PrinterLogic Web Stack Server 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

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 Server, version 18.3.

PrinterLogic Web Stack Server (PL Server) is a web server application that interacts with PrinterLogic Web Stack Clients (PL Clients) in its operational environment. These clients are installed on endpoint systems in an organization to facilitate direct IP printing. The PL Server is used to administer PL Clients, specifically in regards to configuration, user self-service, 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).

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]. The only capabilities covered by the evaluation are those specified in the aforementioned Protection Profile; no additional security functional claims are made by the product. The security functionality specified in [App PP] 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], 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 Server version 18.3 Security Target

ST Version – Version 1.0

ST Date – 27 November 2019

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

1. PrinterLogic Web Stack Web Server (on-premises variant)
    a. Supported on Windows Server 2008 R2, 2012, 2012 R2, or 2016 (64-bit)

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* with the following optional, selection-based, and objective SFRs:
    - FCS_CKM.1(1)
    - FCS_CKM.2
- The following NIAP Technical Decisions apply to this PP 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.
    - 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.
- 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.
- For SFRs that only apply to specific components of the TOE, the SFR is indicated by one or more abbreviations in parentheses placed at the end of the component. Specifically, 'WS' is used to indicate that the component applies to the Web Server and 'WC', 'LC', and 'MC' indicate applicability to Windows Client, Linux Client, and macOS Client, respectively. For example, FCS_HTTPS_EXT.1(WC,LC,MC) applies to all TOE components except for the Web Server component.
- 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 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.

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	A TOE component. 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 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 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 a PL Client application. While configured as a Service Host, a PL Client 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 PrinterLogic Web Stack.
Web Server	A TOE 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 Server 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 Server is an web application that is used to manage on-premise application designed to simplify the management, migration, and provisioning of printers. PrinterLogic Web Stack Server 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 Server is part of a client-server distribution. The TOE is the server portion of this distribution. It interacts with remote PL Client applications in its operational environment.

2.2 Product Overview

This sub-section describes capabilities of the PrinterLogic Web Stack Server 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 Server 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 Server can be used to centrally manage direct IP printing. Printers can be added, modified (e.g. driver, port, name, duplex option), and removed from a centralized Admin Console. These changes are then provisioned to individual PrinterLogic Web Stack Client (PL Client) applications installed on user workstations in the operational environment. End users are provided a Self-Service Portal where they can install additional print drivers above and beyond those provisioned for them. They are also provided a Release Portal where held pull print jobs can be released to a selected printer. Authorizations are based on Active Directory attributes, so users may be given access (or the ability to gain access) to different printers based on role, geographic location, or other attributes.

PrinterLogic Web Stack Server can also be used for centralized auditing and reporting of print jobs by communicating with the PL Client applications in its operational environment. 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, the TOE will interact with the environmental PL Client to ‘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 logging onto the TOE 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. The TOE, along with the environmental PL Client, supports the following methods of cloud printing:

- Email printing: the TOE can be used to configure a PL Client 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: the TOE can be used to configure a PL Client to broadcast itself as an AirPrint-compatible printer so that iOS users can use native iOS printing capabilities. The user provides their credentials to PL Client, which takes the print job and places it into a pull print queue owned by the user.
- Google Cloud Printing: the TOE can be used to configure a PL Client 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 PL Client 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 Server software application (Web Server). The Web Server is deployed on Windows.

The focus of this evaluation is on the TOE functionality supporting the claims in the *Protection Profile for Application Software, Version 1.3*. 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 user and the TOE and between the TOE and the operational environment. Note that the Web Server relies on the underlying Windows OS platform to provide its HTTP server functionality.
- 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 a host platform that is configured for secure operation.
- The ability of the TOE to interface with the low-level components of its host platform in such a manner that the TOE cannot be used as an attack vector to exploit the host platform.
- 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. Administrator opens web browser, navigates to Admin Console on the TOE over TLS/HTTPS connection.
2. Administrator supplies username/password to TOE, which authenticates them locally.
3. If authentication is successful, administrator interacts with Admin Console to change settings for desired clients in the operational environment.
4. Changes are propagated locally to SQL database on Web Server platform and transmitted to desired clients in the operational environment over TLS/HTTPS connection.

User self-service

1. User opens web browser, navigates to Self-Service Portal on the TOE over TLS/HTTPS connection.
5. User supplies username/password to TOE, which authenticates them locally.
2. If authentication is successful, user interacts with Self-Service Portal to manage printer drivers or release held print jobs.
3. If printer drivers are not installed, the TOE will automatically transfer the desired drivers to the environmental PL Client running on the user's system, which then installs the drivers automatically.
4. If print job is released, the TOE will communicate with the environmental PL Client instance that is holding the job over TLS/HTTPS (the instance on the user's local system for pull printing, a Service Host for cloud printing) 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 environmental PL Client.
2. Print job is held by PL Client and PL Client notifies the TOE over TLS/HTTPS that a print job is being held by that user.
3. User releases job by doing one of the following:
   - Navigating to the Release Portal on the TOE, selecting the held job, and specifying a printer to print it to
   - Authenticating to a printer that has been configured to send pull printing requests back to the TOE
4. Regardless of the method used to release the print job, the TOE will notify the PL Client that the job has been released (via TLS/HTTPS) and the PL Client 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 Admin Console, administrator configures an environmental Service Host to poll a particular email box.
2. The configuration change is sent to the Service Host via TLS/HTTPS and also stored by the TOE on the local database.
3. The configuration change is received by the Service Host and stored 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 Service Host will retrieve any emails sent to the polled email box over IMAPS (IMAP over TLS).
6. The Service Host 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 Web Server over HTTPS.
7. If the sender of the email is a valid AD user, the Service Host will hold the print job and notify the TOE (over TLS/HTTPS) that a print job is being held for that user.
8. The user can then release their print job using one of the methods specified in 'pull printing' above.
9. If the sender of the email is not a valid AD user, the Service Host will discard the email.

Email printing (direct)

1. On the Admin Console, administrator configures an environmental 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 Admin Console, 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 Service Host via TLS/HTTPS and also stored by the TOE on the local database.
5. The configuration changes are received by the Service Host and stored 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 Service Host will retrieve any emails sent to the polled email box over IMAPS (IMAP over TLS).
8. The Service Host 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 TOE over HTTPS.
9. If the sender of the email is a valid AD user, the Service Host will immediately print the document using the specified printer.

Email printing (guest)

1. On the Admin Console, administrator configures an environmental 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 in the TOE.
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 Service Host 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 Service Host will then take the email and immediately release it to be printed to the specified guest printer.

AirPrint cloud printing

1. On the Admin Console, administrator enables iOS printing and designates an environmental 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 Service Host as the pull printer.
4. The user will be prompted for their Active Directory credentials, which are validated by the Service Host using LDAPS.

