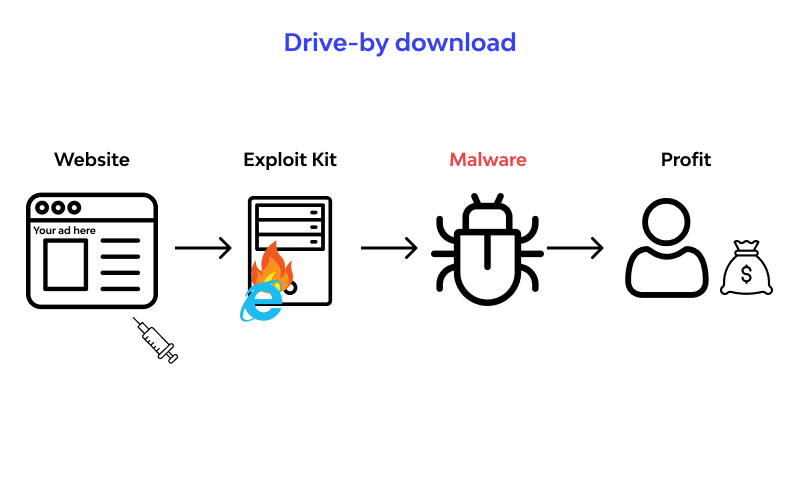
1. **Drive-by download**

Drive-by downloads occur when a user visits a website and a malicious agent downloads onto the victim’s computer automatically. It can happen when the user is downloading something else or upon opening an email, clicking a pop-up window, or merely visiting a page. Since drive-by attacks take advantage of latent security vulnerabilities in apps, browsers, and operating systems, it’s important to keep your environment up to date. Limiting the number of web plug-ins and applications you install also reduces your attack surface.

