網站攻擊技術，一篇打包帶走！

小白哥 黑客技術與網絡安全 今天

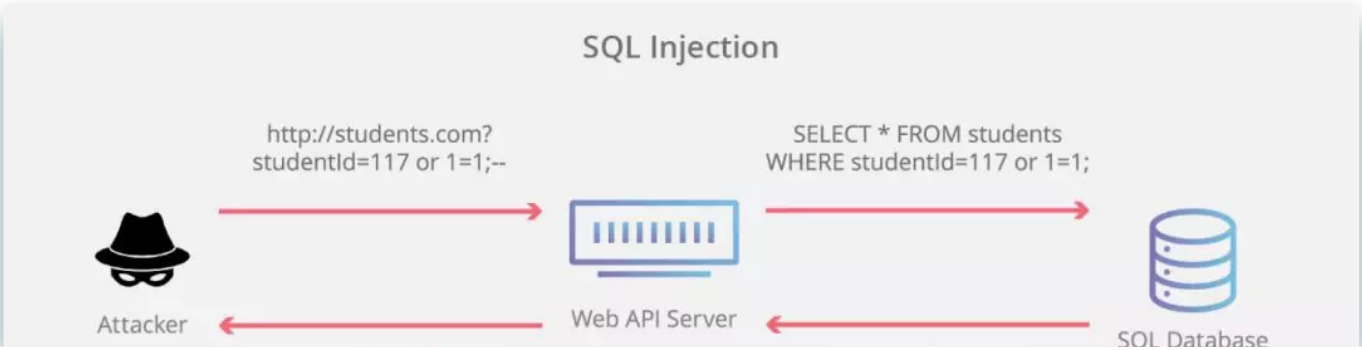
> **來自：SegmentFault，作者：senntyou**
> **鏈接：https://segmentfault.com/a/1190000018004657**

大家好，今天給大家介紹一下，Web安全領域常見的一些安全問題。

1、SQL 注入

SQL注入攻擊的核心在於讓Web服務器執行攻擊者期望的SQL語句，以便得到數據庫中的感興趣的數據或對數據庫進行讀取、修改、刪除、插入等操作，達到其邪惡的目的。

而如何讓Web服務器執行攻擊者的SQL語句呢？SQL注入的常規套路在於將SQL語句放置於Form表單或請求參數之中提交到後端服務器，後端服務器如果未做輸入安全校驗，直接將變量取出進行數據庫查詢，則極易中招。



舉例如下：

對於一個根據用戶ID獲取用戶信息的接口，後端的SQL語句一般是這樣：

```
select name,[...] from t_user whereid=$id
```

