# Relazione Tecnica: Configurazione Servizi e Test di Autenticazione su Kali Linux

**Data:** 16 Gennaio 2026

**Ambiente:** Kali Linux

**Target IP:** 192.168.40.20

## 1) Introduzione

**Hydra** (noto anche come THC-Hydra) è un framework di sicurezza open-source specializzato nell'esecuzione di attacchi di tipo **network login cracking** parallelizzati. Sviluppato dal gruppo *The Hacker's Choice* (THC), è considerato lo standard industriale per testare la robustezza dei meccanismi di autenticazione dei servizi esposti su reti TCP/IP. A differenza dei tools di cracking offline (come **Hashcat** o **John the Ripper**), Hydra opera **online**, interagendo direttamente con i servizi di rete in tempo reale per verificare la validità di coppie di credenziali (**username** e **password**). Il core di Hydra è progettato per il **multithreading**. La sua capacità di gestire connessioni multiple in parallelo (fino a centinaia di thread simultanei) permette di massimizzare il throughput dei tentativi di login, riducendo drasticamente i tempi necessari per completare un attacco a dizionario rispetto a script sequenziali. Uno degli aspetti tecnici più rilevanti è la sua natura modulare. Infatti Hydra supporta oltre **50 protocolli di rete** (**SSH**, **FTP**, **TELNET**, ecc), rendendoli uno strumento universale per il penetration testing.

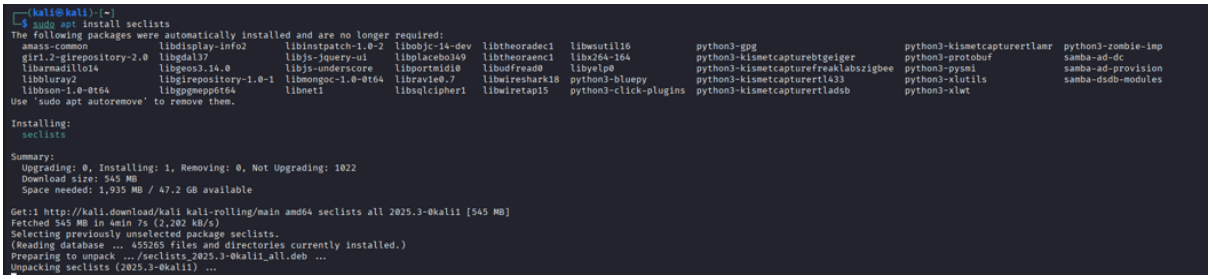
## 2) Obiettivo dell'Attività

L'attività riguarda la configurazione di un ambiente di test controllato, la creazione di utenze di prova e la successiva verifica della robustezza delle credenziali tramite strumenti di auditing per i servizi **SSH** e **FTP**.

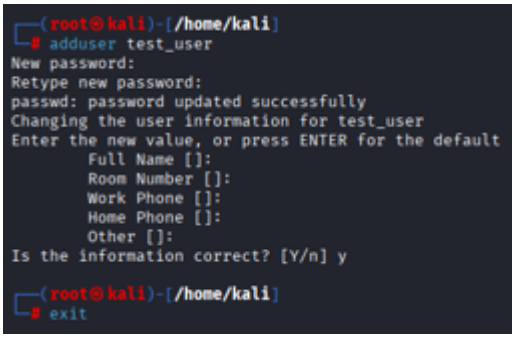
## 3) Preparazione dell'Ambiente e Utenze

Le prime fasi hanno riguardato l'installazione dei tool necessari e la configurazione del sistema **target 192.168.40.20**.

**-Installazione Seclists:** È stato installato il pacchetto seclists, una collezione di liste di nomi utente, password e dizionari essenziali per i test di sicurezza.



**-Creazione Utente:** È stato creato un utente di sistema denominato **test_user** e ad esso è stata associata la password **testpass** tramite il comando "*adduser*".




