Semgrep SAST Scan Report for Repository: bad-python-app

Report Generated at 2024-09-04 21:52

SAST Scan Summary

Vulnerability Severity	Vulnerability Count
Findings- SAST High Severity	32
Findings- SAST Medium Severity	61
Findings- SAST Low Severity	0

Findings Summary- High Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location
var-in-script-tag	Detected a template variable used in a script tag. Although template variables are HTML escaped, HTML escaping does not always prevent cross-site scripting (XSS) attacks when used directly in JavaScript. If you need this data on the rendered page, consider placing it in the HTML portion (outside of a script tag). Alternatively, use a JavaScript-specific encoder, such as the one available in OWASP ESAPI. For Django, you may also consider using the 'json_script' template tag and retrieving the data in your script by using the element ID (e.g., `document.getElementById`).	high	open	pre-commit-diff	templates/base.html#L18
var-in-script-tag	Detected a template variable used in a script tag. Although template variables are HTML escaped, HTML escaping does not always prevent cross-site scripting (XSS) attacks when used directly in JavaScript. If you need this data on the rendered page, consider placing it in the HTML portion (outside of a script tag). Alternatively, use a JavaScript-specific encoder, such as the one available in OWASP ESAPI. For Django, you may also consider using the 'json_script' template tag and retrieving the data in your script by using the element ID (e.g., `document.getElementById`).	high	open	pre-commit-diff	templates/base.html#L19
var-in-script-tag	Detected a template variable used in a script tag. Although template variables are HTML escaped, HTML escaping does not always prevent cross-site scripting (XSS) attacks when used directly in	high	open	pre-commit-diff	templates/base.html#L24

Finding Title	Finding Description & Remediation	severity	status	ref	location
	JavaScript. If you need this data on the rendered page, consider placing it in the HTML portion (outside of a script tag). Alternatively, use a JavaScript-specific encoder, such as the one available in OWASP ESAPI. For Django, you may also consider using the 'json_script' template tag and retrieving the data in your script by using the element ID (e.g., `document.getElementById`).				
var-in-script-tag	Detected a template variable used in a script tag. Although template variables are HTML escaped, HTML escaping does not always prevent cross-site scripting (XSS) attacks when used directly in JavaScript. If you need this data on the rendered page, consider placing it in the HTML portion (outside of a script tag). Alternatively, use a JavaScript-specific encoder, such as the one available in OWASP ESAPI. For Django, you may also consider using the 'json_script' template tag and retrieving the data in your script by using the element ID (e.g., `document.getElementById`).	high	open	pre-commit-diff	templates/base.html#L25
command-injection-os-system	Request data detected in os.system. This could be vulnerable to a command injection and should be avoided. If this must be done, use the 'subprocess' module instead and pass the arguments as a list. See https://owasp.org/www-community/attacks/Command_Injection for more information.	high	open	pre-commit-diff	vulns/sql_injection/sql_injection_login.py#L49
command-injection-os-system	Request data detected in os.system. This could be vulnerable to a command injection and should be avoided. If this must be done, use the 'subprocess' module instead and pass the arguments as a list. See	high	open	pre-commit-diff	vulns/sql_injection/sql_injection_login.py#L54

Finding Title	Finding Description & Remediation	severity	status	ref	location
	https://owasp.org/www-community/attacks/Command_Injection for more information.				
subprocess-injection	Detected user input entering a `subprocess` call unsafely. This could result in a command injection vulnerability. An attacker could use this vulnerability to execute arbitrary commands on the host, which allows them to download malware, scan sensitive data, or run any command they wish on the server. Do not let users choose the command to run. In general, prefer to use Python API versions of system commands. If you must use subprocess, use a dictionary to allowlist a set of commands.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L36
tainted-sql-string	Detected user input used to manually construct a SQL string. This is usually bad practice because manual construction could accidentally result in a SQL injection. An attacker could use a SQL injection to steal or modify contents of the database. Instead, use a parameterized query which is available by default in most database engines. Alternatively, consider using the Django object-relational mappers (ORM) instead of raw SQL queries.	high	open	pre-commit-diff	vulns/sql_injection/sql_injection_login.py#L21
tainted-sql-string	Detected user input used to manually construct a SQL string. This is usually bad practice because manual construction could accidentally result in a SQL injection. An attacker could use a SQL injection to steal or modify contents of the database. Instead, use a parameterized query which is available by default in most database engines. Alternatively, consider using the Django object-relational mappers (ORM) instead of raw SQL queries.	high	open	pre-commit-diff	vulns/sql_injection/sql_injection_search.py#L7

