WIPERAPP sp. z o.o.

WIPERAPP_CORE

version 3.4.0

Security Target

Document version 1.8.1

Date of issue 2023-07-07

Common Criteria version 3 revision 5

Evaluation Assurance Level: EAL 4+

Augmented with: ALC_FLR.1

WIPERAPP – CONFIDENTIAL

Prepared for:	Prepared by:
WIPERAPP sp. z o.o.	**WIPERAPP sp. z o.o.**

(this page is intentionally left blank)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

History of changes in this document

Version ST	Authors	Date	Change description
1.0	Gruca Marcin Wyrwas Marcin	2020-08-20	Preliminary version
1.1	Gruca Marcin Wyrwas Marcin	2020-10-28	Revised version
1.2	Gruca Marcin Wyrwas Marcin	2021-08-22	Revised version according to WIP-OR-ASE
1.3	Gruca Marcin Wyrwas Marcin	2021-12-10	Revised version according to WIP-OR-ASE
1.4	Gruca Marcin Wyrwas Marcin	2021-12-13	Minor editorial changes for consistency and clarity
1.5	Gruca Marcin Wyrwas Marcin	2021-12-16	Minor editorial changes
1.6	Gruca Marcin Wyrwas Marcin	2022-01-05	Minor editorial changes
1.7	Gruca Marcin Wyrwas Marcin	2022-01-12	Minor editorial changes
1.8	Gruca Marcin Wyrwas Marcin	2022-05-05	Revised version according to WIP-OR-ASE-001
1.8.1	Wyrwas Marcin	2023-07-07	Company name change

1. Introduction to ST (ASE_INT)

This ST (Security Target) document describes security features for WIPERAPP EP WIPERAPP_CORE; version 3.4.0 (WIPERAPP_CORE), henceforth called the TOE (Target of Evaluation), and developed by WIPERAPP sp. z o.o.

This section identifies the Security Target (ST) document and the Target of Evaluation (TOE). It also describes the Security Target structure.

The ST document structure has been defined in Annex A to the first part of the Common Criteria (CC) standard [CC_1] and it contains the following:

- Introduction to ST (ASE_INT) – section 1 – identification (reference) of the ST document and the TOE, TOE overview and TOE description, i.e. physical and logical scope of the TOE;
- Conformance Claims (ASE_CCL) – section 2 – claims of conformance with a concrete version of the CC standard, packages (e.g. EAL) and protection profiles (PP);
- Security Problem Definition (ASE_SPD) – section 3 – description of threats, organizational security policies and assumptions regarding the TOE and its operational environment;
- Security Objectives (ASE_OBJ) – section 4 – description of security objectives – solutions that are proposed to solve particular aspects of the security problem in the form of security objectives for the TOE and for the operational environment;
- Extended Components Definition (ASE_ECD) – section 5 – definition of new security functional components (SFR) and security assurances components (SAR) which are not included in the second and third part of the CC standard;
- Security Requirements (ASE_REQ) – section 6 – Security Assurance Requirements (SAR) and expressing Security Objectives for the TOE by means of Security Functional Requirements (SFR);
- TOE Summary Specification (ASE_TSS) – section 7 – description of TOE security functionality (TSF), i.e. technical security mechanisms used by the TOE.

1.1 ST Reference

Table 1. ST Reference

ST Title	SECURITY TARGET for WIPERAPP EP, WIPERAPP_CORE v3.4.0
Full Document ID	WIPERAPP_CORE – Security Target (ST) / v1.8.1 / 2023-07-07 / *WIPERAPP sp. z o.o.*
ST Version	1.8.1
ST Date of issue	2023-07-07
Document language	English
File name	WIPERAPP_CORE_ST_v1.8.1
ST Authors	WIPERAPP sp. z o.o. Address: ul. Kominiarska 42B, 51-180 Wrocław (Poland) Telephone: +48 71 308 98 97 Website: https://www.wiperapp.com
ST Sponsor	WIPERAPP sp. z o.o. Address: ul. Kominiarska 42B, 51-180 Wrocław (Poland) Telephone: +48 71 308 98 97 Website: https://www.wiperapp.com
Certification ID	2020-4
IT Evaluation scheme	PC1 - IT Security Evaluation and Certification Scheme
Evaluation laboratory	ITSEF, Łukasiewicz-EMAG

1.2 TOE Reference

Table 2. TOE Reference

TOE Name:	WIPERAPP EP WIPERAPP_CORE
TOE Version	3.4.0
TOE Short name	WIPERAPP_CORE

TOE Developer	WIPERAPP sp. z o.o. Address: ul. Kominiarska 42B, 51-180 Wrocław (Poland) Telephone: +48 71 308 98 97 Website: https://www.wiperapp.com

1.3 TOE Overview

TOE WIPERAPP_CORE is part of the WIPERAPP application for permanent and irreversible data wipe from mediums. These mediums are storage devices such as:

- hard drives (mechanical and SSD) cooperating with the computer via the ATA, SATA, mSATA or M.2 (NGFF) interface,
- flash memory cards (e.g: Compact Flash, MMC, SD),
- flash drives,
- other data storage devices adapted to work with a computer via the USB interface.

The TOE has been prepared to run on IBM PC architecture compatible hardware platforms with x86-64 processor. The TOE runs in its own software environment (WIPERAPP application and Debian Linux), independent of the file systems, files and software on the computer’s hard disk, as it runs as “live” Linux distribution.

1.3.1 TOE usage and major security features

TOE WIPERAPP_CORE, part of a WIPERAPP application, has the following major security features:

- the ability to generate audit records including: the device (computer) on which the application was run, data medium wiped, recognition of media serial number errors, and recognition of media containing sectors registered by the S.M.A.R.T. as damaged.

  For the computer, the following are included:

  - device model, serial number,
  - processor type,
  - memory size,

  For data mediums, the following are included:

  - type,
  - capacity,
  - model,
  - serial number,
  - technical condition (attributes of the S.M.A.R.T. system, Bad Sectors).

- Ability to carry out the process of safe data wiping from the medium in accordance with predefined data wiping algorithms selected from the list of available algorithms,
- Ability to verify the correctness of the process of safe data wiping from the medium, carried out by reading and comparing with the expected values, and recording the start time of the safe data wiping process, recording the end time of the data wiping process verification process and calculating the total duration of the safe data wiping process (the total duration of data wiping is duration of data wiping process and the duriation of verify the correctness of the wiping process).
- ability to generate and securely (SHA512, ensuring the ability to verify integrity outside the TOE) export data necessary to create a certificate confirming the correctness of the process of secure data erasure from the medium, containing information:
    - about the user-operator,
    - collected during the identification of the device, including the erased data medium,
    - collected during the data erasure process, indicating used erasure method and the duration of the process,
    - collected in the process of verifying the correctness of the process of wiping data from the medium, including duration of the process and verify error, if any,
    - the version of the WIPERAPP software with which the data was deleted from the medium.

1.3.2 TOE Type

The TOE is a software for secure data erasure from storage devices.

1.3.3 Non-TOE hardware and software required by the TOE

WIPERAPP_CORE, the Target of Evaluation (TOE), is a component (set of modules) of the WIPERAPP aplication and does not work independently. For the correct operation of WIPERAPP_CORE, the software environment is required, which is the WIEPRAPP application and the Debian 10 Buster operating system. The software environment together with WIPERAPP_CORE are stored on a medium which is the WIPERBOX device. The TOE is launched via the LAN together with the software environment from the WIPERBOX device on the target computer to which the data erasure media is connected. The WIPERBOX device works with a target computer in the following architecture: Server (WIPERBOX) - Client (computer with data erasure media). The WIPERBOX device, the target computer (client) on which the environment is run and the environment itself (WIPERAPP application and Debian 10 Buster with the required libraries necessary for the operation of WIPERAPP), in which WIPERAPP_CORE is run, constitute only the enviroment of the TOE. The data mediums themselves, from which a data is wiped, are also not part of the Target of Evaluation.

