



HCL BigFix Server Version 11.0.3 Security Target

Version:	1.2
Status:	Released
Last Update:	2025-05-06
Classification:	Public
Authors:	atsec information security corporation

Trademarks

HCL, the HCL logo, BigFix, and Fixlet are registered trademarks of HCL Technologies Limited.

Microsoft, Windows, and SQL server are registered trademarks of Microsoft Corporation.

OpenSSL is a registered trademark of OpenSSL Software Foundation.

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Revision History

Version	Date	Author(s)	Changes to Previous Revision
1.1	2025-03-07	atsec	First published version.
1.2	2025-05-06	atsec	Improved based on the comments from validators.

Table of Contents

1	Introduction	9
1.1	Security Target Identification	9
1.2	TOE Identification	9
1.3	TOE Type	9
1.4	TOE Overview	9
1.5	TOE Description	10
1.5.1	BigFix platform architecture	11
1.5.1.1	BigFix Server (the TOE)	12
1.5.1.2	BigFix Console	13
1.5.1.3	BigFix Administration Tool	14
1.5.1.4	BigFix IEM CLI	14
1.5.1.5	BigFix Client	14
1.5.1.6	BigFix Relay	14
1.5.2	TOE security functionality	15
1.5.2.1	Cryptographic support	15
1.5.2.2	User data protection	15
1.5.2.3	Identification and authentication	16
1.5.2.4	Security management	16
1.5.2.5	Privacy	16
1.5.2.6	Protection of the TSF	16
1.5.2.7	Trusted path/channels	16
1.5.3	TOE boundaries	17
1.5.3.1	Physical boundary	17
1.5.3.2	Logical boundary	17
1.5.3.3	TOE evaluated configuration	18
2	CC Conformance Claim	19
3	Security Problem Definition	21
3.1	Threat Environment	21
3.1.1	Threats countered by the TOE	21
3.2	Assumptions	21
3.3	Organizational Security Policies	21
4	Security Objectives	22
4.1	Objectives for the TOE	22
4.2	Objectives for the Operational Environment	22
4.3	Security Objectives Rationale	23
5	Extended Components Definition	24
6	Security Requirements	25
6.1	TOE Security Functional Requirements	25
6.1.1	Cryptographic support (FCS)	26
6.1.1.1	FCS_CKM_EXT.1 Cryptographic Key Generation Services	26
6.1.1.2	FCS_CKM.1/AK Cryptographic Asymmetric Key Generation	26
6.1.1.3	FCS_CKM_EXT.1/PBKDF Password Conditioning	27
6.1.1.4	FCS_CKM.2 Cryptographic Key Establishment	27

6.1.1.5	FCS_COP.1/HASH Cryptographic Operation - Hashing	28
6.1.1.6	FCS_COP.1/KEYEDHASH Cryptographic Operation - Keyed-Hash Message Authentication	28
6.1.1.7	FCS_COP.1/SIG Cryptographic Operation - Signing	29
6.1.1.8	FCS_COP.1/SKC Cryptographic Operation - Encryption/Decryption	29
6.1.1.9	FCS_HTTPS_EXT.1/CLIENT HTTPS Protocol	29
6.1.1.10	FCS_HTTPS_EXT.1/SERVER HTTPS Protocol	29
6.1.1.11	FCS_RBG_EXT.1 Random Bit Generation Services	30
6.1.1.12	FCS_RBG_EXT.2 Random Bit Generation from Application	30
6.1.1.13	FCS_STO_EXT.1 Storage of Credentials	30
6.1.1.14	FCS_TLS_EXT.1 TLS Protocol	31
6.1.1.15	FCS_TLSC_EXT.1 TLS Client Protocol	31
6.1.1.16	FCS_TLSC_EXT.4 TLS Client Support for Renegotiation	32
6.1.1.17	FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension	32
6.1.1.18	FCS_TLSS_EXT.1 TLS Server Protocol	32
6.1.2	User data protection (FDP)	33
6.1.2.1	FDP_DAR_EXT.1 Encryption Of Sensitive Application Data	33
6.1.2.2	FDP_DEC_EXT.1 Access to Platform Resources	34
6.1.2.3	FDP_NET_EXT.1 Network Communications	34
6.1.3	Identification and authentication (FIA)	34
6.1.3.1	FIA_X509_EXT.1 X.509 Certificate Validation	34
6.1.3.2	FIA_X509_EXT.2 X.509 Certificate Authentication	35
6.1.4	Security management (FMT)	36
6.1.4.1	FMT_CFG_EXT.1 Secure by Default Configuration	36
6.1.4.2	FMT_MEC_EXT.1 Supported Configuration Mechanism	36
6.1.4.3	FMT_SMF.1 Specification of Management Functions	36
6.1.5	Privacy (FPR)	36
6.1.5.1	FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information	36
6.1.6	Protection of the TSF (FPT)	37
6.1.6.1	FPT_AEX_EXT.1 Anti-Exploitation Capabilities	37
6.1.6.2	FPT_API_EXT.1 Use of Supported Services and APIs	37
6.1.6.3	FPT_IDV_EXT.1 Software Identification and Versions	37
6.1.6.4	FPT_LIB_EXT.1 Use of Third Party Libraries	37
6.1.6.5	FPT_TUD_EXT.1 Integrity for Installation and Update	38
6.1.6.6	FPT_TUD_EXT.2 Integrity for Installation and Update	38
6.1.7	Trusted path/channels (FTP)	39
6.1.7.1	FTP_DIT_EXT.1 Protection of Data in Transit	39
6.2	Security Functional Requirements Rationale	39
6.3	Security Assurance Requirements	39
6.4	Security Assurance Requirements Rationale	40
7	TOE Summary Specification	41
7.1	TOE Security Functionality	41
7.1.1	Cryptographic support	41
7.1.1.1	FCS_CKM_EXT.1	43

7.1.1.2	FCS_CKM.1/AK	43
7.1.1.3	FCS_CKM_EXT.1/PBKDF	43
7.1.1.4	FCS_CKM.2	44
7.1.1.5	FCS_COP.1/HASH	44
7.1.1.6	FCS_COP.1/KEYEDHASH	44
7.1.1.7	FCS_COP.1/SIG	44
7.1.1.8	FCS_COP.1/SKC	44
7.1.1.9	FCS_RBG_EXT.1	44
7.1.1.10	FCS_RBG_EXT.2	44
7.1.1.11	FCS_STO_EXT.1	45
7.1.1.12	FCS_HTTPS_EXT.1/CLIENT	45
7.1.1.13	FCS_HTTPS_EXT.1/SERVER	45
7.1.1.14	FCS_TLS_EXT.1	46
7.1.1.15	FCS_TLSC_EXT.1	46
7.1.1.16	FCS_TLSC_EXT.4	47
7.1.1.17	FCS_TLSC_EXT.5	47
7.1.1.18	FCS_TLSS_EXT.1	47
7.1.2	User data protection	48
7.1.2.1	FDP_DAR_EXT.1	48
7.1.2.2	FDP_DEC_EXT.1	48
7.1.2.3	FDP_NET_EXT.1	48
7.1.3	Identification and authentication	49
7.1.3.1	FIA_X509_EXT.1	49
7.1.3.2	FIA_X509_EXT.2	49
7.1.4	Security management	50
7.1.4.1	FMT_CFG_EXT.1	50
7.1.4.2	FMT_MEC_EXT.1	50
7.1.4.3	FMT_SMF.1	50
7.1.5	Privacy	51
7.1.5.1	FPR_ANO_EXT.1	51
7.1.6	Protection of the TSF	51
7.1.6.1	FPT_AEX_EXT.1	51
7.1.6.2	FPT_API_EXT.1	51
7.1.6.3	FPT_IDV_EXT.1	53
7.1.6.4	FPT_LIB_EXT.1	53
7.1.6.5	FPT_TUD_EXT.1	53
7.1.6.6	FPT_TUD_EXT.2	53
7.1.7	Trusted path/channels	53
7.1.7.1	FTP_DIT_EXT.1	54
7.2	Security Assurance	54
7.2.1	Life-cycle support	54
7.2.1.1	ALC_TSU_EXT.1	54
8	Abbreviations, Terminology, and References	55
8.1	Abbreviations	55
8.2	Terminology	57

8.3 References 58

List of Tables

Table 1: NIAP Technical Decisions for [PP_APP_V1.4] 19

Table 2: NIAP Technical Decisions for [PKG_TLS_V1.1] 20

Table 3: SFRs for the TOE 25

Table 4: SARs 39

Table 5: Cryptographic algorithms implemented by the TOE (OpenSSL) and CAVP certificates .. 41

Table 6: Credential list 45

Table 7: TOE network communication paths 48

Table 8: Configuration options 50

Table 9: Windows API functions used by the TOE 51

List of Figures

Figure 1: BigFix platform architecture 12

Figure 2: Logical boundary of the TOE 17

1 Introduction

1.1 Security Target Identification

Title:	HCL BigFix Server Version 11.0.3 Security Target
Version:	1.2
Status:	Released
Date:	2025-05-06
Sponsor:	HCL Technologies Limited
Developer:	HCL Technologies Limited
Validation Body:	NIAP
Validation ID:	VID 11481
Keywords:	HCL BigFix server, HCL Technologies Limited, BigFix, Common Criteria, NIAP, PP_APP_V1.4, PKG_TLS_V1.1

1.2 TOE Identification

The TOE is HCL BigFix Server version 11.0.3.82 (referred to as 11.0.3 in this document).

1.3 TOE Type

The TOE type is Application Software.

1.4 TOE Overview

The Target of Evaluation (TOE) is the HCL BigFix Server version 11.0.3, an application software that is part of HCL BigFix Endpoint Management solution, provided by HCL Technologies Limited.

HCL BigFix Endpoint Management is a centralized endpoint management system that allows authorized users to monitor the system configurations of distributed endpoint systems (client computers) and enables users to take any necessary corrective actions.

HCL BigFix Endpoint Management utilizes a patented Fixlet® technology to identify vulnerable or misconfigured computers in the enterprise and allows authorized users to remediate identified issues across the network.

The Fixlet messages are available to an enterprise by subscribing to any of several Fixlet Sites that are maintained by HCL. Each Fixlet Site contains pre-tested, pre-packaged Fixlet messages that provide out-of-the-box management solutions. They constitute data that the BigFix system collects, distributes, and utilizes to detect and remediate vulnerabilities.

The Fixlet messages enable authorized users to perform the following functions within the enterprise:

- Analyze the vulnerability status (i.e., patched or insecure configurations).
- Distribute patches to vulnerable computers to maintain endpoint security.
- Establish and enforce configuration security policies across the network.
- Distribute and update software.
- Manage the network from a central Console.

- View, modify, and audit the properties and configurations of the enrolled endpoints.

The BigFix server is the TOE and implements the server functionality of HCL BigFix Endpoint Management solution. The TOE runs as a Windows service and provides the following features:

- Maintain a database of enrolled endpoints and subscribed software.
- Accept the enrollment of new endpoints.
- Search and gather software updates and/or endpoint configuration updates (known as Fixlets) for subscribed software from multiple Internet-based HCL Fixlet Sites.
- Download necessary software patches from the vendor sites.
- Provide management services to the Console and REST API applications to administrate endpoints, Fixlets, and corrective actions.
- Distribute Fixlets and corrective actions to enrolled endpoints.
- Receive reports and status updates from enrolled endpoints.

The TOE implements secure channels using the HTTPS protocol to protect all the information flowing between the TOE and other trusted IT products. The TOE does not implement mutual authentication.

The TOE includes the OpenSSL cryptographic module to implement the TLS protocol. The TOE implements the HTTPS Client using the cURL library which uses OpenSSL and is part of the TOE. The TOE implements the HTTPS Server using OpenSSL for TLS and the Windows Sockets (Winsock) API for lower-level sockets. Winsock is provided by the underlying platform; the OpenSSL API is provided by the OpenSSL cryptographic module part of the TOE.

The TOE supports the TLS protocol versions 1.2 and 1.3 for HTTPS implementation. This ST claims conformance to Functional Package for Transport Layer Security (TLS) Version 1.1 [PKG_TLS_V1.1], which does not cover TLS version 1.3. Therefore, TLS version 1.3 is out of scope of this evaluation and this ST focuses on TLS version 1.2.

The TOE claims compliance to the Application Software Protection Profile version 1.4 [PP_APP_V1.4]. The TOE falls under use case 3 ("Communication") scenario, described in section 1.4 of the protection profile as follows:

"The application allows for communication interactively or non-interactively with other users or applications over a communications channel. Example communications include instant messages, email, and voice."

1.5 TOE Description

The Target of Evaluation (TOE) is the HCL BigFix Server version 11.0.3, an application software that is part of HCL BigFix Endpoint Management solution, provided by HCL Technologies Limited. The TOE is installed and runs as a service on a Microsoft® Windows® operating system.

For this evaluation, the TOE runs on Microsoft Windows Server 2019 Standard version 1809, which has been evaluated for conformance with NIAP-approved Protection Profile for General Purpose Operating Systems Version 4.2.1. Microsoft Windows Server 2019 Standard version 1809 was listed on the NIAP Product Compliant List and is now archived [NIAP_PCL]. Because the TOE implements its own cryptography, CAVP certificates have been provided for the TOE implementations executing on the Microsoft Windows Server 2019 Standard version 1809 operating system platform. The platform is only used for the Data Protection API and to provide entropy to the TOE DRBG.

