MOBILE ARMOR FILEARMOR 3.0 & POLICYSERVER V3.1

SECURITY TARGET

VERSION: 1

DATE: 9/7/2010

Mobile Armor® FileArmor 3.0 & PolicyServer v3.1 Security Target

This document is for informational purposes only. Mobile Armor makes no warranties, express or implied, as to the information in this document.

Mobile Armor may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Mobile Armor, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Mobile Armor®, PolicyServer™, DataArmor™, MobileSentinel™, MobileFirewall™, VirusDefense™, RemoteNetwork™, FileArmor™, KeyArmor™, and ColorCode® are trademarks or registered trademarks of Mobile Armor, Inc.

All other trademarks and registered trademarks shown are the property of, and are used by permission of, their respective owners, including, but not limited to: IBM Corporation, Microsoft Corporation, Red Hat, Inc., Novell, Inc., and RSA Security Inc..

Mobile Armor Contact Information:

Mobile Armor, Inc
400 South Woods Mill Road
Suite 300
St. Louis, MO, 63017 USA

Telephone:	+1 (314) 590-0900
Fax:	+1 (314) 590-0995

Website:
http://www.mobilearmor.com

Customer Support:
support@mobilearmor.com

Table of Contents

List of Figures

List of Tables

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is Mobile Armor, Inc. FileArmor and PolicyServer provided by Mobile Armor, Inc. The TOE type is an encryption application. FileArmor can be used to encrypt all data at rest on any supported device without any user intervention to do so. FileArmor requires users to authenticate to it before access to the device, including both the operating system and any data, is granted. PolicyServer is a server application that can be used to manage one or more instances of FileArmor applications from a centralized location in the TOE environment.

The Security Target contains the following additional sections:

- Section 2 – Target of Evaluation (TOE) Description

  This section gives an overview of the TOE, describes the TOE in terms of its physical and logical boundaries, and states the scope of the TOE.

- Section 3 – TOE Security Environment

  This section details the expectations of the environment, the threats that are countered by the TOE and TOE environment.

- Section 4 – TOE Security Objectives

  This section details the security objectives of the TOE and TOE environment.

- Section 5 – IT Security Requirements

  The section presents the security functional requirements (SFR) for the TOE, and details the assurance requirements for EAL4.

- Section 6 – TOE Summary Specification

  The section describes the security functions represented in the TOE that satisfy the security requirements.

- Section 7 – Protection Profile Claims

  This section presents any protection profile claims.

- Section 8 – Rationale

  This section closes the ST with the justifications of the security objectives, requirements and TOE summary specifications as to their consistency, completeness, and suitability.

1.1 Security Target, TOE and Common Criteria Identification

ST Title – Mobile Armor FileArmor and PolicyServer Security Target

ST Version – Version 1.0

ST Date – 9/7/2010

TOE Identification – Mobile Armor PolicyServer 3.1 version 3.1.0.445 and FileArmor 3.0 SP7

TOE Developer – Mobile Armor, Inc.

Evaluation Sponsor – Mobile Armor, Inc.

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 2, September 2007

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 3.1, Revision 2, September 2007.
    - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 3.1, Revision 2, September 2007.
    - Part 3 Conformant
    - Assurance Level: EAL 4 augmented with ALC_FLR.3

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., **[assignment]**). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., **[*[selected-assignment]*]**).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., **[*selection*]**).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.4 Terminology

1.4.1 Acronyms

Acronym	Meaning
AES	Advanced Encryption Standard
CC	Common Criteria
CCTL	CC Testing Laboratory
CEK	Communication Encryption Key
CI	Configuration Item
CM	Configuration Management
CMP	Configuration Management Plan
CVE	Common Vulnerabilities and Exposures
CVS	Concurrent Versioning System
DEK	Disk Encryption Key
DoD	Department of Defense
DoS	Denial of Service
EAL	Evaluation Assurance Level
FADB	File Armor DataBase
FEK	File Encryption Key
FSP	Functional Specification
GUI	Graphical User Interface
HLD	High-level Design
HTTP	Hyper-text Transfer Protocol
ID	Identity/Identification
IP	Internet Protocol
IT	Information Technology
MBR	Master Boot Record
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OCSP	Online Certificate Status Protocol
OS	Operating System
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SOAP	Simple Object Access Protocol
SSO	Single Sign-on
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functions
TSS	TOE Summary Specifications

Table 1 - Acronyms

1.4.2 Definitions

- Authentication Mechanisms: There are several authentication mechanisms available and they are noted under this heading. These mechanisms are further classified into two categories, normal and help.
    - Normal Password Mechanisms: These mechanisms are designed for use in “everyday” authentication.

        - Fixed Password Mechanism: This mechanism is the common method of authentication using letters, numbers and symbols. This method is available on FileArmor and the PolicyServer.
        - Smart Card Mechanism: This mechanism uses smart cards for authentication. This method is available on FileArmor.
    - Help Password Mechanisms: This mechanism is designed for use in situations where the user is unable to login normally. For example, the user has forgotten their password, or lost/misplaced a smart card, or has a locked account.
        - Remote Authentication Mechanism: This mechanism uses a challenge and response sequence passed verbally (such as by phone) from the FileArmor user to an Authorized Administrator in the PolicyServer Console and back to provide authentication to the client. It is initiated through the Reset Help function of the policy server using a Soft Token.
- Authenticators: A term used to refer to both Enterprise and Group Authenticators
- Authorized Administrators: A term used to specify administrators and authenticators in general, but with the understanding that an administrator or authenticator is authorized to provide administration to users at or below their position in the user directory hierarchy. For example, Enterprise Administrators can perform administration at all levels of the hierarchy, but Group Administrators can only perform administration at groups that are at or below their group in the hierarchy. When this term is used, it is to be understood that the administrator or authenticator in question is not able to exceed their hierarchical authority.
- Authorized user: A term used within SFR assignment operations to refer to authenticated users of the TOE.
- Device: this term is used generically to mean any system where FileArmor can be installed, PCs, laptops.
- FADB: This term is defined to mean the protected storage area on a FileArmor-protected client where user, policy and audit data is stored internally.
- FileArmor: When this term is used without versions or other specific identifiers, it is intended to generically mean all clients of the TOE.
- Install: This pertains to the installation of a managed client. Stand Alone installations are out of scope of the evaluation. File Armor client software cannot be updated from the Policy Server.
- Policy Administrators: A term used to specify Enterprise and Group Administrators in general, excluding Authenticators, but with the understanding that an administrator is authorized to provide policy administration at or below their position in the user directory hierarchy. When this term is used, it is to be understood that the administrator in question is not able to exceed their hierarchical authority.
- PolicyServer database: This term is used to talk generically about the two databases (log and policy) that the PolicyServer maintains. The PolicyServer databases are stored in SQL Server.
    - Log database: This term is used to specifically talk about the log database. All PolicyServer logs are stored here directly, and all FileArmor logs are eventually stored here once they have been uploaded to the PolicyServer.
    - Policy database: This term is used to specifically talk about the policy database which stores everything for the management and configuration of PolicyServer and FileArmor except the logs.

2. TOE Description

The Target of Evaluation (TOE) is Mobile Armor FileArmor 3.0 a client, and PolicyServer 3.1, the management server.

FileArmor is an application that can be used to encrypt specific files and folders. Some of these can be configured for minimal user input (such as encrypting a folder) while others are designed for user choice (such as encrypting an individual file). FileArmor requires users to authenticate to it before access to the any encrypted data is granted, and before any data can be encrypted. The client can be installed on computers running Microsoft Windows operating systems.

The PolicyServer is a server application that can be used to manage one or more instances of Mobile Armor applications such as FileArmor or DataArmor from a centralized location in the TOE environment. The PolicyServer provides policy, user and key management of the FileArmor clients as well as centralized audit storage.

The File Armor client must be installed in a Managed configuration. Stand Alone installations are out of scope of this evaluation.

The remainder of this section summarizes the TOE architecture.

2.1 TOE Overview

The FileArmor client provides protection for individual files and folders by employing encryption on the specified objects. When FileArmor is installed on a device, it integrates into the operating system to provide a seamless user experience when encrypting and decrypting. The cryptographic module is paired with an authentication module which authorizes the user for access to encrypted files as well as the ability to create new ones. An additional function of the FileArmor client is the ability to create self-extracting encrypted files. These files can be run on another system without the FileArmor client installed, and prompt the user for credentials (either a smart card certificate or password) to decrypt the file contents.

FileArmor is designed as an application with a file system driver and a service. The service and driver are automatically loaded when the OS starts. The driver actually performs the cryptographic operations, while the service provides support for connectivity to the PolicyServer. The application can be configured to start automatically or manually by the user as determined by the needs of the user. When the application starts, the user is required to authenticate before the user can encrypt new files or decrypt existing files (the exception to decryption is self-extracting archives as they are able to execute independently by design).

FileArmor can be used to handle files manually and automatically; most actions are usually handled automatically. For example, once a user has authenticated, encrypted files will be automatically decrypted for the user when the user opens them (such as by double-clicking on it). Automatic encryption occurs when the file is located in a folder marked for encryption. In such a folder, any files added to the folder will be automatically encrypted without any further user interaction (i.e. they do not need to specify any other information as the configuration is attached to the folder and passed to the file). Folders marked for encryption will cause each file to be encrypted individually, not as a single encrypted block. This allows for access to individual files within an encrypted folder.

Manual actions in FileArmor usually focus on the encryption of individual files (or groups of files, such as but not limited to, a folder) outside an automatic location. For example, to encrypt a file that will stay on the desktop (where automatic encryption would cause problems with application shortcuts), the user would use the FileArmor application to choose to encrypt the file.

To enforce the controls provided by the authentication, FileArmor encrypts the entire contents of every file. FileArmor does not prevent OS actions on the file (such as copying the file, renaming the file, or deleting the file), it does prevent access to the protected contents of the file. Once the file has been encrypted, its “plaintext” contents can only be accessed by an authenticated user.

FileArmor supports three File Encryption Keys (FEKs) for each user: Enterprise, Group and User. All FEKs are AES 256-bit keys. A user will always have a User FEK, though Group and Enterprise FEKs are optional as determined by the administrators. In addition to the FEKs, it is also possible for a file to be encrypted with a password or smart card, independent of the FEKs associated with the user. The determination of the key used to encrypt a file is based on administrator-controlled policy. The first choice is whether a policy has been set to specify a key (such as when a folder is specified for automatic encryption). The second choice if a key is not specified by policy is to ask the user what key to use. The user can use any associated FEK or choose a password (or smart card).

The user’s authentication credentials are used to protect all FEKs associated with the user. Each FileArmor user on the client will have the FEKs associated with their account protected separately. The encrypted FEKs for a user are only changed when the user’s authentication credentials change (i.e. the user changes their password), such that the new credentials will unlock the FEKs. In most cases the authentication credentials are hashed to generate a key which can decrypt the FEKs, though when smart cards are used, the cryptographic functions of the card are used to protect the FEKs. No FEK is ever written in clear text to the hard disk. The FEKs are generated by the PolicyServer as part of the installation process, with a unique FEK generated for each user of FileArmor. Enterprise and Group FEKs are generated for the Enterprise or Group as applicable and assigned to any user in the Enterprise (i.e. an Enterprise FEK will be given to all users) or Group. For recovery purposes, a copy of all FEKs are encrypted and stored on the PolicyServer.

Once the FEKs have been decrypted (by performing a successful authentication), they are loaded into a filter driver designed for the installed OS. The driver executes in the kernel space of the OS and is available for subsequent operations to automatically encrypt and decrypt the contents of files and folders. When the user (or operating system) attempts to access data that has been encrypted, the filter driver will automatically decrypt the data. The user can also use the UI to decrypt files manually (such as permanently removing the encryption). Files placed into folders which have been marked for automatic encryption will be automatically encrypted by the driver as well. Similar to decryption, the user can manually utilize the FEKs loaded into the driver to encrypt individual files or folders. When the user is performing manual encryption operations, they are not shown the FEK used to encrypt the file, but are allowed to choose the FEK by reference (Enterprise, Group or User).

