secure.py Documentation

Release 0.2.1

Caleb Kinney

Apr 23, 2021

Contents:

secure.py is a lightweight package that adds optional security headers and cookie attributes for Python web frameworks.

CHAPTER 1

Supported Python web frameworks:

aiohttp, Bottle, CherryPy, Django, Falcon, Flask, hug, Masonite, Pyramid, Quart, Responder, Sanic, Starlette, Tornado

CHAPTER 2

Install

pip:

```
$ pip install secure
```

Pipenv:

```
$ pipenv install secure
```

After installing secure.py:

```
from secure import SecureHeaders, SecureCookie

secure_headers = SecureHeaders()
secure_cookie = SecureCookie()
```

CHAPTER 3

Documentation

3.1 Secure Headers

Security Headers are HTTP response headers that, when set, can enhance the security of your web application by enabling browser security policies.

You can assess the security of your HTTP response headers at securityheaders.com

Recommendations used by secure,py and more information regarding security headers can be found at the OWASP Secure Headers Project .

3.1.1 Server

Contain information about server software
Default Value: `NULL` *(obfuscate server information, not included by default)*

3.1.2 Strict-Transport-Security (HSTS)

Ensure application communication is sent over HTTPS
Default Value: `max-age=63072000; includeSubdomains`

3.1.3 X-Frame-Options (XFO)

Disable framing from different origins (clickjacking defense)
Default Value: `SAMEORIGIN`

3.1.4 X-XSS-Protection

Enable browser cross-site scripting filters
Default Value: `1; mode=block`

3.1.5 X-Content-Type-Options

Prevent MIME-sniffing
Default Value: `nosniff`

3.1.6 Content-Security-Policy (CSP)

Prevent cross-site injections
Default Value: `script-src 'self'; object-src 'self'` *(not included by default)**

3.1.7 Referrer-Policy

Enable full referrer if same origin, remove path for cross origin and disable referrer in unsupported browsers
Default Value: `no-referrer, strict-origin-when-cross-origin`

3.1.8 Cache-control / Pragma / Expires

Prevent cacheable HTTPS response
Default Value: `no-cache, no-store, must-revalidate, max-age=0` / `no-cache` / `0`

3.1.9 Feature-Policy

Disable browser features and APIs
Default Value: `accelerometer 'none'; ambient-light-sensor 'none'; autoplay 'none'; camera 'none'; encrypted-media 'none'; fullscreen 'none'; geolocation 'none'; gyroscope 'none'; magnetometer 'none'; microphone 'none'; midi 'none'; payment 'none'; picture-in-picture 'none'; speaker 'none'; sync-xhr 'none'; usb 'none'; vr 'none';` *(not included by default)*

Addtional information:

- The `Strict-Transport-Security` (HSTS) header will tell the browser to **only** utilize secure HTTPS connections for the domain, and in the default configuration, including all subdomains. The HSTS header requires trusted certificates and users will unable to connect to the site if using self-signed or expired certificates. The browser will honor the HSTS header for the time directed in the `max-age` attribute *(default = 1 year)*, and setting the `max-age` to `0` will disable an already set HSTS header. Use the `hsts=False` option to not include the HSTS header in Secure Headers.
- The `Content-Security-Policy` (CSP) header can break functionality and can (and should) be carefully constructed, use the `csp=True` option to enable default values.

3.1.10 Usage

```
secure_headers.framework(response)
```

Default HTTP response headers:

```
Strict-Transport-Security: max-age=63072000; includeSubdomains
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Referrer-Policy: no-referrer, strict-origin-when-cross-origin
Cache-control: no-cache, no-store, must-revalidate, max-age=0
Pragma: no-cache
Expires: 0
```

3.1.11 Options

You can toggle the setting of headers with default values by passing `True` or `False` and override default values by passing a string to the following options:

