Check Point Integrity Agent 6.5
Security Target

Version 1.2
6/22/2008

Prepared for:

Check Point Software Technologies

650 Towsen, Suite #575

San Francisco, CA 94103

Prepared By:

Science Applications International Corporation

Common Criteria Testing Laboratory

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

LIST OF TABLES

LIST OF FIGURES

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 Check Point Integrity Agent 6.5.063.145 provided by Check Point Software Technologies. The TOE provides security functions that include the ability to mediate network communications to and from protected workstations as well as to scan protected workstations for evidence of Spyware and take remedial actions.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security Environment (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – Check Point Integrity Agent 6.5 Security Target

ST Version – Version 1.2

ST Date – 6/22/2008

TOE Identification – Check Point Integrity Agent, version 6.5.063.145

TOE Developer – Check Point Software Technologies

Evaluation Sponsor – Check Point Software Technologies

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005

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 2.3, August 2005.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 2.3, August 2005.
  - Part 3 Conformant
  - Assurance Level: EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3
  - Strength of Function Claim: SOF-Medium

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 …").
- Explicitly stated SFRs (i.e., those not found in Part 2 of the CC) are identified with "(EXP)" following the identification of the new functional class, family, or component name (e.g., Spyware Mitigation (EXP) (FSW)).
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

2. TOE Description

The Target of Evaluation (TOE) is Check Point Integrity Agent, version 6.5.063.145.

The TOE is a personal workstation protection application. It is design to be installed on a workstation and to protect that workstation primarily by mediating network communications and by scanning the workstation for Spyware signatures. It can mediate network traffic based on network addresses, protocols, and ports. It can scan the host workstation files and registry for Spyware identifiable by a set of known signatures. Once Spyware is detected, the TOE will delete it so that any potential damage can be limited.

Note that the TOE also includes features to filter e-mail messages (MailSafe) and Instant Messages (IMSecure). However, filtering content in the context of e-mail and IM protocols is somewhat non-deterministic since the protocols are subject to change and support a wide variety of options that allow content to be hidden or disguised. As such, these features are excluded from this evaluation and the evaluated configuration of the product.

2.1 TOE Overview

The TOE consists of a driver and a service application installed on the host workstation. This combination is collectively known as an Integrity client or the Integrity Agent. While some basic management capabilities exist directly on the hosting workstation, the clients are designed to work with an Integrity server.

The Integrity server application provides a more robust and centralized administrator console interface to manage security functions provided by the agents. But, the Integrity server is excluded from the TOE since it is not required to effectively use the client products. However, in the context of this ST, the Integrity server is treated as a 'remote user' that could be configured such that it is authorized to perform the identified TOE management operations.

2.2 TOE Architecture

The Integrity agent TOE mediates network traffic between applications (running in the context of the same workstation in the IT environment) and users and other external IT entities (e.g., computers) in the IT environment accessible via attached network interfaces.

The Integrity agent TOE is able to mediate network traffic as described above based on authorized-user-configurable rules. There are multiple types of rules:

- Firewall rules control packet filtering based on source and destination addresses as well as protocol and port.
- Application rules control the ability of other applications (on the workstation) to establish network connections.

The combination of rules represents a Personal Firewall Policy in the context of this ST.

The Integrity agent TOE can also be configured to scan the hosting workstation for Spyware based on available signatures. Once Spyware is detected that is recorded in an audit log and the Spyware is deleted – these functions are collectively referred to as Spyware Mitigation in this ST.

The Integrity agent TOE can be managed directly by users on the hosting workstation or alternately by remote users using the Integrity server component (outside the TOE). Management of the Integrity agent security functions consists primarily of defining Personal Firewall Policies and configuring the Spyware Mitigation functions. Additionally, the Integrity agent audit log function and the ability to allow remote user management can be configured.

2.2.1 Physical Boundaries

The TOE is the ‘Integrity Agent’ in depicted in **Figure 1**. It consists of a driver in the hosting workstation network stack for the purpose of mediating network information flows; a service application that manages policies, performs Spyware scans, manages the audit log, and facilitates TOE management (via a network connection to an Integrity server product and interpretation of locally stored configuration information); and, a local user interface application to allow workstations users to interact with the TOE (e.g., to review configuration settings). As such, the TOE has interfaces to the external network, the internal (workstation side of the) network, and to other processes. It can also interact with a remote user (i.e., Integrity server) across a connected network.

The TOE is design to operating in the context of Windows 2000 Professional or Windows XP, utilizing execution environments provided by the hosting operating system as well as file storage and network communication services. Interaction with an Integrity Server product is optionally supported by the TOE.

Note that while the figure identifies other components (e.g., RADIUS server), the TOE is dependent only on its host operating system regardless of what that host communicates with. The other things depicted are things that offer capabilities that can be utilized by the host operating system or the optional Integrity Server Product and do not have any direct relationship with the TOE.

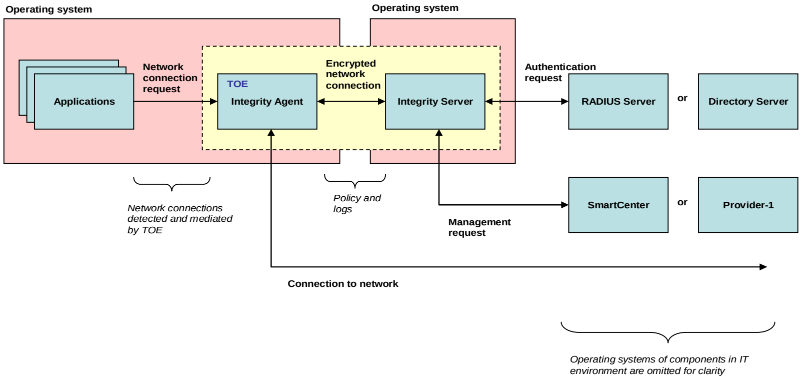

Figure 1: Product Architecture

2.2.2 Logical Boundaries

This section summarizes the security functions provided by Check Point Integrity:

- Security audit,
- User Data Protection (Personal Firewall),
- Identification and authentication,
- Security management, and
- Spyware Mitigation (EXP).

Note that this section provides only a brief overview of each of the security functions. See section 6.1 TOE Security Functions for more details.

Of the available features, MailSafe and IMSecure are not subject to evaluation claims in this Security Target and are excluded from the evaluated configuration. Note that these features are effectively disabled by default since filters would need to be explicitly configured in each case.

2.2.2.1 Security audit

The TOE generates audit records for exceptions encountered while performing Spyware Mitigation and while enforcing the Personal Firewall Policy rules. The resulting audit log is sent to an authenticated Integrity server[1]. Note that the audit log is stored within the hosting workstation, but the events are generated and forwarded to the Integrity server by the TOE.

2.2.2.2 User Data Protection (Personal Firewall)

The TOE implements rules representing a Personal Firewall Policy that can mediate: packets flowing to and from external networks and connections attempted by internal processes to interact with the attached network(s).

