

# PROGETTAZIONE RETE SICURA COMPAGNIA THETA

## Relazione Tecnica

---

Il progetto di rete per la compagnia Theta prevede la progettazione di un'infrastruttura IT scalabile e sicura per un edificio di 6 piani, con 120 computer totali (20 per piano), integrando componenti critici come un web server DVWA su Metasploitable, un firewall perimetrale, un NAS e 3 IDS/IPS per la protezione avanzata.

## Requisiti e Specifiche

L'edificio richiede una struttura gerarchica: switch dedicati per piano collegano i 20 PC ciascuno, interconnessi a uno switch principale anch'esso collegato a un router centrale per il traffico interno. Il NAS si posiziona allo switch del piano terra per storage condiviso accessibile da tutti i dispositivi, mentre i 3 IDS/IPS monitorano il perimetro interno ed esterno rilevando e prevenendo intrusioni in tempo reale. Il web server DVWA, simulato su Metasploitable, opera come DMZ tra firewall e Internet, con opzione per un secondo firewall per maggiore segmentazione.

## Componenti Infrastrutturali

Componente	Quantità	Posizione	Funzione Principale
Switch	6 + 1	Uno per piano + switch principale	Connessione PC locali
Router	1	Centrale	Routing inter-piani
Firewall	1	Perimetrale/DMZ	Blocco minacce esterne
NAS	1	Piano terra	Archiviazione dati
IDS/IPS	3	Perimetro interno ed esterno	Detection/prevenzione
Web Server DVWA	1	DMZ	Servizi web esposti

## Architettura di Rete

La rete interna centralizza il router per unificare gli switch, con il firewall a isolare l'accesso Internet e il NAS ottimizzato per bassa latenza. Esternamente, la DMZ protegge il web server esposto, garantendo comunicazioni sicure senza esposizione diretta della LAN. Questa topologia riduce i punti di failure e migliora la resilienza, con IDS/IPS distribuiti per analisi profonda del traffico.

# Test di Verifica

I test usano script Python: uno verifica verbi HTTP (GET, POST, PUT, DELETE) sul web server, documentando risposte su path come phpMyAdmin di Metasploitable (porta 80), richiedendo prima ping da Kali. Un secondo scanner porte accetta IP e range, elencando stati aperti/chiusi senza tool esterni, essenziale per validare esposizioni. Il report finale include risultati dettagliati, raccomandazioni (es. chiusura porte non necessarie) e conferme di connettività.

Di seguito il codice python per la scansione delle porte:

```
1 import socket
2
3 target = input("IP da scansionare: ")
4 portrange = input("Range di porte (es. 20-100): ")
5
6 lowport = int(portrange.split('-')[0]) # ['20', '100'] -> 20
7 highport = int(portrange.split('-')[1]) # ['20', '100'] -> 100
8
9 print(f"Scannerizzando l'IP {target} da porta {lowport} a porta {highport}")
10
11 for port in range(lowport, highport + 1):
12     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
13     status = s.connect_ex((target, port))
14     if status == 0:
15         print(f"Porta {port} : aperta")
16     else:
17         print(f"Porta {port} : chiusa")
18     s.close()
```

