance configuration the software does not include an operating system and is installed on a host computer. In the appliance configuration, the TOE may be used for erasing attached drives via the USB bus or HBA.

There are no hardware parts on the TOE.

The TOE software is Ziperase Drive Erasure Software v3.0.2. The TOE software is distributed as an ISO package file identified in Sect. 1.1. TOE software for standalone and network configurations includes the operating system, the GUI, the connectivity software and the implementation of all erasure algorithms and other software functions. TOE software for the appliance configuration does not include the operating system but is meant to execute on top of the operating system of the Host PC.

The TOE implements a license mechanism which ensures that drive erasure only occurs if the user has sufficient erasure licenses in place. In the standalone configuration, the licenses are stored on the same USB drive from which the TOE boots up and are accessed with the Boot Media Creator. In the network configuration, the licenses are stored on the same network drive with the bootable TOE and are accessed across the network connection. In the appliance configuration, the licenses are stored on the Host PC of the TOE.

The TOE appears in different representations throughout the life-cycle. The stages constituting the life-cycle of the TOE are described in Table 1.

Table 1 TOE Life-Cycle Stages

Stage	Description
Receipt	The user receives from the developer a link to a .iso file, a SHA-512 checksum of the .iso file, TOE security guidance, and instructions on how to verify the authenticity of the .iso file.
Download	The user downloads the .iso file and stores it on a trusted, local computer which may be connected to a network. The authenticity of the .iso file is verified using the SHA-512 checksum. After successful downloading and verification, the .iso file is handled in accordance with the security practices of the user.
Store on Bootable Media	The configured .iso file is stored on the bootable media if using the TOE in a standalone configuration, within the Command Center if using the TOE in a network configuration, or on the Host PC if using the TOE in appliance configuration. If the TOE is stored on a USB drive for use in standalone configuration,, the Ziperase Boot Media Creator tool must be used.
Boot to RAM	The Host PC is booted up. The TOE boots into the host device RAM bypassing (in standalone and network configurations) the local operating system and makes available to the user the GUI. In appliance configuration, the booting up of the Host PC launches the local operating system which in turn boots up the TOE.
Initialize	Once the host device boots up with the TOE, the GUI makes available to the user an initialization window which may be used for configuring the erasure characteristics of the TOE. Once the user completes the TOE initialization, the TOE may be operated to erase the selected drives.
Operate	After the TOE is initialized, the erasure functions may be used. If sufficient erasure licenses exist, the TOE may be used for erasing any drive connected to the Host PC. The TOE communicates with the USB drive (in standalone configuration), the network drive (in network configuration) or the Host PC (in standalone configuration) for verifying the availability of the erasure licenses.
Terminate	Upon shutting down the Host PC, the TOE is erased from the RAM in which it executes. The TOE becomes inoperative until the booted up again.

1.2.2 TOE Type

The TOE is a disk drive erasure software. It is a software TOE which requires a host device but includes no hardware platform or other hardware parts. The TOE contains an independent boot sequence routine, a Graphical User Interface, connectivity software and erasure software. In the standalone and network configurations, the TOE also includes a minimalistic operating system which bypasses the operating system of the Host PC.

The TOE is booted up in a target host and then used to securely erase one or more drives of the host. There are no readily available Protection Profiles (PP) for drive erasure devices. Therefore, the ST does not claim conformance to any PP and the TOE is not of any exact type defined in a PP.

1.2.3 Physical Scope of the TOE

The physical scope of the TOE consists of TOE Software and TOE Security Guidance. There are no hardware parts in the TOE. A summary of the items in the physical scope of the TOE is given in Table 2.

Table 2 Summary of the Physical Scope of the TOE

TOE Hardware	N/A
TOE Software	Ziperase Drive Erasure Software v3.0.2 distributed as an ISO file appropriate for the selected configuration as identified in Sect. 1.1.
TOE Security Guidance	Common Criteria Guidance Supplement v1.0, Ziperase Drive Erasure Software v3.0.2

TOE Software runs on the Host PC and implements the security functions of the TOE. It interacts with the drive API and other components of the operational environment.

TOE Security Guidance is delivered to all users of the TOE. The TOE must always be deployed and operated in accordance with it. TOE Security Guidance is a Common Criteria Guidance Supplement which extends the existing manuals and other product literature of the TOE and instructs the users of the TOE how to initialize and operate the TOE in the certified configuration.

1.2.4 Logical Scope of the TOE

The logical scope of the TOE includes all security functions and mechanisms required for secure erasure of the disk drives attached to the target host. The security functions and mechanisms constituting the logical scope of the TOE is summarized in Table 3.

Table 3 Logical Scope of the TOE

Function/Mechanism	Description
Secure Erasure	The TOE implements a suite of secure drive erasure algorithms for complete erasure of data from the target drive. The erasure algorithm of the TOE implements the erasure logic in accordance with the selected erasure standard and calls the API of the selected drive to perform the actual operations on the drive. The TOE implements several erasure algorithms but not all of them are included in the logical scope of the TOE. The erasure algorithms included in the logical scope of the TOE are given in Sect. 1.2.1.