其中，$id就是前端提交的用戶id，而如果前端的請求是這樣：

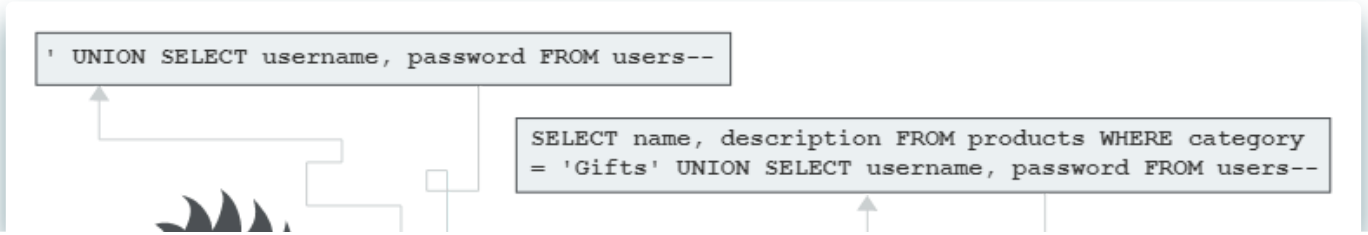
```
GET xx/userinfo?id=1%20or%201=1
```

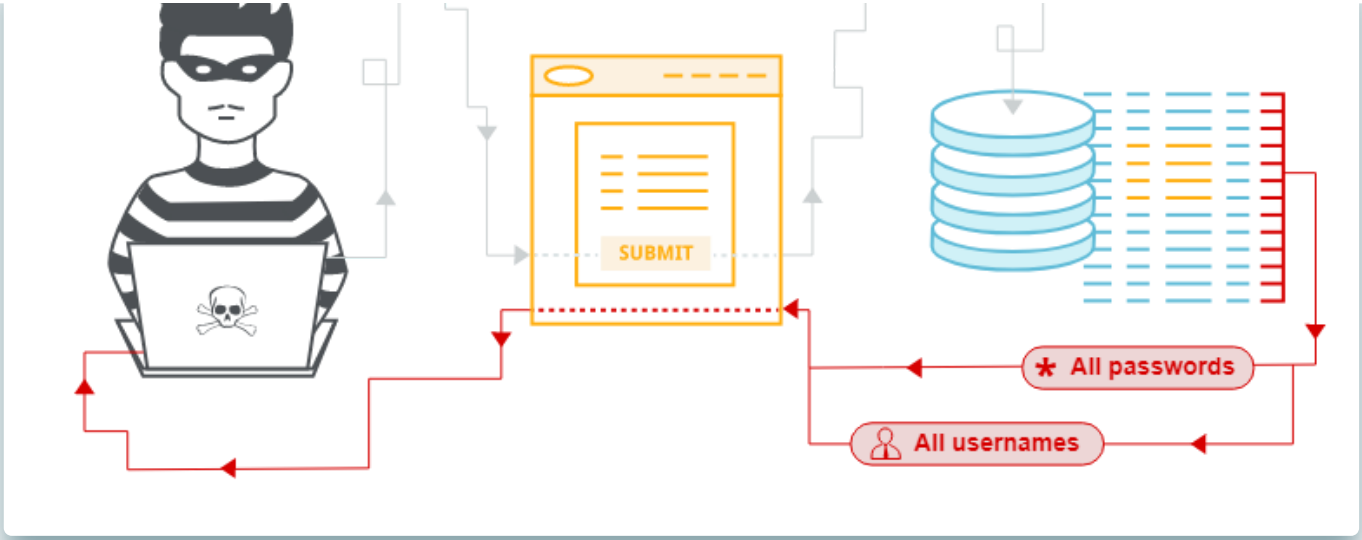
其中請求參數id轉義後就是1 or 1=1，如果後端不做安全過濾直接提交數據庫查詢，SQL語句就變成了：

```
select name,[...] from t_user whereid=1or1=1
```

其結果是把用戶表中的所有數據全部查出，達到了黑客洩露數據的目的。

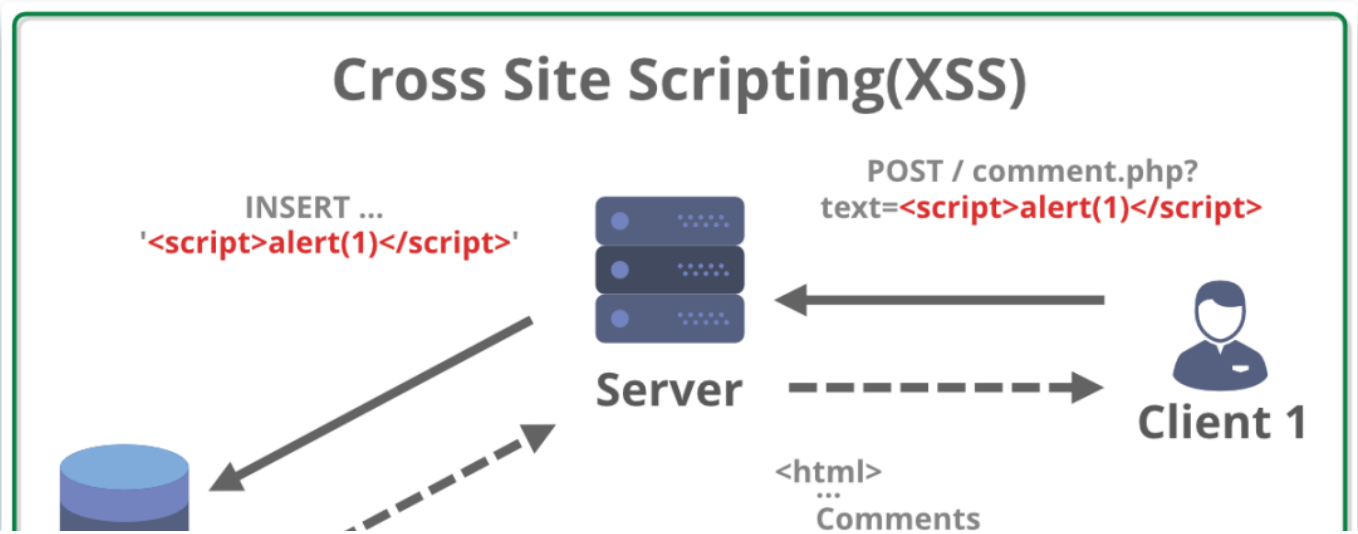
以上只是一個極簡單的示例，在真實的SQL注入攻擊中參數構造和SQL語句遠比這複雜得多，不過原理是一致的。

