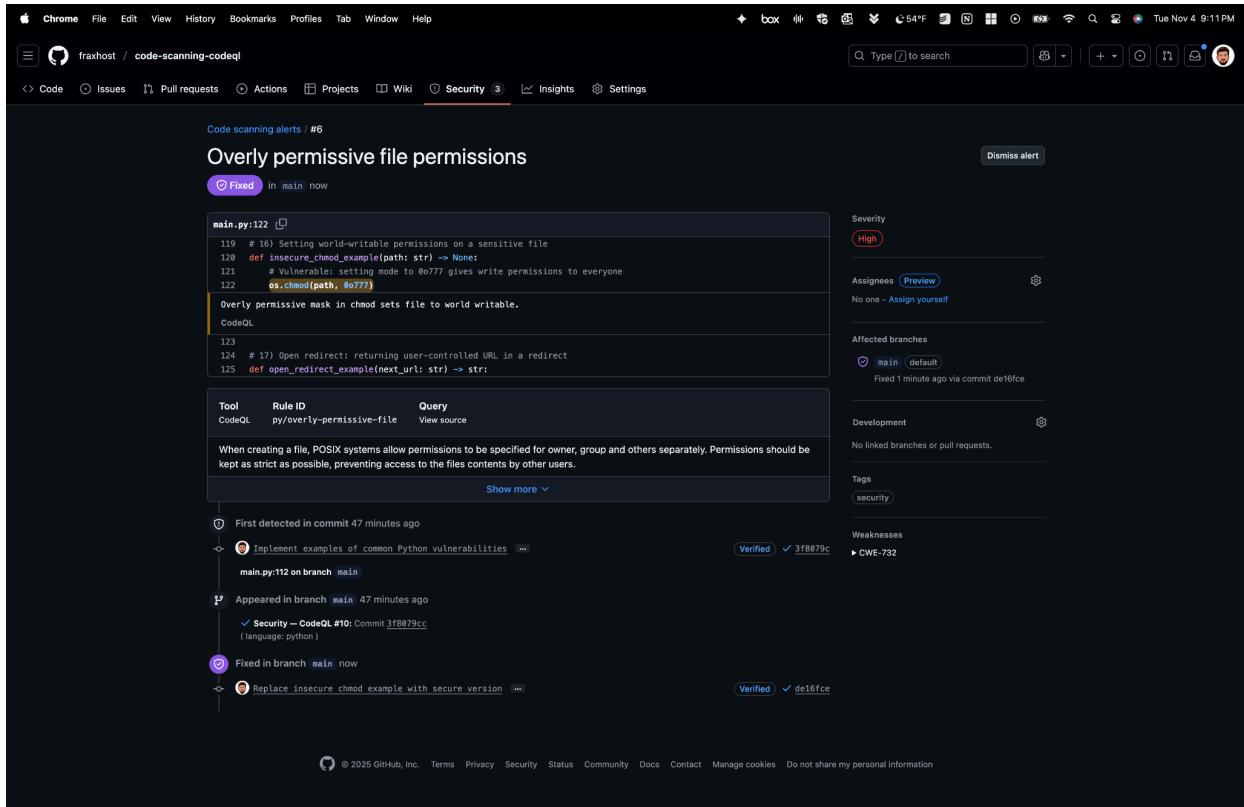


(1) Fix Overly Permissive File Permissions



Screenshot: Fix Overly Permissive File Permissions

(2) Fix Insecure Temporary File

The screenshot shows a GitHub repository named 'fraxhost / code-scanning-codeql' with a pull request titled 'Code scanning alerts / #5'. The alert details an 'Insecure temporary file' vulnerability. The code snippet in main.py:103 shows the use of `tempfile.mktemp`. A red box highlights the call to `tempfile.mktemp(prefix="tmpvol_")`, with a tooltip stating 'Call to deprecated function tempfile.mktemp may be insecure.' The alert is marked as 'Fixed' and was made 2 minutes ago. The severity is 'High'. The affected branches are 'main' and 'default'. The commit 'f22d881' fixed the issue. The alert also links to 'Implement examples of common Python vulnerabilities' and 'Improve security of temporary file creation'.

