HCL BigFix version 10.0.1.41
Common Criteria
Security Target

Version: 1.4

Status: Final

Last Update: 2021-02-26

Contents

Document History

Version	Date	Author	Description
1.4	2021-02-26	HCL Technologies Limited	Public Version

1 Security Target Introduction

1.1 Security Target and TOE Identification

ST Title: HCL BigFix version 10.0.1 Common Criteria Security Target

ST Version: 1.4

ST Date: 2021-02-26

TOE Identification: HCL BigFix version 10.0.1.41

TOE Developer: HCL Technologies Limited

Evaluation Sponsor: HCL Technologies Limited

Certification Body: OCSI

Certification ID: OCSI-CERT-ATS-07-2020

1.2 TOE Overview

The TOE is the *HCL BigFix version 10.0.1.41* (a.k.a. BigFix) software product provided by *HCL Technologies Limited*. The TOE (i.e., TOE type) is a centralized endpoint management system that allows authorized operators to monitor the system configurations of distributed endpoint systems (client computers) and enables operators to take any necessary corrective actions. The TOE also includes the TOE guidance documentation.

The evaluated version of the TOE is: 10.0.1.41

1.2.1 Usage

The TOE contains the following software components.

- BigFix Server (a.k.a. server)
- BigFix Console (a.k.a. console)
- BigFix Relay (a.k.a. relay)
- BigFix Client (a.k.a. client)

Using the Console, TOE administrators monitor and manage the software configurations of enrolled endpoint systems (Client Computers) that run the Client. The TOE supports the use of multiple consoles where each Console runs on a separate system. The consoles connect to the Server to monitor and manage the Client Computers. The Server runs on a separate system and maintains a local database of enrolled Client Computers, the configuration of each enrolled Client Computer, and other data. When the Server detects that an enrolled Client Computer requires one or more corrective actions, the Server informs the Client. The Client then pulls the required corrective actions from the Server and applies the actions to the Client Computer.

In large distributed systems, the Server may offload corrective action deployment to relay systems. Each Relay system runs the Relay. The Server informs each Client of its assigned set of relays from which it can obtain corrective actions. The client software then pulls the corrective actions from its assigned relay(s) and applies the actions to the Client Computer.

A corrective action is known as a Fixlet®. A Fixlet can be a software update or an endpoint configuration update. HCL Technologies Limited maintains multiple Internet-based HCL Fixlet Servers containing commonly deployed Fixlets to which a TOE customer can subscribe. Once subscribed, the Server can download Fixlets from an HCL Fixlet Server and distribute the Fixlets to enrolled Client Computers. TOE administrators can also modify enrolled Client Computer configurations (e.g., add or remove applications) and have the Server and Client enforce the new configurations.

1.2.2 TOE security features

The TOE contains the following major security features.

- Cryptographic support – To implement communication channel protection, digital signature generation and verification and data encryption and decryption.
- User data protection – To manage the information flow control policy between administrators and Client Computers to apply Actions.
- Identification and authentication (I&A) – To allow security management only to authorized administrators.
- Security management – To manage the information flow control policy, manage administrators, Client Computers, sites and certficates.
- Protection of the TSF – To prevent modification and disclosure of data transferred between TOE components.
- Trusted path/channels – To prevent modification and disclosure of data transferred between the TOE and other external IT entities.

1.2.3 Non-TOE hardware and software (operational environment)

Each TOE component requires additional hardware and software that comprise the operational environment. The following lists the software and hardware required by each TOE component.

- BigFix Server:
  - Windows Server 2016
  - MSSQL Server 2016
  - Processor X86-64 (4CPU), 16 GB RAM, 250 GB Disk

- BigFix Console:
  - Windows Server 2016
  - Processor X86-64 (2CPU), 4 GB RAM, 20 GB Disk

- BigFix Relay:
  - Windows 10
  - Processor X86-64 (2CPU), 4 GB RAM, 25 GB Disk

- BigFix Client:
  - Supported on following operating systems:
    - Windows Server 2016

    - Windows 10
    - Red Hat Enterprise Linux (RHEL) 7
- Processor X86-64 (2CPU), 4 GB RAM, 20 GB Disk

The TOE also requires the following service(s) in the operational environment.

- Domain Name System (DNS) service

1.3 Terminology and Acronyms

This section specifies terminology and acronyms used in the ST.

1.3.1 Terminology and Acronyms

The following acronyms and specific terminology pertain to BigFix. Note that CC-specific and other commonly used terms and acronyms are not defined here.

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.

Administered – Client Computers that an operator has authority to manage, the target systems of a BigFix actions.

All Visible - All Fixlets, Client Computers, and Actions visible (or authorized to) to a given operator.

BES - BigFix Enterprise Suite

CGI - Common Gateway Interface

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 providing an automatic fix for it. For the purposes of this ST, the term Fixlet includes all the different types of Fixlet messages including Fixlets, Tasks, Analyses, and Baselines.

BES sites (HCL Fixlet server) **–** Fixlets are available to administrators by subscribing to any of several Internet-based BigFix 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.

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.

Masthead - Created during installation of the TOE that 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 BigFix Fixlet Server as it contains all the site-specific information needed to deploy Fixlets.

Console Operator(s) **–** Master Operator, Operator

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

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.

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.

Signing Password -The password specified during installation of the TOE that is used by an authorized user to sign an action for deployment.

1.4 TOE Description

The TOE is a client-server application that allows monitoring and management of targeted IT systems from a central location. The TOE 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.

Fixlet messages are available to an enterprise by subscribing to any of several Fixlet Sites that are maintained by the BigFix Fixlet Server which is not part of the TOE and is outside from the evaluated configuration. Each Fixlet Site contains pre-tested, pre-packaged Fixlet messages that provide out-of-the-box management solutions. They constitute data that the TOE collects, distributes and otherwise utilizes via the internet from the BigFix Fixlet Server to detect and remediate vulnerabilities.

Fixlets 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; and,
- View, modify and audit properties and configurations of the networked client computers

The TOE contains built-in public/private key cryptographic capabilities to ensure the authenticity of the Fixlet messages and remedial Actions. Each Fixlet and Action received by a BigFix Client is authenticated by verifying a digital signature affixed by the applicable administrator to ensure that it was generated by an administrator authorized to perform corresponding operations. These authorized operations instruct BigFix Clients to view, modify and audit properties and configurations of the networked client computers. The results from those operations — or simply the gathered data — is encrypted and delivered back to the BES server.

