# Security Labs Listing | Veracode

## OWASP 2017 #1: Injection

Exploiting and preventing SQL injection attacks that access sensitive data.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Own the database | Practice SQLi on an app that uses SQLite to retrieve data. | Beginner, Intermediate | Enterprise | 10 min. | • .NET<br>• Golang<br>• Java<br>• Javascript<br>• PHP<br>• Python (Django)<br>• Python (Flask)<br>• Rails<br>• Scala | Lesson |
| Parameterize all the things | Defend against SQL injection using an app that returns data from an H2 database | Beginner, Intermediate | Enterprise | 15 min. | • .NET<br>• Golang<br>• Java<br>• Javascript<br>• PHP<br>• Python (Django)<br>• Python (Flask)<br>• Rails<br>• Scala | Lesson |
| Bobby Tables Challenge | Use SQLi to return sensitive data, then properly parameterize queries to avoid injection attacks. | Advanced | Enterprise | 20 min. | • .NET<br>• Golang | Challenge |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| | | | | | • Java<br>• Javascript<br>• PHP<br>• Python (Django)<br>• Python (Flask)<br>• Rails<br>• Scala | |
| Timing is Everything Challenge | Indirectly reveal sensitive data using SQL 'sleep' commands. | Intermediate, Advanced | Enterprise | 15 min. | • Python (Django)<br>• Python (Flask) | Challenge |

## OWASP 2017 #2: Broken Authentication

Enforcing user password requirements and properly encrypting passwords.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Really, really bad passwords | Enforcing server-side and client-side password requirements for users. | Beginner | Enterprise | 15 min. | • .NET<br>• Java<br>• Python (Django)<br>• Python (Flask)<br>• Rails<br>• Scala | Lesson |
| Hash it, store it, salt - upgrade it | Encrypting user passwords securely. | Beginner | Enterprise | 15 min. | • Golang<br>• Java<br>• Javascript | Lesson |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| | | | | | • PHP<br>• Python (Django)<br>• Python (Flask)<br>• Rails<br>• Scala | |
| Terrible Password Challenge | SQLi and poor hashing lead to exposed passwords. | Advanced | Enterprise | 20 min. | • .NET<br>• Java<br>• Javascript<br>• Python (Flask)<br>• Rails | Challenge |
| Authentication Bypass | Force browse to an unprotected page to discover confidential information. | Intermediate, Advanced | Enterprise | 15 min. | • Golang<br>• Javascript | Lesson |

## OWASP 2017 #3: Sensitive Data Exposure

Stack traces and debug info available in production.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Bugs in Debug | Verbose error messages lead to exposed sensitive data. | Beginner, Intermediate | Enterprise | 15 min. | • .NET<br>• Golang<br>• Java<br>• Javascript<br>• PHP<br>• Python (Django)<br>• Python (Flask) | Lesson |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| | | | | | • Rails<br>• Scala | |
| Helpful Stack Trace Challenge | Use exposed information in stack traces to exploit a vulnerable application. | Intermediate, Advanced | Enterprise | 20 min. | • .NET<br>• Python (Django)<br>• Python (Flask)<br>• Rails | Challenge |
| Secret Logging Challenge | Force an application to throw an error and leak sensitive data in a stack trace. | Intermediate, Advanced | Enterprise | 20 min. | • Python (Flask) | Challenge |

## OWASP 2017 #4: XXE

Injection attacks from unsafe XML parsing.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| eXternal Entity (injection) | Unsafe entity parsing reveals the contents of server files. | Beginner, Intermediate | Enterprise | 15 min. | • .NET<br>• Golang<br>• Java<br>• Javascript<br>• PHP<br>• Python (Django)<br>• Python (Flask)<br>• Rails<br>• Scala | Lesson |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| XML is Always a Challenge | Get access to sensitive data by injecting custom XML. | Intermediate, Advanced | Enterprise | 15 min. | • .NET<br>• Golang<br>• Python (Django)<br>• Python (Flask)<br>• Rails | Challenge |
| External Resolution Challenge | Retrieve a system file by injecting custom XML, then defend against XXE. | Advanced | Enterprise | 15 min. | • Python (Flask) | Challenge |

