

EETHM



PRACTICALS REPORT


REPORT ON ATTACKS

This report contains various web application vulnerabilities.

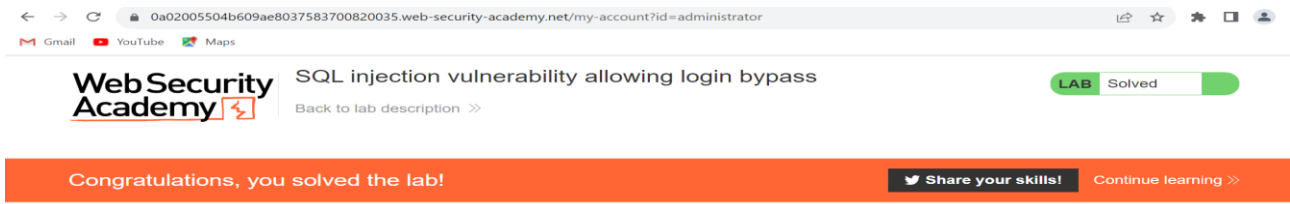
<https://portswigger.net/>

LABS

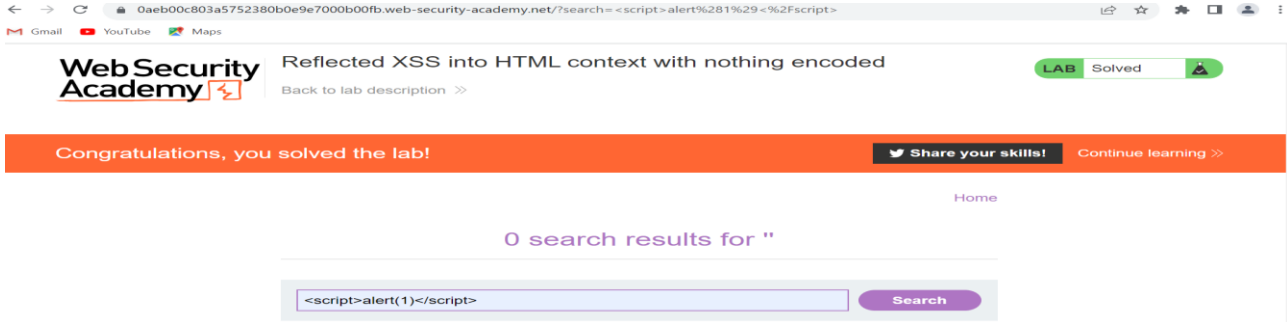
SQL Injection

Threat	SQL injection vulnerability in WHERE clause allowing retrieval of hidden data
VULNERABILITY SUMMARY: This result contains a SQL injection vulnerability in the product category filter in WHERE clause allowing retrieval of hidden data.	
IMPACT: This vulnerability allows user to get access to hidden data that are not meant for the public to access	
POC:  <p>https://0acf00f10443d8cb8132f77f004100ca.web-security-academy.net/</p> <p>open the above link and modify the category parameter, giving it the value '+OR+1=1--</p>	
CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N Exploitability Sub score: 3.9	
RECOMMENDED SOLUTION <ol style="list-style-type: none">1. Special characters must be restricted2. User accessibility must be restricted	

