Ivanti

Security Controls 2022.2

Security Target

Evaluation Assurance Level (EAL): EAL 2+
Document Version: 0.11

Prepared for:

ivanti

Prepared by:

Corsec

Ivanti
10377 South Jordan Gateway
Suite 110
South Jordan, Utah 84095
United States of America

Phone: +1 888 253 6201
www.ivanti.com

Corsec Security, Inc.
12600 Fair Lakes Circle, Suite 210
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
www.corsec.com

Table of Contents

List of Figures

List of Tables

1. Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), and the ST organization. The Target of Evaluation (TOE) is the Ivanti Security Controls 2022.2 (version 9.5.9293) and will hereafter be referred to as the TOE throughout this document. The TOE is an integrated software solution providing patch management, asset inventory, IT[1] administration, and reporting functionality. These functions are supported through the Security Controls application.

1.1 Purpose

This ST is divided into nine sections, as follows:

- Introduction (Section 1) – Provides a brief summary of the ST contents and describes the organization of other sections within this document. It also provides an overview of the TOE security functionality and describes the physical and logical scope for the TOE, as well as the ST and TOE references.
- Conformance Claims (Section 2) – Provides the identification of any Common Criteria (CC), Protection Profile (PP), and Evaluation Assurance Level (EAL) package claims. It also identifies whether the ST contains extended security requirements.
- Security Problem (Section 3) – Describes the threats, organizational security policies, and assumptions that pertain to the TOE and its environment.
- Security Objectives (Section 4) – Identifies the security objectives that are satisfied by the TOE and its environment.
- Extended Components (Section 5) – Identifies new components (extended Security Functional Requirements (SFRs) and extended Security Assurance Requirements (SARs)) that are not included in CC Part 2 or CC Part 3.
- Security Requirements (Section 6) – Presents the SFRs and SARs met by the TOE.
- TOE Security Specification (Section 7) – Describes the security functions provided by the TOE that satisfy the security functional requirements and objectives.
- Rationale (Section 8) – Presents the rationale for the security objectives, requirements, and SFR dependencies as to their consistency, completeness, and suitability.
- Acronyms (Section 9) – Defines the acronyms and terminology used within this ST.

[1] IT – Information Technology

1.2 Security Target and TOE References

Table 1 below shows the ST and TOE references.

Table 1 – ST and TOE References

ST Title	*Ivanti Security Controls 2022.2 Security Target*
ST Version	Version 0.11
ST Author	Corsec Security, Inc.
ST Publication Date	November 7, 2023
TOE Reference	Ivanti Security Controls 2022.2 (version 9.5.9293.0)

1.3 Product Overview

Security Controls provides patch management, asset inventory, scripts for IT management and Information Assurance Vulnerability Alert (IAVA) reporting. These functions combine to provide a centralized and consistent IT management solution that supports efforts to keep all machines up-to-date and protected from vulnerabilities.

Patch management allows for all Windows-based machines and VMware ESXi hypervisors in the network to be scanned. Once scanned, a report detailing the un-patched software vulnerabilities on the network is generated. Based on the scan results, schedules may be created to download and deploy missing patches. E-mail alerts providing patch availability, deployment status, and scan results may be sent to IT personnel to help streamline processes and ensure each machine is up-to-date. Patch management may be performed with or without agents, providing flexibility and minimizing management overhead.

Asset inventory allows for the tracking of hardware, software, and virtual assets. A scan is performed that provides details on installed software, virtual infrastructure, or hardware configuration. Once a scan is complete, reports categorizing information may be generated. Hardware and software specifications may be categorized and collected over time to more effectively manage IT resources.

IT scripts are included with Security Controls. The Windows PowerShell based IT scripts are used to perform a variety of basic administrative tasks. The scripts may be run on a single machine or an established machine group. The IT scripts allow for automating repetitive tasks across a large number of machines. To ensure security, the provided IT scripts are all digitally signed by Ivanti. The following IT script functions are supported:

- Execute scripts against target machines
- Execute scripts from the console
- Create PowerShell templates

PowerShell Templates specify how an IT Script is to be executed. The template defines the script to be executed, parameters to be used in the script, and the number of concurrent machines where the script may be run. Templates may be executed immediately or scheduled to run at a later point in time.

Security Controls supports IAVA specific reporting functionality. The IAVA reports provide a cross-reference between IAVAs and CVEs[2]. CVEs are public listing of report vulnerabilities that feed the U.S. National Vulnerability Database. IAVAs are announcements from U.S. Cyber Command that are based on published CVEs. These IAVAs form the basis of STIG[3] compliance. Security Controls' IAVA reports help administrators better understand which machines have vulnerabilities and establish a plan to address them.

The REST API provides a simple RESTful interface with lightweight JSON-formatted responses that enables TOE users to read and write data to/from the program. The feature allows TOE users to automate many day-to-day operations, saving considerable time and effort. The REST API allows TOE users to fully integrate Ivanti Security Controls into your orchestration and automation systems.

1.4 TOE Overview

The TOE Overview summarizes the usage and major security features of the TOE. The TOE Overview provides a context for the TOE evaluation by identifying the TOE type, describing the TOE, and defining the specific evaluated configuration.

The TOE is an integrated software solution providing patch management, asset inventory, IT administration, and reporting functionality. Based on the provided functionality, the TOE type is identified as "Other Devices and Systems". The TOE is a Windows-based software solution that is comprised of the following components:

- Security Controls Console
- Security Controls Agent
- Security Controls Deployment Tool Chain

The Security Controls Console is the hub of all scanning, deployment, scheduling and reporting tasks. A TOE user must have Windows login credentials with administrative access to the host OS. Functions available are based on the role assigned. The Security Controls Console supports both agent-based and agentless endpoint administration. An agentless configuration is where no persistent software is required on the managed endpoint. Agentless operations are all executed and controlled through the Security Controls Console. Agentless scans are performed to determine the health of machines on the network. Other agentless operations include patch deployment and remote IT Script execution.

The Security Controls Agent is installed on a managed endpoint to support policy-based administration. The Security Controls Agent operates autonomously according to a policy provided by the Security Controls Console, that is created by a TOE user with the Administrator role. This option provides flexibility to overcome network topology challenges such as interrupted connectivity. A policy is a set of operating rules defining what a Security Controls Agent will do. The policy is used by the Security Controls Agent to determine the patch health of the host machine. Based on the health, patches are deployed according to the rules in the policy. Security Controls Agents may get patch updates directly from the Security Controls Console, from a Distribution Server, or from vendor web sites.

Agentless systems are managed remotely by the Security Controls Console. Patch deployment on agentless systems is handled through the Security Controls Deployment Tool Chain. The Security Controls Deployment Tool

[2] CVE – Common Vulnerability Exchange
[3] STIG – Security Technical Implementation Guides

Chain is pushed by the Security Controls Console to the specified agentless machines. This tool facilitates patch execution, scheduling, and status reporting. To perform scheduled operations, the Security Controls Deployment Tool Chain includes the Security Controls Scheduler service. The Security Controls Scheduler can be remotely managed from the Security Controls Console using the Scheduled Tasks Manager application. The Security Controls Scheduler will be installed on demand when a scheduled operation is requested.

The TOE software components can be deployed in a variety of configurations. The configuration for this evaluation is provided in Figure 1 below. The Security Controls Console is the hub of all IT management activity. The Security Controls Console synchronizes its patch repository with a Distribution Server, which is a part of the Security Controls Console machine operational environment. One or more managed endpoints, or Security Controls Agents, get patch information from the Distribution Server or from defined websites on the Web Server. Policies are retrieved from the Security Controls Console by the Security Controls Agents. Scan results and deployment confirmations are sent from the Security Controls Agents back to the Security Controls Console. Schedules are created by the Security Controls Console and executed by the Security Controls Scheduler, which resides on the Security Controls Console and each managed endpoint machine. Electronic mail (e-mail) transmissions are sent from the Security Controls Console to the SMTP[4] Server. All information is transmitted securely across the corporate network. The software and hardware used to run the TOE are not included in the TOE boundary.

1.4.1 Description of the TOE Components

The following sections describe the technologies and concepts related to the TOE.

1.4.1.1 Security Controls Console

The Security Controls Console is the server component of the TOE. The Security Controls Console is a Windows-based application that is installed on Windows Server 2019. The Security Controls Console is composed of the Security Controls Console GUI[5], services and Patch Engine components. The GUI provides a front-end interface to users. The core patch scanning and deployment logic is implemented in the Patch Engine. The Security Controls Console also contains a Windows Service host for various Security Controls Console services including Results Import, Agent Support/STS[6], Deploy Monitor, Data Sync, Scheduler, IT Script Engine, REST API, and Hypervisor Patch. The user must have the Administrator role assigned to their account to have access to all of these functions.

Permissions are enforced by the host OS. Role access within the application is enforced via licensing. At execution, the application checks the user account's permissions and then modifies the active license on the fly in order to remove the user's ability to perform actions for which the user is not authorized.

The Security Controls Console stores encrypted administrative credentials (the encryption is performed by the Windows OS FIPS 140-2 Cryptographic Service Provider, which is outside of the boundary of this evaluation), configuration information, patch deploy audit, and past scan data for the other Windows-based workstations and servers on the monitored network in the attached Microsoft SQL[7] Server database. It is also able to automatically generate reports, export them to a PDF[8], and email them out to a configurable set of email addresses (via a configurable external mail server).

[4] SMTP – Simple Mail Transfer Protocol
[5] GUI – Graphical User Interface
[6] STS – Security Token Service
[7] SQL – Structured Query Language
[8] PDF – Portable Document Format

Patch Management is the core feature of the Security Controls Console. Determining what patches are missing can be performed in an agentless manner, without any additional software or configuration on the target machines. Once an assessment has been performed, missing patches are downloaded and pushed as packages for installation to the target machines. As part of the deployment package, a patch deployment script is generated and pushed to the target endpoint. Once all components of the deployment package are pushed to the target, the deployment script is scheduled for execution via the Security Controls Scheduler.

Distribution Servers can be used in an agent-based or agentless scenario to reduce the impact of patch deployment on the network. A Distribution Server is a local cache of patches available for installation. Patches are stored on a configured Distribution Server (a server with a network file share). The Distribution Server can be the Security Controls Console machine's patch repository or any other network file share. The Security Controls Console synchronizes its patch repository with the Distribution Server (or servers, if more than one is configured). Once a Distribution Server is synchronized, patch deployment targets or Security Controls Agents can get the patches from their configured Distribution Server. (For the purposes of this evaluation, the Distribution Server is located on the same machine as the Security Controls Console.)

The Security Controls Console provides the ability to execute IT Scripts to automate repetitive IT administration tasks. IT Scripts are digitally signed Microsoft PowerShell scripts with credential security and output enhancements.

