Bug 2105419 (CVE-2022-2447) - CVE-2022-2447 Openstack: Application credential token remains valid longer than expected

Keywords: Security ×

Status: NEW

Alias: CVE-2022-2447

Product: Security Response

Component: vulnerability

Version: unspecified

Hardware: All

OS: Linux

Priority: medium

Severity: medium

Target Milestone: ---

Assignee: Red Hat Product Security

QA Contact:

Docs Contact:

URL:

Whiteboard:

Depends On: 2117920 2117923 2117924 2120165 2120167

Blocks: 2105420

TreeView+ depends on / blocked

Reported: 2022-07-08 18:35 UTC by amctagga

Modified: 2022-10-28 13:12 UTC (History)

CC List: 74 users (show)

Fixed In Version:

Doc Type: If docs needed, set a value

Doc Text: A flaw was found in Keystone. There is a time lag (up to one hour in a default configuration) between when security policy says a token should be revoked from when it is actually revoked. This could allow a remote administrator to secretly maintain access for longer than expected.

Clone Of:

Environment:

Last Closed:

Attachments	(Terms of Use)
Add an attachment (proposed patch, testcase, etc.)	

Links

System	ID	Private	Priority	Status	Summary	Last Updated
Launchpad.net	ossa/+bug/1992183	0	None	None	None	2022-10-07 15:36:25 UTC

amctagga 2022-07-08 18:35:59 UTC Description

Description of problem:
Keystone issues tokens with the default lifespan regardless of the lifespan of the application credentials used to issue

them.
If the configured lifespan of an identity token is set to be 1h, and the application credentials expire in 1 minute from now, a newly issued token will outlive the application credentials used to issue it by 59 minutes.

How reproducible: 100%

Steps to Reproduce:
1. Create application credentials with short expiration time (e.g. 10 seconds)
2. openstack token issue
--> the returned token has standard expiration, for example 1 hour. The script below confirms that the token continue being valid after the application credentials expired.

```bash
#!/usr/bin/env bash

set -Eeuo pipefail

openstack image create --disk-format=raw --container-format=bare --file <(echo 'I am a Glance image') testimage -f json > image.json

image_url="$(openstack catalog show glance -f json | jq -r '.endpoints[] | select(.interface=="public").url')$(jq -r '.file' image.json)"

openstack application credential create \
        --expiration="$(date --utc --date '+10 second' +%Y-%m-%dT%H:%M:%S)" \
        token_test \
        -f json \
        > appcreds.json

cat <<EOF > clouds.yaml
clouds:
    ${OS_CLOUD}:
        auth:
            auth_url: <auth_url>
            application_credential_id: '$(jq -r '.id' appcreds.json)'
            application_credential_secret: '$(jq -r '.secret' appcreds.json)'
        auth_type: "v3applicationcredential"
        identity_api_version: 3
        interface: public
        region_name: <region_name>
EOF
# Override ~/.config/openstack/secure.yaml
touch secure.yaml

openstack token issue -f json > token.json

echo "appcreds expiration: $(jq -r '.expires_at' appcreds.json)"
for i in {1..10}; do
        sleep 100
        echo -ne "$(date --utc --rfc-3339=seconds)\t"
        curl -isS -H "X-Auth-Token: $(jq -r '.id' token.json)" --url "$image_url" | head -n1
done

```

```
Actual results (on a cloud with tokens duration of 24h):
appcreds expiration: 2022-07-08T13:55:02.000000
2022-07-08 13:56:38+00:00       HTTP/1.1 200 OK
2022-07-08 13:58:19+00:00       HTTP/1.1 200 OK
2022-07-08 14:00:00+00:00       HTTP/1.1 200 OK
2022-07-08 14:01:42+00:00       HTTP/1.1 200 OK
2022-07-08 14:03:23+00:00       HTTP/1.1 200 OK
2022-07-08 14:05:07+00:00       HTTP/1.1 200 OK
2022-07-08 14:06:49+00:00       HTTP/1.1 200 OK
2022-07-08 14:08:37+00:00       HTTP/1.1 200 OK
2022-07-08 14:10:18+00:00       HTTP/1.1 200 OK
2022-07-08 14:12:00+00:00       HTTP/1.1 200 OK

Expected results:
appcreds expiration: 2022-07-08T13:55:02.000000
2022-07-08 13:54:38+00:00       HTTP/1.1 200 OK
2022-07-08 13:58:19+00:00       HTTP/1.1 401 Unauthorized
2022-07-08 14:00:00+00:00       HTTP/1.1 401 Unauthorized
2022-07-08 14:01:42+00:00       HTTP/1.1 401 Unauthorized
2022-07-08 14:03:23+00:00       HTTP/1.1 401 Unauthorized
2022-07-08 14:05:07+00:00       HTTP/1.1 401 Unauthorized
2022-07-08 14:06:49+00:00       HTTP/1.1 401 Unauthorized
2022-07-08 14:08:37+00:00       HTTP/1.1 401 Unauthorized
2022-07-08 14:10:18+00:00       HTTP/1.1 401 Unauthorized
2022-07-08 14:12:00+00:00       HTTP/1.1 401 Unauthorized
```

Luigi Toscano 2022-07-08 19:29:17 UTC Comment 1

Which RHOSP version? And which keystone version specifically?

amctagga 2022-07-12 19:52:44 UTC Comment 2

In reply to comment #1:
> Which RHOSP version? And which keystone version
specifically?

https://bugzilla.redhat.com/show_bug.cgi?id=2105317 is our original report. I've CC'd Pierre, who made the report, for more info. Thanks!

Gwyn Ciesla 2022-07-12 21:26:44 UTC Comment 3

(In reply to amctagga from comment #2)

Is there any particular reason I'm CCd? I don't have access to the related bugs. Always willing to help, but not sure how here.

Pierre Prinetti 2022-07-13 08:54:17 UTC Comment 4

(In reply to Luigi Toscano from comment #1)
> Which RHOSP version? And which keystone version
specifically?

rhosp: 16.2
puddle id: RHOS-16.2-RHEL-8-20220513.n.2
rhel_version: 8.4

amctagga 2022-07-14 13:40:45 UTC Comment 5

I don't think this flaw should be embargoed, am curious who changed it and why, since we usually are not embargo'ing moderates these days and it was created as a public flaw. Is there a reason it is listed as such? (I also don't think it's a high/important severity flaw, all other credential leak flaws are moderates.)

Nick Tait 2022-07-16 19:40:28 UTC Comment 6

Ana, I agree on the impact and that there is no need for an embargo. Have assigned a CVE

Pierre Prinetti 2022-07-18 08:01:13 UTC Comment 7

Why was https://bugzilla.redhat.com/show_bug.cgi?id=2105317 cloned here?

Nick Tait 2022-08-12 16:27:33 UTC Comment 9

Created openstack-keystone tracking bugs for this issue:

Affects: openstack-rdo [bug 2117920]