Network Administration and System Administration
Spring 2024

Homework 6
B12902080 資工一吳威錡

本作業除了參考額外標記超連結的網址外，沒特別附網址的題目亦會參考 ChatGPT，並與 B12902054 彭詳睿、B12902055 楊翔宇 討論。
另外，因爲我電腦在做此份作業時多次當機導致我需要多次地重新設定 OPNSense，最後一次安裝時並沒有每個都去測試是否正確，所以可能會有我漏設定東西的情況，一切以報告內設定爲準，敬請見諒。

Short Answers

1.

Block：
功能：會靜默地丟棄封包，不讓來源端知道
適合情況：

- 當封包來自無法信任的網路，不想讓來源端知道它已被禁
- 要完全阻止特定封包通過防火牆，如要阻止特定 IP 位址的電腦連線到此網路，或要阻止特定埠的流量通過

Reject：
功能：丟棄封包並讓來源端知道，TCP 回傳 RST 包，UDP 回傳 ICMP Unreachable 包
適合情況：

- 在內部網路可以讓用户馬上知道封包已被禁止，不用等
- 要讓來源端知道封包已被此防火牆擋下
- 要測試防火牆規則是否正確

參考資料：1

2.

interface net 代表的是一整個網路介面，而 interface address 代表的是單一個 IP 位址。
interface net：指定整個介面的網路位址範圍

- 例如 LAN 介面爲 192.168.1.0/24, WAN 介面爲獲得的公有 IP 範圍
- 適用於對整個介面網段進行規則設定

interface address：只指定介面本身的 IP 位址

- 例如 LAN 介面的 192.168.1.1, WAN 介面的公有 IP
- 適用於對單一介面 IP 進行規則設定，而非整個網段

參考資料：1, 2, 3

3.

stateful firewall 和 stateless firewall 的主要區別在於它們處理封包的方式：stateful firewall 會追蹤連接的狀態，而 stateless firewall 則不會。
stateful firewall：

- 功能：能夠追蹤和管理網路上 hosts 間的連接狀態，包括連接的起始、結束和狀態。通常會建立一個狀態表 (state table) 來追蹤這些狀態。
- 原理：當封包通過防火牆時，會記錄封包的來源 IP 位址、目的 IP 位址、Ports 等訊息。當對應的回應封包到達時，它會檢查這些訊息，確定它是否屬於一個已經建立的、允許通過的連線，可以更有效地過濾和管理網路流量。
- 優點：提供安全性和可控性，可以分析封包的前後關連及內容，並對連線內容進行動態的管理和過濾。
- 缺點：會消耗更多的資源來維護狀態及分析訊息。

stateless firewall：

- 功能：僅基於個別封包的特徵，如來源 IP 位址、目的 IP 位址、Ports 等來進行過濾和阻擋。
- 原理：通常不會追蹤封包傳輸的狀態，也不會記錄任何關於封包的前後關連及內容，只是單純地檢查每個封包是否符合預定的規則。

- 優點：處理速度更快，並且對系統資源的要求較少。
- 缺點：由於缺乏封包的前後關連，可能無法有效地應對某些特定類型的攻擊或是應用層面的控制。

OPNsense 是一種 stateful firewall，可以追蹤 TCP、UDP、ICMP 等多種協定的連線狀態。OPNsense 還具有許多其他安全功能，例如入侵偵測、防毒、內容過濾等。
參考資料：1

4.

OPNSense 是 2015 年自 pfSense 衍生而出，都是基於 FreeBSD 的開源路由防火牆，兩者在功能上非常相似，但也有一些技術上的差異。例如：

1. 套件管理：pfSense 使用 FreeBSD 原生的 pkg 進行套件管理，而 OPNSense 用 OPNsense Package Manager (OPM)，由 OPNsense 社群開發的套件管理工具，除提供 pfSense 官方套件庫中的所有套件，也有 OPNsense 社群開發的套件，可提供更方便的套件管理功能。
2. 入侵偵測系統：pfSense 使用 Snort 做爲入侵偵測系統 (IDS)，而 OPNSense 使用 Suricata。Snort 和 Suricata 都是功能強大的 IDS，但 Suricata 的效能較佳，可以處理更大的流量，且其規則語法更易於理解和使用。此外，OPNsense 還提供了一個名爲 OPNsense Firewall 的額外入侵防禦系統 (IPS) 組，可以提供更全面的入侵防禦功能。

參考資料：1

OPNSense

5.

