Web Application Vulnerabilities & Security Models

Stefany C. Cando
scando@masonlive.gmu.edu
Secure Software Engineering
CYSE 411-001
Dr. Sun
Web Apps

- Military Systems
- Government Systems
- Financial Enterprises
- Medical Field
What are Web Applications?

Collection of hyperlink web pages.
- Potentially reduces loading time
- Faster performance

How does it work?
It predicts the users’ actions in order to launch requests for future Web page objects, so that it becomes available when the user needs it.
What is a Vulnerability?

Is a hole or breach in a system caused by:
- Poor design
- Configuration mistakes
- Insecure coding techniques
- Complexity of software
- Weak passwords
- Unchecked input

Affects the
- Confidentiality
- Integrity
- Availability
Input validation vulnerabilities

• Input validation is the ability to verify in each step of the execution progress whether the program is in a secure state by filtering and rejected the input information.

• Can be prevented by constrain input, reject known bad input, sanitize input, validate data for type, length and range.
SQL Injection

Alters the structure of the original SQL query by injecting SQL code in the input fields of the web form to gain unauthorized access.

Cross Site Scripting (XSS)

Application takes untrusted data and sends it to a web browser without proper validation. XSS allows attackers to execute scripts in the victim’s browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
PayPal Input Validation Vulnerability
(June 2015)

• Severity Level: Medium
• Exploitation Technique: Remote
• Problem: The validation and mail encoding web vulnerability allowed remote attackers to inject malicious script codes to the mail header of the portal mails that were being sent to users
• Areas Affected: “Adressbook” core module.
Session Management vulnerabilities

• Session management tracks the user’s activity across sessions of interaction with a website. The most common way to implement session management is by assigning each user to an ID or session token.

Token Generation:

This allows attackers to generate and use a valid token. The tokens can be created by using the user's information like the username or email address.

Session Management Mechanisms:

Even if a token is “completely secure,” attackers can still intercept it by exploiting unencrypted transmissions or weak mechanisms for preserving the keys that generate tokens.

Also, poor session termination policies may give the attacker opportunities to hijack the session.
Application Logical Vulnerabilities

Problems in the logic of the application and not the source code.

Sometimes web application store values and information in the URL, but what happens when we change that information? Would the website still accept the address?

What is a security model?

Scheme or plan for specifying, enforcing, and implementing security policies.
Security by Construction

- Aim to design secure web applications by ensuring that no potential vulnerabilities exist within the structure of the application.

**SIF (Servlet Information Flow):** uses JIF (Java with Information Flow control and access control):
- It labels user input, tracks information flow, enforces the security policies at compile and run time.

**Swift:** Parallel framework to enforce end to end information flow policies.

SIF & SWIFT can be used for building secure website free of input validation vulnerabilities.

**Haskell:** Strong typing language that deals with the weak typing feature of scripting languages. This helps to prevent XSS and SQL injection.
Security by Verification

- Aim to verify that the desired security properties are in place on the application. It could also be referred as a vulnerability assessment.

Program Analysis:

Static
- Data flow
- Pointer analysis
- String analysis
- WebSSARI, Pixy & Rubyx: Tool that applies static analysis and offers functions to secure the application

Dynamic
- Tracks information flow of the user’s input
- Perl: “Taint mode” taints user’s input and ensures that no external data can be used by critical functions

Program Testing:

Penetration Testing
- Web application scanners
- Attacks pattern generators
Security by protection

• Aim to protect vulnerable web applications from being exploited. It builds a run time environment that secures the application’s execution.

• Place safeguards: (proxy) Separate the application from other components in the web environment.

• Use the infrastructure components (language runtime or web browser) to monitor the behavior of the application and identify potential threats.

  - Nemesis: enforces authentication mechanism and keeps track of users credentials.
  - CLAMP: Assigns a virtual web server to each user and ensures that the user can only access his/her own data.
  - Arjun: sets a proxy for monitoring user’s behavior and detecting malicious activity.
  - BLOCK: Black-box approach for inferring the application and detecting violations.
Develop Secure Web Application

Secure software engineering by implementing well-structured processes, from the project formulation all the way through the design and implementation of the application, always having security in mind.

Secure the operational environment.

Penetrate and patch vulnerabilities.
Questions???
Works Cited


Works Cited


