Git-Ctf Challenge Wrriteup

P1 — Pre-amend secret API key

```
# List dangling commits / check reflog

git reflog

git show eca4a6b8b7329952a4762b0a7e83ca243fa5f825:src/config/settings.json

# OR simply view the dangling commit

git show eca4a6b8b7329952a4762b0a7e83ca243fa5f825

# Output contains: djsisaca{
```

P2 — Deleted feature branch

```
# Check for dangling commits

git fsck --lost-found

# Inspect commits

git show e1771af7330385a950ee879486ef3c0f1b86e190

# Output contains: Atl@st_
```

P3 — Commit messages

```
# Show all commit messages in repo

git log --all --pretty=format:"%B"

# Combine the fragments in messages:

# fix: small logging tweak — y0u+

# perf: clean up message f0und_

# Output: y0u+f0und_
```

P4 — Deleted annotated tag

```
git fsck --lost-found

# Look for dangling tag objects

git cat-file -p <dangling_tag_sha>

# Output contains: th3_fl@g_
```

P5 — Dangling blob (gzipped)

```
# List dangling blobs
git fsck --lost-found
# Suppose the blob SHA is: aa1859ebbbe61bc9ffa3e5c2390429e4003617a8
# Decode and gunzip in one command
git cat-file -p aa1859ebbbe61bc9ffa3e5c2390429e4003617a8 | gunzip -c
# Output: fin@LLy
```

P6 — Rebased commit

```
# Check original commits from rebase
git reflog
# Example dangling commit SHA: 1611d1b8e7326a6ae93159c6baeb5a71e7bd88d2
git show 1611d1b8e7326a6ae93159c6baeb5a71e7bd88d2
# Output contains: :)}
```

Final Flag

Concatenate all parts in order:

djsisaca{Atl@st_y0u+f0und_th3_fl@g_fin@LLy:)}