國際 CTF 題目介紹與背景探討

鄭達群
國立交通大學 資訊科學與工程研究所
chengtc@cs.nctu.edu.tw

摘要

CTF，全稱為 Capture the flag，名稱源自於模擬戰爭遊戲中的搶旗模式；在駭客界來說，則是一種模擬入侵與防禦的遊戲，參賽隊伍必須以入侵、分析封包、破解密碼等各種手段獲得 flag，並提交至公正第三方以獲得分數，CTF 類型又可以分為兩種，分別是 attack & defense 及 jeopardy，兩者的題目類型和規則略有不同，但皆以獲得 flag 為主要目的；透過 CTF 可以用遊戲的方式學習資訊安全相關知識，避免真實攻擊別人時造成危害及觸犯法律，達到寓教於樂的目的。

關鍵詞：Capture the flag、入侵、漏洞、防禦、資訊安全、遊戲

壹、 前言

隨著科技的進步，人們的日常生活越來越離不開電腦網路，小則透過網路交談訊息，大則透過網際網路傳輸重要資料及文件，但是，倘若有心人士利用軟體漏洞或社交工程等方式，企圖竊取機密資料，無論是對個人、企業，甚至是國家單位，皆可能造成重大損失，因此資訊安全應該也必須被大家重視；

所謂知己知彼，百戰不殆，要防禦別人的攻擊，就必須了解攻擊的方式，在學習的過程中，不可能只仰賴看書學習的知識紙上談兵，一定得找機會實際演練，但是隨意攻擊別人不僅違法，如果操作不慎還可能造成損失或破壞，因此 CTF 應運而生，透過 CTF 可以想學習資訊安全的大眾，以一種遊戲的方式去學習入侵及防禦的技巧，避免觸犯法律或造成真正上的破壞。

CTF 早期稱之為 wargame，是在模擬戰爭遊戲中，奪下對方旗幟的模式[5]，在駭客界來說，則是一種模擬入侵與防禦的遊戲，在國外已經行之有年，近年來各國更是蓬勃發展，許多國家甚至以政府為首，傾國之力支持資訊安全的發展，由政府出資，提供經費令資安團體舉辦 CTF 供各路好手參戰，像是韓國的 secuinside CTF、大陸的百度盃，舉辦 CTF 一方面可以促進國內資安實力的發展，更吸引國外高手來分享經驗，台灣也不落人後，今年不僅進入 defcon 決賽，HITCON 也將每年議程必備的 wargame 正式更名為 CTF，更在 ctftime 平台上註冊事件，讓任何人都可以報名，讓大家可以與國外高手一較高下。

貳、 什麼是 CTF

CTF，全稱為 Capture the flag[1]，名稱源自於模擬戰爭遊戲中的搶旗模式；在駭客界來說，則是一種模擬入侵與防禦的遊戲，參賽隊伍必須以入侵、分析封包、破解密碼等各種手段獲得 flag，並提交至公正第三方以獲得分數，CTF 類型又可以分為兩種，分別是 Attack & Defense 及 Jeopardy，兩者的題目類型和規則略有不同，但皆以獲得 flag 為主要目的，目前國際上 CTF 大多還是以 jeopardy 為主，只有少部分決賽會採用 Attack & Defense 的模式。

Attack & Defense 的主要形式是「攻擊敵方，防禦己方」，詳細規則由主辦方制定，以最著名的 defcon 規則為例，每組參賽隊伍會被分配一台相同規格的伺服器，上面會運行相同的平台及數個特定的服務，參賽隊伍必須想辦法從運行的服務中，找到可行的攻擊手段，獲得該服務所代表的 flag 並上傳至主辦方，flag 正確則成功獲得分數，被攻擊的一方將失去相對應的分數；找到漏洞後還必須想辦法將漏洞修補，以避免別人利用漏洞來攻擊己方，倘若修補過程中造成服務無法正常運行，參賽隊伍則扣分，並將分數平分給其他所有參賽者。

Jeopardy 則是以解題的形式競賽，主辦方會準備各式各樣的題目，參賽隊伍必須理解題目內容，並想辦法解出題目中隱含的 flag，Jeopardy 的題目類型相較於 Attack & Defense 更為多樣化，除了入侵、找漏洞之外，還會涉及到逆向工程、資訊隱藏、密碼學、程式能力等各方面的技巧，考驗參賽隊伍是否具備全方位的能力。

