# Arbor Peakflow X
# Security Target

Version 1.0
07/08/05

**Prepared for:**
Arbor Networks, Inc.

430 Bedford Street, Suite 160
Lexington, MA 02420

**Prepared By:**
Science Applications International Corporation

**Common Criteria Testing Laboratory**

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

**1. SECURITY TARGET INTRODUCTION ..... 4**

1.1 SECURITY TARGET, TOE AND CC IDENTIFICATION ..... 4
1.2 CONFORMANCE CLAIMS ..... 4
1.3 CONVENTIONS, TERMINOLOGY, ACRONYMS ..... 4
*1.3.1 Conventions ..... 5*

**2. TOE DESCRIPTION ..... 5**

2.1 TOE OVERVIEW ..... 5
2.2 TOE ARCHITECTURE ..... 6
*2.2.1 Physical Boundaries ..... 6*
*2.2.2 Logical Boundaries ..... 7*
2.3 TOE DOCUMENTATION ..... 7

**3. SECURITY ENVIRONMENT ..... 8**

3.1 THREATS ..... 8

**4. SECURITY OBJECTIVES ..... 9**

4.1 SECURITY OBJECTIVES FOR THE TOE ..... 9
4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT ..... 9

**5. IT SECURITY REQUIREMENTS ..... 10**

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS ..... 10
*5.1.1 Identification and authentication (FIA) ..... 10*
*5.1.2 Security management (FMT) ..... 11*
*5.1.3 Protection of the TSF (FPT) ..... 11*
*5.1.4 Network Integrity System Components (NIS) ..... 12*
5.2 TOE SECURITY ASSURANCE REQUIREMENTS ..... 12
*5.2.1 Configuration management (ACM) ..... 13*
*5.2.2 Delivery and operation (ADO) ..... 13*
*5.2.3 Development (ADV) ..... 14*
*5.2.4 Guidance documents (AGD) ..... 14*
*5.2.5 Tests (ATE) ..... 15*
*5.2.6 Vulnerability assessment (AVA) ..... 16*

**6. TOE SUMMARY SPECIFICATION ..... 17**

6.1 TOE SECURITY FUNCTIONS ..... 17
*6.1.1 Identification and authentication ..... 17*
*6.1.2 Security management ..... 17*
*6.1.3 Protection of the TSF ..... 17*
*6.1.4 Network Integrity System Components ..... 18*
6.2 TOE SECURITY ASSURANCE MEASURES ..... 19
*6.2.1 Configuration management ..... 19*
*6.2.2 Delivery and operation ..... 19*
*6.2.3 Development ..... 19*
*6.2.4 Guidance documents ..... 20*
*6.2.5 Tests ..... 20*
*6.2.6 Vulnerability assessment ..... 20*

**7. PROTECTION PROFILE CLAIMS ..... 21**

**8. RATIONALE ..... 22**

8.1 SECURITY OBJECTIVES RATIONALE ..... 22
*8.1.1 Security Objectives Rationale for the TOE and Environment ..... 22*
8.2 SECURITY REQUIREMENTS RATIONALE ..... 23

*8.2.1 Security Functional Requirements Rationale* ..... *23*
8.3 SECURITY ASSURANCE REQUIREMENTS RATIONALE ..... 26
8.4 STRENGTH OF FUNCTIONS RATIONALE ..... 26
8.5 REQUIREMENT DEPENDENCY RATIONALE ..... 26
8.6 EXPLICITLY STATED REQUIREMENTS RATIONALE ..... 27
8.7 TOE SUMMARY SPECIFICATION RATIONALE ..... 27
8.8 PP CLAIMS RATIONALE ..... 28

**LIST OF TABLES**

**Table 1 TOE Security Functional Components** ..... 10
**Table 2 EAL 2 Assurance Components** ..... 13
**Table 3 Environment to Objective Correspondence** ..... 22
**Table 4 Objective to Requirement Correspondence** ..... 24
**Table 5 Dependency Analysis** ..... 27
**Table 6 Security Functions vs. Requirements Mapping** ..... 28

# 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 Peakflow X, version 3.1.4**,** provided by Arbor Networks, Inc.. Peakflow X is a network integrity system (NIS) consisting of collection and controller appliances. The collectors capture network traffic information in order to build and monitor network usage policies. The controller enables management of network usage policy definitions and provides access to the results of monitoring adherence to the defined policies.

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 –** Arbor Peakflow X Security Target

**ST Version** – Version 1.0

**ST Date** – 07/08/05

**TOE Identification** – Arbor Peakflow X, version 3.1.4

**CC Identification** – Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999, ISO/IEC 15408.

## 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 2.1, August 1999, ISO/IEC 15408-2.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.1, August 1999, ISO/IEC 15408-3.
  - Part 3 Conformant
  - EAL 2

## 1.3 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the Security Target.

### 1.3.1 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 in parenthesis 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**]).
    - 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.

# 2. TOE Description

The Target of Evaluation (TOE) is Arbor Peakflow X, version 3.1.4.

Peakflow X is intended for use in large enterprise networks. Peakflow X allows organizations to identify and address internal security issues. Peakflow X constructs a holistic view of the entire network by clustering scores of hosts into groups based on their operational policy. Using this view, Peakflow X generates indications, warnings and alerts of anomalous behavior that may indicate the presence of zero-day threats, worms, misuse, or abuse.

## 2.1 TOE Overview

Peakflow X first establishes an effective-use policy for the network, as it is defined by an audit of its ongoing traffic patterns. Peakflow X then constantly reconciles that policy against actual network use. As the system detects and reports anomalies, the administrator can decide whether to act on these detected anomalous uses of the network, or to adjust the policy, further refining the derived usage policy.

Peakflow X derives its initial policy by learning network behaviors. It accomplishes this by watching the flow of information on the attached network and recording a set of behaviors. Then, once the administrator initiates grouping, Peakflow X automatically groups hosts that have similar behaviors, thus defining the policy for expected network use. The administrator can also refine the group memberships to reflect additional subtleties of network behavior.