The Hypervisor Patch component works with the vSphere API[9] to perform several functions on standalone ESXi hosts, ESXi hosts managed with vCenter Servers, and the ESXi hosts guest Virtual Machines:

- View basic configuration information about the vCenter Servers and the ESXi hypervisors
- Perform a patch scan of the ESXi hypervisors
- View the security bulletins that have been installed on the ESXi hypervisors
- View the security bulletins that have not been deployed on the ESXi hypervisors
- Deploy any missing security bulletins to the ESXi hypervisors
- Power on and off the virtual machines that reside on the ESXi hypervisors
- Add the virtual machines and virtual machine templates to a new or existing machine group

Machine groups are reusable collections of machines or discovery parameters that can be used within an agentless scan. Machine Groups may contain any number of machines, including the Security Controls Console itself. The Machine Group dialog is used to view and configure information about the Machine Group and individual machines within the group. Machines may be added to a Machine Group by name, domain, IP[10] address, IP address range, or Organizational Unit (OU). If a domain is added to the machine group, all machines in the domain at the time of the agentless scan are considered part of the machine group. If an OU is added to the machine group, all machines in the OU, including those in child OUs of the OU being added, are, at the time of the agentless scan considered part of the machine group. Both physical and virtual machines may be added to the same Machine Group.

There are several reports that may be run from the Security Controls Console. Available reports are determined by licensing associated with the user's credentials provided upon the TOE's confirmation of the user's

[9] API – Application Programming Interface
[10] IP – Internet Protocol

authentication. Reports provide detailed information on patch status, power status, and asset inventory. The Government Edition additionally provides multiple IAVA reports:

- Deployment Percentage by Patch (IAVA) – percentage of machines that have each patch installed
- Detailed Summary (IAVA) – detailed scan summary
- Machine Status by Patch Count (IAVA) –listing of machines ordered by the number of missing patches
- Patch Status Detail (IAVA) – detailed patch status information

Agent communication, results rollup, and deployment status are provided over a secured channel between TOE components. Security Controls Console services are exposed as HTTP/HTTPS[11] web services. The Patch Scan Engine and Distribution Server synchronization feature leverages the SMB[12] protocol implemented by the target OSs. Asset inventory scans also leverage SMB in addition to the WMI[13] protocol. The Security Controls Console is also capable of sending automated email messages via the SMTP[14] protocol.

The Security Controls REST API is exposed over an HTTPS interface as web services. API authentication authorizes users to access application functionality in a similar way to the user interface. The API surface allows users to initiate patching operations, observe, and react to patching states in the system.

1.4.1.2 Security Controls Agent

The Security Controls Agent is an agent service that is installed on a physical or virtual machine connected to the network. Actions such as patch scans, asset scans, and patch deployments are defined by an Agent Policy. These policies are configured on the Security Controls Console and retrieved by the Security Controls Agent over a secured channel.

The agent-based configuration is an autonomous service installed on selected target machines. This configuration is useful in organizations with many remote users or distributed networks. In this configuration, the agent machine performs patch management functions and communicates results back to the Security Controls Console. This communication is performed over a secure channel that leverages the Windows OS Cryptographic Service Provider for all cryptographic operations. The Windows OS FIPS 140-2 validated Cryptographic Service Provider is outside the boundary of this evaluation.

1.4.1.3 Security Controls Deployment Tool Chain

The Security Controls Deployment Tool Chain allows agentless machine targets to patch safely. The Security Controls Deployment Tool Chain applies patches, sends progress status, and manages reboot operations. The Security Controls Deployment Tool Chain is pushed by the Security Controls Console to deploy patches on each target machine. The Security Controls Scheduler is a piece of the Security Controls Deployment Tool Chain that schedules patch deployment and allows staging of future deployments. All executables and instructions are digitally signed by the Security Controls Console Windows OS Cryptographic Service Provider prior to being sent to the target machine. The Windows OS Cryptographic Service Provider of the target machine authenticates the digital signature of all files before performing any operations. The Windows OS FIPS 140-2 validated Cryptographic Service Provider is outside the boundary of this evaluation.

[11] HTTP(S) – Hypertext Transfer Protocol/Hypertext Transfer Protocol (Secure)
[12] SMB – Server Message Block
[13] WMI – Windows Management Instrumentation
[14] SMTP – Simple Mail Transfer Protocol

The Security Controls Scheduler service allows remote scheduling and control of patch deployment operations. Communications to the scheduler service are secured using a secure channel.

Agentless and agent-based configurations may be used together ensuring networks are effectively managed while remote users' applications are secure and up-to-date on patches.

1.4.2 TOE Environment

The Security Controls Console component of the TOE is deployed on a general-purpose server or workstation running a supported version of Microsoft Windows with a supported version of the Microsoft .NET Framework. The Security Controls Console leverages the Windows Event Logs and Windows Event Viewer provided by the OS. The agent-based component, Security Controls Agent, of the TOE is deployed on a server or workstation running a supported version of Microsoft Windows. The agentless component, Security Controls Deployment Tool Chain, of the TOE is to be deployed on a server or workstation running a supported version of Microsoft Windows.

All cryptographic functionality is provided by the Windows OS FIPS 140-2 certified Cryptographic Serve Provider on the Security Controls Agent machine. All data associated with the TOE is stored in a Microsoft SQL Server database. For a list of specific build numbers to the Windows OS that Microsoft lists as FIPS 140-2 validated, please refer to their documentation located here: https://docs.microsoft.com/en-us/windows/security/threat-protection/fips-140-validation.

The requirements for the TOE hosts and external servers are listed in Table 2 below.

Table 2 – TOE Host Requirements

Category	Requirements
Security Controls Agent Host	Microsoft Windows 10
Security Controls Deployment Tool Chain Host	Microsoft Windows 10
Security Controls Console Host	Microsoft Windows Server 2019 Microsoft SQL Server 2019 Microsoft Visual C++ Redistributable 2015-2019 .NET Framework 4.8 or later Windows file share for the Distribution Server. Refer to the "Why Use a Distribution Server" section of the Ivanti Security Controls Help Guide.

Table 3 – External Server Requirements

Category	Requirements
SMTP Server	A TLS-enabled SMTP server.
Web Server	This is an external server that requires an internet connection to access.

For host hardware requirements, refer to the "System Requirements" section of the *Ivanti Security Controls Installation Guide*.

The TOE utilizes the network to access the different hosts, SMTP server, and Web Server. All network switches and connections are available in the TOE environment.

An SMTP server is utilized for e-mail messaging. A TOE user establishes a list of recipients to receive e-mail messages regarding patch status and scan results. These messages are sent from the Security Controls Console to the SMTP server.

The TOE environment contains an external Web Server. The Web Server is used in license key validation during installation of the Security Controls application and to gather input from current installation of the TOE to assess functionality being used. In addition, the TOE accesses the Web Server to download end-user application patches from 3rd-party vendors during patch deployment.

1.5 TOE Description

This section primarily addresses the physical and logical components of the TOE that are included in the evaluation.

1.5.1 Physical Scope

Figure 1 illustrates the physical scope and the physical boundary of the overall solution and ties together all of the components of the TOE.

The software-only TOE is a patch and IT management product that is installed on Microsoft Windows. The IvantiSecurityControls_2022.2.exe file for the TOE software is delivered over HTTPS from the Ivanti website. The TOE boundary includes the Security Controls 2022.2 (version 9.5.9293) software components described in section 1.4.1, which are delivered in the single .exe file mentioned above, and excludes the underlying OS, hardware platform, and communications infrastructure.

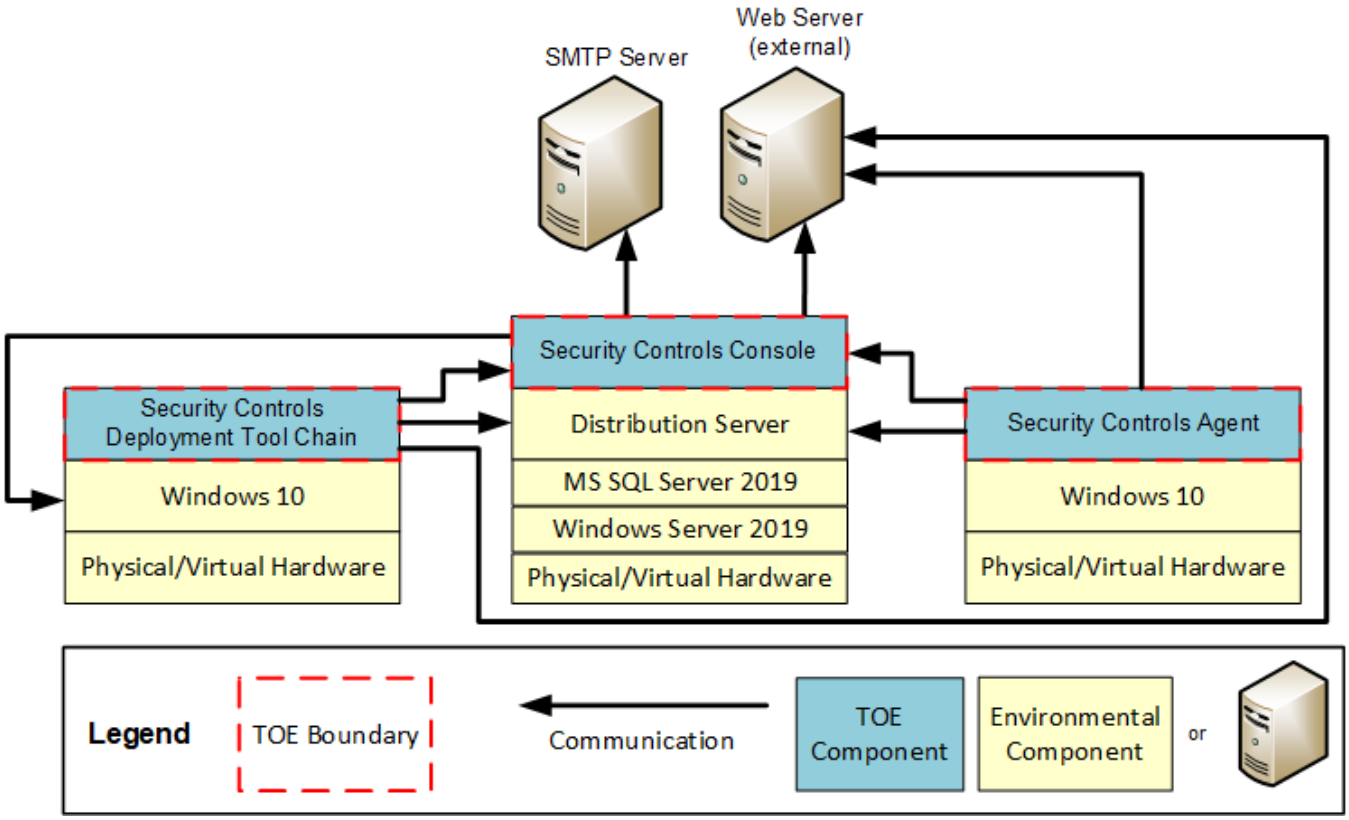

Figure 1 – Physical TOE Boundary

1.5.1.1 Guidance Documentation

The following guides are required reading and part of the TOE:

- *Welcome to Ivanti Security Controls* – A help guide that contains information about the TOE, how to install it, and how to manage it. This Microsoft Compiled HTML Help file can be accessed at https://application.ivanti.com/isec/v9.5/help/9293/ISeCHelp.chm. Users must right-click on the downloaded file, click **Properties** and check the **Unblock** box to use the file.
- *Welcome to the Ivanti Security Controls REST API* – A help guide that contains information about the TOE's REST API usage. This Microsoft Compiled HTML Help file can be accessed at https://application.ivanti.com/isec/v9.5/help/9293/ISeC-API.chm. Users must right-click on the downloaded file, click **Properties** and check the **Unblock** box to use the file.
- *Ivanti Security Controls 2022.2 Guidance Documentation Supplement* – A PDF file that contains information about the evaluated configuration that is not specified in the above documents.