Erasure Verification	The TOE implements a suite of tests to verify the erasure results at the completion of each erasure. Any deviation is reported to the user of the TOE. The erasure reports are protected by a cryptographically secure message digest to prevent falsification. The TOE implements the verification logic and calls the erased drive API for the actual operations on the drive. The verification logic processes the responses from the API calls and reports the results to the user.
Configuration and Management of the TOE	The TOE implements a minimalistic GUI which allows the user to configure and manage the security of the TOE. The GUI is the only method of accessing the TOE and all configuration and management of the TOE must be done using the GUI.
Host Device Testing	Secure and complete erasure of the drives depends on the correct functioning of the underlying hardware. The TOE implements the erasure logic but depends on the API on each drive for accessing the drive. To assist the users of the TOE to determine the authenticity of the drive and to ensure that the drive erased is the one intended by the user, the TOE implements a suite of tools to allow testing of the underlying hardware.
TOE Authenticity and Legitimacy	The TOE implements diagnostics tools for the underlying hardware. The tools allow the user of the TOE to verify the integrity of the host device prior to an erasure. This helps in asserting that the underlying hardware (including the drive to be erased) is likely to function as expected and the erasure results can be trusted. Furthermore, the TOE implements access control mechanisms to ensure that only legitimate accesses to the TOE functions are granted. Erasure of the drive may only proceed if the TOE is legitimate, and the user has access to sufficient erasure licenses.

1.2.5 Non-TOE Hardware, Software and Firmware

The TOE contains the complete software required for the secure erasure of the associated drives. It boots up from a USB drive, network drive or the Host PC and executes independently in the host device to which the drives to be erased are attached. The TOE requires certain environmental components for a complete erasure solution. The mandatory and optional environmental components the TOE requires are stated in Table 4.

Table 4 Environment of the TOE

Component	Description
Host Device (mandatory)	The TOE requires a host device. When used in a standalone configuration, the host device is the host, usually a Personal Computer (PC), which boots up from the bootable media on which the TOE resides in the executable representation. The host device must be a 64-bit Intel compatible host with the minimum of 2GB of RAM. When used in the network configuration, the host device must additionally support Ethernet connection with PXE support. When used in the appliance configuration, the host device is usually a PC. The TOE is loaded to the host device and executed on top of the operating system of

	the PC. The host device must be a 64-bit Intel compatible host with the minimum of 2GB of RAM, executing a Windows operating system. The TOE does not use the drives of the host device which allows secure erasure of any drive attached to the host device. The drive may be any IDE, SATA, SAS, SCSI, NVMe, or eMMC drive.
Image Conversion Environment (mandatory)	The TOE is delivered to the user in a .iso image with an associated SHA-512 checksum. The .iso image must be stored by the user of a TOE and converted into an executable representation which is stored in the bootable media of choice of the user. The checksum verification, the conversion of the .iso image to an executable, and the storage of the executable representation of the TOE must be performed in a dedicated image conversion environment which is not part of the TOE. The image conversion environment is any trusted computing environment in which the representations of the TOE may be processed in a secure manner.
Bootable Media (Optional)	The TOE is downloaded as a non-executable .iso image. The user must produce an executable software from the .iso image and store it on a bootable media. The host device is then booted up from the bootable media for the TOE to become operational. The bootable media may be a network drive (if the TOE is used in the network configuration) or a USB drive (if the TOE is used in a standalone configuration). When used in appliance configuration, the TOE does not require bootable media but is booted by the operating system of the Host PC when the operating system boots up.
Boot Media Creator (Optional)	When the TOE is used in a standalone or network configuration, the .iso image representation must be converted to an executable representation and stored on a bootable media. The conversion is performed by the user of the TOE using a Boot Media Creator tool. The Boot Media Creator used in association with the TOE must be Ziperase Boot Media Creator Version 2.1.4 or newer. The boot media creator is not required when the TOE is used in an appliance configuration.
Network Connectivity (Optional)	When the TOE is used in a network configuration, the operational environment of the TOE must include the network connectivity and the TOE must be configured to use that network connectivity. Network connectivity is not required when the TOE is used in a standalone or appliance configuration.
Host for Installation (Optional)	The TOE maybe installed on a host PC or Server and erase attached disk drives either through the USB bus, disk controller, or HBA.
Message Digest Computation Tool (mandatory)	The .iso image representation of the TOE is distributed with an associated SHA-512 checksum. Prior to the conversion of the .iso image to an executable representation of the TOE, the user must verify the authenticity of the .iso image using an appropriate message digest computation tool. Since the checksum is used for verifying the authenticity of the non-executable .iso representation of the TOE, the message digest verification is not included in the logical scope of the TOE and must be performed off-line in the .iso image conversion environment.

