



Web Vulnerabilities

Module 9

Module 9 Agenda

- Web Vulnerabilities
- OWASP
- OWASP Top Ten

Web Vulnerabilities

Web Vulnerabilities

- As we know, vulnerabilities are weaknesses within a computer that can compromise its security.
- **Web Vulnerabilities** are weaknesses within a website or web application that can allow hackers access to a wide variety of features.

Web Vulnerabilities

- There is a list on the ten most common web vulnerabilities.
- The list is compiled and maintained by an organization named The Open Web Application Security Project (OWASP).
- **OWASP** is a non-profit organization that strives to strengthen security across the web.

*The OWASP Top 10 can be found on OWASP's website: owasp.org

Open Web Application Security Project (OWASP)

OWASP

The top ten web vulnerabilities earn their place on the list based on the following metrics...

- **Exploitability:**

How easily can the vulnerability can be exploited?

- **Prevalence**

How widespread is the vulnerability?

- **Detectability:**

Is it difficult to detect this vulnerability?

- **Impact:**

Vulnerability impact in the real world.

OWASP

The top ten web vulnerabilities are. . .

1. Injections
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities
5. Broken Access Control
6. Security Misconfiguration
7. Cross-Site Scripting XSS
8. Insecure Deserialization
9. Using Components with Known Vulnerabilities
10. Insufficient Logging & Monitoring


OWASP

OWASP Top 10 Overview by Rapid7



AN INTRO TO
OWASP TOP 10 

- ① Injection
- ② Broken Authentication
- ③ Sensitive Data Exposure
- ④ XXE
- ⑤ Broken Access Control
- ⑥ Security Misconfig
- ⑦ XSS
- ⑧ Insecure Deserialization
- ⑨ Known Vulns
- ⑩ Insufficient Logging & Monitoring

owasp.org |  rapid7.com/irg/insightappsec

RAPID7

<https://youtu.be/-7B9Ej4BFgw>



OWASP Top Ten

OWASP Top Ten

1. SQL Injection

- Definition: Taking advantage of input fields where an attacker can “inject” malicious code which are basically database queries used to access data that is not meant to be seen.
- Ex: You are a hacker that was hired to test the security of an online store, it is a fairly new store so their security isn't well-established yet. You conduct a simple security test and go to the login screen and enter some basic SQL commands. You notice that your input is not being filtered, and you are then able to access other people's accounts and data without requiring a password.

OWASP Top Ten

Watch this video from “The Modern Rogue” to learn more about SQL injections (optional)...



OWASP Top Ten

2. Broken Authentication

- Definition: An unwanted user can access a web application without authorization. This is a result of poor authentication and security measures related to session management.
- Examples: Easily guessed default passwords, weak session tokens, credential stuffing using lists of common passwords, using only one metric to authenticate user.
- Protection: Multi factor authentication, strong password complexity, password checking, limit failed login, random session tokens.

OWASP Top Ten

3. Sensitive Data Exposure

- Definition: Hackers can steal sensitive data (financial, healthcare, banking information) due to the improper protection by web applications.
- Examples: Exposure of credit card numbers stored on websites, web applications not using encryption to store data.
- Protection: Apply controls to sensitive data, use strong encryptions for data, encryption of data at rest and in transit, do not store sensitive data if you do not have to.

OWASP Top Ten

4. XML External Entities

- Definition: Taking advantage of a web applications XML processors that allows access to XML documents. XML documents stores data.
- Examples: Attacker can extract data from server, attacker can change the lines of code to probe private networks
- Protection: upgrade XML processors frequently, disable XML external entity, verify XML files frequently, avoid storing sensitive data

OWASP Top Ten

5. Broken Access Control

- Definition: Web applications do not properly assign user privileges. In the worst possible scenario, even a low level guest user might be able to read or write to files that are meant for a system admin.
- Examples: Access of other users' accounts, view sensitive files, modify users' data, change access rights, gain access to prepaid premium accounts.
- Protection: Ensure all authorized functions have an authentication check, change default names, set up access controls properly.

OWASP Top Ten

6. Security Misconfiguration

- Definition: When web applications are not configured properly, the user does not change the default settings, or any other security mishap that came about during the development of the application.
- Examples: Incorrect file permissions, not resetting default usernames and passwords, unnecessary features that compromise security and are of no critical importance are enabled.
- Protection: Reset default username and passwords, no open ports or active rogue accounts (aka minimal platform)

OWASP Top Ten

7. Cross-Site Scripting XSS

- Definition: When an attacker injects an untrusted snippet of code into a web application. Similar to an injection but has more of a permanent and widespread impact.
- Examples: This can be performed through links sent to the web application victim to spread. It also can be done by adding a link directly to the website for users to visit. (redirecting to a malicious site)
- Protection: Implementing input validation and output encoding. Separation of untrusted data from browser content.

OWASP Top Ten

8. Insecure Deserialization

- Definition: Deserialization is the process of turning a stored object back into its original format for later use. Insecure deserialization otherwise known as “object injection” can allow an attacker to pass harmful data into the web application.
- Examples: An attacker alters a serialized object to give themselves root privileges.
- Protection: Implement integrity checks, logging deserialization attempts and failures, monitoring incoming and outgoing traffic from servers that deserialize.

OWASP Top Ten

9. Using Components with Known Vulnerabilities

- Definition: When building a web application a developer will rarely create every component themselves. Sometimes developers either wittingly or unwittingly will use components with known vulnerabilities and jeopardize the security of their work.
- Examples: Web servers have common vulnerabilities, company fails to stay up to date on the latest patches and software bugs.
- Protection: download software from trusted sources, do research on clients and servers, plan to monitor and patch software, use bulletin boards to check for common vulnerabilities in software.

OWASP Top Ten

10. Insufficient Logging & Monitoring

- Definition: When there is insufficient logging and monitoring, not enough is being done to detect a threat when it starts to attack you. No alarms or warnings go off to notify the owner that something is wrong, and when something does happen it is not being recorded, to prevent future mishaps.
- Examples: Your security personnel failed to take action when your intrusion detection systems went off.
- Protection: Log suspicious events with enough detail to be useful when referenced in the future. Stay vigilant on the front-lines when it comes to your organization's security.

Note: The average data breach goes about 200 days before being detected and is often discovered by a third party rather than from internal testing. (Source: OWASP)

To Do

- ❑ Complete Module 9 Activity 1
- ❑ Complete Module 9 Activity 2

End of Module 9!

What questions do you have?

Next Module Topic:

Network & Vulnerability Scanning!

Questions?

Contact IT-Adventures support staff!

email:

ita@iastate.edu

Your school's IP-Range can be found at:

<http://www.it-adventures.org/ip-ranges/>