1.4.1 TOE Architecture

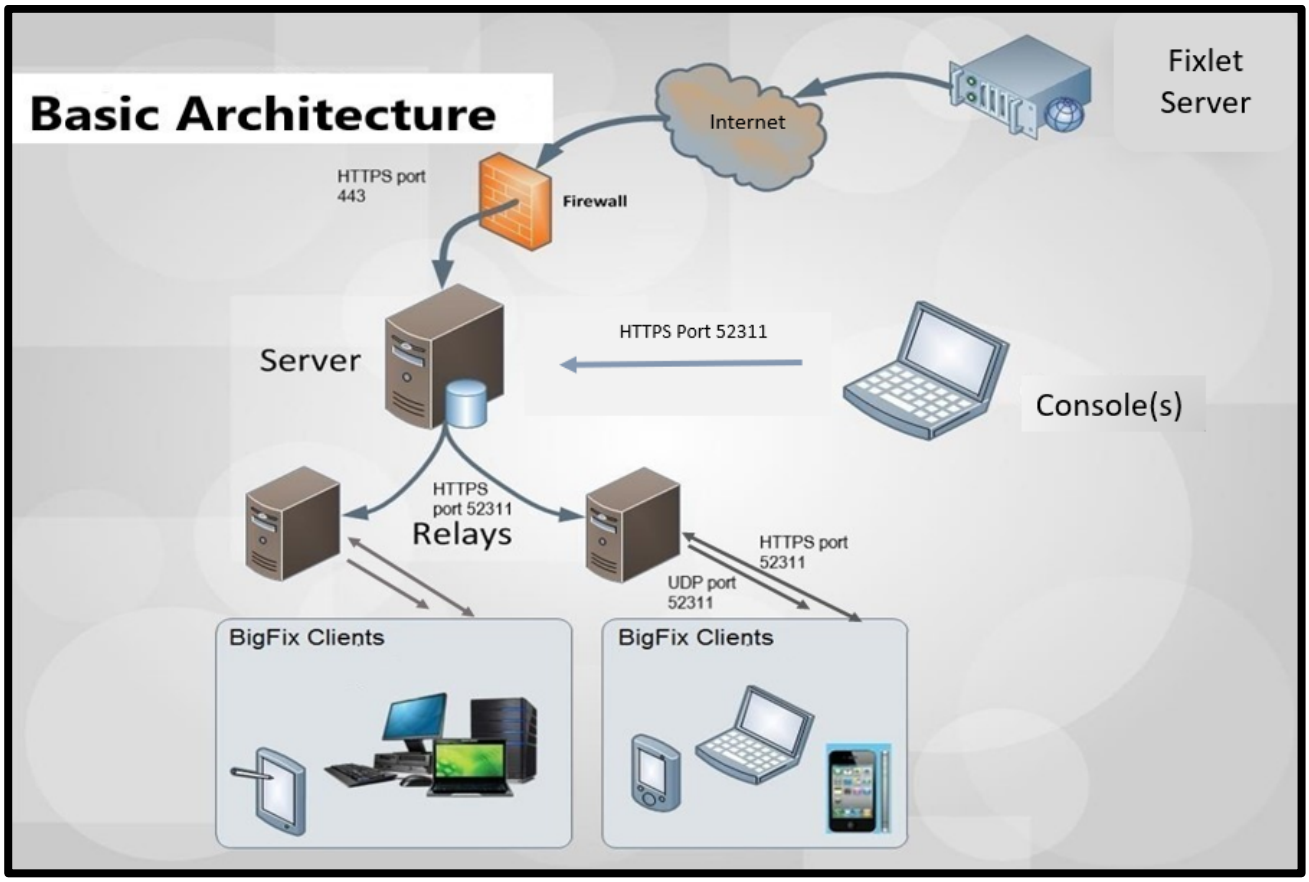
The TOE is comprised of four software components, BigFix Server, BigFix Console, BigFix Client (i.e. Agent) and BigFix Relay. During installation of the TOE, the authorized Site Administrator creates a Masthead that ties the TOE together. Among other things, this Masthead includes a key (signed by the Site Administrator) to authenticate any instructions from the BigFix Server. Following is an overview of each of the components, hereinafter referred to as Server, Console, Client and Relay.

The TOE provides an authorized user the ability to assess the status of client machines Operating System (OS), applications, anti-virus signatures, etc. (using Fixlets) and provides the ability to update these machines as necessary (using Actions). The TOE relies on the ability of client machines to periodically check with the server (or designated relay) in order to obtain the most current Fixlets and/or Actions.

The figures below depict a typical application of the TOE and an overview of the basic TOE architecture. There is at least one server that gathers Fixlets from the BES sites on Internet where

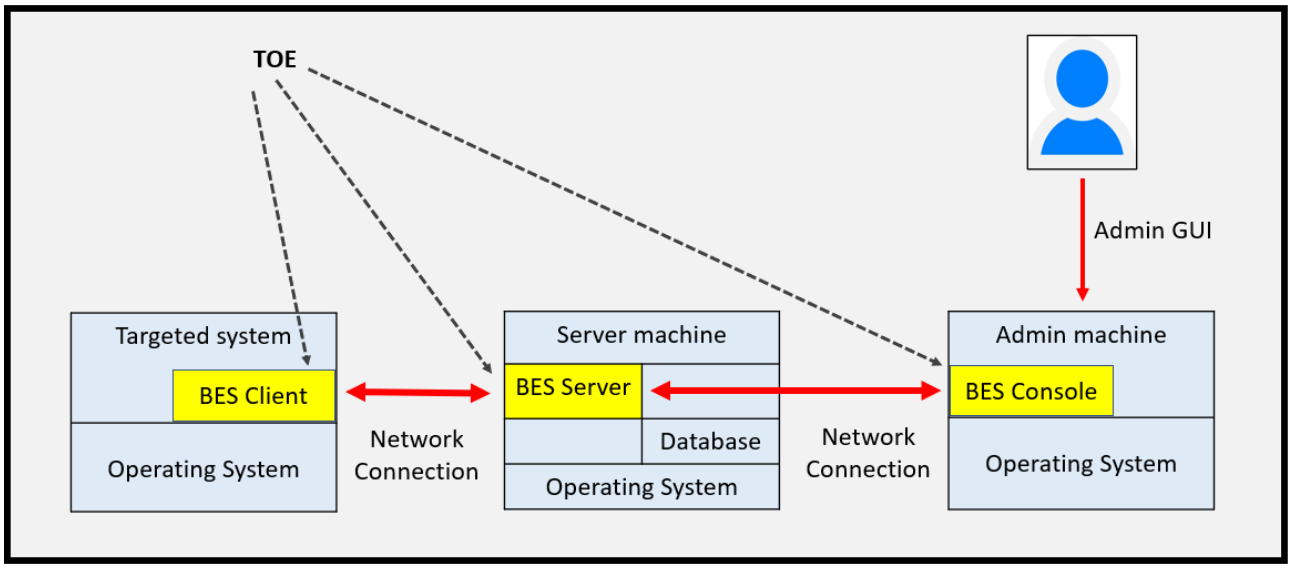
they can be viewed by the console operator and distributed to the relays. Each client inspects its local computer environment and reports any relevant Fixlets back to the relay, which compresses the data and passes it back up to the servers.

Figure 1 - TOE architecture



The solid arrows in Figure 1 reflect the required TOE components as well as the optional Fixlet service in the IT Environment provided by BigFix via the Internet. Note that while the figure depicts the TOE as computers of various types, the TOE consists only of software running in the context of the computers and their installed operating systems. Figure 2 presents a more logical view of the primary TOE components in the context of their host computers. Note that, while not depicted below, a Relay is essentially a combination of Client and Server components acting to store and forward communications in both directions. Relays are optional components that do not affect the security functions of the TOE, but provide for network efficiency in distributing Fixlets and actions.

Figure 2 - TOE Logical View



1.4.1.1 Server

The Server is a collection of interacting services, including application services, and a web server. The Server manages and coordinates the flow of information to and from individual Client Computers and stores the results in the BES database. This database is outside the TOE (i.e., in the operational environment), though it can reside on the same physical computer. The database is used by the TOE to store and retrieve applicable Fixlets and Actions as well as TOE configuration data. The BES database is expected to be configured so that only authorized users can access any contents associated with the TOE. The BES database is also expected to be configured so that its ODBC interface and communications are protected in a manner appropriate to the environment in which it is being used. Note that the BES Server can be configured to periodically collect pre-defined Fixlets from BigFix via a BigFix Fixlet Server. Those, like any locally developed Fixlets, are stored in the BES database and are available for use by administrators in monitoring Clients. The Server offers the following features:

- The Server gathers content from the BES sites on Internet (i.e., Fixlets offered by the BigFix Fixlet Server) and then redistributes the content to the BigFix Clients directly (or through BigFix Relays). This component provides 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 (i.e 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 Server and the Client is given a unique Identifier (ID).
- When a Client detects that a Fixlet has become relevant, it reports to the Server using a secure Hyper-text Transfer Protocol (HTTPS) “POST” operation. The Server 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 Server. All Client data can flow directly to the Server or through relays.

- The Server monitors for changes in Fixlet content for all the Fixlet sites (e.g., BigFix Fixlet Server) to which the TOE is subscribed and it downloads these changes to the Server and makes them available to the rest of the components.
- The Server offers a GUI interface (the Administrator Console, local to the hosting operating system) for the Site Administrator to manage some TOE global options such as the refresh rates, and the Masthead management.

The Server listens on Transport Control Protocol (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 TLS/HTTPS protected messages (Fixlet messages). The TOE can issue User Datagram Protocol (UDP) messages to clients to when new content (e.g., a Fixlet) becomes available, to notify them.
The Server also listens on Transport Control Protocol (TCP) port (52311 by default) for TLS/HTTPS messages from Consoles and REST API clients that connect to the server to perform security management functions.

