基於 OpenStack 建置雲端資安練習平台(CTF)

林家煜
朝陽科技大學資訊工程系
louis70109@gmail.com

王德譽
朝陽科技大學資訊工程系
助理教授
dywang@cyut.edu.tw

中文摘要

傳統對於伺服器的定義都是一機多服務制，如此一來不僅會讓機器應付太繁雜的工作，且有硬體上的限制。在雲端運算出現後，將繁雜的計算服務個別獨立一台，並透過網路將服務串聯起來形成大型的平台，提供 IT 服務，由於雲端運算使用的是「虛擬」資源，不僅增加計算效率，也降低主機的負載，且能同時提供給使用者更快速的服務。

由於網際網路相關資訊技術越來越重要，資訊安全的重要性也隨之增加，而 Capture the flag(CTF)競賽在國內外是相當盛行的資安競賽，許多大學或資安團隊皆會透過此競賽進行資安技術的練習來增加資安相關實務經驗。

基於上述兩點，本研究利用開放原始碼(Open Source) 的 IaaS 雲端平台 — Openstack，設計線上資安攻防練習平台，介紹 CTF 的形式與所需具備的技術，接著與實際網路環境中較容易遭受攻擊之威脅做競賽題目。平台也採用較簡易的題目做為競賽題目，讓一般使用者參與，透過 CTF 學習資訊安全相關知識，既可避免真實攻擊別人時觸犯法律，也可達到學習目的。

關鍵詞：雲端運算、OpenStack、奪旗競賽、虛擬資源、資訊安全

Abstract

Traditional servers are defined lots services in a machine, this way not only make the machine work too complicated, but have restrictions on the hardware. The complicated compute services are separate into single service, and combine the service through internet be a large platform to provide IT services, cause cloud computing is use "virtual" resources, not only increase the computational efficiency, but also reduce host's load, and can also provide most fast service to users.

Because internet information technology is increasingly important, the importance of information security have increased, and Capture the flag(CTF) contest at domestic and overseas is more popular information security contest, many universities and security team are through the contest to increase information security-related practical experience.

For these, in this study, we are use of open source IaaS cloud platform - OpenStack, designing online information security offensive and defensive practice platform, introduced with the desired form of CTF technology and related technology, then we use more vulnerable to the threat of attack as contest topic. The platform also uses a simple topic as the contest topic for general user to join, and through CTF to learn information security-related knowledge, boy only can avoid breaking the law when the real attack, but to achieve the purpose of learning.

Keywords: Cloud Computing、OpenStack、Capture The Flag(CTF)、Virtual Resource、Information Security.

1. 前言

對於使用雲端運算[1]，大部分開發人員會覺得使用相關套件跟尋找配備是非常困擾的事情，通常都需要跟學校的資訊中心或是企業合作才能獲得較大型配備架設環境來提供學術相關研究。

基於以上問題，本研究找到 OpenStack 基金會所開發的雲端套件，此雲平台可用兩台以上的家用電腦架設，並且利用虛擬機建置服務，而服務配置部分也做得相當好，不會讓主程序飢餓，於是我們利用虛擬機來建置雲端資安練習平台(CTF) [2]來提供社群使用者來練習，藉此提升社群交流以及大眾的資安觀念，讓資安圈可以在台灣多點注意。

2. 系統環境與特性

2.1 系統環境

本研究採用高度穩定性的CentOS[4]系統建置 OpenStack[3]雲端平台並結合 Ruby on Rails 打造線上資安練習平台(Capture the flag, CTF)。

OpenStack 是一個自由軟體和開放原始碼項目打造基礎設施即服務(Infrastructure as a Service)，且有 Apache 許可證授權，並以運算(Nova)、網通(Neutron)、儲存服務(Swift)和儀表板(Horizon)服務來打造雲端服務平台，並且利用虛擬資源對外帶來運算服務，以便利彈性擴充或調度。