5. Once the user credentials have been validated, the print job will be transmitted to the Service Host via IPPS (IPP over TLS).
6. The Service Host will notify the TOE 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 an environmental Service Host as a pull printer and configures it for mobile printing.
2. Configuration settings are transmitted to the Service Host using TLS/HTTPS.
3. 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 Admin Console 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 Service Host will poll the Google Cloud print server and retrieve print jobs from the print server's queue.
8. The Service Host will perform a reverse AD lookup of the user that submitted the print job.
9. If the user is recognized, the print job is pulled down 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 Service Host notifies the TOE 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 Admin Console, Administrator specifies one or more environmental Service Hosts to function as a local Google Cloud printer.
2. Configuration settings are transmitted to the Service Host using TLS/HTTPS.
3. The Service Host will automatically broadcast itself as a Google Cloud printer.
4. User on Android or Chromebook device will see the Service Host as a valid printer.
5. When the user selects print, the print job is sent directly to the Service Host using TLS/HTTPS rather than to the Google Cloud print server.
6. The Service Host 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 Service Host.
8. The Service Host notifies the TOE 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 (Web Server)

1. Administrator downloads the PrinterLogic Web Stack update from the PrinterLogic web site.
2. Administrator runs the update file, specifying all desired options.
3. 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:

- Provide a mechanism for administrators to access the TSF remotely over a trusted path.
- Provide a mechanism for users to access the TSF remotely over a trusted path.
- Provide an interface for administrators to modify configuration settings of a PL Client application, for these configuration settings to be stored locally on the Web Server platform, and for them to be transmitted securely to the PL Client application over a trusted channel.
- Notify a PL Client application over a trusted channel that a print job has been released.

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 notifies the relevant environmental PL Client, which 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 Server TOE is a PHP application hosted on IIS

The TOE consists of three subsystems: an Admin Console, which provides a graphical user interface (GUI) for administrative functions; a Self-Service Portal, which provides a GUI for end user configuration functions; a Release Portal, which provides a GUI for end users to release held pull print jobs; and a print management client configuration subsystem, which handles the storage and application of configuration settings.

The TOE includes the following running processes:

- CGI/FastCGI
- PrinterLogic Web Stack Client Interface
- PrinterLogic Web Stack Client Launcher
- PrinterLogic Web Stack Client Manager

2.4.1 Physical Boundary

The TOE consists of the following component:

- Web Server application (for Windows)

In Figure 1 below the TOE and its associated MySQL database are indicated by a red box. In the evaluated configuration all other components in Figure 1, including the PrinterLogic Web Stack Client are considered parts of the environment.

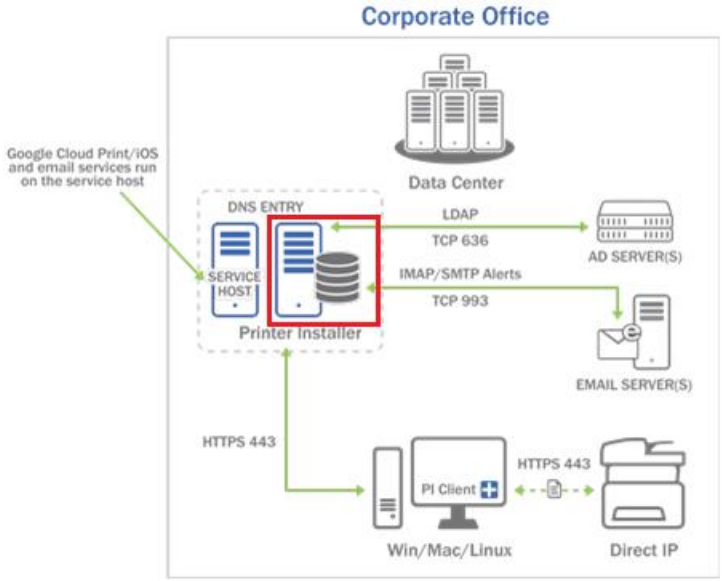

Figure 1: TOE Architecture

TSF-relevant remote interfaces are shown using solid green lines in Figure 1. Note that an environmental Service Host may also reside on a system that is remote from the TOE. In these cases, the same interface that is used by the

TOE 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 platform:

- Windows Server 2008 R2 or higher (64-bit)
- Microsoft IIS 7.0 or higher
- Two 2.0 GHz processors or one 2.0 GHz dual-core processor for up to 15,000 users (add a core for each additional 15,000 users)
- 4GB RAM for up to 15,000 users (add 4GB for each additional 15,000 users)
- 20GB free hard disk space (add 4GB for each additional 100 printers)
- MySQL 5.7 (installed as part of installation package)

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)

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.

The TOE supports the following web browsers:

- Microsoft Edge 40.x and HTML 16.x
- Microsoft Internet Explorer 9 and higher
- Mozilla Firefox 3 and higher
- Google Chrome 45 and higher
- Safari 6.28 and 9.03

The TOE's operational environment includes the following:

- PrinterLogic Web Stack Clients (PL Clients) that are used to facilitate the installation of print drivers and execution of pull/cloud print functions
- Platforms (hardware and software) on which the TOE and PL Client applications are hosted
- Full disk encryption is required for all platforms to ensure adequate data-at-rest protection.
- Windows cryptographic libraries, used to provide cryptographic functionality to the TOE
- Web browser, used to access the TOE GUIs
- MySQL 5.7 database (installed on same local system as the TOE), used to store configuration settings and security data
- Email server, used to hold messages that can be retrieved by a PL Client for pull printing
- Google Cloud print server, used to hold messages that can be retrieved by a PL Client 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 TOE relies on the FIPS-validated cryptographic library cng.sys provided by Windows to perform cryptographic functionality. The TSF encrypts credential data stored by the TOE in the environmental SQL database.

The TOE relies on its underlying OS platform to implement TLS/HTTPS server functionality. The TOE also relies on its underlying OS platform to provide entropy used for key generation.

2.4.2.2 User Data Protection

The TSF leverages functionality provided by their underlying OS platform to secure sensitive data at rest. The TOE uses network resources provided by the underlying platform. All platform services are invoked with user awareness and authorization.

The TOE uses network connectivity to handle interactive user and administrator sessions and to communicate with environmental PL Clients for the purpose of applying configuration changes and updating the status of held print jobs.

2.4.2.3 Security Management

The Web Server provides an Admin Console GUI for configuration of environmental PL Client activity. Specifically, an administrator can designate a PL Client as a Service Host and configure it to work with email printing and mobile printing, thus defining the trusted channels used by a PL Client.

The Web Server also provides Self-Service Portal and Release Portal GUIs that allow users to control printing activity. The Release Portal is used to release print jobs, which prompts secure communications back to environmental PL Clients (Service Hosts) to initiate the print operation.

Authentication to the Web Server is performed using locally-defined credentials. On initial installation, the administrator is prompted to specify credentials to be used for the Admin Console.

TOE configuration data is stored locally in the Windows Registry.

2.4.2.4 Privacy

The TOE does not handle personally identifiable information (PII).

2.4.2.5 Protection of the TSF

The TOE includes measures to integrate securely with its underlying OS platform. The TOE does not perform explicit memory mapping and it does not allocate any memory region with both write and execute permissions. Similarly, the TSF does not write user-modifiable data to directories that contain executable files. The TOE is compatible with its host OS platform when that platform is configured in a secure manner. The TOE is not written in a language that is susceptible to stack-based buffer overflow attacks.