**-Preparazione Dizionari:** Tramite il comando "*grep*", sono stati filtrati i file di *Seclists* per creare file più piccoli e mirati con i seguenti comandi:


**-xato-usernames.txt:** In questo file è presente la lista degli utenti filtrata.


**-xato-passwords.txt:** In questo file è presente la lista di password filtrata.

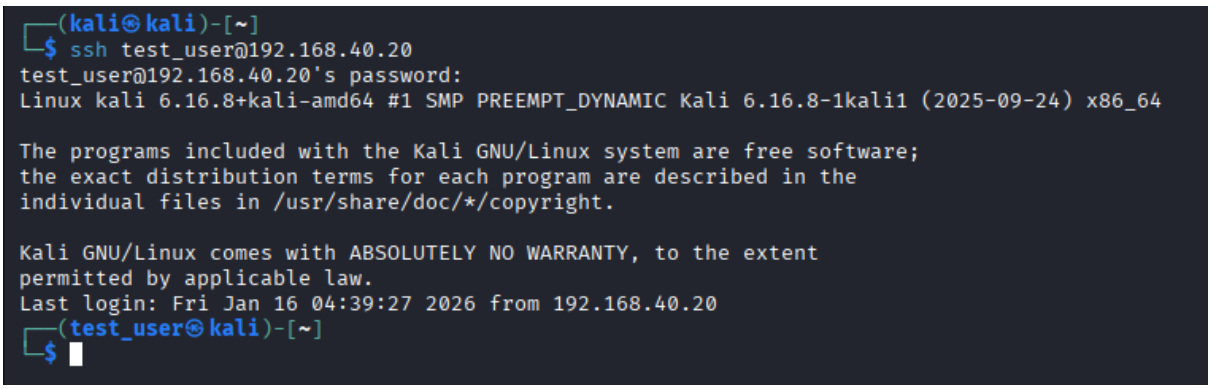
## 4) Configurazione dei Servizi

Per testare l'accessibilità remota, sono stati configurati i seguenti servizi sul target:

**-Servizio SSH**

Il servizio **SSH** è stato verificato tramite un accesso diretto. Come mostrato negli screenshot, l'utente test_user è in grado di autenticarsi correttamente utilizzando, come detto, la password **testpass** sul sistema al target **192.168.40.20.**

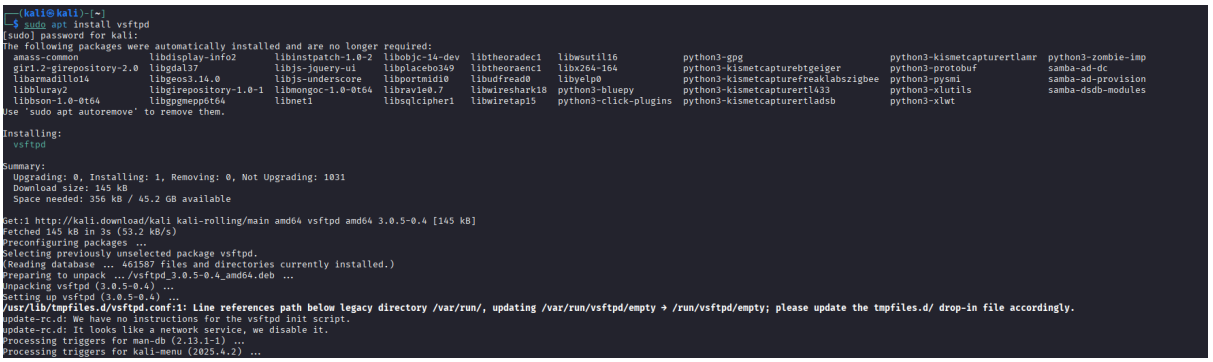
**Risultato: accesso effettuato correttamente!**



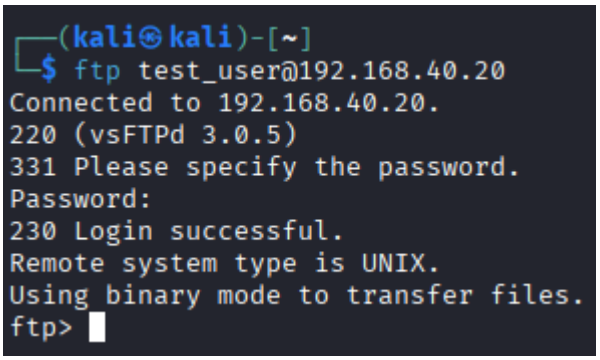
**-Servizio FTP**

È stato installato il server **FTP vsftpd (v3.0.5)** tramite il gestore pacchetti apt. Dopo l'installazione, il servizio è stato avviato per consentire i test di login.