For WIPERAPP_CORE to work correctly, an operational environment is needed in which the TOE is launched. The environment and WIPERAPP_CORE are launched on a computer with connected drives from which data are being erased. The computer on which the environment is launched, and the environment in which WIPERAPP_CORE is triggered, are only the operational environment of the TOE and are not subjected to evaluation. The drives from which data are being erased are not part of the TOE either.

For proper running WIPERAPP, a computer is required, which fulfills the following specifications:

- a standard PC compatible with the IBM/PC x86-64 architecture,
- an Intel Core2Duo processor with 1800 MHz clock rate or higher,
- 1GB RAM or bigger,
- a graphics card supported by Debian 10 Buster Linux,
- LAN network boot support (with PXE),
- the computer must be equipped with a controller that supports drives from which data are to be erased, e.g. ATA controller for ATA discs, SATA controller for SATA discs, USB controller for USB flash drives, etc.

To function properly, WIPERAPP_CORE has to be launched in the following environment:

- Debian 10 Buster Linux,
- The system has to have the following packages installed: xorg, xserver-xorg-video-all, lightdm, matchbox-window-manager, xbacklight, hdparm, net-tools, plymouth, nfs-common, nasm, openssh-server, less, plymouth-themes, binutils, nmon, htop, aspnetcore-runtime-3.1, dmidecode, libudisk2.

The WIPERAPP application, whose part is WIPERAPP_CORE, has to be launched in the system.

1.3.3.1 *Hardware and data mediums used to test WIPERAPP application*

The WIPERAPP application was tested with the use of hardware and drives listed in tables 3 and 4. These devices and mediums are not part of the TOE, but are only a set of test elements of the TOE environment, on which the development team performed the TOE tests.

Table 3 Description of hardware (computer compliant with x86 IBM/PC) on which the WIPERAPP application was tested

Computer compliant with x86-64 IBM/PC		
Type of hardware	Model / Type	Manufacturer

Computer (motherboard)	HP Compaq Pro 6200 Microtower PN XL504AV, Version BIOS J01 v02.33, Serial number CZC132889Y	Hewlett Packard
Processor	Core i3-2100 @ 3100MHz	Intel
Random-access memory	2x 2048MB PC3-10600U, 4096MB DDR3 1333MHz	Samsung
Graphics card	Intel® HD Graphics 2000 Built-in graphic system of the processor	Intel
LAN controller	82579LM on the motherboard	Intel
USB controller	6 Series/C200 Series Chipset Family USB	Intel
SATA controller	SI-PEX40064 based on Marvell 88SE9215*	IOCrest
PATA (IDE) controller	A-16 (PI2IT8212X3B) based on ITE IT8212F	ST-Lab (StarTech)

*Alternatively, it is possibile to use a 6 Series/C200 Series Chipset Family SATA AHCI controller, integrated into the motherboard.

Table 4 Data drives on which the WIPERAPP application was tested

Drives data						
No	Manufacturer	Model	Serial number	Firmware	Type	Sector size
1	Seagate	ST380011A	3JV1S67A	3.06	ATA HDD	512
2	WD	WD800JB-00FMA0	WD-WMAJ 97172164	13.03G13	ATA HDD	512
3	Seagate	ST380815AS	9RW4KAVF	4.ADA	SATA HDD	512
4	Seagate	ST9200423ASG	5TH06G57	DE14	SATA HDD	512
5	WD	WD800ADFS-75SLR2	WD-WMAN S2087297	21.07Q21 (R2)	SATA HDD	512
6	WD	WD1200BEVT-75ZCT2	WD-WXEX 08EU0448	11.01A11 (T2)	SATA HDD	512

7	Samsung	SM481N	S1K2NSAF 412041	DXM03D0Q	SATA SSD	512
8	Lite-On	LCS-128L9S-11	TW0XRV8D5 508548D3405	HC7110B	SATA SSD	512
9	SanDisk	Ultra USB 3.0 16GB	C453000024 0816120052	1.00	USB FLASH	512
10	Intenso (Toshiba)	6002560 (MQ04ABF100)	X7AYT1C3T	JU000U	USB HDD	512

1.4 TOE Description

1.4.1 TOE Physical scope

The Target of Evaluation (TOE) is a software called WIPERAPP_CORE together with the WIPERAPP_CONF configuration file necessary for the correct operation of WIPERAPP_CORE.

WIPERAPP_CORE consists of the following executables (binary dynamically linked libraries - DLL):

1. wiperapp.detect.dll
2. wiperapp.reporter.dll
3. wiperapp.verify.dll
4. wiperapp.wipe.dll
5. wiperapp.common.dll

WIPERAPP_CONF consists of the following configuration files (JSON structure):

1. common.settings.crypt (encrypted configuration file),
2. detect.settings.json.

A WIPERBOX device is delivered to the customer along with a short user manual printed (the full version of the user manual is available via the website in HTML web format). The WIPERBOX device is a minimum requirement computer acting as a server that hosts the WIPERAPP application image of which the TOE (WIPERAPP_CORE) is part.

WIPERAPP_CORE consist of 5 DLL libraries. The first four of them provide the basic security functionality of the TOE. The fifth library – wiperapp.common.dll – provides the function calculating a hash value acordingly to the SHA512.

WIPERAPP_CORE and WIPERAPP_CONF are loaded together with the WIPERAPP application (TOE environment element), from the server - WIPERBOX device (TOE environment element), via the LAN network interface to RAM memory of the device (TOE environment element), to which the data mediums to be erasure are connected. The application then runs on that device from RAM.

1.4.2 TOE logical scope

The logical scheme of the WIPERAPP application is presented in figure 1.

