

Web Application Security For Beginners

Daniel Dixon | @SherlockSec | dan@sherlock-security.com

```
root@kali:~# whoami
```

Name: Daniel Dixon

Username: DigitalSherlock

Twitter: @SherlockSec

GitHub: @Daniel-Dixon-UTC

Website: (email: dan@)sherlock-security.com

Blog: blog.sherlock-security.com

Job: GCSE Student @ UTC Sheffield OLP

Red Teamer

^C

```
root@kali:~# ./starttalk.sh
```

Objectives

- ☐ Learn and understand what a Web Application is, and where they are used.
- ☐ Learn and understand some of the common vulnerabilities in web apps
- ☐ Be able to apply these vulnerabilities and exploit the web apps
- ☐ Be able to fix these vulnerabilities in web apps
- ☐ ... Profit?

Web Applications

Uses and Explanation

What is a Web Application?

- A Web Application is a program whose client runs in a web browser.
- Common languages used in Web Apps include PHP, HTML and JavaScript
- Examples of web apps are Web Mail Clients, eCommerce, Messaging Clients.



Our Example Web Application

- In this talk, we will use OWASP Juice Shop to test some vulnerabilities
- Juice Shop is intentionally vulnerable to attacks
- Juice Shop is FOSS and available online.



OWASP Juice Shop

[https://www.owasp.org/index.php/
OWASP_Juice_Shop_Project](https://www.owasp.org/index.php/OWASP_Juice_Shop_Project)

Objectives

- ☒ Learn and understand what a Web Application is, and where they are used.
- ☐ Learn and understand some of the common vulnerabilities in web apps
- ☐ Be able to apply these vulnerabilities and exploit the web apps
- ☐ Be able to fix these vulnerabilities in web apps
- ☐ ... Profit?

OWASP Vulnerabilities

The most common Web Application vulnerabilities.

OWASP Top 10 Vulnerabilities

A1:Injection

A2:Broken Authentication

A3:Sensitive Data Exposure

A4:XML External Entities (XXE)

A5:Broken Access Control

A6:Security Misconfiguration

A7:Cross-Site Scripting (XSS)

A8:Insecure Deserialization

A9:Using Components with Known Vulnerabilities

A10:Insufficient Logging & Monitoring



OWASP

Open Web Application
Security Project

OWASP | A1: Injection

Explanation:

Injection is an attack vector in which malicious code is embedded in a poorly designed web application.

Use Cases:

Injects are most commonly used against SQL, NoSQL, OS commands and XML.

Severity:

This vuln allows for RCE, so a rating of 5 is given. (Extremely Severe)

Login

Email

Password



Log in

☐ Remember me

[Forgot your password?](#) [Not yet a customer?](#)

```
{
  "error": {
    "message": "SQLITE_ERROR: unrecognized token: \n0cc175b9c0f1b6a831c399e269772661\n",
    "stack": "SequelizeDatabaseError: SQLITE_ERROR: unrecognized token: \n0cc175b9c0f1b6a831c399e269772661\n at Query.formatError (/app/node_modules/sequelize/lib/dialects/sqlite/query.js:423:16)\n at afterExecute (/app/node_modules/sequelize/lib/dialects/sqlite/query.js:119:32)\n at replacement (/app/node_modules/sqlite3/lib/trace.js:19:31)\n at Statement.errBack (/app/node_modules/sqlite3/lib/sqlite3.js:16:21)",
    "name": "SequelizeDatabaseError",
    "parent": {
      "errno": 1,
      "code": "SQLITE_ERROR",
      "sql": "SELECT * FROM Users WHERE email = '' AND password = '0cc175b9c0f1b6a831c399e269772661'",
      "original": {
        "errno": 1,
        "code": "SQLITE_ERROR",
        "sql": "SELECT * FROM Users WHERE email = '' AND password = '0cc175b9c0f1b6a831c399e269772661'",
        "sql": "SELECT * FROM Users WHERE email = '' AND password = '0cc175b9c0f1b6a831c399e269772661'"
      }
    }
  }
}
```

Email

Password



This connection is not secure. Logins entered here could be compromised. [Learn More](#)



Log in

☐ Remember me

[Forgot your password?](#) [Not yet a customer?](#)

Login

Email

' OR 3=3 --

Password


•





Log in


☐ Remember me

[Forgot your password?](#) [Not yet a customer?](#)