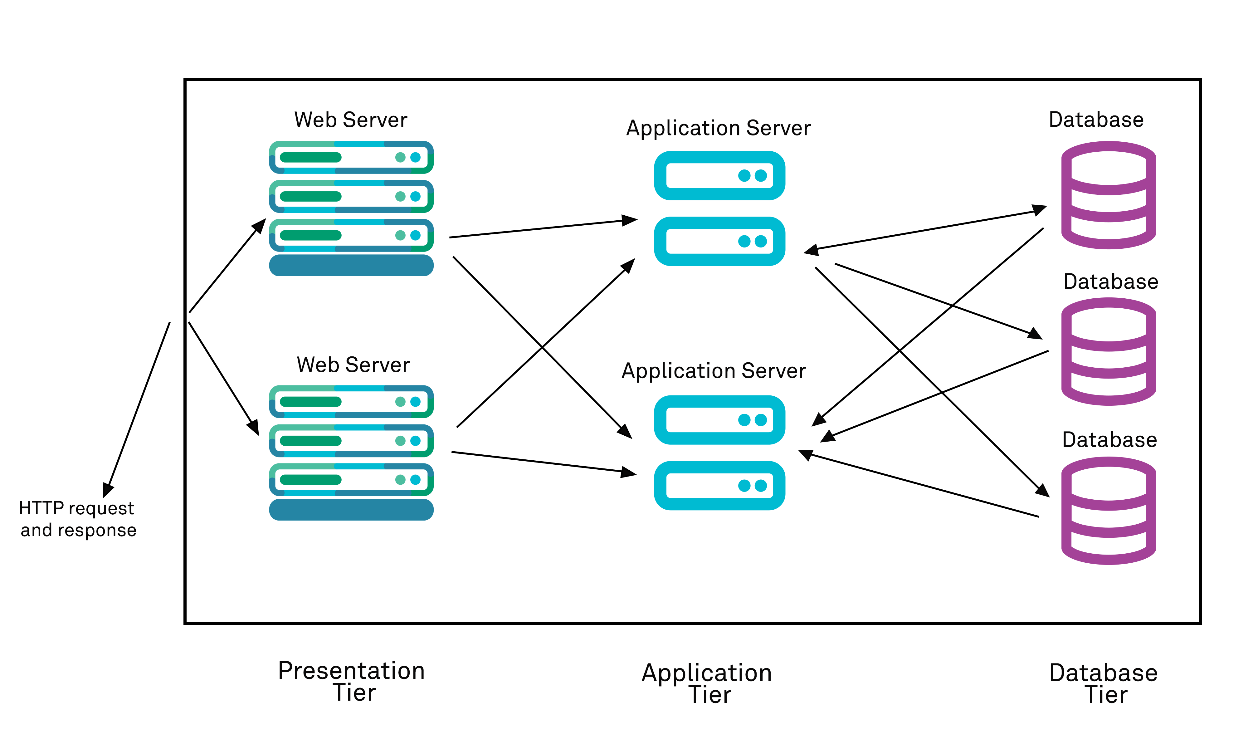


1. **HTTP Parameter Pollution (HPP):**

HPP attacks manipulate HTTP request parameters to confuse the application's processing, potentially causing vulnerabilities like privilege escalation or data corruption.

Server-Side Request Hijacking:

Attackers intercept and manipulate server-to-server communications to alter intended behavior, which can lead to data breaches or system compromise.

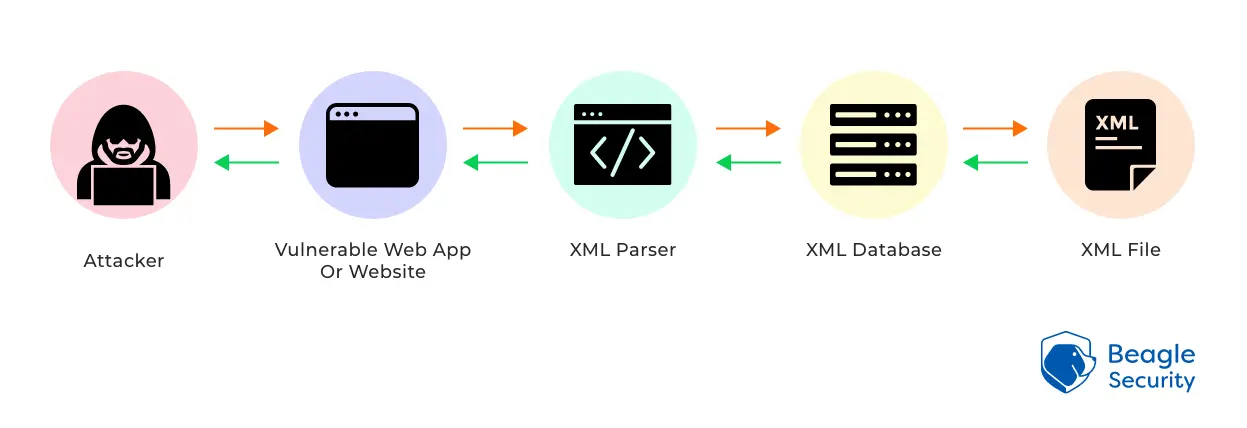


1. **XML Injection:**

XML Injection is an injection attack technique used to manipulate or compromise the logic of an XML application or document. It occurs when user-supplied input is not properly escaped or sanitized before being added to a web application’s XML documents.

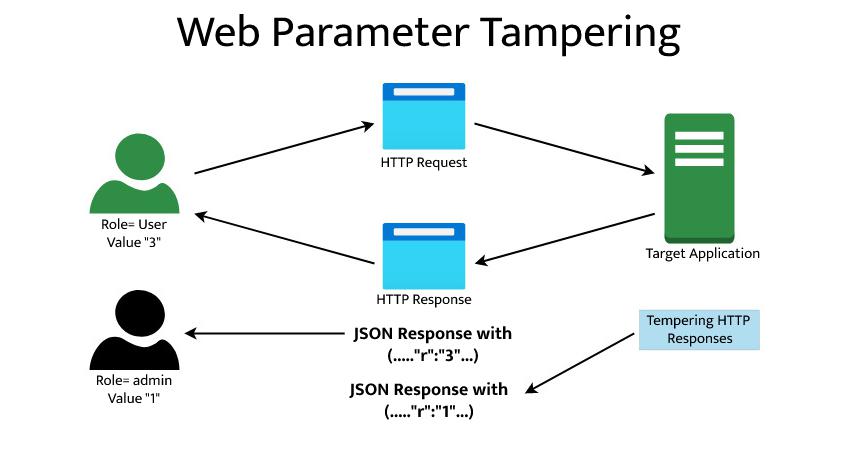
The injected data will be processed and then executed, which completes the attack.

