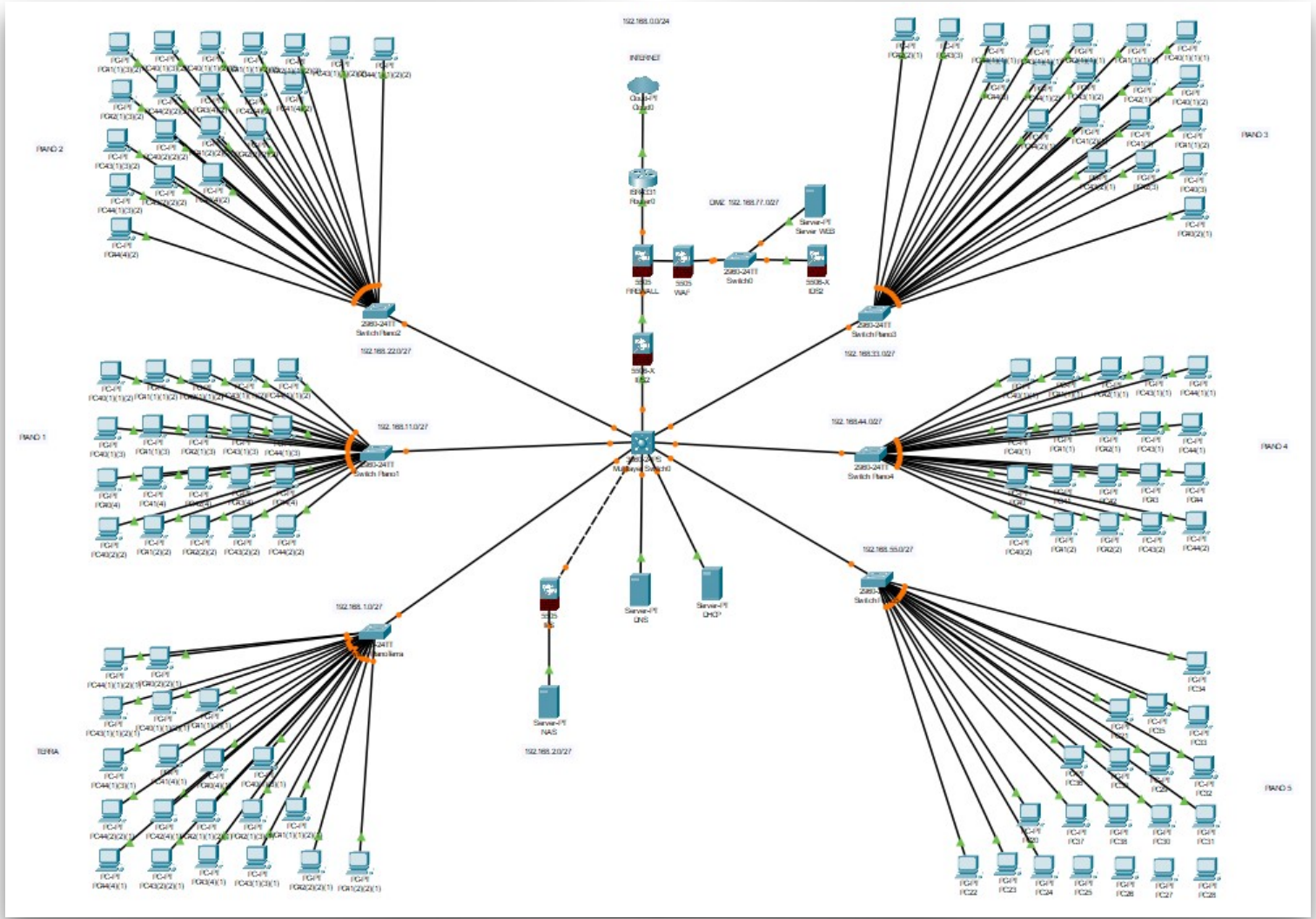
PROGETTO DI RETE PER LA COMPAGNIA THETA

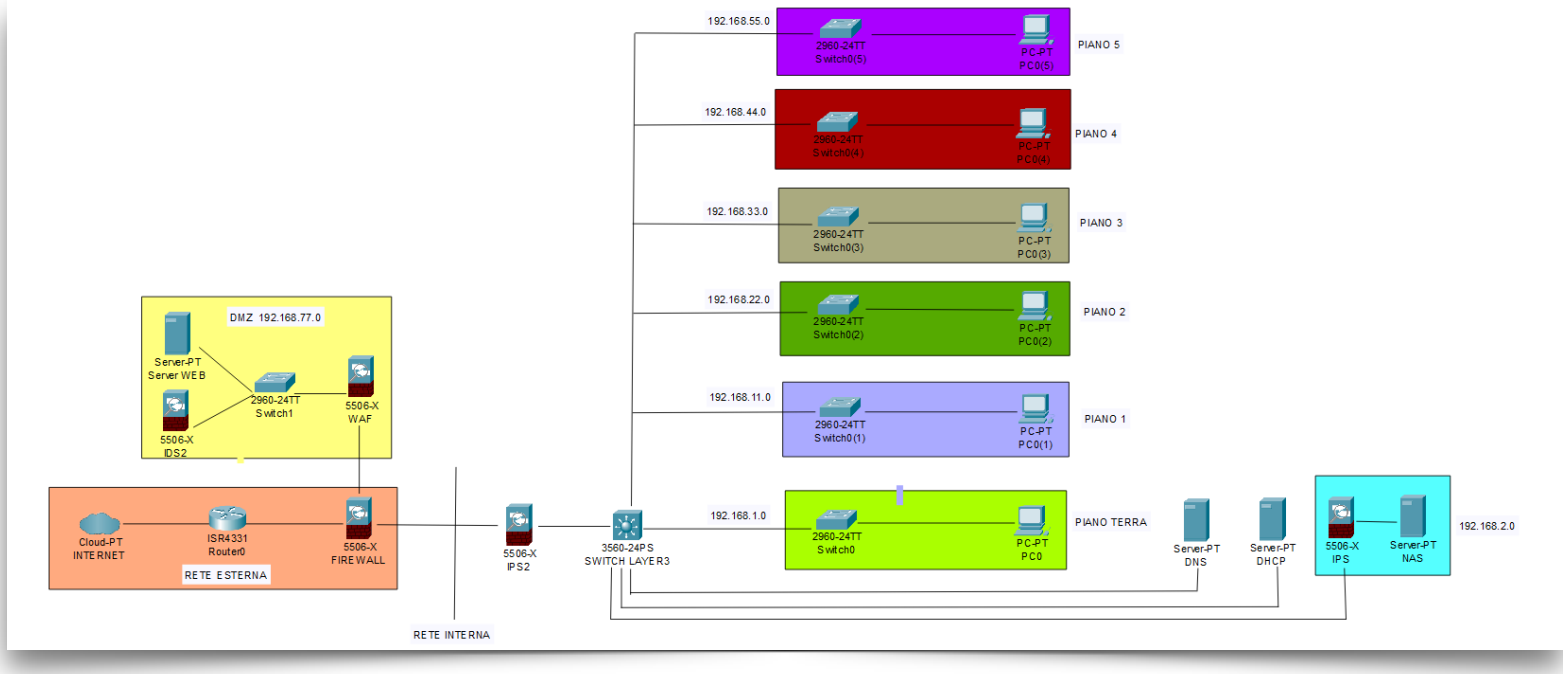
Siamo stati ingaggiati dalla compagnia Theta per sviluppare un progetto di rete e un preventivo di spesa per la loro infrastruttura IT. Sappiamo che l'edificio è suddiviso in 6 piani e ogni piano prevede 20 computer per un totale di **120 computer**. In più devono essere presenti: **1 Web