White Canyon
WipeDrive Version 6.1
Security Target

Version 1.0
January 25, 2011

Prepared for:

White Canyon Inc.

207 E. 860 S.
Orem, UT 84058

Prepared by:

Booz Allen Hamilton

Common Criteria Testing Laboratory
900 Elkridge Landing Road, Suite 100
Linthicum, MD 21090-2950

Table of Contents

List of Figures

List of Tables

1 Security Target Introduction

This chapter presents the Security Target (ST) identification information and an overview. An ST contains the Information Technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by the TOE.

1.1 ST Reference

This section provides information needed to identify and control this ST and its Target of Evaluation. This ST targets Evaluation Assurance Level 4.

1.1.1 ST Identification

ST Title:	White Canyon Inc. WipeDrive Version 6.1
ST Version:	1.0
ST Publication Date:	January 25, 2011
ST Author:	Booz Allen Hamilton

1.1.2 Document Organization

Chapter 1 of this ST provides identifying information for the White Canyon WipeDrive version 6.1. It includes an ST Introduction, ST Reference, ST Identification, TOE Reference, TOE Overview, and TOE Type.

Chapter 2 describes the TOE Description, which includes the physical and logical boundaries.

Chapter 3 describes the conformance claims made by this ST.

Chapter 4 describes the Security Problem Definition as it relates to threats, Operational Security Policies, and Assumptions met by the TOE.

Chapter 5 identifies the Security Objectives of the TOE and the Operational Environment.

Chapter 6 describes the Extended Security Functional and Assurance Requirements.

Chapter 7 describes the Security Functional Requirements (SFRs).

Chapter 8 describes the Security Assurance Requirements (SARs).

Chapter 9 is the TOE Summary Specification (TSS), a description of the functions provided by White Canyon Inc. WipeDrive version 6.1 to satisfy the security functional and assurance requirements.

Chapter 10 is the TOE Summary Specification Rationale and provides a rationale, or pointers to a rationale, for security objectives, assumptions, threats, requirements, dependencies, and PP claims.

Chapter 11 is the Security Problem Definition Rationale and provides a rationale for the chosen EAL, any deviations from CC Part 2 with regards to SFR dependencies, and a mapping of threats to assumptions, objectives, and SFRs. It also identifies the items used to satisfy the Security Assurance Requirements for the evaluation.

1.1.3 Terminology

This section defines the terminology used throughout this ST. The terminology used throughout this ST is defined in Table 1-2: Terminology Definitions. This table is to be used by the reader as a quick reference guide for terminology definitions.

Terminology	Definition
Administrator	Any user of the TOE who maintains physical possession of the WipeDrive application
Administrator Definable Wipe Pattern	A concatenation of static primitives that is not persistent between boots.
ATA HPA	ATA Host Protected Area Refers to as a hidden protected area that is a section of a hard drive that is not normally visible to an Operating System
Kernel	The central component for most Operating Systems (in this case, UNIX) that is primarily responsible for starting and stopping programs, handling the file system, as well as other low level tasks most programs share.
LAB28/LBA48	A common scheme used for specifying the location of blocks of data stored on computer storage devices, generally secondary storage systems such as hard disks. LBA48, in particular, refers to a logical block address that is 28- or 48-bits wide, resulting in a disk size limit.
LiveCD	A Linux-based compact disc based on the Gentoo meta-distribution which is configured to start WipeDrive upon booting up.
Log/Logging	Synonymous with audit/auditing
Preboot eXecution Environment (PXE)	An environment to boot computers using a network interface independently of available data storage devices (e.g. hard disks) or installed Operating Systems.
White Canyon	Vendor
WipeDrive	Product

Table 1-1: Terminology Definitions

1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 1-3: Acronym Definitions. This table is to be used by the reader as a quick reference guide for acronym definitions.

Acronym	Definition
ATA	Advanced Technology Attachment
BIOS	Basic Input/Output System
DCO	Device Configuration Overlay
DHCP	Dynamic Host Configuration Protocol
GNU	*Recursive acronym for GNU's Not Unix*
HPA	Host Protected Area
JSON	JavaScript Object Notation
LBA	Logical Block Addressing
OS	Operating System
PXE	Preboot eXecution Environment
RPC	Remote Procedure Call Protocol
SCSI	Small Computer System Interface
UI	User Interface

Table 1-2: Acronym Definitions

1.1.5 References

[1] White Canyon WipeDrive User Guide

[2] White Canyon WipeDrive Admin Guide

1.2 TOE Reference

White Canyon Inc. WipeDrive Version 6.1

1.3 TOE Overview

This Security Target (ST) defines the Information Technology (IT) security requirements for the White Canyon WipeDrive. White Canyon WipeDrive is a Disk Sanitizing tool that permanently erases hard drive data, operating systems, program files, and all other file data from a system. WipeDrive also provides users with the ability to permanently delete all partitions and drive formats previously configured. The TOE provides 14 disk wipe functions:

- Standard Overwrite
- US Army AR380-19
- US Air Force System Security Instruction 5020
- US DoD 5220.22-M 3-pass

- US DoD 5220.22-M 7-pass
- US Navy Staff Office Publication P-5329-26
- US National Computer Security Center TG-025
- Australian Defense Signals Directorate ACSI-33 (X0-PD)
- Australian Defense Signals Directorate ACSI-33 (X1-P-PD)
- Canadian RCMP TSSIT OPS-II Standard Wipe
- CIS GOST P50739-95
- GB HMG Infosec Standard #5 Baseline
- GB HMG Infosec Standard #5 Enhanced
- German VSITR

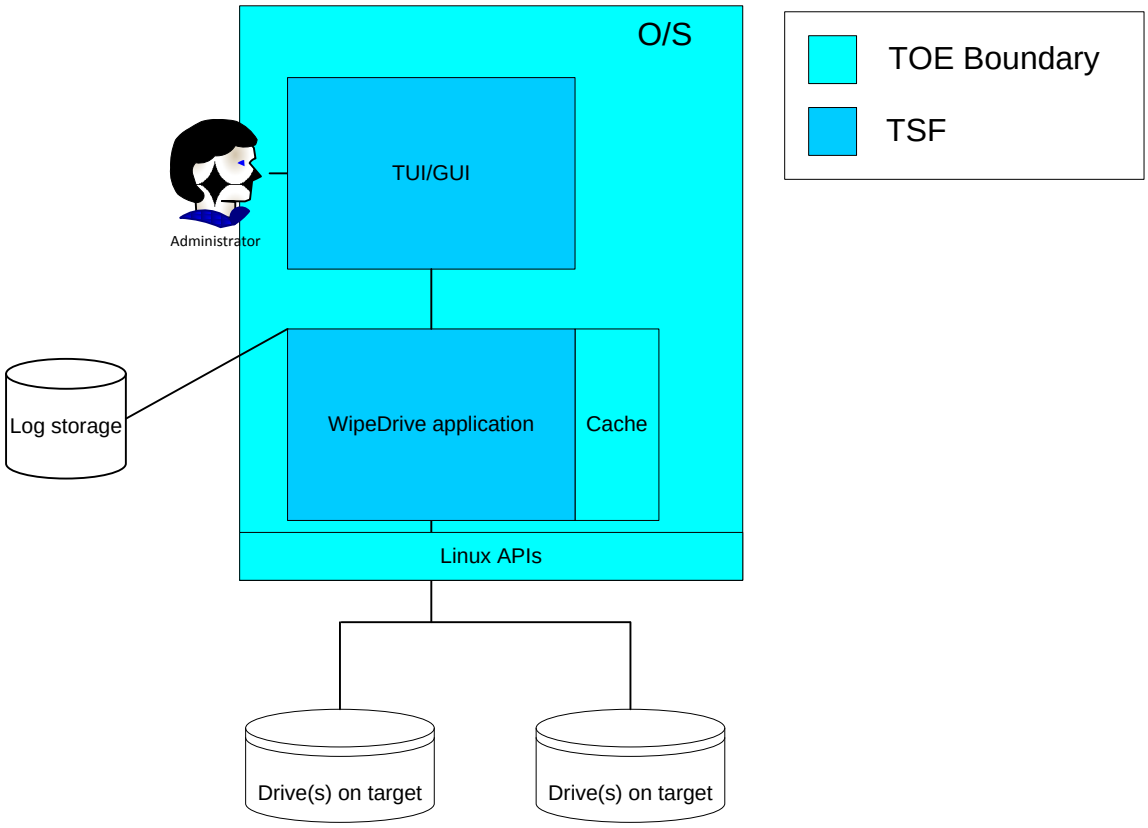
For more information on these wipe methods, please refer to Figure 9-1.

All wipe functions overwrite disk storage to ensure no residual data remains. After the sanitization process has been completed, an audit log is created which compiles verifications that the information contained on the hard drive was in fact erased.

The TOE:

- Is a Linux based OS booted from a LiveCD that resides in memory
- Is a data protection and erasure tool that permanently wipes data from ATA- and SCSI-block devices
- Allows users to create an audit log to capture verifications of the success or failure of hard drive erasure events
- Has the ability to wholly erase Operating Systems, program files, and all file data
- Utilizes user interfaces to allow administrators to graphically see the progress of probing, scanning, and erasure events
- Enables administrators to view sector data

Figure 1-1: TOE Boundary



As shown in Figure 1-1, administrators access the TUI or GUI in order to run the executable file for the WipeDrive application. Once the WipeDrive application has been executed, the cache stores data about scanned and probed devices in order to display the data to users. Scanning and probing are both performed during the initialization of the TOE. The WipeDrive application performs a scanning operation to discover attached devices. For each device that is discovered, a probe operation is run to enumerate device information

The only users of the TOE are referred to as administrators. Administrators, whether through the GUI or TUI, can execute commands to wipe drives by using the administrator definable wipe patterns. Verification of the success or failure of the wipe event is sent to the UI the user is currently using. Also, the audit log data collected from the wipe event is stored in/on a log storage device, which can be a portable flash/thumb drive, FTP server, MySQL database, or other media storage device.

