**HW2 Writeup**

**R10922043 黃政瑋**

**CTF account: cwhuang1937**

1. **Imgura**
   1. 檢查網頁

一開始先用F12仔細檢查網頁，發現什麼都沒有，接著在**DotGit**發現有.git洩漏的資料，接著用**scrabble**這個tool來還原source code。

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

* 1. 查看git log

接著用git log發現最新的commit上面有將dev page刪除，因此用git reset來還原到前次的commit，可以發現多了一個**dev\_test\_page**的資料夾。

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

* 1. 查看<https://imgura.chal.h4ck3r.quest/dev_test_page/>

可以發現這個網頁有上傳照片的功能，接著再看index.php，發現這邊有include的程式碼，故可以試著從**LFI**下手。

* 1. 查看upload.php

首先先上網下載一張小於512x512的.png， 接著看程式碼會發現檔案的第一個extension會被檢查是否符合要求，因此這邊試著將**extension改成.png.php上傳**，仍是可以上傳成功。

* 1. 製作含有木馬的照片

這邊用**pixload中 png.pl**來製作出我們所需的.png檔，並塞入一句話木馬**(這邊用<?=代替<?php echo來躲過偵測)**，接著上傳到網站。

一張含有 文字 的圖片

自動產生的描述

一張含有 文字, 電子用品, 鍵盤 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

* 1. 用command去拿flag

接著將上面上傳成功的網址，放到首頁的?page=後面，並將.php拿掉即可執行shellcode，回到根目錄後即可print出flag。

一張含有 文字, 裝置, 螢幕擷取畫面 的圖片

自動產生的描述

Reference:

Pixload用法: <https://github.com/chinarulezzz/pixload>

1. **DVD Screensaver**
   1. 分析題目

根據source code，可以發現真正的flag藏在db中某個user的底下，但在/login這個底下，只允許輸入字母與數字，故**判斷無法從這個path做SQL injection**。接著發現在/並沒有對username擋任何符號，故這邊可以利用cookies當作跳板，跳過登入的部分，並將我們要的injection塞入到cookies中的username，使得進入/這個path時即會拿我們塞過資料的username來做query。

* 1. 找出SECRET\_KEY

要來自製cookies前，需要先拿到這題的SECRET\_KEY，才能簽我們之後要繞過登入的cookies。仔細觀察source code會發現在/static這邊有讀檔的動作，因此從這個path開始下手。但試了幾次/static/..相關的path會發現，每次傳進去給Handler都會先被轉換過，因此**/static/..並無法走我們要讀檔的path**，接著參考題目給的doc，可以發現用**CONNECT method**即可成功照我們要的進入/static這個path，然後用curl到**/proc/self/environ**即可拿到環境變數。(這邊須注意加上**—path-as-is**，否則curl一樣會將我們傳入的url做轉換，而無法進入/static這個path。



一張含有 文字, 螢幕擷取畫面, 螢幕, 銀色 的圖片

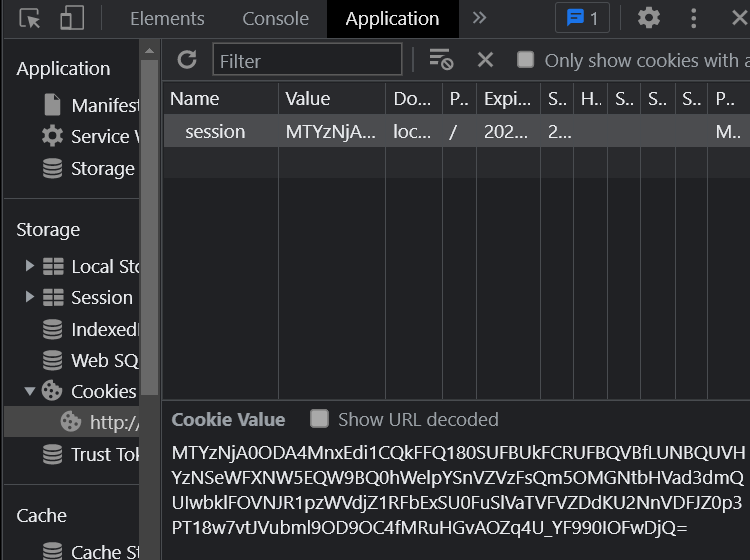
自動產生的描述

* 1. 自製cookies後做SQL injection

這時回到source code給的app.go，首先將上個步驟拿到的SECRET\_KEY放進去，接著將/login那邊的判斷式與query的動作都註解掉，並在 local這邊執行起來，此時這邊不管輸入什麼都能成功登入並產生出一組cookies。接著開始嘗試各種SQL injection的登入帳號，**拿登入後產生出的cookies去curl**，最終即可找到flag。

(SQL injection: 這邊用**LIKE**這個模糊搜尋的指令來找到flag。)

