The Printer Working Group

June 29, 2018
White Paper

IPP Authentication Methods
(IPPAUTH)

Status: Interim

Abstract: This document is a whitepaper that describes the interaction between IPP and various authentication mechanisms used over IPP's HTTP, HTTPS and TLS transports, and how their nuances can affect the authentication user experience on IPP Client systems.

This document is a White Paper. For the definition of a "White Paper", see:

http://ftp.pwg.org/pub/pwg/general/pwg-process30.pdf

This document is available electronically at:

http://ftp.pwg.org/pub/pwg/ipp/whitepaper/tb-ippauth-20180629.odt
http://ftp.pwg.org/pub/pwg/ipp/whitepaper/tb-ippauth-20180629.pdf

Title: IPP Authentication Methods *(IPPAUTH)*

The material contained herein is not a license, either expressed or implied, to any IPR owned or controlled by any of the authors or developers of this material or the Printer Working Group. The material contained herein is provided on an “AS IS” basis and to the maximum extent permitted by applicable law, this material is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material and the Printer Working Group and its members hereby disclaim all warranties and conditions, either expressed, implied or statutory, including, but not limited to, any (if any) implied warranties that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

Table of Contents

List of Figures

List of Tables

1. Introduction

The Internet Printing Protocol (hereafter, IPP) uses HTTP as its underlying transport [RFC8010]. When an IPP Printer is configured to limit access to its services to only those Clients operated by an authorized User, it challenges the User's Client by employing one of the HTTP authentication methods. But an IPP Client isn't usually a typical HTTP User Agent (e.g. it isn't a commonly used Web browser). This white paper examines the common HTTP authentication methods employed today and outlines limits, constraints and conventions that ought to be considered when implementing support for one of these different HTTP authentication methods to ensure a high quality printing user experience.

2. Terminology

2.1. Protocol Roles Terminology

This document defines the following protocol roles in order to specify unambiguous conformance requirements:

Client: Initiator of outgoing IPP session requests and sender of outgoing IPP operation requests (Hypertext Transfer Protocol -- HTTP/1.1 [RFC7230] User Agent).

Printer: Listener for incoming IPP session requests and receiver of incoming IPP operation requests (Hypertext Transfer Protocol -- HTTP/1.1 [RFC7230] Server) that represents one or more Physical Devices or a Logical Device.

2.2. Other Terms Used in This Document

User: A person or automata using a Client to communicate with a Printer.

2.3. Acronyms and Organizations

IANA: Internet Assigned Numbers Authority, http://www.iana.org/

IETF: Internet Engineering Task Force, http://www.ietf.org/

ISO: International Organization for Standardization, http://www.iso.org/

PWG: Printer Working Group, http://www.pwg.org/

3. Overview of IPP Authentication Methods

This white paper describes how various HTTP based authentication systems integrate into IPP communications between a Client and a Printer. Although the authentication protocols themselves do not need to change to be integrated into IPP communications, the IPP Client is not a Web browser, so IPP Client and Printer implementors ought to consider factors that can improve or degrade the user experience.

3.1. Client Authentication Methods

A Printer uses the “authenticated identity” or the “most authenticated user” [RFC8011] to determine whether to allow the requesting Client access to capabilities such as operations, resources, and attributes. Authentication is the process of establishing some level of trust that an entity is who or what they are claiming to be. An IPP Printer specifies its supported authentication methods via several IPP attributes. The “uri-authentication-supported” attribute [RFC8011] indicates the authentication method used for a corresponding URI in “printer-uri-supported” [RFC8011]. The “xri-authentication” member attribute of “printer-xri-supported” [RFC3380] specifies the same corresponding values, if the Printer implements the “printer-xri-supported” attribute.

In some cases, the Printer is not directly involved in the authentication process, and may not be directly aware of the User's identity following authentication. In these cases, the Printer might still need to acquire the User's identity in order to accurately document the User's identity in the Job Object's Job Status attributes, or to support IPP operations such as Get-User-Printer-Attributes [IPPGUPA] that depend on the User's identity to provide meaningfully filtered operation responses.

Each of the authentication method keywords currently registered for “uri-authentication-supported” is described below, with an accompanying sequence diagram for illustration purposes, as well as a discussion of each method's advantages and shortcomings.

3.1.1. The 'none' IPP Authentication Method

The 'none' IPP Authentication Method [RFC8011] very simply indicates that the receiving Printer is provided no method whatsoever to determine the identity of the User who is operating the Client that is making IPP operation requests. The user name for the operation is assumed to be 'anonymous'. This method is not recommended unless the Printer's operator has the objective of providing an anonymous print service. In most cases, the Client SHOULD provide the “requesting-user-name” operation attribute, as described in section 3.1.2.

Figure 3.1 illustrates how the 'none' authentication method integrates into an IPP operation request / response exchange. Other authentication methods will expand on this baseline request / response exchange.

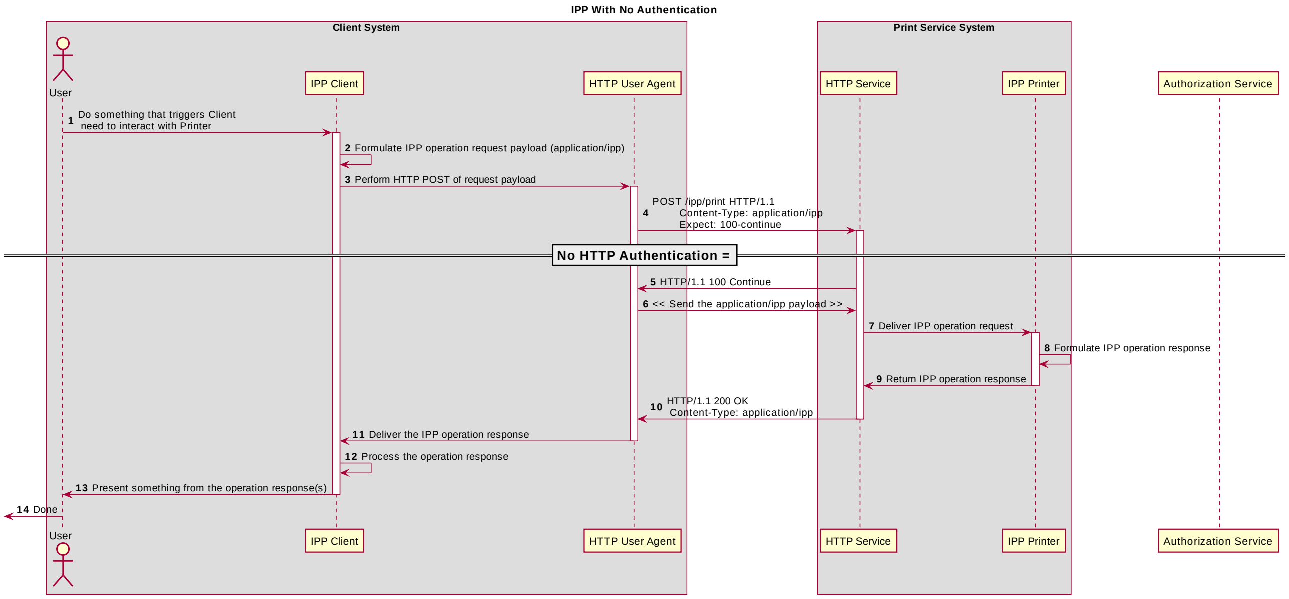
IPP With No Authentication
Client System
Print Service System
User
IPP Client
HTTP User Agent
HTTP Service
IPP Printer
Authorization Service
1 Do something that triggers Client need to interact with Printer
2 Formulate IPP operation request payload (application/ipp)
3 Perform HTTP POST of request payload
4 POST /ipp/print HTTP/1.1 Content-Type: application/ipp Expect: 100-continue
No HTTP Authentication =
5 HTTP/1.1 100 Continue
6 << Send the application/ipp payload >>
7 Deliver IPP operation request
8 Formulate IPP operation response
9 Return IPP operation response
10 HTTP/1.1 200 OK Content-Type: application/ipp
11 Deliver the IPP operation response
12 Process the operation response
13 Present something from the operation response(s)
14 Done
User
IPP Client
HTTP User Agent
HTTP Service
IPP Printer
Authorization Service

Figure 3.1: Sequence diagram for the 'none' IPP Authentication Method

3.1.2. The 'requesting-user-name' IPP Authentication Method

In the 'requesting-user-name' IPP Authentication Method [RFC8011], the Client MUST provides the “requesting-user-name” operation attribute [RFC8011] in its IPP operation request. The Printer uses this unauthenticated name as the identity of the actor operating the Client. This method is not recommended since there is no actual authentication performed as there is no credential provided to prove the identity claimed in the “requesting-user-name”.