The TOE uses a well-defined set of platform APIs and third party libraries.

The TOE provides the ability for a user/administrator to check its version and to apply updates. Updates are delivered in formats appropriate for the platform on which the TOE is installed. Application of an update removes all executable code associated with the application; there is no way for the application to modify its own code. Updates the TOE are digitally signed, and the signature is validated prior to installation.

2.4.2.6 Trusted Path/Channels

TOE components use trusted paths and channels to secure data in transit. The following interfaces are provided by each TOE component:

- Web Server:
  - TLS/HTTPS server for remote user/administrator access
  - TLS/HTTPS server for changes to PL Client configuration data and pull printing status

2.5 TOE Documentation

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

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

Additional information on installation, configuration and use of the TOE can be found in the PrinterLogic Web Stack online help page, located at https://docs.printerlogic.com/.

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 information related to a user's interaction with printer resources may contain sensitive data that a user expects will not be disclosed to anyone, and because the TOE runs on a general purpose operating system 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 Profile (PP):

- *Protection Profile for Application Software*, version 1.3, 21 March 2019 [App PP]

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]. The PP defines the following extended SAR and SFRs; since they have not been redefined in this ST, the [App PP] 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_RBG_EXT.1 (from [App PP]): Random Bit Generation Services
- FCS_STO_EXT.1 (from [App PP]): Storage of Credentials
- 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
- 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_RBG_EXT.1: Random Bit Generation Services
	FCS_STO_EXT.1: Storage of Credentials
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
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
	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] 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]].

FCS_CKM.2	**Cryptographic Key Establishment**
FCS_CKM.2.1	The application shall [invoke platform-provided 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"]].
FCS_CKM_EXT.1	**Cryptographic Key Generation Services**
FCS_CKM_EXT.1.1	The application shall [invoke platform-provided functionality for asymmetric key generation].
FCS_RBG_EXT.1	**Random Bit Generation Services**
FCS_RBG_EXT.1.1	The application shall [invoke platform-provided DRBG functionality] for its cryptographic functions.
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 [*local user, local administrator, and Console Print Application credentials*]] to non-volatile memory.
Application Note:	*Credentials 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).*

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].
FDP_DEC_EXT.1.2	The application shall restrict its access to [[*SQL database*]].
FDP_NET_EXT.1	**Network Communications**
FDP_NET_EXT.1.1	The application shall restrict network communication to [user-initiated communication for [*remote interaction with GUI*], respond to [*remotely-initiated status communication with PL Client*], [*application-initiated status communication with PL Client, application-initiated status communication with PL Client*]].

5.2.3 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].

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

FMT_SMF.1	Specification of Management Functions
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions [[*specification of endpoints for trusted channels, release of print jobs, configuration of client settings*]].

5.2.4 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].

5.2.5 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.

5.2.6 Trusted Path/Channels (FTP)

FTP_DIT_EXT.1	Protection of Data in Transit

FTP_DIT_EXT.1.1 The application shall [

- invoke platform-provided functionality to encrypt all transmitted sensitive data with [*HTTPS, TLS*]

] between itself and another trusted IT product.

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 the PrinterLogic Web Stack Server. Additionally, when updates are made to the bundled third-party capabilities (MySQL, PHP), 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:

- TLS/HTTPS server (for remote user/administrator authentication using the Self-Service Portal and the Release Portal)
- TLS/HTTPS server (for communication of configuration settings to individual PL Client applications at the request of those applications)

The TOE relies on the underlying OS platform cryptography (via the FIPS-validated cryptographic module cng.sys) to implement the cryptographic primitives for TLS/HTTPS communications. The TOE relies on IIS to provide the TLS/HTTPS protocol stack for these communications.

The TOE relies on its operational environment to 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 TOE invokes environmental DRBG functionality for key generation. 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. The TOE relies on the third-party entropy sources provided by the platform vendor (Microsoft); in this case, it is assumed that this platform provides at least 256 bits of entropy. Key generation is the only TOE function that requires the use of random numbers. Random numbers are obtained from the BCryptGenRandom platform API. The TOE relies on IIS to implement all TLS functionality, so the platform API is itself invoked by the platform.

The TOE uses TLS 1.2 implemented by the OS platform for protection of data in transit. The TOE stores local user/administrator credentials (to the Admin, Self-Service, and Release Portals) and credentials for environmental control panel applications (CPAs) in the MySQL database that is bundled with the TOE installation. This data is protected from unauthorized access using AES encryption of the database itself by the TOE platform's cryptographic

module as well as full disk encryption of the entire platform on which the Web Server and MySQL database are installed (see FDP_DAR_EXT.1).

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

- FCS_CKM.1(1) – The TOE platform generates ECC keys for the purpose of TLS key establishment.
- FCS_CKM.2 – The TOE platform performs ECC key establishment for TLS.
- FCS_CKM_EXT.1 – the TSF relies on the underlying OS platform to provide key generation functionality.
- FCS_RBG_EXT.1 – The TSF relies on the underlying OS platform to provide random bit generation functionality.
- FCS_STO_EXT.1 – TOE components use platform-provided services to store all credential data.

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. Note that while CPA credentials are not used by the TSF to fulfill any security functions in the claimed PP, they are stored by the TOE for non-TSF functions. As a result, user/administrator will have a reasonable expectation that the TSF can protect this data. 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 3 Sensitive Data

Sensitive Data	Stored On	Exchange	Protection At Rest	Protection In Transit
Database key	TOE	Inter-process communication	Platform encryption (full disk)	N/A
Admin Console Credentials (Local)	TOE	Admin's browser to Web Server over browser connection	Platform encryption (full disk + FCS_STO_EXT.1 AES)	HTTPS
Self-Service/Release Portal User Credentials (Local)	TOE	User's browser to Web Server over browser connection	Platform encryption (full disk + FCS_STO_EXT.1 AES)	HTTPS
CPA credential	TOE	Queried by remote printer	Platform encryption (full disk + FCS_STO_EXT.1 AES)	HTTPS
PL Client configuration settings	TOE	Pulled by PL Client instances as needed	Platform encryption (full disk + FCS_STO_EXT.1 AES)	HTTPS
Instructions to hold/release print jobs	N/A	Issued from Web Server to Service Host at user direction	N/A	HTTPS

In the evaluated configuration, the TOE will be installed on a platform that has full disk encryption enabled. All data at rest is ultimately secured by the operational environment's platform encryption functionality, including credential data that is stored in the environmental database and encrypted.

The underlying platform functionality that the TOE interacts with is the system's network connectivity. 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: accessing the Web Server from a remote web browser
- Remotely-initiated: initiation of cloud/email printing that a Service Host is configured to handle, remotely-initiated status communication with environmental PL Client (e.g. notification from PL Client that a print job is being held by a user)
- TSF-initiated: status communication with environmental PL Client (e.g. notification to PL Client that a held print job has been released by a user)

The TOE also interacts with the SQL database residing on its local system when user actions requiring access to it are performed.