**Installazione vsftpd**




**Verifica del login: Successo!**




## 5) Test di Autenticazione (Brute-Force Auditing)

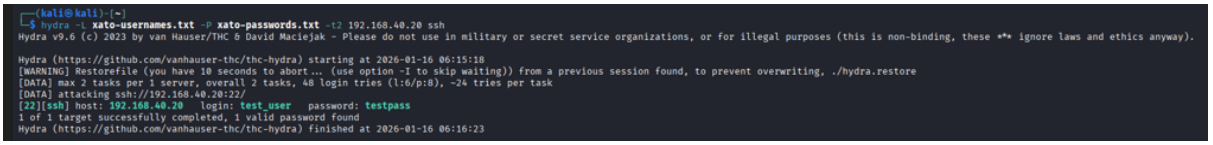
Per trovare **utente** e **password** usiamo un attacco brute force (In cybersecurity, il **brute force** è un metodo di attacco che consiste nel tentare sistematicamente tutte le combinazioni possibili di credenziali fino a trovare quella corretta). Abbiamo usato il comando:

***hydra -L xato-usernames.txt -P xato-passwords.txt -t2 192.168.40.20 ssh***

Questo comando ci permette di fare un brute force al target **192.168.40.20** sull protocollo SSH. In questo caso facciamo finta di non conoscere né utente e nè password, utilizziamo quindi i file di testo che ci siamo precedentemente creati ***xato-usernames.txt, xato-passwords.txt*** con i rispettivi parametri ***-L*** (si usa per caricare una **lista di possibili utenti**) e ***-P (***si usa per caricare la **wordlist delle password**) mentre l'opzione ***-T2*** controlla il numero di **thread**, ovvero quanti tentativi di login il programma esegue contemporaneamente.

Lanciamo il comando e vediamo cosa succede.

**Risultato: Successo!**



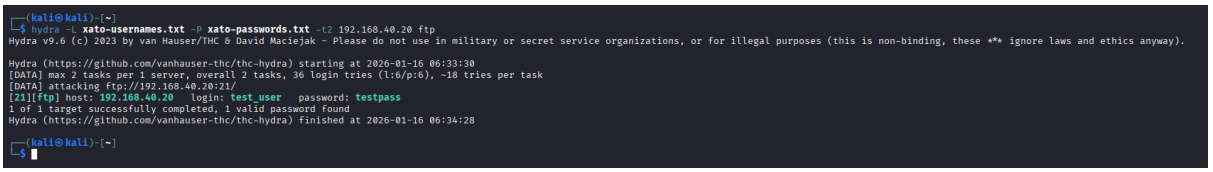

Abbiamo trovato utente: **test_user** e la password: **testpass.**

Facciamo lo stesso con il protocollo **FTP** usando lo stesso comando ma cambiando il protocollo:

***hydra -L xato-usernames.txt -P xato-passwords.txt -t2 192.168.40.20 ftp***

Anche in questo caso lanciamo il comando e vediamo cosa succede.

**Risultato: Successo!**




Abbiamo trovato utente :**test_user** e la password: **testpass.**

## 6) Conclusione

L'analisi effettuata dimostra l'efficacia di **Hydra** come strumento di verifica della sicurezza dei protocolli di rete. L'utilizzo del tool ha permesso di comprendere l'importanza della gestione delle sessioni e della protezione delle interfacce di login. In ottica di difesa (Blue Teaming), i risultati sottolineano come la sicurezza non dipenda solo dalla complessità della password, ma anche dalla capacità del sistema di rilevare e bloccare tentativi di connessione multipli e parallelizzati.

Per difendersi si raccomanda di usare:

**-Implementazione di Meccanismi di Adaptive Blocking:** L'adozione di soluzioni come **Fail2Ban** o **CrowdSec** è indispensabile per analizzare i log in tempo reale e bloccare gli indirizzi IP che mostrano pattern di attacco tipici di Hydra (molteplici tentativi falliti in un lasso di tempo ridotto).

**-Transizione verso l'Autenticazione Passwordless:** Ove possibile, si raccomanda di disabilitare l'autenticazione basata su password a favore di sistemi più robusti, come l'uso di chiavi asimmetriche (SSH Keys) o certificati digitali.

**-Introduzione del Multi-Factor Authentication (MFA): L'MFA** neutralizza l'efficacia di Hydra, poiché la conoscenza della password non è più sufficiente per ottenere l'accesso al sistema.

**-Monitoraggio e Logging:** È necessario configurare sistemi di **SIEM (Security Information and Event Management)** per generare alert immediati in caso di anomalie nei tentativi di login, permettendo così una risposta rapida agli incidenti.