- `server` - set the Server header, e.g. `Server="Secure"` *(string / bool / SecurePolicies, default=False)*
- `hsts` - set the Strict-Transport-Security header *(string / bool / SecurePolicies, default=True)*
- `xfo` - set the X-Frame-Options header *(string / bool / SecurePolicies, default=True)*
- `xxp` - set the X-XSS-Protection header *(string / bool / SecurePolicies, default=True)*
- `content` - set the X-Content-Type-Options header *(string / bool / SecurePolicies, default=True)*
- `csp` - set the Content-Security-Policy *(string / bool / SecurePolicies, default=False)* *
- `referrer` - set the Referrer-Policy header *(string / bool / SecurePolicies, default=True)*
- `cache` - set the Cache-control and Pragma headers *(string / bool / SecurePolicies, default=True)*
- `feature` - set the Feature-Policy header *(SecurePolicies / string / bool / SecurePolicies, default=False)*

Example:

```
from secure import SecureHeaders

secure_headers = SecureHeaders(csp=True, hsts=False, xfo="DENY")

. . .

secure_headers.framework(response)
```

3.2 Secure Cookies

3.2.1 Path

The Path directive instructs the browser to only send the cookie if provided path exists in the URL.

3.2.2 Secure

The Secure flag instructs the browser to only send the cookie via HTTPS.

3.2.3 HttpOnly

The HttpOnly flag instructs the browser to not allow any client side code to access the cookie's contents.

3.2.4 SameSite

The SameSite flag directs the browser not to include cookies on certain cross-site requests. There are two values that can be set for the same-site attribute, lax or strict. The lax value allows the cookie to be sent via certain cross-site GET requests, but disallows the cookie on all POST requests. For example cookies are still sent on links `<a href="x">`, prerendering `<link rel="prerender" href="x"` and forms sent by GET requests `<form-method="get"...`, but cookies will not be sent via POST requests `<form-method="post"...`, images `<img src="x">` or iframes `<iframe src="x">`. The strict value prevents the cookie from being sent cross-site in any context. Strict offers greater security but may impede functionality. This approach makes authenticated CSRF attacks impossible with the strict flag and only possible via state changing GET requests with the lax flag.

3.2.5 Expires

The Expires attribute sets an expiration date for persistent cookies.

3.2.6 Usage

```
secure_cookie.framework(response, name="spam", value="eggs")
```

Default Set-Cookie HTTP response header:

```
Set-Cookie: spam=eggs; Path=/; secure; HttpOnly; SameSite=lax
```

3.2.7 Options

You can modify default cookie attribute values by passing the following options:

- `name` - set the cookie name *(string, No default value)*
- `value` - set the cookie value *(string, No default value)*
- `path` - set the Path attribute, e.g. `path="/secure"` *(string, default="/")*
- `secure` - set the Secure flag *(bool, default=True)*
- `httponly` - set the HttpOnly flag *(bool, default=True)*
- `samesite` - set the SameSite attribute, e.g. `SecureCookie.SameSite.LAX` *(bool / enum, options:* `SecureCookie.SameSite.STRICT`, `SecureCookie.SameSite.LAX` *or* `False`, *default=SecureCookie.SameSite.LAX)*
- `expires` - set the Expires attribute with the cookie expiration in hours, e.g. `expires=1` *(number / bool, default=False)*

You can also import the SameSite options enum from Secure, `from secure import SecureCookie, SameSite`

Example:

```
from secure import SecureCookie
secure_cookie = SecureCookie(expires=1, samesite=SecureCookie.SameSite.STRICT)

secure_cookie.framework(response, name="spam", value="eggs")
```

3.3 Policy Builder

3.3.1 CSP()

Directives: `base_uri(sources)`, `block_all_mixed_content()`, `connect_src(sources)`, `default_src(sources)`, `font_src(sources)`, `form_action(sources)`, `frame_ancestors(sources)`, `frame_src(sources)`, `img_src(sources)`, `manifest_src(sources)`, `media_src(sources)`, `object_src(sources)`, `plugin_types(types)`, `report_to(json_object)`, `report_uri(uri)`, `require_sri_for(values)`, `sandbox(values)`, `script_src(sources)`, `style_src(sources)`, `upgrade_insecure_requests()`, `worker_src(sources)`

Values(): `self_`, `none`, `unsafe_inline`, `unsafe_eval`, `strict_dynamic`, `nonce(nonce_value)`, `all` = "*"

