Esercizio_S6_L5_Hydra

Consegna

L'esercizio si svilupperà in due fasi:

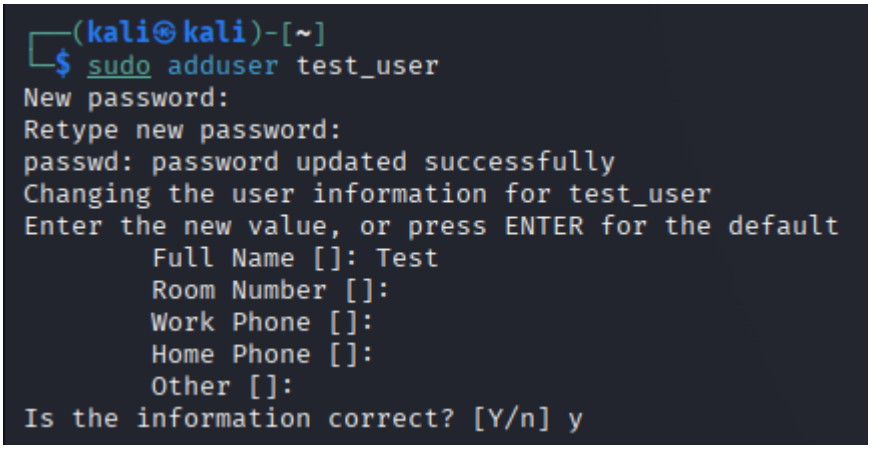
- Una prima fase dove insieme vedremo l'abilitazione di un servizio SSH e la relativa sessione di cracking dell'autenticazione con Hydra.
- Una seconda fase dove sarete liberi di configurare e craccare un qualsiasi servizio di rete tra quelli disponibili, ad esempio ftp, rdp, telnet, autenticazione HTTP.

Svolgimento

Ho deciso di svolgere l'esercizio utilizzando esclusivamente la **VM di Kali Linux**, simulando quindi l'attacco direttamente sul **localhost** tramite l'uso di **Hydra**.

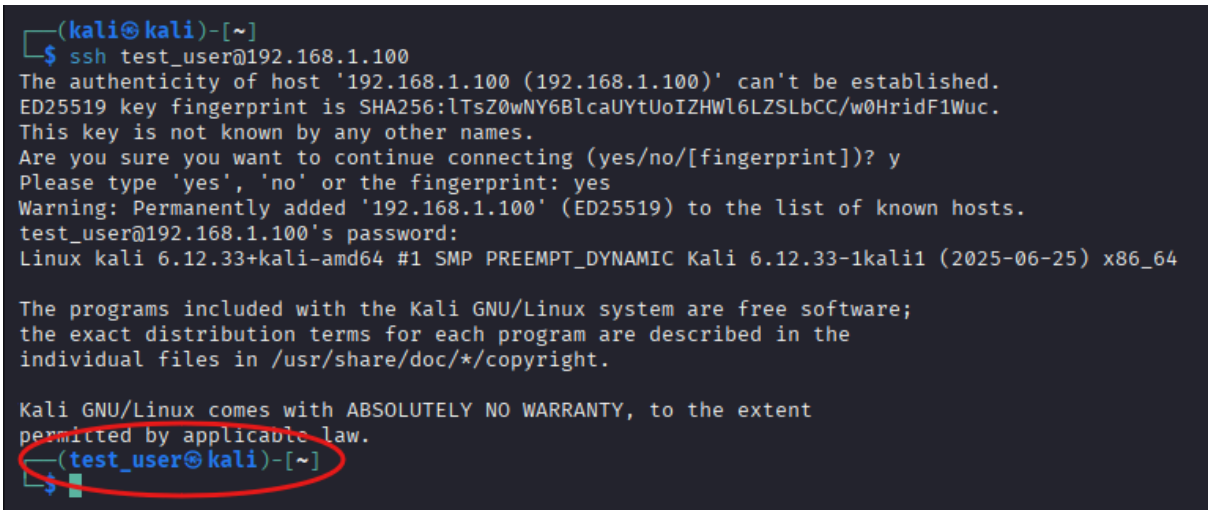
Preparazione Target

Per prima cosa è stato necessario creare un nuovo utente tramite comando adduser seguito dal nome richiesto nella consegna; fatto ciò ci verrà richiesto di configurare la password, che in questo caso doveva essere testpass, assieme ad alcuni altri parametri opzionali di cui ho compilato solamente il nome:



Ho poi proseguito startando il servizio ssh tramite il comando sudo service ssh start e successivamente ne ho testato il funzionamento collegandomi al servizio utilizzando il nuovo account creato in precedenza:

ssh test_user@192.168.1.100



Preparazione Hydra

Per prima cosa ho creato un file di testo contenente una lista di nomi utente da testare durante l'attacco, inserendo tra questi anche l'utente test_user precedentemente configurato.