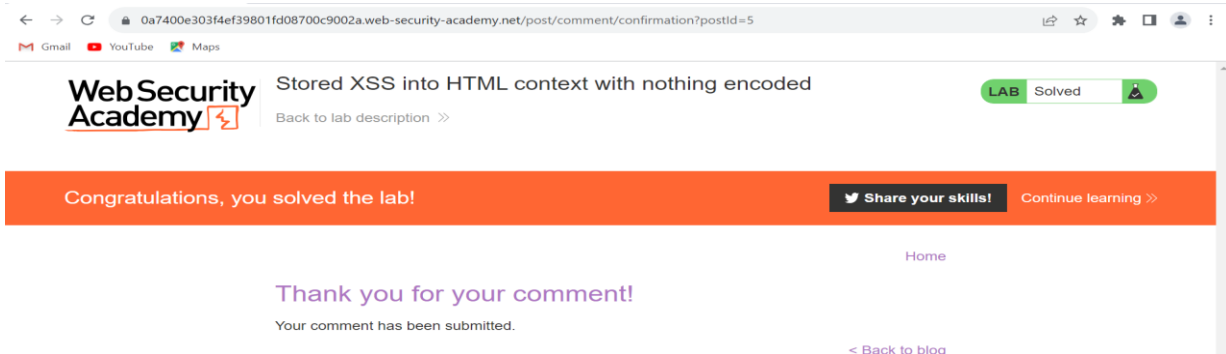
SQL Injection

Threat	SQL injection vulnerability allowing login bypass
VULNERABILITY SUMMARY: This result contains a SQL injection vulnerability in the login function	
IMPACT: This vulnerability contains a SQL injection vulnerability in the login function	
POC:  https://0a2100a004f9d8218289069100fc0082.web-security-academy.net/ open the above link and modify the log in, giving it the value administrator'--	
CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N Exploitability Subscore:2.8	
RECOMMENDED SOLUTION <ol style="list-style-type: none">1. Request must be set on timing2. Strong password must be administered3. Special characters must be set null	


Cross-site scripting

Threat	Reflected XSS into HTML context with nothing encoded.
VULNERABILITY SUMMARY: This report contains a simple reflected cross-site scripting vulnerability in the search functionality.	
IMPACT: This vulnerability contains a SQL injection vulnerability in the login function	
POC:  https://0ac200fc0443127d80a1851900b100ad.web-security-academy.net/ 1. Open the link above the Copy and paste the following into the search box: <code><script>alert(1)</script></code>	
CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N Exploitability Subscore:2.8	
RECOMMENDED SOLUTION <ol style="list-style-type: none">1. Request must be set on timing2. Strong password must be administered3. Special characters must be set null	


Cross-site scripting

Threat	<u>Stored XSS into HTML context with nothing encoded</u>
VULNERABILITY SUMMARY: This report contains a simple reflected cross-site scripting vulnerability in the search functionality.	
IMPACT: This vulnerability contains a SQL injection vulnerability in the login function	
POC:  https://0ac200fc0443127d80a1851900b100ad.web-security-academy.net/	
2. Open the link above the Copy and paste the following into the search box: <script>alert(1)</script>	
CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N Exploitability Subscore:2.8	
RECOMMENDED SOLUTION To prevent XSS attacks, your application must validate all the input data, make sure that only the allowlisted data is allowed, and ensure that all variable output in a page is encoded before it is returned to the user.	

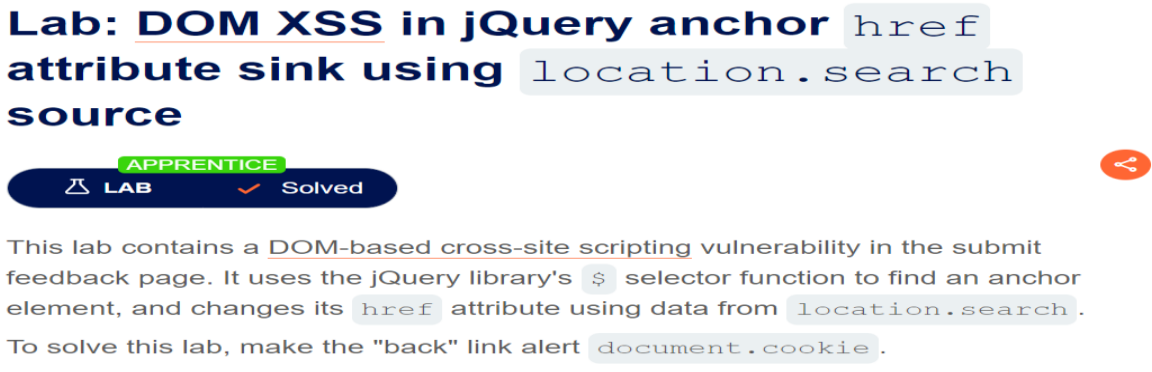
Cross-site scripting

Threat	DOM XSS in document.write sink using source location.search
VULNERABILITY SUMMARY: This report contains a DOM-based cross-site scripting vulnerability in the search query tracking functionality. It uses the JavaScript document.write function, which writes data out to the page. The document.write function is called with data from location.search, which you can control using the website URL.	
IMPACT: DOM-based cross-site scripting vulnerability in the search query tracking functionality. It uses the JavaScript document.write function, which writes data out to the page. The document.write function is called with data from location.search, which you can control using the website URL.	
POC:  https://0aa3004c04f3e47784f8131700d200e6.web-security-academy.net/ Break out of the img attribute by searching for: ><svg onload=alert(1)>	
CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N Exploitability Subscore:2.8	
RECOMMENDED SOLUTION To prevent XSS attacks, your application must validate all the input data, make sure that only the allowlisted data is allowed, and ensure that all variable output in a page is encoded before it is returned to the user.	