Example:

```
csp_value = (
    SecurePolicies.CSP()
    .default_src(SecurePolicies.CSP().Values.none)
    .base_uri(SecurePolicies.CSP().Values.self_)
    .block_all_mixed_content()
    .connect_src(SecurePolicies.CSP().Values.self_, "api.spam.com")
    .frame_src(SecurePolicies.CSP().Values.none)
    .img_src(SecurePolicies.CSP().Values.self_, "static.spam.com")
)

# default-src 'none'; base-uri 'self'; block-all-mixed-content; connect-src 'self'␣
↪api.spam.com; frame-src 'none'; img-src 'self' static.spam.com
```

You can check the effectiveness of your CSP Policy at the CSP Evaluator

3.3.2 HSTS()

Directives: `include_subDomains()`, `max_age(seconds)`, `preload()`

Example:

```
hsts_value = (
    SecurePolicies.HSTS()
    .include_subdomains()
    .preload()
    .max_age(SecurePolicies.Seconds.one_month)
)

# includeSubDomains; preload; max-age=2592000
```

3.3.3 XXP()

Directives: `disabled()` = 0, `enabled()` = 1, `enabled_block()` = 1; mode=block, `enabled_report(uri)` = 1; report=uri

Example:

```
xxp_value = SecurePolicies.XXP().enabled_block()

# 1; mode=block
```

3.3.4 XFO()

Directives: `allow_from(uri)`, `deny()`, `sameorigin()`

Example:

```
xfo_value = SecurePolicies.XFO().deny()

# deny
```

3.3.5 Referrer()

Directives: `no_referrer()`, `no_referrer_when_downgrade()`, `origin()`, `origin_when_cross_origin()`, `same_origin()`, `strict_origin()`, `strict_origin_when_cross_origin()`, `unsafe_url()`

Example:

```
referrer_value = SecurePolicies.Referrer().no_referrer()

# no-referrer
```

3.3.6 Feature()

Directives: `accelerometer(allowlist)`, `ambient_light_sensor(allowlist)`, `autoplay(allowlist)`, `camera(allowlist)`, `document_domain(allowlist)`, `encrypted_media(allowlist)`, `fullscreen(allowlist)`, `geolocation(allowlist)`, `gyroscope(allowlist)`, `magnetometer(allowlist)`, `microphone(allowlist)`, `midi(allowlist)`, `payment(allowlist)`, `picture_in_picture(allowlist)`, `speaker(allowlist)`, `sync_xhr(allowlist)`, `usb(allowlist)`, `Values(allowlist)`, `vr(allowlist)`

Values(): `self_`, `none`, `src`, `all_` = "*"

Example:

```
feature_value = (
    SecurePolicies.Feature()
    .geolocation(SecurePolicies.Feature.Values.self_, "spam.com")
    .vibrate(SecurePolicies.Feature.Values.none)
)

# geolocation 'self' spam.com; vibrate 'none'
```

3.3.7 Cache()

Directives: `immutable()`, `max_age(seconds)`, `max_stale(seconds)`, `min_fresh(seconds)`, `must_revalidate()`, `no_cache()`, `no_store()`, `no_transform()`, `only_if_cached()`, `private()`, `proxy_revalidate()`, `public()`, `s_maxage(seconds)`, `stale_if_error(seconds)`,`stale_while_revalidate(seconds)`,

Example:

```
cache_value = SecurePolicies.Cache().no_store().must_revalidate().proxy_revalidate()

# no-store, must-revalidate, proxy-revalidate
```

3.3.8 Seconds

Values: `five_minutes` = "300", `one_week` = "604800", `one_month` = "2592000", `one_year` = "31536000", `two_years` = "63072000"

