網站滲透技術介紹

報告日期:2021/09/02(四)

報告人:李威辰、吳冠廷

目錄

XSS介紹

XSS

XSS-全名Cross-Site Scripting(跨網站指令碼)

XSS是一種針對網站程式安全漏洞的攻擊，惡意使用者透過注入惡意代碼，使其他使用者在瀏覽網頁時受到影響，攻擊成功後，攻擊者可能獲得更高的權限，如：冒用管理員或使用者身分、竊取cookie等資料

原理

讓輸入的資料變成程式的一部份

輸入Hello的Html為 Hello

當輸入的內容為javascript程式<script>alert(“ XSS attack”);</script> 時

Html變成 <script>alert(“ XSS attack”);</script>

讓使用者彈窗出 XSS attack

常見的XSS攻擊手法

1. Stored XSS ，儲存型
2. Reflected XSS，反射型
3. DOM-Base XSS，文檔物件模型

Stored XSS-儲存型XSS

指被保存在伺服器資料庫的惡意代碼攻擊，由於存於database中每個使用者打開都會看到，是XSS中殺傷力最大的

EX:留言板，因為使用者可以留任意內容，若沒有確實檢查內容，如<script>等程式就會被當成正常的程式碼執行

Reflected XSS-反射型XSS

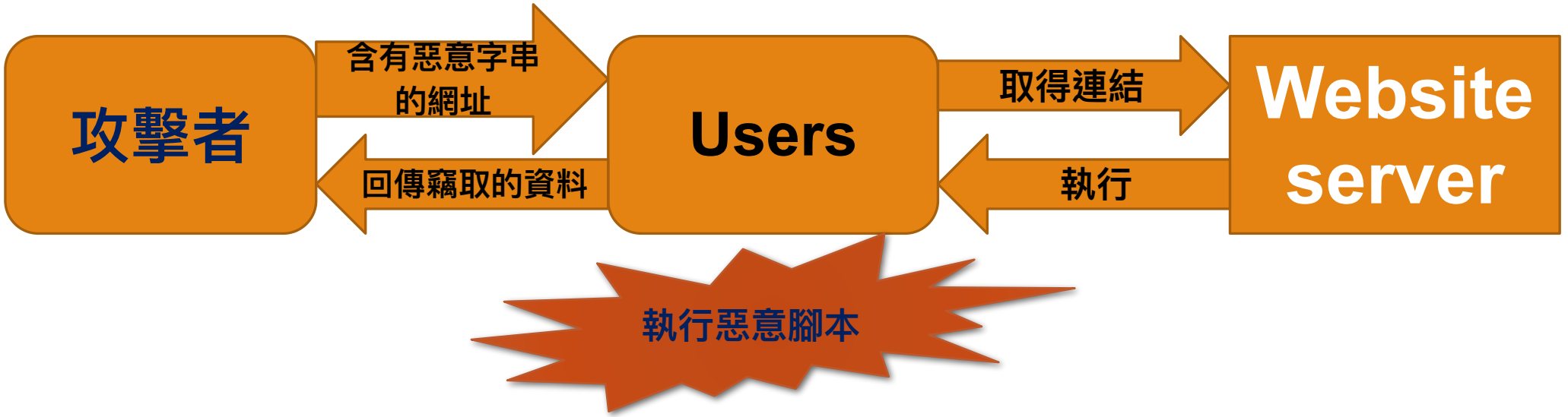
一般手法為透過如email等方式釣魚，誘使使用者點擊有惡意的連結，使用者會將惡意代碼取出拼接在html中並執行。

通常出現在網站的搜尋欄、登入介面來竊取cookie或session資料，冒充使用者在網站上操作

DOM XSS

DOM-document object model(文檔物件模型)

用以描述HTML的表示法，可以不透過伺服器使用javascript動態產生完整網頁(JS前端本身安全漏洞，而非前兩者後端服務器漏洞)



XSS的影響

1. 使加密連線失效，竊取users資料
2. 假冒users身分，存取資料控管限制的網站並操作
3. 將瀏覽器連結重新導向釣魚網站騙取個資
4. 將連結導向惡意網站，並於使用者電腦內植入後門程式
5. 使瀏覽器無法正常運作

預防與檢測XSS攻擊

1. 檢查安全性：

   靜態-檢查網頁原始碼

   動態-編譯並實際測試執行各種配置與變數產生的安全問題

2. 測試及維護：

   靜態-外包團隊或使用fortify等原始碼掃描工具

   動態-

   1.錯誤植入測試：利用fault injection軟體的檢測技術輸入未預期或錯誤的資料，來進行安全評估並找出安全漏洞

   2.滲透測試：滲透測試的測試員除了必須知道如何入侵及成功的原因，更需要知道要探測什麼、使用的工具及方式還有攻擊的時間點。

預防與檢測XSS攻擊-滲透測試

1. 黑箱測試：
   - 測試前不提供任何資料，測試者只知道公開的資料，如公司名稱IP address等。
   - 模擬一個如同真實世界的環境，測試者也如同真實的駭客，測試並利用所有的安全弱點。
2. 白箱測試：
   - 模擬測試者為已知許多內部的資訊的人，如離職員工或惡意員工
   - 測試前提供重要的資訊，如:合法的使用者帳號，網路設備種類，伺服器資訊、作業系統、資料庫平台等等