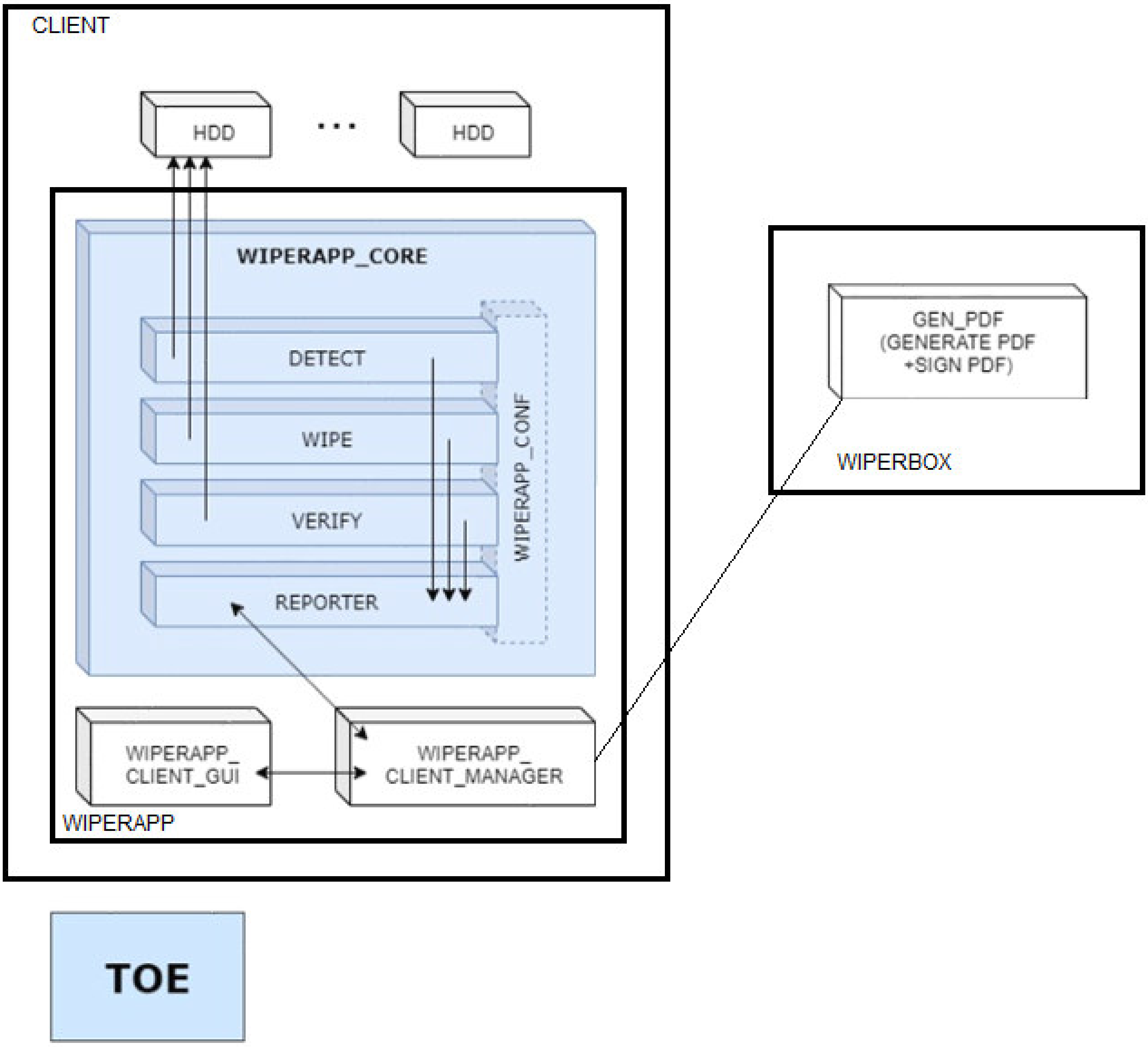

Figure 1. Logical scheme of WIPERAPP software

The application consists of modules of which only a few are part of the Target of Evaluation. The modules of the application are divided into the following blocks, or groups of blocks:

- WIPERAPP_CLIENT_GUI,
- WIPERAPP_CLIENT_MANAGER,
- WIPERAPP_CORE.

WIPERAPP_CLIENT_GUI is responsible for:

- communication of the WIPERAPP application with the system user,
- collecting from the user the orders to fulfil tasks,
- displaying the progress of the tasks execution and the results of the assigned tasks.

WIPERAPP_CLIENT_MANAGER is responsible for the coordination of the whole WIPERAPP software; it enables the cooperation of WIPERAPP_GUI with WIPERAPP_CORE and acts as an intermediate between these two.

WIPERAPP_CORE, which is the Target of Evaluation (TOE), is responsible for the following:

- identifying the specifications of the device on which it was launched,
- identifying data drives connected to the device in order to delete data from them,
- wiping information from the data drives according to the wiping algorithm selected by the user-operator,
- basic verification of the accuracy of the process of erasing data from the drives,
- Creation, collection, cryptographic security and export of data, which are indispensable to generale a report (certificate) confirming that the data have been erased.

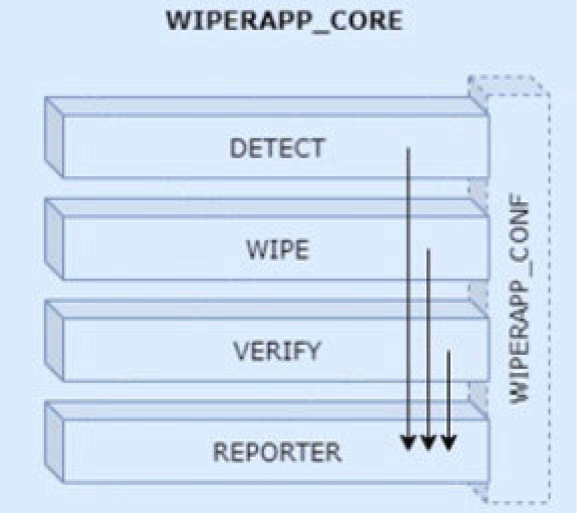

Figure 2. Logical scheme of WIPERAPP_CORE

WIPERAPP_CORE is the Target of Evaluation while the remaining elements are its environment. WIPERAPP_CLIENT_MANAGER, which is part of the TOE environment, ensures a direct interface to communicate with the TOE.

The TOE is composed of four modules:

1. **DETECT** – detection module:

The function of the detection module is to identify the device on which the TOE was launched to which the drives whose data are to be erased were connected. The module identifies the drives too.

The following data of the device are identified:

- manufacturer
- model

- serial number
- capacity of RAM memory
- processor type

The data are collected by reading the content of the DMI area of the device on which the WIPERAPP application is running, the amount of RAM memory is read by means of the "dmidecode" library, which is part of the TOE environment. These data are collected each time the WIPERAPP application is started.

The following information is identified for the drives from which data are to be erased:

- manufacturer,
- model,
- serial number,
- interface,
- software version,
- drive capacity in bytes and in the number of blocks (LBA)
- block size,
- drive status (hibernation, user password, HPA, DCO),
- drive type,
- ATA version ,
- rotation speed,
- S.M.A.R.T. attributes (Self-Monitoring, Analysis and Reporting Technology),

These data are collected by means of the „libudisk2” and „hdparm” libraries, which are part of the TOE environment. The data are always collected once the WIPERAPP application is launched. In addition, the DETECT module can detect drives which were connected after the application had been lauched. This is performed by means of the „libudisk2” library. The application waits for the message from the system that a new drive has been detected. The detection of a drive being disconnected is similar. Once the drive is connected, the same detection is performed for it and the same data are collected. On devices which support the replacement of drives during the device operation, it is possible to have an extra functionality – Hot Swap of drives during the program operation. Thanks to that, it is possible to replace the drives from which data have been already erased to new ones from which data are to be erased without the necessity to restart the device the drives are connected to. One can start the data erasure operation from successively connected drives. This enables to save time during simultaneous erasure of data from drives for which the duration of the erasure and verification process changes within a very wide range.

2. WIPE – module for data erasure from drives:

The data erasure process is consistent with records in the NIST SP 800-88 guideline, revision 1, December 2014. In the first phase of the process, the drive from which data are to be erased is restored to factory settings by resetting the HPA and DCO settings to default values. This allows to gain access to the whole user-accessible space of the drive detected in the DETECT module (the HPA function available for some drives might disable to access a certain space of the drive in which data may be stored; no access to this space makes it impossible to erase the data). In the second phase, the data are erased by logical overwriting or multiple overwriting of the whole accessible drive space with zero values or other pre-defined values (value patterns) according to the algorithms presented in the table below. The table contains exemplary descriptions of other organizations' algorithms in accordance with the provisions records in the NIST SP 800-88 guideline (December 2014).

Table 5 Data erasure algorithms available in WIPERAPP_CORE

Clear	The whole space of the drive, from the first to the last sector, is overwritten once by pre-defined one-byte values. By default, all sectors of the drive are overwritten once by 0x00 value. In one wiping process overwrite can be repeat from 2 to 16 times.
British HMG Infosec Standard 5, Baseline Standard	All sectors of the drive are overwritten once by a pseudo-random character generated by the kernel of the Linux system.
British HMG Infosec Standard 5, Enhanced Standard	All sectors of the drive are overwritten three times. In the first phase, the sectors are overwritten with 0x00 value, in the second phase – with 0xFF value, and in the third phase – with a pseudo-random value drawn from the range <0x00; 0xFF>
U.S.Air Force System Security Instruction 5020	All sectors of the drive are overwritten three times. In the first phase, the sectors are overwritten with 0x00 value, in the second phase – with 0xFF value, in the third phase – with a pseudo-random value drawn from the range <0x00; 0xFF>
DoD 5220.22-M	All sectors of the drive are overwritten three times. In the first phase, the sectors are