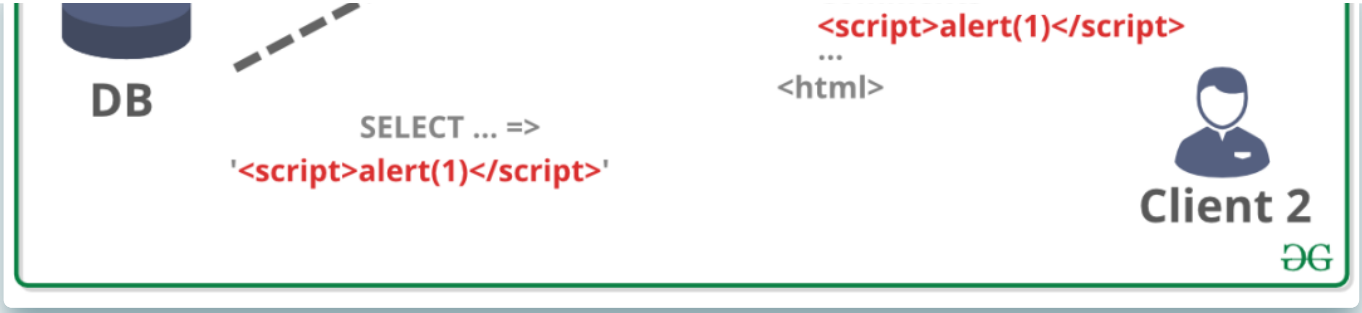




2、XSS 攻擊

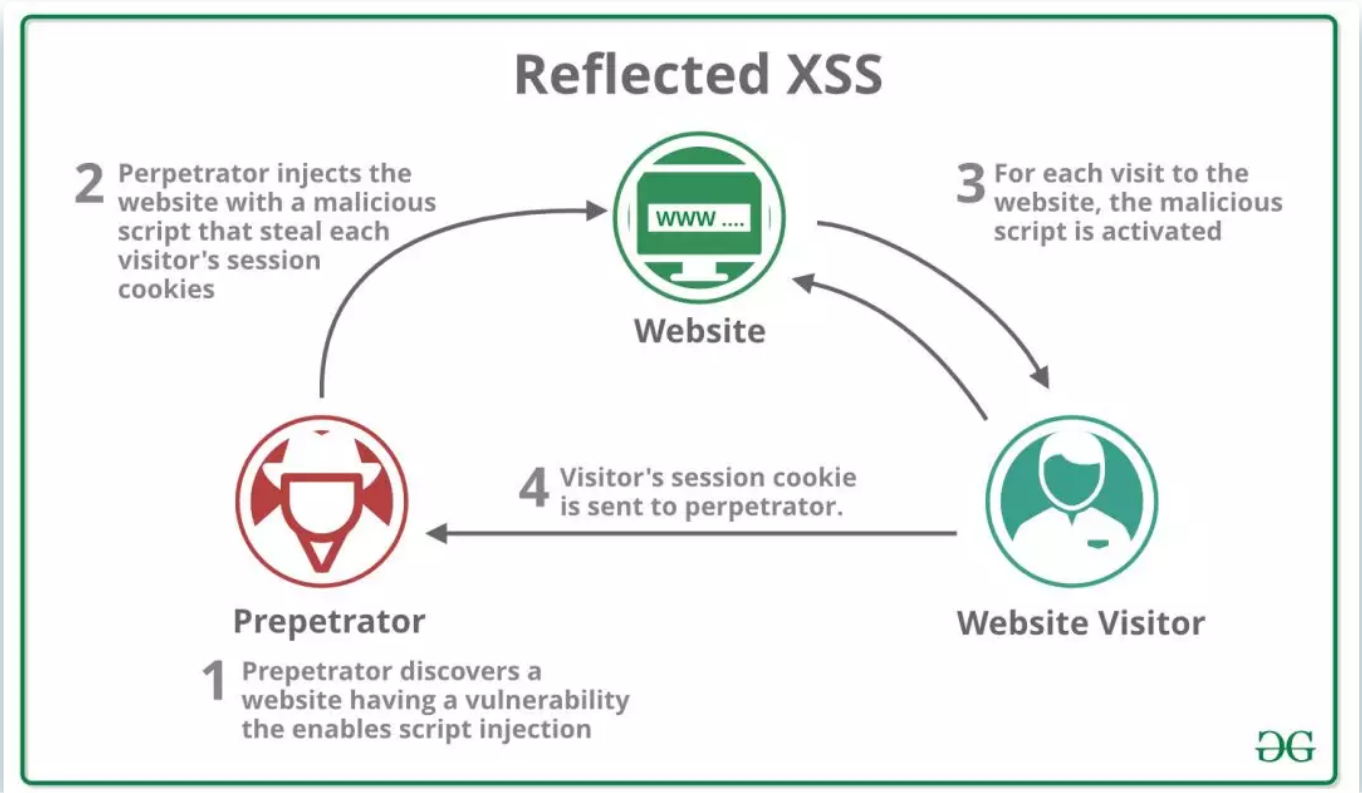
XSS全稱跨站腳本攻擊（Cross Site Scripting），為了與重疊樣式表CSS區分，換了另一個縮寫XSS。