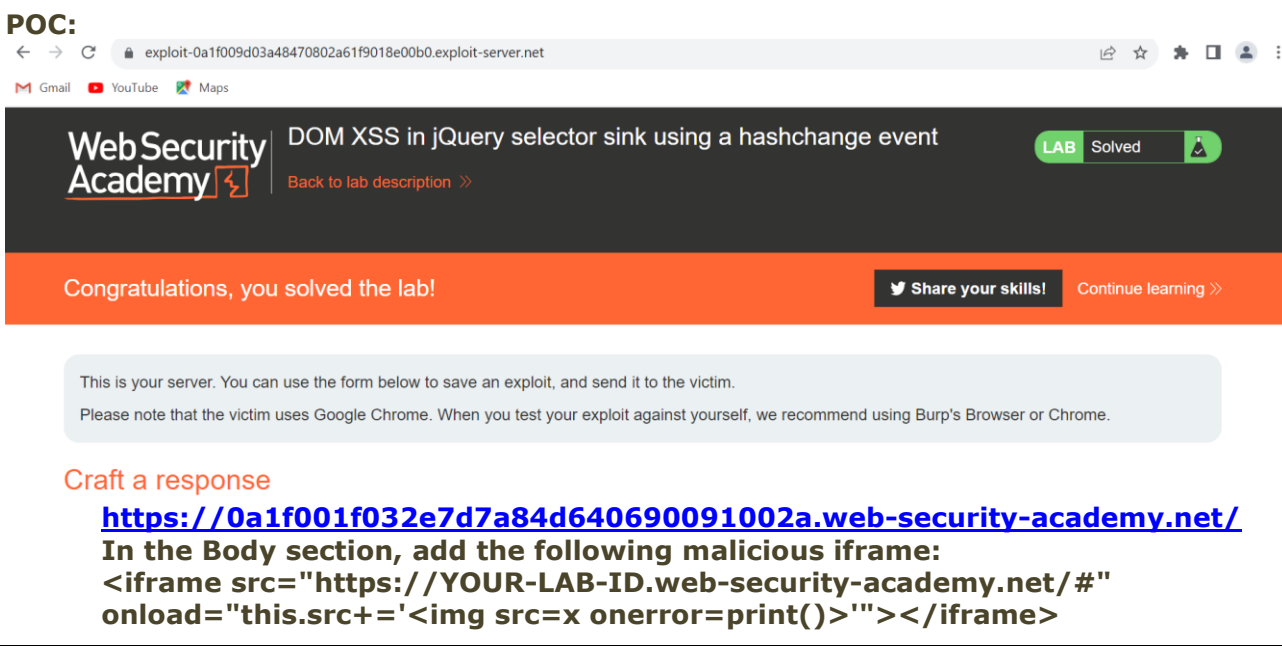
Cross-site scripting

Threat	DOM XSS in innerHTML sink using source location.search
VULNERABILITY SUMMARY: This report contains a DOM-based cross-site scripting vulnerability in the search blog functionality. It uses an innerHTML assignment, which changes the HTML contents of a div element, using data from location.search.	
IMPACT: DOM-based cross-site scripting vulnerability in the search blog functionality. It uses an innerHTML assignment, which changes the HTML contents of a div element, using data from location.search.	
POC:  https://0ab200570380856a81c9a22c003c000a.web-security-academy.net/ Enter the following into the into the search box: Click "Search".	
CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N Exploitability Subscore:2.8	
RECOMMENDED SOLUTION The primary rule that you must follow to prevent DOM XSS is: sanitize all untrusted data, even if it is only used in client-side scripts. If you have to use user input on your page, always use it in the text context, never as HTML tags or any other potential code. Avoid methods such as document.	