	overwritten with 0x00 value, in the second phase – with 0xFF value, and in the third phase – with a pseudo-random character generated by the kernel of the Linux system.
NAVSO P-5239-26	All sectors of the drive are overwritten three times. In the first phase, the sectors are overwritten with 0xFF value, in the second phase – with 0x00 value, and in the third phase – with a pseudo-random value drawn from the range <0x00; 0xFF>
Bruce Schneier's Algorithm	All sectors of the drive are overwritten seven times. In the first phase, the sectors are overwritten with 0xFF value, in the second phase – with 0x00 value, and in the remaining phases – with a pattern in the form of a series of 5 pseudo-randoms characters generated by the kernel of the Linux system.
U.S. DoD Unclassified Computer Hard Drive Disposition	In the first phase, the sectors are overwritten with a pseudo-random value drawn from the range <0x00; 0xFF>, in the second phase all blocks are overwritten with a complementary value (binary negation) to the one saved in this block, and in the third phase – with a another pseudo-random value drawn from the range <0x00; 0xFF>. Whole cycle is repeated 6 times.
German Federal Office for Information Security	In the first pass, the sectors are overwritten once with a pseudo-random character generated by the kernel of the Linux system. In the second pass all blocks are overwritten once with a complementary value (binary negation) to the one saved in this block. Whole cycle is repeated 2 times.
Communications Security Establishment Canada ITSG-06	All sectors of the drive are overwritten once in a three-pass process. In the first pass, all sectors are overwritten with 0xFF values. In the second pass, all blocks are overwritten with a

	complementary value to the one saved in this block during the first phase. In the third phase all sectors are overwritten with a pseudo-random character generated by the kernel of the Linux system.

According to the NIST SP 800-88 guideline (December 2014), overwriting the data once is sufficient to effectively delete data from ATA hard drives. "[2.3 Trends in Data Storage Media (page 6) - for ATA disk drives manufactured after 2001 (over 15 GB) clearing by overwriting the media once is adequate to protect the media from both keyboard and laboratory attack.]" and "[2.4 Types of Sanitization, Table 2-1. Sanitization Types (page 8) - overwriting is an acceptable method for clearing media. There are overwriting software or hardware products to overwrite storage space on the media with non-sensitive data. This process may include overwriting not only the logical storage location of a file(s) (e.g., file allocation table) but also may include all addressable locations. The security goal of the overwriting process is to replace written data with random data. Overwriting cannot be used for media that are damaged or not writeable. Studies have shown that most of today's media can be effectively cleared by one overwrite."

The WIPERAPP application allows the user to choose the methods of data overwriting. One of these methods is the CLEAR method – in this the user can choose the overwriting pattern and the number of overwriting passes. The minimum possible number of overwrite passes is one – therefore it is consistent with NIST 800-88 (September 2006). Other methods of overwriting have been developed by various global institutions and in our application we have used those that provide at least double overwriting pass and this is consistent with NIST 800-88 (September 2006), where the goal of overwriting data with random data was achieved.

3. VERIFY – verification module:

The TOE has a function that ensures basic verification of the data erasure process accuracy. The verification process lies in reading the whole space of the drive and comparing the contents of all read sectors with expected values that should be placed in the drive sectors after the data are deleted. The verification process is consistent with records in the NIST SP 800-88 guideline, revision 1, December 2014. If the content of the compared sector is different from the expected value, such an event is counted as a verification error.

4. REPORTER – module for generating report data:

The function of the module is to collect data about the following:

- user who commissions the operation of data erasure from drives, particularly his/her name and the commission number,
- drives from which data are to be erased, identified by the DETECT module during the detection process, particularly the drive type, capacity, manufacturer, model, and serial number,
- device on which the data erasure operation is performed and to which the drives are connected; these data are also collected automatically by the DETECT module, and are, particularly: the device manufacturer, model, serial number, RAM capacity, and processor type,
- organization or institution which performs the data erasure operation by means of the WIPERAPP application according to the order of the commissioner,
- user-operator who performs the operation of data erasure from the drives,
- program settings regarding the methods and the number of iterations of the data erasure operation for each drive, success or failure of the verification, extra information about errors which occurred during the data erasure and verification process,
- times of starting and completing the data erasure process as well as the whole duration of the process, including verification,
- version of the WIPERAPP software used to erase data from the drives.

These data are collected in one object – DEVICE, stored in the REPORTER module, in a temporary, internal repository of media to be erased. The object is protected with a cryptographic hash SHA512. Then SHA512 and the protected DEVICE object are exported outside the TOE. The TOE environment ensures extra protection of the object and its SHA512 hash during transmission within the TOE environment. Based on the data included in the DEVICE object it is possible to generate a correct certificate confirming that the data have been erased from the drive.

These data are used to generating WIPE REPORT (WIPE CERTIFICATE) document. The final report generated by the REPORTER module is based on the example described in the NIST 800-88 guide (December 2014) - Appendix F on page 35, therefore it complies with the provisions of the guide itself, that's why we refer to it as to the form of the report and the methods used to overwrite data in accordance with the provisions of this guide.

WIPERAPP_CONF in the form of an encrypted file is auxiliary for other modules because it is used to store information about their configuration, necessary for their proper operation, including:

- checksums of key application modules (including TOE modules) and system libraries to verify their integrity,
- patterns for verifying the correctness of serial numbers of data mediums (acceptable characters, permissible number of characters in the serial number).

2. Conformance Claims (ASE_CCL)

2.1 Conformance Claim with CC

This TOE is in conformance with the Common Criteria for Information Technology Security, Version 3.1, Revision 5, April 2017 (CC Part 2 – Conformant, CC Part 3 – Conformant)

2.2 Conformance Claim with PP

The Security Target (ST) does not declare conformance with any Protection Profile (PP).

2.3 Conformance Claim with packages

EAL4 package augmented with ALC_FLR.1 (EAL4+ALC_FLR.1).

2.4 Conformance Rationale

None.

The Security Target (ST) does not declare conformance with any Protection Profile (PP), thus a conformance rationale is not required.

3. Security Problem Definition (ASE_SPD)

In this section of the ST document the authors defined the security problem for the TOE and its development environment. Particular aspects of the security problem are expressed by threats and organizational security policies regarding both the TOE and its operational environment, and by assumptions related only to the operational environment. In addition, the authors defined assets and subjects which are used while describing particular aspects of the Security Problem Definition.

3.1 Assets

Table 6. Assets

Symbol	Description
D.PROTECTED_DATA	Confidentiality of the end user data stored in the device deleted by the TOE.

3.2 Subjects

Table 7. Subjects

Symbol	Description
S.ADMIN	WIPERAPP software administrator, who sets its work parameters, generates reports for end clients, can perform the data erasure operation.
S.USER	User who performs the data erasure operation and the parameter setting operation with the consent of S.ADMIN.
S.ATTACKER	Non-authorized subject who attempts to disturb the data erasure process or distort the report in order to gain access to D.PROTECTED_DATA.

3.3 Threats

Table 8. Threats

Symbol	Description
T.DISK_IDENTITY	The serial number of the device to be wiped is not correct due to either an intentional modification made by S.ATTACKER or to natural damage (e.g. wearing out of the magnetic surface or semiconductor structure) which makes the TOE performing an incorrect identification of the device. If the TOE does not detect the modification of the serial number, it may lead to the generation of an incorrect report confirming the wiping and D.PROTECTED_DATA may remain partially or completely unwiped, which may result in unauthorized and uncontrolled disclosure.

Symbol	Description
T.BAD_SECTOR	"Bad sector" flags are set in the device by either S.ATTACKER or due to normal medium operation (e.g. wearing out of the magnetic surface or semiconductor structure) which makes the TOE identify them as bad sectors and not securely wipe them. If the TOE does not detect the modification made by S.ATTACKER, it may lead to the generation of an incorrect report confirming the wiping and D.PROTECTED_DATA may remain partially or completely unwiped (in particular the information contained in the marked bad sectors), which may result in unauthorized disclosure.
T.CONNECTION	Any subject impersonates the WIPERBOX server and/or: (1) makes the client in which the TOE is supposed to run boot a non-legit OS with a non-legit TOE without detection. The user of the TOE would be under the impression that the data wiping of the storage device was correctly finished, which may result in unauthorized disclosure (2) modifies the data sent to the WIPERBOX from the TOE after a wiping process without being detected
T.BAD_USE	S.USER or S.ADMIN perform a bad use of the TOE, forcing it to work incorrectly by generating fake reports.