[1] Note that the Integrity server is not part of the TOE. The TOE offers access to the audit logs to an Integrity server once that server has been properly identified and authenticated.

2.2.2.3 Identification and authentication

The TOE requires that remote users (i.e., an Integrity server) must be properly identified and authenticated before they can perform TOE operations (e.g., to configure new rules). This is accomplished using SSL-based authentication. The Integrity client and server products support SSL for this purpose and in the evaluated configuration this feature is enabled. Note that the applicable SSL credentials must be configured so that an Integrity client can authenticate the appropriate, corresponding Integrity server. Once SSL-based authentication has occurred, the TOE uses a proprietary encryption scheme to ensure that subsequent communications are appropriately protected.

2.2.2.4 Security management

The TOE offers functions suitable to allow the TOE security functions to be configured and managed appropriately. The ability to configure the TOE in any manner is limited to authorized users. The notion of authorized users includes both local users (i.e., *any* user on the same workstation as the TOE) operating on the hosting workstation and remote users (i.e., Integrity server) that have been identified and authenticated by the TOE.

2.2.2.5 Spyware Mitigation (EXP)

The TOE has the ability to scan the hosting workstation for the presence of known Spyware signatures. Any Spyware that is identified is reported in the audit log and can also be deleted to limit potential future damage.

2.3 TOE Documentation

The TOE includes installation and user guidance (i.e., Check Point Integrity Client Management Guide) designed to facilitate effective use of the TOE in its intended environment. Additional information about these and other documents related to the TOE can be found in section 6.2 TOE Security Assurance Measures.

3. Security Environment

The TOE is intended to protect its hosting workstation (i.e., hosting IT environment[2]) primarily by mitigating threats related associated with Spyware and inappropriate network traffic. It is expected that the hosting workstation will cooperate with the TOE and will not actively seek to impair any of the TOE security functions. Indirectly, it is further expected that the users (i.e., TOE beneficiaries) of the underlying workstation will not actively seek to disable or otherwise impair or bypass the security functions of the TOE. While no assumptions are made about entities that may exist on a connected network, it is assumed that the hosting workstation and associated users will effectively be entirely cooperative with and supportive of the TOE.

3.1 Threats

T.AUDIT Security relevant events detected by the TOE may go unnoticed allowing potential security problems to persist.

T.BAD_APPS The hosting IT environment may also host applications that might attempt to make inappropriate connections or send inappropriate information to an attached network.

T.BAD_MANAGE An unauthorized user may attempt to change the behavior of the TOE security functions.

T.BAD_NETWORK The hosting IT environment may be subjected to malicious or inadvertent attacks or inappropriate traffic coming from an attached network.

T.POOR_MANAGE An authorized user may not be able to configure the TOE security functions.

T.SPYWARE Spyware running in the hosting IT environment may remain undetected and continue to operate.

T.UNSAFETOE An unauthorized user may tamper with or create a bypass around the TOE security functions.

3.2 Assumptions

A.ENVIRONMENT It is assumed that the hosting IT environment and associated users will not actively seek to disable, bypass, or otherwise impair the TOE security functions.

A.INSTALL It is assumed that the TOE will be instantiated in its hosting IT environment, according to the TOE installation guidance, such that it can correctly enforce its security policies.

A.MANAGE It is assumed that the TOE will be managed by authorized users in accordance with the TOE guidance.

[2] For the purpose of this ST, the notion of ‘hosting IT environment’ is intended to represent only the workstation upon which the TOE is installed. It is not intended to include any attach network or additional entities that may be accessible via an attached network.

4. Security Objectives

The following objectives for the TOE, its IT environment, and its non-IT environment are intended to address the threats and assumptions defined in the previous section of this ST. Note that another aspect of the objectives is that there is adequate assurance that these explicit objectives are fulfilled. These assurances are reflected in the claimed assurance target for the TOE: EAL4 augmented with ALC_FLR.2 and AVA_VLA.3.

4.1 Security Objectives for the TOE

O.AUDIT The TOE will be able to record security relevant events and allow an authorized user to review those events.

O.FIREWALL The TOE will be able to filter network traffic and connections according to predefined rules originating from the hosting IT environment and from attached networks.

O.MANAGE_SAFE
The TOE will restrict the ability to manage its security functions to an authorized user.

O.MANAGE_TOOLS
The TOE will provide the functions necessary to manage its security functions.

O.SPYWARE The TOE will be able to scan its hosting IT environment for Spyware signatures, reporting and deleting detected Spyware.

4.2 Security Objectives for the IT Environment

O.AUDITSTORE
The IT environment will facilitate the storage of audit events generated by the TOE.

O.SAFE_TOE The IT environment will instantiate the TOE in its own execution domain and protect it from tampering and bypass attempts.

4.3 Security Objectives for the non-IT Environment

O.ENVIRONMENT
The IT environment of the TOE will not include any software, firmware, or hardware that will actively attempt to alter the security functions of the TOE.

O.GUIDANCE The users (and installers) of the TOE will adhere to the available installation and user guidance.

5. IT Security Requirements

The majority of the security functional requirements and all of the security assurance requirements have been drawn from the Common Criteria (CC) Parts 2 and 3, respectively. The security functional requirements are extended with a class of requirements designed to represent a Spyware Mitigation security function offered by the TOE.

5.1 TOE Security Functional Requirements (SFRs)

The following table describes the SFRs that are satisfied by Check Point Integrity.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit data generation
	FAU_SAR.1: Audit review
FDP: User Data Protection (Personal Firewall)	FDP_IFC.1: Subset information flow control
	FDP_IFF.1: Simple security attributes
FIA: Identification and authentication	FIA_UAU.1: Timing of authentication
	FIA_UID.1: Timing of identification
FMT: Security management	FMT_MOF.1: Management of security functions behaviour
	FMT_MSA.1: Management of security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT : Protection of the TSF	FPT_ITC.1: Inter-TSF confidentiality during transmission
	FPT_ITI.1: Inter-TSF detection of modification
FSW: Spyware Mitigation (EXP)	FSW_RCT.1: Spyware Deletion (EXP)
	FSW_SCN.1: Spyware Signature-based Identification (EXP)
	FSW_SDC.1: Spyware Identification Reporting (EXP)

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 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) [**exceptions detected for the Spyware Mitigation and Personal Firewall security functions**].

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 content**].

5.1.1.2 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide [**remote users**] with the capability to read [**all recorded information**] 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.

5.1.2 User Data Protection (Personal Firewall) (FDP)

5.1.2.1 Subset information flow control (FDP_IFC.1)

FDP_IFC.1.1 The TSF shall enforce the [**Personal Firewall Policy**] on [

a) subjects: (internal subjects) processes running on the same hosting workstation in the IT environment and (external subjects) network entities in the IT environment;

b) information: network traffic (including network packets); and,

c) operations: passing of information and establishment of connections to pass information between internal and external subjects].