XSS攻擊的核心是將可執行的前端腳本代碼（一般為JavaScript）植入到網頁中，聽起來比較拗口，用大白話說就是攻擊者想讓你的瀏覽器執行他寫的JS代碼。那如何辦到呢？一般XSS分為兩種：

反射型



1、攻擊者將JS代碼作為請求參數放置URL中，誘導用戶點擊 示例：

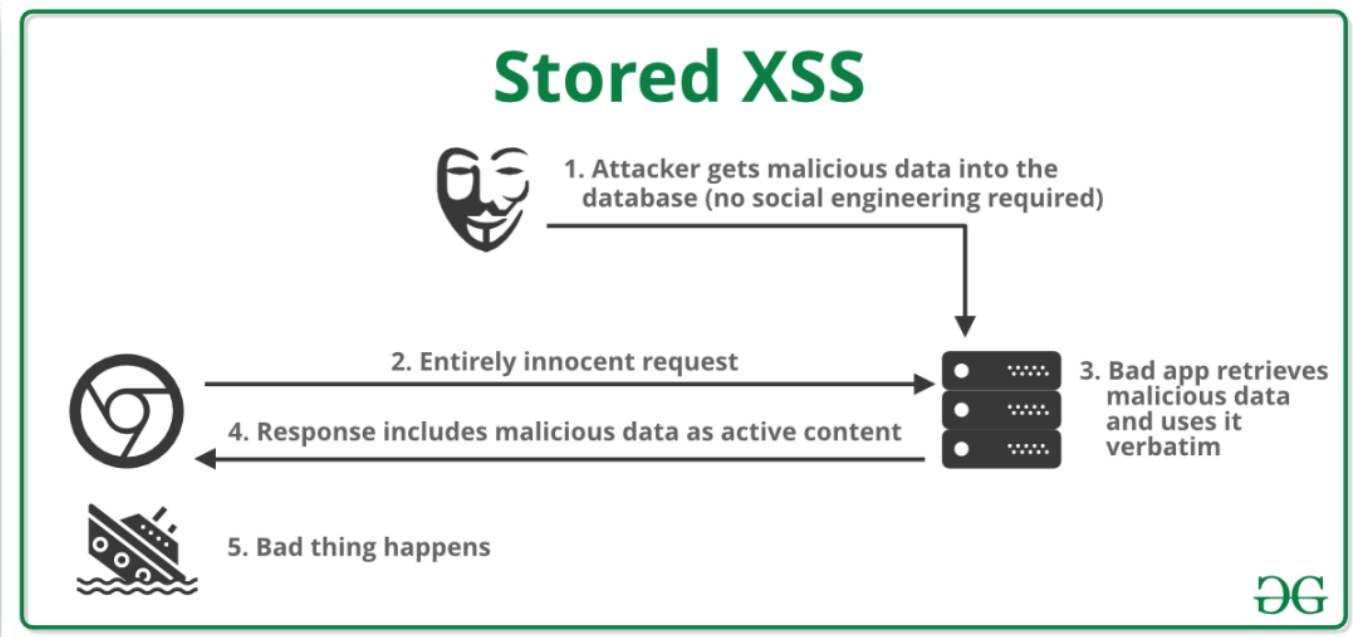
```
http://localhost:8080/test?name=<script>alert("you are under attack!")</script>
```

2、用戶點擊後，該JS作為請求參數傳給Web服務器後端

3、後端服務器沒有檢查過濾，簡單處理後放入網頁正文中返回給瀏覽器

4、瀏覽器解析返回的網頁，中招！

存儲型



上述方式攻擊腳本直接經服務器轉手後返回瀏覽器觸發執行，存儲型與之的區別在於能夠將攻擊腳本入庫存儲，在後面進行查詢時，再將攻擊腳本渲染進網頁，返回給瀏覽器觸發執行。常見的套路舉例如下：

