

fix: insecure user creation #2160

New issue

Merged

ajhollid merged 6 commits into develop from fix/insecure-user-creation last week


Conversation 4

Commits 6

Checks 3

Files changed 5

+34 -25

 **ajhollid** commented last week •  
edited by coderabbitai (bot) Collaborator


This PR addresses a critical security bug wherein a user could modify an invite request to grant elevated permisisions


- ✓ Remove role and teamID from request ody
- ✓ Get role and teamID from invite token
- ✓ Add a default role of "user" level


### Summary by CodeRabbit


- Bug Fixes**
  - Corrected the default role assigned to new invitees, ensuring it is set to "user" instead of "0".
  - Fixed a minor syntax issue in the email input handler for improved consistency.
- New Features**
  - Improved handling of invitation tokens during registration, ensuring invited users are assigned the correct role and team automatically.
- Refactor**
  - Updated validation rules for user registration, making the role field optional and improving code readability.
  - Enhanced consistency in formatting across various components for better maintainability.


Reviewers


 **coderabbitai[bot]**

 **llamapreview[bot]**

 **BrOwnHammer**

 **Owaiseimdad**

 **vishnusr77**

 **mohicody**

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone


Development



Successfully merging this pull request may close these issues.





None yet

2 participants



 **ajhollid** added 4 commits last week

  remove role and teamId from invite form d5adf7a

-  get invite role and teamId from invite instead of client c2ffdd1
-  add default role, remove role requirement from validation 99d416f
-  add user as default role  0c31e8f

  **ajhollid** requested review from **Owaiseimdad**, **Br0wnHammer**, **vishnusr77** and **mohicody** last week



**coderabbitai** bot commented last week • edited ▾

 **Caution**

## Review failed

The pull request is closed.

## Walkthrough

The changes update the handling of user invitation roles throughout the client and server code. The default role for invites is now set to `["user"]` in both frontend and backend. The registration process was adjusted to only use the email from invitation data, no longer extracting role or team ID on the client. On the server, the registration logic now pulls and assigns the invited user's role and team ID from the invite token. Validation schemas were updated to make the `role` field optional during registration, and formatting improvements were applied across several files.

## Changes

Files/Groups
client/src/Components/TabPanels/Account/TeamPanel.jsx
client/src/Pages/Auth/Register/Register.jsx

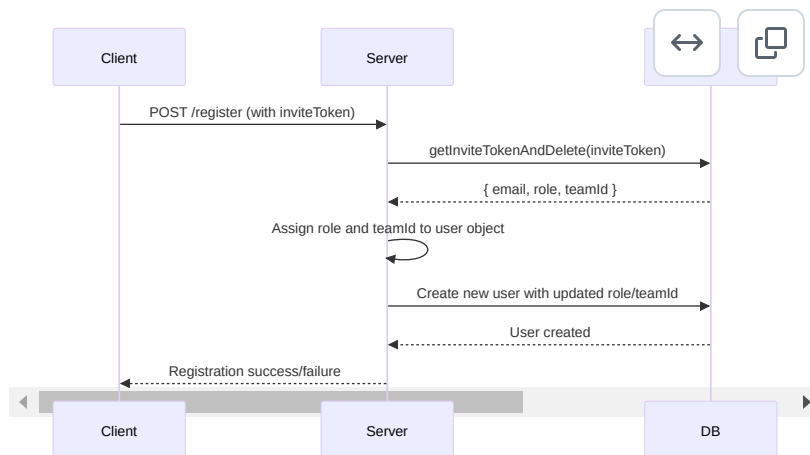
## Files/Groups

server/controllers/authController.js

server/db/models/InviteToken.js

server/validation/joi.js

## Sequence Diagram(s)



## Possibly related PRs

- [fix: edit user #2161](#): Modifies invitation token handling, default role assignment, and validation schema

for the `role` field, closely related to user role management and invitation processing.

## Suggested reviewers

- Owaiseimdad
- vishnusr77
- mohicody

And remember: Canadians apologize when you bump into them, Americans just sue you. So if you get bumped in this code review, feel free to say sorry — no lawsuits needed!

► 📄 Recent review details

▼ ✨ Finishing Touches

☐ 📝 Generate Docstrings

Thanks for using CodeRabbit! It's free for OSS, and your support helps us grow. If you like it, consider giving us a shout-out.

► ❤️ Share

► 📌 Tips



👁 **coderabbitai** bot reviewed  
last week

[View reviewed changes](#)

**coderabbitai** bot left a comment

**Actionable comments posted: 0**

- 🕒 Outside diff range comments (1)
- 🪛 Nitpick comments (1)
- 📄 Review details



👁 **llamapreview** bot reviewed  
last week

[View reviewed changes](#)

**llamapreview** bot left a comment

# Auto Pull Request Review from LlamaPReview

---

## 1. Overview

---

### 1.1 Core Changes

- **Primary purpose and scope:** Address critical security vulnerability preventing privilege escalation through registration requests
- **Key components modified:**
  - Client-side registration forms (TeamPanel.jsx, Register.jsx)
  - Server-side auth controller (authController.js)
  - Invite token schema (InviteToken.js)
  - Joi validation schemas (joi.js)
- **Cross-component impacts:**
  - Changes authentication flow trust boundaries
  - Affects user creation lifecycle end-to-end
- **Business value alignment:**
  - Prevents unauthorized privilege escalation
  - Strengthens security posture for compliance
  - Maintains system integrity for monitoring operations