1.5.2 Logical Scope

The logical boundary of the TOE will be broken down into the following security classes, which are further described in sections 6 and 7 of this ST. The logical scope also provides the description of the security features of the TOE. The security functional requirements implemented by the TOE are usefully grouped under the following Security Function Classes:

- Security Audit
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF[15]
- Resource Utilization
- Data Collection

1.5.2.1 Security Audit

The TOE generates audit records each time a machine is scanned, a patch is applied, and a security violation is discovered. It also allows an authorized user to review the audit records. Audit records are also generated on startup and shutdown of the application, but these audit events are stored in the Windows Event Logs. An authorized user may view the Windows Event Logs through the Windows Event Viewer. Functionality associated with the Windows Event Logs is outside the scope of this evaluation and will not be covered in this Security Target.

1.5.2.2 User Data Protection

The TOE implements an Access Control Security Functional Policy (SFP), which mediates access to the TOE's security functions. The TOE also implements an information flow control SFP, called Protect SFP, which mediates access to machine-scanning functionality and patch-deployment functionality.

The TOE imports end user application[16] patch binaries from vendor websites. When applicable, certificate validation is performed before the information is allowed into the TOE. This validation uses the Windows OS FIPS 140-2 validated Cryptographic Service Provider. If the binaries cannot be validated, then they are not downloaded into the TOE.

The TOE exports end user application patch binaries to the Distribution Server. An authenticated user with appropriate access identifies end user application patch binaries. The files are exported from the TOE to the specified Distribution Server where they will be retrieved by the agentless and agent-based target machines during the patch deployment process.

1.5.2.3 Identification and Authentication

The TOE maintains the unique Windows user account identifier (ID) and assigns a role for each user for access control and auditing purposes.

1.5.2.4 Security Management

The TOE provides following security management functions, upon which Access Control and Protect Control are enforced:

- Management of security functions behavior
- Management of security attributes
- Management of TSF data

[15] TSF -- TOE Security Function

[16] Patch binaries considered as user data are those patch binaries used to patch end user applications such as ERP components, Data Bases, Microsoft Office products, Adobe Acrobat, and other applications installed on a target machine. These patch binaries do not include the patches used for the Windows OS or Security Controls application.

The TOE authorizes access to security functions and attributes based on the user's Windows OS login credentials. (The Windows OS authentication functionality is not a part of this evaluation and will not be covered in this ST.) These credentials are used to identify the user's role and what information is available to be created, modified, and deleted. For further details on roles associated with administration rights, refer to Table 12 below.

1.5.2.5 Protection of the TSF

Ivanti executables[17], patch data[18], and configuration data are protected from modification while being transmitted between separate parts of the TOE. Ivanti executables, TOE patch data, and configuration data are only distributed if the integrity of the data is determined to be valid. The integrity of TOE software is verified upon execution of a TOE component. The TOE component will only allow itself to execute or be executed by appropriately verified software. Integrity checking is based on digital signatures attached to Ivanti executable code, TOE patch data code, and configuration data. The cryptographic functionality related to generating and verifying digital signatures takes place in the Windows OS using a FIPS 140-2 validated Cryptographic Service Provider. (The Windows OS FIPS 140-2 validated Cryptographic Service Provider is outside the scope of this evaluation and will not be discussed further in this Security Target.)

1.5.2.6 Resource Utilization

The TOE implements resource utilization mechanisms when performing patch scans, asset scans, and patch deployments. These engines are multithreaded, which means they may run multiple tasks at one time. When called, a number is passed defining how many threads (at maximum) are to be utilized simultaneously. Security Controls can attempt to scan up to 64 machines per CPU core simultaneously, with the default being 8 per CPU core. For example, on a 16-core system up to 128 machines can be scanned simultaneously by default.

1.5.2.7 Data Collection

The TOE utilizes patch and asset scans to collect data about machines within the network. Patch scans provide updated detail on the health of a machine or machines in a machine group. Asset scans provide information about the hardware and software of physical and virtual machines. Scans on a machine or machine group are executed by an authorized user from the Security Controls Console GUI. If allowed in the agent policy, scans can also be executed by an authorized user on the local machine running the Security Controls Agent. This scan data is collected from the specified target machines, sent to the SQL database, and viewed from the Security Controls Console. Only authorized users may leverage this information to analyze the state of the network and determine key IT tasks to be performed.

1.5.3 Product Physical/Logical Features and Functionality not included in the TOE

The following features and functionality are included in the TOE by default and cannot be disabled. These features have not been tested against Common Criteria standards, and usage of these features is considered to be outside the scope of the evaluation. These features can be accessed only by a trusted administrator, who is assumed to be competent, non-hostile, appropriately trained, and follows all guidance.

Features/Functionality that are not part of the evaluated configuration of the TOE are:

[17] Ivanti executables include code used for installation of agentless and agent-based target machines.

[18] Patch data represents files used to patch components of the Security Controls application, and files used to patch the host Windows OS

- Third-party application control
- PowerShell ITScripts customization
- Security Controls Cloud features
- Power Management features
- Importing CVEs
- Ivanti Application Control

The following user roles are excluded from the evaluated configuration:

- Application Control report only
- Patch and Application Control deploy and report

2. Conformance Claims

This section and Table 4 provide the identification for any CC, PP, and EAL package conformance claims. Rationale is provided for any extensions or augmentations to the conformance claims. Rationale for CC and PP conformance claims can be found in Section 8.1.

Table 4 – CC and PP Conformance

Common Criteria (CC) Identification and Conformance	Common Criteria for Information Technology Security Evaluation, Version 3.1, Release 5, April 2017; CC Part 2 extended; CC Part 3 conformant;
PP Claim	None
Package Claim	EAL 2+ augmented (Augmented with Flaw Remediation (ALC_FLR.2))

3. Security Problem

This section describes the security aspects of the environment in which the TOE will be used and the manner in which the TOE is expected to be employed. It provides the statement of the TOE security environment, which identifies and explains all:

- Known and presumed threats countered by either the TOE or by the security environment
- Organizational security policies with which the TOE must comply
- Assumptions about the secure usage of the TOE, including physical, personnel, and connectivity aspects

3.1 Threats to Security

This section identifies the threats to the IT assets against which protection is required by the TOE or by the security environment. The threat agents are divided into two categories:

- Attackers who are not TOE users: They have public knowledge of how the TOE operates and are assumed to possess a low skill level, limited resources to alter TOE configuration settings or parameters, and no physical access to the TOE.
- TOE users: They have extensive knowledge of how the TOE operates and are assumed to possess a high skill level, moderate resources to alter TOE configuration settings or parameters, and physical access to the TOE. (TOE users are not assumed to be willfully hostile to the TOE.)

Both are assumed to have a low level of motivation. The IT assets requiring protection are the TSF[19] and user data saved on or transitioning through the TOE and the hosts on the protected network. Removal, diminution, and mitigation of the threats are through the objectives identified in Section 4. Table 5 below lists the applicable threats.

Table 5 – Threats

Name	Description
T.AUDACC	Persons may not be accountable for an action that they conduct because there is not an audit event added to the audit trail, thus allowing an attacker to escape detection.
T.BADSTATE	An attacker may exploit vulnerabilities in monitored IT entities that reach an insecure state without the network administrators becoming aware.
T.INT_ATK	An attacker may exploit internal weaknesses in the TOE implementation to gain access to data without authorization.
T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
T.MODIFY	An attacker may attempt to modify or replace TSF data as it is being transmitted between physically separate parts of the TOE or other trusted IT entities.
T.TSF_COMP	An attacker or user may cause, through an unsophisticated attack, the TSF to be inappropriately accessed (viewed, modified, or deleted).
T.UNAUTH	A user may accidently perform actions that are not authorized by the TOE security policy.

[19] TSF – TOE Security Functionality

3.2 Organizational Security Policies

There are no Organizational Security Policies defined for this Security Target.

3.3 Assumptions

This section describes the security aspects of the intended environment for the evaluated TOE. The operational environment must be managed in accordance with assurance requirement documentation for delivery, operation, and user guidance. Table 6 lists the specific conditions that are required to ensure the security of the TOE and are assumed to exist in an environment where this TOE is employed.

Table 6 – Assumptions

Name	Description
A.FIPS	A FIPS 140-2 validated cryptographic module in the TOE environment must provide all cryptographic functionality for the TOE.
A.FIREWALL	All ports needed for proper operation of the TOE will be opened at the firewall. Also, any firewall settings necessary for the TOE's operation will be configured to allow the TOE to operate.
A.INSTALL	The Security Controls Console is installed on a server running Windows Server 2019 that is dedicated to the TOE and its Distribution Server.
A.LOCATE	The TOE is located within a controlled access facility.
A.MANAGE	There are one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NETCON	The TOE environment provides the network connectivity required to allow the TOE to provide secure patch management functions.
A.NOEVIL	The users who manage the TOE are non-hostile, appropriately trained, and follow all guidance.
A.OS_ACCESS	The TOE environment is in a secure state and provides a sufficient level of protection to itself and the TOE components.
A.OS_AUTH	The TOE environment will provide identification and authentication functions for users attempting to manage and use the TOE.
A.SECCOMM	The environment provides a sufficient level of protection to secure communications between Distribution Servers, agents, and other TOE components.
A.TIMESTAMP	The TOE environment provides the TOE with the necessary reliable timestamps.