預防XSS攻擊-客戶端

1. 禁用Javascript(會導致網頁變得難用)
2. 注意陌生連結或陌生人提供的網址
3. 提高使用者對資安的認知與自我保護意識

SQL injection介紹

SQL injection

SQL injection是發生在網頁與資料庫之間的安全漏洞
當輸入的字串含有SQL指令時會導致資料庫將其執行

原因

1. 在應用程式中使用字串聯結方式或聯合查詢方式組合SQL指令。
2. 在應用程式連結資料庫時使用權限過大的帳戶。
3. 在資料庫中開放了不必要但權力過大的功能。
4. 太過於信任使用者所輸入的資料，未限制輸入的特殊字元，以及未對使用者輸入的資料做潛在指令的檢查。

原理

1. SQL命令對於傳入的字串參數是用單引號字元所包起來。（但連續2個單引號字元，在SQL資料庫中，則視為字串中的一個單引號字元）
2. SQL命令中，可以夾帶註解（連續2個減號字元 -- 後的文字為註解，或「/*」與「*/」所包起來的文字為註解）
3. 如果在組合SQL的命令字串時，未針對單引號字元作跳脫處理的話，將導致該字元變數在填入命令字串時，被惡意竄改原本的SQL語法的作用。

常見的 SQL injection 攻擊手法

1. Authorization Bypass（略過權限檢查）
2. Injecting SQL Sub-Statements into SQL Queries（注入 SQL 子語法）
3. Exploiting Stored Procedures（利用預存程序）

Authorization Bypass (略過權限檢查)

登入介面的SQL指令為SELECT * FROM customers WHERE name =' -name- ' AND password = ' -password- ‘

在user中填入'OR 1=1 – –

讓SQL指令變成SELECT * FROM customers WHERE name =' 'OR 1=1 – –'

SELECT * FROM customers WHERE name ="OR 1=1 – –'

等於

SELECT * FROM customers WHERE name ='' OR true – –'

Injecting SQL Sub-Statements into SQL Queries（注入 SQL 子語法）

利用SQL語法去改變資料庫，將SQL語法加在網址後面

http://www.mydomain.com/products/products.asp?productid=123; DROP TABLE Products

http://www.mydomain.com/products/products.asp?productid=123 UNION SELECT Username, Password FROM USERS

Exploiting Stored Procedures (利用預存程序)

將常用的 SQL 語法寫成一組程序並儲存起來，以供後續呼叫。

SomeAsp.asp?city=pune';EXEC master.dbo.xp_cmdshell' cmd.exe dir c:

透過 EXEC 去執行 master.dbo.xp_cmdshell 這個預存程序，並帶一參數 cmd.exe dir c: 代表想讓預存程序執行的內容。

影響

1. 資料表中的資料外洩。
2. 資料庫伺服器被攻擊，系統管理員帳戶被竄改（例如ALTER LOGIN sa WITH PASSWORD='xxxxxx'）。
3. 取得系統較高權限後，有可能得以在網頁加入惡意連結、惡意代碼以及Phishing等。
4. 攻擊者利用資料庫提供的各種功能操縱檔案系統，寫入Webshell，最終導致攻擊者攻陷系統。
5. 網站首頁被竄改。

預防

1. 在設計應用程式時，完全使用參數化查詢（Parameterized Query）來設計資料存取功能。
2. 在組合SQL字串時，先針對所傳入的參數加入其他字元（將單引號字元前加上跳脫字元）。
3. 使用php開發，可寫入html特殊函式，可正確阻擋XSS攻擊。
4. 資料庫設定使用者帳號權限，限制某些管道使用者無法作資料庫存取。

XSS Reference

身為 Web 工程師，你一定要知道的幾個 Web 資訊安全議題-https://medium.com/starbugs/%E8%BA%AB%E7%82%BA-web-%E5%B7%A5%E7%A8%8B%E5%B8%AB-%E4%BD%A0%E4%B8%80%E5%AE%9A%E8%A6%81%E7%9F%A5%E9%81%93%E7%9A%84%E5%B9%BE%E5%80%8B-web-%E8%B3%87%E8%A8%8A%E5%AE%89%E5%85%A8%E8%AD%B0%E9%A1%8C-29b8a4af6e13

淺談XSS-https://net.nthu.edu.tw/netsys/_media/web_site_security.pdf

XSS:跨網站指令碼-https://hitcon.org/2015/CMT/download/day1-a-r4.pdf

前端安全系列（一）：如何防止XSS攻擊https://kknews.cc/zh-tw/tech/grl4lj8.html

跨網站指令碼-https://zh.wikipedia.org/wiki/%E8%B7%A8%E7%B6%B2%E7%AB%99%E6%8C%87%E4%BB%A4%E7%A2%BC

基於跨網站攻擊而造成資訊洩漏的伺服器端防禦系統-https://hdl.handle.net/11296/uxg77d

SQL injection Reference

SQL注入-https://zh.wikipedia.org/wiki/SQL%E6%B3%A8%E5%85%A5

[Postx1] 攻擊行為－SQL 資料隱碼攻擊 SQL injection-https://ithelp.ithome.com.tw/articles/10189201

Mohd Yunus :"Review of SQL Injection : Problems and Prevention",in JOIV:INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION ,vol.2 NO.3–2(2018) ,p215-219 http://joiv.org/index.php/joiv/article/view/144