Finding Title	Finding Description & Remediation	severity	status	ref	location
os-system-injection	User data detected in os.system. This could be vulnerable to a command injection and should be avoided. If this must be done, use the 'subprocess' module instead and pass the arguments as a list.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L15
os-system-injection	User data detected in os.system. This could be vulnerable to a command injection and should be avoided. If this must be done, use the 'subprocess' module instead and pass the arguments as a list.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L21
os-system-injection	User data detected in os.system. This could be vulnerable to a command injection and should be avoided. If this must be done, use the 'subprocess' module instead and pass the arguments as a list.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L27
dangerous-os-exec	Found user controlled content when spawning a process. This is dangerous because it allows a malicious actor to execute commands.	high	open	pre-commit-diff	vuln-1.py#L11
dangerous-subprocess-use	Detected subprocess function 'a' with user controlled data. A malicious actor could leverage this to perform command injection. You may consider using 'shlex.escape()'.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L36
dangerous-system-call	Found user-controlled data used in a system call. This could allow a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	reviewing	pre-commit-diff	vulns/file_upload/file_upload.py#L31
dangerous-system-call	Found user-controlled data used in a system call. This could allow a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier	high	open	pre-commit-diff	vulns/file_upload/file_upload.py#L49

Finding Title	Finding Description & Remediation	severity	status	ref	location
	to use without accidentally exposing a command injection vulnerability.				
dangerous-system-call	Found user-controlled data used in a system call. This could allow a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L15
dangerous-system-call	Found user-controlled data used in a system call. This could allow a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L21
dangerous-system-call	Found user-controlled data used in a system call. This could allow a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L27
dangerous-system-call	Found user-controlled data used in a system call. This could allow a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/sql_injection/sql_injection_login.py#L50
dangerous-system-call	Found user-controlled data used in a system call. This could allow a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/sql_injection/sql_injection_login.py#L55
		high	open		vuln-1.py#L11

Finding Title	Finding Description & Remediation	severity	status	ref	location
dangerous-os-exec-audit	Found dynamic content when spawning a process. This is dangerous if external data can reach this function call because it allows a malicious actor to execute commands. Ensure no external data reaches here.			pre-commit-diff	
dangerous-subprocess-use-audit	Detected subprocess function 'run' without a static string. If this data can be controlled by a malicious actor, it may be an instance of command injection. Audit the use of this call to ensure it is not controllable by an external resource. You may consider using 'shlex.escape()'.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L36
dangerous-system-call-audit	Found dynamic content used in a system call. This is dangerous if external data can reach this function call because it allows a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/file_upload/file_upload.py#L31
dangerous-system-call-audit	Found dynamic content used in a system call. This is dangerous if external data can reach this function call because it allows a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/file_upload/file_upload.py#L49
dangerous-system-call-audit	Found dynamic content used in a system call. This is dangerous if external data can reach this function call because it allows a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L15

Finding Title	Finding Description & Remediation	severity	status	ref	location
dangerous-system-call-audit	Found dynamic content used in a system call. This is dangerous if external data can reach this function call because it allows a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L21
dangerous-system-call-audit	Found dynamic content used in a system call. This is dangerous if external data can reach this function call because it allows a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/semgrep_vulns.py#L27
dangerous-system-call-audit	Found dynamic content used in a system call. This is dangerous if external data can reach this function call because it allows a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/sql_injection/sql_injection_login.py#L50
dangerous-system-call-audit	Found dynamic content used in a system call. This is dangerous if external data can reach this function call because it allows a malicious actor to execute commands. Use the 'subprocess' module instead, which is easier to use without accidentally exposing a command injection vulnerability.	high	open	pre-commit-diff	vulns/sql_injection/sql_injection_login.py#L55
subprocess-shell-true	Found 'subprocess' function 'call' with 'shell=True'. This is dangerous because this call will spawn the command using a shell process. Doing so propagates current shell settings and variables, which makes it	high	open	pre-commit-diff	secretstest.py#L19

