Cross-site scripting

LAB	Reflected XSS into HTML context with nothing encoded »	Solved
LAB	Reflected XSS into HTML context with most tags and attributes blocked »	Solved
LAB	Reflected XSS into HTML context with all tags blocked except custom ones »	Solved
LAB	Reflected XSS with event handlers and `href` attributes blocked »	Solved
LAB	Reflected XSS with some SVG markup allowed »	Solved
LAB	Reflected XSS into attribute with angle brackets HTML-encoded »	Solved
LAB	Stored XSS into anchor `href` attribute with double quotes HTML-encoded »	Solved
LAB	Reflected XSS in canonical link tag »	Solved

Lab: Reflected XSS into HTML context with nothing encoded

APPRENTICE

LAB Solved

This lab contains a simple reflected cross-site scripting vulnerability in the search functionality.

To solve the lab, perform a cross-site scripting attack that calls the `alert` function.

Access the lab

Solution

Community solutions

Lab: Reflected XSS into HTML context with most tags and attributes blocked

PRACTITIONER

LAB Solved

This lab contains a reflected cross-site scripting vulnerability in the search functionality but uses a web application firewall (WAF) to protect against common XSS vectors.

To solve the lab, perform a cross-site scripting attack that bypasses the WAF and alerts `document.cookie`.

Note

Your solution must not require any user interaction. Manually triggering an `alert` in your own browser will not solve the lab.

Access the lab

Solution

Lab: Reflected XSS into HTML context with all tags blocked except custom ones

PRACTITIONER

LAB Solved

This lab blocks all HTML tags except custom ones.

To solve the lab, perform a cross-site scripting attack that injects a custom tag and automatically alerts `document.cookie`.

Access the lab

Solution

Community solutions

Lab: Reflected XSS with event handlers and `href` attributes blocked

EXPERT

LAB Solved

This lab contains a reflected XSS vulnerability with some whitelisted tags, but all events and anchor `href` attributes are blocked..

To solve the lab, perform a cross-site scripting attack that injects a vector that, when clicked, calls the `alert` function.

Note that you need to label your vector with the word "Click" in order to induce the simulated lab user to click your vector. For example: `<a href="">Click me</a>`

Access the lab

Solution

Lab: Reflected XSS with some SVG markup allowed

PRACTITIONER

LAB Solved

This lab has a simple reflected XSS vulnerability. The site is blocking common tags but misses some SVG tags and events.

To solve the lab, perform a cross-site scripting attack that calls the `alert()` function.

Access the lab

Solution

Community solutions

Lab: Reflected XSS into attribute with angle brackets HTML-encoded

APPRENTICE

LAB Solved

This lab contains a reflected cross-site scripting vulnerability in the search blog functionality where angle brackets are HTML-encoded. To solve this lab, perform a cross-site scripting attack that injects an attribute and calls the `alert` function.

Access the lab

Solution

Community solutions

Lab: Stored XSS into anchor `href` attribute with double quotes HTML-encoded

APPRENTICE

LAB Solved

This lab contains a stored cross-site scripting vulnerability in the comment functionality. To solve this lab, submit a comment that calls the `alert` function when the comment author name is clicked.

Access the lab

Solution

Community solutions

Lab: Reflected XSS in canonical link tag

PRACTITIONER

LAB Solved

This lab reflects user input in a canonical link tag and escapes angle brackets.

To solve the lab, perform a cross-site scripting attack on the home page that injects an attribute that calls the `alert` function.

To assist with your exploit, you can assume that the simulated user will press the following key combinations:

- `ALT+SHIFT+X`
- `CTRL+ALT+X`
- `Alt+X`

Please note that the intended solution to this lab is only possible in Chrome.

Access the lab