Server**, **1 Firewall perimetrale**, **1 NAS** e **3 IDS/IPS**.

- Ecco come appare la rete in Cisco Packet Tracer:



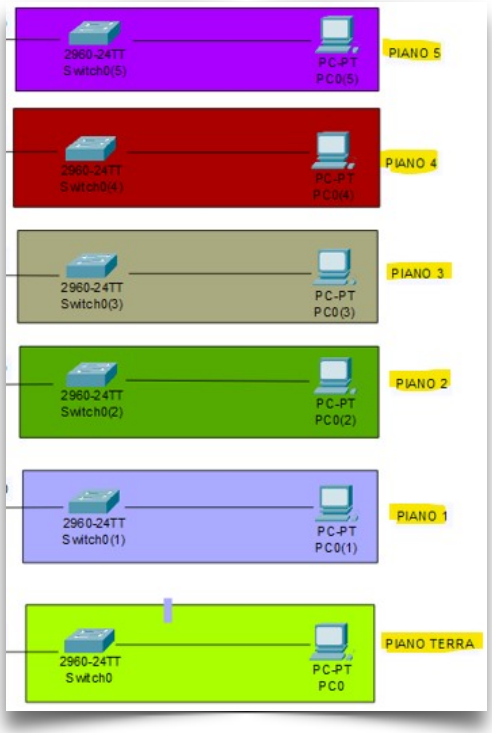
- *STRUTTURA DELLA RETE*

- Abbiamo creato un modello stilizzato in modo da facilitare la comprensione della struttura:



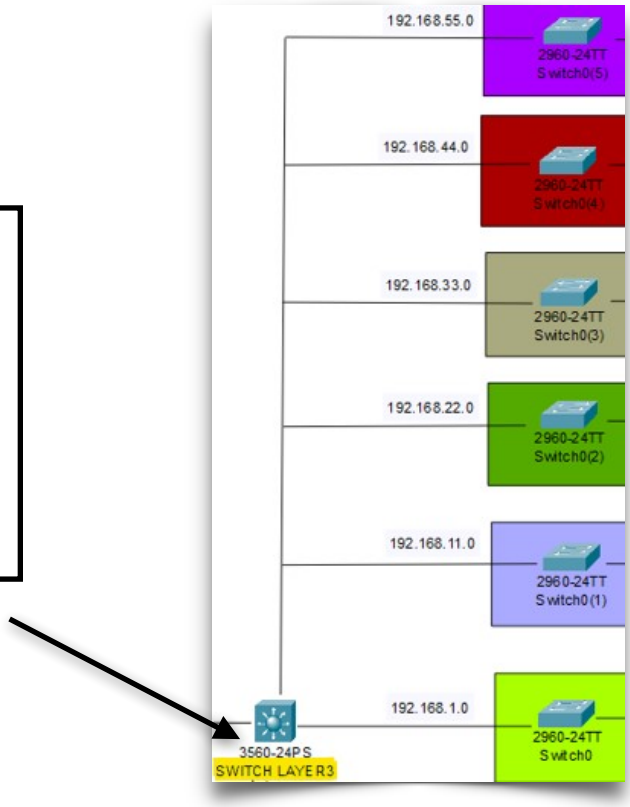
- In particolare abbiamo:

- Collegato i 20 PC di ogni piano ad uno Switch dedicato:

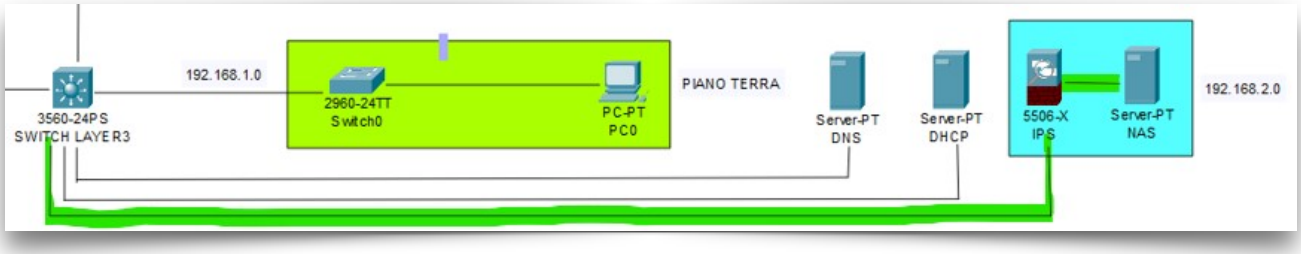


- Successivamente abbiamo collegato ognuno degli Switch dei 6 piani ad uno **Switch (Layer 3)** centrale:

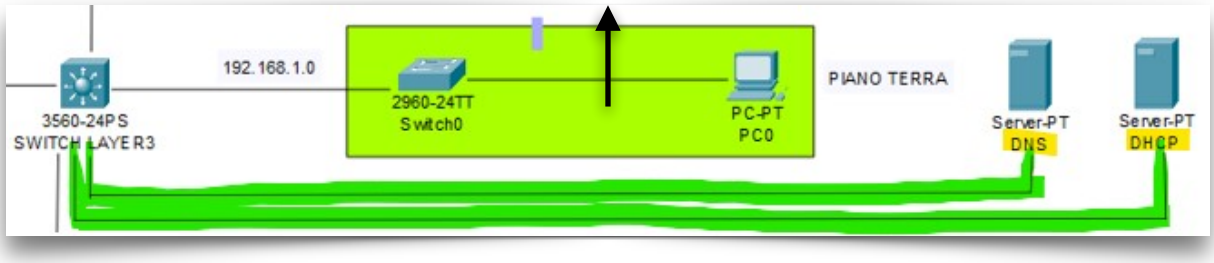
Abbiamo scelto lo **Switch Layer 3** in quanto essendo in grado di eseguire il **routing tra le sottoreti**, ci permette di assegnare al **NAS** (che vedremo subito dopo) una sottorete a sé, aumentando la sicurezza e controllando gli accessi in base ai privilegi di rete che si vogliono impostare.



- Collegato il **NAS** allo Switch centrale e protetto questo collegamento con un **IPS**, che sarà in grado di intervenire in caso di rilevamento di traffico malevolo o accesso non autorizzato:

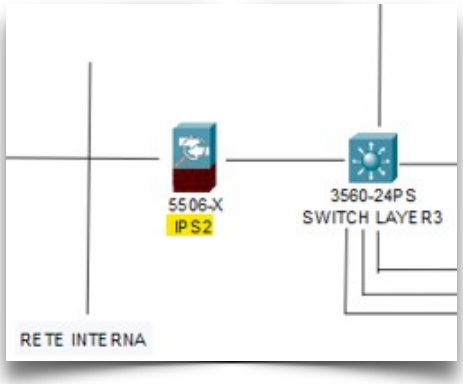


- Aggiunto un **Server DHCP** e un **Server DNS** collegati anche loro allo Switch centrale:



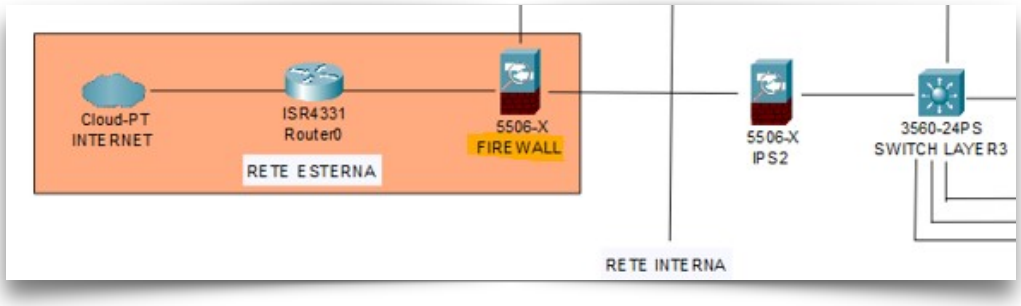
L'esigenza di aggiungere questi 2 server deriva dal fatto che avevamo bisogno di un'assegnazione automatica degli indirizzi IP delle varie sottoreti, da qui il **Server DHCP**. Il **Server DNS** invece è legato all'assegnazione del dominio, ma comunque separato dal Web Server che, come vedremo a breve, si trova nella DMZ.

- Collegato un **IPS** allo Switch centrale, che avrà il compito di separare la rete esterna dalla rete interna appena descritta e soprattutto di creare uno strato di sicurezza operativa prima che si possa raggiungere lo Switch centrale dall'esterno:

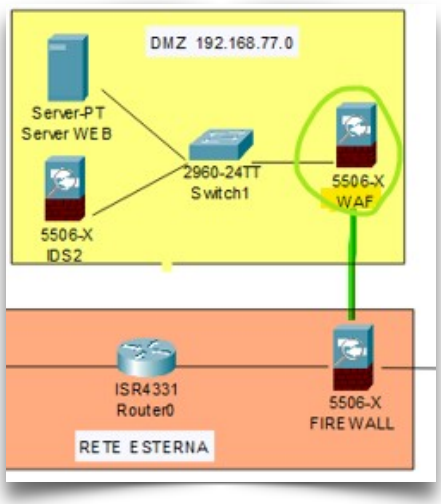


Il ruolo di questo **IPS** è fondamentale in quanto proteggendo lo Switch centrale, protegge anche tutti i dispositivi ad esso collegati.