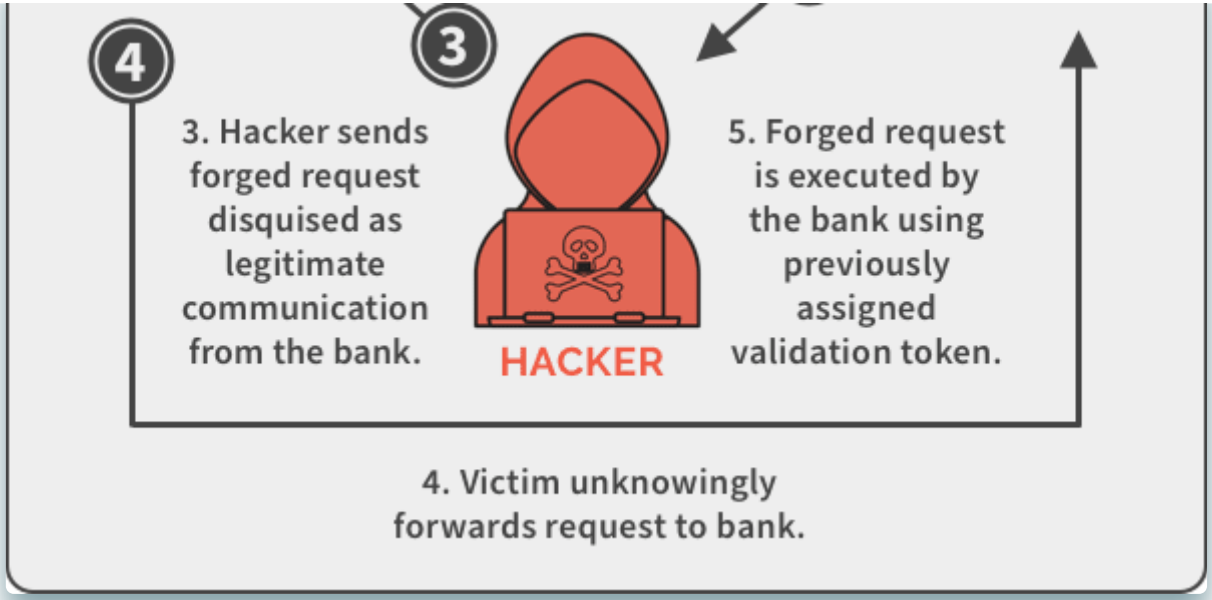
1、攻擊者網頁回帖，帖子中包含JS腳本

2、回帖提交服務器後，存儲至數據庫

3、其他網友查看帖子，後台查詢該帖子的回帖內容，構建完整網頁，返回瀏覽器

4、該網友瀏覽器渲染返回的網頁，中招！

3、CSRF 攻擊



CSRF，跨站請求偽造，其核心思想在於，在打開A網站的情況下，另開Tab頁面打開惡意網站B，此時在B頁面的“唆使”下，瀏覽器發起一個對網站A的HTTP請求。這個過程的危害在於2點：

1、這個HTTP請求不是用戶主動意圖，而是B“唆使的”，如果是一個危害較大的請求操作（發郵件？刪數據？等等）那就麻煩了

2、因為之前A網站已經打開了，瀏覽器存有A下發的Cookie或其他用於身份認證的信息，這一次被“唆使”的請求，將會自動帶上這些信息，A網站後端分不清楚這是否是用戶真實的意願

4、DDoS 攻擊

DDoS全稱Distributed Denial of Service：分佈式拒絕服務攻擊。是拒絕服務攻擊的升級版。拒絕攻擊服務顧名思義，讓服務不可用。常用於攻擊對外提供服務的服務器，像常見的：

- Web服務
- 郵件服務
- DNS服務
- 即時通訊服務
- ……

攻擊者不斷地提出服務請求，讓合法用戶的請求無法及時處理，這就是DoS 攻擊。

攻擊者使用多台計算機或者計算機集群進行DoS 攻擊，就是DDoS 攻擊。

在早期互聯網技術還沒有那麼發達的時候，發起DoS攻擊是一件很容易的事情：一台性能強勁的計算機，寫個程序多線程不斷向服務器進行請求，服務器應接不暇，最終無法處理正常的請求，對別的正常用戶來說，看上去網站貌似無法訪問，拒絕服務就是這麼個意思。

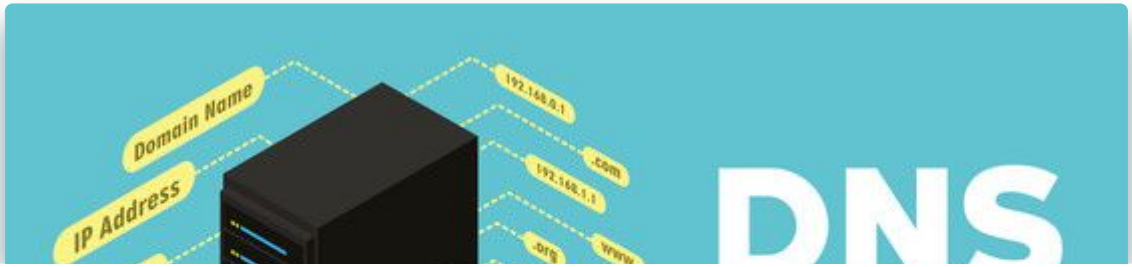
後來隨著技術對發展，現在的服務器早已不是一台服務器那麼簡單，你訪問一個www.baidu.com的域名，背後是數不清的CDN節點，數不清的Web服務器。

這種情況下，還想靠單台計算機去試圖讓一個網絡服務滿載，無異於雞蛋碰石頭，對方沒趴下，自己先趴下了。

技術從來都是一柄雙刃劍，分佈式技術既可以用來提供高可用的服務，也能夠被攻擊方用來進行大規模殺傷性攻擊。攻擊者不再局限於單台計算機的攻擊能力，轉而通過成規模的網絡集群發起拒絕服務攻擊。

