PrinterLogic Web Stack Client version 18.3
Security Target

Version 1.0
27 November 2019

Prepared for:

PrinterLogic
912 West 1600 South
St. George, UT 84770

Prepared by:

Accredited Testing and Evaluation Labs
6841 Benjamin Franklin Drive
Columbia, MD 21046

Table of Contents

1. SECURITY TARGET INTRODUCTION 1

1.1 SECURITY TARGET, TOE AND CC IDENTIFICATION 1
1.2 CONFORMANCE CLAIMS 2
1.3 CONVENTIONS 3
1.3.1 Terminology 3
1.3.2 Acronyms 4

2. PRODUCT AND TOE DESCRIPTION 6

2.1 INTRODUCTION 6
2.2 PRODUCT OVERVIEW 6
2.3 TOE OVERVIEW 7
2.4 TOE ARCHITECTURE 10
2.4.1 Physical Boundary 10
2.4.2 Logical Boundary 12
2.5 TOE DOCUMENTATION 13

3. SECURITY PROBLEM DEFINITION 14

4. SECURITY OBJECTIVES 15

5. IT SECURITY REQUIREMENTS 16

5.1 EXTENDED REQUIREMENTS 16
5.2 TOE SECURITY FUNCTIONAL REQUIREMENTS 17
5.2.1 Cryptographic Support (FCS) 18
5.2.2 User Data Protection (FDP) 20
5.2.3 Identification and Authentication (FIA) 20
5.2.4 Security Management (FMT) 21
5.2.5 Privacy (FPR) 21
5.2.6 Protection of the TSF (FPT) 22
5.2.7 Trusted Path/Channels (FTP) 22
5.3 TOE SECURITY ASSURANCE REQUIREMENTS 23

6. TOE SUMMARY SPECIFICATION 24

6.1 TIMELY SECURITY UPDATES 24
6.2 CRYPTOGRAPHIC SUPPORT 24
6.3 USER DATA PROTECTION 27
6.4 IDENTIFICATION AND AUTHENTICATION 28
6.5 SECURITY MANAGEMENT 29
6.6 PRIVACY 30
6.7 PROTECTION OF THE TSF 30
6.8 TRUSTED PATH/CHANNELS 32

7. PROTECTION PROFILE CLAIMS 33

8. RATIONALE 34

8.1 TOE SUMMARY SPECIFICATION RATIONALE 34

APPENDIX A: TOE USAGE OF THIRD-PARTY COMPONENTS 37

A.1 PLATFORM APIS 37
A.1.1 Windows PL Client 37

A.1.2 Linux PL Client 38
A.1.3 macOS PL Client 38
A.2 THIRD-PARTY LIBRARIES 38
A.2.1 Windows PL Client....38
A.2.2 Linux PL Client 38
A.2.3 macOS PL Client 39

LIST OF TABLES

Table 1 TOE Security Functional Components 17
Table 2 Assurance Components 23
Table 3 Cryptographic Functions 25
Table 4 Cryptographic Functions 25
Table 5 Sensitive Data 27
Table 6 Security Functions vs. Requirements Mapping....35

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is PrinterLogic Web Stack Client, version 18.3.

PrinterLogic Web Stack Client (PL Client) includes stand-alone executable clients for Windows, Mac OS, and Linux systems. These clients are installed on endpoint systems in an organization to facilitate direct IP printing. The PL Client interacts with an environmental Web Server application that is used for administration of the PL Client and handling of mediated printing activities that cannot be performed directly between a computer and an installed printer (e.g., AirPrint, Email Printing Service, Google Could Print). These mediated printing activities are also facilitated by a 'Service Host' capability that is contained within the PL Client application.

The focus of this evaluation is on the TOE functionality supporting the claims of version 1.3 of the Protection Profile for Application Software [App PP] and version 1.1 of the Functional Package for Transport Layer Security (TLS) [TLS Package]. The only capabilities covered by the evaluation are those specified in the aforementioned Protection Profile and Functional Package; no additional security functional claims are made by the product. The security functionality specified in [App PP] and [TLS Package] includes protection of security-relevant data at rest and in transit, any cryptographic functionality used to achieve this, and security of the interactions between the application(s) and their underlying platform(s). Where appropriate and permitted by the [App PP] and [TLS Package], this evaluation will identify areas where the TOE's underlying platform is used to support the TOE's implementation of its claimed security functionality.

The Security Target contains the following additional sections:

- Product and TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
-

- Protection Profile Claims (Section 0)
- Rationale (Section 8)

1.1 Security Target, TOE and CC Identification

ST Title – PrinterLogic Web Stack Client version 18.3 Security Target

ST Version – Version 1.0

ST Date – 27 November 2019

TOE Identification – PrinterLogic Web Stack Client version 18.3. The specific components of the TOE include:

1. PrinterLogic Web Stack Client for Windows
    a. Supported on Windows 7 and later (32-bit and 64-bit)
2. PrinterLogic Web Stack Client for Linux
    a. Supported on Ubuntu 14.04 to 15.10 (32-bit and 64-bit)
3. PrinterLogic Web Stack Client for macOS
    a. Supported on OS X Mavericks (10.9) and up (64-bit)

Also note that the 'Service Host' functionality that is provided by the TOE is part of the PrinterLogic Web Stack Client application.

TOE Developer – PrinterLogic, LLC

Evaluation Sponsor – PrinterLogic, LLC

CC Identification – *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017*

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications: This ST is conformant to:

- *Protection Profile for Application Software, Version 1.3, 01 March 2019* and *Functional Package for Transport Layer Security (TLS), Version 1.1, February 12, 2019* with the following optional, selection-based, and objective SFRs:
    - FCS_CKM.1(1)
    - FCS_CKM.2
    - FCS_COP.1(1)
    - FCS_COP.1(2)
    - FCS_COP.1(3)
    - FCS_COP.1(4)
    - FCS_HTTPS_EXT.1
    - FCS_RBG_EXT.2
    - FCS_TLS_EXT.1
    - FCS_TLSC_EXT.1
    - FCS_TLSC_EXT.5
    - FIA_X509_EXT.1
    - FIA_X509_EXT.2

- The following NIAP Technical Decisions apply to this PP and Functional Package and have been accounted for in the ST development and the conduct of the evaluation, or were considered to be non-applicable :
    - TD0416: Correction to FCS_RBG_EXT.1 Test Activity
        - No change to ST; affects only test activities.
    - TD0427: Reliable Time Source
        - No change to ST; the ST includes the PP's assumptions by reference and therefore any changes to the assumptions are implicity made.
    - TD0434: Windows Desktop Application Test
        - No change to ST; affects only test activities.
    - TD0435: Alternative to SELinux for FPT_AEX_EXT.1.3
        - No change to ST; affects only test activities.
    - TD0437: Supported Configuration Mechanism
        - FMT_MEC_EXT.1.1 has been modified in ST.
    - TD0442: Updated TLS Ciphersuites for TLS Package
        - No change to ST; the TD affects the selections available in FCS_TLSC_EXT.1.1 but ST does not choose any selections that were affected by the TD.
    - TD0444: IPsec selections
        - N/A to TOE; the TD adds a selection for IPsec to FTP_DIT_EXT.1 but the TSF does not include IPsec so this selection is not chosen.
    - TD0445: User Modifiable File Definition
        - No change to ST; affects only test activities.
    - TD0465: Configuration Storage for .NET Apps
        - N/A to TOE; the TOE is not a .NET application.
    - TD0469: Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1
        - N/A to TOE; the TOE does not claim TLS server functionality.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 5, April 2017.
    - Part 3 Extended

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. An iterated SFR is indicated by a number in parentheses placed at the end of the component. For example, FCS_CKM.1(1) and FCS_CKM.1(2) indicate that the ST includes two iterations of the FCS_CKM.1 requirement: (1) and (2).

- Assignment: allows the specification of an identified parameter. Assignments are indicated using italics and are surrounded by brackets (e.g., [*assignment item*]). Note that an assignment within a selection would be identified in both italics and underline, with the brackets themselves underlined since they are explicitly part of the selection text, unlike the brackets around the selection itself (e.g., [selection item, [*assignment item inside selection*]]).
- Selection: allows the specification of one or more elements from a list. Selections are indicated using underlines and are surrounded by brackets (e.g., [selection item]).
- Refinement: allows the addition of details and non-technical changes to grammar and formatting. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Note that minor grammatical changes that do not involve the addition or removal of entire words (e.g., for consistency of quantity such as changing "meets" to "meet") do not have formatting applied.

- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.
- The ST does not highlight operations that have been completed by the PP authors, though it does preserve brackets to show where operations have been made.

1.3.1 Terminology

The following terms and abbreviations are used in this ST:

Admin Console	A GUI that is part of the environmental Web Server application. Used by administrators to configure PL Client settings, including whether to designate a given PL Client instance as a Service Host.
Administrator	Any member of the organization deploying the TOE who has credentials to access the Admin Console of the environmental Web Server.
AirPrint	A feature of Apple devices (Mac/iPhone/iPad) that is used to print documents on those devices via wireless network.
Console Print Application	An interactive program running on a printer or multifunction device, typically controllable through a touchscreen, that can be used to configure device settings and networking.
Delphi	An integrated development environment for the Object Pascal programming language.
Email Printing	A method of remote printing where a user can send an email message to a specific address that is monitored by a Service Host and either printed directly or held for pull printing.
Google Cloud Printing	A feature of Android/Chrome OS devices that is used to print documents on those devices via wireless network.
PrinterLogic Web Stack Client	The TOE. Runs on user machines and is used to handle installation of print drivers and remote printing. Can be configured as a Service Host to provide additional functionality for remote printing.
Pull Printing	A workflow for printing where a user requests to print a document but it is held by a Service Host instead of being immediately printed. The user can then choose to later release the document through the Release Portal, after which it is printed by a desired target printer. Often used in cases where physical custody of a printed document is essential but the user is not physically at or near the desired printer at the time the print job is initiated.
Release Portal	A GUI that is part of the environmental Web Server application. Used to direct a Service Host to release held documents for printing.
Remote Printing	Term used to collectively describe AirPrint, Email Printing, and Google Cloud Printing.