1. 先使用與 lab 相同方法安裝 OPNSense 到虛擬機上，並設立三個 VLAN 並給予其對應的 subnet
2. 同樣參考 lab 方法在 client 上做相同設定
3. 在 GUI 介面中的 Services>ISC DHCPv4> <VLAN interface ID> 中把 range 設成 subnet ID，並在下方 DNS servers 處增加 8.8.8.8 與 8.8.4.4，更改後儲存並 restart

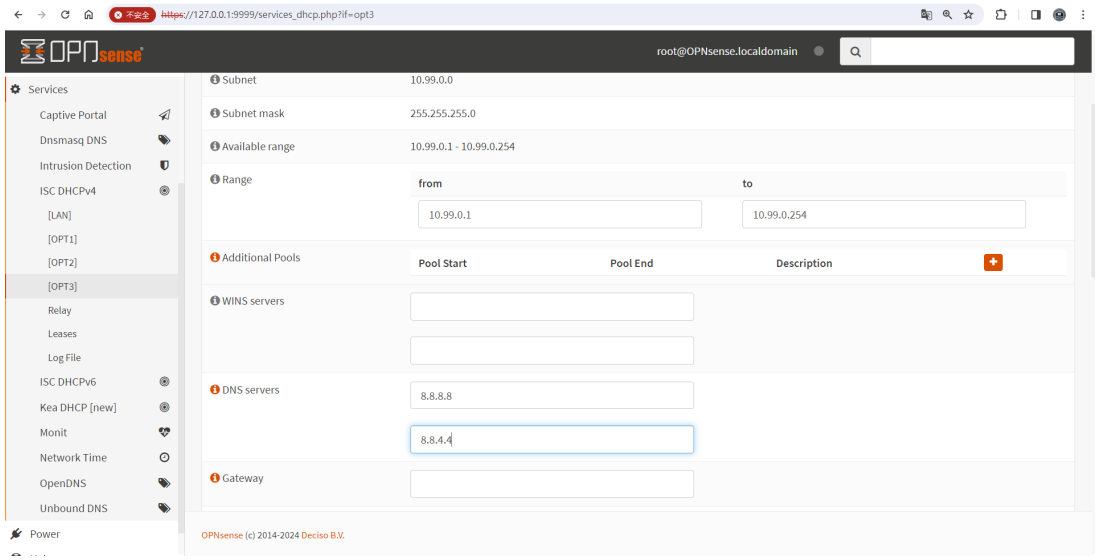


Figure 1: 修改 DNS server

6.

在 GUI 介面中的 Firewall>Aliases 中新增：

- GOOGLE_DNS：
  - Type: Hosts
  - Content: 8.8.8.8, 8.8.4.4
- ADMIN_PORTS：
  - Type: Ports
  - Content: 22, 80, 443
- CSIE_WORKSTASTIONS：

- Type: Hosts
- Content: ws1.csie.org, . . ., ws5.csie.org

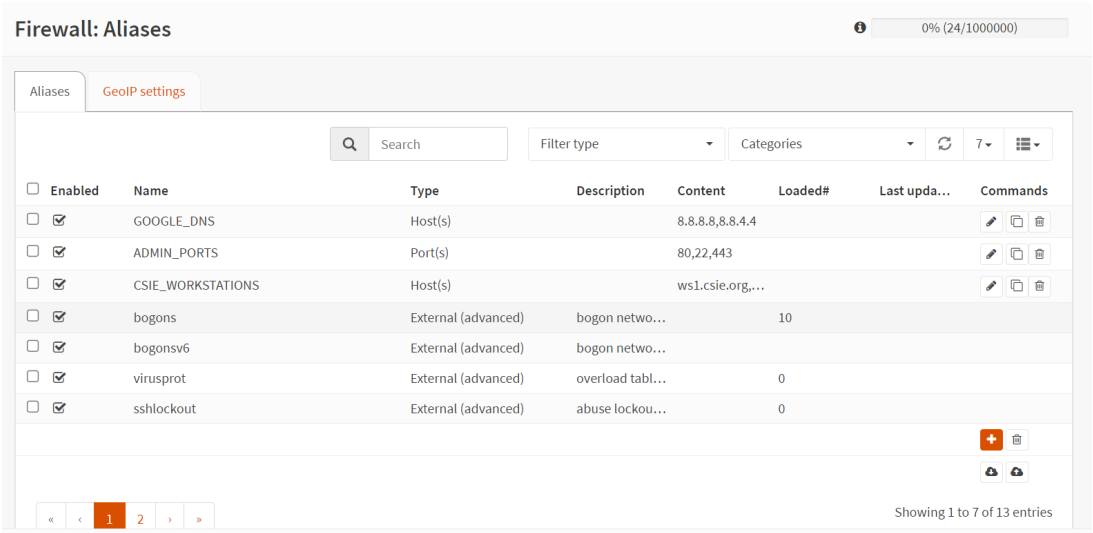


Figure 2: 設定 alias

7.

