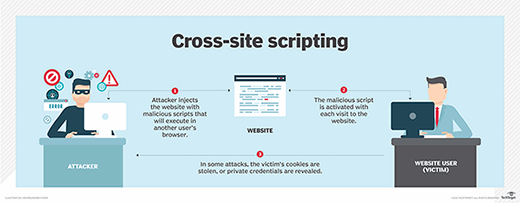
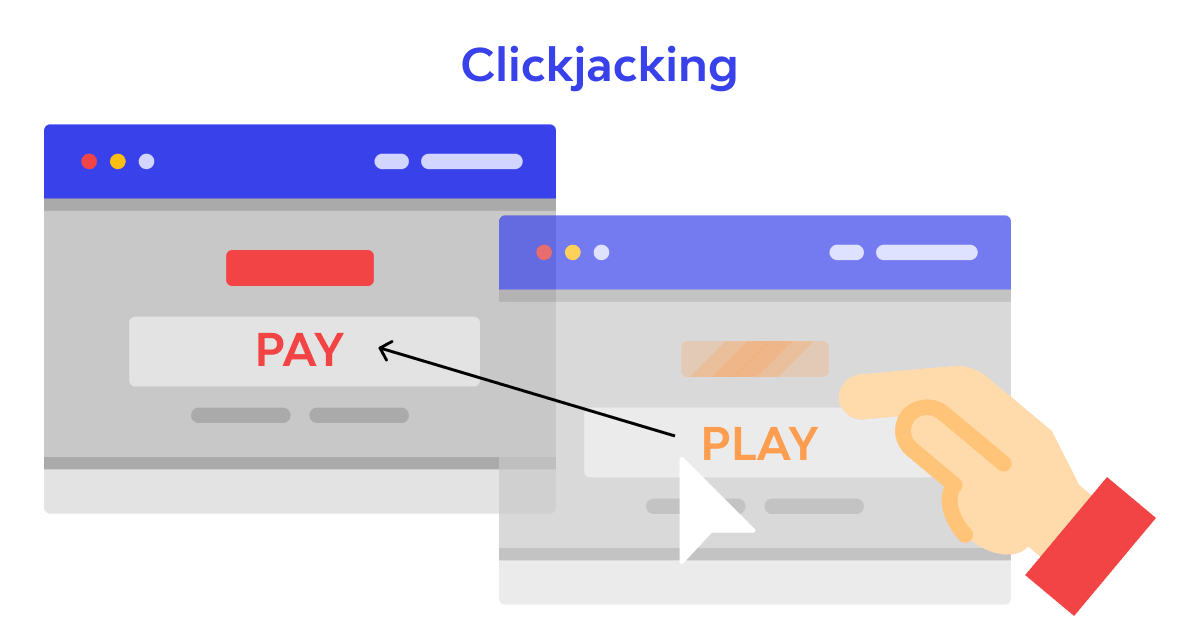
**Non top 10 Web vulnerabilities**

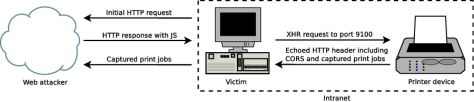
1. Cross-Site Script Inclusion (XSSI): XSSI vulnerabilities occur when an attacker can include untrusted content from a different domain into a web application. This can lead to cross-site scripting (XSS) attacks or information disclosure.