1.4 TOE Type

The TOE type for White Canyon WipeDrive version 6.1 is Sensitive Data Protection. Sensitive Data Protection is defined by CC as “the implementation of administrative, technical, or physical measures to guard against the unauthorized access to data”.

Sensitive Data Protection is the most appropriate designation given to the TOE from the list of TOE types made available by CCEVS. By completely and permanently erasing sensitive data from a given system, the TOE protects a user's data from being unwillingly disseminated and thus, giving individuals unauthorized access to data.

2 TOE Description

2.1 Evaluated Components of the TOE

The TOE provides for 4 distinct components that are included in the evaluated configuration.

2.1.1 WipeDrive Application

The WipeDrive application serves as a single executable file that is primarily responsible for:

- scanning the system for devices that can be erasure targets
- probing the discovered devices for capabilities
- erasing the devices, and performing related operations (such as removing ATA HPA or DCO areas)
- producing progress event messages for consumption by a UI for display to the user
- producing result messages for consumption by a UI and/or logging facilities

Note: Only a single WipeDrive application will be able to run on any single host at any one time.

2.1.2 User Interfaces (UI)

The user interfaces serves as the physical interfaces where controls are used to operate one or more instances of the back-end, each on a distinct host. The interfaces that are included in the evaluated configuration are:

- **TUI** – A text-based UI, run on the same host as the back-end. It is used primarily for systems that do not have framebuffer support – which is typical on many architectures other than x86.
- **GUI** – A graphical UI that is run on the same host as the back-end. This will be the default interface for x86 machines that framebuffer can be accessed.

2.1.3 Cache

The cache stores data about scanned and probed devices in order to display that information to users. The cache component is also responsible for the auditing of data that is collected. The audit data received from the WipeDrive application is stored in the cache, which sends a copy of the same data back to both the Log Storage component and interface the user is currently using.

2.1.4 Linux APIs

Linux APIs provide a logical interface between the application and the target drive(s). For example, when the TOE scans a disk, it relies on Linux to gather the data. This is a built-in function of the Operating System.

2.2 Components in the Operational Environment

2.2.1 Log Storage

The Log Storage component is responsible for the storage of audit information. Log Storage refers to any external device with a file system that the TOE can access. Examples of these are a USB drive or a separate hard disk or partition upon the local machine being wiped.

2.2.1.1 Log Storage Formats

The Log Storage component supports several formats:

- **Regular** – a plain-text synopsis (free-form) of what activities were attempted and their result
- **Delimited** a plain-text file, delimited by a tilde '~', of what activities were attempted and their result in a tabular format
- **XML –** an XML file with a corresponding CSS file that contains both the activities that were attempted and their result as well as a brief system inventory harvested via invoking the lshw Linux utility
- **SQL –** a SQL query that inserts log data into a target database

2.3 Excluded From the TOE

The following optional products and components can be integrated with WipeDrive but are NOT included in the evaluated configuration. They provide no added security related functionality. They are separated into three categories: not installed, installed but requires a separate license, and installed but not part of the TSF.

2.3.1 Not Installed

There are no additional components for WipeDrive version 6.1 that are not installed.

2.3.2 Installed but Requires a Separate License

These components are installed with WipeDrive v6.1, but require a separate license and are therefore not included in the TOE boundary.

- **MediaWiper** – erases all data on partitions, memory cards (compact flash, MMC, memory sticks, smart media, IBM Micro-drives, etc.), USB drives, diskettes, and zip disks. MediaWiper however is outside of the scope of this evaluation because only the wiping of ATA and SCSI-block devices are being evaluated. Additionally, this component requires an additional license which is not provided in the evaluated configuration.

2.3.3 Installed But Not Part of the TSF

These components are installed with WipeDrive v6.1, but are not a part of the TSF.

- **Network GUI** – A GUI that is run on any machine (usually on a PXE server) that can communicate with and control any number of WipeDrive applications running on other hosts. This is not part of the TSF because the communications are not secured by WipeDrive and the threat of being able to wipe remote hosts introduces an increased amount of risk.

Note that while the Network GUI capabilities are technically "installed" as they are part of the LiveCD media, the act of deploying WipeDrive as a network-capable product requires a deliberate configuration effort on a dedicated server. The standard usage of the TOE via single session LiveCD instances do not run a risk of "accidentally" utilizing or deploying this functionality.

2.4 Physical Boundary

The TOE includes the following hardware and software components:

2.4.1 Hardware Components

The following table identifies WipeDrive's hardware components and indicates whether or not each component is in the TOE.

TOE or Environment	Component	Description
TOE	LiveCD	A 156 MB Linux-based disc based on the Gentoo meta-distribution which is configured to start WipeDrive upon booting up. Contained on the disc is an executable file that takes initial input parameter, modifies the boot loader in order to add a Gentoo RAM disc, then reboots the system into the disc where the program is run.
Environment	Target(s) on a Drive	Operating Systems: Windows Mac PC running Linux,

		UNIX CPU – 156 MHz RAM – 64 MB Other: • VGA or higher video support • ATA- or SCSI-block device that has been identified as a candidate for erasure.
	Log Storage	Location in which the audit data is stored and is located separately from the TOE. The data can be stored on any form of file storage medium.
	External Server	A physical server that can utilize FTP or MySQL to optionally be used to store logs of erasure events in lieu of the log storage file if desired.

Table 3 – Hardware Requirements for the TOE

Note: Newer computers often have a BIOS feature to 'enable' or 'disable' devices in the boot order list. It is important that the CD drive be enabled for WipeDrive to wipe the drive. Even if the boot order is properly set to boot from CD before the hard drive, and the CD is a valid copy, WipeDrive will only run if the CD drive is 'enabled' in BIOS.

2.4.2 Memory Requirements

The following table identifies WipeDrive's memory requirements for the UNIX and Windows Operating System.

Component	Operating System	
	UNIX Variant	Windows
Memory (RAM)	64 MB	64 MB

Table 4 – Hardware Requirements for the TOE

Note: In addition to the above requirements, storage space is needed for the log file. This could be stored on any form of file storage medium. The log file will contain the type of wipe, the size of the hard drive, and the timestamp of the wipe.

2.4.3 Software Components

The following table identifies WipeDrive's software components and indicates whether or not each component is in the TOE.

TOE or Environment	Component	Description
TOE	WipeDrive Version 6.1	The disk erasure tool currently being evaluated.
	TUI / GUI	Receives user commands in order to display options to users to run specific wipe operations. The TUI and GUI run on the same host as the WipeDrive application (back-end). The TUI is remote while the GUI is local.
	Cache	Stores data about scanned and probed devices for the purpose of displaying the information to users. It also allows for auditing of the data to a defined log storage device.
	Linux API	Provides a logical interface between the WipeDrive software and target(s) on a drive.
Environment	Log Storage file	A flat file where verification of the success or failure of erasure events are stored.
	FTP Server	A remote FTP server which may be used to receive log data.
	MySQL Server	A remote database server which may be used to receive log data.

Table 5 – Hardware Requirements for the TOE

2.5 Logical Boundary

The logical boundaries of the TOE are described in the terms of the security functionalities that the TOE provides to the systems that utilize this product for information flow control.

The logical boundary of the TOE will be broken down into four security classes: Security Audit, Security Management, Disk Erasure, and User Data Protection. Listed below are the security functions with a listing of the capabilities associated with them:

2.5.1.1 Security Audit

The TOE generates and captures audit data which is used to provide further verification that an erasure event has occurred. Audit logs containing verification data (either denoting a success of failure) is stored in the Log Storage component. The resulting output of a wipe operation is recorded as a JSON-RPC within the user interface and is displayed in an easily interpretable manner. Other events that are recorded include the start-up and shutdown of the audit functions and various parameters related to the TOE's probe, scan, and subsequent erasure of targets on a drive. All audit operations can be associated with the administrator who performed that event. The TOE saves the audit events in a user-readable format outside of the TOE but is not responsible for facilitating the viewing of audit records except for a review of wipe results immediately following a wipe operation.

2.5.1.2 Security Management

The only users of the TOE are referred to as administrators. Administrators are the individuals who maintain physical access to the WipeDrive application, and, as a result, possess several management capabilities. Administrators are able to specify the location for audit storage (in the Log Storage component), specify the format in which this data is stored, create, run, view, or delete an administrator definable wipe pattern, scan for devices, view sector data, and get device info for all devices previously scanned.

The TOE is equipped to operate via various interfaces which are made available to administrators. The administrators of the TOE utilize these interfaces to perform the management functions listed above. The primary purposes of these interfaces are to:

1. Allow commands defined by the TOE to be invoked on the attached WipeDrive application;
2. Visually display the status of the attached WipeDrive application by interpreting the responses and notifications received; and
3. Create audit logs according to the user's preferences. The logs can be stored on any form of media that the user desires, e.g a thumb drive or on an FTP server).