當中的服務可依照管理者的需求去安裝，其內部服務為下列 9 項：

(1). Nova：這部分為主要運算服務，目的在於部署與管理虛擬機器的功能，而此服務是可擴展至多台的運算服務，藉此讓虛擬機的運算資源最大化。

(2). Swift：可擴展的分布式儲存平台，以防止單點故障的情況產生，可存放非結構化的資料。

(3). Glance：虛擬機器磁碟或伺服器的映像檔尋找、註冊以及服務交付等功能。

(4). Horizon：提供簡易 Web 介面和管理控制台，讓 IT 人員可以綜觀雲端服務目前的規模與狀態，並能夠統一存取、部署與管理所有雲端服務所使用到的資源。由於 Horizon 可擴展的網頁式 Application，所以可藉此整合第三方的服務或是產品，像是計費、監控或是額外功能。

(5). Cinder：主要是提供 Block 資料存取，另外整合了運算套件，可讓 IT 人員查看儲存設備的容量使用狀態，具有快照(Snapshot)功能。

(6). Keystone：具有中央目錄，能查看哪位使用者可存取哪些服務，並且，提供了多種驗證方式。

(7). Neutron：提供網路連接即服務(Network-Connectivity-as-a-Service) 功能，讓網路管理功能且可擴展，透過 API 來管理的網路架構系統，使其支援眾多的網路供應商和技術，而 IT 人員可分配 IP 位址、靜態 IP 或是動態 IP。

(8). Ceilometer：提供 OpenStack 雲端服務可藉由監控與量測 OpenStack 的使用，來收集 CPU 與網路的使用資料，以提供收費計價（Billing）、評測（Benchmarking）等使用，或是使用這些資料當作評估系統延展性以及進行系統相關統計之用。

(9). Heat：主要提供一個以模板（Template）為基礎的架構來描述雲端的應用，模板中可以讓使用者建立如虛擬映像實體（Instance）、浮動 IP 位址、安全群組（Security Group）或是使用者等 OpenStack 各種資源，也就是說，Heat 讓使用者可以設定一個雲端應用模板，來串連建立設定相關所需的

OpenStack 服務資源，而不必一個個分別去建立設定。

Ruby on Rails（簡稱 ROR，或 Rails）[5]，是一個使用 Ruby 語言寫的開源 Web 應用框架，它是嚴格按照 MVC 結構開發的，透過 MVC 模式，我們可以有系統的建構程式碼，並且分離程式邏輯和使用者介面，讓前端與後端開發者可以獨立作業，也讓程式碼有著一致性的結構，增加程式的可讀性。

以下三點為 MVC 架構在 Rails 中的分類：

(1). Models：代表了應用程式的資料和操作資料的邏輯。在 Rails 中，Models 主要的功能是負責操作資料庫資料表。在大多數的情況，一個資料表就對應了一個 Model。你的應用程式商業邏輯也會放在 Models 中。

(2). Views：代表了應用程式的使用者介面。在 Rails 中，Views 通常就是有內嵌 Ruby 程式(可以操縱如何顯示資料)的 HTML 檔案。Views 負責提供資料給瀏覽器或其他發 HTTP 請求的軟體。

(3). Controllers：它是 Models 和 Views 之間的黏著劑。在 Rails 中，Controllers 負責處理從瀏覽器進來的 HTTP 請求，並對 Models 詢問資料，然後將資料傳進 Views 中顯示出來。

2.2 系統特性

本研究參考 OpenStack - Documentation for Juno[6]的系統文件建置雲平台，利用分散式運算特性將功能分別以三個以上主機(Controller, Network, Compute1…)呈現，如此一來可增加主機們的運算效率並減少作業系統的負擔。雲平台內的虛擬機是使用 KVM(kernel-based virtual machine)建立，且虛擬機必須使用 nova 套件所建立出來的金鑰才可登入，避免安全性相關的問題。

資安練習平台是以存取控制、密碼學以及安全架構設計網頁與題目，目的是透過模擬各式各樣的安全性問題，讓使用者從中得知漏洞的嚴重性並且了解攻擊是如何進行，增加使用者資安的觀念進而防範漏洞的產生。