4. Security Objectives

Security objectives are concise, abstract statements of the intended solution to the problem defined by the security problem definition (see Section 3). The set of security objectives for a TOE form a high-level solution to the security problem. This high-level solution is divided into two part-wise solutions: the security objectives for the TOE and the security objectives for the TOE's operational environment. This section identifies the security objectives for the TOE and its supporting environment.

4.1 Security Objectives for the TOE

The specific security objectives for the TOE are listed in Table 7 below.

Table 7 – Security Objectives for the TOE

Name	Description
O.EXPORT	The TOE must allow only authorized users to export end user application batch binaries with associated security attributes from within the TOE to the Distribution Server.
O.IMPORT	The TOE must allow only authorized users to import end user application batch binaries with associated security attributes into the TOE from vendor websites.
O.INT_ATK	The TOE implementation must be able to mitigate attacks to stored executable code and thread overuse.
O.INTEGRITY	The TOE must ensure data being transmitted to physically separate parts of the TOE is protected from unauthorized modification.
O.LOG	The TOE must record events of security relevance and provide authorized users with the ability to review the recorded events.
O.MANAGE	The TOE will only provide to a user all the functions and facilities necessary to support the user's management of the security of the TOE.
O.MONITOR	The TOE must be able to monitor machines on the network to ensure that they exist in a secure state and alert TOE users if a system enters an insecure state.
O.ROLE	The TOE must be able to associate users with the appropriate role after the user authenticates.

4.2 Security Objectives for the Operational Environment

This section describes the environmental objectives.

4.2.1 IT Security Objectives

Table 8 below lists the IT security objectives that are to be satisfied by the environment.

Table 8 – IT Security Objectives

Name	Description
OE.CONNECT	The TOE environment must be implemented such that the TOE is appropriately located within and connected to the network to perform its intended function.

Name	Description
OE.FIPS	The operating system that the TOE is installed upon must provide FIPS 140-2 validated cryptographic algorithms for the TOE to use to perform cryptographic functions.
OE.FIREWALL	The firewall must have all ports needed for proper operations of the TOE opened.
OE.OS_ACCESS	The operating system on which the TOE is installed provides a sufficient level of protection for itself and the TOE.
OE.OS_AUTH	The operating system on which the TOE is installed must provide authentication and identification of individuals attempting to use the TOE.
OE.PLATFORM	The TOE environment must include hardware and an operating system on which the TOE can be installed.
OE.SECCOMM	The TOE environment must provide mechanisms to secure communications between TOE agents, Distribution Servers, and other TOE components.
OE.TIME	The TOE environment must provide reliable timestamps to the TOE.

4.2.2 Non-IT Security Objectives

Table 9 below lists the non-IT environment security objectives that are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

Table 9 – Non-IT Security Objectives

Name	Description
NOE.MANAGE	Sites deploying the TOE will provide competent, non-hostile TOE users who are appropriately trained and follow all administrative guidance. TOE users will ensure the system is used securely.
NOE.PHYSICAL	The physical environment must be suitable for supporting a computing device in a secure setting.
NOE.REVIEW	The configuration of the TOE will be inspected on a regular basis to ensure that the configuration continues to meet the organization's security policies in the face of: • Changes to the TOE configuration • Changes in the security objectives • Changes to the Windows OS, including updates to the FIPS 140-2 certified Cryptographic Service Provider • Changes to the hardware on which the TOE is installed • Changes to the VMware ESXi hypervisors • Changes in the threats presented by the hostile network • Changes (additions and deletions) in the services available between the hostile network and the corporate network

5. Extended Components

This section defines the extended SFRs and extended SARs met by the TOE. These requirements are presented following the conventions identified in Section 6.1.

5.1 Extended TOE Security Functional Components

This section specifies the extended SFRs for the TOE. The extended SFRs are organized by class. Table 10 identifies all extended SFRs implemented by the TOE.

Table 10 – Extended TOE Security Functional Requirements

Name	Description
FDC_ANA_EXT.1	System Analysis
FDC_SCN_EXT.1	System Scan
FDC_STG_EXT.1	Scanned Data Storage

5.1.1 Class FDC: Data Collection and Analysis

Data Collection and Analysis functions involve:

- Scanning systems to obtain data
- Storing the collected data
- Performing analysis on collected data and presenting analytical results to users in a format that allows them to take appropriate actions

The FDC: Data Collection and Analysis class was modeled after the CC FAU: Security audit class. The extended family and related components for FDC_ANA_EXT: System Analysis were modeled after the CC family and related components for FAU_SAA: Security audit analysis. The extended family FDC_SCN_EXT: System Scan was modeled after the CC family FAU_GEN: Security audit data generation. The extended family FDC_STG_EXT: Scanned Data Storage was modeled after the CC family FAU_STG: Security audit event storage.

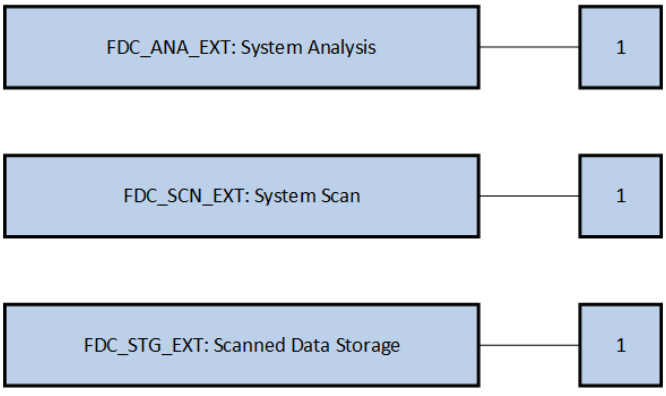

Figure 2 – FDC: Data Collection and Analysis Class Decomposition

5.1.1.1 FDC_ANA_EXT: System Analysis

Family Behavior

This family defines the requirements for the use of tools for the analysis of collected data and that allow users to react to potential security violations found during analysis of collected data.

Component Leveling

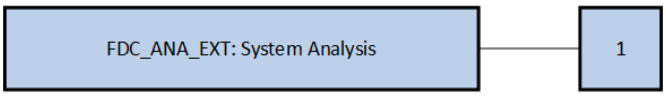

Figure 3 – FDC_ANA_EXT: System Analysis family decomposition

FDC_ANA_EXT.1: System Analysis provides the capability to analyze collected data and present the results to users in a way that easily allows them to respond to potential security violations found during the analysis.

Management: FDC_ANA_EXT.1

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

- Maintenance (deletion, modification, addition) of the analysis rules or the set of systems the rules are applied to.

Audit: FDC_ANA_EXT.1

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

- Minimal: Identity of the entity who initiated a scan or deployed a patch.
- Minimal: Identity of the scanned machines, list of security violations discovered, list of configuration changes made, and list of patches applied to machines.

FDC_ANA_EXT.1 System Analysis

Hierarchical to: No other components

Dependencies: FDC_SCN_EXT.1 System Scan

This component provides the capability to analyze collected data and present the results to users in a way that easily allows them to respond to potential security violations found during the analysis.

FDC_ANA_EXT.1.1

The TSF shall be able to apply a set of rules in monitoring the scanned data and based upon these rules indicate potential security violations:

a) compare applied patches against a list of potential patches and indicate which applications do not have all patches applied.

FDC_ANA_EXT.1.2

The TSF shall enforce the following set of rules for monitoring scanned data:

a) [assignment: *Information Flow Control Policy to be applied to scanned data*];

b) [assignment: *any other rules*].

FDC_ANA_EXT.1.3

The TSF shall be able to indicate a possible security violation to [assignment: *list of users with permission to review analytical results*] and allow [assignment: *list of users with permission to apply patches or configuration updates to scanned machines*] to address security violations that are discovered.

5.1.1.2 FDC_SCN_EXT: System Scan

Family Behavior

This family defines the requirements for scanning systems to retrieve data about their patch deployment and configuration state.

Component Leveling

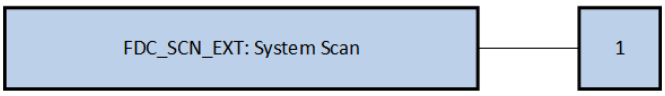

Figure 4 – FDC_SCN_EXT: System Scan family decomposition

FDC_SCN_EXT.1: System Scan defines the scanning function and specifies which machines will have a scan performed on them.

Management: FDC_SCN_EXT.1

- There are no management activities foreseen.

Audit: FDC_SCN_EXT.1

- There are no auditable events foreseen.

FDC_SCN_EXT.1 System Scan
Hierarchical to: No other components
Dependencies: FPT_STM.1 Reliable time stamps

This component provides the ability to scan targeted machines for data related to patch levels.

FDC_SCN_EXT.1.1

The System shall be able to collect the following information from the targeted IT System resource(s):

a) patch levels for [assignment: *list of applications to monitor patch levels for*]

FDC_SCN_EXT.1.2

The TSF shall record within each scan file at least the following information:

a) Date and time of the scan, list of machines scanned, identity of the entity who initiated the scan, list of security violations discovered during the scan

5.1.1.3 FDC_STG_EXT: Scanned Data Storage

Family Behavior

This family defines the requirements for protecting stored scan data.

Component Leveling

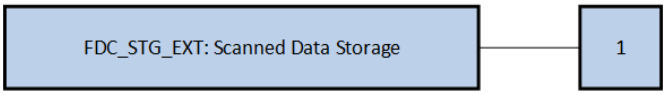

Figure 5 – FDC_STG_EXT: Scanned Data Storage family decomposition

FDC_STG_EXT.1: Scanned Data Storage, defines how the TSF protects stored scan data from unauthorized modification or deletion.

Management: FDC_STG_EXT.1

- There are no management activities foreseen.

Audit: FDC_STG_EXT.1

- There are no auditable events foreseen.

FDC_STG_EXT.1Scanned Data Storage

Hierarchical to: **No other components**

Dependencies: **FDC_SCN_EXT.1 System Scan**

This component provides the ability to protect stored scan data from unauthorized deletion and modification.

FDC_STG_EXT.1.1

The TSF shall protect the stored scan data from unauthorized deletion.

FDC_STG_EXT.1.2

The TSF shall be able to prevent unauthorized modifications to the stored scan data.

5.2 Extended TOE Security Assurance Components

There are no extended SARs defined for this Security Target.

6. Security Requirements

This section defines the SFRs and SARs met by the TOE. These requirements are presented following the conventions identified in Section 6.1.

6.1 Conventions

There are several font variations used within this ST. Selected presentation choices are discussed here to aid the Security Target reader.