Figure 3.2 illustrates how the 'requesting-user-name' authentication method integrates into an IPP operation request / response exchange. This is basically identical to the 'none' method from a protocol perspective.

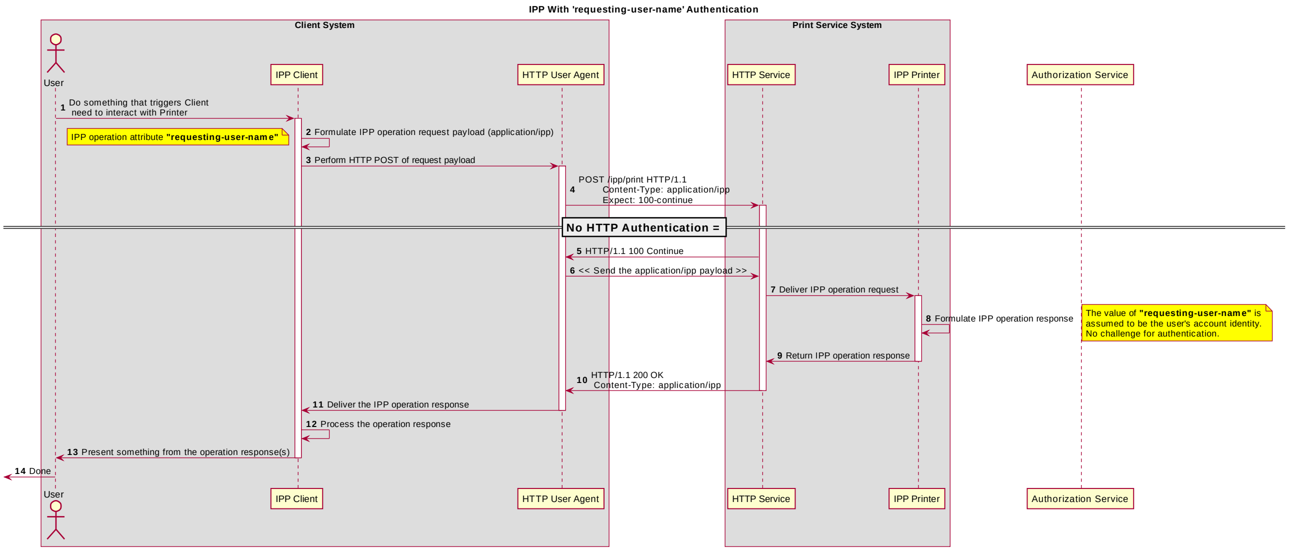
IPP With 'requesting-user-name' Authentication
Client System
Print Service System
User
IPP Client
HTTP User Agent
HTTP Service
IPP Printer
Authorization Service
1 Do something that triggers Client need to interact with Printer
IPP operation attribute "requesting-user-name"
2 Formulate IPP operation request payload (application/ipp)
3 Perform HTTP POST of request payload
4 POST /ipp/print HTTP/1.1 Content-Type: application/ipp Expect: 100-continue
No HTTP Authentication =
5 HTTP/1.1 100 Continue
6 << Send the application/ipp payload >>
7 Deliver IPP operation request
8 Formulate IPP operation response
The value of "requesting-user-name" is assumed to be the user's account identity. No challenge for authentication.
9 Return IPP operation response
10 HTTP/1.1 200 OK Content-Type: application/ipp
11 Deliver the IPP operation response
12 Process the operation response
13 Present something from the operation response(s)
14 Done
User
IPP Client
HTTP User Agent
HTTP Service
IPP Printer
Authorization Service

Figure 3.2: Sequence diagram for the 'requesting-user-name' IPP Authentication Method

3.1.3. The 'basic' IPP Authentication Method

The 'basic' IPP Authentication Method uses HTTP Basic authentication scheme [RFC7617]. It is employed in IPP in much the same way that it is employed in conventional HTTP workflows using a Web browser. When the IPP Client encounters an HTTP 401 Unauthorized response, it evaluates whether it supports the authentication method identified by the value of the "WWW-Authenticated" header in the response. In this case, if it supports 'basic', it will present UI asking the User to provide username and password credentials that may be used to authenticate with the HTTP Server providing access to the IPP Printer. If the HTTP Server successfully authenticates that set of credentials, then the IPP operation request is passed on to the IPP Printer, which responds as usual.

Figure 3.3 illustrates how the 'basic' authentication method integrates into an IPP operation request / response exchange.