3.4 Organizational Security Policies

Table 9. Organizational Security Policies

Symbol	Description
OSP.WIPE	The TOE must wipe the data contained in the target storage device using any of the algorithms included in "Table 5 Data erasure algorithms available in WIPERAPP_CORE"
OSP.REPORT	The TOE must collect all audit data of the wipe process, encapsulate it, and generate a SHA-512 digest of it in order to transmit them to third IT entities for its integrity verification and report generation. Timestamps of wipe process, generated by TOE, must be reliable.
OSP.VERIFICATION	The TOE must verify the data written in the storage media after a wiping process for confirming that the erasure algorithm has worked properly.

3.5 Assumptions

Table 10. Assumptions

Symbol	Description
A.TIME	The TOE environment shall provide reliable timestamps that will be used by the TOE for its operation and reporting.
A.BIOS_SETTINGS	The BIOS settings of the clients in which the TOE will run shall be properly configured so that they allow the correct recognition and wiping data from the media intended for data erase. The device in which the TOE will run shall support booting from LAN (booting from a PXE server).

Symbol	Description
A.USERS	The administrator and users-operators of the system shall be competent people, i.e. they have been trained to use the WIPERAPP application in the ranges corresponding to the functions (roles) they have in the process of data erasure by means of this application.
A.NOEVIL	The administrator and users-operators shall not be irresponsible people who would deliberately cause negligence.
A.LOCATION	Both, the client in which the TOE runs and the WIPERBOX shall be located in secure facilities with controlles access so that no access rights are given to unauthorized or accidental users or persons.
A.KERNEL	All operating system kernel modules and libraries used by the TOE to communicate with data wiping media shall be from official authorized repositories (sources), stable, and will be included in accordance with the TOE addition or replacement procedure.
A.COMMUNICATION	It is assumed that the connection between the WIPERBOX and the client where the TOE runs is protected so that no attackers can access to it and try to disclose or modify the flow of information. In addition, the communication shall be done using cryptographic protected protocols.
A.RELIABLE_MEDIUM_ BEHAVIOUR	Customer organization ensure that disk identifiers and technical parameters are protected against their counterfight before their wiping by applying the procedurad means.

4. Security Objectives (ASE_OBJ)

This section of the Security Target features proposed solutions to particular aspects of the Security Problem in the form of Security Objectives for the TOE and its operational environment.

4.1 Security Objectives for TOE

Table 11. Security Objectives for TOE

Symbol	Description
O.VERIFY_DISK_IDENTITY	The TOE must detect errors in the identifiers (serial number) of the drives and activate an alarm for informing of this fact to the user.
O.BAD_SECTOR_WARNING	The TOE must implement a mechanism for the automatic detection of the drives with bad (inaccessible) sectors, including a notification to the user when bad sectors are found.
O.CONTROLLED_WIPE	The TOE must implement algorithms for wiping all data from the whole accessible space of the drive in a way that it disables future access to these data.
O.BASIC_VERIFY_ERASING_PROCESS	The TOE must implement a mechanism for verifying the data wiping process.
O.PROPER_REPORTING	The TOE must collect all audit data of the wiping process, encapsulate it and generate a SHA-512 digest of it. Both the data encapsulated and the SHA-512 digest must be sent to the TOE environment.

4.2 Security Objectives for Operational Environment

Table 12. Security Objectives for Operational Environment

Symbol	Description
OE.TIME	Before starting the process of wiping data from media, the current time will be set for all devices involved in the wiping of data. When starting the TOE, the current time is automatically taken from the timeserver running on the WIPERBOX, where the time is set at the factory. The time is displayed on the operator interface. The correct setting is ensured by comparing this indication with another device. The time indications should be within the current CEST time +/- 5 minutes. The handling of the larger time difference will be described in the user manual.

Symbol	Description
OE.BIOS_SETTING	Certain BIOS settings in devices used to erase data from media (devices with data erasure media attached) regarding controllers and storage media may prevent correct recognition of the media intended for wiping data, prevent the initiation or proper completion of the data erasure process (e.g. deactivation of SATA channels, deactivation of USB ports, data write lock "BOOT SECTOR VIRUS PROTECTION"). For this reason, the BIOS function settings (e.g. activation of SATA channels and USB ports) will be properly configured so that they do not prevent the correct recognition and wiping data from the media intended for data erase. The TOE requires the device it is run on to support booting from the LAN (booting from a PXE server). The manufacturer of the device (computer, motherboard) provides detailed information on possible BIOS settings.
OE.USERS	The administrator and users-operators of the system will be competent people, equipped with an authorized procedure describing how to properly perform the process of data erasure from the drives. They have been trained to use the WIPERAPP application in the ranges corresponding to the functions (roles) they have in the process of data erasure by means of this application. They will have knowledge and experience in wiping data from mediums. They have a high level of security awareness, compliance with the security policy, respect for procedures and a sense of responsibility for the security of the data wiping process.
OE.NOEVIL	The administrator and users-operators will not be irresponsible people who would deliberately cause neglect. In no circumstances would they deliberately act to distort the results of WIPERAPP work and the data contained in the certificate confirming the erasure of data from drives with the use of this application. They are people who act according to the knowledge and security policies derived from operating manuals, security policy documents, etc.
OE.LOCATION	Both, the client in which the TOE runs and the WIPERBOX will be located in secure facilities with access control so that no access is given to unauthorized or accidental users. The devices for erasing data from drives, along with the WIPERAPP application launched on them, are put in a safe place. The location of these devices will enable easy supervision by the system administrator and users-operators with a view to monitoring the accuracy of the process of data erasure from drives.
OE.KERNEL	All operating system kernel modules and libraries used by the TOE to communicate with data wiping media are from official authorized repositories (sources), are stable, and will be included in accordance with the TOE addition or replacement procedure.

Symbol	Description
OE.COMMUNICATION	The administrator will install the connection between the WIPERBOX and the client in which the TOE runs assuring that it is not possible to attack it by any unauthorized person or entity. All communication between client computer and WIPERBOX will be protected by TLSv1.2 protocol with ECDHE-RSA-AES256-GCM-SHA38 cipher. In addition every payload sent to WIPERBOX API is encrypted with AES. The communication takes place in two stages. In the first step, the data (secured with SHA512 sum) is transferred in an encrypted form and in a controlled manner by the WIPERAPP application between the TOE and the server that is located in the TOE environment. In the second stage, communication with the user's browser takes place in order to generate the certificate. The user obtains a link to download the certificate via the WIPERAPP application interface. The transmission is cryptographically secured; therefore, it is required to use a browser that supports HTTPS encryption. Detailed information on how to obtain certificates will be presented in the user manual.
OE.RELIABLE_MEDIUM _BEHAVIOUR	Customer organization will ensure that disk identifiers and technical parameters are protected against their counterfight before their wiping. All media in the customer organization are under control by the security policy rules.