Ziperase Command Center (Optional)	In the network configuration, the TOE is operated with the Ziperase Command Center. The target host will connect to the Command Center across a network and the TOE will communicate with the Command Center instead of a USB drive for all configuration, licensing, and reporting. The Command Center software must be version 3.0.2 or newer.

2 Conformance Claims

This section states the conformance claims applicable to the ST and the TOE.

2.1 Conformance Claim Statement

The TOE and ST are conformant with Common Criteria Part 1 [CCPart1], Common Criteria Part 2 [CCPart2] and Common Criteria Part 3 [CCPart3].

The TOE and the ST are Common Criteria Part 2 conformant and Common Criteria Part 3 conformant.

The TOE and the ST are package conformant to Evaluation Assurance Level EAL2.

This TOE is Protection Profile conformant to none.

2.2 Conformance Claim Rationale

The ST does not claim conformance to any Protection Profile. Therefore, the Conformance Claims Rationale is not applicable.

3 Security Problem Definition

The Security Problem Definition consists of Threats, Assumptions and Organisational Security Policies (OSP). They are all defined in this section. Each element in the Security Problem Definition is given a unique identifier and a statement of the element. Unique identifiers for the threats consist of prefix T. followed by a descriptor of the threat. Unique identifiers for the Assumptions consist of a prefix A. followed by a descriptor of the assumption. Unique identifiers for the OSPs consist of a prefix OSP. followed by a descriptor of the OSP.

3.1 Threats

The threats applicable to the TOE are given in Table 5.

Table 5 Threats applicable to the TOE

Threat ID	Threat Statement
T.INSECURE_ERASURE	A party not authorized to access the content of a drive gains physical access to an erased drive and succeeds in fully or partially restoring the content of the drive to the state prior to erasure. The restoration is made possible by the selection of an insecure erasure algorithm by the user of the TOE which does not sufficiently erase the content of the drive.
T.INCOMPLETE_ERASURE	A party not authorized to access the content of a drive gains physical access to an erased drive and succeeds in fully or partially restoring the content of the drive to the state prior to erasure. The restoration is made possible by the erasure of the drive not being completed without the user being informed of the incomplete erasure.
T.ILLEGITIMATE_CONFIG	A legitimate user of the TOE inadvertently operates the TOE in an insecure state after configuring the TOE in a manner which is not conformant with the legitimate uses of the TOE.
T.UNAUTHENTIC_HOST	An adversary gaining physical or logical access to the host device or a drive to be erased succeeds in modifying the low-level routines of the host device or the firmware of the drive. This in turn results in the loss of authenticity of the host or the drive, causing the drive API or the low-level access routines to the drive behave in an incorrect manner. Incorrect behaviour causes the drive access functions used for erasure and verification to indicate that a drive has been properly erased while in fact residual information on the drive may be used for fully or partially restoring the content of the drive.
T.TOE_INTEGRITY	An adversary succeeds gaining logical or physical access to the erasure environment and without detection modifying the .iso image of the TOE. This results in an unauthentic behaviour of the bootable TOE which in turn may result in the drives not being properly erased.

3.2 Assumptions

The assumptions applicable to the TOE are given in Table 6.

Table 6 Assumptions applicable to the TOE

Assumption ID	Assumption Statement
A.JURISDICTION	It is assumed that the user of the TOE selects an erasure algorithm, performs the necessary verifications, and handles at all times the TOE (in all representations), the erasure environment, and the host devices and drives in accordance with the policies and practices applicable in the jurisdiction in which the TOE is used.

3.3 Organizational Security Policies

The Organizational Security Policies (OSP) applicable to the TOE are given in Table 7.

Table 7 OSPs applicable to the TOE

OSP ID	OSP Statement
OSP.TRUSTED_USER	The user of the TOE is trusted, is sufficiently trained to use the TOE, shall not abuse or misuse the TOE, and operates the TOE at all times in accordance with the security guidance.
OSP.DIGEST_VERIFICATION	The message digest of the .iso image in which the TOE is distributed is verified against the message digest associated to the distribution at each time prior to the creation of a bootable media from the .iso image.
OSP..ENVIRONMENT	The entire operational environment in which the TOE is used - including storage as an .iso image, verification of the message digest of the .iso image, generation of a bootable media, booting up a target host from the bootable media, and operation of the TOE takes place in a secure environment. Physical and logical access to the representations of the TOE and the environment which the TOE is used is only granted to the legitimate users of the TOE.
OSP.NETWORK	The organization using the TOE implements a policy which states that the TOE must only be used in a network configuration when the environment in which the TOE is used, including the network connection, is trusted.
OSP.ERASURE_LIMITS	Users of the TOE are aware of the limits of erasure algorithms used (e.g. when used with RAID disks) and report to the risk owners the residual risks associated to erased drives.

4 Security Objectives

The security objectives are stated for the TOE Sect. 4.1 and for the operational environment of the TOE in Sect. 4.2. The security objectives rationale is given in Sect. 4.3.

4.1 Security Objectives for the TOE

The security objectives for the TOE are stated in Table 8.