5、DNS劫持

當今互聯網流量中，以HTTP / HTTPS為主的Web服務產生的流量佔據了絕大部分。Web服務發展的如火如荼，這背後離不開一個默默無聞的大功臣就是域名解析系統：

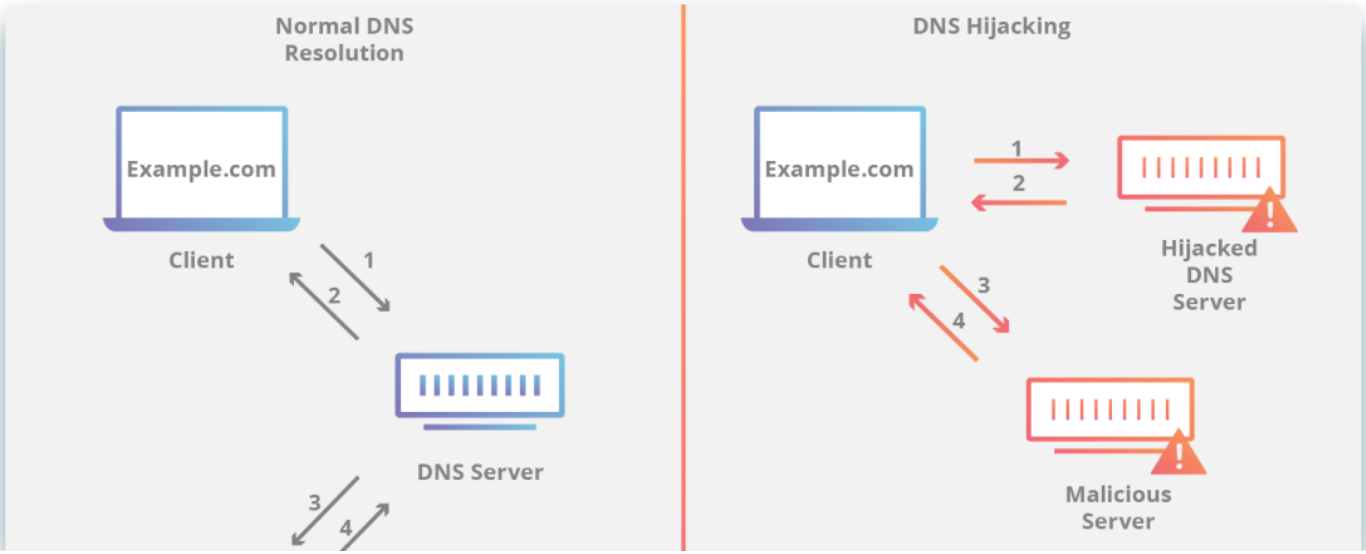


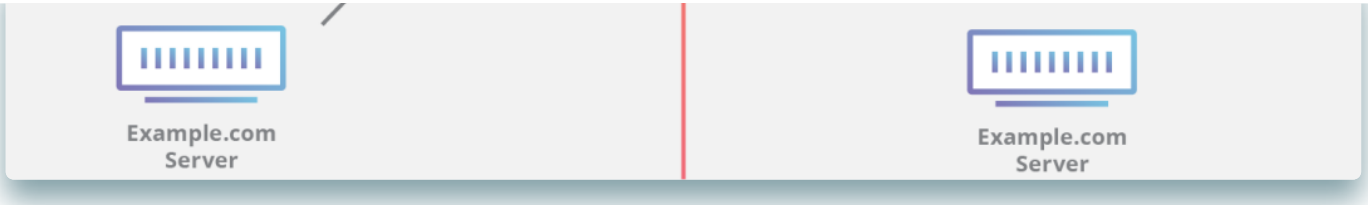
如果沒有DNS，我們上網需要記憶每個網站的IP地址而不是他們的域名，這簡直是災難，好在DNS默默在背後做了這一切，我們只需要記住一個域名，剩下的交給DNS來完成吧。