The remainder of this section provides an overview of the BigFix platform architecture to understand the context in which the TOE runs.

1.5.1 BigFix platform architecture

The BigFix platform is comprised of the following main components:

- BigFix Server (a.k.a. Server), the TOE.
- BigFix Administration Tool (a.k.a. Admin Tool).
- BigFix Console (a.k.a. Console).
- BigFix IEM Command-Line Interface (a.k.a. IEM CLI).
- BigFix Client (a.k.a. Client or Agent).
- BigFix Relay (a.k.a. Relay).

The evaluated configuration for the TOE includes all the components above except the BigFix Relay.

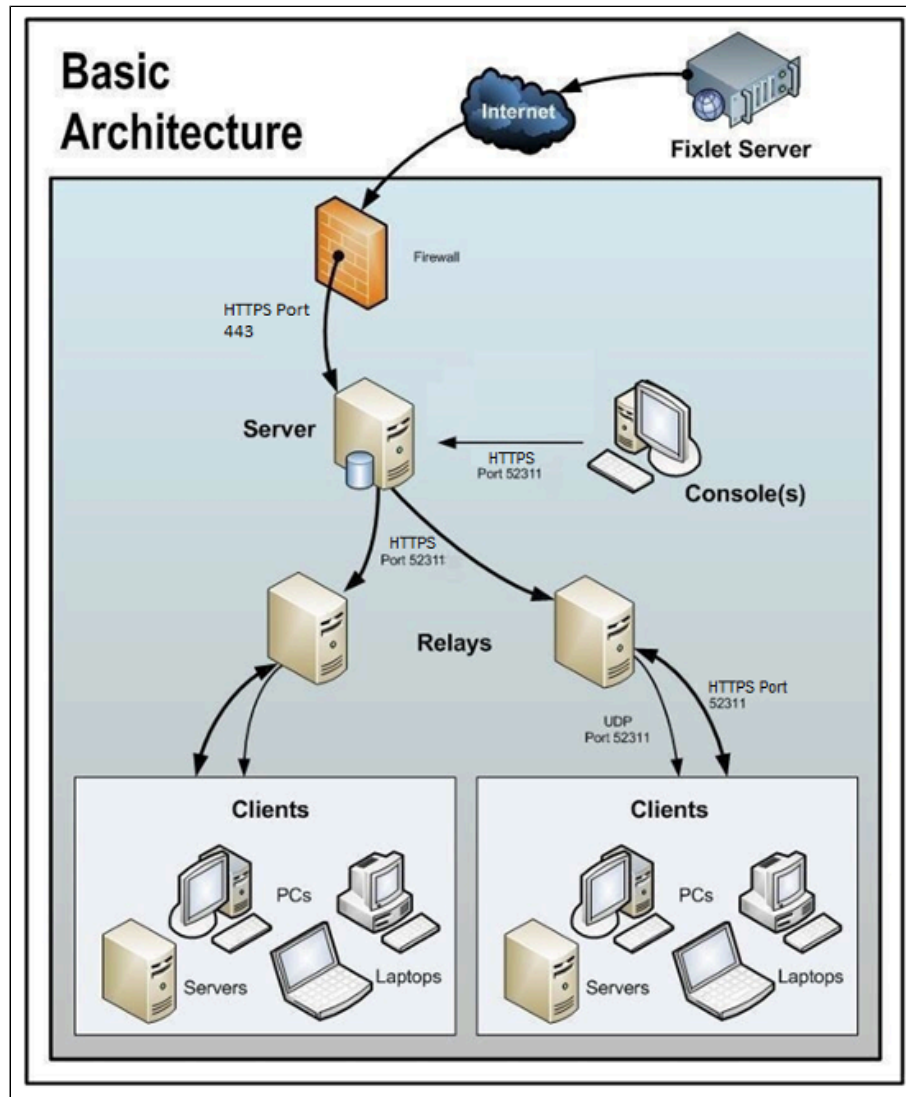
The BigFix platform provides an authorized user (known as Console Operator) the ability to

- assess the status of the Operating System (OS), applications, anti-virus signatures, and other subscribed software installed in the Client Computers (using Fixlets), and
- issue instructions for updating and remediating the Client Computers (using Actions).

The BigFix platform relies on the ability of Client Computers to periodically check with the TOE (directly or through Relays) in order to obtain the most current Fixlets and/or Actions. The TOE can also send UDP "new information" messages to the Clients (directly or through Relays) to promptly notify the Clients that there are new contents available. The UDP messages can help reduce the response time.

During the installation of the platform, the authorized Site Administrator creates a masthead file that ties the platform applications together. Among other things, this file includes a public key (signed by the Site Administrator) to authenticate any instructions from the TOE.

The diagram below depicts a typical application of the BigFix platform and an overview of its basic architecture. There is one BigFix Server (the TOE) that gathers Fixlets from HCL Fixlet Servers on the Internet. The Fixlets can be reviewed by the Console Operator and distributed to the Clients directly or through Relays. Each Client inspects its local computer environment and reports any relevant Fixlets back to the Server directly, or through a Relay which compresses the data and passes it back up to the Server. Please note that the BigFix Relay component is excluded from the evaluated configuration; the BigFix Clients communicate directly with the BigFix Server in the evaluated configuration.

Figure 1: BigFix platform architecture

The following sections describe the role of each of these applications to provide HCL BigFix Endpoint Management solution.

1.5.1.1 BigFix Server (the TOE)

The TOE is a collection of interacting services, including application services and a web server. The TOE offers the following features:

- The TOE collects contents from external Internet sites (i.e., HCL Fixlet servers and software vendor sites) and then redistributes the content to the BigFix Clients, either directly or through BigFix Relays. These mechanisms provide bandwidth advantages, as well as removing the need to configure individual BigFix Clients to connect to the Internet directly. Although it is possible to have BigFix Clients communicate directly over the Internet to download any software image required by the Fixlet (e.g., Windows patches binaries), that configuration can cause additional network traffic.

- When the Client is installed on a new Client Computer, it registers itself with the client registration component of the TOE and the Client is given a unique identifier (ID).
- When a Client detects that a Fixlet has become relevant, it reports to the TOE using an HTTPS "POST" operation. The TOE identifies the relevant Fixlet along with the registered ID of the Client Computer; this information is passed on to the database and then becomes viewable in the Console. Also, other state changes are periodically reported by the Clients to the TOE. All Client data can flow directly to the TOE or through Relays.
- The TOE monitors the changes in Fixlet content for all the Fixlet servers to which it is subscribed. If any changes are detected, they are downloaded and made available to the rest of the BigFix Platform components.

The TOE listens on a TCP port (52311 by default) for TLS/HTTPS messages from Clients and Relays. Data files containing Fixlets, Actions, or responses to Actions performed on clients are communicated between the TOE and Clients using HTTPS protected messages. The TOE can send out UDP messages to notify Clients when the new content (e.g., a Fixlet) becomes available.

The TOE also listens on a TCP port (52311 by default) for TLS/HTTPS messages from Consoles and REST API applications that connect to the TOE to perform security management functions.

The TOE uses a database to store and retrieve applicable data. The TOE manages and coordinates the flow of information to and from Clients and stores the results in the database. The database is also used by the TOE to store and retrieve applicable Fixlets and Actions as well as TOE configuration data.

The database is outside the boundary of the TOE (i.e., in the operational environment). The database can reside in a separate machine or the same machine where the TOE is installed. In the evaluated configuration, the TOE uses MSSQL database system which is installed on the same machine as the TOE. The database is expected to be configured so that only authorized users can access any contents associated with the TOE. Also, it is expected that the ODBC interface and communications are protected in a manner appropriate to the environment in which it is being used.

HCL has published guidance so that users could potentially develop their own applications to access the database, provided they have applicable database authorizations. However, the development and use of other applications to access the database, while not forbidden, is outside the scope of this evaluation.

1.5.1.2 BigFix Console

The Console provides the ability for an authorized administrator to view and manage their entire network of computers by enabling automated distribution of fixes, software deployment, vulnerability analysis (i.e., systems requiring patches, updated Service Packs, configuration violations and/or enterprise security policy violations), and remediation from a central location.

Console users, also known as Console Operators, can be in charge of flexibly defined groups of computers (running the Client) with varying degrees of freedom. The platform supports two classes of Console Operators: Master Operators, and (ordinary) Operators. A Master Operator has overall control of each Operator's domain and the specific rights they have over that domain.

The Console is invoked as an interactive application. The TOE enforces the use of TLS/HTTPS to protect the communications channel of the Console. The TOE also enforces authenticity and integrity of all Console Operators through the use of usernames and passwords. Credentials are managed by the TOE and stored in the database.

Multiple Consoles can connect to the TOE simultaneously.

1.5.1.3 BigFix Administration Tool

The BigFix Administration Tool is the tool for configuration changes and maintenance operations on the BigFix Server (TOE). This program is located on the computer hosting the TOE. On Windows, it has both a GUI (Graphical User Interface) and a CLI (Command-Line Interface).

With this tool, the Site Administrator can edit the masthead file, check the signatures of the objects in the database, enable and disable enhanced security, clear historical data, rotate the server private key, configure the Console and synchronize the masthead with the updated license.

1.5.1.4 BigFix IEM CLI

The TOE provides a REST (REpresentational State Transfer) API, which allows for using a set of standardized and operating system independent methods to perform the majority of the tasks available in the BigFix Console.

The IEM Command-Line Interface (CLI) is a utility that facilitates programmatic control of the TOE using the REST API. It is a lightweight wrapper for user authentication, session management, HTTPS request generation, and response parsing.

In the evaluated configuration, the TOE enforces the use of TLS/HTTPS to protect the communications channel of the REST API. The TOE also enforces authenticity and integrity of all REST API users through the use of usernames and passwords. Credentials are managed by the TOE and stored in the database.

1.5.1.5 BigFix Client

Clients are installed on every computer (personal computer, server, workstation, desktop, laptop, etc.) within the enterprise that will be managed by the TOE. Clients are also referred to as Agents and these terms are interchangeable.

After the Client is installed on a machine, the machine has to register to the TOE for the first time to become a Client Computer. Once registered, the Client Computer is known to the BigFix platform and can operate to obtain updates. A renewal of the Client Computer registration may occur per a Console Operator request.

Clients access a collection of Fixlet messages that detect security holes, vulnerabilities, and other configuration issues and Action messages capable of implementing corrective actions received from the TOE. In most cases, the Client operates silently in the background so that users are not aware of what actions are taking place on their system; however, when an action requires user input, the Console Operator is able to provide friendly screen prompts for the user.

The Clients listen on a UDP port (default 52311) for notifications from the TOE or Relays indicating that updated data is available for retrieval. The Clients use HTTPS to connect to Relays and/or Servers in order to request or renew a registration, retrieve Fixlets and Actions, and send back results of applying Fixlets and Actions.

1.5.1.6 BigFix Relay

Relays can increase the efficiency of the BigFix platform. Instead of forcing each networked computer to directly access the TOE, Relays can be installed on any computer within the enterprise to distribute the workload by storing and forwarding data (i.e., messages) passing between Servers and Clients. Relays query the TOE (or another Relay) for Fixlet and Action messages and Client machines connect to Relays in the same manner as they would do with the TOE.

Relays are considered optional in the BigFix platform, that is, they are not required for the operation but are available as part of the product and so can be installed and enabled for use. In the CC evaluated configuration, BigFix Relays were not tested and, therefore, are not allowed.

Relays listen on a TCP port (52311 by default) for TLS/HTTPS messages from Clients, and other Relays, so that they can establish connections to Clients, and then in turn connect to a TCP port on a Server or another Relay in a chain in order to forward TLS/HTTPS messages appropriately. Similarly, Relays proxy a UDP port (default 52311) so that notifications from Servers regarding availability of updated content can be forwarded and acted upon by the Relay so that it can store and forward the updates to minimize network traffic to the extent it can.

The UDP messages are used to send update notifications to Clients earlier than their individual schedules might allow. The unreliable nature of UDP is not considered to be especially important given that it will take time to distribute updates in a large enterprise regardless. TOE users can mitigate any perceived issue by configuring the Client polling interval to be as short as necessary.

1.5.2 TOE security functionality

The TOE provides the security functionality required by [PP_APP_V1.4] and [PKG_TLS_V1.1], which is described briefly in the following sections.

1.5.2.1 Cryptographic support

The TOE provides cryptographic support using the OpenSSL cryptographic module that is bundled in the TOE, and Windows Cryptography API: Next Generation (CNG), which is provided by the underlying Windows platform.

The TOE uses the OpenSSL cryptographic module for the following security functionality:

- Trusted channels for incoming and outgoing connections using the TLS protocol version 1.2.
- Conditioning of passwords for storing credentials (Console Operator's passwords).

The TOE uses the Windows CNG for the following security functionality:

- Protect private keys and database credentials using the Data Protection Application Programming Interface (DPAPI).
- Provide entropy to the SP800-90A compliant DRBG implemented in the OpenSSL cryptographic module.

The SP800-90A DRBG implemented in the TOE by the OpenSSL cryptographic module obtains the seed data from the underlying platform by calling the BCryptGenRandom API function. The seed data has a minimum of 256 bits of entropy. The Entropy Assessment Report [BIGFIX_EAR] provides a detailed description of the entropy source.