4.3 Security Objectives Rationale

4.3.1 Tracings between Security Problem Definition and Security Objectives

Table 13. Mapping of Threats to Objectives

Security Objective Threat	O.VERIFY_DISK_IDENTITY	O.BAD_SECTOR_WARNING	O.CONTROLLED_WIPE	O.BASIC_VERIFY_ERASING_PROCESS	O.PROPER_REPORTING	OE.TIME	OE.BIOS_SETTING	OE.USERS	OE.NOEVIL	OE.LOCATION	OE.KERNEL	OE.COMMUNICATION	OE.RELIABLE_MEDIUM_BEHAVIOUR
T.DISK_IDENTITY	X						X				X		X
T.BAD_SECTOR		X					X				X		X

Security Objective **Threat**	O.VERIFY_DISK_IDENTITY	O.BAD_SECTOR_WARNING	O.CONTROLLED_WIPE	O.BASIC_VERIFY_ERASING_PROCESS	O.PROPER_REPORTING	OE.TIME	OE.BIOS_SETTING	OE.USERS	OE.NOEVIL	OE.LOCATION	OE.KERNEL	OE.COMMUNICATION	OE.RELIABLE_MEDIUM_BEHAVIOUR
T.CONNECTION					X				X	X		X	
T.BAD_USE								X	X				

Table 14. Mapping of OSPs to Objectives

Security Objective **OSP**	O.VERIFY_DISK_IDENTITY	O.BAD_SECTOR_WARNING	O.CONTROLLED_WIPE	O.BASIC_VERIFY_ERASING_PROCESS	O.PROPER_REPORTING	OE.TIME	OE.BIOS_SETTING	OE.USERS	OE.NOEVIL	OE.LOCATION	OE.KERNEL	OE.COMMUNICATION	OE.BEHAVED_MEDIUM
OSP.WIPE			X										
OSP.REPORT					X	X							
OSP.VERIFICATION				X									

Table 15. Mapping of Assumptions to Objectives

Security Objective **Assumption**	OE.TIME	OE.BIOS_SETTING	OE.USERS	OE.NOEVIL	OE.LOCATION	OE.KERNEL	OE.COMMUNICATION	OE.RELIABLE_MEDIUM_BEHAVIOUR
A.TIME	X							
A.BIOS_SETTINGS		X						
A.USERS			X					
A.NOEVIL				X				
A.LOCATION					X			
A.KERNEL						X		
A.COMMUNICATION							X	
A.RELIABLE_MEDIUM_BEHAVIOUR								X

4.3.2 Tracings justification

Table 16. Threats tracings – justification

Threat	Security Objective	Justification
T.DISK_IDENTITY	O.VERIFY_DISK_IDENTITY	This objective counters the threat by ensuring that the TOE will be able to verify correctly the parameters associated to the storage media, including its identity, and if any error is encountered an alarm is triggered. The TOE will therefore be able to wipe the storage media according to correct identity parameters.
	OE.BIOS_SETTING	This objective counters this threat by ensuring that the BIOS settings of the computer in which the TOE runs allow a proper recognition of the storage media to be wiped and thus

Threat	Security Objective	Justification
		the TOE will be able to check all the information regarding the identity of such media.
	OE.KERNEL	This objective counters this threat by ensuring that all software modules and libraries running on the computer in which the TOE runs are legit and therefore, there is nothing that could lead into an incorrect behaviour of the TOE in terms of disk identity recognition.
	OE.RELIABLE_MEDIUM_BEHAVIOUR	This objective counters this threat by ensuring that disk identifier can not be counterfight outside the operational environment of the WIPERAPP application.
T.BAD_SECTOR	O.BAD_SECTOR_WARNING	This objective counters the threat by ensuring that the TOE will be able to identify correctly the bad sectors contained in the storage media (if any). If an error is encountered an alarm is triggered. Therefore, the TOE will be able to wipe the storage media entirely.
	OE.BIOS_SETTING	This objective counters this threat by ensuring that the BIOS settings of the computer in which the TOE runs allow a proper recognition of the storage media to be wiped and thus the TOE will be able to check all the information regarding the bad sectors of such media.
	OE.KERNEL	This objective counters this threat by ensuring that all software modules and libraries running on the computer in which the TOE runs are legit and therefore, there is nothing that could lead into an incorrect behaviour of the TOE in terms of bad sectors recognition.
	OE.RELIABLE_MEDIUM_BEHAVIOUR	This objective counters this threat by ensuring that disk technical parameters can not be counterfight outside the operational environment of the WIPERAPP application.
T.CONNECTION	O.PROPER_REPORTING	This objective counters this threat by implementing a verification mechanism based on a cryptographic digest (SHA-512) of the information sent from the client

Threat	Security Objective	Justification
		in which the TOE runs to the WIPERBOX server.
	OE.NOEVIL	This objective counters this threat by ensuring that the users and administrators that are allowed to access to the premises in which the TOE is deployed will never try to attack the connection between the server and the client.
	OE.LOCATION	This objective counters this threat by ensuring that the server and the client are in a protected environment and not accessible to unauthorized persons or entities.
	OE.COMMUNICATION	This objective counters this threat by ensuring that connection between the server and the client is in a protected environment and not accessible to unauthorized persons or entities. In addition, the communication shall be done using cryptographic protected protocols.
T.BAD_USE	OE.USERS	This objective counters this threat by ensuring that the administrators and users will behave according to the guidance and are properly trained with experience of this kind of products.
	OE.NOEVIL	This objective counters this threat by ensuring that the administrators and users are trusted and will not use the TOE incorrectly.

Table 17. Mapping the organization's Security Policies – justification

OSP	Security Objective	Justification
OSP.WIPE	O.CONTROLLED_WIPE	This OSP is fulfilled by this objective by making the TOE perform the wipe operation of the media storage using the referenced algorithms.
OSP.REPORT	OE.TIME	This OSP is fulfilled by this objective by providing correct time stamps of start & finish time of a data wiping process.
	O.PROPER_REPORTING	This OSP is fulfilled by this objective by making the TOE collect the data of the wipe process, encapsulate it,

OSP	Security Objective	Justification
		generate a SHA-512 digest and transmit it to a third party for its verification.
OSP.VERIFICATION	O.BASIC_VERIFY_ERASING_PROCESS	This OSP is fulfilled by this objective by making the TOE perform a verification test for confirming that the data in the media storage after the wipe process is the one written by the TOE using a selected algorithm.

Table 18. Mapping assumptions – justification

Assumption	Security Objective	Justification
A.TIME	OE.TIME	This assumption is directly upheld by this objective.
A.BIOS_SETTINGS	OE.BIOS_SETTING	This assumption is directly upheld by this objective.
A.USERS	OE.USERS	This assumption is directly upheld by this objective.
A.NOEVIL	OE.NOEVIL	This assumption is directly upheld by this objective.
A.LOCATION	OE.LOCATION	This assumption is directly upheld by this objective.
A.KERNEL	OE.KERNEL	This assumption is directly upheld by this objective.
A.COMMUNICATION	OE.COMMUNICATION	This assumption is directly upheld by this objective.
A.RELIABLE_ MEDIUM_BEHAVIOUR	OE.RELIABLE_ MEDIUM_BEHAVIOUR	This assumption is directly upheld by this objective.

5. Extended Components Definition (ASE_ECD)

This section of the Security Target (ST) contains definitions of extended components, i.e. SAR and SFR newly-defined components not included in the catalogue of components defined in the second and third part of the CC standard.

5.1 Extended SAR Components Definition

There were no extended SAR components defined.

5.2 Extended SFR Components Definition

There were no extended SFR components defined.

6. Security Requirements (ASE_REQ)

This section of the Security Target (ST) features Security Functional Requirements (SFR) and Security Assurance Requirements (SAR), which are fulfilled by the Target of Evaluation (TOE).

The operations are written in the following manner:

- for assignments or selection operation: [*text in italics, in square brackets*];
- for iterations: the name of the component followed by the number of iteration in brackets, e.g. FAU_GEN.1(5).

6.1 Security Functional Requirements

Table 19. SFR Components

SFR Component	SFR Element	SFR Element description
FAU_GEN.1 Audit data generation Hierarchical to: No other components. Dependencies: FPT_STM.1 Reliable time stamps	FAU_GEN.1.1	The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the [selection: *not specified*] level of audit; and c) [assignment: - *Detection of a medium (containing bad sectors or not and its own medium serial number).* - *Completion of the verification process.*].