3. 系統概述

3.1 系統架構

本節將介紹我們的研究方法及步驟，系統架構如圖1所示，後續我們將對系統的流程與架構做詳細介紹，其系統架構如下：

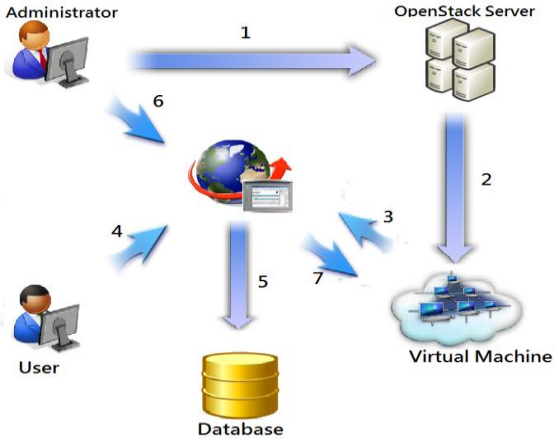

圖 1 系統架構流程圖

步驟一： 管理者可連接至伺服器裡並透過指令選用虛擬機的資源來分配大小並建立。

步驟二： Controller 會透過遠端程序呼叫(Remote Procedure Call, RPC)[7]相關套件呼叫 Network 以及 Compute Node 裡的 Nova 跟 Neutron 套件建立並配置網路給虛擬機。

步驟三： 由於是自製映像檔，檔案內已經將整個服務建置完成，所以管理者只需將資安練習平台啟動即可。

步驟四： 使用者透過網頁連接平台，註冊並登入後即可練習。

步驟五：網頁會將使用者資料存至資料庫中方便管理者管理。

步驟六：管理者可透過 noVNC[8]所提供的功能在網頁查看虛擬機是否有出錯，以及可在網頁上觀察 OpenStack 整體服務狀況。

步驟七：Horizon 則將管理者的要求傳回虛擬

機去變更。

OpenStack 實體主機規格如表 1 所示，因 Compute Node 需使用虛擬機來架設服務，所以需要規格較好的 CPU 與較大的記憶體空間，而 Controller Node 與 Network Node 雖然需處理不同的服務，但相較於 Compute Node，所需要的記憶體比較少，CPU 規格也不需跟 Compute Node 一致。

表 1 OpenStack 主機規格

Node	CPU	Core	Memory
Controller	Intel E7400 2.80GHz	2	4GB
Network	Intel E7400 2.80GHz	2	4GB
Compute1	Intel i7-4790 3.6GHz	4	4GB

本研究使用 Ruby on Rail(RoR)來開發 CTF 網站，使用 MVC(model-view-controller)系統，model 負責與資料庫溝通，view 負責將介面顯示在使用者的瀏覽器上，controller 則分析使用者的 Http Request，並轉交給 model 和 view 做下一步處理，由於邏輯跟使用者介面分離，在程式碼維護上也容易許多，架構下如圖 2 所示。

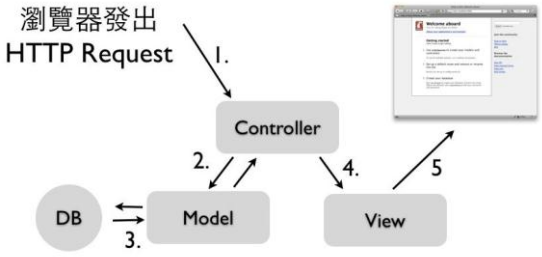

圖 2 Rails MVC 架構圖

除此之外，RoR 有兩個開發原則，分別是不做重複的事(Don't Repeat Yourself)和慣例優於設定(Convention Over Configuration)，第一個因 RoR 是基於 Ruby 的網頁架構，所以擁有 Ruby 的動態性質，能夠輕鬆把程式碼縮減至最少，第二個是 RoR 會預設好大部分的設定，在開發過程中不需在設定檔上花費心力，也因為這些特色，讓開發人員可以在短時間內提升生產力。