5.1.2.2 Simple security attributes (FDP_IFF.1)

FDP_IFF.1.1 The TSF shall enforce the [**Personal Firewall Policy**] based on the following types of subject and information security attributes: [

a) subject security attributes: application identifier for internal subjects and presumed address for external subjects;

b) information security attributes: address of destination subject, TOE interface, protocol, port].

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [

a) external subjects can cause information to flow through the TOE to internal subjects if:

1. the combination of presumed address of the external subject, TOE interface, protocol, and service are unambiguously permitted by the Personal Firewall Policy rules, where such rules may be composed from all possible combinations of the values of the identified attributes, created by the authorized user

and

2. the presumed address of the external subject translates to an external network address;

b) internal subjects can cause information to flow through the TOE to external subjects if:

1. the combination of address of destination subject, TOE interface, protocol, and service are unambiguously permitted by the Personal Firewall Policy rules, where such rules may be composed from all possible combinations of the values of the identified attributes, created by the authorized user].

FDP_IFF.1.3 The TSF shall enforce the [**no additional rules**].

FDP_IFF.1.4 The TSF shall provide the following [**no additional SFP capabilities**].

FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules: [**none**].

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules: [

a) when an internal subject attempts to open a network socket,

1. if this feature is enabled and if the application identifier associated with the internal subject is not specifically permitted, it will not be able to open a network socket].

5.1.3 Identification and authentication (FIA)

5.1.3.1 Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow [**local user operations on the TOE and information flows in accordance with the Personal Firewall Policy**] on behalf of the user to be performed before the user is authenticated.

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

5.1.3.2 Timing of identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow [**local user operations on the TOE and information flows in accordance with the Personal Firewall Policy**] on behalf of the user to be performed before the user is identified.

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

5.1.4 Security management (FMT)

5.1.4.1 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to [***modify the behaviour of***] the functions [**Spyware Mitigation and Personal Firewall**] to [**authorized users**].

5.1.4.2 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the [**Personal Firewall Policy**] to restrict the ability to [***query, modify, delete, [create]***] the security attributes [**Personal Firewall Policy rules**] to [**authorized users**].

5.1.4.3 Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the [**Personal Firewall Policy**] to provide [***permissive***] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [**authorized users**] to specify alternative initial values to override the default values when an object or information is created.

5.1.4.4 Specification of Management Functions (FMT_SMF.1)

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

- **a) Create, query, modify, and delete Personal Firewall Policy rules;**
- **b) Enable and configure the Spyware Mitigation function;**
- **c) Configure the audit trail; and,**
- **d) Configure remote management capabilities for remote users**].

5.1.4.5 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles [**authorized user, local user, remote user**].

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

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Inter-TSF confidentiality during transmission (FPT_ITC.1)

FPT_ITC.1.1 The TSF shall protect all TSF data transmitted from the TSF to a remote trusted IT product from unauthorised disclosure during transmission.

5.1.5.2 Inter-TSF detection of modification (FPT_ITI.1)

FPT_ITI.1.1 The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and a remote trusted IT product within the following metric: [**80 bits**].

FPT_ITI.1.2 The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and a remote trusted IT product and perform [**reject the TSF data**] if modifications are detected.

5.1.6 Spyware Mitigation (EXP) (FSW)

5.1.6.1 Spyware Deletion (EXP) (FSW_RCT.1)

FSW_RCT.1.1 The TSF shall be able to automatically delete identified Spyware.

5.1.6.2 Spyware Signature-based Identification (EXP) (FSW_SCN.1)

FSW_SCN.1.1 The TSF shall be able to scan its hosting IT environment for the presence of Spyware based on known Spyware signatures.

5.1.6.3 Spyware Identification Reporting (EXP) (FSW_SDC.1)

FSW_SDC.1.1 The TSF shall maintain a list of identified Spyware including an identification of the identified Spyware.

5.2 IT Environment Security Functional Requirements

The following table describes the SFRs that are to be satisfied by the hosting IT environment of Check Point Integrity.

Requirement Class	Requirement Component
FAU: Security audit	FAU_STG.1: Protected audit trail storage
FPT: Protection of the TSF	FPT_RVM.1: Non-bypassability of the TSP
	FPT_SEP.1: TSF domain separation
	FPT_STM.1: Reliable time stamps

Table 2 IT Environment Security Functional Components

5.2.1 Security audit (FAU)

5.2.1.1 Protected audit trail storage (FAU_STG.1)

FAU_STG.1.1 The ~~TSF~~**IT environment** shall protect the stored audit records from unauthorised deletion.

FAU_STG.1.2 The ~~TSF~~**IT environment** shall be able to [***prevent***] unauthorised modifications to the audit records in the audit trail.

5.2.2 Protection of the TSF (FPT)

5.2.2.1 Non-bypassability of the TSP (FPT_RVM.1)

FPT_RVM.1.1 The ~~TSF~~**IT environment** shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.2.2.2 TSF domain separation (FPT_SEP.1)

FPT_SEP.1.1 The ~~TSF~~**IT environment** shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The ~~TSF~~**IT environment** shall enforce separation between the security domains of subjects in the TSC.

5.2.2.3 Reliable time stamps (FPT_STM.1)

FPT_STM.1.1 The ~~TSF~~**IT environment** shall be able to provide reliable time stamps for its own use.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ACM: Configuration management	ACM_AUT.1: Partial CM automation
	ACM_CAP.4: Generation support and acceptance procedures
	ACM_SCP.2: Problem tracking CM coverage
ADO: Delivery and operation	ADO_DEL.2: Detection of modification
	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.2: Fully defined external interfaces
	ADV_HLD.2: Security enforcing high-level design
	ADV_IMP.1: Subset of the implementation of the TSF
	ADV_LLD.1: Descriptive low-level design
	ADV_RCR.1: Informal correspondence demonstration
	ADV_SPM.1: Informal TOE security policy model
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
	AGD_USR.1: User guidance
ALC: Life cycle support	ALC_DVS.1: Identification of security measures
	ALC_FLR.2: Flaw reporting procedures
	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.1: Testing: high-level design
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_MSU.2: Validation of analysis
	AVA_SOF.1: Strength of TOE security function evaluation
	AVA_VLA.3: Moderately resistant

Table 3 EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3Assurance Components

5.3.1 Configuration management (ACM)

5.3.1.1 Partial CM automation (ACM_AUT.1)

ACM_AUT.1.1d The developer shall use a CM system.
ACM_AUT.1.2d The developer shall provide a CM plan.
ACM_AUT.1.1c The CM system shall provide an automated means by which only authorised changes are made to the TOE implementation representation.
ACM_AUT.1.2c The CM system shall provide an automated means to support the generation of the TOE.
ACM_AUT.1.3c The CM plan shall describe the automated tools used in the CM system.
ACM_AUT.1.4c The CM plan shall describe how the automated tools are used in the CM system.
ACM_AUT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.2 Generation support and acceptance procedures (ACM_CAP.4)

ACM_CAP.4.1d The developer shall provide a reference for the TOE.
ACM_CAP.4.2d The developer shall use a CM system.
ACM_CAP.4.3d The developer shall provide CM documentation.
ACM_CAP.4.1c The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.4.2c The TOE shall be labelled with its reference.