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 user-initiated directly through the TSF, remotely initiated by a user performing an action in the operational environment, or initiated by the TOE itself.

6.4 Security Management

The TOE provides a graphical user interface (GUI) that requires user authentication to access. As part of initial setup, the user must specify the username/password of an administrator account before any additional access is granted.

The TOE is protected from unauthorized access via the host platform's file system. By default, the binaries and application data for the TOE are owned by Administrators.

The TOE enforces its security functionality by default upon initial installation and configuration. Configuration settings are defined on the Web Server's local system and stored in the Windows registry.

The TOE can be used to specify the endpoints for trusted channels through configuration of environmental PL Clients. Specifically, a user can specify one or more email servers that a Service Host will retrieve data from for email printing. A user can also designate a Service Host to receive and hold print jobs for AirPrint and Google cloud printing. The TOE specifies this behavior by saving settings in its SQL database. When an environmental PL Client communicates with the TOE, it will be notified if it has been designated as a Service Host; in response to this, it will spawn a process to provide that functionality. The behavior of that process (e.g., what mailbox to monitor for email printing, whether to print or discard email printing requests that come from guest users, what Active Directory repository to use when doing reverse lookups for user identities) is controlled by the TOE settings as well. As this configuration data is stored remotely from the PL Client, it isstored as sensitive data in the SQL database and so it is not subject to unauthorized manipulation.

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 TOE 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 TOE to release the job

In the Admin Console, an administrator may also configure the following PL Client settings:

- Time interval for client check-in (default 240 minutes)
- Enable/disable server initiation of client updates
- Enable/disable designation of individual client as Service Host
- Specify use of HTTPS for client-server communications (Service Host communications are not configurable and use TLS/HTTPS by default)

All other management functionality provided by the product is non-TSF and all other security-relevant settings are established during TOE installation and are non-configurable.

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

- FMT_CFG_EXT.1 – The TOE requires credentials to be defined before use. It is also prevented from direct modification by untrusted users via their host OS platform.
- FMT_MEC_EXT.1 – Locally-modifiable configuration settings for the TOE are stored in an appropriate location.
- FMT_SMF.1 – The TOE can be used by administrators to configure environmental PL Client settings including configuring individual clients to act as Service Hosts and configuring trusted channel endpoints by identifying where Service Hosts will retrieve print jobs. The TOE can also be used by end users to authorize the release of print jobs that are held by Service Hosts.

6.5 Privacy

The TOE does not handle personally identifiable information (PII). The TOE facilitates interaction between users and printer resources but does not directly accept Active Directory credentials or handle print spool data.

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

- FPR_ANO_EXT.1 – the TOE does not handle personally identifiable information.

6.6 Protection of the TSF

The TOE implements several mechanisms to protect against exploitation. The TOE implement address space layout randomization (ASLR) and relies fully on its underlying host platform to perform memory mapping. There is no situation where the TSF maps memory to an explicit address. The TOE (PHP) is interpreted code that does not use just-in-time compilation. It also does not use both PROT_WRITE and PROT_EXEC on the same memory regions. The TOE writes data to the underlying OS platform; however, no data is considered to be user-modifiable. The following directories are used to write and store data:

- web application resides in IIS node of administrator's choosing (e.g., C:\inetpub); PHP writes log data to %TMP% directory on OS platform; configuration data is stored in SQL database installed at location of administrator's choosing.

The TOE is written in interpreted language that relies on the runtime environment to dynamically allocate memory and is therefore not subject to stack-based buffer overflows.

The TOE is designed to run on a host OS platform where platform security features have been enabled (e.g. Windows Defender Export Guard). The TOE uses only documented platform APIs. Appendix A.1 lists the APIs used by the TOE. The TOE also makes use of third-party libraries. Appendix A.2 lists the libraries used by the TOE. The TOE is versioned in the format 'x.y' where x is the year of release and y is the nth release of that year (e.g. in the case of version 18.3, it's the 3rd version released in 2018).

The TOE provides the means to check for, apply, and verify software updates. This is implemented as follows:

A user can check for updates by visiting a link in the Admin Console to check for the latest release. Updates are packaged as .exe files. There is no method to uninstall the application; application of an update will modify existing components as opposed to removing and re-installing components. The current version of the application can be queried by navigating to About PrinterLogic Web Stack in the Admin Console. All updates are digitally signed by

PrinterLogic using 2048-bit RSA signatures. The digital signature is verified automatically by Windows APIs prior to installation.

The TOE is made available as a stand-alone installer 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). Updating the TOE software is the only method of changing its executable code; it does not change its own code. 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 – The TOE interacts with its host OS platform in a manner that does not expose the system to memory-related exploitation.
- FPT_API_EXT.1 – The TOE uses documented platform APIs.
- FPT_IDV_EXT. 1 – The TOE is versioned with the year and month of release.
- FPT_LIB_EXT.1 – The set of third-party libraries used by the TOE 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 platform and removal of the TOE does not preserve any executable code on the platform.

6.7 Trusted Path/Channels

The TOE uses HTTPS / TLSv1.2 to secure sensitive data in transit over trusted channels and paths. The channels and paths supported by the TOE and the protocols used to establish them are listed in section 6.2. All trusted channel communications are provided by the TOE platform.

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

- User and administrator credential data (from user to TOE)
- Configuration information (from user to TOE and between TOE and PL Client)
- Credential data for environmental components (CPA) (from TOE to/from environmental components)
- Authorizations to hold/release print jobs (between TOE and PL Client)

The secure protocols are supported by NIST-validated cryptographic mechanisms provided by the operational environment. 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.

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

- FTP_DIT_EXT.1 – The TOE relies on platform-provided functionality to secure 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] 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.

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.

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]. 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 4 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_RBG_EXT.1	X						
FCS_STO_EXT.1	X						
FDP_DAR_EXT.1		X					
FDP_DEC_EXT.1		X					
FDP_NET_EXT.1		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	
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 TOE.

A.1 Platform APIs

The Windows platform APIs used by the Web Server are listed below. The Web Server also relies on the standalone IIS and MySQL components referenced in section 2.4.1.