Once grouping is complete and the administrator has determined that the learning phase has sufficiently captured the network traffic data, the administrator switches Peakflow X from learning to active mode.

At this point, Peakflow X starts observing actual network traffic and reporting any network use that represents policy violation. When Peakflow X detects a policy violation, it generates events that trigger interactive or synchronous policy refinements. For example, Peakflow X notifies the administrator when a host that belongs to a group of mail servers opens a secure shell connection to an accounting machine.

The administrator can then choose to:

- Accept the anomaly as an acceptable use, which updates the policy.
- Explicitly forbid the behavior, which updates the policy.
- Ignore the anomaly and defer modifying the policy.

Other important events, such as network scans or new hosts appearing on the network, also generate alerts, and the administrator can address these issues individually, or in batch mode.

In addition to recording and reconciling actual network usage against policy, Peakflow X maintains a database of usage over time. Peakflow X can respond to usage queries and provide details of network flows or traffic levels over a given period of time.

Peakflow X also offers a number of special-purpose reports using the network usage and policy information to improve network administration.

## 2.2 TOE Architecture

Peakflow X operates using a high-level representation of observed traffic. Peakflow X records individual connections on the attached network (flows), and groups of related connections (sessions). Peakflow X identifies individual flows using flow rules that capture the significant aspects of the connection.

Flow rules can include network host IP addresses, network service ports (for protocols such as UDP (User Datagram Protocol) or TCP (Transmission Control Protocol)), or ICMP (Internet Control Message Protocol) types and codes. Flow rules identify both ongoing and historical traffic and they can be used to describe the network policy - which operations are permitted and which operations are denied within the network. In the case of ongoing and historical traffic, flow rules are augmented with usage statistics, including total numbers of bytes flowing in and out of a connection.

Peakflow X establishes its initial network policy based on actual traffic patterns. The policy is established relative to the operator's home network, which is specified as a CIDR (Classless Inter-Domain Routing) block. The policy can be expressed in either positive or negative terms — flow connections can be either explicitly permitted or denied. After a learning period where Peakflow X has recorded typical flow transactions, it creates an initial policy based on those transactions. The initial policy is described in terms of permitted host-to-host operations.

While Peakflow X is running, it will notify the administrator of violations in the policy by sending flow rule alerts. The administrator can then act on the alert to refine the policy by explicitly accepting the new communication type, explicitly denying the communication type, or ignoring the alert and deferring the decision. The administrator can also process multiple violations synchronously as a batch.

When the administrator accepts or denies a flow rule, they either accept the specific host-host rule, or a more general host-group or group-group rule. The administrator can also refine policy by manually entering a flow rule that may actually correspond to a type of traffic the system has not yet seen.

### 2.2.1 Physical Boundaries

A deployment of Peakflow X comprises a Controller appliance and zero or more Collector appliances. The role of the Collector is to collect all network traffic, either as raw network packets (via a SPAN - Switched Port Analyzer - port) or as NetFlow[1] information. The Collector summarizes network traffic into flow information, which is then passed to the Controller.

The Controller receives the summarized flow information from one or more Collectors (or generates this information itself in an installation without any Collectors). While in learning mode, the Controller uses the flow information to build up its view of the network and its behavior. When in active mode, it compares flow information against its model of the network generates alerts if it detects anomalous behavior. All flow information is stored in a traffic flow log, available for subsequent anomaly or traffic flow analysis.

Both the Collector and Controller appliances are based on Intel commodity servers and utilize the Arbor Networks Operating System (ArbOS), which is based on OpenBSD.

Both of the appliances provide external network interfaces. One network interface is used to collect network traffic flow information via a SPAN port or Netflow from an existing network device such as a router. The second network interface is used for administration and to isolate administration traffic from the network that is being monitored.

[1] Cisco IOS® NetFlow technology is an integral part of Cisco IOS Software that collects and measures data as it enters specific routers or switch interfaces.

Either network interface can be configured to carry network traffic flow information between Collector and Controller appliances, but that communication is protected using SSL regardless.

### 2.2.2 Logical Boundaries

Peakflow X implements security functions that support Identification & Authentication, Security Management, Protection of the TSF and Network Integrity System.

#### 2.2.2.1 Identification and authentication

Both the Controller and Collector components require that administrators must be identified and authenticated before allowing them to perform any other functions. Peakflow X associates a userid and authentication data with each user.

#### 2.2.2.2 Security management

Peakflow X defines a single security management role of Administrator. The Administrator is able to manage the behavior of the network monitoring policy, by switching it between learning and monitoring modes, as well as manage user accounts to control access to the Peakflow X appliances. The Administrator is able to modify the rules that specify the network monitoring policy.

#### 2.2.2.3 Protection of the TSF

Peakflow X protects from disclosure the traffic flow data transmitted by Collectors to the Controller. It also detects modifications to traffic flow data transmitted by Collectors to the Controller and discards modified data. Peakflow X ensures that its security functions cannot be bypassed. All users (i.e., Administrators) must be identified and authenticated prior to performing any other functions. All network traffic that is collected (either by a Collector or by the Controller in a Controller-only installation) is summarized as a traffic flow and used to build the network monitoring policy (in learning mode) or is compared against the network monitoring policy for anomalous behavior (in active mode). Peakflow X is implemented on a dedicated network appliance and, as such, maintains a domain for its own execution. It protects itself against tampering by presenting limited, well-defined and -controlled external interfaces.

#### 2.2.2.4 Network Integrity System Components

Peakflow X monitors network traffic and distills captured network information (either raw packets or NetFlow data) into traffic flow data. Peakflow X uses this traffic flow data to build a policy of allowed network flows and then monitors network traffic against this policy. Peakflow X generates alerts if it identifies: a traffic flow that is inconsistent with the network monitoring policy; a traffic flow involving a previously unknown host; a traffic flow indicating an unauthorized scan; a traffic flow indicating an unusual increase in traffic volumes.

## 2.3 TOE Documentation

Arbor offers a single document (*Peakflow X Installation and User Guide version 3.1.4 (PX-UG-314)*) that describes the installation process for Peakflow X as well as guidance for subsequent security management of the product. Given that the only 'users' are administrators; this single document is designed to fulfill all of the guidance documentation requirements. Refer to section 6 for information about additional documentation.