ACM_CAP.4.3c The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.
ACM_CAP.4.4c The configuration list shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.4.5c The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.4.6c The CM documentation shall describe the method used to uniquely identify the configuration items.
ACM_CAP.4.7c The CM system shall uniquely identify all configuration items.
ACM_CAP.4.8c The CM plan shall describe how the CM system is used.
ACM_CAP.4.9c The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.
ACM_CAP.4.10c The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.
ACM_CAP.4.11c The CM system shall provide measures such that only authorised changes are made to the configuration items.
ACM_CAP.4.12c The CM system shall support the generation of the TOE.
ACM_CAP.4.13c The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.
ACM_CAP.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.3 Problem tracking CM coverage (ACM_SCP.2)

ACM_SCP.2.1d The developer shall provide a list of configuration items for the TOE.
ACM_SCP.2.1c The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.
ACM_SCP.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2 Delivery and operation (ADO)

5.3.2.1 Detection of modification (ADO_DEL.2)

ADO_DEL.2.1d The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.2.2d The developer shall use the delivery procedures.
ADO_DEL.2.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
ADO_DEL.2.2c The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.
ADO_DEL.2.3c The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.
ADO_DEL.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Installation, generation, and start-up procedures (ADO_IGS.1)

ADO_IGS.1.1d The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.
ADO_IGS.1.1c The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.
ADO_IGS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADO_IGS.1.2e The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.3.3 Development (ADV)

5.3.3.1 Fully defined external interfaces (ADV_FSP.2)

ADV_FSP.2.1d The developer shall provide a functional specification.
ADV_FSP.2.1c The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.2.2c The functional specification shall be internally consistent.
ADV_FSP.2.3c The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.
ADV_FSP.2.4c The functional specification shall completely represent the TSF.
ADV_FSP.2.5c The functional specification shall include rationale that the TSF is completely represented.
ADV_FSP.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.2 Security enforcing high-level design (ADV_HLD.2)

ADV_HLD.2.1d The developer shall provide the high-level design of the TSF.
ADV_HLD.2.1c The presentation of the high-level design shall be informal.
ADV_HLD.2.2c The high-level design shall be internally consistent.
ADV_HLD.2.3c The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.2.4c The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.2.5c The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.
ADV_HLD.2.6c The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.2.7c The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.2.8c The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_HLD.2.9c The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.
ADV_HLD.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.2.2e The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Subset of the implementation of the TSF (ADV_IMP.1)

ADV_IMP.1.1d The developer shall provide the implementation representation for a selected subset of the TSF.
ADV_IMP.1.1c The implementation representation shall unambiguously 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 internally consistent.
ADV_IMP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_IMP.1.2e The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.4 Descriptive low-level design (ADV_LLD.1)

ADV_LLD.1.1d The developer shall provide the low-level design of the TSF.
ADV_LLD.1.1c The presentation of the low-level design shall be informal.
ADV_LLD.1.2c The low-level design shall be internally consistent.
ADV_LLD.1.3c The low-level design shall describe the TSF in terms of modules.
ADV_LLD.1.4c The low-level design shall describe the purpose of each module.

ADV_LLD.1.5c The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.1.6c The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.1.7c The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.1.8c The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.1.9c The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_LLD.1.10c The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

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

ADV_LLD.1.2e The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.5 Informal correspondence demonstration (ADV_RCR.1)

ADV_RCR.1.1d The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.1.1c For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

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

5.3.3.6 Informal TOE security policy model (ADV_SPM.1)

ADV_SPM.1.1d The developer shall provide a TSP model.

ADV_SPM.1.2d The developer shall demonstrate correspondence between the functional specification and the TSP model.

ADV_SPM.1.1c The TSP model shall be informal.

ADV_SPM.1.2c The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

ADV_SPM.1.3c The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4c The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

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

5.3.4 Guidance documents (AGD)

5.3.4.1 Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1d The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1c The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2c The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3c The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4c The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.

AGD_ADM.1.5c The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6c The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7c The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8c The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

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

5.3.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1d The developer shall provide user guidance.

AGD_USR.1.1c The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2c The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3c The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4c The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

AGD_USR.1.5c The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6c The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

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

5.3.5 Life cycle support (ALC)

5.3.5.1 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.1.2c The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

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.

5.3.5.2 Flaw reporting procedures (ALC_FLR.2)

ALC_FLR.2.1d The developer shall provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3d The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.2.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5c The flaw remediation procedures documentation shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6c The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC_FLR.2.7c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

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

5.3.5.3 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.

5.3.5.4 Well-defined development tools (ALC_TAT.1)

ALC_TAT.1.1d The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2d The developer shall document the selected implementation-dependent options of the development tools.

ALC_TAT.1.1c All development tools used for implementation shall be well-defined.

ALC_TAT.1.2c The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3c The documentation of the development tools 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.

5.3.6 Tests (ATE)

5.3.6.1 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 identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2c The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

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

5.3.6.2 Testing: high-level design (ATE_DPT.1)

ATE_DPT.1.1d The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1c The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

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

5.3.6.3 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, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2c The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3c The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4c The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5c The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

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

5.3.6.4 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 test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.7 Vulnerability assessment (AVA)

5.3.7.1 Validation of analysis (AVA_MSU.2)

AVA_MSU.2.1d The developer shall provide guidance documentation.

AVA_MSU.2.2d The developer shall document an analysis of the guidance documentation.

AVA_MSU.2.1c The guidance documentation 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.

AVA_MSU.2.2c The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3c The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.2.4c The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5c The analysis documentation shall demonstrate that the guidance documentation is complete.

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

AVA_MSU.2.2e The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.2.3e The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_MSU.2.4e The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

5.3.7.2 Strength of TOE security function evaluation (AVA_SOF.1)

AVA_SOF.1.1d The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1c For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2c For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

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

AVA_SOF.1.2e The evaluator shall confirm that the strength claims are correct.

5.3.7.3 Moderately resistant (AVA_VLA.3)

AVA_VLA.3.1d The developer shall perform a vulnerability analysis.

AVA_VLA.3.2d The developer shall provide vulnerability analysis documentation.

AVA_VLA.3.1c The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.3.2c The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.3.3c The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.3.4c The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA_VLA.3.5c The vulnerability analysis documentation shall show that the search for vulnerabilities is systematic.

