

Created Wed 8th Sep 2021 | Snapshot taken 5 days ago

PROJECT OWNER

+ Add a project owner

Issues 6

Search...

SEVERITY

☐ High0

☐ Medium6

☐ Low0

PRIORITY SCORE

Scored between 0 - 1000

STATUS

☒ Open6

☐ Ignored0

LANGUAGES

☐ Python6

Share your feedback!

6 of 6 issues

Group by noneSort by highest severity

M

Jinja2NoDefaultAutoEscape

SCORE588

SNYK CODE

```
105         self.resolve_template = jinja2.Template(RESOLV_TMPL.read_text())
106     except OSError as err:
107         _LOGGER.error("Can't read resolve.tmpl: %s", err)
108     try:
109         self.hosts_template = jinja2.Template(HOSTS_TMPL.read_text())
```

jinja2.Template

 is called with no autoescape argument (autoescaping is disabled by default). This increases the risk of Cross-Site Scripting (XSS) attacks.

supervisor/plugins/dns.py1 step in 1 file

Full details

M

Jinja2NoDefaultAutoEscape

SCORE588

SNYK CODE

```
101     async def load(self) -> None:
102         """Load DNS setup."""
103         # Initialize CoreDNS Template
104         try:
105             self.resolve_template = jinja2.Template(RESOLV_TMPL.read_text())
```

jinja2.Template

 is called with no autoescape argument (autoescaping is disabled by default). This increases the risk of Cross-Site Scripting (XSS) attacks.

supervisor/plugins/dns.py1 step in 1 file

Full details

M

insecureHash

SCORE538

SNYK CODE

```
11 def get_hash_from_repository(name: str) -> str:
12     """Generate a hash from repository."""
13     key = name.lower().encode()
14     return hashlib.sha1(key).hexdigest()[:8]
```

hashlib.sha1

 is insecure. Consider changing it to a secure hashing algorithm (e.g. SHA256).

supervisor/store/utils.py1 step in 1 file

Full details

M

insecureHash

SCORE538

SNYK CODE

```
143         range_args[group_name[2:]] = float(group_value)
144     if typ.startswith(_STR) or typ.startswith(_PASSWORD):
145         if typ.startswith(_PASSWORD) and value:
146             self.pwned.add(hashlib.sha1(str(value).encode()).hexdigest())
```

hashlib.sha1

 is insecure. Consider changing it to a secure hashing algorithm (e.g. SHA256).

supervisor/addons/options.py1 step in 1 file

Full details