Self-Service Portal	A GUI that is part of the environmental Web Server application. Used for user installation of printer drivers and other basic configuration functionality that does not need to be restricted to administrators.
Service Host	A special configuration for the TOE. While configured as a Service Host, a the TOE can process remote printing and pull printing workflows. It does this by acting as a 'dummy' printer for AirPrint/Google Cloud Printing and/or by monitoring email boxes used for email printing.
User	An individual in the organization that lacks any specific privileges to administer the TOE.
Web Server	An environmental component. Application that runs various user- and administrator-facing user interfaces and acts as a central point for distributing configuration settings changes and release of held pull printing jobs to the various PL Clients (including Service Hosts).

1.3.2 Acronyms

AA	Assurance Activity
API	Application Programming Interface
ASLR	Address Space Layout Randomization
AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher-Block Chaining
CC	Common Criteria for Information Technology Security Evaluation
CEM	Common Evaluation Methodology for Information Technology Security
CPA	Control Panel Application
CRL	Certificate Revocation List
FIPS	Federal Information Processing Standard
GUI	Graphical User Interface
HMAC	Hashed Message Authentication Code
HTTP(S)	Hypertext Transfer Protocol (Secure)
IP	Internet Protocol
LDAPS	Lightweight Directory Access Protocol Secure
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
PL	PrinterLogic Web Stack
PII	Publicly Identifiable Information
PP	Protection Profile
RSA	Rivest, Shamir and Adleman (algorithm for public-key cryptography)
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Socket Layer Protocol
ST	Security Target
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functions
XMPP	Extensible Messaging and Presence Protocol

2. Product and TOE Description

The TOE is the PrinterLogic Web Stack Client v18.3 product. This section provides an overview of the capabilities of the product and then proceeds to describe the TOE itself in terms of its evaluated components and functional claims.

2.1 Introduction

PrinterLogic Web Stack Client is an on-premise application designed to simplify the management, migration, and provisioning of printers. PrinterLogic Web Stack Client facilitates features including centrally-managed direct IP printing, self-service installation of printer drivers, automated deployment of drivers, centralized reporting of printer usage, and pull/mobile printing.

PrinterLogic Web Stack Client is part of a client-server distribution. The TOE is the client portion of this distribution. It interacts with a central Web Server application in its operational environment.

2.2 Product Overview

This sub-section describes capabilities of the PrinterLogic Web Stack Client product as a whole. It should be noted that many of these capabilities are not covered within the scope of the evaluation. The scope of the evaluation is covered in the subsequent sub-sections that provide the TOE overview and describe the TOE architecture and physical and logical boundaries.

PrinterLogic Web Stack Client is a product that provides centralized services for user installation of print drivers as well as pull printing and cloud printing functionality.

PrinterLogic Web Stack Client can be used to facilitate direct IP printing. The application interacts with an environmental web server to allow users to install centrally-provisioned printer drivers. It also provides the ability to facilitate pull print jobs that are then released to a selected printer.

PrinterLogic Web Stack Client can also be used for centralized auditing and reporting of print jobs. This allows the organization to identify operational costs based on printer usage so that cost savings can be identified. It also uses SNMP to provide monitoring of individual printers and can generate emails in response to specific SNMP notifications.

Pull printing, also known as secure printing, refers to the case where a user will initiate a print job from their desktop workstation but it will not be printed immediately. Instead, PrinterLogic Web Stack Client will pass the print job to the environmental Web Server, which will ‘hold’ the print job until the user signals that they wish to ‘release’ (i.e. print) the job once they are physically present at the printer that they wish to have the job printed to. This is signaled either by the user notifying the server themselves (e.g. through a mobile device) or through the user signing on to the printer, which then uses its embedded control panel application (CPA) to retrieve and print the job. Cloud printing refers to the use of various network services to initiate print jobs in the absence of having installed printer drivers on the system. PrinterLogic Web Stack Client supports the following methods of cloud printing:

- Email printing: PrinterLogic Web Stack Client can be configured to communicate with an email inbox and automatically place any attachments in a pull print queue owned by the enterprise identity of the user that sent an email to the inbox (via reverse Active Directory lookup).
- Apple AirPrint: PrinterLogic Web Stack Client can be configured to broadcast itself as an AirPrint-compatible printer so that iOS users can use native iOS printing capabilities. The user provides their credentials to the TOE, which takes the print job and places it into a pull print queue owned by the user.
- Google Cloud Printing: PrinterLogic Web Stack Client can be configured either to interface with Google Cloud or to impersonate a Google Cloud server so that Android/Chromebook users can use native printing capabilities. Similar to email printing, The TOE will perform a reverse lookup of the user’s enterprise identity and hold the print job in a queue until released by that user.

2.3 TOE Overview

The Target of Evaluation (TOE) is comprised of the PrinterLogic Web Stack Client software application (PL Client). The PL Client software may be configured either as an end user client, or as a Service Host, which uses the PL Client software to provide cloud printing functionality. The TOE may be deployed on Windows, Linux, or macOS.

The focus of this evaluation is on the TOE functionality supporting the claims in the *Protection Profile for Application Software, Version 1.3* and *Functional Package for Transport Layer Security (TLS), Version 1.1*. Specifically, the following capabilities are within the scope of the evaluation:

- Trusted communications of user credential data, print spool data, and configuration data between the TOE and the operational environment. Note that the Windows PL Client relies on the underlying Windows OS platform to provide its HTTPS functionality when not operating as a Service Host. All other cryptographic functionality, including the Windows PL Client when operating as a Service Host, is implemented by TOE cryptography.
- The extent to which the TSF relies on platform-provided and third-party capabilities to perform its functionality.
- The extent to which data used to determine the behavior of the TSF is secured while at rest and in transit.
- The ability for the TOE to function on host platforms that are configured for secure operation.
- The ability of the TOE to interface with the low-level components of its host platforms in such a manner that the TOE cannot be used as an attack vector to exploit the host platforms.
- Pull printing and cloud printing functionality that require the TSF to handle sensitive print spool data.
- The ability of the organization deploying the TOE to perform timely and trusted security updates to it.

The basic workflows of the application that relate to the TSF are listed below:

Administrator change to client settings

1. In the operational environment, Administrator logs on to Web Server application and makes administrative changes.
2. Changes are propagated to SQL database on Web Server platform and transmitted to the TOE over TLS/HTTPS connection.
3. When received by the TOE, configuration files/registry values changed as needed.

User self-service

1. In the operational environment, user logs on to Web Server application.
2. On the environmental Web Server, user performs management of printer drivers or releases a held print job.
3. If printer drivers are not installed, Web Server will automatically transfer the desired drivers to the TOE running on the user's system, which then installs the drivers automatically.
4. If print job is released, Web Server will communicate with the TOE over TLS/HTTPS to instruct it to send the job to the desired printer.

Pull printing

1. User prints document on local system, choosing a pull printer installed by the TOE.
2. Print job is held by the TOE, which notifies the environmental Web Server over TLS/HTTPS that a print job is being held by that user.
3. User releases job by doing one of the following:
   - Logs in to the environmental Web Server, selects the held job, and specifies a printer to print it to
   - Authenticating to a printer that has been configured to send pull printing requests back to the environmental Web Server
4. Regardless of the method used to release the print job, the environmental Web Server will notify the TOE that the job has been released (via TLS/HTTPS) and the TOE will take the held job and send it to the platform's print spool for printing to the desired printer.

Email printing (standard)

1. On the environmental Web Server, an administrator configures a Service Host to poll a particular email box.
2. The configuration change is sent to the TOE (configured as a Service Host) via TLS/HTTPS and stored remotely in the environmental Web Server's database.
3. The configuration change is received by the TOE and stored locally in the registry/configuration files as needed.
4. When the user wishes to print a document, they will email it to the polled email box.
5. The TOE will retrieve any emails sent to the polled email box over IMAPS (IMAP over TLS).
6. The TOE will connect to the environmental Active Directory server over TLS to do a reverse lookup of the sender of the email (i.e., the user). BIND credential used to do this is retrieved from the environmental Web Server over HTTPS.
7. If the sender of the email is a valid AD user, the TOE will hold the print job and notify the environmental Web Server (over TLS/HTTPS) that a print job is being held for that user.
8. The user can then release their print job from the operational environment using one of the methods specified in 'pull printing' above.
9. If the sender of the email is not a valid AD user, the TOE will discard the email.

Email printing (direct)

1. On the environmental Web Server, an administrator configures the TOE (configured as a Service Host) to poll a particular email box.
2. In the Operational Environment, the administrator will configure one or more sub-domains of that email box to forward inbound messages to that email box via mail routing rules.
3. On the environmental Web Server, administrator configures the printer(s) that should be printed to based on the sub-domains of inbound messages.
4. The configuration changes are sent to the TOE via TLS/HTTPS and stored remotely in the environmental Web Server's database.
5. The configuration changes are received by the TOE and stored locally in the registry/configuration files as needed.
6. When the user wishes to print a document, they will email it to the polled email box.
7. The TOE will retrieve any emails sent to the polled email box over IMAPS (IMAP over TLS).
8. The TOE will connect to the environmental Active Directory server over LDAPS to do a reverse lookup of the sender of the email (i.e., the user). BIND credential used to do this is retrieved from the environmental Web Server over HTTPS.
9. If the sender of the email is a valid AD user, the TOE will immediately print the document using the specified printer.

Email printing (guest)

1. On the environmental Web Server, an administrator configures the TOE (configured as a Service Host) to poll a particular email box.
2. The administrator also designates a specific printer as a guest printer for that email box using the 'Allow print jobs to be emailed directly to this printer from guests' option.
3. When a guest user (i.e., not defined in the organizational Active Directory) wishes to print a document, they send it as an email to the polled email box.
4. The TOE will retrieve any emails sent to the polled email box over IMAPS (IMAP over TLS).
5. Since the user is a guest, there will be no AD user to look up.
6. The TOE will then take the email and immediately release it to be printed to the specified guest printer.