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

AVA_VLA.3.2e The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.3.3e The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.3.4e The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.3.5e The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a moderate attack potential.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Security audit

The TOE uses a file provided by the hosting workstation to store events that can be sent to a remote user (i.e., an Integrity server). The TOE records exceptions that occur when Spyware is identified based on known signatures as well as when the Personal Firewall Policy rules block network traffic. In each case, the audit record is created with the current date/time, the type of exception, and additional information about the exception (e.g., its source or location). The audit records are stored in log files in a proprietary format that can be interpreted by the intended users (e.g., Integrity server).

The TOE can be configured to audit only specific audit events based on their event type. However, there are some events that will always be generated and hence auditing cannot be disabled though it can be configured such that only limited audit records are generated if desired.

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

- FAU_GEN.1: The TOE generates exceptions related to Spyware Mitigation and the Personal Firewall security functions. Note that logging is always on (i.e., it cannot be disabled), so there are no audit start-up or shut-down events. Each audit record includes the date/time, the type of event, the subject identity in terms of the source of the event, and the outcome (implied in the event type).
- FAU_SAR.1: Remote users can retrieve and interpret the audit log from the TOE.

6.1.2 User Data Protection (Personal Firewall)

The TOE includes a driver that is installed in the network stack of the hosting workstation. As such, it can observe, alter, and even stop network traffic flowing between the attached networks and processes on the hosting workstation.

The TOE includes a number of mediation capabilities that together represent the Personal Firewall Policy described in this ST.

Packet Filtering

This aspect of the Personal Firewall Policy involves limiting the flow of network traffic into and out of the hosting workstation based on the relevant addresses, protocol, port, and direction of flow.

Rules can be established for essentially any combination of those attributes to prevent unwanted network packets from being received in the hosting workstation or from being sent to an attached network.

Application Restriction

This aspect of the Personal Firewall Policy involves limiting the set of applications that can open network sockets in order to interact with an attached network. Rules can be established, specifically allowing identified applications to open network sockets.

If this feature is enabled, when an application attempts to open a network socket, the TOE would only allow the operation to succeed if the TOE configuration specifically permits that application to do so. Note that the application is identified by the program that is being run and the TOE is not able to ensure that the application is correctly identified.

Note that these capabilities are not necessarily exclusive of one another. In each case, all of the applicable rules would be applied regardless of the other rules and in effect all of the applicable rules must be simultaneously satisfied in order for the information to successfully flow in or out of the hosting workstation.

Note also that, as indicated in the Security Audit section above, when exceptions result from the application of these rules, the TOE will record the applicable details in the audit log.

The User Data Protection (Personal Firewall) function is designed to satisfy the following security functional requirements:

- FDP_IFC.1: The TOE mediates the flow of network packets between other processes executing on the same hosting workstation and the attached networks.
- FDP_IFF.1: The TOE implements rules to mediate the flow of network packets through the TOE (i.e., to and from the hosting workstation), including limiting the workstation applications that can establish network connections depending on the service/protocol.

6.1.3 Identification and authentication

The TOE does not require identification or authentication for users that are local on the same workstation. Similarly, it does not require identification or authentication when processing network information flows. However, in order for a remote user (i.e., an Integrity Server) to manage the TOE, that *user* must be identified and authenticated.

The TOE identifies Integrity Servers via their network IP addresses and supports SSL (implemented using OpenSSL) for the purpose of authenticating the Integrity Server. Once the server has been authenticated, the TOE and the Integrity Server will negotiate and exchange cryptographic credentials according to a proprietary Check Point encryption scheme. These credentials are then used to ensure (by traffic encryption) the integrity and confidentiality of all subsequent traffic exchanged between the TOE and the Integrity Server. Note that integrity is ensured with an 80-bit checksum and any modified traffic is discarded. Hence, in the evaluated configuration, the TOE will accept management direction from only an Integrity Server that can be authenticated using configured SSL credentials. Whenever the negotiated cryptographic credentials become out of sync or expire, the process begins again with SSL authentication and a new credential negotiation.

Note that the Check Point implementation of the underlying cryptographic mechanisms used in this security function have not been subject to FIPS certification.

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

- FIA_UAU.1: Other than local user management and mediation of network traffic via the Personal Firewall security functions, the only interactions supported by the TOE involve remote user management via an Integrity server. The TOE requires a remote user (i.e., Integrity server) to be properly authenticated using SSL credentials prior to accepting any management instructions.
- FIA_UID.1: Other than local user management and mediation of network traffic via the Personal Firewall security functions, the only interactions supported by the TOE involve remote user management via an Integrity server. The TOE requires a remote user (i.e., Integrity Server) to be properly identified based on its network IP address.
- FPT_ITC.1: A Check Point proprietary encryption scheme is used to ensure confidentiality of communications with that remote user (i.e., Integrity Server).
- FPT_ITI.1: A Check Point proprietary encryption scheme is used to ensure integrity of communications with that remote user (i.e., Integrity Server).

6.1.4 Security management

The TOE provides functions to local users and remote users (via an authenticated Integrity Server) to manage its primary security functions (Spyware Mitigation and Personal Firewall), as well as its audit logs and support for remote users (i.e., Integrity Server configuration). In particular, the following functions are available (though not necessarily always to *both* local and remote users):

- create, query, modify, and delete Personal Firewall rules;
- enable and configure the Spyware Mitigation function;
- configure the audit log; and,
- configure the remote management capabilities for remote users.

All of the management functions are limited to authorized users (which includes local users and authenticated remote users) based on the architecture of the TOE. More specifically, it offers direct interfaces to local users and it offers an interface to remote users that requires authentication before allowing any management operations to be performed.

Note that by default, the policies enforced by the TOE are essentially 'permissive'. For example, it initially has no Personal Firewall rules and when rules are created, they affect only the entities within the scope of their definition. Hence, if no rules are defined that encompass a particular network address, then that address is not subject to any restrictions otherwise imposed by the TOE.

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

- FMT_MOF.1: All management functions are limited to local users and/or authenticated remote users (i.e., authorized users) based on either their presence on the same hosting workstation as the TOE or their having been authenticated by the TOE.
- FMT_MSA.1: All management functions (including creation, modification, review, and deletion of Personal Firewall rules) are limited to local users and/or authenticated remote users (i.e., authorized users) based on either their presence on the same hosting workstation as the TOE or their having been authenticated by the TOE.
- FMT_MSA.3: All management functions (including assigning initial Personal Firewall rules) are limited to local users and/or authenticated remote users (i.e., authorized users) based on either their presence on the same hosting workstation as the TOE or their having been authenticated by the TOE. As indicated above, the Personal Firewall rules are effectively permissive by default since the TOE initially has no rules at all and subsequently defined rules apply only to those things specifically within the scope of their definition.
- FMT_SMF.1: The required management functions are available as identified in the SFR definition and above.
- FMT_SMR.1: The notions of authorized user, local user, and remote user are largely logical in nature where the authorized users are a superset of local and remote users. Local users are assumed to be any user operating on the same hosting workstation as the TOE and remote users are specifically authenticated as such by the TOE.