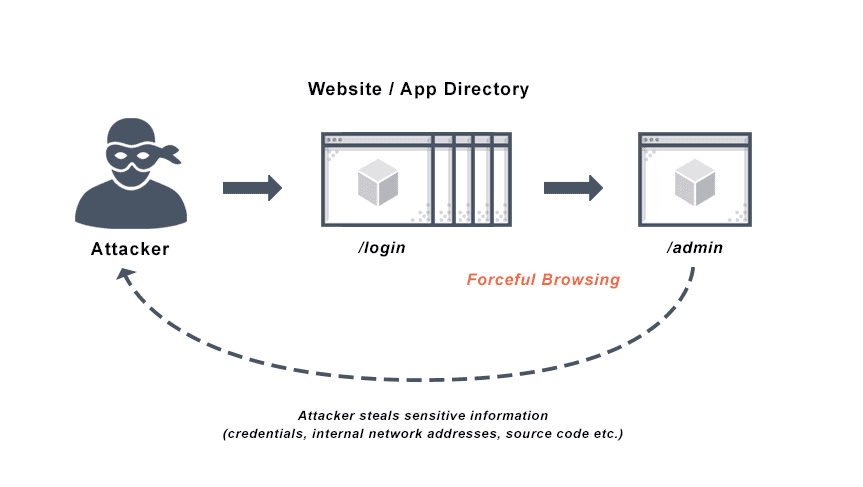
2. Clickjacking: Clickjacking involves tricking users into clicking on a malicious element disguised as a legitimate one on a webpage. This can lead to unintended actions performed by the user, such as unknowingly liking a post or sharing sensitive information.



3. Cross-Site Printing: Cross-Site Printing attacks occur when an attacker abuses the browser's print functionality to print arbitrary content. This can lead to the disclosure of sensitive information or the manipulation of printed documents.



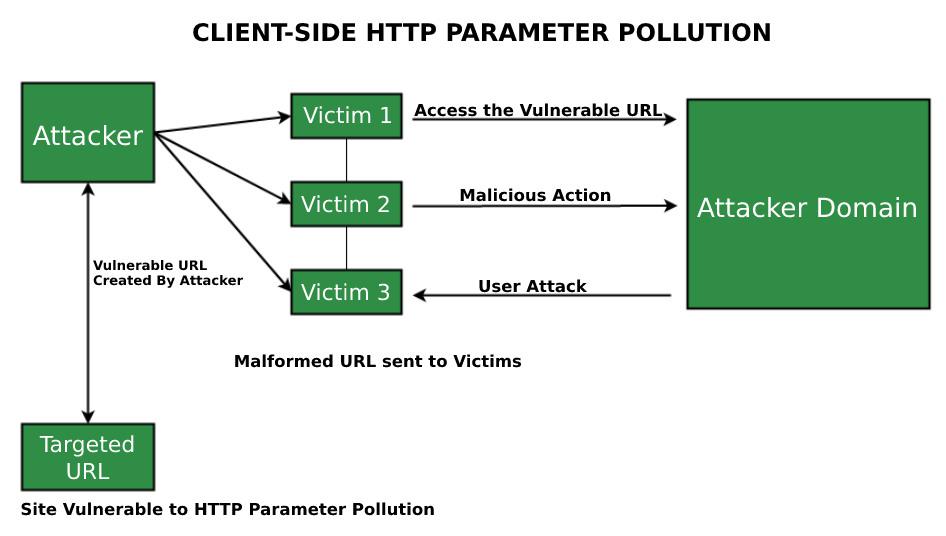
4. Predictable Resource Location: Predictable resource location vulnerabilities occur when sensitive resources, such as files or directories, have predictable or easily discoverable URLs. This can allow attackers to directly access or manipulate sensitive information.



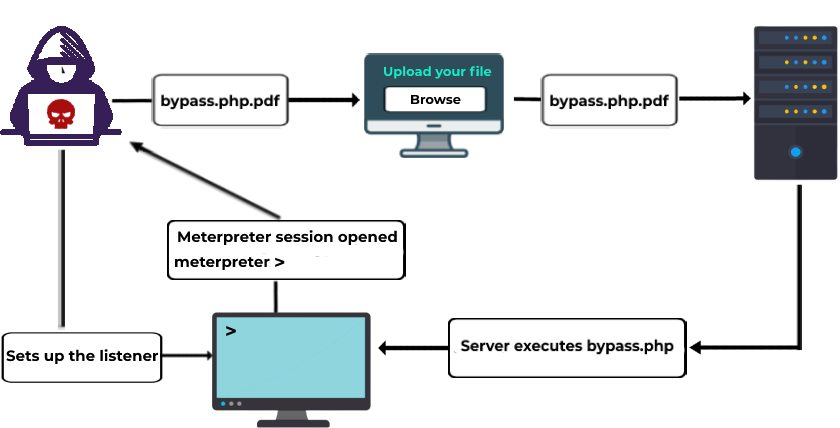
5. Security Through Obscurity: Relying on security through obscurity involves using secrecy or hidden mechanisms as the primary means of protection. This can lead to vulnerabilities being overlooked and exploited once the underlying mechanisms are discovered.