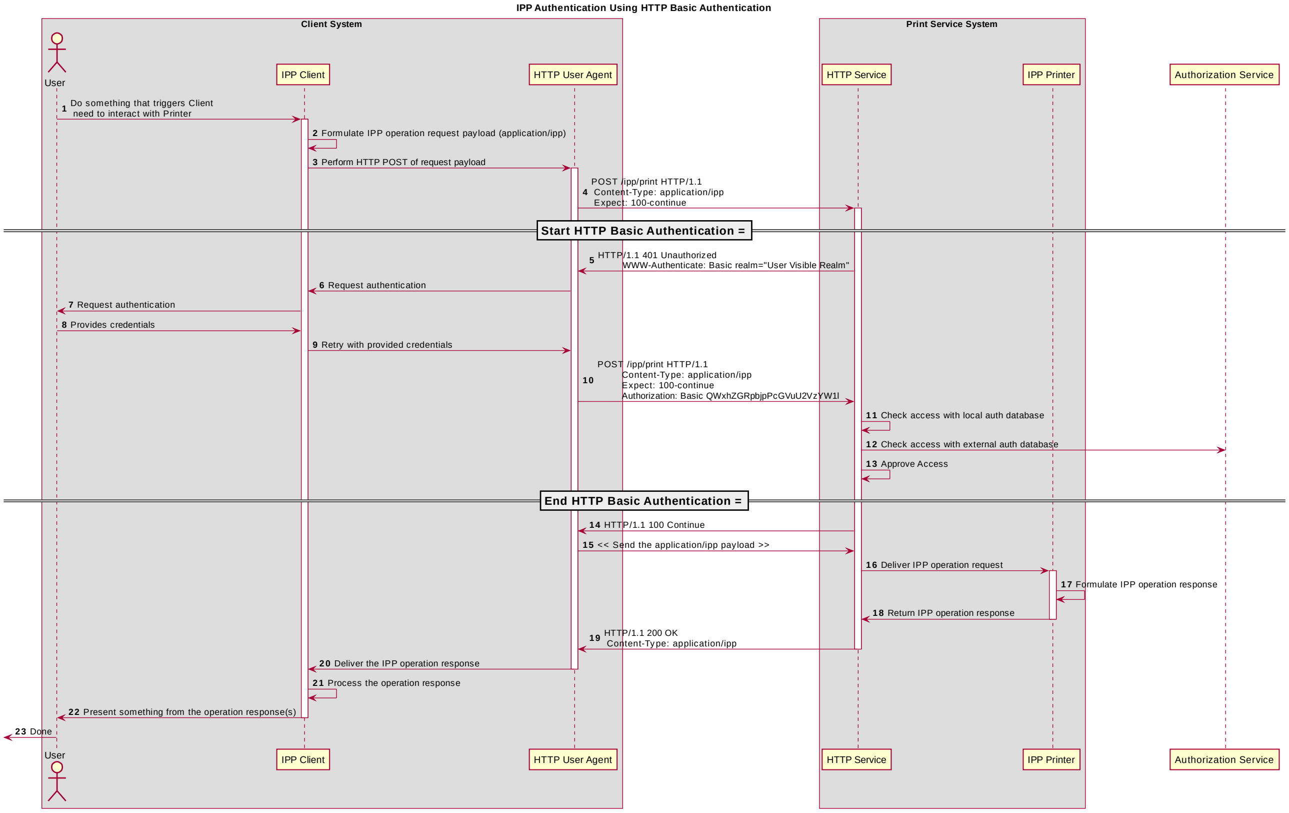
IPP Authentication Using HTTP Basic Authentication
Client System
Print Service System
User
IPP Client
HTTP User Agent
HTTP Service
IPP Printer
Authorization Service
1 Do something that triggers Client need to interact with Printer
2 Formulate IPP operation request payload (application/ipp)
3 Perform HTTP POST of request payload
4 POST /ipp/print HTTP/1.1 Content-Type: application/ipp Expect: 100-continue
Start HTTP Basic Authentication =
5 HTTP/1.1 401 Unauthorized WWW-Authenticate: Basic realm="User Visible Realm"
6 Request authentication
7 Request authentication
8 Provides credentials
9 Retry with provided credentials
10 POST /ipp/print HTTP/1.1 Content-Type: application/ipp Expect: 100-continue Authorization: Basic QWxhZGRpbjpPcGVuU2VzYW1l
11 Check access with local auth database
12 Check access with external auth database
13 Approve Access
End HTTP Basic Authentication =
14 HTTP/1.1 100 Continue
15 << Send the application/ipp payload >>
16 Deliver IPP operation request
17 Formulate IPP operation response
18 Return IPP operation response
19 HTTP/1.1 200 OK Content-Type: application/ipp
20 Deliver the IPP operation response
21 Process the operation response
22 Present something from the operation response(s)
23 Done
User
IPP Client
HTTP User Agent
HTTP Service
IPP Printer
Authorization Service

Figure 3.3: Sequence diagram for the 'basic' IPP Authentication Method

3.1.4. The 'digest' IPP Authentication Method

The 'digest' IPP Authentication method uses the HTTP Digest authentication scheme [RFC7616]. It is employed in IPP in much the same way that it is employed in conventional HTTP workflows using a Web browser; when the IPP Client encounters an HTTP 401 Unauthorized response, it evaluates whether it supports the authentication method identified by the value of the “WWW-Authenticated” header in the response. In this case, if it supports 'digest', it will present UI asking the User to provide username and password credentials that may be used to authenticate with the HTTP Server providing access to the IPP Printer. If the HTTP Server successfully authenticates that set of credentials, then the IPP operation request is passed on to the IPP Printer, which responds as usual.

Figure 3.4 illustrates how the 'digest' authentication method integrates into an IPP operation request / response exchange.

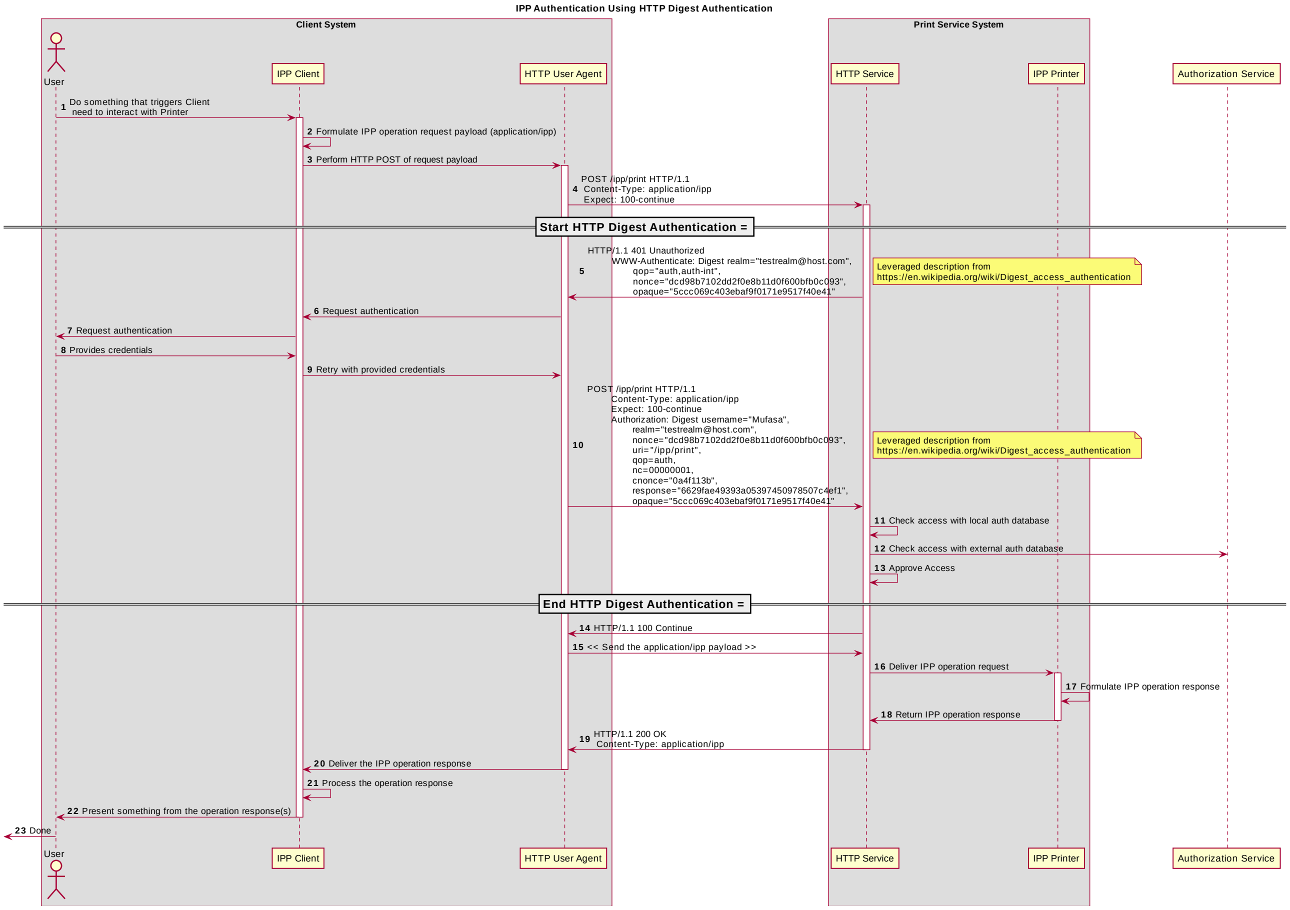
IPP Authentication Using HTTP Digest Authentication
Client System
Print Service System
User
IPP Client
HTTP User Agent
HTTP Service
IPP Printer
Authorization Service
1 Do something that triggers Client need to interact with Printer
2 Formulate IPP operation request payload (application/ipp)
3 Perform HTTP POST of request payload
4 POST /ipp/print HTTP/1.1
Content-Type: application/ipp
Expect: 100-continue
Start HTTP Digest Authentication =
5 HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest realm="testrealm@host.com",
qop="auth,auth-int",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
opaque="5ccc069c403ebaf9f0171e9517f40e41"
Leveraged description from https://en.wikipedia.org/wiki/Digest_access_authentication
6 Request authentication
7 Request authentication
8 Provides credentials
9 Retry with provided credentials
10 POST /ipp/print HTTP/1.1
Content-Type: application/ipp
Expect: 100-continue
Authorization: Digest username="Mufasa",
realm="testrealm@host.com",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
uri="/ipp/print",
qop=auth,
nc=00000001,
cnonce="0a4f113b",
response="6629fae49393a05397450978507c4ef1",
opaque="5ccc069c403ebaf9f0171e9517f40e41"
Leveraged description from https://en.wikipedia.org/wiki/Digest_access_authentication
11 Check access with local auth database
12 Check access with external auth database
13 Approve Access
End HTTP Digest Authentication =
14 HTTP/1.1 100 Continue
15 << Send the application/ipp payload >>
16 Deliver IPP operation request
17 Formulate IPP operation response
18 Return IPP operation response
19 HTTP/1.1 200 OK
Content-Type: application/ipp
20 Deliver the IPP operation response
21 Process the operation response
22 Present something from the operation response(s)
23 Done
User
IPP Client
HTTP User Agent
HTTP Service
IPP Printer
Authorization Service