Table 8 Security Objectives for the TOE

Security Objective ID	Security Objective Statement
O.SECURE_ERASE	The drive selected for erasure is, upon successfully completion of the erasure, completely erased. No residual information remains on the drive in quantity or format which could be exploited for fully or partially recovering data residing on the drive prior to the erasure.
O.COMPLETE_ERASE	The TOE ensures that each erasure is completed, or the user is given clear indicator of the incompleteness of a failed erasure. Outcome of each erasure is verified by the TOE prior to reporting completion of the erasure.
O.CONFIG	Configuration of the TOE is as intended by the user, and only produced using a legitimate management interface of the TOE. The TOE implements a GUI which is used for all configuration and management of the TOE and ensures that there are no alternative means of modifying the configuration or issuing commands to the TOE.
O.HOST_TESTING	The TOE allows the user to examine the integrity of the host device on which the TOE is executed. A set of tests on the underlying host may be executed by the TOE and the outcome reported to the user.

4.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are given in Table 9.

Table 9 Security Objective for the Operational Environment

Security Objective ID	Security Objective Statement
OE.LEGIT_ERASURE	The environment in which the TOE is used ensures that the use of erasure is consistent with all the laws, policies and other regulations governing the erasure of sensitive data. Sufficient evidence of the application of the necessary governance measures in the erasure of drives is produced to demonstrate conformance.
OE.USER	The user of the TOE is trained into the use of the TOE, commits to the use of the TOE in accordance with the security guidance at all times and shall not attempt to misuse or abuse the TOE for any gain.
OE.TOE_AUTHENTICITY	The .iso image representation of the TOE is at all times used in an environment which ensures that only legitimate users of the TOE gain access to it. The authenticity of the

	.iso image is verified prior to the creation of each bootable media from the image. No bootable media shall be generated from the .iso image if the verification fails.
OE.NETWORK	The network in which the TOE is used when operated in a network configuration is trusted and does not allow violation of the security of the TOE.
OE.ENVIRONMENT	The environment in which the .iso image of the TOE is handled and bootable media generated is physically and logically secure. Access to the TOE and the environment on which the TOE is used is only granted to legitimate users of the TOE. The environment, including all tools used in the handling of the TOE and generation of bootable media, is set up in full conformance with the security guidance of the TOE.

4.3 Security Objectives Rationale

Security objectives for the TOE and the operational environment are traced to the elements of the Security Problem Definition in Table 10. Each tracing is then justified to argue for the accuracy and completeness of the tracing.

Table 10 Tracing of Security Objectives to the Security Problem Definition

	T.INSECURE_ERASe	T.INCOMPLETE_ERASURE	T.ILLEGITIMATE_CONFIG	T.UNAUTHENTIC_HOST	T.TOE_INTEGRITY	A.JURISDICTION	OSP.TRUSTED_USER	OSP.DIGEST_VERIFICATION	OSP.ENVIRONMENT	OSP.NETWORK	OSP.ERASURE_LIMITS
O.SECURE_ERASURE	X										X
O.COMPLETE_ERASURE		X									X
O.CONFIG			X				X		X		
O.HOST_TESTING				X							
OE.LEGIT_ERASURE						X					X
OE.USER							X				
OE.TOE_AUTHENTICITY					X			X			
OE.NETWORK										X	
OE.ENVIRONMENT									X		

O.SECURE_ERASURE concerns with the TOE ensuring that each erasure is carried out in a secure manner. Secure erasure is one where no residual information remains on the erased drive such that could be exploited to fully or partially recover the erased data. This is achieved if threat T.INSECURE_ERASURE is prevented from occurring, and no adversary with access to an erased drive may recover the erased data. Furthermore, given the nature of the erasure being carried out using the API of the drive, and the possibility of self-recovering drives (e.g. RAID disks), there is a residual risk of erasure that under rare circumstances, the erasure may not be as complete as could be

reasonably expected. The organization using the TOE must ensure that by policy, each erasure report includes a statement of the residual risk associated to the erasure. This is fulfilled if OSP.ERASURE_LIMITS is fulfilled by the organization using the TOE.

O.COMPLETE_ERASURE concerns with the TOE always completing an erasure successfully, or informing the user of an incomplete erasure of a drive. At no time shall the erasure status be left ambiguous. Threat T.INCOMPLETE_ERASURE concerns with the possibility of an incomplete erasure facilitating restoration of the content of a drive after an erasure. O.COMPLETE_ERASURE is enforced if threat T.INCOMPLETE_ERASURE is prevented from occurring. There is always a residual risk of incompleteness associated to each erasure. This residual risk must be communicated to the user for completeness of the enforcement of O.COMPLETE_ERASURE. The communication of residual risk is sufficiently carried out if OSP.ERASURE_LIMITS is enforced in the organization using the TOE.