The TOE can be operated via two user interfaces – a TUI or GUI. The TUI runs on the same host as the WipeDrive application (back-end). It is used primarily for systems that do not have framebuffer support – which is typical on many architectures other than x86. The GUI is also run on the same host as the back-end. This will be the default interface for x86 machines that framebuffer can be accessed.

2.5.1.3 Disk Erasure

The TOE is able to perform three distinct operations under the guise of Disk Erasure – scanning of devices, probing of devices, and the erasure of the devices. Scanning and probing are both performed during the initialization of the TOE while the probe operation is run each time a device is discovered. Administrators, whether via the GUI or TUI, can execute commands to wipe drives. The wipe command applies the administrator definable wipe pattern to each selected disk instance, which performs the overwrite operations directly on the disk.

2.5.1.4 User Data Protection

The TOE provides for the erasure of residual information. This erasure is initiated at the user-facing interfaces and requires communication with the information repository (disk). The erasure of residual information is performed when a deviceOp instance is executed – which is a direct result an administrator definable wipe pattern. No residual information will reside in the RAM subsequent to a wipe event.

3 Conformance Claims

3.1 CC Version

This ST is compliant with *Common Criteria for Information Technology Security Evaluation*, CCMB-2007-09-004, Version 3.1 Revision 3, July 2009

3.2 CC Part 2 Extended

This ST and Target of Evaluation (TOE) is Part 2 extended for EAL4 to include all applicable NIAP and International interpretations through 14 August 2009.

3.3 CC Part 3 Augmented

This ST and Target of Evaluation (TOE) is Part 3 augmented plus flaw remediation for EAL4 to include all applicable NIAP and International interpretations through 14 August 2009.

3.4 PP Claims

This ST does not claim Protection Profile (PP) conformance.

3.5 Package Claims

This TOE has a package claim of EAL 4.

3.6 Package Name Conformant or Package Name Augmented

This ST and Target of Evaluation (TOE) is conformant to EAL package claims augmented with ALC_FLR.2 and ASE_TSS.2**.**

3.7 Conformance Claim Rationale

There is no Conformance Claim rationale for this ST.

4 Security Problem Definition

4.1 Threats

The TOE itself has threats and the TOE is also responsible for addressing threats to the environment in which it resides. The assumed level of expertise of the attacker for all the threats is moderately sophisticated. The following are threats addressed by the TOE.

T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.

T.AUDIT_COMPROMISE A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.

T.AUDIT_FAILURE A malicious user or process failure may cause the TOE to fail to record or improperly record audit data, thus masking a user's action.

T.RESIDUAL Any person with access to a target environmental resource can access residual data, either due to a wipe operation being incomplete or a completed wipe operation being insufficient.

T.UNAUTH An unauthorized user obtains the physical medium which contains the TOE and uses it to perform a wipe operation against an environmental resource which there has been no authorization to wipe.

4.1 Organizational Security Policies

The TOE addresses the organizational security policies described below.

P.REUSE All drive data must be securely erased to allow the reuse of drives.

P.STANDARD The TOE will be used to securely erase drive data in conformance with the standards of the organization.

4.2 Assumptions

The specific conditions listed in this section are assumed to exist in the environment in which the TOE is deployed. These assumptions are necessary as a result of practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

4.2.1 Personnel Assumptions

A.ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.

A.NOEVIL Users of the TOE are not careless, wilfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.

4.2.2 Physical Assumptions

A.LOCAL The TOE will be loaded onto the same physical machine as the target resource so that commands are not exposed over the network.

A.LOCATE The physical medium which contains the TOE will be located in a secure location and physical custody is maintained by one or more authorized administrators.

4.2.3 Logical Assumptions

A.PATCHES Administrators of the Operational Environment exercise due diligence to acquire updated versions of the TOE and patch the Operational Environment (e.g., OS and database) so they are not susceptible to attacks resulting in malfunction of the TOE or associated audit data.

5 Security Objectives

5.1 Security Objectives for the TOE

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

O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.

O.ERASE The TOE will provide measures for erasing data contained on block devices on a target system as well as sufficient assurance that the desired data was erased and that the erasure method was sufficient for permanent erasure.

O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor the set of disk wipe patterns made available to the TOE and the storage of audit data generated by the TOE.

O.ROBUST_ADMIN_GUIDANCE The vendor will provide administrators with the necessary information for secure delivery and management of the TOE.

5.1.1 Security Objectives for the Operational Environment of the TOE

The following security objectives for the Operational environment of the TOE must be satisfied in order for the TOE to fulfill its security objectives.

OE.ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.

OE.LOCAL The TOE will be loaded onto the same physical machine as the target resource so that commands are not exposed over the network.

OE.LOCATE	The physical medium which contains the TOE will be located in a secure location and physical custody is maintained by one or more authorized administrators.
OE.NOEVIL	Users of the TOE are not careless, wilfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.
OE.PATCHES	Administrators of the Operational Environment exercise due diligence to acquire updated versions of the TOE and patch the Operational Environment (e.g., OS and database) so they are not susceptible to attacks resulting in malfunction of the TOE or associated audit data.
OE.SYSTIME	The operating environment will provide reliable system time.

6 Extended Security Functional Requirements

6.1 Extended Security Functional Requirements for the TOE

Security Function	Security Functional Components
Disk Erasure	FDE_SCN_EXT.1 Scan of Devices
	FDE_PRB_EXT.1 Probe of Devices
	FDE_ERS_EXT.1 Erasure of Devices

Table 6-1: Extended Security Functional Requirements for the TOE

6.1.1 Class FDE: Disk Erasure

The FDE family defines requirements for the scanning, probing, and erasure of devices. This family identifies the types of disk erasures capable of being performed (FDE_ERS_EXT.1) by enumerating the methods made available to users of the TOE. In addition to listing the wipe patterns the TOE can perform, the FDE_SCN_EXT.1 requirement scans a system for potential erasure targets. The FDE_PRB_EXT.1 requirement communicates with the devices that were located subsequent to the scan operation in order to discover that target's parameters. Both the FDE_SCN_EXT.1 and FDE_PRB_EXT.1 extended requirements are performed automatically as the WipeDrive application is run.

FDE_SCN_EXT.1 Scan of devices, requires the TSF to discover storage devices on a system based on some criteria.

FDE_PRB_EXT.1 Probe of devices, requires the TSF to identify certain parameter values on the devices it has discovered from the scanning process.

FDE_ERS_EXT.1 Erasure of devices, requires the TSF to erase devices which have been scanned and probed according to some specific overwrite sequence.

Management: FDE_SCN_EXT.1, FDE_PRB_EXT.1, FDE_ERS_EXT.1

The following actions could be considered for management actions in FMT:

a) Initiating a scan, probe, or erasure
b) Viewing the results of a scan (or the combination of a scan and a probe)
c) Managing (create/view/edit/delete) administrator definable wipe patterns

Audit: FDE_ERS_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Not Specified: target device erased, identify of individual performing erasure, type of erasure performed, data about the device, errors encountered during erasure operation.

The FDE_SCN_EXT.1, FDE_PRB_EXT.1, and FDE_ERS_EXT.1 requirements are extended because there is no security functional requirement described in CC Part 2 that directly applies to the primary functionality of the TOE. The most closely related security functional requirement in CC Part 2 is the FDP_RIP.1 requirement, which speaks to the allocation and deallocation of resources on a given system. This requirement does indeed speak to a portion of the primary functionality of the TOE, however it does not encapsulate all of the TOE's purpose. It is necessary to introduce the Disk Erasure family in order to fully capture the most important functionality of the TOE, i.e. the scanning of devices, the probing of devices, and the erasure of targets on a given drive.

6.1.1.1 FDE_SCN_EXT.1 Scan of Devices

Hierarchical to: No other components.

FDE_SCN_EXT.1.1 The TSF shall be able to discover all [***ATA and SCSI block devices***] on a system as potential erasure targets.

Dependencies: No dependencies

6.1.1.2 FDE_PRB_EXT.1 Probe of Devices

Hierarchical to: No other components.

FDE_PRB_EXT.1.1 The TSF shall communicate with devices that are discovered as the result of the scan in order to determine the following parameters for the device:

- [***Model Name***
- ***Name of manufacturer***
- ***Serial Number***
- ***Drive Capacity***]

Dependencies: FDE_SCN_EXT.1 Scan of devices

6.1.1.3 FDE_ERS_EXT.1 Erasure of Devices

Hierarchical to: No other components.

FDE_ERS_EXT.1 The TSF shall be able to erase devices that are discovered by a scan using one of the following sequences:

- Standard Overwrite
- US Army AR380-19
- US Air Force System Security Instruction 5020
- US DoD 5220.22-M 3-pass
- US DoD 5220.22-M 7-pass
- US Navy Staff Office Publication P-5329-26
- US National Computer Security Center TG-025
- Australian Defense Signals Directorate ACSI-33 (X0-PD)
- Australian Defense Signals Directorate ACSI-33 (X1-P-PD)
- Canadian RCMP TSSIT OPS-II Standard Wipe
- CIS GOST P50739-95
- GB HMG Infosec Standard #5 Baseline
- GB HMG Infosec Standard #5 Enhanced
- German VSITR

Dependencies: FDE_SCN_EXT.1 Scan of Devices
FDE_PRB_EXT.1 Probe of Devices

6.2 Proper Dependencies

There are no dependencies that were derived from CC Part 2.

6.3 Extended Security Assurance Requirements

There are no extended Security Assurance Requirements in this ST.