 OWASP Juice Shop v7.4.0

 admin@juice-sh.op

 Logout

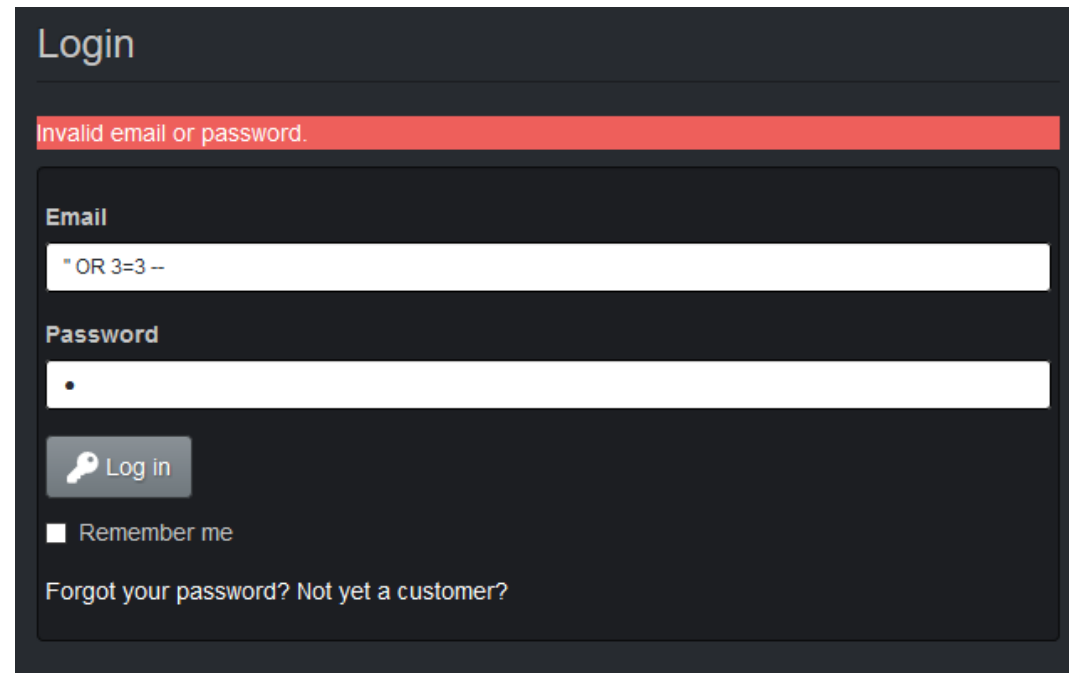
 English ▾

OWASP | A1: Injection

Mitigation Techniques:

Who needs a database anyways?

Sanitize any inputs by escaping special characters like ' " ; -



The screenshot shows a dark-themed login interface. At the top, the word "Login" is displayed. Below it, a red error message reads "Invalid email or password." The "Email" input field contains the text "' OR 3=3 --", which is an SQL injection payload. The "Password" field is empty and masked with dots. Below the password field is a "Log in" button with a key icon. Underneath the button is a "Remember me" checkbox, which is currently unchecked. At the bottom of the form, there are two links: "Forgot your password?" and "Not yet a customer?".

OWASP | A2: Broken Authentication

Explanation:

This attack vector allows the attacker to capture or bypass the authentication methods used in a web app.

Use Cases:

Brute force; cred stuffing; Exploiting unexpired session tokens

Severity:

This allows the attacker to potentially obtain admin privileges, so is rated at a 4

Forgot Password

Email

 Change

Forgot Password

Email

jim@juice-sh.op

Your eldest siblings middle name?

New Password

Repeat New Password



Change

James Tiberius Kirk

Star Trek character



William Shatner as Kirk in a publicity photograph for the original *Star Trek*

James Kirk's brother, George Samuel Kirk,

Forgot Password

Email

jim@juice-sh.op

Your eldest siblings middle name?


.....

New Password

.....

Repeat New Password

.....

 Change

Forgot Password

Your password was successfully changed.