Figure 3.4: Sequence diagram for the 'digest' IPP Authentication Method

3.1.5. The 'negotiate' IPP Authentication Method

The 'negotiate' IPP Authentication method uses the HTTP Negotiate authentication scheme [RFC4559], which is used to support Kerberos and NTLM authentication methods with HTTP.

Figure 3.5 illustrates how the 'negotiate' authentication method integrates into an IPP operation request / response exchange.

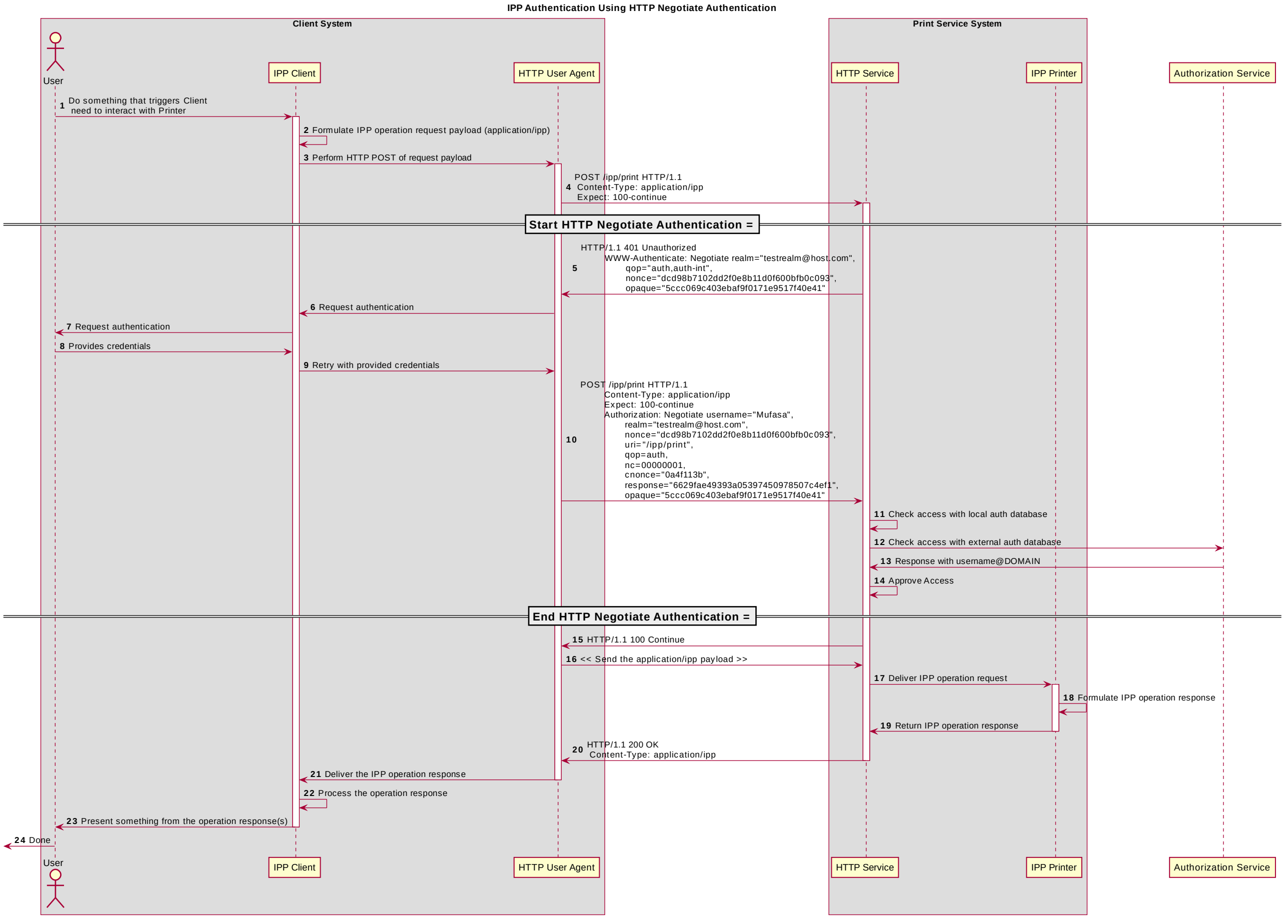
IPP Authentication Using HTTP Negotiate Authentication
Client System
Print Service System
User
IPP Client
HTTP User Agent
HTTP Service
IPP Printer
Authorization Service
1 Do something that triggers Client need to interact with Printer
2 Formulate IPP operation request payload (application/ipp)
3 Perform HTTP POST of request payload
4 POST /ipp/print HTTP/1.1 Content-Type: application/ipp Expect: 100-continue
Start HTTP Negotiate Authentication =
5 HTTP/1.1 401 Unauthorized WWW-Authenticate: Negotiate realm="testrealm@host.com", qop="auth,auth-int", nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093", opaque="5ccc069c403ebaf9f0171e9517f40e41"
6 Request authentication
7 Request authentication
8 Provides credentials
9 Retry with provided credentials
10 POST /ipp/print HTTP/1.1 Content-Type: application/ipp Expect: 100-continue Authorization: Negotiate username="Mufasa", realm="testrealm@host.com", nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093", uri="/ipp/print", qop=auth, nc=00000001, cnonce="0a4f113b", response="6629fae49393a05397450978507c4ef1", opaque="5ccc069c403ebaf9f0171e9517f40e41"
11 Check access with local auth database
12 Check access with external auth database
13 Response with username@DOMAIN
14 Approve Access
End HTTP Negotiate Authentication =
15 HTTP/1.1 100 Continue
16 << Send the application/ipp payload >>
17 Deliver IPP operation request
18 Formulate IPP operation response
19 Return IPP operation response
20 HTTP/1.1 200 OK Content-Type: application/ipp
21 Deliver the IPP operation response
22 Process the operation response
23 Present something from the operation response(s)
24 Done
User
IPP Client
HTTP User Agent
HTTP Service
IPP Printer
Authorization Service

Figure 3.5 : Sequence diagram for the 'negotiate' IPP Authentication Method

3.1.6. The 'oauth' IPP Authentication Method

The 'oauth' IPP Authentication method uses the OAuth2 authentication scheme [RFC6749] [RFC6749] and the OAuth2 Bearer Token [RFC6750]. Figure 3.6 illustrates how the 'oauth' authentication method integrates into an IPP operation request / response exchange.