AirPrint cloud printing

1. On the environmental Web Server, administrator enables iOS printing and designates the TOE (configured as a Service Host) as a pull printer for this.
2. In the operational environment, an administrator creates pointer records on the DNS server to let any iOS device see the pull printer.
3. User prints a document on iOS device, specifying the TOE as the pull printer.
4. The user will be prompted for their Active Directory credentials, which are validated by the TOE using LDAPS.
5. Once the user credentials have been validated, the print job will be transmitted to the TOE via IPPS (IPP over TLS).

6. The TOE will notify the environmental Web Server that a job is being held for that user.
7. The user releases the job using one of the methods specified in 'pull printing' above.

Google Cloud printing (traditional)

1. On the Admin Console, administrator registers the TOE (configured as a Service Host) as a pull printer and configures it for mobile printing.
2. Configuration settings are transmitted to the TOE using TLS/HTTPS.
3. On the environmental Web Server, the administrator enables Google Cloud printing and specifies the email address and password of the Google Cloud print account where documents will be published.
4. The administrator registers the printer in Google (via pop-up redirect from the environmental Web Server to Google).
5. A user prints a document on their Android or Chromebook device to the pull printer.
6. When the user selects print, the print job is sent to the Google Cloud print server and subsequently converted to a PDF document.
7. The TOE will poll the Google Cloud print server and retrieve print jobs from the print server's queue.
8. The TOE will perform a reverse AD lookup of the user that submitted the print job.
9. If the user is recognized, the TOE pulls down the print job from the Google Cloud print server over TLS/HTTPS (XMPP channel over TLS created by Google Cloud for job status and TLS/HTTPS for retrieval of the job itself) and held.
10. The TOE notifies the environmental Web Server over TLS/HTTPS that a pull print job is being held for the user.
11. The user can release the print job for printing using any of the methods specified in 'pull printing' above.

Google Cloud printing (local)

1. On the environmental Web Server, an administrator specifies the TOE (configured as a Service Host) to function as a local Google Cloud printer.
2. Configuration settings are transmitted to the TOE using TLS/HTTPS.
3. The TOE will automatically broadcast itself as a Google Cloud printer.
4. User on Android or Chromebook device will see the TOE as a valid printer.
5. When the user selects print, the print job is sent directly to the TOE using TLS/HTTPS rather than to the Google Cloud print server.
6. The TOE will perform a reverse AD lookup of the user that submitted the print job.
7. If the user is recognized, the print job is held by the TOE.
8. The TOE notifies the environmental Web Server over TLS/HTTPS that a pull print job is being held for the user.
9. The user can release the print job for printing using any of the methods specified in 'pull printing' above.

Software update

There are three separate methods of updating the PL Client application:

1. Manual download – the user acquires the installer file directly and runs it to update the PL Client.
2. Automatic download – the environmental Web Server contains installer files for the PL Client. This is triggered when a PL client 'checks in' with the Web Server and is notified that an update is available. This occurs when a user logs in to the workstation on which the PL Client resides (all platforms) or manually checks for an update by right-clicking the tray icon (Windows only).
3. Deployed update – the Administrator can download a copy of the PL Client installer from the environmental Web Server to the user workstation and deploy the update using command prompt, script, batch file, or group policy.

In all cases, the old version of the application will automatically be replaced with the updated version as part of the update process.

From these use cases, the user-facing responsibilities of the TOE include the following:

- Securely store configuration settings, received over a trusted channel, on the host platform.
- Notify the Web Server application over a trusted channel that a print job has been held.

- Receive a print job over a trusted channel.
- Securely store the contents of a print job on the host platform while the job is being held.

The TSF includes all security data and configuration settings needed to support this behavior. Not all configuration settings are security-relevant; information about how the PL Client is displayed to the user or the installation of print drivers is outside the scope of the TOE.

Once a print job has been released, the TSF sends the job to the print spool for printing by the host platform. Any transmission of the print job data from the host platform itself to the target printer is not under the control of the TSF and is therefore outside the scope of the TOE. Similarly, all configuration of network settings and email servers that allow print data to be received by the TOE are outside the scope of the TOE. The TSF is not responsible for the security of print data that is sent by the user to a component in the TOE's operational environment (e.g., the communication from the user to a mailbox used for email printing is non-TSF, but the communication between that mailbox and the TOE is part of the TSF).

2.4 TOE Architecture

The PrinterLogic Web Stack Client TOE is a software application that runs on several different platforms:

- PL Client (Windows) – a Delphi application with Python components (when configured as Service Host only)
- PL Client (Linux) – a Python application
- PL Client (macOS) – a Python application

The TOE consists of two subsystems: a local printer interface subsystem, which handles installation of print drivers and interfaces between the host platform and the Web Server (e.g., to prompt a user to hold or release a pull print job and notify the Web Server if the job is held); and a Service Host subsystem, which allows the PL Client to act as a virtual pull printer to securely retrieve emails and mobile print jobs to be held or released.

The TOE includes the following running processes:

- PrinterLogic Web Stack Client Interface
- PrinterLogic Web Stack Client Launcher (Windows only)
- PrinterLogic Web Stack Client Manager
- PrinterInstaller_SNMPMonitor (Service Host only)
- PrinterLogicServiceAirprint (Service Host only)
- PrinterLogicServiceEmail (Service Host only)
- PrinterLogicServiceGoogleCloudPrint (Service Host only)
- PrinterLogicControlPanelApp (Service Host only)
- PrinterLogicServiceManager (Service Host only)

2.4.1 Physical Boundary

The TOE consists of the following component:

- PL Client application (for Windows, Linux, and macOS; configured as a Service Host or not)

In Figure 1 below the TOE and its host platform are indicated by a red box. Everything else in Figure 1 including the server labeled as Printer Installer is considered a part of the environment.

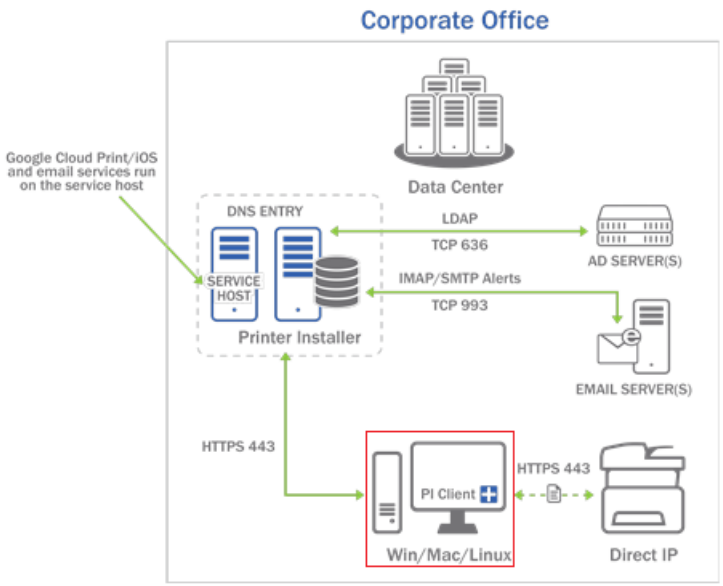

Figure 1: TOE Architecture

TSF-relevant remote interfaces are shown using solid green lines in Figure 1. Note that a Service Host may also reside on a system that is remote from the environmental Web Server. In these cases, the same interface that is used by the Web Server to communicate with a remote PL Client is used. Printing functionality is shown in this diagram using a dotted line. This is because facilitating printing activities is the primary purpose of the product; however, the actual act of sending documents from a system's print spool to a networked printer is still the responsibility of the underlying operating system.

The TOE has the following system requirements for its host platforms:

- Supported Windows platforms: Windows 7 (32-bit), Windows 8/8.1/10 (32- and 64-bit), Windows Server 2008 R2 (64-bit), Windows 2012/2016 (64-bit)
- Supported Linux platforms: Ubuntu (32- and 64-bit), Red Hat (64-bit)
- Supported macOS platforms: OSX 10.9 (Mavericks) and higher (64-bit)

The following network ports must be open for the TOE to function:

- 443/TCP (for HTTPS communications)
- 587/TCP and 993/TCP (for secure SMTP/IMAP communications)
- 636/TCP (for LDAPS communications)

The product itself also requires TCP port 9100 to be open for network printing, UDP ports 161/162 to be open for SNMP communications, and TCP ports 80/139/445 and UDP ports 137/138 for installation of printer drivers; however, these functions do not pertain to the storage and transmission of sensitive data so they are non-TSF.

Red Hat systems require CUPS 2.0 or higher to be installed on the workstation as a prerequisite to installing the PL Client application. Client systems should have at least 200 MB of storage space available. Systems used as Service Hosts should have an additional 30 MB of storage space. Any computer capable of running a compatible operating system will have sufficient RAM to run the TOE.

The TOE's operational environment includes the following:

- PrinterLogic Web Stack Server (Web Server) for centralized configuration of the PL Client application.
- Platforms (hardware and software) on which the TOE is hosted
- Full disk encryption is required for all platforms to ensure adequate data-at-rest protection

- Windows cryptographic libraries (Windows PL Client only), used to provide cryptographic functionality in some cases
- Web browser, used to access the environmental Web Server GUIs
- MySQL 5.7 database (installed on environmental Web Server), used to store configuration settings and security data
- Email server, used to hold messages that can be retrieved by the TOE for pull printing
- Google Cloud print server, used to hold messages that can be retrieved by the TOE for pull printing
- Active Directory, used for user authentication – in the evaluated configuration, it is assumed that all users belong to the organization's Active Directory domain; however, the TOE does require the use of at least one locally-defined administrator account to be used during initial setup of the TOE
- Mobile devices, used to initiate mobile print jobs—the following mobile operating systems are supported:
    - iOS 9+
    - Android 4.4+
    - Chrome OS (all versions)
- Printers, used to execute print jobs released by the user—supported manufacturers include HP, Xerox, Konica Minolta, and Ricoh. For the full list of compatible devices, refer to http://docs.printerlogic.com/Content/B_GettingStarted/RequirementsAndSupportedEnvironments.htm?Highlight=hardware.