1.5.2.2 User data protection

The application provides user data protection by protecting sensitive data at rest, as well as restricting access to only those platform-based resources that are needed in order to provide the required functionality.

The following information is considered sensitive data:

- The private key corresponding to the server certificate used for the TLS communication.
- The private key corresponding to the CA certificate used for signing Client certificates. Each Client is assigned a certificate to prove its identity and that certificate is signed using the private key of the CA certificate. This feature is implemented outside the TLS protocol (i.e., it is not TLS mutual authentication) at the application layer. It is also beyond the security requirements defined in Protection Profiles and Functional Packages to which the ST claims conformance. Therefore, this feature is out of scope of this evaluation.
- Passwords of Console Operators.

The TOE uses cryptographic functionality implemented in the TOE or in the underlying platform to protect sensitive data.

Access to network communication is restricted to the application for:

- HTTPS incoming requests from the Console, REST API applications, and Clients;
- HTTPS requests initiated by the TOE to connect to Fixlet servers and vendor sites; and
- UDP messages to notify Clients that new information is available for download.

HTTPS uses the TLS version 1.2 protocol for establishing the secure channel.

1.5.2.3 Identification and authentication

The TOE authenticates the identity of the endpoint server when connecting as a TLS client by validating the X.509 certificates received from the server during the TLS protocol handshake. The TOE uses the cURL library and the OpenSSL cryptographic module, which are part of the TOE.

The TOE also authenticates Console Operators for the purpose of managing the TOE from the Console or the REST API. The Console Operator credential consists of a username and a password, both stored in the database. The password is conditioned using the PBKDF algorithm.

1.5.2.4 Security management

The TOE provides the ability to set various configuration options for communication paths. These options are stored, as recommended by Microsoft, in the Windows Registry.

The TOE also provides security functionality to create Console Operators including, among other information, a username and a password. Console Operators can change their own passwords.

Console Operators can also verify the name and version of the TOE and check whether TOE updates are available for download.

Security Management functions can be triggered via the Console as well as the REST API.

1.5.2.5 Privacy

The TOE does not specifically request Personally Identifiable Information (PII).

1.5.2.6 Protection of the TSF

The TOE uses only documented Windows APIs. The TOE is also packaged with third-party libraries which provide additional functionality. These are listed in [section 6.1.6.4](#).

The TOE does not write user-modifiable files to directories that contain executable files.

The TOE is compiled using stack buffer overrun protection and Address Space Layout Randomization (ASLR) techniques. The TOE does not request to map memory at explicit addresses.

All TOE binaries are signed using the Microsoft Authenticode process. The TOE is delivered as an InstallShield installation package, signed by HCL America Inc.

1.5.2.7 Trusted path/channels

The TOE protects all incoming and outgoing transmitted data by using trusted channels with HTTPS, using the TLS version 1.2 as the underlying protocol. The TOE implements the TLS protocol using the OpenSSL cryptographic module which is part of the TOE.

1.5.3 TOE boundaries

1.5.3.1 Physical boundary

The physical boundary of the TOE consists of the application installer executable and guidance documentation.

The TOE installer is bundled in the BigFix platform installation package that can be obtained from the BigFix Enterprise Suite Download Center at [\[BIGFIX_DOWNLOAD\]](#). The TOE also includes the TOE guidance listed below, which provides information for installing, configuring, and maintaining the evaluated configuration:

- HCL BigFix Server Version 11.0.3 Common Criteria Configuration Guide [\[CCGUIDE\]](#)

The hardware platform used during the evaluation was a Dell PowerEdge R430 with Intel Broadwell Xeon E5-2620 v4 processor.

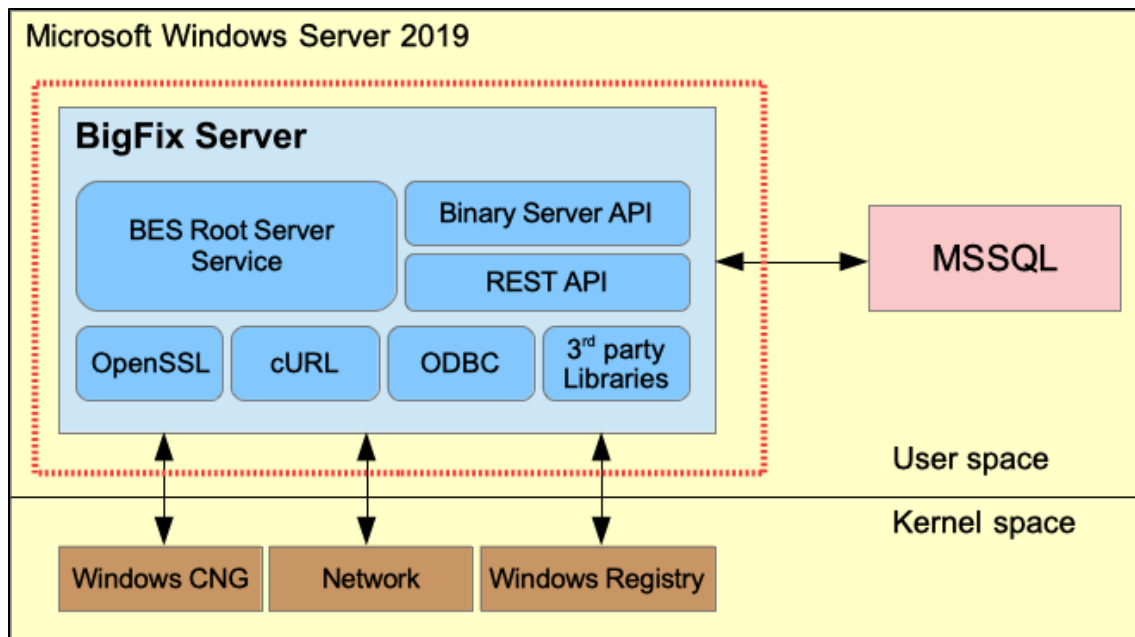
For the evaluated configuration, the TOE requires the following software components installed in the same machine:

- Microsoft Windows Server 2019 Standard version 1809
- Microsoft SQL Server 2019

1.5.3.2 Logical boundary

The diagram below depicts the logical boundary of the TOE in the dotted red box. The TOE includes several third party libraries, such as cURL and OpenSSL. The TOE also interacts with the underlying operating system to access the network, the Windows Registry, and request cryptographic services from Windows Cryptography API: Next Generation (CNG).

Figure 2: Logical boundary of the TOE



1.5.3.3 TOE evaluated configuration

In the evaluated configuration, the Operational Environment where the TOE runs is restricted to the following BigFix components:

- BigFix Administration Tool 11.0.3 (in the same machine where the TOE runs)
- BigFix Console 11.0.3
- BigFix IEM CLI 11.0.3
- BigFix Client 11.0.3

The following BigFix components and features are not allowed in the evaluated configuration:

- BigFix Relay
- BigFix Web Reports
- BigFix WebUI
- BigFix Explorer
- BigFix Asset Discovery
- Disaster Server Architecture (DSA)

The following constraints apply:

- The TOE must be configured to use "FIPS mode".
- The MSSQL database must reside in the same system as the TOE.

The specifications for configuring the TOE in the evaluated configuration are located in the TOE guidance documentation listed in [section 1.5.3.1](#). The consumer must read, understand, and follow the guidance documentation provided as part of the TOE for the evaluated configuration.

2 CC Conformance Claim

This Security Target (ST) is CC Part 2 extended and CC Part 3 extended. Common Criteria [CC] version 3.1 revision 5 is the basis for this conformance claim.

This ST claims exact conformance to the following Protection Profiles (PPs) and Functional Packages:

- [PP_APP_V1.4][\[1\]](#): Protection Profile for Application Software. Version 1.4 as of 2021-10-07.
- [PKG_TLS_V1.1][\[2\]](#): Functional Package for Transport Layer Security (TLS). Version 1.1 as of 2019-03-01.

Table 1 below contains the NIAP Technical Decisions (TDs) for the [PP_APP_V1.4][\[1\]](#) protection profile at the time of the evaluation and a statement of applicability to the evaluation. TDs are marked as applicable if any of the documentation or evaluation activities included in this evaluation were updated, even if these evaluation activities were not performed. The applicability of a TD does not imply that the related security functionality is claimed or not claimed by the TOE.

Table 1: NIAP Technical Decisions for [PP_APP_V1.4]

TD #	Description	Applicable?	Non-applicability rationale
TD0893	Addition of Recommended Configuration Locations for Windows in FMT_MEC_EXT.1.1	Yes	
TD0865	Consistency of Cryptographic Key Sizes	Yes	
TD0860	Updating FIPS 186-4 to 186-5 in PP_APP_V1.4	Yes	
TD0844	Addition of Assurance Package for Flaw Remediation V1.0 Conformance Claim	No	The ST does not claim conformance to Assurance Package for Flaw Remediation V1.0.
TD0823	Update to Microsoft Windows Exploit Protection link in FPT_AEX_EXT.1.3	Yes	
TD0822	Correction to Windows Manifest File for FDP_DEC_EXT.1	Yes	
TD0815	Addition of Conditional TSS Activity for FPT_AEX_EXT.1.5	Yes	
TD0798	Static Memory Mapping Exceptions	Yes	
TD0780	FIA_X509_EXT.1 Test 4 Clarification	Yes	
TD0756	Update for platform-provided full disk encryption	Yes	
TD0747	Configuration Storage Option for Android	No	The TOE is not for Android platform.
TD0743	FTP_DIT_EXT.1.1 Selection exclusivity	Yes	
TD0736	Number of elements for iterations of FCS_HTTPS_EXT.1	Yes	
TD0719	ECD for PP APP V1.3 and 1.4	Yes	
TD0717	Format changes for PP_APP_V1.4	Yes	
TD0664	Testing activity for FPT_TUD_EXT.2.2	Yes	
TD0650	Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	No	The ST does not claim conformance to MOD_VPNC_V2.3/2.4.

TD #	Description	Applicable?	Non-applicability rationale
TD0628	Addition of Container Image to Package Format	Yes	

Table 2 contains the NIAP Technical Decisions (TDs) for the [PKG_TLS_V1.1] functional package at the time of the evaluation and a statement of applicability to the evaluation. TDs are marked as applicable if any of the documentation or evaluation activities included in this evaluation were updated, even if these evaluation activities were not performed. The applicability of a TD does not imply that the related security functionality is claimed or not claimed by the TOE.

Table 2: NIAP Technical Decisions for [PKG_TLS_V1.1]

TD #	Description	Applicable?	Non-applicability rationale
TD0779	Updated Session Resumption Support in TLS package V1.1	Yes	
TD0770	TLSS.2 connection with no client cert	No	The ST does not claim selection-based FCS_TLSS_EXT.2.
TD0739	PKG_TLS_V1.1 has 2 different publication dates	Yes	
TD0726	Corrections to (D)TLSS SFRs in TLS 1.1 FP	Yes	
TD0513	CA Certificate loading	Yes	
TD0499	Testing with pinned certificates	Yes	
TD0469	Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	Yes	
TD0442	Updated TLS Ciphersuites for TLS Package	Yes	

3 Security Problem Definition

3.1 Threat Environment

3.1.1 Threats countered by the TOE

T.LOCAL_ATTACK

An attacker can act through unprivileged software on the same computing platform on which the application executes. Attackers may provide maliciously formatted input to the application in the form of files or other local communications.

T.NETWORK_ATTACK

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter communications between the application software and other endpoints in order to compromise it.

T.NETWORK_EAVESDROP

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the application and other endpoints.

T.PHYSICAL_ACCESS

An attacker may try to access sensitive data at rest.

3.2 Assumptions

A.PLATFORM

The TOE relies upon a trustworthy computing platform with a reliable time clock for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.

A.PROPER_ADMIN

The administrator of the application software is not careless, willfully negligent or hostile, and administers the software in compliance with the applied enterprise security policy.

A.PROPER_USER

The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.

3.3 Organizational Security Policies

No Organizational Security Policies (OSPs) are part of the Security Problem Definition described in [PP_APP_V1.4].

4 Security Objectives

4.1 Objectives for the TOE

O.INTEGRITY

Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom, if ever, shipped without errors. The ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options.

O.MANAGEMENT

To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII.

O.PROTECTED_COMMS

To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application.

O.PROTECTED_STORAGE

To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data.

O.QUALITY

To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs.

4.2 Objectives for the Operational Environment

OE.PLATFORM

The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.

OE.PROPER_ADMIN

The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

OE.PROPER_USER

The user of the application software is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.

4.3 Security Objectives Rationale

The security objectives rationale is defined in the [PP_APP_V1.4][📄](#) protection profile.

5 Extended Components Definition

This Security Target claims exact conformance to [PP_APP_V1.4] and [PKG_TLS_V1.1]; therefore, it does not extend the security requirements defined by these documents.

6 Security Requirements

6.1 TOE Security Functional Requirements

The table below summarizes the SFRs for the TOE and the operations performed on the components according to CC part 1. Operations in the SFRs use the following convention:

- Iterations (Iter.) are identified by appending a suffix to the original SFR.
- Refinements (Ref.) added to the text are shown in *italic text*, deletions are shown as ~~strikethrough text~~.
- Assignments (Ass.) are shown in **bold text**.
- Selections (Sel.) are shown in **bold text**.

