### Report on 5 Real-World Web Application Attacks

#### 1. Local File Inclusion

**Description**: LFI exploits vulnerabilities in web applications that allow attackers to include files from the server’s filesystem. By manipulating input parameters, attackers can gain unauthorized access to sensitive files or execute malicious scripts.

* **Threats**:
  + Access to configuration files (e.g., config.php).
  + Exposure of sensitive information like database credentials.
  + Execution of malicious scripts that can compromise the server.
* **Vulnerabilities**:
  + Lack of proper input validation and sanitization.
  + Direct inclusion of user-supplied file paths.
* **Affected Security Pillars**:
  + **Confidentiality**: Unauthorized access to sensitive data.
  + **Integrity**: Potential manipulation or deletion of critical files.
* **Risks and Impact**:
  + **Legal**: Violations of data protection laws (e.g., GDPR).
  + **Financial**: Costs related to incident response, data breach notifications, and potential fines.
  + **Reputational**: Loss of customer trust due to data exposure.
* **Remediation Measures**:
  + Validate and sanitize all user inputs.
  + Restrict file access permissions to essential files only.
  + Disable functions like include, require if not necessary.
* **Mitigation Strategies**:
  + Use secure coding practices, such as whitelisting allowed file paths.
  + Implement a Web Application Firewall (WAF) to block malicious inputs.
  + Regularly update the web server and application framework.

#### 2. DNS Server Hijacking

**Description**: DNS Server Hijacking, also known as DNS Redirection, occurs when attackers compromise DNS servers to redirect users from legitimate websites to malicious ones.

* **Threats**:
  + Redirecting users to fraudulent websites to steal credentials or distribute malware.
  + Interception and modification of DNS traffic.
* **Vulnerabilities**:
  + Weak authentication mechanisms.
  + Outdated or unpatched DNS server software.
* **Affected Security Pillars**:
  + **Confidentiality**: Exposure of user data to malicious entities.
  + **Integrity**: Manipulation of DNS responses.
  + **Availability**: Disruption of legitimate DNS resolution.
* **Risks and Impact**:
  + **Legal**: Potential lawsuits from affected users and businesses.
  + **Financial**: Loss of business revenue and increased recovery costs.
  + **Reputational**: Severe damage to brand reputation and loss of customer trust.
* **Remediation Measures**:
  + Implement DNS Security Extensions (DNSSEC) to ensure DNS data integrity.
  + Regularly update and patch DNS server software.
  + Use strong authentication for DNS management.
* **Mitigation Strategies**:
  + Employ secure DNS protocols like DNS over HTTPS (DoH) or DNS over TLS (DoT).
  + Conduct regular security audits and monitoring of DNS infrastructure.
  + Set up alerts for unauthorized changes to DNS records.

#### 3. HTTP Response Splitting

**Description**: HTTP Response Splitting is a web vulnerability where attackers manipulate HTTP headers by injecting malicious content, leading to unintended server responses.

* **Threats**:
  + Injection of malicious scripts, potentially leading to Cross-Site Scripting (XSS).
  + Cache poisoning attacks, which can serve malicious content to users.
* **Vulnerabilities**:
  + Inadequate validation of user-supplied data in HTTP headers.
  + Failure to encode special characters in HTTP responses.
* **Affected Security Pillars**:
  + **Confidentiality**: Unintended data exposure.
  + **Integrity**: Corruption of server responses.
* **Risks and Impact**:
  + **Legal**: Non-compliance with privacy and security regulations.
  + **Financial**: Costs associated with attack mitigation and user compensation.
  + **Reputational**: Loss of user trust and credibility.
* **Remediation Measures**:
  + Validate and encode user input in HTTP headers.
  + Use frameworks and libraries that handle header encoding automatically.
  + Implement strict error handling and response sanitation.
* **Mitigation Strategies**:
  + Regularly review and update web server configurations.
  + Conduct periodic security testing to identify and fix vulnerabilities.
  + Use a Web Application Firewall (WAF) to detect and block malicious requests.

#### 4. Directory Traversal

**Description**: Directory Traversal attacks exploit input validation weaknesses to gain unauthorized access to files and directories outside the web root.

* **Threats**:
  + Access to sensitive files such as configuration files and user credentials.
  + Potential modification or deletion of critical system files.
* **Vulnerabilities**:
  + Insufficient input validation allowing the use of path traversal characters (../).
  + Misconfigured file permissions.
* **Affected Security Pillars**:
  + **Confidentiality**: Unauthorized data access.
  + **Integrity**: Possible alteration of sensitive files.
* **Risks and Impact**:
  + **Legal**: Breach of sensitive data, leading to legal repercussions.
  + **Financial**: Costs for data recovery, system hardening, and incident response.
  + **Reputational**: Negative impact on customer confidence and brand image.
* **Remediation Measures**:
  + Sanitize and validate all file path inputs.
  + Implement access controls to limit file system access to essential directories.
  + Use secure APIs that prevent directory traversal.
* **Mitigation Strategies**:
  + Apply the principle of least privilege for file access.
  + Regularly monitor and log file access patterns to detect anomalies.
  + Use relative paths and avoid using user-supplied data to construct file paths.

#### 5. Web Defacement Attack

**Description**: Web Defacement occurs when attackers gain unauthorized access to a web server and alter the website's content, often for propaganda or malicious purposes.

* **Threats**:
  + Unauthorized modification of web pages.
  + Spread of offensive or misleading information.
* **Vulnerabilities**:
  + Weak authentication and access controls.
  + Unpatched or outdated web server software.
* **Affected Security Pillars**:
  + **Integrity**: Alteration of website content.
  + **Availability**: Temporary loss of website availability.
* **Risks and Impact**:
  + **Legal**: Legal actions due to offensive or illegal content.
  + **Financial**: Costs for restoring the website and improving security measures.
  + **Reputational**: Significant damage to the organization's public image and trust.
* **Remediation Measures**:
  + Enforce strong authentication mechanisms, such as multi-factor authentication.
  + Regularly update and patch the web server and applications.
  + Implement Content Security Policy (CSP) headers to prevent unauthorized script execution.
* **Mitigation Strategies**:
  + Use a Web Application Firewall (WAF) to block unauthorized access attempts.
  + Set up intrusion detection and prevention systems.
  + Maintain regular backups of the website to facilitate quick recovery.