## OWASP 2017 #5: Broken Access Control

Session fixation leads to account hijacking.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Fix the Sessions | Unsafe entity parsing reveals the contents of server files. | Intermediate, Advanced | Enterprise | 15 min. | • Java | Lesson |
| To Protect and to Serve Secure Cookies | Tamper with an insecure cookie for privilege escalation. | Intermediate, Advanced | Enterprise | 15 min. | • .NET<br>• Golang<br>• Javascript<br>• PHP<br>• Python (Django)<br>• Python (Flask)<br>• Rails<br>• Scala | Lesson |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Bad Cookie Challenge | Decrypt cookies and hijack another user account. | Intermediate, Advanced | Enterprise | 20 min. | • .NET<br>• Golang<br>• Javascript<br>• Python (Flask)<br>• Rails | Challenge |

## OWASP 2017 #6: Security Misconfiguration

Insecure secret keys lead to compromised authentication measures.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Jot down this key | Modify JWTs by exploiting knowledge of an insecure secret key. | Intermediate, Advanced | Enterprise | 25 min. | • .NET<br>• Java<br>• Javascript | Lesson |
| Can you keep a secret? | Modify JWTs by exploiting knowledge of an insecure secret key. | Intermediate, Advanced | Enterprise | 25 min. | • Golang<br>• Javascript<br>• PHP<br>• Python (Django)<br>• Python (Flask)<br>• Scala | Lesson |
| Bulky Updates | Access hidden attributes to take unauthorized actions. | Intermediate, Advanced | Enterprise | 15 min. | • Rails | Lesson |
| Secret Admin Challenge | Escalate JWT user privileges by exploiting knowledge of an insecure secret key. | Advanced | Enterprise | 25 min. | • .NET<br>• Java<br>• Python | Challenge |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| | | | | | (Flask)<br>• Rails | |

## OWASP 2017 #7: XSS

Reflected and persistent cross-site scripting attacks. Content Security Policy.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Can you see your reflection? | Practice exploiting simple cross-site scripting vulnerabilities to deliver JavaScript payloads. | Beginner, Intermediate | Enterprise | 15 min. | • .NET<br>• Golang<br>• Java<br>• Javascript<br>• PHP<br>• Python (Django)<br>• Python (Flask)<br>• Rails<br>• Scala | Lesson |
| Down with Uploads | Exploit stored cross-site scripting via "image" uploads. | Beginner, Intermediate | Enterprise | 20 min. | • .NET<br>• Java<br>• Javascript<br>• Python (Django)<br>• Python (Flask)<br>• Scala | Lesson |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Check your sources | Content Security Policy to prevent XSS and other code injection. | Beginner, Intermediate | Enterprise | 20 min. | • Java<br>• Python (Django) | Lesson |
| Alert Challenge | Exploit a non-persistent XSS vulnerability in a poorly protected app. | Intermediate, Advanced | Enterprise | 20 min. | • Java<br>• Javascript<br>• Python (Django)<br>• Python (Flask) | Challenge |
| Persistence Challenge | Exploit directory traversal and persistent XSS vulnerabilities in a poorly protected app. | Advanced | Enterprise | 20 min. | • Java<br>• Javascript<br>• Python (Django)<br>• Python (Flask)<br>• Rails | Challenge |
| Stored XSS Versus CSP | Defense in depth using CSP against XSS attacks. | Advanced | Enterprise | 20 min. | • .NET<br>• Javascript<br>• Scala | Lesson |
| Angular ERB sanitization | Cause XSS through improper sanitization and poor variable handoff with Angular. | Intermediate | Enterprise | 15 min. | • Rails | Lesson |
| React Sanitization | Cause XSS through improper sanitization and poor variable handoff with React. | Intermediate | Enterprise | 15 min. | • Rails | Lesson |
| Angular HTML and URL Sanitization | Cause XSS through improper sanitation and poor variable handoff with Angular. | Intermediate | Enterprise | 15 min. | • Rails | Lesson |