Finding Title	Finding Description & Remediation	severity	status	ref	location
	much easier for a malicious actor to execute commands. Use 'shell=False' instead.				
dangerous-subprocess-use-audit	Detected subprocess function 'call' without a static string. If this data can be controlled by a malicious actor, it may be an instance of command injection. Audit the use of this call to ensure it is not controllable by an external resource. You may consider using 'shlex.escape()'.	high	open	pre-commit-diff	secretstest.py#L19

Findings Summary- Medium Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location
missing-integrity	This tag is missing an 'integrity' subresource integrity attribute. The 'integrity' attribute allows for the browser to verify that externally hosted files (for example from a CDN) are delivered without unexpected manipulation. Without this attribute, if an attacker can modify the externally hosted resource, this could lead to XSS and other types of attacks. To prevent this, include the base64-encoded cryptographic hash of the resource (file) youâ€™re telling the browser to fetch in the 'integrity' attribute for all externally hosted files.	medium	open	pre-commit-diff	semgrep_sast_findings_bad-python-app_20231201-0103.html#L4
missing-integrity	This tag is missing an 'integrity' subresource integrity attribute. The 'integrity' attribute	medium	open	pre-commit-diff	semgrep_sast_findings_bad-python-app_20231201-0103.html#L247

Finding Title	Finding Description & Remediation	severity	status	ref	location
	allows for the browser to verify that externally hosted files (for example from a CDN) are delivered without unexpected manipulation. Without this attribute, if an attacker can modify the externally hosted resource, this could lead to XSS and other types of attacks. To prevent this, include the base64-encoded cryptographic hash of the resource (file) youâ€™re telling the browser to fetch in the 'integrity' attribute for all externally hosted files.				
missing-integrity	This tag is missing an 'integrity' subresource integrity attribute. The 'integrity' attribute allows for the browser to verify that externally hosted files (for example from a CDN) are delivered without unexpected manipulation. Without this attribute, if an	medium	open	pre-commit-diff	semgrep_sast_findings_bad-python-app_20231201-0103.html#L753

Finding Title	Finding Description & Remediation	severity	status	ref	location
	attacker can modify the externally hosted resource, this could lead to XSS and other types of attacks. To prevent this, include the base64-encoded cryptographic hash of the resource (file) youâ€™re telling the browser to fetch in the 'integrity' attribute for all externally hosted files.				
missing-integrity	This tag is missing an 'integrity' subresource integrity attribute. The 'integrity' attribute allows for the browser to verify that externally hosted files (for example from a CDN) are delivered without unexpected manipulation. Without this attribute, if an attacker can modify the externally hosted resource, this could lead to XSS and other types of attacks. To prevent this, include the base64-encoded cryptographic hash of the resource	medium	open	pre-commit-diff	semgrep_sast_findings_bad-python-app_20231201-0103.html#L783

Finding Title	Finding Description & Remediation	severity	status	ref	location
	(file) youâ€™re telling the browser to fetch in the 'integrity' attribute for all externally hosted files.				
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).	medium	open	pre-commit-diff	vuln-main-10.java#L15
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to	medium	open	pre-commit-diff	vuln-main-10.java#L46

Finding Title	Finding Description & Remediation	severity	status	ref	location
	securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).				
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).	medium	open	pre-commit-diff	vuln-main-2.java#L15
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to	medium	open	pre-commit-diff	vuln-main-2.java#L46

Finding Title	Finding Description & Remediation	severity	status	ref	location
	securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).				
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).	medium	open	pre-commit-diff	vuln-main-3.java#L15
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to	medium	open	pre-commit-diff	vuln-main-3.java#L46

Finding Title	Finding Description & Remediation	severity	status	ref	location
	securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).				
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).	medium	open	pre-commit-diff	vuln-main-4.java#L15
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to	medium	open	pre-commit-diff	vuln-main-4.java#L46

Finding Title	Finding Description & Remediation	severity	status	ref	location
	securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).				
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).	medium	open	pre-commit-diff	vuln-main-7.java#L15
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to	medium	open	pre-commit-diff	vuln-main-7.java#L46

Finding Title	Finding Description & Remediation	severity	status	ref	location
	securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).				
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).	medium	open	pre-commit-diff	vuln-main-9.java#L15
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to	medium	open	pre-commit-diff	vuln-main-9.java#L46

