Blancco File Eraser Security Target

For the Common Criteria Certification of Blancco File Eraser Version 2.0

www.blancco.com

Table of Content

1 Security Target Introduction 3
1.1 ST Reference 3
1.2 TOE Reference 3
1.3 TOE Overview 3
1.4 TOE Description 3
2 Conformance Claims 7
3 Security Problem Definition 8
3.1 Threats 8
3.2 Organisational security policies (OSPs) 8
3.3 Assumptions 8
4 Security Objectives 10
4.1 Security Objectives for the TOE 10
4.2 Security Objectives for the Operational Environments 10
4.3 Security Objectives Rationale 10
5 Extended Components Definition 12
6 Security Requirements 13
6.1 Security Functional Requirements 13
6.2 Security Functional Requirements Rationale 13
6.3 Security Assurance Requirements 14
6.4 Security assurance requirements rationale 15
7 TOE Summary Specification 16
7.1 Security Functions and Associated Security Functional Requirements 16
7.2 SF.Erase_Data 16
7.3 SF.Report_Results 16
8 Abbreviations and Terms 17

1 Security Target Introduction

1.1 ST Reference

ST Title – Blancco File Eraser Security Target

ST Version – 2.0

ST Date – 2017-11-24

1.2 TOE Reference

TOE Identification – Blancco File Eraser (Home Edition), Blancco File Eraser (Enterprise Edition) and Blancco File Eraser (Data Center Edition)

TOE Version – 8.2 (applicable to all the editions)

1.3 TOE Overview

This is the Security Target for Blancco File Eraser (Home Edition), Blancco File Eraser (Enterprise Edition) and Blancco File Eraser (Data Center Edition). When not explicitly stated, descriptions of Blancco File Eraser (BFE) apply to all the versions.

Blancco File Eraser (BFE) enables the secure deletion of selective data on PCs, servers and virtual machines. They are also optimized for erasure of selected files and folders in a corporate network. It can be used to erase specified paths or commanded with automated tasks using various rules.

Detailed information of each erasure performed by BFE is stored in an erasure report. This report provides proof that the erasure has been performed successfully.

BFE is a software application running on Windows. The software is operated via a graphical user interface (GUI) or by using a Command Line Interface (CLI).

1.3.1 System Requirements

BFE is a Windows based solution, both 32 and 64 bit systems are supported. BFE works on single machines or in a network. It can erase selected files on both clients and servers.

The following Windows versions are included in the evaluation:

- Windows 7, 8 and 10
- Windows Server 2008 and 2012

BFE supports the following file systems: NTFS, FAT32 and exFAT.

1.4 TOE Description

The key functionality offered by BFE is the erasure of files and folders on a local computer or remotely executed in storage areas across a given network. BFE can be installed on both PCs and servers (physical/virtual) and remote installation and deployment is available through the use of Microsoft Installer (MSI) packages. Granular control of software distribution is enabled through the centralized configuration of clients and multiple installations through group policy files. Additionally, BFE can be run from a standalone USB stick when plugged into a computer or server to aid mobility and operations on-demand.

BFE is storage media agnostic i.e. it does not matter what storage medium is underpinning the system on which data is being erased. BFE works with the Windows API to access files and does not consider if the device is, for example, a Hard Disk Drive (HDD), Solid State Drive (SSD) or any other relevant medium.

There are three named editions of Blancco File Eraser:

- Blancco File Eraser (Home Edition)
- Blancco File Eraser (Enterprise Edition)
- Blancco File Eraser (Data Center Edition)

All the editions are operable via the use of a GUI (TSFI 2) using the *BlanccoFileEraser.exe* application.

Enterprise Edition and Data Center Edition also include an additional application to operate the tool via a CLI (TSFI 1). The application is called *BlanccoFileEraserCmd.exe*, and it provides more granular control of the file erasure process. However, the logic used to perform the actual erasure of files and generating reports is the same for GUI and CLI. The CLI tool can be used both manually and with scripting, but the usage method does not affect to the functionality.

Only Data Center Edition can be installed on server environments running a Windows Server operating system. All the editions can be installed on PC environments that have a Windows desktop operating system.

The core functionality is the same for all the editions. The security functionality (i.e. erasing files and reporting) is exactly the same. It is built on the same code base and using the same application extensions.