The BES database, which is often collocated on the computer hosting the Server is accessible via ODBC. The Server is the TOE component that uses the BES database to store and retrieve applicable data. Note that BigFix has published guidance so that users could potentially develop their own applications to access TOE-related data, provided they have applicable BES database authorizations. However, the development and use of other applications to access TOE data, while not forbidden, is outside the scope of this evaluation.

1.4.1.2 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 (SPs), configuration violations and/or enterprise security policy violations), and remediation from a central location.

Console users, also known as Operators, can be in charge of flexibly defined groups of computers (running the Client) with varying degrees of freedom. A Master Operator has overall control of each Operator's domain and the specific rights they have over that domain. The TOE supports two classes of Console users: Master Operators, and (ordinary) Operators. See section 7.1.3 for details about their respective responsibilities.

The Console is invoked as an interactive application. The TOE enforces the use of TLS/HTTPS (Hypertext Transfer Protocol Secure) to protect the communications channel of the Console. The TOE also enforces authenticity and integrity of all remote console users through the use of user names and passwords. The account information for the Console is managed by the Server software component and is located remotely on the database server.

Multiple Consoles can connect to the TOE simultaneously.

1.4.1.3 BigFix Administration Tool

This program is installed with the Server component and it is located on the computer hosting the Server, it provides a graphical interface (GUI) for BigFix administrative operations. 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, resign the content in the database, rotate the server private key, configure the Console and synchronize the masthead with the updated license.

1.4.1.4 REST API

The BigFix Server provides a REST API programming interface. It allows the majority of the tasks available in the BigFix console by using a set of standardized and operating system independent methods.

1.4.1.5 IEM CLI

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

1.4.1.6 Clients

Clients are installed on every Client 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. 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 Server. 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 Operator is able to provide friendly screen prompts for the user.

The Clients listen on a UDP port (default 52311) for messages from the Server or Relays indicating that updated data is available for retrieval. The Clients use HTTPS to connect to Relays and/or Servers in order to retrieve Fixlets and Actions and to send results of applying Fixlets and Actions back to the Server or Relays. The TOE provides secure settings to specify that the Clients encrypt these results before they are transmitted over the network.

1.4.1.7 Relays

Relays can increase the efficiency of the TOE. Instead of forcing each networked computer to directly access the Server, 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 Server (or another Relay) for Fixlet and Action messages and Client machines utilize Relays exactly as they would Servers. Relays do not need to be dedicated computers and can connect to other Relays for additional efficiency. When Relays are installed they report themselves to their corresponding Server, and subsequently the Clients are made aware of them and can access their Server via available Relays. Relays work by:

- Relieving Downstream Traffic: The Server distributes files such as patches or software packages and Fixlet messages to Clients. Relays can be set up to ease this burden so that the Server does not need to distribute the same file to every Client. Instead the file is sent

once to the Relay, which in turn distributes it to the Clients. In this model, the Client connects directly to the Relay and does not need to connect to the Server.

- Reducing Upstream Traffic: In the upstream direction, Relays can compress and package data (including Fixlet relevance, action status and retrieved properties) from the Clients for greater efficiencies. During this process Relays may optionally decrypt and re-encrypt data sent from clients to ensure compression efficiencies. Administrators must designate which Relays are able to re-encrypt data.
- Reducing Congestion on Low-Bandwidth Connections: If the Server is communicating with computers in a remote office over a slow connection, designation of one of those computers as a Relay can help. Then, instead of sending patches over the slow connection to every Client independently, the Server only sends a single copy to the Relay(s) as needed and then the Relay distributes the file to the other computers in the remote office over its own fast LAN to effectively remove the slow connection bottleneck for remote groups on the network.
- Reducing Load on the Server: The Server has many duties including handling connections from Clients and Relays. At any given instant, the Server is limited in how many connections it can effectively service; however, Relays can buffer multiple Clients and upload the compressed results to the Server. Relays also distribute downloads to individual Clients, further reducing the workload of the Server and allowing the TOE to operate faster and more efficiently.

Note that Relays are considered an optional TOE component – they are not required for the operation of the TOE but are available as part of the product and so can be installed and enabled for use in the evaluated configuration.

Relays listens 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 them 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 messages from Servers regarding 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.

1.4.2 Physical Boundaries and delivery

The TOE is a set of software components and the TOE's guidance documents. The TOE's physical boundary is defined by the TOE installation image.

- TOE installation image. This image contains the following.
  - BigFix Server 10.0.1.41
  - BigFix Client 10.0.1.41
  - BigFix Relay 10.0.1.41
  - BigFix Console 10.0.1.41

- Administration tool hot fix. This image contains the fixed version of the BigFix Administration tool required by the Common Criteria configuration.
    - BigFix Administration Tool 10.0.1.45

  Refer to the “BigFix version 10.0.1 Common Criteria Configuration Guide” for the installation instructions.

The TOE documentation consists of the following documents.

- BigFix Installation guide version 10.0 (PDF)
- BigFix Configuration guide version 10.0 (PDF)
- BigFix version 10.0.1 Action Script Guide (PDF)
- BigFix version 10.0.1 REST API (PDF)
- BigFix Console Operator's Guide version 10.0 (PDF)
- BigFix version 10.0.1 Relevance Guide (PDF)
- BigFix version 10.0.1 Common Criteria Configuration Guide

To obtain the TOE, the HCL website contains the downloadable TOE image and guidance documentation.

1.4.3 TOE security features

This section identifies the security functions provided by the TOE and the logical boundary of the TOE.

- Cryptographic Support,
- User Data Protection,
- Identification and Authentication (I&A),
- Security Management,
- Protection of the TOE Security Functions (TSF),
- Trusted path/channels.

1.4.3.1 Cryptographic support

The TOE performs cryptographic operations by providing Rivest-Shamir-Adleman (RSA) public/private key pairs for the purpose of digitally signing Actions within the infrastructure. These signatures enable the TOE to authenticate and ensure the integrity of remedial Actions as they are collected, distributed and deployed by various components of the TOE across the network.

To protect the data collected from the clients, the TOE generates RSA public/private key pairs used for encryption that are distributed from the Server to Clients and Relays. The public key is distributed in the Masthead: a container that is digitally signed using the separate signing key pairs described above. The data gathered on Clients is encrypted using the encryption key pair, delivered over the network, and decrypted on the Server.

1.4.3.2 User data protection

The TOE provides an Action Information Flow Control SFP that controls the application of Actions via Clients. Actions are provided by Operators. The TOE Server facilitates the distribution of applicable Actions to Clients and those Clients will only accept and apply Actions when it can be validated that they have come from an authorized source (e.g., an Operator assigned to manage that Client).

1.4.3.3 Identification and authentication

The TOE requires users (i.e., administrators) to be identified and authenticated before completing any security management related actions. Once the administrator is authenticated, the TOE enforces role-based rules and only Master Operator can change the rules and attributes on behalf of users.

1.4.3.4 Security management

The TOE provides security management functions that can only be accessed by authorized administrators. The TOE restricts the ability to determine the behavior of, disable, enable, modify the behavior of the functions (i.e., security policy rules and privileges) by role and the TOE also provides the functions necessary for effective management of the TOE security functions. All authorized administrators (i.e., the Site Administrator, Master Operators, and Operators) must login to the TOE with unique credentials. Access to management functions is based on assigned roles.

1.4.3.5 Protection of the TSF

The TOE enforces the use of TLS v1.2/HTTPS (Hypertext Transfer Protocol Secure) to protect the communications channel among all TOE components (Server, Console, Relay and Clients).

The TOE protects the security of data and operation results data gathered on networked client computers by encrypting this data before it is transmitted over the network.

1.4.3.6 Trusted Path / channels

The TOE enforces the use of TLS v1.2/HTTPS (Hypertext Transfer Protocol Secure) to protect the communications channel between the TOE and Fixlet Servers, which are considered external IT entities.

The TOE enforces the use of TLS v1.2 for the REST API interface provided by the TOE to allow external IT entities to perform security management functions.

1.4.4 Evaluated configuration

The evaluated configuration consists of the software, hardware, and guidance documentation specified in the above section 1.4.2. The evaluated configuration also imposes some limitations on the configuration of the product.

The specifications for configuring the TOE in the evaluated configuration are located in the guidance documentation listed in section 1.4.2. The consumer must read, understand, and follow the guidance documentation provided as part of the TOE for the evaluated configuration.

