RSA NetWitness Suite v11.0
Security Target

Version 1.0
May 31, 2018

Prepared for:

RSA Security LLC

10700 Parkridge Blvd.
Suite 600
Reston, VA 20191

Prepared By:

Leidos Inc.
Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive, Columbia, Maryland 21046

leidos

TABLE OF CONTENTS

LIST OF TABLES

1 Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE), ST conventions, ST conformance claims, and the ST organization. The TOE is RSA NetWitness Suite v11.0 (NetWitness). NetWitness is a collection of appliances that form a security infrastructure for an enterprise network. This architecture provides converged network security monitoring and centralized security information and event management (SIEM). NetWitness provides real-time visibility into the monitored network and long-term network data storage to provide detection, investigation, analysis, forensics, and compliance reporting. The NetWitness Capture infrastructure collects log and packet data from the network. Packet collection extracts metadata, reassembles, and globally normalizes all network traffic at layers 2 through 7 of the Open Systems Interconnection (OSI) model. This data allows NetWitness to perform real-time session analysis; incident detection, drill-down investigation, reporting, and forensic analysis functions.

The Security Target contains the following additional sections:

- Security Target Introduction (Section 1)
- TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Rationale (Section 7)

1.1 Security Target, TOE and CC Identification

ST Title –RSA NetWitness Suite v11.0 Security Target

ST Version – Version 1.0

ST Date– May 31, 2018

TOE Identification –RSA NetWitness Suite v11.0

TOE Developer – RSA The Security Division of EMC[2]

Evaluation Sponsor – RSA The Security Division of EMC[2]

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 3.1 Revision 4, September 2012.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
    - Part 3 Conformant
    - Assurance Level: EAL2 augmented with ALC_FLR.1

1.3 Conventions

The following conventions have been applied in this document:

Extended requirements – Security Functional Requirements not defined in Part 2 of the CC are annotated with a suffix of _EXT.

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 identified with a number in parentheses following the base component identifier. For example, iterations of FCS_COP.1 are identified in a manner similar to FCS_COP.1(1) (for the component) and FCS_COP.1.1(1) (for the elements).

Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).

Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).

Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").

Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.4 Glossary

Acronym	Description
API	Application Programming Interface
ESA	Event Stream Analysis
ATD	Automated Threat Detection
CC	Common Criteria
EAL	Evaluation Assurance Level
EPS	Events per Second
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDS	Intrusion Detection System
LAN	Local Area Network
OOTB	Out of the Box
OS	Operating System
OSI	Open Systems Interconnection
PP	Protection Profile
SDEE	Security Device Event Exchange
SIEM	Security Information and Event Management
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function

1.5 Terminology

The terminology below is described in order to clarify the terms used in the ST as well as those used in the TOE product documentation.

Analyzer	The function of an IDS that applies analytical processes to collected IDS data in order to derive conclusions about potential or actual intrusions.
CloudTrail	An application program interface (API) call-recording and log-monitoring Web service offered by Amazon Web Services (AWS).
Command and Control Malware	A system compromise where the malware is sending unauthorized data back to a system source.
Concentrator	A concentrator that receives network packet metadata.
NetWitness Core Database	An RSA proprietary repository used in the capture architecture, comprising the packet, session, and meta databases.
Decoder	A decoder that captures network packets.
Feed	A list of data that is compared to sessions as they are captured or processed. For each match, additional metadata is created. This data could identify and classify malicious IPs or incorporate additional information such as department and location based on internal network assignments.
IDS	Intrusion Detection System —a combination of services or functions such as an Analyzer that monitors an IT System for activity that may inappropriately affect the IT System or its resources, and that can send alerts if such activity is detected.
IDS data	Refers both to raw data collected by the TOE and to the results of analysis applied by the TOE to that data.
Index	Indexes are internal RA data structures that organize for searching the metadata elements of sessions and are generated during data processing for a collection. The content of the index, and consequently the metadata elements that are displayed in the Navigation view, are controlled by settings in effect during collection processing.
Log Concentrator	A concentrator that receives log metadata,
Log Decoder	A decoder that captures log data.
Metadata	Specific data types (Service Type, Action Event, Source IP Address, etc.) created by the parsers which are counted and itemized in the captured data. A detailed list of metadata for each parser may be found in the NETWITNESS Guidance.
MongoDB	A Free and open-source cross-platform document-oriented database program.
Parser	A software module that defines tokens and instructions for lexical processing of network streams. Processing includes stream identification and metadata extraction.
Services	Components of the product that work together to provide the security functions of the TOE such as Analyzer, Concentrator, and Decoder
SIEM	Security Information and Event Management—combines security information management (SIM) and security event management (SEM)

	to provide real-time analysis of security alerts generated by network hardware and applications.
WHOIS	A query and response protocol used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address block, or an autonomous system.

2 TOE Description

The Target of Evaluation (TOE) is RSA NetWitness Suite v11.0, hereinafter referred to NetWitness or the TOE.

2.1 TOE Overview

NetWitness is a collection of appliances that form a security infrastructure for an enterprise network. This architecture provides converged network security monitoring and centralized security information and event management (SIEM). NetWitness provides real-time visibility into the monitored network and long-term network data storage to provide detection, investigation, analysis, forensics, and compliance reporting. The NetWitness capture infrastructure imports log data and collects packet data from the network. Packet collection extracts metadata, reassembles, and globally normalizes all network traffic at layers 2 through 7 of the OSI model. This data allows NetWitness to perform real-time session analysis. NetWitness recognizes over 250 event source types, which are aggregated, analyzed, and stored for long-term use. The TOE implements Collection Methods to support collection from the event sources.

Data is collected and aggregated by the Decoder and Concentrator appliances. Log Collectors support data collection for use-cases such as importing Legacy Windows log data. Collected data is aggregated into a complete data structure across all network layers, logs, events, and applications. The Event Stream Analysis (ESA) appliance uses this data to provide advanced stream analytics such as correlation and complex event processing at high throughputs and low latency. ESA uses Event Processing Language to bring meaning to the event flows. The TOE's user interface uses this aggregated data to provide incident detection, and drill-down investigation[1]. The Archiver appliance is a specialized concentrator or variant that receives, indexes, and compresses logs. The Archiver is adapted to hold indexed and compressed raw log and metadata, and indices for an extended period of time. The Reporting Engine and TOE user interface use the data to provide compliance reporting and in-depth network analysis. Raw packets and packet metadata are not stored in the Archiver.

The NetWitness Suite provides functions for Data Privacy Management. The functions provide users with the Data Privacy Officer or Administrator role the ability to manage and protect privacy-sensitive data, without significantly reducing analytical capability. NetWitness Suite can be configured to limit exposure of meta data and raw content (packets and logs) using a combination of techniques. The methods available to protect data in NetWitness Suite include Data Obfuscation, Data Retention Enforcement, and Audit Logging. Data privacy officers and administrators can specify which meta keys in their environment are privacy-sensitive and limit where the meta values and raw data for those keys are displayed in the NetWitness Suite network. In place of the original values, NetWitness Suite can provide obfuscated representations to enable investigation and analytics. In addition, DPOs and administrators can prevent persistence of privacy-sensitive meta values and raw logs or packets. The Audit Logging feature generates audit log entries that are relevant to data privacy.

The TOE implements additional security functions such as identification and authentication of TOE users; auditing; security management; and trusted path.

The security management functions of the TOE are performed via the NetWitness Suite User Interface (UI), which is a web-based GUI. This interface allows authorized administrators to manage the user accounts, session lockout values and other TSF data, and view the IDS data and alerts. Navigation in the UI is based on Roles and is divided into major functional areas including Respond, Investigate, and Admin. The Respond view consolidates all alerts such as ESA Correlation Rules, ESA Automated Threat Detection, Malware Analytics, and Reporting Alerts into one location and is used for incident tracking and triage. The Investigate view presents three different views into a set of data, allowing authorized users to see metadata, events, and potential indicators of compromise. In the Admin view, Administrators can manage network hosts and services; manage system-level security; and manage Collection Methods/event sources.

NetWitness v11.0 includes the following pre-configured, out of the box (OOTB) dashboards:

- Investigation

[1] The NetWitness product provides additional capabilities for reporting, and forensic analysis functions, which are not included in the scope of evaluation.

- Operations - File Analysis
- Operations - Protocol Analysis
- Threat - Malware Indicators

The dashboards consist of dashlets that provide the ability to view the key snapshots of the various components of interest to the user in a single space. In NetWitness Suite, users can compose custom dashboards to obtain high-level information and metrics that portray the overall picture of a NetWitness Suite deployment, displaying only the information that is most relevant to the day-to-day operations.

The TOE associates users with administrative roles and maintains the pre-defined roles: Administrator, Analyst, Operator, SOC_Manager, Respond Administrator, Malware Analyst, and Data Privacy Officer. Note that pre-defined roles are not initially assigned to any user. Note also that though the administrator guidance refers to the roles as: 'Administrators', 'Analysts', 'Operators', 'SOC_Managers', and 'Malware Analysts'; the roles identified in this ST are the same roles whether or not the 's' is included at the end.

2.2 TOE Architecture

2.2.1 NetWitness Product Components

NetWitness is composed of multiple components that can be combined on appliances or deployed with multiple appliances depending on network needs. The components are broken into the Capture Architecture and the Analysis Architecture.

2.2.1.1 Capture Architecture

The NetWitness Capture architecture is composed of the Decoder, Windows Legacy Log Collector, Concentrator, and Broker. Each component is described below.

Decoder: The Decoder performs capture for either packets or logs. When deployed, either the packet or log capture capability is enabled. The term 'Decoder' is used for Decoder (packet) and 'Log Decoder' for Decoder (log). A Decoder collects packets, extracts metadata, reassembles and normalizes network traffic. A Log Decoder imports logs by either retrieving (pulling) the log records from an event source or by receiving the log records from the event sources (pushed). Each appliance sends its collected data to an assigned Concentrator.

Within a Log Decoder appliance is a Log Collector service[2] that imports logs utilizing various Collection Methods. The Collection Methods supported as part of the baseline are listed below.

- Syslog
- SNMP Trap
- NetFlow
- File (pushed by SFTP and FTPS)
- Windows (WinRM)
- ODBC
- Check Point LEA
- VMWare
- SDEE
- Cloud (Including AWS CloudTrail and Microsoft Azure)
- Office365

[2] The Log Collector service can also be deployed separately from a Log Decoder appliance as a Virtual Log Collector.

Windows Legacy Log Collector - also identified as Windows (legacy): The Windows Legacy Log Collector is deployed in a Windows Legacy domain(s). This appliance performs log capture by retrieving (pulling) the log records from a Legacy Windows event source. Each appliance sends its collected data to an assigned Log Decoder.

Concentrator: Concentrators are deployed as either a packet or log Concentrator. These appliances aggregate and store metadata received from multiple Decoders. Metadata received on a Concentrator is indexed and also may be sent to an ESA device for further analysis for detection and alerting. Concentrators also perform queries to retrieve stored metadata, as requested by external users or the NetWitness Server.

Broker: Brokers facilitate queries between Concentrators, allowing the NetWitness Server access to metadata across the network. The Broker can be installed on the same appliance as the Netwitness Server or as an independent Broker server, based on topology and traffic requirements.

2.2.1.2 Analysis Architecture

The Analysis architecture is composed of the NetWitness Server, Archiver, ESA, Malware Analysis, Automated Threat Detection, Respond, and Reporting Engine. Each component is described below. Unless otherwise stated, each component is deployed on the same appliance as the NetWitness Server.

NetWitness Server: The NetWitness Server hosts the user interface. This interface enables an administrator to perform incident detection, management, investigation, and device and user administration. The NetWitness User Interface uses the NetWitness Administration Server as the backend service for administrative tasks in the NetWitness UI. It abstracts authentication, global preferences management, and authorization support for the UI. Some of these abstractions are represented as 'Servers' in the UI. For example, the Security Server, Orchestration Server, and Config Server are all services that reside on the NetWitness Server and control different underlying capabilities for NetWitness services. The UI is accessed through HTTPS only (i.e., HTTP over TLS in FIPS mode).

Archiver: The Archiver is a stand-alone appliance. Archiver receives, indexes, and compresses log data from Log Decoders. The Archiver is adapted to hold indexed and compressed raw log and metadata and indices for an extended period of time. The Reporting Engine and UI use the data (via the Broker) to provide compliance reporting and in-depth network analysis.

ESA: ESA is installed on its own appliance (along with the ATD Service). ESA provides advanced stream analytics such as correlation and event processing. ESA receives event data from multiple Concentrators. ESA uses an advanced Event Processing Language (EPL) to filter, aggregate, join, correlate, and recognize patterns across multiple disparate event streams. ESA provides incident detection and alerting[3].

Malware Analysis Enterprise: The Malware Analysis is a stand-alone appliance. The Malware Analysis service analyzes file objects to assess the likelihood the file is malicious. This service uses network session analysis and static file analysis[4] to check for malware. The service can perform continuous or on-demand polling of Decoders or Brokers to extract sessions identified as potentially carrying malware.

Automated Threat Detection: The Automated Threat Detection (ATD) Service is deployed on the same appliance as ESA and receives metadata data from multiple Concentrators. The ATD applies rule logic across metadata to identify outliers, abnormal behavior, and malicious activity. Currently included with the service is rule logic associated with http network traffic that produces alerts and incidents for possible command and control activity. The service also performs behavior analytics automated threat detection on supported web proxy logs, such as Blue Coat Cache Flow (cacheflowelff), and Cisco IronPort WSA (ciscoiportwsa) and Zscaler (zscalernss).

Respond: Collects Alerts, displays the alerts on the NetWitness Suite Respond user interface, and provides authorized users the ability to group the alerts logically and start an Incident response workflow to investigate and

[3] NetWitness can send alerts over email, syslog or SNMP traps, but these types of alert notification are not within the scope of evaluation.
[4] The NetWitness product provides additional capabilities for dynamic file analysis, and security community analysis, which are not included in the scope of evaluation.

remediate the security issues raised. Respond allows the user to configure rules to automate the aggregation of Alerts into Incidents. The Respond service periodically runs rules to aggregate multiple Alerts into an Incident and set some attributes of the Incident (e.g. severity, category, etc.). Users can access these functions through the NetWitness UI.

Reporting Engine: The Reporting Engine supports the definition and generation of reports and alerts. Administrators can create rules that govern how data is represented in reports and alerts. The Reporting Engine also manages the alert queue, allowing administrators to enable and disable alerts.

Each appliance in the NetWitness solution can also be deployed as a virtual appliance. The functionality of the virtual appliance is the same as the hardware-based solution, though there are differences in supported throughput.

