2021-05-24a Registry Advisory

Summary

Registry v3.3.0 introduced CO-specific API users, which can be either privileged (having full access to the CO via the API) or unprivileged (having no specific access unless granted). By crafting a specific sequence of API requests, a privileged CO-specific API user can obtain escalated privileges in another CO, including the COmanage CO (and therefore obtaining platform level privileges).

Severity

The severity of this issue is *medium*, as a privileged API user is required to escalate privileges.

Exposure

The exposure will generally be *low*, as this advisory only meaningfully affects multi-tenant deployments, and only those that have enabled CO-specific API users.

Recommended Mitigation

Deployments not using the described configuration need not take any action, though should plan an upgrade as soon as plausible in case CO-specific API users are created later.

Deployments using the described configuration should immediately upgrade to Registry v3.3.3, or to develop commit b014a9301a or later.

Deployments may also perform an audit, as described in *Discussion*, below.

Alternate Mitigations

Deployments may alternately disable any privileged CO-specific API users until an upgrade can be performed.

Discussion

Registry v3.3.0 introduced CO-specific API users, which can be either privileged (having full access to the CO via the API) or unprivileged (having no specific access unless granted). Previously, the REST API was only available to platform-wide superusers.

Certain data validation routines were not correctly updated as part of this new feature, and as a result a carefully crafted series of API calls could allow a CO-specific privileged API user to create elevated access in another CO, including to a platform administrator identity. This condition does not allow an unprivileged API user to elevate to a privileged API user.

To check for exploits, SQL queries can be used to compare the `actor_identifier` to the CO of the relevant record. For example:

```
WITH t1 AS (
  SELECT cm_identifiers.id,cm_identifiers.actor_identifier,cm_org_identities.co_id
  FROM cm_identifiers
  INNER JOIN cm_org_identities on cm_identifiers.org_identity_id=cm_org_identities.id
  WHERE cm_identifiers.actor_identifier IN (
    SELECT username
    FROM cm_api_users
    WHERE co_id > 1
    AND privileged = true
  )
), t2 AS (
  SELECT username,co_id
  FROM cm_api_users
  WHERE co_id > 1
  AND privileged = true
)
SELECT * FROM t1
LEFT JOIN t2 on t1.actor_identifier=t2.username
WHERE t1.co_id <> t2.co_id;
```

Tables to examine include `cm_co_group_members`, `cm_identifiers`, and `cm_co_person_roles`.

References

- CO-2146