The authentication credentials necessary to access FileArmor are controlled by the PolicyServer administrators (for example, the type of credential to be used:

Fixed password or Smart Card

and restrictions about the credentials

Description	Value Range	Default
Specify whether passwords can contain alpha, numeric, special or a combination.	Alpha,Numeric,Special	Alpha,Numeric,Special
Specify (in days) when to force a user to change their password.	1 to 5000	22
Specify the number of consecutive characters allowed in a password.	1 to 10	3
Specify the minimum length allowed for passwords.	1 to 255	6

Description	Value Range	Default
Specify the number of alpha characters that must be used in a password.	0 to 15	0
Specify the number of numbers that must be used in a password.	0 to 15	0
Specify the number of special characters that must be used in a password.	to 15	0
	0	

The PolicyServer provides an interface for reviewing all logs generated in the system. This includes logs generated on the server related to administrative activity as well as all logs uploaded from clients to the PolicyServer. The records can be reviewed both on a group-by-group basis as well as a system-wide basis. Additionally, alerts can be triggered based on incoming client audit records, notifying administrators of potential problems, such as multiple failed logins.

2.2 TOE Architecture

The TOE can be described in terms of the following components:

- Mobile Armor FileArmor application – Provides encryption and invocation and enforcement of authentication decisions implemented within the TOE or in the TOE environment, depending on how the TOE is configured. Includes pre-access authentication components (to invoke configured authentication services) and data encryption components.
- Mobile Armor PolicyServer – Provides administrative interfaces that can be used to manage FileArmor encryption and authentication policy functions. The administrative PolicyServer interface implemented as a Microsoft Management Console (MMC) “snap-in”, which displays PolicyServer GUI components within a MMC GUI window pane called a “console”. Only the Policy Server elements that are used for file Armor have been included in the evaluation

Figure 1 below illustrates the TOE as it can be deployed in a customer environment. The pieces in the configuration are color coded to illustrate the different components and how they relate to the TOE. The red box indicates FileArmor clients. The blue box indicates PolicyServer components, including both the server pieces and the management client. The orange box indicates external services which are required for the evaluated configuration, in this case an email server. The green boxes indicate optional services which can be connected to the system, but which are not required.

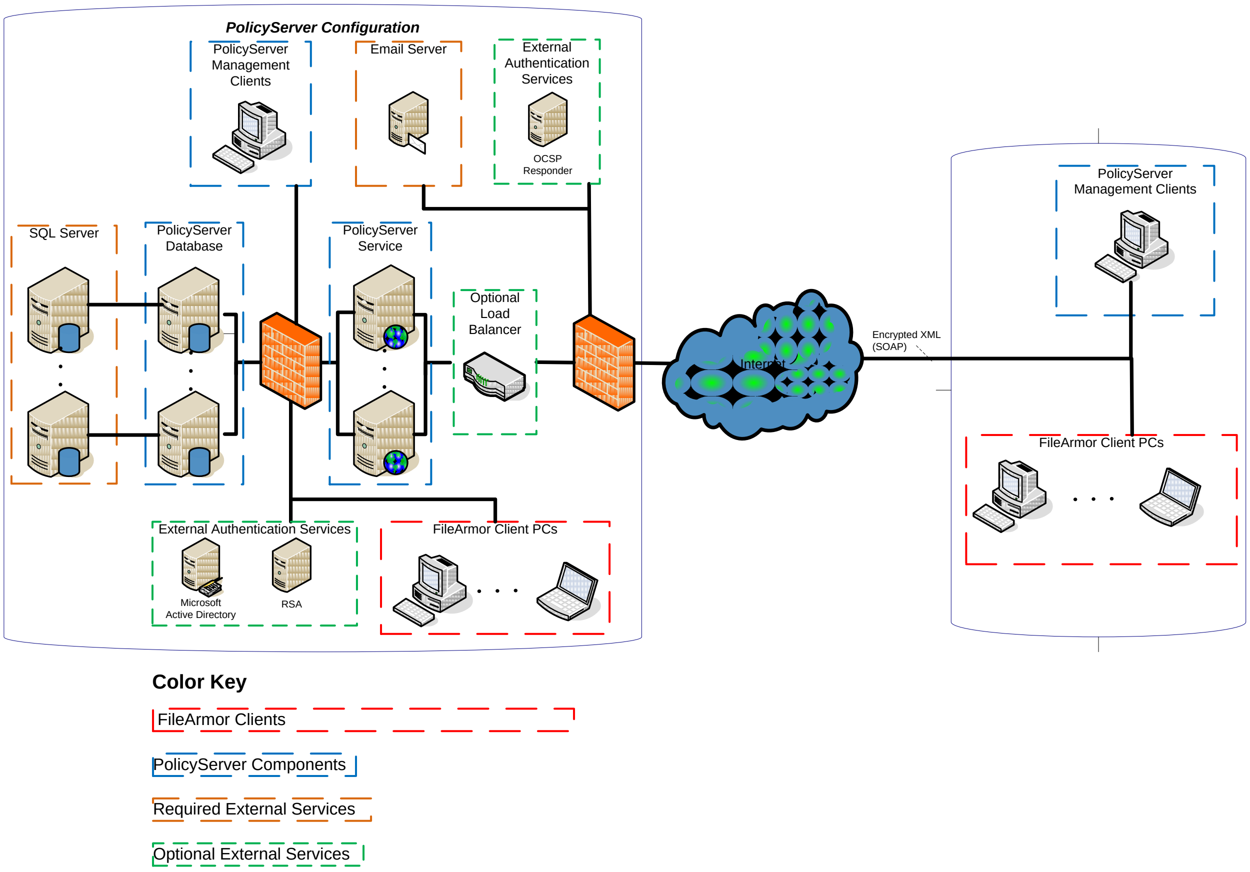
PolicyServer Configuration
PolicyServer Management Clients
Email Server
External Authentication Services
OCSP Responder
SQL Server
PolicyServer Database
PolicyServer Service
Optional Load Balancer
Internet
Encrypted XML (SOAP)
PolicyServer Management Clients
FileArmor Client PCs
External Authentication Services
Microsoft Active Directory
RSA
FileArmor Client PCs
Color Key
FileArmor Clients
PolicyServer Components
Required External Services
Optional External Services

Figure 1 - Mobile Armor Solution Architecture

The intended environment of the TOE is dependent on the piece of the TOE being described.

The FileArmor portion of the intended environment can be described in terms of the following components:

- Operating systems – Provides the runtime environment for FileArmor application components.
- Authentication servers – Provides optional external authentication services for FileArmor.

The PolicyServer portion of the intended environment can be described in terms of the following components:

- Operating systems – Provides the runtime environment for the PolicyServer application components. Provides operating system GUI interfaces for PolicyServer. Provides web server for interface between PolicyServer and clients.
- SQL Database – Provides the storage for FileArmor and PolicyServer user and device information, policies and centralized audit storage. PolicyServer maintains two separate databases, one specifically for log events and one for all other data.
- Mail Server – Provides the SMTP server for use in email alert configuration. Authentication servers – Provides optional external authentication services for FileArmor users. The PolicyServer acts as a proxy for authentication.

- Load Balancer – Provides optional scalability by allowing multiple PolicyServers to be configured together as one.

2.2.1 Physical Boundaries

The TOE is a software product, and as such the physical boundary of the TOE is defined as the files and information stored on the device where it is installed. The TOE functions are implemented uniformly across all supported OS platforms.

The following software packages are considered to be the TOE:

- PolicyServer Service
- PolicyServer Database (the database created in SQL Server)
- PolicyServer Management Console
- Active Directory Plug-in
- FileArmor Client

Any other products which may be attached to this configuration are not considered part of the evaluated configuration. However, it is acceptable to use other evaluated products, e.g. Data Armor, in conjunction with the evaluated configuration.

The operational environment of TOE depends on the following:

- PolicyServer
    - Operating systems
        - For the PolicyServer Service
            - Microsoft Windows Server 2003 SP1+, Standard or Enterprise Editions
        - For the PolicyServer Management console
            - Microsoft Windows Server 2003 SP1+, Standard or Enterprise Editions
    - Database
        - Microsoft SQL Server 2005 with Service Pack 2+
    - External Mail Server
- FileArmor platforms
    - Microsoft Windows XP SP3
    - Microsoft Windows Vista SP2
- **Optionally -** Authentication servers (for external authentication integration):
    - Microsoft Active Directory

2.2.2 Logical Boundaries

This section summarizes the security functions provided by the TOE:

- Security audit
- Cryptographic support

- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- Resource Utilization
- TOE Access

2.2.2.1 Security audit

The TOE generates audit records for actions taken in FileArmor and the PolicyServer. The management console provides a way to restrict access to the audit records to authorized administrators. The OS is relied on to provide reliable time stamps for use in audit records.

The audit records that are taken can be divided into three broad categories: authentication actions, management actions and status messages. Authentication logs cover all attempts to login to the client or server, and record success and failure, as well as conditions such as locking the user. Management actions cover all actions taken on the PolicyServer (managing users, policies, etc). Status messages cover events such as the startup and shutdown of functions which can provide audit records, failure messages related to TOE functions (such as a client not being able to contact the server), and encryption status.

Audit records recorded on the client are stored there until a connection is made with the PolicyServer at which time they are sent to the PolicyServer for central storage (they will be stored in the PolicyServer log database). While stored on the client, the audit records are stored in a secure area and protected against tampering. The clients are capable of storing at least 4000 log messages before wrapping will occur and the oldest messages will be lost (a first in-first out system). Audit records generated on the PolicyServer are stored directly into the PolicyServer log database.

The PolicyServer provides an alert notification system via email to notify administrators of potential security violations. The FileArmor login will notify authenticated users of potential security violations on systems where they login.

2.2.2.2 Cryptographic support

The TOE provides its own FIPS-validated cryptographic module which performs symmetric encryption and decryption operations on cryptographic keys, storage media, and data or commands sent over a network. AES algorithm is used for this encryption. Additional algorithms are also supported for random number generation and various hashing functions. All algorithms are FIPS-validated.

While random numbers are only generated on the PolicyServer (for use in creating encryption keys), all other functions of the cryptographic module are used by both FileArmor and PolicyServer.

2.2.2.3 User data protection

The TOE provides the ability to restrict access to data that are encrypted whether the data is kept on that device or any other. All users are subject to the Data Access Control Policy when accessing the protected data.

2.2.2.4 Identification and authentication

The TOE requires users to be identified and authenticated before access is allowed to protected data. Administrators can choose from several different authentication mechanisms based on the needs of the organization, including the ability to link to external authentication services.

2.2.2.5 Security management

The TOE provides the ability to manage users and groups, encryption settings, and authentication server settings. The TOE provides five levels of authority: Enterprise Administrator, Enterprise Authenticator, Group Administrator, Group Authenticator and User. Administrators and Authenticators have access to the management console, with their access being determined by where their authority is granted in the hierarchy. Users only have the ability to login to FileArmor clients.

2.2.2.6 Protection of the TSF

For the FileArmor component, the TOE provides authentication components and filter driver components. Encryption of all specified data prevents bypassing the authentication components. When the TOE starts, it performs several tests to ensure it is properly functioning and that security has not been compromised.

For the PolicyServer component, the TOE relies on the operating system to restrict access to its software as well as the database where the TOE information is stored. The TOE encrypts its communication between FileArmor and the PolicyServer by encrypting and decrypting SOAP messages sent and received using HTTP.

2.2.2.7 Resource Utilization

FileArmor is able to maintain and ensure the proper application of its existing policies even when communications are unavailable.

2.2.2.8 TOE Access

The TOE can provide a login banner to all users accessing FileArmor as well as all authorized administrators accessing the management console.

2.3 TOE Documentation

Mobile Armor offers a series of documents that describe the installation process for the TOE as well as guidance for subsequent use and administration of the applicable security features.