Figure 2-1 below depicts the TOE in its evaluated configuration. Note that each NetWitness Server and ESA host also contains a Mongo database though not shown in the figure. Also not depicted is the Windows Legacy Log Collector deployed in a Windows Legacy domain(s). The Windows Legacy Log Collector sends log data over the network to the Log Decoder. The Malware Analysis component aggregates data from a Packet Decoder. This communication channel is not depicted in the figure.

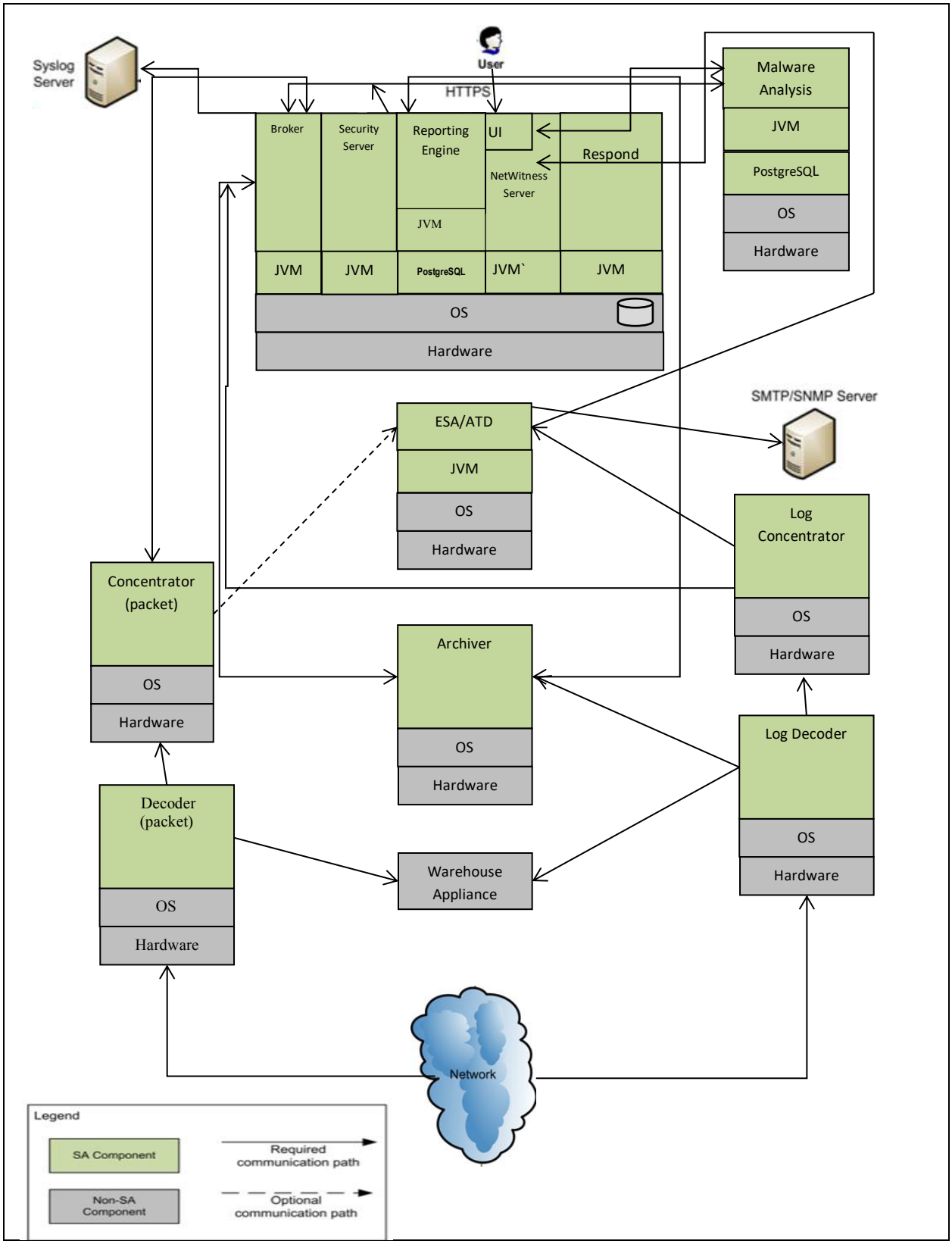
Syslog Server
User
HTTPS
Malware Analysis
JVM
PostgreSQL
OS
Hardware
Broker
Security Server
Reporting Engine
UI
NetWitness Server
Respond
JVM
JVM
JVM
PostgreSQL
JVM`
JVM
OS
Hardware
SMTP/SNMP Server
ESA/ATD
JVM
OS
Hardware
Log Concentrator
OS
Hardware
Concentrator (packet)
OS
Hardware
Archiver
OS
Hardware
Log Decoder
OS
Hardware
Decoder (packet)
OS
Hardware
Warehouse Appliance
Network
Legend
SA Component
Non-SA Component
Required communication path
Optional communication path

Figure 2-1 Evaluated Configuration

Communications between components are protected using TLS. The TOE configuration is described further in Section 2.2.2.4.

2.2.2 TOE Physical Boundaries

2.2.2.1 Included Product Components

Product components included in the TOE are listed below. **Figure 2-1 Evaluated Configuration** shows a representative deployment of the TOE in its evaluated configuration.

1. Windows Legacy Log Collector (zero or more)
   Note: A NetWitness deployment includes at least one Windows Legacy Log Collector
2. Decoder (zero or more)
3. Log Decoder (zero or more)
   Note: A NetWitness deployment includes at least one Decoder and one Log Decoder.
4. Concentrator (zero or more)
5. Log Concentrator (zero or more)
   Note: A NetWitness deployment that contains a Log Decoder must include a Log Concentrator. Likewise, a deployment that includes a Decoder for network packets must include a Concentrator for network packets.
6. Broker (zero or more)
   Note: A NetWitness deployment includes at least one Broker.
7. Event Stream Analysis (ESA) (zero or more)
   Note: A NetWitness deployment includes at least one ESA.
8. Archiver (zero or more)
   Note: A NetWitness deployment includes at least one Archiver.
9. NetWitness Server (one or more)
10. Respond (zero or more)
    Note: A NetWitness deployment includes at least one Respond.
11. Malware Analysis (zero or more)
    Note: A NetWitness deployment includes at least one Malware Analysis.
12. Automated Threat Detection (one for each ESA host)
13. Reporting Engine (one per NetWitness Server)
14. Java Virtual Machine (JVM) (one for each of the following services on the NetWitness Server: Broker, Respond, Malware Analysis, Reporting Engine Services, and one for the UI and NetWitness Server itself. Additionally, the ESA runs in its own JVM)
15. PostgreSQL database (one for each of the following services: Malware Analysis, and Reporting Engine)
16. Mongo database (one for each NetWitness Server and ESA)

2.2.2.2 Excluded Product Components

NetWitness product components excluded from the TOE in the evaluated configuration are:

1. Warehouse appliance
2. RSA Live (content delivery and Live Connect)
3. Malware Community
4. Malware Sandbox

NetWitness product features excluded from the TOE in the evaluated configuration are:

1. Direct-Attached Capacity (DAC) storage for Archiver
2. Representational State Transfer, Application Programming Interface (REST API)

3. External authentication services (such as RADIUS, LDAP, and Windows Active Directory)
4. Export of security audit records to Syslog server
5. Sending SMTP, SNMP, or Syslog alerts
6. Integrated Dell Remote Access Controller (iDRAC) out-of-band appliance management capabilities
7. Serial and USB device connections (Used during installation and maintenance only)

2.2.2.3 Services and Products in the Operational Environment

The TOE relies on the following services and products in the operational environment:

1. Operating System: provides execution environment for NetWitness components. The OS is CentOS version 7.3 running on a Dell R630 or R730xd.
2. Customer provided hardware and Windows operating system for Legacy Windows Log Collector meeting minimum system requirements below:
    a. Windows 2008 R2 SP1 64-Bit or Windows 2012 64-bit
    b. Processor – Intel Xeon CPU @2.0Ghz or faster
    c. Memory – 4GB or faster
    d. Available Disk Space - 320GB
3. Hypervisor: provides virtualization for NetWitness virtual appliances. The hypervisor is ESXi version 5.5, 6.0 and 6.5 or later.
4. Administrator Workstation / Browser: provides human users access to NetWitness Server user interface. NetWitness supports Microsoft Internet Explorer (version 11), Firefox (latest), Safari (version 11.0), and Chrome (latest).
5. Network Traffic Sources: source of network traffic. **Note:** The TOE has a direct physical connection to a network traffic source (Decoder (packet) network connection)
6. Log Decoder and Collector Collection Methods: provide log data to the TOE. Within a Log Decoder appliance is a Log Collector service[5] that imports logs utilizing various Collection Methods. The Collection Methods supported as part of the baseline are::
    a. Syslog
    b. SNMP Trap
    c. NetFlow
    d. File (pushed by SFTP and FTPS)
    e. Windows (WinRM)
    f. Windows (Legacy)
    g. ODBC
    h. Check Point LEA
    i. VMWare
    j. SDEE
    k. Cloud (Including AWS CloudTrail and Microsoft Azure)
    l. Office365

The following services can be deployed in the operational environment but were not covered by the evaluation:

1. Syslog server: NetWitness Server can forward security audit records and alerts to an external Syslog server.
2. SMTP Server: NetWitness Server can send email messages via SMTP server.
3. SNMP Server: NetWitness Server can send SNMP traps.
4. Authentication Servers: provides external authentication methods (such as Windows Active Director, RADIUS, and LDAP).

[5] The Log Collector service can also be deployed separately from a Log Decoder appliance as a Virtual Log Collector.

2.2.2.4 TOE Configurations

RSA deploys NetWitness as a collection of appliances providing services. RSA provides the TOE as either hardware appliances, virtual appliances or a combination thereof. The deployment of appliances varies from customer to customer. A customer with a small volume of network or log data would combine services onto a few appliances. A large enterprise customer would have appliances for each service with multiple Decoder, Concentrator, and Broker appliances. The evaluated configuration represents the range of deployments.

2.2.2.4.1 Hardware Appliance Deployments

The evaluated configuration includes one of each of the following appliances; deployed as either hardware or virtual hosts (see also **Figure 2-1 Evaluated Configuration**):

- NetWitness Server (hosting NetWitness Server UI, Broker, Respond, and Reporting Engine services)
    a. Broker, Respond, and Reporting Engine Services each have their own JVM. Reporting Engine also includes a PostgreSQL database.
    b. Also contains a Mongo database.
- One Malware Analysis appliance
    a. Malware Analysis includes a JVM and a PostgreSQL
- One Legacy Windows Log Collector appliance
- One Decoder appliance
    a. Decoder includes a NetWitness Core Database
- One Log Decoder appliance
    a. Log Decoder includes a NetWitness Core Database

  Note: A NetWitness deployment includes at least one Decoder or Log Decoder with the following deployment.
  a. Can be deployed standalone or on appliance with Concentrator.
  b. Typical deployment is one-to-one Decoder/ Concentrator pairs; though multiple Decoders per Concentrators or multiple Concentrators per Decoder are technically possible.
- One Concentrator appliance for packet data
    a. Concentrator includes a NetWitness Core Database
- One Concentrator appliance for log data
    a. Concentrator includes a NetWitness Core Database

  Note: A NetWitness deployment that contains a Log Decoder must include a Log Concentrator. Likewise, a deployment that includes a Decoder for network packets must include a Concentrator for network packets.
  a. Can be deployed standalone or on appliance with Decoder
  b. Typical deployment is one-to-one Decoder/ Concentrator pairs; though it is possible for a single Concentrator to aggregate from multiple Decoders or for a single Decoder to aggregate to multiple Concentrators.
- One Event Stream Analysis appliance (and JVM)
  a. Deployed standalone
  b. Receives event data from multiple Concentrators (packet and log)
  c. Includes Automated Threat Detection service and a Mongo database.

  Note: Deployments could have more than one ESA appliance.
- One Archiver appliance
    a. Archiver includes a NetWitness Core Database

  Note: Deployments could have more than one Archiver.
  a. Deployed standalone
  b. Only aggregates capture data from Log Decoder

The deployment described above includes sufficient appliances to demonstrate the TOE security functions even when additional appliances are used in a deployment. The deployment uses each of the TOE components. The interactions between TOE components remain the same when multiple components are installed on a single appliance, albeit without the need for protected communication.

Hardware Specifications per Appliance Model:

Series 4S Packet Decoder
Throughput: 2 Gbps
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 2.6GHz
RAM: 96GB
RAID Controller Card: 6Gb/s
Capacity Drive Count: 2 - 1TB HDD
Available Capacity: N/A

Series 5 Packet Decoder
Throughput: 2-10 Gbps
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 3.2GHz
RAM: 128GB
RAID Controller Card: 12Gb/s
Capacity Drive Count: 2 - 1TB HDD & 2 – 2TB HDD
Available Capacity: N/A

Series 4S Log Decoder
Throughput: up to 60K EPS
Form Factor: 1U, Full Depth
Processors:Dual Eight Core, 2.6GHz
RAM; 96GB
RAID Controller Card: 6Gb/s
Capacity Drive Count: 2 - 1TB HDD
Available Capacity: N/A

Series 5 Log Decoder
Throughput: up to 60K EPS
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 3.2GHz
RAM: 128GB
RAID Controller Card: 12Gb/s
Capacity Drive Count: 2 - 1TB HDD & 2 – 2TB HDD
Available Capacity: N/A

Series 4S Log Concentrator/ Series 4S Packet Concentrator
Throughput: N/A
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 2.6GHz
RAM: 96GB
RAID Controller Card: 6Gb/s
Capacity Drive Count: 2 - 1TB HDD
Available Capacity: N/A

Series 5 Log Concentrator/ Series 5 Packet Concentrator

Throughput: N/A
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 3.2GHz
RAM: 128GB
RAID Controller Card: 12Gb/s
Capacity Drive Count: 2 - 1TB HDD & 2 – 2TB HDD
Available Capacity: N/A

Series 4S Hybrid for Packets (Packet Decoder and Concentrator)
Throughput: 622 Mbps
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 2.6GHz
RAM: 96GB
RAID Controller Card: 6Gb/s
Capacity Drive Count: 10 - 1TB HDD
Available Capacity: 4.2 TB

Series 5 Hybrid for Packets (Packet Decoder and Concentrator)
Throughput: 1Gbps
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 3.2GHz
RAM: 128GB
RAID Controller Card: 12Gb/s
Capacity Drive Count: 4 - 1TB HDD, 8 – 6TB HDD, & 2 – 800GB SSD
Available Capacity: 48TB

Series 4S Hybrid for Logs (Log Collector, Log Decoder, and Concentrator)
Throughput: up to 20K EPS
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 2.6GHz
RAM: 96GB
RAID Controller Card: 6Gb/s
Capacity Drive Count: 10 - 1TB HDD
Available Capacity: 8 TB