2.4.2 Logical Boundary

This section summarizes the security functions provided by the TOE:

- Cryptographic Support
- User Data Protection
- Identification and authentication
- Security Management
- Privacy
- Protection of the TSF
- TOE access
- Trusted Path/Channels

2.4.2.1 Cryptographic Support

The TOE uses NIST-validated cryptographic algorithms to secure data in transit. The Windows PL Client application relies on the FIPS-validated cryptographic library cng.sys provided by Windows to perform cryptographic functionality, while the Linux and macOS PL Clients include their own copies of the OpenSSL FIPS Object Module. The Windows PL Client also uses its own instance of OpenSSL-FIPS when running as a Service Host.

The PL Client application (for all platforms) provides TLS/HTTPS client, TLS client, and TLS server functionality. All components rely on their underlying OS platforms to provide entropy used for key generation.

2.4.2.2 User Data Protection

The TOE leverages functionality provided by its underlying OS platforms to secure sensitive data at rest. The TOE uses network resources provided by the underlying platforms. The TOE also interfaces with the print spool of its underlying platform. All platform services are invoked with user awareness and authorization.

The TOE uses network connectivity to interact with the Web Server to receive configuration changes and to communicate the status of held print jobs. If configured as a Service host, it will also use network connectivity to securely retrieve print jobs from the operational environment (email server, iOS device, Android device, and/or Google Cloud print server), and to perform reverse AD lookups of users who submit these print jobs.

2.4.2.3 Identification and Authentication

The TOE uses X.509 certificates to authenticate endpoints for TLS and HTTPS trusted communications. As with cryptographic functionality in general, the Windows PL Client relies on the operational environment to provide this functionality in some cases while the Linux and macOS PL Clients implement it entirely within the TSF. For all platforms, revocation status is checked using CRLs but connections are authorized if the revocation status of an otherwise valid certificate cannot be confirmed.

2.4.2.4 Security Management

PL Client configuration data is stored remotely on the Web Server system. Configuration of the PL Client is performed via administration of the environmental Web Server.

2.4.2.5 Privacy

The TOE handles Active Directory credentials as well as print spool data, which could contain personally identifiable information (PII). However, any such handling of data is done with the user's explicit authorization. No transmission of PII occurs that is not in direct response to user activity.

2.4.2.6 Protection of the TSF

The TOE includes measures to integrate securely with their underlying OS platforms. The TOE does not perform explicit memory mapping, nor does it allocate any memory region with both write and execute permissions. Similarly, the TOE does not write user-modifiable data to directories that contain executable files. The TOE is compatible with its supported host OS platforms when those platforms are configured in a secure manner. For all platforms, the TOE is not written in languages that are susceptible to stack-based buffer overflow attacks.

For each supported platform, the TOE uses a well-defined set of platform APIs and third party libraries.

The TOE provides the ability for a user to check its version and to apply updates. Updates are delivered in a format that is appropriate for the TOE's platform (e.g., .deb files for Ubuntu Linux). Application of an update removes all executable code associated with the TOE; there is no way for the TOE to modify its own code. Updates to the TOE are digitally signed, and the signature is validated prior to installation.

2.4.2.7 Trusted Path/Channels

The TOE uses trusted paths and channels to secure data in transit. The following interfaces are provided by the TOE:

- PL Client:
  - TLS/HTTPS client for changes to configuration data and pull printing status from Web Server
- PL Client (Service Host only):
  - TLS client for retrieval of print jobs over email (IMAPS)
  - TLS server for retrieval of print jobs over AirPrint (IPPS)
  - TLS client for transmission of printer status data to Google Cloud (XMPP)
  - TLS/HTTPS client for retrieval of print jobs from Google Cloud
  - TLS client for Active Directory communications (LDAPS)

2.5 TOE Documentation

PrinterLogic provides the following product documentation in support of the installation and secure use of the TOE:

- PrinterLogic Web Stack Admin Guide, Version 18.3
- PrinterLogic Web Stack Common Criteria Evaluated Configuration Guide, Version 1.0

3. Security Problem Definition

This Security Target includes by reference the Security Problem Definition, composed of threats and assumptions, from the [App PP]. The Common Criteria also provides for organizational security policies to be part of a security problem definition, but no such policies are defined in the [App PP].

In general, the threat model of the [App PP] is designed to protect against the following:

- Disclosure of sensitive data at rest or in transit that the user has a reasonable expectation of security for
- Excessive or poorly-implemented interfaces with the underlying platform that allow an application to be used as an intrusion point to a system

This threat model is applicable to the TOE because the contents of a print job may contain sensitive data that a user expects will not be disclosed to anyone other than whoever takes possession of the physical printed document, and because the TOE runs on general purpose operating systems that may contain other data, applications, or network services that enforce their security in part through the assumption that the underlying operating system is trusted.

4. Security Objectives

Like the Security Problem Definition, this Security Target includes by reference the security objectives define in [App PP]. This includes security objectives for the TOE (used to mitigate threats) and for its operational environment (used to satisfy assumptions).

5. IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the following Protection Profiles (PP) and Functional Packages:

- *Protection Profile for Application Software*, version 1.3, 21 March 2019 [App PP]
- *Functional Package for Transport Layer Security (TLS),* Version 1.1, February 12, 2019 [TLS Package]

As a result, any selection/assignment/refinement operations already performed by that PP on the claimed SFRs are not identified here (i.e., they are not for matted in accordance with the conventions specified in section 1.3 of this Security Target). Formatting conventions are only applied on SFR text that was chosen at the ST author's discretion.

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the [App PP] and [TLS Package]. The PP and package defines the following extended SAR and SFRs; since they have not been redefined in this ST, the [App PP] and [TLS Package] should be consulted for more information regarding these extensions to CC Parts 2 and 3.

- ALC_TSU_EXT.1 (from [App PP]): Timely Security Updates
- FCS_CKM_EXT.1 (from [App PP]): Cryptographic Key Generation Services
- FCS_HTTPS_EXT.1 (from [App PP]): HTTPS Protocol
- FCS_RBG_EXT.1 (from [App PP]): Random Bit Generation Services
- FCS_RBG_EXT.2 (from [App PP]): Random Bit Generation from Application
- FCS_STO_EXT.1 (from [App PP]): Storage of Credentials
- FCS_TLS_EXT.1 (from [TLS Package]): TLS Protocol
- FCS_TLSC_EXT.1 (from [TLS Package]): TLS Client Protocol
- FCS_TLSC_EXT.5 (from [TLS Package]): TLS Client Support for Supported Groups Extension
- FDP_DAR_EXT.1 (from [App PP]): Encryption of Sensitive Application Data
- FDP_DEC_EXT.1 (from [App PP]): Access to Platform Resources
- FDP_NET_EXT.1 (from [App PP]): Network Communications
- FIA_X509_EXT.1 (from [App PP]): X.509 Certificate Validation
- FIA_X509_EXT.2 (from [App PP]): X.509 Certificate Authentication
- FMT_CFG_EXT.1 (from [App PP]): Secure by Default Configuration
- FMT_MEC_EXT.1 (from [App PP]): Supported Configuration Mechanism
- FPR_ANO_EXT.1 (from [App PP]): User Consent for Transmission of Personally Identifiable Information
- FPT_AEX_EXT.1 (from [App PP]): Anti-Exploitation Capabilities
- FPT_API_EXT.1 (from [App PP]): Use of Supported Services and APIs
- FPT_IDV_EXT.1 (from [App PP]): Software Identification and Versions
- FPT_LIB_EXT.1 (from [App PP]): Use of Third Party Libraries
- FPT_TUD_EXT.1 (from [App PP]): Integrity for Installation and Update
- FPT_TUD_EXT.2 (from [App PP]): Integrity for Installation and Update
- FTP_DIT_EXT.1 (from [App PP]): Protection of Data in Transit

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the TOE.

Table 1 TOE Security Functional Components

Requirement Class	Requirement Component
FCS: Cryptographic Support	FCS_CKM.1(1): Cryptographic Asymmetric Key Generation
	FCS_CKM.2: Cryptographic Key Establishment
	FCS_CKM_EXT.1: Cryptographic Key Generation Services
	FCS_COP.1(1): Cryptographic Operation – Encryption/Decryption
	FCS_COP.1(2): Cryptographic Operation – Hashing
	FCS_COP.1(3): Cryptographic Operation – Signing
	FCS_COP.1(4): Cryptographic Operation – Keyed-Hash Message Authentication
	FCS_HTTPS_EXT.1: HTTPS Protocol (Windows Client, Linux Client, macOS Client)
	FCS_RBG_EXT.1: Random Bit Generation Services
	FCS_RBG_EXT.2: Random Bit Generation from Application
	FCS_STO_EXT.1: Storage of Credentials
	FCS_TLS_EXT.1: TLS Protocol (Windows Client, Linux Client, macOS Client)
	FCS_TLSC_EXT.1: TLS Client Protocol (Windows Client, Linux Client, macOS Client)
	FCS_TLSC_EXT.5: TLS Client Support for Supported Groups Extension
FDP: User Data Protection	FDP_DAR_EXT.1: Encryption of Sensitive Application Data
	FDP_DEC_EXT.1: Access to Platform Resources
	FDP_NET_EXT.1: Network Communications
FIA: Identification and authentication	FIA_X509_EXT.1: X.509 Certificate Validation
	FIA_X509_EXT.2: X.509 Certificate Authentication
FMT: Security Management	FMT_CFG_EXT.1: Secure by Default Configuration
	FMT_MEC_EXT.1: Supported Configuration Mechanism
	FMT_SMF.1: Specification of Management Functions
FPR: Privacy	FPR_ANO_EXT.1: User Consent for Transmission of Personally Identifiable Information
FPT: Protection of the TSF	FPT_AEX_EXT.1: Anti-Exploitation Capabilities
	FPT_API_EXT.1: Use of Supported Services and APIs
	FPT_IDV_EXT.1: Software Identification and Versions
	FPT_LIB_EXT.1: Use of Third Party Libraries
	FPT_TUD_EXT.1: Integrity for Installation and Update