Finding Title	Finding Description & Remediation	severity	status	ref	location
	securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).				
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).	medium	open	pre-commit-diff	vuln-main.java#L15
java-jwt-hardcoded-secret	A hard-coded credential was detected. It is not recommended to store credentials in source-code, as this risks secrets being leaked and used by either an internal or external malicious adversary. It is recommended to use environment variables to	medium	open	pre-commit-diff	vuln-main.java#L46

Finding Title	Finding Description & Remediation	severity	status	ref	location
	securely provide credentials or retrieve credentials from a secure vault or HSM (Hardware Security Module).				
var-in-href	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. If using a relative URL, start with a literal forward slash and concatenate the URL, like this: href='/ {{link}}'. You may also consider setting the Content Security Policy (CSP) header.	medium	open	pre-commit-diff	templates/components/navbar.html#L16
var-in-href	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. If using a relative URL, start with a literal	medium	open	pre-commit-diff	templates/components/navbar.html#L26

Finding Title	Finding Description & Remediation	severity	status	ref	location
	forward slash and concatenate the URL, like this: href='/{{link}}'. You may also consider setting the Content Security Policy (CSP) header.				
[var-in-href]()	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. If using a relative URL, start with a literal forward slash and concatenate the URL, like this: href='/{{link}}'. You may also consider setting the Content Security Policy (CSP) header.	medium	open	pre-commit-diff	[templates/components/navbar.html#L29]()
[var-in-href]()	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks.	medium	open	pre-commit-diff	[templates/components/navbar.html#L41]()

Finding Title	Finding Description & Remediation	severity	status	ref	location
	If using a relative URL, start with a literal forward slash and concatenate the URL, like this: href='/{{link}}'. You may also consider setting the Content Security Policy (CSP) header.				
var-in-href	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. If using a relative URL, start with a literal forward slash and concatenate the URL, like this: href='/{{link}}'. You may also consider setting the Content Security Policy (CSP) header.	medium	open	pre-commit-diff	templates/components/navbar.html#L44
var-in-href	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and	medium	open	pre-commit-diff	templates/components/navbar.html#L54

Finding Title	Finding Description & Remediation	severity	status	ref	location
	is subject to cross- site scripting (XSS) attacks. If using a relative URL, start with a literal forward slash and concatenate the URL, like this: href='/{{link}}'. You may also consider setting the Content Security Policy (CSP) header.				
var-in-href	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. If using a relative URL, start with a literal forward slash and concatenate the URL, like this: href='/{{link}}'. You may also consider setting the Content Security Policy (CSP) header.	medium	open	pre-commit-diff	templates/components/navbar.html#L58
var-in-href	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a	medium	open	pre-commit-diff	templates/components/navbar.html#L62

Finding Title	Finding Description & Remediation	severity	status	ref	location
	malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. If using a relative URL, start with a literal forward slash and concatenate the URL, like this: href='/{{link}}'. You may also consider setting the Content Security Policy (CSP) header.				
var-in-href	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. If using a relative URL, start with a literal forward slash and concatenate the URL, like this: href='/{{link}}'. You may also consider setting the Content Security Policy (CSP) header.	medium	open	pre-commit-diff	templates/components/navbar.html#L66
var-in-href	Detected a template variable used in an	medium	open		templates/components/navbar.html#L70

Finding Title	Finding Description & Remediation	severity	status	ref	location
	anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. If using a relative URL, start with a literal forward slash and concatenate the URL, like this: href='/{{link}}'. You may also consider setting the Content Security Policy (CSP) header.			pre-commit-diff	
template-autoescape-off	Detected a template block where autoescaping is explicitly disabled with '{% autoescape off %}'. This allows rendering of raw HTML in this segment. Turn autoescaping on to prevent cross-site scripting (XSS). If you must do this, consider instead, using `mark_safe` in Python code.	medium	open	pre-commit-diff	semgrep_sast_findings_bad-python-app_20231201-0103.html#L458
template-autoescape-off	Detected a template block where autoescaping is explicitly	medium	open	pre-commit-diff	semgrep_sast_findings_bad-python-app_20231201-0103.html#L479

Finding Title	Finding Description & Remediation	severity	status	ref	location
	disabled with '{% autoescape off %}'. This allows rendering of raw HTML in this segment. Turn autoescaping on to prevent cross-site scripting (XSS). If you must do this, consider instead, using `mark_safe` in Python code.				
template-href-var	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. Use the 'url' template tag to safely generate a URL. You may also consider setting the Content Security Policy (CSP) header.	medium	open	pre-commit-diff	templates/components/navbar.html#L16
template-href-var	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site	medium	open	pre-commit-diff	templates/components/navbar.html#L26

