

f / prompts.chat Public

<> Code Issues 9 Pull requests 33 Actions Projects Models Se

# Commit 1464475



mdisec committed 2 weeks ago · ✖ 2 / 9

Fix username case-collision vulnerability across write and read paths

main (#1098)

1 parent [0bd2b10](#) commit 1464475

6 files changed +33 -9 lines changed

[↑ Top](#)

- ✓ src
  - ✓ \_\_tests\_\_/api
    - user-profile.test.ts
  - ✓ app
    - ✓ [username]
      - opengraph-image.tsx
      - page.tsx
    - ✓ api
      - ✓ admin/import-prompts
        - route.ts
      - ✓ user/profile
        - route.ts
    - ✓ lib/auth
      - index.ts

6 files changed +33 -9 lines changed

```

src/___tests___/api/user-profile.test.ts
  ...
  8      8      db: {
  9      9      user: {
 10     10      findUnique: vi.fn(),
 11     11 +     findFirst: vi.fn(),
 11     12      update: vi.fn(),
 12     13      },
 13     14      },
  ...
  ...
 162    163
 163    164      it("should return 400 if username is taken", async () => {
 164    165        vi.mocked(auth).mockResolvedValue({ user: { id: "user1", username:
 165    166          "olduser" } } as never);
 166    167        vi.mocked(db.user.findUnique).mockResolvedValue({ id: "other-user" } as
 167    168          never);
 168    169        vi.mocked(db.user.findFirst).mockResolvedValue({ id: "other-user" } as
 169    170          never);
 170    171        const request = new Request("http://localhost:3000/api/user/profile", {
 171    172          method: "PATCH",
 172    173          body: JSON.stringify({ name: "Test", username: "TakenUser" }),
 173    174        });
 174    175        const response = await PATCH(request);
 175    176        const data = await response.json();
 176    177        expect(data.error).toBe("username_taken");
 177    178      });
 178    179
 179    180 +   it("should check username case-insensitively", async () => {
 180    181 +     vi.mocked(auth).mockResolvedValue({ user: { id: "user1", username:
 181    182 +       "olduser" } } as never);
 182    183 +     vi.mocked(db.user.findFirst).mockResolvedValue({ id: "other-user" } as
 183    184 +       never);
 184    185 +     const request = new Request("http://localhost:3000/api/user/profile", {
 185    186 +       method: "PATCH",
 186    187 +       body: JSON.stringify({ name: "Test", username: "TakenUser" }),
 187    188 +     });
 188    189 +     const response = await PATCH(request);
 189    190 +     const data = await response.json();
 190    191 +
 191

```

```

192 + expect(response.status).toBe(400);
193 + expect(data.error).toBe("username_taken");
194 + expect(db.user.findFirst).toHaveBeenCalledWith({
195 +   where: { username: { equals: "TakenUser", mode: "insensitive" } },
196 +   select: { id: true },
197 + });
198 + });
199 +
179 200   it("should allow keeping the same username", async () => {
180 201     vi.mocked(auth).mockResolvedValue({ user: { id: "user1", username:
"sameuser" } } as never);
181 202     vi.mocked(db.user.update).mockResolvedValue({
@@ -197,12 +218,12 @@ describe("PATCH /api/user/profile", () => {
197 218     expect(response.status).toBe(200);
198 219     expect(data.name).toBe("Updated Name");
199 220     // Should NOT check for existing username when keeping the same one
200 -     expect(db.user.findUnique).not.toHaveBeenCalled();
221 +     expect(db.user.findFirst).not.toHaveBeenCalled();
201 222   });
202 223
203 224   it("should update profile successfully", async () => {
204 225     vi.mocked(auth).mockResolvedValue({ user: { id: "user1", username:
"olduser" } } as never);
205 -     vi.mocked(db.user.findUnique).mockResolvedValue(null); // Username not
taken
226 +     vi.mocked(db.user.findFirst).mockResolvedValue(null); // Username not taken
206 227     vi.mocked(db.user.update).mockResolvedValue({
207 228       id: "user1",
208 229       name: "New Name",
@@ -318,7 +339,7 @@ describe("PATCH /api/user/profile", () => {
318 339
319 340     it("should accept valid username with underscores", async () => {
320 341       vi.mocked(auth).mockResolvedValue({ user: { id: "user1", username: "old" }
} as never);
321 -     vi.mocked(db.user.findUnique).mockResolvedValue(null);
342 +     vi.mocked(db.user.findFirst).mockResolvedValue(null);
322 343     vi.mocked(db.user.update).mockResolvedValue({
323 344       id: "user1",
324 345       name: "Test",