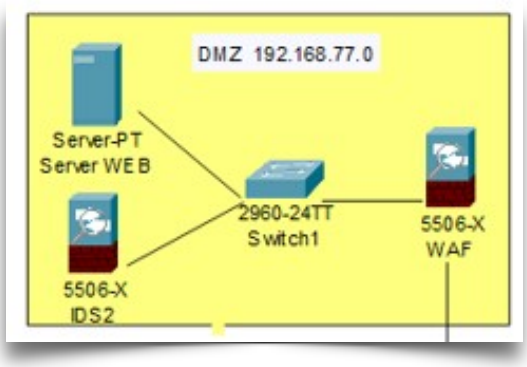
- Per quanto riguarda la rete esterna abbiamo posizionato un **Firewall** tra la rete interna e la connessione ad internet in modo da bloccare tutto il traffico illecito:



- Abbiamo aggiunto un **WAF** (Web Application Firewall) tra il Firewall e la DMZ, nella quale si trova il Server Web, in modo da prevenire attacchi piuttosto pericolosi (XSS, SQL Injection, ecc…):



- Nella **DMZ** abbiamo inserito uno Switch capace di indirizzare i dati verso un **IDS** aggiuntivo, in grado di proteggere il Web Server segnalando all'amministratore eventuali attività sospette in entrata e in uscita dal Server:



• *SUBNETTING*

Una consegna bonus dell'esercizio richiedeva di eseguire il **Subnetting** in modo da scegliere la Subnet più appropriata per la rete.

- Sappiamo che l'azienda dispone di 6 piani e per ogni piano ci sono 20 host, quindi abbiamo bisogno di **6 Subnet** dove ognuna di esse possa disporre di almeno 20 host, più altre **2 Subnet**, una per la DMZ e una per il NAS
- Per fare ciò bisogna scegliere il numero di bit più appropriato per determinare il numero di host all'interno della Subnet, in questo caso **5** poiché:
    - 2^5 - 2 = 32 - 2 = **30 host per sottorete**
- Nel nostro caso ci servirà una rete di classe C (192.160.0.0). Scegliendo **5 bit per gli host**, la maschera di sottorete sarà 255.255.255.224, con 29 bit destinati alla rete e i restanti 5 agli host.
- Dividiamo la rete 192.168.0.0 in 8 Subnet (ogni Subnet avrà 32 indirizzi per gli host di cui 30 utilizzabili).
- Le Subnet saranno:

-192.168.1.0/27: Per il Piano Terra

-192.168.11.0/27: Per il Primo Piano

-192.168.22.0/27: Per il Secondo Piano

-192.168.33.0/27: Per il Terzo Piano

-192.168.44.0/27: Per il Quarto Piano

-192.168.55.0/27: Per il Quinto Piano

-**192.168.77.0/27**: Per la DMZ

-192.168.2.0/27: Per il NAS

- *PREVENTIVO*

Per il preventivo abbiamo scelto:

- **Firewall**: Fortinet Fortigate 100F 22 X GE RJ45 (1.868 €)
  link: https://www.siimsrl.it/router-e-firewall-fortinet-fortigate-100f-22-x-ge-rj45-ports.1.1.137.gp.721227.uw

- **NAS**: QNAP TS-873XU-RP (1.573€) + x3 Seagate IronWolf 12 TB (800€)
  link: https://www.tonitrus.com/it/data-storage/qnap/rack-server/smb-entry-level/10249651-003-qnap-ts-832pxu-rp-4g-ts-832pxu-nas-server-8-bays-rack-mountable-sata-6gb/s-raid-0-1-5-6-10-50-jbod-ram-4-gb-gigabit-ethernet/2.5-gigabit-ethernet/10-gigabit-ethernet-iscsi-support-2u/?number=10249651-003&gad_source=1&gclid=CjwKCAiAyJS7BhBiEiwAyS9uNSV6KAztbM-Qw4jrUfwraJ1JnZUIwheZXG1nuao6TOTouJIX_OrabxoCb-8QAvD_BwE.
  link Hard Disk: https://www.amazon.it/Seagate-St12000Vnz008-IronWolf-Interno-Argento/dp/B08147LZFD/ref=sr_1_2?adgrpid=123721510726&dib=eyJ2IjoiMSJ9.ZgAqZ28lhzDw2B-sqLTISJgAd_I6Akvif7YjWHEKxLP5z7EEgZdt6Cq-7MCJnbDx-JvuLeGiv3tjox50wgzb9IXruqudp83ATA26_vcOMZd8rFqvVkoRmKn4iGDd9-LHSBc7X4l1VjCCTUk_ABFp5b5nZCrZE0VU0w3k16p1789zipgUUVXw4egZWC_Rq5gUxFR7euDkmdOTu20vZjJINJmGzsVjy19bwBhFIorc2CetVD9K8EOjgXffychJcLQqlyNyc0w4GKcN2ktDtYyGH12ciz2egToQfgdcVxCzlbsLvrbpuj7tjE4ra4BrJN7yYqmJZwtl9GJrG-aYkJO4aW0uYa5e46fOYIGZXnQ2OTDhst0cojnn5d3FCoAvpju_Aw0pSSKWEvAv4pm71W1-8V5GsCfsOxXj6hm6R9mk1eLjgno2RYa8Nq8WiK0D1Wb5.N7AS5q6EyrbT4-uc_ThzPDZ9mRAFkoEQ5gx8_gH4LG8&dib_tag=se&hvadid=591546549276&hvdev=c&hvlocphy=1008736&hvnetw=g&hvqmt=e&hvrand=13391994289586884956&hvtargid=kwd-381797929649&hydadcr=15886_2272061&keywords=seagate%2Bironwolf%2B12%2Btb&nsdOptOutParam=true&qid=1734692606&sr=8-2&ufe=app_do%3Aamzn1.fos.9d4f9b77-768c-4a4e-94ad-33674c20ab35&th=1