Series 5 Hybrid for Logs (Log Collector, Log Decoder, and Concentrator)
Throughput: up to 20K EPS
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 3.2GHz
RAM: 128GB
RAID Controller Card: 12Gb/s
Capacity Drive Count: 4 - 1TB HDD, 8 – 6TB HDD, & 2 – 800GB SSD
Available Capacity: 48TB

Series 4S Broker
Throughput: N/A
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 2.6GHz
RAM: 96GB
RAID Controller Card: 6Gb/s
Capacity Drive Count: 2 - 1TB HDD
Available Capacity: N/A

Series 5 Broker
Throughput: N/A

Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 3.2GHz
RAM: 128GB
RAID Controller Card: 12Gb/s
Capacity Drive Count: 2 - 1TB HDD & 2 – 2TB HDD
Available Capacity: N/A

Series 4S NetWitness Server[6]
Throughput: N/A
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 2.6GHz
RAM: 96GB
RAID Controller Card: 6Gb/s
Capacity Drive Count: 2 - 1TB HDD
Available Capacity: N/A

Series 5 NetWitness Server[6]
Throughput: N/A
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 3.2GHz
RAM: 128GB
RAID Controller Card: 12Gb/s
Capacity Drive Count: 2 - 1TB HDD & 2 – 2TB HDD
Available Capacity: N/A

Series 4S Archiver
Throughput: N/A
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 2.6GHz
RAM: 96GB
RAID Controller Card: 6Gb/s
Capacity Drive Count: 2 - 1TB HDD
Available Capacity: N/A

Series 5 Archiver
Throughput: N/A
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 3.2GHz
RAM: 128GB
RAID Controller Card: 12Gb/s
Capacity Drive Count: 2 - 1TB HDD & 2 – 2TB HDD
Available Capacity: N/A

Series 4S Event Stream Analytics
Throughput: N/A
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 2.6GHz
RAM: 96GB
RAID Controller Card: 6Gb/s
Capacity Drive Count: 10 - 1TB HDD
Available Capacity:10TB

Series 5 Event Stream Analytics
Throughput: N/A

[6] Series 4S and Series 5 NetWitness Server appliances include Broker software.

Form Factor: 1U, Full Depth
Processors: Dual Twelve Core, 2.5GHz
RAM: 256GB
RAID Controller Card: 12Gb/s
Capacity Drive Count: 2 - 1TB HDD & 4 – 2TB HDD
Available Capacity: 8TB

Series 4S Malware Analysis
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 2.6GHz
RAM: 96GB
RAID Controller Card: 6Gb/s
Capacity Drive Count: 10 - 1TB HDD
Available Capacity: 8 TB

Series 5 Malware Analysis
Form Factor: 1U, Full Depth
Processors: Dual Eight Core, 3.2GHz
RAM: 128GB
RAID Controller Card: 12Gb/s
Capacity Drive Count: 2 - 1TB HDD & 2 – 2TB HDD
Available Capacity: N/A

2.2.2.4.2 Virtual and Cloud Deployments

The TOE components can also be deployed as virtual and cloud (AWS) hosts in the evaluated configuration. Both the virtual and cloud hosts are the same TOE image and operating system as the physical appliances. Virtual appliances differ from physical appliances in capacity. Hence evaluation on hardware appliances is adequate validation of virtual and cloud hosts and vice versa.

The NetWitness Server, Respond and Reporting Engine services are to be hosted as one virtual appliance in the virtual environment. Broker is not included in this virtual appliance and must be deployed as a separate virtual appliance. The hardware requirements for the virtual machine are cumulative for these components.

The following table lists CPU, Memory, and OS Disk partition minimum requirements for the virtual appliances.

- The disk requirements are fixed sizes for the OVA packages.
- RAM and CPU metrics are minimums and are also dependent on the capture and ingest environment.

Virtual Appliance Type	Quantity of CPUs	CPU Specifications	RAM	Disk
Packet Decoder	4	Intel Xeon CPU @2.59 Ghz	32 GB	320 GB
Log Decoder	4	Intel Xeon CPU @2.59 Ghz	32 GB	320 GB
Concentrator	4	Intel Xeon CPU @2.59 Ghz	32 GB	320 GB
Archiver	4	Intel Xeon CPU @2.59 Ghz	32 GB	320GB
Broker	4	Intel Xeon CPU @2.59 Ghz	32 GB	320 GB
NetWitness Server	4	Intel Xeon CPU @2.59 Ghz	32 GB	320 GB
ESA	4	Intel Xeon CPU @2.59 Ghz	32 GB	320 GB
Malware Analysis	4	Intel Xeon CPU @2.59 Ghz	32 GB	320 GB
Legacy Windows Log Collector[7]	4	Intel Xeon CPU @2.59 Ghz	32 GB	320 GB
Virtual Log Collector	2	Intel Xeon CPU @2.00 Ghz	2 GB	150 GB

[7] Basic requirements for Legacy Windows Log Collector are the same whether virtual or physical. See also Section 2.2.2.3.

The following table lists CPU, Memory, and OS Disk partition minimum requirements for the AWS deployments.

- The minimum instance type required for any NetWitness Suite component AMI is **m4-2xlarge**.
- RAM and CPU metrics are minimums and are also dependent on the capture and ingest environment.
- EBS Storage types are leveraged for different Hosts and amount of storage per type is dependent on capacity requirements in the environment.

AWS Cloud Host	Quantity of CPUs	RAM	EBS Volume Types Required
Packet Decoder	8	16 GB	General Purpose SSD and Throughput Optimized HDD
Log Decoder	8	16 GB	General Purpose SSD and Throughput Optimized HDD
Concentrator	4	16 GB	General Purpose SSD, Provisioned IOPs, Throughput Optimized HDD
Archiver	4	16 GB	General Purpose SSD and Throughput Optimized HDD
Broker	4	16 GB	General Purpose SSD
NetWitness Server	8	32 GB	General Purpose SSD
ESA	8	32 GB	General Purpose SSD
Legacy Windows Log Collector[8]	8	16 GB	General Purpose SSD
Virtual Log Collector	8	16 GB	General Purpose SSD

The TOE relies on each hosting OS to protect its applications, processes, and any locally stored data. The TOE also relies on the hosting OS for reliable time to use with the audit, IDS data.

2.2.3 TOE Logical Boundaries

This section identifies the security functions that RSA NetWitness Suite v11.0 provides. The logical boundaries of the TOE include the security functions of the TOE interfaces. The TOE logically supports the following security functions:

- Security Audit
- Cryptographic Support
- Identification & Authentication
- Security Monitoring with Security Information and Event Management (SIEM)
- Security Management
- Protection of the TSF
- TOE Access
- Trusted path/channels

2.2.3.1 Security Audit

The TOE generates audit records of security relevant events that include at least date and time of the event, subject identity and outcome for security events. The TOE provides the default Administrator and Operator roles with the ability to read the audit events. The environment stores the audit records and also provides the system clock information that is used by the TOE to timestamp each audit record.

[8] Basic requirements for Legacy Windows Log Collector are the same whether virtual or physical. See also Section 2.2.2.3.

2.2.3.2 Cryptographic Support

The Transport Layer Security (TLS 1.2) protocol in FIPS mode is used to provide protection of the communications surrounding the remote administrative sessions from disclosure and from modification. TLS is also used for distributed internal TOE component communications. The TOE uses a FIPS-validated module for SSH protected communication pathways for the transfer of file event source data from log data sources to the TOE.

The TOE uses Crypto-C ME 4.1.2 (FIPS 140-2 validation certificates #2300) for both SSH and TLS communications.

The TOE uses the RSA BSAFE Crypto-J cryptographic library: BSAFE SSL-J 6.2.1.1 for Java applications, which incorporates BSAFE Crypto-J 6.2 (FIPS 140-2 Certificates #2468).

2.2.3.3 Identification & Authentication

The TOE allows the users to acknowledge end-user license agreements and view warning banners prior to providing identification and authentication data. No other access to the TOE is permitted until the user is successfully authenticated. The TOE maintains the following security attributes belonging to individual human users: username, password and role.

The TOE provides authentication failure handling that allows administrators to configure the number of times a user may attempt to login and the time that the user will be locked out if the configured number of attempts has been surpassed. The TOE detects when the defined number of unsuccessful authentication attempts has been surpassed, and enforces the described behavior (locks the user account for a specified time period).

2.2.3.4 Security Monitoring with Security Information and Event Management (SIEM)

The TOE receives network packets, reconstructs network transactions, extracts metadata, and applies rules. The rules identify interesting events, effectively matching signatures and performing statistical analysis. Likewise, the TOE receives log data, parses the data, extracts metadata, correlates events, and applies rules. Through statistical and signature analysis, the TOE can identify potential misuse or intrusions and send an alarm to NetWitness Respond User Interfaces. The NetWitness Respond User Interfaces provide the analytical results to authorized users in a manner suitable for the user to interpret the information. The analytical results are recorded with information such as date and time. Only users with the Analysis, Administrator, and Respond Administrator roles can read the metadata, raw logs, raw packet data, and incident management (including alerts) from the IDS data.

2.2.3.5 Security Management

Authorized administrators manage the security functions and TSF data of the TOE via the web-based User Interface. The ST defines and maintains the administrative roles: Administrator, Respond Administrator, Analyst, Operator, SOC_Manager, Malware Analyst, and Data Privacy Officer. Authorized administrators perform all security functions of the TOE including starting and stopping the services and audit function, creating and managing user accounts, manage authentication failure handling and session inactivity values and read the audit and analyzer data.

2.2.3.6 Protection of the TSF

The TOE provides protection mechanisms for its security functions. One of the protection mechanisms is that users must authenticate and have the appropriate permissions before any administrative operations or access to TOE data and resources can be performed on the TSF. The TOE is a collection of special-purpose appliances. Each appliance provides only functions for the necessary operation of the TOE, and limits user access to authorized users with an administrative role.

Communication with remote administrators is protected by TLS in FIPS mode, protecting against the disclosure and undetected modification of data exchanged between the TOE and the administrator. The TOE runs in a FIPS compliant mode of operation and uses FIPS-validated cryptographic modules.

2.2.3.7 TOE Access

The TOE terminates interactive sessions after administrative configured period of time. The TOE also allows user-initiated termination of the user's own interactive session by closing the browser or explicitly logging off.

Before establishing a user session, the TOE displays an advisory warning message regarding unauthorized use of the TOE.

2.2.3.8 Trusted path/channels

The TOE requires remote users to initiate a trusted communication path using TLS for initial user authentication. The TOE also requires that the trusted path be used for the transmission of all NetWitness interface session data. The use of the trusted path provides assured identification of end points and protection of the communicated data from modification, and disclosure. The TOE uses a FIPS-validated module for SSH protected communication pathways for the transfer of file event source data from log data sources to the TOE. TLS and SSH ensure the administrative session and file transfer communication pathways are secured from disclosure and modification.

2.3 TOE Documentation

RSA has a number of administration and configuration guides for NetWitness which include the following:

- *NetWitness Suite Getting Starting Guide for Version 11.0: https://community.rsa.com/docs/DOC-79525*
- *RSA NetWitness Suite 11.0 Deployment Guide: https://community.rsa.com/docs/83063*
- *Data Privacy Management Guide for Version 11.0: https://community.rsa.com/docs/DOC-79239*
- *Automated Threat DetectionGuide for Version 11.0: https://community.rsa.com/docs/DOC-79484*
- *System Configuration Guide: https://community.rsa.com/docs/DOC-79255*
- *System Security and User Management:* https://community.rsa.com/docs/DOC-79524
- *Licensing Management Guide:* https://community.rsa.com/docs/DOC-77658
- *Core Database Tuning Guide: https://community.rsa.com/docs/DOC-60104*
- *Host and Services Getting Started Guide:* https://community.rsa.com/docs/DOC-83319
- *Archiver Configuration Guide:* https://community.rsa.com/docs/DOC-77657
- *Broker and Concentrator Configuration Guide:* https://community.rsa.com/docs/DOC-79527
- *Decoder/Log Decoder Configuration Guide: https://community.rsa.com/docs/DOC-79471*
- *ESA Configuration Guide:* https://community.rsa.com/docs/DOC-79485
- *Investigate and Malware Analysis User Guide: https://community.rsa.com/docs/DOC-79452*
- *Malware Analysis Configuration Guide:* https://community.rsa.com/docs/DOC-81596
- *Reporting Engine Configuration Guide:* https://community.rsa.com/docs/DOC-78004
- *Reporting User Guide: https://community.rsa.com/docs/DOC-79534*
- *Virtual Host Setup Guide:* https://community.rsa.com/docs/DOC-82978
- *Event Source Management User Guide: https://community.rsa.com/docs/DOC-77678*
- *Microsoft Office 365: Event Source Log Configuration Guide: https://community.rsa.com/docs/DOC-79883*
- *Log Collection Configuration Guide:* https://community.rsa.com/docs/DOC-77591
- *RSA NetWitness System Maintenance Guide:* https://community.rsa.com/docs/DOC-79254
- *NetWitness Respond Configuration Guide: https://community.rsa.com/docs/DOC-79489*
- *NetWitness Respond User Guide: https://community.rsa.com/docs/DOC-79457*
- *Alerting Using ESA Guide: https://community.rsa.com/docs/DOC-79234*
- *AWS Virtual Public Cloud Deployment Guide: https://community.rsa.com/docs/DOC-84098*
- *RSA Product Verification Checklist (Secure Acceptance Procedures):* RSA portal/customer section

3 Security Problem Definition

The TOE security environment describes the security aspects of the intended environment in which the TOE is to be used and the manner in which it is expected to be employed. The statement of the TOE security environment defines the following:

- Threats that the TOE and the environment of the TOE counter
- Assumptions made about the operational environment and the intended method of use for the TOE

The statement of TOE security environment does not include any Organizational Policies.

The TOE is intended to be used in environments where the relative assurance that its security functions are enforced is commensurate with EAL 2 augmented with ALC_FLR.1 as defined in the CC.

3.1 Assumptions

The following conditions are assumed to exist in the operational environment.

3.1.1 Intended Usage Assumptions

A.AUDIT_PROTECTION	The operational environment will provide the capability to protect audit information.
A.DATA_SOURCES	The data sources in the environment provide complete and reliable data to the TOE.
A.TIME	The environment will provide reliable time sources for use by the TOE.

3.1.2 Physical Assumptions

A.DEPLOY	TOE Administrators will properly configure the network in the TOE operational environment and configure adequate network capacity for the deployed TOE components.
A.PHYSICAL	The TOE hardware and software critical to the security policy enforcement will be located within controlled access facilities which will prevent unauthorized physical access.