Di seguito il codice python per catturare il socket di rete:

```
1 import socket
2 import sys #Permette la comunicazione tra programma e sistema operativo
3
4 # --- CONFIGURAZIONE SERVER ---
5 # '0.0.0.0' ascolta su tutte le interfacce disponibili (necessario se ti connetti da un altro PC)
6 # Se vuoi ascoltare solo su un IP specifico, usa ad esempio: SRV_ADDR = "192.168.1.34"
7 SRV_ADDR = input("Inserisci l'IP a cui collegarti: ")
8 SRV_PORT = int(input("Inserisci la porta: "))
9 BUFFER_SIZE = 1024 # Dimensione standard per la ricezione dei dati (BIT)
10
11 # Creazione del socket (TCP/IPv4)
12 try:
13     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
14     print("Socket creato con successo.")
15 except socket.error as err:
16     print(f"Errore nella creazione del socket: {err}")
17     sys.exit()
18
19 # 1. Assegnazione di IP e porta (Bind)
20 try:
21     s.bind((SRV_ADDR, SRV_PORT))
22     print(f"Socket legato a {SRV_ADDR}:{SRV_PORT}")
23 except socket.error as err:
24     print(f"Bind fallito. Errore: {err}")
25     sys.exit()
26
27 # 2. Metti il socket in ascolto (Listen)
28 # 5 Ã" il numero di connessioni client che possono essere in coda.
29 s.listen(5)
30 print("Socket in ascolto...")
31
32 # --- LOOP PRINCIPALE (Accetta Connessioni Multiple) ---
33 # Questo loop permette al server di accettare nuovi client dopo che i precedenti si sono disconnessi.
34 while True:
35     connection = None #default
36     try:
37         # 3. Accettazione della connessione (Accept)
38         # Il programma si blocca qui in attesa di un client
39         connection, address = s.accept()
40         print(f"\n[!] Connessione accettata da: {address[0]}:{address[1]}")
```

```

41     # --- GESTIONE DEL CLIENT SINGOLO (LOOP INTERNO) ---
42     while True:
43         # 4. Ricezione dei dati
44         data = connection.recv(BUFFER_SIZE)
45
46         # Controllo fondamentale: se recv() restituisce un oggetto vuoto (b''),
47         # significa che il client ha chiuso la connessione.
48         if not data:
49             print(f"[-] Client {address[0]} disconnesso.")
50             break # Esci dal loop interno
51
52         # Decodifica e stampa del messaggio
53         try:
54             message = data.decode('utf-8').strip()
55             print(f"  << Ricevuto: {message}")
56         except UnicodeDecodeError:
57             print("  << Dati ricevuti non decodificabili in UTF-8.")
58
59         # 5. Risposta al client
60         # Invia una conferma di ricezione.
61         response = b"SERVER: Messaggio ricevuto.\n"
62         connection.sendall(response)
63         print(f"  >> Invia: {response.decode().strip()}")
64
65     except KeyboardInterrupt:
66         # Gestisce la chiusura con Ctrl+C
67         print("\n\n[-] Server terminato da utente (Ctrl+C).")
68         break
69
70     except Exception as e:
71         print(f"\n[!] Errore critico durante la comunicazione: {e}")
72
73     finally:
74         # 6. Chiusura della connessione (Cruciale)
75         if connection:
76             connection.close()
77
78     # Chiusura finale del socket del server
79     s.close()
80
81     print("Socket principale chiuso. Programma terminato.")

```

Di seguito il codice python per la verifica dei verbi HTTP:

```

1  import http.client
2
3  def test_metodi(host, port, path):
4      methods = ["OPTIONS", "GET", "POST", "PUT", "DELETE", "HEAD", "PATCH"]
5      results = {}
6      conn = None
7
8      try:
9          conn = http.client.HTTPConnection(host, port, timeout=5)
10
11         for method in methods:
12             conn.request(method, path)
13             response = conn.getresponse()
14             response.read() # svuota il body
15             results[method] = (response.status, response.reason)
16
17     except ConnectionRefusedError:
18         print("Connessione rifiutata dal server")
19
20     except Exception as e:
21         print("Errore:", e)
22
23     finally:
24         if conn:
25             conn.close()
26
27     return results
28
29
30
31 host = input("Inserire host/IP del sistema target: ")
32 port = input("Inserire la porta del sistema target (default 80): ")
33 path = input("inserisci il root: ")
34
35 port = 80 if port == "" else int(port)
36 path = "/" if path == "" else str(path)
37
38 results = test_metodi(host, port, path)
39
40 print("\nRisultati verifica verbi HTTP:")
41 for method, (status, reason) in results.items():
42     print(f"{method}: {status} {reason}")

```

## Aspetti Avanzati e Bonus

Per il subnetting bonus, si calcola una rete /24 (es. 192.168.1.0/24) per 120 host, dividendola in subnet per piani (es. /27 per 30 host/piano) ottimizzando IP e broadcast. Un packet sniffer Python cattura socket di rete per analisi forense aggiuntiva. Il preventivo di spesa (non dettagliato nel brief) copre hardware enterprise-level, priorizzando ROI (return on investment) su sicurezza e scalabilità per Theta.