- **Server WEB**: Lenovo ThinkSystem SR530 server Rack (1U) (1.979€)
  link: https://computer.milano.it/lenovo-thinksystem-sr530-1xintel-xeon-silver-4208-8c-21ghz-85w-1x32gb-2rx4-thinksystem-m2-with-mirroring-enablement-kit-1x750w-xcc-enterprise-thinksystem-toolless-slide-rail-7x08a0cbea.html
- **IDS-IPS**: Cisco ASA5512-IPS-K8 x3 (1.275€)
  link: https://it-planet.com/it/p/cisco-asa5512-ips-k8-211452.html
- **Switch**: Cisco SF350-24P-K9 259 euro –x7-tot (1.813€)
  link: https://it-planet.com/it/p/cisco-sf350-24p-k9-eu-19382.html?c=0
- **Switch Layer3**: Cisco WS-C2960XR-24TS-I (1.850€)
  link: https://it-planet.com/it/p/cisco-ws-c2960xr-24ts-i-14865.html?number=2866861000&gad_source=1&gclid=CjwKCAiAyJS7BhBiEiwAyS9uNfgvcm0U0AoFjTTydufDoyFUVQ1N6zLwF8w7cEOoRBy8gaWCbuiu8hoCv1QQAvD_BwE
- **Server DHCP e DNS**: HPE ProLiant MicroServer Gen10 (1.236x2= 2.472€)
  link: https://www.senetic.it/product/P70335-425?gad_source=1&gclid=CjwKCAiApY-7BhBjEiwAQMrrEZRRFZ_Cww0nuQqSJ0caljAtH0qmpvJ9F7VE93yu_ur_lMKpmYWwVhoCSa4QAvD_BwE
- **Router**: Cisco RV340-K9-G5 (1.995€)
  link: https://it-planet.com/it/p/cisco-rv340-k9-g5-16256.html?number=5304907000

TOTALE PREVENTIVO = 15.626€

• *TESTING DELLA RETE*

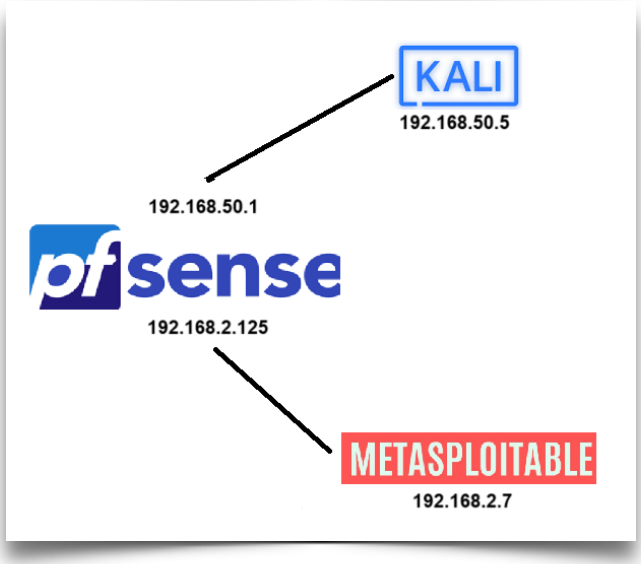
- Per poter completare i collegamenti di rete e fare diverse richieste abbiamo avuto la necessità di integrare parti di codice con **programmi in Python**, questa fase è stata necessaria per alcuni tipi di configurazione, per effettuare richieste di informazioni o inserire dati all'interno della rete tra macchine collegate tra loro.
  Nello specifico **Kali è il Client** e **Metasploitable è il Server**.

- Suddividiamo il processo in passaggi:

  1. **Configurazione del sistema**
  2. **Scansione delle porte**
  3. **Considerazioni di sicurezza sulle porte**
  4. **Richiesta verbi ammessi**
  5. **Test dei verbi**
  6. **Socket di rete (Bonus)**

1. Configurazione del sistema

- Abbiamo iniziato con la configurazione del sistema con **Pfsense** per far si che **Kali** e **Metasploitable** potessero comunicare tra loro:



- Le configurazioni di rete per le macchine sono:

Pfsense → WAN 10.0.2.15/24
LAN 192.168.50.1/24
OPT1 192.168.2.125/24

Kali → LAN 192.168.50.5/24

Metasploitable → LAN 192.168.2.7/24

- Il primo ostacolo sul quale ci siamo soffermati è stato dover accedere con user e password nella piattaforma phpMyAdmin nel browser.
  Per la risoluzione abbiamo avuto la necessità di entrare nel database per modificare la Password di accesso.
  Abbiamo inoltre garantito all'utente root tutti i privilegi:

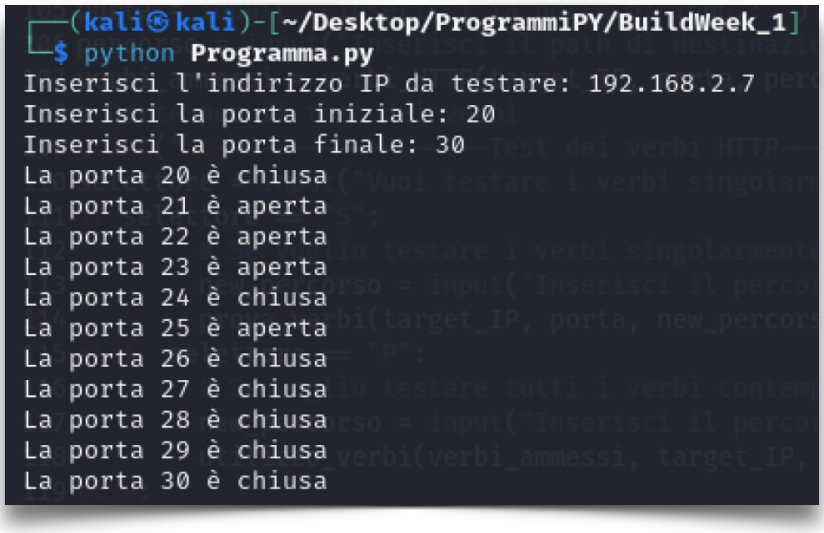
```
mysql> UPDATE user SET password=PASSWORD('root') WHERE user='root' AND host='%';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT user, host, password FROM user WHERE user='root';
+------+------+-------------------------------------------+
| user | host | password                                  |
+------+------+-------------------------------------------+
| root | %    | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
+------+------+-------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