3.1.3 Personnel Assumptions

A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.TRUSTED_ADMIN	TOE Administrators will follow and apply all administrator guidance in a trusted manner.
A.USER	Users will protect their authentication data.

3.2 Threats

T.UNPROTECTED_PRIVACY_DATA	IDS Data that should be protected as privacy sensitive is not obscured, access restricted or retained appropriately allowing unauthorized users to view the data.
T.MALICIOUS_ACTIVITY	Malicious activity by an attacker may occur on the network the TOE monitors may go undetected.
T.INADVERTENT_ACTIVITY	Inadvertent activity and access by a user or a process that may occur on the network the TOE monitors may go undetected.
T.MISUSE	Unauthorized accesses and activity indicative of misuse by a user or a process that may occur on the network the TOE monitors may go undetected.

T.TSF_COMPROMISE	A user may cause, through an unsophisticated attack, TSF data, or security functions to be inappropriately accessed (viewed, modified, or deleted).
T.UNACCOUNTABLE_USERS	Authorized users of the TOE might not be held accountable for their actions.

4 Security Objectives

This chapter identifies the security objectives of the TOE and its environment. Security objectives identify the responsibilities of the TOE and the support needed by the TOE from its environment.

4.1 Security Objectives for the TOE

O.ANALYZE	The TOE will apply analytical processes and information to derive conclusions about potential unauthorized/malicious intrusions and send appropriate alerts.
O.PRIVACY_DATA_PROTECT	The TOE will protect data determined to be privacy sensitive.
O.AUDIT_GENERATION:	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.AUDIT_PROTECTION:	The TOE will provide the capability for protection of the audit information from unauthorized users via the TOE interfaces.
O.MANAGE:	The TOE will provide all the functions necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions from unauthorized use.
O.TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE.
O.PROTECTED_COMMS	The TOE will provide protected communication channels for remote administrators, IT entities and for TOE device to TOE device communications.

4.2 Security Objectives for the Operational Environment

The TOE's operating environment must satisfy the following objectives.

OE.AUDIT_PROTECTION	The operational environment provides the capability to protect audit information.
OE.DATA_SOURCES	The data sources in the environment provide complete and reliable data to the TOE.
OE.DEPLOY	The TOE Administrators will properly configure the network in the TOE operational environment and configure adequate network capacity for the deployed TOE components
OE.MANAGE	Personnel working as authorized administrators shall be carefully selected and trained for proper operation of the TSF.
OE.PHYSICAL	The TOE hardware and software critical to the security policy enforcement will be located within controlled access facilities which will prevent unauthorized physical access.
OE.TIME	The environment provides reliable time sources for use by the TOE.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.USER	Users must ensure that their authentication data is held securely and not disclosed to unauthorized persons.

5 IT Security Requirements

The security requirements for the TOE have been drawn from Parts 2 and 3 of the Common Criteria. The security functional requirements have been selected to correspond to the actual security functions implemented by the TOE while the assurance requirements have been selected to offer a low to moderate degree of assurance that those security functions are properly realized.

5.1 Extended Component Definition

This Security Target includes Security Functional Requirements (SFRs) that are not drawn from CC Part 2. These Extended SFRs are identified by having a label '_EXT' in the requirement name for TOE SFRs. The structure of the extended SFRs is modeled after the SFRs included in CC Part 2. The structure is as follows:

A. Class – The extended SFRs included in this ST are part of the identified classes of requirements.

B. Family – The extended SFRs included in this ST are part of several SFR families including the new families defined below.

C. Component – The extended SFRs are not hierarchical to any other components, though they may have identifiers terminating on other than "1". The dependencies for each extended component are identified in the TOE SFR Dependencies section of this ST (Section 7.3, Requirement Dependency Rationale).

5.1.1 Extended Family Definitions

5.1.1.1 **Transport Layer Security Components**

Class FCS: Cryptographic support

Family: Transport Layer Security (FCS_TLS)

Family Behavior

This family identifies the behavior of the TOE when the Transport Layer Security (TLS) protocol is implemented. The TOE must implement one or more of the identified protocols and ciphersuites.

The FCS_TLS family contains one component.

Management:

There are no management activities foreseen.

Audit:

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

Basic:

- Failure to establish a TLS Session
- Establishment/Termination of a TLS session

5.1.1.1.1 Definition

FCS_TLS_EXT.1 – Transport Layer Security Protocol

Hierarchical to: No other components.

Dependencies: None

FCS_TLS_EXT.1.1 The TSF shall implement the following protocols [selection: TLS 1.0, TLS 1.1, TLS 1.2)] supporting the following ciphersuites:

[selection:
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_ SHA384
TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE-RSA-AES256-GCM-SHA384
TLS_ECDHE-RSA-AES128-GCM-SHA256
TLS_DHE-RSA-AES256-GCM-SHA384
TLS_DHE-RSA-AES128-GCM-SHA256
].

5.1.1.1.2 Extended Requirements Rationale:

FCS_TLS_EXT.1 is modeled on the standard component FCS_CKM.1: Cryptographic key generation. FCS_TLS_EXT.1 needed to be defined as an extended family/component because the Transport Layer Security functionality does not exist in the CC part 2.

5.1.1.2 **SSH Protocol Components**

Class FCS: Cryptographic support

Family: SSH Protocol (FCS_SSH)

Family Behavior

This family identifies the behavior of the TOE when the SSH protocol is implemented. The TOE must implement one or more of the identified protocols and ciphersuites.

The FCS_SSH family contains one component with 6 elements.

Management:

There are no management activities foreseen.

Audit:

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

Basic:

- Failure to establish an SSH Session
- Establishment/Termination of an SSH session

5.1.1.2.1 Definition

FCS_SSH_EXT.1 – SSH Protocol

Hierarchical to: No other components.

Dependencies: None

FCS_SSH_EXT.1.1 The TSF shall implement the SSH protocol.

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication method as described in RFC 4252: [**selection:** password, public key-based].

FCS_SSH_EXT.1.3 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-256, [**selection:** AES-CBC-128, AES-CBC-192, AES-CTR-128, AES-CTR-192, AES-CTR-256, 3DES-CBC, no other algorithms].

FCS_SSH_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses [**selection:** SSH_RSA, SSH_DSS] and [**selection:** PGP-SIGN-RSA, PGP-SIGN-DSS, ecdsa-sha2-nistp384, no other public key algorithms,] as its public key algorithm(s).

FCS_SSH_EXT.1.5 The TSF shall ensure that data integrity algorithms used in SSH transport connection is [**selection**: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512].

FCS_SSH_EXT.1.6 The TSF shall ensure that allowed key exchange method used for the SSH protocol is [**selection**: diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group-exchange-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, diffie-hellman-group14-sha256].

5.1.1.2.2 Extended Requirements Rationale:

FCS_SSH_EXT.1 is modeled on the standard component FCS_CKM.1 Cryptographic key generation. FCS_SSH_EXT.1 needed to be defined as an extended family/component because the SSH Protocol functionality does not exist in the CC part 2

5.1.1.3 Intrusion Detection System

Class IDS: Intrusion Detection System

This class is defined specifically for the security functionality provided by the NetWitness TOE that is not defined in CC Part 2. This class of requirements covers the security functions provided by the TOE regarding the analyzing and reporting (alerts) of the information from targeted IT System resource(s) (the IT network monitored by the TOE). This functionality is typical of an Intrusion Detection System (IDS).

The IDS Class, Families and Components are modeled on the FAU Class, Families and Components defined CC Part 2.

5.1.1.4 Analyzer

Family: Intrusion Detection System Analyzer analysis (IDS_ANL)

Family Behavior

This family defines the NetWitness functionality to perform analysis on all log and network traffic in the monitored network (IDS traffic). The TOE must also derive conclusions about potential intrusions and record the result of the analysis.

The IDS_ANL family contains one component.

Management:

There are no management activities foreseen.

Audit:

There are no auditable events foreseen.

5.1.1.4.1 Definition

IDS_ANL_EXT.1 – Analyzer analysis

Hierarchical to: No other components.

Dependencies: none

IDS_ANL_EXT.1.1 The TSF shall perform the following analysis function(s) on IDS data:

a) [**selection:** statistical, signature, integrity]; and

b) [**assignment:** other analytical functions].

Application Note: Statistical analysis involves identifying deviations from normal patterns of behavior. For example, it may involve mean frequencies and measures of variability to identify abnormal or malicious usage. Signature analysis involves the use of patterns corresponding to known attacks or misuse. For example, patterns of System settings and user activity can be compared against a database of known attacks. Integrity analysis involves comparing System settings or user activity at some point in time with those of another point in time to detect differences.

IDS_ANL_EXT.1.2 The TSF shall record within each analytical result at least the following information:

a. Date and time of the result, type of result, identification of data source; and

b. [**assignment:** other security relevant information about the result].

5.1.1.4.2 Extended Requirements Rationale:

IDS_ANL_EXT.1 is modeled on the standard components from CC part 2 and needed to be defined as an extended component because there is no requirement in the CC part 2 to cover this functionality of the TOE.

5.1.1.5 **Data Privacy Protection**

Family: Data Privacy Protection (IDS_DOR)

Family Behavior

This family defines the NetWitness functionality to protect Privacy Sensitive Data.

The IDS_DOR family contains one component.

Management: IDS_DOR.1

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

a) maintenance of the parameters that control how IDS Privacy Sensitive Data is persisted,

b) maintenance of the parameters control data retention limits.

Audit: IDS_DOR.1

The following actions should be auditable if FAU_GEN.1 Security audit data generation is included:

Basic Level:

- Modifications to permissions and users assigned to roles
- Data deletion
- Attempts (successful or not) to view or modify privacy-sensitive data, including an identification of the user who made the attempt.

5.1.1.6 **Analyzer React**

Family: Intrusion Detection System Analyzer react (IDS_RCT)

Family Behavior

This family defines the NetWitness functionality to send an alarm and perform other actions when analysis of the network traffic in the monitored network (performed in IDS_ANL) indicates there have been potential intrusions.

The IDS_RCT family contains one component.

Management:

There are no management activities foreseen.

Audit:

There are no auditable events foreseen.

5.1.1.6.1 Definition

IDS_DOR_EXT.1 – Data Privacy Protection

Hierarchical to: No other components.

Dependencies: IDS_ANL_EXT.1

IDS_DOR_EXT.1.1 The TSF shall be capable of protecting IDS Privacy Sensitive Data using the following methods: [**assignment:** Privacy Sensitive Data Protection method(s)].

5.1.1.6.2 Definition

IDS_RCT_EXT.1 – Analyzer React

Hierarchical to: No other components.

Dependencies: IDS_ANL_EXT.1

IDS_RCT_EXT.1.1 The TSF shall send an alarm to [**assignment:** alarm destination] and take [**assignment:** appropriate actions] when an intrusion is detected.

Application Note: There must be an alarm, though the ST should refine the nature of the alarm and define its target (e.g., administrator console, audit log). The Analyzer may optionally perform other actions when intrusions are detected; these actions should be defined in the ST. An intrusion in this requirement applies to any conclusions reached by the analyzer related to past, present, and future intrusions or intrusion potential.

5.1.1.6.3 Extended Requirements Rationale:

IDS_DOR_EXT.1 is modeled closely on the standard components from CC part 2 and needed to be defined as an extended component because there is no requirement in the CC part 2 to cover this functionality of the TOE.

5.1.1.6.4 Extended Requirements Rationale:

IDS_RCT_EXT.1 is modeled closely on the standard components from CC part 2 and needed to be defined as an extended component because there is no requirement in the CC part 2 to cover this functionality of the TOE.

5.1.1.7 **Restricted Data Review**

Family: Intrusion Detection System Restricted Data Review (IDS_RDR)

Family Behavior

This family defines the NetWitness functionality for data analyzing tools that must be available to authorized users to assist in the review of data collected from the monitoring network. This family indicates that the TOE must provide authorized users the capability to obtain, review and interpret the information.

This family consists of one component.

Management:

There are no management activities foreseen.

Audit:

There are no auditable events foreseen.

5.1.1.7.1 Definition

IDS_RDR_EXT.1 – Restricted Data Review

Hierarchical to: No other components.

Dependencies: IDS_ANL_EXT.1

IDS_RDR_EXT.1.1 The TSF shall provide [**assignment:** authorised users] with the capability to read [**assignment:** list of IDS data] from the IDS data.

Application Note: This requirement applies to authorised users of the TSF. The requirement is left open for the writers of the ST to define which authorised users may access what TSF data.

IDS_RDR_EXT.1.2 The TSF shall provide the IDS data in a manner suitable for the user to interpret the information.

IDS_RDR_EXT.1.3 The TSF shall prohibit all users read access to the IDS data, except those users that have been granted explicit read-access.

5.1.1.7.2 Extended Requirements Rationale:

IDS_RDR_EXT.1 is modeled closely on the standard components from CC part 2 (more specifically, this SFR is modeled on FAU_SAR components) and needed to be defined as an extended component because there is no requirement in the CC part 2 to cover this functionality of the TOE

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by RSA NetWitness.

Requirement Class	Requirement Component
FAU: Security Audit	FAU_GEN.1: Audit data generation
	FAU_GEN.2: User identity association
	FAU_SAR.1: Audit review
	FAU_SAR.2: Restricted audit review
	FAU_STG.1: Protected audit trail storage
FCS: Cryptographic support	FCS_SSH_EXT.1: SSH Protocol
	FCS_TLS_EXT.1: Transport Layer Security Protocol
FIA: Identification and authentication	FIA_AFL.1: Authentication failure handling (Human user)
	FIA_ATD.1: User attribute definition
	FIA_UAU.1: Timing of authentication
	FIA_UAU.5: Multiple authentication mechanisms
	FIA_UID.1: Timing of identification
Security Monitoring with Security Information and Event Management (SIEM)	IDS_ANL_EXT.1: Analyzer analysis
	IDS_DOR_EXT.1: Data Privacy Protection
	IDS_RCT_EXT.1: Analyzer react
	IDS_RDR_EXT.1: Restricted Data Review
FMT: Security management	FMT_MOF.1(1): Management of security functions behaviour
	FMT_MOF.1(2): Management of security functions behaviour
	FMT_MTD.1: Management of TSF data
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_ITT.1: Basic internal TSF data transfer protection
TOE Access	FTA_SSL.3: TSF-initiated Termination
	FTA_SSL.4: User-initiated Termination
	FTA_TAB.1: Default TOE access banners
FTP: Trusted path/channels	FTP_TRP.1: Trusted Path

Table 5-1 TOE Security Functional Components

5.2.1 Security audit (FAU)

5.2.1.1 **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) **The specifically defined auditable events listed in Table 5-2 Auditable Events**].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**information specified in column three of Table 5-2 Auditable Events**].