Table 3: SFRs for the TOE

Security functional class	Security functional requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
FCS - Cryptographic support	FCS_CKM_EXT.1 Cryptographic Key Generation Services	PP_APP_V1.4	No	No	No	Yes
	FCS_CKM.1/AK Cryptographic Asymmetric Key Generation	PP_APP_V1.4	No	No	No	Yes
	FCS_CKM_EXT.1/PBKDF Password Conditioning	PP_APP_V1.4	No	No	Yes	Yes
	FCS_CKM.2 Cryptographic Key Establishment	PP_APP_V1.4	No	No	No	Yes
	FCS_COP.1/HASH Cryptographic Operation - Hashing	PP_APP_V1.4	No	No	No	Yes
	FCS_COP.1/KEYEDHASH Cryptographic Operation - Keyed-Hash Message Authentication	PP_APP_V1.4	No	No	Yes	Yes
	FCS_COP.1/SIG Cryptographic Operation - Signing	PP_APP_V1.4	No	No	No	Yes
	FCS_COP.1/SKC Cryptographic Operation - Encryption/Decryption	PP_APP_V1.4	No	No	No	Yes
	FCS_HTTPS_EXT.1/CLIENT HTTPS Protocol	PP_APP_V1.4	No	No	No	Yes
	FCS_HTTPS_EXT.1/SERVER HTTPS Protocol	PP_APP_V1.4	No	No	No	Yes
	FCS_RBG_EXT.1 Random Bit Generation Services	PP_APP_V1.4	No	No	No	Yes
	FCS_RBG_EXT.2 Random Bit Generation from Application	PP_APP_V1.4	No	No	No	Yes
	FCS_STO_EXT.1 Storage of Credentials	PP_APP_V1.4	No	No	Yes	Yes
	FCS_TLS_EXT.1 TLS Protocol	PKG_TLS_V1.1	No	No	No	Yes
	FCS_TLSC_EXT.1 TLS Client Protocol	PKG_TLS_V1.1	No	No	No	Yes
	FCS_TLSC_EXT.4 TLS Client Support for Renegotiation	PKG_TLS_V1.1	No	No	No	No
	FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension	PKG_TLS_V1.1	No	No	No	Yes
	FCS_TLSS_EXT.1 TLS Server Protocol	PKG_TLS_V1.1	No	No	No	Yes

Security functional class	Security functional requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
FDP - User data protection	FDP_DAR_EXT.1 Encryption Of Sensitive Application Data	PP_APP_V1.4	No	No	No	Yes
	FDP_DEC_EXT.1 Access to Platform Resources	PP_APP_V1.4	No	No	Yes	Yes
	FDP_NET_EXT.1 Network Communications	PP_APP_V1.4	No	No	Yes	Yes
FIA - Identification and authentication	FIA_X509_EXT.1 X.509 Certificate Validation	PP_APP_V1.4	No	No	No	Yes
	FIA_X509_EXT.2 X.509 Certificate Authentication	PP_APP_V1.4	No	No	No	Yes
FMT - Security management	FMT_CFG_EXT.1 Secure by Default Configuration	PP_APP_V1.4	No	No	No	No
	FMT_MEC_EXT.1 Supported Configuration Mechanism	PP_APP_V1.4	No	No	No	Yes
	FMT_SMF.1 Specification of Management Functions	PP_APP_V1.4	No	No	Yes	Yes
FPR - Privacy	FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information	PP_APP_V1.4	No	No	No	Yes
FPT - Protection of the TSF	FPT_AEX_EXT.1 Anti-Exploitation Capabilities	PP_APP_V1.4	No	No	Yes	Yes
	FPT_API_EXT.1 Use of Supported Services and APIs	PP_APP_V1.4	No	No	No	No
	FPT_IDV_EXT.1 Software Identification and Versions	PP_APP_V1.4	No	No	No	Yes
	FPT_LIB_EXT.1 Use of Third Party Libraries	PP_APP_V1.4	No	No	Yes	No
	FPT_TUD_EXT.1 Integrity for Installation and Update	PP_APP_V1.4	No	No	No	Yes
	FPT_TUD_EXT.2 Integrity for Installation and Update	PP_APP_V1.4	No	No	No	Yes
FTP - Trusted path/channels	FTP_DIT_EXT.1 Protection of Data in Transit	PP_APP_V1.4	No	No	Yes	Yes

6.1.1 Cryptographic support (FCS)

6.1.1.1 FCS_CKM_EXT.1 Cryptographic Key Generation Services

Origin: PP_APP_V1.4

Applied TDs: [TD0717](#)

FCS_CKM_EXT.1.1 The application shall

- **implement asymmetric key generation**

TSS Link: [TSS for FCS_CKM_EXT.1](#)

6.1.1.2 FCS_CKM.1/AK Cryptographic Asymmetric Key Generation

Origin: PP_APP_V1.4

Applied TDs: [TD0717](#), [TD0860](#)

- FCS_CKM.1.1/AK** The application shall
- **implement functionality** to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm
 - **[ECC schemes] using ["NIST curves" P-384 and P-256, P-521] that meet the following: [FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix B.4]**
 - **[FFC schemes] using cryptographic key sizes of [2048-bit or greater] that meet the following: [FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix B.1]**
 - **[FFC Schemes] using ["safe-prime" groups] that meet the following: [NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and RFC 7919]**

TSS Link: [TSS for FCS_CKM.1/AK](#)

6.1.1.3 FCS_CKM_EXT.1/PBKDF Password Conditioning

Origin: PP_APP_V1.4

Applied TDs: [TD0717](#), [TD0865](#)

- FCS_CKM_EXT.1.1/PBKDF** A password/passphrase shall perform **Password-based Key Derivation (PBKDF) using HMAC-SHA-256** in accordance with a specified cryptographic algorithm as specified in FCS_COP.1/KeyedHash, with **2,000** iterations, and output sizes **512 bits** that meet the following [NIST SP 800-132].

- FCS_CKM_EXT.1.2/PBKDF** The TSF shall generate salts using a RBG that meets FCS_RGB_EXT.1 and with entropy corresponding to the security strength selected for PBKDF in FCS_CKM_EXT.1.1/PBKDF.

TSS Link: [TSS for FCS_CKM_EXT.1/PBKDF](#)

6.1.1.4 FCS_CKM.2 Cryptographic Key Establishment

Origin: PP_APP_V1.4

- FCS_CKM.2.1** The application shall **implement functionality** to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:
- **[RSA-based key establishment schemes] that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1"**
 - **[Elliptic curve-based key establishment schemes] that meets the following: [NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"]**

- **[Finite field-based key establishment schemes] that meets the following: [NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"]**
- **[FFC Schemes using "safe-prime" groups] that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and RFC 7919 .**

TSS Link: [TSS for FCS_CKM.2](#)

6.1.1.5 FCS_COP.1/HASH Cryptographic Operation - Hashing

Origin: PP_APP_V1.4

Applied TDs: [TD0717](#)

FCS_COP.1.1/HASH The application shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm

- **SHA-1**
- **SHA-256**
- **SHA-384**
- **SHA-512**

and message digest sizes

- **160**
- **256**
- **384**
- **512**

bits that meet the following: [FIPS Pub 180-4].

TSS Link: [TSS for FCS_COP.1/HASH](#)

6.1.1.6 FCS_COP.1/KEYEDHASH Cryptographic Operation - Keyed-Hash Message Authentication

Origin: PP_APP_V1.4

Applied TDs: [TD0717](#)

FCS_COP.1.1/KEYEDHASH The application shall perform [keyed-hash message authentication] in accordance with a specified cryptographic algorithm

- **HMAC-SHA-256**
- **HMAC-SHA-384**

and

- **HMAC-SHA-1**

with key sizes **160, 256, and 384 bits** and message digest sizes **256, 384** and **160** bits that meet the following: [FIPS Pub 198-1 "The Keyed-Hash Message Authentication Code" and FIPS Pub 180-4 "Secure Hash Standard"].

TSS Link: [TSS for FCS_COP.1/KEYEDHASH](#)

6.1.1.7 FCS_COP.1/SIG Cryptographic Operation - Signing

Origin: PP_APP_V1.4

Applied TDs: [TD0717](#), [TD0860](#)

- FCS_COP.1.1/SIG** The application shall perform [cryptographic signature services (generation and verification)] in accordance with a specified cryptographic algorithm
- **RSA schemes using cryptographic key sizes of [2048-bit or greater] that meet the following: [FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5]**
 - **ECDSA schemes using ["NIST curves" P-256, P-384 and P-521] that meet the following: [FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6]**

TSS Link: [TSS for FCS_COP.1/SIG](#)

6.1.1.8 FCS_COP.1/SKC Cryptographic Operation - Encryption/Decryption

Origin: PP_APP_V1.4

Applied TDs: [TD0717](#)

- FCS_COP.1.1/SKC** The application shall perform [encryption/decryption] in accordance with a specified cryptographic algorithm
- **AES-CBC (as defined in NIST SP 800-38A) mode**
 - **AES-GCM (as defined in NIST SP 800-38D) mode**
- and cryptographic key sizes **128-bit, 256-bit**.

TSS Link: [TSS for FCS_COP.1.1/SKC](#)

6.1.1.9 FCS_HTTPS_EXT.1/CLIENT HTTPS Protocol

Origin: PP_APP_V1.4

FCS_HTTPS_EXT.1.1/CLIENT The application shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2/CLIENT The application shall implement HTTPS using TLS as defined in the Functional Package for TLS.

FCS_HTTPS_EXT.1.3/CLIENT The application shall **not establish the application-initiated connection** if the peer certificate is deemed invalid.

TSS Link: [TSS for FCS_HTTPS_EXT.1/CLIENT](#)

6.1.1.10 FCS_HTTPS_EXT.1/SERVER HTTPS Protocol

Origin: PP_APP_V1.4

Applied TDs: [TD0736](#)

FCS_HTTPS_EXT.1.1/SERVER The application shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.3/SERVER The application shall implement HTTPS using TLS as defined in the Functional Package for TLS.

FCS_HTTPS_EXT.1.3/SERVER The application shall **not establish the connection** if the peer certificate is deemed invalid.

TSS Link: [TSS for FCS_HTTPS_EXT.1/SERVER](#)

6.1.1.11 FCS_RBG_EXT.1 Random Bit Generation Services

Origin: PP_APP_V1.4

FCS_RBG_EXT.1.1 The application shall

- **implement DRBG functionality** for its cryptographic operations.

TSS Link: [TSS for FCS_RBG_EXT.1](#)

6.1.1.12 FCS_RBG_EXT.2 Random Bit Generation from Application

Origin: PP_APP_V1.4

FCS_RBG_EXT.2.1 The application shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using **CTR_DRBG (AES)**.

FCS_RBG_EXT.2.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and

- **a software-based noise source**
- **a hardware-based noise source**

with a minimum of

- **256 bits**

of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

TSS Link: [TSS for FCS_RBG_EXT.2](#)

6.1.1.13 FCS_STO_EXT.1 Storage of Credentials

Origin: PP_APP_V1.4

Applied TDs: [TD0865](#)

FCS_STO_EXT.1.1 The application shall

- **invoke the functionality provided by the platform to securely store**
 - **Private key of the TOE Signing Certificate**
 - **Private key of the CA certificate**
- **implement functionality to securely store**
 - **Console Operator password**

according to **FCS_CKM_EXT.1/PBKDF**
to non-volatile memory.

TSS Link: [TSS for FCS_STO_EXT.1](#)

6.1.1.14 FCS_TLS_EXT.1 TLS Protocol

Origin: PKG_TLS_V1.1

FCS_TLS_EXT.1.1 The product shall implement

- **TLS as a client**
- **TLS as a server**

Application Note: *The TOE supports session renegotiation when acting as a TLS client but not when acting as a TLS server.*

TSS Link: [TSS for FCS_TLS_EXT.1](#)

6.1.1.15 FCS_TLSC_EXT.1 TLS Client Protocol

Origin: PKG_TLS_V1.1

Applied TDs: [TD0442](#)

FCS_TLSC_EXT.1.1 The product shall implement TLS 1.2 (RFC 5246) and **no earlier TLS versions** as a client that supports the cipher suites

- **TLS_RSA_WITH_AES_128_CBC_SHA** as defined in RFC 5246
- **TLS_RSA_WITH_AES_256_CBC_SHA** as defined in RFC 5246
- **TLS_RSA_WITH_AES_128_CBC_SHA256** as defined in RFC 5246
- **TLS_RSA_WITH_AES_256_CBC_SHA256** as defined in RFC 5246
- **TLS_RSA_WITH_AES_128_GCM_SHA256** as defined in RFC 5288
- **TLS_RSA_WITH_AES_256_GCM_SHA384** as defined in RFC 5288
- **TLS_DHE_RSA_WITH_AES_128_CBC_SHA256** as defined in RFC 5246
- **TLS_DHE_RSA_WITH_AES_256_CBC_SHA256** as defined in RFC 5246
- **TLS_DHE_RSA_WITH_AES_128_GCM_SHA256** as defined in RFC 5288
- **TLS_DHE_RSA_WITH_AES_256_GCM_SHA384** as defined in RFC 5288
- **TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256** as defined in RFC 5289
- **TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** as defined in RFC 5289
- **TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384** as defined in RFC 5289
- **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384** as defined in RFC 5289
- **TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256** as defined in RFC 5289
- **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256** as defined in RFC 5289

- **TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384** as defined in RFC 5289
- **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384** as defined in RFC 5289

and also supports functionality for

- **session renegotiation**

.