3.2 系統特色

本研究以 CentOS 7 作業系統架設 OpenStack 雲端平台並與 Ruby on Rails 作為練習平台開發軟體，本研究有以下幾點特色：

(1). 管理性（manageability）：使用Horizon所提供的Dashboard管理其虛擬機狀態以及虛擬機所提供的服務。

(2). 資訊安全（security）：透過CTF網站可練習到密碼學、資訊隱藏、網路安全以及系統安全相關的技術。

(3). 高可用性（high availability）：使用KVM所提供的快照(Snapshot)備份，如此一來即使在使用過程中系統故障，也可藉此功能還原系統找出錯誤。

(4). 分散式管理 （distributed management）：將主機分別以Controller、Network以及Compute去作分類，讓主機分別去處理計算量較複雜的部份，例如：網路分配計算、虛擬機資源分配計算等，降低主機負載且可增加虛擬機的運轉速度，如此一來也可將所消耗之總能源降至最低。

我們的系統包含以下二大部分：

1. OpenStack 雲平台：可在網頁(Horizon)上得知虛擬機目前使用狀態以及主機資源被利用多少。

2. CTF 練習平台：不同分類的題型讓使用者可以依照自己喜好去作練習，分數也依照題目難度而定，且有排行榜可讓使用者得知自身的程度在哪。

4. 系統實現

本研究分為二大部分：OpenStack 雲平台、CTF 練習平台，以下為本研究規劃之操作介面流程圖，並將針對這二大部分詳細說明。

4.1 OpenStack 雲平台

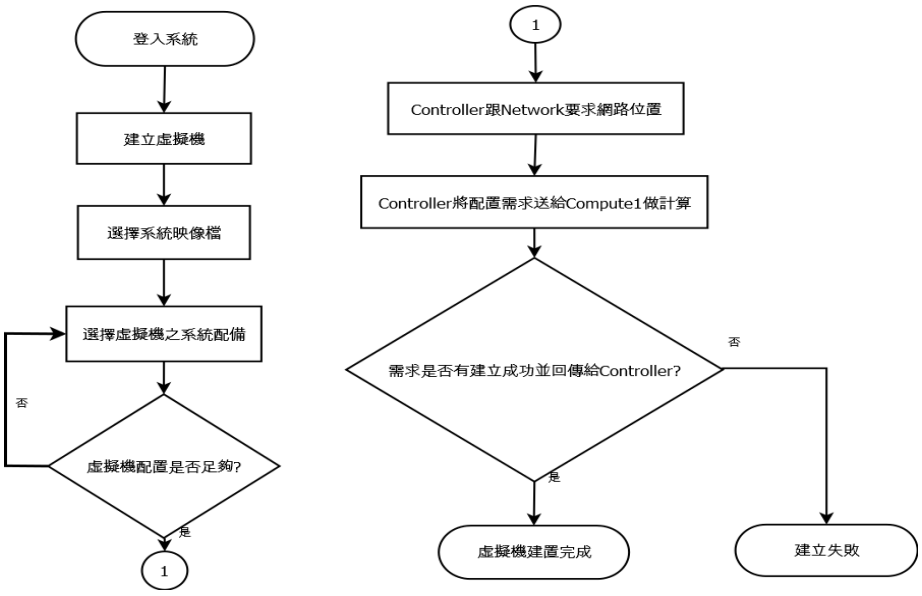

圖 3 OpenStack 流程圖

圖 3 顯示整個流程，管理者在登入系統後，若要建置虛擬機，Controller Node 會要求管理者所需要配置給虛擬機之資源，其後將配置內容傳送給 Compute1 Node，讓 Compute1 利用 KVM 內部指令來建置管理者所需要的虛擬機，若資源不足或網路錯誤則會回傳錯誤，建置成功後則會跟 Network Node 要求一組 IP，讓管理者方便登入。

