OTF-003: Improper Access Control: Anyone with access to the chat environment can write messages disguised as another chat participant

Moderate **micahflee** published **GHSA-gjj5-998g-v36v** on Jan 18

Package

OnionShare (OnionShare)

Affected versions	Patched versions
>= 2.3	2.5

Description

Between September 26, 2021 and October 8, 2021, Radically Open Security conducted a penetration test of OnionShare 2.4, funded by the Open Technology Fund's Red Team lab. This is an issue from that penetration test.

- Vulnerability ID: OTF-003
- Vulnerability type: Improper Access Control
- Threat level: Moderate

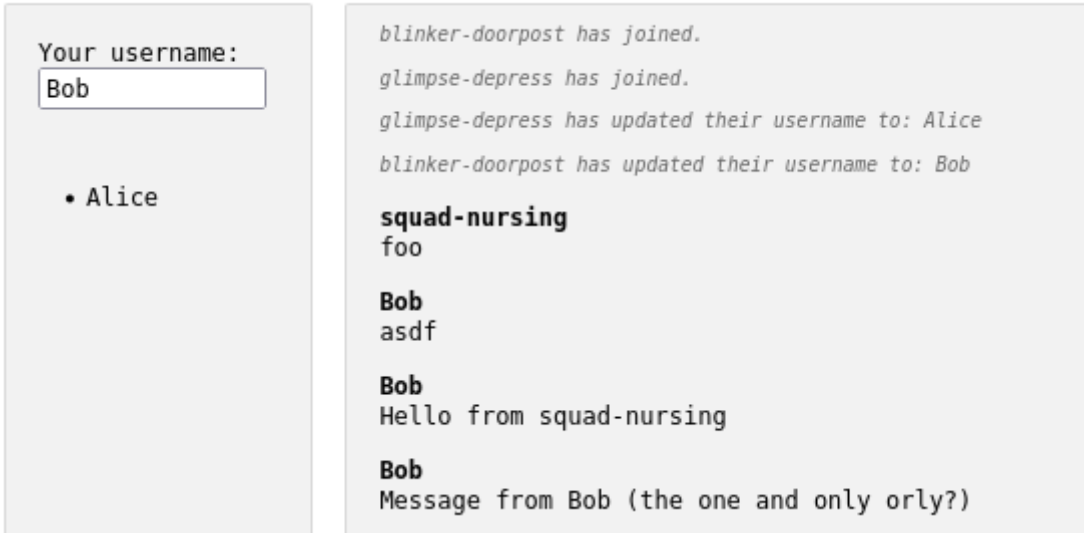
Description:

Anyone with access to the chat environment can write messages disguised as another chat participant.

Technical description:

Prerequisites:

- Alice and Bob are legitimate users
- A third user has access to the chat environment



This screenshot shows Alice (`glimpse-depress`) and Bob (`blinker-doorpost`) joined a chatroom and are the only participants in the chatroom. Then the non-listed user squad-nursing writes a message in the chatroom without being visible in the list of users. The sending of the message itself is not required but was done here to show the initial access. The non-listed participant now renames himself to Bob and writes another message, seemingly coming from Bob.

This can be reproduced by slightly modifying the client-side JavaScript. The `joined` emit needs to be removed from the `socket.on(connect)` event handler. Therefore a client is not listed in the userlist and has no active session.

onionshare/cli/onionshare_cli/resources/static/js/chat.js
Lines 16 to 18 in `d08d5f0`

```
16      socket.on('connect', function () {
17        socket.emit('joined', {});
18      });
```

This can be done either via a crafted client or runtime modification of the `chat.js` script in the browser's internal debugger.

It is still possible to call the text method and send text to the chat via websocket.

onionshare/cli/onionshare_cli/web/chat_mode.py
Lines 131 to 139 in `d08d5f0`

```
131             @self.web.socketio.on("text", namespace="/chat")
132             def text(message):
133                 """Sent by a client when the user entered a new message.
134                 The message is sent to all people in the room."""
135                 emit(
136                     "message",
137                     {"username": session.get("name"), "msg": message["msg"]},
138                     room=session.get("room"),
139                 )
```

It is also possible to call the `update_username` function and choose an existing username from the chat.

onionshare/cli/onionshare_cli/web/chat_mode.py
Lines 141 to 162 in `d08d5f0`

```
141             @self.web.socketio.on("update_username", namespace="/chat")
142             def update_username(message):
143                 """Sent by a client when the user updates their username.
144                 The message is sent to all people in the room."""
145                 current_name = session.get("name")
146                 if message.get("username", ""):
147                     session["name"] = message["username"]
148                     self.connected_users[
149                         self.connected_users.index(current_name)
150                     ] = session.get("name")
151                 emit(
152                     "status",
```

Afterwards the hidden user can send messages that are displayed as coming from the impersonated user. There is no way to distinguish between the fake and original message.

Impact:

An adversary with access to the chat environment can impersonate existing chat participants and write messages but not read the conversation. The similar exploit described in OTF-004 (page 19) has only slightly more requirements but also allows for reading.

Recommendation:

- Implement proper session handling

Severity

Moderate

CVE ID

CVE-2022-21692

Weaknesses

No CWEs