FCS_TLSC_EXT.1.2 The product shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3 The product shall not establish a trusted channel if the server certificate is invalid

- **with no exceptions**

.

TSS Link: [TSS for FCS_TLSC_EXT.1](#)

6.1.1.16 FCS_TLSC_EXT.4 TLS Client Support for Renegotiation

Origin: PKG_TLS_V1.1

FCS_TLSC_EXT.4.1 The product shall support secure renegotiation through use of the "renegotiation_info" TLS extension in accordance with RFC 5746.

TSS Link: [TSS for FCS_TLSC_EXT.4](#)

6.1.1.17 FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension

Origin: PKG_TLS_V1.1

FCS_TLSC_EXT.5.1 The product shall present the Supported Groups Extension in the Client Hello with the supported groups

- **secp256r1**
- **secp384r1**
- **secp521r1**
- **ffdhe2048(256)**
- **ffdhe3072(257)**
- **ffdhe4096(258)**
- **ffdhe6144(259)**
- **ffdhe8192(260)**

.

TSS Link: [TSS for FCS_TLSC_EXT.5](#)

6.1.1.18 FCS_TLSS_EXT.1 TLS Server Protocol

Origin: PKG_TLS_V1.1

Applied TDs: [TD0442](#), [TD0726](#), [TD0779](#)

- FCS_TLSS_EXT.1.1** The product shall implement TLS 1.2 (RFC 5246) and **no earlier TLS versions** as a server that supports the cipher suites
- **TLS_DHE_RSA_WITH_AES_128_CBC_SHA256** as defined in RFC 5246
 - **TLS_DHE_RSA_WITH_AES_256_CBC_SHA256** as defined in RFC 5246
 - **TLS_DHE_RSA_WITH_AES_128_GCM_SHA256** as defined in RFC 5288
 - **TLS_DHE_RSA_WITH_AES_256_GCM_SHA384** as defined in RFC 5288
 - **TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256** as defined in RFC 5289
 - **TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256** as defined in RFC 5289
 - **TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384** as defined in RFC 5289
 - **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384** as defined in RFC 5289
 - **TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256** as defined in RFC 5289
 - **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256** as defined in RFC 5289
 - **TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384** as defined in RFC 5289
 - **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384** as defined in RFC 5289
- and also supports functionality for
- **session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2)**
 - **session resumption based on session tickets according to RFC 5077**
- , and
- **none**
- .

FCS_TLSS_EXT.1.2 The product shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and **TLS 1.1**.

- FCS_TLSS_EXT.1.3** The product shall perform key establishment for TLS using
- **Diffie-Hellman parameters with size 3072 bits and no other sizes**
 - **ECDHE parameters using elliptic curves secp256r1, secp384r1, secp521r1 and no other curves**
- .

TSS Link: [TSS for FCS_TLSS_EXT.1](#)

6.1.2 User data protection (FDP)

6.1.2.1 FDP_DAR_EXT.1 Encryption Of Sensitive Application Data

Origin: PP_APP_V1.4

- FDP_DAR_EXT.1.1** The application shall
- **protect sensitive data in accordance with FCS_STO_EXT.1** in non-volatile memory.

TSS Link: *TSS for FDP_DAR_EXT.1*

6.1.2.2 FDP_DEC_EXT.1 Access to Platform Resources

Origin: PP_APP_V1.4

- FDP_DEC_EXT.1.1** The application shall restrict its access to
- **network connectivity**
 - .

- FDP_DEC_EXT.1.2** The application shall restrict its access to
- **Windows Registry**
 - **MSSQL database**
 - .

TSS Link: *TSS for FDP_DEC_EXT.1*

6.1.2.3 FDP_NET_EXT.1 Network Communications

Origin: PP_APP_V1.4

- FDP_NET_EXT.1.1** The application shall restrict network communication to
- **respond to communication initiated from the following endpoints:**
 - **BigFix Console**
 - **REST API applications**
 - **BigFix Clients**
 - **communication initiated by the TOE to connect to the following endpoints**
 - **Fixlet servers**
 - **Vendor sites**
 - **BigFix Clients**
 - **OCSP responders**
 - .

TSS Link: *TSS for FDP_NET_EXT.1*

6.1.3 Identification and authentication (FIA)

6.1.3.1 FIA_X509_EXT.1 X.509 Certificate Validation

Origin: PP_APP_V1.4

- FIA_X509_EXT.1.1** The application shall **implement functionality** to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field
- The application shall validate the revocation status of the certificate using **OCSP as specified in RFC 6960, an OCSP TLS Status Request Extension (OCSP stapling) as specified in RFC 6066.**
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
 - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
 - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the EKU field.
 - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
 - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
 - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
 - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

FIA_X509_EXT.1.2 The application shall treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

TSS Link: [TSS for FIA_X509_EXT.1](#)

6.1.3.2 FIA_X509_EXT.2 X.509 Certificate Authentication

Origin: PP_APP_V1.4

FIA_X509_EXT.2.1 The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for **HTTPS, TLS**.

FIA_X509_EXT.2.2 When the application cannot establish a connection to determine the validity of a certificate, the application shall **not accept the certificate**.

TSS Link: [TSS for FIA_X509_EXT.2](#)

6.1.4 Security management (FMT)

6.1.4.1 FMT_CFG_EXT.1 Secure by Default Configuration

Origin: PP_APP_V1.4

FMT_CFG_EXT.1.1 The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

FMT_CFG_EXT.1.2 The application shall be configured by default with file permissions which protect the application binaries and data files from modification by normal unprivileged users.

TSS Link: [TSS for FMT_CFG_EXT.1](#)

6.1.4.2 FMT_MEC_EXT.1 Supported Configuration Mechanism

Origin: PP_APP_V1.4

FMT_MEC_EXT.1.1 The application shall **invoke the mechanisms recommended by the platform vendor for storing and setting configuration options**

TSS Link: [TSS for FMT_MEC_EXT.1](#)

6.1.4.3 FMT_SMF.1 Specification of Management Functions

Origin: PP_APP_V1.4

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

- **Create Console Operators.**
- **Change the password of the logged-in Console Operator.**
- **Obtain TOE name and version.**
- **Verify updates for the TOE.**

.

TSS Link: [TSS for FMT_SMF.1](#)

6.1.5 Privacy (FPR)

6.1.5.1 FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information

Origin: PP_APP_V1.4

FPR_ANO_EXT.1.1 The application shall

- **not transmit PII over a network**

.

TSS Link: [TSS for FPR_ANO_EXT.1](#)

6.1.6 Protection of the TSF (FPT)

6.1.6.1 FPT_AEX_EXT.1 Anti-Exploitation Capabilities

Origin: PP_APP_V1.4

FPT_AEX_EXT.1.1 The application shall not request to map memory at an explicit address except for **none**.

FPT_AEX_EXT.1.2 The application shall

- **not allocate any memory region with both write and execute permissions**

FPT_AEX_EXT.1.3 The application shall be compatible with security features provided by the platform vendor.

FPT_AEX_EXT.1.4 The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

FPT_AEX_EXT.1.5 The application shall be built with stack-based buffer overflow protection enabled.

TSS Link: [TSS for FPT_AEX_EXT.1](#)

6.1.6.2 FPT_API_EXT.1 Use of Supported Services and APIs

Origin: PP_APP_V1.4

FPT_API_EXT.1.1 The application shall use only documented platform APIs.

TSS Link: [TSS for FPT_API_EXT.1](#)

6.1.6.3 FPT_IDV_EXT.1 Software Identification and Versions

Origin: PP_APP_V1.4

FPT_IDV_EXT.1.1 The application shall be versioned with **SWID tags that comply with minimum requirements from ISO/IEC 19770-2:2015**.

TSS Link: [TSS for FPT_IDV_EXT.1](#)

6.1.6.4 FPT_LIB_EXT.1 Use of Third Party Libraries

Origin: PP_APP_V1.4

FPT_LIB_EXT.1.1 The application shall be packaged with only

- **boost 1.78**
- **cURL 8.9.1**
- **ICU 73.2**

- **OpenSSL 3.2.2**
- **OpenSSL FIPS Provider 3.0.8**
- **rapidxml 1.13**
- **SQLite 3.45.1**
- **Xerces-C 3.2.4**
- **zlib 1.3.1**

TSS Link: [TSS for FPT_LIB_EXT.1](#)

6.1.6.5 FPT_TUD_EXT.1 Integrity for Installation and Update

Origin: PP_APP_V1.4

- FPT_TUD_EXT.1.1** The application shall **provide the ability** to check for updates and patches to the application software.
- FPT_TUD_EXT.1.2** The application shall **provide the ability** to query the current version of the application software.
- FPT_TUD_EXT.1.3** The application shall not download, modify, replace or update its own binary code.
- FPT_TUD_EXT.1.4** Application updates shall be digitally signed such that the application platform can cryptographically verify them prior to installation.
- FPT_TUD_EXT.1.5** The application is distributed **as an additional software package to the platform OS.**

TSS Link: [TSS for FPT_TUD_EXT.1](#)

6.1.6.6 FPT_TUD_EXT.2 Integrity for Installation and Update

Origin: PP_APP_V1.4

Applied TDs: [TD0628](#)

- FPT_TUD_EXT.2.1** The application shall be distributed using **the format of the platform-supported package manager.**
- FPT_TUD_EXT.2.2** The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.
- FPT_TUD_EXT.2.3** The application installation package shall be digitally signed such that its platform can cryptographically verify them prior to installation.

TSS Link: [TSS for FPT_TUD_EXT.2](#)

6.1.7 Trusted path/channels (FTP)

6.1.7.1 FTP_DIT_EXT.1 Protection of Data in Transit

Origin: PP_APP_V1.4

Applied TDs: [TD0743](#)

FTP_DIT_EXT.1.1 The application shall

- **encrypt all transmitted data with HTTPS as a client in accordance with FCS_HTTPS_EXT.1/Client for gathering and downloading information from Fixlet servers and vendor sites, HTTPS as a server in accordance with FCS_HTTPS_EXT.1/Server for accepting security management actions from the BigFix Console and/or REST API applications, accepting information download from BigFix Clients** between itself and another trusted IT product.

TSS Link: [TSS for FTP_DIT_EXT.1](#)

6.2 Security Functional Requirements Rationale

The SFR rationale is defined in the [PP_APP_V1.4] [\[1\]](#) protection profile.

6.3 Security Assurance Requirements

The security assurance requirements (SARs) for the TOE are defined in the PP_APP_V1.4 protection profile.

The following table shows the SARs, and the operations performed on the components according to CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Table 4: SARs

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ADV Development	ADV_FSP.1 Basic functional specification	PP_APP_V1.4	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	PP_APP_V1.4	No	No	No	No
	AGD_PRE.1 Preparative procedures	PP_APP_V1.4	No	No	No	No
ALC Life-cycle support	ALC_CMC.1 Labelling of the TOE	PP_APP_V1.4	No	No	No	No
	ALC_CMS.1 TOE CM coverage	PP_APP_V1.4	No	No	No	No
	ALC_TSU_EXT.1	PP_APP_V1.4	No	No	No	No
ATE Tests	ATE_IND.1 Independent testing - conformance	PP_APP_V1.4	No	No	No	No
AVA Vulnerability assessment	AVA_VAN.1 Vulnerability survey	PP_APP_V1.4	No	No	No	No
ASE Security Target evaluation	ASE_CCL.1 Conformance claims	PP_APP_V1.4	No	No	No	No
	ASE_ECD.1 Extended components definition	PP_APP_V1.4	No	No	No	No
	ASE_INT.1 ST introduction	PP_APP_V1.4	No	No	No	No

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
	ASE_OBJ.1 Security objectives for the operational environment	PP_APP_V1.4	No	No	No	No
	ASE_REQ.1 Stated security requirements	PP_APP_V1.4	No	No	No	No
	ASE_SPD.1 Security problem definition	PP_APP_V1.4	No	No	No	No
	ASE_TSS.1 TOE summary specification	PP_APP_V1.4	No	No	No	No

6.4 Security Assurance Requirements Rationale

The SAR rationale is defined in the [PP_APP_V1.4] protection profile.

7 TOE Summary Specification

7.1 TOE Security Functionality

As per [PP_APP_V1.4] and [PKG_TLS_V1.1], the TOE supports the following major security features.

- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Privacy
- Protection of the TSF
- Trusted path/channels

7.1.1 Cryptographic support











The TOE uses the following cryptographic libraries:

- OpenSSL cryptographic module that is included in the TOE;
- Windows Cryptography API: Next Generation (CNG), provided by the underlying Windows platform.

The table below shows the cryptographic services used by the TOE and provided by the OpenSSL cryptographic module that is included in the TOE, describing the algorithms, their supported key sizes, applicable standard and purpose. The table also includes the certificates obtained from the Cryptographic Algorithm Validation Program (CAVP) in the evaluated configuration for each of the cryptographic algorithms.