- 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.2 Third-Party Libraries

The following section lists the third-party libraries used by the TOE:

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

The Web Server component also uses the following third-party libraries for PHP and Javascript, with applicable version information:

- PHP libraries:
  - adambrett/shell-wrapper 0.8
  - aws/aws-sdk-php 3.19.17
  - aws/aws-sdk-php-laravel 3.1.0
  - barryvdh/laravel-async-queue dev-master de900c0
  - barryvdh/laravel-ide-helper 2.3.2
  - barryvdh/laravel-snappy 0.3.1
  - barryvdh/reflection-docblock 2.0.4
  - bosnadev/repositories 0.9
  - classpreloader/classpreloader 3.0.0
  - danielstjules/stringy 2.3.2
  - dnoegel/php-xdg-base-dir 0.1
  - doctrine/annotations 1.2.7
  - doctrine/cache 1.6.0
  - doctrine/collections 1.3.0
  - doctrine/common 2.6.1
  - doctrine/dbal 2.5.5
  - doctrine/inflector 1.1.0
  - doctrine/instantiator 1.0.5
  - doctrine/lexer 1.0.1
  - evenement/evenement 2.0.1
  - fzaninotto/faker 1.7.0
  - guzzlehttp/guzzle 6.2.2
  - guzzlehttp/promises 1.2.0
  - guzzlehttp/psr7 1.3.1
  - h4cc/wkhtmltoimage-amd64 0.12.3

- h4cc/wkhtmltopdf-amd64 0.12.3
- hamcrest/hamcrest-php 1.2.2
- intervention/image 2.3.8
- jakub-onderka/php-console-color 0.1
- jakub-onderka/php-console-highlighter 0.3.2
- jakub-onderka/php-parallel-lint 0.9.2
- jeremeamia/SuperClosure 2.2.0
- jeroen-g/laravel-packager 1.5
- knplabs/knp-snappy 0.4.3
- laravel/framework 5.2.45
- laravelcollective/html 5.2
- league/event 2.1.2
- league/flysystem 1.0.30
- league/fractal 0.13.0
- league/oauth2-server 4.1.6
- lucadegasperi/oauth2-server-laravel 5.1.4
- mockery/mockery 0.9.5
- monolog/monolog 1.21.0
- mtdowling/cron-expression 1.1.0
- mtdowling/jmespath.php 2.3.0
- myclabs/deep-copy 1.5.4
- nesbot/carbon 1.21.0
- nikic/php-parser 2.1.1
- paragonie/random_compat 1.4.1
- phpdocumentor/reflection-common 1.0
- phpdocumentor/reflection-docblock 3.1.1
- phpdocumentor/type-resolver 0.2
- phpspec/php-diff 1.0.2
- phpspec/phpspec 2.5.3
- phpspec/prophecy 1.6.1
- phpunit/php-code-coverage 4.0.1
- phpunit/php-file-iterator 1.4.1
- phpunit/php-text-template 1.2.1
- phpunit/php-timer 1.0.8
- phpunit/php-token-stream 1.4.8
- phpunit/phpunit 5.6.1
- phpunit/phpunit-mock-objects 3.4.0

- predis/predis 1.1.1
- printerlogic/copy-scan-tracking-pkg 1.0.3
- printerlogic/ms-auth-key-pkg 1.3
- printerlogic/php-coding-standards-pkg 1.1.0
- printerlogic/queuecommber 2.1.2
- printerlogic/site-id-pkg 1.0.0
- psr/http-message 1.0.1
- psr/log 1.0.2
- psy/psysh 0.7.2
- react/cache 0.4.2
- react/dns 0.4.13
- react/event-loop 0.5.2
- react/http 0.8.3
- react/promise 2.5.1
- react/promise-stream 1.1.1
- react/promise-timer 1.3.0
- react/socket 0.8.11
- react/stream 0.7.7
- ringcentral/psr7 1.2.2
- rlerdorf/opcache-status dev-master 4867346
- sebastian/code-unit-reverse-lookup 1.0.0
- sebastian/comparator 1.2.0
- sebastian/diff 1.4.1
- sebastian/environment 1.3.8
- sebastian/exporter 1.2.2
- sebastian/global-state 1.1.1
- sebastian/object-enumerator 1.0.0
- sebastian/recursion-context 1.0.2
- sebastian/resource-operations 1.0.0
- sebastian/version 2.0.0
- spatie/laravel-fractal 1.9.1
- squizlabs/php_codesniffer 3.0.2
- swiftmailer/swiftmailer 5.4.3
- symfony/class-loader 3.2.7
- symfony/console 3.0.9
- symfony/css-selector 3.1.5
- symfony/debug 3.0.9

  - symfony/dom-crawler 3.1.5
  - symfony/event-dispatcher 3.1.6
  - symfony/finder 3.0.9
  - symfony/http-foundation 3.0.9
  - symfony/http-kernel 3.0.9
  - symfony/polyfill-mbstring 1.3.0
  - symfony/polyfill-php56 1.2.0
  - symfony/polyfill-util 1.2.0
  - symfony/process 3.0.9
  - symfony/routing 3.0.9
  - symfony/translation 3.0.9
  - symfony/var-dumper 3.0.9
  - symfony/yaml 3.1.5
  - vlucas/phpdotenv 2.4.0
  - webmozart/assert 1.1.0
  - webpatser/laravel-uuid 2.2.1
  - wemersonjanuario/wkhtmltopdf-windows 0.12.2.3
  - wimg/php-compatibility 8.0.1
- Javascript libraries:
  - JSONStream 1.3.2
  - abbrev 1.1.1
  - accounting 0.4.1
  - acorn 4.0.13
  - ajv 5.5.2
  - ajv-keywords 2.1.1
  - align-text 0.1.4
  - amdefine 1.0.1
  - ansi-gray 0.1.1
  - ansi-regex 2.1.1
  - ansi-styles 2.2.1
  - ansi-wrap 0.1.0
  - aproba 1.2.0
  - archy 1.0.0
  - are-we-there-yet 1.1.4
  - argparse 1.0.9
  - arr-diff 4.0.0
  - arr-flatten 1.1.0

- arr-union 3.1.0
- array-differ 1.0.0
- array-each 1.0.1
- array-filter 0.0.1
- array-find-index 1.0.2
- array-map 0.0.0
- array-reduce 0.0.0
- array-slice 1.1.0
- array-union 1.0.2
- array-uniq 1.0.3
- array-unique 0.3.2
- arrify 1.0.1
- asap 2.0.6
- asn1 0.2.3
- asn1.js 4.9.2
- assert 1.4.1
- assert-plus 1.0.0
- assertion-error 1.1.0
- assign-symbols 1.0.0
- ast-types 0.9.6
- astw 2.2.0
- async 1.5.2
- async-foreach 0.1.3
- asynckit 0.4.0
- atob 2.0.3
- autoprefixer 7.2.5
- aws-sign2 0.7.0
- aws4 1.6.0
- babel-code-frame 6.26.0
- babel-core 6.24.0
- babel-generator 6.26.0
- babel-helper-builder-react-jsx 6.26.0
- babel-helper-call-delegate 6.24.1
- babel-helper-define-map 6.26.0
- babel-helper-function-name 6.24.1
- babel-helper-get-function-arity 6.24.1
- babel-helper-hoist-variables 6.24.1