The CC allows for assignment, refinement, selection, and iteration operations to be performed on security functional requirements. All of these operations are used within this ST. These operations are performed as described in Part 2 of the CC, and are shown as follows:

- Completed assignment statements are identified using [*italicized text within brackets*].
- Completed selection statements are identified using [underlined text within brackets].
- Refinements are identified using **bold text**. Any text removed is stricken (Example: ~~TSF Data~~) and should be considered as a refinement.
- Extended Functional and Assurance Requirements are identified using "_EXT" at the end of the short name.
- Iterations are identified by appending a letter in parentheses following the component title. For example, FAU_GEN.1(a) Audit Data Generation would be the first iteration and FAU_GEN.1(b) Audit Data Generation would be the second iteration.

6.2 Security Functional Requirements

This section specifies the SFRs for the TOE. This section organizes the SFRs by CC class. Table 11 identifies all SFRs implemented by the TOE and indicates the ST operations performed on each requirement.

Table 11 – TOE Security Functional Requirements

Name	Description	S	A	R	I
FAU_GEN.1	Audit Data Generation	✓	✓		
FAU_SAR.1	Audit review		✓		
FDP_ACC.1	Subset access control		✓		
FDP_ACF.1	Security attribute based access control		✓		
FDP_ETC.2	Export of user data with security attributes		✓		
FDP_IFC.1(a)	Subset information flow control (Scan Data Analysis)		✓		✓
FDP_IFC.1(b)	Subset information flow control (Deployment)		✓		✓
FDP_IFF.1(a)	Simple security attributes (Scan Data Analysis)		✓		✓
FDP_IFF.1(b)	Simple security attributes (Deployment)		✓		✓
FDP_ITC.1	Import of user data without security attributes		✓		

Name	Description	S	A	R	I
FIA_ATD.1	User attribute definition		✓		
FMT_MOF.1	Management of security functions behaviour	✓	✓		
FMT_MSA.1(a)	Management of security attributes (user roles)	✓	✓		✓
FMT_MSA.1(b)	Management of security attributes (machine properties)	✓	✓		✓
FMT_MSA.3(a)	Static attribute initialisation (Access Control SFP)	✓	✓		✓
FMT_MSA.3(b)	Static attribute initialisation (Protect SFP)	✓	✓		✓
FMT_MTD.1	Management of TSF data	✓	✓		
FMT_SMF.1	Specification of management functions		✓		
FMT_SMR.1	Security roles		✓		
FPT_ITT.1	Basic internal TSF data transfer protection	✓			
FPT_ITT.3	TSF data integrity monitoring	✓	✓		
FPT_TST.1	TSF testing	✓	✓	✓	
FRU_RSA.1	Maximum quotas	✓	✓		
FDC_ANA_EXT.1	System analysis		✓		
FDC_SCN_EXT.1	System scan		✓		
FDC_STG_EXT.1	Scanned data storage				

Note: S=Selection; A=Assignment; R=Refinement; I=Iteration

6.2.1 Class FAU: Security Audit

FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

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) [*machines scanned, patches applied, discovered security violations*].

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, [*no other information*].

FAU_SAR.1 Audit review

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAR.1.1

The TSF shall provide [*all TOE users*] with the capability to read [*machines scanned, patches applied, discovered security violations*] 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.

6.2.2 Class FDP: User Data Protection

FDP_ACC.1 Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1

The TSF shall enforce the [*Access Control SFP*] on [

- *Subjects: TOE users*
- *Objects: User interface menu items, policies, machine groups, scans, and end user application patch binaries*
- *Operations: All interactions between the subjects and objects identified above*

].

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1

The TSF shall enforce the [*Access Control SFP*] to objects based on the following: [

- *Subject attributes:*
  - *Role*
  - *Windows user ID*
- *and Object attributes:*
  - *Permissions assigned to objects*

].

FDP_ACF.1.2

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

- *If user requests access to an object and the role associated with that user has permission to access that object, then access is granted. A mapping of role to permissions is provided in Table 12 below.*
- *If the rules above do not apply, then access is denied.*

].

FDP_ACF.1.3

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [*no other rules*].

FDP_ACF.1.4

The TSF shall explicitly deny access of subjects to objects based on the [*no other rules*].

FDP_ETC.2 Export of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control
FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1

The TSF shall enforce the [*Protect SFP*] when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.2.2

The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3

The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.

FDP_ETC.2.4

The TSF shall enforce the following rules when user data is exported from the TOE: [*no rules specified*].

FDP_IFC.1(a) Subset information flow control (Scan Data Analysis)

Hierarchical to: No other components.
Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1(a).1

The TSF shall enforce the [*Protect SFP*] on [

a) Subjects: Machines that are members of machine groups
b) Information: Data obtained by scanning the machines
c) Operations: Analysis of scanned data against a patch list

].

FDP_IFC.1(b) Subset information flow control (Deployment)

Hierarchical to: No other components.
Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1(b).1

The TSF shall enforce the [*Protect SFP*] on [

a) Subjects: Machines that are members of machine groups
b) Information: End user application patch binaries to be deployed to end user applications
c) Operations: Deployment of end user application patch binaries to machines

].

FDP_IFF.1(a) Simple security attributes (Scan Data Analysis)

Hierarchical to: No other components.
Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

FDP_IFF.1(a).1

The TSF shall enforce the [*Protect SFP*] based on the following types of subject and information security attributes: [

Subject Attributes:

a) Machine group membership

Information Attributes:

a) Machine of origin

].

FDP_IFF.1(a).2

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

a) *An authorized user requests that a machine be scanned*

].

FDP_IFF.1(a).3

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

FDP_IFF.1(a).4

The TSF shall explicitly authorise an information flow based on the following rules: [*an authorized user with appropriate permissions has scheduled a scan to be performed at some point in the future*].

FDP_IFF.1(a).5

The TSF shall explicitly deny an information flow based on the following rules: [*no additional rules*].

FDP_IFF.1(b) Simple security attributes (Deployment)

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

FDP_IFF.1(b).1

The TSF shall enforce the [*Protect SFP*] based on the following types of subject and information security attributes: [

Subject Attributes:

a) Machine group membership

Information Attributes:

a) Machine of origin
b) Installed applications
c) Installed patches
d) Digital signature of the patch file (if applicable)

].

FDP_IFF.1(b).2

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

a) *An authorized user requests that an end user application patch be deployed to a machine*

].

FDP_IFF.1(b).3

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

FDP_IFF.1(b).4

The TSF shall explicitly authorise an information flow based on the following rules: [*an authorized user with appropriate permissions has scheduled an end user application patch deployment to be performed at some point in the future*].

FDP_IFF.1(b).5

The TSF shall explicitly deny an information flow based on the following rules: *[the patch does not match its signature (if applicable)].*

FDP_ITC.1 Import of user data without security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_MSA.3 Static attribute initialisation

FDP_ITC.1.1

The TSF shall enforce the [*Protect SFP*] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2

The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3

The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [*no additional rules*].

6.2.3 Class FIA: Identification and Authentication

FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_ATD.1.1

The TSF shall maintain the following list of security attributes belonging to individual users: [*Role, Windows user account ID*].

6.2.4 Class FMT: Security Management

FMT_MOF.1 Management of security functions behaviour

Hierarchical to: No other components.

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MOF.1.1

The TSF shall restrict the ability to [determine the behaviour of, modify the behaviour of] the functions [*in the 'Permissions' column of Table 12*] to [*the roles indicated in the 'Role' column of Table 12*].

Table 12 – Security functions behaviour by role

Role	Permissions
Administrator	Create, delete, modify users Create, delete, modify machine groups Initiate, schedule scans Initiate, schedule patch updates Create, delete, modify patch groups Create, view reports Create, delete, modify deployment templates Delete scan/deployment results Create, delete, modify agent policy Install, remove Security Controls Agent

Role	Permissions
Full user	Create, delete, modify machine groups Initiate, schedule scans Initiate, schedule patch updates Create, delete, modify patch groups Create, view reports Create, delete, modify deployment templates Delete scan/deployment results Create, delete, modify agent policy Install, remove Security Controls Agent
Scan and report only	Initiate, schedule scans Create, view reports
Patch deploy and report only	Initiate, schedule patch updates Create, view reports
Reporting > Patch Report Only	Create, view patch reports

FMT_MSA.1(a) Management of security attributes (User roles)

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security fdroles

FMT_MSA.1(a).1

The TSF shall enforce the [*Access Control SFP*] to restrict the ability to [change_default, modify] the security attributes [*Role*] to [*Administrator*].

FMT_MSA.1(b) Management of security attributes (Machine properties and scan schedules)

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MSA.1(b).1

The TSF shall enforce the [*Protect SFP*] to restrict the ability to [change default, query, modify, delete] the security attributes [*Machine group membership*] to [*Administrator and Full user*].

FMT_MSA.3(a) Static attribute initialisation (Access Control SFP)

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3(a).1

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

FMT_MSA.3(a).2

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

FMT_MSA.3(b) Static attribute initialisation (Protect SFP)

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3(b).1

The TSF shall enforce the [*Protect SFP*] to provide [restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3(b).2

The TSF shall allow the [*Administrator, Full user, Patch deploy and report only*] to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MTD.1.1

The TSF shall restrict the ability to [query, delete] the [*data from scanned machines*] to [*the Administrator and Full user*].

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No Dependencies

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions: [*management of security functions behavior, management of security attributes, management of TSF data*].

FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1

The TSF shall maintain the roles [

For the TOE:

a) Administrator
b) Full user
c) Scan and report only
d) Patch deploy and report only
e) Reporting
a. Patch Report only

].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

6.2.5 Class FPT: Protection of the TSF

FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_ITT.1.1

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

FPT_ITT.3 TSF data integrity monitoring

Hierarchical to: No other components.

Dependencies: FPT_ITT.1 Basic internal TSF data transfer protection

FPT_ITT.3.1

The TSF shall be able to detect [modification of data, substitution of data] for TSF data transmitted between separate parts of the TOE.

FPT_ITT.3.2

Upon detection of a data integrity error, the TSF shall take the following actions: [*drop the corrupted data*].

FPT_TST.1 TSF testing

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST.1.1

The TSF shall run a suite of self tests [at the conditions [*during execution of a TOE component*]] to demonstrate the correct operation of [the TSF].

FPT_TST.1.2

The TSF shall ~~provide authorised users with the capability to~~ **automatically** verify the integrity of [digitally signed TSF data].

FPT_TST.1.3

The TSF shall ~~provide authorised users with the capability to~~ **automatically** verify the integrity of [*stored TSF executable code*].

6.2.6 Class FRU: Resource Utilization

FRU_RSA.1 Maximum quotas

Hierarchical to: No other components.

Dependencies: No dependencies

FRU_RSA.1.1

The TSF shall enforce maximum quotas of the following resources: [*threads dedicated to scanning machines*] that [a defined group of users] can use [simultaneously]].