Requirement Class	Requirement Component
	FPT_TUD_EXT.2: Integrity for Installation and Update
FTP: Trusted Path/Channels	FTP_DIT_EXT.1: Protection of Data in Transit

5.2.1 Cryptographic Support (FCS)

FCS_CKM.1(1)	**Cryptographic Asymmetric Key Generation**
FCS_CKM.1.1(1)	The application shall [invoke platform-provided functionality, implement functionality] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [[ECC schemes] using ["NIST curves" P-256, P-384 and [no other curves]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4]].
Application Note:	*The capabilities that rely on platform-provided functionality versus implementing it within the TOE boundary are identified in Table 3.*

FCS_CKM.2	**Cryptographic Key Establishment**
FCS_CKM.2.1	The application shall [invoke platform-provided functionality, implement functionality] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [[Elliptic curve-based key establishment schemes] that meet the following: [NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"]].
Application Note:	*The capabilities that rely on platform-provided functionality versus implementing it within the TOE boundary are identified in Table 3.*

FCS_CKM_EXT.1	**Cryptographic Key Generation Services**
FCS_CKM_EXT.1.1	The application shall [invoke platform-provided functionality for asymmetric key generation, implement asymmetric key generation].
Application Note:	*The capabilities that rely on platform-provided functionality versus implementing it within the TOE boundary are identified in Table 3.*

FCS_COP.1(1)	**Cryptographic Operation – Encryption/Decryption**
FCS_COP.1.1(1)	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].

FCS_COP.1(2)	**Cryptographic Operation – Hashing**
FCS_COP.1.1(2)	The application shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [SHA-256, SHA-384] and message digest sizes [256, 384] bits that meet the following: FIPS Pub 180-4.

FCS_COP.1(3)	**Cryptographic Operation – Signing**

FCS_COP.1.1(3)	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 186-4, "Digital Signature Standard (DSS)", Section 4, ECDSA schemes using "NIST curves" P-256, P-384 and [no other curves] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5].
FCS_COP.1(4)	**Cryptographic Operation – Keyed-Hash Message Authentication**
FCS_COP.1.1(4)	The application shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-SHA-256 and [SHA-384] with key sizes [*256 bits, 384 bits*] and message digest sizes 256 and [384] bits that meet the following: FIPS PUB 198-1 The Keyed-Hash Message Authentication Code and FIPS PUB 180-4 Secure Hash Standard.
FCS_HTTPS_EXT.1	**HTTPS Protocol**
FCS_HTTPS_EXT.1.1	The application shall implement the HTTPS protocol that complies with RFC 2818.
FCS_HTTPS_EXT.1.2	The application shall implement HTTPS using TLS as defined in the TLS package.
FCS_HTTPS_EXT.1.3	The application shall [not establish the application-initiated connection] if the peer certificate is deemed invalid.
FCS_RBG_EXT.1	**Random Bit Generation Services**
FCS_RBG_EXT.1.1	The application shall [invoke platform-provided DRBG functionality, implement DRBG functionality] for its cryptographic functions.
Application Note:	*The capabilities that rely on platform-provided functionality versus implementing it within the TOE boundary are identified in Table 3.*
FCS_RBG_EXT.2	**Random Bit Generation from Application**
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 [Hash_DRBG (any)].
FCS_RBG_EXT.2.2	The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [no other 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.
FCS_STO_EXT.1	**Storage of Credentials**
FCS_STO_EXT.1.1	The application shall [invoke the functionality provided by the platform to securely store [*X.509 certificates*]] to non-volatile memory.
Application Note:	*Each platform version of the TOE relies on the platform's typical security mechanisms for X.509 certificate storage. The remaining credentials, maintained only by the environmental Web Server, are stored by the platform through invocation of platform-provided AES (i.e., the same mechanism that the TSF would use if it was the component responsible for the secure storage).*
FCS_TLS_EXT.1	**TLS Protocol**
FCS_TLS_EXT.1.1	The product shall implement [TLS as a client].
FCS_TLSC_EXT.1	**TLS Client Protocol**
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_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289]

and also supports functionality for [none].

Application Note: *The capabilities that rely on platform-provided functionality versus implementing it within the TOE boundary are identified in Table 3.*

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].

FCS_TLSC_EXT.5	**TLS Client Support for Supported Groups Extension**

FCS_TLSC_EXT.5.1 The product shall present the Supported Groups Extension in the Client Hello with the supported groups [secp256r1, secp384r1].

5.2.2 User Data Protection (FDP)

FDP_DAR_EXT.1	**Encryption of Sensitive Application Data**

FDP_DAR_EXT.1.1 The application shall [[leverage platform-provided functionality to encrypt sensitive data]] in non-volatile memory.

FDP_DEC_EXT.1	**Access to Platform Resources**

FDP_DEC_EXT.1.1 The application shall restrict its access to [network connectivity, [*network-connected printers*]].

FDP_DEC_EXT.1.2 The application shall restrict its access to [[*print spool directory/folder*]].

FDP_NET_EXT.1	**Network Communications**

FDP_NET_EXT.1.1 The application shall restrict network communication to [respond to [*remotely-initiated cloud/email print requests, remote Web Server configuration, status communication with Web Server*], [*application-initiated LDAP communications, application-initiated status communication with Web Server*]].

5.2.3 Identification and Authentication (FIA)

FIA_X509_EXT.1	**X.509 Certificate Validation**

FIA_X509_EXT.1.1 The application shall [invoke platform-provided functionality, 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.
- The application shall validate the revocation status of the certificate using [a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3].
- The application shall validate the extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage field.

Application Note: *The capabilities that rely on platform-provided functionality versus implementing it within the TOE boundary are identified in section 6.4.*

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.

FIA_X509_EXT.2	**X.509 Certificate Authentication**

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 [accept the certificate].

5.2.4 Security Management (FMT)

FMT_CFG_EXT.1	**Secure by Default Configuration**

FMT_CFG_EXT.1.1 The application shall only provide 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's binaries and data files from modification by normal unprivileged users.

FMT_MEC_EXT.1	**Supported Configuration Mechanism**

FMT_MEC_EXT.1.1[1] The application shall [invoke the mechanisms recommended by the platform vendor for storing and setting configuration options].

FMT_SMF.1	**Specification of Management Functions**

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions [no management functions].

Application Note: *The TOE does not contain its own management interface. All configuration of the TOE is performed by the Web Server in the TOE's operational environment.*

5.2.5 Privacy (FPR)

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

FPR_ANO_EXT.1.1 The application shall [not transmit PII over a network].

Application Note: *Note that as per the [App PP], this requirement applies only to PII that is specifically requested by the application and does not apply if the user volunteers PII without prompting. This is why the chosen selection was made even though the application can*

[1] This SFR has been modified as per NIAP TD0437

facilitate the user-initiated transfer of arbitrary data to a networked printer (which can potentially include PII).

5.2.6 Protection of the TSF (FPT)

FPT_AEX_EXT.1	**Anti-Exploitation Capabilities**
FPT_AEX_EXT.1.1	The application shall not request to map memory at an explicit address except for [*no exceptions*].
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 compiled with stack-based buffer overflow protection enabled.
FPT_API_EXT.1	**Use of Supported Services and APIs**
FPT_API_EXT.1.1	The application shall only use documented platform APIs.
FPT_IDV_EXT.1	**Software Identification and Versions**
FPT_IDV_EXT.1.1	The application shall be versioned with [[*other version information*]].
FPT_LIB_EXT.1	**Use of Third Party Libraries**
FPT_LIB_EXT.1.1	The application shall be packaged with only [*third-party libraries listed in Appendix A.2*].
Application Note:	*The TOE uses a large number of third-party libraries so this information has been provided in an Appendix for readability purposes.*
FPT_TUD_EXT.1	**Integrity for Installation and Update**
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	The application installation package and its updates shall be digitally signed such that its 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].
FPT_TUD_EXT.2	**Integrity for Installation and Update**
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.

5.2.7 Trusted Path/Channels (FTP)

FTP_DIT_EXT.1	**Protection of Data in Transit**
FTP_DIT_EXT.1.1	The application shall [

- encrypt all transmitted sensitive data with [*HTTPS in accordance with FCS_HTTPS_EXT.1, TLS as defined in the TLS Package*],
- invoke platform-provided functionality to encrypt all transmitted sensitive data with [*HTTPS, TLS*]

] between itself and another trusted IT product.

Application Note: *The TOE uses platform-provided cryptography for the TLS/HTTPS Client for the Windows PL Client, but ONLY when it is not configured as a Service Host. All other HTTPS/TLS communication is provided by the TSF (see Table 3 Cryptographic Functions).*

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference to [App PP].

Table 2 Assurance Components

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1 Basic Functional Specification
AGD: Guidance Documents	AGD_OPE.1: Operational User Guidance
	AGD_PRE.1: Preparative Procedures
ALC: Life-Cycle Support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM coverage
	ALC_TSU_EXT.1: Timely Security Updates
ATE: Tests	ATE_IND.1 Independent Testing – Conformance
AVA: Vulnerability Assessment	AVA_VAN.1 Vulnerability Survey

Consequently, the assurance activities specified in the [App PP] apply to the TOE evaluation, including any changes made to them by subsequent NIAP Technical Decisions as summarized in section 1.2 above.

6. TOE Summary Specification

This chapter describes the security functions of the TOE:

- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Privacy
- Protection of the TSF
- Trusted Path/Channels

It also describes the process put in place by the TOE vendor to provide timely security updates to the TOE as per the ALC_TSU_EXT.1 requirements of the [App PP].