The table below summarizes the technical differences between the product editions.

Table 1. Differences between the available product editions

Feature	Home Edition	Enterprise Edition	Data Center Edition
Installation on PCs	Yes	Yes	Yes
Installation on servers	No	No	Yes
Installation by .exe and .msi	Yes	Yes	Yes
Graphical user interface	Yes	Yes	Yes
Command line interface	No	Yes	Yes

The command line program uses parameters to achieve desired operations. Parameters are passed to the operation according to the needs of the user. For example, the erasure algorithm is selected by passing a number parameter that calls the required process.

Using the Windows build in scheduling, such as the Task Scheduler found in the Administrative Tools of Microsoft Windows 7, BFE allows the scheduling of erasure for selected files on local or remote servers and work stations. It is also possible to create rules and automatic routines to erase files and folders and policy-based scheduling and integration through Windows standard components. Please, note that the scheduling is a feature of the Windows environment and not of BFE.

The erasure of a file is performed using overwriting techniques defined by erasure algorithms. The erasure algorithm used for erasure can be selected by the user at the time of use. The contents of the file are overwritten multiple times using different data patterns, depending on the erasure algorithm chosen.

An erasure algorithm originate usually from an information security standard (see Table 2). The standards describe erasure methods in terms of the amount of times that data is overwritten and what data pattern (for example, all binary ones or zeroes) is used. While the standards do not provide explicit instruction for erasing files, the overwrite patterns they mandate for securely erasing data are applicable to file erasure. Therefore, it is only the patterns for erasure that are extracted from the listed standards – no other conformance is claimed.

Only the erasure algorithms listed in Table 2 are included in the evaluation.

Table 2. Included erasure algorithms and the related standards

Algorithm Name	Document Title and Version	Additional Information
HMG Infosec Standard 5, Lower standard	HMG Infosec Standards No. 5, Secure Sanitisation of Protectively Marked or Sensitive Information, September 2007	See the Appendix C of the document
HMG Infosec Standard 5, Higher standard		
U.S Department of Defense Sanitizing (DoD 5220.22-M)	1. DoD 5220.22-M National Industrial Security Program Operating Manual (NISPOM), February 28, 2006 2. DSS Clearing and Sanitization Matrix, June 28, 2007	See the method *d* in the document 2. The document 2 is a technical supplement to the document 1.
NSA 130-1	1. NSA/CSS Manual 130-1, Information System Security Training Requirements, September 2001 2. Joint DoDIIS/Cryptologic SCI Information Systems Security Standards, 31 March 2001, Revision 2	See the section 20.4.2.1 of the document 2. The document 2 is a technical supplement to the document 1.
NIST 800-88 Clear	NIST Special Publication 800-88, Guidelines for Media Sanitization, Revision 1, December 2014	See Appendix A of the document
Aperiodic random overwrite	N/A	The erasure algorithm consists of one overwriting pass with pseudo-random data

Some standards mandate a verification record to be produced. For verification reports the BFE generates a report as a result of the erasure process presenting detailed information about the time, date, process used and more to ensure compliance with the auditing requirements of various standards and legislation listed above. The reports are save into the Windows file system as HTML files.

The erasure report contains for example the following information:

- If the job was successful (otherwise a warning will be shown)
- File(s) erased
- Time when erasure operation was completed
- Erasure algorithm used
- Computer name on which the job was performed.

By using the Blancco Management Console (not within the scope of the evaluation) it is possible generate reports also in XML and PDF formats. The user may decide to store the reports in a centralized in a dedicated repository, but no such functionality is within the scope of the TOE.

The BFE also provide a log module that allow the reporting to also be handled by the Windows Event-log to log events and allow them to be viewed in the Windows Event Viewer.

The TOE is software only and consists of the executable, along with user documentation. The physical scope of the TOE is:

- Software executable (*BlanccoFileEraser.exe* or *BlanccoFileEraserCmd.exe*)

- *Blancco File Eraser, User Manual for version 8.2*
- *Blancco File Eraser, Administrator's manual for version 8.2*
- *Blancco File Eraser, Common Criteria Supplement for version 8.2*

The software and the documentation can be downloaded from the Blancco web page.