6.2.7 Class FDC: Data Collection and Analysis

FDC_ANA_EXT.1 System Analysis

Hierarchical to: No other components

Dependencies: FDC_SCN_EXT.1 System Scan

FDC_ANA_EXT.1.1

The TSF shall be able to apply a set of rules in monitoring the scanned data and based upon these rules indicate potential security violations.

a) compare applied patches against a list of potential patches and indicate which applications do not have all patches applied.

FDC_ANA_EXT.1.2

The TSF shall enforce the following set of rules for monitoring scanned data:

a) [*Protect SFP*];

b) [*no other rules*].

FDC_ANA_EXT.1.3

The TSF shall be able to indicate a possible security violation to [*Administrator, Full user, Scan and report only, and Patch deploy and report only*] and allow [*Administrator, Full user, and Patch deploy and report only*] to address security violations that are discovered.

FDC_SCN_EXT.1 System Scan

Hierarchical to: No other components

Dependencies: FPT_STM.1 Reliable time stamps

FDC_SCN_EXT.1.1

The System shall be able to collect the following information from the targeted IT System resource(s):

a) Patch levels for [*the list of applications supported under the Protect SFP*]

FDC_SCN_EXT.1.2

The TSF shall record within each scan file at least the following information:

a) Date and time of the scan, list of machines scanned, identity of the entity who initiated the scan, list of security violations discovered during the scan

FDC_STG_EXT.1 Scanned Data Storage

Hierarchical to: No other components

Dependencies: FDC_SCN_EXT.1 System Scan

FDC_STG_EXT.1.1

The TSF shall protect the stored scan data from unauthorized deletion.

FDC_STG_EXT.1.2

The TSF shall be able to prevent unauthorized modifications to the stored scan data.

6.3 Security Assurance Requirements

This section defines the assurance requirements for the TOE. Assurance requirements are taken from the CC Part 3 and are EAL2 augmented with ALC_FLR.2. Table 13 summarizes these requirements.

Table 13 – Assurance Requirements

Assurance Requirements	
Class 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
Class ALC : Life Cycle Support	ALC_CMC.2 Use of a CM[20] system
	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_DEL.1 Delivery Procedures
	ALC_FLR.2 Flaw reporting procedures
Class ADV: Development	ADV_ARC.1 Security Architecture Description
	ADV_FSP.2 Security-enforcing functional specification
	ADV_TDS.1 Basic design
Class AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
Class ATE: Tests	ATE_COV.1 Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
Class AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

[20] CM – Configuration Management

7. TOE Security Specification

This section presents information to detail how the TOE meets the functional requirements described in previous sections of this ST.

7.1 TOE Security Functionality

Each of the security requirements and the associated descriptions correspond to a security functionality. Hence, each security functionality is described by how it specifically satisfies each of its related requirements. This serves to both describe the security functionality and rationalize that the security functionality satisfies the necessary requirements. Table 14 lists the security functionality and their associated SFRs.

Table 14 – Mapping of TOE Security Functionality to Security Functional Requirements

TOE Security Functionality	SFR ID	Description
Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_SAR.1	Audit review
User Data Protection	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute based access control
	FDP_ETC.2	Export of user data with security attributes
	FDP_IFC.1(a)	Subset information flow control (Scan Data Analysis)
	FDP_IFC.1(b)	Subset information flow control (Deployment)
	FDP_IFF.1(a)	Simple security attributes (Scan Data Analysis)
	FDP_IFF.1(b)	Simple security attributes (Deployment)
	FDP_ITC.1	Import of user data without security attributes
Identification and Authentication	FIA_ATD.1	User attribute definition
Security Management	FMT_MOF.1	Management of security functions behaviour
	FMT_MSA.1(a)	Management of security attributes (user roles)
	FMT_MSA.1(b)	Management of security attributes (machine properties)
	FMT_MSA.3(a)	Static attribute initialisation (Access Control SFP)
	FMT_MSA.3(b)	Static attribute initialisation (Protect SFP)
	FMT_MTD.1	Management of TSF data
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
Protection of TOE Security Functions	FPT_ITT.1	Basic internal TSF data transfer protection
	FPT_ITT.3	TSF data integrity monitoring
	FPT_TST.1	TSF testing
Resource Utilization	FRU_RSA.1	Maximum quotas
Data Collection and Analysis	FDC_ANA_EXT.1	System analysis

TOE Security Functionality	SFR ID	Description
	FDC_SCN_EXT.1	System scan
	FDC_STG_EXT.1	Scanned data storage

7.1.1 Security Audit

The TOE generates audit records each time a machine is scanned, a patch is applied, and a security violation is discovered. Audit records are also generated upon startup and shutdown of Security Controls audit functions. These startup/shutdown events are logged in the Windows Event Log.

The TOE generates audit logs that contain the information provided in Table 15 below.

Table 15 – Audit Record Contents

Field	Content
Date/Time	Date and time of the event
Event Type	Description of the event
Subject Identity	Unique ID of subject initiating the event; may not always be applicable
Outcome	Success or failure of the event

The TOE provides audit logs for all authenticated users of the TOE to review in a form suitable for interpretation of the information in the logs. The logs containing scan, patch, and security violation information are available via the Ivanti Security Controls Console. Only authorized users of the TOE are permitted to view the audit records. Users with Windows administrative abilities may view startup/shutdown events through the Windows Event Viewer.

TOE Security Functional Requirements Satisfied: FAU_GEN.1, FAU_SAR.1.

7.1.2 User Data Protection

The TOE implements an Access Control SFP and a Protect SFP.

The Access Control SFP manages access to Security Controls security functions. When a local user invokes the Security Controls Console, the application checks the assigned role and then only grants permission to access the management options (“objects”) for which that user’s role is authorized.

Access to machine-scanning functionality and patch-deployment functionality is controlled based on the Protect SFP. Only authorized users may initiate a manual (immediate) or scheduled (delayed) machine scan or patch deployment. A machine scan is performed to determine the status of applications on a machine and the current patch status. A machine scan is initiated from the Security Controls Console to one or more machines. The machine scan can be performed against a machine running the agentless configuration or on the Security Controls Agent. Machine scans can be run on machine groups containing machines with either configuration. The TOE ensures the integrity of a patch update file used during patch deployment is verified before it is used, and any patch update file that fails integrity verification is not used. The TOE requests that the Windows OS Cryptographic Service

provider perform integrity verification on the digital signatures of the patch data. The TOE ensures the Windows OS verification passes prior to installing the patch.

Patch binary data with a digital signature (if available) is imported from vendor websites into the TOE. Transport of this information may only be performed by an authorized user when authenticated upon login to the Windows environment. An authorized user may check the vendor website location, file name, file date, and version number. If the end user application patch binaries are valid, then the authorized user can export the end user application patch binaries from the TOE to a specified Distribution Server.

TOE Security Functional Requirements Satisfied: FDP_ACC.1, FDP_ACF.1, FDP_ETC.2, FDP_IFC.1(a), FDP_IFC.1(b), FDP_IFF.1(a), FDP_IFF.1(b), FDP_ITC.1.

7.1.3 Identification and Authentication

The users of the TOE are authenticated by the underlying Windows OS before the TOE is invoked. After the TOE is invoked, it uses the user's Windows user account ID (Windows username) and role (assigned by the TOE) for identification and access control purposes.

TOE Security Functional Requirements Satisfied: FIA_ATD.1.

7.1.4 Security Management

The TOE provides three security management functions:

- Management of security functions behavior
- Management of security attributes
- Management of TSF data

The TOE implements administrative roles and associates each TOE user with one or more of these roles. The Security Controls application implements five administrative roles:

- Administrator
- Full user
- Scan and report only
- Patch deploy and report only
- Reporting
    - Patch Report only

Roles are used by the TOE to determine which users may manage the behavior of the TOE's security functions. The TOE determines which Security Controls security functions each user may manage based on the assigned role and the permissions available to that role. Table 12 above provides this access control matrix.

Administrative roles are also used by the TOE to determine which users may manage user roles and machine group membership.

The TOE manages the Access Control SFP and the Protect SFP to provide restrictive default values for SFP security attributes. These attributes can be overridden by users with authorized roles.

The TOE protects access to patch data, vulnerability data, and policy data, only allowing authorized users to view, modify, or delete the data.

TOE Security Functional Requirements Satisfied: FMT_MOF.1, FMT_MSA.1(a), FMT_MSA.1(b), FMT_MSA.3(a), FMT_MSA.3(b), FMT_MTD.1, FMT_SMF.1, FMT_SMR.1.

7.1.5 Protection of the TSF

Security Controls digitally signs all executables and policy data pushed to a machine for deployment. The TOE ensures that all patch data, which includes patches for the Security Controls application and Windows OS, are digitally signed using the cryptographic operations provided by the Windows OS. The TOE ensures that the integrity of the data is verified on the target machine prior to installation, and if the integrity verification fails, the TOE does not install it. Integrity verification is based on digital signatures of the Ivanti executables and policy data. The digital signatures are verified by a FIPS 140-2 validated Cryptographic Service Provider on the Windows OS.

In order to prevent tampering by malicious software (such as viruses), each executable file and most library files[21] composing the TOE are digitally signed. The TOE verifies the integrity of stored signed code prior to allowing it to be deployed. Integrity verification is based on digital signatures of the stored executable code. The TOE requests that the Windows OS FIPS 140-2 validated Cryptographic Service Provider verify the digital signature prior to deployment and will not perform the update until verification is received from the OS that the signature is verified. (The Windows OS FIPS 140-2 validated Cryptographic Service Provider is outside the scope of this evaluation and will not be discussed further in this Security Target.)

TOE Security Functional Requirements Satisfied: FPT_ITT.1, FPT_ITT.3, FPT_TST.1.

7.1.6 Resource Utilization

In order to prevent resource exhaustion, the TOE limits the number of simultaneous scans that users may initiate. By default, Security Controls will allow up to 8 simultaneous scans per CPU core; however, it can be configured to allow up to 64 simultaneous scans per CPU core.

TOE Security Functional Requirements Satisfied: FRU_RSA.1.

7.1.7 Data Collection and Analysis

The Security Controls application can scan a machine or machine group on the network. Scans can be performed from the Security Controls Console against an agentless target machine or a machine running the Security Controls Agent. An authorized user selects the machine or machine group to be scanned from the GUI. The scan can be performed immediately or scheduled to run at a future point in time. When a scan is run, the TOE generates collection logs that contain the following information:

[21] Library files provided by Developer Express and Grape City are not digitally signed. This is specific to Non-English Developer Express Resource localization modules.

- Date and time of the scan
- List of machines scanned
- Identity of the entity (user or process on behalf of a user) who initiated the scan
- List of installed and missing patches

The TOE protects the scan data collection logs from unauthorized deletion and modification. Only authorized users with the Administrator or Full user role may use the Security Controls Console GUI to clear the logs or delete scan data.