Screenshot: Fix Insecure Temporary File

(3) Fix Request without Certificate Validation

The screenshot shows a GitHub repository named 'fraxhost / code-scanning-codeql'. The 'Security' tab is selected, displaying a list of alerts. The first alert is titled 'Request without certificate validation' and is categorized as 'High' severity. It points to a line of Python code in 'main.py:64':

```
61
62 # 8) Disabling SSL certificate verification
63 def disable_ssl_verification(url: str) -> requests.Response:
64     resp = requests.get(url, verify=False) # disabled TLS verification
65
66     return resp
67 # 9) Logging sensitive data
```

A red box highlights the line `resp = requests.get(url, verify=False)` with the note: "This request may run without certificate validation because it is disabled." Below the code, there's a summary table:

Tool	Rule ID	Query
CodeQL	py/request-without-cert-validation	View source

The summary text explains: "Encryption is key to the security of most, if not all, online communication. Using Transport Layer Security (TLS) can ensure that communication cannot be interrupted by an interceptor. For this reason, it is unwise to disable the verification that TLS provides. Functions in the `requests` module provide verification by default, and it is only when explicitly turned off using `verify=False` that no verification occurs."

On the right side, the alert details include:

- Severity:** High
- Assignees:** Preview (No one - Assign yourself)
- Affected branches:** main (default) - Fixed 2 minutes ago via commit 441cfaa
- Development:** No linked branches or pull requests.
- Tags:** security
- Weaknesses:** CWE-295

Below the alert, a timeline shows the following events:

- First detected in commit 1 hour ago
- Add vulnerable examples for CodeQL testing ... (Verified ✓ b6cce75f)
- Appeared in branch main 1 hour ago
 - ✓ Security - CodeQL #9: Commit b6cce75f (language: python)
- Fixed in branch main 1 minute ago
 - Update SSL verification to use secure requests ... (Verified ✓ 441cfaa)

At the bottom, there's a footer with links to GitHub's Terms, Privacy, Security, Status, Community, Docs, Contact, Manage cookies, and a 'Do not share my personal information' link.

Screenshot: Request without Certificate Validation

(4) Fix Use of a Broken or Weak Cryptographic Hashing Algorithm on Sensitive Data

The screenshot shows a GitHub repository named 'fraxhost / code-scanning-codeql' with a single commit. A security alert is displayed for the file 'main.py' at line 39. The alert details:

- Severity:** High
- Assignees:** No one - Assign yourself
- Affected branches:** main (default) - Fixed 1 minute ago via commit 97fda2a
- Tool:** CodeQL
- Rule ID:** py/weak-sensitive-data-hashing
- Query:** View source
- Description:** Using a broken or weak cryptographic hash function can leave data vulnerable, and should not be used in security related code.
- First detected in commit:** 1 hour ago
- Associated commit:** b6cce75f (language: python)
- Weaknesses:** CWE-327, CWE-328, CWE-916

Below the alert, there are two suggestions:

- Add vulnerable examples for CodeQL testing
- Implement secure password hashing with bcrypt

At the bottom of the page, there is a footer with links to GitHub's Terms, Privacy, Security, Status, Community, Docs, Contact, Manage cookies, and a note about not sharing personal information.

Screenshot: Use of a Broken or Weak Cryptographic Hashing Algorithm on Sensitive Data

(5) Fix Clear-Text Logging of Sensitive Information

The screenshot shows a GitHub repository named 'fraxhost / code-scanning-codeql'. A specific commit has triggered a 'Code scanning alerts' warning. The alert details a 'Clear-text logging of sensitive information' issue found in 'main.py:82'. The code snippet shows a logging statement that logs the password in clear text:

```
80 # 9) Logging sensitive data
81 def sensitive_logging_example(username: str, password: str) -> None:
82     logging.warning(f"User login attempt: user={username}, password={password}")
83
84 # 10) Exposing stack trace / information disclosure
85 def expose_stack_trace_example() -> None:
```

The alert is categorized under 'Severity: High' and 'Tool: CodeQL'. It includes a 'Query' link to view the source code. A note states: 'If sensitive data is written to a log entry it could be exposed to an attacker who gains access to the logs.' Below the alert, a timeline shows the detection of the vulnerability in the commit, its appearance in the branch, and its subsequent fix. The fix was committed 1 minute ago and is verified. The alert also lists related weaknesses: CWE-312, CWE-359, and CWE-532.

Screenshot: Clear-Text Logging of Sensitive Information