在 GUI 介面中的 System>Settings>Administration 中，勾選 `Enable Secure Shell`, `Permit root user login`, `Permit password login`，並在 Listen Interfaces 選擇 LAN, OPT3(VLAN 99 net)，使其可以與外網及 VLAN 99 中的機器以 SSH 連接。

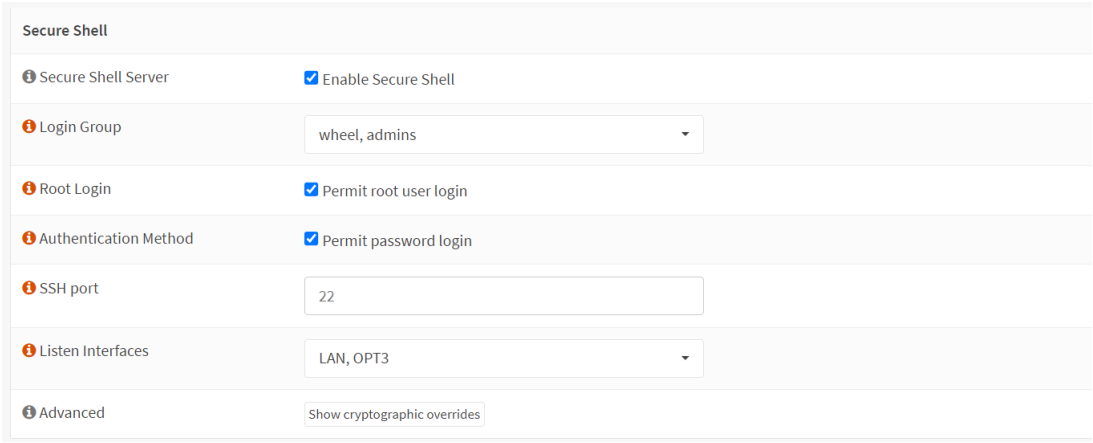
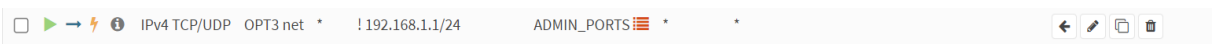


Figure 3: OPNSense 的 SSH 設定

接著在 OPT3 的防火牆 rules 下新增以下規則：



其中，`192.168.1.1` 爲 OPNSense 的 LAN IP。

8.

