

# Semgrep Pattern

Hritesh Sonawane

## Task 1:

metavariables\_class\_task\_1-copy

Saved

Share

Add to Policy

structure

NEW

advanced

```
1 rules:
2   - id: metavariables_class_task_1-copy
3     languages:
4       - python
5     severity: ERROR
6     message: Semgrep found a match
7     pattern: requests.get($BLAH, timeout=$TIMEOUT, verify=True)
```

test code

live code

NEW

metadata

docs

Pro Turbo

```
1 import requests
2
3 # match these
4 requests.get("semgrep.dev", timeout=30, verify=True)
5 requests.get("semgrep.dev", verify=True, timeout=10)
6
7 # but do not match these
8 requests.get("semgrep.dev", timeout=10)
9 requests.get("semgrep.dev", verify=True)
10 requests.get("semgrep.dev", verify=False, timeout=10)
11 requests.get("semgrep.dev", secure=True, timeout=10)
12
13
14 # Write a Semgrep pattern to match on calls to requests.get which specify any string as a url
   (positional argument), any timeout value, and that verify is true.
```

--

pattern: requests.get(\$BLAH, timeout=\$TIMEOUT, verify=True)

---

## Task 2:

metavariables\_class\_task\_2-copy

Saved

Share

Add to Policy

structure

NEW

advanced

```
1 rules:
2   - id: metavariables_class_task_2-copy
3     message: Semgrep found a match
4     languages:
5       - python
6     severity: WARNING
7     pattern: requests.get($BLAH, verify=False, ...)
```

test code

live code

NEW

metadata

docs

Pro Turbo

```
1 import requests
2
3 # safe
4 requests.get('https://github.com')
5
6 # safe
7 requests.get(url, path_to_cert_file)
8
9 # safe
10 requests.get('https://github.com', verify=True)
11
12 # vulnerable
13 requests.get('https://github.com', verify=False)
14
15 # vulnerable
16 requests.get(url, verify=False, timeout=30)
17
18
19 # get calls with verify=False are vulnerable, by default verify=True
```

--

pattern: requests.get(\$BLAH, verify=False, ...)

--