Email



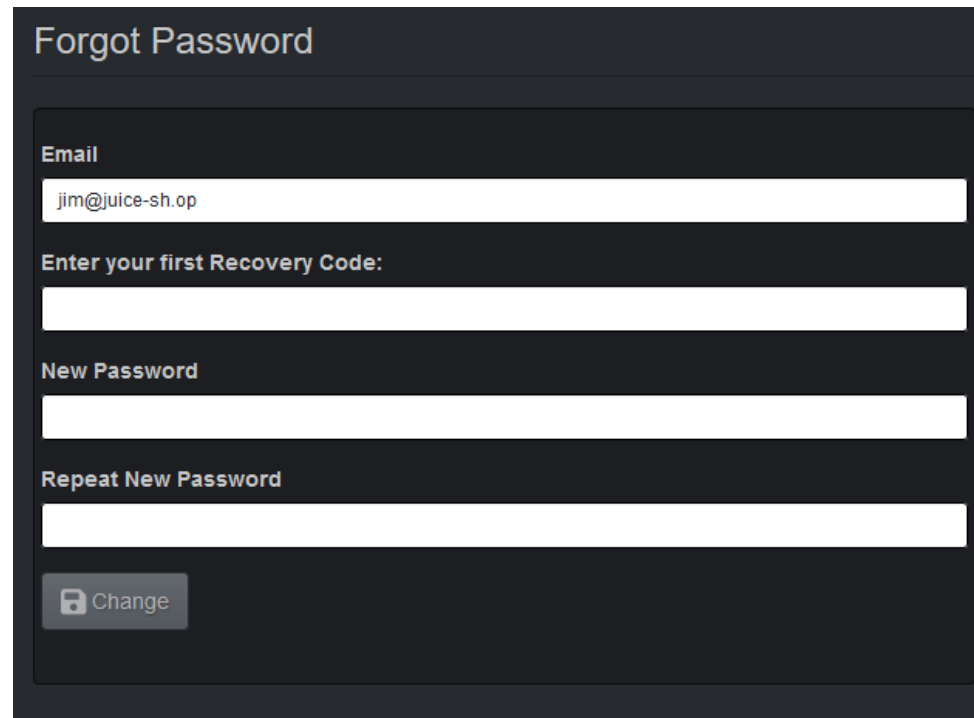
Change

OWASP | A2: Broken Authentication

Mitigation Techniques:

Remove Authentication altogether

Don't use weak security questions. Require MFA. Use Recovery Codes to reset passwords.



Forgot Password

Email

jim@juice-sh.op

Enter your first Recovery Code:

New Password

Repeat New Password

Change

OWASP | A3: Sensitive Data Exposure

Explanation:

This attack vector allows the attacker to obtain sensitive data that the application did not correctly protect i.e. credit card information

Use Cases:

Credit Card Fraud; Obtaining access to another users account; Obtaining the full password file of the application.

Severity:

Due to the recent GDPR, this vulnerability has been especially more important, and as such is rated at a 5


About Us Corporate History & Policy

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. **Check out our boring terms of use if you are interested in such lame stuff.** Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Lorem ipsum dolor sit amet, consetetur adipiscing elit, sed diam nonumy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, At accusam aliquyam diam diam dolore dolores duo eirmod eos erat, et nonumy sed tempor et et invidunt justo labore Stet clita ea et gubergren, kasd magna no rebum. sanctus sea sed takimata ut vero voluptua. est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur.

Opening legal.md



You have chosen to open:

 **legal.md**

which is: Markdown Source File (3.0 kB)

from: <http://dsherlock-owasp.herokuapp.com>

What should Firefox do with this file?



Open with

Visual Studio Code (default)



Save File



Do this automatically for files like this from now on.

OK

Cancel

dsherlock-owasp.herokuapp.com/ftp/legal.md?md_debug=true



dsherlock-owasp.herokuapp.com/ftp/

~ / ftp /



acquisitions.md



incident-support.kdbx



suspicious_errors.yml

Opening acquisitions.md



You have chosen to open:



acquisitions.md

which is: Markdown Source File (909 bytes)

from: <http://dsherlock-owasp.herokuapp.com>

What should Firefox do with this file?



Open with

Visual Studio Code (default)



Save File



Do this automatically for files like this from now on.