# OWASP 2017 #8: Insecure Deserialization

Deserialization attacks lead to remote code execution.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| In a Pickle | A vulnerable serialization library allows dangerous user payloads. | Intermediate, Advanced | Enterprise | 15 min. | • .NET<br>• Golang<br>• Java<br>• PHP<br>• Python (Django)<br>• Python (Flask)<br>• Scala | Lesson |
| Mongo: Like SQL, but Messier | View non-public posts by supplying a document query as user input. | Intermediate | Enterprise | 15 min. | • Javascript | Lesson |
| Tell Mongo "no-go" for Untrusted Code | Defend against NoSQL IDOR on a NodeJS app that uses MongoDB to store and retrieve data. | Intermediate, Advanced | Enterprise | 15 min. | • Javascript | Lesson |
| Deserialization Challenge | Use pickling to reveal the code of the underlying application. | Advanced | Enterprise | 15 min. | • Python (Flask) | Challenge |
| User-Provided Users | Exposed, unhashed user IDs are modifiable by users. | Intermediate | Enterprise | 15 min. | • Rails | Lesson |

# OWASP 2017 #9: Using Known Vulnerabilities

Keep tabs on outdated dependencies with known security weaknesses.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Suspicious Packages | Find and exploit vulnerabilities in an outdated Java Spring application. | Beginner, Intermediate | Enterprise | 15 min. | • .NET<br>• Golang<br>• Java<br>• Javascript<br>• PHP<br>• Python (Django)<br>• Python (Flask)<br>• Rails<br>• Scala | Lesson |
| Outdated Challenge | Find and upgrade an outdated, vulnerable dependency. | Advanced | Enterprise | 20 min. | • .NET<br>• Java<br>• Python (Flask)<br>• Rails | Challenge |

## OWASP 2017 #10: Lack of Sufficient Logging + Monitoring

Rate-limit sensitive actions and block attacks as they happen.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Slow Down | Brute-force a user's password on a non-rate-limited login page. | Intermediate, Advanced | Enterprise | 15 min. | • .NET<br>• Golang<br>• Java<br>• Javascript<br>• PHP<br>• Python (Django)<br>• Python (Flask) | Lesson |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| | | | | | • Rails<br>• Scala | |
| Brute Force Challenge | Bruteforce a user's credentials, then implement rate limiting. | Advanced | Enterprise | 15 min. | • .NET<br>• Java<br>• Python (Flask)<br>• Rails | Challenge |

## CWE-319: Cleartext Transmission of Sensitive Data

Sensitive traffic is sent over unencrypted HTTP.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| See-through Traffic | Sniff a user's credentials via insecure HTTP requests. | Beginner, Intermediate | Enterprise | 15 min. | • Golang<br>• Javascript | Lesson |

## CWE-601 #22: Open Redirects

Unchecked URL redirection to untrusted sites.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| The Art of Redirection | URL redirects cause users to automatically visit untrusted sites. | Beginner, Intermediate | Enterprise | 15 min. | • Javascript | Lesson |
| No Going Back Challenge | Work around a URL redirect safety check, then provide an allowlist. | Advanced | Enterprise | 15 min. | • Javascript | Challenge |

## CWE-352 #12: Cross-Site Request Forgery

Forge valid requests from authenticated users.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Forging User Requests | Cause a user to take unexpected, pre-authenticated actions. | Beginner, Intermediate | Enterprise | 15 min. | • Golang<br>• Python (Django)<br>• Rails | Lesson |

## CWE-1021: Improper Restriction of Frames

A lack of response header allows the application to load in an external frame.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| You've Been Framed | A clickjacking attack tricks users into taking intended actions. | Beginner, Intermediate | Enterprise | 15 min. | • Rails | Lesson |