2 Conformance Claims

This Security Target is CC Part 2 and CC Part 3 conformant. This Security Target claims conformance to CC version 3.1 Revision 4.

This ST claims no conformance to any Protection Profile. This ST claims conformance to the EAL2 package of security assurance requirements, augmented with ALC_FLR.2.

3 Security Problem Definition

The TOE aims to address the threat to data security posed by the improper deletion of sensitive data contained within any type of computer file. The nature of 'normal' file deletion processes performed by an Operating System (Windows, in the case of the TOE) is that the data that populates a file remains on the physical storage medium when deleted. Only the reference to the file is removed at the file system level i.e. a logical level deletion, which does not allow the host or user to access the file.

However, the recovery of logically deleted files is possible with widely available tools. For example, data recovery software applications can 'rebuild' files by scanning the storage and locating and identifying the necessary data that makes up the content of a given file. This is because the data is physically still present on the disk after a delete command. The area of storage that previously held the file is now considered as unallocated and the contents may be eventually overwritten with other data. However, the user has no control over this and it is possible to recover sensitive information from fragments of files - for example, plain text from parts of documents.

The threat agent may be an attacker using manual tools or an automatic attack data by an IT product.

The assets are data stored on a storage device.

3.1 Threats

This section identifies the threats that are to be countered by the TOE and the TOE environment.

Threat	Description
T.Recovery	A threat agent gains access to a storage device after sensitive data files have been improperly (logically only) erased and is able to recover the contents of the file(s) using software or hardware tools.

3.2 Organisational security policies (OSPs)

This section identifies the organizational security policies that are enforced by the TOE and the TOE environment.

OSP	Description
P.Report	The TOE will report the results of an erasure process providing an indication of the success (or otherwise) and details about how and when it was performed.

3.3 Assumptions

This section specifies the assumptions on the TOE environment that are necessary for the TOE to meet its security objectives.

Assumption	Description
A.Users	Personnel using the TOE must have been trained, competent and follow all applicable guidance documentation.
A.Platform	The underlying hardware, firmware and the operating system functions needed by the TOE to guarantee secure operation, are working correctly and have no undocumented security critical side effect on the functions of the TOE.
A.Time	The platform must provide a time stamp and ensure that the time is correctly set.

A.Repository	The operational environment must provide storage to retain reports generated by the TOE in order to use them later for auditing/erasure proof requirements.

4 Security Objectives

The TOE mitigates this aforementioned threats by providing a more thorough erasure of a given file (and any previous versions associated with it) by replacing its contents with redundant data before performing a logical delete. This ensures that sensitive files cannot be recovered if a threat agent were to gain access to a storage device. Additionally, the slack space associated with a given file is erased by the TOE, removing the potential for sensitive data to be recovered from this.

The TOE also enables the erasure of free space on a hard drive, which removes remnants of other potentially sensitive data that was previously written to the storage medium but is no longer required.

4.1 Security Objectives for the TOE

The following are the security objectives to be met by the TOE.

Security Objective	Description
O.Erasure	The TOE is capable of erasing all data from the targeted files (along with previous versions, if selected) and/or free storage space, depending on what has been selected by the user, in such way that attempting to read the original data will fail.
O.Report	The TOE shall provide information of the erasure process, consisting of erasure success or failure, the date erasure was performed, the erasure standard used and information about the content that was erased.

4.2 Security Objectives for the Operational Environments

The following are the security objectives to be met by the TOE environment.

Security Objective	Description
OE.Users	Personnel using the TOE must have been trained, competent and follow all applicable guidance documentation.
OE.Platform	The underlying hardware, firmware and the operating system functions needed by the TOE to guarantee secure operation, are working correctly and have no undocumented security critical side effect on the functions of the TOE.
OE.Time	The platform must provide a time stamp and ensure that the time is correctly set.
OE.Repository	The operational environment must provide storage to retain reports generated by the TOE in order to use them later for auditing/erasure proof requirements.

4.3 Security Objectives Rationale

4.3.1 Security Objective Completeness

The following tables provide a mapping of security objectives both for the TOE and the TOE environment to the environment defined by the threats, policies and assumptions, illustrating that each security objective for the TOE covers at least one threat or policy, and that each security objective for the TOE environment covers at least one policy, threat or assumption.