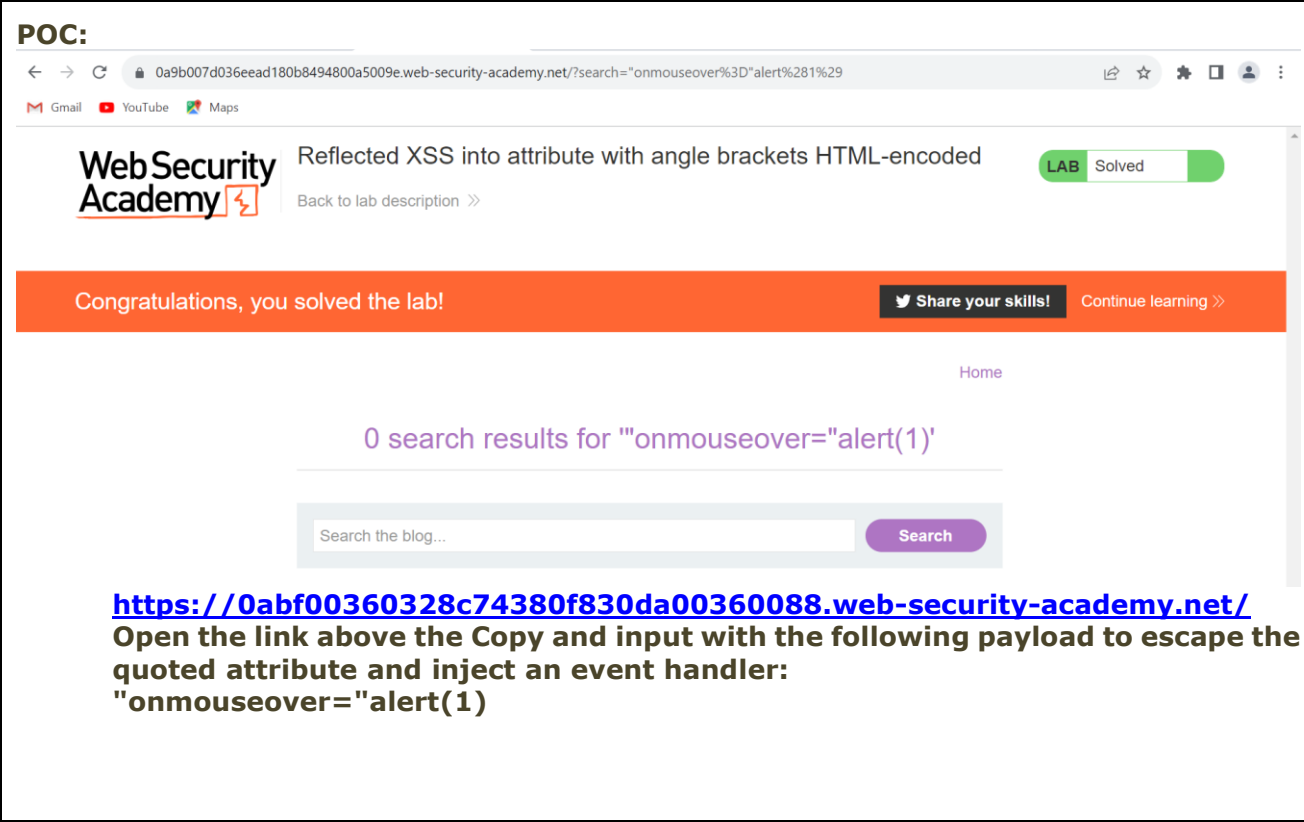
Cross-site scripting

Threat	DOM XSS in jQuery anchor href attribute sink using location.search source
VULNERABILITY SUMMARY: This report contains a simple reflected cross-site scripting vulnerability in the search functionality.	
IMPACT: This vulnerability contains a SQL injection vulnerability in the login function	
POC: Lab: <u>DOM XSS</u> in jQuery anchor href attribute sink using location.search source  https://0a1f001f032e7d7a84d640690091002a.web-security-academy.net/ In the Body section, add the following malicious iframe: <iframe src="https://YOUR-LAB-ID.web-security-academy.net/#" onload="this.src+="></iframe>	
CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N Exploitability Subscore:2.8	
RECOMMENDED SOLUTION To prevent XSS attacks, your application must validate all the input data, make sure that only the allowlisted data is allowed, and ensure that all variable output in a page is encoded before it is returned to the user.	




Cross-site scripting

Threat	DOM XSS in jQuery selector sink using a hashchange event
VULNERABILITY SUMMARY: This report contains a DOM-based cross-site scripting vulnerability on the home page. It uses jQuery's <code>\$()</code> selector function to auto-scroll to a given post, whose title is passed via the <code>location.hash</code> property.	
IMPACT: This report contains a DOM-based cross-site scripting vulnerability on the home page. It uses jQuery's <code>\$()</code> selector function to auto-scroll to a given post, whose title is passed via the <code>location.hash</code> property.	
POC:  <p>Craft a response</p> <p>https://0a1f001f032e7d7a84d640690091002a.web-security-academy.net/</p> <p>In the Body section, add the following malicious iframe:</p> <pre><iframe src="https://YOUR-LAB-ID.web-security-academy.net/#" onload="this.src+=''"></iframe></pre>	
CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N Exploitability Subscore:2.8	
RECOMMENDED SOLUTION The primary rule that you must follow to prevent DOM XSS is: sanitize all untrusted data, even if it is only used in client-side scripts.	