O.CONFIG concerns with the TOE at all times residing in a well-defined state, and only being managed through a legitimate management interface. This ensures the authenticity of the TOE when in the operational state. The authenticity of the .iso image representation of the TOE is addressed by OE.TOE_AUTHENTICITY. The TOE authenticity requires that the TOE is only managed through a well-defined management interface, that the TOE is only used by trusted users who do not seek to abuse or misuse the TOE, and that the operational environment of the TOE fulfills to a reasonable level that only authentic and approved software and other tools are used in association with the TOE. The first concern is addressed when threat T.ILLEGITIMATE_CONFIG is prevented from occurring and the TOE is at all times in a configuration intended by the user and only managed through a well-defined management interface. The second concerns is addressed if the users of the TOE are trusted, trained into the use of the TOE, always operate the TOE in accordance with the security guidance and do not seek to misuse or abuse the TOE. That is addressed if the organization using the TOE enforces OSP.TRUSTED_USER. Finally, the operational environment can be trusted to be sufficiently authentic so that the software and the tools of the operational environment behave as expected when OSP.ENVIRONMENT is enforced by the organization using the TOE.

O.HOST_TESTING concerns with the user of the TOE being able to unambiguously determine the state of the hardware of the host device and the drive to be erased. There are limits to the assurance on the completeness of the drive erasure caused by the fact that the actual erasure is carried out by the API of the drive and the TOE only implements the erasure logic in accordance with the erasure standards. O.HOST_TESTING is fulfilled by the TOE when reasonable assurance is gained for the authenticity of the hardware of the host device, including the drive to be erased. This is achieved if threat T.UNAUTHENTIC_HOST is prevented from occurring.

OE.LEGIT_ERASURE concerns with ensuring that the erasure algorithms and processes surrounding the use of the TOE are in accordance with the legal and other requirements for the erasure applicable to the geographic location in which the TOE is used. The TOE ensures that the erasure algorithms function correctly and that each erasure is completely carried out (or the user is clearly informed of an error) but there are no technical means by which the TOE could enforce that the policy requirements governing the use of the TOE are fulfilled. Therefore, the TOE must be used in accordance with A.JURISDICTION to ensure that the TOE is used in accordance with the applicable regulations. Further, the acceptance of residual risk of erasure is dependent on the applicable regulations and the user of the TOE must at all times report to the risk owner the residual risk in accordance with OSP.ERASURE_LIMITS. Jointly, operating the TOE in accordance with A.JURISDICTION and enforcing OSP.ERASURE_LIMITS in the operational environment enforce OE.LEGIT_ERASURE.

OE.USER concerns with ensuring that the TOE is at all times only used by legitimate users who do not abuse or misuse the TOE. The organisation using the TOE must ensure that they have a well defined personnel policy which is in force when designating personnel for using the TOE. Therefore, enforcing OSP.USER enforces OE.USER.

OE.TOE_AUTHENTICITY concerns with the authenticity of the TOE when stored in the .iso image representation. The .iso image is not executable and can therefore not include security measures to protect itself. Instead, protection of the authenticity of the TOE in .iso image representation requires measures applied to the environment of the TOE. The measures are twofold: First, the authenticity of the .iso must at all times when converted into a bootable representation of the TOE be verified using the cryptographic checksum as stated in OSP.VERIFICATION. Further, threats to the .iso representation of the TOE, i.e. T.TOE_INTEGRITY, must be prevented from occurring. The two concerns together enforce OE.TOE_INTEGRITY in the environment of the TOE.

OE.NETWORK concerns with the security of the network connection between the TOE and the Ziperase Command Center when the TOE is used in a network configuration. The TOE does not protect the network connection, and therefore the organization using the TOE must have in place a policy which only allows the use of the TOE in a network configuration when the network connection can be trusted. Enforcing OSP.NETWORK in the operational environment of the TOE fulfills OE.NETWORK.

OE.ENVIRONMENT is enforced if at all times the operational environment of the TOE is set up in accordance with the security guidance and all software and other tools are known and trusted. Enforcing OSP.ENVIRONMENT in the environment of the TOE fulfills OE.ENVIRONMENT.

5 Security Requirements

This section states the Security Functional Requirements and Security Assurance Requirements for the TOE.

5.1 Extended Components Definition

There are no extended components defined for this Security Target.

5.2 Notation and Conventions

The conventions used in the statement of the SFRs are as follows:

- Iteration is identified by repeating the identifier of the security functional requirement with a string indicating a specific iteration separated from the SFR identification by a slash (e.g. FCS_COP.1/AES, FCS_COP.1/DSIG).
- Refinement is identified by a) indicating in square brackets in **bold font** any added text, in form of **[Refinement: added text]** and b) indicating any removed words using ~~overstrike~~ font. Whenever a refinement is used, the rationale and justification of the refinement is given immediately after the statement of the security requirement.
- Selection is identified by indicating the selected values in **[square brackets using bold font]**.
- Assignment is identified by indicating the assigned values in ***[square brackets using bold, italic font]***.

Application notes may be added after the formal statement of the security requirements to assist the reader in understanding the specific security requirement in the context of this ST and TOE.