7 Security Functional Requirements

7.1 Security Functional Requirements for the TOE

Security Function	Security Functional Components
Security Audit	FAU_GEN.1 Audit Data Generation
	FAU_GEN.2 User Identity Association
	FAU_SAR.1 Audit Review
Security Management	FMT_SMF.1 Specification of Management Functions
User Data Protection	FDP.RIP.1 Residual Information Protection

Table 7-1: Security Functional Requirements for the TOE

7.1.1 Class FAU: Security Audit

7.1.1.1 FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

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 [**not specified**] level of audit; and

c) [**erasure events**]

Dependencies: FPT_STM.1 Reliable Time Stamps

Application Note: The audit functions cannot be disabled and are run automatically.

Application Note: System time is provided by the BIOS hardware clock. The BIOS clock is synchronized with the system clock contained on the Linux LiveCD.

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,

[***WipeDrive version number, drive model information, serial number of physical drive, current user (user-defined, name of person performing the wipe), computer ID (user-defined, name of the drive or system), type of operation performed (***administrator definable wipe pattern***), number of overwrites performed, date and time operation was completed, total elapsed time, operation result, total number of disk sector read/write errors, if any, total uncleaned or unreadable disk sectors, if any***].

Dependencies: No dependencies

Application Note: Users are self-identified when performing operations therefore the TOE does not ensure correct authentication.

7.1.1.2 FAU_GEN.2 User Identity Association

Hierarchical to:	No other components.
FAU_GEN.2.1	The TSF shall be able to associate each auditable event with the identity of the user that caused the event.
Dependencies:	FAU_GEN.1 Audit Data Generation FIA_UID.1 Timing of Authentication

7.1.1.3 FAU_SAR.1 Audit Review

Hierarchical to:	No other components.
FAU_SAR.1.1	The TSF shall provide [***administrators***] with the capability to read [***information about the most recent wipe performed during the active session***].
FAU_SAR.1.2	The TSF shall provide the audit records in a manner suitable for the user to interpret the information.
Dependencies:	FAU_GEN.1 Audit Data Generation

7.1.2 Class FMT: Security Management

7.1.2.1 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

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

- ***Specify location for audit storage***
- ***Specify format for log storage***
- ***Create an administrator definable wipe pattern***
- ***Delete an administrator definable wipe pattern***
- ***Run an administrator definable wipe pattern***
- ***View all administrator defined wipe patterns***
- ***Scan for devices***
- ***View sector data***
- ***Get device info for all devices previously scanned***

Dependencies: No dependencies.

Application Note: The security management functions can be initiated either via GUI or TUI.

Application Note: Disk scanning and probing are processes that are started automatically upon initialization of the TOE.

7.1.3 Class FDP: User Data Protection

7.1.3.1 FDP.RIP.1 Residual Information Protection

Hierarchical to: No other components

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a ~~resource~~ block device upon which the TSF is acting is made unavailable upon [***allocation of the resources to***] the following objects: [***any object created as the result of an administrator definable wipe pattern***].

Dependencies: No dependencies

Application Note: The administrator definable wipe pattern determines the target resource(s) and the method(s) by which its previous information content will be made unavailable. Residual data will be removed from system memory as well as the target resource(s)

7.2 Operations Defined

The notation, formatting, and conventions used in this security target (ST) are consistent with version 3.1 of the Common Criteria for Information Technology Security Evaluation. All of the components in this ST are taken directly from Part 2 of the CC except the ones noted with "_EXT" in the component name. Font style and clarifying information conventions were developed to aid the reader.

The CC permits four functional component operations: assignment, iteration, selection, and refinement to be performed on functional requirements. These operations are defined in Common Criteria, Part 1 as:

7.2.1 Assignments Made

An assignment allows the specification of parameters and is specified by the ST author in [***italicized bold text***].

7.2.2 Iterations Made

An iteration allows a component to be used more than once with varying operations and is identified with the iteration number within parentheses after the short family name, FAU_GEN.1(1), FAU_GEN.1(2).

7.2.3 Selections Made

A selection allows the specification of one or more items from a list and is specified by the ST author in [**bold text**].

7.2.4 Refinements Made

A refinement allows the addition of details and is identified with "Refinement:" right after the short name. ~~The old text is shown with a strikethrough~~ and ***the new text is specified by italicized bold and underlined text***.

8 Security Assurance Requirements

This section identifies the Security Assurance Requirement components met by the TOE. These assurance components meet the requirements for EAL4 augmented with ALC_FLR.2 and ASE_TSS.2.

8.1 Security Architecture

8.1.1 Security Architecture Description (ADV_ARC.1)

ADV_ARC.1.1D:	The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.
ADV_ARC.1.2D:	The developer shall design and implement the TSF so that it is able to protect itself from tampering by un-trusted active entities.
ADV_ARC.1.3D:	The developer shall provide a security architecture description of the TSF.
ADV_ARC.1.1C:	The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.
ADV_ARC.1.2C:	The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.
ADV_ARC.1.3C:	The security architecture description shall describe how the TSF initialization process is secure.
ADV_ARC.1.4C:	The security architecture description shall demonstrate that the TSF protects itself from tampering.
ADV_ARC.1.5C:	The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.
ADV_ARC.1.1E:	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.1.2 Functional Specification with Complete Summary (ADV_FSP.4)

ADV_FSP.4.1D	The developer shall provide a functional specification.
ADV_FSP.4.2D	The developer shall provide a tracing from the functional specification to the SFRs.
ADV_FSP.4.1C	The functional specification shall completely represent the TSF.

ADV_FSP.4.2C	The functional specification shall describe the purpose and method of use for all TSFI.
ADV_FSP.4.3C	The functional specification shall identify and describe all parameters associated with each TSFI.
ADV_FSP.4.4C	The functional specification shall describe all actions associated with each TSFI.
ADV_FSP.4.5C	The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.
ADV_FSP.4.6C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
ADV_FSP.4.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.4.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

8.1.3 Implementation Representation of the TSF (ADV_IMP.1)

ADV_IMP.1.1D	The developer shall make available the implementation representation for the entire TSF.
ADV_IMP.1.2D	The developer shall provide a mapping between the TOE design description and the sample of the implementation representation. Content and presentation elements:
ADV_IMP.1.1C	The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.
ADV_IMP.1.2C	The implementation representation shall be in the form used by the development personnel.
ADV_IMP.1.3C	The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence. Evaluator action elements:
ADV_IMP.1.1E	The evaluator *shall confirm* that, for the selected sample of the implementation representation, the information provided meets all requirements for content and presentation of evidence.

8.1.4 Architectural Design (ADV_TDS.3)

ADV_TDS.3.1D	The developer shall provide the design of the TOE.
ADV_TDS.3.2D	The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.
ADV_TDS.3.1C	The design shall describe the structure of the TOE in terms of subsystems.
ADV_TDS.3.2C	The design shall describe the TSF in terms of modules.
ADV_TDS.3.3C	The design shall identify all subsystems of the TSF.
ADV_TDS.3.4C	The design shall provide a description of each subsystem of the TSF.
ADV_TDS.3.5C	The design shall provide a description of the interactions among all subsystems of the TSF.
ADV_TDS.3.6C	The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.
ADV_TDS.3.7C	The design shall describe each SFR-enforcing module in terms of its purpose and interaction with other modules.
ADV_TDS.3.8C	The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, interaction with and called interfaces to other modules.
ADV_TDS.3.9C	The design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules.
ADV_TDS.3.10C	The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.
ADV_TDS.3.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_TDS.3.2E	The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

8.2 Guidance Documents

8.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1D	The developer shall provide operational user guidance.

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

8.3 Lifecycle Support

8.3.1 Authorization Controls (ALC_CMC.4)

ALC_CMC.4.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.4.2D The developer shall provide the CM documentation.

ALC_CMC.4.3D The developer shall use a CM system.

ALC_CMC.4.1C The TOE shall be labeled with its unique reference.

ALC_CMC.4.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.4.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.4.4C The CM system shall provide automated measures such that only authorized changes are made to the configuration items.

ALC_CMC.4.5C The CM system shall support the production of the TOE by automated means.

ALC_CMC.4.6C The CM documentation shall include a CM plan.

ALC_CMC.4.7C The CM plan shall describe how the CM system is used for the development of the TOE.

ALC_CMC.4.8C The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ALC_CMC.4.9C The evidence shall demonstrate that all configuration items are being maintained under the CM system.

ALC_CMC.4.10C The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

ALC_CMC.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.2 CM Scope (ALC_CMS.4)

ALC_CMS.4.1D	The developer shall provide a configuration list for the TOE.
ALC_CMS.4.1C	The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.
ALC_CMS.4.2C	The configuration list shall uniquely identify the configuration items.
ALC_CMS.4.3C	For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ALC_CMS.4.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.3 Delivery Procedures (ALC_DEL.1)

ALC_DEL.1.1D	The developer shall document procedures for delivery of the TOE or parts of it to the consumer.
ALC_DEL.1.2D	The developer shall use the delivery procedures.
ALC_DEL.1.1C	The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.
ALC_DEL.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.4 Identification of Security Measures (ALC_DVS.1)

ALC_DVS.1.1D	The developer shall produce development security documentation.
ALC_DVS.1.1C	The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

8.3.5 Life-cycle Definition (ALC_LCD.1)

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_LCD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.6 Tools and techniques (ALC_TAT.1)

ALC_TAT.1.1D The developer shall identify each development tool being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of each development tool.

ALC_TAT.1.1C Each development tool used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.

ALC_TAT.1.3C The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.

