



TRUSTVAULT

OTTOBRE 2024

IMPLEMENTAZIONE RETE IT E SICUREZZA

Oggi, per il domani

PROGETTO “Theta Group”

PRESENTATO DA

TrustVault

Indice

Introduzione	→	3
Chi Siamo	→	4
Rete interna	→	5
Rete esterna	→	9
Analisi del Codice Python	→	10
Preventivo e spesa	→	25
Considerazione finali	→	30



Introduzione

La presente relazione ha l'obiettivo di delineare e mostrare il progetto di rete IT commissionato per conto della Theta Group, includendo un preventivo di spesa ed una dettagliata descrizione dell'infrastruttura IT stessa.

La rete è progettata per supportare un ambiente di lavoro dinamico, implementando 120 computer, distribuiti 20 per piano, nei 6 piani in cui la configurazione prenderà luogo.

Il progetto comprende anche componenti essenziali per la sicurezza della struttura IT, come: 3 server, tra cui uno per il backup dati, un Firewall, dispositivi NAS e sistemi IDS/IPS. Per tener conto della salvaguardia della sicurezza e dell'efficienza operativa.



Chi siamo

TrustVault nasce nel 2015, con sede a Zurigo.
Nata per fornire a piccole e grandi aziende soluzioni tecnologiche innovative.

Siamo specializzati in sviluppo Software, Consulenza IT e CyberSecurity. Il nostro team vanta tra le più eccelse menti del mondo IT, con esperienze certificate di Problem Solving, Innovazione, Comunicazione, Capacità di analisi, e Certificazione professionali tra cui:

- CompTIA Security+
- CISSP
- CISM
- CEH
- CSM



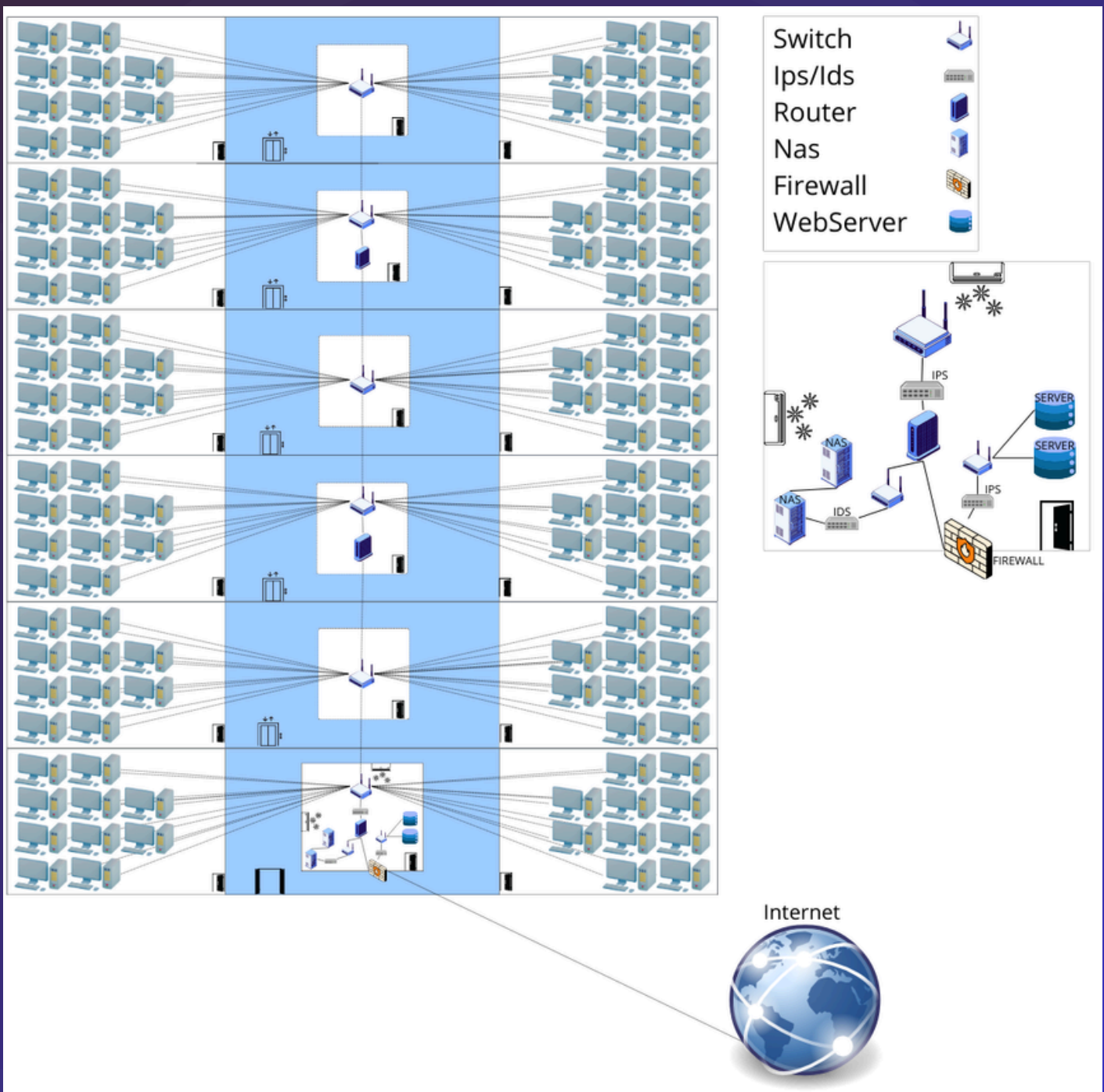
Rete interna

La rete aziendale interna sarà costituita da:

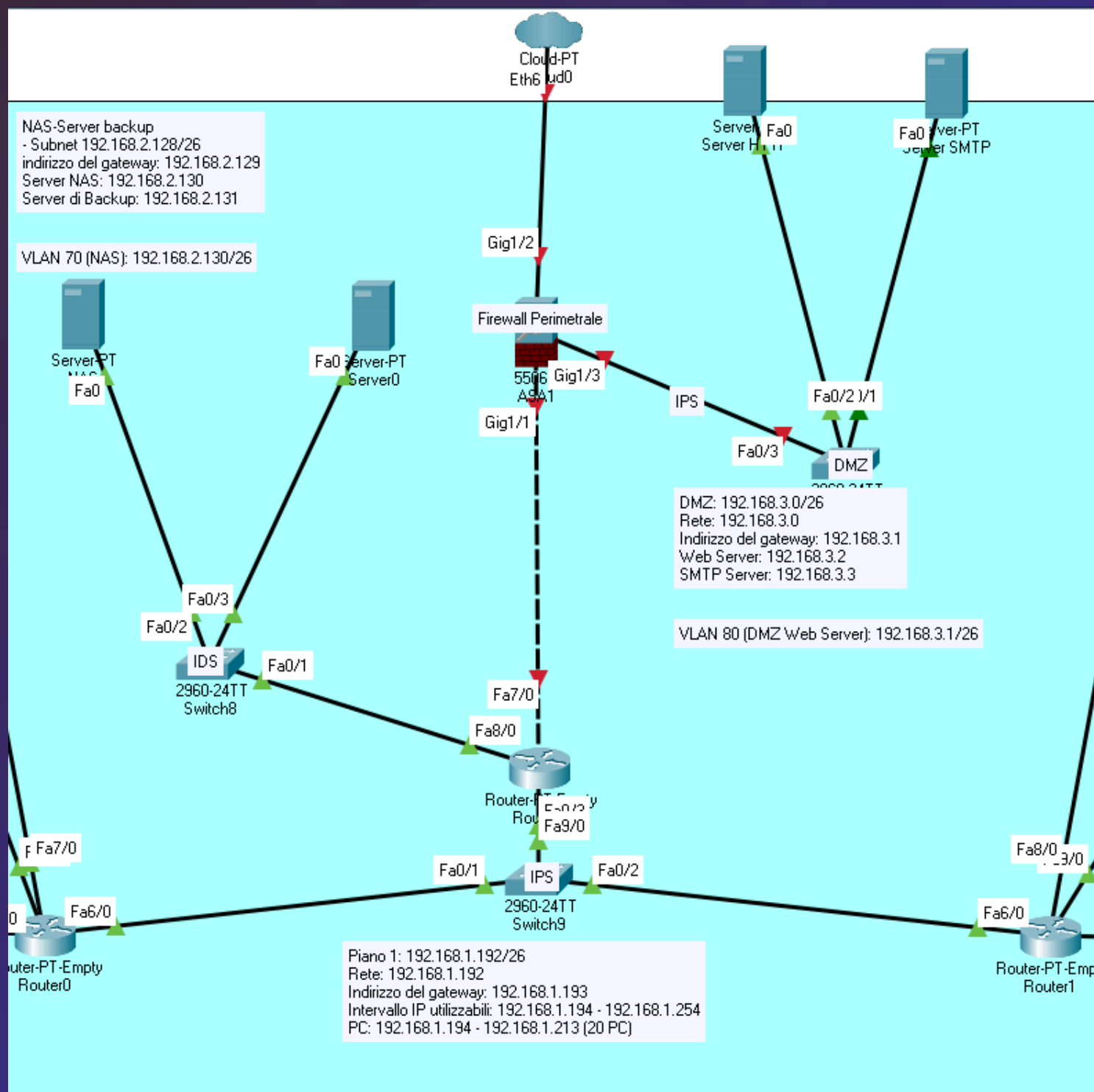
- Switch Layer2: ogni piano ne sarà dotato di uno, per connettere i 20 computer, inoltre permetterà di segmentare il traffico attraverso la creazione di VLAN dedicate.
- Router: collegato agli Switch, gestirà il traffico interno e permetterà l'accesso ad internet.
- Firewall Perimetrale: garantirà protezione da minacce esterne e controllerà il traffico in ingresso ed in uscita (filtraggio pacchetti, prevenzione e segnalazione di intrusioni, VPN).
- NAS: collegato al Router, consentirà un accesso centrale e protetto ai dati da parte di tutti i computer.
- Server di backup: per l'archiviazione dati.
- IDS/IPS: faranno da monitor per il traffico in tempo reale.