The documentation for the TOE is:

- Mobile Armor™ FileArmor™ v3.0
- Mobile Armor™ PolicyServer™ v3.1
- PolicyServer™ v3.1 Administration Guide
- PolicyServer™ v3.1 Installation Guide
- PolicyServer™ v3.1 Administration Appendices
- FileArmor™ v3.0 PC Installation Guide
- FileArmor™ v3.0 PC User Guide
- FileArmor™ v3.0 PC Administration Guide
- Mobile Armor™ Certification Guide

3. TOE Security Environment

This section summarizes the threats addressed by the TOE and assumptions about the intended environment of the TOE. Note that while the identified threats are mitigated by the security functions implemented in the TOE, the overall assurance level (EAL 4) also serves as an indicator of whether the TOE would be suitable for a given environment.

3.1 Threats

T.ACCOUNTABILITY A user may not be held accountable for their actions.

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

T.MASQUERADE An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.

T.SUBVERT A malicious user may cause non-configuration data at rest to be inappropriately accessed (viewed, modified or deleted).

T.TSF_COMPROMISE A malicious user may cause configuration data to be inappropriately accessed (viewed, modified or deleted).

T.UNAUTH_ACCESS A user may gain unauthorized access (view, modify, delete) to configuration data.

3.2 Assumptions

A.LOCATE The server portion of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

A.NO_EVIL The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation.

A.NO_EVIL_USER Users of the TOE are properly trained in the use of the TOE and will cooperate with those responsible for administration in maintaining TOE security.

A.DEVICE_USE Users of the TOE will follow policies to prevent unauthorized physical access to a TOE-protected device.

4. Security Objectives

This section summarizes the security objectives for the TOE and its environment.

4.1 Security Objectives for the TOE

O.ACCESS The TOE will ensure that users gain only authorized access to the TOE and to the resources that the TOE controls.

O.ADMIN_ROLE The TOE will provide authorized administrator roles to isolate administrative actions.

O.ALERT The TOE will provide the capability to alert authorized users of potential security violations.

O.AUDIT_GENERATION The TOE will provide the capability to create records of security relevant events associated with users.

O.AUDIT_REVIEW The TOE will provide the capability to view audit information.

O.CRYPTO_OPS The TOE will ensure that all cryptographic operations are compliant with FIPS 140-2 Level 1 & 2 based on installed OS (cryptographic module), FIPS 197 (AES), FIPS 46-3 (3DES), FIPS 180-2 (SHS), FIPS 198 (HMAC) and ANSI X9.31 (RNG) and that the keys for those operations are managed accordingly.

O.DATA_TRANSFER The TOE will protect system data in transmission between the client and server.

O.FAULT_TOLERANCE The TOE must continue to enforce access control policies if communications with the server are not available or if the server has detected violations of policy integrity.

O.MANAGE The TOE will allow administrators to effectively manage the TOE and its security functions, and must ensure that only authorized administrators are able to access such functionality.

O.TOE_PROTECTION The client portion of the TOE will protect itself and its assets from external interference or tampering.

O.USER_AUTHENTICATION The TOE will ensure that users are reliably identified and are authenticated before any access to TOE-protected assets is granted.

4.2 Security Objectives for the TOE Environment

OE.AUDIT_PROTECTION The TOE Environment will be configured to protect audit information on the server portion of the TOE.

OE.CONFIG The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation.

OE.USER_GUIDANCE Users of the TOE will be properly trained in the secure usage and protection of the TOE and the devices where it is installed.

OE.PHYCAL The server portion of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

OE.TIME The TOE Environment will provide reliable time stamps for use in audit records.

OE.TOE_PROTECTION The TOE Environment will protect the server portion of the TOE and the assets under its control from external interference or tampering.

5. IT Security Requirements

5.1 Extended Components Definition

There are no extended security requirements defined within this Security Target.

5.2 TOE Security Functional Requirements

The security functional requirements for the TOE are drawn from Part 2 of the Common Criteria, as represented below.

Requirement Class	Requirement Component
FAU: Security audit	FAU_ARP.1a Security Alarms
	FAU_ARP.1b Security Alarms
	FAU_GEN.1: Audit data generation
	FAU_GEN.2: User identity association
	FAU_SAA.1a Potential Violation Analysis
	FAU_SAA.1b Potential Violation Analysis
	FAU_SAR.1: Audit review
	FAU_SAR.2: Restricted audit review
	FAU_SAR.3: Selectable audit review
	FAU_STG.1a: Protected audit trail storage
	FAU_STG.1b: Protected audit trail storage
FCS: Cryptographic support	FCS_CKM.1: Cryptographic key generation
	FCS_CKM.3: Cryptographic key access
	FCS_CKM.4: Cryptographic key destruction
	FCS_COP.1a: Cryptographic operation
	FCS_COP.1b: Cryptographic operation
	FCS_COP.1c: Cryptographic operation
FDP: User data protection	FDP_ACC.1: Subset access control
	FDP_ACF.1: Security attribute based access control
FIA: Identification and authentication	FIA_AFL.1: Authenitcation failure handling
	FIA_ATD.1: User attribute definition
	FIA_UAU.2a: User authentication before any action
	FIA_UAU.2b: User authentication before any action
	FIA_UAU.5: Multiple authentication mechanisms
	FIA_UAU.6: Re-authenticating
	FIA_UAU.7: Protected authentication feedback
	FIA_UID.2a: User identification before any action
	FIA_UID.2b: User identification before any action
FMT: Security management	FMT_MSA.1: Management of security attributes
	FMT_MSA.2: Secure security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_MTD.1a: Management of TSF data
	FMT_MTD.1b: Management of TSF data
	FMT_MTD.1c: Management of TSF data
	FMT_MTD.1d: Management of TSF data
	FMT_MTD.1e: Management of TSF data
	FMT_MTD.1f: Management of TSF data
	FMT_MTD.1g: Management of TSF data

Requirement Class	Requirement Component
	FMT_MTD.2: Management of limits on TSF data
	FMT_REV.1: Revocation
	FMT_SAE.1: Time-limited authorization
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_FLS.1a: Failure with preservation of secure state
	FPT_FLS.1b: Failure with preservation of secure state
	FPT_ITT.1: Basic internal TSF data transfer protection
	FPT_TST.1: TSF Testing
FRU: Resource Utilization	FRU_FLT.1: Degraded fault tolerance
FTA: TOE Access	FTA_TAB.1:Default TOE Access Banner

Table 2 - TOE Security Functional Components

5.2.1 Security audit (FAU)

5.2.1.1 Security alarms (FAU_ARP.1a)

FAU_ARP.1a.1 The TSF shall take **[action to notify the user on the next successful login after the potential client violation]** upon detection of a potential security violation.

Application note: This requirement is specifically focused on the client portion of the TOE and an end user login to the TOE after the potential violation has been detected.

5.2.1.2 Security alarms (FAU_ARP.1b)

FAU_ARP.1b.1 The TSF shall take **[action to notify the specified Administrators by email]** upon detection of a potential security violation.

Application note: This requirement is specifically focused on alert notification of administrators as events are uploaded and recorded into the central PolicyServer audit database

5.2.1.3 Audit data generation (FAU_GEN.1)

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) **[see the table below]**.

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, 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, **[no additional information]**.

Security Functional Requirement	Auditable Event(s)
FAU_ARP.1a FAU_ARP.1b	None
FAU_GEN.1	Start-up and shutdown.
FAU_GEN.2	None
FAU_SAA.1a FAU_SAA.1b	None

FAU_SAR.1-3	None
FAU_STG.1	None
FCS_CKM.1	Success of cryptographic operations
FCS_CKM.3	None
FCS_CKM.4	Success of cryptographic operations
FCS_COP.1a	Success of cryptographic operations
FCS_COP.1b	Failure of cryptographic operations
FCS_COP.1c	Failure of cryptographic operations
FDP_ACC.1	None
FDP_ACF.1	None
FIA_AFL.1	None
FIA_ATD.1	None
FIA_UAU.2	All use of the authentication mechanism
FIA_UAU.5	All use of the authentication mechanism and which authentication mechanism was chosen. Although audited, client SmartCard authentication is recorded as a fixed password event.
FIA_UAU.6	Failure of reauthentication
FIA_UAU.7	None
FIA_UID.2	All use of the user identification mechanism, including the user identity provided.
FMT_MSA.1	All modifications of the values of user and group security attributes on the policy server. No audit messages are provided for any client modification.
FMT_MSA.2	All offered and rejected values for a security attribute (password). No audit messages are provided for any client modification.
FMT_MSA.3	None
FMT_MTD.1a FMT_MTD.1b FMT_MTD.1c FMT_MTD.1d FMT_MTD.1e FMT_MTD.1f FMT_MTD.1g	None
FMT_MTD.2	None
FMT_REV.1	None
FMT_SAE.1	None
FMT_SMF.1	Use of the management functions.
FMT_SMR.1	Modifications to the group of users that are part of a role
FPT_FLS.1	None
FPT_ITT.1	None
FPT_TST.1	None
FRU_FLT.1	None
FTA_TAB.1	None

5.2.1.4 User identity association (FAU_GEN.2)

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.2.1.5 Potential violation analysis (FAU_SAA.1a)

FAU_SAA.1a.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.

FAU_SAA.1a.2 The TSF shall enforce the following rules for monitoring audited events:

a.) Accumulation or combination of **[Two or more consecutive failed login attempt on a PC client]**

known to indicate a potential security violation;

b.) **[no other rules]**.

Application note: This requirement is specifically focused on the client portion of the TOE and an end user login to the TOE after the potential violation has been detected.

5.2.1.6 Potential violation analysis (FAU_SAA.1b)

FAU_SAA.1b.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.

FAU_SAA.1b.2 The TSF shall enforce the following rules for monitoring audited events:

a.) Accumulation or combination of **[**

- **Multiple failed login attempts on the client that reach the threshold defined by the administrator**
- **Policy value tampering events identified through invalid policy HMAC values in the PolicyServer policy database**
- **Log tampering events identified through invalid policy HMAC values in the PolicyServer log database]**

known to indicate a potential security violation;

b.) **[no other rules]**.

Application note: This requirement is specifically focused on notification of administrators as events are uploaded and recorded into the central PolicyServer audit database.

5.2.1.7 Audit Review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **[Authorized Administrators]** with the capability to read **[all audit information according to the administrator's authority]** from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

5.2.1.8 Restricted audit review (FAU_SAR.2)

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.2.1.9 Selectable audit review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to perform **[*searches*]** of audit data based on **[user identity, device name, message ID and date range]**.

Application note: This requirement is specifically focused on the server portion of the TOE.

5.2.1.10 Protected audit trail storage (FAU_STG.1a)

FAU_STG.1a.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1a.2 The TSF shall be able to **[*detect*]** unauthorized modifications to the stored audit records in the audit trail.

Application note: This requirement is specifically focused on the client portion of the TOE.

5.2.1.11 Protected audit trail storage (FAU_STG.1b)

FAU_STG.1b.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1b.2 The TSF shall be able to **[*detect*]** unauthorized modifications to the stored audit records in the audit trail.

Application note: This requirement is specifically focused on the server portion of the TOE.

5.2.2 Cryptographic support (FCS)

5.2.2.1 Cryptographic key generation (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[ANSI X9.31 RNG]** and specified cryptographic key sizes **[AES 256 bits]** that meet the following: **[FIPS 140-2, Section 4.7.2 Key Generation]**.

5.2.2.2 Cryptographic key access (FCS_CKM.3)

FCS_CKM.3.1 The TSF shall perform **[cryptographic key escrow]** in accordance with a specified cryptographic key access method **[duplication]** that meets the following: **[no standard]**.

5.2.2.3 Cryptographic key destruction (FCS_CKM.4)

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a **[zeroization]** that meets the following: **[FIPS PUB 140-2 Section 4.7.6 Key Zeroization]**.

5.2.2.4 Cryptographic operation (FCS_COP.1a)

FCS_COP.1a.1 The TSF shall perform **[symmetric encryption and decryption]** in accordance with a specified cryptographic algorithm **[AES]** and cryptographic key sizes **[OFB, CBC and ECB modes with 256 bit keys]** that meet the following: **[FIPS 197]**.