## Common React Pitfalls

Vulnerabilities frequently encountered in ReactJS application development.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| React String Sanitization | Cause XSS through improper sanitization and poor variable handoff with React. | Beginner | Community | 15 min. | • Javascript | Lesson |
| Sneaky Links | Learn about a feature of HTML that can leave your React app open to XSS. | Intermediate | Community | 20 min. | • Javascript | Lesson |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Dangerously Set HTML Links | React's dangerouslySetInnerHtml and markdown rendering craft a malicious href. | Intermediate | Community | 15 min. | • Javascript | Lesson |

## Basic Terminal Usage

Navigate around system file and folders using the bash shell.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Introduction to Bash 1 | Shell commands to navigate around directories and modify files. | Beginner | Community | 10 min. | • Bash Shell | Lesson |
| Introduction to Bash 2 | Navigate files and folders more efficiently, and search for file contents. | Beginner | Community | 10 min. | • Bash Shell | Lesson |
| Introduction to Bash 3 | Preview the contents of files; create new folders and move files around. | Beginner | Community | 10 min. | • Bash Shell | Lesson |

## Intermediate Terminal Usage

Additional bash skills: text editing, scripting, and command line tools.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Nano for text editing | Use Nano, a basic text editor, for creating and editing files. | Beginner | Community | 15 min. | • Bash Shell | Lesson |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Encrypting, encoding, and hashing | Common encoding patterns, cryptographic techniques, and command line tools. | Beginner | Community | 15 min. | • Bash Shell | Lesson |
| Introduction to bash scripting | Automate tasks by writing and running basic scripts in bash. | Beginner | Community | 15 min. | • Bash Shell | Lesson |

## Juice Shop

Very vulnerable MEAN web app full of practice challenges.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Hidden Pages | Find (not so) carefully hidden pages. | | Community, Enterprise | 15 min. | • Javascript | Challenge |
| Login Bypass | Log in with other users' accounts via SQL injection. | | Community, Enterprise | 20 min. | • Javascript | Challenge |
| XSS Levels | Reflected and persistent XSS attacks of increasing difficulty. | | Community, Enterprise | 25 min. | • Javascript | Challenge |
| Credentials Dump | Retrieve a list of all user credentials via SQL injection. | | Community, Enterprise | 20 min. | • Javascript | Challenge |
| Account Hijack | Access and modify another user's shopping cart. | | Community, Enterprise | 20 min. | • Javascript | Challenge |
| Confidential Documents | Access unprotected confidential documents. | | Community, Enterprise | 25 min. | • Javascript | Challenge |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Open Redirects | Redirect from the Juice Shop to external untrusted sites. | | Community, Enterprise | 20 min. | • Javascript | Challenge |
| File Uploads | Improper input validation in user file uploads. | | Community, Enterprise | 20 min. | • Javascript | Challenge |
| Error Handling Challenge | Provoke an error that is not very gracefully handled. | | Community, Enterprise | 20 min. | • Javascript | Challenge |

## Forensics

Work with disk images and investigate the contents of system files.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Creating a Disk Image | Learn how to acquire a disk image using the forensic tool dc3dd. | | Enterprise | 15 min. | • Forensics | Lesson |
| Working with a Disk Image | Hard disk image analysis with the sleuthkit (TSK), a standard forensic tool. | | Enterprise | 15 min. | • Forensics | Lesson |
| Analyzing Log Files | Uncover evidence of an attack by analyzing a system's logs. | | Enterprise | 15 min. | • Forensics | Lesson |
| Metadata with ExifTool | View and modify the metadata associated with multimedia files. | | Enterprise | 15 min. | • Forensics | Lesson |

## Best Practices

Prevent the compilation of programs using unsafe functions with banned function headers.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Forbidden Functions | A banned function header prevents the compilation of programs using unsafe functions. | | Enterprise | 15 min. | • C++ | Lesson |
| Time and Time Again | A side-channel timing attack reveals sensitive information. | | Enterprise | 20 min. | • C++ | Lesson |

## Bitwise Shifts

Prevent the compilation of programs using unsafe functions with banned function headers.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Shifty RSA | An RSA implementation allows for invalid bit shifts. | | Enterprise | 15 min. | • C++ | Lesson |