3.3.9 Usage

Example:

```
from sanic import Sanic
from secure import SecureHeaders, SecurePolicies

csp_value = (
    SecurePolicies.CSP()
    .default_src(SecurePolicies.CSP().Values.none)
    .base_uri(SecurePolicies.CSP().Values.self_)
    .block_all_mixed_content()
    .connect_src(SecurePolicies.CSP().Values.self_, "api.spam.com")
    .frame_src(SecurePolicies.CSP().Values.none)
    .img_src(SecurePolicies.CSP().Values.self_, "static.spam.com")
)

hsts_value = (
    SecurePolicies.HSTS()
    .include_subdomains()
    .preload()
    .max_age(SecurePolicies.Seconds.one_month)
)

xxp_value = SecurePolicies.XXP().enabled_block()

xfo_value = SecurePolicies.XFO().deny()

referrer_value = SecurePolicies.Referrer().no_referrer()

feature_value = (
    SecurePolicies.Feature()
    .geolocation(SecurePolicies.Feature.Values.self_, "spam.com")
    .vibrate(SecurePolicies.Feature.Values.none)
)

cache_value = SecurePolicies.Cache().no_store().must_revalidate().proxy_revalidate()
```

(continues on next page)

(continued from previous page)

```
secure_headers = SecureHeaders(
    csp=csp_value,
    hsts=hsts_value,
    xfo=xfo_value,
    xxp=xxp_value,
    referrer=referrer_value,
    feature=feature_value,
    cache=cache_value,
)
secure_cookie = SecureCookie()

app = Sanic()

. . .

@app.middleware("response")
async def set_secure_headers(request, response):
    secure_headers.sanic(response)

. . .
```

Response Headers:

```
Strict-Transport-Security: includeSubDomains; preload; max-age=2592000
X-Frame-Options: deny
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Content-Security-Policy: default-src 'none'; base-uri 'self'; block-all-mixed-content;
↪ connect-src 'self' api.spam.com; frame-src 'none'; img-src 'self' static.spam.com
Referrer-Policy: no-referrer
Cache-control: no-store, must-revalidate, proxy-revalidate
Feature-Policy: geolocation 'self' spam.com; vibrate 'none'
```

3.4 Supported Frameworks

3.4.1 Framework Agnostic

Return Dictionary of Headers:
`secure_headers.headers()`

Example:

```
secure_headers.headers(csp=True, feature=True)
```

Return Value:

```
{'Strict-Transport-Security': 'max-age=63072000; includeSubdomains',
'X-Frame-Options': 'SAMEORIGIN', 'X-XSS-Protection': '1; mode=block',
'X-Content-Type-Options': 'nosniff', 'Content-Security-Policy':
"script-src 'self'; object-src 'self'", 'Referrer-Policy': 'no-referrer,
strict-origin-when-cross-origin', 'Cache-control': 'no-cache, no-store,
must-revalidate', 'Pragma': 'no-cache', 'Feature-Policy': "accelerometer
```

```
'none'; ambient-light-sensor 'none'; autoplay 'none'; camera 'none';
encrypted-media 'none';fullscreen 'none'; geolocation 'none'; gyroscope
'none'; magnetometer 'none'; microphone 'none'; midi 'none';payment 'none';
picture-in-picture 'none'; speaker 'none'; sync-xhr 'none'; usb 'none'; vr
'none';"}
```

3.4.2 aiohttp

Headers

```
secure_headers.aiohttp(resp)
```

Example:

```
from aiohttp import web
from aiohttp.web import middleware
from secure import SecureHeaders

secure_headers = SecureHeaders()

. . .

@middleware
async def set_secure_headers(request, handler):
    resp = await handler(request)
    secure_headers.aiohttp(resp)
    return resp

. . .

app = web.Application(middlewares=[set_secure_headers])

. . .
```

Cookies

```
secure_cookie.aiohttp(resp, name="spam", value="eggs")
```

Example:

```
from aiohttp import web
from secure import SecureCookie

secure_cookie = SecureCookie()

. . .

@routes.get("/secure")
async def set_secure_cookie(request):
    resp = web.Response(text="Secure")
    secure_cookie.aiohttp(resp, name="spam", value="eggs")
    return resp

. . .
```

3.4.3 Bottle

Headers

`secure_headers.bottle(response)`

Example:

```
from bottle import route, run, response, hook
from secure import SecureHeaders

secure_headers = SecureHeaders()

. . .

@hook("after_request")
def set_secure_headers():
    secure_headers.bottle(response)

. . .
```