5.2.2.5 Cryptographic operation (FCS_COP.1b)

FCS_COP.1b.1 The TSF shall perform **[secure hashing]** in accordance with a specified cryptographic algorithm **[SHA-1, SHA-224, SHA-256, SHA-384, SHA-512]** and cryptographic key sizes **[not applicable]** that meet the following: **[FIPS 180-2]**.

5.2.2.6 Cryptographic operation (FCS_COP.1c)

FCS_COP.1c.1 The TSF shall perform **[message authentication]** in accordance with a specified cryptographic algorithm **[HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512]** and cryptographic key sizes **[KS<BS, KS=BS, KS>BS]** that meet the following: **[FIPS 198]**.

5.2.3 User data protection (FDP)

5.2.3.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the **[Data Access Control Policy]** on **[subjects: authorized users; objects: specified files and folders; and, all operations on the plain-text contents of the specified files and folders by users]**.

Application note: The TOE protects file and folder contents by encrypting them and only allowing users with the appropriate fixed password or FEK credentials to access the plain-text contents of the object. Hence, while other operations (e.g., erase) could be performed outside the control of the TOE, the operations on the plain-text, or decrypted, content of the object is subject to this access control policy.

5.2.3.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the **[Data Access Control Policy]** to objects based on the following: **[security attributes: subject: user identity and authentication data; object: encryption keys]**.

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

- **The requested operation is allowed to operate on the plain-text contents of the applicable object if the authorized user identity and authentication data can successfully decrypt the symmetric key associated with the applicable object,**
- **Otherwise the plain-text object contents cannot be accessed].**

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[there are no additional authorization rules]**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the: **[there are no additional denial rules]**.

5.2.4 Identification and authentication (FIA)

5.2.4.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when **[*an administrator configurable positive integer within [1-10]*]** unsuccessful authentication attempts occur related to **[any logon]**.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **[perform one of the following actions based on the configured policy:**

- **On FileArmor, one of the following:**
  - **Lock the user account until unlocked with Remote Authentication,**
  - **Specify a Time Delay before further login attempts**
- **On PolicyServer Management Console**
  - **Lock the user account until unlocked by a Policy Administrator].**

5.2.4.2 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: **[**

- **On FileArmor:**
  - **User identity,**
  - **Group membership,**
  - **Role,**
  - **File Encryption Keys**
- **On the PolicyServer**
  - **User identity,**
  - **Authentication data as related to PolicyServer administration roles,**
  - **Group membership,**
  - **Role].**

5.2.4.3 User authentication before any action (FIA_UAU.2a)

FIA_UAU.2a.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application note: This requirement is specifically focused on the client portion of the TOE.

5.2.4.4 User authentication before any action (FIA_UAU.2b)

FIA_UAU.2b.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application note: This requirement is specifically focused on the server portion of the TOE.

5.2.4.5 Multiple authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide **[**

- **Fixed Password Mechanism**
- **Smart Card Mechanism**
- **Remote Authentication Mechanism]**

to support user authentication.

Application Note: SmartCard Mechanism is specifically focused on the client portion of the TOE.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the **[following rules:**

- **The Fixed Password or Smart Card must be used for normal login of all users to FileArmor**
- **The Fixed Password must be used for any login to the PolicyServer management console**
- **The Remote Authentication Mechanism may be used in cases where normal login is not possible (such as a forgotten password or lost smart card)**
- **The Remote Authentication mechanism may be required when a user account has become locked based on policies set by the administrator].**

5.2.4.6 Re-authenticating (FIA_UAU.6)

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions **[the PolicyServer management interface has been inactive for an administrator specified time]**.

Application note: This requirement is specifically focused on the server portion of the TOE.

5.2.4.7 Protected authentication feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only **[obscured feedback]** to the user while the authentication is in progress.

5.2.4.8 User identification before any action (FIA_UID.2a)

FIA_UID.2a.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application note: This requirement is specifically focused on the client portion of the TOE.

5.2.4.9 User identification before any action (FIA_UID.2b)

FIA_UID.2b.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application note: This requirement is specifically focused on the server portion of the TOE.

5.2.5 Security management (FMT)

5.2.5.1 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the **[Data Access Control Policy]** to restrict the ability to **[*manage*]** the security attributes **[of users]** to **[Authorized Administrators]**.

Application note: The term 'manage' in the SFR above is intended to ensure that ANY means of manipulation of the applicable security attributes is appropriately controlled, rather than limiting the protection to a discrete set of possible management operations.

5.2.5.2 Secure security attributes (FMT_MSA.2)

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

5.2.5.3 Static attribute initialization (FMT_MSA.3)

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

FMT_MSA.3.2 The TSF shall allow the **[no role]** to specify alternative initial values to override the default values when an object or information is created.

5.2.5.4 Management of TSF data (FMT_MTD.1a)

FMT_MTD.1a.1 The TSF shall restrict the ability to **[*[search]*]** the **[audit trail]** to **[Authorized Administrators]**.

5.2.5.5 Management of TSF data (FMT_MTD.1b)

FMT_MTD.1b.1 The TSF shall restrict the ability to [[**create**], delete, modify] the [**Enterprise Alert configuration**] to [**Enterprise Administrators**].

5.2.5.6 Management of TSF data (FMT_MTD.1c)

FMT_MTD.1c.1 The TSF shall restrict the ability to **[*query, modify*]** the **[security-relevant roles] to [Policy Administrators]**.

5.2.5.7 Management of TSF data (FMT_MTD.1d)

FMT_MTD.1d.1 The TSF shall restrict the ability to **[*[initialize], delete*]** the **[encryption keys]** to **[Authorized Administrators]**.

5.2.5.8 Management of TSF data (FMT_MTD.1e)

FMT_MTD.1e.1 The TSF shall restrict the ability to **[*modify*]** the **[authentication data]** to **[Authorized Administrators, and users (for their own authentication data)]**.

5.2.5.9 Management of TSF data (FMT_MTD.1f)

FMT_MTD.1f.1 The TSF shall restrict the ability to **[*query, modify*]** the **[authentication data policy settings (such as the type of mechanism used, and the requirements for the method] to [Policy Administrators]**.

5.2.5.10 Management of TSF data (FMT_MTD.1g)

FMT_MTD.1g.1 The TSF shall restrict the ability to **[*query, modify*]** the **[Authentication failure settings]** to **[Policy Administrators]**.

5.2.5.11 Management of limits on TSF data (FMT_MTD.2)

FMT_MTD.2.1 The TSF shall restrict the specification of the limits for **[Authentication failure settings]** to **[Policy Administrators]**.

FMT_MTD.2.2 The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: **[**

- **On FileArmor, one of the following:**
  - **Lock the user account until unlocked by an Authorized Administrator with Remote Authentication**
  - **Specify a Time Delay before further login attempts**
- **On PolicyServer Management console**
  - **Lock the user account until unlocked by a Policy Administrator].**

5.2.5.12 Revocation (FMT_REV.1)

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the **[*users*]** under the control of the TSF to **[Policy Administrators]**.

FMT_REV.1.2 The TSF shall enforce the rules **[**

- **Revocation will take place at the current login on a device in contact with the server, or**
- **At the next login after connection to the server if the device is not in contact at the time of the login].**

5.2.5.13 Time-limited authorization (FMT_SAE.1)

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for [**fixed password authentication data for the User role used for FileArmor logins]** to **[Policy Administrators]**.

FMT_SAE.1.2 For each of these security attributes, the TSF shall be able to **[prevent user login until the authentication data is successfully changed]** after the expiration time for the indicated security attribute has passed.

5.2.5.14 Specification of Management Functions (FMT_SMF.1)

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

- **Query the audit trail**
- **Manage user security attributes**
- **Management of enterprise alerts**
- **Management of device policy settings and security attributes**
- **Manage authentication policy settings].**

5.2.5.15 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles **[**

- **Enterprise Administrator**
- **Group Administrator**
- **Enterprise Authenticator**
- **Group Authenticator**
- **User].**

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

5.2.6 Protection of the TSF (FPT)

5.2.6.1 Failure with preservation of secure state (FPT_FLS.1a)

FPT_FLS.1a.1 The TSF shall preserve a secure state when the following types of failures occur: **[communications with the PolicyServer are not available]**.

5.2.6.2 Failure with preservation of secure state (FPT_FLS.1b)

FPT_FLS.1b.1 The TSF shall preserve a secure state when the following types of failures occur: **[policies are found to not have the proper HMAC value]**.

5.2.6.3 Basic internal TSF data transfer protection (FPT_ITT.1)

FPT_ITT.1.1 The TSF shall protect TSF data from **[*disclosure, modification*]** when it is transmitted between separate parts of the TOE.

Application note: The PolicyServer component encrypts its communication with FileArmor by encrypting and decrypting SOAP messages sent and received using HTTP.

5.2.6.4 TSF testing (FPT_TST.1)

FPT_TST.1.1 The TSF shall run a suite of self tests **[*during initial start-up*]** to demonstrate the correct operation of **[*the TSF*]**.

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of [***Mobile Armor Cryptographic Module***].

FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of stored TSF executable code.

5.2.7 Resource Utilization (FRU)

5.2.7.1 Degraded fault tolerance (FRU_FLT.1)

FRU_FLT.1.1 The TSF shall ensure the operation of **[normal user functionality and device access]** when the following failures occur: **[communications with the PolicyServer are not available]**.

5.2.8 TOE Access (FTA)

5.2.8.1 Default TOE access banner (FTA_TAB.1)

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

Application note: This requirement is specifically focused on the server portion of the TOE.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 4 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.4: Complete functional specification
	ADV_IMP.1: Implementation representation of the TSF
	ADV_TDS.3: Basic modular design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.4: Production support, acceptance procedures and automation
	ALC_CMS.4: Problem tracking CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_DVS.1: Identification of security measures
	ALC_FLR.3: Systematic flaw remediation
	ALC_LCD.1: Developer defined life-cycle model
	ALC_TAT.1: Well-defined development tools
ATE: Tests	ATE_COV.2: Analysis of coverage
	ATE_DPT.2: Testing: security enforcing modules
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.3: Focused vulnerability analysis

Table 3 - EAL 4 Assurance Components

5.3.1 Development (ADV)

5.3.1.1 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 untrusted 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 initialisation 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.

5.3.1.2 Complete functional specification (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 security enforcing effects and exceptions associated with 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.

5.3.1.3 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.

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.

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.

5.3.1.4 Basic modular 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.

ADV_TDS.3.8c The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, 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.

5.3.2 Guidance documents (AGD)

5.3.2.1 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.

5.3.2.2 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.

5.3.3 Life-cycle support (ALC)

5.3.3.1 Production support, acceptance procedures and automation (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.4.1c 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 authorised 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.

5.3.3.2 Problem tracking CM coverage (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.

5.3.3.3 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.

5.3.3.4 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.

5.3.3.5 Systematic flaw remediation (ALC_FLR.3)

ALC_FLR.3.1d The developer shall document flaw remediation procedures addressed to TOE developers.

ALC_FLR.3.2d 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.3.3d The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.3.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.3.2c 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.3.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.3.4c 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.3.5c 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.3.6c The flaw remediation procedures shall include a procedure requiring timely response and the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.

ALC_FLR.3.7c 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.3.8c 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.3.9c 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.3.10c The flaw remediation guidance shall describe a means by which TOE users may register with the developer, to be eligible to receive security flaw reports and corrections.

ALC_FLR.3.11c The flaw remediation guidance shall identify the specific points of contact for all reports and enquiries about security issues involving the TOE.

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

5.3.3.6 Developer defined life-cycle model (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.

5.3.3.7 Well-defined development tools (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.

5.3.4 Tests (ATE)

5.3.4.1 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.

5.3.4.2 Testing: security enforcing modules (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 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.

5.3.4.3 Functional testing (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.

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.

5.3.4.4 Independent testing - sample (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 interfaces to confirm that the TSF operates as specified.

5.3.5 Vulnerability assessment (AVA)

5.3.5.1 Focused 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.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 Security audit

The TOE generates audit records for start-up and shutdown of the TOE components which generate audit records, as well as the “not specified” level of audit as defined in the Common Criteria. Audit records are generated on FileArmor and the PolicyServer.