當前OpenStack主要用於測試各項伺服器映像檔之建置、雲端運算實驗、以及實驗室同學一同使用。本論文就是在OpenStack 上開發CTF雲端練習平台，管理者可以透過網頁介面(Horizon)登入OpenStack，並可在登入時看到整體資源使用率。

圖 4 為管理者在網頁上建立虛擬機所需選擇大小以及映像檔的地方，可在當中選擇所要配置的私鑰以及對外以及內部網路，如圖 5 所示。

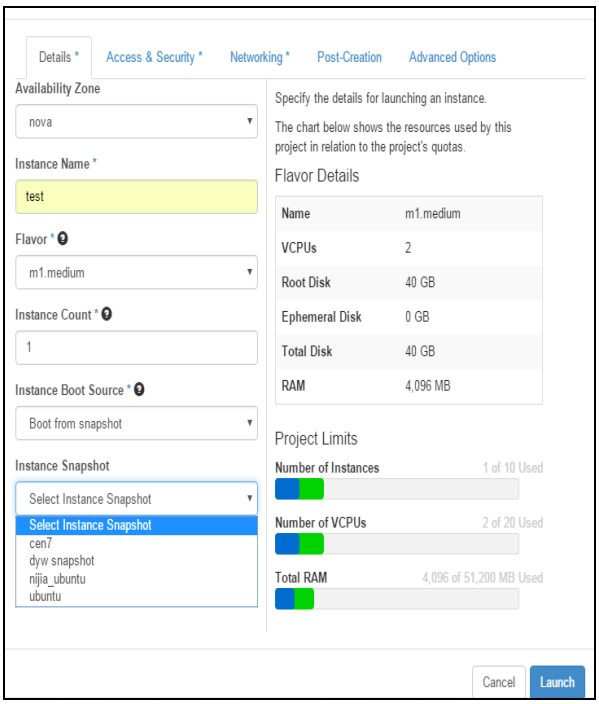

圖 4 虛擬機建立圖

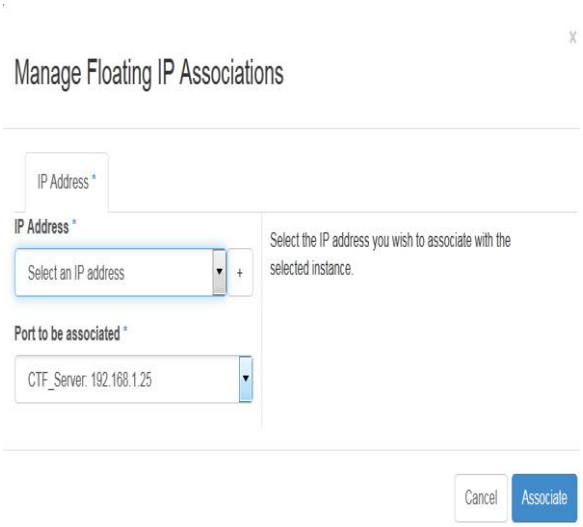

圖 5 對外 IP 分配圖

圖 6 則是在 OpenStack 中虛擬機的資訊圖，當中可以知道對內外 IP、私鑰、資源大小以及能在下拉選單中使用快照、重新配置資源、虛擬機主控台等功能。

如圖 7 所示，若管理者較習慣在命令自元操作，也可在指令列上直接下建立指令，執行速度也較網頁上快速許多，另外也可在主控台中利用 noVNC 所提供的功能直接在網頁上使用虛擬機來做軟體服務，如管理者是用非 Linux 時也可操作，圖 8 為實際使用情況。