Cookies

```
secure_cookie.bottle(response, name="spam", value="eggs")
```

Example:

```
from bottle import route, run, response, hook
from secure import SecureCookie

secure_cookie = SecureCookie()

. . .

@route("/secure")
def set_secure_cookie():
    secure_cookie.bottle(response, name="spam", value="eggs")
    return "Secure"

. . .
```

3.4.4 CherryPy

Headers

`"tools.response_headers.headers": secure_headers.cherrypy()`

Example:

CherryPy Application Configuration:

```
import cherrypy
from secure import SecureHeaders

```

(continues on next page)

(continued from previous page)

```
secure_headers = SecureHeaders()

. . .

config = {
    "/": {
        "tools.response_headers.on": True,
        "tools.response_headers.headers": secure_headers.cherrypy(),
    }
}

. . .
```

Cookies

```
response_headers = cherrypy.response.headers
secure_cookie.cherrypy(response_headers, name="spam", value="eggs")
```

Example:

```
import cherrypy
from secure import SecureCookie

secure_cookie = SecureCookie()

. . .

class SetSecureCookie(object):
    @cherrypy.expose
    def set_secure_cookie(self):
        response_headers = cherrypy.response.headers
        secure_cookie.cherrypy(response_headers, name="spam", value="eggs")
        return "Secure"

. . .
```

3.4.5 Django

Headers

```
secure_headers.django(response)
```

Example:

Django Middleware Documentation:

```
# securemiddleware.py
from secure import SecureHeaders

secure_headers = SecureHeaders()

. . .
```

(continues on next page)

(continued from previous page)

```
def set_secure_headers(get_response):
    def middleware(request):
        response = get_response(request)
        secure_headers.django(response)
        return response

    return middleware

. . .
```

```
# settings.py

...

MIDDLEWARE = [
    'app.securemiddleware.set_secure_headers'
]

...
```

Cookies

```
secure_cookie.django(response, name="spam", value="eggs")
```

Example:

```
from django.http import HttpResponse
from secure import SecureCookie

secure_cookie = SecureCookie()

. . .

def set_secure_cookie(request):
    response = HttpResponse("Secure")
    secure_cookie.django(response, name="spam", value="eggs")
    return response

. . .
```

3.4.6 Falcon

Headers

```
secure_headers.falcon(resp)
```

Example:

```
import falcon
from secure import SecureHeaders

secure_headers = SecureHeaders()
```

(continues on next page)

(continued from previous page)

```
. . .

class SetSecureHeaders(object):
    def process_request(self, req, resp):
        secure_headers.falcon(resp)

. . .

app = api = falcon.API(middleware=[SetSecureHeaders()])

. . .
```

Cookies

```
secure_cookie.falcon(resp, name="spam", value="eggs")
```

Example:

```
import falcon
from secure import SecureCookie

secure_cookie = SecureCookie()

. . .

class SetSecureCookie(object):
    def on_get(self, req, resp):
        resp.body = "Secure"
        secure_cookie.falcon(resp, name="spam", value="eggs")

. . .
```

3.4.7 Flask

Headers

```
secure_headers.flask(response)
```

Example:

```
from flask import Flask, Response
from secure import SecureHeaders

secure_headers = SecureHeaders()

app = Flask(__name__)

. . .

@app.after_request
def set_secure_headers(response):
    secure_headers.flask(response)
    return response
```

(continues on next page)

(continued from previous page)

```
. . .
```

Cookies

```
secure_cookie.flask(resp, name="spam", value="eggs")
```

Example:

```
from flask import Flask, Response
from secure import SecureCookie

secure_cookie = SecureCookie()

. . .

@app.route("/secure")
def set_secure_cookie():
    resp = Response("Secure")
    secure_cookie.flask(resp, name="spam", value="eggs")
    return resp

. . .
```

3.4.8 hug

Headers

```
secure_headers.hug(response)
```

Example:

```
import hug
from secure import SecureHeaders

secure_headers = SecureHeaders()

 . . .

@hug.response_middleware()
def set_secure_headers(request, response, resource):
    secure_headers.hug(response)

 . . .
```

Cookies

```
secure_cookie.hug(response, name="spam", value="eggs")
```

Example:

```
import hug
from secure import SecureCookie

secure_cookie = SecureCookie()

 . . .

@hug.get("/secure")
def set_secure_cookie(response):
  secure_cookie.hug(response, name="spam", value="eggs")
  return "Secure"

 . . .
```

3.4.9 Masonite

Headers

`secure_headers.masonite(self.request)`

Example:

Masonite Middleware:

```
# SecureMiddleware.py

from masonite.request import Request

from secure import SecureHeaders

secure_headers = SecureHeaders()

class SecureMiddleware:
    def __init__(self, request: Request):

        self.request = request

    def before(self):
        secure_headers.masonite(self.request)

 . . .
```

```
 # middleware.py

 ...

HTTP_MIDDLEWARE = [
    SecureMiddleware,
]

 ...
```

Cookies

```
secure_headers.masonite(self.request)
```

Example:

```
 . . .

def show(self, view: View, request: Request, response: Response):
    secure_cookie.masonite(request, name="spam", value="eggs")
    return response.view('Secure')

 . . .
```

3.4.10 Pyramid

Headers

Pyramid Tween:

```
def set_secure_headers(handler, registry):
    def tween(request):
        response = handler(request)
        secure_headers.pyramid(response)
        return response

    return tween
```

Example:

```
from pyramid.config import Configurator
from pyramid.response import Response
from secure import SecureHeaders

secure_headers = SecureHeaders()

. . .

def set_secure_headers(handler, registry):
    def tween(request):
        response = handler(request)
        secure_headers.pyramid(response)
        return response

    return tween

. . .

config.add_tween(".set_secure_headers")

. . .
```

Cookies

```
response = Response("Secure")
secure_cookie.pyramid(response, name="spam", value="eggs")
```

Example:

```
from pyramid.config import Configurator
from pyramid.response import Response
from secure import SecureCookie

secure_cookie = SecureCookie()

. . .

def set_secure_cookie(request):
    response = Response("Secure")
    secure_cookie.pyramid(response, name="spam", value="eggs")
    return response

. . .
```

3.4.11 Quart

Headers

```
secure_headers.quart(response)
```

Example:

```
from quart import Quart, Response
from secure import SecureHeaders

secure_headers = SecureHeaders()

app = Quart(__name__)

. . .

@app.after_request
async def set_secure_headers(response):
    secure_headers.quart(response)
    return response

. . .
```

Cookies

```
secure_cookie.quart(resp, name="spam", value="eggs")
```

Example:

```
from quart import Quart, Response
from secure import SecureCookie

secure_cookie = SecureCookie()

app = Quart(__name__)

. . .

@app.route("/secure")
async def set_secure_cookie():
    resp = Response("Secure")
    secure_cookie.quart(resp, name="spam", value="eggs")
    return resp

. . .
```

3.4.12 Responder

Headers

```
secure_headers.responder(resp)
```

Example:

```
import responder
from secure import SecureHeaders

secure_headers = SecureHeaders()

api = responder.API()

. . .

@api.route(before_request=True)
def set_secure_headers(req, resp):
    secure_headers.responder(resp)

. . .
```

You should use Responder's built in HSTS and pass the `hsts=False` option.

Cookies

```
secure_cookie.responder(resp, name="spam", value="eggs")
```

Example:

```
import responder
from secure import SecureCookie

secure_cookie = SecureCookie()

api = responder.API()
```

(continues on next page)

(continued from previous page)

```
. . .

@api.route("/secure")
async def set_secure_cookie(req, resp):
    resp.text = "Secure"
    secure_cookie.responder(resp, name="spam", value="eggs")

. . .
```

3.4.13 Sanic

Headers

`secure_headers.sanic(response)`

Example:

```
from sanic import Sanic
from secure import SecureHeaders

secure_headers = SecureHeaders()

app = Sanic()

. . .

@app.middleware("response")
async def set_secure_headers(request, response):
    secure_headers.sanic(response)

. . .
```

Cookies

```
secure_cookie.sanic(response, name="spam", value="eggs")
```

Example:

```
from sanic import Sanic
from sanic.response import text
from secure import SecureCookie

secure_cookie = SecureCookie()

app = Sanic()

. . .

@app.route("/secure")
async def set_secure_cookie(request):
    response = text("Secure")
    secure_cookie.sanic(response, name="spam", value="eggs")
```

(continues on next page)

(continued from previous page)

```
    return response

. . .
```

To set Cross Origin Resource Sharing (CORS) headers, please see sanic-cors .

