From: Carl Kugler [kugler@us.ibm.com]
Sent: Tuesday, January 16, 2001 14:32
To: ipp@pwg.org
Subject: IPP> Bakeoff issues 3.1 and 3.2

<<<<<<<<<<<<

Issue 3.1: AGREED
     IPP Client failed when an unexpected HTTP "100 continue" was received.
     Some printers sent a "100 continue" even before the Client sent a
     request.

Issue 3.2: OPEN
     Some IPP Clients issues a zero length HTTP Post. The Client assumed
     that this would force a challenge if security is enabled on the
     Printer. The Client would have a problem if a subsequent print
     operation were challenged.

>>>>>>>>>>>>>>

It occurs to me that these two issues are related, and that issue 3.1
          contains the solution to issue 3.2.

The crux of the 3.2 problem is this: for Digest Authentication, the client
wants to provoke a challenge so it can get the "nonce" it needs in order to
form the authentication-info for a request. It wants to get this challenge
BEFORE it sends the document data to the printer; otherwise, the request
will be rejected (Unauthorized) and will have to be resent with
authentication-info.

This is exactly the type of problem that the "100-Continue" mechanism is
designed to solve! If a request includes an "Expect: 100-Continue"
header, the Printer MUST either respond with 100 (Continue) status and
continue to read from the input stream, or reject the request with a final
status code. The Printer MUST NOT wait for the request body before sending
the 100 (Continue) response.

Problem 3.2 is solved if a client sends an HTTP request containing the
"Expect: 100-Continue" header and waits for a 100 (Continue) response
before sending the request body. When a request includes the 100-Continue
expectation, and security is enabled on a Printer, the Printer will respond
with 401 (Unauthorized) and include a WWW-Authenticate header containing
the challenge, instead of sending 100 (Continue). This response MUST be
sent after the Printer processes the HTTP headers, without waiting for the
request body. At this point, the client can form the appropriate
WWW-Authenticate request-header, and retry the request. This time it
should receive 100 (Continue), indicating it should proceed to send the
request body.

So the client has successfully provoked a challenge BEFORE sending its
Print-Job request, using only standard mechanisms that already are
required.

-Carl

-----Original Message-----
From: Carl Kugler [mailto:kugler@us.ibm.com]
Sent: Thursday, December 14, 2000 09:39
To: ipp@pwg.org
Subject: Re: IPP> BakeOff3 Issue 3.2 - Do URLs have to be different if the security is different?

--- Tom wrote:
> At the IPP WG meeting, we agreed to resolution 2 for Issue 3.2. However, on
> the IPP telecon today, Ira pointed out that HTTP security is
> connection-based, not transaction-based.
> There is a new experimental RFC
> 2660 for SHTTP (August 1999), which has transaction-based security, but we
> don't want IPP to have to use that.
>
> So resolution 2 won't work; the challenge has to be issued for the
> connection, not on an operation-by-operation basis. Therefore, each
> different security regime that a Printer supports MUST have a distinct URL.
> What about authentication?
>

This seems overly general to me. By "HTTP security" are you refering to Digest authentication, TLS, Kerberos, or what?

You seem to be implying that each operation requires a separate connection. That is not the normal case for HTTP/1.1: all connections in HTTP/1.1 are persistent by default. Also, Basic and Digest authentication can work over non-persistent connections (they worked for HTTP/1.0, didn't they?).

AFAIK, a transaction is a series of operations that succeeds or fails as a unit, with the properties of atomicity, consistency, isolation and durability. Is this a new requirement for IPP?

-Carl

-----Original Message-----
From: Hastings, Tom N [mailto:hastings@cp10.es.xerox.com]
Sent: Wednesday, December 13, 2000 17:54
To: ipp (E-mail)

Subject: IPP> BakeOff3 Issue 3.2 - Do URLs have to be different if the security is different?

At the IPP WG meeting, we agreed to resolution 2 for Issue 3.2. However, on the IPP telecon today, Ira pointed out that HTTP security is connection-based, not transaction-based. There is a new experimental RFC 2660 for SHTTP (August 1999), which has transaction-based security, but we don't want IPP to have to use that.

