2.2 Details of High-Risk Vulnerabilities

2.2.1 CQ-001: Stored Cross-Site Scripting (XSS)

CQ-001: Stored Cross-Site Scripting (XSS)

Severity	High	CVSS	10.0	Status	Open

References	CVSS Metric	
● OWASP Top Ten A03:2021-Injection	Attack Vector (AV)	Network
	Attack Complexity (AC)	Low
	Privileges Required (PR)	None
	User Interaction (UI)	None
	Scope (S)	Changed
	Confidentiality (C)	High
	Integrity (I)	High
	Availability (A)	None

Affected
`https://instance.cq.firebird.sh/`

Details

The application currently faces a security vulnerability known as Cross-Site Scripting (XSS). This occurs when an attacker injects malicious code into the application, which can then be executed by other users.

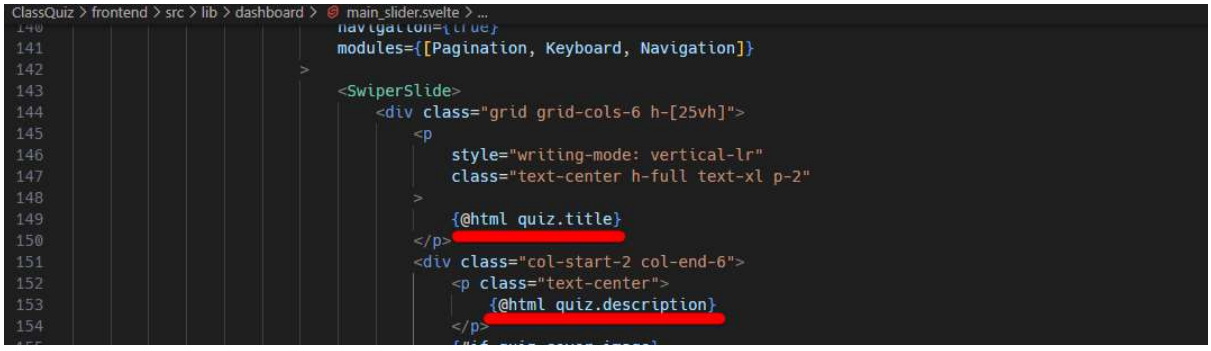

Figure 69: The Svelte frontend parses the quiz title and description as HTML

The attempt to secure user input by filtering HTML tags using the Python library "`bleach`" was made, but unfortunately, it is ineffective due to an incorrect use of the "`html.unescape`" function.

```
ClassQuiz > classquiz > routers > editor.py > finish_edit
69  @router.post("/finish")
70  async def finish_edit(edit_id: str, quiz_input: QuizInput):
71      session_data = await redis.get(f"edit_session:{edit_id}")
72      if session_data is None:
73          raise HTTPException(status_code=401, detail="Edit ID not found!")
74      session_data = EditSessionData.parse_raw(session_data)
75      quiz_input.title = html.unescape(bleach.clean(quiz_input.title, tags=ALLOWED_TAGS_FOR_QUIZ, strip=True))
76      quiz_input.description = html.unescape(bleach.clean(quiz_input.description, tags=ALLOWED_TAGS_FOR_QUIZ, strip=True))
77      if quiz_input.background_color is not None:
78          quiz_input.background_color = html.unescape(bleach.clean(quiz_input.background_color, tags=[], strip=True))
79
```

Figure 70: The uses of `html.unescape` function was incorrect and allows bypasses

The following example shows a possible sanitization bypass, by setting the question title to this value, an alert box would be shown in Dashboard:

```
&lt;img src=x onerror='alert(1)'>
```

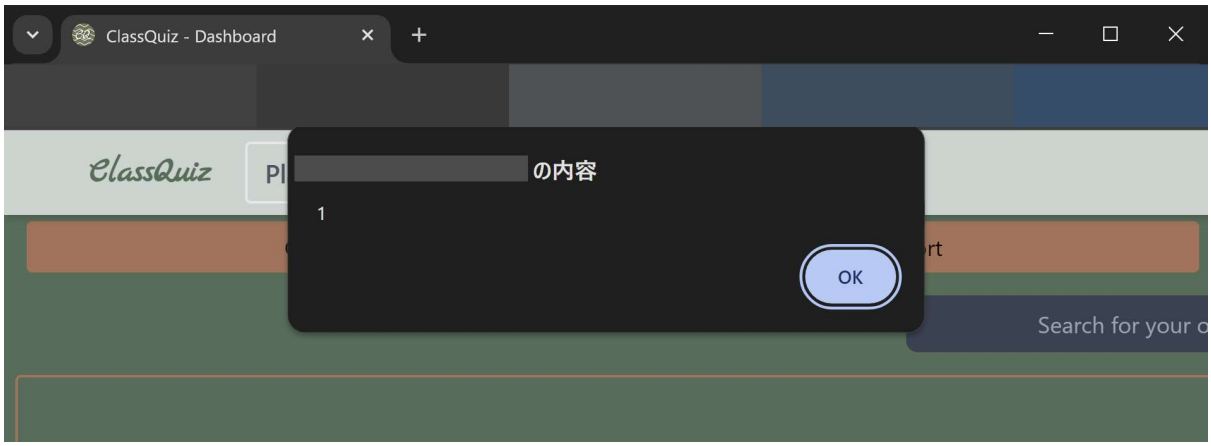

Figure 71: Alert box was shown in Dashboard when the payload was injected

Impact

With this vulnerability, attackers could potentially gain unauthorized access to any user account within the application. This includes sensitive accounts such as administrative accounts, allowing malicious individuals to manipulate and misuse the system's functionalities.

Recommendations

To address this issue, it is crucial to fix the incorrect usage of the "`html.unescape`" function and ensure that the filtering of HTML tags is properly implemented.

Remediation Responses

Jan 1, 2024 – While it may seem like a high-risk vulnerability, it's just displaying "1" in alert boxes – harmless, really! Well, as an extra layer of precaution, we've updated the ClassQuiz application to its latest version, and implemented XSS Protection and Content-Security-Policy headers to fortify the site's security.

We've put in some effort, hoping it meets your expectations. Please review and approve our fix; we'd appreciate it if you could refrain from further inquiries. Cheers!

Status

Open - Remediation applied by vendor