The following restrictions apply to the evaluated configuration:

- The Server component must be configured as an authenticating server.
- The Server component must be configured to use HTTPS to gather from external sites.
- The Server component must be configured to require TLS v1.2 for all HTTPS communications.
- The Server component must be configured to use “Enhanced security”.
- The Server component must be configured to use “FIPS mode”.
- The Relay components must be configured as and authenticating relay.
- The Client components must be configured to send “encrypted reports” only.
- Each user account can have only one role assigned to it.
- FTP must be disabled.
- SSH must be disabled.
- The Web Reports interface must be disabled or not installed.
- The WebUI interface must be not installed.

2 CC Conformance Claim

This Security Target is CC Part 2 conformant and CC Part 3 conformant, with a claimed Evaluation Assurance Level of EAL2.

Common Criteria [CC] version 3.1 revision 5 is the basis for this conformance claim.

This Security Target does not claim conformance to any Protection Profile.

3 Security Problem Definition (SPD)

This section summarizes the threats addressed by the TOE and assumptions about the intended environment of the TOE.

3.1 Threats

This section describes the threat model for the TOE and identifies the individual threats that are assumed to exist in the TOE environment.

The assets to be protected by the TOE are information or resources to be protected by the countermeasures of the TOE.

The threat agents having an interest in manipulating the data model can be categorized as either:

- Unauthorized individuals ("attackers") which are unknown to the TOE and its runtime environment.
- Authorized users of the TOE (i.e., administrators) who try to manipulate data that they are not authorized to access.

Threat agents originate from a well-managed user community within an organization internal network. Hence, only inadvertent or casual attempts to breach system security are expected from this community.

TOE administrators, including administrators of the TOE environment, are assumed to be trustworthy, trained and to follow the instructions provided to them with respect to the secure configuration and operation of the systems under their responsibility. Hence, only inadvertent attempts to manipulate the safe operation of the TOE are expected from this community.

3.1.1 Threats countered by the TOE

T.UNAUTHORIZED_ACTION

Threat agents may supply either counterfeit Actions to Client Computers, or valid Actions to Client Computers that compromise the security features of the TOE or contain incorrect updates for the Client Computers.

T.UNAUTHORIZED_ACCESS

Threat agents may gain access to the TSF, TSF data or user data without proper authorization (i.e. identification and authentication).

T.DATA_IN_TRANSIT

Threat agents may eavesdrop communication between TOE components located in different machines, or between the TOE and external IT entities, and be able to read or modify TSF data or user data.

3.2 Assumptions

3.2.1 Intended usage of the TOE

A.TRUSTED_DNS

When a Domain Name System (DNS) service is used by the network, the DNS provides trustworthy services.

A.PROTECTED_HW

The hardware providing the runtime environment for the TOE is protected against unauthorized physical access and modification.

A.DEDICATED_RTE

The hardware and software providing the runtime environment for the TOE Server and TOE Relays are used solely for this purpose and not to run other application software, except as required for the support of the TOE and for the management and maintenance of the underlying operating system and hardware.

3.3 Organizational Security Policies

P.TRAINED_ADMIN

The organization will ensure that administrators are aware of the security policies and procedures of their organization, are trained and competent to follow the guidance and documentation for managing the TOE, and correctly configure and operate the TOE in accordance with those policies and procedures.

4 Security Objectives

4.1 TOE Objectives

O.AUTHENTIC_ACTION

The TOE must ensure that Actions received by Client Computers are originated from the TOE server and are tamper-evident.

O.AUTHORIZED_ACTION

The TOE shall ensure that Actions are applied only to the Client Computers authorized by an administrator.

O.I&A_ADMIN

The TOE shall ensure that TOE administrators are identified and authenticated prior to performing security-relevant actions.

O.MANAGE

The TOE shall provide security management capability to manage the TOE's claimed security functionality and ensure that these management capabilities are only available to TOE administrators.

O.PROTECTED_COMM

The TOE shall protect sensitive data (TSF data and user data) in transit between TOE components, and between the TOE and external IT entities, from disclosure and modification.

4.2 Operational Environment Objectives

OE. TRUSTED_DNS

When a Domain Name System (DNS) service is used by the network, the DNS service shall be trustworthy.

OE.PROTECTED_HW

The hardware providing the runtime environment for the TOE shall be protected against unauthorized physical access and modification.

OE.DEDICATED_RTE

The hardware and software providing the runtime environment for the TOE Server and TOE Relays shall be used solely for this purpose and not to run other application software, except as required for the support of the TOE and for the management and maintenance of the underlying operating system and hardware.

OE.TRAINED_ADMIN

Administrators shall be aware of the security policies and procedures of their organization, shall be trained and competent to follow the guidance and documentation for managing the TOE, and shall correctly configure and operate the TOE in accordance with those policies and procedures.

4.3 Security Objectives Rationale

4.3.1 Coverage analysis

The following table provides a mapping of security objectives to policies, threats and assumptions, showing that each objective covers at least one policy, threat, or assumption, and viceversa.

Table 4-1: Objective-to-SPD coverage analysis

Objective	**P.TRAINED_ADMIN**	**T.DATA_IN_TRANSIT**	**T.UNAUTHORIZED_ACCESS**	**T.UNAUTHORIZED_ACTION**	**A. TRUSTED_DNS**	**A.DEDICATED_RTE**	**A.PROTECTED_HW**
O.AUTHORIZED_ACTION				X			
O.AUTHENTIC_ACTION				X			
O.I&A_ADMIN			X				
O.MANAGE	X						
O.PROTECTED_COMM		X					
OE.TRUSTED_DNS					X		
OE.PROTECTED_HW							X
OE.DEDICATED_RTE						X	
OE.TRAINED_ADMIN	X						

4.3.2 Sufficiency analysis

The following tables provide sufficiency rationale for the mappings of objectives to the security problem definition.

Table 4-2: Sufficiency of objectives countering threats

Threat	**Rationale for security objectives**
T.UNAUTHORIZED_ACTION	This threat is diminished by: • O.AUTHENTIC_ACTION requires that Actions received by Client Computers are originated in the TOE and protected from tampering. • O.AUTHORIZED_ACTION requires that Actions are applied only to the Client Computers that are authorized by an operator.
T.UNAUTHORIZED_ACCESS	This threat is diminished by: • O.I&A_ADMIN requires that TOE administrators are identified and authenticated prior to performing security-relevant actions.
T.DATA_IN_TRANSIT	This threat is diminished by: • O.PROTECTED_COMM requires protection for sensitive data (TSF

Threat	Rationale for security objectives
	data and user data) in transit between TOE components, and between the TOE and external IT entities, from disclosure and modification.

Table 4-3: Sufficiency of objectives satisfying assumptions

Assumption	Rationale for security objectives
A. TRUSTED_DNS	This assumption is satisfied by: • OE. TRUSTED_DNS requires that the Domain Name System (DNS) service used by the network is trustworthy.
A.PROTECTED_HW	This assumption is satisfied by: • OE.PROTECTED_HW requires that the hardware providing the runtime environment for the TOE is protected against unauthorized physical access and modification.
A.DEDICATED_RTE	This assumption is satisfied by: • OE.DEDICATED_RTE requires that the hardware and software providing the runtime environment for the TOE Server and TOE Relays is used solely for this purpose and not to run other application software, except as required for the support of the TOE and for the management and maintenance of the underlying operating system and hardware.

Table 4-4: Sufficiency of objectives satisfying organizational security policies

Policy	Rationale for security objectives
P.TRAINED_ADMIN	This organization security policy is satisfied by: • O.MANAGE requires security management capability to manage the TOE's claimed security functionality and ensure that these management capabilities are only available to TOE administrators. • OE.TRAINED_ADMIN requires that administrators are aware of the security policies and procedures of their organization, are trained and competent to follow the guidance and documentation for managing the TOE, and that they correctly configure and operate the TOE in accordance with those policies and procedures.

5 Extended Components Definition

No extended components definitions are used in this ST.

6 Security Requirements

This section describes the Security Functional Requirements (SFRs) for the TOE as well as the Security Assurance Requirements (SARs) for the TOE.