Cross-site scripting

Threat	Reflected XSS into attribute with angle brackets HTML-encoded
VULNERABILITY SUMMARY: This report contains a reflected cross-site scripting vulnerability in the search blog functionality where angle brackets are HTML-encoded.	
IMPACT: This lab contains a reflected cross-site scripting vulnerability in the search blog functionality where angle brackets are HTML-encoded allowing an attacker to insert payload through input.	
POC:  https://0abf00360328c74380f830da00360088.web-security-academy.net/ Open the link above the Copy and input with the following payload to escape the quoted attribute and inject an event handler: "onmouseover="alert(1)	
CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N Exploitability Subscore:3.9	
RECOMMENDED SOLUTION Filter input on arrival. At the point where user input is received, filter as strictly as possible based on what is expected or valid input. Encode data on output. ... Use appropriate response headers. ... Content Security Policy.	

Cross-site scripting

Threat	Stored XSS into anchor href attribute with double quotes HTML-encoded
VULNERABILITY SUMMARY: This lab contains a stored cross-site scripting vulnerability in the comment functionality	
IMPACT: This lab contains a stored cross-site scripting vulnerability in the comment functionality where an attacker can inject malicious payload into the database.	
POC: <div><h3>Lab: <u>Stored XSS</u> into anchor href attribute with double quotes HTML-encoded</h3><div><div>APPRENTICE</div><div> LAB  Solved</div><div></div></div><p>https://0a0b00ff0444d85f82ce60f800fc00f8.web-security-academy.net/ Open the above link and replace your input with the following payload to inject a JavaScript URL that calls alert: javascript:alert(1)</p></div>	
CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N Exploitability Subscore: 3.9	
RECOMMENDED SOLUTION The HTTP X-XSS-Protection response header is a feature of Internet Explorer, Chrome, and Safari that stops pages from loading when they detect reflected cross-site scripting (XSS) attacks.	

Cross-site scripting

Threat	Reflected XSS into a JavaScript string with angle brackets HTML encoded
VULNERABILITY SUMMARY: This report contains a reflected cross-site scripting vulnerability in the search query tracking functionality where angle brackets are encoded. The reflection occurs inside a JavaScript string	
IMPACT: This lab contains a reflected cross-site scripting vulnerability in the search query tracking functionality where angle brackets are encoded. The reflection occurs inside a JavaScript string.	
POC:	

Congratulations, you solved the lab!

[Share your skills!](#)[Continue learning >>](#)[Home](#)

0 search results for "-alert(1)-"

Search the blog...

Search

<https://0ac200fc0443127d80a1851900b100ad.web-security-academy.net/>

1. Open the link above and Replace your input with the following payload to break out of the JavaScript string and inject an alert:
'-alert(1)-'

CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N
Exploitability Subscore:2.8

RECOMMENDED SOLUTION

To prevent XSS attacks, your application must validate all the input data, make sure that only the allowlisted data is allowed, and ensure that all variable output in a page is encoded before it is returned to the user.

Cross-site request forgery (CSRF)

Threat	CSRF vulnerability with no defenses
VULNERABILITY SUMMARY: This report contains email change functionality is vulnerable to CSRF.	
IMPACT: This vulnerability contains to change the viewer's email address and upload it to your exploit server.	
POC: <div><div><div>Lab: <u>CSRF vulnerability with no defenses</u></div><div>APPRENTICE</div><div>LAB Solved</div><div></div></div><p>This lab's email change functionality is vulnerable to CSRF.</p><p>To solve the lab, craft some HTML that uses a <u>CSRF attack</u> to change the viewer's email address and upload it to your exploit server.</p><p>You can log in to your own account using the following credentials: <code>wiener:peter</code></p><p>https://0a45007903068e618424bd8800140004.web-security-academy.net/ Open the link above the Copy and paste the following into the search box:</p><pre><form method="POST" action="https://YOUR-LAB-ID.web-security-academy.net/my-account/change-email"> <input type="hidden" name="email" value="anything%40web-security-academy.net"> </form> <script> document.forms[0].submit(); </script></pre><p>Change the email and deliver to the target.</p></div>	
CVE, OWASP, CWE, REFERENCE – A3: AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:N Exploitability Subscore:2.8	
RECOMMENDED SOLUTION To prevent CSRF attacks on the server side, banks and merchants should transition from cookies that perform session-tracking to session tokens that are dynamically generated. This would make it more difficult for an attacker to get a hold of a client's session	