# 3. Security Environment

The TOE security environment consists of the threats as they relate to the TOE and the IT environment protected by the TOE.

## 3.1 Threats

| | |
|---|---|
| T.ACCESS | Users may gain unauthorized access to the functions of the TOE. |
| T.POLICY | The TOE may operate with a poorly defined or incomplete policy. |
| T.PROTECT | The TOE may be vulnerable to attacks against itself or may be bypassable. |
| T.VIOLATION | Inappropriate network activity may occur and go undetected. |

# 4. Security Objectives

This section identifies the security objectives of the TOE and its supporting environment. Note that all of the IT security objectives are directed at the TOE, while all of the non-IT security objectives are directed at the TOE's intended environment. All of the identified threats are addressed by the security objectives described below.

## 4.1 Security Objectives for the TOE

| | |
|---|---|
| O.I&A | Users must be identified and authenticated to ensure that security-related functions can be appropriately restricted. |
| O.MANAGE | The TOE must protect security management functions from unauthorized use and must provide the functions necessary to effectively manage the security functions of the TOE. |
| O.MONITOR | The TOE must be able to monitor actual network traffic to identify and report violations to the administrator. |
| O.POLICY | The TOE must be able to build a default policy automatically and subsequently allow the administrator to adjust the policy. |
| O.PROTECT | The TOE must protect its functions from tampering and ensure that its security functions cannot be bypassed. |
| O.TRAIL | The TOE must store traffic flow data ensuring that it is protected from unauthorized access and also that the most current data is always retained in the event that available storage space becomes exhausted. |

## 4.2 Security Objectives for the Environment

| | |
|---|---|
| OE.CONFIGURE | The TOE appliances must be installed and configured in their target networks so that all applicable network traffic will be directed to the appliances. |
| OE.MANAGE | The TOE appliances must be installed, configured, and managed by suitable administrator personnel in accordance with the applicable guidance documentation. |
| OE.PHYSICAL | The TOE appliances and their connections must be protected from unauthorized physical access and potential tampering. |

# 5. IT Security Requirements

This section describes all of the security functional requirements for the TOE. Note that some explicit (i.e., not defined in the Common Criteria) security functional requirements pertaining to network monitoring are defined within a new class (NIS - Network Integrity System) and are identified with '(EXP)'.

The overall strength of function claim for the TOE is SOF-basic. The only security functional requirements that are associated with permutational or probabilistic mechanisms are related to user authentication (FIA_UAU.2) and encryption (FPT_ITT.1 and FPT_ITT.3).

## 5.1 TOE Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by Peakflow X.

| Requirement Class | Requirement Component |
|---|---|
| **FIA: Identification and authentication** | FIA_ATD.1: User attribute definition |
| | FIA_UAU.2: User authentication before any action |
| | FIA_UID.2: User identification before any action |
| **FMT: Security management** | FMT_MOF.1: Management of security functions behaviour |
| | FMT_MTD.1a: Management of TSF data |
| | FMT_MTD.1b: Management of TSF data |
| | FMT_SMF.1: Specification of Management Functions *(per International Interpretation #65)* |
| | FMT_SMR.1: Security roles |
| **FPT: Protection of the TSF** | FPT_ITT.1: Basic internal TSF data transfer protection |
| | FPT_ITT.3: TSF data integrity monitoring |
| | FPT_RVM.1: Non-bypassability of the TSP |
| | FPT_SEP.1: TSF domain separation |
| | FPT_STM.1: Reliable time stamps |
| **NIS: Network Integrity System Components** | NIS_NMP.1: Information flow policy (EXP) |
| | NIS_STG.1: Protection of traffic flow data (EXP) |
| | NIS_STG.2: Mitigation of traffic flow data loss (EXP) |
| | NIS_TFA.1: Traffic flow alert (EXP) |
| | NIS_TFC.1: Traffic flow collection (EXP) |
| | NIS_TFM.1: Information flow monitoring (EXP) |
| | NIS_TFP.1: Traffic flow profiling (EXP) |

**Table 1 TOE Security Functional Components**

### 5.1.1 Identification and authentication (FIA)

#### 5.1.1.1 User attribute definition (FIA_ATD.1)

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

#### 5.1.1.2 User authentication before any action (FIA_UAU.2)

**FIA_UAU.2.1** The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

#### 5.1.1.3 User identification before any action (FIA_UID.2)

**FIA_UID.2.1** The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

### 5.1.2 Security management (FMT)

#### 5.1.2.1 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 [**Network Monitoring Policy**] to [**the administrator**].

#### 5.1.2.2 Management of TSF data (FMT_MTD.1a)

**FMT_MTD.1a.1** The TSF shall restrict the ability to [***query* and *modify***] the [**security attributes belonging to individual users**] to [**the administrator**].

#### 5.1.2.3 Management of TSF data (FMT_MTD.1b)

**FMT_MTD.1b.1** The TSF shall restrict the ability to [***query* and *modify***] the [**Network Monitoring Policy definition**] to [**the administrator**].

#### 5.1.2.4 Specification of Management Functions *(per International Interpretation #65)* (FMT_SMF.1)

**FMT_SMF.1.1** The TSF shall be capable of performing the following security management functions: [**a) review traffic flow data in a manner suitable for the administrator to interpret the information; b) examine and modify the Network Monitoring Policy; and c) assign and remove user security attributes**]. *(per International Interpretation #65)*

#### 5.1.2.5 Security roles (FMT_SMR.1)

**FMT_SMR.1.1** The TSF shall maintain the roles [**administrator**].
**FMT_SMR.1.2** The TSF shall be able to associate users with roles.

### 5.1.3 Protection of the TSF (FPT)

#### 5.1.3.1 Basic internal TSF data transfer protection (FPT_ITT.1)

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

#### 5.1.3.2 TSF data integrity monitoring (FPT_ITT.3)

**FPT_ITT.3.1** The TSF shall be able to detect [***modification of data* and *substitution of data***] for TSF data transmitted between separate parts of the TOE.
**FPT_ITT.3.2** Upon detection of a data integrity error, the TSF shall take the following actions: [**disregard the data**].