After scan data is collected, the TOE performs automated analysis of the scan data to identify missing patches. When potential security violations (missing patches) are detected, the Protect SFP is enforced, allowing a user to view and address the violations.

TOE Security Functional Requirements Satisfied: FDC_ANA_EXT.1, FDC_SCN_EXT.1, FDC_STG_EXT.1.

8. Rationale

8.1 Conformance Claims Rationale

This Security Target extends Part 2 and conforms to Part 3 of the *Common Criteria for Information Technology Security Evaluation*, Version 3.1 Release 4.

8.2 Security Objectives Rationale

This section provides a rationale for the existence of each threat, policy statement, and assumption that compose the Security Target. Sections 8.2.1, 8.2.2, and 8.2.3 demonstrate the mappings between the threats, policies, and assumptions to the security objectives are complete. The following discussion provides detailed evidence of coverage for each threat, policy, and assumption.

8.2.1 Security Objectives Rationale Relating to Threats

Table 16 below provides a mapping of the objectives to the threats they counter.

Table 16 – Threats: Objectives Mapping

Threats	Objectives	Rationale
T.AUDACC Persons may not be accountable for an action that they conduct because there is not an audit event added to the audit trail, thus allowing an attacker to escape detection.	O.LOG The TOE must record events of security relevance and provide authorized users with the ability to review the recorded events.	O.LOG counters this threat by ensuring that an audit trail of management events on the TOE is generated.
	OE.TIME The TOE environment must provide reliable timestamps to the TOE.	OE.TIME counters this threat by ensuring that accurate timestamps are provided for all audit records, allowing the order of events to be preserved.
T.BADSTATE An attacker may exploit vulnerabilities in monitored IT entities that reach an insecure state without the network administrators becoming aware.	O.MONITOR The TOE must be able to monitor machines on the network to ensure that they exist in a secure state and alert TOE users if a system enters an insecure state.	O.MONITOR counters this threat by ensuring that systems on the network are monitored by the TOE and that the TOE alerts TOE users when a security violation occurs.
T.INT_ATK An attacker may exploit internal weaknesses in the TOE implementation to gain access to data without authorization.	O.INT_ATK The TOE implementation must be able to mitigate attacks to stored executable code and thread overuse.	O.INT_ATK counters this threat by ensuring that the TOE is implemented in such a way as to prevent attackers from substituting TOE executable code and preventing the overuse of threads.
T.MASQUERADE A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	O.EXPORT The TOE must allow only authorized users to export end user application batch binaries with associated security attributes from within the TOE to the Distribution Server.	O.EXPORT counters this threat by ensuring the validity of all end user application patch binary data exported from the TOE to the Distribution Server.

Threats	Objectives	Rationale
	O.IMPORT The TOE must allow only authorized users to import end user application batch binaries with associated security attributes into the TOE from vendor websites.	O.IMPORT counters this threat by ensuring the validity of all end user application patch binary data imported from vendor websites into the TOE.
	O.ROLE The TOE must be able to associate users with the appropriate role after the user authenticates.	O.ROLE counters this threat by ensuring that the TOE is able to associate users with roles according to their operating system user identifier.
	OE.OS_AUTH The operating system where the TOE is installed must provide authentication and identification of individuals attempting to use the TOE.	OE.OS_AUTH counters this threat by ensuring that the operating system identifies and authenticates TOE users.
T.MODIFY An attacker may attempt to modify or replace TSF data as it is being transmitted between physically separate parts of the TOE or other trusted IT entities.	O.INTEGRITY The TOE must protect data being transmitted to physically separate parts of the TOE from unauthorized modification.	O.INTEGRITY counters this threat by ensuring that data transferred between physically separate parts of the TOE is not modified or replaced during transmission.
T.TSF_COMP An attacker or user may cause through an unsophisticated attack, the TSF to be inappropriately accessed (viewed, modified, or deleted).	O.MANAGE The TOE will only provide to a user all the functions and facilities necessary to support the user's management of the security of the TOE.	O.MANAGE counters this threat by restricting the management functions of the TOE to authorized users.
T.UNAUTH A user may accidently perform actions that are not authorized by the TOE security policy.	O.EXPORT The TOE must allow only authorized users to export end user application batch binaries with associated security attributes from within the TOE to the Distribution Server.	O.EXPORT counters this threat by ensuring that only authenticated users of the TOE with the appropriate role may export end user patch application data from the TOE to the Distribution Server.
	O.IMPORT The TOE must allow only authorized users to import end user application batch binaries with associated security attributes into the TOE from vendor websites.	O.IMPORT counters this threat by ensuring that only authenticated users of the TOE with the appropriate role may import end user application patch binary data from vendor websites into the TOE.
	O.MANAGE The TOE will only provide to a user all the functions and facilities necessary to support the user's management of the security of the TOE.	O.MANAGE counters this threat by limiting the management functions made available to users.
	O.ROLE The TOE must be able to associate users with the appropriate role after the user authenticates.	O.ROLE counters this threat by ensuring that users are associated with roles while logged into the TOE.
	OE.OS_AUTH The operating system where the TOE is installed must provide authentication and identification of individuals attempting to use the TOE.	OE.OS_AUTH counters this threat by ensuring that the operating system identifies and authenticates all TOE users.

Every threat is mapped to one or more objectives in the table above. This complete mapping demonstrates that the defined security objectives counter all defined threats.

8.2.2 Security Objectives Rationale Relating to Policies

There are no Organizational Security Policies defined for this Security Target.

8.2.3 Security Objectives Rationale Relating to Assumptions

Table 17 below gives a mapping of assumptions and the environmental objectives that uphold them.

Table 17 – Assumptions: Objectives Mapping

Assumptions	Objectives	Rationale
A.FIPS A FIPS 140-2 validated cryptographic algorithms in the TOE environment must provide all cryptographic functionality for the TOE.	OE.FIPS The operating system that the TOE is installed upon must provide a FIPS 140-2 validated cryptographic algorithms for the TOE to use to perform cryptographic functions.	OE.FIPS upholds this assumption by ensuring that FIPS 140-2 cryptographic algorithms are available for the TOE to use within the operating system the TOE is installed upon.
A.FIREWALL All ports needed for proper operation of the TOE will be opened at the firewall. Also, any firewall settings necessary for the TOE's operation will be configured to allow the TOE to operate.	OE.FIREWALL The firewall must have all ports needed for proper operations of the TOE opened.	OE.FIREWALL upholds this assumption by ensuring that all ports necessary for the operation of the TOE are opened.
A.INSTALL The Security Controls Console is installed on a server running Windows Server 2019 that is dedicated to the TOE and its Distribution Server.	OE.PLATFORM The TOE environment must include hardware and an operating system for the TOE to be installed on.	OE.PLATFORM upholds this assumption by ensuring that an appropriate operating system and hardware is available for the TOE to be installed on.
	NOE.MANAGE Sites deploying the TOE will provide competent, non-hostile TOE users who are appropriately trained and follow all administrative guidance. TOE users will ensure the system is used securely.	OE.MANAGE upholds this assumption by ensuring that the TOE users read and follow the guidance for installation and deployment of the TOE.
A.LOCATE The TOE is located within a controlled access facility.	NOE.PHYSICAL The physical environment must be suitable for supporting a computing device in a secure setting.	OE.PHYCAL upholds this assumption by ensuring that the environment provides protection against physical attack.
A.MANAGE There are one or more competent individuals assigned to manage the TOE and the security of the information it contains.	NOE.MANAGE Sites deploying the TOE will provide competent, non-hostile TOE users who are appropriately trained and follow all administrative guidance. TOE users will ensure the system is used securely.	OE.MANAGE upholds this assumption by ensuring that those responsible for the TOE will provide competent individuals to perform management of the security of the environment and restrict these functions and facilities from unauthorized use.

Assumptions	Objectives	Rationale
	NOE.REVIEW The configuration of the TOE will be inspected on a regular basis to ensure that the configuration continues to meet the organization's security policies in the face of: • Changes to the TOE configuration • Changes in the security objectives • Changes to the Windows OS, including updates to the FIPS 140-2 certified Cryptographic Service Provider • Changes to the hardware on which the TOE is installed • Changes to the VMware ESXi hypervisors • Changes in the threats presented by the hostile network • Changes (additions and deletions) in the services available between the hostile network and the corporate network	OE.REVIEW upholds this assumption by ensuring that users assigned to manage the TOE will review the configuration on a regular basis to ensure that it accurately reflects the intended configuration.
A.NETCON The TOE environment provides the network connectivity required to allow the TOE to provide secure patch management functions.	OE.CONNECT The TOE environment must be implemented such that the TOE is appropriately located within and connected to the network to perform its intended function.	OE.CONNECT upholds this assumption by ensuring that the environment provides the TOE with the appropriate configuration to provide secure patch and configuration management functions.
A.NOEVIL The users who manage the TOE are non-hostile, appropriately trained, and follow all guidance.	NOE.MANAGE Sites deploying the TOE will provide competent, non-hostile TOE users who are appropriately trained and follow all administrative guidance. TOE users will ensure the system is used securely.	OE.MANAGE upholds this assumption by ensuring that all users assigned to manage the TOE are not careless, negligent, or willfully hostile; are appropriately trained; and follow all administrative guidance.
A.OS_ACCESS The TOE environment is in a secure state and provides a sufficient level of protection to itself and the TOE components.	OE.OS_ACCESS The operating system where the TOE is installed provides a sufficient level of protection for itself and the TOE.	OE.OS_ACCESS upholds this assumption by ensuring that the operating system where the TOE is installed provides enough protection for itself and the TOE.
A.OS_AUTH The TOE environment will provide identification and authentication functions for users attempting to manage and use the TOE.	OE.OS_AUTH The operating system where the TOE is installed must provide authentication and identification of individuals attempting to use the TOE.	OE.OS_AUTH upholds this assumption by ensuring that the operating system where the TOE is installed will provide authentication and identification of users attempting to use the TOE.
A.SECCOMM The environment provides a sufficient level of protection to secure communications between Distribution Servers, agents, and other TOE components.	OE.SECCOMM The TOE environment must provide mechanisms to secure communications between TOE agents, Distribution Servers, and other TOE components.	OE.SECCOMM upholds this assumption by ensuring that the TOE environment will provide adequate security to protect the TOE.
A.TIMESTAMP The TOE environment provides the TOE with the necessary reliable timestamps.	OE.TIME The TOE environment must provide reliable timestamps to the TOE.	OE.TIME upholds this assumption by ensuring that the operating system where the TOE is installed will provide reliable time stamps for the TOE.

Every assumption is mapped to one or more objectives in the table above. This complete mapping demonstrates that the defined security objectives uphold all defined assumptions.