Table 5-2 Auditable Events

Security Functional Requirement	RSA Identified Auditable Event	Additional Audit Record Contents
FCS_TLS_EXT.1	Failed and successful establishment of a TLS Session	No additional information
FIA_AFL.1	Each failed authentication attempt is audited	No additional information
FIA_UAU.1	Failed login Successful login	No additional information
FIA_UAU.5	An attempt to authenticate to a SFTP connection will generate logs for both failed and successful login authentication events. See FIA_UAU.1/FIA_UID.	No additional information
FIA_UID.1	Failed login Successful login	No additional information
FMT_MOF.1	Modifications to permissions and users assigned to roles related to management of the Privacy Data Function	No additional information
FMT_MOF.1	Data deletions related to the management of the Privacy Data Function.	No additional information
FMT_MOF.1	Attempts (successful or not) to view or modify privacy-sensitive data.	No additional information
FMT_SMF.1	Create, modify, delete users	Module or Service where audit record originated Connection information (IP address)
FMT_SMF.1	Create, modify LockBox password	Module or Service where audit record

Security Functional Requirement	RSA Identified Auditable Event	Additional Audit Record Contents
		originated
FMT_SMR.1	Creating/updating/enabling/disabling users; Creating/updating roles	No additional information
FTA_SSL.3	TSF initiated termination of interactive user sessions due to session inactivity	No additional information
FTA_SSL.4	User logouts are audited, which terminates session	No additional information
FTP_TRP.1	Initiation of trusted channel (Successful logins) FCS_TLS_EXT.1 or FIA_UAU.1 audit events serve here since TLS and authentication are the trusted path mechanisms.	No additional information
	Termination of trusted channel (User logouts) FTA_SSL.3 and FTA_SSL.4 audit events serve here since TLS and authentication are the trusted path mechanisms.	No additional information
	Failures of the trusted path functions (Failed logins) FCS_TLS_EXT.1 or FIA_UAU.1 audit events serve here since TLS and authentication are the trusted path mechanisms. Note: A system-level log is generated for FCS_TLS_EXT.1 in cases where the negotiated TLS handshake has not completed and a failure occurs: for example, the client doesn't have the right cyphers.	No additional information
IDS_DOR_EXT.1	Modifications to permissions and users assigned to roles.	No additional information
	Data deletion	No additional information
	Attempts (successful or not) to view or modify privacy-sensitive data, including an identification of the user who made the attempt.	No additional information

5.2.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.2.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide [**Operator, Administrator, and Data Privacy Officer**] with the capability
to read [**all audit 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.

5.2.1.4 Audit Restricted audit review (FAU_SAR.2)

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

5.2.1.1 Protected audit trail storage (FAU_STG.1)

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

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

5.2.2 Cryptographic Support (FCS)

5.2.2.1 SSH Protocol (FCS_SSH_EXT.1)

FCS_SSH_EXT.1.1 The TSF shall implement the SSH protocol.

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication method as described in RFC 4252: [***public key-based***].

FCS_SSH_EXT.1.3 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-256, [***AES-CTR-128, AES-CTR-192, AES-CTR-256***].

FCS_SSH_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses [***SSH_RSA, SSH_DSS***] and [***no other public key algorithms***] as its public key algorithm(s).

FCS_SSH_EXT.1.5 The TSF shall ensure that data integrity algorithms used in SSH transport connection is [***hmac-sha1***, ***hmac-sha2-256, hmac-sha2-512***].

FCS_SSH_EXT.1.6 The TSF shall ensure that allowed key exchange method used for the SSH protocol is [***ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group-exchange-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, diffie-hellman-group14-sha256***]

5.2.2.2 Transport Layer Security Protocol (FCS_TLS_EXT.1)

FCS_TLS_EXT.1.1 The TSF shall implement the following protocols [***TLS 1.2***] supporting the following ciphersuites: [

TLS_DHE-RSA-AES256-GCM-SHA384
TLS_DHE-RSA-AES128-GCM-SHA256
TLS_ECDHE-RSA-AES256-GCM-SHA384
TLS_ECDHE-RSA-AES128-GCM-SHA256].

5.2.3 Identification and authentication (FIA)

5.2.3.1 Authentication failure handling (Human user) (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when [***[an administrator configurable positive integer within [0 and no maximum value]]***] unsuccessful authentication attempts occur related to [**NetWitness UI user authentication**].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [***surpassed***], the TSF shall [**lock account for a specified time period as configured by authorized administrator**].

5.2.3.2 User attribute definition (FIA_ATD.1)

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

- **Username;**
- **password; and**
- **role**].

5.2.3.3 Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow [**acknowledge end-user license agreement and view warning banner**] 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.

5.2.3.4 Multiple authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide [**SSH public-key, and password-based authentication mechanisms**] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [**TOE users authenticate using password-based and authorized IT entities authenticate using SSH public-key authentication mechanisms**].

5.2.3.5 Timing of identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow [**acknowledge end-user license agreement and view warning banner**] 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.

5.2.4 Security Monitoring with Security Information and Event Management

5.2.4.1 Analyzer analysis (IDS_ANL_EXT.1)

IDS_ANL_EXT.1.1 The TSF shall perform the following analysis function(s) on IDS data:

a. [***statistical, signature***]; and

b. [**no other analytical functions**].

IDS_ANL_EXT.1.2 The TSF shall record within each analytical result at least the following information: a. Date and time of the result, type of result, identification of data source; and b. [**no additional information**].

5.2.4.2 Data Privacy Protection (IDS_DOR_EXT.1)

IDS_DOR_EXT.1.1 The TSF shall be capable of protecting IDS Privacy Sensitive Data using the following methods: [

- **Prevent persistence of both original and obfuscated data**
- **Prevent persistence of original data and store only obfuscated data**
- **Persisting both original and obfuscated data and restrict access to privileged access role**
- **Prevent transfer of Data Privacy protected IDS data to other components**
- **Enforce data retention limits by date or age**].

5.2.4.3 Analyzer react (IDS_RCT_EXT.1)

IDS_RCT_EXT.1.1 The TSF shall send an alarm to [**NetWitness Respond User Interfaces**] and take [**no other action**] when an intrusion is detected.

5.2.4.4 Restricted data review (IDS_RDR_EXT.1(1))

IDS_RDR_EXT.1.1(1) The TSF shall provide [**Respond Administrator**, **SOC_Manager**, **Analyst, and Data Privacy Officer**] with the capability to read [**all of the following non-protected Analyzer data: metadata, raw logs, raw packet data, and Incident Management data**] from the IDS data.

IDS_RDR_EXT.1.2(1) The TSF shall provide the IDS data in a manner suitable for the user to interpret the information.

IDS_RDR_EXT.1.3(1) The TSF shall prohibit all users read access to the IDS data, except those users that have been granted explicit read-access.

5.2.4.5 Restricted data review (IDS_RDR_EXT.1(2))

IDS_RDR_EXT.1.1(2) The TSF shall provide [**Administrator**, **Data Privacy Officer**] with the capability to read [**all of the Data Privacy Sensitive Protected data**] from the IDS data.

IDS_RDR_EXT.1.2(2) The TSF shall provide the IDS data in a manner suitable for the user to interpret the information.

IDS_RDR_EXT.1.3(2) The TSF shall prohibit all users read access to the IDS data, except those users that have been granted explicit read-access.

5.2.4.6 Restricted data review (IDS_RDR_EXT.1(3))

IDS_RDR_EXT.1.1(3) The TSF shall provide [**Administrator**, **Data Privacy Officer, Analyst**] with the capability to read [**the original Data Privacy Sensitive Protected data**] from the IDS data.

IDS_RDR_EXT.1.2(3) The TSF shall provide the IDS data in a manner suitable for the user to interpret the information.

IDS_RDR_EXT.1.3(3) The TSF shall prohibit all users read access to the IDS data, except those users that have been granted explicit read-access.

5.2.5 Security management (FMT)

5.2.5.1 Management of security functions behaviour (FMT_MOF.1(1))

FMT_MOF.1.1(1) The TSF shall restrict the ability to [***disable, enable***] the functions [**audit function**, **the specified start/stop services (decoder, concentrator, broker, ESA, Archiver, Respond, Malware Analysis, Reporting Engine, and Automated Threat Detection)**] to [**Administrator, Operator**].

5.2.5.2 Management of security functions behaviour (FMT_MOF.1(2))

FMT_MOF.1.1(2) The TSF shall restrict the ability to [***determine the behaviour of, disable, enable, modify the behaviour of***] the functions [**Data Privacy Protection**] to [**Administrator, Data Privacy Officer**].

5.2.5.3 Management of TSF data (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to [***query, [manage]***] the [**TSF data**] to [**authorized identified roles in Table 5-3 Management of TSF Data**].

Table 5-3: Management of TSF Data

TSF data	Operation(s)	Role
Login failure limit	Change	Administrator, Data Privacy Officer
Lockout period	Change	Administrator, Data Privacy Officer
Session inactivity period	Change	Administrator, Data Privacy Officer
User accounts	Create, modify, delete	Administrator, Data Privacy Officer
Security audit	Read (query)	Operators, Administrator, Data Privacy Officer
Analyzer data	Read (query)	Analysts, SOC_Manager, Respond Administrator, Data Privacy Officer
Malware Analysis data	Read (query)	Administrator, SOC_Manager, Malware Analyst, Data Privacy Officer, Respond Administrator (Incident Data only)
Privacy Sensitive data	Create, modify, delete,	Administrator, Data Privacy Officer
Privacy Sensitive data	Read (Query)	Administrator, Data Privacy Officer, Analyst
Text (system settings)	Enable/disable, customize	Administrator, Data Privacy Officer
Devices	Add. Remove	Administrator, Data Privacy Officer
Log Decoder event sources connections	Add, update, delete	Administrator, Operator, Data Privacy Officer
Signatures	Add, modify, remove	Administrator, Operator, Data Privacy Officer
LockBox password for Log Collectors	Create, modify	Administrator, Operator, Data Privacy Officer

5.2.5.4 Specification of Management Functions (FMT_SMF.1)

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

- **Manage TSF functions as specified in FMT_MOF.1(1), and FMT_MOF.1(2)**
- **Manage TSF data as specified in FMT_MTD.1**
- **Manage security audit as specified in FMT_MTD.1, FAU_STG.1**

].

5.2.5.5 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles: [

- **Administrator**
- **Respond Administrator**
- **Analyst**
- **Operator**
- **SOC_Manager**
- **Malware Analyst**
- **Data Privacy Officer**].

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

5.2.6 Protection of the TSF (FPT)

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

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

5.2.7 TOE Access (FTA)

5.2.7.1 TSF-initiated termination (FTA_SSL.3)

FTA_SSL.3.1 The TSF shall terminate an interactive session after a [**time interval of user inactivity configured by authorized administrator**].

5.2.7.2 TSF-initiated termination (FTA_SSL.4)

FTA_SSL.4.1 The TSF shall allow user-initiated termination of the user's own interactive session.

5.2.7.3 Default TOE access banners (FTA_TAB.1)

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

5.2.8 Trusted path/channels (FTP)

5.2.8.1 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1 The TSF shall provide a communication path between itself and **[*remote*]** users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **[*modification, disclosure*]**.

FTP_TRP.1.2 The TSF shall permit [***remote users***] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [***initial user authentication [all NetWitness interface session data, transfer of file event source data]***].

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 augmented with ALC_FLR.1 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.2: Security-enforcing functional specification
	ADV_TDS.1: Basic design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.2: Use of a CM system
	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_FLR.1: Basic flaw remediation
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements

	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.1: Evidence of coverage
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing — sample
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

Table 5-4 EAL2 Augmented with ALC_FLR.1 Assurance Components

5.3.1 Development (ADV)

5.3.1.1 Security architecture description (ADV_ARC.1)

ADV_ARC.1.1d The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2d The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3d The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1c The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2c The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3c The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4c The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5c The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

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

5.3.1.2 Security-enforcing functional specification (ADV_FSP.2)

ADV_FSP.2.1d The developer shall provide a functional specification.

ADV_FSP.2.2d The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.2.1c The functional specification shall completely represent the TSF.

ADV_FSP.2.2c The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3c The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.2.4c For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.2.5c For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.

ADV_FSP.2.6c The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

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 SFRs.

5.3.1.3 Basic design (ADV_TDS.1)

ADV_TDS.1.1d The developer shall provide the design of the TOE.

ADV_TDS.1.2d The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.1.1c The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.1.2c The design shall identify all subsystems of the TSF.

ADV_TDS.1.3c The design shall describe the behaviour of each SFR-supporting or SFR-non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.

ADV_TDS.1.4c The design shall summarise the SFR-enforcing behaviour of the SFR-enforcing subsystems.

ADV_TDS.1.5c The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.

ADV_TDS.1.6c The mapping shall demonstrate that all TSFIs trace to the behaviour described in the TOE design that they invoke.

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

ADV_TDS.1.2e The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance documents (AGD)

5.3.2.1 Operational user guidance (AGD_OPE.1)

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

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

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

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

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

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

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

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

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

5.3.2.2 Preparative procedures (AGD_PRE.1)

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

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

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

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

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

5.3.3 Life-cycle support (ALC)

5.3.3.1 Use of a CM system (ALC_CMC.2)

ALC_CMC.2.1d The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.2.2d The developer shall provide the CM documentation.

ALC_CMC.2.3d The developer shall use a CM system.

ALC_CMC.2.1c The TOE shall be labelled with its unique reference.

ALC_CMC.2.2c The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.2.3c The CM system shall uniquely identify all configuration items.

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

5.3.3.2 Parts of the TOE CM coverage (ALC_CMS.2)

ALC_CMS.2.1d The developer shall provide a configuration list for the TOE.

ALC_CMS.2.1c The configuration list shall include the following: The TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.

ALC_CMS.2.2c The configuration list shall uniquely identify the configuration items.

ALC_CMS.2.3c For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

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

5.3.3.3 Delivery procedures (ALC_DEL.1)

ALC_DEL.1.1d The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2d The developer shall use the delivery procedures.

ALC_DEL.1.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

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

5.3.3.4 Basic flaw remediation (ALC_FLR.1)

ALC_FLR.1.1d The developer shall document and provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.1.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2c 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.1.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4c 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.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Security Target Evaluation (ASE)

5.3.4.1 Conformance Claims (ASE_CCL.1)

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

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

5.3.4.2 Extended components definition (ASE_ECD.1)

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

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

ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

5.3.4.3 ST introduction (ASE_INT.1)

ASE_INT.1.1D The developer shall provide an ST introduction.

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall identify the TOE.