#### 5.1.3.3 Non-bypassability of the TSP (FPT_RVM.1)

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

#### 5.1.3.4 TSF domain separation (FPT_SEP.1)

**FPT_SEP.1.1** The TSF 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 shall enforce separation between the security domains of subjects in the TSC.

#### 5.1.3.5 Reliable time stamps (FPT_STM.1)

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

### 5.1.4 Network Integrity System Components (NIS)

#### 5.1.4.1 Information flow policy (EXP) (NIS_NMP.1)

**NIS_NMP.1.1** The TSF shall enforce the Network Monitoring Policy on traffic flow data collected from the target network. (EXP)

#### 5.1.4.2 Protection of traffic flow data (EXP) (NIS_STG.1)

**NIS_STG.1.1** The TSF shall protect the stored traffic flow data from unauthorized deletion or modification. (EXP)

#### 5.1.4.3 Mitigation of traffic flow data loss (EXP) (NIS_STG.2)

**NIS_STG.2.1** The TSF shall overwrite the oldest stored traffic flow data if the available storage capacity has been reached. (EXP)

#### 5.1.4.4 Traffic flow alert (EXP) (NIS_TFA.1)

**NIS_TFA.1.1** The TSF shall send an alert to the administrator when any Network Monitoring Policy violation is detected. (EXP)

#### 5.1.4.5 Traffic flow collection (EXP) (NIS_TFC.1)

**NIS_TFC.1.1** The TSF shall be able to collect traffic flow data consisting of layer 2-4 network traffic headers of the OSI model from the target network. (EXP)

**NIS_TFC.1.2** At a minimum, the TSF shall collect and record the following traffic flow information: a) Date and time of the traffic; and b) the entire layer 2-4 traffic headers of the OSI model. (EXP)

#### 5.1.4.6 Information flow monitoring (EXP) (NIS_TFM.1)

**NIS_TFM.1.1** The TSF shall be able to enforce the Network Monitoring Policy based on the collected traffic flow data. (EXP)

#### 5.1.4.7 Traffic flow profiling (EXP) (NIS_TFP.1)

**NIS_TFP.1.1** The TSF shall be able to create a default Network Monitoring Policy definition based on traffic flow data collected from the target network. (EXP)

**NIS_TFP.1.2** The TSF shall begin and end the creation a default Network Monitoring Policy definition only at the request of the administrator. (EXP)

## 5.2 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 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_CAP.2: Configuration items |
| **ADO: Delivery and operation** | ADO_DEL.1: Delivery procedures |
| | ADO_IGS.1: Installation, generation, and start-up procedures |
| **ADV: Development** | ADV_FSP.1: Informal functional specification |
| | ADV_HLD.1: Descriptive high-level design |
| | ADV_RCR.1: Informal correspondence demonstration |

| | |
|---|---|
| **AGD: Guidance documents** | AGD_ADM.1: Administrator guidance |
| | AGD_USR.1: User guidance |
| **ATE: Tests** | ATE_COV.1: Evidence of coverage |
| | ATE_FUN.1: Functional testing |
| | ATE_IND.2: Independent testing - sample |
| **AVA: Vulnerability assessment** | AVA_SOF.1: Strength of TOE security function evaluation |
| | AVA_VLA.1: Developer vulnerability analysis |

**Table 2 EAL 2 Assurance Components**

## 5.2.1 Configuration management (ACM)

### 5.2.1.1 Configuration items (ACM_CAP.2)

**ACM_CAP.2.1d** The developer shall provide a reference for the TOE.
**ACM_CAP.2.2d** The developer shall use a CM system.
**ACM_CAP.2.3d** The developer shall provide CM documentation.
**ACM_CAP.2.1c** The reference for the TOE shall be unique to each version of the TOE.
**ACM_CAP.2.2c** The TOE shall be labeled with its reference.
**ACM_CAP.2.3c** The CM documentation shall include a configuration list.
**ACM_CAP.2.4c** The configuration list shall describe the configuration items that comprise the TOE.
**ACM_CAP.2.5c** The CM documentation shall describe the method used to uniquely identify the configuration items.
**ACM_CAP.2.6c** The CM system shall uniquely identify all configuration items.
**ACM_CAP.2.7c** The configuration list shall uniquely identify all configuration items that comprise the TOE. *(per International Interpretation #3)*
**ACM_CAP.2.1e** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

## 5.2.2 Delivery and operation (ADO)

### 5.2.2.1 Delivery procedures (ADO_DEL.1)

**ADO_DEL.1.1d** The developer shall document procedures for delivery of the TOE or parts of it to the user.
**ADO_DEL.1.2d** The developer shall use the delivery procedures.
**ADO_DEL.1.1c** 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.1.1e** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

### 5.2.2.2 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. *(per International Interpretation #51 (rev 1))*
**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.

## 5.2.3 Development (ADV)

### 5.2.3.1 Informal functional specification (ADV_FSP.1)

**ADV_FSP.1.1d** The developer shall provide a functional specification.
**ADV_FSP.1.1c** The functional specification shall describe the TSF and its external interfaces using an informal style.
**ADV_FSP.1.2c** The functional specification shall be internally consistent.
**ADV_FSP.1.3c** The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.
**ADV_FSP.1.4c** The functional specification shall completely represent the TSF.
**ADV_FSP.1.1e** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
**ADV_FSP.1.2e** The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

### 5.2.3.2 Descriptive high-level design (ADV_HLD.1)

**ADV_HLD.1.1d** The developer shall provide the high-level design of the TSF.
**ADV_HLD.1.1c** The presentation of the high-level design shall be informal.
**ADV_HLD.1.2c** The high-level design shall be internally consistent.
**ADV_HLD.1.3c** The high-level design shall describe the structure of the TSF in terms of subsystems.
**ADV_HLD.1.4c** The high-level design shall describe the security functionality provided by each subsystem of the TSF.
**ADV_HLD.1.5c** 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.1.6c** The high-level design shall identify all interfaces to the subsystems of the TSF.
**ADV_HLD.1.7c** The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
**ADV_HLD.1.1e** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
**ADV_HLD.1.2e** The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

