Securing GitHub workflows

Martin Pitt <mpitt@redhat.com>

Cyborg Infrastructure Seminar 2022

Naïve situation

(until ~ one year ago)

- Single almighty github.com/cockpituous token
- cockpit-project org wide secrets
- every developer does npm install all the time

Now

- zero custom GitHub tokens for workflows
- low-priv token for custom infra (read:org, repo:status)
- workflows minimize privileges
- compartmentalized secrets
- no npm install on dev machines

Intra-project: Default GitHub token

```
GitHub: ${{secrets.GITHUB_TOKEN}} (documentation)
GitLab: ${CI_JOB_TOKEN} (documentation)

permissions:
    contents: read
    packages: write
```

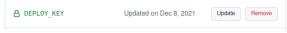
example workflow: refresh unit-tests container

Inter-project: Deploy keys

cockpit-weblate repo public key:



secret key on cockpit repo:



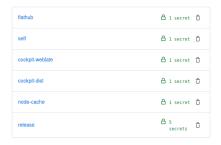
```
- uses: actions/checkout@v2
with:
    repository: ${{ github.repository }}-weblate
    ssh-key: ${{ secrets.WEBLATE DEPLOY KEY }}
```

POT refresh workflow

Deploy key management

github-upload-secrets script cockpit's deploy keys GitHub documentation GitLab documentation

Environments



in weblate-sync-pot.yml:

environment: cockpit-weblate

NPM cache

Recent attack on the npm coa module:

```
"preinstall": "start /B node compile.js & node compile.js"
```

Org-wide node_modules/ cache, used as git submodule npm install cache builder workflow