VLAN 99 的詳細設定如下圖：

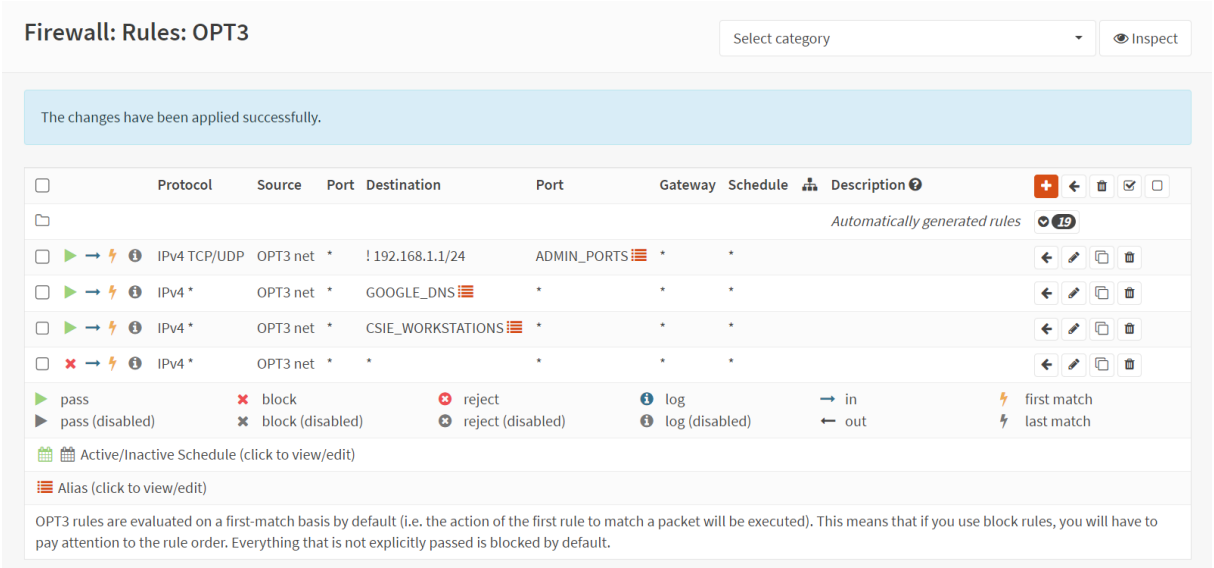


Figure 4: OPT3 setting

允許存取 CSIE_WORKSTATIONS 與 GOOGLE_DNS。

成功 ping DNS server 截圖：

```
localhost:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=55 time=7.044 ms
64 bytes from 8.8.8.8: seq=1 ttl=55 time=8.084 ms
64 bytes from 8.8.8.8: seq=2 ttl=55 time=7.641 ms
64 bytes from 8.8.8.8: seq=3 ttl=55 time=7.582 ms
64 bytes from 8.8.8.8: seq=4 ttl=55 time=6.874 ms
64 bytes from 8.8.8.8: seq=5 ttl=55 time=12.419 ms
64 bytes from 8.8.8.8: seq=6 ttl=55 time=6.421 ms
64 bytes from 8.8.8.8: seq=7 ttl=55 time=7.015 ms
64 bytes from 8.8.8.8: seq=8 ttl=55 time=7.016 ms
64 bytes from 8.8.8.8: seq=9 ttl=55 time=6.721 ms
64 bytes from 8.8.8.8: seq=10 ttl=55 time=7.371 ms
64 bytes from 8.8.8.8: seq=11 ttl=55 time=7.722 ms
64 bytes from 8.8.8.8: seq=12 ttl=55 time=7.859 ms
64 bytes from 8.8.8.8: seq=13 ttl=55 time=7.309 ms
64 bytes from 8.8.8.8: seq=14 ttl=55 time=10.205 ms
64 bytes from 8.8.8.8: seq=15 ttl=55 time=7.688 ms
64 bytes from 8.8.8.8: seq=16 ttl=55 time=6.838 ms
64 bytes from 8.8.8.8: seq=17 ttl=55 time=7.033 ms
64 bytes from 8.8.8.8: seq=18 ttl=55 time=12.753 ms
64 bytes from 8.8.8.8: seq=19 ttl=55 time=11.300 ms
^C
--- 8.8.8.8 ping statistics ---
20 packets transmitted, 20 packets received, 0% packet loss
round-trip min/avg/max = 6.421/8.144/12.753 ms
localhost:~# ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1): 56 data bytes
^C
--- 10.8.0.1 ping statistics ---
12 packets transmitted, 0 packets received, 100% packet loss
localhost:~#
```