5.3 Security Functional Requirements

Security Functional requirements for the TOE are stated in an alphabetical order on a per security functional class.

5.3.1 Class FCS: Cryptographic Support

FCS_COP.1 Cryptographic operation

FCS_COP.1.1 The TSF shall perform [*Message digest Computation*] in accordance with a specified cryptographic algorithm [*SHA-256*] and cryptographic key sizes [*none*] that meet the following: [*Federal Information Processing Standard FIPS PUB 180-4*].

5.3.2 Class FDP: User Data Protection

FDP_ACC.1 Subset access control

FDP_ACC.1.1 The TSF shall enforce the [*Operation SFP*] on [

Subjects: User;

Objects: TOE Function;

Operations: Executing a TOE Function

].

FDP_ACF.1 Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the [*Operation SFP*] to objects based on the following: [

Subjects: User;

Object: TOE Function;

Security attributes of subject User: Licenses held;

Security attributes of object TOE Function: Licenses required;

].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

User is only allowed to execute a TOE Function if any of the Licenses held by the user is equivalent to the License required for the execution of the requested TOE Function

].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [*none*].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [*none*].

FDP_RIP.1 Residual Information Protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource [refinement: assured by the TOE] is made unavailable upon the [deallocation of the resource from] the following objects: [*Drive selected for erasure by the user of the TOE*].

Rationale: User of the TOE allocates a resource (i.e. a drive) for erasure by the TOE when the drive is selected for erasure. At that point of time no erasure takes place yet. Deallocation occurs when the user of the TOE commences with the actual erasure. Thereupon, the TOE erases that drive in accordance with the selected erasure standard and performs the necessary verifications. Because the drive selected for erasure is not part of the TOE, FDP_RIP.1 is not directly applicable and must be refined to indicate that the TOE performs a secure erasure of the data on the resource but that resource is not part of the TSF but a resource which is assured by the TOE. Therefore, the refinement is necessary to ensure precise statement of the SFR.

5.3.3 Class FIA: Identification and Authentication

FIA_UID.1 Timing of Identification

FIA_UID.1.1 The TSF shall allow [*Configuring and Managing the TOE*] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.3.4 Class FMT: Security management

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the [*Operation SFP*] to restrict the ability to [modify] the security attributes [*Licenses held*] to [*none*].

FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the [*Operation SFP*] to provide [restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [*none*] to specify alternative initial values to override the default values when an object or information is created.

Application note: The default value for the licences held by the user is none (which allows no access).

FMT_SMF.1 Specification of Management Functions

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

- *Initialization of the TOE;*
- *Setting the date and time of the TOE;*
- *Entering drive information not retrievable in a programmatic manner;*
- *Selecting a drive for erasure;*
- *Selecting erasure algorithm;*
- *Updating the erasure licenses;*
- *Managing erasure reports;*
- *Blinking the LED of the target drive;*
- *Testing the hardware of the host device;*
- *Displaying erasure log messages;*
- *Displaying available licenses;*
- *Powering off and rebooting the host device;*
- *Displaying server connection status; and*
- *Displaying drive content*].

5.3.5 Class FPT: Protection of the TSF

FPT_RCV.1 Manual recovery

FPT_RCV.1.1 After [*unscheduled termination of TOE execution*] the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_STM.1 Reliable time stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

FPT_TEE.1 Testing of external entities

FPT_TEE.1.1/Drive The TSF shall run a suite of tests [periodically during normal operation] to check the fulfillment of [*complete erasure of the drive*] .

FPT_TEE.1.2/Drive If the test fails, the TSF shall [*Report to the user*] .

Application note: Verification of the erasure is performed after each round of erasure and the outcome is reported to the user.

FPT_TEE.1.1/Platform The TSF shall run a suite of tests [periodically during normal operation] to check the fulfillment of [*correct functioning of the host device*] .

FPT_TEE.1.2/Platform If the test fails, the TSF shall [*Report to the user*].

5.4 Security Assurance Requirements

Security assurance requirements for the TOE constitute the evaluation assurance package EAL2 and are fully defined with reference to CC Part 3. The security assurance requirements constituting EAL2 are the following:

- Assurance Class ADV: Development
    - ADV_ARC.1 Security architecture description
    - ADV_FSP.2 Security-enforcing functional specification
    - ADV_TDS.1 Basic design
- Assurance Class AGD: Guidance documents
    - AGD_OPE.1 Operational user guidance
    - AGD_PRE.1 Preparative procedures
- Assurance Class ALC: Life-cycle support
    - ALC_CMC.2 Use of a CM system
    - ALC_CMS.2 Parts of the TOE CM coverage
    - ALC_DEL.1 Delivery procedures
- Assurance Class ASE: Security Target evaluation
    - 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_SPD.1 Security problem definition
    - ASE_TSS.1 TOE summary specification
- Assurance Class ATE: Tests
    - ATE_COV.1 Evidence of coverage
    - ATE_FUN.1 Functional testing
    - ATE_IND.2 Independent testing – sample
- Assurance Class AVA: Vulnerability assessment
    - AVA_VAN.2 Vulnerability analysis

5.5 Security Requirements Rationale

This section is the Security Requirements Rationale. It commences by demonstrating the fulfillment of the security functional requirement dependencies in Sect. 5.5.1. That is followed by the tracing of the security functional

components to the security objectives for the TOE and justifying that tracing in Sect. 5.5.2. Finally, the security assurance requirements are justified in Sect. 5.5.3.

