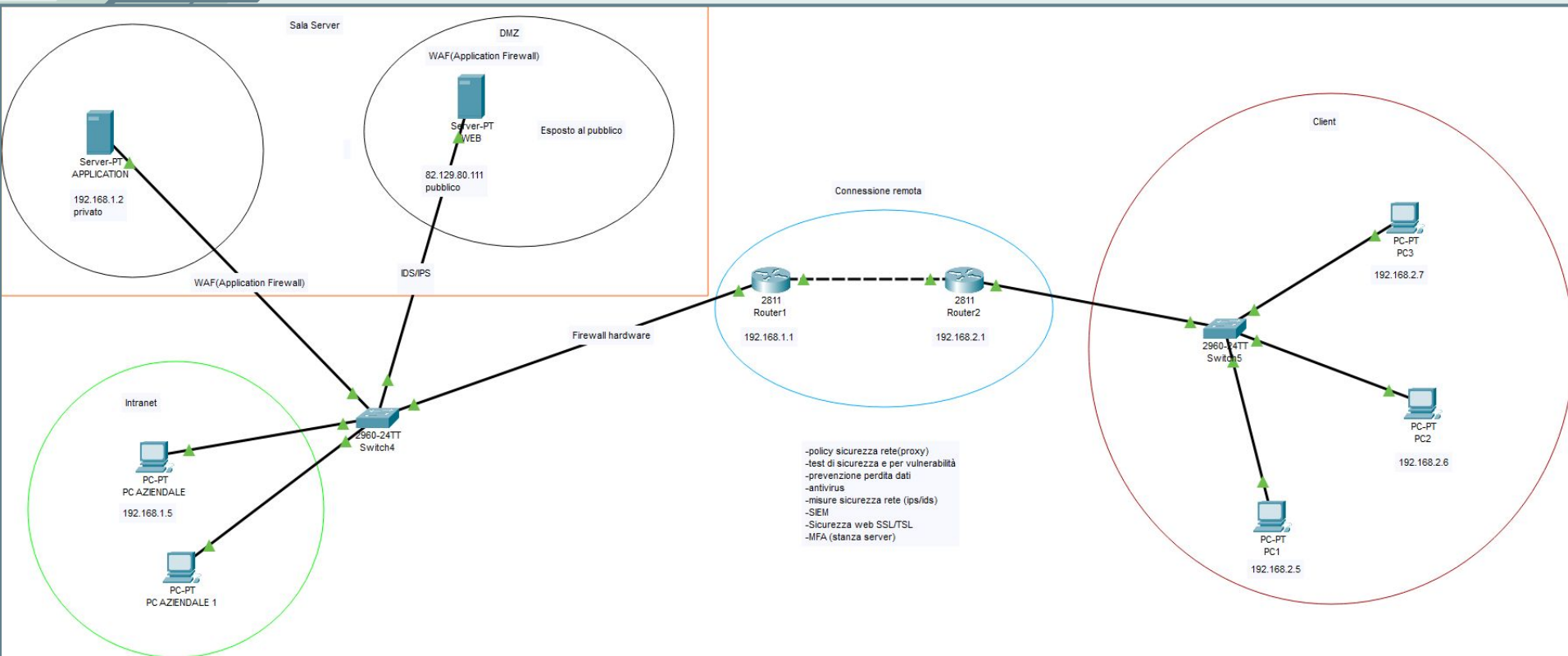




# DESIGN DI RETE

Compagnia Theta



# NETWORK SECURITY

Design di rete suggerito:

1. **Rafforzare sicurezza Web** (tecniche codifica sicure, protocolli SSL/TSL, scansione vulnerabilità applicazioni Web e test di penetrazione)
2. **Security Policies** (regole accesso ed architettura rete)
3. **Test sicurezza delle applicazioni** (identificare e mitigare difetti di codice)
4. **Gestione delle vulnerabilità** (identificazione, definizione, correzione e segnalazione vulnerabilità)
5. **Network Penetration Testing** (misurare e valutare la sicurezza dell'infrastruttura)
6. **Prevenzione della perdita di dati** (rileva e previene potenziali violazioni dati)
7. **Software antivirus** (previene, scansiona, rileva ed elimina virus)
8. **Misure di sicurezza IDS/IPS** (rilevamento e prevenzione delle intrusioni)
9. **Soluzione SIEM** (rilevamento e gestione incidenti)
10. **Autenticazione a più fattori** (MFA)