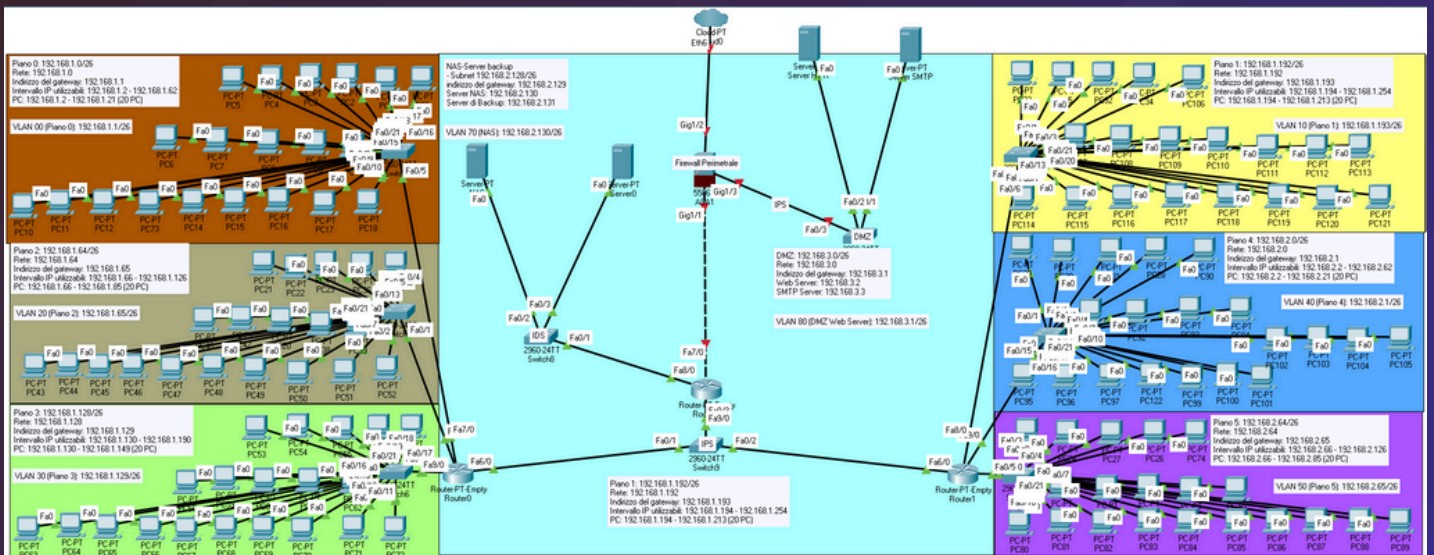


Da questa configurazione grafica possiamo vedere come verrà progettata e configurata la rete per i 6 piani, inoltre è stato previsto di inserire in un'apposita area, i Server, implementando Sistemi di Raffreddamento e Analisi della Temperatura generale.



Questa presentazione rende ancora più chiara la rete interna, mostrando la segmentazione tramite la SubNetting e VLAN.





La subnet mask di /26 consente di avere un totale di 64 indirizzi IP in ogni subnet, di cui 62 utilizzabili (due indirizzi sono riservati: uno per il network address e uno per il broadcast address).

Per calcolare il numero di indirizzi IP in una subnet data una subnet mask di /26, puoi usare la formula:

Numero totale di indirizzi = $2^{(32 - \text{maschera})}$

La maschera in questo caso è 26.

Esempi di Indirizzi con /26

192.168.1.0/26:

Indirizzo di rete: 192.168.1.0

Indirizzo di broadcast: 192.168.1.63

Indirizzi utilizzabili: 192.168.1.1 - 192.168.1.62

Questo è il calcolo dietro la divisione della subnet mask, su più reti.



Rete esterna

La rete esterna permetterà:

Connessione ad internet:

- il Firewall darà accesso ad Internet, garantendo un traffico esterno sicuro e monitorato.

Web Server:

- il DVWA sarà posizionato nella Zona Demilitarizzata (DMZ), permettendo la connessione di esterni salvaguardando la rete interna.

Il nostro obiettivo per Theta Group è quella di fornire una rete IT di facile utilizzo ed accessibile, garantendo la protezione.

Garantiamo per i nostri clienti la massima professionalità, costruendo relazioni solide, sulla base di risultati eccellenti.



