

OWASP Top 10

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration .
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery

Broken Access Control

List of Mapped CWEs:

1. CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
2. CWE-23 Relative Path Traversal
3. CWE-35 Path Traversal: '.../.../'
4. CWE-59 Improper Link Resolution Before File Access ('Link Following')
5. CWE-200 Exposure of Sensitive Information to an Unauthorized Actor
6. CWE-201 Exposure of Sensitive Information Through Sent Data
7. CWE-219 Storage of File with Sensitive Data Under Web Root
8. CWE-264 Permissions, Privileges, and Access Controls (should no longer be used)
9. CWE-275 Permission Issues
10. CWE-276 Incorrect Default Permissions
11. CWE-284 Improper Access Control

Etc.

Viewing previous chat transcripts to gain access to password:

The screenshot shows a web browser window with the URL `https://0a48007804b48bc280...`. The page is from 'WebSecurity Academy' and displays a lab titled 'Insecure direct object references' with a 'LAB Solved' badge. Below the lab title, there's a 'Back to lab description' link. A large orange banner says 'Congratulations, you solved the lab!' with 'Share your skills!' and 'Continue learning >>' buttons. The 'My Account' section shows the username 'carlos' and an 'Update email' button. In the background, Burp Suite is open, showing an HTTP request and response. The request is a GET to `download=transcript/1.txt HTTP/2`. The response is an HTTP 200 OK with a Content-Type of `text/plain; charset=utf-8` and a Content-Disposition of `attachment; filename='1.txt'`. The response body contains a chat transcript where a user named 'Hal Pline' asks for a password and receives a confirmation message.

Cryptographic failures:

List of Mapped CWEs

1. CWE-261 Weak Encoding for Password
2. CWE-296 Improper Following of a Certificate's Chain of Trust
3. CWE-310 Cryptographic Issues
4. CWE-319 Cleartext Transmission of Sensitive Information
5. CWE-321 Use of Hard-coded Cryptographic Key
6. CWE-322 Key Exchange without Entity Authentication
7. CWE-323 Reusing a Nonce, Key Pair in Encryption
8. CWE-324 Use of a Key Past its Expiration Date
9. CWE-325 Missing Required Cryptographic Step
10. CWE-326 Inadequate Encryption Strength
11. CWE-327 Use of a Broken or Risky Cryptographic Algorithm
12. CWE-328 Reversible One-Way Hash
13. CWE-329 Not Using a Random IV with CBC Mode
14. CWE-330 Use of Insufficiently Random Values

Etc.

Source code disclosure via backup files

The screenshot displays a web browser window with a Java source code file open. The code is for a class named `ProductTemplate` that implements `Serializable`. It contains a static `serialVersionUID`, private fields for `id` and `product`, and a public constructor. The `readObject` method is overridden to read an `ObjectInputStream` and execute a SQL query to retrieve product information from a database. The code is partially obscured by a blue highlight.

Below the code, a Burp Suite HTTP history table is visible, showing a list of requests and responses. The table has columns for #, Host, Method, URL, Params, Edited, Status code, and Length. The selected request is a GET request to `https://0ae3005b04bf7e6f813e026200c40a8.web-security-academy.net/backup/ProductTemplate.java.bak` with a status code of 200 and a length of 1776.

#	Host	Method	URL	Params	Edited	Status code	Length
573	https://0ae3005b04bf7e6f813e026200c40a8.web-security-academy.net	GET	/robots.txt			200	139
572	https://0ae3005b04bf7e6f813e026200c40a8.web-security-academy.net	GET	/robots.txt			404	131
571	https://0ae3005b04bf7e6f813e026200c40a8.web-security-academy.net	GET	/resources/labheader/images/ps-lab-s...			200	707
570	https://0ae3005b04bf7e6f813e026200c40a8.web-security-academy.net	POST	/submitSolution			200	100
569	https://0ae3005b04bf7e6f813e026200c40a8.web-security-academy.net	GET	/academyLabHeader			101	147
568	https://0ae3005b04bf7e6f813e026200c40a8.web-security-academy.net	GET	/			200	10866
567	https://0ae3005b04bf7e6f813e026200c40a8.web-security-academy.net	GET	/robots.txt			404	131
566	https://0ae3005b04bf7e6f813e026200c40a8.web-security-academy.net	GET	/backup/ProductTemplate.java.bak			200	1776
565	https://googleads.g.doubleclick.net	GET	/pagead/clkid=1			200	836
564	https://googleads.g.doubleclick.net	GET	/pagead/clkid=1			302	745
563	https://0ae3005b04bf7e6f813e026200c40a8.web-security-academy.net	GET	/robots.txt			200	139