# TEST WEB SERVER



# SCANSIONE SERVIZI ATTIVI

```
kali@kali: ~/Desktop/prove
File Actions Edit View Help

(kali@kali)~[~/Desktop/prove]
$ sudo nano PORT_scanner.py

(kali@kali)~[~/Desktop/prove]
$ python3 PORT_scanner.py
Enter the IP address to scan: 192.168.32.101
Enter the port range to scan (e.g., 5-200): 1-9999
Scanning host 192.168.32.101 from port 1 to port 9999
*** Port 21 - OPEN ***
*** Port 22 - OPEN ***
*** Port 23 - OPEN ***
*** Port 25 - OPEN ***
*** Port 53 - OPEN ***
*** Port 80 - OPEN ***
*** Port 111 - OPEN ***
*** Port 139 - OPEN ***
*** Port 445 - OPEN ***
*** Port 512 - OPEN ***
*** Port 513 - OPEN ***
*** Port 514 - OPEN ***
*** Port 1099 - OPEN ***
*** Port 1524 - OPEN ***
*** Port 2049 - OPEN ***
*** Port 2121 - OPEN ***
*** Port 3306 - OPEN ***
*** Port 3632 - OPEN ***
*** Port 5432 - OPEN ***
*** Port 5900 - OPEN ***
*** Port 6000 - OPEN ***
*** Port 6667 - OPEN ***
*** Port 6697 - OPEN ***
*** Port 8009 - OPEN ***
*** Port 8180 - OPEN ***
*** Port 8787 - OPEN ***

(kali@kali)~[~/Desktop/prove]
$
```

```
kali@kali: ~/Desktop/prove
File Actions Edit View Help

GNU nano 7.2 PORT_scanner.py
import socket

target = input('Enter the IP address to scan: ')
portrange = input('Enter the port range to scan (e.g., 5-200): ')

lowport = int(portrange.split('-')[0])
highport = int(portrange.split('-')[1])

print('Scanning host', target, 'from port', lowport, 'to port', highport)

for port in range(lowport, highport + 1):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    status = s.connect_ex((target, port))
    if status == 0:
        print('*** Port', port, '- OPEN ***')
    #else:
        #print('Port', port, '- CLOSED')
    s.close()

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^_ Justify  ^_ Go To Line
```

# ENUMERAZIONE METODI HTTP

kali@kali: ~/Desktop/prove

File Actions Edit View Help

(kali@kali)-[~/Desktop/prove]

\$ python3 HTTP\_req.py

Inserisci l'URL di destinazione http:// o digita 'x' per uscire: http://192.168.32.101

Il metodo HTTP 'GET' è consentito

Il metodo HTTP 'POST' è consentito

Il metodo HTTP 'PUT' è consentito

Il metodo HTTP 'DELETE' è consentito

Il metodo HTTP 'OPTIONS' è consentito

Il metodo HTTP 'HEAD' è consentito

Il metodo HTTP 'TRACE' è consentito

Il metodo HTTP 'PATCH' è consentito

(kali@kali)-[~/Desktop/prove]

\$

kali@kali: ~/Desktop/prove

File Actions Edit View Help

GNU nano 7.2

HTTP\_req.py

```
import requests #Libreria contenente le richieste HTTP
import sys #Questa libreria fornisce accesso a funzionalità del sistema operativo
#enumera i metodi HTTP, parametro è indirizzo di destinazione
def enumerare_metodi_http(url_destinazione):
    metodi = ['GET', 'POST', 'PUT', 'DELETE', 'OPTIONS', 'HEAD', 'TRACE', 'PATCH'] #metodi HTTP

    for metodo in metodi:
        try:
            #per ogni metodo controlla il codice di stato
            risposta = requests.request(metodo, url_destinazione)
            #Se il codice di stato è 200
            if risposta.status_code == 200: #richiesta e metodo consentito
                print(f"Il metodo HTTP '{metodo}' è consentito")
            #Se il codice di stato è 405
            elif risposta.status_code == 405: #richiesta e metodo non consentito
                print(f"Il metodo HTTP '{metodo}' non è consentito")
            #Se il codice non è 202 e 405
            else: #stampa il metodo ed il suo codice di stato
                print(f"Il metodo HTTP '{metodo}' - Codice di stato: {risposta.status_code}")
            #Se viene colta un'eccezione durante la richiesta
        except requests.exceptions.RequestException as e:
            print(f"Si è verificato un errore durante l'invio della richiesta con il metodo '{metodo}': {str(e)}")

while 1:
    #Chiede all'utente di inserire l'URL
    destinazione = input("Inserisci l'URL di destinazione http:// o digita 'x' per uscire: ")
    #Possiamo digitare x o X per uscire dal programma
    if destinazione.lower() == 'x':
        #Funzione della libreria sys, precisamente l'uscita dal programma
        sys.exit(0)
    #Se URL inizia con http://
    elif destinazione.startswith('http://'):
        #funzione enumerare_metodi_http viene chiamata
        enumerare_metodi_http(destinazione)
        #Funzione della libreria sys, precisamente l'uscita dal programma
        sys.exit(0)
    else:
        #Output se digitiamo altro
        print("L'URL di destinazione deve iniziare con 'http://'")
```

⌘ Help ⌘ Write Out ⌘ Where Is ⌘ Cut ⌘ Execute ⌘ Location ⌘ Undo  
⌘ Exit ⌘ Read File ⌘ Replace ⌘ Paste ⌘ Justify ⌘ Go To Line ⌘ Redo