The auditable events recorded by FileArmor include:

- Start-up and shutdown of FileArmor client which generates audit records
- Success of cryptographic operations;
- Identification and authentication attempts;
- Remote Authentication Mechanism usage;

The auditable events recorded by PolicyServer include:

- Start-up and shutdown of the PolicyServer
- Successful requests to perform an operation on an object covered by the SFP;
  - Use of the management functions, specifically, manage user security attributes through the PolicyServer Management Console
- Use of the management functions, specifically,
  - Manage user and group security attributes,
  - Manage policy settings, and
  - Manage authentication server policy settings;
- Modifications to the group of users that are part of a role;
- Identification and authentication attempts;
- Rejection of unacceptable password changes;

FileArmor generates all audit events associated with its own users (e.g., attempts to login and decrypt protected data) on the client. Those audit records are created (using the specific event type, time/date from the local operating system, the TOE user identity, and success or failure of the event), stored in the FADB, and then sent in encrypted SOAP messages (see section 6.6) to the PolicyServer, which then records the logs into the PolicyServer log database. While these records are stored on the client, the client is able to detect the deletion of the log. The audit log is started/stopped when the FileArmor client is started/stopped.

The PolicyServer generates all audit events associated with security management of the TOE as well as authentication attempts. Each record identifies the event, time/date from the local operating system, the user identity, and success or failure of the event.

The PolicyServer component of the TOE provides a management console that can be used to read from the PolicyServer log database used to store audit records and to generate reports. The management console requires identification and authentication of the administrators, restricting the number of users able to review the audit log. These administrators are able to fully search the collected audit records, and then can export the results of their search into a report for use outside the TOE. The TOE also provides several built-in reports which can be used to gather information about the health of the system.

While the IT Environment should protect the audit trail from tampering once they have been uploaded to the PolicyServer, a mechanism is provided for detecting if changes have occurred in the audit records once they have been stored. As each record is recorded in the PolicyServer log database (whether uploaded from FileArmor or directly entered from the PolicyServer) an HMAC value is generated and stored in the log database. This value also contains entry information, allowing the sequence of events entered into the log database to be determined. Enterprise Administrators can define a scheduled process be set to run to verify the records and detect potential changes to the log database. This process, managed by the TOE (and not reliant on OS task scheduling) will verify the HMAC values to check for the integrity of the audit records as well as completeness (that none have been deleted or false entries added).

FileArmor and PolicyServer are able to report on potential audit violations based on reviews of the audit records.

FileArmor reports potential violations to the next user to successfully authenticate, ensuring a local user knows that potential violations have been noted.

On the PolicyServer, Enterprise Administrators are able to schedule and run alerts on all uploaded logs, and inform administrators via email when potential violation events have been reported. The PolicyServer maintains a scheduling system for running the alerts. Each scheduled alert can have the email addresses set uniquely for specific administrators to allow customized notifications. Any time an alert generates results (i.e. does not return an empty query from the log database) a PDF is generated and sent in an email to the specified addresses.

These alerts are used to track potential violations on the client and server. In addition the PolicyServer monitors the integrity of all the policy configuration values in the policy database and generates audit events for each value that fails the integrity check.

The following potential violations are reported by PolicyServer by email from a scheduled alert notification:

- Multiple failed login attempts on FileArmor that reach the threshold defined by the administrator for the alert
- Policy value tampering events identified through invalid policy HMAC values in the PolicyServer policy database
- Log tampering events identified through invalid policy HMAC values in the PolicyServer log database

The Security audit function is designed to satisfy the following security functional requirements:

- **FAU_ARP.1ab** and **FAU_SAA.1ab**: The TOE detects several potential security violations and provides reports about them. On FileArmor, the violation is reported to the next authenticated user.
- **FAU_GEN.1**: The TOE generates audit records for the minimum level of audit. The operating system is relied on to provide reliable time stamps for use in audit records. Audit records are stored in a database in the IT environment. The operating system and database server in the IT environment is relied on to protect audit data from tampering.
- **FAU_GEN.2**: The TOE generates audit records that include the identity of the user that caused the event.
- **FAU_SAR.1**: The TOE provides administrative interfaces that can be used to read from the database stored in the IT environment.

- **FAU_SAR.2**: The TOE provides restricted access to the audit records by requiring the user to be an authorized administrator.
- **FAU_SAR.3**: The TOE provides the ability to search the audit records which have been uploaded to the PolicyServer log database.
- **FAU_STG.1a**: The TOE provides the ability to prevent modifications to the audit trail while it is stored on the clients.
- **FAU_STG.1b**: The TOE provides the ability to detect modifications to the audit trail while it is stored in the log database.

6.2 Cryptographic support

Both the FileArmor and PolicyServer applications use instances of the same FIPS-evaluated Mobile Armor cryptomodules to perform all cryptographic operations. The cryptomodules only operate in FIPS mode.

FileArmor comes with two versions of the Mobile Armor Cryptographic Module, v3.0 and v3.5. Version 3.5 is built from 3.0 with no changes to the 3.0 core code, but adds enhancements to support FIPS 140-2 Level 2 validation (version 3.0 supports only FIPS 140-2 Level 1 validation). This allows the FileArmor client to operate in the highest FIPS 140-2 Level available on the installed operating system. The PolicyServer only uses v3.5 of the module. The following tables list the algorithm and module certificates for these modules.

Algorithm	v3.0 Certificate	v3.5 Certificate
AES	820	920
Triple-DES (3DES)	692	740
SHS (all SHA)	818	907
ANSI X9.31 PRNG	472	528
HMAC	453	514

Table 4 - Cryptographic Module Algorithm Certificates

Module Version	Certificate
v3.0	1134
v3.5	1123

Table 5 - Cryptographic Module Certificates

Note that while Triple-DES (3DES) is listed as a validated algorithm, it is not claimed directly in the TOE. This algorithm is included indirectly as the basis for the ANSI X9.31 PRNG.

When the PolicyServer is installed, its cryptomodule is used to generate data that is used to create a 256-bit company-unique AES encryption key. The data necessary to generate this key is stored in the PolicyServer policy database and is used for encrypting FileArmor File Encryption Keys (FEKs) for escrow purposes to support any necessary FileArmor recovery operations. The keys are not stored on the local disk.

The PolicyServer uses its cryptomodule to create 256-bit FEKs for each Enterprise, Group and User of FileArmor for use with AES.

These keys are sent, encrypted using a Diffie--Hellman-negotiated session key, from the PolicyServer to the FileArmor instance once when the FileArmor instance is initially defined in the PolicyServer. Unlike FEKs, a new CEK is generated each time a client connects to the PolicyServer, with a unique

CEK for each client connected at the time. Once the communications channel between the client and server is closed, the CEK is securely deleted from memory on both sides and not used again.

Each FileArmor instance includes its own cryptomodule and, when installed, receives its FEKs from the associated PolicyServer. The keys are encrypted using a 256-bit key (derived from a hash of the current user's identity and authentication data) with AES and stored in the FADB. The FEKs sent to the user are determined by the group membership of the user associated with the computer where FileArmor is installed.

Once the FileArmor client is installed, each time the OS is started the FileArmor application will start and present the user a login dialog. The user must identify and authenticate to FileArmor. The user identity and authentication data is hashed to create a 256-bit AES key. This key is used to decrypt the FEKs and if the key cannot be decrypted the authentication will fail. Once the FEKs are decrypted, any protected files can be decrypted using the appropriate FEK automatically, without further user intervention via the operating system where the FEKs are maintained in the memory of the device driver in the OS kernel. The FEKs are protected in memory by the secure execution environment provided by the OS.

Furthermore, once the FileArmor product is installed, two-way communications between the PolicyServer and FileArmor instance are implemented using SOAP messages sent over the HTTP protocol. The FileArmor client uses its cryptomodule to generate unique 256-bit AES Communication Encryption Keys (CEK) for each client session for the purpose of protecting communication between the PolicyServer and the FileArmor instance. Unlike the DEK, a new CEK is generated each time a client connects to the PolicyServer, with a unique CEK for each client connected at the time. Once the communications channel between the client and server is closed, the CEK is securely deleted from memory on both sides and not used again. The Body of each SOAP message is encrypted using the 256-bit CEK with AES. One-way communications from the PolicyServer to mobile device clients is encrypted on the server using the appropriate device's DEK.

The Cryptographic support function is designed to satisfy the following security functional requirements:

- **FCS_CKM.1**: The TOE generates keys for encryption and decryption of data using a FIPS-validated RNG.
- **FCS_CKM.3**: The TOE escrows the generated FEK keys on the server in accordance with FIPS 140-2.
- **FCS_CKM.4**: The TOE destroys keys stored temporarily in filter drivers by overwriting them in memory. The TOE will also destroy the keys from memory as well as from the disk for users when the user is deleted from the device.
- **FCS_COP.1abc**: The TOE provides its own FIPS-evaluated cryptographic module which performs symmetric encryption and decryption operations using AES. Additionally algorithms HMAC, RNG, and SHS are provided for additional security related to those functions.

6.3 User data protection

The client portion of the TOE implements the Data Access Control Policy which enforces access to TOE-protected objects such as encrypted files and folders. Access control decisions to determine access to TOE-protected information are based on user identities and authentication credentials which are used to decrypt the symmetric keys used to encrypt specified contents. Simply, if the user identity and authentication credentials can be successfully used to decrypt the symmetric key that decrypts the encrypted data, the user may access the unencrypted or plain-text data.

The User data protection function is designed to satisfy the following security functional requirements:

- **FDP_ACC.1** and **FDP_ACF.1**: The TOE provides the ability to restrict access to data stored on the device's persistent storage. All users are subject to the Data Access Control Policy for all available operations devices by a combination of authentication and encryption.

6.4 Identification and authentication

The TOE defines the following user security attributes:

- User identity (username)
- Authentication data
- Group membership
- Role
- File Encryption Keys (FEKs).

These attributes are stored, as appropriate, in different parts of the TOE as follows:

- On FileArmor:
  - User identity (username)
  - Group membership
  - Role
  - User's encrypted File Encryption Keys (FEKs)
- On the PolicyServer
  - User identity (username)
  - Authentication data as related to PolicyServer administration roles
  - Group membership
  - Role

To access the contents of encrypted files or folders, the TOE requires that users first use FileArmor login dialogs to authenticate. FileArmor authentication credentials are not stored directly, but are used to generate a hash value which is used to encrypt all FEKs associated with the user. The exception to this is when smart cards are used for authentication where a private key from the smart card is used to encrypt the FEKs. Successful authentication in FileArmor will generate the proper hash value to decrypt the FEKs.

FileArmor always attempts to authenticate via the PolicyServer (using encrypted SOAP messages, see section 6.6) to have the latest data in the FADB. By contacting the PolicyServer during authentication it is possible to enforce conditions such as user locking or removal. If the user is configured to use an external authentication server, the PolicyServer will act as a proxy for the authentication to the external server and then pass that information back to the FADB, ensuring the user must authenticate with the proper credentials. When the PolicyServer is not available, such as when FileArmor is off-line (i.e. using a laptop on an airplane), the authentication data currently in the FADB will be used for authentication.

FileArmor supports multiple users on a protected device. All users in a group can login to any PC device in the same group. This list would include all accounts with the User role, all Group Administrators and Group Authenticators in the group, as well as all Enterprise Administrators and Enterprise Authenticators.

The TOE requires administrators to use PolicyServer management console logon interfaces to identify and authenticate themselves. The PolicyServer authenticates administrators directly using credentials stored in the policy database. The PolicyServer will enforce the authentication decision so that only authentic administrators can access corresponding security functions.

In all login attempts, the user password credentials are obscured from display on the screen.

Before users can decrypt or encrypt data, users must first log into FileArmor login interfaces. When FileArmor starts, it prompts the user for credentials such as username and password for the authentication as summarized above.