ASE_INT.1.4C The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

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

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

5.3.4.4 Security objectives (ASE_OBJ.2)

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

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

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

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

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

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

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

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

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

5.3.4.5 Derived security requirements (ASE_REQ.2)

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

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

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

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

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

5.3.4.6 Security problem definition (ASE_SPD.1)

ASE_SPD.1.1D The developer shall provide a security problem definition.

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

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

5.3.4.7 TOE summary specification (ASE_TSS.1)

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

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

ASE_TSS.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

5.3.5 Tests (ATE)

5.3.5.1 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 in the test documentation and the TSFIs 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.

5.3.5.2 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, expected test results and actual test results.

ATE_FUN.1.2c The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3c The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4c The actual test results shall be consistent with the expected test results.

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

5.3.5.3 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 execute a sample of tests in the test documentation to verify the developer test results.

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

5.3.6 Vulnerability assessment (AVA)

5.3.6.1 Vulnerability analysis (AVA_VAN.2)

AVA_VAN.2.1d The developer shall provide the TOE for testing.

AVA_VAN.2.1c The TOE shall be suitable for testing.

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

AVA_VAN.2.2e The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3e The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4e The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6 TOE Summary Specification

This chapter describes the following security functions:

- Security audit
- Cryptographic support
- Identification and authentication
- Security Monitoring with Security Information and Event Management (SIEM)
- Security management
- Protection of the TSF
- TOE Access
- Trusted path/channels

6.1 Security audit

The TOE generates audit records for the following auditable events:

- Start-up and shutdown of the audit function,
- Start-up and shutdown of the TOE,
- All auditable events as specified in Table 5-2 Auditable Events in Section 5.2.1.1.

Each audit record includes the date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event.

The operating system in the environment provides protection, and storage of the audit records. The operating system also provides the system clock information that is used by the TOE to timestamp each audit record. The audit records are stored on the local file system of the host appliance. The TOE is a multi-host distributed architecture, where the TOE subsystems run on a number of hosts. The audit records are stored on the local file system of the host on which the related auditable event is detected. Consequently, the aggregate audit record for an entire TOE system is distributed across multiple hosts, rather than being stored in a single location.

The TOE provides a web-based NetWitness Server UI, through which an authorized user with the Operator or Administrator role has the ability to read all audit information from the audit records on each appliance. Additionally, the TOE does not provide any interfaces to delete or modify audit records.

The TOE relies upon the environment to provide typical operating system file services including protected data storage. The TOE relies upon the operating system in the operational environment to provide the file system which allows the TOE to store information securely. The TOE relies upon the environment to prevent unauthorized modification or deletion of the audit files.

The TOE does not provide the ability to start and stop the audit mechanism independently from the starting and stopping of the services. The audit mechanism is started and stopped when the service is started and stopped. The TOE generates an audit event when each service is started and stopped.

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

- FAU_GEN.1: Audit records are generated for security relevant events and include the date and time of the event, type of event, subject identity, outcome of the event, and other data identified earlier in this section.
- FAU_GEN.2: The TOE associates each auditable event resulting from actions of identified users with the identity of the user that caused the event.
- FAU_SAR.1: The TOE provides users with the Operator and Administrator roles with the capability to read all audit information from the audit records. The TSF provides the audit records in a manner suitable for the user to interpret the information.
- FAU_SAR.2: The TOE prohibits all users read access to the audit records, except those users that have been granted explicit read-access.

- FAU_STG.1: The TOE does not provide the capability to delete or modify audit records. Hence, the TOE protects the stored audit records in the audit trail from unauthorised deletion. The TOE is able to prevent unauthorised modifications to the stored audit records in the audit trail.

6.2 Cryptographic support

The TOE uses cryptography to support the protection of the following types of communication pathways:

- Administrative login and management sessions,
- TOE appliance to TOE appliance, and
- File event source to Log Collector.

A remote administrative management session is initiated by a login and occurs only over HTTPS using TLS (TLS version 1.2 in FIPS mode). The TOE performs TLS cryptographic operations in a FIPS-compliant mode of operation using a FIPS-validated cryptographic module. TOE to TOE communication occurs for the purpose of TOE device/appliance communication with one another. Each instance of the TOE ensures that such communication occurs only over a TLS in FIPS mode protected communication pathway.

The TOE uses the RSA BSAFE Crypto-J cryptographic library: BSAFE SSL-J 6.2.1.1 for Java applications, which incorporates BSAFE Crypto-J 6.2. The latter is certified under FIPS 140-2 Certificate #2468. The Lockbox uses the Common Security Toolkit, version 3.3.0.12. Crypto-C ME 4.1.2, and is covered by Certificates #2300.[9] The lockbox uses TLS_RSA_WITH_AES_256_CBC_SHA and TLS_RSA_WITH_AES_256_CBC_SHA256.

The TOE uses a FIPS-validated module for SSH protected communication pathways for the transfer of file event source data from log data sources to the TOE. The TOE implements the SSH protocol and supports public key-based authentication as described in RFC 4252. The TSF uses the following encryption algorithms: AES-CTR-128, AES-CTR-192, and AES-CTR-256 for SSH transport. The SSH transport implementation uses SSH_RSA, SSH_DSS public key algorithms; and hmac-sha1, hmac-sha2-256, and hmac-sha2-512 data integrity algorithms for the SSH transport connection. The key exchange methods used for the SSH protocol include: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group-exchange-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, and diffie-hellman-group14-sha256.

The TOE uses Crypto-C ME 4.1.2 version 4.1.2, which has undergone a FIPS 140-2 certification (certificate #2300) for both SSH and TLS. The TOE operates in FIPS mode in the evaluated configuration. FIPS is enabled by default on all services. FIPS 140-2 Certified Cryptographic Modules are enabled for all services that perform cryptographic operations. Although the FIPS Cryptographic Module is leveraged, it is not enforced for the following services: NTP, CollectD, ssh (on Log Collector), Salt, Decoder, Log Collector and Log Decoder. Based on this, any internal communications between NetWitness services will utilize FIPS Cipher Suite while external clients that do not support FIPS cipher suites and interact with these services will still be able to support SSL/TLS handshake and connection.

The TOE implements TLS version 1.2 as specified by RFC 5246 using the following ciphersuites.

- TLS_DHE-RSA-AES256-GCM-SHA384
- TLS_DHE-RSA-AES128-GCM-SHA256
- TLS_ECDHE-RSA-AES256-GCM-SHA384
- TLS_ECDHE-RSA-AES128-GCM-SHA256

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

- FCS_SSH_EXT.1: The TOE implements the SSH protocol according to RFC 4252 for public key-based and using the algorithms and key exchange method as described in the text above.

[9] Lockbox protects the passwords required for some Log Sources by encryption (using a FIPS-validated cryptographic module). This protection does not contribute to satisfying any TOE security functional requirements.

- FCS_TLS_EXT.1: The TOE implements TLS Version 1.2 protocol which is used as described in the text above. The TOE implementation of TLS provides the ciphersuites listed above.

6.3 Identification and authentication

The TOE maintains user accounts for the authorized users of the TOE and a list of security attributes for each user which includes the username, role membership, and password. The TOE maintains the security relevant roles of Administrator, Respond Administrator, Operator, Analyst, SOC_Manager, Malware Analyst, and Data Privacy Officer.

The TOE requires users to provide unique identification and passwords before any access to the TOE is granted other than acknowledging the end-user license agreement and viewing the warning banner. The TOE authenticates claimed identities for TOE users using password-based mechanisms and for authorized IT entities using SSH public-key authentication mechanisms. IT entities are log collection sources that have been configured to send logs to the TOE. SSH public-key authentication is used when log sources send file event sources (SFTP).

The TOE maintains role and user attributes on each device. The TOE associates attributes with a user through NetWitness Server login and NetWitness trusted connections. The NetWitness server login authenticates the identity of a human user and NetWitness trusted management connections initiated by administrators by providing the user identity from the NetWitness server to other NetWitness devices. NetWitness trusted connections use client authentication for TLS connections. For example, when NetWitness server establishes a NetWitness trusted connection to a Concentrator, the Concentrator (TLS server) authenticates the NetWitness server (TLS client). Once authenticated, the NetWitness server provides the Concentrator with the user's authenticated identity and roles. TOE devices will trust the certificate of the NetWitness server when an administrator has added the TLS client's PEM encoded certificate to the server's trusted peer list. When the Concentrator then subsequently establishes a connection to other devices, it will present an authentication token for the user, which contains the username, and role. If the device recognizes the TLS client device's certificate as trusted and if the role is defined locally, it will allow the user access. If the TLS server device does not accept the TLS client device's certificate, then the client must authenticate by providing the username and role(s) for that session. If a role is not defined locally, it is ignored. For example for Archiver, Log Collector, Log and Packet Decoder/Concentrator; if the device does not have the role defined it is ignored and NetWitness Server has to connect using the legacy method (present usrname/passwd credentials). For all other devices/services (that is, Broker, ESA, Malware Analysis devices) trust is determined as follows. If the TLS client device reports user "tim" has roles "A", "B", "C" but the TLS server device only defines roles "X" and "Y" then when user tim makes an API call the API call will be rejected on any APIs that are controlled with roles "X" and "Y". The NetWitness Server acts as the TLS client and the device acts as the TLS server. The device uses the NetWitness Server certificate in the TLS exchange to authenticate the NetWitness Server. The TLS client device does not authenticate the TLS server device.

Once access to the TOE is granted, authorization to access functions and data is implemented via the user's role membership. User roles are the central point of authorization in the TOE's security model. User roles are created with a specific set of permissions which apply to all users assigned to the role.

The TOE is able to detect when an administrator configurable positive integer of unsuccessful authentication attempts occur related to NetWitness Server UI user authentication. When the defined number of unsuccessful authentication attempts has been surpassed, the TOE locks the user account for a specified time period as configured by authorized administrator. The range of time values which can be configured for the maximum number of login attempts is min 0, no max and default is 5 (Maximum number of login attempts allowed before an account is locked out is 5). The range of time values which can be configured for the lockout time is min 0, no max and default is 20 minutes (Period of time where locked out accounts remain locked out). Note that 0 means lockout is disabled. Once the session is locked and the timeframe for locking the session has passed the user may log back in by providing their username and password. When an unsuccessful authentication attempt has been detected the TOE audits the failed authentication attempt.

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

- FIA_AFL.1: The TOE is able to detect when an administrator configurable positive integer of unsuccessful authentication attempts occur related to NetWitness UI user authentication. When the defined number of

unsuccessful authentication attempts has been surpassed, the TOE locks the user account for a specified time period as configured by authorized administrator.

- FIA_ATD.1: The TOE maintains a list of security attributes: Username, password, role for individual users.
- FIA_UAU.1: The TOE allows "acknowledge end-user license agreement and view warning banner" on behalf of the user to be performed before the user is authenticated. Otherwise the TOE requires each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UAU.5: The TOE provides SSH public-key, and password-based authentication mechanisms to support user authentication. The TOE authenticates claimed identities for TOE users using password-based and for authorized IT entities using SSH public-key authentication mechanisms.
- FIA_UID.1: The TOE allows "acknowledge end-user license agreement and view warning banner" on behalf of the user to be performed before the user is identified. Otherwise the TOE requires each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.4 Security Monitoring with Security Information and Event Management

The TOE accepts network and log data and analyzes the data for anomalous and inappropriate activity. The TOE:

- Reconstructs network traffic,
- Parses network traffic and log data to identify metadata,
- Analyzes and adds metadata with Decoder rules and feeds,
- Indexes metadata on the Concentrator for analysis,
- Analyzes metadata by applying ESA, Malware Analysis, and ATD rules,
- Generates an alert when a scoring threshold is met or when a rule matches.

The parsers, indices, and rules encapsulate signatures of anomalous and inappropriate activity. Feeds are used to create metadata based on externally defined metadata values. The analysis process includes signature analysis and malware analysis when indicated. The TOE identifies known patterns based on parsed metadata from transactions. The out of the box rules identify patterns representing known attacks. Additional rule content is developed specifically for threats and can include a file type rule which when matched would result in Malware Analysis processing. Malware analysis uses network session analysis, and static file analysis to check for malware. The device can perform continuous or on-demand polling to extract sessions identified as potentially carrying malware.

There are three types of Decoder rules: Correlation, Network, and Application.

Basic Correlation Rules are applied at the session level and alert the user to specific activities that may be occurring in their environment. Correlation rules are applied over a configurable slice of time on a Decoder. When the conditions are met, alert metadata is created for this activity and there is a visible indicator of the suspicious activity.

Network rules do not apply to Log Decoders. Network layer rules are applied at the packet level on a Decoder and are made up of rule sets from Layer 2 - Layer 4. Network rules can apply to multiple network layers (for example, when a network rule filters out specific ports for a specific IP address).

Application rules can be applied to both Decoders and Log Decoders. Application layer rules are applied at the session level. Rule conditions trigger an action when matched. One of the following actions is applied when a matching packet is found depending on the defined rule.

- Keep: The packet payload and associated meta are saved when they match the rule.
- Filter: The packet is not saved when it matches the rule.
- Truncate: The packet payload is not saved when it matches the rule, but packet headers and associated meta are retained.
- Alert: The packet payload and associated meta are saved and an alert is triggered when they match the rule.

Automated Threat Detection service residing on ESA applies rule logic using behavior-based statistical algorithms to analyze metadata to identify outliers, abnormal behavior, and malicious activity. The current algorithms and rules focus on metadata associated with network http traffic and web proxy logs. The service is built to allow for expansion to add additional algorithms and rules for other types of data. ESA ingests metadata from the Concentrators and utilizes the metadata for both for ESA rules and Automated Threat Detection. The service only selects metadata associated with algorithms/rules it had loaded for its analysis. As data is analyzed, certain behaviors (or conditions) are checked, and if behaviors or conditions occur, it moves to the next step in the process. The following includes some examples of the workflow and behaviors or conditions. (IDS_ANL_EXT.1)

- Whitelist is checked- If a whitelist was created, the ESA checks this list to rule out domains. If a domain in the event is whitelisted, the event is ignored.
- The domain profile is checked - Automated Threat Detection checks to see if the domain is newly seen (approximately three days), has few source IP connections, has many connections without a referer, or has connections with a rare user agent. If one or several of these conditions is true, the domain is next checked for periodic beaconing.
- The domain is checked for periodic beaconing. Beaconing occurs when the malware regularly sends communications back to the command and control server to notify it that a machine has been compromised and the malware is awaiting further instructions. If the site displays beaconing behavior, then the domain registration information is checked.
- Domain registration information is checked. WHOIS is used to see if the domain is recently registered or nearly expired. Domains that have a very short lifespan are often hallmarks of malware.
- Command and control (C2) aggregates scores. Each of the above factors generates a separate score which is weighted to denote various levels of importance. The weighted scores determine if an alert should be generated. If an alert is generated, the aggregated alerts appear in the Incident Manager and can then be investigated further from there.