### 1.2 Technical Architecture

- **System design modifications:**
  - Shifted privilege assignment to server-side token resolution
  - Removed client-controlled role parameters
  - Implemented defense-in-depth through schema defaults
- **Integration points impact:**
  - Registration endpoint now strictly token-dependent
  - First-user creation flow modified for superadmin handling
- **Dependency changes:**
  - Increased reliance on token service validity
  - Added server-side validation requirements for email casing

## 2. Critical Findings

### 2.1 Must Fix (P0🔴)

**Issue:** First user superadmin regression risk

- **Analysis Confidence:** High
- **Impact:** Could lock administrators out of new installations
- **Resolution:**

```
// server/controllers/authController.js
else {
  const jwtSecret = crypto.randomBytes(64).toString("
  await this.db.updateAppSettings({ jwtSecret });
  req.body.role = ["superadmin"]; // Explicit assignm
}
```



**Issue:** Null token dereference crash

- **Analysis Confidence:** High
- **Impact:** Server crashes from invalid tokens create DoS vulnerability
- **Resolution:** Add null check before role assignment

```
const invitedUser = await this.db.getInviteToken
if (!invitedUser) throw new Error("Invalid token...")
```



**Issue:** Joi schema permission injection vector

- **Analysis Confidence:** Medium
- **Impact:** Residual client-controlled role assignment possibility
- **Resolution:** Forbid client-specified roles in validation

```
// server/validation/joi.js
role: joi.forbidden().error(new Error("Roles cannot be"))
```



### 2.2 Should Fix (P1🟡)

**Issue:** Token consumption race condition

- **Analysis Confidence:** Medium
- **Impact:** Potential token reuse in high-concurrency scenarios

- **Suggested Solution:** Use MongoDB's findOneAndDelete for atomic operation

**Issue:** Default role array validation gap

- **Analysis Confidence:** Low
- **Impact:** Schema defaults could be overridden in edge cases
- **Suggested Solution:** Add pre-save validation middleware to InviteToken model

## 2.3 Consider (P2🟢)

**Area:** Email case normalization implementation

- **Analysis Confidence:** High
- **Improvement Opportunity:** Move case enforcement to server-side validation for stronger consistency

**Area:** Token metadata logging

- **Analysis Confidence:** Medium
- **Improvement Opportunity:** Enhanced audit trails for security investigations

## 2.4 Summary of Action Items

Priority	Action Item	Timeline
P0	Implement superadmin first-user fix	Immediate
P0	Add null token validation	Immediate
P0	Forbid client role specification	Immediate
P1	Atomic token consumption	Next release
P1	Schema validation middleware	Next release

## 3. Technical Analysis

### 3.1 Code Logic Analysis

📁 **server/controllers/authController.js - registerUser**

- Submitted PR Code:

```

else {
  const jwtSecret = crypto.randomBytes(64).toString('hex');
  await this.db.updateAppSettings({ jwtSecret });
}

```

- Analysis:
  - Missing superadmin role assignment for first user
  - Could prevent administrative access to new installations
  - Critical impact on system usability
- LlamaPReview Suggested Improvements:

```

else {
  const jwtSecret = crypto.randomBytes(64).toString('hex');
  await this.db.updateAppSettings({ jwtSecret });
  user.role = ["superadmin"]; // Ensure first user
}

```

- Improvement rationale:
  - Guarantees system accessibility
  - Maintains expected admin workflow

### 📁 server/validation/joi.js - registrationBodyValidation

- Submitted PR Code:

```

role: joi.array().items(joi.string().valid(...

```

- Analysis:
  - Allows empty role arrays from client
  - Potential privilege assignment bypass
  - Security-critical validation gap
- LlamaPReview Suggested Improvements:

```

role: joi.forbidden().error(new Error("Roles not

```

- Improvement rationale:
  - Eliminates client-controlled privilege vector
  - Enforces server-side role management

## 3.2 Key Quality Aspects

- Testing strategy:



- Requires new test cases for first-user superadmin flow
- Need negative testing for invalid token scenarios
- **Documentation needs:**
  - Update API docs for removed role/teamID parameters
  - Add security considerations to registration flow docs

## 4. Overall Evaluation

- **Technical assessment:** Addresses critical security flaw with generally sound approach
- **Business impact:** Prevents high-risk privilege escalation scenarios
- **Risk evaluation:** P0 fixes required before merge to prevent regression risks
- **Notable positives:**
  - Effective removal of client-controlled privileges
  - Improved token-centric security model
- **Implementation quality:** Requires additional validation safeguards
- **Final recommendation:** Request Changes (Approve after P0 resolutions)

### 💡 LlamaPReview Community

Have feedback on this AI Code review tool? Join our [GitHub Discussions](#) to share your thoughts and help shape the future of LlamaPReview.

🔗  disallow role ✓ 67ae722




**Br0wnHammer** approved these changes last week View reviewed changes

**Br0wnHammer** left a comment • edited ▾

Member

LGTM!

Crisp and clean PR.

🔗  Merge pull request [#2161](#) from Verified ✓ b387eba  
bluewave-labs/fix/edit-user ...



**ajhollid** merged commit **d4a6072** into

develop last week

2 of 3 checks passed

[View details](#)



**ajhollid** deleted the `fix/insecure-user-creation`  
branch last week