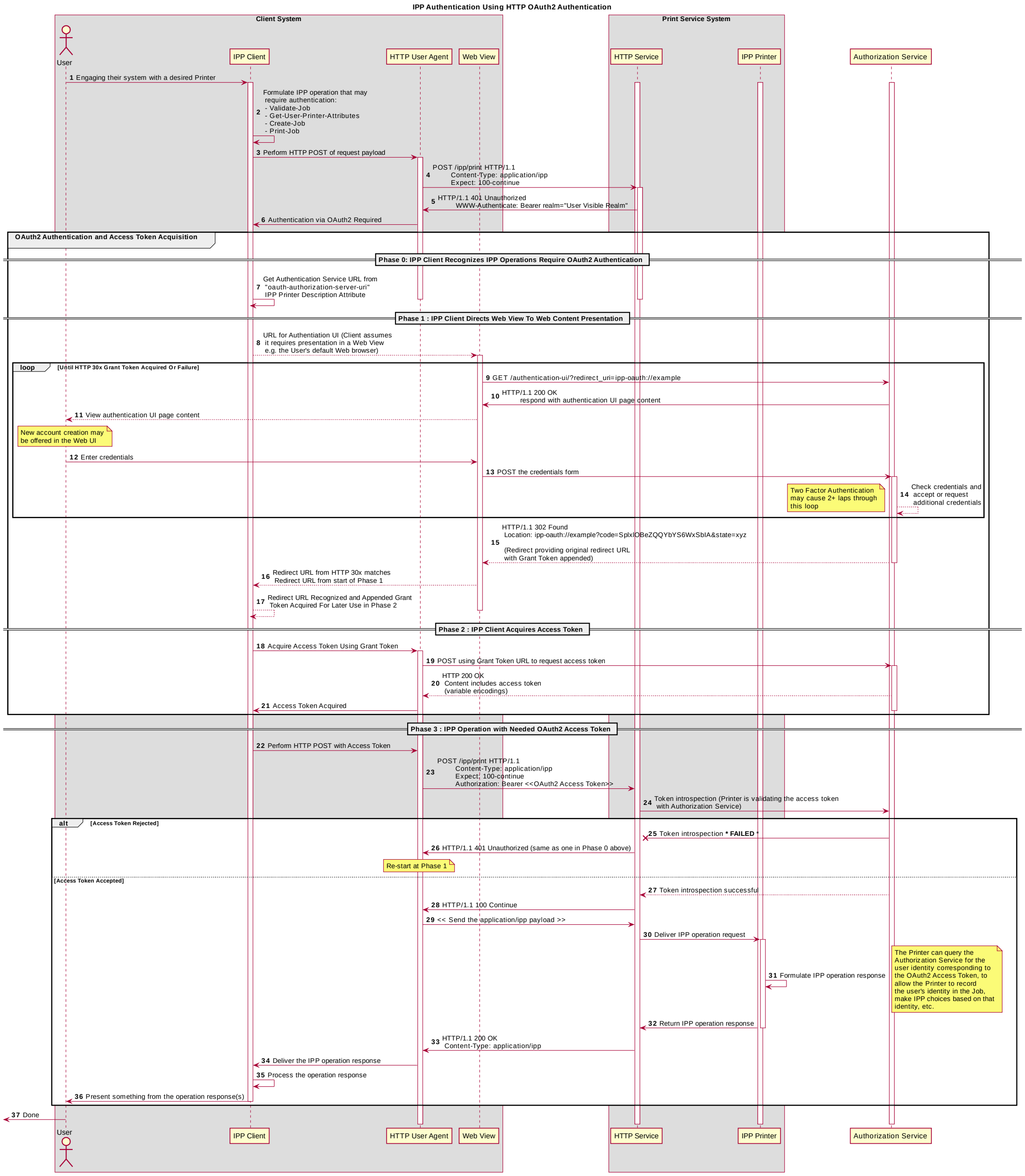
IPP Authentication Using HTTP OAuth2 Authentication
Client System
Print Service System
User
IPP Client
HTTP User Agent
Web View
HTTP Service
IPP Printer
Authorization Service
1 Engaging their system with a desired Printer
2 Formulate IPP operation that may require authentication: - Validate-Job - Get-User-Printer-Attributes - Create-Job - Print-Job
3 Perform HTTP POST of request payload
4 POST /ipp/print HTTP/1.1 Content-Type: application/ipp Expect: 100-continue
5 HTTP/1.1 401 Unauthorized WWW-Authenticate: Bearer realm="User Visible Realm"
6 Authentication via OAuth2 Required
OAuth2 Authentication and Access Token Acquisition
Phase 0: IPP Client Recognizes IPP Operations Require OAuth2 Authentication
7 Get Authentication Service URL from "oauth-authorization-server-uri" IPP Printer Description Attribute
Phase 1 : IPP Client Directs Web View To Web Content Presentation
8 URL for Authentiation UI (Client assumes it requires presentation in a Web View e.g. the User's default Web browser)
loop [Until HTTP 30x Grant Token Acquired Or Failure]
9 GET /authentication-ui/?redirect_uri=ipp-oauth://example
10 HTTP/1.1 200 OK respond with authentication UI page content
11 View authentication UI page content
New account creation may be offered in the Web UI
12 Enter credentials
13 POST the credentials form
Two Factor Authentication may cause 2+ laps through this loop
14 Check credentials and accept or request additional credentials
15 HTTP/1.1 302 Found Location: ipp-oauth://example?code=SplxlOBeZQQYbYS6WxSbIA&state=xyz (Redirect providing original redirect URL with Grant Token appended)
16 Redirect URL from HTTP 30x matches Redirect URL from start of Phase 1
17 Redirect URL Recognized and Appended Grant Token Acquired For Later Use in Phase 2
Phase 2 : IPP Client Acquires Access Token
18 Acquire Access Token Using Grant Token
19 POST using Grant Token URL to request access token
20 HTTP 200 OK Content includes access token (variable encodings)
21 Access Token Acquired
Phase 3 : IPP Operation with Needed OAuth2 Access Token
22 Perform HTTP POST with Access Token
23 POST /ipp/print HTTP/1.1 Content-Type: application/ipp Expect: 100-continue Authorization: Bearer <<OAuth2 Access Token>>
24 Token introspection (Printer is validating the access token with Authorization Service)
alt [Access Token Rejected]
25 Token introspection * FAILED *
26 HTTP/1.1 401 Unauthorized (same as one in Phase 0 above)
Re-start at Phase 1
[Access Token Accepted]
27 Token introspection successful
28 HTTP/1.1 100 Continue
29 << Send the application/ipp payload >>
30 Deliver IPP operation request
31 Formulate IPP operation response
The Printer can query the Authorization Service for the user identity corresponding to the OAuth2 Access Token, to allow the Printer to record the user's identity in the Job, make IPP choices based on that identity, etc.
32 Return IPP operation response
33 HTTP/1.1 200 OK Content-Type: application/ipp
34 Deliver the IPP operation response
35 Process the operation response
36 Present something from the operation response(s)
37 Done
User
IPP Client
HTTP User Agent
Web View
HTTP Service
IPP Printer
Authorization Service

Figure 3.6 : Sequence diagram for the 'oauth' IPP Authentication Method

3.1.7. The 'certificate' IPP Authentication Method

The 'certificate' IPP Authentication method uses X.509 certificate authentication via TLS. X.509 certificate authentication via TLS is initiated by the Printer by sending a Certificate Request message during the Transport Layer Security (TLS) [RFC5246] handshake. The Client then sends the X.509 certificate identifying the User and/or Client in a corresponding Certificate message, and a subsequent Certificate Verify message to prove to the Printer that the Client has the corresponding private key. If the Client has no configured X.509 certificate to provide, it sends an empty Certificate message.