	T.Recovery	P.Report	A.Users	A.Platform	A.Time	A.Repository
O.Erasure	X					
O.Report		X				
OE.Users			X			
OE.Platform				X		
OE.Time					X	
OE.Repository						X

4.3.2 Security objectives sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat actually contributes to the mitigation of that threat.

Threat	Rational for the security objectives
T.Recovery	The security objective O.Erasure ensures that all data from the targeted files (along with previous versions, if selected) and/or free storage space will be erased.

The following rationale provides justification that the security objectives of the TOE and the TOE environment are suitable to address each individual OSP and that each security objective tracing back to a OSP actually contributes in addressing the OSP.

OSP	Rational for the security objectives
P.Report	The security objective O.Report ensures that information of the erasure process, consisting of erasure success or failure, the date erasure was performed, the erasure standard used and information about the content that was erased will be reported to the user and audited.

The following rationale provides justification that the security objectives of the TOE environment are suitable to address each individual assumption and that each security objective tracing back to an assumption actually contributes in addressing the assumption.

Assumption	Rational for the security objectives
A.Users	The security objective OE.Users is literally the same as the assumption.
A.Platform	The security objective OE.Platform is literally the same as the assumption.
A.Time	The security objective OE.Time is literally the same as the assumption.
A.Repository	The security objective O.Repository is literally the same as the assumption.

5 Extended Components Definition

There are no extended components defined or used in this ST.

6 Security Requirements

6.1 Security Functional Requirements

The following convention is used for operations applied to the Security Functional Requirements: Assignment and selection are indicated by **bold**. Refinements are indicated by **bold underscore** for additions and by ~~**bold strike through**~~ for deletions.

6.1.1 FDP_RIP.1 – Residual Information Protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource from** the following objects: **files**, **free space on the storage device**.

Application note: Blancco File Eraser (BFE) is storage media agnostic i.e. it does not matter what storage medium is underpinning the system on which data is being erased. BFE works with the Windows API to access files and does not consider if the device is, for example, a Hard Disk Drive (HDD), Solid State Drive (SSD) or any other relevant medium. Please refer to Table 2 for supported erasure standards.

6.1.2 FAU_GEN.1 – Audit data generation

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**~~ **erasure**
- b) All auditable events for the **not specified** level of audit; and
- c) **The following event:**
    1. **erasure as described in FDP_RIP.1**.

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; and
- b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST,
    1. **the data erased (files, free space etc)**
    2. **the erasure standard used**
    3. **process duration**
    4. **software information (OS name, OS version, computer name and user name)**.

Application note: There is no start-up or shutdown of the audit function since audit is always active and cannot be deactivated, meaning that all events are always being audited all the time. This means that the refinement is not a limitation of the SFR, but rather a clarification of the audit function.

6.2 Security Functional Requirements Rationale

6.2.1 Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective and that all security objectives are addressed by one or more SFRs.

	O.Erasure	O.Report
FDP_RIP.1	X	
FAU_GEN.1		X

6.2.2 Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives.

Security Objective	Fulfilment of the security objective
O.Erasure	The security objective is met by FDP_RIP.1, ensuring that erasure of the specific computer files/free space is made on the storage device.
O.Report	The security objective is met by FAU_GEN.1, ensuring that records of the erasure is audited, providing the verification of the erasure documenting success or failure, the date erasure was performed, the erasure standard used and information about the content that was erased.

6.2.3 Dependency analysis between security functional components

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives.

SFR	Dependencies	Fulfilment of the security objective
FDP_RIP.1	No dependencies	–
FAU_GEN.1	FPT_STM.1	No, this is provided by OE.Time

6.3 Security Assurance Requirements

The security assurance requirements of this Security are those defined for the assurance level EAL2 augmented with ALC_FLR.2.

Assurance Class	Assurance Components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.2 Basic functional specification
	ADV_TDS.1 Basic design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures

ALC: Life-cycle support	ALC_CMC.2 Use of a CM system
	ALC_CMS.2 Use of a CM system
	ALC_DEL.1 Delivery procedures
	ALC_FLR.2 Flaw reporting procedures (augmentation)
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
ATE: Tests	ATE_COV.1 Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

6.4 Security assurance requirements rationale

Dependencies within the EAL package selected (EAL2) for the security assurance requirements have been considered by the authors of CC Part 3 and are not analysed here again. The augmentation by flaw remediation, ALC_FLR.2, has no dependencies on other requirements. The security functional requirements in this Security Target do not introduce dependencies on any security assurance requirement; neither do the security assurance requirements in this Security Target introduce dependencies on any security functional requirement.