- babel-helper-optimise-call-expression 6.24.1
- babel-helper-regex 6.26.0
- babel-helper-replace-supers 6.24.1
- babel-helpers 6.24.1
- babel-messages 6.23.0
- babel-plugin-check-es2015-constants 6.22.0
- babel-plugin-syntax-flow 6.18.0
- babel-plugin-syntax-jsx 6.18.0
- babel-plugin-transform-es2015-arrow-functions 6.22.0
- babel-plugin-transform-es2015-block-scoped-functions 6.22.0
- babel-plugin-transform-es2015-block-scoping 6.26.0
- babel-plugin-transform-es2015-classes 6.23.0
- babel-plugin-transform-es2015-computed-properties 6.24.1
- babel-plugin-transform-es2015-destructuring 6.23.0
- babel-plugin-transform-es2015-duplicate-keys 6.24.1
- babel-plugin-transform-es2015-for-of 6.23.0
- babel-plugin-transform-es2015-function-name 6.24.1
- babel-plugin-transform-es2015-literals 6.22.0
- babel-plugin-transform-es2015-modules-amd 6.24.1
- babel-plugin-transform-es2015-modules-commonjs 6.26.0
- babel-plugin-transform-es2015-modules-systemjs 6.24.1
- babel-plugin-transform-es2015-modules-umd 6.24.1
- babel-plugin-transform-es2015-object-super 6.24.1
- babel-plugin-transform-es2015-parameters 6.24.1
- babel-plugin-transform-es2015-shorthand-properties 6.24.1
- babel-plugin-transform-es2015-spread 6.22.0
- babel-plugin-transform-es2015-sticky-regex 6.24.1
- babel-plugin-transform-es2015-template-literals 6.22.0
- babel-plugin-transform-es2015-typeof-symbol 6.23.0
- babel-plugin-transform-es2015-unicode-regex 6.24.1
- babel-plugin-transform-flow-strip-types 6.22.0
- babel-plugin-transform-react-display-name 6.25.0
- babel-plugin-transform-react-jsx 6.23.0
- babel-plugin-transform-react-jsx-self 6.22.0
- babel-plugin-transform-react-jsx-source 6.22.0
- babel-plugin-transform-regenerator 6.26.0
- babel-plugin-transform-strict-mode 6.24.1

- babel-preset-es2015 6.24.0
- babel-preset-flow 6.23.0
- babel-preset-react 6.23.0
- babel-register 6.26.0
- babel-runtime 6.26.0
- babel-template 6.26.0
- babel-traverse 6.23.1
- babel-types 6.26.0
- babelify 7.3.0
- babylon 6.18.0
- balanced-match 1.0.0
- base 0.11.2
- base62 0.1.1
- base64-js 1.2.1
- bcrypt-pbkdf 1.0.1
- beeper 1.1.1
- bl 0.9.5
- block-stream 0.0.9
- bn.js 4.11.8
- boom 4.3.1
- bower 1.8.2
- brace-expansion 1.1.8
- braces 2.3.0
- brorand 1.1.0
- browser-pack 6.0.3
- browser-resolve 1.11.2
- browser-stdout 1.3.0
- browserify 14.3.0
- browserify-aes 1.1.1
- browserify-cipher 1.0.0
- browserify-des 1.0.0
- browserify-rsa 4.0.1
- browserify-sign 4.0.4
- browserify-zlib 0.1.4
- browserslist 2.11.3
- buffer 5.0.8
- buffer-xor 1.0.3

- builtin-modules 1.1.1
- builtin-status-codes 3.0.0
- cache-base 1.0.1
- cached-path-relative 1.0.1
- camelcase 2.1.1
- camelcase-keys 2.1.0
- caniuse-lite 1.0.30000792
- capitalize 1.0.0
- caseless 0.12.0
- center-align 0.1.3
- chai 3.5.0
- chalk 1.1.3
- cipher-base 1.0.4
- circular-json 0.3.3
- class-utils 0.3.6
- classnames 2.2.5
- cliui 3.2.0
- clone 1.0.3
- clone-buffer 1.0.0
- clone-regexp 1.0.0
- clone-stats 0.0.1
- cloneable-readable 1.0.0
- co 4.6.0
- code-point-at 1.1.0
- collection-visit 1.0.0
- color-convert 1.9.1
- color-name 1.1.3
- color-support 1.1.3
- combine-source-map 0.8.0
- combined-stream 1.0.5
- commander 2.9.0
- commoner 0.10.8
- component-emitter 1.2.1
- concat-map 0.0.1
- concat-stream 1.5.2
- concat-with-sourcemaps 1.0.5
- console-browserify 1.1.0

- console-control-strings 1.1.0
- constants-browserify 1.0.0
- convert-source-map 1.5.1
- copy-descriptor 0.1.1
- core-js 1.2.7
- core-util-is 1.0.2
- cosmiconfig 2.2.2
- create-ecdh 4.0.0
- create-hash 1.1.3
- create-hmac 1.1.6
- create-react-class 15.6.2
- cross-spawn 3.0.1
- cryptiles 3.1.2
- crypto-browserify 3.12.0
- css 2.2.1
- currently-unhandled 0.4.1
- dashdash 1.14.1
- date-now 0.1.4
- dateformat 2.0.0
- debug 2.6.9
- debug-fabulous 0.0.4
- decamelize 1.2.0
- decode-uri-component 0.2.0
- deep-eql 0.1.3
- deep-is 0.1.3
- defaults 1.0.3
- define-property 1.0.0
- defined 1.0.0
- del 2.2.2
- delayed-stream 1.0.0
- delegates 1.0.0
- deprecated 0.0.1
- deps-sort 2.0.0
- des.js 1.0.0
- detect-file 1.0.0
- detect-indent 4.0.0
- detect-newline 2.1.0

- detective 4.7.1
- diff 1.4.0
- diffie-hellman 5.0.2
- dom-helpers 3.3.1
- domain-browser 1.1.7
- duplexer 0.1.1
- duplexer2 0.1.4
- ecc-jsbn 0.1.1
- electron-to-chromium 1.3.31
- element-class 0.2.2
- elliptic 6.4.0
- encoding 0.1.12
- end-of-stream 0.1.5
- error-ex 1.3.1
- escape-string-regexp 1.0.5
- escodegen 1.8.1
- esprima 2.7.3
- estraverse 1.9.3
- esutils 2.0.2
- event-stream 3.3.4
- events 1.1.1
- evp_bytestokey 1.0.3
- execall 1.0.0
- exenv 1.2.0
- expand-brackets 2.1.4
- expand-range 1.8.2
- expand-tilde 2.0.2
- extend 3.0.1
- extend-shallow 2.0.1
- extglob 2.0.4
- extsprintf 1.3.0
- faker 4.1.0
- fancy-log 1.3.2
- fast-deep-equal 1.0.0
- fast-json-stable-stringify 2.0.0
- fast-levenshtein 2.0.6
- fbjs 0.8.16

- file-entry-cache 2.0.0
- filename-regex 2.0.1
- fill-range 4.0.0
- find-index 0.1.1
- find-up 1.1.2
- findup-sync 2.0.0
- fined 1.1.0
- first-chunk-stream 1.0.0
- flagged-respawn 1.0.0
- flat-cache 1.3.0
- flatten 1.0.2
- for-in 1.0.2
- for-own 1.0.0
- forever-agent 0.6.1
- form-data 2.3.1
- fragment-cache 0.2.1
- from 0.1.7
- fs.realpath 1.0.0
- fstream 1.0.11
- function-bind 1.1.1
- gauge 2.7.4
- gaze 0.5.2
- get-caller-file 1.0.2
- get-stdin 4.0.1
- get-value 2.0.6
- getpass 0.1.7
- glob 7.1.1
- glob-base 0.3.0
- glob-parent 2.0.0
- glob-stream 3.1.18
- glob-watcher 0.0.6
- glob2base 0.0.12
- global-modules 1.0.0
- global-prefix 1.0.2
- globals 9.18.0
- globby 5.0.0
- globjoin 0.1.4