### 5.2.3.3 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.

## 5.2.4 Guidance documents (AGD)

### 5.2.4.1 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.

#### 5.2.4.2 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.

### 5.2.5 Tests (ATE)

#### 5.2.5.1 Evidence of coverage (ATE_COV.1)

**ATE_COV.1.1d** The developer shall provide evidence of the test coverage.
**ATE_COV.1.1c** The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.
**ATE_COV.1.1e** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

#### 5.2.5.2 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.

#### 5.2.5.3 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.

### 5.2.6 Vulnerability assessment (AVA)

#### 5.2.6.1 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.

#### 5.2.6.2 Developer vulnerability analysis (AVA_VLA.1)

**AVA_VLA.1.1d** The developer shall perform a vulnerability analysis. *(per International Interpretation #51 (rev 1))*
**AVA_VLA.1.2d** The developer shall provide vulnerability analysis documentation. *(per International Interpretation #51 (rev 1))*
**AVA_VLA.1.1c** The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP. *(per International Interpretation #51 (rev 1))*
**AVA_VLA.1.2c** The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities. *(per International Interpretation #51 (rev 1))*
**AVA_VLA.1.3c** The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE. *(per International Interpretation #51 (rev 1))*
**AVA_VLA.1.1e** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
**AVA_VLA.1.2e** The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

# 6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

## 6.1 TOE Security Functions

### 6.1.1 Identification and authentication

The TOE appliances are based on a custom operating system, ArbOS, which requires all users to be identified an authenticated prior to accessing any other functions. Each user is defined by associating a user name with a password. Each time a user attempts to login, they must provide a valid user name and the matching password or access will be refused.

In addition to the interfaces that require identification and authentication in order to access functions provided by the TOE, the TOE also has interfaces by which it collects and shares (with other TOE appliances) network traffic information. These interfaces using Netflow or SPAN port protocols and SSL do not require identification or authentication, but they also do not offer any TOE functions. Rather, they simply passively collect and transmit information from the attached network and between TOE components.

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

- FIA_ATD.1: The TOE defines each user including a user identifier and associated password.
- FIA_UAU.2: The TOE offers no functions to users prior to authenticating their claimed identity.
- FIA_UID.2: The TOE offers no functions to users prior to their identification and subsequent authentication.

### 6.1.2 Security management

As descried above, the TOE defines user accounts based on user identities. Since the TOE offers no general user services, the only accounts that are defined are always associated with administrators of the TOE. As a result, the authorized administrator is the only role supported by the TOE and in effect the identification and authentication mechanism effectively restricts every function, including all available administrative functions, to an authorized administrator.

Among the functions offered to administer the TOE are the ability to define new user accounts (i.e., administrators), assign and change user passwords, initiate network policy collection, initiate network policy monitoring, examine and modify the network policy, and to review network monitoring results (e.g., potential violations, alerts, etc.).

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

- FMT_MOF.1: The TOE restricts the ability to change the state of policy enforcement to the administrator.
- FMT_MTD.1a: The TOE restricts the ability to manage user accounts, including their user identifiers and passwords, to administrators.
- FMT_MTD.1b: The TOE restricts the ability to review and modify the current policy to administrators.
- FMT_SMF.1: The TOE provides interfaces to effectively review traffic flow data, examine and modify the network policy, and to manage user account attributes.
- FMT_SMR.1: The TOE supports only users in the administrator role.

### 6.1.3 Protection of the TSF

Each TOE appliance is a self-contained device including hardware, a custom operating system (ArbOS), and custom applications for network traffic monitoring, analysis, and policy management. The interfaces for each appliance are

well defined and limited to specific, necessary functions. The first is the management interface is via an isolated and physically secure connection that requires identification and authentication in order to access the available functions. The second is the monitoring interface is either a SPAN port or Netflow connection to existing network devices, such as routers, through which the TOE passively collects network traffic information. This interface does not offer any TOE functions that could be used to exploit or tamper with the TOE. The third is the interface between TOE appliances. This interface is a network interface that can optionally be physically separate from the network(s) being monitored and could be physically protected. However, the TOE appliances utilize an implementation of SSL (OpenSSL) to ensure the security and integrity of data transmitted between the appliances. Furthermore, when data integrity is compromised (e.g., modification or substitution is detected) the TOE will discard the corrupted data.

Given that the TOE always processes all information that is collected via its monitoring interface, there is no way to bypass the security mechanisms of the TOE.

ArbOS provides timestamps to its applications based on time information derived from an embedded hardware clock.

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

- FPT_ITT.1: The TOE uses encryption and optionally physical isolation to protect data flowing among the TOE components from disclosure.
- FPT_ITT.3: The TOE uses encryption and optionally physical isolation to protect data flowing among the TOE components from substitution and modification, discarding any data with detected integrity errors.
- FPT_RVM.1: The TOE ensures that its functions cannot be bypassed by examining all network traffic that it receives.
- FPT_SEP.1: The TOE isolates itself in physically distinct network appliances and protects itself from tampering by offering only well-defined and controlled access points. Its only users are separated using by differentiating and control their sessions.
- FPT_STM.1: The TOE generates its own timestamps to be associated with traffic flow data.

### 6.1.4 Network Integrity System Components

The TOE provides two modes of operation that are selectable by an authorized administrator: analysis and monitoring. In the analysis mode, the TOE collects network traffic headers and constructs a default policy based on identified entities (e.g., host) on the network and observed interactions among those entities. In the monitoring mode, the TOE collects network traffic headers and identifies policy violations.

The TOE collection appliances collect network traffic headers via SPAN port or Netflow connections to existing network devices, such as routers. The traffic headers collected include OSI layers 2 through 4 and once the headers are collected, the TOE collection appliances add a timestamp and also perform some consolidation functions. The consolidation functions include merging connection related packets (e.g., both ends of a connection negotiation) to represent a single network operation and combining repeated activity in a small period of time.