Finding Title	Finding Description & Remediation	severity	status	ref	location
	scripting (XSS) attacks. Use the 'url' template tag to safely generate a URL. You may also consider setting the Content Security Policy (CSP) header.				
template-href-var	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. Use the 'url' template tag to safely generate a URL. You may also consider setting the Content Security Policy (CSP) header.	medium	open	pre-commit-diff	templates/components/navbar.html#L44
template-href-var	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. Use the 'url' template tag to safely generate a URL. You may also	medium	open	pre-commit-diff	templates/components/navbar.html#L54

Finding Title	Finding Description & Remediation	severity	status	ref	location
	consider setting the Content Security Policy (CSP) header.				
template-href-var	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. Use the 'url' template tag to safely generate a URL. You may also consider setting the Content Security Policy (CSP) header.	medium	open	pre-commit-diff	templates/components/navbar.html#L58
template-href-var	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. Use the 'url' template tag to safely generate a URL. You may also consider setting the Content Security Policy (CSP) header.	medium	open	pre-commit-diff	templates/components/navbar.html#L62

Finding Title	**Finding Description & Remediation**	**severity**	**status**	**ref**	**location**
template-href-var	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. Use the 'url' template tag to safely generate a URL. You may also consider setting the Content Security Policy (CSP) header.	medium	open	pre-commit-diff	templates/components/navbar.html#L66
django-no-csrf-token	Manually-created forms in django templates should specify a csrf_token to prevent CSRF attacks	medium	open	pre-commit-diff	templates/file_upload.html#L5
django-no-csrf-token	Manually-created forms in django templates should specify a csrf_token to prevent CSRF attacks	medium	open	pre-commit-diff	templates/idor/idor_login.html#L15
django-no-csrf-token	Manually-created forms in django templates should specify a csrf_token to prevent CSRF attacks	medium	open	pre-commit-diff	templates/ssrf.html#L9

Finding Title	Finding Description & Remediation	severity	status	ref	location
django-no-csrf-token	Manually-created forms in django templates should specify a csrf_token to prevent CSRF attacks	medium	open	pre-commit-diff	templates/xss-stored.html#L10
user-eval	Found user data in a call to 'eval'. This is extremely dangerous because it can enable an attacker to execute arbitrary remote code on the system. Instead, refactor your code to not use 'eval' and instead use a safe library for the specific functionality you need.	medium	open	pre-commit-diff	vulns/semgrep_vulns.py#L31
avoid_app_run_with_bad_host	Running flask app with host 0.0.0.0 could expose the server publicly.	medium	open	pre-commit-diff	vulns/sql_injection/sql_injection_login.py#L56
debug-enabled	Detected Flask app with debug=True. Do not deploy to production with this flag enabled as it will leak sensitive information. Instead, consider using Flask configuration variables or setting 'debug' using	medium	open	pre-commit-diff	vulns/sql_injection/sql_injection_login.py#L56

Finding Title	Finding Description & Remediation	severity	status	ref	location
	system environment variables.				
render-template-string	Found a template created with string formatting. This is susceptible to server-side template injection and cross-site scripting attacks.	medium	open	pre-commit-diff	middlewares.py#L16
secure-set-cookie	Found a Flask cookie with insecurely configured properties. By default the secure, httponly and samesite ar configured insecurely. cookies should be handled securely by setting \`secure=True\`, \`httponly=True\`, and \`samesite='Lax'\` in response.set_cookie(...). If these parameters are not properly set, your cookies are not properly protected and are at risk of being stolen by an attacker. Include the \`secure=True\`, \`httponly=True\`, \`samesite='Lax'\` arguments or set these to be true in the Flask configuration.	medium	open	pre-commit-diff	vulns/idor/idor.py#L33

Finding Title	Finding Description & Remediation	severity	status	ref	location
secure-set-cookie	Found a Flask cookie with insecurely configured properties. By default the secure, httponly and samesite ar configured insecurely. cookies should be handled securely by setting \`secure=True\`, \`httponly=True\`, and \`samesite='Lax'\` in response.set_cookie(...). If these parameters are not properly set, your cookies are not properly protected and are at risk of being stolen by an attacker. Include the \`secure=True\`, \`httponly=True\`, \`samesite='Lax'\` arguments or set these to be true in the Flask configuration.	medium	open	pre-commit-diff	vulns/idor/idor.py#L34
template-autoescape-off	Detected a segment of a Flask template where autoescaping is explicitly disabled with '{% autoescape off %}'. This allows rendering of raw HTML in this segment. Ensure no user data is rendered here, otherwise	medium	open	pre-commit-diff	templates/xss-reflected.html#L13