ALC_TAT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.3.7 Flaw reporting procedures (ALC_FLR.2)

ALC_FLR.2.1D	The developer shall document flaw remediation procedures addressed to TOE developers.
ALC_FLR.2.2D	The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.
ALC_FLR.2.3D	The developer shall provide flaw remediation guidance addressed to TOE users.
ALC_FLR.2.1C	The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.2.2C	The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.2.3C	The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.2.4C	The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_FLR.2.5C	The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.
ALC_FLR.2.6C	The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.
ALC_FLR.2.7C	The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.
ALC_FLR.2.8C	The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.
ALC_FLR.2.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

8.4 Security Target Evaluation

8.4.1 Conformance Claims (ASE_CCL.1)

ASE_CCL.1.1D	The developer shall provide a conformance claim.
ASE_CCL.1.2D	The developer shall provide a conformance claim rationale.
ASE_CCL.1.1C	The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.
ASE_CCL.1.2C	The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.
ASE_CCL.1.3C	The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.
ASE_CCL.1.4C	The CC conformance claim shall be consistent with the extended components definition.
ASE_CCL.1.5C	The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.
ASE_CCL.1.6C	The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.
ASE_CCL.1.7C	The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.
ASE_CCL.1.8C	The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

8.4.2 Extended Components Definition (ASE_ECD.1)

ASE_ECD.1.1D	The developer shall provide a statement of security requirements.
ASE_ECD.1.2D	The developer shall provide an extended components definition.

ASE_ECD.1.1C	The statement of security requirements shall identify all extended security requirements.
ASE_ECD.1.2C	The extended components definition shall define an extended component for each extended security requirement.
ASE_ECD.1.3C	The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.
ASE_ECD.1.4C	The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.
ASE_ECD.1.5C	The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.
ASE_ECD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_ECD.1.2E	The evaluator shall confirm that no extended component can be clearly expressed using existing components.

8.4.3 ST Introduction (ASE_INT.1)

ASE_INT.1.1D	The developer shall provide an ST introduction.
ASE_INT.1.1C	The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
ASE_INT.1.2C	The ST reference shall uniquely identify the ST.
ASE_INT.1.3C	The TOE reference shall identify the TOE.
ASE_INT.1.4C	The TOE overview shall summarize the usage and major security features of the TOE.
ASE_INT.1.5C	The TOE overview shall identify the TOE type.
ASE_INT.1.6C	The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.
ASE_INT.1.7C	The TOE description shall describe the physical scope of the TOE.
ASE_INT.1.8C	The TOE description shall describe the logical scope of the TOE.
ASE_INT.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

8.4.4 Security Objectives (ASE_OBJ.2)

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

ASE_OBJ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.5 Security Requirements (ASE_REQ.2)

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C	All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.
ASE_REQ.2.3C	The statement of security requirements shall identify all operations on the security requirements.
ASE_REQ.2.4C	All operations shall be performed correctly.
ASE_REQ.2.5C	Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.
ASE_REQ.2.6C	The security requirements rationale shall trace each SFR back to the security objectives for the TOE.
ASE_REQ.2.7C	The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.
ASE_REQ.2.8C	The security requirements rationale shall explain why the SARs were chosen.
ASE_REQ.2.9C	The statement of security requirements shall be internally consistent.
ASE_REQ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.6 Security Problem Definition (ASE_SPD.1)

ASE_SPD.1.1D	The developer shall provide a security problem definition.
ASE_SPD.1.1C	The security problem definition shall describe the threats.
ASE_SPD.1.2C	All threats shall be described in terms of a threat agent, an asset, and an adverse action.
ASE_SPD.1.3C	The security problem definition shall describe the OSPs.
ASE_SPD.1.4C	The security problem definition shall describe the assumptions about the operational environment of the TOE.
ASE_SPD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.4.7 TOE Summary Specification (ASE_TSS.2)

ASE_TSS.2.1D	The developer shall provide a TOE summary specification.
ASE_TSS.2.1C	The TOE summary specification shall describe how the TOE meets each SFR.
ASE_TSS.2.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
ASE_TSS.2.2C	The TOE summary specification shall describe how the TOE protects itself against interference and logical tampering.
ASE_TSS.2.2E	The evaluator *shall confirm* that the TOE summary specification is consistent with the TOE overview and the TOE description.
ASE_TSS.2.3C	The TOE summary specification shall describe how the TOE protects itself against bypass.

8.5 Tests

8.5.1 Analysis of Coverage (ATE_COV.2)

ATE_COV.2.1D	The developer shall provide an analysis of the test coverage.
ATE_COV.2.1C	The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.
ATE_COV.2.2C	The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.
ATE_COV.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.5.2 Basic Design (ATE_DPT.2)

ATE_DPT.2.1D	The developer shall provide the analysis of the depth of testing.
ATE_DPT.2.1C	The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and SFR-enforcing modules in the TOE design.

ATE_DPT.2.2C	The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.
ATE_DPT.2.3C	The analysis of the depth of testing shall demonstrate that the SFR-enforcing modules in the TOE design have been tested.
ATE_DPT.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.5.3 Functional Tests (ATE_FUN.1)

ATE_FUN.1.1D	The developer shall test the TSF and document the results.
ATE_FUN.1.2D	The developer shall provide test documentation
ATE_FUN.1.1C	The test documentation shall consist of test plans, expected test results and actual test results.
ATE_FUN.1.2C	The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.3C	The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.4C	The actual test results shall be consistent with the expected test results.
ATE_FUN.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

8.5.4 Independent Testing (ATE_IND.2)

ATE_IND.2.1D	The developer shall provide the TOE for testing.
ATE_IND.2.1C	The TOE shall be suitable for testing.
ATE_IND.2.2C	The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

8.6 Vulnerability Assessment

8.6.1 Vulnerability Analysis (AVA_VAN.3)

AVA_VAN.3.1D The developer shall provide the TOE for testing.

AVA_VAN.3.1C The TOE shall be suitable for testing.

AVA_VAN.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.3.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.3.3E The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE.

AVA_VAN.3.4E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Enhanced-Basic attack potential.

9 TOE Summary Specification

The following sections identify the security functions of the TOE. They include Security Audit, Disk Erasure, Security Management, and User Data Protection,

9.1 Security Audit

The TOE generates audit logs which serve as verification of the erasure events that the administrator has performed. The Log Storage component of the TOE is responsible for the erasure of media. The Log Storage component supports several formats:

- **Regular** – a plain-text synopsis (free-form) of what activities were attempted and their result
- **Delimited** – a plain-text file, delimited by a tilde (~), of what activities were attempted and their result in a tabular format
- **XML** – an XML file with a corresponding CSS file that contains both the activities that were attempted and their result as well as a brief system inventory harvested via invoking the lshw Linux utility
- **SQL** – connecting to an external MySql database to store log information

. The TSF writes audit records in a format suitable for a TOE user to view and print the individual Disk Log Files that the TOE records from any other operating system. The TOE provides these in a user-readable format outside of the TOE but is not responsible for facilitating the viewing of those audit records.

9.1.1 Log Files

For verification of the wipe, the software allows logging to a USB drive. The log file will contain the type of wipe, the size of the hard drive and the date and time of the wipe.

Event logs of wiping sessions can be created and saved when using the LiveCD to wipe drives. The log file can be turned on/off and configured from the main WipeDrive menu via one of the available user interfaces. When fully configured, WipeDrive has the ability to log the following 12 items:

1. WipeDrive version number
2. Drive model information
3. Serial number of physical drive
4. Current User (user-defined, name of person performing the wipe)
5. Computer ID (user-defined, name of the drive or system)
6. Type of operation performed
7. Number of overwrites performed
8. Date & Time operation was completed
9. Total elapsed time (HH:MM:SS)

10. Operation Result (Success or Failure)

11. Total number of disk sector read/write errors, if any

12. Total uncleaned or unreadable disk sectors, if any

The log can be found on the media the administrator has chosen to save the data on with the filename: 'LOG.TXT'. A special delimited log is also created on the same media with the filename: 'LOGDEL.TXT' for administrators desiring to maintain the logs in a database. When logging is turned on, additional wipes will append the new data to the end of an existing log file.

9.1.2 Disk Errors

If the TOE reports errors during a wipe or verification, the application has encountered some issue reading or writing to the drive. This means the drive is beginning to fail. If errors are encountered, it is recommended that the computer be rebooted and the wipe process started again. If errors persist and the drive currently in use is used in the future, it is recommended that important data is backed up immediately and frequently after the initial backup. The drive could fail further or completely at any time.

9.2 Disk Erasure

The TOE erases data present by overwriting it with a particular pattern of data, thus eradicating the previous contents of the disk.

9.2.1 Patterns of Wipe Level Definitions

Each wipe pattern adheres to a specific approved standard, including official government and military standards in use today. Specific patterns such as all 'ones' and all 'zeros' are used in various wipe standards as defined below. The implementation of the wipe functions utilized is vendor-asserted. Some wipes are designed to use random data, or to include full verification of each character written. The following list details the 13 types of disk sanitization methods made available to administrators of the TOE:

- Standard Overwrite
    - A single pass overwriting algorithm that overwrites all data with a fixed value (0x00).
- US Army AR380-19
    - A 2-pass overwriting algorithm where the first pass is user defined, and the second pass is the inverse of that user definition.
- US Air Force System Security Instruction 5020

    - An deletion algorithm that first overwrites the target data with a fixed value (0x00), then with another fixed value (0xff), and finally user defined data.