6.1.5 Spyware Mitigation (EXP)

The TOE provides the ability to scan (and schedule scans) the hosting workstation for potential Spyware. The TOE can be configured to scan for a number of different types (e.g., keystroke loggers, Trojan horses, dialers), selectable by the authorized user, of Spyware based on known Spyware signatures. The scans performed by the TOE examine both the file system and registry of the hosting workstation looking for matches to the signatures it has been configured to use.

When the TOE detects Spyware, it is deleted to remove the threat of subsequent potential damage. The TOE also logs the identification of Spyware in the audit log so that the authorized user may become aware that Spyware has been identified.

The Spyware Mitigation (EXP) function is designed to satisfy the following security functional requirements:

- FSW_RCT.1: The TOE automatically deletes any Spyware that is detected during a Spyware scan.
- FSW_SCN.1: The TOE can be configured to perform Spyware scans based on a selectable set of Spyware signatures.
- FSW_SDC.1: The TOE records detected Spyware (identification) in the audit log when it is identified in a Spyware scan.

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

The configuration management measures applied by Check Point ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. Check Point ensures changes to the implementation representation are controlled with the support of automated tools and that TOE associated configuration item modifications are properly controlled. Check Point performs configuration management on the TOE implementation representation, design documentation, tests and test documentation, user and administrator guidance, delivery and operation documentation, life-cycle documentation, vulnerability analysis documentation, configuration management documentation, and security flaws.

These activities are documented in:

- Check Point Integrity 6.5 Development Plan

The Configuration management assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3 assurance requirements:

- ACM_AUT.1
- ACM_CAP.4
- ACM_SCP.2

6.2.2 Delivery and operation

Check Point provides delivery documentation and procedures to identify the TOE, allow detection of unauthorized modifications of the TOE and installation and generation instructions at start-up. Check Point's delivery procedures describe all applicable procedures to be used to detect modification to the TOE. Check Point also provides documentation that describes the steps necessary to install Integrity in accordance with the evaluated configuration.

These activities are documented in:

- Check Point Integrity 6.5 Delivery Plan
- Check Point Integrity Client Management Guide

The Delivery and operation assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3 assurance requirements:

- ADO_DEL.2
- ADO_IGS.1

6.2.3 Development

Check Point has numerous documents describing all facets of the design of the TOE. In particular, they have a functional specification that describes the accessible TOE interfaces; a high-level design that decomposes the TOE architecture into subsystems and describes each subsystem and their interfaces; a low-level design that further decomposes the TOE architecture into modules and describes each module and their interfaces; and, correspondence documentation that explains how each of the design abstractions correspond from the TOE summary specification in the Security Target to the actual implementation of the TOE. Furthermore, Check Point has a security model that describes each of the security policies implemented by Integrity. Of course, the implementation of the TOE itself is also available as necessary.

These activities are documented in:

- Check Point Integrity 6.5 Functional Specification
- Check Point Integrity 6.5 High-level Design
- Check Point Integrity 6.5 Low-level Design
- Check Point Integrity 6.5 Security Policy Model
- Check Point Integrity 6.5 source code

The Development assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3 assurance requirements:

- ADV_FSP.2
- ADV_HLD.2
- ADV_IMP.1
- ADV_LLD.1
- ADV_RCR.1
- ADV_SPM.1

6.2.4 Guidance documents

Check Point provides administrator and user guidance on how to utilize the TOE security functions and warnings to administrators and users about actions that can compromise the security of the TOE.

These activities are documented in:

- Check Point Integrity Client Management Guide

The Guidance documents assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3 assurance requirements:

- AGD_ADM.1
- AGD_USR.1

6.2.5 Life cycle support

Check Point ensures the adequacy of the procedures used during the development and maintenance of the TOE through the use of a comprehensive life-cycle management plan. Check Point applies security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure the secure development of the TOE. Check Point has procedures that define the process for accepting and acting upon user reports of security flaws. These procedures describe the acceptance criteria for security flaws, how all security flaws and the status of fixes for each security flaw are tracked, and how corrections and corrective measures are made available as applicable. Check Point has a documented model of the TOE life cycle that ensures that the TOE is developed and maintained in a well-defined manner. Check Point uses well-defined development tools in order to ensure consistent and predictable results while developing the TOE.

These activities are documented in:

- Check Point Integrity 6.5 Development Plan
- Check Point Integrity 6.5 Flaw Remediation Plan

The Life cycle support assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3 assurance requirements:

- ALC_DVS.1
- ALC_FLR.2
- ALC_LCD.1
- ALC_TAT.1

6.2.6 Tests

Check Point has a test plan that describes how each of the necessary security functions is tested, along with the expected test results. Check Point has documented each test as well as an analysis of test coverage and depth demonstrating that the security aspects of the design evident from the functional specification and high-level design are appropriately tested. Actual test results are created on a regular basis to demonstrate that the tests have been applied and that the TOE operates as designed.

These activities are documented in:

- Check Point Integrity 6.5 Test Plan
- Check Point Integrity 6.5 test results

The Tests assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3 assurance requirements:

- ATE_COV.2
- ATE_DPT.1
- ATE_FUN.1
- ATE_IND.2

6.2.7 Vulnerability assessment

The TOE administrator and user guidance documents describe the operation of Integrity and how to maintain a secure state. These guides also describe all necessary operating assumptions and security requirements outside the scope of control of the TOE. They have been developed to serve as complete, clear, consistent, and reasonable administrator and user references. Furthermore, Check Point has conducted a misuse analysis demonstrating that the provided guidance is complete.

Since the only permutational or probabilistic mechanism within Check Point Integrity is cryptographic in nature, no further SOF analysis has been conducted.

Check Point performs regular and systematic vulnerability analyses of the entire TOE (including documentation) to identify weaknesses that can be exploited in the TOE.

These activities are documented in:

- Check Point Integrity 6.5 Vulnerability Analysis

The Vulnerability assessment assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3 assurance requirements:

- AVA_MSU.2
- AVA_SOF.1
- AVA_VLA.3

7. Protection Profile Claims

This ST makes no PP conformance claims.

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;
- Strength of Functions;
- Requirement Dependencies;
- TOE Summary Specification; and,
- PP Claims.

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.

	T.AUDIT	T.BAD_APPS	T.BAD_MANAGE	T.BAD_NETWORK	T.POOR_MANAGE	T.SPYWARE	T.UNSAFETOE	A.ENVIRONMENT	A.INSTALL	A.MANAGE
O.AUDIT	X									
O.FIREWALL		X		X						
O.MANAGE_SAFE			X							
O.MANAGE_TOOLS					X					
O.SPYWARE						X				
O.AUDITSTORE	X									
O.SAFE_TOE							X			
O.ENVIRONMENT								X		
O.GUIDANCE								X	X	X

Table 4 Environment to Objective Correspondence

8.1.1.1 T.AUDIT

Security relevant events detected by the TOE may go unnoticed allowing potential security problems to persist.