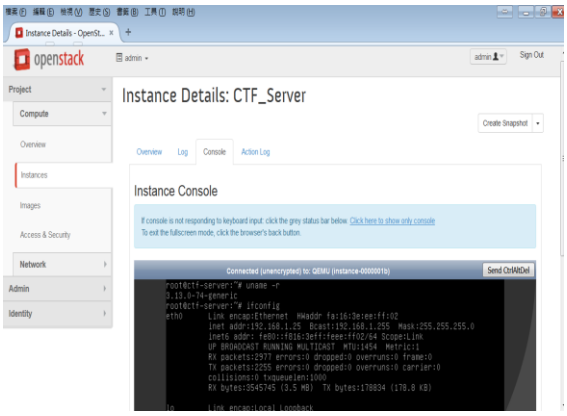

圖 8 遠端命令字元使用圖

Horizon 中有選項可察看網路拓樸圖，因網路部分是使用 Open vSwitch[9]來打造整體環境，可讓管理者在當中看到使用中的 Router 以及整體配置狀況，如要限制特定網域使用，也可直接點選 Router 進去設定其限制。

4.2 CTF 練習平台

我們主要套用 Bootstrap 這個框架，它是由 Twitter[10]所開發的專案，它提供了很多版面配置樣式，可以輕鬆的套用各種功能，再外觀上面也非常講究，而且說明文件中詳細的解說每個樣式跟功能的用法，非常容易上手。

CTF 主要是透過在網頁上建置題目提供使用者練習，藉由 CTF 來增加大眾的資安觀念，題目當中也會有提示，讓系統給予使用者練習中的回饋，流程圖如下圖 9 所示：

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
CTF_Server	nijia_ubuntu	192.168.1.25 163.17.11.80	m1.medium	key1	Active	nova	None	Running	3 days, 22 hours	Create Snapshot

圖 6 虛擬機資訊圖

```
$ nova boot --flavor m1.tiny --image cirros-0.3.3-x86_64 --nic net-id=DEMO_NET_ID \
  --security-group default --key-name demo-key demo-instance1
+--------------------------------------+-----------------------------------------------------------------+
| Property                             | Value                                                           |
+--------------------------------------+-----------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                          |
| OS-EXT-AZ:availability_zone          | nova                                                            |
| OS-EXT-STS:power_state               | 0                                                               |
| OS-EXT-STS:task_state                | scheduling                                                      |
| OS-EXT-STS:vm_state                  | building                                                        |
| OS-SRV-USG:launched_at               | -                                                               |
| OS-SRV-USG:terminated_at             | -                                                               |
| accessIPv4                           |                                                                 |
| accessIPv6                           |                                                                 |
| adminPass                            | vFW7Bp8PQGNo                                                    |
| config_drive                         |                                                                 |
| created                              | 2014-04-09T19:24:27Z                                            |
| flavor                               | m1.tiny (1)                                                     |
| hostId                               |                                                                 |
| id                                   | 05682b91-81a1-464c-8f40-8b3da7ee92c5                            |
| image                                | cirros-0.3.3-x86_64 (acafc7c0-40aa-4026-9673-b879898e1fc2)      |
| key_name                             | demo-key                                                        |
| metadata                             | {}                                                              |
| name                                 | demo-instance1                                                  |
| os-extended-volumes:volumes_attached | []                                                              |
| progress                             | 0                                                               |
| security_groups                      | default                                                         |
| status                               | BUILD                                                           |
| tenant_id                            | 7cf50047f8df4824bc76c2fdf66d11ec                                |
| updated                              | 2014-04-09T19:24:27Z                                            |
| user_id                              | 0e47686e72114d7182f7569d70c519c9                                |
+--------------------------------------+-----------------------------------------------------------------+
```