Figure 5: ping 8.8.8.8

成功 traceroute 截圖：

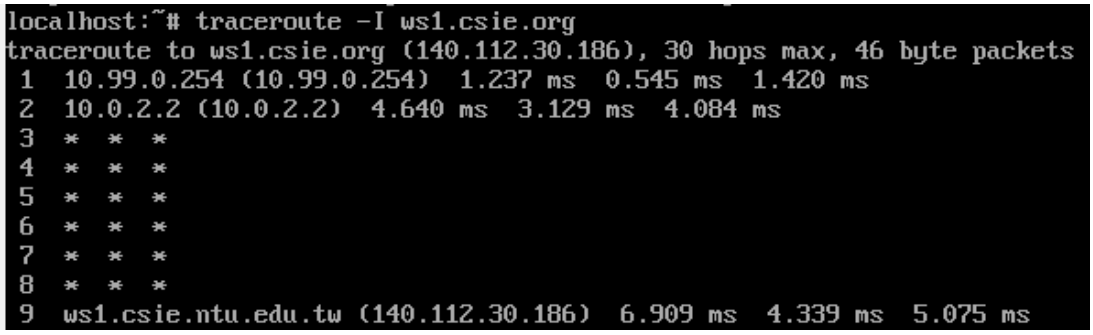


Figure 6: traceroute ws1.csie.org

成功 ssh 到 OPNSense 的截圖：

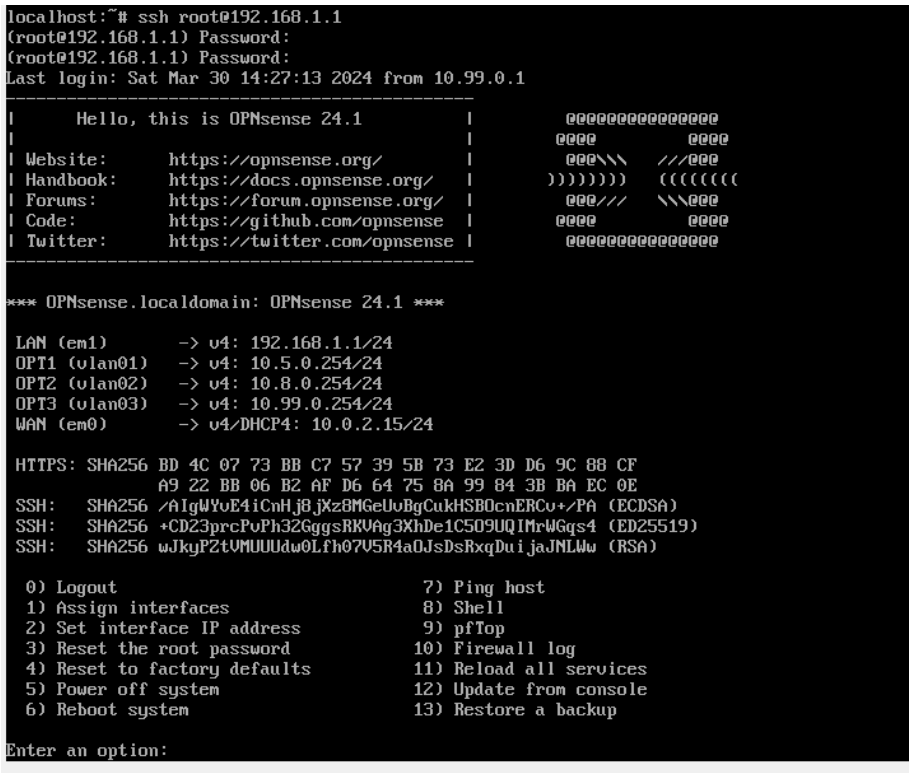


Figure 7: ssh into OPNSense management interface

參考資料：1

9.

第九題的操作同 lab，不多做描述。

```
localhost:~# ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1): 56 data bytes
64 bytes from 10.8.0.1: seq=0 ttl=63 time=4.472 ms
64 bytes from 10.8.0.1: seq=1 ttl=63 time=6.018 ms
64 bytes from 10.8.0.1: seq=2 ttl=63 time=3.090 ms
64 bytes from 10.8.0.1: seq=3 ttl=63 time=5.391 ms
64 bytes from 10.8.0.1: seq=4 ttl=63 time=3.902 ms
^C
--- 10.8.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 3.090/4.574/6.018 ms
```

Figure 8: VLAN 5 ping VLAN 8

```
localhost:~# ping 10.5.0.1
PING 10.5.0.1 (10.5.0.1): 56 data bytes
^C
--- 10.5.0.1 ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
localhost:~#
```

Figure 9: VLAN 8 ping VLAN 5

```
localhost:~# ssh root@192.168.1.1
```

Figure 10: VLAN 5 can't ssh into OPNSense server

10.

1. 在 GUI 介面中 Firewall>Settings>Schedules 新增名爲 `March14` 的時間表，代表 2024/3/14 這天的早上 0:00 至晚上 23:59
2. 在 OPT1 的 rules 中新增：在 March14 的時候禁止封包傳入與傳出的 rules