This Threat is satisfied by ensuring that:

- O.AUDIT: By recording and providing interfaces to review audit records, the TOE will ensure that potential security problems are brought to the attention of authorized users.
- O.AUDITSTORE: By storing audit data generated by the TOE, the IT environment will support the TOE in maintaining an audit trail.

8.1.1.2 T.BAD_APPS

The hosting IT environment may also host applications that might attempt to make inappropriate connections or send inappropriate information to an attached network.

This Threat is satisfied by ensuring that:

- O.FIREWALL: By enforcing connection and traffic filters for traffic destined for attached networks, the TOE will mitigate potential inappropriate communications originating from the hosting IT environment.

8.1.1.3 T.BAD_MANAGE

An unauthorized user may attempt to change the behavior of the TOE security functions.

This Threat is satisfied by ensuring that:

- O.MANAGE_SAFE: By restricting the ability to manage the TOE security functions, the TOE will prevent unauthorized users from changing its own behavior.

8.1.1.4 T.BAD_NETWORK

The hosting IT environment may be subjected to malicious or inadvertent attacks or inappropriate traffic coming from an attached network.

This Threat is satisfied by ensuring that:

- O.FIREWALL: By enforcing network traffic filters for traffic from attached networks, the TOE will mitigate potential network attacks and inappropriate communications.

8.1.1.5 T.POOR_MANAGE

An authorized user may not be able to configure the TOE security functions.

This Threat is satisfied by ensuring that:

- O.MANAGE_TOOLS: By providing the functions necessary to manage each of the available TOE security functions, the TOE will ensure that it can be appropriately configured.

8.1.1.6 T.SPYWARE

Spyware running in the hosting IT environment may remain undetected and continue to operate.

This Threat is satisfied by ensuring that:

- O.SPYWARE: By scanning its host IT environment and reporting on and deleting any identified Spyware, the TOE will ensure that Spyware will not continue to operate undetected.

8.1.1.7 T.UNSAFETOE

An unauthorized user may tamper with or create a bypass around the TOE security functions.

This Threat is satisfied by ensuring that:

- O.SAFE_TOE: By protecting the TOE from tamper and bypass attempts, the IT environment will ensure that the TOE can continue to enforce its security functions appropriately.

8.1.1.8 A.ENVIRONMENT

It is assumed that the hosting IT environment and associated users will not actively seek to disable, bypass, or otherwise impair the TOE security functions.

This Assumption is satisfied by ensuring that:

- O.ENVIRONMENT: By excluding elements that may be harmful to the TOE security functions, the IT environment will ensure that the TOE security functions operate properly.
- O.GUIDANCE: By following the applicable user guidance, users will not actively seek to disable, bypass, or impair any TOE security function.

8.1.1.9 A.INSTALL

It is assumed that the TOE will be instantiated in its hosting IT environment, according to the TOE installation guidance, such that it can correctly enforce its security policies.

This Assumption is satisfied by ensuring that:

- O.GUIDANCE: By following the applicable installation guidance, users can ensure that the TOE has been properly instantiated in its intended environment.

8.1.1.10 A.MANAGE

It is assumed that the TOE will be managed by authorized users in accordance with the TOE guidance.

This Assumption is satisfied by ensuring that:

- O.GUIDANCE: By following the applicable user guidance, users can ensure that the TOE is being properly managed in its intended environment.

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that **Table 5** indicates the requirements that effectively satisfy the individual objectives. .

8.2.1 Security Functional Requirements Rationale

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

	O.AUDIT	**O.FIREWALL**	**O.MANAGE_SAFE**	**O.MANAGE_TOOLS**	**O.SPYWARE**	**O.AUDITSTORE**	**O.SAFE_TOE**
FAU_GEN.1	X						
FAU_SAR.1	X						
FDP_IFC.1		X					
FDP_IFF.1		X					
FIA_UAU.1			X				

FIA_UID.1			X				
FMT_MOF.1			X				
FMT_MSA.1			X				
FMT_MSA.3			X				
FMT_SMF.1				X			
FMT_SMR.1			X				
FPT_ITC.1			X				
FPT_ITI.1			X				
FSW_RCT.1					X		
FSW_SCN.1					X		
FSW_SDC.1					X		
FAU_STG.1						X	
FPT_RVM.1							X
FPT_SEP.1							X
FPT_STM.1	X						

Table 5 Objective to Requirement Correspondence

8.2.1.1 O.AUDIT

The TOE will be able to record security relevant events and allow an authorized user to review those events.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: The TOE must record identified detected Spyware and attempts to violate the firewall rules (i.e., exceptions in the context of those security functions)
- FAU_SAR.1: The TOE must provide access to the record audit events so they can be reviewed.
- FPT_STM.1: The IT environment must provide reliable time stamps in support of the TOE's audit generation function.

8.2.1.2 O.FIREWALL

The TOE will be able to filter network traffic and connections according to predefined rules originating from the hosting IT environment and from attached networks.

This TOE Security Objective is satisfied by ensuring that:

- FDP_IFC.1: The TOE must implement appropriate rules to control the flow of information between its hosting workstation and attached networks.
- FDP_IFF.1: The TOE must control the flow of information between its hosting workstation and attached networks to mitigate inappropriate disclosure of information or attacks against the hosting workstation.

8.2.1.3 O.MANAGE_SAFE

The TOE will restrict the ability to manage its security functions to an authorized user.

This TOE Security Objective is satisfied by ensuring that:

- FIA_UAU.1: The TOE must authenticate remote authorized users before it accepts security management instructions.
- FIA_UID.1: The TOE must identify remote authorized users before it accepts security management instructions.
- FMT_MOF.1: The TOE must limit the ability to change the behavior of its own security functions to authorized users.
- FMT_MSA.1: The TOE must limit the ability to change the firewall policy security attributes to authorized users.

- FMT_MSA.3: The TOE must have reasonable firewall policy defaults and limit the ability to change them to authorized users.
- FMT_SMR.1: The TOE must support appropriate roles for the restriction of security management functions.
- FPT_ITC.1: The TOE must ensure that remote management communications are protected from unauthorized disclosure that might lead to a security management compromise.
- FPT_ITI.1: The TOE must ensure that remote management communications are protected from modification that might lead to a security management compromise.

8.2.1.4 O.MANAGE_TOOLS

The TOE will provide the functions necessary to manage its security functions.

This TOE Security Objective is satisfied by ensuring that:

- FMT_SMF.1: The TOE must ensure that functions are offered to manage each of its security functions.

8.2.1.5 O.SPYWARE

The TOE will be able to scan its hosting IT environment for Spyware signatures, reporting and deleting detected Spyware.

This TOE Security Objective is satisfied by ensuring that:

- FSW_RCT.1: The TOE must be able to stop the execution of any detected Spyware so that it will not continue to do any potential damage.
- FSW_SCN.1: The TOE must be able to scan its hosting workstation in order to determine whether any Spyware signatures are evident.
- FSW_SDC.1: The TOE must be able to record identified Spyware so that the user can be aware that it was present and identified.