It is performed by inserting unauthorized information into existing XML files or data streams. This is done by utilizing the special characters (metacharacters such as <, >. “ and &) contained within XML documents that can be used to add or edit data or XML syntax. Usage of these characters can allow an attacker to carry out desired operations in a target server.



1. **HTTP Verb Tampering:**

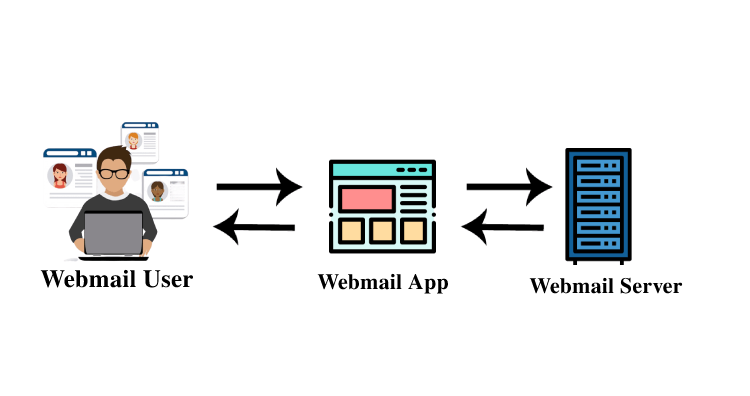
HTTP Verb Tampering is an attack that bypasses an authentication or control system that is based on the HTTP Verb. Sometimes, Web Server authentication mechanisms use verb-based authentication with access controls. Such security mechanisms include access control rules for requests with specific HTTP methods. Due to the HTTP specification that includes request methods other than the standard GET and POST requests, a standards-compliant web server may respond to these alternative methods in ways not anticipated by developers. So if an application restricts only GET requests it might still be possible to access the page using a POST, PUT, PATCH, or other method.



1. **Server Header Manipulation:**

Server Header Manipulation is a web server attack where attackers attempt to modify or conceal information found in the HTTP response header sent by a web server. This header often includes details about the server's technology stack and version. Attackers use this vulnerability to hide server information, making it harder for security researchers to identify potential vulnerabilities. It's also sometimes used to mimic trusted servers and evade security mechanisms.

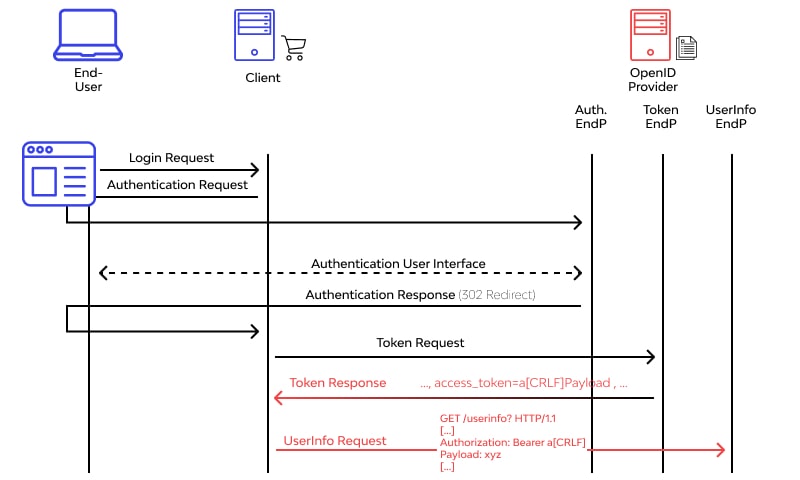
To defend against Server Header Manipulation, web administrators should configure servers securely, keep software updated, employ security headers, use web application firewalls (WAFs), monitor logs, and follow best security practices. However, it's important to note that obscuring server information should complement, not replace, comprehensive security measures like patch management and vulnerability assessments.