6.1 TOE Security Functional Requirements

The following table shows the SFRs for the TOE:

Requirement Class	**SFR List**
FCS: Cryptographic Support	FCS_CKM.1(1): Cryptographic key generation (asymmetric)
	FCS_CKM.1(2): Cryptographic key generation (symmetric)
	FCS_CKM.2: Cryptographic key distribution
	FCS_CKM.4: Cryptographic key destruction
	FCS_COP.1: Cryptographic operations
FDP: User Data Protection	FDP_IFC.1: Subset information flow control
	FDP_IFF.1: Simple security attributes
FIA: Identification and Authentication	FIA_ATD.1: User attribute definition
	FIA_UAU.2: User authentication before any action
	FIA_UID.2: User identification before any action
FMT: Security Management	FMT_MSA.1: Management of security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_MTD.1: Management of TSF data
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_ITT.1: Basic internal TSF data transfer protection
FTP: Trusted path/channels	FTP_ITC.1 Inter-TSF trusted channel

6.1.1 Conventions

SFR assignments are in **bold text**. SFR selections are in ***bold and italic text***. SFR refinement additions are in <u>underlined</u> text. SFR refinement deletions are in ~~strikethrough~~ text.

6.1.2 FCS: Cryptographic Support

6.1.2.1 FCS_CKM.1(1) Cryptographic key generation (asymmetric)

FCS_CKM.1.1 The TSF shall generate <u>asymmetric</u> cryptographic keys in accordance with a specified cryptographic key generation algorithm **defined in Table 6-1** and specified cryptographic key sizes **defined in Table 6-1** that meet the following: **the standards defined in Table 6-1**.

Table 6-1: Asymmetric cryptographic key generation

Key generation algorithms	Key sizes	Standards	Purpose
Digital Signature Algorithm (DSA)	2048-bit	[FIPS 186-4] Appendix B.1	Key exchange in TLSv1.2 protocol.
Elliptic Curve DSA (ECDSA)	P-256, P-384, P-521	[FIPS 186-4] Appendix B.4	Key exchange in TLSv1.2 protocol.
RSA	2048, 4096 bits	[FIPS 186-4] Appendix B.3	Client and server authentication in TLSv1.2 protocol. Digital signature generation and verification of Actions. Key wrapping of symmetric keys.

6.1.2.2 FCS_CKM.1(2) Cryptographic key generation (symmetric)

FCS_CKM.1.1 The TSF shall generate symmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm **defined in Table 6-2** and specified cryptographic key sizes **defined in Table 6-2** that meet the following: **the standards defined in Table 6-2**.

Table 6-2: Symmetric cryptographic key generation

Key generation algorithms	Key sizes	Standards	Purpose
AES	256 bits	[FIPS 186-4] Appendix B.1	Encryption and decryption of Message Level Encryption and Mailboxes.

6.1.2.3 FCS_CKM.2 Cryptographic key distribution

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **defined in Table 6-3** that meets the following: **standards defined in Table 6-3**.

Table 6-3: Cryptographic key establishment

Key distribution methods	Standards	Purpose
Diffie-Hellman key agreement	[SP800-56A-Rev3] section 6.1.2, [RFC5246]	Key exchange in TLSv1.2 protocol.
Elliptic Curve Diffie-Hellman key agreement	[SP800-56A-Rev3] section 6.1.2, [RFC4492]	Key exchange in TLSv1.2 protocol.
RSA key wrapping	[SP800-56B-Rev2]	Key wrapping of AES keys used for symmetric encryption.
TLSv1.2 Key Derivation Function (Pseudorandom function)	[RFC5246] section 6.3	Derivation of encryption keys, HMAC keys and IVs in TLSv1.2 protocol.

6.1.2.4 FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in volatile memory in accordance with a specified cryptographic key destruction method **zeroization of CSPs** that meets the following: **[FIPS140-2]**.

6.1.2.5 FCS_COP.1 Cryptographic operations

FCS_COP.1.1 The TSF shall perform **the cryptographic operations defined in Table 6-4** in accordance with a specified cryptographic algorithm **defined in Table 6-4** and cryptographic key sizes **defined in Table 6-4** that meet the following: **the standards defined in Table 6-4**.

Table 6-4: Cryptographic operations

Algorithm	Operations	Key sizes	Standards	Purpose
AES in CBC mode	Encryption and decryption.	128 and 256 bits	[FIPS 197] [SP800-38D]	TLSv1.2 protocol. Encryption and decryption of Message Level Encryption and Mailboxes.
AES in GCM mode	Encryption and decryption.	128 and 256 bits	[FIPS 197] [SP800-38D]	TLSv1.2 protocol.
SHA-1, SHA-256, SHA-384	Message digest generation.	none	[FIPS 180-4]	TLSv1.2 protocol. Digital signature generation and verification. X.509 certificate generation and validation.
HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384	Message authentication code.	256 bits	[FIPS 198-1]	TLSv1.2 protocol.
RSA PKCS#v1.5	Digital signature generation and verification.	2048, 4096 bits	[FIPS 186-4]	TLSv1.2 protocol. Digital signature generation and verification of Actions. X.509 certificate generation and validation.
RSA PKCS#v1.5	Encryption and decryption	2048, 4096 bits	[FIPS 186-4]	RSA key wrapping.

6.1.3 FDP: User Data Protection

6.1.3.1 FDP_IFC.1: Subset information flow control

FDP_IFC.1.1 The TSF shall enforce the **Action Information Flow SFP** on

- **subjects: User, Client Computer;**
- **information: Action;**
- **operations: Deploy Action on Client Computer**.

6.1.3.2 FDP_IFF.1: Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the **Action Information Flow SFP** based on the following types of subject and information security attributes:

- **subjects:**
  - **User**
    - **Username**
    - **Role (Master Operator or Operator)**
    - **Permission to create and deploy actions (for Operator role)**
    - **Client Computers authorized to administer (for Operator role)**
  - **Client Computer**
    - **Computer Identity**
    - **Inspectable properties**
    - **Users authorized to administer the Client Computer**
    - **Subscribed Operator Sites**
    - **Server Signing Certificate**
    - **Client Computer's Certificate**
- **information:**
  - **Action**
    - **Site where the action was deployed (Master Action Site, Operator Site, Mailbox)**
    - **Username (issuer, only for Mailbox)**
    - **Relevance Clauses**
    - **Digital signature**

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

a) the User:
   a. has the Master Operator role, or
   b. has the Operator role, and
      i. has permission to deploy (take) actions;
      ii. has authorization to administer the Client Computer;

b) the Action was deployed by the User:
   a. in the Master Action Site, and the User has the Master Operator role;
   b. in the Operator Site corresponding to the User, the User has the Operator role, and the Client Computer is subscribed to that site;

c. **in the Client Computer's mailbox;**

c) **if the Action was deployed in the Client Computer's mailbox, it can be decrypted using the Client Computer's Certificate;**
d) **the Digital Signature of the Action is successfully verified against the Server Signing Certificate;**
e) **if the Action was deployed in the Client Computer's mailbox, the Username of the action corresponds to a User authorized to administer the Client Computer;**
f) **the Relevance Clauses of the Action are evaluated against inspectable properties of the Client Computer and all the conditions in the Relevance Clauses are met;**
g) **If all the above rules are met, the Action is applied on the Client Computer.**

FDP_IFF.1.3 The TSF shall enforce the **no additional information flow control SFP rules**.

FDP_IFF.1.4 The TSF shall explicitly authorize an information flow based on the following rules: **no additional access rules**.