When network traffic data is received by a TOE controller appliance, the TOE ensures it cannot be inappropriately access by restricting access to all TOE functions to an authorized administrator (see Identification and Authentication, above). If the available storage space for network traffic were to become exhausted, the TOE overwrites the oldest data to ensure that the newest data is always available. Note, however, the since only network headers are collected and even those undergo some consolidation, the TOE is able to store a very large amount of data before the space normally available in the TOE would become exhausted.

The TOE controller appliance records all perceived policy violations, allowing the authorized administrator to review them and decide whether to modify the policy, accept the single instance, or to otherwise respond as they see fit. In addition to recording network activity and perceived network policy violations, the TOE will issues alerts to the administrator display on the TOE controller appliance when policy violations are detected.

The Network Integrity System Components function is designed to satisfy the following security functional requirements:

- NIS_NMP.1: The TOE allows the administrator to put the TOE in monitoring mode where it monitors conformance with its defined policy.
- NIS_STG.1: The TOE protects traffic flow data from unauthorized modification or deletion.
- NIS_STG.2: The TOE retains as the newest traffic flow data by overwriting old traffic flow data when the available space is exhausted.
- NIS_TFA.1: The TOE alerts the administrator when applicable policy violations are detected.
- NIS_TFC.1: The TOE collects traffic flow information from attached routers and similar devices using Netflow or SPAN port protocols.
- NIS_TFM.1: The TOE enforces its network monitoring policy as configured by an administrator, identifying and reporting traffic flows in violation of the policy.
- NIS_TFP.1: The TOE allows the administrator to put the TOE in profiling mode where it will analyze traffic flows to create a default network policy.

## 6.2 TOE Security Assurance Measures

### 6.2.1 Configuration management

The configuration management measures applied by Arbor ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. Arbor performs configuration management on the TOE implementation representation, design, tests, user and administrator guidance, and the CM documentation.

These activities are documented in:

- Peakflow X Configuration Management Plan, revision 2.0, June 28, 2005

The Configuration management assurance measure satisfies the following EAL 2 assurance requirements:

- ACM_CAP.2

### 6.2.2 Delivery and operation

Arbor provides delivery documentation and procedures to identify the TOE and installation and generation instructions at start-up. Arbor's delivery procedures describe all applicable procedures to be used to ensure the secure delivery of the TOE. Arbor also provides documentation that describes the steps necessary to install Peakflow X in accordance with the evaluated configuration.

These activities are documented in:

- Peakflow X Delivery Procedures, revision 2.0, June 28, 2005
- Peakflow X Installation and User Guide Version 3.1.4

The Delivery and operation assurance measure satisfies the following EAL 2 assurance requirements:

- ADO_DEL.1
- ADO_IGS.1

### 6.2.3 Development

The Design Documentation provided for Peakflow X is provided in two documents. These documents serve to describe the security functions of the TOE, its interfaces both external and between subsystems, the architecture of the TOE (in terms of subsystems), and correspondence between the available design abstractions (including the ST).

These activities are documented in:

- Peakflow X Development Documentation, revision 2.0, June 28, 2005

The Development assurance measure satisfies the following EAL 2 assurance requirements:

- ADV_FSP.1
- ADV_HLD.1
- ADV_RCR.1

### 6.2.4 Guidance documents

Arbor provides administrator guidance documents that describe the administrative functions and the administrative interface available to authorized administrators of Peakflow X. These documents are consistent with other supplied documentation and describe how to administer Peakflow X in a secure manner. The guidance documents describe the assumptions regarding user behavior that is relevant to the secure operation of the appliance, and describes the parameters that are under the control of the authorized administrators.

These activities are documented in:

- Peakflow X Installation and User Guide Version 3.1.4

The Guidance documents assurance measure satisfies the following EAL 2 assurance requirements:

- AGD_ADM.1
- AGD_USR.1

### 6.2.5 Tests

The test documents describe the overall test plan, testing procedures, the tests themselves, including expected and actual results. In addition, these documents describe how the functional specification has been appropriately tested.

These activities are documented in:

- Peakflow X Testing Documentation, revision 2.0, June 28, 2005

The Tests assurance measure satisfies the following EAL 2 assurance requirements:

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

### 6.2.6 Vulnerability assessment

Arbor has conducted a strength of function analysis wherein all permutational or probabilistic security mechanisms have been identified and analyzed resulting in a demonstration that all of the relevant mechanisms fulfill the minimum strength of function claim, SOF-basic.

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

These activities are documented in:

- Arbor Peakflow X Strength of Functions Analysis, revision 2.0, June 28, 2005
- Arbor Peakflow X Vulnerability Analysis, revision 2.0, June 28, 2005

The Vulnerability assessment assurance measure satisfies the following EAL 2 assurance requirements:

- AVA_SOF.1
- AVA_VLA.1

# 7. Protection Profile Claims

There is no Protection Profile claim.

# 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.ACCESS | T.POLICY | T.PROTECT | T.VIOLATION |
|---|---|---|---|---|
| **O.I&A** | X | | | |
| **O.MANAGE** | X | X | | X |
| **O.MONITOR** | | | | X |
| **O.POLICY** | | X | | |
| **O.PROTECT** | X | | X | |
| **O.TRAIL** | X | | | X |
| **OE.CONFIGURE** | | | X | |
| **OE.MANAGE** | | | X | |
| **OE.PHYSICAL** | | | X | |

**Table 3 Environment to Objective Correspondence**

#### 8.1.1.1 T.ACCESS

*Users may gain unauthorized access to the functions of the TOE.*

This Threat is satisfied by ensuring that:

- O.I&A: Users are identified and authenticated so that security related functions can be appropriately restricted.
- O.MANAGE: The TOE protects security management functions from unauthorized use and provides the functions necessary to effectively manage the security functions of the TOE.
- O.PROTECT: The TOE protects its functions from tampering and ensures that its security functions cannot be bypassed.
- O.TRAIL: The TOE stores traffic flow data which aids in the protection against unauthorized access, and the most current data must be retained in the event that available space becomes exhausted.