The Printer SHOULD allow both empty and valid X.509 certificates. The Printer SHOULD return the IPP status code listed in Table 3.1 when the corresponding authentication exception occurs. The Client SHOULD respond to the reported status code with the corresponding response listed in Table 3.1.

Operation Status Code	Authentication Exception	Recommended Client Response
'client-error-not-authenticated'	Authentication required but no X.509 certificate supplied	Close the connection; select a certificate (with possible user interaction); retry connection with selected certificate
'client-error-not-authorized'	Access denied for the identity specified by the provided X.509 certificate; try again	Close the connection; select a different certificate (with possible user interaction); retry connection with selected certificate
'client-error-forbidden'	Access denied for the identity specified by the provided X.509 certificate; don't try again	Close the connection and present User with error dialog (“Access denied”)

Table 3.1 : IPP 'certificate' Authentication Method Error Condition Status Codes

Figure 3.7 illustrates how the TLS authentication method integrates into an IPP operation request / response exchange.

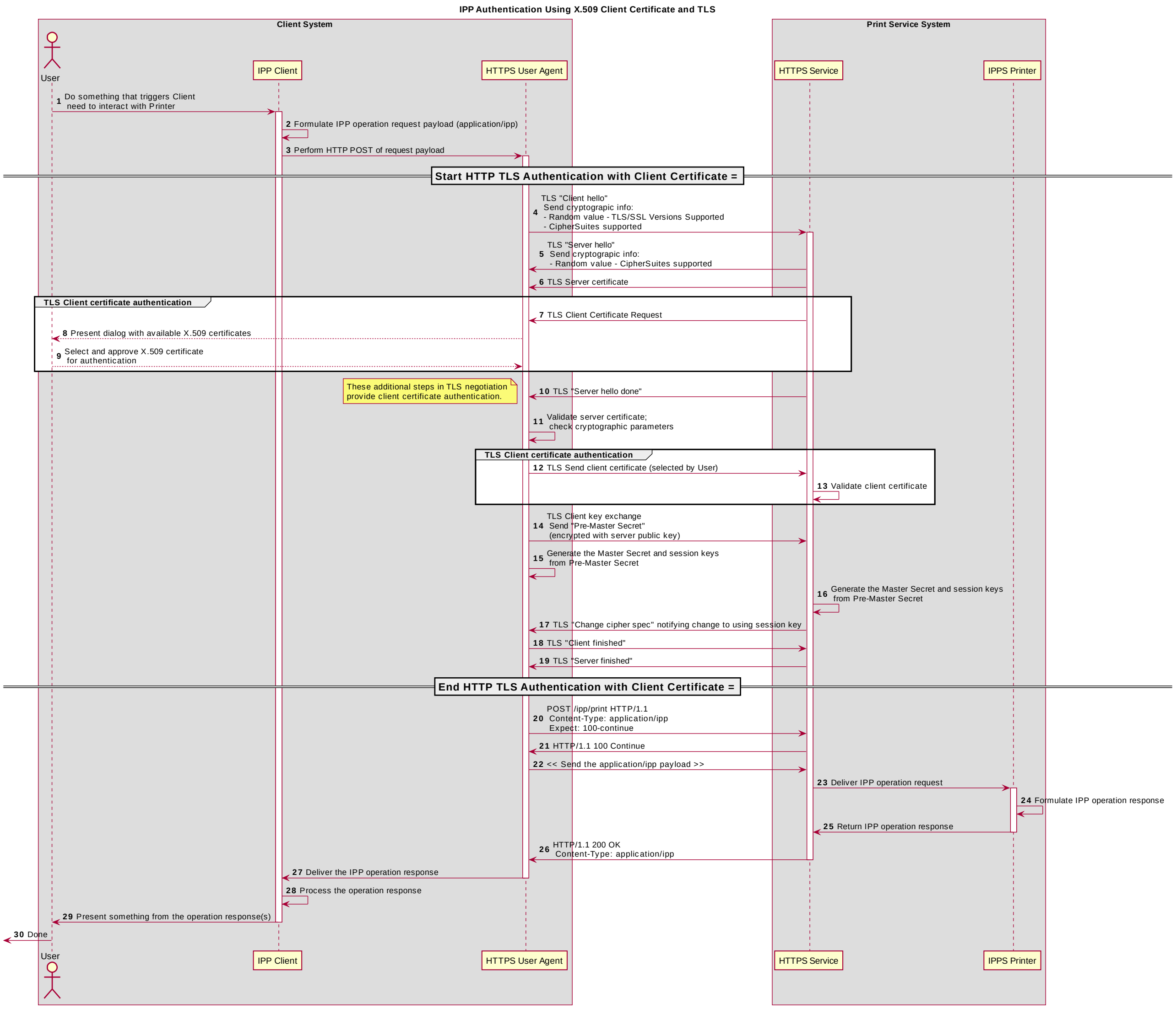
IPP Authentication Using X.509 Client Certificate and TLS
Client System
Print Service System
User
IPP Client
HTTPS User Agent
HTTPS Service
IPPS Printer
1 Do something that triggers Client need to interact with Printer
2 Formulate IPP operation request payload (application/ipp)
3 Perform HTTP POST of request payload
Start HTTP TLS Authentication with Client Certificate =
4 TLS "Client hello" Send cryptograpic info: - Random value - TLS/SSL Versions Supported - CipherSuites supported
5 TLS "Server hello" Send cryptograpic info: - Random value - CipherSuites supported
6 TLS Server certificate
TLS Client certificate authentication
7 TLS Client Certificate Request
8 Present dialog with available X.509 certificates
9 Select and approve X.509 certificate for authentication
These additional steps in TLS negotiation provide client certificate authentication.
10 TLS "Server hello done"
11 Validate server certificate; check cryptographic parameters
TLS Client certificate authentication
12 TLS Send client certificate (selected by User)
13 Validate client certificate
14 TLS Client key exchange Send "Pre-Master Secret" (encrypted with server public key)
15 Generate the Master Secret and session keys from Pre-Master Secret
16 Generate the Master Secret and session keys from Pre-Master Secret
17 TLS "Change cipher spec" notifying change to using session key
18 TLS "Client finished"
19 TLS "Server finished"
End HTTP TLS Authentication with Client Certificate =
20 POST /ipp/print HTTP/1.1 Content-Type: application/ipp Expect: 100-continue
21 HTTP/1.1 100 Continue
22 << Send the application/ipp payload >>
23 Deliver IPP operation request
24 Formulate IPP operation response
25 Return IPP operation response
26 HTTP/1.1 200 OK Content-Type: application/ipp
27 Deliver the IPP operation response
28 Process the operation response
29 Present something from the operation response(s)
30 Done
User
IPP Client
HTTPS User Agent
HTTPS Service
IPPS Printer

Figure 3.7 : Sequence diagram for X.509 Certificate Authentication Via TLS

4. Implementation Recommendations

Provide possible technical solutions/approaches in this section. Include pros and cons for each technical solution or approach. Include references to specific protocols and/or data models when appropriate. Include mapping and gateway considerations when appropriate.

4.1. Client Implementation Recommendations

4.1.1. General Recommendations

A Client SHOULD limit the number of additional windows presented to the user during the course of an authentication workflow, to avoid causing a fragmented, disruptive user experience.