The EAL2 level was also deemed sufficient because this will provide a necessary assurance for a product that is operated in an environment that is not directly exposed to external attackers, but still able to resist attacker with basic attack potential.

The assurance requirements of the EAL2 package provides a full Security Target and requires an analysis using a functional and interface specification and a basic description of the architecture of the TOE, which would give sufficient confidence in the design and architecture and for the evaluator to perform an analysis of the design and architecture for the vulnerability analysis.

7 TOE Summary Specification

7.1 Security Functions and Associated Security Functional Requirements

The TOE summary specification identifies the security functions that the TOE implements to meet the requirements defined in chapter 6 of the Security Target.

The table below shows which SFRs are satisfied by each TSF.

Security Functions	Security Functional Requirements
SF.Erase_Data	FDP_RIP.1 - Subset residual information protection
SF.Report_Results	FAU_GEN.1 - Audit data generation

7.2 SF.Erase_Data

To fulfil the requirements of FDP_RIP.1, the TOE erases the data that constitutes a file or the unallocated space (Free Disc Space Erasure) on a target storage device by overwriting it with the selected overwrite pattern. A file erasure operation consists of a series of steps to access a file, get its size, overwrite the content and finally delete it from file system. In the case of a journaling file system, such as Windows NTFS, previous versions and its meta-data may also be erased at the request of the user (Previous Versions Erasure). This functionality is the same both for local erasure of files and folders as well as when remotely executed in storage areas across a given network. Table 2 describes supported erasure standards. The TOE erase functionality is presented to the user as:

- Files or Folders Erasure (Previous versions can be included)
- System Files Erasure
- Recycle Bin Erasure
- Free Disc Space Erasure

7.3 SF.Report_Results

The reporting functionality of the TOE implements the FAU_GEN.1 requirement. Reports that are designed to meet requirements for an audit of erased data are generated every time a file or drive free space is erased. The TOE receives a reliable date and time stamp, which is taken from the Windows Operating System.

The report contains the date and time that the actions were performed along with an indication of success or failure for the erasure events taking place. The report contains the following details:

- Erasure results
- Operation – how many files were erased (e.g. Erasing 2 file(s))
- Duration of the erasure operation
- The method used to overwrite the files
- The target files erased
- The size of the files being erased
- If previous versions were erased or not
- OS name:
- OS version:
- Computer name:
- Windows User name:
- Report UUID:
- Report date:
- Blancco File Eraser software version:

8 Abbreviations and Terms

This Security Target uses following abbreviations and terms.

Term	Description
ATA	Magnetic media interface specification. Also known as “IDE” – Integrated Drive Electronics.
Batch file	A batch file is a text file consisting of a sequence of commands to be implemented by the command interpreter (computer program).
BFE	Blancco File Eraser – a software application for erasing individual files and free space on a storage device. The BFE is also the TOE.
MSI package	The Microsoft Windows Installer package is an .msi file that contains explicit instructions about installing and removing specific applications.
CLI	Command line interface. The line on the display screen where a command is expected. Generally, the command line is the line that contains the most recently displayed command prompt.
GUI	Graphical user interface. The desktop app interface using the Microsoft Windows graphical interface.
HDD	Hard Disk Drive
HTML	HTML, which stands for Hyper Text Markup Language, is the predominant markup language for web pages. It provides a possibility to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists etc as well as for links, quotes, and other items.
Parameters	A parameter is the same as a command line argument. The argument essentially communicates how the program should perform and execute the commands.
Shred	A legacy term for erasing data securely. Means the same thing than “erase”.
Slack space	When a computer file does not need all the space it has been allocated, slack space is the part of the area storage that is left over and may contain remnant data from previous use.
SSD	A Solid State Drive (SSD) is a storage device that uses solid state memory to store persistent data.
Windows registry	Windows registry is a database used within Windows operating systems that stores configurations and option settings.
XML	eXtensible Markup Language is a markup language that defines a set of rules for interpreting documents.
ST	Security Target