RPM Support - Story #1912

Add MD5 checksum element to data type elements in repomd.xml file

05/12/2016 03:33 PM - robnester

Status:	CLOSED - WONTFIX	**Start date:**	
Priority:	High	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:			
Platform Release:		**Tags:**	Pulp 2
Groomed:	No	**Sprint:**	
Sprint Candidate:	No	**Quarter:**	

Description

As part of downstream content validation testing, it would be beneficial to be able to have the MD5 checksum of generated metadata files present in their respective entries within the repomd.xml file.

Currently there are values for checksum and open-checksum. While these checksums algorithms can be set to sha1/sha256, I'd like to see another value (I have no naming requirements) which holds an MD5 checksum of the file. See the following examples:

An example from repomd.xml as it is today:

```
<data type="other_db">
  <location href="repodata/25cd804ef053f6b3693a8ec0e6dc711db7af4bd4-other.sqlite.bz2"/>
  <checksum type="sha">25cd804ef053f6b3693a8ec0e6dc711db7af4bd4</checksum>
  <timestamp>1462981265.0</timestamp>
  <size>5849902</size>
  <open-size>50198528</open-size>
  <open-checksum type="sha">3da280326cb00e28d14ffd40a6c752a9c91867c7</open-checksum>
  <database_version>10</database_version>
</data>
```

and an example of what i'd like to see, note I'm calling the desired value "validation-checksum" with a type of MD5 here, but the value name can be arbitrary in my opinion.

```
<data type="other_db">
  <location href="repodata/25cd804ef053f6b3693a8ec0e6dc711db7af4bd4-other.sqlite.bz2"/>
  <checksum type="sha">25cd804ef053f6b3693a8ec0e6dc711db7af4bd4</checksum>
  <validation-checksum type="md5">(md5 checksum here)</checksum>
  <timestamp>1462981265.0</timestamp>
  <size>5849902</size>
  <open-size>50198528</open-size>
  <open-checksum type="sha">3da280326cb00e28d14ffd40a6c752a9c91867c7</open-checksum>
  <database_version>10</database_version>
</data>
```

History

#1 - 05/13/2016 05:27 PM - dkliban@redhat.com

- Tracker changed from Issue to Story

- Priority changed from Normal to High

- Groomed set to No

- Sprint Candidate set to No

#2 - 05/13/2016 05:43 PM - mhrivnak

If schema changes are desired in repomd.xml, it would be better to start the conversation with the createrepo_c project. They are the defacto owners of that schema. If they agree to a change that meets your needs, we could then make sure it also gets supported in pulp. But we would not want to add non-standard data to that file without buy-in from them.

Or it's possible that multiple <checksum> elements are allowed, in which case we could add one that has md5. That would similarly need to be

confirmed with the createrepo_c project.

As an alternative, maybe there is a simpler way to get you that information. It's common to put a file named something like "MD5_SUMS" in a web directory that is the output of "md5sum *", and which can be fed back in for verification with "md5sum -c MD5_SUMS". Would something like that meet your needs?

Generally, it would be helpful to know more about why this is a requirement for you. What are you trying to accomplish, and what does the workflow look like end-to-end?

#4 - 05/13/2016 06:16 PM - jcline@redhat.com

- Priority changed from High to Normal

https://github.com/rpm-software-management/yum/blob/master/docs/repomd.dtd is the repomd schema, as far as I can tell. We should definitely talk with the rpm folks, but based on the DTD, it looks like having a ``data`` element with multiple ``checksum`` or ``open-checksum`` child elements is okay. Of course, just because the schema allows it doesn't mean it's what was intended or that clients will deal with it properly.

#5 - 05/13/2016 06:16 PM - jcline@redhat.com

- Priority changed from Normal to High

#6 - 05/13/2016 06:29 PM - robnester

mhrivnak wrote:

> If schema changes are desired in repomd.xml, it would be better to start the conversation with the createrepo_c project. They are the defacto owners of that schema. If they agree to a change that meets your needs, we could then make sure it also gets supported in pulp. But we would not want to add non-standard data to that file without buy-in from them.
>
> Or it's possible that multiple <checksum> elements are allowed, in which case we could add one that has md5. That would similarly need to be confirmed with the createrepo_c project.