Analisi del Codice Python per la Scansione delle Porte e Verifica dei Verbi http

Scansione delle Porte:

```
1 import socket
2
3 # Funzione per la scansione delle porte
4 def scan_ports(ip, start_port, end_port):
5     open_ports = []
6     print(f"Inizio scansione delle porte da {start_port} a {end_port} su {ip}... ")
7
8     for port in range(start_port, end_port + 1):
9         with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
10             result = s.connect_ex((ip, port))
11             if result == 0:
12                 open_ports.append(port)
13                 print(f"Porta {port} aperta")
14             else:
15                 print(f"Porta {port} chiusa")
16
17     print(f"\nScansione completata. Porte aperte su {ip}: {open_ports}")
18     return open_ports
19
20 # Chiedo all'utente l'indirizzo IP e il range di porte
21 ip_address = input("Inserisci l'indirizzo IP del dispositivo: ")
22 start_port = int(input("Inserisci la porta iniziale del range: "))
23 end_port = int(input("Inserisci la porta finale del range: "))
24
25 # Avvio la scansione
26 open_ports = scan_ports(ip_address, start_port, end_port)
```

Viene importata la libreria socket, fondamentale per la creazione di connessioni di rete.



1 - Definizione della funzione scan_ports:

Questa funzione accetta tre parametri:

- ip: l'indirizzo IP del dispositivo da scansionare.
- start_port: la porta iniziale del range da controllare.
- end_port: la porta finale del range da controllare.

2 - Inizio della scansione:

Viene stampato un messaggio che indica l'inizio della scansione.

3 - Ciclo di scansione delle porte:

Si itera su ogni porta nel range specificato.

Per ogni porta:

- Viene creato un socket con `socket.socket(socket.AF_INET, socket.SOCK_STREAM)`.
- Il metodo `connect_ex` viene utilizzato per tentare di connettersi all'indirizzo IP sulla porta specificata. Se la porta è aperta, `connect_ex` restituisce 0.



4 - Controllo dello stato della porta:

Se la porta è aperta, viene aggiunta a `open_ports` e viene stampato un messaggio corrispondente. In caso contrario, viene indicato che la porta è chiusa.

5 - Conclusione della scansione:

Al termine della scansione, viene stampato un riepilogo delle porte aperte trovate e la lista `open_ports` viene restituita.

6 - Interazione con l'utente

Dopo la definizione della funzione, il programma chiede all'utente di inserire:

- Un indirizzo IP.
- La porta iniziale e quella finale del range da scansionare.

Infine, la scansione viene avviata chiamando la funzione `scan_ports` con i parametri forniti dall'utente.



Verifica dei Verbi http

```
1 import http.client
2 from urllib.parse import urlparse
3 def verifica_metodi_http(url):
4     if not url.startswith('http://'): # Se l'input non include "http://", aggiungilo per poter usare urlparse
5         url = 'http://' + url
6     host = urlparse(url).hostname # Parsing dell'URL per ottenere l'host
7     metodi_da_provare = ['OPTIONS', 'GET', 'HEAD', 'POST', 'PUT', 'DELETE'] # Lista dei metodi HTTP da testare
8     metodi_abilitati = set()
9     try: # Provo il metodo OPTIONS per ottenere i metodi supportati
10         connection = http.client.HTTPConnection(host)
11         connection.request('OPTIONS', '/') # Invia la richiesta OPTIONS
12         response = connection.getresponse() # Controlla l'header "Allow" nella risposta
13         allow_header = response.getheader('Allow')
14         if allow_header:
15             metodi = allow_header.replace(' ', '') # Rimuove gli spazi
16             print(f"Metodi abilitati (da 'Allow'): {metodi}") # Stampa direttamente i metodi abilitati
17         else:
18             print("Nessun metodo abilitato trovato nell'header 'Allow'.")
19         connection.close()
20     except Exception as e:
21         print(f"Errore: {e}")
22         return
23     if not metodi_abilitati: # Se 'Allow' non fornisce informazioni, proviamo gli altri metodi
24         for metodo in metodi_da_provare:
25             try:
26                 connection = http.client.HTTPConnection(host)
27                 connection.request(metodo, '/')
28                 response = connection.getresponse()
29                 if response.status in (200, 204):
30                     metodi_abilitati.add(metodo)
31                 print(f"Risposta per {metodo}: {response.status} - {response.reason}")
32                 connection.close()
33             except Exception as e:
34                 print(f"Errore con il metodo {metodo}: {e}") # Mostra i metodi abilitati
35
36     if metodi_abilitati:
37         print(f"Metodi abilitati finali: {'', '.join(metodi_abilitati)}")
38     else:
39         print("Nessun metodo abilitato trovato.")
40
41 if __name__ == "__main__":
42     url_input = input("Inserisci l'URL o IP del target (es. www.example.com o 192.168.1.1): ")
43     verifica_metodi_http(url_input)
```

1 - Importazione librerie:

- `http.client`: Importa la libreria che fornisce classi per la gestione delle comunicazioni HTTP. Sarà utilizzata per inviare richieste HTTP al server.
- `urlparse`: Questa funzione serve a analizzare un URL, permettendo di estrarre diverse parti (come `hostname`, `porta`, ecc.).