The Burp Suite interface also shows a 'Request' tab with the raw HTTP request and a 'Response' tab with the raw HTTP response. The response is a 200 OK status with a content type of `text/plain; charset=utf-8` and a content length of 1667. The response body contains the source code of the `ProductTemplate` class.

Injections:

List of Mapped CWEs

1. CWE-20 Improper Input Validation
2. CWE-74 Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection')
3. CWE-75 Failure to Sanitize Special Elements into a Different Plane (Special Element Injection)
4. CWE-77 Improper Neutralization of Special Elements used in a Command ('Command Injection')
5. CWE-78 Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
6. CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
7. CWE-80 Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)
8. CWE-83 Improper Neutralization of Script in Attributes in a Web Page
9. CWE-87 Improper Neutralization of Alternate XSS Syntax
10. CWE-88 Improper Neutralization of Argument Delimiters in a Command ('Argument Injection')
11. CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
12. CWE-90 Improper Neutralization of Special Elements used in an LDAP Query ('LDAP Injection')
13. CWE-91 XML Injection (aka Blind XPath Injection)

Etc

SQL injection vulnerability allowing login bypass:

The screenshot displays a web browser window on the left and the Burp Suite proxy interface on the right. The browser window shows a 'Web Security Academy' lab titled 'SQL injection vulnerability allowing login bypass'. The lab status is 'Solved'. Below the title, there is a 'Login' form with fields for 'Username' (containing 'administrator') and 'Password' (containing '1c3456789'). A 'Log in' button is at the bottom of the form. The Burp Suite interface on the right shows an intercepted HTTP request. The request is a POST to '/login' with a body containing a SQL injection payload. The payload is: `username=administrator'--&password=1c3456789`. The Burp Suite interface also shows the request headers, including 'Host', 'Cookie', 'Content-Length', 'Cache-Control', 'Sec-CH-UA', 'User-Agent', 'Accept', 'Referer', and 'Accept-Encoding'.

Insecure Design

List of Mapped CWEs

1. CWE-73 External Control of File Name or Path
2. CWE-183 Permissive List of Allowed Inputs
3. CWE-209 Generation of Error Message Containing Sensitive Information
4. CWE-213 Exposure of Sensitive Information Due to Incompatible Policies
5. CWE-235 Improper Handling of Extra Parameters
6. CWE-256 Unprotected Storage of Credentials
7. CWE-257 Storing Passwords in a Recoverable Format
8. CWE-266 Incorrect Privilege Assignment
9. CWE-269 Improper Privilege Management
10. CWE-280 Improper Handling of Insufficient Permissions or Privileges

Etc.

Multi-step Clickjacking

Web! Acad Multistep clickjacking LAB Not solved

[Go to exploit server](#)
[Back to lab description >>](#)

[Home](#) | [My account](#) | [Log out](#)

My Account

Your username is: wiener [Test me next](#)

Email

[Update email](#)

[Delete account](#)

Security Misconfiguration

List of Mapped CWEs

1. CWE-2 7PK - Environment
2. CWE-11 ASP.NET Misconfiguration: Creating Debug Binary
3. CWE-13 ASP.NET Misconfiguration: Password in Configuration File
4. CWE-15 External Control of System or Configuration Setting
5. CWE-16 Configuration
6. CWE-260 Password in Configuration File
7. CWE-315 Cleartext Storage of Sensitive Information in a Cookie
8. CWE-520 .NET Misconfiguration: Use of Impersonation
9. CWE-526 Exposure of Sensitive Information Through Environmental Variables
10. CWE-537 Java Runtime Error Message Containing Sensitive Information
11. CWE-541 Inclusion of Sensitive Information in an Include File