1. **HTTP Response Splitting:**

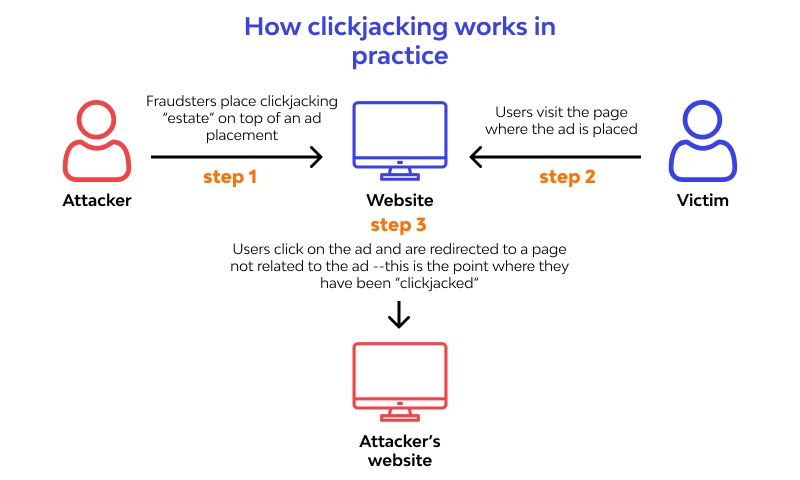
HTTP response splitting is a means to an end, not an end in itself. At its root, the attack is straightforward: an attacker passes malicious data to a vulnerable application, and the application includes the data in an HTTP response header.

To mount a successful exploit, the application must allow input that contains CR (carriage return, also given by %0d or \r) and LF (line feed, also given by %0a or \n) characters into the header AND the underlying platform must be vulnerable to the injection of such characters. These characters not only give attackers control of the remaining headers and body of the response the application intends to send, but also allow them to create additional responses entirely under their control.



1. **Clickjacking (UI Redress Attack):**

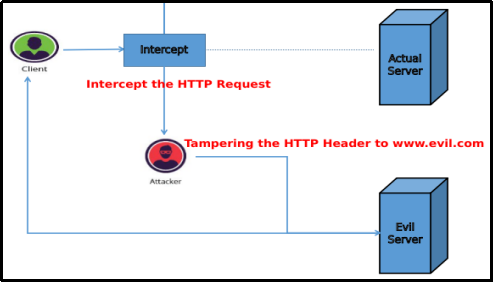
Clickjacking is an attack that fools users into thinking they are clicking on one thing when they are actually clicking on another. Its other name, user interface (UI) redressing, better describes what is going on. Users think they are using a web page’s normal UI, but in fact, there is a hidden UI in control; in other words, the UI has been redressed. When users click something, they think is safe, the hidden UI performs a different action.



1. **HTTP Host Header Attacks:**

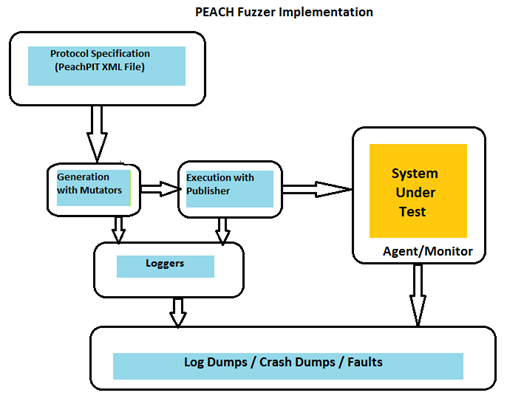
HTTP Host header attacks exploit vulnerable websites that handle the value of the Host header in an unsafe way. If the server implicitly trusts the Host header and fails to validate or escape it properly, an attacker may be able to use this input to inject harmful payloads that manipulate server-side behavior. Attacks that involve injecting a payload directly into the Host header are often known as "Host header injection" attacks.

Off-the-shelf web applications typically don't know what domain they are deployed on unless it is manually specified in a configuration file during setup.



1. **Fuzzing**

Fuzz testing works by initially inputting a large amount of random data (fuzz) into an application to get it to crash. The next step is using a fuzzer software tool to identify the weak spots. If there are any loopholes in the target's security, the attacker can further exploit it. The best way to combat a fuzzing attack is by keeping your security and other applications updated. This is especially true for any security patches that come out with an update that the perpetrators can exploit if you haven’t made the update yet.



1. **Man in the Middle Attack:**

Man-in-the-middle attacks are common among sites that haven’t encrypted their data as it travel from the user to the servers (sites using HTTP instead of HTTPS). The perpetrator intercepts the data as it’s being transferred between two parties. If the data isn’t encrypted, the attacker can easily read personal, login, or other sensitive details that travel between two locations on the Internet. A straightforward way to mitigate the man-in-the-middle attack is to install a Secure Sockets Layer (SSL) certificate on your site. This certificate encrypts all the information that travels between parties, so the attacker won’t easily make sense of it. Typically, most modern hosting providers already feature an SSL certificate with their hosting package.