Questa funzione prende un singolo argomento, url, che è l'indirizzo da testare.

- `parsed_url.scheme` è un attributo dell'oggetto che contiene il protocollo o lo schema dell'url (`http`, `HTTPS`, ...).
- `parsed_url.scheme` prende in esame la parte del protocollo.
- `Host = parsed_url.hostname`: memorizza il corpo dell'url, esempio • `http://esempio.com`, salverà `esempio.com`.
- `port = parsed_url.port`: Questa parte memorizza in una variabile "port" un numero di porta



2 - Inizializzazione dei metodi da testare:

- `metodi_da_provare`: Elenco di metodi HTTP che verranno testati per determinare quali sono supportati dal server.
- `metodi_abilitati`: Un insieme vuoto che memorizzerà i metodi che il server supporta.

3 - Prova del metodo OPTIONS:

- `try/except`: Gestisce eventuali errori che possono verificarsi durante la connessione o la richiesta.
- `HTTPConnection`: Crea una connessione HTTP con l'host e la porta specificati.
- `request('OPTIONS', '/')`: Invia una richiesta HTTP OPTIONS al server per ottenere i metodi supportati.



4 – Controllo dell'header "Allow":

- `getheader('Allow')`: Recupera il valore dell'header "Allow" dalla risposta, che elenca i metodi HTTP che il server supporta.
- Aggiornamento dell'insieme `metodi_abilitati`: Se l'header è presente, aggiunge i metodi all'insieme, rimuovendo eventuali spazi.
- `allow_header.replace(' ', '')`: Questo rimuove eventuali spazi presenti tra i metodi. Non dobbiamo usare `.strip()` su ogni metodo perché stiamo rimuovendo gli spazi direttamente dalla stringa intera. Ad esempio, se `allow_header` è "GET, POST, OPTIONS", diventerà "GET,POST,OPTIONS".
- Else: ... `connection.close()`: se l'allow è vuoto stampa a schermo il messaggio, Chiude la connessione una volta completata la richiesta.
- Cattura eccezioni generali: Questo blocco `except` cattura qualsiasi eccezione che non è stata gestita dai precedenti blocchi `except`. Questo include errori non specifici che possono verificarsi durante l'esecuzione del codice all'interno del blocco `try`.



Prova degli altri metodi:

- Se l'header "Allow" non è presente, la funzione prova gli altri metodi HTTP (OPTION, GET, POST, PUT, DELETE, HEAD).
- Invia una richiesta per ciascun metodo e controlla se il server risponde con uno stato di successo (200 OK o 204 No Content). Se sì, aggiunge il metodo all'insieme metodi_abilitati.

La condizione `if __name__ == "__main__":` serve a controllare se il file sta venendo eseguito come programma principale.

Solo se la condizione è vera (cioè se il file è eseguito direttamente), il codice all'interno di questo blocco verrà eseguito.

Ecco un elenco delle porte comuni e degli attacchi informatici più frequenti associati a ciascuno di esse:



Porta 21: FTP

Attacchi:

- Brute Force: Tentativi di accesso non autorizzato mediante l'indovinare le credenziali.
- Packet Sniffing: Cattura di dati non crittografati in transito.

Porta 22: SSH

Attacchi:

- Brute Force: Tentativi di accesso non autorizzato attraverso username e password.
- Man-in-the-Middle: Attacco in cui un malintenzionato intercetta la comunicazione.

Porta 23: Telnet

Attacchi:

- Packet Sniffing: Cattura di dati non crittografati.
- Session Hijacking: Assunzione di controllo su una sessione Telnet attiva.

Porta 25: SMTP

Attacchi:

- Email Spoofing: Invio di email mascherate come provenienti da fonti legittime.
- Open Relay: Utilizzo del server per inviare spam se non configurato correttamente.

Porta 53: DNS

Attacchi:

- DNS Spoofing: Manipolazione delle risposte DNS per reindirizzare il traffico a siti malevoli.
- DDoS (Distributed Denial of Service): Attacco che sovraccarica il server DNS con richieste.

Porta 80: HTTP

Attacchi:

- Cross-Site Scripting (XSS): Iniezione di codice maligno in pagine web per attaccare utenti.
- SQL Injection: Iniezione di comandi SQL per manipolare il database.

Porta 111: RPC

Attacchi:

- RPC DDoS: Sfruttamento delle chiamate RPC per sovraccaricare i servizi.
- Remote Code Execution: Esecuzione di codice malevolo su un server vulnerabile.

Porta 139: NetBIOS

Attacchi:

- NetBIOS Name Spoofing: Inganno dei sistemi per dirottare il traffico di rete.
- Brute Force: Tentativi di accesso non autorizzato a risorse condivise.
- leverei solo la porta 111(RCP)



REPORT FINALE

Rapporto sulla Scansione delle Porte del Dispositivo:

A seguito della scansione di alcune porte sul dispositivo con indirizzo IP 192.168.1.144, sono state rilevate le seguenti porte aperte:

- Porta 21: FTP
- Porta 22: SSH
- Porta 23: Telnet
- Porta 25: SMTP
- Porta 53: DNS
- Porta 80: HTTP
- Porta 111: RPC
- Porta 139: NetBIOS

Osservazioni e Raccomandazioni:

Valutazione delle Porte Aperte:

Le porte aperte possono rappresentare potenziali punti di accesso per attacchi esterni. È fondamentale valutare la necessità di ciascuna porta aperta in base alle applicazioni in uso.



