OWASP
Open Web Application
Security Project

OWASP Top 10

David Quisenberry
@quizsec

Agenda

- What is OWASP?
- Goals for this Workshop
- Intro to OWASP Juice Shop
- What is the OWASP Top 10?
- Exploring the OWASP Top 10 using Juice Shop

Goals

- Become More Familiar with OWASP Top 10
- Help Each Other
- Have Fun!

What Is OWASP

OWASP

Projects

Conferences

Chapters

The Portland Chapter

Twitter: @PortlandOWASP

Website: pdxowasp.org

Meetup: OWASP-Portland-Chapter

Slack: owasp.slack.com #chapter-pdx

OWASP Juice Shop

The Project Repo

This is not how we are going to set up today, but if you want it for future reference...

https://github.com/bkimminich/juice-shop

OWASP Juice Shop Setup

Create a Juice Shop

https://juicyctf.com

Register for the CTF

http://ctf.juicyctf.com

OWASP Juice Shop Walk Through

- Setup
- Creating CTF Account
- Exploring the Application in Firefox

What is the OWASP Top Ten?

What are they?

I Bought Six Xylophones Because Someone Could Imagine Cool Instruments

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access Control
- Security Misconfiguration
- Cross-Site Scripting (XSS)
- Insecure Deserialization
- Components w/ Vulns
- Insufficient Logging / Monitor

Untrusted Data

Interpreter Executes Unintended Commands

Injection

Injection

- Injection Vectors
- Prevalent in Legacy Code
- Discoverable
- Bad

Injection

What makes you vulnerable?

- Untrusted Data Not Validated, Filtered, or Sanitized
- Dynamic Queries Containing Untrusted Data

Injection - Dynamic Query Example

```
String sql = "SELECT UserID FROM User WHERE " +
             "UserName = '" + UserName + "' AND " +
             "Password = '" + Password + "'";
Using (SqlCommand cmd = new SqlCommand(sql))...
```

https://software-security.sans.org/developer-how-to/fix-sql-injection-microsoft-.net-with-parameterized-queries

Injection - Dynamic Query Example

```
String sql = "SELECT UserID FROM User WHERE " +
             "UserName = '" + 'OR 1=1-- + "' AND " +
             "Password = '" + Password + "'";
Using (SqlCommand cmd = new SqlCommand(sql))...
```

https://software-security.sans.org/developer-how-to/fix-sql-injection-microsoft-.net-with-parameterized-queries

Injection

Prevention

- Prepared Statements / Stored Procedures
- Whitelist Input Validation
- Escape Untrusted Data
- Code Review
- LIMIT Controls

Injection - Parameterized Query Example

```
String sql = “SELECT UserID FROM User WHERE “ +
                    “UserName = @UserName “ +
                    “AND Password = @Password“;
Using (SqlCommand cmd = new SqlCommand(sql))
{
    cmd.Parameters.Add(“@UserName”, System.Data.SqlDbType.NVarChar, 50); …
    cmd.Parameters[“@UserName”].Value = UserName; …
}
```

https://software-security.sans.org/developer-how-to/fix-sql-injection-microsoft-.net-with-parameterized-queries

Injection

Try to identify where in the Juice Shop SQL queries might be being executed.

Can you alter the way the SQL query executes?

Auth / Session Management Implemented Incorrectly

Broken Authentication

Broken Authentication

- Brute Force Is Prevalent
- Session Management Attacks Understood

Broken Authentication

What Makes You Vulnerable?

- Permits Credential Stuffing
- Permits Weak/Well Known Passwords
- Weak Credential Recovery Methods
- Does Not Rotate Session IDs
- Improperly Invalidates Session IDs

Broken Authentication

Prevention

- MFA
- Implement Weak Password Checks
- Limit and Log Failed Login Attempts
- Session Tokens Out of URL, Securely Stored, Invalidated after Logout, Idle, Defined Times

Broken Authentication

Can you guess a valid login for the administrator's account?

Does the site enforce strong passwords?

Financial, Health, PII Data not Properly Protected

Sensitive Data Exposure

Sensitive Data Exposure

What Makes you Vulnerable?

- Data Transmitted Clear Text
- Old/Weak Crypto
- Encryption Not Enforced
- Not Determining the Protection of Data

Sensitive Data Exposure

Prevention

- Identify and Classify Data Processed, Stored, and Transmitted - Applying Controls
- Encrypt Sensitive Data at Rest
- Ensure Strong Crypto
- Disable Caching for Sensitive Data Responses

Sensitive Data Exposure

Can you find any files you think should be confidential?