6.1 Timely Security Updates

PrinterLogic provides maintenance releases as needed in between major releases. The purpose of the maintenance release is to provide bug fixes and security updates for all platform versions of the PrinterLogic Web Stack Client. Additionally, when updates are made to the bundled third-party capabilities, they are obtained by PrinterLogic and pushed to customers. Customers are notified by the Customer Support team when a maintenance release is made available. Maintenance release notes identify the security vulnerabilities that are fixed in the release. The only mechanism to deploy security updates is through maintenance releases. Upon discovery of a vulnerability, the impact will be assessed for priority. Any critical security fixes are immediately implemented, with a target release of 72 hours from discovery. Lower-risk items are targeted for resolution in 30-45 days depending on priority and severity. All security reports are communicated from customers to Customer Support via live phone support or through an HTTPS form on the printerlogic.com website.

6.2 Cryptographic Support

TOE components use cryptography to secure data in transit to and from each application instance. The following cryptographic interfaces are used by each component when the TOE is configured to be in its evaluated configuration:

- PL Client – User (Windows, Linux, and macOS):
  - TLS/HTTPS client (for communication with environmental Web Server to receive updated configuration settings)
- PL Client – Service Host (Windows, Linux, and macOS):
  - TLS/HTTPS client (for communication with Web Server to receive updated configuration settings/LDAP credential data)
  - TLS/HTTPS client (for retrieval of print spool data from AirPrint-compatible devices over IPPS)
  - TLS/HTTPS client (for retrieval of print spool data from Google Cloud)
  - TLS client (for reverse Active Directory lookup to establish user identity of email print requesters)
  - TLS client (for communication to email server using IMAP tunneled through TLS)

The Windows PL Client relies on the underlying OS platform cryptography (via the FIPS-validated cryptographic module cng.sys) to implement the cryptographic primitives for TLS/HTTPS communications when it is not configured as a Service Host. In this configuration, it also rlies on WinHTTP to provide the TLS/HTTPS protocol stack. The Windows PL Client also uses OpenSSL 1.0.2h wrapped with version 2.0.12 of the OpenSSL FIPS Object Module for its outbound LDAP communications as well as inbound remote print services (AirPrint, Google Cloud, email server) when configured as a Service Host. The Windows PL Client relies on the platform to provide TLS client communications to the Web Server for periodic status checks and transmission of configuration settings.

For the Linux and macOS PL Client components, all TLS/HTTPS client communications are implemented by the TSF. This is done using the Python Requests library, which uses OpenSSL 1.0.2h, wrapped with version 2.0.12 of the OpenSSL FIPS Object Module as the underlying library for cryptographic primitives.

The following table lists each component and cryptographic interface included in the TOE boundary. For each entry, if that component makes use of that interface, the table indicates whether the cryptographic functionality is implemented by the TSF, the TOE platform, or both.

Table 3 Cryptographic Functions

	Windows PL Client	Linux PL Client	macOS PL Client
HTTPS Client (administration)	Both*	TSF	TSF
TLS Client (LDAP)	TSF	TSF	TSF
TLS Client (cloud printing)	TSF	TSF	TSF
TLS Client (email printing)	TSF	TSF	TSF

*The Windows PL Client uses platform cryptography when not configured as a Service Host, and TSF cryptography when configured as a Service Host.

The cryptographic algorithms supplied by the TOE are NIST-validated. The following table identifies the cryptographic algorithms used by the TSF, the associated standards to which they conform, and the NIST certificates that demonstrate that the claimed conformance has been met.

Table 4 Cryptographic Functions

Functions	Standards	Certificates
FCS_CKM.1(1) Cryptographic Asymmetric Key Generation		
ECC key pair generation (NIST curves P-256, P-384)	FIPS PUB 186-4	CAVP cert #C351
FCS_CKM.2 Cryptographic Key Establishment		
ECDSA based key establishment	NIST SP 800-56A Revision 2	CAVP cert #C351
FCS_COP.1(1) Cryptographic Operation – Encryption/Decryption		
AES CBC, GCM (128, 256 bits)	CBC as defined in NIST SP 800-38A GCM as defined in NIST SP 800-38D	CAVP cert #C351
FCS_COP.1(2) Cryptographic Operation – Hashing		
SHA-256 and SHA-384 (digest sizes 256 and 384 bits)	FIPS PUB 180-4	CAVP cert #C351
FCS_COP.1(3) Cryptographic Operation – Signing		
RSA Digital Signature Algorithm (rDSA) (modulus 2048)	FIPS PUB 186-4, Section 4	CAVP cert #C351
ECDSA (NIST curves P-256 and P-384)	FIPS PUB 186-4, Section 5	CAVP cert #C351

Functions	Standards	Certificates
FCS_COP.1(4) Cryptographic Operation – Keyed Hash Message Authentication		
HMAC-SHA-256 and SHA-384	FIPS PUB 189-1	CAVP cert #C351
FCS_RBG_EXT.2 Random Bit Generation from Application		
Hash_DRBG	NIST SP 800-90A	CAVP cert #C351

In the evaluated configuration the TOE's Windows and Linux platform versions were running on computers with Intel Core I3-4010U CPUs. The macOS platform version was running on a compuer with an Intel Core I5 CPU. This is consistent with the algorithm certificate.

The TOE capabilities that supply their own cryptographic algorithm implementations generate asymmetric keys in support of trusted communications. The TSF generates ECC keys using P-256 and P-384. These keys are generated in support of the ECDHE key establishment schemes that are used for TLS/HTTPS communications. To ensure sufficient key strength, the same TOE capabilities implement DRBG functionality for key generation. These capabilities use the Hash_DRBG random bit generator. The proprietary Entropy Analysis Report (EAR) describes how the TSF extracts random data from software-based sources to ensure that an amount of entropy that is at least equal to the strength of the generated keys is present (i.e., at least 256 bits when the largest supported keys are generated) when seeding the DRBG for key generation purposes. Windows and macOS TOE platform versions rely on the third-party entropy sources provided by the respective platform vendors; in these cases, it is assumed that these platforms provide at least 256 bits of entropy. For those Windows PL Client functions that do not rely on TSF-internal cryptography, the TSF invokes the platform-provided cryptographic functionality to perform key establishment. Key generation is the only TOE function that requires the use of random numbers. Random numbers are obtained from the following platform APIs, depending on the platform used:

- Windows: BCryptGenRandom
- Linux and macOS: invocation of /dev/random pseudo-device

Per Table 3, the BCryptGenRandom function will be invoked by either the TOE or by the platform, depending on the TOE configuration.

Each TOE platform version uses TLS 1.2 for client communications. As stated previously, the Windows PL Client uses both TSF and platform cryptography to perform TLS communications while the Linux and macOS PL Clients implement TLS 1.2 themselves exclusively. For all TOE platform versions, the TLS client implementation supports the following TLS cipher suites in the TOE's evaluated configuration:

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

Note that for the TLS interfaces that rely on the TOE platform, configuration of the platform will be required to enforce this set of supported ciphersuites.

All supported ciphersuites use elliptic curves as the method of key establishment. In all cases, the TSF presents secp256r1 and secp384r1 as the supported values in the Supported Elliptic Curves (Supported Groups) extension.

As part of certificate validation in the establishment of TLS connectivity, each TOE platform version will validate the reference identifier of a presented server certificate. This is done through validation of the Common Name (CN) and Subject Alternative Name (SAN) certificate fields. IP addresses are supported. Wildcards are only supported for the left-most label immediately preceding the public suffix. Certificate pinning is not supported.

In the event of a certificate validation failure, the connection attempt will be dropped and the user will be notified via generation of an audit record (the user notification is not presented in real-time by the TOE). This is applicable to all TLS and TLS/HTTPS communications.

The TOE uses platform storage for X.509 certificates that are used to establish TLS communications. Windows, Linux, and macOS Client platform versions use the Windows Certificate Store, Linux keyrings, and Keychain respectively.

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

- FCS_CKM.1(1) – The TOE and the TOE platform generate ECC keys for the purpose of TLS key establishment.
- FCS_CKM.2 – The TOE and the TOE platform perform ECC key establishment for TLS.
- FCS_CKM_EXT.1 – Depending on the platform version and interface, the TSF either implements key generation functionality itself or relies on the underlying OS platform to provide it.
- FCS_COP.1(1) – TOE capabilities that implement their own cryptographic functionality use AES in support of TLS communications.
- FCS_COP.1(2) – TOE capabilities that implement their own cryptographic functionality use various hash algorithms in support of TLS communications.
- FCS_COP.1(3) – TOE capabilities that implement their own cryptographic functionality use RSA and ECDSA signature algorithms in support of TLS communications.
- FCS_COP.1(4) – TOE capabilities that implement their own cryptographic functionality use various HMAC algorithms in support of TLS communications.
- FCS_HTTPS_EXT.1 – All of the PL Client platform versions implement HTTPS functionality for remote access to the environmental Web Server and, when configured as a Service Host, retrieval of print spool data.
- FCS_RBG_EXT.1 – Depending on the platform version and interface, the TSF either implements random bit generation functionality itself or relies on the underlying OS platform to provide it.
- FCS_RBG_EXT.2 – TOE capabilities that implement their own cryptographic functionality use a NIST SP 800-90A compliant DRBG algorithm in support of TLS communications.
- FCS_STO_EXT.1 – The TOE uses platform-provided services to store all credential data.
- FCS_TLS_EXT.1 – The TOE implements TLS as a client.
- FCS_TLSC_EXT.1 – The TOE acts as a TLS client to provide various functions. Depending on the capability, the TLS client functionality may be implemented by either the TSF or by the platform.
- FCS_TLSC_EXT.5 - When the TOE attempts to establish TLS communications as a client, it will present the specific elliptic curves it supports for ECDHE.