以下是國際 CTF 中常見的題目分類：

1. Pwn：題目通常會給一組 host & port，或是讓參賽者直接 ssh 連線到題目環境，參賽者必須從題目程式中找出可以利用的漏洞，並獲取此題所對應的 flag，Attack & Defense 大部分皆是此類型的題目。
2. Web：題目會給參賽者一個網址，參賽者必須找到這個網站中設計不當的地方，利用各種方式，像是 SQL injection、偽造認證身分…等，獲得此題所對應的 flag。
3. Reverse：題目通常是一個執行檔，參賽者必須想辦法透過逆向工程的方式，理解執行此檔案後的行為，並達成某些條件讓檔案執行後會顯示此題對應的 flag。
4. Crypto：題目會提供密文和加密時的工具，參賽者必須找出加密時的缺陷，將密文破譯以獲取此題的 flag。
5. Recon：題目通常會問一個問題和一些線索，答案可能會藏在某些網站中，考驗參賽者的情報搜查能力。
6. PPC：題目可能是一個遊戲，參賽者要設計出一個演算法來突破層層關卡，過關後才將 flag 顯示給參賽者，考驗參賽者的程式能力。
7. Misc：其他難以分類的題目皆屬於此，可能包含分析封包（Forensic）、找出圖檔或音訊檔隱藏的訊息（Steganography）、資料救援…等，考驗參賽者全面的能力。

參、 實例分析－stand back, we have PHDays!

PHDays CTF 是由俄羅斯駭客團體 Positive Technologies 於 2014/1/25 所舉辦 CTF 競賽[2]，為期兩天，類型是 jeopardy，其中有一題價值 2700 分的題目——「stand back, we have PHDays!」，需要結合 Crypto 和 Web 兩方面的知識，才可以成功拿到 flag。

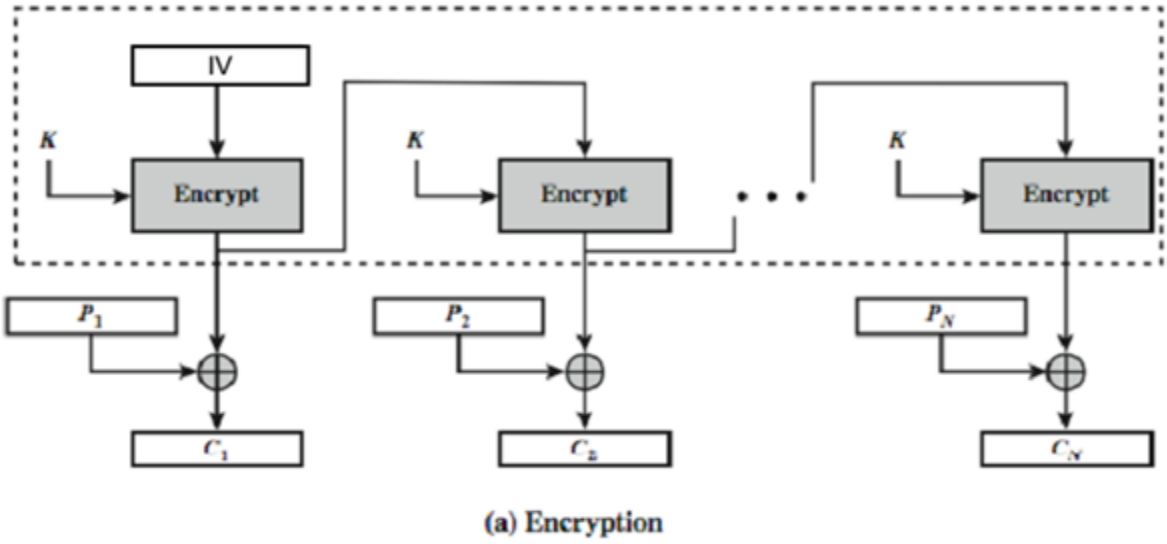
連到題目所給的網址，大部分都是靜態頁面，只有一個 login page 有點可疑，但嘗試做 SQL injection 也沒有成功，但是仔細檢查後會發現此網站竟然存在 index.php~（此檔案為編輯檔案時的備份），仔細檢查檔案內容，有一行是：

```
$query = "SELECT username FROM users WHERE id='$uid'";
```

程式並沒有對 uid 做過濾，因此如果能控制 uid 的內容就可以做 SQL injection 了，但是 uid 的來源卻是由 cipher 解密而來：

```
if(isset($_COOKIE['uid'])){
   $uid = openssl_decrypt ($_COOKIE['uid'], $method, $key, false, $iv);
}
```