Table 5: Cryptographic algorithms implemented by the TOE (OpenSSL) and CAVP certificates

Cryptographic service	Algorithm	Key sizes	Standard	Purpose	CAVP Cert.
FCS_CKM.1/AK - Asymmetric Key Generation	Elliptic Curve Cryptography (ECC)	P-256, P-384, P-521 (256 to 521 bits)	[FIPS186-5]	Ephemeral asymmetric key generation for TLS key exchange	A5321
	Finite Field Cryptography (FFC)	3072 bits	[FIPS186-4]	Ephemeral asymmetric key generation for TLS key exchange	A5321
	Finite Field Cryptography (FFC) with safe primes	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 (2048 to 8192 bits)	[SP800-56A-Rev3]	Ephemeral asymmetric key generation for TLS key exchange	A5331
FCS_CKM.2 - Key Establishment	RSA RSAES-PKCS1-v1_5	2048 bits or greater	[RFC8017]	TLS key exchange	Tested by the CCTL see Note 1

Cryptographic service	Algorithm	Key sizes	Standard	Purpose	CAVP Cert.
	Elliptic Curve Cryptography (ECC)	P-256, P-384, P-521 (256 to 521 bits)	[SP800-56A-Rev3] 	TLS key exchange	A5321
	Finite Field Cryptography (FFC)	3072 bits	[SP800-56A-Rev3] 	TLS key exchange	A5331
	Finite Field Cryptography (FFC) safe primes	ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192 (2048 to 8192 bits)	[SP800-56A-Rev3] 	TLS key exchange	A5331
FCS_COP.1/SKC - Data Encryption and Decryption	AES in CBC mode	128, 256 bits	[SP800-38A] 	Data encryption and decryption in the TLS protocol.	A5311
	AES in GCM mode	128, 256 bits	[SP800-38D] 	Authenticated data encryption and decryption in the TLS protocol.	A5316
FCS_COP.1/Hash - Message Digest	SHA-1, SHA2-256, SHA2-384, SHA2-512	N/A	[FIPS180-4] 	HMAC Conditioning of Console Operator's password Digital Signature Generation and Verification	A5321
FCS_COP.1/Sig - Digital Signature Generation and Verification	RSA digital signature generation and verification	2048, 3072, 4096 bits	[FIPS186-5] 	Server authentication in the TLS protocol X.509 Certificate Validation	A5321
	ECDSA digital signature generation and verification	P-256, P-384, P-521	[FIPS186-5] 	Server authentication in the TLS protocol X.509 Certificate Validation	A5321
FCS_COP.1/KeyedHash - Message Authentication Code	HMAC with SHA-1, SHA2-256, SHA2-384	160, 256, 384 bits	[FIPS198-1] 	Pseudorandom function (PRF) for the TLS protocol Data Integrity in the TLS protocol Conditioning of Console Operator's password	A5321
FCS_CKM_EXT.1/PBKDF - Password-Based Key Derivation	PBKDF	512 bits	[SP800-132] 	Conditioning of Console Operator's password	A5321

Cryptographic service	Algorithm	Key sizes	Standard	Purpose	CAVP Cert.
FCS_RBG_EXT.2 - Random Number Generator	DRBG (CTR_DRBG)	256 bits	[SP800-90A-Rev1]	Asymmetric Key Generation Salt for PBKDF Client and server random secrets in TLS protocol	A5309

Note 1: RSA key exchange was tested by the CCTL following the testing Assurance Activity for FCS_CKM.2.

7.1.1.1 FCS_CKM_EXT.1

The TOE implements asymmetric key generation for the following purposes:

- Generation of ephemeral asymmetric key pairs for the TLS protocol.

See [Table 5](#) for more information.

7.1.1.2 FCS_CKM.1/AK

[Table 5](#) provides information about asymmetric key generation methods used, key sizes and purpose.

All asymmetric key generation functionality is provided by the TOE with the bound OpenSSL cryptographic module.

7.1.1.3 FCS_CKM_EXT.1/PBKDF

The TOE implements password-based key derivation function (PBKDF) compliant with [SP800-132] for the following purposes:

- Conditioning of Console Operator's passwords

See [Table 5](#) for more information.

Credentials are needed to authenticate Console Operators for accessing the TOE from the Console or the REST interface. The Console Operator credential consists of a username and a password, both stored in the database. The password is conditioned by using the PBKDF algorithm before storing it in the database. The TOE also performs the same conditioning when the user attempts to authenticate to the TOE; the TOE compares the conditioned value of the password provided with the stored value and if they match the authentication succeeds.

The TOE invokes the PKCS5_PBKDF2_HMAC() function provided by the OpenSSL cryptographic module, using SHA2-256 as the message digest algorithm. The following parameters are used:

- **Password:** the TOE uses the Console Operator's password as it is, without any padding or encoding, before invoking the function. The password has a minimum of eight characters.
- **Salt:** randomly generated, 512 bits long.
- **Number of iterations:** 2000.
- **Message digest:** SHA2-256 algorithm.
- **Output length:** the conditioned password is 512 bits long.

The salt used for the PBKDF2 algorithm is generated with the DRBG provided by the OpenSSL cryptographic module. The salt is stored in the database together with the Console Operator's credentials.

7.1.1.4 FCS_CKM.2

The TOE implements key establishment schemes for the TLS protocol when using RSA, Diffie-Hellman (DH), and Elliptic Curve Diffie-Hellman (ECDH) as key exchange. See [Table 5](#) for more information.

7.1.1.5 FCS_COP.1/HASH

The TOE implements the SHA-1, SHA2-256, SHA2-384 and SHA2-512 message digest algorithms, compliant with [\[FIPS180-4\]](#), that are used for the following algorithms:

- HMAC: SHA-1, SHA2-256, and SHA2-384.
- RSA Signature Generation and Verification: SHA-1, SHA2-256, SHA2-384, and SHA2-512.
- ECDSA Signature Generation and Verification: SHA-1, SHA2-256, SHA2-384, and SHA2-512.
- PBKDF: SHA2-256.

See [Table 5](#) for more information.

7.1.1.6 FCS_COP.1/KEYEDHASH

The TOE implements the HMAC-SHA1, HMAC-SHA2-256, and HMAC-SHA2-384 functions, compliant with [\[FIPS198-1\]](#), that are used for the following purposes:

- Data integrity and PRF in the TLS protocol.
- PRF used in PBKDF.

See [Table 5](#) for more information.

7.1.1.7 FCS_COP.1/SIG

The TOE implements signature generation and verification using the RSA and ECDSA schemes with SHA-1, SHA2-256, SHA2-384, and SHA2-512, compliant with [\[FIPS186-5\]](#), that are used for the following purposes:

- X.509 Certificate validation.

See [Table 5](#) for more information.

7.1.1.8 FCS_COP.1/SKC

The TOE implements data encryption and decryption compliant with [\[FIPS197\]](#), using AES keys of 128 and 256 bits, in CBC and GCM modes, compliant with [\[SP800-38A\]](#) and [\[SP800-38D\]](#), respectively. These algorithms are used for the following purposes:

- Data encryption and decryption in the TLS protocol.
- Data integrity in the TLS protocol.

See [Table 5](#) for more information.

7.1.1.9 FCS_RBG_EXT.1

The TOE implements its own deterministic random bit generator (DRBG) functionality. The TOE includes the OpenSSL cryptographic module, which provides an implementation of an SP800-90A compliant DRBG.

7.1.1.10 FCS_RBG_EXT.2

The OpenSSL cryptographic module provides by default the CTR_DRBG mechanism using an AES key of 256 bits, providing a security strength of 256 bits. The TOE invokes this cryptographic service for random bit generation with the default settings, and there is no ability to specify the use of an alternative DRBG.

The TOE obtains entropy from an entropy source provided by the underlying platform to seed and reseed the DRBG. The OpenSSL cryptographic module obtains 384 bits of entropy to seed the DRBG during initialization, and 256 bits during reseeding. The OpenSSL cryptographic module invokes the BCryptGenRandom API function to obtain entropy from the entropy source. The amount of entropy used for seeding and reseeding the DRBG is always equal or greater than 256 bits.

An assessment of the entropy source is described in more detail in the proprietary Entropy Assessment Report [BIGFIX_EAR].

7.1.1.11 FCS_STO_EXT.1

The TOE securely stores credentials, using cryptographic functionality provided by the OpenSSL cryptographic module included in the TOE, and by the Windows Data Protection API (DPAPI) that is provided by the underlying platform. The following table shows the credentials necessary for the operation of the TOE, their purpose, their storage and how they are protected.

Table 6: Credential list

Credential	Purpose	Storage	Protection
Server signing certificate and private key	Acts as the server certificate for the TLS protocol.	In filesystem	Private key (EncryptedServerSigningKey file) is protected using CryptProtectData() and CryptUnprotectData() functions (Windows DPAPI)
CA certificate and private key	Sign and verify Client certificates. Each Client is assigned a certificate to prove its identity and that certificate is signed using the private key of the CA certificate.	In filesystem	Private key (EncryptedClientCAKey file) is protected using CryptProtectData() and CryptUnprotectData() functions (Windows DPAPI)
Console Operator username and password	Enforce Identification and Authentication of users from the Console and the REST API.	Database	Password is conditioned using PKCS5_PBKDF2_HMAC() function (OpenSSL in TOE)

Credentials and Private keys are stored in the filesystem at C:\Program Files (x86)\BigFix Enterprise\BES Server.

7.1.1.12 FCS_HTTPS_EXT.1/CLIENT

The TOE implements the HTTPS protocol to connect to Fixlet servers and vendor sites. The TOE uses HTTP over TLS compliant with [RFC2818].

The HTTPS Client is implemented by the cURL library which is bundled in the TOE. The cURL library runs over OpenSSL cryptographic module which is also bundled in the TOE. The cURL library initiates a connection to the server on the appropriate port. When the TLS handshake has successfully finished, the cURL library initiates the first HTTP request.

The X.509 certificate presented by the server endpoint is validated before establishing the secure connection; the TOE does not establish the connection if the peer certificate is deemed invalid. See section 7.1.3.1 for details about certificate validation.

7.1.1.13 FCS_HTTPS_EXT.1/SERVER

The TOE accepts incoming connections using the HTTPS protocol from the BigFix Console, client applications using REST, and BigFix Clients. The TOE uses HTTP over TLS compliant with [RFC2818].

The TOE implements the HTTPS Server using OpenSSL for TLS and the Windows Sockets (Winsock) API for lower-level sockets. Winsock is provided by the underlying platform; the OpenSSL API is provided by the OpenSSL cryptographic module part of the TOE. The HTTPS Server accepts incoming connections on the appropriate port. When the TLS handshake has successfully finished, the HTTPS Server starts the process of HTTP responses. The URL and port number of the HTTPS Server are configured during the TOE installation process.

7.1.1.14 FCS_TLS_EXT.1

The TOE implements the TLS protocol both as a client and server for protecting communication paths. The TLS protocol is implemented by the OpenSSL cryptographic module which is bundled in the TOE.

7.1.1.15 FCS_TLSC_EXT.1

The OpenSSL cryptographic module implements the TLS protocol version 1.2 compliant with [RFC5246][\[1\]](#). The TOE enables the following cipher suites in TLS when acting as a client.

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

Once the TLS handshake with the TLS server endpoint is finished, the TOE verifies:

- the validity of the X.509 certificate received from the server; and
- the server identity, by verifying the information contained in the server certificate.

The TOE performs X.509 certificate validation of the server certificate following the requirements described in [section 7.1.3.1](#). In case the certificate is deemed to be invalid, the TOE does not establish the trusted channel.

The TOE verifies the identity of the TLS server by following section 3.1 "Server Identity" of [RFC2818][\[1\]](#), as well as [RFC3280][\[2\]](#) and [RFC6125][\[3\]](#). The reference identifier is established from the URLs that the TOE uses when connecting to the Internet sites. The TOE performs the following steps.

1. The TOE first determines whether the reference identifier corresponds to a Domain Name System (DNS) domain name or an IP address, so it can find in the certificate the proper presented identifier.
2. The TOE then searches a presented identifier in the Subject Alternative Name (SAN) extension with the type that matches the one found in the URL. If multiple identifiers of the same type present in the certificate (e.g., two presented identifiers of `dNSName` type), it is considered acceptable if the reference identifier matches any one in the set.
3. For a DNS domain name, the `dNSName` type entry can include wildcards, but only at the beginning (e.g. `*.bigfix.com`).
 - If a SAN extension of type `dNSName` is found, then it is used as the identity. The TOE verifies that the hostname matches this presented identifier, otherwise it continues searching in the rest of the SAN extensions.
 - If a SAN extension of type `dNSName` is not found, then the TOE uses the most specific Common Name field in the Subject field of the certificate. The TOE verifies that the hostname matches this presented identifier.
4. For an IP Address, a SAN extension of type `iPAddress` must exist in the certificate, which is used as the identity. The TOE verifies that the IP Address exactly matches this presented identifier.

The TOE does not provide a general-purpose capability to "pin" public certificates.

The TOE does not provide an option for authorizing the override of invalid certificates.

7.1.1.16 FCS_TLSC_EXT.4

The OpenSSL cryptographic module supports secure renegotiation in compliance with [\[RFC5746\]](#).

7.1.1.17 FCS_TLSC_EXT.5

The OpenSSL cryptographic module supports the Supported Groups extension with the following values.

- For Elliptic Curve Groups (ECDHE), using the following elliptic curves compliant with [\[RFC8422\]](#).
 - `secp256r1`
 - `secp384r1`
 - `secp521r1`
- For Finite Field Groups (DHE), using the following safe primes compliant with [\[RFC7919\]](#).
 - `ffdhe2048(256)`
 - `ffdhe3072(257)`
 - `ffdhe4096(258)`
 - `ffdhe6144(259)`
 - `ffdhe8192(260)`

7.1.1.18 FCS_TLSS_EXT.1

The OpenSSL cryptographic module implements the TLS protocol version 1.2 compliant with [\[RFC5246\]](#). The TOE enables the following cipher suites in TLS when acting as a server.