OK

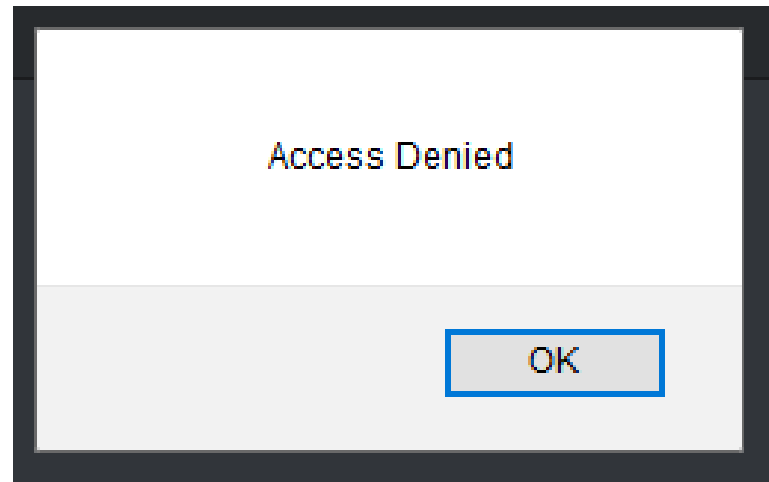
Cancel

OWASP | A3: Sensitive Data Exposure

Mitigation Techniques:

Maybe don't publicly host company secrets?

Ensure proper access control; IP Whitelist; Discard of unneeded data



OWASP | A4: XML External Entities (XXE)

Explanation:

This attack vector allows the attacker to exploit a poor XML parser to execute arbitrary code.

Use Cases:

Disclosure of confidential data; Denial of Service; Anything that is possible with code.

Severity:

As this involves RCE, this attack vector is rated as a 5.

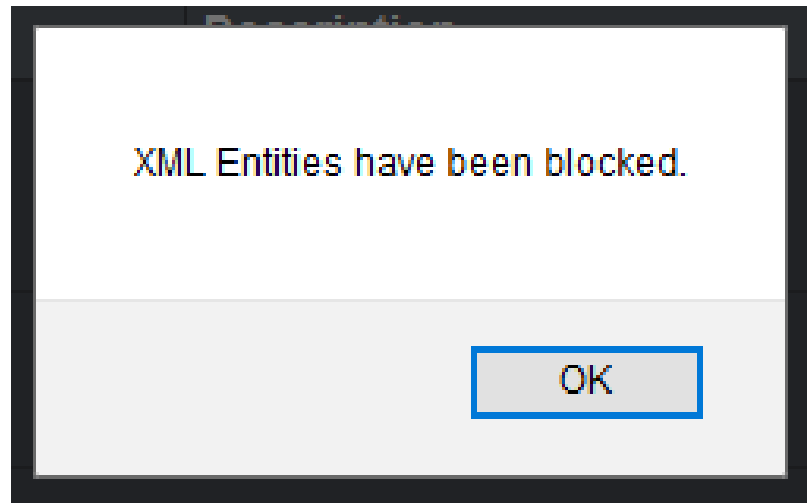
Live Demonstration

OWASP | A4: XML External Entities (XXE)

Mitigation Techniques:

Don't have a relationship with other businesses.

Don't allow unrestricted XML upload; Don't allow XML File Entities;



OWASP | A5: Broken Access Control

Explanation:

This attack vector allows the attacker to access areas of the web app of which they do not have the permissions to.

Use Cases:

Accessing Admin Panels; Accessing Account Info

Severity:

This attack vector is rated as a 3.

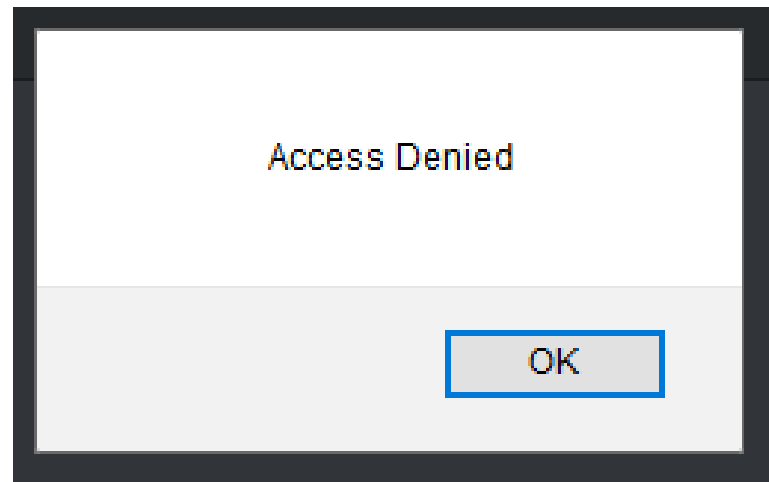
Live Demonstration

OWASP | A5: Broken Access Control

Mitigation Techniques:

Do we really even need administrators?