- globule 0.1.0
- glogg 1.0.1
- graceful-fs 3.0.11
- graceful-readlink 1.0.1
- growl 1.9.2
- gulp 3.9.1
- gulp-babel 6.1.2
- gulp-clean 0.3.2
- gulp-concat 2.6.1
- gulp-rename 1.2.2
- gulp-sass 3.1.0
- gulp-sourcemaps 2.5.1
- gulp-util 3.0.8
- gulplog 1.0.0
- handlebars 4.0.11
- har-schema 2.0.0
- har-validator 5.0.3
- has 1.0.1
- has-ansi 2.0.0
- has-flag 1.0.0
- has-gulplog 0.1.0
- has-unicode 2.0.1
- has-value 1.0.0
- has-values 1.0.0
- hash-base 2.0.2
- hash.js 1.1.3
- hawk 6.0.2
- hmac-drbg 1.0.1
- hoek 4.2.0
- home-or-tmp 2.0.0
- homedir-polyfill 1.0.1
- hosted-git-info 2.5.0
- html-tags 2.0.0
- htmlescape 1.1.1
- http-signature 1.2.0
- https-browserify 1.0.0
- iconv-lite 0.4.19

- ieee754 1.1.8
- ignore 3.3.7
- imurmurhash 0.1.4
- in-publish 2.0.0
- indent-string 2.1.0
- indexes-of 1.0.1
- indexof 0.0.1
- inflight 1.0.6
- inherits 2.0.3
- ini 1.3.5
- inline-source-map 0.6.2
- insert-module-globals 7.0.1
- interpret 1.1.0
- invariant 2.2.2
- invert-kv 1.0.0
- is-absolute 1.0.0
- is-accessor-descriptor 1.0.0
- is-arrayish 0.2.1
- is-buffer 1.1.6
- is-builtin-module 1.0.0
- is-data-descriptor 1.0.0
- is-descriptor 1.0.2
- is-directory 0.3.1
- is-dotfile 1.0.3
- is-equal-shallow 0.1.3
- is-extendable 0.1.1
- is-extglob 2.1.1
- is-finite 1.0.2
- is-fullwidth-code-point 1.0.0
- is-glob 3.1.0
- is-number 3.0.0
- is-odd 1.0.0
- is-path-cwd 1.0.0
- is-path-in-cwd 1.0.0
- is-path-inside 1.0.1
- is-plain-object 2.0.4
- is-posix-bracket 0.1.1

- is-primitive 2.0.0
- is-regexp 1.0.0
- is-relative 1.0.0
- is-stream 1.1.0
- is-supported-regexp-flag 1.0.0
- is-typedarray 1.0.0
- is-unc-path 1.0.0
- is-utf8 0.2.1
- is-windows 1.0.1
- isarray 1.0.0
- isexe 2.0.0
- isobject 3.0.1
- isomorphic-fetch 2.2.1
- isstream 0.1.2
- istanbul 0.4.5
- js-base64 2.4.3
- js-tokens 3.0.2
- js-yaml 3.10.0
- jsbn 0.1.1
- jsesc 1.3.0
- json-schema 0.2.3
- json-schema-traverse 0.3.1
- json-stable-stringify 0.0.1
- json-stringify-safe 5.0.1
- json3 3.3.2
- json5 0.5.1
- jsonify 0.0.0
- jsonparse 1.3.1
- jsprim 1.4.1
- jstransform 10.1.0
- keycode 2.1.9
- kind-of 6.0.2
- known-css-properties 0.3.0
- labeled-stream-splicer 2.0.0
- lazy-cache 2.0.2
- lazy-debug-legacy 0.0.1
- lcid 1.0.0

- levn 0.3.0
- lexical-scope 1.2.0
- liftoff 2.5.0
- load-json-file 1.1.0
- lodash 4.17.4
- lodash._baseassign 3.2.0
- lodash._basecopy 3.0.1
- lodash._basecreate 3.0.3
- lodash._basetostring 3.0.1
- lodash._basevalues 3.0.0
- lodash._escapehtmlchar 2.4.1
- lodash._escapestringchar 2.4.1
- lodash._getnative 3.9.1
- lodash._htmlescapes 2.4.1
- lodash._isiterateecall 3.0.9
- lodash._isnative 2.4.1
- lodash._objecttypes 2.4.1
- lodash._reescape 3.0.0
- lodash._reevaluate 3.0.0
- lodash._reinterpolate 3.0.0
- lodash._reunescapedhtml 2.4.1
- lodash._root 3.0.1
- lodash._shimkeys 2.4.1
- lodash.assign 4.2.0
- lodash.clonedeep 4.5.0
- lodash.create 3.1.1
- lodash.defaults 2.4.1
- lodash.escape 3.2.0
- lodash.isarguments 3.1.0
- lodash.isarray 3.0.4
- lodash.isobject 2.4.1
- lodash.keys 3.1.2
- lodash.memoize 3.0.4
- lodash.mergewith 4.6.0
- lodash.restparam 3.6.1
- lodash.template 3.6.2
- lodash.templatesettings 3.1.1

- lodash.values 2.4.1
- log-symbols 2.2.0
- longest 1.0.1
- loose-envify 1.3.1
- loud-rejection 1.6.0
- lru-cache 2.7.3
- make-iterator 1.0.0
- map-cache 0.2.2
- map-obj 1.0.1
- map-stream 0.1.0
- map-visit 1.0.0
- mathml-tag-names 2.0.1
- md5.js 1.3.4
- meow 3.7.0
- micromatch 3.1.5
- miller-rabin 4.0.1
- mime-db 1.30.0
- mime-types 2.1.17
- minimalistic-assert 1.0.0
- minimalistic-crypto-utils 1.0.1
- minimatch 3.0.4
- minimist 0.0.8
- mixin-deep 1.3.0
- mkdirp 0.5.1
- mocha 3.2.0
- module-deps 4.1.1
- ms 2.0.0
- multipipe 0.1.2
- nan 2.8.0
- nanomatch 1.2.7
- natives 1.1.1
- node-fetch 1.7.3
- node-gyp 3.6.2
- node-sass 4.5.3
- nopt 3.0.6
- normalize-package-data 2.4.0
- normalize-path 2.1.1