- `TLS_DHE_RSA_WITH_AES_128_CBC_SHA256` as defined in RFC 5246
- `TLS_DHE_RSA_WITH_AES_256_CBC_SHA256` as defined in RFC 5246
- `TLS_DHE_RSA_WITH_AES_128_GCM_SHA256` as defined in RFC 5288

- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

The TOE supports the TLS protocol version 1.2. The TOE denies connections from clients requesting invalid or previous versions of the protocol, namely, SSL v2.0, SSL v3.0, TLS v1.0 and TLS v1.1.

The OpenSSL cryptographic module uses the following key exchange methods based on the cipher suites enabled by the TOE.

- Elliptic Curve Cryptography (ECC) scheme using NIST curves P-256, P-384 and P-521, compliant with [SP800-56A-Rev3] and [RFC8422].
- Finite Field Cryptography (FFC) scheme using parameter with size of 3072 bits, compliant with [SP800-56A-Rev3].

7.1.2 User data protection

7.1.2.1 FDP_DAR_EXT.1

The TOE protects sensitive data in accordance with FCS_STO_EXT.1 when it is stored in non-volatile memory. See [section 7.1.1.11](#) for further information. No other forms of sensitive data are stored by the TOE.

7.1.2.2 FDP_DEC_EXT.1

The TOE restricts its access to network connectivity; the TOE does not access to any other hardware resources or peripherals provided by the Operational Environment.

The TOE has access to the following sensitive information repositories provided by the Operational Environment:

- Windows Registry
- MSSQL database

Sensitive information in those repositories is protected by using mechanisms implemented in the TOE as well as provided by the underlying platform. See [section 7.1.2.1](#) and [section 7.1.1.11](#) for further information.

7.1.2.3 FDP_NET_EXT.1

The TOE restricts network communication to the remote endpoints summarized in the following table.

Table 7: TOE network communication paths

Initiated by	Connected to	Purpose	Protocol	Default port
BigFix Server (TOE)	OCSP responder	Certificate revocation checking.	OCSP	80

Initiated by	Connected to	Purpose	Protocol	Default port
BigFix Server (TOE)	HCL Fixlet server	Search and gather Fixlets.	HTTPS	443
BigFix Server (TOE)	Software vendor site	Download patches from vendors.	HTTPS	443
BigFix Console	BigFix Server (TOE)	Security management of BigFix platform.	HTTPS	52311
REST API application	BigFix Server (TOE)	Security management of BigFix platform.	HTTPS	52311
BigFix Server (TOE)	BigFix Client	Notify clients that information is available for download.	UDP	52311
BigFix Client	BigFix Server (TOE)	Register or Re-Register the Client computer.	HTTPS	52311
		Download update information from the Server, upload reporting information to the Server.	HTTPS	52311

7.1.3 Identification and authentication

7.1.3.1 FIA_X509_EXT.1

The TOE conducts certificate validation to verify the identity of the TLS server to which the TOE attempts to connect. Certificate validation is implemented in the OpenSSL cryptographic module meeting the following requirements:

- Certificate validation and certificate path validation conforms to [\[RFC5280\]](#).
- The certificate path must terminate with a trusted CA certificate.
- All CA certificates must have the basicConstraints extension present and the CA flag set to TRUE.
- The certificate must not be a revoked certificate. The TOE uses OCSP stapling and OCSP to check the certificate revocation status. The TOE first attempts using OCSP stapling; if the TLS server does not provide OCSP stapling, the TOE contacts the OCSP responder.
- The certificate presented by the TLS server must have the Server Authentication purpose in the extendedKeyUsage field.

7.1.3.2 FIA_X509_EXT.2

When establishing the connection to a TLS server, the TOE verifies its identity by validating the certificate received from the server as part of the TLS handshake. The TOE validates the certificate path by using the certificate bundle included in the BigFix server installation.

The purpose of certificate validation is to authenticate the TLS server. Therefore, the TOE uses the certificate received from the TLS server in the "Server Certificate" message during the TLS handshake. This behavior is by design and cannot be configured.

The TOE will automatically reject a certificate if it is found to be invalid; a certificate with unknown revocation status (because the TOE is unable to reach the OCSP responder or validate the OCSP response) is rejected.

If the validation of the server certificate fails, the TOE closes the connection. The TOE does not provide a means for defining an alternative action in case of the certificate validation failure.

7.1.4 Security management

7.1.4.1 FMT_CFG_EXT.1

The TOE does not include default credentials. The installer of the TOE launches the BigFix Administration tool, which allows the Site Administrator to set Master Operator credentials.

7.1.4.2 FMT_MEC_EXT.1

The table below provides the configuration options that are related to the security functionality claimed in this ST. The TOE stores all these options in the Windows Registry, that is, it uses the mechanism supported by the platform.

Table 8: Configuration options

Registry key	Setting and Purpose
_BESGather_Use_Https	Set to 1 to enforce HTTPS for gathering from Fixlet servers.
_BESRelay_Download_UseHttps	Set to 1 to enforce HTTPS for downloading from vendor sites.
_BESServer_HTTPServer_ForceTLS	Set to 1 to enforce HTTPS for accepting requests from Clients.
_BESGather_CAVerifyStrict	Set to 1 to enable strict root CA validation when gathering.
_BESRelay_Download_CAVerifyStrict	Set to 1 to enable strict root CA validation when downloading.
_BESGather_OcspVerify	Set to 1 to enable OCSP stapling and OCSP certificate revocation checks when gathering.
_BESRelay_Download_OcspVerify	Set to 1 to enable OCSP stapling and OCSP certificate revocation checks when downloading.
_BESRelay_HTTPRequester_OCSPCacheHours	Set to 0 to turn off OCSP caching.
_APIServer_HTTPServer_IsEnabled	Set to 0 to disable WebUI port (52315).
_RESTAPI_HTTPServer_PortNumber	In case of using a port number different from the default (52311), set the port for REST API. A non-default port is required when using a custom TLS server certificate for REST API.
_RESTAPI_HTTPServer_SSLCertificateFilePath	Indicate the path to the custom TLS server certificate file for REST API.
_RESTAPI_HTTPServer_SSLPrivateKeyFilePath	Indicate the path to the custom TLS server private key file for REST API.
_BESGather_CACert	Specify the directory storing the customized set of trusted certificates for certificate validation, for gathering information from Fixlet servers as a TLS client
_BESRelay_Download_CaCertDirectory	Specify the directory storing the customized set of trusted certificates for certificate validation, for downloading patches from vendor sites as a TLS client

7.1.4.3 FMT_SMF.1

The TOE provides the following Security Management functions related with the security claims made in this ST:

- Create Console Operators.

- Change the password of the logged-in Console Operator.
- Obtain TOE name and version.
- Verify updates for the TOE.

All these functions are accessible using the BigFix Console, which accesses the TOE through the TSF interface after a Console Operator authenticates.

7.1.5 Privacy

7.1.5.1 FPR_ANO_EXT.1

The TOE does not contain any functionality that relates to Personally Identifiable Information (PII).

7.1.6 Protection of the TSF

7.1.6.1 FPT_AEX_EXT.1

The following compiler and linker flags are enabled when the TOE is built:

- /DYNAMICBASE (Use address space layout randomization).
- /GS (Buffer Security Check).

The /DYNAMICBASE flag generates an executable image that can be randomly rebased at load time by using the address space layout randomization (ASLR) feature of Windows. This linker option is enabled by default.

The /GS flag instructs the compiler to perform buffer security checks. This compiler option is included in the building procedures.

All the executables and Dynamic Linked Libraries (DLLs) that are part of the TOE are compiled with these compiler and linker flags. There is no static memory mapping instructed in the building procedure.

7.1.6.2 FPT_API_EXT.1

The table below lists all API functions provided by the Windows platform that are loaded when the TOE is launched.

Table 9: Windows API functions used by the TOE

Dynamic Link Library	Windows API functions
ADVAPI32.dll	AddAccessAllowedAce AdjustTokenGroups AdjustTokenPrivileges AllocateAndInitializeSid BuildExplicitAccessWithName ChangeServiceConfig2 CloseServiceHandle ControlService ConvertSecurityDescriptorToStringSecurityDescriptor ConvertStringSecurityDescriptorToSecurityDescriptor CopySid CreateProcessAsUser CreateProcessWithLogonW CreateService CryptAcquireContext CryptGenRandom CryptReleaseContext DeleteService DeregisterEventSource DuplicateTokenEx EnumDependentServices EnumServicesStatus EnumServicesStatusEx EqualSid FreeSid GetAce GetAclInformation GetEffectiveRightsFromAcl GetExplicitEntriesFromAcl GetLengthSid GetNamedSecurityInfo GetSecurityDescriptorControl GetSecurityDescriptorDacl GetSecurityDescriptorGroup GetSecurityDescriptorOwner GetSecurityDescriptorSacl GetSecurityInfo GetSidIdentifierAuthority GetSidSubAuthority GetSidSubAuthorityCount GetTokenInformation GetUserName ImpersonateLoggedOnUser ImpersonateSelf InitializeAcl InitializeSecurityDescriptor IsValidSid LogonUser LogonUserW LookupAccountName LookupAccountSid LookupPrivilegeValue LsaClose LsaNtStatusToWinError LsaOpenPolicy OpenProcessToken OpenSCManager OpenService OpenThreadToken

Dynamic Link Library	Windows API functions
	QueryServiceConfig QueryServiceStatus QueryServiceStatusEx RegCloseKey RegConnectRegistry RegCreateKey RegCreateKeyEx RegDeleteKey RegDeleteKeyEx RegDeleteValue RegEnumKeyEx RegEnumValue RegFlushKey RegisterEventSource RegisterServiceCtrlHandler RegisterServiceCtrlHandlerEx RegNotifyChangeKeyValue RegOpenKeyEx RegQueryInfoKey RegQueryValueEx RegSetValueEx ReportEvent RevertToSelf SetEntriesInAcl SetNamedSecurityInfo SetSecurityDescriptorDacl SetSecurityDescriptorGroup SetSecurityDescriptorOwner SetServiceObjectSecurity SetServiceStatus StartService StartServiceCtrlDispatcher
CRYPT32.dll	CryptProtectData CryptUnprotectData
KERNEL32.dll	AcquireSRWLockExclusive AcquireSRWLockShared AssignProcessToJobObject CloseHandle CompareString CompareStringA CompareStringW CopyFile CreateDirectory CreateEvent CreateFile CreateFileMapping CreateIoCompletionPort CreateJobObject CreateMutex CreatePipe CreateProcess CreateSemaphore CreateThread CreateToolhelp32Snapshot DeleteCriticalSection DeleteFile DeleteFileW EnterCriticalSection ExpandEnvironmentStrings FindClose FindCloseChangeNotification FindFirstChangeNotification FindFirstFile FindFirstFileEx FindFirstFileW FindNextChangeNotification FindNextFile FindNextFileW FindResource FlushFileBuffers FormatMessage FormatMessageA FreeEnvironmentStrings FreeLibrary FreeResource GetACP GetCommandLineW GetComputerName GetComputerNameEx GetConsoleOutputCP GetCurrentDirectory GetCurrentProcess GetCurrentProcessId GetCurrentThread GetCurrentThreadId GetDateFormat GetDiskFreeSpace GetDiskFreeSpaceEx GetDriveType GetEnvironmentStrings GetEnvironmentVariable GetExitCodeProcess GetFileAttributes GetFileAttributesW GetFileInformationByHandle GetFileSize GetFileType GetLastError GetLocaleInfo GetLocaleInfoA GetLocaleInfoW GetModuleFileName GetModuleHandle GetNumberFormatW GetPriorityClass GetProcAddress GetProcessHeap GetProcessTimes GetProfileInt GetStdHandle GetSystemDefaultUILanguage GetSystemDirectory GetSystemTime GetTempFileName GetTempPath GetTickCount GetTickCount64 GetTimeFormat GetTimeZoneInformation GetUserDefaultUILanguage GetVersionEx GetVolumeInformation GetVolumeInformationA GetWindowsDirectory GlobalAlloc GlobalFree GlobalLock GlobalMemoryStatus GlobalMemoryStatusEx GlobalUnlock HeapCompact InitializeConditionVariable InitializeCriticalSection InitializeSRWLock IsDebuggerPresent LCMAPStringW LeaveCriticalSection LoadLibrary LoadLibraryEx LoadResource LocalFree LockResource lstrcmp lstrcmpi MapViewOfFile MoveFile MoveFileEx MulDiv MultiByteToWideChar OpenProcess OutputDebugString Process32First Process32Next QueryPerformanceCounter QueryPerformanceFrequency RaiseException ReadDirectoryChanges ReadDirectoryChangesW ReadFile ReleaseMutex ReleaseSemaphore ReleaseSRWLockExclusive ReleaseSRWLockShared RemoveDirectory RemoveDirectoryW ResetEvent ResumeThread SetConsoleCtrlHandler SetEndOfFile SetEnvironmentVariable SetEvent SetFileAttributes SetFileAttributesW SetFilePointer SetFileTime SetHandleInformation SetInformationJobObject SetLastError SetStdHandle SetUnhandledExceptionFilter Sleep SleepConditionVariableCS TerminateProcess TlsAlloc TlsFree TlsGetValue TlsSetValue TryEnterCriticalSection UnhandledExceptionFilter UnmapViewOfFile VirtualQuery WaitForMultipleObjects WaitForMultipleObjectsEx WaitForSingleObject WaitForSingleObjectEx WakeAllConditionVariable WakeConditionVariable WideCharToMultiByte WriteFile
NTDSAPI.dll	DsBind DsBindW DsBindWithCred DsBindWithCredW DsCrackSpn DsFreePasswordCredentials DsMakePasswordCredentials DsMakePasswordCredentialsW DsServerRegisterSpn DsServerRegisterSpnW DsUnBind DsWriteAccountSpn DsWriteAccountSpnW
ole32.dll	CoCreateGuid CoCreateInstance CoInitialize CoTaskMemAlloc CoTaskMemFree CoUninitialize ReleaseStgMedium StringFromGUID2

