jamk

A03:2021 – Injection

Issue Report

Daniil Lebedev, AD1375, TTV22S2

Assignment report
Web Application Security TTC6500 / Joonatan Ovaska
16.02.2025
School of Technology / Information and Communication Technology

jamk | Jyväskylän ammattikorkeakoulu
University of Applied Sciences

1 Cross-Site Scripting

1.1 Main Target – Stored XSS

Title: XSS Injection on create product in the wasdat.fi

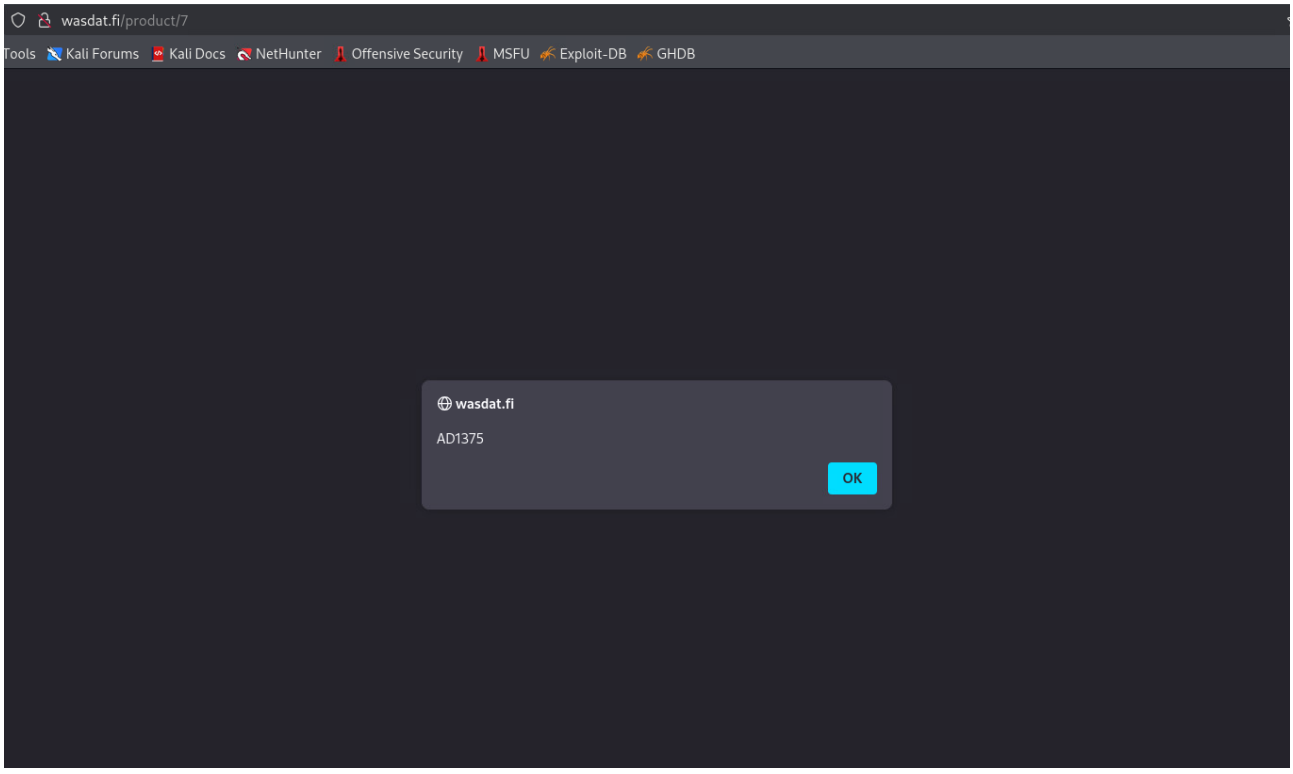
Description: I have found that if you inject your Java script in the Description when creating a new product, it triggers when you visit this product in the marketplace.

Steps to reproduce:

- Visit wasdat.fi and go to your profile, then select “Create your first product”.
- Put the payload on the Description box.

XSS Payload: <script>alert(‘AD1375’)</script>

- Visiting this product in the marketplace causes script to trigger (see Picture 1).



Picture 1

Impact estimation:

- Medium severity: This XSS Vulnerability in the product description field allows attackers to inject malicious JavaScript, which executes when users visit the affected product page. This could lead to session hijacking, phishing attacks, or defacement of the marketplace. Risk can be increased if an admin is the affected user, potentially compromising security of the platform.

Mitigation:

- Check prevention: https://cheatsheetseries.owasp.org/cheatsheets/DOM_based_XSS_Prevention_Cheat_Sheet.html
- See CAPEC-592: https://capec.mitre.org/data/definitions/592.html

2 SQL Injection

2.1 Main Target – Retrieve Flag

Title: Website's database in vulnerable to SQL injections.

Description: Target is vulnerable to SQL Injections using searchbar, custom SQL queries can be executed without proper validation. Unauthorized user can access information stored in the database.

Steps to reproduce:

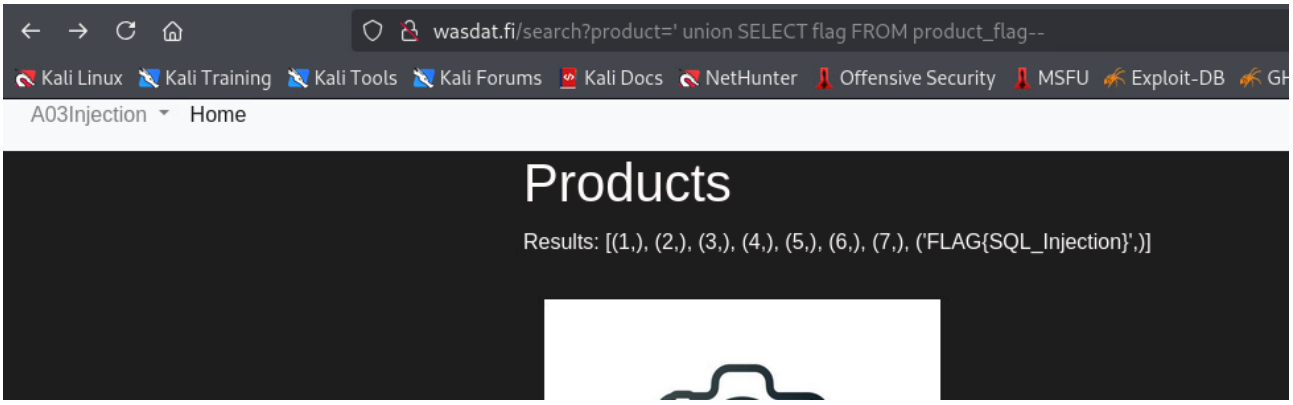
- Extract database structure using this payload:

http://wasdat.fi/search?product=%27%20union%20SELECT%20sql%20FROM%20sqlite_schema--

- Flag is stored in the "product_flag" table.
- Retrieve the flag using this payload:

  http://wasdat.fi/search?product=%27%20union%20SELECT%20flag%20FROM%20product_flag--

- Retrieved flag (see Picture 2).



Picture 2

Impact estimation:

- High severity: unauthorized user can dump all data and delete all database data.

Mitigation:

- How to prevent: https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
- https://capec.mitre.org/data/definitions/66.html

3 Time management

Date	Used hours	Description
14.02.2025	1h	Reading chapter and watching recording
16.02.2025	0.5h	Exploiting the target weakness
16.02.2025	1h	Writing the documentation and finding the suitable resource recommendations
Total	2.5h	Finishing the assignment and reporting it