```


 src/app/[username]/opengraph-image.tsx
 ...


```
@@ -50,6 +50,7 @@ export default async function OGImage({ params }: { params:
  Promise<{ username:
```

50 50

51 51 `const user = await db.user.findFirst({`52 52  `where: { username: { equals: username, mode: "insensitive" } },`53 +  `orderBy: { createdAt: "asc" },`53 54  `select: {`54 55  `id: true,`55 56  `name: true,`
 src/app/[username]/page.tsx
 ...


```
@@ -38,6 +38,7 @@ export async function generateMetadata({ params }:
  UserProfilePageProps): Promis
```

38 38

39 39 `const user = await db.user.findFirst({`40 40  `where: { username: { equals: username, mode: "insensitive" } },`41 +  `orderBy: { createdAt: "asc" },`41 42  `select: { name: true, username: true },`42 43 `});`

43 44



```
@@ -72,6 +73,7 @@ export default async function UserProfilePage({ params,
  searchParams }: UserProf
```

72 73

73 74 `const user = await db.user.findFirst({`74 75  `where: { username: { equals: username, mode: "insensitive" } },`76 +  `orderBy: { createdAt: "asc" },`75 77  `select: {`76 78  `id: true,`77 79  `name: true,`
 src/app/api/admin/import-prompts/route.ts
 ...


```
@@ -150,7 +150,7 @@ export async function POST(request: NextRequest) {
```

150 150 `let user = await db.user.findFirst({`151 151  `where: {`

```

152 152          OR: [
153 -          { username: normalizedUsername },
153 +          { username: { equals: normalizedUsername, mode: "insensitive" } },
154 154          { email: pseudoEmail },
155 155        ],
156 156      },

```



src/app/api/user/profile/route.ts



```

@@ -46,8 +46,8 @@ export async function PATCH(request: NextRequest) {
46 46
47 47      // Check if username is taken by another user
48 48      if (username !== session.user.username) {
49 -      const existingUser = await db.user.findUnique({
50 -      where: { username },
49 +      const existingUser = await db.user.findFirst({
50 +      where: { username: { equals: username, mode: "insensitive" } },
51 51      select: { id: true },
52 52      });
53 53

```



src/lib/auth/index.ts



```

@@ -26,7 +26,7 @@ async function generateUsername(email: string, name?:
string | null): Promise<st
26 26      // Check if username exists and append number if needed
27 27      let username = baseUsername;
28 28      let counter = 1;
29 -      while (await db.user.findUnique({ where: { username } })) {
29 +      while (await db.user.findFirst({ where: { username: { equals: username, mode:
"insensitive" } } })) {
30 30      username = `${baseUsername}${counter}`;
31 31      counter++;
32 32      }
@@ -81,7 +81,7 @@ function CustomPrismaAdapter(): Adapter {
81 81      const baseUsername = username;
82 82      let finalUsername = baseUsername;
83 83      let counter = 1;

```

```
84 - while (await db.user.findUnique({ where: { username: finalUsername } }))  
    {  
84 + while (await db.user.findFirst({ where: { username: { equals:  
    finalUsername, mode: "insensitive" } } })) {  
85 85     finalUsername = `${baseUsername}${counter}`;  
86 86     counter++;  
87 87     }  
    ↓
```

## Comments 0



Please [sign in](#) to comment.