- normalize-range 0.1.2
- normalize-selector 0.2.0
- npmlog 4.1.2
- num2fraction 1.2.2
- number-is-nan 1.0.1
- oauth-sign 0.8.2
- object-assign 4.1.1
- object-copy 0.1.0
- object-keys 0.4.0
- object-visit 1.0.1
- object.defaults 1.1.0
- object.map 1.0.1
- object.omit 2.0.1
- object.pick 1.3.0
- once 1.4.0
- optimist 0.6.1
- optionator 0.8.2
- orchestrator 0.3.8
- ordered-read-streams 0.1.0
- os-browserify 0.1.2
- os-homedir 1.0.2
- os-locale 1.4.0
- os-tmpdir 1.0.2
- osenv 0.1.4
- pako 0.2.9
- parents 1.0.1
- parse-asn1 5.1.0
- parse-filepath 1.0.2
- parse-glob 3.0.4
- parse-json 2.2.0
- parse-passwd 1.0.0
- pascalcase 0.1.1
- path-browserify 0.0.0
- path-exists 2.1.0
- path-is-absolute 1.0.1
- path-is-inside 1.0.2
- path-parse 1.0.5

- path-platform 0.11.15
- path-root 0.1.1
- path-root-regex 0.1.2
- path-type 1.1.0
- pause-stream 0.0.11
- pbkdf2 3.0.14
- performance-now 2.1.0
- pi-observer
- pi-react-components
- pify 2.3.0
- pinkie 2.0.4
- pinkie-promise 2.0.1
- posix-character-classes 0.1.1
- postcss 6.0.16
- postcss-less 1.1.3
- postcss-media-query-parser 0.2.3
- postcss-reporter 5.0.0
- postcss-resolve-nested-selector 0.1.1
- postcss-scss 1.0.3
- postcss-selector-parser 2.2.3
- postcss-sorting 3.1.0
- postcss-value-parser 3.3.0
- prelude-ls 1.1.2
- preserve 0.2.0
- pretty-hrtime 1.0.3
- private 0.1.8
- process 0.11.10
- process-nextick-args 1.0.7
- promise 7.3.1
- prop-types 15.6.0
- prop-types-extra 1.0.1
- pseudomap 1.0.2
- public-encrypt 4.0.0
- punycode 1.4.1
- q 1.5.1
- qs 6.5.1
- querystring 0.2.0

- querystring-es3 0.2.1
- randomatic 1.1.7
- randombytes 2.0.6
- randomfill 1.0.3
- react 15.4.2
- react-bootstrap 0.31.5
- react-bootstrap-table 4.3.1
- react-dom 15.4.2
- react-list-select 0.3.0
- react-modal 1.7.3
- react-overlays 0.7.4
- react-s-alert 1.4.1
- react-tools 0.13.3
- reactify 1.1.1
- read-only-stream 2.0.0
- read-pkg 1.1.0
- read-pkg-up 1.0.1
- readable-stream 2.3.3
- recast 0.11.23
- rechoir 0.6.2
- redent 1.0.0
- regenerate 1.3.3
- regenerator-runtime 0.11.1
- regenerator-transform 0.10.1
- regex-cache 0.4.4
- regex-not 1.0.0
- regexpu-core 2.0.0
- regjsgen 0.2.0
- regjsparser 0.1.5
- remove-trailing-separator 1.1.0
- repeat-element 1.1.2
- repeat-string 1.6.1
- repeating 2.0.1
- replace-ext 0.0.1
- request 2.83.0
- require-dir 0.3.2
- require-directory 2.1.1

- require-from-string 1.2.1
- require-main-filename 1.0.1
- resolve 1.5.0
- resolve-dir 1.0.1
- resolve-from 3.0.0
- resolve-url 0.2.1
- riek 1.1.0
- right-align 0.1.3
- rimraf 2.6.2
- ripemd160 2.0.1
- run-sequence 1.2.2
- safe-buffer 5.1.1
- sass-graph 2.2.4
- scss-tokenizer 0.2.3
- semver 4.3.6
- sequencify 0.0.7
- set-blocking 2.0.0
- set-getter 0.1.0
- set-value 2.0.0
- setimmediate 1.0.5
- sha.js 2.4.10
- shasum 1.0.2
- shell-quote 1.6.1
- sigmund 1.0.1
- signal-exit 3.0.2
- slash 1.0.0
- slice-ansi 1.0.0
- snapdragon 0.8.1
- snapdragon-node 2.1.1
- snapdragon-util 3.0.1
- sntp 2.1.0
- source-map 0.5.7
- source-map-resolve 0.5.1
- source-map-support 0.4.18
- source-map-url 0.4.0
- sparkles 1.0.0
- spdx-correct 1.0.2

- spdx-expression-parse 1.0.4
- spdx-license-ids 1.2.2
- specificity 0.3.2
- split 0.3.3
- split-string 3.1.0
- sprintf-js 1.0.3
- sshpk 1.13.1
- static-extend 0.1.2
- stdout-stream 1.4.0
- stream-browserify 2.0.1
- stream-combiner 0.0.4
- stream-combiner2 1.1.1
- stream-consume 0.1.0
- stream-http 2.8.0
- stream-splicer 2.0.0
- string-width 1.0.2
- string_decoder 0.10.31
- stringstream 0.0.5
- strip-ansi 3.0.1
- strip-bom 1.0.0
- strip-bom-string 1.0.0
- strip-indent 1.0.1
- style-search 0.1.0
- stylelint 8.1.1
- stylelint-order 0.7.0
- subarg 1.0.0
- sugarss 1.0.1
- supports-color 2.0.0
- svg-tags 1.0.0
- syntax-error 1.3.0
- table 4.0.2
- tar 2.2.1
- through 2.3.8
- through2 2.0.3
- tildify 1.2.0
- time-stamp 1.1.0
- timers-browserify 1.4.2

- to-arraybuffer 1.0.1
- to-fast-properties 1.0.3
- to-object-path 0.3.0
- to-regex 3.0.1
- to-regex-range 2.1.1
- tough-cookie 2.3.3
- trim-newlines 1.0.0
- trim-right 1.0.1
- tty-browserify 0.0.1
- tunnel-agent 0.6.0
- tweetnacl 0.14.5
- type-check 0.3.2
- type-detect 1.0.0
- typedarray 0.0.6
- ua-parser-js 0.7.17
- uglify-js 2.8.29
- uglify-to-browserify 1.0.2
- umd 3.0.1
- unc-path-regex 0.1.2
- uncontrollable 4.1.0
- union-value 1.0.0
- uniq 1.0.1
- unique-stream 1.0.0
- unset-value 1.0.0
- urix 0.1.0
- url 0.11.0
- use 2.0.2
- user-home 1.1.1
- utf8 2.1.2
- util 0.10.3
- util-deprecate 1.0.2
- uuid 3.2.1
- v8flags 2.1.1
- validate-npm-package-license 3.0.1
- verror 1.10.0
- vinyl 0.5.3
- vinyl-buffer 1.0.0

- vinyl-fs 0.3.14
- vinyl-source-stream 1.1.0
- vinyl-sourcemaps-apply 0.2.1
- vm-browserify 0.0.4
- warning 3.0.0
- whatwg-fetch 2.0.3
- which 1.3.0
- which-module 1.0.0
- wide-align 1.1.2
- window-size 0.1.0
- wordwrap 1.0.0
- wrap-ansi 2.1.0
- wrappy 1.0.2
- write 0.2.1
- xtend 4.0.1
- y18n 3.2.1
- yallist 2.1.2
- yargs 7.1.0
- yargs-parser 5.0.0