## Compilers

Sensitive data leaked through insecure compiler optimizations.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Optimal Memory | A program that checks user input against a password file leaves sensitive data in memory. | | Enterprise | 20 min. | • C++ | Lesson |

## Files

Overwriting system files through race conditions.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Race Condition | An encryption program allows system files to be overwritten through a race condition. | | Enterprise | 20 min. | • C++ | Lesson |

## Heap Overflows

Unsafe character arrays, null terminators, and use of GDB to examine heap memory.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Take Note! | A note-taking program copies strings to the heap unsafely. | | Enterprise | 25 min. | • C++ | Lesson |

## Integer Overflows

Overflowing short integers and wraparound of unsigned integers.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Short Scores | A program to add golf scores is susceptible to overflowing. | | Enterprise | 15 min. | • C++ | Lesson |
| Unsigned Messages | A message parsing utility uses unsafe range checks. | | Enterprise | 20 min. | • C++ | Lesson |
| Coercive Login | Use integer coercion to log in as an admin user. | | Enterprise | 15 min. | • C++ | Lesson |

## Iterators and Sequence Containers

Leaked data through unsafe iteration and unsafe access of container indices.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Go the Distance (but not too far) | A program to parse input from a file iterates unsafely, resulting in leaked data. | | Enterprise | 20 min. | • C++ | Lesson |
| Pinball Wizard | A program to display high scores trusts user input, leading to multiple vulnerabilities. | | Enterprise | 25 min. | • C++ | Lesson |

## Overreads

Buffer overruns common to parsing utilities, and the dangers of relying on side effects.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Passed Date | A date parsing and formatting utility allows for buffer over-reads. | | Enterprise | 20 min. | • C++ | Lesson |
| Trivial Side Effects | A trivia program reveals sensitive data by poorly tracking player scores. | | Enterprise | 20 min. | • C++ | Lesson |

## Stack Overflows

Unsafe string copying and incomplete string comparisons.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Triple Word Score | A Scrabble score calculator copies user input unsafely. | | Enterprise | 20 min. | • C++ | Lesson |

## Memory Management

Accessing freed memory when unsafe parsing keeps deallocated pointers accessible.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Use After Free | A note-taking program copies strings to the heap unsafely . | | Enterprise | 20 min. | • C++ | Lesson |

## Threads

Poor use of mutex locks leads to exceptions.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Lock Down the Threads | Poor use of mutex locks leads to exceptions. | | Enterprise | 15 min. | • C++ | Lesson |

## User Data Privacy

Improve data handling practices on an app that tracks users' jogging habits

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| PII Storage | De-identify and limit or do not collect sensitive user data. | | Enterprise | 15 min. | • Javascript | Lesson |
| Informed Consent | Let users actively choose to give consent for clear, specific data collection, as well as opting out. | | Enterprise | 15 min. | • Javascript | Lesson |
| Access and Erasure | Let users see their stored data, delete their data, and have the 'right to be forgotten. | | Enterprise | 15 min. | • Javascript | Lesson |
| Rectification | Let users supply corrections to their data. | | Enterprise | 15 min. | • Javascript | Lesson |
| Data Portability | Let users export their data in a machine-readable format. | | Enterprise | 15 min. | • Javascript | Lesson |

## Mobile Security

Important concepts related to secure mobile app development.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Custom URL Handling | How to handle custom protocol in different operation systems and launching applications using custom browser protocols. | Beginner, Intermediate | Enterprise | 20 min. | • kotlin<br>• swift | Lesson |
| Secrets Storage | How to securely store secret credentials or API keys that you need to have in your app. | Beginner, Intermediate | Enterprise | 20 min. | • kotlin<br>• swift | Lesson |
| Forced Browsing & API Security | Prevent attackers from accessing resources that they should not be able to access. | Beginner, Intermediate | Enterprise | 20 min. | • kotlin<br>• swift | Lesson |
| Mobile Logging | Best practices for logging when developing a mobile application. | Beginner, Intermediate | Enterprise | 20 min. | • kotlin<br>• swift | Lesson |