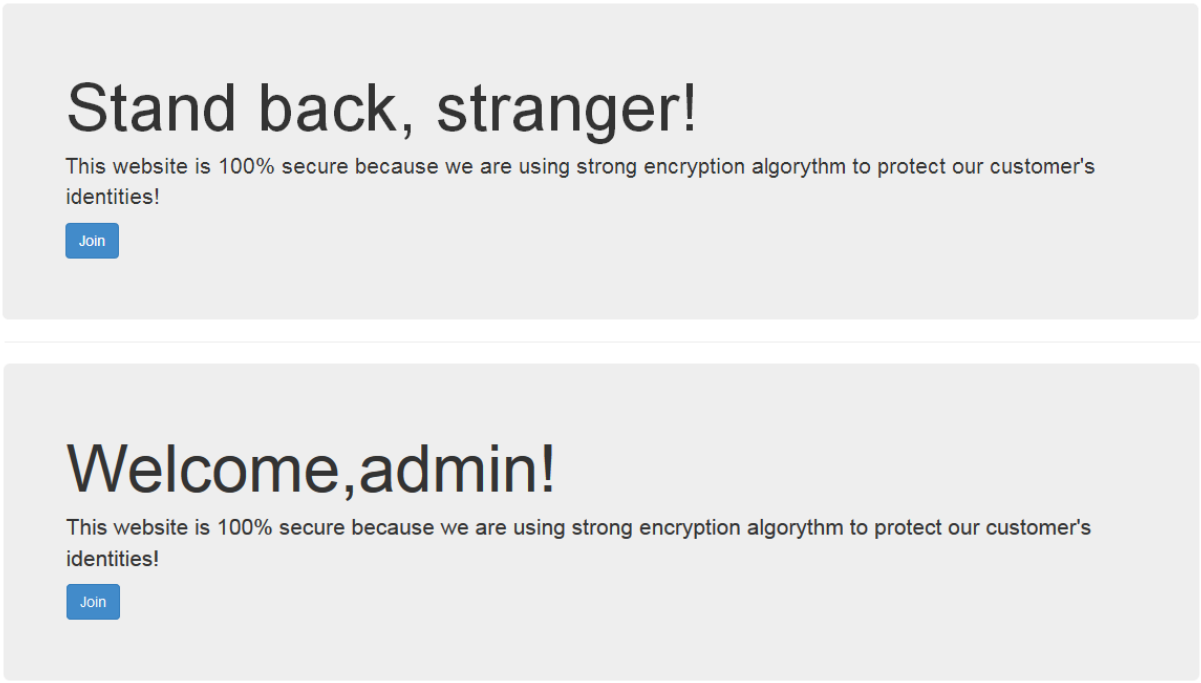
其中 $method 代表所使用的加密方式，為 AES-256-OFB，AES 加密是現今網路世界依然很常見的一種加密方式，至少在 CTF 時限內沒辦法用暴力破解的方式得到明文，但是仔細研究 OFB（Output feedback）這種 stream cipher 的流程會發現是有破綻的，如圖一所示，OFB 是利用 AES 產生出 xor key 以後，再以 xor key 做 xor 加密，如果 IV 和 K 的值重複使用，則產生出的 xor key 會相同，因此只要蒐集到夠多的 cipher，就可以透過頻率分析的方式去解出 xor key，我們就可以把 payload 加密成想要的密文，再做 SQL injection。



圖一：OFB encryption

將「' or 1=1 or "='」以 xor key 加密後再存到 cookie 內測試猜想是否正確，可以成功得到歡迎訊息變成「Welcome, admin!」，如圖二，但是也就僅此於此，網站內沒有其他後台藏有線索了，於是猜測 admin 的密碼就是此題的 flag，我們可以透過歡迎訊息是否變化來做 blind SQL injection，最後得到此題的 flag 為：

50M37IM35_Y0U_D0_N07_N33D_A_K3Y_70_D3CRYP7_AES



圖二：透過 welcome message 分辨 SQL 語法是否成功

肆、 實例分析－yet-another-javascript-jail

此題出自於由韓國駭客團體 ASRT（Advanced Security Research Team）所舉辦的 Secuinside CTF[3]，屬於典型的 Pwn 類型，需要找到此題的漏洞並撰寫 shellcode 來獲得此題所對應的 flag，此題漏洞由知名駭客 geohot 於 2014 年二月回報給 google，google 給予獎金 15 萬美元作為回報[4]。

如題目名稱所述，此題是一個可以任意執行 javascript 的 jail 環境，連到題目提供的 ip 和 port 以後，題目會先印出一串訊息，要我們繞過「check()」的限制，並將變數 flag 的值印出來，這部分只要透過 overwrite「Array.prototype.toString」重新定義 javascript 中 array 的行為即可。

但是題目印出的不是 flag，而是一個 url，內容如下：

Seems like easy? this is a pwnable task. binary:

http://54.178.138.53/29084554f8e41b34912a3aebddff81de (release date : Feb 2014)

從 url 中我們會下載到一個 elf 檔案，執行後發現此檔案就是題目的環境，也就是說，此題是由 google v8 engine 所建置而成的 javascript jail。

