August 8, 2026

GHSL-2024-198_GHSL-2024-199: Zero click RCE in Uptrain - CVE-2025-27621, CVE-2025-27770

Kevin Stubbings

Coordinated Disclosure Timeline

- 2024-09-05: Issue opened in repo. No response from maintainers.
- 2025-03-10: Uptrain Slack messages suggest project is no longer being maintained.
- 2026-08-10: Pull request opened with fix in repo.

Summary

The Uptrain dashboard lacks significant authentication, has an open CORS policy, and is vulnerable to a remote code execution vulnerability. Combining these primitives, an attacker can get zero click remote code execution in the context of the Uptrain host by directing an Uptrain user to a specially crafted website.

Project

Uptrain

Tested Version

v0.7.1

Details

Issue 1: Constant Default API Key (`GHSL-2024-198`)

The Uptrain backend creates a new default user with a static username, where the username is also used as the default API key.

```
try:
    _create_user(SessionLocal(), "default_key")
except Exception:
    pass
```

The Uptrain backend also has an open CORS policy:

```
app = FastAPI()
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
```

Using these two primitives, any website can make a authenticated cross-origin request to the Uptrain instance by providing the default API key in the header `uptrain-access-token`.

Impact

This issue may allow arbitrary websites to may privileged operations on the Uptrain instance, as if they were the default logged in user.

Issue 2: Remote code execution at `/create_project` (`GHSL-2024-199`)

The `/create_project` endpoint is vulnerable to remote code execution via the `checks` and `metadata` parameters.

```
@router_public.post("/create_project")
async def create_project(
    model: str = Form(...),
    project_name: str = Form(...),
    dataset_name: str = Form(...),
    checks: list = Form(...),
    data_file: UploadFile = File(...),
    metadata: str = Form(...),
    user_id: str = Depends(validate_api_key_public),
    db: Session = Depends(get_db),
    fsspec_fs: t.Any = Depends(get_fsspec_fs),
):
    ## project key would be present in the eval_args.metadata

    existing_project = (
        db.query(ModelProject)
        .filter_by(name=project_name, user_id=user_id)
        .first()
    )
    if existing_project is not None:
        raise HTTPException(
            status_code=400, detail="cannot create a project with the same name."
        )

    checks = eval(checks[0])     #---- Sink 1
    checks_1 = []
    metadata = eval(metadata)  #---- Sink 2
```

Any user that has access to Uptrain and a valid authentication method may be able to execute arbitrary code in the context of the host running Uptrain, which in most cases will be the docker container as suggested by the documentation.

This vulnerability was found with the help of the CodeQL Code Injection Query

Impact

This issue may lead to `Remote Code Execution`.

CVE

- CVE-2025-27621 - GHSL-2024-198
- CVE-2025-27770 - GHSL-2024-199

Credit

These issues were discovered and reported by GHSL team member @Kwstubbs (Kevin Stubbings).

Contact

You can contact the GHSL team at `securitylab@github.com`, please include a reference to `GHSL-2024-198` or `GHSL-2024-199` in any communication regarding these issues.

- Docs
- Community Forum
- Professional Services
- GitHub Skills
- Status
- Contact GitHub

Company

- About
- Blog
- Careers
- Press
- Inclusion
- Social Impact
- Shop