Resolving SoftwareUpdate metadata validation

Summary

Updating contents of an object in Configuration Manager (CM) usually results in a version change (or a new object is created). However with SoftwareUpdates, the contents could be updated without any change in its ID or version (which always remains at 1). Such changes to content without versioning would cause issues in peer-to-peer (P2P) sharing, because peers could serve stale content. To cope with changes like this, an additional validation has been introduced for software updates, ensuring clients download the latest content.

This additional validation ensures Nomad downloads content as the CM client expects. With software updates the metadata of its files is available in the CM client's WMI, and is compared against the details in LSZ. If there is a mismatch the client tells the peers to delete the stale content from their cache and the client retries the download. If the client was downloading from a DP, a message is sent to Nomad on that DP and the LSZ is recreated is using the latest content. Once a client has downloaded the LSZ with the expected metadata, the rest of the download logic continues as expected.

How it works

File metadata is obtained from the 'DownloadInfoEx2' class in CM's WMI and as a part of the validation process, the name and hash of every file is compared against those in the LSZ. In the following example, the hashes are SHA1.

> ⚠ The file hashes currently in the LSZ are of SHA2. LSZ generation is updated to put in SHA1 hashes for software updates.

Sample metadata in WMI:

```
<Content ContentId="c54879c8-215e-413f-afbf-a701afa2d637" Version="1">
<FileContent Name="18362.30.190401-1528.19h1_release_svc_refresh_CLIENTBUSINESS_VOL_x86FRE_en-us.esd" Hash="
1F01343ACDE39AE567FF520F9FC51DC2F9507305" HashAlgorithm="SHA1" Size="2345415546"/> <FileContent Name="
WindowsUpdateBox.exe" Hash="A215D07C5B28BA2270D9E2F54FB51C1010777762" HashAlgorithm="SHA1" Size="19524440"/>
</Content>
```

If the metadata in the LSZ does not match, the LSZ is deleted and the download is retried and downloading of stale LSZ is avoided in the next attempt. If the client is downloading from a Distribution Point (DP), an LSZGen request is sent with a force option, so the existing LSZ file on the DP is deleted and generated again using the latest content, this means that any changes to content after the initial LSZ generation is accounted for.

If the client was downloading from a peer, then a "Validate Metadata" message is broadcaste to peers. When peers receve the message, clients will validate the LSZ metadata in their cache using the 'MetadataHash' included in the message. If there is no match, clients delete the entire content from their cache. This means that when the original download is retried, any clients with stale content will not take part in elections. If there are any SSD /LocalSSD peers included, they are informed individually with a similar message.

The 'MetadataHash' used for validation is a hash of details of every file in the content. It is an SHA2 hash, calculated using Name&Hash of each file sorted alphabetically using their hashes (there is no specific reason for choosing hashes as the sort criteria, using the name also would work the same way).

Configuration

These changes apply when the `0x40000000` flag is enabled in CompatibilityFlags.

> ⚠ This bit must be added to CompatibilityFlags for this to work, it is not set by default.

Before applying the `0x40000000` flag, consider that:

- This validation may not work for updates deployed before this fix, because file hashes in LSZ for existing packages would be of SHA2, which is different from CM client's metadata
- Because older clients will ignore the validation msg, the fix should be deployed on all clients
- Clients will not take explicit action (For example, to disqualify) if a peer does not act on the validation msg, (But it may happen automatically, due to the resilience in an existing P2P implementation).