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **no additional access rules.**

6.1.4 FIA: Identification and Authentication

6.1.4.1 FIA_ATD.1: User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: **user name, password, role**.

6.1.4.2 FIA_UAU.2: User authentication before any action

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

6.1.4.3 FIA_UID.2: User identification before any action

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

6.1.5 FMT: Security Management

6.1.5.1 FMT_MSA.1: Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the **Action Information Flow Control SFP** to restrict the ability to **perform the operations defined in Table 6-5** on the security attributes **defined in Table 6-5** to **the roles defined in Table 6-5.**

Table 6-5: Security attributes for the Action Information Control SFP

Subject	Security attribute	Operations	Roles
User	Username	Create User	Master Operator
	Role	Assign User	Master Operator
	Permission to Create and Deploy Actions	Enable, Disable	Master Operator
	Authorized Client Computers	Add, Remove	Master Operator
Client Computer	Computer ID	Manage Client Computer	Master Operator
	Inspectable properties	None	N/A
	Users authorized to administer the Client Computer	Add, Remove	Master Operator
	Subscribed Operator sites	Add, Remove	Master Operator
	Server signing certificate	Create	Site Administrator
		Manage Client Computer	Master Operator
	Client Computer's certificate	Manage Client Computer	Master Operator
Action	Site where the action was deployed	Deploy Action	Master Operator Operator
	Username (issuer, only for Mailbox)	Deploy Action	
	Relevance Clauses	Create Action	
	Digital signature	None	N/A

6.1.5.2 FMT_MSA.3: Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **Action Information Flow Control SFP** to provide ***restrictive*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow **the roles defined in Table 6-5** to specify alternative initial values to override the default values when an object or information is created.

6.1.5.3 FMT_MTD.1: Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to ***perform the operations defined in Table 6-6*** on the **TSF data defined in Table 6-6** to **the authorized roles marked as "Yes" in Table 6-6.**

Table 6-6: Management of TSF data

TSF data	Operations	Site Administrator	Master Operator	Operator
Certificates	Manage	Yes	No	No
Master Action Site	Initialize, Manage	No	Yes	No
User	Manage	No	Yes	No
Client Computers	Manage	No	Yes	Yes
BigFix Roles	Manage	No	Yes	No
User Permissions	Manage	No	Yes	No
Authorization of Operator users to Client Computers	Add, Remove	No	Yes	Yes
Action	Create, Deploy (take)	No	Yes	Yes

Application Note: Permissions in the Operator column marked as "Yes" also depends on the BigFix roles and user permissions assigned to users with the Operator role.

6.1.5.4 FMT_SMF.1: Specification of Management Functions

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

- **Master Action Site Initialization**
- **Configuration of communication to external Fixlet Servers**
- **Manage Master Operator users**
- **Manage BigFix roles and user permissions**
- **Manage Operator users**
- **Manage Client Computers**
- **Manage assignment of Client Computers to Operator users**
- **Manage Certificates**
- **Create Custom Actions**
- **Deploy (take) Actions**

6.1.5.5 FMT_SMR.1: Security roles

FMT_SMR.1.1 The TSF shall maintain the roles **Site Administrator, Master Operator, Operator**.

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

6.1.6 FPT: Protection of the TSF

6.1.6.1 FPT_ITT.1: Basic internal TSF data transfer

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

6.1.7 FTP: Trusted path/channels

6.1.7.1 FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit ***the TSF, another trusted IT product*** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **requesting user data from a Fixlet Server, accepting Security Management function requests via the REST API interface**.

6.2 Security Functional Requirements Rationale

6.2.1 Coverage

The following table provides a mapping of SFR to the security objectives showing that each security functional requirement addresses at least one security objective.

Table 6-7: Sufficiency of SFR satisfying Security Objectives

Security Functional Requirements	**O. AUTHENTIC_ACTION**	**O. AUTHORIZED_ACTION**	**O.I&A_ADMIN**	**O. MANAGE**	**O. PROTECTED_COMM**
FCS_CKM.1(1): Cryptographic key generation (asymmetric)				X	X
FCS_CKM.1(2): Cryptographic key generation (symmetric)				X	
FCS_CKM.2: Cryptographic key distribution					X
FCS_CKM.4: Cryptographic key destruction					X
FCS_COP.1: Cryptographic operations	X	X			X
FDP_IFC.1: Subset information flow control		X			
FDP_IFF.1: Simple security attributes		X			
FIA_ATD.1: User attribute definition			X		
FIA_UAU.2: User authentication before any action			X		
FIA_UID.2: User identification before any action			X		
FMT_MSA.1: Management of security attributes				X	
FMT_MSA.3: Static attribute initialization				X	
FMT_MTD.1: Management of TSF Data				X	
FMT_SMF.1: Specification of management functions				X	
FMT_SMR.1: Security roles				X	
FPT_ITT.1: Basic internal TSF data transfer protection				X	X
FTP_ITC.1: Inter-TSF trusted channel					X

6.2.2 Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives.

Table 6-8: Sufficiency of SFR addressing Security Objectives

Security Objectives	Rationale
O.AUTHENTIC_ACTION	FCS_COP.1: The TOE is required to perform RSA signature generation and verification operations in order to ensure that Actions are generated by the server and received by Client Computers without tampering.
O.AUTHORIZED_ACTION	FCS_COP.1: The TOE is required to perform RSA signature generation and verification operations in order to ensure that Actions are generated by authorized administrators. FDP_IFC.1 and FDP_IFF.1: The TOE is required to enforce an information flow control policy on Actions between authorized users and Client Computers.
O.I&A_ADMIN	FIA_ATD.1: The TOE is required to maintain security attributes for users. FIA_UAU.2: The TOE is required to authenticate users before allowing any TSF-mediated actions. FIA_UID.2: The TOE is required to identify users before allowing any TSF-mediated actions.
O.MANAGE	FCS_CKM.1(1) and FCS_CKM.1(2): The TOE is required to generate asymmetric and symmetric cryptographic keys . FMT_MSA.1: The TOE is required to restrict access to security attributes appropriately. FMT_MSA.3: The TOE is required to enforce the information flow control SFPs to provide restrictive access to authorized users. FMT_MTD.1: The TOE restricts the ability to perform management operations on TSF Data to authorized roles. FMT_SMF.1: The TOE is required to offer the functions necessary for effective management of the TOE security functions as well as the targeted IT environment machines. FMT_SMR.1: The TOE is required to define authorized administrators that will be able to perform the applicable security management functions. FPT_ITT.1: The TOE is required to protect communications between TOE components so that instructions cannot be corrupted, modified, or observed (where access to sensitive information might allow a potential attacker to identify a weakness).
O.PROTECTED_COMM	FCS_CKM.1(1), FCS_CKM.2, FCS_CKM.4 and FCS_COP.1 implement the cryptographic functionality required for the TLS v1.2 protocol. FPT_ITT.1: The TSF protects inter-TSF data transfer between the distributed parts of the TOE by using the TLSv1.2 protocol. FTP_ITC.1: The TSF provides a communication channel between itself and another trusted IT.

6.2.3 Security requirements dependency analysis

The following table demonstrates the dependencies of SFRs modeled in CC Part 2 and how the SFRs. for the TOE resolve those dependencies.

Table 6-9: TOE SFR dependency analysis

Security functional requirement	Dependencies	Resolution
FCS_CKM.1(1)	[FCS_CKM.2 or FCS_COP.1]	FCS_CKM.2
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.1(2)	[FCS_CKM.2 or FCS_COP.1]	FCS_CKM.2
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.2	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(1)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(1), FCS_CKM.1(2)
FCS_COP.1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(1), FCS_CKM.1(2)
	FCS_CKM.4	FCS_CKM.4
FDP_IFC.1	FDP_IFF.1	FDP_IFF.1
FDP_IFF.1	FDP_IFC.1	FDP_IFC.1
	FMT_MSA.3	FMT_MSA.3
FIA_ATD.1	No dependencies	
FIA_UAU.2	FIA_UID.2	FIA_UID.2
FIA_UID.2	No dependencies	
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.1
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3	FMT_MSA.1	FMT_MSA.1
	FMT_SMR.1	FMT_SMR.1
FMT_MTD.1	FMT_SMF.1	FMT_SMF.1
	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1	No dependencies	
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FPT_ITT.1	No dependencies	
FTP_ITC.1	No dependencies	

6.3 Security Assurance Requirements

The Security Assurance Requirements (SARs) for the TOE are the EAL2 components as specified in Part 3 of the CC. No operations are applied to the assurance components.