Per la lista di password ho deciso di utilizzare la wordlist rockyou.txt, una delle più note e utilizzate in ambito di penetration testing.
La scelta è ricaduta su rockyou.txt **dopo aver verificato la presenza della password testpass al suo interno**, utilizzando il seguente comando:

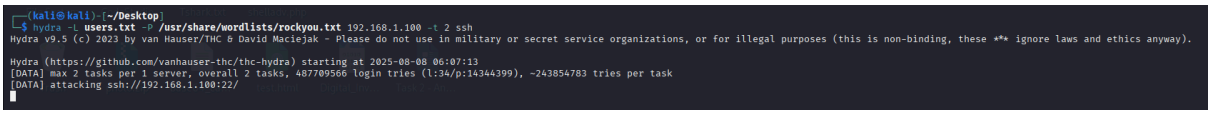
grep -Fx "testpass" /usr/share/wordlists/rockyou.txt

Questo controllo ha confermato che la password era effettivamente presente nella lista, rendendola adatta per il test di cracking con Hydra.

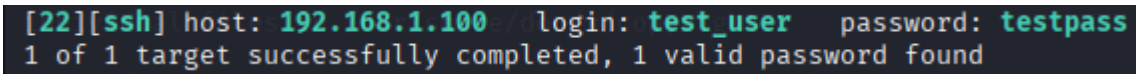
Cracking Credenziali Servizio SSH

Successivamente possiamo avviare il processo per il Brute force del servizio ssh tramite Hydra.

hydra -L users.txt -P /usr/share/wordlists/rockyou.txt 192.168.1.100 -t 4 ssh



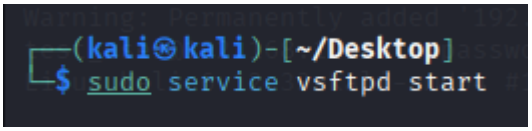
Dopo diverse ore il processo termina ed abbiamo il risultato: hydra è riuscito a validare le credenziali che cercavamo.



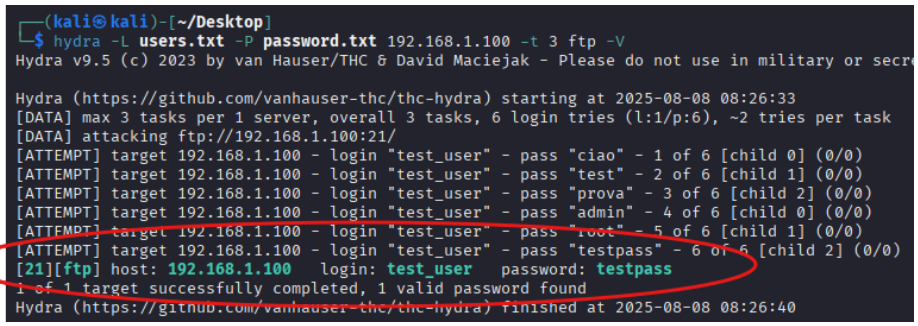
Cracking Credenziali servizio FTP

Per testare e fare ulteriore pratica con Hydra ho poi installato svftpd per avviare un server FTP:

Ho poi startato il server FTP:



Ed ho infine avviato nuovamente Hydra, stavolta utilizzando una lista di psw minore ed un unico username contenuto all'interno dello stesso file precedente. Ho dunque aggiunto altresì il parametro -V per avviare il programma in modalità verbose e vedere passo a passo i test eseguiti dal tool:

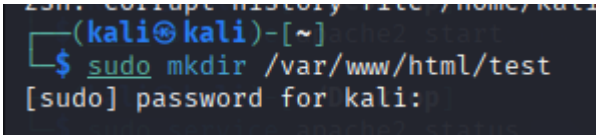


Cracking Credenziali HTTP

Terminiamo la simulazione con il cracking delle credenziali su un servizio HTTP;

Per l'occasione ho deciso di utilizzare apache2, webserver che avevo già installato in precedenza.