FileArmor is designed to ensure that the contents of any encrypted file or folder are not accessible until after successful authentication. Further, the authenticated user must have access to the same FEK which was used to encrypt the file. For example, the contents of a file encrypted with a user FEK will not be accessible to a different user as the second user will not have access to the original user's user FEK. Files encrypted with Group FEKs (or Enterprise FEKs) can be accessed by any user in the same Group (or Enterprise). Files which are encrypted as self-extracting prompt for user authentication when the executable is run. Only successful authentication will extract the file.

When configured, authentication servers in the IT environment are relied on to maintain user identities and authentication data for FileArmor users. The TOE maintains user identities as well as information that identifies the authentication mechanism that the user must use (and when the TOE is authenticating users, the authentication data itself), and TOE-defined group information in the FADB. Additionally, when using the IT environment authentication credentials for login, it is possible to configure the FileArmor client for Single Sign-on from the OS login. When configured, the OS login credentials are used to automatically authenticate the FileArmor user. The Single Sign-on function only works when using fixed password authentication in a domain environment. The end result of this is the user will only see the OS login and will automatically be authenticated to FileArmor without further prompts.

FileArmor provides many different authentication mechanisms, from strong fixed passwords, smart cards, and Remote Authentication logins for situations where normal authentication is not possible (such as a forgotten password or a lost smart card).

The Remote Authentication Mechanism is designed as a one-time use method of authentication requiring the verbal assistance of an Authorized Administrator. When a user initiates this mechanism it generates a unique X9.9-based challenge and displays it to the user. This is read to the Authorized Administrator and input into the PolicyServer management console where a response is generated. This is read back to the user and entered into the login. This process will unlock the FEKs and force the user to immediately enter new authentication data to re-encrypt the FEKs for the user. A new X9.9 key is then generated to re-encrypt the FEKs for the next use, preventing the re-use of the same challenge and response.

In addition to configuring the authentication mechanisms available for a user, the administrator can configure the action to be taken after a specified number of failed login attempts. FileArmor supports locking of the File Armor client user account (Remote Authentication can be used to unlock the account) and setting a time delay before more login attempts can be made. Policy Server management console accounts will always lock the user account when the allowed number of invalid attempts have been made.

The User role does not have any administrative authority in the TOE, but is assigned access to FileArmor-protected devices through group membership. The TOE uses groups as a way to apply configuration settings (for example the authentication mechanism that must be used) to more than one user at a time, both to the PolicyServer and to FileArmor. Permissions inside the management console are determined by a combination of assigned role and group membership. Roles are described in the Security management description below. Similarly this combination is used to determine access to FileArmor clients.

FileArmor associates a TOE user with all FileArmor processes that are started after authentication. For example, the cryptographic module is started as a process for that TOE user, and all actions that are taken are specifically tied to that TOE user. PolicyServer associates a TOE user with all actions taken inside the PolicyServer management console. This ensures that all audit records are properly associated with the TOE user responsible for the process. Note that the TOE user is not necessarily the same as the OS user.

The TOE administrators configure an inactivity policy for the management console which will cause the administration console to become closed after a specified period of inactivity. When this happens, the user must authenticate to the TOE before regaining access to the console.

The Identification and authentication function is designed to satisfy the following security functional requirements:

- **FIA_AFL.1**, **FIA_ATD.1**, and **FIA_UAU.5**: The TOE maintains security attributes belonging to individual users and administrators including user identity and authentication mechanism identification information to accomplish this. The TOE also maintains a policy to control the actions to be taken if a number of unsuccessful login attempts occur in succession.
- **FIA_UAU.2** and **FIA_UID.2**: The TOE authenticates all users prior to access of the PolicyServer management console except to specify the name of the server and Enterprise to connect to.
- **FIA_UAU.6**: The TOE provides a re-authentication mechanism to ensure that inactive administration consoles are closed to prevent unauthorized access.
- **FIA_UAU.7**: The TOE prevents the display of the password portion of a user's credentials on the screen.

6.5 Security management (FMT)

The TOE defines the following roles:

- Enterprise Administrator
- Group Administrator
- Enterprise Authenticator
- Group Authenticator
- User

On the PolicyServer, a user that possesses the Enterprise Administrator role can access all PolicyServer management interfaces. A user that possess the Group Administrator role can manage user information (including roles, groups, and permissions) for one or more user groups that the user possessing this role has been assigned to. Similar access is given to Enterprise and Group Authenticators, respectively, though an Authenticator is not allowed to access the policies.

Authenticators can modify a subset of user data as follows: - UserID, FirstName, LastName, EmployeeID, email address, and assign a one-time password. An Authenticator cannot change the role of the user.

The specific management functions available to an Authenticator through the PolicyServer Management console are (subject to their position in the hierarchy):

- Review and search the audit trail
- Modify authentication data by forcing a password change with a onetime password

- Unlock FileArmor user accounts with Remote Authentication
- Initialize and delete encryption keys

Through the PolicyServer Management console, an Authenticator explicitly cannot modify the role of another user (preventing improper elevation of authority).

Permission information maintained by the PolicyServer identifies which groups a user that possesses the Group Administrator role has been assigned to. A user that possesses the User role is simply a user of the FileArmor application, i.e. a user who uses FileArmor, and provides no access to login to the PolicyServer.

Authority is assigned to the user account (a user account must exist to be given authority) within the group (or Enterprise) where the user needs to have the authority. When this user logs into the management console, they will only see what which they have access to from the authority granted. When a user account is assigned the Administrator or Authenticator role, the account is also assigned a separate, administrator credentials (fixed password). Without both the role and administrator credentials, it is not possible to login to the management console.

The TOE provides administrator console interfaces to perform the following:

- Search the audit trail
- Manage user security attributes
- Management of Enterprise Alerts
- Manage device policy settings and security attributes
- Manage authentication server policy settings

There are three types of FEK, user, enterprise and group. FEK's are used to encrypt/decrypt files on the client. There is also a device encryption key that is used to transport data between the Policy Server and the client. The user's File Encryption Key is created at the time the Authorized Administrator creates the user in the policy server. The group FEK is created when the group is created in the Policy Server by the Authorized Administrator. The enterprise FEK is created when the enterprise is created in the Policy Server by the Authorized Administrator. The device encryption key is created after the user registers with the policy server using the one time password supplied by the Authorized Administrator. The user and group FEK is then supplied, using the device encryption key to secure transportation, to the client form the Policy Server.

A user can be removed from the group or enterprise by an Authorized Administrator. Once removed, the FEK for that user are revoked. The next time that the client synchronizes with the Policy Server, the user FEK is removed.

The TOE provides a password expiration policy for accounts logging into the FileArmor clients with the User role. As a policy set in the PolicyServer Management console by Policy Administrators, it is possible to set a period of days before a password will expire. When setting an explicit date, once that date has been reached and the password changed, the next expiration time will be determined by the policy value. Authorized Administrators can force a password change by using the "Assign One Time Password" function. This allows the Authorized Administrator to set a new password for the next login (the client must be able to connect to the PolicyServer for this to take effect) which will automatically force the user to choose a new password.

There is an application-based administrative interface to the PolicyServer providing a GUI-based interface. TOE settings are configured using sets of rules that are grouped into policies. For example, “Password Policies” policy settings include “AuthenticationMethods” which specify which authentication methods are available to a user for login at the FileArmor client.

The FileArmor client provides protection for individual files and folders by employing encryption on the specified objects. A Policy can be set in the Policy Server that defines a list the folders that will be encrypted on the hard drive (device policy settings). Non-existent folders are created. A valid drive letter to the hard drive must also be supplied.

The FileArmor client provides the ability for the user to modify their own password. After successfully authenticating via login, the user can access the change password function. Even though they have been authenticated, they must provide the current password again before any change can be made. A new password can be entered and is confirmed before that change is made.

The Security management function is designed to satisfy the following security functional requirements:

- **FMT_MSA.1**: The ability to manage user security attributes is restricted to Authorized Administrators at the enterprise and group level.
- **FMT_MSA.2** and **FMT_MSA.3**: By default, all protected devices require authentication to access TOE-protected files which are encrypted and hence protected by default.
- **FMT_MTD.1a**: Authorized Administrators are able to search the audit trail.
- **FMT_MTD.1b**: Enterprise Administrators are able to manage the alert configuration for potential violation notifications.
- **FMT_MTD.1c and FMT_MTD.1f**: Policy Administrators are able to access and as appropriate, manage, the data on the PolicyServer including the audit trail, security roles, and authentication policies.
- **FMT_MTD.1d:** Authorized Administrators are able to initialize and delete the encryption keys.
- **FMT_MTD.1e**: Authorized Administrators are able to access and as appropriate, manage authentication data. Users are able to manage their own authentication data within the restrictions set by Policy Administrators (FMT_MTF.1f).
- **FMT_MTD.1g** and **FMT_MTD.2**: The TOE provides the ability to determine the results of successive failed FileArmor login attempts, to lock the user account, or specify a time delay before further login attempts. Control of these settings is limited to Policy Administrators.
- **FMT_REV.1**: The TOE provides the ability to revoke FileArmor user access to Policy Administrators. This access will be revoked at the current login on a device in contact with the server or at the next login where connectivity to the PolicyServer is available.
- **FMT_SAE.1**: The ability to manage fixed authentication data by requiring it to expire periodically is restricted to Policy Administrators.
- **FMT_SMF.1**: The TOE provides the ability to query audit records, manage Enterprise alerts, manage user security attributes, and device policy settings and security attributes, and authentication server policy settings.
- **FMT_SMR.1**: The TOE provides administrative, authenticator and user authority. These are further divided by the level of the hierarchy where the role is assigned to the user. This creates the roles of Enterprise Administrator, Enterprise Authenticator, Group Administrator, Group Authenticator and User.

6.6 Protection of the TSF

While the FileArmor application can be readily bypassed after installation by choosing not to login, access to the FEKs used for decrypting existing and encrypting new data are only available after successful authentication. This ensures that any data protected by FileArmor remains protected, even if FileArmor is removed from the computer. While FileArmor is being loaded, it implements a series of tests on itself to ensure it is functioning properly. The application and the filter drivers that use the FEKs are digitally signed and verified as they are loaded by the OS.

FileArmor is designed to be automatically started with the OS. Backend components, such as the services and filter driver are started automatically with the OS, though they are not active in terms of providing encryption or decryption until the user has successfully authenticated. After the FEKs have been decrypted using the authentication credentials, they are loaded into the filter driver where it is available to encrypt and decrypt encrypted contents automatically or manually for the user.

FileArmor attempts to contact the PolicyServer on a regular basis. This is done to provide ongoing and up to date information from the PolicyServer. In cases where FileArmor is unable to contact the PolicyServer it will continue to function using its last loaded configuration and policies to maintain the security of the TOE. When communications are re-established, any updated policies will be downloaded and all logs will be uploaded to the PolicyServer at that time.

The PolicyServer relies largely on the TOE environment for protection, but has been carefully designed to restrict access to its own functions in accordance with the TOE security policies by ensuring that users are authentic and that they are authorized to perform the functions they attempt. The PolicyServer further ensures that invalid policies cannot be sent to the clients by validating the HMAC values for all policy values before they are sent to the client. Any invalid value will prevent the policy update from being sent to the client until the policy has been updated (this will generate a new HMAC value along with ensuring the current value itself). When invalid policy values are discovered, a log entry is created specifying the invalid policy. In addition to protecting the policy values, the server can be scheduled to verify the integrity of the log entries that have been stored on the PolicyServer (see section 6.1).

All two way communications between the PolicyServer and FileArmor use SOAP messages sent over HTTP. FileArmor relies on the OS to provide access to the network, but nothing else for connectivity. The Body of each SOAP message is encrypted using a 256-bit AES CEK that was generated uniquely for each communications session and from each device. As such, all policy information, audit records, authentication data, etc. that passes among the TOE components are protected from modification and disclosure and are all acknowledged by both client and server.