I've opened the following with the createrepo_c project: https://github.com/rpm-software-management/createrepo_c/issues/60

> As an alternative, maybe there is a simpler way to get you that information. It's common to put a file named something like "MD5_SUMS" in a web directory that is the output of "md5sum *", and which can be fed back in for verification with "md5sum -c MD5_SUMS". Would something like that meet your needs?
>
> Generally, it would be helpful to know more about why this is a requirement for you. What are you trying to accomplish, and what does the workflow look like end-to-end?

I'm responsible for content validation on the CDN.
The CDN writes an MD5 of a given file in the etag header when making a request for the file.
Currently in order to ensure that the repo metadata files are complete we parse repomd.xml and extract the sha1/sha256 checksums from the file's respective entry.
We then download the file, and perform a corresponding checksum to compare.
Some of the repo metadata files can be large and it would be beneficial to have the "expected" md5 of the file present in the repomd.xml file (or, alternatively, another file in the repodata directory) which can be parsed and compared to the md5 of the file "on-disk" on the CDN.

#7 - 05/20/2016 04:19 PM - robnester

I opened https://github.com/rpm-software-management/createrepo_c/issues/60 for the createrepo_c, which was closed indicating that additional entries in the schema wouldn't affect createrepo_c. It wasn't clear how other clients would interpret non-required elements in the schema. So this, I think, brings us back to how pulp will proceed?

#8 - 01/06/2017 05:28 PM - mhrivnak

There are two challenges I see to this approach to content validation.

1. Does the CDN provider guarantee that the etag value will be the md5 hash of the file? Is that guarantee documented? The etag header is not required by the HTTP spec to use any particular hash algorithm, or even use a hash at all. So unless the CDN provider has committed to this as part of their API spec, they are free to change any time they want.

https://tools.ietf.org/html/rfc7232#section-2.3

2. This approach won't work for other kinds of data on the CDN. If we assume that the answer to challenge number 1 above is "Yes, they guarantee this", then I think storing the md5 hash of each file somewhere outside of repomd.xml is appropriate. I don't know what your validation tool has access to, but there are lots of options. A text file could be uploaded in the same directory, as described in comment #2. Or whatever uploads files to the CDN provider could calculate an md5 hash on the spot and compare it with what the CDN provider returns post-upload. Or those hashes could be stored in some other database for later use by your tool. Or ....

And as we keep adding more types of content to the CDN, it would be quite valuable to have one way of validating files that doesn't depend on parsing type-specific metadata and requiring the use of a particular hash algorithm.

Perhaps the CDN provider themselves can offer suggestions on verifying content in a general-purpose way.

So given those two problems, plus the inherent risk of doing anything unconventional in yum repo metadata, I don't see much upside to doing this work in pulp. But I am very happy to be convinced otherwise if we can address all of these concerns. Let me know if it would be helpful to setup a live discussion on this topic.

#9 - 01/06/2017 06:02 PM - robnester

mhrivnak wrote:

> So given those two problems, plus the inherent risk of doing anything unconventional in yum repo metadata, I don't see much upside to doing this work in pulp. But I am very happy to be convinced otherwise if we can address all of these concerns. Let me know if it would be helpful to setup a live discussion on this topic.

I think that it might be beneficial to setup a live discussion sometime in the near future to discuss the challenges and options.

#10 - 04/12/2019 10:17 PM - bmbouter

- Status changed from NEW to CLOSED - WONTFIX

Pulp 2 is approaching maintenance mode, and this Pulp 2 ticket is not being actively worked on. As such, it is being closed as WONTFIX. Pulp 2 is still accepting contributions though, so if you want to contribute a fix for this ticket, please reopen or comment on it. If you don't have permissions to reopen this ticket, or you want to discuss an issue, please reach out via the developer mailing list.

#11 - 04/15/2019 10:30 PM - bmbouter

- Tags Pulp 2 added