5.5.1 Security Requirements Dependency Rationale

The dependencies of the security functional components are identified and their fulfillment justified in Table 11.

Table 11 Security Requirements Dependency Justification and fulfilment

Security Functional Component	Dependencies	Fulfillment
FCS_COP.1	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	None of the dependencies are applicable. Dependencies of FCS_COP.1 concern with the management of the cryptographic keys but the TOE only implements message digest computation with a cryptographically secure hash function. Such function does not use cryptographic keys and, therefore, the dependencies are not applicable.
FDP_ACC.1	FDP_ACF.1 Security attribute based access control	FDP_ACF.1 by the TOE
FDP_ACF.1	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1 by the TOE FMT_MSA.3 by the TOE
FDP_RIP.1	No Dependencies	N/A
FIA_UID.1	No Dependencies	N/A
FMT_MSA.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1 by the TOE FMT_SMR.1 is not fulfilled. The TOE does not implement user authentication. Therefore, there is also no meaningful role assignment but the authorization of a user to perform erasures is based in the availability of licenses. Configuration and management of the TOE using the GUI is allowed to any user without licenses being required. FMT_SMF.1 is fulfilled by the TOE.
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1 by the TOE FMT_SMR.1 is not fulfilled by the TOE (see argument in FMT_MSA.1).
FMT_SMF.1	No Dependencies	N/A
FPT_RCV.1	AGD_OPE.1	Fulfilled by the TOE Security Guidance.
FPT_STM.1	No Dependencies	N/A

FPT_TEE.1/Drive	No Dependencies	N/A
FPT_TEE.1/Platform	No Dependencies	N/A

5.5.2 Tracing of Security Functional Components to Security Objectives

The tracing of the Security Functional Components to the security objectives is given in Table 12. The tracing is followed by a justification of the tracing.

Table 12 Tracing of the Security Functional Components to the Security Objectives

Security Functional Component	O.SECURE_ERASE	O.COMPLETE_ERASE	O.CONFIG	O.HOST_TESTING
FCS_COP.1		X		
FDP_ACC.1			X	
FDP_ACF.1			X	
FDP_RIP.1	X			
FIA_UID.1			X	
FMT_MSA.1			X	
FMT_MSA.3			X	
FMT_SMF.1			X	
FPT_RCV.1		X		
FPT_STM.1			X	
FPT_TEE.1/Drive		X		
FPT_TEE.1/Platform				X

O.SECURE_ERASE concerns with a secure erasure of all information on the selected drives. This is ensured of the TOE completely erases the selected drive in accordance with the selecgted erasure standard. The TOE implements secure erasure as stated in FDP_RIP.1 for complete fulfillment of O.SECURE_ERASE.

O.COMPLETE_ERASE concerns with ensuring that the drive selected for erasure is completely erased so that no residual information may be used for recovering the data erased from a drive. This is fulfilled by three means which jointly fulfill O.COMPLETE_ERASE completely: 1) The integrity of the erasure reports is protected by a message digest computed with a cryptographically secure hash function (FCS_COP.1). This ensures that any report indicating incomplete erasure is truly reported to the user and threat agents may not modify the content without detection. 2) In case the erasure is incomplete, the erasure is not complete and the TOE enters a mode where all operations are inhibited (and no erasure report is generated) until the operator re-executes the erasure (FPT_RCV.1). 3) After each erasure, the TOE verifies the erasure for testing the erased drive for the presence of any residual information (FPT_TEE.1/Drive). if residual information remains, the user is informed of the incomplete erasure.

O.CONFIG concerns with ensuring that the TOE is at all times in an authentic configuration. This is a somewhat complex security objective which depends partly on the environment of the TOE for protecting the authenticity of

the TOE when stored in the .iso representation, but also on the ability of the TOE to ensure that the configuration is authentic, and that the TOE is only used in a legitimate mode when sufficient licenses are available.

The former concerns with ensuring that the TOE may only be used through a legitimate management interface which does not allow commands that could be used for bypassing the TOE controls. The TOE implements a GUI which is the only management interface to the TOE and implements a set of well defined management commands (FMT_SMF.1). The TOE also maintains a trusted time to ensure that all time stamps on erasure reports are accurate (FPT_STM.1). For license control, the TOE implements an access control model which allows access to the erasure functions only when a sufficient number of erasure licenses is present (FDP_ACC.1 and FDP_ACF.1). Users are identified through the licenses (FIA_UID.1) and the erasure may only take place if sufficient licenses have been explicitly allocated to the TOE. By default, the TOE contains no licenses and does not allow erasure of drives (FMT_SMF.1, FMT_SMF.3).