4.1.2. Handling Authentication Failure

If a Printer rejects authentication credentials provided by a Client in response to an authentication challenge following an IPP operation request, the Printer MAY return an IPP operation response. If it does not, and the connection is left open, it SHOULD treat the connection the same way it handles a stalled connection, and close it after a reasonably brief amount of time.

4.1.3. OAuth2 Recommendations

The OAuth2 authorization service may have a complicated user presentation. If possible, select a presentation alternative that is the least complicated or the most similar to the user experience provided for older authentication methods (HTTP Basic or HTTP Digest) that may be more familiar to the user.

4.2. Printer Implementation Recommendations

4.2.1. Handling Authentication Failure

If a Printer receives an IPP operation request, challenges the Client for authentication, and the authentication process fails, the Printer SHOULD send an appropriate IPP operation response indicating the cause of the failure.

4.2.2. OAuth2 Recommendations

To align with existing Client authentication user experience for HTTP Basic or HTTP Digest authentication, the OAuth2 Authentication Server SHOULD use HTTP Basic or HTTP Digest authentication rather than presenting an authentication dialog page using its own web content. If that isn't practical, an OAuth2 Authorization Service used in an IPP printing workflow SHOULD direct a Client to an authentication page that facilitates an appropriate presentation on even limited Client systems such as smart phones.

5. Internationalization Considerations

For interoperability and basic support for multiple languages, conforming implementations MUST support the Universal Character Set (UCS) Transformation Format -- 8 bit (UTF-8) [RFC3629] encoding of Unicode [UNICODE] [ISO10646] and the Unicode Format for Network Interchange [RFC5198].

Implementations of this specification SHOULD conform to the following standards on processing of human-readable Unicode text strings, see:

- Unicode Bidirectional Algorithm [UAX9] – left-to-right, right-to-left, and vertical
- Unicode Line Breaking Algorithm [UAX14] – character classes and wrapping
- Unicode Normalization Forms [UAX15] – especially NFC for [RFC5198]
- Unicode Text Segmentation [UAX29] – grapheme clusters, words, sentences
- Unicode Identifier and Pattern Syntax [UAX31] – identifier use and normalization
- Unicode Collation Algorithm [UTS10] – sorting
- Unicode Locale Data Markup Language [UTS35] – locale databases

Implementations of this specification are advised to also review the following informational documents on processing of human-readable Unicode text strings:

- Unicode Character Encoding Model [UTR17] – multi-layer character model
- Unicode in XML and other Markup Languages [UTR20] – XML usage
- Unicode Character Property Model [UTR23] – character properties
- Unicode Conformance Model [UTR33] – Unicode conformance basis

6. Security Considerations

6.1. Human-readable Strings

Implementations of this specification SHOULD conform to the following standard on processing of human-readable Unicode text strings, see:

- Unicode Security Mechanisms [UTS39] – detecting and avoiding security attacks

Implementations of this specification are advised to also review the following informational document on processing of human-readable Unicode text strings:

- Unicode Security FAQ [UNISECFAQ] – common Unicode security issues

6.2. Client Security Considerations

An IPP Client SHOULD follow these recommendations:

1. A Client SHOULD securely store at rest any personally identifiable information (PII) and authentication credentials such as passwords.

2. A Client SHOULD only respond to an authentication challenge over a secure connection (TLS) [RFC8010][RFC8011] unless TLS is not supported over that transport (e.g. IPP USB).

3. A Client SHOULD validate the identity of the Printer by whatever means are available for that connection type. If the connection is secured via TLS [RFC8010], the Client SHOULD validate the server's TLS certificate, match it to the originating host, cross-check it to match the host name or IP address in the IPP URI for the target Printer, and otherwise follow industry best practices for validating the Printer's identity using X.509 certificates over TLS [RFC6125]. If the connection is not secured via TLS, other means may be necessary to validate the Printer's identity.

4. A Client SHOULD provide a means to allow the User to examine a Printer's provided identity.

5. A Client SHOULD provide one or more means of notification when it is engaging with a previously encountered Printer whose identity has changed.

6. OAuth2 Considerations

    1. The recommendations in “Proof Key for Code Exchange by OAuth Public Clients” [RFC7636] SHOULD be followed, since the threats described therein has been observed in practice.

    2. The recommendations in “OAuth 2 for Native Apps” [RFC8252] should be followed if the print system provides its own user interface presentation and controls for handling the OAuth2 authentication steps, to mitigate the risks described therein.

6.3. Printer Security Considerations

An IPP Printer:

1. SHOULD securely store at rest any personally identifiable information (PII) and authentication credentials such as passwords that are local to the Printer.

2. SHOULD only challenge a Client for authentication over a secure connection (TLS) [RFC8010][RFC8011] unless TLS is not supported over that transport (e.g. IPP USB).

3. SHOULD support User-provisioned X.509 certificates:

    1. The certificate MUST persist across power cycles

    2. The certificate MUST NOT be automatically renewed or replaced

    3. The certificate SHOULD have a maximum expiration of 3 year from the date of issuance

    4. The certificate SHOULD NOT use MD5 or SHA-1 hashes

4. SHOULD support self-generated self-signed X.509 certificates:

    1. The certificate persists across power cycles

    2. The certificate has a minimum default expiration of 5 years from the date of issuance / generation

    3. The certificate is automatically renewed (regenerated), using a new private key if the previous certificate has expired

    4. The certificate is generated using the mDNS, DHCP and/or manually-configured DNS hostname(s) and regenerated whenever these change

    5. The Printer MUST be able to generate RSA certificates with a key length of 2048 bits using SHA-256 hash

    6. The Printer SHOULD be able to generate ECDSA certificates using the secp256r1(P-256), secp384r1 (P-384), or secp521r1 (P-521) curves and a SHA-256 hash.

    7. The Printer MUST NOT generate self-signed certificates using MD5 or SHA-1 hashes

7. References

7.1. Normative References

[IANA-HTTP-AUTH] Hypertext Transfer Protocol (HTTP) Authentication Scheme Registry, Internet Assigned Numbers Authority, https://www.iana.org/assignments/http-authschemes/http-authschemes.xml

[ISO10646] "Information technology -- Universal Coded Character Set (UCS)", ISO/IEC 10646:2011

[PWG5100.12] R. Bergman, H. Lewis, I. McDonald, M. Sweet, "IPP Version 2.0, 2.1, and 2.2", PWG 5100.12-2015, October 2015, http://ftp.pwg.org/pub/pwg/standards/std-ipp20-20151030-5100.12.pdf

[PWG5100.13] M. Sweet, I. McDonald, P. Zehler, "IPP: Job and Printer Extensions - Set 3 (JPS3)", PWG 5100.13-2012, July 2012, http://ftp.pwg.org/pub/pwg/candidates/cs-ippjobprinterext3v10-20120727-5100.13.pdf

[PWG5100.14] M. Sweet, I. McDonald, A. Mitchell, J. Hutchings, "IPP Everywhere", 5100.14-2013, January 2013, http://ftp.pwg.org/pub/pwg/candidates/cs-ippeve10-20130128-5100.14.pdf

[PWG5100.19] S. Kennedy, "IPP Implementor's Guide v2.0", PWG 5100.19-2015, August 2015, http://ftp.pwg.org/pub/pwg/candidates/cs-ippig20-20150821-5100.19.pdf

[PWG5100.SYSTEM] I. McDonald, M. Sweet, “IPP System Service v1.0”, PWG 5100.SYSTEM, TBD, https://ftp.pwg.org/pub/pwg/ipp/wd/wd-ippsystem10-20180502.pdf