Once the session has ended, both the client and server will securely delete the CEK from memory. The next time the client connects to the server, a new CEK is generated.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- **FPT_TST.1**: The TOE client ensures that the security is functioning properly when the device is initialized by running a series of checks on the integrity of the software.
- **FPT_FLS.1a**: The ability to maintain a secure configuration in the advent of a non-connected state with the PolicyServer is provided.
- **FPT_FLS.1b**: The ability to maintain a secure configuration in the advent invalid policy values are found.
- **FPT_ITT.1**: The PolicyServer component encrypts its communication with FileArmor by encrypting and decrypting SOAP messages sent and received using HTTP.

6.7 Resource Utilization

FileArmor attempts to contact the PolicyServer on a regular basis. This is done to provide ongoing and up to date information from the PolicyServer. In cases where FileArmor is unable to contact the PolicyServer it will continue to function using its last loaded configuration and policies to maintain the security of the TOE. When communications are re-established, any updated policies will be downloaded and all logs will be uploaded to the PolicyServer at that time.

The Resource Utilization function is designed to satisfy the following security functional requirements:

- **FRU_FLT.1**: The ability to maintain a secure configuration in the advent of a non-connected state with the PolicyServer is provided.

6.8 TOE Access

.The PolicyServer management console provides a login banner that can be configured by Enterprise Administrators. This banner can contain any pertinent information as determined by the organization about the use of the console. The banner can be set at the Enterprise level and is presented prior to login of the Policy Server Management console. Banners at the group level or banner timeouts are not in scope of this evaluation.

The TOE Access function is designed to satisfy the following security functional requirements:

- **FTA_TAB.1**: Policy Administrators have the ability to create and manage login banners on the PolicyServer console.

7. Protection Profile Claims

There is no Protection Profile claim in this Security Target.

8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Requirement Dependencies; and
- TOE Summary Specification.

8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

8.1.1 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of organizational policies and usage assumptions by the security objectives.

	O.ACCESS	O.ADMIN_ROLE	O.ALERT	O.AUDIT_GNEERATION	O.AUDIT_REVIEW	O.CRYPTO_OPS	O.DATA_TRANSFER	O.FAULT_TOLERANCE	O.MANAGE	O.TOE_PROTECTION	O.USER_AUTHENTICATION	OE.AUDIT_PROTECTION	OE.TIME	OE.TOE_PROTECTION	OE.CONFIG	OE.PHYCAL	OE.USER_GUIDANCE
T.ACCOUNTABILITY			X	X	X							X	X				
T.ADMIN_ERROR		X							X								
T.MASQERADE											X						
T.SUBVERT						X											
T.TSF_COMPROMISE							X	X		X				X			
T.UNAUTH_ACCESS	X																
A.LOCATE																X	
A.NO_EVIL															X		
A.NO_EVIL_USER																	X
A.DEVICE_USE																	X

Table 6 - Environment to Objective Correspondence

8.1.1.1 T.ACCOUNTABILITY

A user may not be held accountable for their actions.

This Threat is countered by ensuring that:

- O.ALERT: The TOE will provide the capability to alert authorized users of potential security violations.
- O.AUDIT_GENERATION: The TOE will provide the capability to create records of security relevant events associated with users.
- OE.AUDIT_PROTECTION: The TOE Environment will provide the capability to protect audit information on the server portion of the TOE.
- O.AUDIT_REVIEW: The TOE will provide the capability to view audit information.
- OE.TIME: The TOE Environment will provide reliable time stamps for use in audit records so that audit records are associated with when the events occurred.

8.1.1.2 T.ADMIN_ERROR

An authorized administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

This Threat is countered by ensuring that:

- O.ADMIN_ROLE: The TOE will provide authorized administrator roles to isolate administrative actions.
- O.MANAGE: The TOE will allow administrators to effectively manage the TOE and its security functions, and must ensure that only authorized administrators are able to access such functionality.

8.1.1.3 T.MASQUERADE

An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.

This Threat is countered by ensuring that:

- O.USER_AUTHENTICATION: The TOE will ensure either by itself or using services of the TOE environment that user identities are authentic prior to allowing access to its functions.

8.1.1.4 T.SUBVERT

A malicious user may cause non-configuration data at rest to be inappropriately accessed (viewed, modified or deleted).

This Threat is countered by ensuring that:

- O.CRYPTO_OPS: All data at rest on a TOE-protected device is encrypted using FIPS-validated algorithms and modules.

8.1.1.5 T.TSF_COMPROMISE

A malicious user may cause configuration data to be inappropriately accessed (viewed, modified or deleted).

This Threat is countered by ensuring that:

- O.DATA_TRANSFER: The TOE encrypts all data transferred between the client and server and acknowledges the receipt of the transfers.
- O.FAULT_TOLERANCE: The TOE must continue to enforce access control policies if communications with the server are not available or if the server has detected violations of policy integrity.
- O.TOE_PROTECTION: The client portion of the TOE will protect itself and its assets from external interference or tampering.
- OE.TOE_PROTECTION: The TOE Environment will protect the server portion of the TOE and its assets from external interference or tampering.

8.1.1.6 T.UNAUTH_ACCESS

A user may gain unauthorized access (view, modify, delete) to configuration data.

This Threat is countered by ensuring that:

- O.ACCESS: The TOE will ensure that users gain only authorized access to the TOE and to the resources that the TOE controls.

8.1.1.7 A.LOCATE

The server portion of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

This Assumption is satisfied by ensuring that:

- OE.PHYCAL: The server portion of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

8.1.1.8 A. NO_EVIL

The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation.

This Assumption is satisfied by ensuring that:

- OE.CONFIG: The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation.

8.1.1.9 A. NO_EVIL_USER

Users of the TOE are properly trained in the use of the TOE and will cooperate with those responsible for administration in maintaining TOE security.

This Assumption is satisfied by ensuring that:

- OE.USER_GUIDANCE: Users of the TOE will be properly trained in the secure usage and protection of the TOE and the devices where it is installed.

8.1.1.10 A. DEVICE_USE

Users of the TOE will follow policies to prevent unauthorized physical access to a TOE-protected device.

This Assumption is satisfied by ensuring that:

- OE.USER_GUIDANCE: Users of the TOE will be properly trained in the secure usage and protection of the TOE and the devices where it is installed.

8.2 Security Requirements Rationale

All Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

Security Functional Requirements	O.ACCESS	O.ADMIN_ROLE	O.ALERT	O.AUDIT_GENERATION	O.AUDIT_REVIEW	O.CRYPTO_OPS	O.DATA_TRANSFER	O.FAULT_TOLERANCE	O.MANAGE	O.TOE_PROTECTION	O.USER_AUTHENTICATION
FAU_ARP.1a			X								
FAU_ARP.1b			X								
FAU_GEN.1				X							
FAU_GEN.2				X							
FAU_SAA.1a			X								
FAU_SAA.1b			X								
FAU_SAR.1					X						
FAU_SAR.2					X						
FAU_SAR.3					X						
FAU_STG.1a					X						
FAU_STG.1b					X						
FCS_CKM.1	X					X					
FCS_CKM.3						X					
FCS_CKM.4	X					X					
FCS_COP.1a	X					X					
FCS_COP.1b	X					X					
FCS_COP.1c	X					X					
FDP_ACC.1	X										
FDP_ACF.1	X										
FIA_AFL.1											X
FIA_ATD.1									X		X
FIA_UAU.2a											X
FIA_UAU.2b											X
FIA_UAU.5											X
FIA_UAU.6											X
FIA_UAU.7											X
FIA_UID.2a											X
FIA_UID.2b											X
FMT_MSA.1									X		
FMT_MSA.2									X		
FMT_MSA.3									X		

Security Functional Requirements	O.ACCESS	O.ADMIN_ROLE	O.ALERT	O.AUDIT_GENERATION	O.AUDIT_REVIEW	O.CRYPTO_OPS	O.DATA_TRANSFER	O.FAULT_TOLERANCE	O.MANAGE	O.TOE_PROTECTION	O.USER_AUTHENTICATION
FMT_MTD.1a									X		
FMT_MTD.1b									X		
FMT_MTD.1c									X		
FMT_MTD.1d									X		
FMT_MTD.1e									X		
FMT_MTD.1f									X		
FMT_MTD.1g									X		
FMT_MTD.2									X		
FMT_REV.1									X		
FMT_SAE.1									X		
FMT_SMF.1									X		
FMT_SMR.1		X							X		
FPT_FLS.1a								X			
FPT_FLS.1b								X			
FPT_ITT.1							X			X	
FPT_TST.1										X	
FRU_FLT.1								X			
FTA_TAB.1	X										

Table 7 - Objective to Requirement Correspondence

8.2.1.1 O.ACCESS

The TOE will ensure that users gain only authorized access to the TOE and to the resources that the TOE controls.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1, FDP_ACF.1: The TOE provides the ability to encrypt and decrypt specified files and folders. All users are subject to the Data Access Control Policy for all actions related to the encryption and decryption of data protected by the TOE.
- FCS_CKM.1, FCS_CKM.4 and FCS_COP.1abc: The TOE provides its own FIPS-evaluated cryptographic module with applicable FIPS-validated algorithms which performs symmetric encryption and decryption operations, as well as key generation and destruction. These are used to control access to data at rest and secure communications between FileArmor and PolicyServer.
- FTA_TAB.1: The TOE provides a means for providing a warning or appropriate use message when the user is attempting to gain access to the persistent storage and the management console.

8.2.1.2 O.ADMIN_ROLE

The TOE will provide authorized administrator roles to isolate administrative actions.

This TOE Security Objective is satisfied by ensuring that:

- FMT_SMR.1: The TOE provides administrative, authenticator and user authority. These are further divided by the level of the hierarchy where the role is assigned to the user. This creates the roles of Enterprise Administrator, Enterprise Authenticator, Group Administrator, Group Authenticator and User. The user role corresponds to users who access TOE-protected data.

8.2.1.3 O.ALERT

The TOE will provide the capability to alert authorized users of potential security violations.

This TOE Security Objective is satisfied by ensuring that:

- FAU_ARP.1ab and FAU_SAA.1ab: The TOE has rules which analyze the audit records and provides alerts when specific events, or sequences of events, have been detected.

8.2.1.4 O.AUDIT_GENERATION

The TOE will provide the capability to create records of security relevant events associated with users.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: The TOE generates audit records for the minimum level of audit that. The operating system is relied on to provide reliable time stamps for use in audit records. Audit records are stored in audit trail files in the TOE environment. The operating system in the TOE environment is relied on to protect audit trail files.
- FAU_GEN.2: The TOE generates audit records that include the identity of the user that caused the event.

8.2.1.5 O.AUDIT_REVIEW

The TOE will provide the capability to view audit information.

This TOE Security Objective is satisfied by ensuring that:

- FAU_SAR.1 and FAU_SAR.2: The TOE provides administrators an interface that can be used to read from log files stored in the TOE environment.
- FAU_SAR.3: The TOE provides administrators with the ability to search through the audit records and then to further sort that data based on the results. The administrator can then generate reports from this information that can be presented outside the administrative interface.
- FAU_STG.1a: Records stored in the client audit trail will be protected against modification to ensure availability for review until uploaded to the server.
- FAU_STG.1b: Records stored in the server audit trail will be protected such that it is possible to detect modification of the records to ensure integrity for review.

8.2.1.6 O.CRYPTO_OPS

The TOE will ensure that all cryptographic operations are compliant with FIPS 140-2 Level 1 & 2 based on installed OS (cryptographic module), FIPS 197 (AES), FIPS 46-3 (3DES), FIPS 180-2

(SHS), FIPS 198 (HMAC) and ANSI X9.31 (RNG) and that the keys for those operations are managed accordingly.

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM.1 and FCS_COP.1abc: The TOE utilizes only FIPS-validated cryptographic modules and algorithms for encryption and decryption operations. These functions are provided transparently to the user.
- FCS_CKM.3: The cryptographic keys used by the TOE to enforce the Data Access Policy are escrowed for recovery purposes in accordance with best practices.
- FCS_CKM.4: The TOE ensures that all keys used for cryptographic operations are properly deleted and cannot be recovered once they are deleted in accordance with FIPS 140-2 requirements.

8.2.1.7 O.DATA_TRANSFER

The TOE will protect system data in transmission between the client and server.