## Security Labs — Getting Started

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Lesson Zero | Explore the Security Labs and learn how to use lesson step features. This lab helps developers become familiar with use of the lab environment, so they can be successful while finding and remediating vulnerabilities in the lessons. | Beginner | Enterprise | 15 min. | • .NET | Lesson |

## OWASP API #2: Broken User Authentication

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Terrible Password | Learn how attackers crack a password hashed with SAH512 using online rainbow tables, also learn how to avoid it. | Intermediate | Enterprise | 15 min. | • .NET | Challenge |
| Really, really bad passwords | Best practices when it comes to user passwords, including sensible ways and really, really bad ways to enforce user password requirements. | Beginner | Enterprise | 15 min. | • .NET | Lesson |

## OWASP API #1: Broken Object Level Authorization

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Stronger IDs | Learn how to create stronger object IDs to discourage malicious users from being able to attack your API at the object level. | Intermediate | Enterprise | 20 min. | • .NET | Lesson |
| One ID to Access All Objects | Learn how to remediate object-level authorization vulnerabilities to protect the objects from unauthorized clients. | Intermediate | Enterprise | 20 min. | • .NET | Lesson |

## OWASP API #3: Excessive Data Exposure

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Bugs in Debug | In this lesson, you will learn:<br>• How exception messages can reveal information to attackers<br>• How the debug mode compromises the security<br>• To protect your API from excessive data exposure | Intermediate | Enterprise | 15 min. | • .NET | Lesson |
| Revealing Schemas | In this lesson, you will learn:<br>• The JSON responses in API reveal information to attackers | Intermediate | Enterprise | 20 min. | • .NET | Lesson |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| | • The difference between response models and the domain models<br>• How to organize your code to avoid revealing internal information<br>• How to organize your code to increase maintainability | | | | | |

## OWASP API #4: Lack of Resources & Rate Limiting

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Slow Down | In this lesson, you will learn:<br>• How attackers perform brute-force attacks<br>• How to set up password requirements<br>• How to apply rate limit to the login process | Intermediate | Enterprise | 10 min. | • .NET | Lesson |
| Brute Force | In this challenge, you will learn:<br>• How attackers guess passwords by brute-force attacks<br>• How the implementation of rate-limiting prevents password guessing | Intermediate | Enterprise | 15 min. | • .NET | Challenge |
| Denial of Service | Don't let attackers drain your computational resources, be prepared with this lesson where you will learn about Denial of Service and how to avoid it. | Intermediate | Enterprise | 25 min. | • .NET | Lesson |

## OWASP API #5: Broken Function Level Authorization

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Neglected Endpoints | In this lesson, you will learn:<br>• How API endpoints structure can be exploited by attackers<br>• How attackers scan for hidden API endpoints<br>• How to implement policies to protect API endpoints | Intermediate | Enterprise | 20 min. | • .NET | Lesson |

## OWASP API #6: Mass Assignment

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Bad Design Compromises Security | In this lesson, you will learn:<br>• How API payloads can be exploited by attackers<br>• How to control model properties to avoid unauthorized access<br>• How to apply the *separation of concern* principle to your models | Intermediate | Enterprise | 15 min. | • .NET | Lesson |

## OWASP API #8: Injection

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Bobby Tables | In this challenge, you will use your knowledge to:<br>• Query the system tables to access database table and column information using SQL injection<br>• Get access to private information using data collected from system tables | Intermediate | Enterprise | 20 min. | • .NET | Challenge |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| | Parameterize a query to avoid SQL Injection | | | | | |
| Parameterize all the things | In this lesson, you will learn:<br>• How to verify that queries are vulnerable to SQL Injection<br>How to parameterize the queries to avoid SQL Injection | Intermediate | Enterprise | 15 min. | • .NET | Lesson |
| Own the database | In this lesson, you will learn:<br>• How SQL injection attacks work<br>• How attackers get access to sensitive content using SQL Injection<br>The best practices to avoid SQL Injection | Intermediate | Enterprise | 20 min. | • .NET | Lesson |