Disabilitazione dei Servizi Non Necessari:

Si consiglia di disabilitare i servizi associati a porte che non sono necessarie per il funzionamento del dispositivo. Tra le porte aperte rilevate sul dispositivo, i seguenti protocolli possono essere considerati non necessari in molti contesti:

Porta 21 (FTP):

- FTP è un protocollo di trasferimento file non sicuro, vulnerabile a intercettazioni e attacchi. È consigliabile utilizzare SFTP o FTPS, che offrono un livello di sicurezza maggiore.

Porta 23 (Telnet):

- Telnet trasmette dati in chiaro, rendendolo insicuro. È preferibile utilizzare SSH (porta 22) per le connessioni remote sicure.

Porta 139 (NetBIOS):

- Questo protocollo è spesso non necessario per le configurazioni moderne e può essere utilizzato per attacchi di rete. È consigliabile disabilitarlo se non utilizzato.



Implementazione di Firewall:

- È consigliabile utilizzare un firewall per filtrare il traffico in entrata e in uscita. Configurando regole appropriate, potrai limitare l'accesso solo alle porte e ai servizi essenziali.

Monitoraggio Continuo:

- Implementare un sistema di monitoraggio per rilevare attività anomale o tentativi di accesso non autorizzato. Ciò consente una risposta tempestiva a potenziali minacce.

Aggiornamenti e Patch:

- Assicurati che tutti i servizi in esecuzione siano aggiornati con le ultime patch di sicurezza per mitigare vulnerabilità note.

Seguendo queste raccomandazioni, potrai migliorare significativamente la sicurezza del dispositivo e ridurre il rischio di attacchi informatici. Per ulteriori informazioni o assistenza nella messa in atto di queste misure, non esitare a contattarci.



Raccomandazioni per la Sicurezza dei Metodi HTTP Abilitati

A seguito della verifica dei metodi HTTP abilitati sul vostro server, è emerso che sono attivi i seguenti metodi: OPTIONS, POST, PUT, DELETE, HEAD, e GET. È fondamentale considerare la gestione di questi metodi per migliorare la sicurezza complessiva della vostra infrastruttura.

Raccomandazioni:

Disabilitare Metodi Non Necessari:

- Se i metodi PUT e DELETE non sono utilizzati per le operazioni quotidiane, si consiglia di disabilitarli. Questi metodi possono rappresentare un rischio significativo, poiché consentono la modifica e la cancellazione di risorse sul server.

Implementare Autenticazione e Autorizzazione:

- È essenziale garantire che tutte le richieste ai metodi abilitati siano protette mediante adeguati meccanismi di autenticazione e autorizzazione. Ciò contribuirà a prevenire accessi non autorizzati alle risorse del server.

Monitoraggio e Logging:

- Si raccomanda di implementare un sistema di monitoraggio e logging per rilevare eventuali tentativi di accesso non autorizzati o utilizzi anomali dei metodi HTTP abilitati. Questo può fornire un'importante visibilità sulla sicurezza delle vostre applicazioni.



Eseguire Test di Sicurezza Regolari:

- Infine, si consiglia di effettuare test di sicurezza periodici per identificare nuove vulnerabilità e garantire che le misure di sicurezza siano sempre aggiornate e adeguate.

Seguendo queste raccomandazioni, potrete ridurre significativamente il rischio di exploit legati ai metodi HTTP abilitati. Per ulteriori chiarimenti o assistenza nella messa in opera di queste misure, non esitate a contattarci.