The TOE provides a means of protecting privacy-sensitive data. The Administrator and DPO can configure NetWitness Suite to limit exposure of meta data and raw content (packets and logs) using a combination of techniques. The methods available to protect data in NetWitness Suite include: Data Obfuscation, and Data Retention Enforcement. Audit Logging supports accountability by generating audit events of related activity such as Modifications to permissions and users assigned to roles; Data deletion events; and attempts (successful or not) to view or modify privacy-sensitive data, including an identification of the user who made the attempts.

Data privacy officers and administrators can specify which meta keys in their environment are privacy-sensitive and limit where the meta values and raw data for those keys are displayed in the NetWitness Suite network. In place of the original values, NetWitness Suite can provide obfuscated representations to enable investigation and analytics. In addition, DPOs and administrators can prevent persistence of privacy-sensitive meta values and raw logs or packets.

Meta keys configured as protected can be represented by obfuscated values at the time of creation on a Decoder or Log Decoder; the obfuscated values are hashed using FIPS approved SHA-256 and optional salt values. Meta keys configured as protected can also be configured to be persisted with access to the original data restricted to authorized roles. Finally, meta keys configured as protected can also be configured as transient, allowing the Analytical functions to utilize the data in the analysis functions. The data is not persisted, but is only in memory long enough for the analysis functions to take place.

The NetWitness Privacy Sensitive Data Protection Function ensures that data is retained only as long as necessary or as specified. An Administrator or DPO can configure data retention using age and time thresholds on a per-service basis. Schedulers running on each service automatically delete data meeting those thresholds. Once the data is deleted, it is no longer available through user interfaces, queries, or application programming interface (API) calls. Some of the NetWitness Suite components also support purging of data through overwrites. An administrator can manage data retention in several ways:

- Configure how long data persists in storage on the system.

- For Core services, strategically remove privacy-sensitive data that may have been written by configuring automatic removal of data of a specific age.
- Configure NetWitness Suite so that original data is not sent or saved to the other components. If privacy-sensitive data makes its way into another database on the Reporting Engine, Malware Analysis, and NetWitness Servers, data retention can be managed there as well.
- If a situation arises where the DPO decides that already collected data is privacy-sensitive after the system is functional, the administrator can manually overwrite the data from databases or files where the data is saved.

The recommended configuration to obtain the best analytical value with data obfuscation enabled is to define privacy-sensitive meta data and keep both original and obfuscated (hash) values of privacy-sensitive data on disk for Decoders, Log Decoders, Concentrators, and Brokers. The built-in and automatic data retention enforcement function deletes data at a certain threshold configured the authorized administrator. To manage cache storage, the NetWitness Server clears cache related to investigations of events every 24 hours.

NetWitness Suite provides alternative controls that the administrator can apply to enforce stronger restrictions on privacy-sensitive data storage when data obfuscation is enabled. The first option is to store only the obfuscated value and eliminate the persistence of sensitive data to disk. In this scenario, meta data generated during parsing on the Decoders and Log Decoders is used only in memory and not written to disk. Administrators can configure individual meta keys on a Decoder or Log Decoder as transient to ensure that sensitive meta data is not written to disk. Downstream services do not see original values and must use obfuscated values to conduct investigation and analytics. Original values identified as sensitive are extracted from the raw data during parsing on the Decoder and Log Decoder and are accessible to the system during parsing (parsers, rules, feeds). The Decoder does not save the original values for meta keys identified as sensitive, storing only the hash of original values along with other non-sensitive meta data related to the event.

The second option is eliminate the persistence of the original value to disk entirely if the risk of exposure is too great. Neither the original value nor obfuscated values are persisted. As in Option 1, in this scenario, meta data generated during parsing on the Decoders and Log Decoders is used only in memory and not written to disk. Administrators can configure individual meta keys on a Decoder or Log Decoder as transient to ensure that sensitive meta data is not written to disk. Downstream services do not see original values and have no obfuscated values to conduct investigation and analytics.

The Administrator and DPO can Purge Data Using String and Pattern Redaction Options. The Data purging option provides a mechanism to strategically overwrite a specific subset of data from the system in case any sensitive data has been persisted either on purpose or by accident. The NetWitness Suite wipe utility allows for unique patterns to be written over the data in the meta and packet databases for Core services, which may contain RAW packets or logs for existing sessions, based on a session identifier. All Core components have the capability to overwrite a subset of data that has been found by executing a query string, including regex patterns. The session identifiers resulting from the query are fed into the NetWitness Suite wipe utility.

The TOE is capable of receiving events from different source types (e.g. Syslog, SNMP Trap, NetFlow…) covering over 350 specific devices. RSA content team authors rules for parsing content from a particular device (Log Decoder). These parser rules make up the signatures for log events (and network packets) to be collected (IDS_ANL_EXT.1). Parsing rules or signatures are included for the following event sources:

a. Syslog
b. SNMP Trap
c. NetFlow
d. File
e. Windows (WinRM)
f. Windows (Legacy)
g. ODBC
h. Check Point LEA
i. VMWare
j. SDEE

k. Cloud (Including AWS CloudTrail and Microsoft Azure)
l. Office365

Note: LockBox protects credentials for log sources, but the functionality does not implement any security functional requirements. The TOE protects log source by encryption (using a FIPS-validated cryptographic module). This protection does not contribute to satisfying any TOE security functional requirements.

Categories of ESA Rules (out of the box):

- Log Events with certain criteria
- Active Directory Policy Modification
- Adapter Events
- Backdoor Activity
- Brute Force Login
- Traffic detection with certain criteria
- Port Activity and port scan
- Login and attempted login activity including account lockouts
- DNS activity
- Account creation and password changes
- Connection attempts
- Scan Events
- P2P Software detection
- Privilege escalation
- Configuration changes
- Windows account creation with subsequent management activity
- Windows audit log cleared

Malware Analysis Rules / Methodologies (out of the box):

- Network Session Analysis: metadata check: attribution checks (e.g. from China, new domain, …) and scores metadata.
- Static File Analysis: 2000 rules consisting of WinPE, Office, and PDF checks such as Payload, Header, High-Risk scripting, Obfuscation, Artifacts, Country, and Meta.

Automated Threat Detection Rules (out of the box):

- HTTP network traffic Analysis: metadata is analyzed for possible command and control activity.
- Analysis of web proxy logs: Blue Coat Cache Flow (cacheflowelff), Cisco IronPort WSA (ciscoiportwsa), and Zscaler (zscalernss): metadata is analyzed for possible command and control activity.

Analytical results are recorded with the following information: date and time of the result, type of result, and identification of data source. Type of result corresponds to the rule that generates an alert. There are original data sources and metadata sources. The original sources are from the operational environment (Cisco, Juniper, etc.). The metadata sources are internal to NetWitness and are also identified in the user interface. It is the original sources that correspond to data source in the requirement.

The TOE uses results of the analysis to determine whether or not to send an alarm. If analysis identifies potential intrusion, malware or misuse an alarm (alert) is sent to the NetWitness Respond User Interfaces (IDS_RCT.1). 'intrusions' are anomalous events or events that merit further investigation. Authorized administrators can view, and work with alarm notifications via the Respond menu available remotely through the NetWitness Server User Interface. The IDS data (metadata, raw logs, raw packet data, and incident management data) can also be viewed from the UI by users with the Respond Administrator, Analyst and Administrator administrative roles. The data are provided in a readable format to authorized users (IDS_RDR.1). Updates to the rules can be obtained by licensed customers at the vendor's website Live. Only authorized Administrators are permitted to download these updates.

The Security Monitoring with Security Information and Event Management function is designed to satisfy the following security functional requirements:

- IDS_ANL_EXT.1: The TOE performs statistical and signature analysis on IDS data. The TOE records within each analytical result the following information: date and time of the result, type of result, and identification of the data source.
- IDS_DOR_EXT.1: The TOE is capable of protecting IDS Privacy Sensitive Data using various methods.
- IDS_RCT_EXT.1: The TOE sends an alarm to NetWitness Respond User Interfaces when an intrusion is detected.
- IDS_RDR_EXT.1: The TOE provides the authorized administrators with the capability to read Analyzer and Data Privacy Sensitive Protected data from the IDS data. The TOE provides the IDS data in a manner suitable for the user to interpret the information. The TOE prohibits all users read access to the IDS data, except those users that have been granted explicit read-access.

6.5 Security management

The ST defines the Administrator, Respond Administrator, Analyst, Operator, SOC_Manager, Malware Analyst, and DPO roles to distinguish users who can perform various security management functions. The TOE defines roles with various security management functions protected with privileges. A user that is assigned a role has the associated privileges assigned to that role and can perform those associated security management functions. Note that internally, some TOE components associate groups to users. However these groups are essentially the same as the roles previously described differing only in reference (e.g. group rather than role).

Only authorized administrators with the Administrator role can create, modify, or delete users. Only authorized users with the Administrator or DPO roles can manage the Data Privacy Protection, including management of the users authorized to access the protected data. The ability to query and manage the TSF data is restricted to the users as identified in **Table 5-3**: Management of TSF Data.

The NetWitness Server UI Interface provides the interface through which the authorized administrator manages the security functions of the TOE and the TSF data. There are no local administrative interfaces provided in the evaluated configuration for either management or installation. Only authorized administrators with the Administrator or Operator role can start/stop services and start or stop the Audit function. Users with the Operator role only have permissions to those services explicitly assigned to them by the user with the Administrator role. User accounts are per service. The Administrator should not create an account for the Respond Administrator, Analyst or Operator on services they should not have access to.

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

- FMT_MOF.1(1): The TOE restricts the ability to disable, enable, the audit function, start/stop services (Decoder, Concentrator, Broker, ESA, …) functions to the authorised identified roles. Starting and stopping the services also starts and stops the Audit collection function as each TOE appliance/service generates its own audit records.
- FMT_MOF.1(2): The TOE restricts the ability to determine the behaviour of, disable, enable, and modify the behaviour of the Data Privacy Protection to the authorised identified roles.
  .
- FMT_MTD.1: The TOE restricts the ability to query and manage the TSF data to the authorised identified roles.

- FMT_SMF.1: The TOE provides management functions identified in the text above to support the authorised identified role's ability to manage the TSF data, functions, and security audit as described in the above section.
- FMT_SMR.1: The TOE maintains the security roles: Respond Administrator, Administrator, Analyst, Operator, SOC_Manager, Malware Analyst, and Data Privacy Officer. The TSF is able to associate users with roles.

6.6 Protection of the TSF

The TOE uses TLS in FIPS-compliant mode to protect the TSF data transmitted between distributed parts of the TOE. This protection is enforced across all TOE Device components/appliances (e.g. device to device). The data that is protected across the communication channel consist of audit data, and collected data: all metadata, raw logs, raw packet data, rules, and incident management data.

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

- FPT_ITT.1: The TOE utilizes TLS to protect data transmitted between distributed parts of the TOE (Device to Device only).

6.7 TOE Access

The TOE terminates an interactive session after a time interval of user inactivity configured by an authorized administrator. Authorized administrators can configure the interval to be between min 0 and no max but default is 600 (Expiry for all sessions in minutes) via the remote administrative GUI interface. Note that zero is permitted, which disables lockout. An interactive remote session that is inactive (i.e., no commands issued and no activity from the remote client browser to the NetWitness Server UI) for the defined timeout value will be terminated.

The TOE allows user-initiated termination of the user's own interactive session by closing the browser or explicitly logging off.

Before establishing an interactive user session, the TOE displays an advisory warning message regarding unauthorised use of the TOE.

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

- FTA_SSL.3: The TOE terminates an interactive session after a time interval of user inactivity configured by an authorized administrator.
- FTA_SSL.4: The TOE allows user-initiated termination of the user's own interactive session.
- FTA_TAB.1: Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorised use of the TOE.

6.8 Trusted path/channels

The TOE requires an HTTPS connection for remote users to authenticate to the TOE from a browser that is part of the environment. To successfully establish an interactive administrative session, the administrator must be able to provide acceptable user credentials (e.g., user id and password), after which they will be able to access the GUI interface. This initial authentication action occurs over TLS in FIPS mode negotiated using the ciphers defined as valid for a TLS in FIPS mode session as described in section 6.2. Subsequently, all NetWitness interface session data transmission also occurs over TLS. The TOE uses a FIPS-validated module for SSH protected communication pathways for the transfer of file event source data from log data sources to the TOE. TLS and SSH ensure the administrative session and file transfer communication pathways are secured from disclosure and modification.

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FTP_TRP.1: The TOE provides a communication path between itself and remote administrative and authorized IT Entity users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification and disclosure.
- FTP_TRP.1: The TOE permits remote administrative and authorized IT Entity users to initiate communication via the trusted path.
- FTP_TRP.1: The TOE requires the use of the trusted path for initial user authentication, all NetWitness interface session data, and for the transfer of file event source data from log data sources to the TOE.

7 Rationale

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

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

7.1 Security Objectives Rationale

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

7.1.1 Security Objectives Rationale for the TOE and Environment

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

	T.MALICIOUS_ACTIVITY	T.UNPROTECTED_PRIVACY_DATA	T.INADVERTENT_ACTIVITY	T.MISUSE	T.TSF_COMPROMISE	T.UNACCOUNTABLE_USERS	A.AUDIT_PROTECTION	A.DATA_SOURCES	A. DEPLOY	A. MANAGE	A. PHYSICAL	A.TIME	A.TRUSTED_ADMIN	A.USER
O.ANALYZE	X		X	X										
O. PRIVACY_DATA_PROTECT		X												
O.AUDIT_GENERATION					X	X								
O.AUDIT_PROTECTION					X									
O.MANAGE	X		X	X	X									
O.PROTECTED_COMMS					X									
O.TOE_ACCESS					X	X								
OE.AUDIT_PROTECTION							X							
OE. DATA_SOURCES								X						
OE.DEPLOY									X					
OE.MANAGE										X				
OE.PHYSICAL											X			
OE.TIME												X		
OE.TRUSTED_ADMIN													X	
OE.USER														X

7.1.1.1 T.MALICIOUS_ACTIVITY

Malicious activity by an attacker may occur on the network the TOE monitors may go undetected.

This Threat is countered by ensuring that:

- O.ANALYZE: The TOE applies analytical processes and collects information to derive conclusions and send alerts about potential unauthorized/malicious activities in the monitored network.
- O.MANAGE: The TOE provide tools necessary to support the authorized administrators in their management of the security of the TOE, this includes reviewing the alarms and IDS data.