SFR Component	SFR Element	SFR Element description
	FAU_GEN.1.2	The TSF 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 the outcome (success or failure) of the event; b.) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment**:** *- Outcome of the bad sectors detection mechanism* *- Outcome of the incorrect identifier detection mechanism* *- Outcome of the verification process (success or error)* *- Total duration of the wiping process (duration of wiping and verification)*].
FAU_ARP.1 Security alarms Hierarchical to: No other components. Dependencies: FAU_SAA.1 Potential violation analysis	FAU_ARP.1.1	The TSF shall take [assignment: *- Display message about wrong identification of the drive* *- Display message about detecting a drive with bad sectors*] upon detection of a potential security violation.
FAU_SAA.1 Potential violation analysis Hierarchical to: No other components. Dependencies: FAU_GEN.1 Audit data generation	FAU_SAA.1.1	The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.
	FAU_SAA.1.2	The TSF shall enforce the following rules for monitoring audited events: a) Accumulation or combination of [assignment: *events regarding bad identification of drives and events regarding bad sectors detection*] known to indicate a potential security violation; b) [assignment: *none*].

SFR Component	SFR Element	SFR Element description
FDP_RIP.1 Subset residual information protection Hierarchical to: No other components. Dependencies: No dependencies.	FDP_RIP.1.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection**:** *deallocation of the resource from*] the following objects: [assignment**:** *data storage device*]. Application note: the algorithms for the data wipe are contained in "Table 5 Data erasure algorithms available in WIPERAPP_CORE"
FPT_TST.1 TSF testing Hierarchical to: No other components. Dependencies: No dependencies.	FPT_TST.1.1	The TSF shall run a suite of self tests [selection: *at the conditions [assignment: after a wiping process]*] to demonstrate the correct operation of [selection: *[assignment: the wiping process]*].
	FPT_TST.1.2	The TSF shall provide authorised users with the capability to verify the integrity of [selection: *[assignment: the data written on the wiped storage device by the TOE]*].
	FPT_TST.1.3	The TSF shall provide authorised users with the capability to verify the integrity of [selection: *[assignment: the wiping verification function]*].
FCS_COP.1 Hierarchical to: No other components. Dependencies: [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	FCS_COP.1.1	The TSF shall perform [assignment: *cryptographic hash function*] in accordance with a specified cryptographic algorithm [assignment: *SHA-512*] and cryptographic key sizes [assignment: *not applicable*] that meet the following: [assignment: *none*].
FPT_ITI.1 Inter-TSF detection of modification Hierarchical to: No other components. Dependencies: No dependencies.	FPT_ITI.1.1	The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and another trusted IT product within the following metric: [assignment: *SHA512 checksum. In order to detect modifications to the transmitted data, the SHA512 hash generated for this data will be attached to it*].

SFR Component	SFR Element	SFR Element description
	FPT_ITI.1.2	The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and another trusted IT product and perform [assignment: *In case of detecting the lack of integrity of the transmitted data, the product receiving the data will ignore the data, thus preventing the generation of a false report on the course of the data deletion process and informing the system administrator about it*] if modifications are detected.

6.2 Security Assurance Components

Table 20. SAR Components

SAR Class	SAR Component	SAR Component description
ASE: Security Target evaluation	ASE_INT.1	ST introduction
	ASE_CCL.1	Conformance claims
	ASE_SPD.1	Security problem definition
	ASE_OBJ.2	Security objectives
	ASE_ECD.1	Extended components definition
	ASE_REQ.2	Derived security requirements
	ASE_TSS.1	TOE summary specification
ADV: Development	ADV_ARC.1	Security architecture description
	ADV_FSP.4	Complete functional specification
	ADV_IMP.1	Implementation representation of the TSF
	ADV_TDS.3	Basic modular design
AGD: Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
ALC: Life-cycle support	ALC_CMC.4	Production support, acceptance procedures and automation
	ALC_CMS.4	Problem tracking CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_DVS.1	Identification of security measures
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.1	Well-defined development tools
	ALC.FLR.1	Basic flaw remediation
ATE: Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: basic design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.3	Focused vulnerability analysis

6.3 Security Requirements Rationale

This subsection of the Security Target (ST) contains tables which feature tracings between Security Objectives and SFR components, the justifications of the tracings and the rationale for selecting a given set of SAR components.

6.3.1 Tracings between Security Objectives and SFR Components

Table 21. Tracings between Security Objectives and SFR Components

Security Objective **SFR**	FAU_GEN.1	FAU_SAA.1	FAU_ARP.1	FDP_RIP.1	FPT_TST.1	FCS_COP.1	FPT_. ITI 1
O.VERIFY_DISK_IDENTITY	X	X	X				
O.BAD_SECTOR_WARNING	X	X	X				
O.CONTROLLED_WIPE				X			
O.BASIC_VERIFY_ERASING_PROCESS					X		
O.PROPER_REPORTING	X					X	X

6.3.2 Tracings Justifications

Table 22. Tracings of Security Objectives for the TOE – justification

Security Objective	**SFR Component**	**Justification**
O.VERIFY_DISK_IDENTITY	FAU_GEN.1	This SFR contributes to the fulfillment of this objective by generating audit records of the detection of medium, including identification information, which will later be analyzed.
	FAU_SAA.1	This SFR contributes to the fulfillment of this objective by analyzing the audit records generated by the TOE and searching for errors in the identification process of media storage devices.
	FAU_ARP.1	This SFR contributes to the fulfillment of this objective by displaying a message to the user upon a detection of an error in the identification process.
O.BAD_SECTOR_WARNING	FAU_GEN.1	This SFR contributes to the fulfillment of this objective by generating audit records of the detection of medium, including “bad sector” analysis.
	FAU_SAA.1	This SFR contributes to the fulfillment of this objective by analyzing the audit records generated by the TOE and searching for “bad sectors” issues in the identification and wipe process.
	FAU_ARP.1	This SFR contributes to the fulfillment of this objective by displaying a message

Security Objective	SFR Component	Justification
		to the user upon a detection of an issue associated to “bad sectors”.
O.CONTROLLED_WIPE	FDP_RIP.1	This SFR contributes to the fulfillment of this objective by performing an active wiping process using the algorithms defined in Table 5.
O.BASIC_VERIFY_ERASING_PROCESS	FPT_TST.1	This SFR contributes to the fulfillment of this objective by performing a post-process test for the verification of the data written in the storage media during the wiping process.
O.PROPER_REPORTING	FAU_GEN.1	This SFR contributes to the fulfillment of this objective by generating audit data that will be collected and put in the reports.
	FCS_COP.1	This SFR contributes to the fulfillment of this objective by carrying out a hash calculation with SHA-512 of the data collected and encapsulated during the wipe process.
	FPT_ITI.1	This SFR contributes to the fulfillment of this objective by providing a mean for the verification of the information sent to a third party, using the SHA-512 algorithm.

6.3.3 Dependencies Justification

The dependencies between SARs are all satisfied within the EAL4 package selected. The augmentation ALC_FLR.1 has no dependencies.

Table 23. SFR dependencies – justification

SFR Component	Dependent Component	Dependency satisfied	Justification of non-enforcement
FAU_GEN.1	FPT_STM.1 Reliable time stamps	No	Dependent on FPT_STM.1, however, time deviation control is provided by the environment (OE.TIME).
FAU_SAA.1	FAU_GEN.1	Yes	-
FAU_ARP.1	FAU_SAA.1 Potential violation analysis	Yes	-
FDP_RIP.1	None	-	-
FPT_TST.1	None	-	-
FCS_COP.1	[FDP_ITC.1 Import of user data without security attributes, or	No	The algorithm SHA-512 does not use cryptographic keys so there is no need to

SFR Component	Dependent Component	Dependency satisfied	Justification of non-enforcement
	FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction		either import them, generate them or delete them.
FPT_ITI.1	None	-	-

6.3.4 SAR Components Rationale

For the TOE, a coherent set of components was selected in the form of the EAL4 package with the conventionally added ALC_FLR.1 component. The selection of SAR (EAL4+) components does not contradict the attack potential of TOE-defined threat agents.

7. TOE Summary Specification (ASE_TSS)

This section of the Security Target (ST) describes in details the TOE Security Functionalities (TSF), i.e. how the TOE fulfills Security Functional Requirements (SFR).