Properly configure access controls; Check page and file permissions



OWASP | A6: Security Misconfiguration

Explanation:

This attack vector allows the attacker to take advantage of incorrectly set safeguards

Use Cases:

Accessing older software; Directory Listing; Viewing Error Messages

Severity:

This attack vector is rated as a 2.

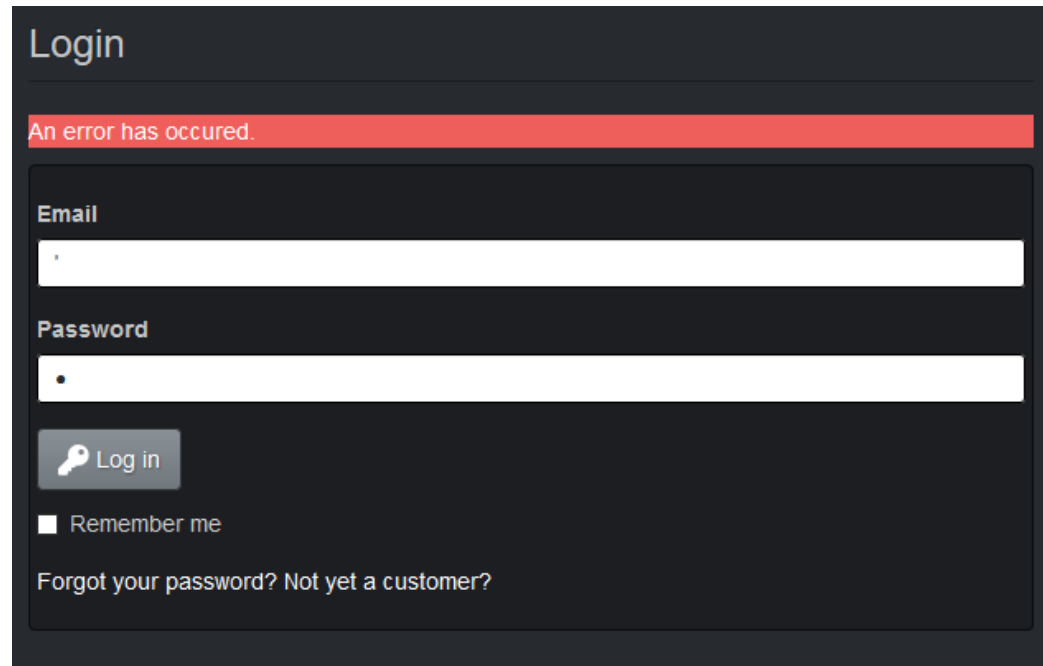
Live Demonstration

OWASP | A6: Security Misconfiguration

Mitigation Techniques:

Write good code that never has errors.

Disable verbose error messages; Trust no-one.



The screenshot shows a login interface with the title "Login". Below the title, a red error bar displays the message "An error has occurred." This is a security misconfiguration because it provides a generic error message that could be exploited by attackers to determine the validity of their input. The form includes fields for "Email" and "Password", a "Log in" button with a key icon, a "Remember me" checkbox, and links for "Forgot your password?" and "Not yet a customer?".

OWASP | A7: Cross Site Scripting (XSS)

Explanation:

This attack vector allows the attacker to inject malicious JavaScript into in another user's browser. It is the second most common vulnerability found in web apps.


Use Cases:

Anything possible with JavaScript

Severity:

This attack vector is rated as a 5.