7.1.1.2 T.UNPROTECTED_PRIVACY_DATA

IDS Data that should be protected as privacy sensitive is not obscured, access restricted or retained appropriately allowing unauthorized users to view the data.

This Threat is countered by ensuring that:

- O.PRIVACY_DATA_PROTECT: The TOE protects data determined to be privacy sensitive.

7.1.1.3 T.INADVERTENT_ACTIVITY

Inadvertent activity and access by a user or a process that may occur on the network the TOE monitors may go undetected.

This Threat is countered by ensuring that:

- O.ANALYZE: The TOE applies analytical processes and collects information to derive conclusions and send alerts about potential unauthorized/malicious activities in the monitored network.
- O.MANAGE: The TOE provide tools necessary to support the authorized administrators in their management of the security of the TOE, this includes reviewing the alarms and IDS data.

7.1.1.4 T.MISUSE

Unauthorized accesses and activity indicative of misuse by a user or a process that may occur on the network the TOE monitors may go undetected.

This Threat is countered by ensuring that:

- O.ANALYZE: The TOE applies analytical processes and collects information to derive conclusions and send alerts about potential misuse in the monitored network.
- O.MANAGE: The TOE provide tools necessary to support the authorized administrators in their management of the security of the TOE, this includes reviewing the alarms and IDS data.

7.1.1.5 T.TSF_COMPROMISE

A user may cause, through an unsophisticated attack, TSF data, or security functions to be inappropriately accessed (viewed, modified, or deleted).

This Threat is countered by ensuring that:

- O.MANAGE: The TOE restricts access to the security functions and TSF data to authorized administrators.
- O. AUDIT_GENERATION: To reduce the potential of unauthorized access attempts that might go unnoticed, the TOE is expected to log security relevant events.
- O.AUDIT_PROTECTION: The TOE helps to protect the audit records by not providing interfaces to modify or delete the audit records.
- O.PROTECTED_COMMS: Administrative communications with the TOE; IT entities (log sources) and TOE device to TOE Device communications are protected from disclosure and medication.

- O.TOE_ACCESS: To reduce the potential of unauthorized access to TOE security functions and data, the TOE ensures that only authorized administrators can log in and access security management functions and TOE data.

7.1.1.6 T.UNACCOUNTABLE_USERS

Authorized users of the TOE might not be held accountable for their actions.

.

This threat is satisfied by ensuring that:

- O.AUDIT_GENERATION: To reduce the potential of security relevant actions occurring without notice, the TOE is expected to audit security relevant events and associates the user that the caused the event with the audit record. This helps to mitigate the threat by ensuring that the user that caused the security relevant events can be identified.
- O.TOE_ACCESS: This objective helps to mitigate this threat by ensuring each user is uniquely identified and authenticated.

7.1.1.7 A.AUDIT_PROTECTION

The operational environment will provide the capability to protect audit information.

This Assumption is satisfied by ensuring that:

- OE.AUDIT_PROTECTION: The operational environment provides the capability to protect audit information.

7.1.1.8 A.DATA_SOURCES

The data sources in the environment will provide complete and reliable data to the TOE.

This Assumption is satisfied by ensuring that:

- OE.DATA_SOURCES: The data sources in the environment provide complete and reliable data to the TOE.

7.1.1.9 A.DEPLOY

TOE Administrators will properly configure the network in the TOE operational environment and configure adequate network capacity for the deployed TOE components.

This assumption is countered by ensuring that:

- OE.DEPLOY: The TOE Administrators will properly configure the network in the TOE operational environment and configure adequate network capacity for the deployed TOE components.

7.1.1.10 A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

This assumption is countered by ensuring that:

- OE.MANAGE: Personnel working as authorized administrators shall be carefully selected and trained for proper operation of the TSF.

7.1.1.11 A.PHYSICAL

The TOE hardware and software critical to the security policy enforcement will be located within controlled access facilities which will prevent unauthorized physical access.

This assumption is countered by ensuring that:

- OE.PHYSICAL The TOE hardware and software critical to the security policy enforcement will be located within controlled access facilities which will prevent unauthorized physical access.

7.1.1.12 A.TIME

The environment will provide reliable time sources for use by the TOE.

This assumption is countered by ensuring that:

- OE.TIME: The environment must provide a time source for use by the TOE

7.1.1.13 A.TRUSTED_ADMIN

TOE Administrators will follow and apply all administrator guidance in a trusted manner.

This assumption is countered by ensuring that:

- OE.TRUSTED_ADMIN: TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

7.1.1.14 A.USER

Users will protect their authentication data.

This assumption is countered by ensuring that:

- OE.USER: Users must ensure that their authentication data is held securely and not disclosed to unauthorized persons.

7.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note Table 7-1 indicates the requirements that effectively satisfy the individual objectives.

7.2.1 Security Functional Requirements Rationale

All of the 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.ANALYZE	O. PRIVACY_DATA_PROTECT	O.AUDIT_PROTECTION	O.AUDIT_GENERATION	O.MANAGE	O.PROTECTED_ COMMS	O.TOE_ACCESS
FAU_GEN.1				X			
FAU_GEN.2				X			
FAU_SAR.1					X		
FAU_SAR.2					X		
FAU_STG.1			X				
FCS_SSH_EXT.1						X	
FCS_TLS_EXT.1						X	
FIA_AFL.1							X
FIA_ATD.1							X
FIA_UAU.1							X
FIA_UAU.5							X
FIA_UID.1							X
FMT_MOF.1(1)					X		

	O.ANALYZE	O. PRIVACY_DATA_PROTECT	O.AUDIT_PROTECTION	O.AUDIT_GENERATION	O.MANAGE	O.PROTECTED_ COMMS	O.TOE_ACCESS
FMT_MOF.1(2)					X		
FMT_MTD.1					X		
FMT_SMF.1					X		
FMT_SMR.1					X		
FPT_ITT.1						X	
FTA_SSL.3							X
FTA_SSL.4							X
FTA_TAB.1							X
FTP_TRP.1						X	
IDS_ANL_EXT.1	X						
IDS_DOR_EXT.1		X					
IDS_RCT_EXT.1	X						
IDS_RDR_EXT.1*					X		

Table 7-1 Objective to Requirement Correspondence

7.2.1.1 O.ANALYZE

The TOE will apply analytical processes and information to derive conclusions about potential unauthorized/malicious intrusions and send appropriate alerts.

This TOE Security Objective is satisfied by ensuring that:

- IDS_ANL_EXT.1: The TOE performs statistical and signature analysis functions on IDS data. The TOE records each analytical result and includes at least the following information in the record: Date and time of the result, type of result, identification of data source.
- IDS_RCT_EXT.1: The TOE sends an alarm to NetWitness Respond User Interfaces when an intrusion is detected.

7.2.1.2 O.PRIVACY_DATA_PROTECT

The TOE will protect data determined to be privacy sensitive.

This TOE Security Objective is satisfied by ensuring that:

- IDS_DOR_EXT.1: The TOE provides the following functions to protect the privacy sensitive data: data obfuscation; functions to control the persistence of data; and functions to prevent the transfer of protected IDS data.

7.2.1.3 O.AUDIT_GENERATION

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

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: The TOE is required to provide a set of events that it is capable of recording. Among these events the TOE is able to audit must be security relevant events occurring within the TOE. This requirement also defines the information that must be recorded for each auditable event.

- FAU_GEN.2: The TOE is required to associate a user identity with the auditable events being recorded.

7.2.1.4 O.AUDIT_PROTECTION

The TOE will provide the capability for protection of the audit information from unauthorized users via the TOE interfaces.

This TOE Security Objective is satisfied by ensuring that:

- FAU_STG.1: The TOE is required to protect the stored audit records from unauthorized modification or deletion.

7.2.1.5 O.MANAGE

The TOE will provide all the functions necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions from unauthorized use.

This TOE Security Objective is satisfied by ensuring that:

- FAU_SAR.1: The TOE is required to provide authorized administrators with the capability to read all audit information from the audit records. The TOE is required to provide the audit records in a manner suitable for the user to interpret the information.
- FAU_SAR.2: The TOE is required to prohibit all users read access to the audit records, except those users that have been granted explicit read-access.
- FMT_MOF.1(1): The TOE is required to restrict the ability to enable and disable the security functions to authorized administrators.
- FMT_MOF.1(2): The TOE is required to restrict the ability to manage the Data Privacy Protection to Only authorized administrators.
- FMT_MTD.1: The TOE is required to restrict to authorized administrators the ability to manipulate TOE data used to enforce the TOE security functions.
- FMT_SMF.1: The TOE is required to provide at least the identified management functions for use by the authorized administrators.
- FMT_SMR.1: The TOE is required to establish, maintain and enforce authorized administrator roles.
- IDS_RDR_EXT.1: The TOE is required to provide the authorized administrators with the capability to read all metadata, raw logs, raw packet data, rules, and incident management data from the IDS data. The TOE is required to provide the authorized administrators with the capability to read the original and obfuscated Data Privacy Sensitive Protected data from the IDS data. The TOE provides the data in a manner suitable for the user to interpret the information; and prohibits all users read access to the IDS data, except those users that have been granted explicit read-access.

7.2.1.6 O.PROTECTED_COMMS

The TOE will provide protected communication channels for remote administrators, IT entities and for TOE device to TOE device communication channels.

This TOE Security Objective is satisfied by ensuring that:

- FCS_SSH_EXT.1: The TOE implements SSH to protect log data being sent to the TOE from authorized IT entities (log sources).
- FCS_TLS_EXT.1: The TOE is required to implement TLS to protect applicable network communication channels.
- FPT_ITT.1: The TOE is required to protect communications from disclosure and detect the modification of those communications when it is transmitted between distributed parts of the TOE.

- FTP_TRP.1: The TOE is required to protect communication between itself and its remote administrative users from disclosure and detect the modification of those communications. The TOE is required to use HTTP over TLS to provide these protections.

7.2.1.7 O.TOE_ACCESS

The TOE will provide mechanisms that control a user's logical access to the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FIA_AFL.1: The TOE must detect when an administrator configurable positive integer within the configured timeframe of unsuccessful authentication attempts occur related to NetWitness UI user authentication. When the defined number of unsuccessful authentication attempts has been met, the TSF shall lock account for a specified time period as configured by authorized administrator.
- FIA_ATD.1: The TOE maintains the following list of security attributes belonging to individual human users: Username, password, role. The attributes of users, are used by the TOE to determine a user's identity and role memberships and enforce what type of access the user has to the TOE.
- FIA_UAU.1: The TOE is required to ensure that users must be authenticated in order to access functions, other than those specifically identified (view the warning banner and acknowledging the end-user license).
- FIA_UAU.5: The TOE provides password-based authentication for administrative users and SSH public-key based authentication for authorized IT entities (log sources) sending log data to the TOE. Administrative users must successfully authenticate by providing a valid username and password in order to access functions, other than those specifically identified (view the warning banner and acknowledging the end-user license). IT entities must successfully authenticate using public key-based authentication prior to sending any log data to the TOE.
- FIA_UID.1: The TOE is required to ensure that users must be identified in order to access functions of the TOE other than those specifically identified (view the warning banner and acknowledging the end-user license).
- FTA_SSL.3: The TOE will terminate an interactive session after a time period configured by an authorized administrator.
- FTA_SSL.4: The TOE must allow user-initiated termination of the user's own interactive session.
- FTA_TAB.1: Before establishing a user session, the TOE shall display an advisory warning message regarding unauthorised use of the TOE.

7.2.2 Security Assurance Requirements Rationale

The security assurance requirements for the TOE are the EAL 2 augmented with ALC_FLR.1 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

EAL 2 augmented with ALC_FLR.1 was selected as the assurance level because the TOE is a commercial product whose users require a low to moderate degree of independently assured security. ALC_FLR.1 was selected to augment EAL2 assurance requirements in order to ensure that identified flaws are addressed. The TOE is targeted at a relatively benign environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have little attack potential. As such, EAL 2 augmented with ALC_FLR.1 is appropriate to provide the assurance necessary to counter the limited potential for attack.

7.3 Requirement Dependency Rationale

The following table demonstrates the dependencies among the claimed security requirements. It shows that all dependencies are satisfied. Therefore the requirements work together to accomplish the overall objectives defined for the TOE.

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	See TimeStamp Note Below.

FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN.1 and FIA_UID.1
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FCS_SSH_EXT.1	None	None
FCS_TLS_EXT.1	None	None
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1	None	None
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UAU.5	None	None
FIA_UID.1	None	None
FMT_MOF.1(1), FMT_MOF.1(2)	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1	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.1
FPT_ITT.1	None	None
FTA_SSL.3	None	None
FTA_SSL.4	None	None
FTA_TAB.1	None	None
FTP_TRP.1	None	None
IDS_ANL_EXT.1	None	None
IDS_DOR_EXT.1	IDS_ANL_EXT.1	IDS_ANL_EXT.1
IDS_RCT_EXT.1	IDS_ANL_EXT.1	IDS_ANL_EXT.1
IDS_RDR_EXT.1*	IDS_ANL_EXT.1	IDS_ANL_EXT.1

Timestamp Note: The TOE is not a physical device and operates as an application within a process provided by the environment. Thus, the environment is providing resources for the TOE. The environmental objective OE.TIME requires that the TOE's environment provide a reliable timestamp which the TOE can use as needed (e.g., within audit records). Thus, the functionality reflected in the dependency of FAU_GEN.1 upon FPT_STM.1 is available to the TOE from the environment.

7.4 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 7-2 Security Requirements to Security Functions** Mapping demonstrates the relationship between security requirements and security functions.

	Security audit	**Cryptographic support**	**Identification and authentication**	**Security Monitoring with SIEM**	**Security management**	**Protection of the TSF**	**TOE Access**	**Trusted path/channels**
FAU_GEN.1	X							
FAU_GEN.2	X							

FAU_SAR.1	X							
FAU_SAR.2	X							
FAU_STG.1	X							
FCS_SSH_EXT.1		X						
FCS_TLS_EXT.1		X						
FIA_AFL.1			X					
FIA_ATD.1			X					
FIA_UAU.1			X					
FIA_UAU.5			X					
FIA_UID.1			X					
IDS_ANL_EXT.1				X				
IDS_DOR_EXT.1				X				
IDS_RCT_EXT.1				X				
IDS_RDR_EXT.1*				X				
FMT_MOF.1(1)					X			
FMT_MOF.1(2)					X			
FMT_MTD.1					X			
FMT_SMF.1					X			
FMT_SMR.1					X			
FPT_ITT.1						X		
FTA_SSL.3							X	
FTA_SSL.4							X	
FTA_TAB.1							X	
FTP_TRP.1								X

Table 7-2 Security Requirements to Security Functions Mapping