嘗試做 reverse 分析並對原始的 v8 engine 做比對，發現除了「runshell()」加入幾行程式碼，使 v8 engine 預設去載入 /home/jj/init.js 並執行其內容，除此以外沒有發現任何不同，因此猜測此題的漏洞並非出題者刻意設計，而是原始就存在於 v8 engine 的漏洞！再根據前一階段的提示訊息「release date : Feb 2014」，初步判定此題是考真正存在於 google chrome 的漏洞 CVE-2014-1705，根據 CVE 的說明，此漏洞可以任意改寫記憶體，poc 的關鍵程式碼如以下三行：

```
var ab4 = new ArrayBuffer(8);
ab4.__defineGetter__("byteLength", function() { return 0xFFFFFFFC; });
var aaaa = new Uint32Array(ab4)
```

將 array 的長度設超大以後，只要執行「aaaa[index] = 0xAAAAAAAA」這種形式的 javascript 代碼就可以任意讀寫記憶體位置，index 不支援負數，看似不能存取 aaaa 之前的記憶體位置，但是在執行組合語言時會變成像這樣的代碼「mov eax,[esi+index*4]」，可以透過 interger overflow 的特性，去存取在 aaaa 之前的記憶體

剩下的就是該怎麼利用這個漏洞，在讀取指令時是利用「fget()」去得到指令，因此我們將 fget 的 got table 所代表的位置改成 system 的位置，並將 esp 所指的位址內容改寫為為「/bin/sh」，即可拿到此題的 shell。

```
cat FLLLAGGGGG
66237850ab03afbc721cf358be3812b7
```

伍、 結論

現今 CTF 的題目其實包羅萬象，這邊列舉兩題只占了冰山一角，透過 CTF 不僅可學習到電腦安全的知識，還有助於程式能力的提升，解題的過程中有時也需要去學習一些環境部屬或電腦指令的操作，非常適合資訊相關科系的學生作為課外娛樂。

過去 CTF 的場次其實不多，可能一兩個月才會有一次參加的機會，但近年來資訊安全日益受到重視，CTF 的場次也越來越多，依照 ctftime 平台統計的場次，2011 年僅僅只有 18 場 CTF，2013 年的統計則高達 55 場 CTF，而 2014 年至目前為止(2014/07/30)，已經登記了 33 場 CTF，其中許多 CTF 都提供高額獎金吸引高手參賽。

台灣雖然目前風氣還不盛行，但隨著前陣子 Chroot 和台大 217 聯軍，在大陸舉辦的百度盃一鳴驚人，吸引了社會大眾的目光，今年更是 HITCON 聯軍更是成功打入 defcon 決賽，相信如果決賽也能有亮眼的表現，一定能引起更多人對資訊安全的興趣，讓台灣朝著成為資安強國的目標邁進[6]。

［致謝］

首先，我要感謝黃世昆老師收我進 SQLAB，由學長的介紹下開始接觸 CTF，這過程中認識到很多大神，也感謝 HITCON 的賞識，讓我有機會去國外參加 CTF 決賽，最後特別感謝向我邀稿的 Alan，也希望這篇文章能讓大家能對 CTF 有更深的認識。

參考文獻

[1] Capture the flag, http://en.wikipedia.org/wiki/Capture_the_flag. Retrieved 2014-07-30.

[2] Eindbazen, Hackers from All Over the World Competed to Join PHDays IV CTF, http://www.phdays.com/ctf/, 2014-0. Retrieved 2014-07-30.

[3] Schedule of SECUINSIDE 2014, http://www.secuinside.com/2014/notice.html?id=52, 2014-05-14. Retrieved 2014-07-30.

[4] The google chrome team, Stable Channel Update for Chrome OS, http://googlechromereleases.blogspot.tw/2014/03/stable-channel-update-for-chrome-os_14.html, 2014-03-14. Retrieved 2014-07-30.

[5] Chroot.org，*The Wargame 駭客訓練基地 - 決戰台灣版*，臺北：旗標出版股份有限公司，2008。

[6] Benson，我們可以成為資安強國——資安危機就是國安危機系列之三，http://www.appledaily.com.tw/appledaily/article/headline/20140724/35978177/%E6%88%91%E5%80%91%E5%8F%AF%E4%BB%A5%E6%88%90%E7%82%BA%E8%B3%87%E5%AE%89%E5%BC%B7%E5%9C%8B%EF%BC%88Benson%EF%BC%89 ，2014 年 07 月 24 日。於 2014 年 7 月 30 日查閱。

[作者簡介]

鄭達群，目前就讀於國立交通大學資科工所二年級，因對資訊安全深感興趣而加入軟體品質實驗室，實驗室研究主要的方向是找各式軟體的漏洞和利用漏洞，接觸 CTF 的經歷約一年，興趣是打 CTF 和寫解題流程。