```
msfadmin@metasploitable:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> SELECT user, host, password FROM mysql.user WHERE user = 'root';
+------+------+----------+
| user | host | password |
+------+------+----------+
| root | %    |          |
+------+------+----------+
1 row in set (0.00 sec)
```

2. Scansione delle porte

- Successivamente siamo passati alla fase relativa allo Scan delle porte, che consisteva appunto nello **scansionare le porte di Metasploitable2**, per avere un riscontro su quali porte fossero aperte e quali chiuse.
  Lo Scan è stato effettuato in un range prestabilito di porte.

- Abbiamo effettuato lo scan utilizzando un programma Python sulle porte più comuni utilizzate:



- Tra le porte principali abbiamo avuto i seguenti riscontri:

Numero Porta	Tipo Porta	Stato
20	FTP	Chiusa
21	FTP	Aperta
22	SSH	Aperta
23	TELNET	Aperta
25	SMTP	Aperta
53	DNS	Aperta
67 - 68	DHCP	Chiuse
80	HTTP	Aperta
110	POP	Chiusa
111	PORT MAPPER	Aperta
137-138	NETBIOS	Chiusa
139	NETBIOS	Aperta
143	IMAP	Chiusa
161-162	SNMP	Chiuse
443	HTTPS	Chiusa
445	SAMBA	Aperta
989 - 990	FTPS	Chiuse

3. Considerazioni di sicurezza sulle porte

- Nello specifico segnaliamo le seguenti porte aperte:

21 - 22 - 23 - 25 - 53 - 80 - 111 - 139 - 445

- **Porta 21 (FTP, File Transfer Protocol)**: È una porta che ospita un server FTP, non è sicura dato che spesso questi server usano l'autenticazione anonima e non possiedono crittografia.

- **Porta 22 (SSH, Secure Shell)**: È una porta spesso associata ad un account con password poco sicure, per questo sono soggette a molte vulnerabilità ed è necessario distribuire patch di sicurezza costantemente.

- **Porta 23 (Telnet)**: È una porta collegata ad un servizio di comunicazione ormai arretrato che invia dati senza crittografia, per questo è esposto a molte minacce.

- **Porta 25 (SMTP, Simple Mail Transfer Protocol)**: Serve al proprio Client di porta per il trasferimento di messaggi su Internet, i contro sono la mancanza di crittografia e di contrasto allo spam, proprio per questo spesso viene bloccata dagli ISP.

- **Porta 53 (DNS, Domain Name Service)**: È la porta assegnata al DNS per la conversione di nomi in indirizzi IP leggibili dalla macchina, il problema è che viene raramente monitorata, infatti viene spesso attaccata per l'esfiltrazione di dati. Un modo per renderla più sicura è il monitoraggio e il filtraggio del traffico DNS.

- **Porta 80 (HTTP)**: Una delle porte più utilizzate perché serve per stabilire connessioni HTTP in modo da poter scambiare dati tra un browser e un server, il problema principale è che non essendo sicura viene spesso utilizzata per effettuare attacchi DDos, il consiglio è bloccarla ed utilizzare solo la porta 443 (HTTPS) che invece utilizza crittografia.