8.3 Rationale for Extended Security Functional Requirements

A class of FDC requirements was created to specifically address the data collected and analyzed by patch management devices. The audit class of the CC (FAU) was used as a model for creating these requirements. The purpose of this class of requirements is to address the unique nature of patch deployments and provide requirements about collecting, analyzing, storing, and reviewing the data. FDC_SCN_EXT.1 is dependent on FPT_STM.1, since FDC_SCN_EXT.1 requires the TSF to record the date and time of scans, and these timestamps must be accurate and reliable . FDC_ANA_EXT.1 and FDC_STG_EXT.1 are dependent on FDC_SCN_EXT.1 since they apply to scan data that must first be collected by the TOE. These requirements exhibit functionality that can be easily documented in the ADV assurance evidence and thus do not require any additional Assurance Documentation.

8.4 Rationale for Extended TOE Security Assurance Requirements

There are no extended Security Assurance Requirements defined in this Security Target.

8.5 Security Requirements Rationale

The following discussion provides detailed evidence of coverage for each security objective.

8.5.1 Rationale for Security Functional Requirements of the TOE Objectives

Table 18 below shows a mapping of the objectives and the SFRs that support them.

Table 18 – Objectives: SFRs Mapping

Objective	Requirements Addressing the Objective	Rationale
O.EXPORT The TOE must allow only authorized users to export end user application batch binaries with associated security attributes from within the TOE to the Distribution Server.	FDP_ETC.2 Export of user data with security attributes	This requirement supports O.EXPORT by requiring the TOE to enforce an access control policy on users that are allowed to export validated end user application patch binaries from the TOE to the Distribution Server.
O.IMPORT The TOE must allow only authorized users to import end user application batch binaries with associated security attributes into the TOE from vendor websites.	FDP_ITC.1 Import of user data without security attributes	This requirement supports O.IMPORT by requiring the TOE to enforce an access control policy on users that are allowed to import validated end user application patch binaries from vendor websites into the TOE.
O.INT_ATK The TOE implementation must be able to mitigate attacks to stored executable code and thread overuse.	FPT_TST.1 TSF testing	This requirement supports O.INT_ATK by requiring the TOE to be able to perform a self-test verifying the integrity of stored TOE executable code.

Objective	Requirements Addressing the Objective	Rationale
	FRU_RSA.1 Maximum quotas	This requirement supports O.INT_ATK by requiring the TOE to set a limit on the number of threads available for scanning machines simultaneously.
O.INTEGRITY The TOE must protect data being transmitted to physically separate parts of the TOE from unauthorized modification.	FPT_ITT.1 Basic internal TSF data transfer protection	This requirement supports O.INTEGRITY by requiring the TOE to protect TSF data from unauthorized modification while it is being transmitted between separate parts of the TOE.
	FPT_ITT.3 TSF data integrity monitoring	This requirement supports O.INTEGRITY by requiring the TOE to drop TSF data that has been modified or replaced by an unauthorized entity.
O.LOG The TOE must record events of security relevance and provide authorized users with the ability to review the recorded events.	FAU_GEN.1 Audit Data Generation	This requirement supports O.LOG by requiring the TOE to produce audit records for the system security events and for actions caused by enforcement of the Access Control and Protect SFPs.
	FAU_SAR.1 Audit review	This requirement supports O.LOG by requiring the TOE to make the recorded audit records available for review.
O.MANAGE The TOE will only provide to a user all the functions and facilities necessary to support the user's management of the security of the TOE.	FDP_ACC.1 Subset access control	This requirement supports O.MANAGE by requiring the TOE to enforce an access control policy on users connecting to the TOE.
	FDP_ACF.1 Security attribute based access control	This requirement supports O.MANAGE by defining the access control policy that controls interactions between users and the TOE.
	FMT_MOF.1 Management of security functions behaviour	This requirement supports O.MANAGE by defining the management functions available to each type of user.
	FMT_MSA.1(a) Management of security attributes (user roles)	This requirement supports O.MANAGE by restricting the users who can manage user roles.
	FMT_MSA.1(b) Management of security attributes (machine properties)	This requirement supports O.MANAGE by restricting the users who can manage machine groups.
	FMT_MSA.3(a) Static attribute initialisation (Access Control SFP)	This requirement supports O.MANAGE by defining restrictive default values for the Access Control policy.
	FMT_MSA.3(b) Static attribute initialisation (Protect SFP)	This requirement supports O.MANAGE by defining restrictive default values for the Protect policy.
	FMT_MTD.1 Management of TSF data	This requirement supports O.MANAGE by restricting the users who can manage scanned data used for making security decisions.

Objective	Requirements Addressing the Objective	Rationale
	FMT_SMF.1 Specification of management functions	This requirement supports O.MANAGE by specifying the types of management functions available to users of the TOE.
	FMT_SMR.1 Security roles	This requirement supports O.MANAGE by specifying user roles and allowing the TOE to associate users with roles.
O.MONITOR The TOE must be able to monitor machines on the network to ensure that they exist in a secure state and alert TOE users if a system enters an insecure state.	FDC_ANA_EXT.1 System analysis	This requirement supports O.MONITOR by requiring the TOE to be able to analyze scanned data according to the Protect SFP and alert users when security violations are discovered.
	FDC_SCN_EXT.1 System scan	This requirement supports O.MONITOR by requiring the TOE to be able to obtain system data from monitored machines.
	FDC_STG_EXT.1 Scanned data storage	This requirement supports O.MONITOR by requiring the TOE to prevent unauthorized modification and deletion of scanned data.
	FDP_IFC.1(a) Subset information flow control (Scan Data Analysis)	This requirement supports O.MONITOR by defining the subject, operations, and information for the Protect SFP.
	FDP_IFC.1(b) Subset information flow control (Deployment)	This requirement supports O.MONITOR by defining the subject, operations, and information for the Protect SFP.
	FDP_IFF.1(a) Simple security attributes (Scan Data Analysis)	This requirement supports O.MONITOR by defining the attributes and information flow control rules for the Protect SFP.
	FDP_IFF.1(b) Simple security attributes (Deployment)	This requirement supports O.MONITOR by defining the attributes and information flow control rules for the Protect SFP.
O.ROLE The TOE must be able to associate users with the appropriate role after the user authenticates.	FIA_ATD.1 User attribute definition	This requirement supports O.ROLE by requiring the TOE to maintain a list of user identifiers and their associated roles.
	FMT_SMR.1 Security roles	This requirement supports O.ROLE by requiring the TOE to be able to associate user roles with their respective users.

8.5.2 Security Assurance Requirements Rationale

EAL 2+ was chosen to provide a low to moderate level of assurance that is consistent with good commercial practices. As such, minimal additional tasks are placed upon the vendor assuming the vendor follows reasonable software engineering practices and can provide support to the evaluation for design and testing efforts. The chosen assurance level is appropriate with the threats defined for the environment. The TOE is expected to be in a non-hostile position and embedded in or protected by other products designed to address threats that correspond with the intended environment. At EAL 2+, the TOE will have incurred a search for obvious flaws to support its introduction into the non-hostile environment.

The augmentation of ALC_FLR.2 was chosen to give greater assurance of the developer's on-going flaw remediation processes.

8.5.3 Dependency Rationale

The SFRs in this ST satisfy all of the required dependencies listed in the Common Criteria, applicable PPs, and SFRs explicitly stated in this ST. Table 19 lists each requirement to which the TOE claims conformance and indicates whether the dependent requirements are included. As the table indicates, all dependencies have been met.

Table 19 – Functional Requirements Dependencies

SFR ID	Dependencies	Dependency Met	Rationale
FAU_GEN.1	FPT_STM.1	✓	FPT_STM.1 is satisfied by the host OS providing timestamps for the TOE.
FAU_SAR.1	FAU_GEN.1	✓	
FDP_ACC.1	FDP_ACF.1	✓	
FDP_ACF.1	FMT_MSA.3	✓	
	FDP_ACC.1	✓	
FDP_ETC.2	FDP_IFC.1	✓	
	FDP_ACC.1	✓	
FDP_IFC.1(a)	FDP_IFF.1	✓	
FDP_IFC.1(b)	FDP_IFF.1	✓	
FDP_IFF.1(a)	FDP_IFC.1	✓	
	FMT_MSA.3	✓	
FDP_IFF.1(b)	FDP_IFC.1	✓	
	FMT_MSA.3	✓	
FDP_ITC.1	FDP_ACC.1	✓	
	FMT_MSA.3	✓	
FIA_ATD.1	No dependencies	N/A	
FMT_MOF.1	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MSA.1(a)	FDP_ACC.1	✓	
	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MSA.1(b)	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
	FDP_IFC.1	✓	
FMT_MSA.3(a)	FMT_MSA.1(a)	✓	
	FMT_SMR.1	✓	
FMT_MSA.3(b)	FMT_MSA.1(b)	✓	

SFR ID	Dependencies	Dependency Met	Rationale
	FMT_SMR.1	✓	
FMT_MTD.1	FMT_SMR.1	✓	
	FMT_SMF.1	✓	
FMT_SMF.1	No dependencies	N/A	
FMT_SMR.1	FIA_UID.1	✓	FIA_UID.1 is satisfied by the host OS providing user identification.
FPT_ITT.1	No dependencies	N/A	
FPT_ITT.3	FPT_ITT.1	✓	
FPT_TST.1	No dependencies	N/A	
FRU_RSA.1	No dependencies	N/A	
FDC_ANA_EXT.1	FDC_SCN_EXT.1	✓	
FDC_SCN_EXT.1	FPT_STM.1	✓	FPT_STM.1 is satisfied by the host OS providing timestamps for the TOE.
FDC_STG_EXT.1	FDC_SCN_EXT.1	✓	

9. Acronyms

Table 20 defines the acronyms used throughout this document.

Table 20 – Acronyms

Acronym	Definition
API	Application Programming Interface
CC	Common Criteria
CEM	Common Evaluation Methodology
CM	Configuration Management
CVE	Common Vulnerability Exchange
EAL	Evaluation Assurance Level
E-Mail	Electronic Mail
EXP	Extended Package
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IAVA	Information Assurance Vulnerability Alert
ID	Identifier
IP	Internet Protocol
IT	Information Technology
MN	Minnesota
OS	Operating System
OU	Organizational Unit
PDF	Portable Document Format
PP	Protection Profile
RSA	Rivest, Shamir-Adleman
SAR	Security Assurance Requirement
SFP	Security Functional Policy
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
ST	Security Target

Acronym	Definition
STIG	Security Technical Implementation Guides
STS	Security Token Service
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSS	TOE Security Specification
VA	Virginia
WMI	Windows Management Instrumentation

Prepared by:
Corsec Security, Inc.

12600 Fair Lakes Circle
Suite 210
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
Email: info@corsec.com
http://www.corsec.com