Web Application Security

Description

This class is designed to raise awareness of the most dangerous, costly, and prevalent threats to the security of websites, both internal and external. Students will not only learn to identify these threats but will also leave the class with practical and workable defenses for each one.

You'll be given an e-commerce website to practice on. After learning about each topic, you'll carry out the attack on that actual website. Then you'll harden your website against that attack using what was learned in the lecture and watch the attack fail the second time. This hands-on approach will anchor your understanding of web application security.

The lectures are packed with interesting stories from newspaper headlines and videos as practical examples of each of the attacks. You will learn how the hackers successfully carried out those attacks, including the tools used so that we learn exactly how to defend our sites against these attacks and ones like them.

We will focus on OWASP's Top Ten Security Threats, seeing examples, learning hackers' methods, and the best practices for protecting our sites against similar attacks.

Objectives

At the end of this course, students will be able to:

- Identify and understand the most prevalent and dangerous security threats in today's web applications.
- Know how to defend his or her organization against each one of them.
- Explain cryptographic ciphers like SHA1, MD5, Blowfish, and RSA and know which ones are stronger.
- Know how to best use .Net to manage security including creating user accounts, logging in/out, and handling forgotten passwords.

Topics

- Overview of web security
- Overview of the OWASP top ten vulnerabilities
- Clickjacking
- Phishing
- Denial of service attacks
- A10 Unvalidated redirects and forwards
- A9 Insufficient transport layer protection
- A8 Failure to restrict URL access
- Cryptography overview
- A7 Insecure cryptographic storage
- A6 Security misconfiguration
- Password management
- A5 Cross site request forgery
- A4 Insecure direct object references
- Padding oracle attack
- Information leakage and improper error handling
- A3 Broken authentication and session management
- A2 Cross site scripting
- A1 Injection flaws
- Bringing the top ten together – best practices for security overall.

Audience

This class is most appropriate for intermediate to advanced developers who want to enhance their knowledge of security threats and who want to know the practical steps on how to protect their web applications.

Duration Five days

Web Application Security Essentials
Course Outline

Overview of web security

A. PCI DSS
B. OWASP publications and projects
C. Overview of the OWASP top ten vulnerabilities

Secure Coding Principles

A. Systemic views to security
B. Bolted-on security
C. Baked-in security
D. Best practices

Threat Risk Modeling

A. Proper analysis of a web project
B. STRIDE
C. DREAD
D. Threat Trees
E. How to predict attack vectors

A10 Unvalidated redirects and forwards

A. Real-world example
B. How attackers do it
C. How we protect ourselves
D. Whitelists and mapping

Fuzz testing

A. What is it?
B. How to use it.

A9 Insufficient transport layer protection

A. Real-world example
B. How attackers do it
C. How we protect ourselves
D. AJAX, web services

Denial of service attacks

A. Real-world example
B. How attackers do it
C. How we protect ourselves

A8 Failure to restrict URL access

A. Real-world example
B. How attackers do it
C. How we protect ourselves

Cryptography overview

A. Vulnerability examples
B. History
C. Functions of cryptography
D. Types of encryption with ciphers
E. When to use each type
F. Which ciphers to avoid
G. How SSL/TLS works

A7 Insecure cryptographic storage

A. Real-world example
B. How attackers do it
C. War stories of lost & stolen data
D. How we protect ourselves
E. How to encrypt web.config
F. Demo of cracking passwords

Phishing attacks

A. Real-world example

B. How attackers do it
C. How we protect ourselves

A6 Security misconfiguration

A. Real-world example
B. How attackers do it
C. How we protect ourselves
D. Best practices for installation
E. Internal attackers

ASP.NET Authentication and Authorization

A. Deep dive into how ASP.NET implements
B. Creating the database
C. Setting up web.config
D. Using it with MVC and WebForms

Password management

A. Real-world story of bad password policies
B. How to use.Net membership & role providers
C. Easy and secure user controls
D. Best practices for security architects

A5 Cross site request forgery

A. Real-world example
B. How attackers do it
C. Demo of a CSRF attack
D. How NOT to protect against CSRF
E. How we protect ourselves
F. Synchronizer token pattern made easy

ClickJacking

A. Real-world example
B. How attackers do it
C. How we protect ourselves

A4 Insecure direct object references

A. Real-world examples
B. How attackers do it
C. How we protect ourselves
D. Mapping references

Web services security

A. Real-world example
B. How it differs from web site security
C. Special considerations for XML web services
D. How attackers do it
E. How we protect ourselves

Padding oracle attack

A. Real-world example
B. How attackers do it
C. How we protect ourselves
D. The POET tool

Information leakage and improper error handling

A. Real-world example
B. How attackers do it
C. How we protect ourselves

Code access security

A. Definition
B. Implementation using .Net and C#

Role-based security

A. Definition
B. Using the provider model
C. Extending the provider model

D. Implementation using .Net and C#

A3 Broken authentication and session management

A. Real-world example
B. How sessions & cookies work
C. How attackers exploit
D. How we protect ourselves

Visual Studio Code Analysis

A. Demo of the tool
B. How to use it to analyze your site's security weaknesses
C. Best practices

A2 Cross site scripting

A. Real-world example
B. How attackers do it
C. Live demo of a XSS attack
D. Reflected vs. stored attacks
E. How we protect ourselves
F. Installing and using the Anti-XSS Toolkit

A1 Injection flaws

A. Sample attack vectors
B. How attackers do it
C. How we protect ourselves
D. Parameterized queries
E. ORMs (LINQ-to-SQL, EF, NHibernate)
F. Stored procedures
G. Whitelisting vs. blacklisting

Bringing the top ten together – best practices for security

A. Review of the top ten (and more)
B. Handout: Checklist for protecting against all top ten attacks
C. Handout: Many links to tools, articles and further study