RPM Support - Issue #9224

Pulp 3.14 - can't sync a repository because of a checksum

08/10/2021 03:38 PM - ares

Status:	CLOSED - NOTABUG	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	dalley	**Estimated time:**	0:00 hour
Category:			
Sprint/Milestone:			
Severity:	3. High	**Groomed:**	No
Version:		**Sprint Candidate:**	No
Platform Release:		**Tags:**	Katello
OS:	RHEL 7	**Sprint:**	Sprint 102
Triaged:	No	**Quarter:**	

Description

I'm trying to sync a yum repo in Katello with pulp-rpm 3.14.0. Syncing shows the following error

```
A file located at the url https://ftp.redhat.com/redhat/convert2rhel/7/os/repodata/842112544c39a31
f7481ca043705a424d6f2e50cd589f56ba33999d1e81da2a7-primary.xml.gz failed validation due to checksum
.
```

The repo URL I'm trying to sync - http://ftp.redhat.com/redhat/convert2rhel/7/os/

The repo metadata seems correct, I checked all the checksums. It works fine on older Katello system using Pulp2 and I don't have a proof right now but I believe it worked with Pulp 3.12. This is kind of a blocker for me.

History

#1 - 08/10/2021 03:57 PM - jsherril@redhat.com

I added some debugging and it looks like pulp is mixing up the open checksum and the checksum values:

```
    def validate_digests(self):
        """
        Validate all digests validate if ``expected_digests`` is set

        Raises:
            :class:`~pulpcore.exceptions.DigestValidationError`: When any of the ``expected_digest``
                values don't match the digest of the data passed to
                :meth:`~pulpcore.plugin.download.BaseDownloader.handle_data`.
        """
        if self.expected_digests:
            for algorithm, expected_digest in self.expected_digests.items():
                if expected_digest != self._digests[algorithm].hexdigest():
                    raise DigestValidationError(str(self.url) + " == " + str(self._digests[algorithm].hexdiges
t()) + " == " + str(expected_digest) )
```

and the error i got was:

A file located at the url
http://ftp.redhat.com/redhat/convert2rhel/7/os/repodata/28ffc65b7ccaf7f8bdc280e0e0a3055950f51ff38517bf2fbdf44ca301363d5e-filelists.xml.gz ==
5e32e2d2d238845433993bf3965df7924c02ad561748fe86538e0fb86dc71fd3 ==
28ffc65b7ccaf7f8bdc280e0e0a3055950f51ff38517bf2fbdf44ca301363d5e failed validation due to checksum.

So it seems like its expecting the open digest, but getting the gzipd digest. Why is this happening only for this repo? I have no idea

#2 - 08/10/2021 04:04 PM - ttereshc

- Tags Katello added

#3 - 08/10/2021 08:31 PM - dalley

- Status changed from NEW to ASSIGNED

- Assignee set to dalley

- Sprint set to Sprint 102

#4 - 08/12/2021 02:13 AM - dalley

The problem is that the http client sees "Content-Encoding: x-gzip" and interprets that as the server having applied a gzip encoding, which it then reverses by de-compressing the content transparently.

```
< HTTP/1.1 200 OK

< Date: Wed, 11 Aug 2021 23:26:54 GMT
< Server: Apache
< Last-Modified: Tue, 13 Jul 2021 07:24:09 GMT

< ETag: "830f546f-479-5c6fc1fac5840"
< Accept-Ranges: bytes
< Content-Length: 1145

< Cache-Control: max-age=3600
< Expires: Thu, 12 Aug 2021 00:26:54 GMT
< Connection: close

< Content-Type: application/x-gzip

< Content-Encoding: x-gzip
```

From what I can tell, this means the web server is serving these files is doing so incorrectly.

> Content-Encoding is used solely to specify any additional encoding done by the server before the content was transmitted to the client. Although the HTTP RFC outlines these rules pretty clearly, some web sites respond with "gzip" as the Content-Encoding even though the server has not gzipped the content.
>
> Our testing has shown this problem to be limited to some sites that serve Unix/Linux style "tarball" files. Tarballs are gzip compressed archives files. By setting the Content-Encoding header to "gzip" on a tarball, the server is specifying that it has additionally gzipped the gzipped file.

https://docs.microsoft.com/en-us/archive/blogs/wndp/content-encoding-content-type

According to MDN:

> The Content-Encoding representation header lists any encodings that have been applied to the representation (message payload), and in what order. This lets the recipient know how to decode the representation in order to obtain the original payload format. Content encoding is mainly used to compress the message data without losing information about the origin media type.
>
> Note that the original media/content type is specified in the Content-Type header, and that the Content-Encoding applies to the representation, or "coded form", of the data. If the original media is encoded in some way (e.g. a zip file) then this information would not be included in the Content-Encoding header.

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Encoding

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Type

Since the metadata was already in gzip format before being served, the web server marking the content-encoding as gzip is not standard-compliant.

#5 - 08/12/2021 02:27 AM - dalley

- Status changed from ASSIGNED to CLOSED - NOTABUG

#6 - 08/12/2021 02:28 AM - dalley

Closing this as not a bug, because Pulp is doing the appropriate thing... we will however try to get into contact with the owners of that FTP server.

The reason this only started failing in 3.13+ is because the metadata library transparently handles both compressed and noncompressed files, so it didn't matter that they were being transparently decompressed by the web client, and because we only recently began checking the checksums of the downloaded metadata files. I don't think we should *ignore* those checksums, so there's not much we can do from the client side to work around this.

Changing the decompression behavior is a no-go as it would break other things. We've tried that before: https://pulp.plan.io/issues/3907

#7 - 08/12/2021 03:35 AM - dalley

Compare to the response of the fedora servers

```
< HTTP/1.1 200 OK
< Date: Thu, 12 Aug 2021 01:32:47 GMT
< Server: Apache

< X-Frame-Options: DENY
< X-Xss-Protection: 1; mode=block

< X-Content-Type-Options: nosniff

< Referrer-Policy: same-origin
< Content-Security-Policy: default-src 'none'; img-src 'self'

< Strict-Transport-Security: max-age=31536000; preload
< Last-Modified: Tue, 25 May 2021 01:46:45 GMT

< ETag: "64732f-5c31db2d300dd"

< Accept-Ranges: bytes
< Content-Length: 6583087

< AppTime: D=2561
< X-Fedora-AppServer: dl01.iad2.fedoraproject.org
< Content-Type: application/x-gzip
```

No Content-Encoding header

And CentOS

```
< HTTP/1.1 200 OK
< Date: Thu, 12 Aug 2021 01:34:33 GMT
< Server: Apache/2.4.6 (CentOS)
< X-Xss-Protection: 1; mode=block
< X-Content-Type-Options: nosniff
< Referrer-Policy: same-origin
< X-Frame-Options: SAMEORIGIN
< Last-Modified: Wed, 11 Aug 2021 18:57:22 GMT
< ETag: "47bc81-5c94d303613b0"
< Accept-Ranges: bytes
< Content-Length: 4701313
< Content-Type: application/x-gzip
```

No Content-Encoding header

#8 - 08/23/2021 05:22 PM - dalley

This is now fixed by a configuration change to the webserver hosting this repository.