也正是因為其重要性，別有用心的人自然是不會放過它，DNS劫持技術被發明了出來。

DNS提供服務用來將域名轉換成IP地址，然而在早期協議的設計中並沒有太多考慮其安全性，對於查詢方來說：

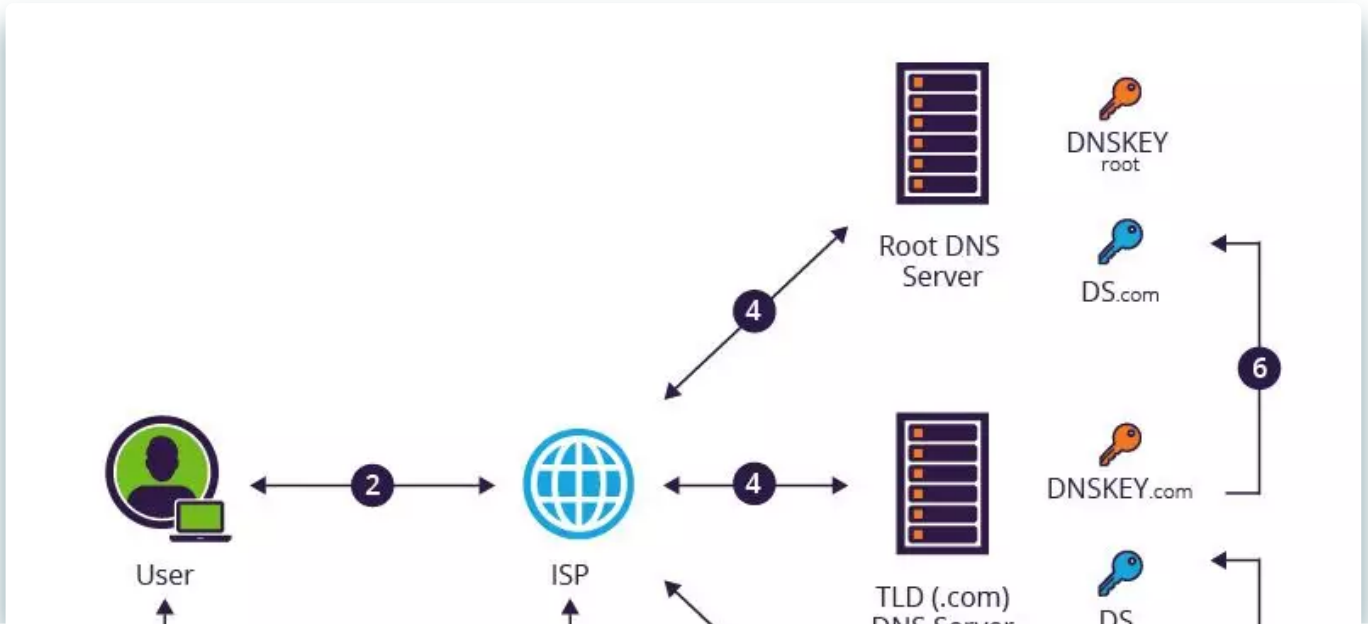
- 我去請求的真的是一個DNS服務器嗎？是不是別人冒充的？
- 查詢的結果有沒有被人篡改過？這個IP真是這個網站的嗎？

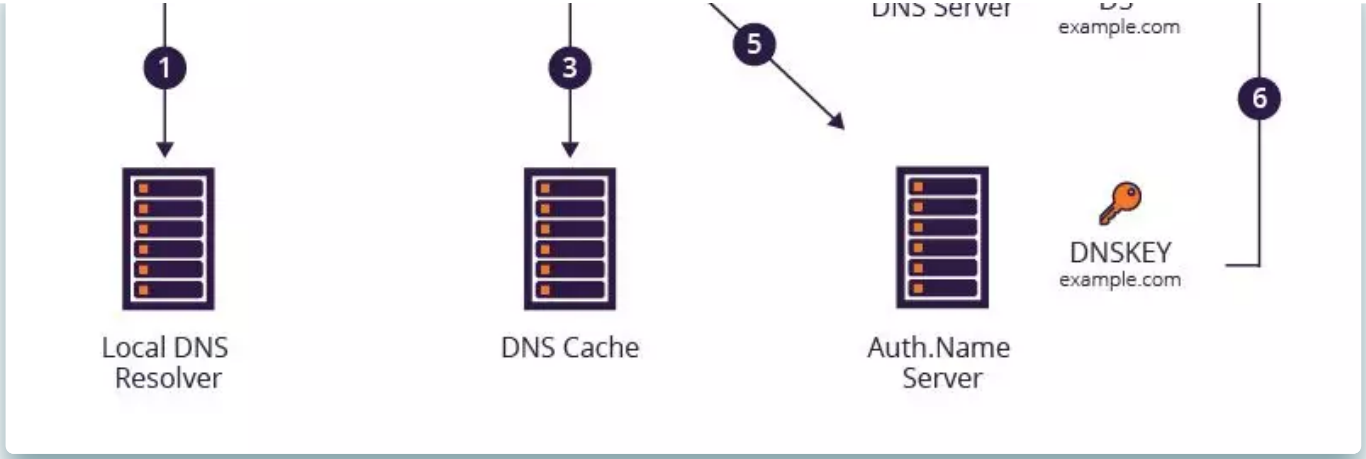




DNS協議中沒有機制去保證能回答這些問題，因此DNS劫持現象非常氾濫，從用戶在地址欄輸入一個域名的那一刻起，一路上的凶險防不勝防：

- 本地計算機中的木馬修改hosts文件
- 本地計算機中的木馬修改DNS數據包中的應答
- 網絡中的節點（如路由器）修改DNS數據包中的應答
- 網絡中的節點（如運營商）修改DNS數據包中的應答
- ……