7.1 Tracings between SFR Components and TOE Security Functionalities

Table 24. Tracings between SFR Components and TOE Security Functionalities

TSF Functionality **SFR Component**	TSF_1_DETECT	TSF_2_WIPE	TSF_3_VERIFY	TSF_4_REPORTER
FAU_GEN.1	X			X
FAU_SAA.1	X			
FAU_ARP.1	X			
FDP_RIP.1		X		
FPT_TST.1			X	
FCS_COP.1				X
FPT_ITI.1				X

7.2 Description of TOE Security Functionality

7.2.1 TSF_1_DETECT

TSF_1_DETECT is a function responsible for identifying the device on which the TOE has been run, recognizing media, recognizing certain classes of media serial number errors, and recognizing media containing sectors registered by the S.M.A.R.T. as damaged. TSF_1_DETECT to meet all component requirements: FAU_GEN.1, FAU_SAA.1, and FAU_ARP.1 performs the following operations:

- Every time a media intended for data wiping is connected to the system, it performs a media detection along with retrieving information about this media (FAU_GEN.1) from its registers using the operating system's kernel. The connected medium, if it is functional, reports its presence in the operating system. Such events are monitored by the TOE and, if necessary, upon detecting the appearance of the medium in the system, the data from which is to be deleted, the TOE retrieves data from the carrier's registers such as: interface, logical name in the system, model, serial number, manufacturer, software version, size in GB and in the number of LBA blocks, Block Size, media status (hibernation, user password, HPA, DCO), media type, ATA version, rotational speed, S.M.A.R.T attributes. Additionally, the event of detecting the connection of the medium to the system, stamped with a time stamp, is saved to the log file. The quality of this timestamp is controlled by the environment (OE.TIME), as it is responsible for providing these timestamps to the TOE.

- The TOE has an implemented a mechanism for automatic detection of errors in identifiers (serial number) of the drives (FAU_SAA.1).In the process of a drive identification the TOE may get inaccurate identification data of the drive, e.g. in the case when the drive is partially damaged, incompatible with any module of the erasing system, etc. The TOE automatically detects certain errors of that type: serial number with too few characters - less than 5, serial numbers with characters outside the acceptable range: "a-z", "A-Z", "0-9" and "-", identifiers with too many characters, above 32, empty identifiers. If such serial numbers are detected, the TOE will display a window informing about the detected problem and asking the operator whether to continue the data wiping process on such a medium (FAU_ARP.1), and if the wiping is continued, it will attach information about the occurrence of such an event to the certificate confirming the wiping of data from the drive. When connecting the medium intended for data erasure, after downloading identifiers from its registers, the attributes are downloaded from the S.M.A.R.T. system registers.
- The TOE has an implemented mechanism for the automatic detection of the drives with bad (inaccessible) sectors (FAU_SAA.1). In the process of the disc detection, the TOE collects parameters from the S.M.A.R.T. system records of the drive: the number of end-to-end errors (attribute No. 184), the number of sector relocation attempts (attribute No. 196), the number of sectors identified as unstable, waiting for relocation (attribute No. 197), the number of sectors which cannot be relocated (attribute No. 198) and the number of relocated parameters (attribute No. 5). If it is detected that any of these parameters has values different than zero, the TOE display a window informing about the detected problem and asking the user-operator whether the process of data erasure should be continued (FAU_ARP.1). If so, there will be information about the occurrence of such an event in the certificate confirming the data erasure from the drive.

The configuration data necessary for the correct operation of TSF_1_DETECT are stored in the WIPERAPP_CONF configuration file. It also stores the checksum of the DETECT module, which performs all TSF_1_DETECT tasks, in order to verify the integrity of this module.

7.2.2 TSF_2_WIPE

TSF_2_WIPE is a function responsible for carrying out the process of secure data wiping from the medium in accordance with predefined wiping algorithms. TSF_2_WIPE meets FDP_RIP.1, because in the case of a correctly performed data erasure process from the data medium, the TOE shall perform the following operations:

- It carries out the process of restoring HPA and DCO settings to their default state. Thanks to that, the TOE gains access to the whole user-accessible space of the data drive (to each user-accessible sector).
- It carries out single or multiple overwriting (FDP_RIP.1) of each drive sector with 0x00 and 0xFF values or random values in compliance with records in the NIST SP 800-88 guideline, revision 1, December 2014, according to algorithms described thoroughly in Table 5.

The configuration data necessary for the correct operation of TSF_2_WIPE are stored in the WIPERAPP_CONF configuration file. It also stores the checksum of the WIPE module, which performs all TSF_2_WIPE tasks, to enable the integrity of this module to be verified.

7.2.3 TSF_3_VERIFY

TSF_3_VERIFY is a function responsible for performing basic verification of the correctness of the safe data wiping process. TSF_3_VERIFY meets FPT_TST.1 (in a range of verifying the content of the medium after wiping process) by performing the following operations:

- Conducting the process of basic verification of the correctness of the process of wiping data from the medium by confirming that the data written by the wiping process is the expected according to the algorithm used (FPT_TST.1).

The configuration data necessary for the correct operation of TSF_3_VERIFY are stored in the WIPERAPP_CONF configuration file. It also stores the checksum of the VERIFY module, which performs all TSF_3_VERIFY tasks, in order to enable verification of the integrity of this module.

7.2.4 TSF_4_REPORTER

TSF_4_REPORTER is a function responsible for: recording the start time of the safe data wiping process, recording the completion time of the data wiping process verification process and calculation of the total duration of the secure wiping process, generating and safely exporting data necessary to create a certificate confirming the correctness of the process of safe data wiping from the medium. TSF_4_REPORTER fulfills: FAU_GEN.1, FCS_COP.1 and FPT_ITI.1 by the following operations:

- Collecting (FAU_GEN.1) and recording informations about the start time of the safe data wiping process, recording the completion time of the data wiping process verification process delivered from other TOE modules (other TSFs).
- Calculating the total duration of the secure wiping process based on these informations.
- The TOE sends to the WIPERAPP application the information about the completion of the verification process, and thus the entire process of removing data from the drive, along with the status of the verification performed, extended by information on the duration of the entire process of removing data from the drive and verification.
- The TOE collects all information regarding the device on which the TOE is running, the detected wiping drive connected to that device, and the data wiping and verification process flow. It then secures all this data by computing a hash (FCS_COP.1) using SHA512 method for the object containing the data and appending this hash to the data each time the data is sent outside the TOE to enable its integrity check (FPT_ITI.1). The function calculating the hash value is provided by the wiperapp.common.dll.
- If the lack of integrity of the received data is detected, the data is ignored due to the possibility of its corruption and the possibility of generating a report based on this data is blocked.

The configuration data necessary for the correct operation of TSF_4_REPORTER are stored in the WIPERAPP_CONF configuration file. It also stores the checksum of the REPORTER module, which carries out all TSF_4_REPORTER tasks, in order to verify the integrity of this module.

8. Appendix

8.1 Abbreviations

Abbreviation	Explanation
CC	Common Criteria
EAL	Evaluation Assurance Level
IT	Information Technology
OSP	Organisational Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SPD	Security Problem Definition
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality

8.2 Terms and Definitions

The glossary explains those terms used in the document whose meaning may be unclear or is specific with respect to the Common Criteria standard. Terms explained in [CC_1] were not repeated here.

8.3 References

[CC_1] Common Criteria, Part 1: Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1, Rev.5, April 2017.

[CC_2] Common Criteria, Part 2: Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements, Version 3.1, Rev.5, April 2017.

[CC_3] Common Criteria, Part 3: Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements, Version 3.1, Rev.5, April 2017.

[CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Rev.5, April 2017.

[ST_Guide] ISO/IEC TR 15446 Guide for the production of Protection Profiles and Security Targets.

[AB_SI] Białas A.: Semiformal Common Criteria Compliant IT Security Development Framework. Studia Informatica vol. 29, Number 2B(77), Silesian University of Technology Press, Gliwice 2008.