Risultato del test eseguito con Python

```
root@kali: /home/kali/Desktop/progetto_vpn
File Actions Edit View Help
(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
(root@kali)-[/home/kali]
# cd Desktop/progetto_vpn

(root@kali)-[/home/kali/Desktop/progetto_vpn]
# python3 msf.py
Inserisci l'URL o IP del target (es. www.example.com o 192.168.1.1): http://192.168.1.144/
phpMyAdmin/
Nessun metodo abilitato trovato nell'header 'Allow'.
Risposta per OPTIONS: 200 - OK
Risposta per GET: 200 - OK
Risposta per HEAD: 200 - OK
Risposta per POST: 200 - OK
Risposta per PUT: 200 - OK
Risposta per DELETE: 200 - OK
Metodi abilitati finali: PUT, GET, HEAD, OPTIONS, POST, DELETE

(root@kali)-[/home/kali/Desktop/progetto_vpn]
#
```




```
root@kali: /home/kali/Desktop/progetto_vpn
File Actions Edit View Help
(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
(root@kali)-[/home/kali]
# cd Desktop/progetto_vpn

(root@kali)-[/home/kali/Desktop/progetto_vpn]
# python port_scan.py
Inserisci l'indirizzo IP del dispositivo: 192.168.1.144
Inserisci la porta iniziale del range: 0
Inserisci la porta finale del range: 1024
Inizio scansione delle porte da 0 a 1024 su 192.168.1.144 ...
Porta 0 chiusa
Porta 1 chiusa
Porta 2 chiusa
Porta 3 chiusa
Porta 4 chiusa
Porta 5 chiusa
Porta 6 chiusa
Porta 7 chiusa
Porta 8 chiusa
Porta 9 chiusa
Porta 10 chiusa
Porta 11 chiusa
Porta 12 chiusa
Porta 13 chiusa
Porta 14 chiusa
```

```
root@kali: /home/kali/Desktop/progetto_vpn
File Actions Edit View Help
Porta 1004 chiusa
Porta 1005 chiusa
Porta 1006 chiusa
Porta 1007 chiusa
Porta 1008 chiusa
Porta 1009 chiusa
Porta 1010 chiusa
Porta 1011 chiusa
Porta 1012 chiusa
Porta 1013 chiusa
Porta 1014 chiusa
Porta 1015 chiusa
Porta 1016 chiusa
Porta 1017 chiusa
Porta 1018 chiusa
Porta 1019 chiusa
Porta 1020 chiusa
Porta 1021 chiusa
Porta 1022 chiusa
Porta 1023 chiusa
Porta 1024 chiusa

Scansione completata. Porte aperte su 192.168.1.144: [21, 22, 23, 25, 53, 80, 111, 139, 445, 512, 513, 514]

(root@kali)-[/home/kali/Desktop/progetto_vpn]
#
```



Preventivo Spesa

Area Server – Data Center

BUDGET PROGETTO THET				
COMPONENTI HARDWARE				
SERVER & Amministrazione				
Quantità	Tipologia	Prezzo Unità	Prezzo Totale	Descrizione
2	Server	€ 9.800,00	€ 19.600,00	HPE Matrice intelligente
1	Server backup	€ 9.800,00	€ 9.800,00	HPE Matrice intelligente
1	NAS	€ 9.000,00	€ 9.000,00	QNAP TS-h3087XU-RP-E2378-64G Intel Xeon E-2378 8 Core/16 Th
74	RAM (Server+NAS - 24+24+2)	€ 80,00	€ 5.920,00	Kingston FURY Beast (1 x 32GB, DDR4-3600, DIMM 288 pin)
40	HD SATAIII (Server+NAS - 8+8+16)	€ 170,00	€ 6.800,00	Seagate Exos 7E10 512E/4kn SATA 8 tb
2	Alimentatori	€ 450,00	€ 900,00	HPE AF538A
1	IDS	€ 3.350,00	€ 3.350,00	Cisco C3725-VPN-IDS/K9
2	IPS	€ 4.000,00	€ 8.000,00	Stormshield SN520 Security Appliance
1	Firewall Hardware	€ 22.000,00	€ 22.000,00	Cisco SECURE FIREWALL 3110 NXFG
2	Batterie/UPS	€ 8.600,00	€ 17.200,00	APC USV SRT10KXLI, 10000W
1	RACK	€ 2.500,00	€ 2.500,00	Tripp Lite SR42UBEIS SmartRack Enclosure Rack Cabinet, NEMA
1	RACK Accessori	€ 2.000,00	€ 2.000,00	
SUBTOTALE			€ 107.070,00	

In questa sezione abbiamo scelto l'hardware per l'area Server e l'amministrazione degli stessi. I server scelti sono tra i più venduti al mondo e offrono sicurezza, prestazioni ed espandibilità eccezionali. In questa configurazione che abbiamo progettato la vostra azienda potrà fare affidamento su un servizio telematico veloce ed efficiente, inoltre, con prospettive di crescita, non dovrete temere il carico di lavoro sui server. Il NAS (Network Attached Storage) verrà equipaggiato con 124 TB di memoria.



Area Piani / Workstation

FLOORS & HOST				
30	PC Laptop	€ 997,00	€ 29.910,00	LENO
90	PC Desktop	€ 560,00	€ 50.400,00	INTEL
92	Monitor 24"	€ 80,00	€ 7.360,00	Sams
8	Switch	€ 960,00	€ 7.680,00	Cisco
3	Router Gateway - Firewall(opzionale)	€ 710,00	€ 2.130,00	Cisco
100	Mouse&Keyboard set	€ 35,00	€ 3.500,00	Logite
SUBTOTALE			€ 100.980,00	

Abbiamo pensato all'acquisto di 30 PC Laptop (portatili) per dare più mobilità ai dipendenti e di 90 PC Desktop con relativi monitor da 24". Per la distribuzione della rete lungo tutto il building sono stati pensati due Router Gateway (che all'occorrenza possono fungere anche da Firewall) e di 8 Switch da 24 porte ognuno.