Table 6-10: SARs

Security assurance class	Security assurance requirement
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.2: Security-enforcing functional specification (FSP)
	ADV_TDS.1: Basic design
AGD: Guidance documents	AGD_OPE.1: Operation User Guidance
	AGD_PRE.1: Preparative Procedures
ALC: Life-cycle support	ALC_CMC.2: Use of a CM system
	ALC_CMS.2: Parts of the TOE CM coverage
	ALC_DEL.1: Delivery procedures
ASE: Security Target evaluation	ASE_CCL.1: Conformance claims
	ASE_ECD.1: Extended components definition
	ASE_INT.1: ST introduction
	ASE_OBJ.2: Security objectives
	ASE_REQ.2: Derived security requirements
	ASE_SPD.1: Security problem definition
	ASE_TSS.1: TOE summary specification
ATE: Tests	ATE_COV.1: Evidence of coverage
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2: Vulnerability analysis

7 TOE Summary Specification

7.1 TOE Security Functionality

The TOE supports the following major security features:

- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF

The following subsections provide more detail on how the TOE meets the requirements.

7.1.1 Cryptographic support

The TOE provides the following cryptographic functionality:

- The TLSv1.2 protocol, for protecting communication between:
  - the BigFix server, BigFix relays, and BigFix clients (except UDP communications);
  - the BigFix console and the BigFix server;
  - the TOE and the Fixlet servers; and
  - the REST API interface (exposed by the TOE) and a REST API client.
- RSA Digital Signature Generation and Verification, to provide integrity and authenticity of Actions.
- Validation of X.509 certificates, for verifying the chain of trust in TLSv1.2 and signed actions.
- AES Key Generation, AES encryption/decryption and RSA key wrapping for:
  - Data Encryption of BigFix client mailboxes stored in BigFix servers or relays.
  - Data Encryption of reports sent by BigFix clients to BigFix relays and servers, and by BigFix relays to BigFix relays and servers.
- AES encryption/decryption for the protection of user passwords.

The TOE implements all cryptographic functionality using the OpenSSL package (i.e. OpenSSL version 1.0.2u) in FIPS-enabled mode. OpenSSL performs zeroization of cryptographic keys in volatile memory in accordance to the requirements stated in [FIPS140-2].

7.1.1.1 Transport Layer Security (TLS)

The TOE implements the TLS protocol for providing integrity and confidentiality in all communication paths between the TOE components, and between the TOE and external IT entities (e.g. BigFix Fixlet servers, external web sites, REST API applications). The TOE acts both as a TLS client and a TLS server.
In the evaluated configuration, the TOE supports TLS version 1.2 with the following cipher suites:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

The TOE uses the following cryptographic algorithms to implement these cipher suites:

- Ephemeral Diffie-Hellman (DHE) and Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) for key establishment.
- TLSv1.2 KDF (pseudorandom function) for deriving session keys.
- RSA signature generation and signature verification for TLS client and server authentication.
- AES 128-bit and 256-bit cryptographic algorithms in GCM and CBC modes for symmetric cryptography.
- HMAC-SHA-256 and HMAC-SHA-384 for keyed-hash message authentication.

The TOE creates Elliptic Curve Diffie-Hellman ephemeral keys of at least 256 bits, per RFC 7919. The TOE uses automatic curve selection for the TLS server, which chooses the same preferred curves as the client uses by default. The order of preference is the following:

- sect571r1
- sect571k1
- secp521r1
- sect409k1
- sect409r1
- secp384r1
- sect283k1
- sect283r1
- secp256k1
- prime256v1

The TOE's HTTPS implementation complies with [RFC2818]. The BigFix Console verifies the validity of the TOE's X.509v3 certificate during the TLS handshake using signature verification. Once the HTTPS session is established, the TOE uses the identification and authentication information provided by the administrator to identify and authenticate the TLS client.

7.1.1.2 SFR coverage

The TOE security functionality satisfies the following SFRs:

- FCS_CKM.1(1)
- FCS_CKM.1(2)
- FCS_CKM.2
- FCS_CKM.4
- FCS_COP.1

7.1.2 User data protection

The TOE provides the Action Information Flow Control Security Function Policy (SFP), (hereinafter referred to as Action SFP), which controls the ability to apply Actions sent by authorized

administrators to Client Computers. The TOE has the ability to control actions reflected in the Information Flow Control SFP based on several rules that are based on properties of the Client Computer.

Administrators are the subjects that initiate the information flow. Administrators can either have:

- the Master Operator role, in which case they are authorize to apply Actions to all Client Computers;
- the Operator role, in which case they must be granted to “take Actions” and entitled to administer the Client Computer

Administrators connect to the TOE through the security management interface and instruct the TOE to apply a certain Action; this Action is stored in one of the repositories located in the TOE Server and known as “Sites”. All Master Operator administrators use the “Master Action Site” to store actions that are targeted to all Client Computers. Each Operator administrator has its own “Operator Site” to store actions that are targeted to the Client Computers that he/she administers. In addition, an Operator administrator can instruct the TOE to apply a certain Action to a specific Client Computer; in that case, the Action is stored in the repository dedicated to the Client Computer, known as Mailbox. There is one Mailbox for each Client Computer administered by the TOE.

The TOE server and (optionally) the TOE relays notify via UDP messages when new information is available to the Client Computers. The TOE Client (install in the Client Computer), either when it receives the notification from the TOE server or one of the TOE relays, or on a predetermined frequency, connects to the TOE server or the TOE relay and pull information from the Master Action site, all Operator sites where the Client Computer is subscribed to, and its own dedicated Mailbox.

Actions stored in the Master Action and Operator sites are digitally signed with the TOE server’s signing certificate, so the TOE Client can verify the authenticity of the Action. If the digital signature cannot be verified against the TOE server certificate, the Action is rejected.

Similarly, Actions in the Mailbox are stored in encrypted form; the TOE server generates a AES session key to encrypt the message, and wraps the symmetric key using the Client Computer’s certificate. The TOE Client pulls information from the Mailbox and verifies that the Action can be decrypted.

The TOE Client analyzes the applicability of the Action by obtaining properties from its own operating environment and evaluating the Relevance Clauses, which are conditions or rules that needs to be met in order to apply the Action. It also verifies, in the case of Actions retrieved from its Mailbox, that the administrator that sent the Action (the Issuer of the Action) is authorized to apply actions in the Client Computer.

If all the aforementioned conditions are met, the TOE client applies the action on the Client Computer.

Each Computer Client is initially installed with the TOE client and the “masthead” (containing the server signing X.509 certificate) of the TOE server. Every client generates a key pair (Client X.509 Certificate) at installation time, and sends the public key to the server.

The Server Signing certificate is used by the TOE client to verify the authenticity and integrity of the Actions received from the TOE server through the Master Action and Operator sites; the Client certificate is used to decrypt information received from the TOE server through the Computer Client’s mailbox.

The TOE Server coordinates the flow of information to and from Client Computers and stores the results in the TOE database. Server components operate quietly in the background without any direct intervention from an administrator.

The TOE enforces role-based rules and only a Master Operator can change and assign the rules on behalf of users. The master operator creates a set of rules and assigns them to the Operators. To enforce security only the Operators having an associated rule can login to the TOE console.

7.1.2.1 SFR coverage

The TOE security functionality satisfies the following SFRs:

- FDP_IFC.1
- FDP_IFF.1

7.1.3 Identification and authentication

All administrative interfaces require each user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that user. The TOE's administrative interfaces are:

- BigFix Administration Console (local GUI), used exclusively by the Site Administrator role.
- REST API (Programming Interface), used by administrators with the Master Operator and Operator roles. Administrators can use either the IEM CLI provided in the TOE package or any other REST API client.
- BigFix Console (remote GUI) , used by administrators with the Master Operator and Operator roles.

All administrative interfaces require each user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that user.

Identification and authentication from the Console or the REST API interfaces are performed by using the username and password of the administrator. The TOE (actually the Server component) retrieves the credentials stored in the database server and verifies that they match with the user and password entered by the administrator during logon.

The TOE supports passwords with a length of eight characters or greater, and a combination of upper-case letters, lower case letters, numbers, and special characters. The interfaces used to identify and authenticate provide obscured feedback of authentication data to the administrative user during the login process. Passwords are stored in the database server in encrypted form.

Identification and authentication from the Administrator Console are performed by using the Site Administrator Certificate created during the TOE installation, and the passphrase to access the private key, which is only known by the Site Administrator. The TOE verifies that the private key selected by the Site Administrator during logon matches the Site Administrator Certificate and that the private key can be accessed using the passphrase entered by the Site Administrator.