Old / Poorly Configured XML Processors Evaluating External Entities

XML External Entities (XXE)

XML External Entities (XXE)

What May Make you Vulnerable?

- App Accepts XML, XML Uploads
- XML processors or SOAP based Web Services with DTD's Enabled
- SOAP Prior to Version 1.2
- App Uses SAML for Identity Processing

XML External Entities (XXE)

Prevention

- Patch/Upgrade all XML Processors/Libraries
- Disable XML External Entity and DTD Processing
- Whitelist XML Inputs / Validate Files with XSD
- SAST Tooling / Code Reviews

XML External Entities (XXE)

Challenges for XXE only work on non-Docker based environments cause certain XXE attacks can crash the docker process.

But…. Where would you try it? How? Spin up Juice shop in Node and give it a whirl.

Restrictions on Authenticated Users Not Properly Enforced

Broken Access Control

Broken Access Control

What Makes you Vulnerable?

- Users Bypassing Access Control Checks
- Allowing Primary Key to Be Changed to Another Users Record
- Unauthorized API Access

Broken Access Control

Prevention

- Enforce Server Side or Serverless API
- Model Access Controls Record Ownership
- Log Access Control Failures
- Rate Limit API
- Invalidate JWT Tokens on Server at Logout

Broken Access Control

Can you access areas of the site you don't think you should have permission to?

Insecure Default Configurations

Verbose Error Messages

OS, Framework, Libraries

Security Misconfiguration

Security Misconfiguration

What Makes you Vulnerable?

- Missing Security Hardening
- Unnecessary Features Installed/Enabled
- Default Accounts
- Verbose Error Reporting
- Out of Date Packages

Security Misconfiguration

Prevention

- Repeatable, Automated Hardening Process
- Unique Creds for Dev, QA, Prod Environments
- Patch Management Process
- Segmented Application Architecture (Secure Separation)

Security Misconfiguration

Provoke an error that isn’t gracefully handled.

Untrusted Data Included without Validation or Escaping

Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS)

Vulnerabilities

- Reflected XSS
- Stored XSS
- DOM XSS

Cross-Site Scripting (XSS)

Prevention

- Use Frameworks that Escape XSS by Design
- Escape Untrusted Data in HTML output
- Apply Context-Sensitive Encoding
- Enable Content Security Policy

Cross-Site Scripting (XSS)

Where do you think you could *Reflect* a XSS attack into the browser?

Can you identify any areas you might be able to execute a *Stored* XSS attack?

Accepting Serialized Objects from Untrusted Sources

Insecure Deserialization

Insecure Deserialization

What Makes you Vulnerable?

- App Deserializes Hostile or Tampered Objects

Attacks

- Modification of App Logic or RCE
- Data Tampering

Insecure Deserialization

Prevention

- Do Not Accept Serialized Objects from Untrusted Sources
- Integrity Checks
- Log Deserialization Exceptions and Failures

Insecure Deserialization

Similar to XXE this challenge doesn't work in docker.

But… can you discover an area of the site you might be able to get the app to deserialize a tampered object?

Component run same privileges as the application

Components with Vulnerabilities

Components with Known Vulns

What Makes you Vulnerable?

- Out of Date Components
- Not Scanning for Vulnerabilities
- Not Fixing / Upgrading Components Timely
- Not Securing Components Configurations

Components with Known Vulns

Prevention

- Removed Unused Components
- Inventory of Client/Server Side Components
- Monitor for Components Not Being Maintained
- File Integrity

Attacker Persistence, Deepen Attack

Insufficient Logging and Monitoring

Insufficient Logging / Monitoring

What Makes you Vulnerable?

- Auditable Events are not Logged
- Warnings/Errors Generate Inadequate Logs
- Logs aren't Monitored For Suspicious Activity
- Logs Only Stored Locally
- Pen Testing / DAST Tools do not Trigger Alerts

Insufficient Logging / Monitoring

Prevention

- Login, Access Control Failures Logged With Context
- Log Format Machine Readable
- Log Integrity
- Alerts / Incident Plans

Resources

Resources

OWASP Top Ten

OWASP Cheat Sheets

OWASP ASVS

OWASP Testing Guide

OWASP Juice Shop

Environment Variables CTF Setup

If you are setting up your own juiceshop using node, docker, etc make sure to set these environment variables to be able to earn points.

NODE_ENV=ctf

CTF_KEY=zLp@.-6fMW6L-7R3b!9uR_K!NfkkTr