6.3 User Data Protection

The [App PP] defines ‘sensitive data’ as follows: “Sensitive data may include all user or enterprise data or may be specific application data such as emails, messaging, documents, calendar items, and contacts. Sensitive data must minimally include PII, credentials, and keys. Sensitive data shall be identified in the application’s TSS by the ST author.”

The table below lists the data that is considered to be ‘sensitive data’ for this TOE along with where that data resides. The TSF does not examine printed documents for content so they are all considered to be sensitive data since a user has a reasonable expectation that if they print a document, it will not be stored on a separate server for others to view.

Table 5 Sensitive Data

Sensitive Data	Stored On	Exchange	Protection At Rest	Protection In Transit
Instructions to hold/release print jobs	N/A	Issued from Web Server to Service Host at user direction	N/A	HTTPS

Sensitive Data	Stored On	Exchange	Protection At Rest	Protection In Transit
Documents for print spooling	TOE (as Service Host)	Retrieval from mail server, mobile device, Google cloud (depending on print method)	Platform encryption (full disk)	Email printing: IMAPS (TLS) Cloud printing (all): HTTPS Cloud printing (Google cloud only): XMPP (TLS) Cloud printing (AirPrint only): IPPS (TLS)
X.509 certificates for TLS	TOE	Public key portion only	Platform storage (FCS_STO_EXT.1)	N/A

In the evaluated configuration, each TOE platform version will be installed on platforms that have full disk encryption enabled. All data at rest is ultimately secured by the operational environment's platform encryption functionality.

The underlying platform functionality that the TOE interacts with is the system's network connectivity and print spool. Through access to the print spool, the TOE will also interact with printers themselves. Network usage of the TOE is authorized implicitly through user guidance; it does not make any specific requests on its own to use network services once installed. The TOE restricts network connectivity to the following uses only:

- User-initiated: none
- Remotely-initiated: initiation of cloud/email printing that a Service Host is configured to handle, remote Web Server configuration, status communication (i.e. receiving notification from the environmental Web Server that a held pull print job has been released)
- TSF-initiated: LDAP communications, status communication (i.e. notifying the environmental Web Server that a new pull print job is being held)

The TOE only interacts with the print spool/environmental printers when users explicitly initiate this process with a reasonable expectation that the print spool will be used. This is done one of two ways:

- Direct email printing, where the user sends a document to an email box monitored by a Service Host that is automatically configured to print any emails it receives from that box to a specific printer.
- Pull printing, where the user initiates a print job that is held by the Service Host until being released to a specific printer by direct user action either on the Release Portal or on the target printer itself.

The User Data Protection security function is designed to satisfy the following security functional requirements:

- FDP_DAR_EXT.1 – Sensitive data at rest is protected by full disk encryption of the underlying OS platforms for each TOE component.
- FDP_DEC_EXT.1 – The TOE's use of platform services is well understood by users prior to authorizing the TOE activity.
- FDP_NET_EXT.1 – The TOE communicates over the network for well-defined purposes. Depending on the function, the use of network resources is remotely initiated by a user performing an action in the operational environment or initiated by the TOE itself.

6.4 Identification and Authentication

The TOE uses X.509 certificates for authentication of trusted communications for Web Server validation, LDAP server validation (Service Host only), mail server validation (Service Host only), and cloud print service validation (Service Host only)

Depending on the capability, certificate validation is performed either by the TSF or by the TOE platform, as follows:

- Windows PL Client
  - TSF: LDAP client interface, cloud print interface (Service Host only)
  - TOE platform: HTTPS client interface (when not configured as Service Host)
- Linux PL Client
  - TSF: LDAP client interface, cloud print interface (Service Host only)
  - TOE platform: none
- macOS PL Client
  - TSF: LDAP client interface, cloud print interface (Service Host only)
  - TOE platform: none

The following functional behavior is identical for all cases where the TSF is responsible for certificate validation:

- Certificate validation and certificate path validation is performed in accordance with RFC 5280.
- The certificate path is checked to ensure that it terminates with a trusted CA certificate.
- The certificate path is validated by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- Revocation status is checked using CRLs in accordance with RFC 5280 Section 6.3.
- The application shall validate the extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage field.

In the event that the revocation status of a certificate cannot be verified (i.e. the CRL cannot be acquired), each TOE component will accept the certificate.

Because the TOE's use of the certificate validation function is to validate the authenticity of servers, the TSF chooses what certificates to use based on what is presented to it as part of establishing the TLS session.

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

- FIA_X509_EXT.1 – X.509 certificates are validated by the TSF when establishing trusted communications. This does not apply to the Web Server because that component relies on the TOE platform to implement TLS functionality which includes the dependent certificate validation function.
- FIA_X509_EXT.2 – When revocation status of a certificate cannot be determined, the TSF accepts the certificate by default. This does not apply to the Web Server because that component relies on the TOE platform to implement TLS functionality which includes the dependent certificate validation function.

6.5 Security Management

All TOE components are protected from unauthorized access via the host platform's file system. By default, the following file permissions are used for the binaries and application data of each of the different TOE platform versions:

- Windows PL Client: Owned by Administrators
- Linux PL Client: 644; owned by user root, group root
- macOS PL Client: 644; owned by user root, group wheel

The TOE enforces its security functionality by default upon initial installation and configuration. PL Client configuration settings are defined remotely, maintained by the environmental Web Server, and transmitted to the TOE over TLS/HTTPS. Read-only copies of these settings reside locally on the PL Client's local OS platform as backups in the event of network outages.

The TOE will hold pull print jobs in protected files on the host platform. In addition to being protected using file permissions, this data is protected by the platform encryption expected by FDP_DAR_EXT.1. When a user initiates a pull print, the print job will be stored on the user's local system and maintained by the TOE there. When a user initiates a cloud or email print, the print job will be stored on a system where the TOE is configured as a Service Host. The TOE provides a mechanism for a user to complete a pull print or cloud print through release of the print job. The release method differs based on the print method, as follows:

- Pull print: PL Client (on user's host system) prompts user to hold or release print job
  - If print job is held, user will use the environmental Web Server to release the job at a later time
  - If print job is released, PL Client will release the job
- Cloud print:
  - User can use the environmental Web Server to release the job

When a held print job is released, the TOE will use the underlying host platform's network printing functionality to perform the actual print operation.

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

- FMT_CFG_EXT.1 – The TOE is prevented from direct modification by untrusted users via the host OS platform.
- FMT_MEC_EXT.1 – Locally-modifiable configuration settings for each TOE platform version are stored in appropriate locations for each host OS platform.
- FMT_SMF.1 – The TOE does not possess an interface to directly manage its functionality.

6.6 Privacy

The TOE handles several types of personally identifiable information (PII). A user can perform an email print or cloud print operation that transmits print spool data (which may or may not contain PII) to the TOE over the network. However, in no circumstance does the TSF transmit PII at the application's request; any transmission of PII is performed as a direct result of user behavior that is reasonably understood to involve transmission of that data over a network.

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

- FPR_ANO_EXT.1 – the TOE prevents the unnoticed/unauthorized transmission of PII across a network by ensuring that any such transmission is the result of explicit user action.

6.7 Protection of the TSF

The TOE implements several mechanisms to protect against exploitation. All TOE platform versions implement address space layout randomization (ASLR) and rely fully on their underlying host platforms to perform memory mapping. There is no situation where the TSF maps memory to an explicit address. The Linux/macOS PL Clients (Python) are interpreted code and are not just-in-time compiled. These two platform versions also do not use both PROT_WRITE and PROT_EXEC on the same memory regions. The Windows PL Client is a Delphi application; Delphi intrinsically does not provide the ability to allocate memory that is both writable and executable. When configured as a Service Host, the Windows PL Client will also use Python components with the same characteristics as those described for the Linux and macOS platform versions. The various TOE platform versions write data to the underlying OS platforms; however, no data is considered to be user-modifiable. The following directories are used to write and store data:

- Windows PL Client: executable code installed to Program Files; cached configuration data resides in Program Files; log files are written to the %TMP% directory for the SYSTEM account; print spool data is written to the OS print spool directory.
- Linux PL Client: everything is installed to /opt/PrinterInstallerClient and various subfolders
- macOS PL Client: everything is installed to /opt/PrinterInstallerClient and various subfolders

The Windows PL Client is written in Object Pascal using the Delphi IDE. Delphi does not include a compiler flag for stack protection but the Object Pascal language itself mitigates this by using the heap instead of the stack. The heap is then protected using the memory protections described above. As a result, the Windows PL Client is not susceptible to stack-based buffer overflows. The remaining TOE platform versions, including the Service Host capability of the Windows PL Client, are written in interpreted languages (PHP, Python) that rely on the runtime environment to dynamically allocate memory and are therefore also not subject to stack-based buffer overflows.

All TOE platform versions are designed to run on host OS platforms where platform security features have been enabled (e.g. Windows Defender Export Guard, SELinux). TOE platform versions use only documented platform APIs. Appendix A.1 lists the APIs used by each platform version. The various TOE components also make use of third-party libraries. Appendix A.2 lists the libraries used by each platform version. The TOE is versioned with major and minor version numbers as well as build numbers. As an example, a hypothetical version 10.1 build 150 would be displayed as 10.1.150.

All TOE platform versions provide the means to check for, apply, and verify software updates. This is implemented for each TOE platform version as follows:

- Windows PL Client: in the evaluated configuration, the application is automatically configured to check for updates every time it communicates with the Web Server (no user interaction required). The PL Client can then receive software updates from the Web Server. Updates are packaged as .msi files. If the application is removed from the system, the application itself is removed but no cached configuration data is removed. Held print jobs are preserved but can only be released if the PL Client is reinstalled. The current version of the application can be queried by reviewing the log file. All updates are digitally signed by PrinterLogic using 2048-bit RSA signatures. The digital signature is verified automatically by Windows APIs prior to installation.
- Linux PL Client: in the evaluated configuration, the application is automatically configured to check for updates every time it communicates with the Web Server (no user interaction required). The PL Client can receive software updates from the Web Server. Updates are packaged as .deb files and installed using dpkg (for Ubuntu) and .rpm files, installed using yum (for Red Hat). If the application is removed from the system, all residual data is removed. The current version of the application can be queried by reviewing the log file or by using the functionality provided by the package manager used to install it. All updates are digitally signed by PrinterLogic using 2048-bit RSA signatures. The digital signature is verified either manually (for Ubuntu systems) or via the package manager (for Red Hat) prior to installation.
- macOS PL Client: in the evaluated configuration, the application is automatically configured to check for updates every time it communicates with the Web Server (no user interaction required. The PL Client can receive software updates from the Web Server. Updates are packaged as .pkg files. If the application is removed from the system, all residual data is removed. All updates are digitally signed by PrinterLogic using 2048-bit RSA signatures, issued by Apple. The digital signature is verified by the OS prior to execution but can also be verified manually using the package utility.