[RFC2817] R. Khare, S. Lawrence, “Upgrading to TLS Within HTTP/1.1”, RFC 2817, May 2000, https://www.ietf.org/rfc/rfc2817.txt

[RFC3380] T. Hastings, R. Herriot, C. Kugler, H. Lewis, “Internet Printing Protocol (IPP): Job and Printer Set Operations”, RFC 3380, September 2002, https://www.ietf.org/rfc/rfc3380.txt

[RFC3629] F. Yergeau, “UTF-8, a transformation format of ISO 10646”, RFC 3629, November 2003, https://www.ietf.org/rfc/rfc3629.txt

[RFC4559] K. Jaganathan, L. Zhu, J. Brezak, “SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows”, RFC 4559, June 2006, https://www.ietf.org/rfc/rfc4559.txt

[RFC5198] J. Klensin, M. Padlipsky, “Unicode Format for Network Interchange”, RFC 5198, March 2008, https://www.ietf.org/rfc/rfc5198.txt

[RFC5246] T. Dierks, E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2”, August 2008, https://www.ietf.org/rfc/rfc5246.txt

[RFC6749] D. Hardt, Ed., “The OAuth 2.0 Authorization Framework”, RFC 6749, October 2012, https://www.ietf.org/rfc/rfc6749.txt

[RFC6750] M. Jones, D. Hardt, “The OAuth 2.0 Authorization Framework: Bearer Token Usage”, RFC 6750, October 2012, https://www.ietf.org/rfc/rfc6750.txt

[RFC7230] R. Fielding, J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014, https://www.ietf.org/rfc/rfc7230.txt

[RFC7616] R. Shekh-Yusef, D. Ahrens, S. Bremer, “HTTP Digest Access Authentication”, RFC 7616, September 2015, https://www.ietf.org/rfc/rfc7616.txt

[RFC7617] J. Reschke, “The 'Basic' HTTP Authentication Scheme”, RFC 7617, September 2015, https://www.ietf.org/rfc/rfc7617.txt

[RFC7636] N. Sakimura, Ed., J. Bradley, N. Agarwal, "Proof Key for Code Exchange by OAuth Public Clients", RFC 7636, September 2015, https://www.ietf.org/rfc/rfc7636.txt

[RFC8010] M. Sweet, I. McDonald, “Internet Printing Protocol/1.1: Encoding and Transport”, RFC 8010, January 2017, https://www.ietf.org/rfc/rfc8010.txt

[RFC8011] M. Sweet, I. McDonald, “Internet Printing Protocol/1.1: Model and Semantics”, RFC 8011, January 2017, https://www.ietf.org/rfc/rfc8011.txt

[RFC8252] W. Denniss, J. Bradley, “OAuth 2.0 for Native Apps”, RFC 8252, October 2017, https://www.ietf.org/rfc/rfc8252.txt

[UAX9] Unicode Consortium, “Unicode Bidirectional Algorithm”, UAX#9, May 2016, http://www.unicode.org/reports/tr9

[UAX14] Unicode Consortium, “Unicode Line Breaking Algorithm”, UAX#14, June 2016, http://www.unicode.org/reports/tr14

[UAX15] Unicode Consortium, “Normalization Forms”, UAX#15, February 2016, http://www.unicode.org/reports/tr15

[UAX29] Unicode Consortium, “Unicode Text Segmentation”, UAX#29, June 2016, http://www.unicode.org/reports/tr29

[UAX31] Unicode Consortium, “Unicode Identifier and Pattern Syntax”, UAX#31, May 2016, http://www.unicode.org/reports/tr31

[UNICODE] The Unicode Consortium, “Unicode® 10.0.0”, June 2017, http://unicode.org/versions/Unicode10.0.0/

[UTS10] Unicode Consortium, “Unicode Collation Algorithm”, UTS#10, May 2016, http://www.unicode.org/reports/tr10

[UTS35] Unicode Consortium, “Unicode Locale Data Markup Language”, UTS#35, October 2016, http://www.unicode.org/reports/tr35

[UTS39] Unicode Consortium, “Unicode Security Mechanisms”, UTS#39, June 2016, http://www.unicode.org/reports/tr39

7.2. Informative References

[IPPGUPA] S. Kennedy, "IPP Get-User-Printer-Attributes (GUPA)", December 2017, https://ftp.pwg.org/pub/pwg/ipp/registrations/reg-ippgupa-20171214.pdf

[IPPUSB] S. Kennedy, A. Mitchell, “USB Print Interface Class IPP Protocol Specification”, December 2012, http://www.usb.org/developers/docs/devclass_docs/IPP.zip

[RFC6125] P. Saint-Andre, J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011, https://www.ietf.org/rfc/rfc6125.txt

[UNISECFAQ] Unicode Consortium “Unicode Security FAQ”, November 2016, http://www.unicode.org/faq/security.html

[UTR17] Unicode Consortium “Unicode Character Encoding Model”, UTR#17, November 2008, http://www.unicode.org/reports/tr17

[UTR20] Unicode Consortium “Unicode in XML and other Markup Languages”, UTR#20, January 2013, http://www.unicode.org/reports/tr20

[UTR23] Unicode Consortium “Unicode Character Property Model”, UTR#23, May 2015, http://www.unicode.org/reports/tr23

[UTR33] Unicode Consortium “Unicode Conformance Model”, UTR#33, November 2008, http://www.unicode.org/reports/tr33

8. Authors' Addresses

Primary authors:

Smith Kennedy
HP Inc.
11311 Chinden Blvd.
Boise ID 83714
smith.kennedy@hp.com

Michael Sweet
Apple Inc.
One Apple Park Way
MS 111-HOMC
Cupertino, CA 95014
msweet@apple.com

The authors would also like to thank the following individuals for their contributions to this standard:

Ira McDonald – High North, Inc.

9. Change History

9.1. June 29, 2018

Updated as per feedback from PWG May 2018 F2F:

- Added line numbers
- Resolved typos in diagrams in figures 3.5, 3.6, and the “new” 3.7 (TLS)
- Removed the second OAuth2 diagram

- Rewrote the TLS client authentication scheme description (contributed by Mike Sweet) and re-titled the section for its corresponding “uri-authentication-supported” keyword ('certificate')

9.2. May 10, 2018

Updated figures 6 and 7 (relating to OAuth2) to add a note indicating where the Printer might be able to acquire a user identifier suitable for making policy choices. Also made a few minor editorial updates.

9.3. April 30, 2018

Changed to Apache OpenOffice template. Added Mike Sweet as a co-author since he has contributed a great deal of content to the document. Resolved all “to-do” highlighted areas and resolved issues identified in the February 2018 vF2F minutes (https://ftp.pwg.org/pub/pwg/ipp/minutes/ippv2-f2f-minutes-20180207.pdf):

- Added sequence diagram for X.509 client authentication
- Added sequence diagram for hybrid 'oauth' / 'digest' authentication
- Many other changes

9.4. January 23, 2018

Updated as per email feedback and discussion:

- Fixed some editorial issues with naming HTTP Basic, HTTP Digest, and HTTP Negotiate, and some names of sections.
- Added mention of “printer-xri-supported”.
- Added additional references.
- Added additional sub-sections to capture Client and Printer recommendations for appropriate behavior when authentication is unsuccessful since the negative cases can vary widely.

9.5. December 5, 2017

Updated as per feedback from the November 2017 PWG vF2F and subsequent work with IPP WG members on specific details:

- Corrected OAuth2 sequence diagram to more correctly describe the sequence of operations and actors involved in an OAuth2 authenticated IPP Printer scenario.

- Added Implementation Recommendations that were revealed during the course of correcting the OAuth2 sequence diagram.

9.6. August 3, 2017

Initial revision.