- US DoD 5220.22-M (1 verify)
    - A 3-pass overwriting algorithm where the first pass overwrites with zeroes, the next pass with ones, and the last pass with random bytes.
- US DoD 5220.22-M (3 verifies)
    - A 6-pass overwriting algorithm where the first the first pass overwrites with zeroes, the second pass runs a complete verify, the third pass overwrites with ones, the fourth pass runs a complete verify, the fifth pass overwrites with the number 97, and the 6th pass runs a complete verify.
- US Navy Staff Office Publication P-5329-26
    - A 3-pass overwriting algorithm where the first pass overwrites with zeroes, the next pass with ones, and the last pass with random bytes.
- US National Computer Security Center TG-025
    - An overwriting algorithm which performs 3 overwrites where the first pass overwrites with zeroes, the next pass with ones, and the last pass with random bytes.
- Australian Defense Signals Directorate ACSI-33 (X0-PD)
- Australian Defense Signals Directorate ACSI-33 (X1-P-PD)
- Canadian RCMP TSSIT OPS-II Standard Wipe
    - A 7-pass overwriting algorithm featuring three alternating passes of zeroes and ones, with the last pass using random characters.
- CIS GOST P50739-95
    - A 5-pass overwriting algorithm that is characterized as performing a slower speeds yet providing more security of data remanence.
- GB HMG Infosec Standard #5 Baseline
    - A 1-pass overwriting algorithm where data is overwritten using zeroes.
- GB HMG Infosec Standard #5 Enhanced
    - A 3-pass overwriting algorithm where the first pass uses zeroes, the second uses ones, and the last pass uses random bytes.
- German VSITR

- A 3-pass overwriting algorithm that uses alternating passes of zeroes and ones, and the last pass using random characters.

Figure 9-1 details each wipe method and its associated wipe pattern.

In order to perform a wipe of a hard drive, the following steps must be performed:

1. In order to wipe a target, a wipe pattern must be selected from an administrator defined list. A wipe pattern consists of disk operations. The following disk operations are supported:
   - Performed prior to methods mentioned in Figure 9-1:
     - ATA REMOVE HPA
     - ATA REMOVE DCO
     - ATA SECURITY ERASE UNIT ENHANCED
   - Performed in conjunction with the methods mentioned in Figure 9-1:
     - Write value
     - Verify value
     - Write random
     - Verify random

   Alternatively, a wipe pattern can be defined by the user using an arbitrary collection of operations, patterns, and/or patterns. These wipe patterns determine the specific method that is used to wipe data from the target.
2. Once the wipe pattern has been defined, a list of one or more target block devices on the system must be specified.
3. The wipe pattern sequence is executed against each target device in order.
4. During the execution of the wipe pattern sequence, progress is displayed to the UI.
5. Whenever a process is completed, a response is sent out, and the log data is received by any process listening for it. These processes comprise the preconfigured log source(s): one or more of standard output, UNIX pipe, or network socket.

Type	Wipe Standard	Wipe Pattern						
		1st Pass	2nd Pass	3rd Pass	4th Pass	5th Pass	6th Pass	7th Pass
L1	Single Overwrite	0's						
L2	DoD 5220.22-M (1 Verify)	0's	1's	R with FV				
L3	HMG IS5 Baseline	0's with FV						
L4	HMG IS5 Enhanced	0's	1's	R with FV				
L5	Canadian OPS-II	0's	1's	0's	1's	0's	1's	R with FV
L6	US Army AR380-19	UD	IUD					
L7	US Air Force 5020	1's	0's	UD				
L8	German VSITR	0's	1's	0's	1's	0's	1's	A's
L9	Navso P-5329-26	0's	1's	R with FV				
La	NCSC-TG-025	0's	1's	R with FV				
Lb	GOST P50739-95	R						
Lc	DoD 5220.22-M (3 Verifies)	0's	FV	1's	FV	97's	FV	
Ld	Custom Wipe	UD	(UD)	(UD)	(UD)	(UD)	(UD)	(UD)

Key

0's	Write logical 0's this pass	UD	Write a user defined character this pass
1's	Write logical 1's this pass	IUD	Write inverse of user defined character this pass
A's	Write the character A this pass	(UD)	Write a user defined character this pass (Optional)
97's	Write the number 97 this pass		
R	Write random characters this pass	FV	Run a Complete Verify on the drive

Figure 9-1: Wipe Patterns

9.3 Security Management

9.3.1 User-Accessible Interfaces

The user interfaces serve as the physical interface where controls are used to operate one or more instances of the WipeDrive application, each on a distinct host. The interfaces that are included in the evaluated configuration are:

- **TUI** – A text-based UI, ran on the same host as the WipeDrive Application, primarily for systems that do not have framebuffer support, typical on many architectures other than x86.
- **GUI** – A graphical UI, ran on the same host as the WipeDrive Application. This will be the default interface for x86 machines which have access to framebuffer

9.3.2 Administrator Capabilities

Administrators of the TOE have the ability to perform the following operations:

- Specify location for audit storage
- Specify format for log storage
- Ability to view sector data
- Create an administrator definable wipe pattern
- Delete an administrator definable wipe pattern
- Run an administrator definable wipe pattern
- View all administrator definable wipe patterns
- Scan for devices
- Get device info for all devices previously scanned

\

If an individual has physical possession of the application, they are then considered to be an administrator of the TOE.

9.3.3 WipeDrive Operations

TOE users have the ability to perform three distinct operations when using the TOE – scanning a drive, probing a drive, and performing the erasure.

9.3.3.1 Drive Scanning

The steps necessary to scan a drive that is a candidate for erasure are listed below:

1. At boot, the OS will run a shell script to launch the UI and the backend (wd).
2. The UI is loaded with parameters from an initial configuration file
3. wd is loaded with a separate configuration file.
4. Once loaded, wd executes a series of commands reserved for its startup sequence as defined in the configuration file, including drive scanning.
5. wd I/O loop receives the instruction to scan the system for all applicable block devices
6. I/O loop passes this command to the parser
7. Parser produces a lexical representation of the data in order to determine that it is valid JSON. If it finds a complete token, the I/O loop will call a function to extract a semantic value.
8. This semantic value is passed to the command factory
9. The command factory determines if it represents valid syntax (correct number and type of parameters) and returns this to the I/O loop as a command object
10. The I/O loop sends the command object to the internal kernel for processing.
11. Kernel loop executes the command object, the outcome of this is dependent on the correctness of the parameters provided
12. The command object opens the /sys/block directory and gathers data from it. This data was originally created by the Linux kernel.
13. This data is cached in wd for viewing in the UI and used as input for probing.

9.3.3.2 Drive Probing

The steps necessary to probe a drive that is a candidate for erasure are listed below:

1. For each item in /sys/block, the following sequence is performed:
2. First, it attempts to instantiate an ATA device object whose constructor performs an ATA IDENTIFY DEVICE command. This command attempts to gather the 512b block of data which defines the drive information and is characteristic of ATA devices. If this fails, it cannot be accessed through the ATA command set.
3. Next, it attempts to instantiate a SCSI device object whose constructor performs a SCSI inquiry. This gathers the following data:
    a. Drive model information
        i. Manufacturer
        ii. Model name
        iii. Serial Number
        iv. Drive Capacity
4. If this fails, the process terminates.
5. Once basic data about the device is gathered, additional SCSI inquiries are run to determine additional information about the device.

6. When the process completes for each device, the data is cached for use in the GUI.

9.3.3.3 Drive Erasure

The steps necessary to wipe a drive that is a candidate for erasure are listed below:

1. In order to wipe a target, a wipe pattern must be selected from a pre-defined list. A wipe pattern consists of disk operations. The following disk operations are supported:
    - Performed prior to methods mentioned in Figure 9-1:
        - ATA REMOVE HPA
        - ATA REMOVE DCO
        - ATA SECURITY ERASE UNIT ENHANCED
    - Performed in conjunction with the methods mentioned in Figure 9-1:
        - Write value -
        - Verify value
        - Write random
        - Verify random
    - Alternatively, a wipe pattern can be defined by the administrator using an arbitrary collection of operations, sequences, and/or patterns. These wipe patterns determine the specific method that is used to wipe data from the target. [gather the operations which comprise specific patterns]
2. Once the wipe pattern has been defined, a list of one or more target block devices on the system must be specified.
3. The administrator definable wipe pattern inserts operations as necessary to identify a valid license is present and decrement the number of licenses remaining.
4. The wipe pattern sequence is executed against each target device in order.
5. During the execution of the wipe pattern sequence, progress is displayed to the UI.
6. Whenever a process is completed, a response is sent out, and the log data is received by any process listening for it. These processes comprise the preconfigured log source(s): one or more of standard output, UNIX pipe, or network socket.

9.3.3.4 Booting from standalone ISO

The TOE can be run from the included Linux LiveCD for wiping an environment booted to a target environment. As a caveat, this may require the boot order to be changed in the BIOS (refer to the physical boundary section for more information on this).

9.3.4 Commands

Commands are processes that implement the user-accessible API of the WDL6.0 grammar. Operations that directly affect devices and that require license usage are abstracted into "device ops" that are invoked indirectly by first defining an administrator definable wipe pattern, then invoking the wipe pattern on specific devices. When an administrator definable wipe pattern is invoked, deviceOp processes correlating to the wipe pattern's listed operations which are instantiated and queued to run by the kernel. These operations can also invoke additional, internal processes to complete their task(s), and/or to allow for code sharing.

9.4 User Data Protection

The TOE provides for the erasure of residual information. This erasure is initiated at the user-facing interfaces and requires communication with the information repository (disk). The erasure of residual information is performed when a deviceOp instance is executed – which is a direct result of the administrator definable wipe pattern. No residual information will reside in the RAM subsequent to a wipe event.