#### 8.1.1.2 T.POLICY

*The TOE may operate with a poorly defined or incomplete policy.*

This Threat is satisfied by ensuring that:

- O.MANAGE: The TOE protects security management functions from unauthorized use and provides the functions necessary to effectively manage the security functions of the TOE.
- O.POLICY: The TOE is able to build a default policy automatically and subsequently allows the administrator to adjust the policy.

#### 8.1.1.3 T.PROTECT

*The TOE may be vulnerable to attacks against itself or may be bypassable.*

This Threat is satisfied by ensuring that:

- O.PROTECT: The TOE protects its functions from tampering and ensures that its security functions cannot be bypassed.
- OE.CONFIGURE: The TOE appliances are installed and configured in their target networks so that all applicable network traffic will be directed to the appliances.
- OE.MANAGE: The TOE appliances are installed, configured, and managed by suitable administrator personnel in accordance with the applicable guidance documentation.
- OE.PHYSICAL: The TOE appliances and their connections are protected from unauthorized physical access and potential tampering.

#### 8.1.1.4 T.VIOLATION

*Inappropriate network activity may occur and go undetected.*

This Threat is satisfied by ensuring that:

- O.MANAGE: The TOE protects security management functions from unauthorized use and provides the functions necessary to effectively manage the security functions of the TOE.
- O.MONITOR: The TOE is able to monitor actual network traffic in order to identify and report violations to the administrator.
- O.TRAIL: The TOE stores traffic flow data which aids in the detection of inappropriate network activity. The most current data must be retained in the event that available space becomes exhausted.

## 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 4** indicates the requirements that effectively satisfy the individual objectives. .

### 8.2.1 Security Functional 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.

| | O.I&A | O.MANAGE | O.MONITOR | O.POLICY | O.PROTECT | O.TRAIL |
|---|---|---|---|---|---|---|
| **FIA_ATD.1** | X | | | | | |
| **FIA_UAU.2** | X | | | | | |
| **FIA_UID.2** | X | | | | | |
| **FMT_MOF.1** | | X | | | | |
| **FMT_MTD.1a** | | X | | | | |
| **FMT_MTD.1b** | | X | | | | |
| **FMT_SMF.1** | | X | | X | | |
| **FMT_SMR.1** | | X | | | | |
| **FPT_ITT.1** | | | | | | X |
| **FPT_ITT.3** | | | | | X | X |
| **FPT_RVM.1** | | | | | X | |
| **FPT_SEP.1** | | | | | X | |
| **FPT_STM.1** | | | X | | | |
| **NIS_NMP.1** | | | X | | | |
| **NIS_STG.1** | | | | | | X |
| **NIS_STG.2** | | | | | | X |
| **NIS_TFA.1** | | | X | | | |
| **NIS_TFC.1** | | | X | | | |
| **NIS_TFM.1** | | | X | | | |
| **NIS_TFP.1** | | | | X | | |

**Table 4 Objective to Requirement Correspondence**

#### 8.2.1.1 O.I&A

*Users must be identified and authenticated to ensure that security-related functions can be appropriately restricted.*

This TOE Security Objective is satisfied by ensuring that:

- FIA_ATD.1: The TOE maintains user identifiers and passwords belonging to each user.
- FIA_UAU.2: The TOE requires each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UID.2: The TOE requires each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

#### 8.2.1.2 O.MANAGE

*The TOE must protect security management functions from unauthorized use and must provide the functions necessary to effectively manage the security functions of the TOE.*

This TOE Security Objective is satisfied by ensuring that:

- FMT_MOF.1: Network monitoring policies can only be modified by the administrator.
- FMT_MTD.1a: Security attributes belonging to individual users can only be viewed and modified by the administrator.
- FMT_MTD.1b: The network monitoring policy definition can only be viewed and modified by the administrator.

- FMT_SMF.1: The TOE is capable of reviewing traffic flow data in a manner suitable for the administrator to interpret the information, and of assigning and removing user security attributes.
- FMT_SMR.1: The TOE maintains the administrator role.

#### 8.2.1.3 O.MONITOR

*The TOE must be able to monitor actual network traffic to identify and report violations to the administrator.*

This TOE Security Objective is satisfied by ensuring that:

- FPT_STM.1: The TOE provides reliable time stamps for its own use.
- NIS_NMP.1: The TOE enforces the Network Monitoring Policy on traffic flow data collected from the target network.
- NIS_TFA.1: The TOE sends an alert to the administrator when any Network Monitoring Policy violation is detected.
- NIS_TFC.1: The TOE collects and records the following traffic flow data: date and time of the traffic and the entire layer 2 -4 traffic headers of the OSI model.
- NIS_TFM.1: The TOE is able to enforce the Network Monitoring Policy based on the collected traffic flow data.

#### 8.2.1.4 O.POLICY

*The TOE must be able to build a default policy automatically and subsequently allow the administrator to adjust the policy.*

This TOE Security Objective is satisfied by ensuring that:

- FMT_SMF.1: The TOE provides the administrator with the functions to review and modify the Network Monitoring Policy.
- NIS_TFP.1: The TOE is able to create a default Network Monitoring Policy definition based on traffic flow data collected from the target network.

#### 8.2.1.5 O.PROTECT

*The TOE must protect its functions from tampering and ensure that its security functions cannot be bypassed.*

This TOE Security Objective is satisfied by ensuring that:

- FPT_ITT.3: The TOE is able to detect modification and substitution of data for TSF data transmitted between separate parts of the TOE and upon such detection, the TSF will disregard the data.
- FPT_RVM.1: The TOE ensures that the TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.
- FPT_SEP.1: The TOE maintains a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

#### 8.2.1.6 O.TRAIL

*The TOE must store traffic flow data ensuring that it is protected from unauthorized access and also that the most current data is always retained in the event that available storage space becomes exhausted.*

This TOE Security Objective is satisfied by ensuring that:

- FPT_ITT.1: The TOE protects TSF data from disclosure when it is transmitted between separate parts of the TOE.
- FPT_ITT.3: The TOE is able to detect modification and substitution of data for TSF data transmitted between separate parts of the TOE and upon such detection, the TSF will disregard the data.
- NIS_STG.1: The TOE protects the stored traffic flow data from unauthorized deletion or modification.
- NIS_STG.2: The TOE will overwrite the oldest stored traffic flow data if the available storage capacity has been reached.