The TOE is made available as stand-alone installers that can be obtained from PrinterLogic's website, and can be distribed by any variety of methods (e.g. pushed out through AD or made available in the Software Center). For all platform versions, updating the TOE software is the only method of changing its executable code; it does not change its own code. In all cases, removal of the application will result in the deletion of all traces of the application except for any related configuration settings, log events, or output files.

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

- FPT_AEX_EXT.1 – Each TOE platform version interacts with its host OS platform in a manner that does not expose the system to memory-related exploitation.
- FPT_API_EXT.1 – Each TOE platform version uses documented platform APIs.
- FPT_IDV_EXT. 1 – Each TOE platform version is versioned with the year and month of release.
- FPT_LIB_EXT.1 – The set of third-party libraries used by each TOE platform version is well-defined.
- FPT_TUD_EXT.1 – The TOE can be updated through installation packages. Updates are signed by the vendor and validated by the host OS platform prior to installation.

- FPT_TUD_EXT.2 – Updates to the TOE are packaged using formats native to the supported OS platforms and removal of the TOE does not preserve any executable code on the platform.

6.8 Trusted Path/Channels

The TOE uses HTTPS / TLSv1.2 to secure sensitive data in transit over trusted channels. The channels supported by each TOE platform and the protocols used to establish them are listed in section 6.2. Not all trusted channel communications are provided by the TOE; specifically, the Windows platform invokes Microsoft IIS when acting as a web server and WinHTTP when acting as a client. All other trusted communications for all versions of the TOE will use TSF capabilities.

The following data is considered by the TOE to be 'sensitive' and is therefore protected in transit to/from the system on which the TOE resides:

- Configuration information (between Web Server and PL Client)
- Credential data for environmental components (CPA, LDAP) (PL Client to/from environmental components)
- Authorizations to hold/release print jobs (between Web Server and PL Client)
- Print spool data (from email server or iOS/Android device to PL Client acting as a Service Host)

The secure protocols are supported by NIST-validated cryptographic mechanisms included in the TOE implementation and provided by the operational environment. Refer to Table 3 for TOE provided vs platform provided HTTPS/TLS per platform version and section 6.2 for the specific NIST validation information. In each of these cases, the administrator must configure the interfaces to use the trusted channels before the TOE has been placed into its evaluated configuration.

The TOE can also interact with printers in the operational environment to query status information using SNMPv3, but this is not considered to be sensitive data as per section 6.2 and is therefore not protected with any of the trusted protocols specified in FTP_DIT_EXT.1. Actual communication of the print spool data from a user's host system or a system running a PL Client as a Service Host is handled by the underlying OS platform; the print job itself is no longer under the TOE's control once it has been released.

The Trusted Path/Channels security function is designed to satisfy the following security functional requirements:

- FTP_DIT_EXT.1 – The TOE secures sensitive data in transit using TLS and HTTPS.

7. Protection Profile Claims

This ST is conformant to the *Protection Profile for Application Software, Version 1.3, 1 March 2019* [App PP] and *Functional Package for Transport Layer Security (TLS), Version 1.1, February 12, 2019* [TLS Package]. along with all applicable errata and interpretations from the certificate issuing scheme.

As explained in section 3, Security Problem Definition, the Security Problem Definition of [App PP] has been included by reference into this ST.

As explained in section 4, Security Objectives, the Security Objectives of [App PP] has been included by reference into this ST. The [TLS Package] is a functional package and does not describe a Security Problem Definition or Security Objectives.

All claimed SFRs are defined in [App PP]. All mandatory SFRs are claimed. No optional or objective SFRs are claimed. Selection-based SFR claims are consistent with the selections made in the mandatory SFRs that prompt their inclusion. The inclusion of [TLS Package] is prompted by the TOE's implementation of the TLS protocol.

8. Rationale

This Security Target includes by reference the [App PP] Security Problem Definition, Security Objectives, and Security Assurance Requirements. The Security Target does not add, remove, or modify any of these items. Security Functional Requirements have been reproduced with the Protection Profile operations completed. All selections, assignments, and refinements made on the claimed Security Functional Requirements have been performed in a manner that is consistent with what is permitted by the [App PP] and [TLS Package]. The proper set of selection-based requirements have been claimed based on the selections made in the mandatory requirements. Consequently, the claims made by this Security Target are sufficient to address the TOE's security problem. Rationale for the sufficiency of the TOE Summary Specification is provided below.

8.1 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. The table below demonstrates the relationship between security requirements and security functions.

Table 6 Security Functions vs. Requirements Mapping

	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Privacy	Protection of the TSF	Trusted Path/Channels
FCS_CKM.1(1)	X						
FCS_CKM.2	X						
FCS_CKM_EXT.1	X						
FCS_COP.1(1)	X						
FCS_COP.1(2)	X						
FCS_COP.1(3)	X						
FCS_COP.1(4)	X						
FCS_HTTPS_EXT.1	X						
FCS_RBG_EXT.1	X						
FCS_RBG_EXT.2	X						
FCS_STO_EXT.1	X						
FCS_TLS_EXT.1	X						
FCS_TLSC_EXT.1	X						
FCS_TLSC_EXT.5	X						
FDP_DAR_EXT.1		X					
FDP_DEC_EXT.1		X					
FDP_NET_EXT.1		X					
FIA_X509_EXT.1			X				
FIA_X509_EXT.2			X				
FMT_CFG_EXT.1				X			
FMT_MEC_EXT.1				X			
FMT_SMF.1				X			
FPR_ANO_EXT.1					X		
FPT_AEX_EXT.1						X	
FPT_API_EXT.1						X	
FPT_IDV_EXT.1						X	
FPT_LIB_EXT.1						X	
FPT_TUD_EXT.1						X	
FPT_TUD_EXT.2						X	

	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Privacy	Protection of the TSF	Trusted Path/Channels
FTP_DIT_EXT.1							X

Appendix A: TOE Usage of Third-Party Components

This Appendix lists the platform APIs and third-party libraries that are used by the various TOE platform versions:

A.1 Platform APIs

Listed below are the platform APIs used by each platform version. The TOE does not use different platform APIs if it is configured as a Service Host.

A.1.1 Windows PL Client

- COM controls
  - Microsoft XMLDOM (COM control)
  - NameTranslate
- APIs in the following platform libraries:
  - activeds.dll
  - advapi32.dll
  - bcrypt.dll
  - comctl32.dll
  - comdig32.dll
  - cryp32.dll
  - fbwflib.dll
  - gdi32.dll
  - iphlpapi.dll
  - kernel32.dll
  - mpr.dll
  - msvcrt.dll
  - netapi32.dll
  - ole32.dll
  - oleaut32.dll
  - shell32.dll
  - user32.dll
  - userenv.dll
  - version.dll
  - winhttp.dll
  - wintrust.dll
  - wsock32.dll
  - wtsapi32.dll
- APIs in the following platform drivers:
  - winspool.drv

A.1.2 Linux PL Client

The only API used by the Linux PL Client is the Common UNIX Printing System (CUPS) to interface with the print spooler.

A.1.3 macOS PL Client

The only API used by the macOS PL Client is the Common UNIX Printing System (CUPS) to interface with the print spooler.

A.2 Third-Party Libraries

The following section lists the third-party libraries used by each TOE platform version. Libraries that only apply when the TOE is configured as a Service Host are listed separately from those that always apply.

A.2.1 Windows PL Client

The Windows PL Client uses the following libraries regardless of configuration:

- calwin32.dll
- netwin32.dll
- locwin32.dll
- clxwin32.dll
- nwcalls.dll
- nwnet.dll
- nwlocale.dll

In addition, it uses the following third-party libraries when configured as a Service Host only:

- pysnmp 4.4.4
- python-dateutil 2.7.3
- python-magic 0.4.15
- pytz 2018.5
- requests 2.19.1
- requests-toolbelt 0.8.0
- unittest-xml-reporting 2.2.0
- zeep 3.0.0
- zeroconf 0.19.0

A.2.2 Linux PL Client

The Linux PL Client uses the following third-party libraries:

- For all usage:
    - altgraph 0.15
    - appdirs 1.4.3
    - appscript 1.0.1
    - asn1crypto 0.24.0

  - certifi 2018.4.16
  - cffi 1.11.5
  - chardet 3.0.4
  - cryptography 2.3
  - cx-Freeze 5.0.2
  - decorator 4.3.0
  - dnspython3 1.12.0
  - gssapi 1.2.0
  - idna 2.6
  - ldap3 2.4
  - macholib 1.7
  - netifaces 0.10.4
  - packaging 17.1
  - psutil 2.1.1
  - pyaes 1.6.0
  - pyasn1 0.4.3
  - pycparser 2.18
  - pycups 1.9.72
  - pyOpenSSL 18.0.0
  - pyparsing 2.2.0
  - requests 2.18.4
  - six 1.11.0
  - urllib3 1.22
  - xmlrunner 1.7.7
  - yappi 0.98
- When configured as Service Host only: same as the Service Host libraries specified in A.2.1 above.

A.2.3 macOS PL Client

The macOS PL Client uses the same libraries as the Linux PL Client, both for standard usage and when configured as a Service Host.