Finding Title	Finding Description & Remediation	severity	status	ref	location
	this is a cross-site scripting (XSS) vulnerability, or turn autoescape on.				
template-autoescape-off	Detected a segment of a Flask template where autoescaping is explicitly disabled with '{% autoescape off %}'. This allows rendering of raw HTML in this segment. Ensure no user data is rendered here, otherwise this is a cross-site scripting (XSS) vulnerability, or turn autoescape on.	medium	open	pre-commit-diff	templates/xss-stored.html#L29
dynamic-urllib-use-detected	Detected a dynamic value being used with urllib. urllib supports 'file://' schemes, so a dynamic value controlled by a malicious actor may allow them to read arbitrary files. Audit uses of urllib calls to ensure user data cannot control the URLs, or consider using the 'requests' library instead.	medium	open	pre-commit-diff	vulns/ssrf/ssrf.py#L35
eval-detected		medium	open		vulns/semgrep_vulns.py#L31

Finding Title	Finding Description & Remediation	severity	status	ref	location
	Detected the use of eval(). eval() can be dangerous if used to evaluate dynamic content. If this content can be input from outside the program, this may be a code injection vulnerability. Ensure evaluated content is not definable by external sources.			pre-commit-diff	
md5-used-as-password	It looks like MD5 is used as a password hash. MD5 is not considered a secure password hash because it can be cracked by an attacker in a short amount of time. Use a suitable password hashing function such as scrypt. You can use `hashlib.scrypt`.	medium	open	pre-commit-diff	vulns/idor/idor.py#L14
md5-used-as-password	It looks like MD5 is used as a password hash. MD5 is not considered a secure password hash because it can be cracked by an attacker in a short amount of time. Use a suitable password hashing function such as	medium	open	pre-commit-diff	vulns/sql_injection/sql_injection_login.py#L19

Finding Title	Finding Description & Remediation	severity	status	ref	location
	scrypt. You can use `hashlib.scrypt`.				
md5-used-as-password	It looks like MD5 is used as a password hash. MD5 is not considered a secure password hash because it can be cracked by an attacker in a short amount of time. Use a suitable password hashing function such as scrypt. You can use `hashlib.scrypt`.	medium	open	pre-commit-diff	vulns/sql_injection/sql_injection_login.py#L44
flask-duplicate-handler-name	Looks like `route_param_concat` is a flask function handler that registered to two different routes. This will cause a runtime error	medium	open	pre-commit-diff	vulns/semgrep_vulns.py#L17
pass-body-fn	`pass` is the body of function before_request. Consider removing this or raise NotImplementedError() if this is a TODO	medium	open	pre-commit-diff	app.py#L30
unspecified-open-encoding	Missing 'encoding' parameter. 'open()' uses device locale encodings by default, corrupting	medium	open	pre-commit-diff	semgrep_sast_findings_report_sh.py#L237

Finding Title	Finding Description & Remediation	severity	status	ref	location
	files with special characters. Specify the encoding to ensure cross-platform support when opening files in text mode (e.g. encoding="utf-8").				
unspecified-open-encoding	Missing 'encoding' parameter. 'open()' uses device locale encodings by default, corrupting files with special characters. Specify the encoding to ensure cross-platform support when opening files in text mode (e.g. encoding="utf-8").	medium	open	pre-commit-diff	semgrep_sast_findings_report_sh.py#L264
is-function-without-parentheses	Is "is_admin" a function or an attribute? If it is a function, you may have meant self.is_admin() because self.is_admin is always true.	medium	open	pre-commit-diff	db_models.py#L6
unchecked-subprocess-call	This is not checking the return value of this subprocess call; if it fails no exception will be raised. Consider subprocess.check_call() instead	medium	open	pre-commit-diff	secretstest.py#L19

Finding Title	Finding Description & Remediation	severity	status	ref	location
return-not-in-function	`return` only makes sense inside a function	medium	open	pre-commit-diff	secretstest.py#L20

Findings Summary- Low Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location