O.HOST_TESTING allows the user to examine the host device before and after erasure of associated drives to gain confidence that the host device is in an authentic state and the erasure is likely to be performed in accordance with authentic drive API calls. This is fulfilled by the TOE implementing a suite of tests for testing the host device and reporting to the user the findings (FPT_TEE.1/Platform).

5.5.3 Justification of the Security Assurance Requirements

The Security Assurance Requirements selected for the TOE constitute a well-defined evaluation assurance package EAL2 and as such, are an internally consistent set of security assurance requirements.

6 TOE Summary Specification

This section is the TOE Summary Specification. It describes first the fulfillment of each Security Functional Requirements. The fulfillment of the Security Functional Requirements is given in Table 14.

Table 13 Fulfilment of the Security Functional Components

Security Functional Component	Fulfilment
FCS_COP.1	The TOE implements a message digest computation for protecting the integrity of erasure reports. The TOE generates an erasure report in the JavaScript Object Notation (JSON) format and computes the message digest for subset of the fields on the report. The report is then exported in PDF with the message digest included. The user may upload the JSON file to the Command Center which verifies the message digest. There are no key management functions required as the TOE implements no cryptographic functions operating on the keys.
FDP_ACC.1 FDP_ACF.1	The TOE maintains in the configuration files the number of erasure licenses associated to each user. The number of licenses is verified prior to each erasure and the TOE will only perform the erasure of a sufficient number of licenses is available. The licenses are issued in quantity and one license is required per erasure. On the standalone configuration the licenses are stored on the USB drive and on the network configuration on the network.
FDP_RIP.1	The TOE ensures that the drive selected for erasure is completely erased in accordance with the selected erasure standard. Each erasure algorithm included in the certified configuration is considered secure and the TOE ensures that the algorithm is fully implemented using the standard API calls offered by the drive.
FIA_UID.1	The TOE does not maintain traditional users identified with a username and authenticated with a password. Instead, the users are only identified by the number of licenses they possess and erasure functions are only made available to those users possessing sufficient erasure licenses. As such, the only identification the TOE implements is through the erasure licenses.
FMT_MSA.1 FMT_MSA.3	By default, the TOE contains no licenses and, therefore, does not allow erasure of drives to take place. There are also no functions for modifying the default license values. License management may only take place over the GUI of the TOE. This ensures that at all times the default-deny policy is enforced on the erasures and licenses must have been explicitly allocated to the user to allow drive erasure to take place.
FMT_SMF.1	The TOE implements a GUI which executes at the start up of the TOE. The GUI is the only method of accessing the TOE and all TOE management and operation may only take place using the GUI. Each management function available to the user has a corresponding GUI function which implements the management function.
FPT_RCV.1	The TOE ensures that each erasure is complete or the TOE enters a maintenance state where the host device has to be rebooted with the TOE and the erasure re-executed. In case of an incomplete erasure when the erasure is interrupted by an error from which the TOE may not recover, the erasure halts and the TOE must be

	rebooted for the operation of the TOE to continue. No erasure report is generated.
FPT_STM.1	The TOE allows the user to set the time which is used for generating time stamps on erasure reports. Each erasure report is associated with a time stamp which allows the user to unambiguously determine the exact date and time at which the erasure occurred. The time may only be changed through a GUI function for setting date and time.
FPT_TEE.1/Drive	At the completion of each erasure, the TOE verifies the erasure results. The TOE implements a verification function which reads the content of the drive using the drive API and verifies that the erasure has been completely carried out in accordance with the erasure standard. The results are reported to the user in the erasure report.
FPT_TEE.1/Platform	The TOE implements a suite of functions to test the hardware of the host device. This allows the users to ensure that the host device is functioning correctly and has likely not been tampered with. This in turn increases the confidence in the erasure results.

7 Acronyms and Abbreviations

The acronyms and abbreviations specific to the TOE are given below. Common Criteria terms, acronyms and abbreviations are defined in Sect. 4 and Sect. 5 of [CCPart1] and are not reproduced here.

API	Application Programming Interface
eMMC	embedded Multi Media
GB	Giga Bit
GUI	Graphical User Interface
HBA	Host Bus Adapter
HMG	His/Her Majesty's Government
IDE	Integrated Drive Electronics
ISO	International Standardization Organisation
JSON	JavaScript Object Notation
LED	Light Emitting Diode
NIST	National Institute of Standards and Technology
NVM	Non-Volatile Memory
NVMe	NVM Express
PC	Personal Computer
PXE	Pre-Boot Execution
RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
SAS	Serial Attached SCSI
SATA	Serial Advanced Technology Attachment
SCSI	Small Computer System Interface
SHA	Secure Hash Algorithm
SSD	Solid State Disk
US	United States
USB	Universal Serial Bus