- **Porta 111 (RPC, Remote Procedure Call)**: Serve per collegare 2 computer non facenti parte della stessa rete, ad esempio usa il NFS (Network File System) che permette ad un computer di accedere ai dati di un altro computer come se fossero entrambi nella stessa rete locale. Presenta diverse vulnerabilità e un modo per contrastarle è la creazione di una regola nel Firewall che permetta l'utilizzo di Portmapper solo all'interno della tua rete o locale o tramite specifici indirizzi IP.

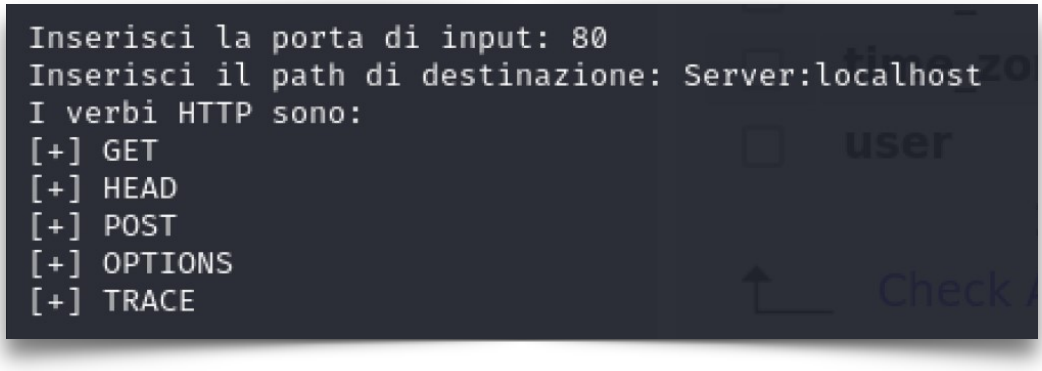
- **Porta 139 (SMB, Server Message Block)**: Permette alle applicazioni su diversi computer di comunicare all'interno di una rete locale; è stata sviluppata negli anni 80, periodo in cui la sicurezza non era di certo la priorità, per proteggersi bisognerebbe creare una regola all'interno del Firewall che blocchi qualsiasi connessione in entrata da fonti non autorizzate.

- **Porta 445 (SMB, Server Message Block)**: Nuove versioni di SMB che non utilizzano più la porta 139 ma la 445, per proteggerla si seguono le stesse istruzioni descritte per la porta 139.

4. *Richiesta dei verbi ammessi*

- Per questa fase abbiamo creato un programma in Phyton ed effettuato una richiesta **OPTIONS** al server phpMyAdmin tramite porta **80** (HTTP).

- Abbiamo ricevuto come risultati i seguenti verbi ammessi:



- **GET**: Viene utilizzato per richiedere una risorsa (come una pagina web, un'immagine, un file, ecc.) dal server.
  Non modifica nulla all'interno del server, ma si occupa solo della visualizzazione o del recupero della risorsa richiesta.
  Esempio: Se visiti una pagina web, il tuo browser invia una richiesta GET per ottenere la pagina.

- **HEAD**: Funziona in modo simile a GET, ma non restituisce il corpo della risposta. Viene utilizzato per ottenere solo le intestazioni (Headers).
  Esempio: Un metodo di utilizzo è quello di verificare se una pagina esiste prima di scaricarla completamente.

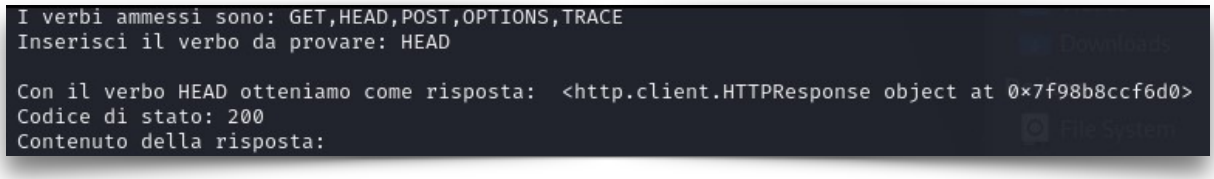
- **POST**: Viene utilizzato per inviare dati al server, tipicamente per creare o aggiornare una risorsa. I dati vengono inviati nel corpo della richiesta.
  Esempio: Quando compili un modulo su una pagina web (come il login), i dati vengono inviati al server tramite una richiesta POST.

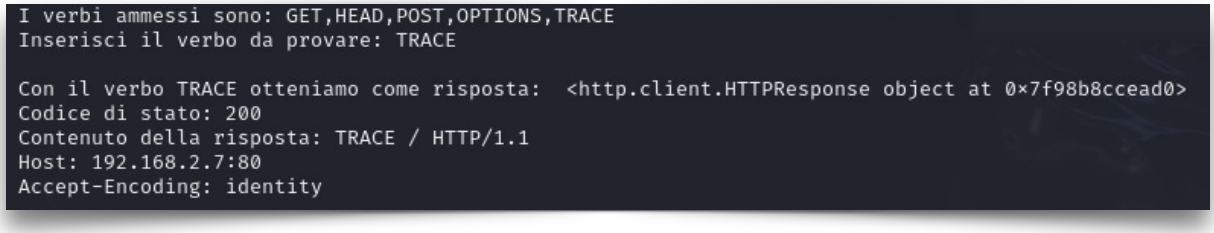
- **OPTIONS**: Viene utilizzato per ottenere i verbi HTTP disponibili per una risorsa o un server e riceve come risposta i verbi supportati (GET, POST, PUT, DELETE…).
  Esempio: Quando un'applicazione web fa una richiesta OPTIONS è per verificare se può comunicare e con quali verbi può farlo.

- **TRACE**: Utilizzato per tracciare il percorso che una richiesta HTTP fa attraverso i vari Server.
  Esempio: In alcune situazioni di debugging, TRACE può essere usato per vedere come una richiesta arriva al server.

5. Test dei verbi

