✓ What payload did you use for the lab "HTML's are easy!" * 1/1

- ◯ "><h1>Hello World</h1>
- ◉ <h1>Hello World</h1> ✓
- ◯ "><h1>Hello World</h1>"<
- ◯ None of the above

✓ The "ALLOW-FROM" URI means * 1/1

- ◉ Permit the specified "uri" to frame this page ✓
- ◯ Allow from anyone except the URI mentioned
- ◯ Allow only images from the URI
- ◯ Allow only text from the URI

✓ ClickJacking on non-sensitive pages comes under which category? * 1/1

- ◉ P5 ✓
- ◯ P4
- ◯ P3
- ◯ P2

✓ The correct sequence of HTML tags for starting a webpage is * 1/1

- ○ HTML, Head, Body, Title,
- ◉ HTML, Head, Title, Body ✓
- ○ HTML, Body, Title, Head
- ○ Head, Title, HTML, body

✓ HTML Injection can be prevented by * 1/1

- ○ Checking if input contains tags or not
- ○ Sanitizing the input
- ○ Never trust user input
- ◉ All of the above ✓

✓ What payload did you use for the lab "File Content and HTML Injection a perfect pair!" * 1/1

- ○ A HTML file
- ○ An SVG file with HTML tags
- ○ A CSS file

All of the above ✓

✓ The impact of HTML Injection is * 1/1

- Phishing
- Social Engineering
- Stealing Credentials
- All of the above ✓

✓ ClickJacking is also known as ? * 1/1

- User Interface redress attack
- UI redressing
- Both A and B ✓
- None of the above

✓ What payload did you use for the lab "Injecting HTML using URL" * 1/1

- <h1>Hello</h1> in the URL
- ?<h1>Hello</h1> in the URL ✓
- !<h1>Hello</h1> in the URL
- None of the above

✓ HTML stands for ? * 1/1

- (●) HyperText Markup Language ✓
- ( ) HyperText and links Markup Language
- ( ) HighText Machine Language
- ( ) None of the above

✓ HTML Injection is exploited with? * 1/1

- ( ) Open Source Intelligence
- (●) Social Engineering ✓
- ( ) Remote Code Execution
- ( ) None of the Above

✓ ClickJacking on Logout and Contact form is sensitive * 1/1

- ( ) TRUE
- (●) FALSE ✓
- ( ) Maybe
- ( ) Cant Say

✓ _____ defines that this document is an HTML5 document * 1/1

- <html>
- (●) <!DOCTYPE html> ✓
- <!DOCUMENT html>
- <!DOCUMENT html5>

✓ If you find an HTML Injection there is a good chance of finding * 1/1

- (●) XSS ✓
- XXE
- CSRF
- MFLAC

✓ What website you would use in order to check if the website has the necessary headers or not * 1/1

- https://google.com
- (●) https://securityheaders.com ✓
- https://bing.com

All of the above

✓ Can Click Jacking be used to download a malware? * 1/1

- (●) True ✓
- ( ) False

✓ The impact of Click Jacking is * 1/1

- ( ) To gain followers on social media
- ( ) To gain RSS subscribers
- ( ) To transfer funds unknowingly from a victim
- (●) All of the above ✓

✓ What payload did you use for the lab "Let me Store them!" * 1/1

- (●) ">abc ✓
- ( ) abc
- ( ) <abc></b/
- ( ) <abc></h1>

✓ Which of the following is used to prevent Clickjacking? * 1/1

- ○ HTTPS Connection
- ◉ X-Frame-Options HTTP Header ✓
- ○ Content-Security-Policy HTTP Header
- ○ None of the above

✓ The CVSS score of HTML Injection is * 1/1

- ◉ 0.1 - 3.9 ✓
- ○ 4.0 - 6.9
- ○ 7.0 - 8.9
- ○ 9.0 - 10.0

✓ Which of the following should be checked to know if page is vulnerable to clickjacking? * 1/1

- ○ Content Security Policy
- ○ X-Content-Type-Options HTTP Header
- ◉ X-Frame-Options HTTP Header ✓

X-Powered-By

✓ What payload did you use for the lab "Encode IT!" * 1/1

ROT encode of the payload <h1>Hello World</h1>

Base64 encode of the payload <h1>Hello World</h1>

URL encode of the payload <h1>Hello World</h1> ✓

<h1>Hello World</h1>

✓ The severity of HTML Injection is * 1/1

P5

P4

P3 ✓

P2

✓ Which of the following should X-Frame-Options should be set to * 1/1

DENY

SAMEORIGIN ✓

All of the above

None of the above

✓ The severity of ClickJacking on sensitive pages is * 1/1

- ○ P5
- ◉ P3 ✓
- ○ P4
- ○ P2

✓ The recommended clickjacking protection is to incorporate the frame-ancestors in CSP. The value of frame-ancestors should be set to * 1/1

- ○ none
- ○ self
- ○ allow
- ◉ Both A and B ✓

✓ What payload did you use for the lab "File Names are also vulnerable!" * 1/1

- ◉ "><iframe src="malware_iframe.html">.txt ✓
- ○ <iframe src="malware_iframe.html">.txt
- ○ "><iframe src="malware_iframe.html".txt

"><iframe><iframe src="malware_iframe.html">.txt

✓ The Clickjacking vulnerability we saw in "Let's Hijack!" was to _____ * 1/1

- Delete User Account
- Login into Google Account
- Delete Admin account ✓
- Both A and C

✓ Which of the following might be an injection point for HTML Injection * 1/1

- ?profileId=
- ?search= ✓
- ?account=
- ?redirect=

✓ The Clickjacking vulnerability we saw in "Let's Re-Hijack!" was to _____ * 1/1

- Login into Google Account ✓
- Delete User Account
- Delete Admin account
- All of the above

This form was created inside of VT.

Forms