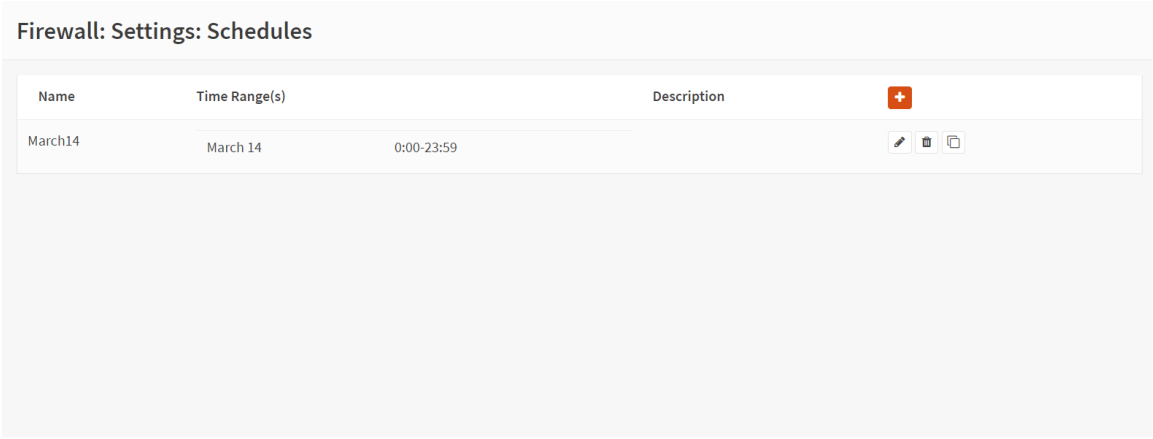


Figure 11: set schedule

OPT1, OPT2 的詳細設定如下圖，實際操作同 lab：

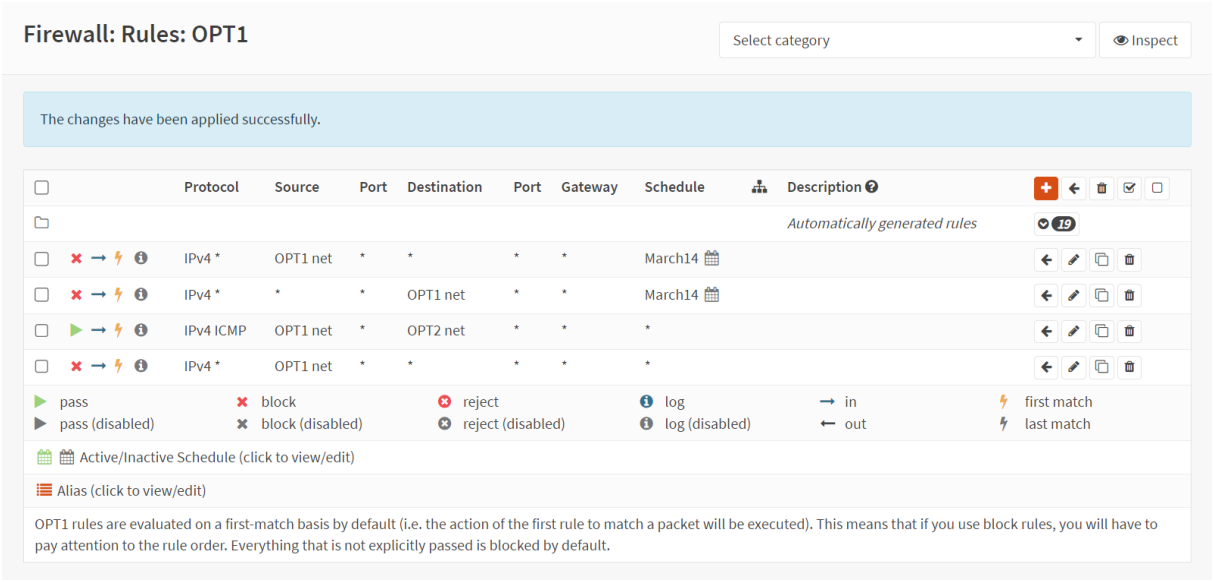


Figure 12: OPT1 setting

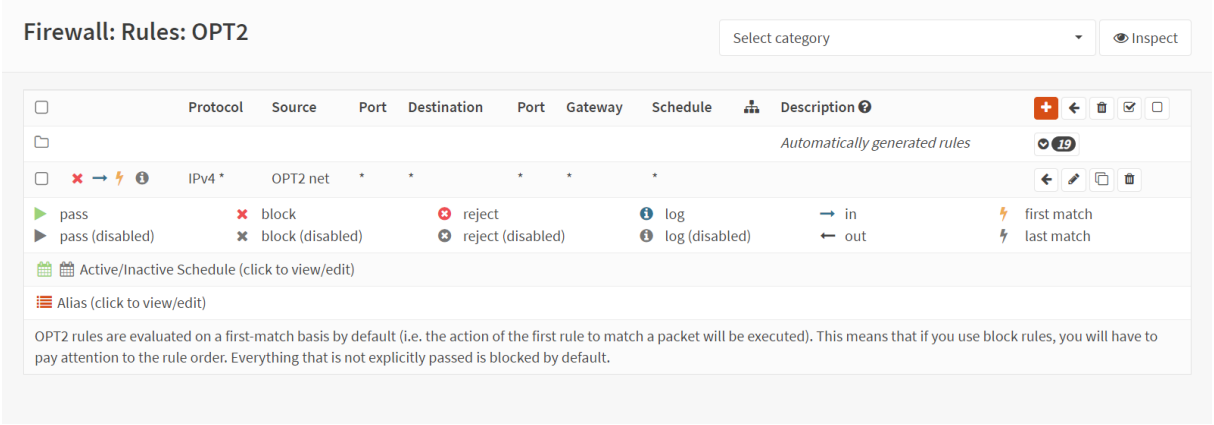


Figure 13: OPT2 setting

11.

另外開一台 alpine linux VM，下載 hping3 並將其連到 VLAN 99。
使用指令下載 hping3：`apk add hping3 --update-cache --repository http://dl-cdn.alpinelinux.org/alpine/edge/testing`
使用指令：`hping3 -i 1 --data <data_size> 192.168.1.1` 發送資料給 OPNSense server。