3.4.14 Starlette

Headers

```
secure_headers.starlette(response)
```

Example:

```
from starlette.applications import Starlette
import uvicorn
from secure import SecureHeaders

secure_headers = SecureHeaders()

app = Starlette()

. . .

@app.middleware("http")
async def set_secure_headers(request, call_next):
    response = await call_next(request)
    secure_headers.starlette(response)
    return response

. . .
```

Cookies

```
secure_cookie.starlette(response, name="spam", value="eggs")
```

Example:

```
from starlette.applications import Starlette
from starlette.responses import PlainTextResponse
import uvicorn
from secure import SecureHeaders, SecureCookie

secure_cookie = SecureCookie()

app = Starlette()

. . .

@app.route("/secure")
async def set_secure_cookie(request):
    response = PlainTextResponse("Secure")
    secure_cookie.starlette(response, name="spam", value="eggs")
```

(continues on next page)

(continued from previous page)

```
    return response

. . .
```

3.4.15 Tornado

Headers

`secure_headers.tornado(self)`

Example:

```
import tornado.ioloop
import tornado.web
from secure import SecureHeaders

secure_headers = SecureHeaders()

. . .

class BaseHandler(tornado.web.RequestHandler):
    def set_default_headers(self):
        secure_headers.tornado(self)

. . .
```

Cookies

```
secure_cookie.tornado(self, name="spam", value="eggs")
```

Example:

```
import tornado.ioloop
import tornado.web
from secure import SecureCookie

secure_cookie = SecureCookie()

. . .

class SetSecureCookie(BaseHandler):
    def get(self):
        secure_cookie.tornado(self, name="spam", value="eggs")
        self.write("Secure")

. . .
```

3.5 Resources

3.5.1 Frameworks

- aiohttp - Asynchronous HTTP client/server framework for asyncio and Python
- Bottle - A fast and simple micro-framework for python web-applications.
- CherryPy - A pythonic, object-oriented HTTP framework.
- Django - The Web framework for perfectionists with deadlines.
- Falcon - A bare-metal Python web API framework for building high-performance microservices, app backends, and higher-level frameworks.
- Flask - The Python micro framework for building web applications.
- hug - Embrace the APIs of the future. Hug aims to make developing APIs as simple as possible, but no simpler.
- Masonite - The Modern And Developer Centric Python Web Framework.
- Pyramid - A Python web framework
- Quart - A Python ASGI web microframework.
- Responder - A familiar HTTP Service Framework
- Sanic - An Async Python 3.5+ web server that's written to go fast
- Starlette - The little ASGI framework that shines.
- Tornado - A Python web framework and asynchronous networking library, originally developed at FriendFeed.

3.5.2 General

- OWASP - Secure Headers Project
- OWASP - Session Management Cheat Sheet
- Mozilla Web Security
- securityheaders.com

3.5.3 Policies

- **CSP:** CSP Cheat Sheet | Scott Helme, Content-Security-Policy | MDN, Content Security Policy Cheat Sheet | OWASP, Content Security Policy CSP Reference & Examples
- **XXP:** X-XSS-Protection | MDN
- **XFO:** X-Frame-Options | MDN
- **HSTS:** Strict-Transport-Security | MDN, HTTP Strict Transport Security Cheat Sheet | OWASP
- **Referrer:** A new security header: Referrer Policy | Scott Helme, Referrer-Policy | MDN
- **Feature:** A new security header: Feature Policy | Scott Helme, Feature-Policy | MDN, Introduction to Feature Policy | Google Developers
- **Cache:** Cache-Control | MDN

CHAPTER 4

Indices and tables

- genindex
- modindex
- search