rgw - Bug #38638

S3 policy evaluated incorrectly

03/08/2019 10:48 AM - Davide Dal Bianco

Status:	Resolved	**% Done:**	0%
Priority:	High	**Spent time:**	0.00 hour
Assignee:	Pritha Srivastava		
Category:			
Target version:			
Source:		**Affected Versions:**	
Tags:		**ceph-qa-suite:**	
Backport:	luminous mimic nautilus	**Pull request ID:**	27309
Regression:	No	**Crash signature (v1):**	
Severity:	3 - minor	**Crash signature (v2):**	
Reviewed:			

Description

Hi,

I noticed a bug when accessing Ceph via Hadoop. I am using some shared buckets with read/write access for all users. Here is the policy for the bucket:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAll",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::<bucket>/*",
                "arn:aws:s3:::<bucket>"
            ]
        }
    ]
}
```

However, if a user different from the owner (or even an anonymous user) does a GetObject/HeadObject on a non existing object, Radosgw returns status code 403 which makes the Hadoop write fail.

From the official S3 documentation:

If a requested object doesn't exist in the bucket and the requester doesn't have s3:ListBucket access, then the requester receives an HTTP 403 (Access Denied) error rather than the HTTP 404 (Not Found) error.

I tried in AWS and a bucket with the same policy returns 404, which should be the correct behaviour since ListBucket is allowed.

Related issues:

Copied to rgw - Backport #39272: luminous: rgw: S3 policy evaluated incorrectly	**Resolved**
Copied to rgw - Backport #39273: nautilus: S3 policy evaluated incorrectly	**Resolved**
Copied to rgw - Backport #39274: mimic: S3 policy evaluated incorrectly	**Resolved**

History

#1 - 03/09/2019 01:27 AM - Brad Hubbard

- Project changed from Ceph to rgw

#2 - 03/14/2019 05:49 PM - Matt Benjamin

- Status changed from New to In Progress

- Assignee set to Pritha Srivastava

pritha, could you take a look?

Matt

#3 - 03/14/2019 05:49 PM - Matt Benjamin

- Priority changed from Normal to High

#4 - 03/15/2019 10:37 AM - Pritha Srivastava

This error can also be reproduced by attaching the above policy to a bucket, and then a user (not bucket owner)trying to get a non-existent object from the bucket. This happens even before the bucket policy rule is evaluated. It fails while reading permissions. (I think a non owner isn't allowed to Read permissions for the bucket). I need to dig deeper and will update when I find the reason.

#5 - 03/19/2019 04:32 AM - Pritha Srivastava

I checked further and can see that while evaluating permissions, if the object is not found, the bucket acls are checked to see if there is any read acl set for the non owner user, and if it is not, then AccessDenied is returned. I also tried by setting a read acl on the bucket, and in that case the correct error 404 is returned. So we may need to modify code to return 404 when the object is not found, irrespective of whether a read acl is set on the bucket (for the non owner user) or not.

#6 - 03/27/2019 08:52 AM - Pritha Srivastava

Pritha Srivastava wrote:

> I checked further and can see that while evaluating permissions, if the object is not found, the bucket acls are checked to see if there is any read acl set for the non owner user, and if it is not, then AccessDenied is returned. I also tried by setting a read acl on the bucket, and in that case the correct error 404 is returned. So we may need to modify code to return 404 when the object is not found, irrespective of whether a read acl is set on the bucket (for the non owner user) or not.

On second thoughts,

RGW behaves correctly as far as acl evaluation is concerned and returns the correct error code (that is 404 when acl corresponding to ListBucket is set and 403 when that acl is not set), the correct way to solve this would be evaluate bucket policies as well after evaluating the ACLs.

#7 - 03/28/2019 05:38 PM - Adam Emerson

We could add a special case to the permission check to return 404 if the object doesn't exist?

#8 - 03/29/2019 04:08 AM - Pritha Srivastava

Adam Emerson wrote:

> We could add a special case to the permission check to return 404 if the object doesn't exist?

Adam, this error is returned while reading the permissions while building object policies (before perm evaluation), it already has a special case for non-existent objects, where it checks for acls and if the correct acl is present on the bucket, then 404 is returned else 403 is returned (which is correct according to AWS spec). We need to evaluate bucket policies also in this place to be able to return the correct error code (may be just call the correct verify_permission method which takes both acls and bucket policies into account)

#9 - 04/02/2019 08:32 AM - Pritha Srivastava

https://github.com/ceph/ceph/pull/27309

#10 - 04/02/2019 04:18 PM - Casey Bodley

- Status changed from In Progress to 7

- Backport set to luminous mimic nautilus

- Pull request ID set to 27309

#11 - 04/11/2019 05:31 PM - Casey Bodley

- Status changed from 7 to Pending Backport

#12 - 04/12/2019 12:04 PM - Nathan Cutler

- Copied to Backport #39272: luminous: rgw: S3 policy evaluated incorrectly added

#13 - 04/12/2019 12:04 PM - Nathan Cutler

- Copied to Backport #39273: nautilus: S3 policy evaluated incorrectly added

#14 - 04/12/2019 12:04 PM - Nathan Cutler

- Copied to Backport #39274: mimic: S3 policy evaluated incorrectly added

#15 - 10/11/2019 09:20 AM - Nathan Cutler

- Status changed from Pending Backport to Resolved

While running with --resolve-parent, the script "backport-create-issue" noticed that all backports of this issue are in status "Resolved" or "Rejected".