This TOE Security Objective is satisfied by ensuring that:

- FPT_ITT.1: The TOE ensures that all communications between the client and server portions of the TOE are protected from disclosure through the use of encrypted communications.

8.2.1.8 O.FAULT_TOLERANCE

The TOE must continue to enforce access control policies if communications with the server are not available.

This TOE Security Objective is satisfied by ensuring that:

- FPT_FLS.1a and FRU_FLT.1: The client portion of the TOE ensures that the Data Access Control Policy remains in effect at all times, even when communications with the server portion of the TOE are not available.
- FPT_FLS.1b: The PolicyServer portion of the TOE ensures that the Data Access Control Policy remains in effect at all times, even when invalid policy values are found on the server, but not allowing invalid values to be passed to the client.

8.2.1.9 O.MANAGE

The TOE will allow administrators to effectively manage the TOE and its security functions, and must ensure that only authorized administrators are able to access such functionality.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MTD.1g and FMT_MTD.2: The TOE provides an administrator with the ability to configure the consequences of failed user authentication to the TOE.
- FIA_ATD.1: The TOE maintains security attributes belonging to individual users and administrators including user identity and authentication mechanism identification information to accomplish this.
- FMT_MTD.1ef and FMT_SAE.1: The TOE provides an administrator with the ability to manage the user authentication requirements. The administrator can choose the types of mechanisms are available to the user as well as conditions on the user of the mechanisms.
- FMT_MTD.1d: The ability to manage device encryption keys is restricted to Authorized Administrators.FMT_MTD.1a: Authorized Administrators are able to search the audit trail.

- FMT_MSA.1: The ability to manage user security attributes, most importantly those used for access control decisions is restricted to Authorized Administrators.
- FMT_MSA.2: The TOE utilizes FIPS-validated cryptographic modules which ensure that all cryptographic functions utilize secure values.
- FMT_MSA.3: By default, access to a protected device is minimal and all data at rest is encrypted.
- FMT_SMF.1: The TOE provides the ability to manage user and group security attributes, and device policy settings, and authentication server policy settings.
- FMT_MTD.1b: The ability to manage the alert configuration for potential violation notifications is restricted to Enterprise Administrators.
- FMT_MTD.1c: The ability to query or modify security relevant roles is restricted to Policy Administrators.
- FMT_SMR.1: The TOE provides administrator, authenticator and user roles. Administrative users who posses either one of Enterprise Administrator or Group Administrator roles are considered authorized administrators. An authenticator possesses either Enterprise Authenticator or Group Authenticator roles. The user role corresponds to users who access TOE-protected client devices. The ability to manage user security attributes, including role and group assignments, is restricted to Policy Administrators

8.2.1.10 O.TOE_PROTECTION

The TOE will protect the TOE and its assets from external interference or tampering.

This TOE Security Objective is satisfied by ensuring that:

- FPT_TST.1: The TOE client ensures that the security is functioning properly when the device is initialized by running a series of checks on the integrity of the software.
- FPT_ITT.1: The TOE ensures that all communications between the client and server portions of the TOE are protected from disclosure through the use of encrypted communications.

8.2.1.11 O.USER_AUTHENTICATION

The TOE will ensure that users are reliably identified and are authenticated before any access to TOE-protected assets is granted.

This TOE Security Objective is satisfied by ensuring that:

- FIA_AFL.1: Users authenticating to the TOE are subject to restrictions on the number of times the login attempts can fail before the TOE initiates a corrective action determined by an administrator.
- FIA_ATD.1: The TOE will either authenticate users directly or invoke services of the TOE environment to ensure that users are authenticated properly.
- FIA_UAU.2a and FIA_UID.2a: The TOE will require that users be authenticated before any access is granted to a protected data.
- FIA_UAU.2b and FIA_UID.2b: The TOE authenticates all users prior to access of the PolicyServer management console.
- FIA_UAU.5: Users of the TOE are able to utilize one of many different types of authentication mechanisms. Some of these mechanisms may be used in combination under specific circumstances.

- FIA_UAU.6: After a specified period of inactivity, users will be forced to re-authenticate to the administration console.
- FIA_UAU.7: Entered authentication data will be obscured from view on the device.

8.3 Security Assurance Requirements Rationale

EAL4 was selected as the assurance level because the TOE is a commercial product whose users require a moderate to high level of independently assured security. The TOE is targeted at a relatively benign environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have little attack potential.

This ST contains the assurance requirements from the CC EAL4 assurance package augmented with ALC_FLR.3. The CC allows assurance packages to be augmented, which allows the addition of assurance components from the CC not already included in the EAL. Augmentation was chosen to provide the added assurance that is provided by defining flaw remediation procedures and correcting security flaws (ALC_FLR.3). This ST is based on good rigorous commercial development practices and has been developed for a generalized environment for a TOE that is generally available and does not require modification to meet the security needs of the environment specified in this ST.

8.4 Requirement Dependency Rationale

The following table demonstrates that all dependencies among the claimed security requirements are satisfied, except as explicitly noted below, and therefore the requirements work together to accomplish the overall objectives defined for the TOE.

ST Requirement	CC Dependencies	ST Dependencies
FAU_ARP.1a	FAU_SAA.1a	FAU_SAA.1a
FAU_ARP.1b	FAU_SAA.1b	FAU_SAA.1b
FAU_GEN.1	FPT_STM.1	FPT_STM.1 is not satisfied by the TOE. Rather, the TOE is dependent upon its environment to provide time stamps in support of audit record generation.
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN.1 and FIA_UID.1
FAU_SAA.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FAU_STG.1a	FAU_GEN.1	FAU_GEN.1
FAU_STG.1b	FAU_GEN.1	FAU_GEN.1
FCS_CKM.1	(FCS_CKM.2 or FCS_COP.1) and FCS_CKM.4 and FMT_MSA.2	FCS_COP.1, FCS_CKM.4 and FMT_MSA.2
FCS_CKM.3	(FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1), FCS_CKM.4 and FMT_MSA.2	FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2
FCS_CKM.4	(FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1) and FMT_MSA.2	FCS_CKM.1 and FMT_MSA.2
FCS_COP.1a	(FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4 and FMT_MSA.2	FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2
FCS_COP.1b	(FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4 and FMT_MSA.2	FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2

ST Requirement	CC Dependencies	ST Dependencies
FCS_COP.1c	(FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4 and FMT_MSA.2	FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.2 and FMT_MSA.3
FIA_AFL.1	FIA_UAU.2	FIA_UAU.2
FIA_ATD.1	none	none
FIA_UAU.2a	FIA_UID.2a	FIA_UID.2a
FIA_UAU.2b	FIA_UID.2b	FIA_UID.2b
FIA_UAU.5	none	none
FIA_UAU.6	none	None
FIA_UAU.7	FIA_UAU.2	FIA_UAU.2
FIA_UID.2a	none	none
FIA_UID.2b	none	none
FMT_MSA.1	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.2
FMT_MSA.2	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.2
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1 and FMT_SMR.1
FMT_MTD.1a	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1b	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1c	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1d	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1e	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1f	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1g	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.2	FMT_MTD.1 and FMT_SMR.1	FMT_MTD.1 and FMT_SMR.1
FMT_REV.1	FMT_SMR.1	FMT_SMR.1
FMT_SAE.1	FMT_SMR.1 and FPT_STM.1	FMT_SMR.1 FPT_STM.1 is not satisfied by the TOE. Rather, the TOE is dependent upon its environment to provide time stamps in support of audit record generation.
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FPT_FLS.1a	none	none
FPT_FLS.1b	none	none
FPT_ITT.1	none	none
FPT_TST.1	none	none
FRU_FLT.1	FPT_FLS.1	FPT_FLS.1
FTA_TAB.1	none	none
ADV_ARC.1	ADV_FSP.1 and ADV_TDS.1	ADV_FSP.4 and ADV_TDS.3
ADV_FSP.4	ADV_TDS.1	ADV_TDS.3
ADV_IMP.1	ADV_TDS.3 and ALC_TAT.1	ADV_TDS.3 and ALC_TAT.1
ADV_TDS.3	ADV_FSP.4	ADV_FSP.4
AGD_OPE.1	ADV_FSP.1	ADV_FSP.4
AGD_PRE.1	none	none
ALC_CMC.4	ALC_CMS.1 and ALC_DVS.1 and ALC_LCD.1	ALC_CMS.4 and ALC_DVS.1 and ALC_LCD.1
ALC_CMS.4	none	none

ST Requirement	CC Dependencies	ST Dependencies
ALC_DEL.1	none	none
ALC_DVS.1	none	none
ALC_FLR.3	None	none
ALC_LCD.1	none	none
ALC_TAT.1	ADV_IMP.1	ADV_IMP.1
ATE_COV.2	ADV_FSP.2 and ATE_FUN.1	ADV_FSP.4 and ATE_FUN.1
ATE_DPT.2	ADV_ARC.1 and ADV_TDS.3 and ATE_FUN.1	ADV_ARC.1 and ADV_TDS.3 and ATE_FUN.1
ATE_FUN.1	ATE_COV.1	ATE_COV.2
ATE_IND.2	ADV_FSP.2 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.1 and ATE_FUN.1	ADV_FSP.4 and AGD_OPE.1 and AGD_PRE.1 and ATE_COV.2 and ATE_FUN.1
AVA_VAN.3	ADV_ARC.1 and ADV_FSP.2 and ADV_TDS.3 and ADV_IMP.1 and AGD_OPE.1 and AGD_PRE.1	ADV_ARC.1 and ADV_FSP.4 and ADV_TDS.3 and ADV_IMP.1 and AGD_OPE.1 and AGD_PRE.1

Table 8 - Requirements Dependency Analysis

The TOE provides cryptography as a service in that it allows users to arbitrarily encrypt and decrypt disk contents using operating system and TOE environment application interfaces. Cryptographic operations in support of web browsers and communication between components are not offered as a service and are not reflected in the FCS SFRs in this Security Target, rather in the FPT_ITT.1 iterations. During TOE operation, when operating system and TOE environment application interfaces are invoked to encrypt and decrypt disk content, the cryptographic operations are transparent to the user, for example no algorithm parameters are passed. Cryptographic operations are transparent given the implementation of a disk filter driver as described in section 2.1.

8.5 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 9 - Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Security audit	Cryptographic support	User data protection	Identification and authentication	Security management	Protection of the TSF	Resource Utilization	TOE Access
FAU_ARP.1ab	X							
FAU_GEN.1	X							
FAU_GEN.2	X							
FAU_SAA.1a	X							
FAU_SAA.1b	X							
FAU_SAR.1	X							
FAU_SAR.2	X							
FAU_SAR.3	X							
FAU_STG.1a	X							
FAU_STG.1b	X							
FCS_CKM.1		X						
FCS_CKM.3		X						
FCS_CKM.4		X						
FCS_COP.1a		X						
FCS_COP.1b		X						
FCS_COP.1c		X						
FDP_ACC.1			X					
FDP_ACF.1			X					
FIA_AFL.1				X				
FIA_ATD.1				X				
FIA_UAU.2a				X				
FIA_UAU.2b				X				
FIA_UAU.5				X				
FIA_UAU.6				X				
FIA_UAU.7				X				
FIA_UID.2a				X				
FIA_UID.2b				X				
FMT_MSA.1					X			
FMT_MSA.2					X			
FMT_MSA.3					X			
FMT_MTD.1a					X			
FMT_MTD.1b					X			
FMT_MTD.1c					X			
FMT_MTD.1d					X			
FMT_MTD.1e					X			
FMT_MTD.1f					X			
FMT_MTD.1g					X			
FMT_MTD.2					X			

	Security audit	Cryptographic support	User data protection	Identification and authentication	Security management	Protection of the TSF	Resource Utilization	TOE Access
FMT_REV.1					X			
FMT_SAE.1					X			
FMT_SMF.1					X			
FMT_SMR.1					X			
FPT_FLS.1a						X		
FPT_FLS.1b						X		
FPT_ITT.1						X		
FPT_TST.1						X		
FRU_FLT.1							X	
FTA_TAB.1								X

Table 9 - Security Functions vs. Requirements Mapping