8.2.1.6 O.AUDITSTORE

The IT environment will facilitate the storage of audit events generated by the TOE.

This IT Environment Security Objective is satisfied by ensuring that:

- FAU_STG.1: The IT environment must store and protect audit records so they are not inappropriately accessed or modified.

8.2.1.7 O.SAFE_TOE

The IT environment will instantiate the TOE in its own execution domain and protect it from tampering and bypass attempts.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_RVM.1: The IT environment must ensure that the TOE (and its own mechanisms) are not bypassable.
- FPT_SEP.1: The IT environment must separate the security domain of the TOE from other domains and protect the TOE (and itself) from tampering.

8.3 Security Assurance Requirements Rationale

This security target claims an assurance rating of EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3. This target was chosen to ensure that the TOE has a moderate level of assurance in enforcing its security functions when instantiated in its intended environment which imposes no restrictions on assumed activity on applicable networks. Augmentation was chosen to provide the added assurances that result from systematic vulnerability analyses and from having flaw remediation procedures and correcting security flaws as they are reported.

8.4 Strength of Functions Rationale

In accordance with EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3, a Strength of Functions claim of SOF-medium has been made. EAL 4 augmented with ALC_FLR.2 and AVA_VLA.3 represents a moderate level of security assurance and hence SOF-medium should represent a moderate strength of function. There are no applicable mechanisms, given that cryptographic mechanisms are outside the scope of this claim, Given this and the fact that there are no security objectives that directly indicate any particular strength of function, this claim is essentially irrelevant in any case.

8.5 Requirement Dependency Rationale

As indicated in the following table, all of the dependencies defined in the CC and also for the explicitly stated requirements have been satisfied. In the following table (ST Dependencies column), TOE security functional requirements are presented normally; TOE security assurance requirements are underlined; and, IT environment security functional requirements are *italicized*.

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	*FPT_STM.1*
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FDP_IFC.1	FDP_IFF.1	FDP_IFF.1
FDP_IFF.1	FDP_IFC.1 and FMT_MSA.3	FDP_IFC.1 and FMT_MSA.3
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UID.1	none	none
FMT_MOF.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
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_IFC.1
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1 and FMT_SMR.1
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FPT_ITC.1	none	none
FPT_ITI.1	none	none
FSW_RCT.1	FSW_SCN.1	FSW_SCN.1
FSW_SCN.1	none	none
FSW_SDC.1	FSW_SCN.1	FSW_SCN.1
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FPT_RVM.1	none	none
FPT_SEP.1	none	none
FPT_STM.1	none	none
ACM_AUT.1	ACM_CAP.3	ACM_CAP.4
ACM_CAP.4	ALC_DVS.1	ALC_DVS.1
ACM_SCP.2	ACM_CAP.3	ACM_CAP.4
ADO_DEL.2	ACM_CAP.3	ACM_CAP.4
ADO_IGS.1	AGD_ADM.1	AGD_ADM.1
ADV_FSP.2	ADV_RCR.1	ADV_RCR.1
ADV_HLD.2	ADV_FSP.1 and ADV_RCR.1	ADV_FSP.2 and ADV_RCR.1
ADV_IMP.1	ADV_LLD.1 and ADV_RCR.1 and ALC_TAT.1	ADV_LLD.1 and ADV_RCR.1 and ALC_TAT.1
ADV_LLD.1	ADV_HLD.2 and ADV_RCR.1	ADV_HLD.2 and ADV_RCR.1
ADV_RCR.1	none	none
ADV_SPM.1	ADV_FSP.1	ADV_FSP.2
AGD_ADM.1	ADV_FSP.1	ADV_FSP.2
AGD_USR.1	ADV_FSP.1	ADV_FSP.2
ALC_DVS.1	none	none

ALC_FLR.2	none	none
ALC_LCD.1	none	none
ALC_TAT.1	ADV_IMP.1	ADV_IMP.1
ATE_COV.2	ADV_FSP.1 and ATE_FUN.1	ADV_FSP.2 and ATE_FUN.1
ATE_DPT.1	ADV_HLD.1 and ATE_FUN.1	ADV_HLD.2 and ATE_FUN.1
ATE_FUN.1	none	none
ATE_IND.2	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1	ADV_FSP.2 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1
AVA_MSU.2	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1	ADO_IGS.1 and ADV_FSP.2 and AGD_ADM.1 and AGD_USR.1
AVA_SOF.1	ADV_FSP.1 and ADV_HLD.1	ADV_FSP.2 and ADV_HLD.2
AVA_VLA.3	ADV_FSP.1 and ADV_HLD.2 and ADV_IMP.1 and ADV_LLD.1 and AGD_ADM.1 and AGD_USR.1	ADV_FSP.2 and ADV_HLD.2 and ADV_IMP.1 and ADV_LLD.1 and AGD_ADM.1 and AGD_USR.1

8.6 Explicitly Stated Requirements Rationale

This ST introduces a new class of security functional requirements designed to support claims about Spyware mitigation. The class, **FSW - Spyware Mitigation (EXP)**, includes three families:

- **FSW_SCN - Spyware Scanning (EXP)** has a single component (**FSW_SCN.1 - Spyware Signature-based Identification (EXP)**) to require that the TOE must be able to scan for Spyware based on known signatures;
- **FSW_SDC - Spyware Data Collection (EXP)** has a single component (**FSW_SDC.1 - Spyware Identification Reporting (EXP)**) to require that information must be recorded about detected Spyware; and,
- **FSW_RCT - Spyware Reaction (EXP)** has a single component (**FSW_RCT.1 - Spyware Deletion (EXP)**) to require that identified Spyware must be deleted so that it will not continue to operate.

The CC does not have suitable requirements to effectively represent this security function and while they may seem related to security audit (FAU), the CC security audit requirements are designed to audit the behavior of the TOE itself and not things occurring in the TOE's environment.

Of these new requirements, there are no hierarchical relationships and the only dependencies of the new security functional components are as follows:

- FSW_SCN.1 - no dependencies
- FSW_SDC.1 - depends on FSW_SCN.1
- FSW_RCT.1 - depends on FSW_SCN.1

8.7 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 6 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Security audit	User Data Protection (Personal Firewall)	Identification and authentication	Security management	Spyware Mitigation (EXP)
FAU_GEN.1	X				
FAU_SAR.1	X				
FDP_IFC.1		X			
FDP_IFF.1		X			
FIA_UAU.1			X		
FIA_UID.1			X		
FMT_MOF.1				X	
FMT_MSA.1				X	
FMT_MSA.3				X	
FMT_SMF.1				X	
FMT_SMR.1				X	
FPT_ITC.1			X		
FPT_ITI.1			X		
FSW_RCT.1					X
FSW_SCN.1					X
FSW_SDC.1					X

Table 6 Security Functions vs. Requirements Mapping

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.