Maintenance e Security

Maintenance & Security				
2	Condizionatori 16000BTU/Deumid.	€ 650,00	€ 1.300,00	ECOFORT Klimagerät CoolAir 1
1	Deumidificatore	€ 1.780,00	€ 1.780,00	Airecoler Deumidificatore comm
8	Cavo Rete Cat6 - 100 mt	€ 38,00	€ 304,00	Gembird Cavo di rete F/UTP, CAT6, 1
1	Cavo Rete Cat8 - 100 mt	€ 150,00	€ 150,00	Datenkabel CAT.8 2000 MHz AW
10	Connettori RJ45 CAT6 pz40	€ 8,00	€ 80,00	Greluma 40 Pz Connettori passa
15	Connettori RJ45 CAT8 pz4	€ 42,00	€ 630,00	Primewire 4x connettore di rete F
0	Rilevatore Umidità	€ 70,00	€ 0,00	CO2 Messgerät WL1025
1	Rilevatore Umidità/temp	€ 130,00	€ 130,00	WLAN-Funk-Wetterstation, Aussens
2	Door Locker - fingerprint	€ 110,00	€ 220,00	dnt Fingerprint-Türgriff DoorAcce
	SUBTOTALE		€ 4.594,00	

Verranno acquistati due condizionatori ed un deumidificatore. Verrà installato anche un rilevatore di umidità e temperatura. Inoltre abbiamo pensato di dotare la porta di ingresso del data center e la porta di accesso al rack dei server di un Door Locker biometrico e con codice PIN in modo tale da prevenire l'accesso a personale non autorizzato. I cavi di connessione saranno, per quanto riguarda tutta la rete network nella struttura, tutti di Cat6 - che assicurano una trasmissione dati sino a 10 Gigabit, mentre il data center verrà dotato di cavi Cat8 che assicurano una trasmissione dati sino a 40 Gigabit.



Costo installazione / Licenze

Work Costs /				
1	Progettazione/Configurazione	€ 7.995,00	€ 7.995,00	
3	Messa in opera/Installazione	€ 7.995,00	€ 23.985,00	
2	Installazione condizionatori	€ 200,00	€ 400,00	
1	Windows Server - annuale	€ 1.100,00	€ 1.100,00	
120	Windows 11 pro - annuale	€ 60,00	€ 7.200,00	

Il processo di progettazione, configurazione e messa in opera richiede personale specializzato con competenze tecniche acquisite con anni di esperienza nel campo della sicurezza e dell'informatica. I nostri compensi sono in linea col mercato. Inoltre ci saranno i costi legati alle licenze di Windows Server e del software Windows 11 pro per ogni computer.



Riassunto

BUDGET PROGETTO THETA

SERVER & Amministrazione		Normale	Pro
	SUBTOTALE	€ 107.070,00	€ 107.070,00
Floor & HOST			
	SUBTOTALE	€ 100.980,00	€ 122.010,00
Maintenance & Security			
	SUBTOTALE	€ 4.594,00	€ 4.594,00
Work Costs / Licenze			
	SUBTOTALE	€ 40.680,00	€ 40.680,00
	TOTALE	€ 253.324,00	€ 274.354,00

Qui potrete avere un grafico riassuntivo dell'investimento previsto (IVA inclusa) per la Theta Group. Quello descritto in precedenza è la versione normale; potreste optare per un investimento "Pro" che comprende un videoproiettore, 6 stampanti (una per piano) e monitor da 32" per i computer Desktop.

Si prega di accedere al file Excel allegato per avere maggiori dettagli: ...



Considerazioni finali

La TrustVault ha ritenuto opportuno l'installazione di due Server (uno HTTP ed uno SMTP) per un regolare flusso dei dati e di un NAS per l'archiviazione dei dati interni. Tra rete LAN e WAN è stato configurato un Firewall perimetrale che assicura una prima, necessaria ed opportuna scannerizzazione dei dati in entrata. Inoltre i due server sono equipaggiati da un sistema IPS (Intrusion Prevention Systems) e da un NAS ulteriormente messo in sicurezza da un IDS (Intrusion Detection Systems) che assicurerà un regolare svolgimento dell'attività lavorativa. È stato possibile creare la DMZ (Demilitarized zone) che andrà a prevenire possibili minacce dirette dall'esterno verso l'interno dell'azienda.



Conclusioni

Il progetto, così come è stato strutturato, assicura alla Theta Group sicurezza e fruibilità. Inoltre, visti i componenti hardware scelti, la Theta Group avrà a disposizione tutti gli strumenti necessari per raggiungere i propri obiettivi avendo, senza dubbio, un grande vantaggio competitivo sui propri competitor a livello hardware e strutturale.

Progetto realizzato da:

Lorenzo Oliva
Alessio di Donato
Antonio Bevilacqua
Mattia Montis
Daniel Gabriel Costeanu
Matteo Pasqualini
Umberto Valentini