So resolution 2 won't work; the challenge has to be issued for the connection, not on an operation-by-operation basis. Therefore, each different security regime that a Printer supports MUST have a distinct URL. What about authentication?

As to whether sending a zero length HTTP Post (also ISSUE 3.2) and being guaranteed that the server will always issue the challenge (if the URL is one that supports security that challenges), needs further work.

NEW ISSUE: The "Job and Printer Set Operation" specification has two different security regimes with the same URL. See the extracted text following this issue text. What to do about that?

Issue 3.2: OPEN

Some IPP Clients issues a zero length HTTP Post. The Client assumed that this would force a challenge if security is enabled on the Printer. The Client would have a problem if a subsequent print operation were challenged.

Proposed Resolutions:

There are two competing resolutions.

Resolution 1 is that a challenge should be issued whenever an HTTP operation is received on a particular URL. (assuming the URL is part of an authentication space) The client must accept and respond to a challenge the first time a URL is accessed.

Resolution 2 allows the vendor to determine when a challenge is issued. The vendor is free to use the contents of the HTTP request to determine if the operation mandates a challenge. The client must accept and respond to a challenge at any time.

The Client should use the IPP operation "validate-job" to check if a job will be accepted. This operation will cause the Printer to issue a challenge and check the print request before sending the data. The IPP Client should also be able to handle a challenge when issuing an IPP operation since there is no guarantee the connection has not been torn down.

Furthermore, a Printer should accept an empty HTTP post and issue a challenge based on the URL of the post.

Resolution 1:

From Bob Herriot:

I raised the issue about whether a Printer should perform the authentication

challenge based solely on the URL or whether it could react differently to

an empty request than to a Validate-Job request.

I asked an HTTP expert and received the following information.

1) An HTTP server can have any policy.
This means that resolution 2 is allowable.
2) It is best for a client if it can associate the URL tree with the authentication space.
This means that our decision could be better. That is, we should require an IPP Printer to decide whether to issue an authentication challenge by examining the URL and nothing else, e.g. a Printer receiving a request for a particular URL, gives the same challenge to an empty request as to a Validate-Job request.
This solution allows a client to use Validate-Job to request a challenge as we decided to allow. It also allows a client to use the empty request.
The important difference between our decision and what I am proposing is that the Printer must perform an authentication challenge consistently for a URL regardless of the contents of the message body. This rule make IPP behavior consistent with good HTTP policy.

Resolution 2:
From Peter Zehler:
Allowing IPP Printers to use the contents of an IPP request to determine if a challenge should be issued allows for increased usability. The client does not have to keep track of multiple instances of the same printer and select the appropriate one based on the operation to be performed. The printer is free to determine when authentication is required. This allows the client to use a single URL and authenticate himself when the printer places restrictions on operations or features.
This resolution does not prohibit challenges based statically on a URL. Resolution 2 does require a client to be ready at any time to receive a challenge. This should be done anyway since the client application may be unaware that an HTTP connection has dropped after authenticating the connection, resulting in a new challenge. Some HTTP servers have security realms that apply only to a transaction as well as being connection based.

From the Job and Print Set spec:

```
  "printer-xri-supported =
      {  "xri-uri" = ipp://abc.com/p1
         "xri-authentication" = basic, digest
         "xri-security" = tls
      },
      {  "xri-uri" = http://abc.com/pq
         "xri-authentication" = none
         "xri-security" = none
      }
```

would cause the Printer to set the three corresponding IPP/1.1 READ-ONLY attributes, each with three parallel values as follows:

```
    "printer-uri-supported" = { ipp://abc.com/p1, ipp://abc.com/p1,
                                http://abc.com/pq }
    "uri-authentication-supported" = { basic, digest, none }
    "uri-security-supported" = { tls, tls, none }
```

Because there were two authentication values for the ipp://abc.com/p1 URL, that URL value is repeated. Had the ipp URL had 2 authentication values and 3 security values, then there would have been 7 (2*3 + 1) parallel values for each of the three attributes, 6 with the same ipp URI and 1 with the http URI.