- Per eseguire il test abbiamo creato un programma in Python che permette di scegliere tra test singolo, inserendo il verbo specifico a cui vogliamo effettuare la richiesta, oppure il test di tutti i verbi disponibili in sequenza.
  Ecco alcuni esempi:





- Nella richiesta ci soffermiamo su un primo codice di risposta, il **codice di stato = 200**, che ci assicura la corretta elaborazione della richiesta, la quale verrà esposta nelle righe successive.

6. *Socket di rete (Bonus)*

- È stato sviluppato un altro programma su Metasploitable che si occupa di fare il pocket di rete sulla porta 80.

↓

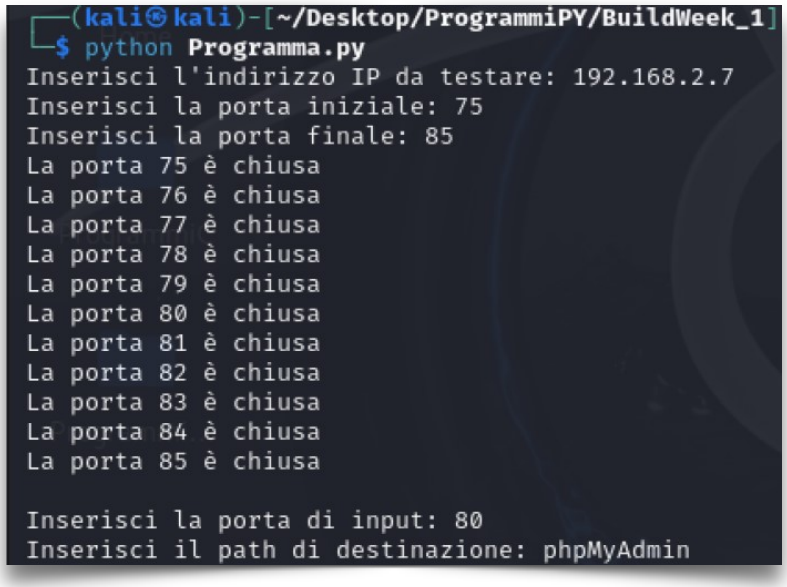
- Per poter attivare il socket abbiamo avuto la necessità di spegnere il servizio apache2 per liberare la porta 80.

↓

- Quindi abbiamo fatto partire i 3 programmi in contemporanea nelle due macchine Client e Server, in Kali lo Scan e la richiesta dei verbi, in Metasploitable il socket.

↓

- Da kali visualizziamo questo risultato:



↓

- Questo invece l’output di Metasploitable:

- Possiamo notare la prima riga di comando dove spegniamo apache2, successivamente abbiamo avviato il programma Python socket di rete e gli abbiamo assegnato la porta 80.

- Quando si avvia il socket visualizziamo Server Started, Waiting for connections, a questo punto da Kali iniziamo ad avviare il programma che effettua lo Scan delle porte e otteniamo su Metasploitable la riga Client connected con l’IP di Kali 192.168.50.5 con porta 44774. Effettuato lo scan correttamente si disconnette e riavviamo il socket di rete, poi continuiamo con il programma in Kali con la richiesta OPTIONS per ricevere i verbi ammessi.

- Visualizziamo sempre client connection di Kali, Kali invia OPTIONS che è proprio il verbo che vogliamo inviare a Metasploitable.

- Dall’output di Metasploitable vediamo la voce OPTIONS.

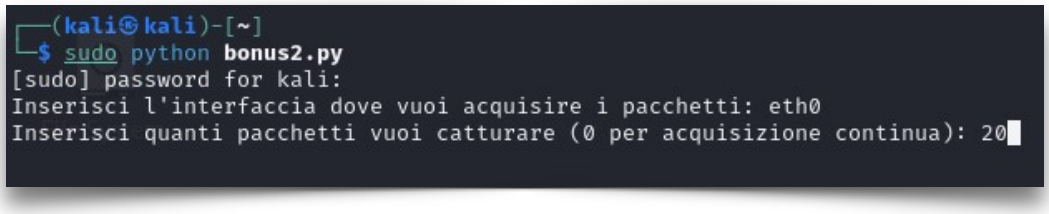
- A questo punto dato che abbiamo dovuto spegnere apache2, Kali non otterrà risposta.

ESERCIZIO BONUS 2

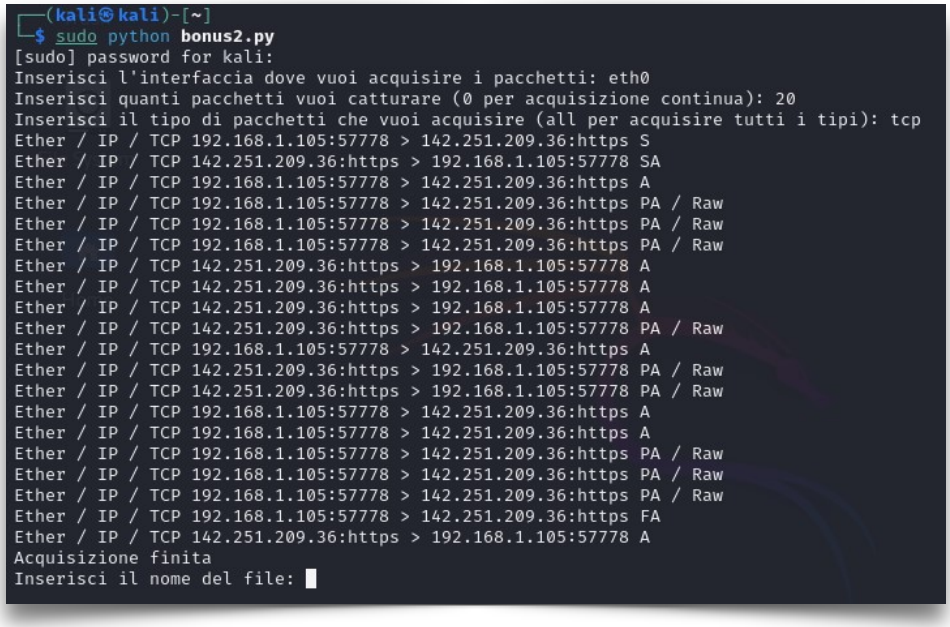
- L’ultimo esercizio bonus richiedeva un programma in Python che utilizzasse **Scapy** per catturare e analizzare il traffico di rete.

- Avvio il programma con **sudo**, dato che Scapy richiede i privilegi di amministratore.

- Inserisco l’interfaccia **eth0** e inserisco 20 per catturare **20 pacchetti**:

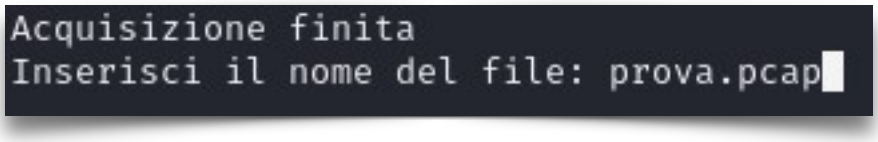


- Filtro i pacchetti in base al protocollo **tcp** e cercando qualsiasi cosa sul web ottengo i 20 pacchetti:



- Come si vede in figura ottengo informazioni come **IP sorgente**, **IP destinazione**, **protocollo** e **porte** utilizzate.

- A questo punto salvo le informazioni nel file **prova.pcap** in modo da poterle analizzare successivamente.



- Per accedere al file dal terminale faccio **sudo scapy** e seleziono il file di interesse ovvero **prova.pcap**: