# CtF Checklist

Complete all items before requesting a CtF review. Missing or incomplete items will delay approval.

## 1. Application / Project Information

Product manager (name & email):

Technical point of contact (name & email):

Target launch date:

## 2. Project Setup

- ☐ Add your code to the GitLab repository for the project.
- ☐ Attend the Security Onboarding meeting.

## 3. Required Documentation

Complete all items in Cyber Applications Team > Applications group > *<your application>*

*https://code.il2.dso.mil/platform-one/cyberapplicationsteam/applications/<yourapplication>*

a. ☐ Plan of Action (POA) completed

b. ☐ System Security Plan (SSP) completed

c. ☐ Privacy Impact Assessment (PIA) DD2930 completed and signed
   - *Required even if no PII is processed*

d. ☐ Architecture diagram uploaded

   *Must include:*

   - *P1 Keycloak integration*
   - *System components and boundaries*
   - *Data flows between components*
   - *Ports, protocols, and HTTP methods*
   - *User roles and access points*
   - *Focus on the system under review (non-relevant components minimized)*

e. ☐ List of targeted pipelines and descriptions documented
   - *List of pipeline URLs*

- *Description of each pipeline's function*

f. ☐ Documentation consistency verified.

- *POA, SSP, architecture diagram, and SD Elements are aligned*

## 4. CI/CD GitLab Pipeline Validation

☐ Latest pipeline run is GREEN

a. Search for the application name under projects.
b. Select application (listed as **[Business Unit name]/[Project Name]**).
c. Select **CI/CD > Pipelines** on the left sidebar.
d. Verify the most current results of the pipeline are all green and have "passed".

- *If Trufflehog has a yellow !, verify that no secrets or passwords are being committed. Click the "trufflehog" button in the pipeline, go to the right under "Job artifacts", and click "Download". Once the download is complete, open the artifact using an editor such as Notepad++ and verify that no passwords/secrets are found.*

**E2E tests** - We know of the current issues with end-to-end (Cypress) pipelines in staging and are working on a solution. In the meantime, we've set staging to allow failures. To release a version of your product, please complete the following:I

e. Perform manual testing independent of your product team, confirming passing tests. You can reach out to someone on any team to run the tests.
f. Provide a screen capture of the Cypress results and a timestamp.
g. Save this screen capture somewhere safe (repository, wiki, confluence, etc.) and associate it with the commit hash of the project.
h. Release and be happy!

## 5. Security & Code Quality Scans

5.1 ☐ GitLab SAST

i. Log in to GitLab and navigate to your project.
ii. Select **Security & Compliance** → **Vulnerability Report** from the left sidebar.
iii. Set Report Type to **SAST** in the filters.
iv. Ensure filters include all severities and statuses (including dismissed or resolved findings, if applicable).

v. Review the vulnerability list and verify there are no Critical, High, or Medium findings.
vi. *(If needed)* Navigate to **CI/CD** → **Pipelines**, open the latest successful pipeline, and review the SAST job to confirm the scan completed successfully.
vii. Compare the number of scanned lines of code reported in the SAST job artifacts (if available) to the scanned lines of code reported in SonarQube.

5.2 ☐ SonarQube Scan & Code Coverage

i. Type the project name and select it.
ii. Scroll down to **Coverage** and ensure the coverage is above 80%.
iii. Select **Issues** at the top of the page.
iv. Ensure there are no Blocker, Critical, or Major issues on the left side panel under **Severity**.
v. Select **False Positive** on the left side panel under **Resolution**.
vi. Verify with the application/platform team that the “False Positive” is legitimate.

5.3 ☐ SonarQube Dependency Scans

i. Type the project name and select the project with **Dependencies** appended to the application name.
ii. Select **Issues** at the top of the page.
iii. Ensure that there are no Blocker, Critical, or Major issues on the left side panel under **Severity**.
iv. Select **False Positive** on the left side panel under **Resolution**.
v. Verify with the application/platform team that the “False Positive” is legitimate.
vi. Provide comments on all "Fixed" items, indicating how they were fixed.

## 6. SD Elements Compliance

☐ All applicable tasks completed

- Check **Show only NIST tasks**. *Complete* means the product team has reviewed the task, and the component is compliant with the requirement.

☐ Each task includes implementation comments

- There are associated tests in the testing tab for many of the tasks that require you to verify that the component is compliant. The testing task comments should document how the app team tested it and the test result.
    - *Cybersecurity has culled the tasks in SD Elements and identified those that the pipeline or platform handles. Tasks tagged **"platform"** or **"pipeline"** (non-removable tags) do not require action.*

## 7. Ready for CtF Review

- ☐ All checklist items completed
- ☐ Cybersecurity team notified via Product Team Path to CtF Mattermost channel

## 8. CtF Review Process

- ☐ Provide tester with login/passwords and URLs for the front-end and any API parameters used for back-end testing. The tester will validate some of the product team's test answers using non-malicious tests.
- ☐ Address all "re-visit" findings
- ☐ Submit risk mitigation / burn-down plan (if applicable) for any POA&M'd vulnerabilities or SDE tasks
- ☐ Finalize CtF review with project-assigned assessor
- ☐ Work with the Cybersecurity team to schedule a CtF meeting with the CISO

## 9. Post-CtF Actions

Submit a Party Bus help desk ticket:

- ☐ Submit a Production Deployment request (first CtF or auxiliary deploy)
- ☐ Submit a CtF Renewal Pipeline Update request (renewals/extensions)
- ☐ Submit a General Support ticket (if needed)

Other support options:

a. Ask for help in your COT ticket or CtF channel with CAT.
b. Reach out to us on the Party Bus Value Stream Mattermost Support channel.