## OWASP API #7: Security Misconfiguration

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| XML is always a | In this challenge, you will use your knowledge to:<br>• Get access to a source code file through XML<br>• Remediate the XML external entity vulnerability | Intermediate | Enterprise | 20 min. | • .NET | Challenge |

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| eXternal Entity (injection) | In this lesson, you will learn:<br>• How the XML external entity attack works<br>• How attackers get access to private resources using malicious XML payloads<br>• How to avoid the XML external entity attack | Intermediate | Enterprise | 15 min. | • .NET | Lesson |
| Secret Admin | In this challenge, you will use your knowledge to:<br>• Decode JSON Web Tokens (JWT)<br>• Modify token claims to authenticate as an admin user<br>• Modify old tokens to make them valid | Intermediate | Enterprise | 20 min. | • .NET | Challenge |
| Jot down this key | In this lesson, you will learn:<br>• How to Decode JSON Web Tokens (JWT)<br>• How attackers modify JWTs<br>• The best practices to protect secret keys | Intermediate | Enterprise | 15 min. | • .NET | Lesson |

## OWASP #10: Server-Side Request Forgery

SSRF flaws can occur when a web application fetches a remote resource without validating the user-supplied URL.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Get there from here | • How server-side request forgery works.<br>• How attackers get access to sensitive content using SSRF.<br>• The best practices to avoid SSRF vulnerabilities. | Intermediate | Enterprise | 15 min. | • Java<br>• Javascript | Lesson |

## OWASP API #9: Improper Assets Management

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Unprotected deployments | In this lesson, you will learn:<br>• How attackers scan API routes to find other environments<br>• How security failures in non-production environments can affect the entire system<br>• How to protect API deployments in multiple environments | Intermediate | Enterprise | 20 min. | • .NET | Lesson |

## OWASP API #10: Insufficient Logging & Monitoring

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| The Importance of Logging and Monitoring | In this lesson, you will learn:<br>• The importance of logging for detecting some attacks on your API<br>• How to produce good log entries<br>• How logging is related to monitoring tools | Intermediate | Enterprise | 15 min. | • .NET | Lesson |
| Logging in the API Infrastructure | In this lesson, you will learn:<br>• The different levels to implement logging<br>• How to implement logging at the API infrastructure level<br>• The importance of monitoring tools to detect anomalous activities in your API | Intermediate | Enterprise | 15 min. | • .NET | Lesson |

## OWASP #9: Security Logging and Monitoring Failures

Rate-limit sensitive actions and block attacks as they happen.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Hold the Line | Learn how attackers can use CRLF injection to flood log files with false events and how to remediate CRLF vulnerabilities (CWE-117). | Beginner | Enterprise | 10 min. | • .NET<br>• Java<br>• Javascript | Lesson |

## OWASP #4: Insecure Design

Failing to initially think about and address security vulnerabilities at the design phase can lead to vulnerabilities and defects.

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Valid Deficit | In this lesson, you will learn how insufficient validation of user data can lead to impacts on the usability and integrity of a site. | Beginner | Enterprise | 10 min. | • Javascript | Lesson |
| Making Secure Decisions | In this lesson, you will learn that insecure design decisions can lead to vulnerabilities at every level of an application. | Beginner | Enterprise | 10 min. | • .NET | Lesson |
| Valid Deficit | In this lesson, you will learn how insufficient validation of user data can lead to impacts on the usability and integrity of a site. | Beginner | Enterprise | 10 min. | • .NET | Lesson |

## OWASP #1: Broken Access Control

| Lab | Description | Difficulty | Edition | Est. Time | Languages | Type |
|---|---|---|---|---|---|---|
| Loose Lips Sink Servers | In this lesson, you will learn how information leakage can lead to the exposure of sensitive data. | Beginner | Enterprise | 10 min. | • Javascript | Lesson |