7.1.3.1 BigFix Console (local or remote GUI)

The administrator starts the BigFix Console program, which prompts the administrator with the "Login to HCL BigFix Console" dialog containing input fields for the "Server" (i.e., IP address of the TOE), and the administrator's "User name" and "Password." The administrator fills in all three fields and clicks Login. The GUI connects to the TOE using TLS/HTTPS and transmits the administrator's login credentials. When the login is successful, a window containing the full BigFix Console GUI appears. When the login is unsuccessful, a "Login Failed" dialog appears displaying "Incorrect username or password". The administrator clicks OK to dismiss the "Login Failed" dialog and is presented with the previous "Login to HCL BigFix Console" dialog.

7.1.3.2 REST API Client (local or remote)

The administrator uses a REST API client (e.g. IEM CLI, provided in the TOE package) to connect to the TOE. The BigFix REST API server (part of the TOE server component) uses the REST API “login” resource. The login resource requires input fields for the "Server" (i.e., IP address of the TOE), and the administrator's "User name" and "Password". The REST API client connects to the TOE using TLS/HTTPS and the specified credentials.

7.1.3.3 BigFix Administrative Console (local only)

The administrator starts the *BigFix Administration tool program*. The GUI presents a dialog to select the “Site signing key” (license.pvk) file, on which the administrator selects the file and clicks OK. The administrator is then presented with the "Site Admin Private Key Password” dialogue to which the administrator fills the Site Administrator password. When the login is successful, the administrator is placed into a window containing the full “BigFix Administration Tool” program. When the login is unsuccessful, a "Login Failed" dialog appears displaying "Incorrect password." The GUI connects to the TOE locally thus there is not transmission of credentials, or other data.

7.1.3.4 SFR coverage

The TOE security functionality satisfies the following SFRs:

- FIA_ATD.1
- FIA_UAU.2
- FIA_UID.2

7.1.4 Security management

The TOE provides the following security management functions that can only be accessed by authorized administrators:

- Master Action Site Initialization
- Configuration of communication to external Fixlet Servers
- Manage Master Operator users
- Manage BigFix roles and user permissions

- Manage Operator users
- Manage Client Computers
- Manage assignment of Client Computers to Operator users
- Manage Certificates
- Create Custom Actions
- Deploy (take) Actions

These security management functions can be performed by administrators depending on the administrator role assigned. The TOE provides three administration roles: Site Administrator, Master Operator and Operator. The Site Administrator uses the BigFix Administrator Tool to manage perform security management, and must identify and authenticate with the Site Certificate issued during the installation of the TOE. The Master Operator and the Operator roles use the BigFix Console and, optionally, the REST API interface. In both cases, administrators must identify and authenticate using their username and password.

7.1.4.1 Site Administrator role/user

This role is actually entitled to only one user (also called the Site Administrator user) that creates a set of keys that grants TOE administrator privileges. The site Administrator is unique and it is created at TOE installation time.

The Site Administrator performs security management functions using the TOE Administration Console. The Site Administrator role is reserved to perform top-level management tasks including:

- Create the first administrator with the Master Operator role.
- Manage Certificates
- Set global system and advanced options
- Security settings, replication settings, encryption settings
- Edit the Masthead.

7.1.4.2 Master Operator role

This role administers the configuration of the TOE by doing the following tasks:

- Create users with the Master Operator or Operator role
- Edit the management rights settings for administrators with the Operator role by assigning BigFix roles and user permissions, thus restricting the administrator to only perform a subset of the security management functions;
- Edit the management rights settings for administrators with the Operator role by assigning Client Computers;
- Subscribe or unsubscribe from Fixlet sites.

The first Master Operator is created at TOE installation time by the Site Administrator.

7.1.4.3 Operator role

This role manages the day-to-day operation of the TOE by doing the following tasks:

- Manage Client Computers
- Create and Edit Custom Actions

- Deploy (take) Actions

The security management functions allowed to an administrator with this role is subject to the management rights assigned by a Master Operator administrator when the administrator is created.

7.1.4.4 SFR coverage

The TOE security functionality satisfies the following SFRs:

- FMT_MOF.1
- FMT_MSA.1
- FMT_MSA.3
- FMT_MTD.1
- FMT_SMF.1
- FMT_SMR.1

7.1.5 Protection of the TSF

The TOE is comprised by the following software components running on different computers:

- the BigFix Server (one instance);
- the BigFix Console (one or more instances), used by administrators to perform security management functions;
- the BigFix Relay (optional, one or more instances), used as a proxy between BigFix clients and the BigFix server;
- the BigFix Client (one per each of the Client Computer administered by the TOE), running in Client Computers to apply Actions.

Communication between the TOE components is performed using the secure Hypertext Transfer Protocol (HTTPS) and the Transport Control Protocol (TCP). The TOE provides integrity and confidentiality of all communication using the TLS protocol version 1.2. See section 7.1.1 for a detailed description of this protocol.

The BigFix Server and the BigFix relays use Datagram Protocol (UDP) messages to notify availability of new content; also BigFix clients use the Internet Control Message Protocol (ICMP) to find the closest relay to communicate with. These messages do not expose sensitive information and are not protected.

The following table summarizes all the communication paths between the different TOE components and how they are protected:

Table 7-1: Communication paths between TOE components

TOE Components		Protocol	Description / Purpose	Protection Mechanism
Initiated by	**Connected To**			
Console	Server	TCP/IP	TOE Security Management by Administrators.	TLSv1.2
Client	Relay	ICMP	Find the closest relay available based on the number of network hops.	None
Client	Relay	HTTPS	Gathering Actions and Fixlet messages. Downloading files (such as patches). Sending results about client computer properties or status of actions/Fixlets.	TLSv1.2
	Server			
Relay	Client	UDP	Sending notifications about new information about new Fixlet messages, new Actions or downloads available.	None
Server				
Relay	Parent Relay	HTTPS	Gathering Actions and Fixlet messages. Downloading files (such as patches). Sending results about client computer properties or status of actions/Fixlets.	TLSv1.2
	Server			
Parent Relay	Relay	TCP/IP	Sending notifications about new information about new Fixlet messages, new Actions or downloads available.	TLSv1.2
Server				

The TOE provides authentication between the TLS endpoints through the use of X.509 certificates issued by the server. The TOE also enforces Identification and Authentication (I&A) of all BigFix Console users through the corresponding credentials (username and password).

7.1.5.1 SFR coverage

The TOE security functionality satisfies the following SFRs:

- FPT_ITT.1

7.1.6 Trusted Path / channels

The TOE starts communication with external BigFix Fixlets Servers and other external servers to obtain Fixlets and other information that are used by the TOE server. The TOE uses HTTPS and TLSv1.2 to protect the communication path. The TOE also verifies the authenticity of the external servers by validating the X.509 certificate provided by the TLS server endpoint.

The TOE server provides a REST API interface for allowing external IT products to connect to the TOE server and perform security management functions. For instance, the IEM Command-Line Interface (CLI) is a utility provided in the TOE package (not part of the TOE) that facilitates programmatic control of the BigFix Server using this interface. The TOE protects the REST API interface with TLSv1.2 acting as a TLS server. Once the TLS session is established, the TOE enforces

Identification and Authentication (I&A) of users through the corresponding credentials (username and password).

7.1.6.1 SFR coverage

The TOE security functionality satisfies the following SFRs:

- FTP_ITC.1

8 References

Reference	Description
[CC]	Common Criteria for Information Technology Security Evaluation Version 3.1R5, April 2017
[FIPS 140-2]	FIPS PUB 140-2 - Security Requirements For Cryptographic Modules
[FIPS 180-4]	Secure Hash Standard (SHS)
[FIPS 186-4]	Digital Signature Standard (DSS)
[FIPS 197]	Advanced Encryption Standard
[FIPS 198-1]	The Keyed-Hash Message Authentication Code (HMAC)
[RFC 2818]	HTTP Over TLS
[RFC 4492]	Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)
[RFC 5246]	The Transport Layer Security (TLS) Protocol Version 1.2
[RFC 7919]	Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS)
[SP800-38D]	NIST Special Publication 800-38D - Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC
[SP800-56A-Rev3]	NIST Special Publication 800-56A Revision 3 - Recommendation for Pair Wise Key Establishment Schemes Using Discrete Logarithm Cryptography
[SP800-90A]	Recommendation for Random Number Generation Using Deterministic Random Bit Generators