Dynamic Link Library	Windows API functions
Secur32.dll	DeleteSecurityContext FreeContextBuffer FreeCredentialsHandle GetAdaptersAddresses GetProcessMemoryInfo GetUserNameEx InitializeSecurityContext InitializeSecurityContextW InitSecurityInterface RevertSecurityContext
SHELL32.dll	CommandLineToArgvW SHBrowseForFolder ShellExecute ShellExecuteEx SHGetFolderLocation SHGetFolderPath SHGetPathFromIDList
SHLWAPI.dll	PathFindFileName PathMatchSpec PathRemoveExtension PathStripPath
USERENV.dll	CreateEnvironmentBlock DestroyEnvironmentBlock LoadUserProfile UnloadUserProfile
UxTheme.dll	CloseThemeData DrawThemeBackground DrawThemeText GetThemePartSize IsAppThemed OpenThemeData
VERSION.dll	GetFileVersionInfoEx GetFileVersionInfoSizeEx VerQueryValue
WINHTTP.dll	WinHttpCloseHandle WinHttpGetIEProxyConfigForCurrentUser WinHttpGetProxyForUrl WinHttpOpen
WS2.dll	freeaddrinfo getaddrinfo GetAddrInfo inet_ntop WSAAddressToString WSAEnumProtocols WSALoctl

7.1.6.3 FPT_IDV_EXT.1

The TOE uses a software identification (SWID) tag as defined in the ISO/IEC 19770-2:2015 standard.

The SWID tag includes a Software Identity element and an Entity element, and is stored in the hcltechsw.com-BigFix_Platform_Server.swidtag file.

7.1.6.4 FPT_LIB_EXT.1

The TOE is comprised with the third-party libraries shown in [FPT_LIB_EXT.1](#).

7.1.6.5 FPT_TUD_EXT.1

The TOE and updates to the TOE are available for download at the BigFix Enterprise Suite Download Center ([\[BIGFIX_DOWNLOAD\]](#)).

Both the TOE and its updates are distributed as an InstallShield installation package in EXE format. The installation package, as well as the binaries of the TOE, are signed by HCL America Inc. using Microsoft Authenticode, with RSA and SHA2-256 as the algorithms that are part of the digital signature generation. The authorized source of installation packages is HCL America Inc.

7.1.6.6 FPT_TUD_EXT.2

See TSS for [FPT_TUD_EXT.1](#).

7.1.7 Trusted path/channels

The TOE provides protection of all data in transit between the TOE and other trusted endpoints as described in [TSS_FDP_NET_EXT.1](#). The trusted channels are established using HTTPS and TLS.

7.1.7.1 FTP_DIT_EXT.1

The TOE includes the OpenSSL cryptographic module to implement the TLS protocol. The TOE implements the HTTPS Client using the cURL library which uses OpenSSL and is part of the TOE. The TOE implements the HTTPS Server using OpenSSL for TLS and the Windows Sockets (Winsock) API for lower-level sockets. Winsock is provided by the underlying platform; the OpenSSL API is provided by the OpenSSL cryptographic module part of the TOE.

7.2 Security Assurance

7.2.1 Life-cycle support

7.2.1.1 ALC_TSU_EXT.1

HCL does not publicly disclose or confirm security vulnerabilities until HCL has conducted an analysis on the vulnerabilities and issued fixes and/or mitigations. HCL uses version 3.1 of the Common Vulnerability Scoring System (CVSS) as part of its standard process of evaluating potential vulnerabilities; CVSS is used for prioritizing responses and resources according to vulnerability severity levels.

For a validated security vulnerability, HCL will provide fixes and/or mitigations within no more than 180 days from the public disclosure of the vulnerability. The fixes for security vulnerabilities that result in the TOE security updates are rolled into BigFix product patch releases, and are delivered via the TOE update process.

Once the fixes and/or mitigations to security vulnerabilities are available, the information relating to addressed vulnerabilities will be published in Security Bulletins, which are available from the Knowledge Base on the HCL Customer Support portal (https://support.hcltechsw.com/csm?id=kb_search).

The security bulletin information is also distributed to users through mailing lists. Users can sign up for the email notifications of security bulletins by visiting the HCL Product Security Incident Response Team (PSIRT) Blog (https://support.hcltechsw.com/csm?id=community_forum&sys_id=038a2b921b7bb34c77761fc58d4bcb0d).

Users can report a potential security vulnerability through logging into the HCL Customer Support portal (<https://support.hcltechsw.com/csm>) via the HTTPS protocol and opening a support case.

8 Abbreviations, Terminology, and References

8.1 Abbreviations

AES	Advanced Encryption System
API	Application Programming Interface
ASLR	Address Space Layout Randomization
CA	Certificate Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CNG	Windows Cryptography API: Next Generation
CN	Common Name
CRL	Certificate Revocation List
cURL	Client for URLs
CVSS	Common Vulnerability Scoring System
DLL	Dynamic Link Library
DN	Distinguished Name
DNS	Domain Name System
DPAPI	Data Protection Application Programming Interface
DRBG	Deterministic Random Bit Generator
EAR	Entropy Analysis Report
ECC	Elliptic Curve Cryptography

ECDSA	Elliptic Curve Digital Signature Algorithm
EMS	Endpoint Management System
FCC	Finite Field Cryptography
FIPS	Federal Information Processing Standard
GCM	Galois/Counter Mode
HMAC	Keyed-hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
LDAP	Lightweight Directory Access Protocol
LDAPS	Secure LDAP
NIAP	National Information Assurance Partnership
ODBC	Open Database Connectivity
OSP	Organizational Security Policies
PCL	Product Compliant List
PII	Personally Identifiable Information
PP	Protection Profile
PRF	Pseudorandom Function
PSIRT	Product Security Incident Response Team
REST	Representational State Transfer
RSA	Rivest-Shamir-Adleman
SAN	Subject Alternative Name

SN	Subject Name
ST	Security Target
SWID	Software ID
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TOE Security Functionality Interface
TSS	TOE Security Summary

8.2 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

Action

An Action is a change applied to a target system to remediate issues identified by Fixlets. They are typically scripts written in the BigFix Action Language. A Fixlet that detects an issue may offer several different remediation Actions that authorized operators may choose from and deploy. For example, a Fixlet may detect a missing Windows Service Pack and offer an Action to download and install it on the relevant target systems.

BES

BigFix Enterprise Suite.

BES sites (HCL Fixlet server)

Fixlets are available to administrators by subscribing to any of several Internet-based HCL Fixlet servers. BES sites are outside of TOE; the Fixlet sites are maintained by HCL and are global. Each BES site contains pre-tested, pre-packaged Fixlet messages that provide out-of-the-box management solutions.

Console Operator(s)

Master Operator, Operator.

Custom Sites

Fixlet messages can also be developed in-house by administrators to address policy, configuration, and vulnerability concerns specific to the customer's environment. In-house fixes are known as Custom Fixlets and are developed by an authorized administrator to address specific situations. Both Fixlets and Custom Fixlets are supported by the TOE. Custom Fixlets are published on custom sites locally on the BigFix server, thus they are not global.

Fixlet

The TOE utilizes a patented Fixlet® technology to identify vulnerable or misconfigured endpoints and allows authorized users to remediate identified issues. Fixlets are sent via Fixlet messages to the target endpoints by the TOE, and provide an automatic fix for the identified issues. For the purposes of this ST, the term Fixlet includes all the different types of Fixlet messages including Fixlets, Tasks, Analyses, and Baselines.

Master Operator

A TOE Console operator with administrative rights. A Master Operator can do almost everything a Site Administrator can do except for some configuration operations that affects the masthead.

Masthead

Created during installation of the TOE; includes URLs for the BigFix server's CGI programs and other site information in a signed MIME message. The Masthead is central to accessing and authenticating the enterprise action site. The TOE brings content into the enterprise based on subscribed Mastheads. A Masthead is required for communicating with the HCL Fixlet Server as it contains all the site-specific information needed to deploy Fixlets.

Operator

An authorized user of the TOE Console. Ordinary Operators can deploy Fixlet actions and edit certain computer settings. Management rights are assigned by Master Operators.

Site Administrator

The only TOE user (besadmin) with the right to edit and change the masthead. Those changes are TOE advanced settings, security settings and other configuration settings.

8.3 References

BIGFIX_DOWNLOAD	BigFix Enterprise Suite Download Center Date 2024-09-23 received Location https://support.bigfix.com/bes/release
BIGFIX_EAR	BigFix Server Entropy Assessment Report Version 1.0 Date 2025
CC	Common Criteria for Information Technology Security Evaluation Version 3.1R5 Date April 2017 Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.p df Location <a href="http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.p
df">http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.p df Location <a href="http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.p
df">http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.p df
CCGUIDE	HCL BigFix Server Version 11.0.3 Common Criteria Configuration Guide Version 1.0 Date 2025

FIPS180-4	Secure Hash Standard (SHS) Date 2015-08-04 Location https://csrc.nist.gov/pubs/fips/180-4/upd1/final
FIPS186-4	Digital Signature Standard (DSS) Date 2013-07-19 Location https://csrc.nist.gov/pubs/fips/186-4/final
FIPS186-5	Digital Signature Standard (DSS) Date 2023-02-03 Location https://csrc.nist.gov/pubs/fips/186-5/final
FIPS197	Advanced Encryption Standard (AES) Date 2023-05-09 Location https://csrc.nist.gov/pubs/fips/197/final
FIPS198-1	The Keyed-Hash Message Authentication Code (HMAC) Date 2008-07-16 Location https://csrc.nist.gov/pubs/fips/198-1/final
NIAP_PCL	NIAP Product Compliant List Date 2019-09-07 Location https://www.niap-ccevs.org/products
PKG_TLS_V1.1	Functional Package for TLS Version 1.1 Version 1.1 Date 2019-03-01 Location https://www.niap-ccevs.org/protectionprofiles/439
PP_APP_V1.4	Protection Profile for Application Software Version 1.4 Version 1.4 Date 2021-10-07 Location https://www.niap-ccevs.org/protectionprofiles/462
RFC2818	HTTP Over TLS Author(s) E. Rescorla Date 2000-05-01 Location http://www.ietf.org/rfc/rfc2818.txt
RFC3280	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile Author(s) R. Housley, W. Polk, W. Ford, D. Solo Date 2002-04-01 Location http://www.ietf.org/rfc/rfc3280.txt
RFC5246	The Transport Layer Security (TLS) Protocol Version 1.2 Author(s) T. Dierks, E. Rescorla Date 2008-08-01 Location http://www.ietf.org/rfc/rfc5246.txt
RFC5280	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile Author(s) D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk

	Date 2008-05-01 Location http://www.ietf.org/rfc/rfc5280.txt
RFC5746	Transport Layer Security (TLS) Renegotiation Indication Extension Author(s) E. Rescorla, M. Ray, S. Dispensa, N. Oskov Date 2010-02-01 Location http://www.ietf.org/rfc/rfc5746.txt
RFC6125	Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS) Author(s) P. Saint-Andre, J. Hodges Date 2011-03-01 Location http://www.ietf.org/rfc/rfc6125.txt
RFC7919	Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS) Author(s) D. Gillmor Date 2016-08-01 Location http://www.ietf.org/rfc/rfc7919.txt
RFC8017	PKCS #1: RSA Cryptography Specifications Version 2.2 Author(s) K. Moriarty, B. Kaliski, J. Jonsson, A. Rusch Date 2016-11-01 Location http://www.ietf.org/rfc/rfc8017.txt
RFC8422	Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier Author(s) Y. Nir, S. Josefsson, M. Pegourie-Gonnard Date 2018-08-01 Location http://www.ietf.org/rfc/rfc8422.txt
SP800-132	Recommendation for Password-Based Key Derivation: Part 1: Storage Applications Date 2010-12-22 Location https://csrc.nist.gov/pubs/sp/800/132/final
SP800-38A	Recommendation for Block Cipher Modes of Operation: Methods and Techniques Date 2001-12-01 Location https://csrc.nist.gov/pubs/sp/800/38/a/final
SP800-38D	Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC Date 2007-11-28 Location https://csrc.nist.gov/pubs/sp/800/38/d/final
SP800-56A-Rev3	Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography Date 2018-04-16 Location https://csrc.nist.gov/pubs/sp/800/56/a/r3/final

SP800-90A-Rev1

**Recommendation for Random Number Generation Using Deterministic
Random Bit Generators**

Date 2015-06-24

Location <https://csrc.nist.gov/pubs/sp/800/90/a/r1/final>