## 8.3 Security Assurance Requirements Rationale

This Security Target (ST) contains the assurance requirements from the Common Criteria (CC) EAL2 assurance package. This ST is based on good commercial development practices to provide a low to moderate level of assurance. While the System may monitor a hostile environment, it is expected to be in a non-hostile position protected by physical circumstances or other products designed to address threats that correspond with the intended environment. The TOE itself can be managed only via interfaces unrelated to the interfaces used to perform its primary network monitoring function offering essentially no opportunity for an attacker to subvert the security policies without physical access. As such, it is believed that EAL 2 provides an appropriate level of assurance in the security functions offered by the TOE.

## 8.4 Strength of Functions Rationale

The claimed minimum strength of function level of SOF-basic was selected since it generally corresponds with the claimed assurance level of EAL 2 and because the only applicable mechanism is the administrator authentication interface available only within the physically protected TOE environment.

## 8.5 Requirement Dependency Rationale

The following table identifies each security functional and assurance requirement in this ST. The table enumerates the dependencies of each requirement as specified in the CC and then identifies the requirement in this ST that satisfies each of those dependencies. Note that in some cases a dependency is satisfied by a hierarchically (as defined in the CC) greater requirement component (identified in **bold**), but there are no unfulfilled dependencies.

Note also that the explicitly defined requirements are included in this analysis. The dependencies for the explicit requirements are defined within this analysis.

| ST Requirement | CC Dependencies and NIS Dependencies | ST Dependencies |
|---|---|---|
| **FIA_ATD.1** | none | none |
| **FIA_UAU.2** | FIA_UID.1 | **FIA_UID.2** |
| **FIA_UID.2** | none | none |
| **FMT_MOF.1** | FMT_SMR.1 and FMT_SMF.1 | FMT_SMR.1 and FMT_SMF.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_SMF.1** | none | none |
| **FMT_SMR.1** | FIA_UID.1 | **FIA_UID.2** |
| **FPT_ITT.1** | none | none |
| **FPT_ITT.3** | FPT_ITT.1 | FPT_ITT.1 |
| **FPT_RVM.1** | none | none |
| **FPT_SEP.1** | none | none |
| **FPT_STM.1** | none | none |
| **NIS_NMP.1** | NIS_TFM.1 | NIS_TFM.1 |
| **NIS_STG.1** | NIS_TFC.1 | NIS_TFC.1 |
| **NIS_STG.2** | NIS_TFC.1 | NIS_TFC.1 |
| **NIS_TFA.1** | NIS_TFM.1 | NIS_TFM.1 |
| **NIS_TFC.1** | FPT_STM.1 | FPT_STM.1 |
| **NIS_TFM.1** | NIS_NMP.1 and NIS_TFC.1 | NIS_NMP.1 and NIS_TFC.1 |
| **NIS_TFP.1** | NIS_TFC.1 | NIS_TFC.1 |
| **ACM_CAP.2** | none | none |

| | | |
|---|---|---|
| **ADO_DEL.1** | none | none |
| **ADO_IGS.1** | AGD_ADM.1 | AGD_ADM.1 |
| **ADV_FSP.1** | ADV_RCR.1 | ADV_RCR.1 |
| **ADV_HLD.1** | ADV_FSP.1 and ADV_RCR.1 | ADV_FSP.1 and ADV_RCR.1 |
| **ADV_RCR.1** | none | none |
| **AGD_ADM.1** | ADV_FSP.1 | ADV_FSP.1 |
| **AGD_USR.1** | ADV_FSP.1 | ADV_FSP.1 |
| **ATE_COV.1** | ADV_FSP.1 and ATE_FUN.1 | ADV_FSP.1 and ATE_FUN.1 |
| **ATE_FUN.1** | ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 | ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 |
| **ATE_IND.2** | none | none |
| **AVA_SOF.1** | ADV_FSP.1 and ADV_HLD.1 | ADV_FSP.1 and ADV_HLD.1 |
| **AVA_VLA.1** | ADV_FSP.1 and ADV_HLD.1 and AGD_ADM.1 and AGD_USR.1 | ADV_FSP.1 and ADV_HLD.1 and AGD_ADM.1 and AGD_USR.1 |

**Table 5 Dependency Analysis**

## 8.6 Explicitly Stated Requirements Rationale

A class of network integrity system (NIS) requirements was created to specifically address the unique network monitoring security function provided by this TOE. There are no comparable requirements within the CC that correspond to this security function. The class, families, and components are all identified in section 5 of this ST. None of the defined components are hierarchically related and all dependencies are specified in the dependency analysis in the previous section.

## 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.

| | **Identification and authentication** | **Security management** | **Protection of the TSF** | **Network Integrity System Components** |
|---|---|---|---|---|
| **FIA_ATD.1** | X | | | |
| **FIA_UAU.2** | X | | | |
| **FIA_UID.2** | X | | | |
| **FMT_MOF.1** | | X | | |

| | | | | |
|---|---|---|---|---|
| **FMT_MTD.1a** | | X | | |
| **FMT_MTD.1b** | | X | | |
| **FMT_SMF.1** | | X | | |
| **FMT_SMR.1** | | X | | |
| **FPT_ITT.1** | | | X | |
| **FPT_ITT.3** | | | X | |
| **FPT_RVM.1** | | | X | |
| **FPT_SEP.1** | | | X | |
| **FPT_STM.1** | | | X | |
| **NIS_NMP.1** | | | | X |
| **NIS_STG.1** | | | | X |
| **NIS_STG.2** | | | | X |
| **NIS_TFA.1** | | | | X |
| **NIS_TFC.1** | | | | X |
| **NIS_TFM.1** | | | | X |
| **NIS_TFP.1** | | | | X |

**Table 6 Security Functions vs. Requirements Mapping**

## 8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.