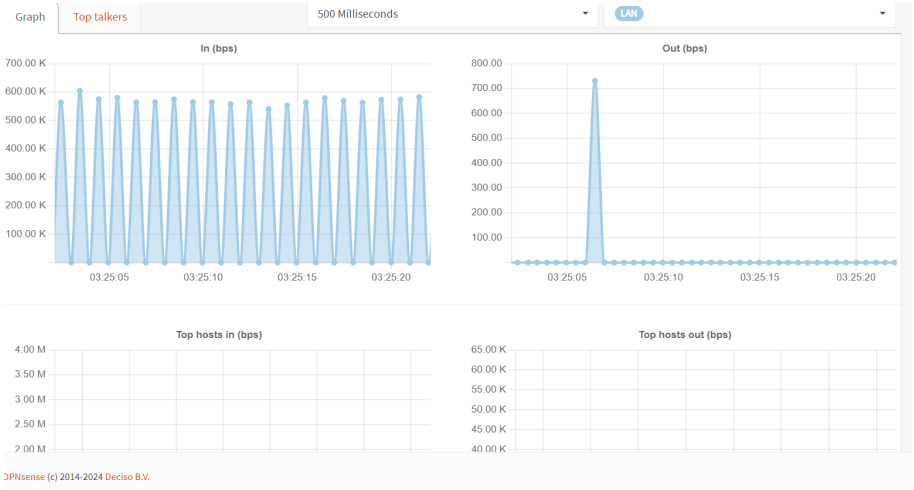


Figure 14: 0.1MB

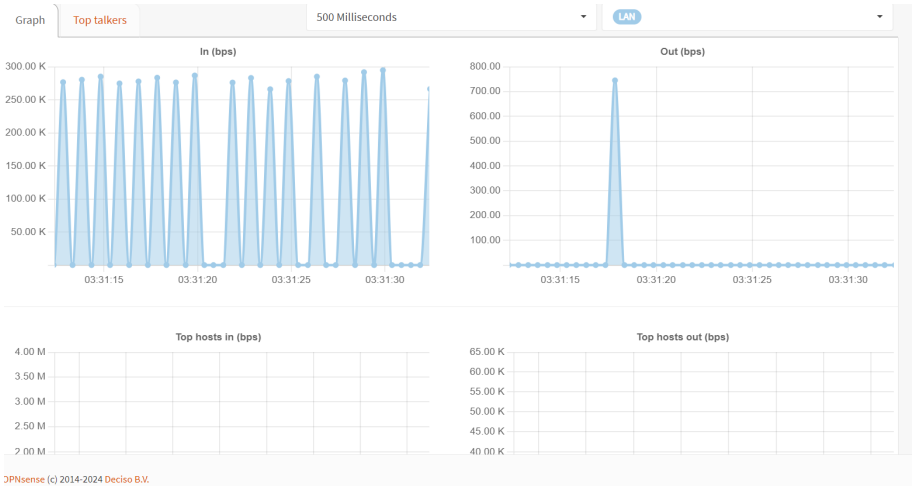


Figure 15: 1MB

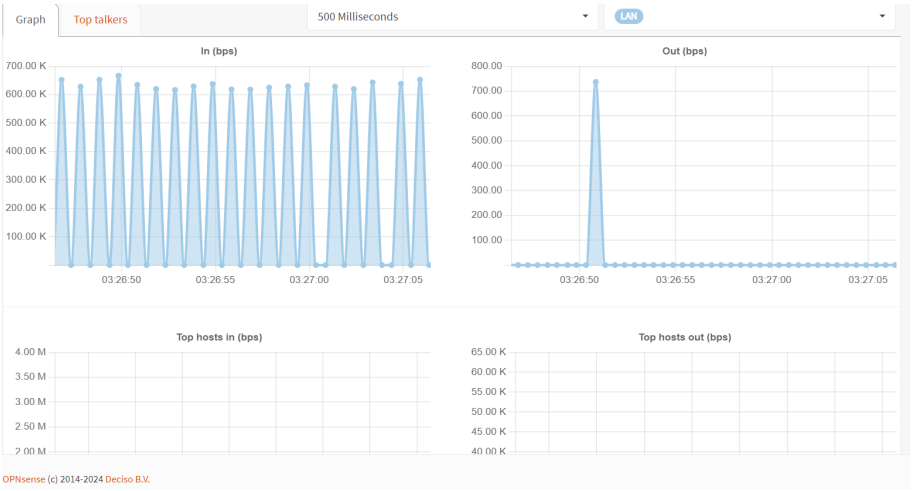


Figure 16: 10MB

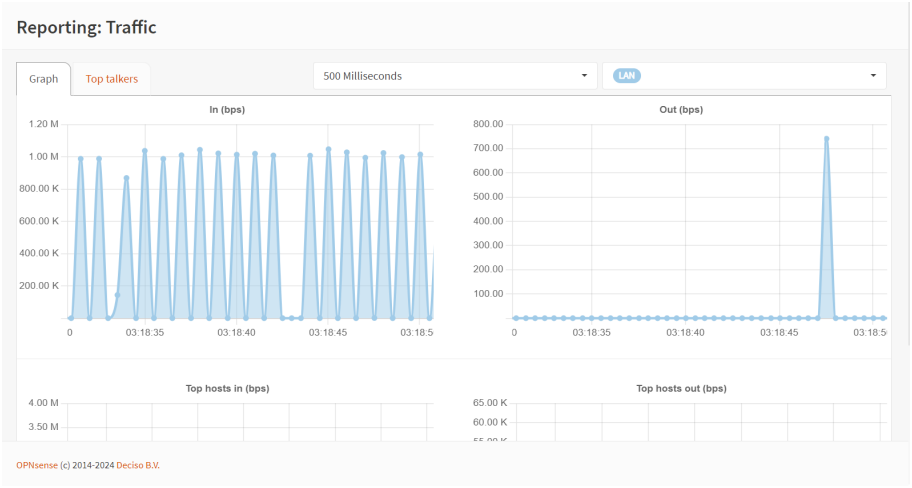


Figure 17: 50MB

分析圖形：

1. 除了 0.1MB 至 1MB 的區間外，每次發送的數據量越大，防火牆收到的流量也越大。
   我詢問 ChatGPT 了爲何發送 0.1MB/s 時防火牆爲何會收到相較 1MB/s 更大的流量，可能有以下原因：(但我無法查證這些原因是否正確)

   - 大的封包在 fragmentation 的過程中產生封包遺失
   - 以較低的速率發送數據時，可能會伴隨著較多的連接建立和拆除的開銷，以及數據包的控制開銷，導致防火牆在統計上接收到的流量量較大。
   - 防火牆可能會針對較小的數據包進行優化，以提高處理效率。因此，儘管每秒發送的數據量較少，防火牆可能仍然會接收到較大的流量。

2. 送出去的流量會有不平均的情況，大約每秒會產生一個高峰

3. 即便每秒送出的數據量相同，防火牆接收的流量不會固定相同，甚至在有些時刻會有完全沒流量的情況，而傳送的數據量越大越容易發生此現象。

參考資料：1, 2, 3, 4, 5

12.

見 zip 檔