**Nikto**

1. HTTP Server Apache-Coyote/1.1︰是Tomcat的连接器。处理http请求的请求行，请求头等信息，创建Request和Response对象，然后调用Servlet容器的invoke方法
2. [X-Frame-Options](https://a42033.gitbooks.io/system/content/security/user/X_Frame_Options.html)：這是為了防止IFrame式Clickjacking攻擊
3. [**OSVDB**](https://en.wikipedia.org/wiki/Open_Source_Vulnerability_Database)**：是開源漏洞庫，**Open Source Vulnerability Database
4. [X-XSS](https://zh.wikipedia.org/zh-tw/%E8%B7%A8%E7%B6%B2%E7%AB%99%E6%8C%87%E4%BB%A4%E7%A2%BC)︰一種網站應用程式的安全漏洞攻擊，允許惡意使用者將程式碼注入到網頁上
5. [X-Content-Type-Options：](https://ithelp.ithome.com.tw/articles/10273029)在講 header ，先談談什麼content sniffing：一般來說瀏覽器會透過 Content-Type 來判斷請求到的資源是什麼類型，因此瀏覽器看到之後就會拿來執行，但有些網站（尤其是十幾二十年前的舊網站）在開發時並沒有把 Content-Type 設，瀏覽器除了參考 Content-Type 之外，也會做 content sniffing 從檔案內容分析是什麼類型，如果分析出是 JS 那就會拿去執行，這樣舊網站才不會壞掉
6. Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, OPTIONS
7. HTTP method ('Allow' Header): 'PUT' method could allow clients to save files on the web server.
8. HTTP method ('Allow' Header): 'DELETE' may allow clients to remove files on the web server.
9. [Web Server returns a valid response with junk HTTP methods, this may cause false positives.（](https://github.com/sullo/nikto/issues/556)廢棄HTTP方法

**Wapiti**

1. [盲目SQL注入](https://owasp.org/www-community/attacks/Blind_SQL_Injection)（與一般SQL注入 從數據庫檢所數據方式，當數據庫不向網頁輸出數據時，攻擊者需用真假問題竊取數據。困難
2. 脆弱的身分驗證（常見密碼、短密碼）解決執行strong password
3. CRLF的注入（將CRLF文字插入影響程式碼，並可結合其他攻擊技術影響程式碼ex：XSS 頁面注入 Web緩存中毒 緩存的破壞等。由\rASCII13\nASCII10組成
4. 內容安全策略配置：是附加的安全層，有助於檢測和緩解某些類型的攻擊
5. [跨站請求偽造CSRF](https://tech-blog.cymetrics.io/posts/jo/zerobased-cross-site-request-forgery/)：駭客以B網站的網域，用A網站給使用者的Cookie對A發送請求，之後可能加入些Javascript讓A支付費用給B網站
6. 潛在危險檔案：因有些服務器允許將任何文件類型上傳到服務器。
7. [Command execution](https://ithelp.ithome.com.tw/articles/10242080)命令執行注入：有點像SQL，但更危險，有可能取得伺服器控制權，影響運作。
8. 目錄遍歷（用網站的安全驗證缺陷或用戶請求驗證缺陷，列出伺服器目錄的漏洞，簡單來說就是利用url ../這符號抓到伺服器資料。
9. [Htaccess](https://beaglesecurity.com/blog/vulnerability/htaccess-bypass.html) 繞過（Htaccess 用密碼保護資價夾 允許或禁止用戶使用IP地址
10. [HTTP 安全標頭](https://www.searchenginejournal.com/http-security-headers/415404/)：4、16、HSTS、X-Content、X-Frame
11. HttpOnly 標誌 cookie（HttpOnly 是一個包含在 Set-Cookie HTTP 響應標頭中的附加標誌。在生成 cookie 時使用 HttpOnly 標誌有助於降低客戶端腳本訪問受保護 cookie 的風險)
12. 打開重定向（輸入內容後經過網址重新跳轉到其他惡意網址
13. 安全標誌 cookie（因http狀態是無狀態的 因此 cookie辨別客戶端身分，若被竊取或使用，可能造成數據洩漏。因此需要
14. SQL 注入
15. [服務器端請求偽造](https://ithelp.ithome.com.tw/articles/10242449)
16. 跨站腳本XSS：跨站腳本 (XSS) 是一種通常在 Web 應用程序中發現的計算機安全漏洞，它允許惡意 Web 用戶將代碼注入到其他用戶查看的網頁中。此類代碼的示例包括 HTML 代碼和客戶端腳本。
17. XML 外部實體：會發現在 XML 引用內部或外用部資源，由於 XML 解析器不會阻止及限制外部實體解析，如此可導致解析器遭受 XML 外部實體攻擊，用相關工具可以重送 Request。
18. 內部服務器錯誤：服務器端發生錯誤，服務器遇到了阻止它完成請求的意外情況。這可能是漏洞的跡象。
19. 資源消耗：無法正確控制有限資源的分配和維護，從而使參與者能夠影響所消耗的資源量，最終導致可用資源耗盡。
20. 指紋網絡技術：用於 Web 服務器指紋識別的技術包括[橫幅抓取](https://en.wikipedia.org/wiki/Banner_grabbing)、引發對格式錯誤的請求的響應，以及使用自動化工具執行更強大的掃描，這些掃描使用多種策略。他們都努力從 Web 服務器引出一些響應，然後可以將其與已知響應和行為的數據庫進行比較，從而與已知的服務器類型相匹配