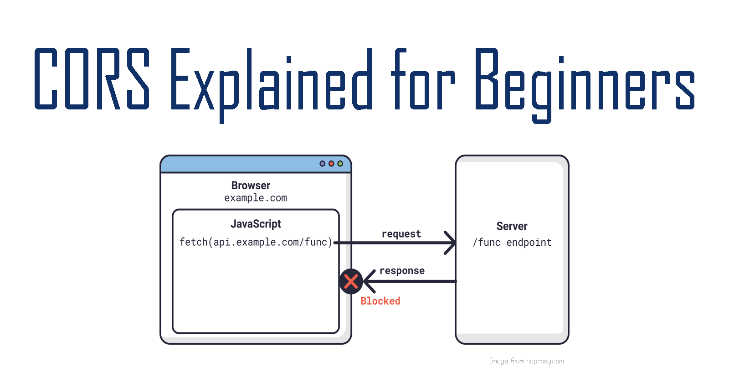
6. HTTP Parameter Pollution (HPP): HPP vulnerabilities occur when multiple values are assigned to the same parameter, causing conflicts or unexpected behavior in the application. This can lead to data corruption, privilege escalation, or bypassing security controls.



7. Unvalidated File Upload: Unvalidated file upload vulnerabilities allow users to upload arbitrary files to a web application without proper validation. This can lead to the execution of malicious code, denial of service, or unauthorized access to the server.



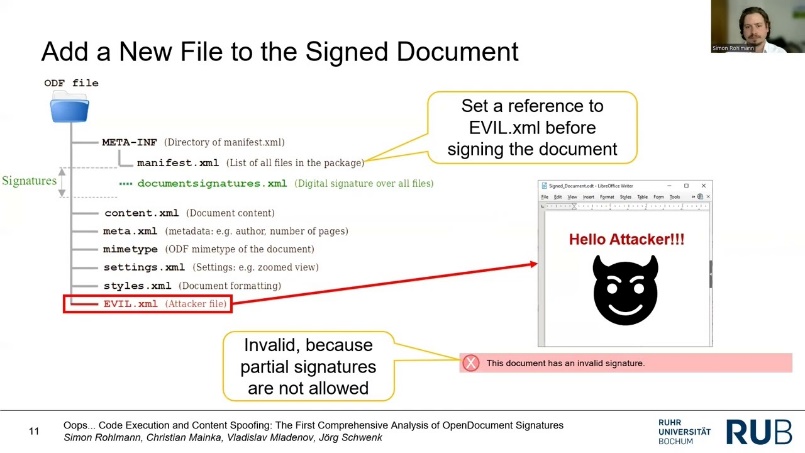
8. Insecure Cross-Origin Resource Sharing (CORS): CORS vulnerabilities occur when a web application improperly configures cross-origin resource sharing, allowing unauthorized domains to access sensitive data or perform privileged actions.



9. HTTP Response Splitting: HTTP response splitting vulnerabilities occur when an attacker can manipulate the application's HTTP response header, leading to cache poisoning, session hijacking, or injection attacks.



10. Content Spoofing: Content spoofing involves manipulating the content displayed to users, making it appear as if they are interacting with a legitimate website or application. This can lead to phishing attacks or the theft of sensitive information.



While these vulnerabilities may not always make it to the top 10 lists, they can still pose significant risks to web applications. It is important to employ secure coding practices, perform regular security assessments, and stay informed about emerging vulnerabilities to protect against a wide range of threats.