9.5 Self-Protection (ADV_ARC.1)

Domain separation is the security architecture property whereby the TSF defines separate security domains for administrators and for the TSF; it ensures that no user process can affect the contents of a security domain of another administrator or of the TSF.

The TSF is designed in such a manner that requires administrators to have physical possession of the WipeDrive application before any TSF-mediated operations can occur. Therefore, an individual's authorization is based on the possession of the Linux LiveCD containing the WipeDrive application. With the Linux LiveCD, the administrator is able to perform management functions through either the Local GUI or Console UI. There are no specific access control features or authentication methods in place as access is granted to the individual with possession of the Linux LiveCD.

All requests for protected resources, i.e. data located on drives on a target, are processed through the User Interface, either Local GUI or Console UI. When an erasure event is performed on a target, the verification data that denotes whether the erasure was a success or failure is stored in the cache located on the WipeDrive application. This data is then securely transmitted between TOE components to both the administrators interface and is placed in the Log Storage component. This allows the administrator to receive a graphical representation of the status of the erasure performed.

Administrators do have the ability to perform management functions remotely; however the user interfaces related to remote access are not included in this evaluated configuration.

10 TOE Summary Specification Rationale

This section identifies the security functions provided by the TOE mapped to the security functional requirement components contained in this ST. This mapping is provided in the following table.

Security Function	Security Functional Components
Security Audit	FAU_GEN.1 Audit Data Generation
	FAU_GEN.2 User Identity Association
Disk Erasure	FDE_SCN_EXT.1 Scan of Devices
	FDE_PRB_EXT.1 Probe of Devices
	FDE_ERS_EXT.1 Erasure of Devices
User Data Protection	FDP_RIP.1 Residual Information Protection
Security Management	FMT_SMF.1 Specification of Management Functions

Table 10-1: Security Functional Components

10.1.1 Security Audit

The security audit function of the TOE enforces the FAU_GEN.1 and FAU_GEN.2 requirements. FAU_GEN.1 requires a reliable timestamp, which is provided by the Operating System that is bundled on the LiveCD.

By default, audit data is created by scanning, probing, and/or wiping of a target on a device. This data produces a verification message of the success or failure of this event; this notification is sent to the console, the user who performed the event, as well as it being stored in the Log Storage component. Along with the success or failure of events being recorded, the TSF records the date and time of that event, the type of event (i.e. erasure, probe, scan), the identity of the user performing the event, the serial number of the target drive, the wipe pattern sequence used to erase the drive, number of overwrites performed, elapsed time of the operation, physical blocks of target, and the logical blocks of the target. Audit data is also created for the start-up and shutdown of audit functions. All audit operations listed above are inextricably linked to the user who caused the event.

In the evaluated configuration, this audit data contained in the Log Storage component could be stored on any media device, e.g. thumb drive or remotely on an FTP server. When the TOE is configured to log audit data to an FTP server or MySQL database, the logs and all associated data are sent as clear-text.

. The TSF will ensure that this information is logged in a clear and coherent manner so that the reader is able to accurately interpret the data. The TOE saves the audit events in a user-readable format outside of the TOE but is not responsible for facilitating the viewing of those audit records.

The audit functions available to the users of the TOE cannot be disabled and are run automatically.

Users are self-identified when performing operations, therefore the TOE does not ensure correct authentication. As a result, self-identification does not necessitate having the FIA_UID.1 requirement.

10.1.2 Disk Erasure

The disk erasure function of the TOE enforces the FDE_SCN_EXT.1, FDE_PRB_EXT.1, and FDE_ERS_EXT.1 requirements.

The TSF is able to scan a system for (target(s) on (a)) devices that are eligible for erasure. More specifically, the Linux kernel recognizes all ATA and SCSI block devices on the given system as potential erasure targets. Probing is allows the TSF to communicate with devices that are discovered as the result of a scan; this is done in order to determine the parameters for the device. Both scanning and probing of targets are performed during the initialization process of the TOE.

The TSF has the ability, once devices have been targeted through the probing and scanning process, to erase ATA and SCSI block devices by using any of the following 14 available wipe functions: Standard Overwrite, US Army AR380-19, US Air Force System Security Instruction 5020, US DoD 5220.22-M 3-pass, US DoD 5220.22-M 7-pass, US Navy Staff Office Publication P-5329-26, US National Computer Security Center TG-025, Australian Defense Signals Directorate ACSI-33 (X0-PD), Australian Defense Signals Directorate ACSI-33 (X1-P-PD), Canadian RCMP TSSIT OPS-II Standard Wipe, CIS GOST P50739-95, GB HMG Infosec Standard #5 Baseline, GB HMG Infosec Standard #5 Enhanced, and German VSITR.

10.1.3 User Data Protection

The user data protection function of the TOE enforces the FDP_RIP.1 requirement.

Subsequent to an erasure event on a targeted device, the TSF ensures that any previous information is made unavailable based upon the allocation of resources to any object created as a result of an administrator definable wipe pattern. This protects any information that may have remained after an erasure event.

10.1.4 Security Management

The management function of the TOE enforces the FMT_SMF.1 requirement.

The TOE provides management capabilities that only administrators can perform. The management functions available to these users can be initiated either by the Local GUI or Console UI..

11 Security Problem Definition Rationale

11.1 Security Objectives Rationale

The following table provides a mapping with rationale to identify the security objectives that address the stated assumptions and threats.

Assumption	Objective	Rationale
A. ADMIN There will be one or more authorized administrators assigned to install, configure, and manage the TOE and the security information it contains.	OE.ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.	OE. ADMIN maps to A. ADMIN in order to ensure that authorized administrators install, manage and operate the TOE in a manner that maintains its security objectives.
A.PATCHES System Administrators exercise due diligence to acquire updated versions of the TOE and patch the Operational environment (e.g. OS and database) so they are not susceptible to network attacks.	OE.PATCHES Administrators of the Operational Environment exercise due diligence to acquire updated versions of the TOE and patch the Operational Environment (e.g., OS and database) so they are not susceptible to attacks resulting in malfunction of the TOE or associated audit data.	OE. PATCHES maps to A. PATCHES in order to ensure that the authorized administrators properly patch the Operational environment in a manner that maintains its security objectives.
A.NOEVIL All users of the TOE are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the TOE documentation.	OE.NOEVIL All users of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the guidance documentation.	OE.NOEVIL directly maps to A.NOEVIL and ensures that all users of the TOE are properly trained in the configuration and usage of the TOE and will follow the guidance provided.
A.LOCATE The processing resources of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.	OE.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.	OE.LOCATE directly maps to A.LOCATE to ensure that those responsible for the TOE locate the TOE in a controlled access facility that will prevent unauthorized physical access.
A.LOCAL The TOE will be loaded onto the same physical machine as the target resource so that commands are not exposed over the network.	OE.LOCAL The TOE will be loaded onto the same physical machine as the target resource so that commands are not exposed over the network.	OE.LOCAL directly maps to A.LOCAL to ensure that the TOE will be loaded onto the same physical machine as the target resource so that commands are not exposed over the network.

Table 11-1: Assumption to Objective Mapping

Threat/Policy	Objective	Rationale

T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.	O.ROBUST_ADMIN_GUIDANCE The vendor will provide administrators with the necessary information for secure delivery and management of the TOE.	O,ROBUST_ADMIN_GUIDANCE (AGD_OPE.1, AGD_PRE.1, ALC_DEL.1) mitigates the risk of an administrator incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms by providing administrators with the necessary information for secure delivery and management of the TOE.
T.AUDIT_COMPROMISE A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.	O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	O.AUDIT (FAU_GEN.1, FAU_GEN.2, and FAU_SAR.1) addresses T.AUDIT_COMPROMISE by providing the necessary measures to be put in place for recording security relevant events that will only assist authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE.
T.AUDIT_FAILURE A malicious user or process failure may cause the TOE to fail to record or improperly record audit data, thus masking a user's action.	O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	O.AUDIT (FAU_GEN.1, FAU_GEN.2, FAU_SAR.1) addresses T.AUDIT_FAILURE by providing the necessary measures to be put in place for recording security relevant events that will only assist authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE.
	OE.NOEVIL Users of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the organization's guidance documentation.	OE.NOEVIL directly maps to T.AUDIT_FAILURE and ensures users of the TOE are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the organization's guidance documentation.
T.RESIDUAL Any person with access to a target environmental resource can access residual data, either due to a wipe operation being incomplete or a completed wipe operation being	O.ERASE The TOE will provide measures for erasing data contained on block devices on a target system as well as sufficient assurance that the desired data was erased and that the erasure method was sufficient	O.ERASE (FDE_PRB_EXT.1, FDE_SCN_EXT.1, FDE_ERS_EXT.1, FDP_RIP.1) mitigates this risk by ensuring that once an erasure event has occurred, that

insufficient.	for permanent erasure.	no residual information should remain on the target being wiped
T.UNAUTH An unauthorized user obtains the physical medium which contains the TOE and uses it to perform a wipe operation against an environmental resource which there has been no authorization to wipe.	O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor the set of disk wipe patterns made available to the TOE and the storage of audit data generated by the TOE.	O.MANAGE (FMT_SMF.1, FAU_SAR.1) mitigates the risk of unauthorized users being able to access confidential data because there are specific TSF-mediated actions that only authorized users will be able to perform. It also mitigates the threat that unauthorized users would be able to read data contained in the audit records by limiting such management functions to administrators of the TOE.
	OE.ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.	OE.ADMIN (FMT_SMF.1) mitigates the risk of unauthorized users accessing the TOE by having administrators of the TOE select designating individuals to manage the TOE as well as be able to perform specific management functions.