後來，為了在客戶端對收到對DNS應答進行校驗，出現了DNSSEC技術，一定程度上可以解決上面的部分問題。但限於一些方面的原因，這項技術並沒有大規模用起來，尤其在國內，鮮有部署應用。

再後來，以阿里、騰訊等頭部互聯網廠商開始推出了httpDNS服務，來了一招釜底抽薪，雖然這項技術的名字中還有DNS三個字母，但實現上和原來但DNS已經是天差地別，通過這項技術讓DNS變成了在http協議之上的一個應用服務。

6、JSON 劫持

JSON是一種輕量級的數據交換格式，而劫持就是對數據進行竊取（或者應該稱為打劫、攔截比較合適。惡意攻擊者通過某些特定的手段，將本應該返回給用戶的JSON數據進行攔截，轉而將數據發送回給惡意攻擊者，這就是JSON劫持的大概含義。一般來說進行劫持的JSON數據都是包含敏感信息或者有價值的數據。

7、暴力破解

這個一般針對密碼而言，弱密碼（Weak Password）很容易被別人（對你很了解的人等）猜到或被破解工具暴力破解。

解決方案密碼複雜度要足夠大，也要足夠隱蔽限制嘗試次數

8. HTTP 報頭追踪漏洞

HTTP/1.1（RFC2616）規範定義了HTTP TRACE 方法，主要是用於客戶端通過向Web 服務器提交TRACE 請求來進行測試或獲得診斷信息。

當Web 服務器啟用TRACE 時，提交的請求頭會在服務器響應的內容（Body）中完整的返回，其中HTTP 頭很可能包括Session Token、Cookies 或其它認證信息。攻擊者可以利用此漏洞來欺騙合法用戶並得到他們的私人信息。

解決方案:

禁用HTTP TRACE 方法。

9. 信息洩露

由於Web 服務器或應用程序沒有正確處理一些特殊請求，洩露Web 服務器的一些敏感信息，如用戶名、密碼、源代碼、服務器信息、配置信息等。

所以一般需注意：

應用程序報錯時，不對外產生調試信息過濾用戶提交的數據與特殊字符保證源代碼、服務器配置的安全

10、目錄遍歷漏洞

攻擊者向Web 服務器發送請求，通過在URL 中或在有特殊意義的目錄中附加../、或者附加../ 的一些變形（如..\ 或..// 甚至其編碼），導致攻擊者能夠訪問未授權的目錄，以及在Web 服務器的根目錄以外執行命令。

11、命令執行漏洞

命令執行漏洞是通過URL 發起請求，在Web 服務器端執行未授權的命令，獲取系統信息、篡改系統配置、控制整個系統、使系統癱瘓等。

12. 文件上傳漏洞

如果對文件上傳路徑變量過濾不嚴，並且對用戶上傳的文件後綴以及文件類型限制不嚴，攻擊者可通過Web 訪問的目錄上傳任意文件，包括網站後門文件（webshell），進而遠程控製網站服務器。

所以一般需注意：

在開發網站及應用程序過程中，需嚴格限制和校驗上傳的文件，禁止上傳惡意代碼的文件限制相關目錄的執行權限，防範webshell 攻擊

13. 其他漏洞

- SSLStrip 攻擊
- OpenSSL Heartbleed 安全漏洞
- CCS 注入漏洞
- 證書有效性驗證漏洞

14. 業務漏洞

一般業務漏洞是跟具體的應用程序相關，比如參數篡改（連續編號ID / 訂單、1 元支付）、重放攻擊（偽裝支付）、權限控制（越權操作）等。

另外可以參考：6種常見web漏洞坑

15. 框架或應用漏洞

- WordPress 4.7 / 4.7.1：REST API 內容注入漏洞
- Drupal Module RESTWS 7.x：Remote PHP Code Execution
- SugarCRM 6.5.23：REST PHP Object Injection Exploit
- Apache Struts：REST Plugin With Dynamic Method Invocation Remote Code Execution
- Oracle GlassFish Server：REST CSRF
- QQ Browser 9.6：API 權限控制問題導致洩露隱私模式
- Hacking Docker：Registry API 未授權訪問

--- EOF ---

推薦↓↓↓

Linux學習

專注分享Linux/Unix相關內容，包括Linux命令、Linux內核、Linux系統開發、Linux運維、網絡編程、開發工具等Linux相關知識和技術

公眾號

閱讀原文

喜歡此內容的人還喜歡

水澤-信息收集自動化工具

滲透雲筆記

瀟湘信安技術交流微信①群

瀟湘信安

滲透測試中常見的那些編碼和加密

瀟湘信安