Proseguo creando una cartella test all'interno di /var/www/html/

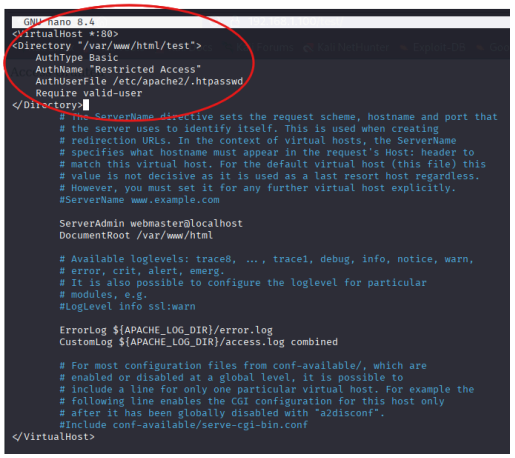


Ed ho infine creato l'utente all'interno del webserver tramite comando

sudo htpasswd -c /etc/apache2/.htpasswd test_user

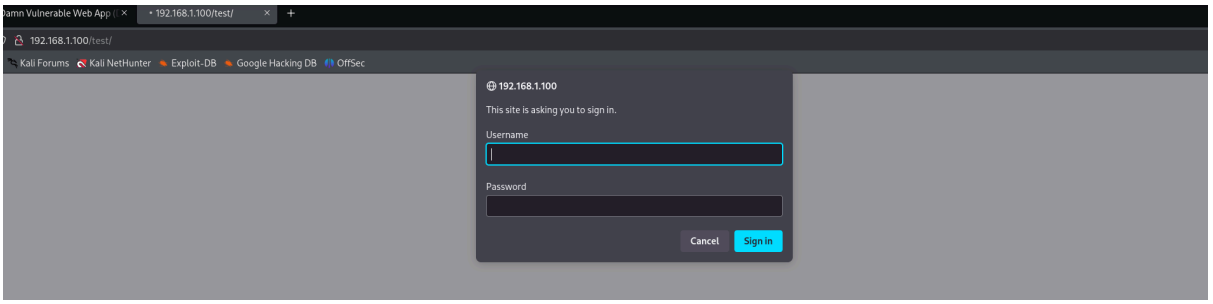
Implemento poi, inserendo il codice sottostante all'interno del file di configurazione di apache2, un metodo di autenticazione per il webserver.

sudo nano /etc/apache2/sites-enabled/000-default.conf



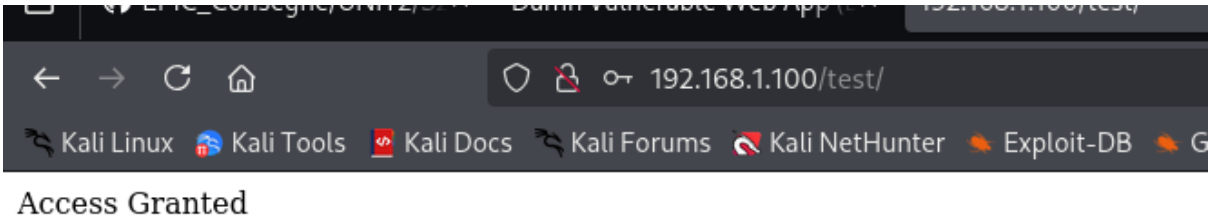
Riavviamo poi il servizio apache2 affinché rielabori le modifiche attuate.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo service apache2 restart
```



Ed infine creiamo un semplicissimo file html che mostri un messaggio *"Access Granted"* qualora si acceda alla pagina web.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo "Access Granted" | sudo tee /var/www/html/test/index.html
Access Granted
```



Access Granted

Eseguiamo nuovamente il comando di Hydra come fatto in precedenza variando ovviamente il servizio target:

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -L users.txt -P password.txt 192.168.1.100 -t 3 http-get /test -V
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secre

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-08-08 09:03:22
[DATA] max 3 tasks per 1 server, overall 3 tasks, 6 login tries (l:1/p:6), ~2 tries per task
[DATA] attacking http-get://192.168.1.100:80/test
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "ciao" - 1 of 6 [child 0] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "test" - 2 of 6 [child 1] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "prova" - 3 of 6 [child 2] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "admin" - 4 of 6 [child 1] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "root" - 5 of 6 [child 0] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "testpass" - 6 of 6 [child 2] (0/0)
[80][http-get] host: 192.168.1.100   login: test_user   password: testpass
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-08 09:03:22
```

Notiamo nuovamente che **hydra** riesce a lavorare anche con http, dimostrando dunque una grande versatilità!

Precisazioni finali

Per ridurre drasticamente il tempo necessario per completare l'attacco di tipo brute-force, è possibile distribuire il carico di lavoro su più macchine virtuali.

In particolare, si possono creare istanze VM aggiuntive che condividono la lista di password da testare, suddividendola in porzioni distinte.
Ogni macchina può quindi eseguire una parte dell'attacco in parallelo, permettendo così di accelerare significativamente il processo di cracking.

Questo approccio simula un attacco distribuito (Distributed Brute Force) e si dimostra particolarmente efficace in ambienti controllati.

Terminiamo il tutto spegnendo i servizi tramite comando:

sudo service ssh stop
sudo service svftp stop
sudo service apache2 stop

E rimuovo infine l'utente creato allo scopo del test:

sudo deluser --remove-home test_user

Conclusioni

L'esercizio ha dimostrato l'efficacia (e i limiti) di un attacco brute-force tramite Hydra su diversi servizi di rete (SSH, FTP, HTTP).
L'uso della wordlist rockyou.txt ha evidenziato l'importanza di:

- **Ottimizzare le liste di password**, evitando wordlist enormi se non necessario.
- **Valutare l'impatto del numero di thread** (-t) sull'efficienza dell'attacco.
- **Conoscere le configurazioni dei servizi target** (es. tipo di autenticazione HTTP).

L'attacco ha avuto esito positivo in tutti i casi, con identificazione corretta delle credenziali.
Questo laboratorio ha permesso di comprendere:

- l'importanza di usare **password robuste.**
- il funzionamento degli **attacchi a dizionario.**
- l'uso corretto di strumenti come **Hydra.**