Table 11-2: Threat to Objective Mapping

11.2 EAL 4 Justification

The threats that were chosen are consistent with attacker of medium attack potential, therefore EAL4 was chosen for this ST.

11.3 Requirement Dependency Rationale

All Security Functional Requirement component dependencies have been met by the TOE as defined by the CEM with the exception of FPT_STM.1 and FAU_GEN.1. Rationale for these exclusions is included in Section 10.1.1 above.

11.4 Security Functional Requirements Rationale

The following table provides a mapping with rationale to identify the security functional requirement components that address the stated TOE and environment objectives.

Objective	Security Functional Component	Rationale
O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	AGD_OPE.1 Operational User Guidance	AGD_OPE.1 describes the proper use of the TOE from a user standpoint.
	AGD_PRE.1 Preparative Procedures	AGD_PRE.1 documents the procedures necessary and describes the steps required for the secure installation,

		generation, and start-up of the TOE.
	ALC_DEL.1 Delivery Procedures	ALC_DEL.1 describes product delivery and a description of all procedures used to ensure objectives are not compromised in the delivery process.
O.MANAGE The vendor will provide administrators with the necessary information for secure delivery and management of the TOE.	FMT_SMF.1 Specification of Management Functions	FMT_SMF.1 states that the administrator of the TOE will be able to perform various management functions.
O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	FAU_GEN.1 Audit Data Generation	FAU_GEN.1 states that the TSF shall be able to generate an audit record of the start-up and shutdown of the audit functions
	FAU_GEN.2 User Identity Association	FAU_GEN.2 ensures that the audit records associate a user identity with the auditable event.
	FAU_SAR.1 Audit Review	FAU_SAR.1 ensures that only authorized users of the TOE will be able to read the TOE's audit records. Additionally, FAU_SAR.1 ensures that this information will be presented to the user in a format that is coherent and easily understandable.
O.ERASE The TOE will provide measures for erasing data contained on block devices on a target system as well as sufficient assurance that the desired data was erased and that the erasure method was sufficient for permanent erasure.	FDE_SCN_EXT.1 Scan of Devices	FDE_SCN_EXT.1 states that the TSF shall be able to scan a system for devices that are erasure targets.
	FDE_PRB_EXT.1 Probe of Devices	FDE_PRB_EXT.1 states that the TSF shall communicate with devices that are discovered as the result of the scan in order to determine the parameters for the device.
	FDE_ERS_EXT.1 Erasure of Devices	FDE_ERS_EXT.1 states that the TSF shall be able to erase devices that are discovered by a scan using any combination of the 13 available wipe functions.

	FDP_RIP.1 Residual Information Protection	FDP_RIP.1 states that the TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to any object created as the result of an administrator definable wipe pattern.
	FAU_GEN.1 Audit Data Generation	FAU_GEN.1 states that the TSF shall be able to generate an audit record of the start-up and shutdown of the audit functions
	FAU_GEN.2 User Identity Association	FAU_GEN.2 ensures that the audit records associate a user identity with the auditable event.

Table 11-3**: Security Functional Requirements Rationale**

11.5 Assurance Measures

This section identifies the assurance measures provided by the developer in order to meet the security assurance requirement components for EAL4 augmented with ASE_TSS.2 and ALC_FLR.2. A description of each of the TOE assurance measures follows in Table 11-4.

Component	Document(s)	Rationale
ADV_ARC.1 Security Architecture Design	Functional Specification Document for WhiteCanyon, Inc. WipeDrive version 6.1	This document describes the security architecture of the TOE.
ADV_FSP.4 Functional Specification with complete summary	Functional Specification Document for WhiteCanyon, Inc. WipeDrive version 6.1	This document describes the functional specification of the TOE with complete summary.
ADV_IMP.1 Implementation Representation of the TSF	Low level data flows.xlsx	This document describes the implementation of the TOE.
ADV_TDS.3 Architectural Design	• TOE Design Specification for WhiteCanyon, Inc. WipeDrive version 6.1 • LLDs.zip • LLD eval docs.zip	This document describes the architectural design of the TOE.
AGD_OPE.1 Operational User Guidance	WipeDrive Enterprise User Guide Software Version 6.1	This document describes the operational user guidance for.
AGD_PRE.1 Preparative Procedures	WipeDrive Enterprise User Guide Software Version 6.1	This document describes the preparative procedures that need to be done prior to installing.

Component	Document(s)	Rationale
ALC_CMC.4 Authorizations Controls	WhiteCanyon Source Control Management Version 1.1	This document describes the authorization controls for the TOE.
ALC_CMS.4 CM Scope	• Fogbugz_screenshot • NIAP-Snap-Nov1	These documents describe the CM scope of the TOE.
ALC_DEL.1 Delivery Procedures	WhiteCanyon Product Delivery Process Version 1.1	This document describes product delivery for and a description of all procedures used to ensure objectives are not compromised in the delivery process.
ALC_DVS.1 Identification of Security Measures	WhiteCanyon Security Policies and Procedures Version 1	This document provides an identification of security measures for the TOE.
ALC_FLR.2 Flaw reporting procedures	WhiteCanyon Flaw Remediation Process Version 1.1	This document provides the policies for issuing new releases of the TOE as corrective actions.
ALC_LCD.1 Life-Cycle Definition	WhiteCanyon Product Life Cycle Version 1.1	This document provides the life-cycle definition of the TOE.
ALC_TAT.1 Tools and Techniques	WipeDrive Tools and Technologies Version 1.1	This document describes the tools and techniques used in the life cycle development of the TOE.
ASE_CCL.1 Conformance Claims	White Canyon WipeDrive Version 6.1 Security Target version 1.0	This document describes the CC conformance claims made by the TOE.
ASE_ECD.1 Extended Components Definition	White Canyon WipeDrive Version 6.1 Security Target version 1.0	This document provides a definition for all extended components in the TOE.
ASE_INT.1 Security Target Introduction	White Canyon WipeDrive Version 6.1 Security Target version 1.0	This document describes the Introduction of the Security Target.
ASE_OBJ.2 Security Objectives	White Canyon WipeDrive Version 6.1 Security Target version 1.0	This document describes all of the security objectives for the TOE.
ASE_REQ.2 Security Requirements	White Canyon WipeDrive Version 6.1 Security Target version 1.0	This document describes all of the security requirements for the TOE.
ASE_SPD.1 Security Problem Definition	White Canyon WipeDrive Version 6.1 Security Target version 1.0	This document describes the security problem definition of the Security Target.
ASE_TSS.2 TOE Summary Specification	White Canyon WipeDrive Version 6.1 Security Target version 1.0	This document describes the TSS section of the Security Target.

Component	Document(s)	Rationale
ATE_COV.2 Analysis of Coverage	• Booz_Allen_WC_Wipe_v0+3ATE_4_Matrix_20100930 • WipeDrive_Test-Case_Spreadsheet_Nov_1_2010	This document provides an analysis of coverage for the TOE.
ATE_DPT.2 Testing: Security enforcing modules	• Booz_Allen_WC_Wipe_v0+3ATE_4_Matrix_20100930 • WipeDrive_Test-Case_Spreadsheet_Nov_1_2010	This document describes the security enforcing modules of the TOE.
ATE_FUN.1 Functional Tests	• WipeDrive_Test-Case_Spreadsheet_Nov_1_2010 • Booz_Allen_WC_Wipe_v0+3ATE_4_Matrix_20100930 • Contents of 'On-site Test Results.zip'	This document describes the functional tests for the TOE.
ATE_IND.2 Independent Testing	• Booz Allen_WP_INDTestProcedures • White Canyon WipeDrive Version 6.1 Evaluation Team Test Report Version 1.0 • Contents of 'On-site Test Results.zip'	This document describes the independent testing for the TOE.
AVA_VAN.3 Vulnerability Analysis	• Vulnerability Analysis WHITE CANYON, INC. WIPE DRIVE VERSION 6.1 Version 1.0 • Contents of 'On-site Test Results.zip'	This document describes the vulnerability analysis of the TOE.

Table 11-4: Assurance Requirements Evidence

11.6 Extended Requirements Rationale

This TOE contains the following extended security functions:
FDE_SCN_EXT.1
FDE_PRB_EXT.1
FDE_ERS_EXT.1

11.6.1 FDE_SCN

FDE_SCN_EXT.1 was created to capture the basic functionality provided by the TOE. FDE_SCN_EXT.1 allows for the TOE to be able to scan a given system for devices, e.g. hard drives, partitions, that are targets for erasure. Through this requirement, the WipeDrive application is capable via the Linux kernel of recognizing all block devices located on a system as potential erasure targets. Scanning is performed automatically upon initialization of the WipeDrive application.

11.6.2 FDE_PRB

FDE_PRB_EXT.1 was created to capture the basic functionality provided by the TOE. FDE_PRB_EXT.1 allows for the TOE to communicate with devices that are discovered as a result of the scan in order to determine the parameters for the given device. Probing, along with scanning, is performed automatically upon initialization of the WipeDrive application.

11.6.3 FDE_ERS

FDE_ERS_EXT.1 was created to capture the basic functionality provide by the TOE. FDE_ERS_EXT.1 allows the administrators of the TOE to select from 13 different wipe patterns in order to erase devices discovered through the scanning process.