圖 7 虛擬機建立指令

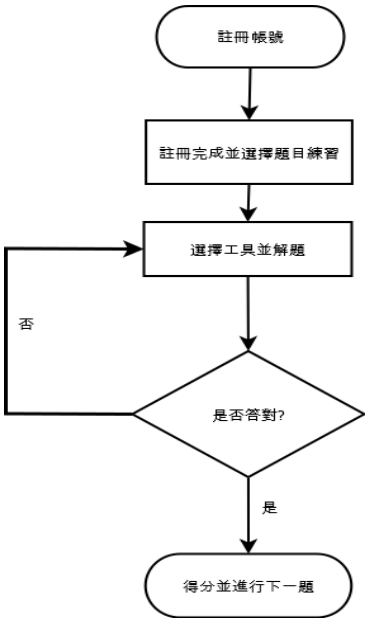

圖 9 CTF 流程圖

網頁主要分為幾個部份，主頁、題目、排名、註冊及登入，會由路由設定檔中將使用者的請求做導向，設定檔如圖 10 所示。

```
Rails.application.routes.draw do
 devise_for :users
 root 'welcome#index'
 match "/topic/index" => "topic#index", :via => [:get, :post]
 get "rank" => "rank#index"
 get 'testflag/flag3'
 get 'welcome/download'
end
```

圖 10 路由設定檔

devise_for會分別導向到註冊及登入，root則是主頁導向，其他分別將題目、排名，使用get導向到對應的頁面第並用post將提交的旗幟(flag)送到題目頁面作驗證。題目頁面是整個CTF的核心部份，它必須顯示題目、驗證答案以及資料庫的更新，圖11是題目頁面主要驗證程式碼。

```
def chk_Flag()
   @notice = ""
   flag = params[:flag] #
   task = params[:task] # task

   if Task.find(current_user.id).read_attribute_for_serialization(task)
     @notice = "Already"
   elsif flag_list[task] == flag
     @notice = "Success"

     #Topic UPdate
     t = Task.find(current_user.id)
     condition = {task => 1}
     t.update(condition)

     #score to user
     sum = t.point + correct_point[task]
     score = {"point" => sum}
     t.update(score)
     redirect_to :back
   else
     @notice = "Fail"
   end
end
```

圖 11 答案驗證函式圖

網站內題目類別分為綜合練習、Liunx 提權以及網站資訊安全，在第一項內主要分為檔案漏洞、資訊洩漏以及 Javascript 練習，第二項則是需要使用者連線至並透過一些指令提權找到檔案內的訊息，藉此找到答案，最後一項主要是要讓使用者觀察網頁程式碼中的漏洞，透過一些例外請求來讓主機發生錯誤，利用錯誤讓答案產生出來。

5. 結論

本論文以 IaaS 作為基礎架構所製作的 OpenStack 並結合 Ruby on Rails 來打造 CTF 練習平台，供使用者能在雲端學習資安領域的資源，且使用者也不知雲端主機所在的位置在哪，即使損毀只要利用 OpenStack 所提供的功能即能快速還原雲端虛擬機，不須重新將主機重灌來還原系統，比起過往一服務一主機有著不同的好處。

但由於主機配備較低，我們可以認為這是一個初步的概念，其應該尋找更大型的配備來建置雲平台，提供使用者更好的體驗。然而，目前的研究雖只讓使用者練習到最近熱門的資安，日後也可在雲平台上架設不同的性質的作業系統來供他人使用，讓 OpenStack 能有更多不同的研究。

參考文獻

[1] "Cloud Computing," https://zh.wikipedia.org/wiki/%E9%9B%B2%E7%AB%AF%E9%81%8B%E7%AE%97

[2] Raghu Raman, Sherin Sunny, Vipin Pavithran, Krishnasree Achuthan, "Framework for evaluating Capture The Flag(CTF) security competitions," **International Conference for Convergence of Technology**, pp.1-5, 2014

[3] "OpenStack 官網," https://www.openstack.org/

[4] "CentOS 官網, "https://www.centos.org/

[5] "Ror introduction," http://rubyonrails.org/

[6] "OpenStack - Documentation for Juno," http://docs.openstack.org/juno/install-guide/install/yum/content/ch_preface.html

[7] "Remote procedure call," https://en.wikipedia.org/wiki/Remote_procedure_call

[8] "noVNC," http://www.vpsee.com/2013/07/integrating-novnc-with-our-vm-control-panel/

[9] "Open vSwitch," http://roan.logdown.com/posts/238771-openvswitch-debug-enviroment

[10] "Twitter bootstrap," http://getbootsrap.com/2.3.2/index.html