---

## CATALOGO PREZZI – INFRASTRUTTURA IT AZIENDALE

Scenario: Azienda su 6 piani, 120 PC (20 per piano). Il catalogo include Offerta A (Intermedia) e Offerta B (Avanzata).

### OFFERTA A – INTERMEDIA (PC i5 / 16GB / SSD 512GB)

Voce	Q.tà	Prezzo unit. (IVA incl.)	Totale (IVA incl.)
PC Intermedio x120 (i5, 16GB RAM, 512gb)	120	700	84.000
Sistema Operativo	120	180	21.600
<b>Totale Offerta A</b>			<b>105.600</b>

### OFFERTA B – AVANZATA (PC i7 / 32GB / SSD 1TB)

Voce	Q.tà	Prezzo unit. (IVA incl.)	Totale (IVA incl.)
PC Avanzato x120 (i7, 32GB RAM, 1TB)	120	1.100	132.000
Sistema Operativo	120	180	21.600
<b>Totale Offerta B</b>			<b>153.000</b>

## NETWORKING

Voce	Q.tà	Prezzo unit. (IVA incl.)	Totale (IVA incl.)
Router aziendale - Cisco ISR4431 SEC/K9	1	3.699	3699
Switch Layer 3 (core) - S2200-24T4X L3 Managed Switch	1	623	623
Switch Layer 2 per piano- IES5100-24TF L2+ Managed Switch	6	718	4308
<b>Totale sezione</b>			<b>8630</b>

## CABLAGGIO STRUTTURATO

Voce	Q.tà	Prezzo unit. (IVA incl.)	Totale (IVA incl.)
Punti presa dati – prezzo unitario medio	264	95	25080
<b>Totale sezione</b>			<b>25080</b>

## SICUREZZA

Voce	Q.tà	Prezzo unit. (IVA incl.)	Totale (IVA incl.)
Firewall perimetrale NGFW	1	1090	1090
IDS/IPS	2	360	1080
Licenze sicurezza (1 anno)		1190	1190
<b>Totale sezione</b>			<b>3360</b>

## SERVER & STORAGE

Voce	Q.tà	Prezzo unit. (IVA incl.)	Totale (IVA incl.)
Web server aziendale (tower server entry – Dell PowerEdge T150)	1	1416	1416
NAS 4-bay (Synology DS923+ – senza dischi)	1	881	881
Software backup (AOMEI Backupper Server – indicazione prezzo)	1	90	90
<b>Totale sezione</b>			<b>2387</b>

## INSTALLAZIONE E CONFIGURAZIONE

Attività	Ruolo	Giorni	RAL ref (€/anno)	Costo azienda/giorno	Costo interno	Tariffa vendita/giorno
Progettazione rete + subnetting + VLAN	Network Engineer	3	32900	202	606	650
Configurazione switch/router + messa in servizio	Network Engineer	3	32900	202	606	650
Firewall + regole + DMZ	Security Engineer	2	45900	282	563	750
IDS/IPS (install + tuning base)	Network Security Engineer	2	34200	210	420	700
Server web + NAS + permessi/backup	Sistemista	2	30000	184	368	600
Test (HTTP verbs + port scanner custom) + report	Security Engineer	2	45900	282	563	750
Project management / coordinamento fornitori	IT Project Manager	2	44250	272	543	700
<b>Totale</b>					<b>3669</b>	<b>10900</b>

## RIEPILOGO FINALI

Offerta	Totale (IVA incl.)
Offerta A – Intermedia	<b>155.957</b>
Offerta B – Avanzata	<b>203.357</b>

**19/12/25 - Team Samurai**

*(Vincenzo Di Salvo, Vincenzo Zarola, Mirko Imbrogno, Victor Rosati, Gerald Bejte, Daniele Di Martino)*