一張含有 文字 的圖片

自動產生的描述 

一張含有 文字 的圖片

自動產生的描述

1. **Double SSTI**
   1. 分析題目

首先用F12找到/source可以看source code，發現這題需先拿到secret才能發request給對應的path，並由server用proxy幫我們連到target url。接著又發現這個網站是使用handlebars這個template engine，於是上他的官網查語法。

* 1. SSTI拿secret

在handlebars的官網可以用@data variables來拿到特定的值，並搭配build-in helper的**#each**可以將**@root**底下的東西拿出來。(@root即template的initial context)。Injection後即成功拿到key，接著將url導向/2nd\_stage\_77777me0w\_me0w\_s3cr3t77777，proxy即會幫我們導向第二關。

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

* 1. SSTI分析:

首先用 {{"7"\*7}}嘗試發現得到的是7777777而不是49，可以確定這邊template engine為**Jinja2**而不是Twig。由於這邊沒有找到source code，故不斷嘗試輸入來確定可以injection的symbol有什麼。發現config、.(dot)、\_、[]皆會被blacklist擋掉，於是上網搜尋Jinja2 SSTI bypass相關的文章，發現只要用()，’’| 這三個符號即可做到SSTI。

* 1. 變換symbol來bypass

參考網路上的文章後，這邊將需要用到的symbol轉換皆列出來。

. (dot): 可以用**|attr()**來替換。

\_\_ : **\x5f\x5f**來encode即可。

[]: 用**.\_\_getitem\_\_**來做到，而這個function再用上述的替換在變換一次，**.\_\_getitem\_\_ = >|attr('\x5f\x5fgetitem\x5f\x5f')()**。

將以上的三種符號變換之後，即可成功bypass，並SSTI拿到flag。

Final payload:

**{{ ()|attr('\x5f\x5fclass\x5f\x5f')|attr('\x5f\x5fbase\x5f\x5f')|attr('\x5f\x5fsubclasses\x5f\x5f')()|attr('\x5f\x5fgetitem\x5f\x5f')(132)|attr('\x5f\x5finit\x5f\x5f')|attr('\x5f\x5fglobals\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('popen')('cat /y000\*')|attr('read')() }}**

一張含有 文字 的圖片

自動產生的描述

Reference:

1. Handlebars官網: <https://handlebarsjs.com/>
2. Jinja2 SSTI bypass參考文章: <https://medium.com/@nyomanpradipta120/jinja2-ssti-filter-bypasses-a8d3eb7b000f>
3. **Log me in FINAL**
   1. 分析題目

一開始試著查看F12與登入後，發現什麼東西都沒有，也沒有cookies被記錄。從助教的hint似乎要不斷地戳出500才能看到東西，因此這邊不斷嘗試輸入各種特殊符號當作input，最終用**反斜線**終於戳出500，不過這邊只能看到query的語法，但是sqli\_waf()與addslashes()做了什麼並無法知道。於是這邊再試著用curl只丟username的欄位進去，會發現在sqli\_waf會噴error，但無法看到trackback的code，因此這邊用**BurpSuite**來丟一個不完整的payload。即成功看到那兩個function的source code頁面。

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

* 1. 分析sqli\_waf與addslashes:

sqli\_waf:

keyword :中間**加上空白**即可繞過。

空白 : 則用**/\*\*/繞**過。

= : 用**LIKE**即可替代

addslashes:

由於’與”都會被自動加上反斜線，因此這邊用**\’**繞過，使得\被跳脫出來成為data即可。而後續若需要用到’’時，則用**CHAR()**繞過即可。

這邊以\'||true#實驗，是可成功登入的，確定可以injection後才往後繼續測試。

* 1. 測試user table的欄位

\'/\*\*/UNI ON/\*\*/SEL ECT/\*\*/1,2,3/\*\*/FROM/\*\*/users#

從1,2,3…不斷試，發現只有3不會噴error(error based)，故users有三個column，而第三個column**手動用二分法**不斷地猜，猜出來是uid。接著再搭配LIKE的模糊搜尋，來找看看當中有沒有什麼重要內容，於是在password找到了一個提示FLAG(is\_in\_another\_table)，這邊都是用手動不斷的試，故這邊花了非常非常多的時間，於是決定後面的猜長度與猜字串皆自己寫**python script**來跑。

* 1. 暴力跑table name:

這邊用LIMIT手動測試出一共有2個table，接著用自己寫的腳本先跑出table name的長度再跑出table\_name(h3y\_here\_15\_the\_flag\_y0u\_w4nt,meow,flag)。這邊再參考助教的github，可以用error bases的方式爆出column name。

一張含有 文字, 監視器, 螢幕擷取畫面, 黑色 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

* 1. 暴力跑出FLAG:

拿到table name與column name後，用前面的腳本改一下payload再搭配LIKE的模糊搜尋( %FLAG{% => CHAR(37,70,76,65,71,123,37))，即可成功暴力跑出FLAG。

一張含有 文字 的圖片

自動產生的描述

Reference:

助教大神的github: <https://github.com/w181496/Web-CTF-Cheatsheet#sql-injection>