`<script>alert("XSS")</s`

 Search

XSS

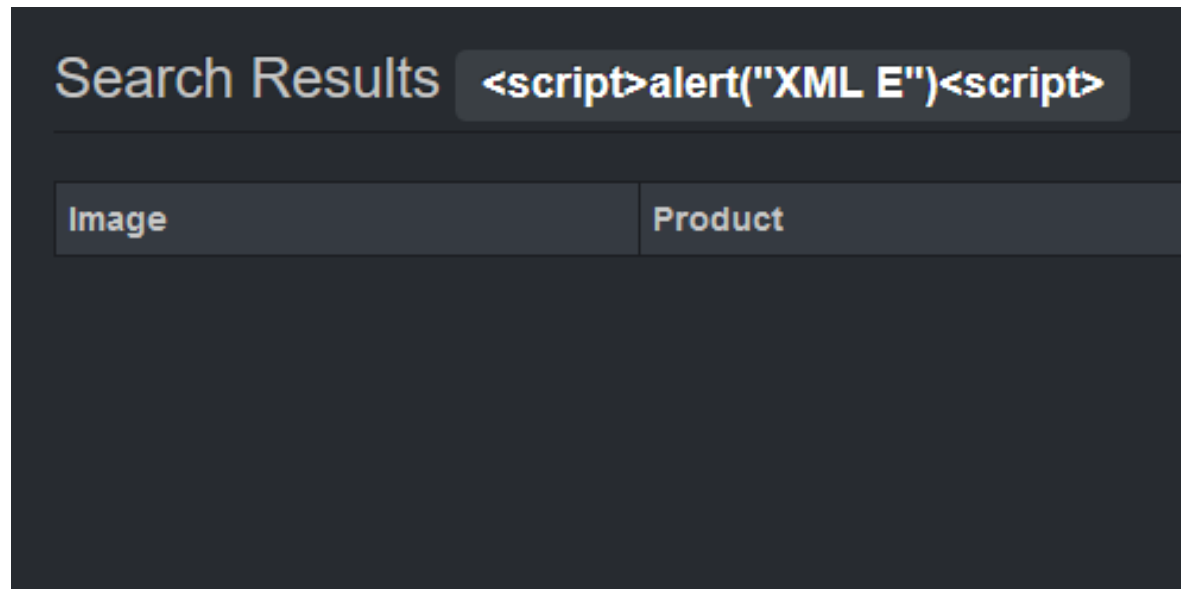
OK

OWASP | A7: Cross Site Scripting (XSS)

Mitigation Techniques:

Don't use JavaScript.

Escape HTML tags; Trust no-one; Use XSS blocking frameworks.



OWASP | A8: Insecure Deserialization

Explanation:

This attack vector allows the attacker to abuse the logic of a program with untrusted data.

Use Cases:

Denial Of Service; RCE

Severity:

This attack vector is rated as a 5.

NextGen B2B API 2.0.0 OAS3

New & secure JSON-based API for our enterprise customers. (Deprecates previously offered XML-based endpoints)

[MIT](#)

Authorize



Server

/b2b/v2



Customer order to be placed

Example Value | Model

```
{"orderLinesData": "(function dos() {while(true); })()"}|
```

Response body

```
{  
  "error": {  
    "message": "Infinite loop detected - reached max iterations",  
    "stack": "/app/node_modules/notevil/lib/infinite-checker.js:15\
```

OWASP | A8: Insecure Deserialization

Mitigation Techniques:

APIs are bad anyways, don't bother using them.

Run the code in a sandbox first; Check the integrity; Enforce strict constraints

Response body

```
{  
  "error": {  
    "message": "Infinite loop detected"
```

OWASP | A9: Using Components with Known Vulnerabilities

Explanation:

This attack vector allows the attacker to exploit already documented vulnerabilities in components added to the code.

Use Cases:

Exploit Dependent

Severity:

This attack vector is rated as a 5.

OWASP | A9: Using Components with Known Vulnerabilities

Mitigation Techniques:

WAFS – Web App From Scratch

Check the components that you are using; Double check for any typing errors; Keep up to date.



You're Up to Date!

OWASP | A10: Insufficient Logging/Monitoring

Explanation:

This occurs when security-critical events are not recorded.

Use Cases:

Exploit Dependent

Severity:

This attack vector is rated as a 5.

OWASP | A10: Insufficient Logging/Monitoring

Mitigation Techniques:

Just make your own cyber threat task force

Keep a verbose set of logs offline. Always monitor for suspicious activity. Have a damage control plan just in case.



Objectives

- ☒ Learn and understand what a Web Application is, and where they are used.
- ☒ Learn and understand some of the common vulnerabilities in web apps
- ☒ Be able to apply these vulnerabilities and exploit the web apps
- ☒ Be able to fix these vulnerabilities in web apps
- ☒ ... Profit?

Thank you!

Questions and credits