Remote Access for Kali Linux Powered Machines

Abstract:

Generally, when starting in cyber security the first learning point in the "practical knowledge" zone is undoubtedly Kali Linux. For newcomers in this field it is very important to get their hands dirty with Kali and in-depth understanding of this Linux distribution can build a strong foundation for the future. There are various forms of the distro available that range right from ARM based CPU's to virtual machines to full install options. This self-project can help enhance the user experience and can prove to be very helpful for headless connections.

Possible Use Cases:

1. Specifically, ARM based Kali Linux is the most portable and cheapest option. Small card sized computers like the Raspberry Pi running Kali Linux have now become a very vital tool in the cyberspace and get the job done really quick. But to have a headless connection to the Raspberry Pi can save the user from the hassle of wires running all over and provide a clean and tidy solution. Portable hacking machines built using the Raspberry Pi can be connected to a user's laptop or desktop over the network through a headless connection through this script.

2. In Cyber or to be specific in a “Cyber Range” (virtual environment that companies can use for cyber warfare training and software development) when security professionals use many monitors and many laptops for work, this script can prove to be very useful for using many computers on one single computer.

Implementation:

→Asks the user for permission to install the open-ssh server

→If user says Y:

    Installs the open-ssh server

If the user says N:

    Terminates the program and exits

→Creates a new directory for backing up the default ssh keys

→Backs up the default ssh keys into the new directory for safety purposes

→Regenerates the ssh keys to continue using the ssh server

→Enables ssh service to start at every instance of bootup

→Enables ssh services for that instance of bootup

→Displays the status of ssh service for the user

→Displays the sshd_config file to the user

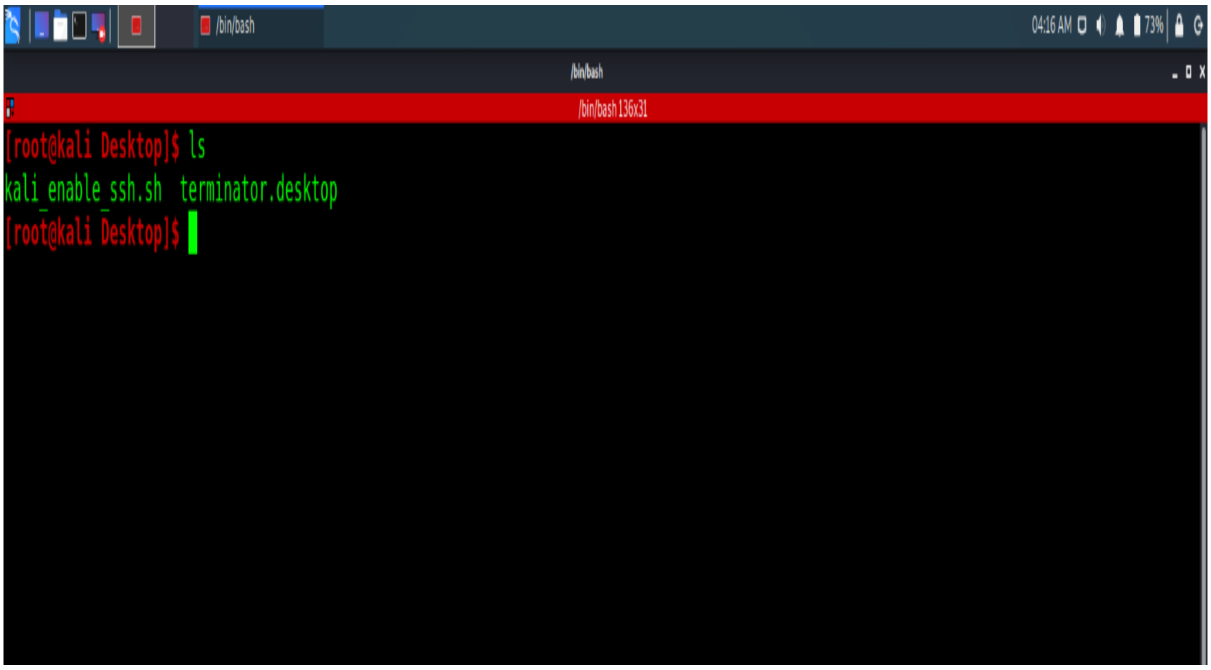
→Instructs the user about the next steps to take

→Program terminates

→User then configures the sshd_config file

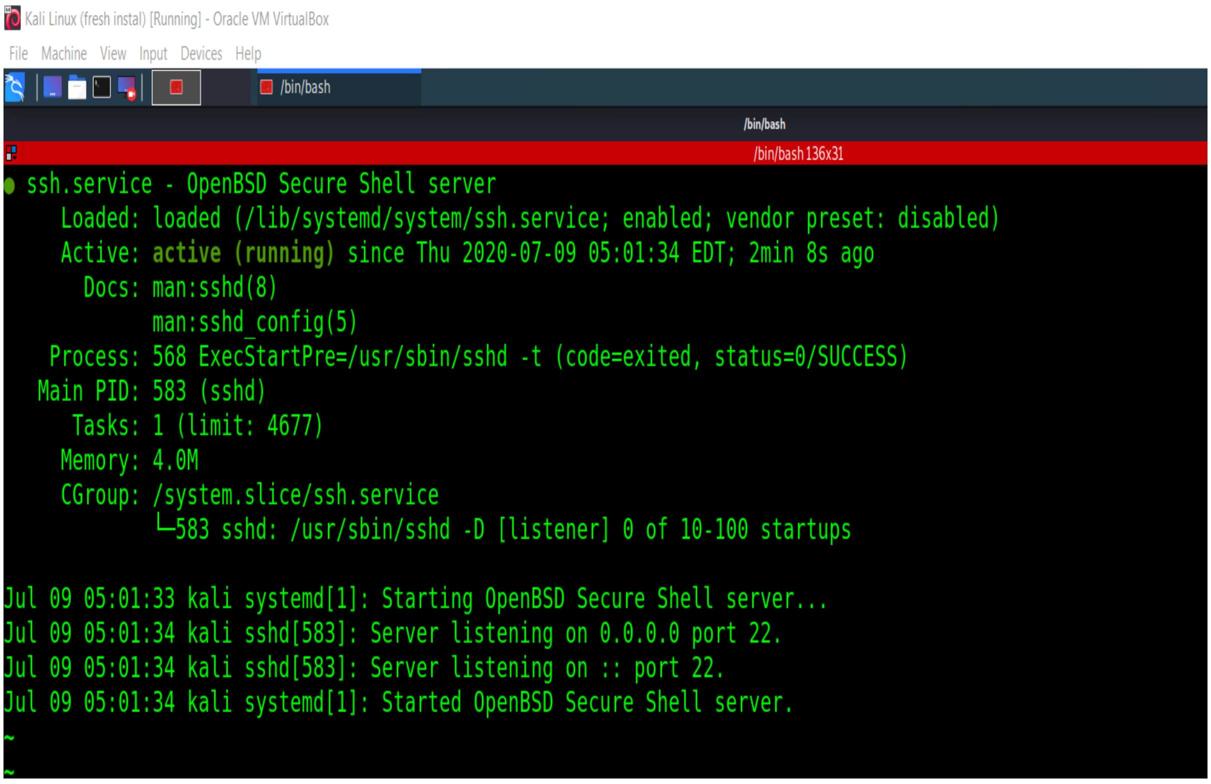
NOTE: To preserve the integrity of the system it is very important to configure the file manually.

Screenshots:



```
[root@kali Desktop]$ ls
kali_enable_ssh.sh  terminator.desktop
[root@kali Desktop]$ 
```

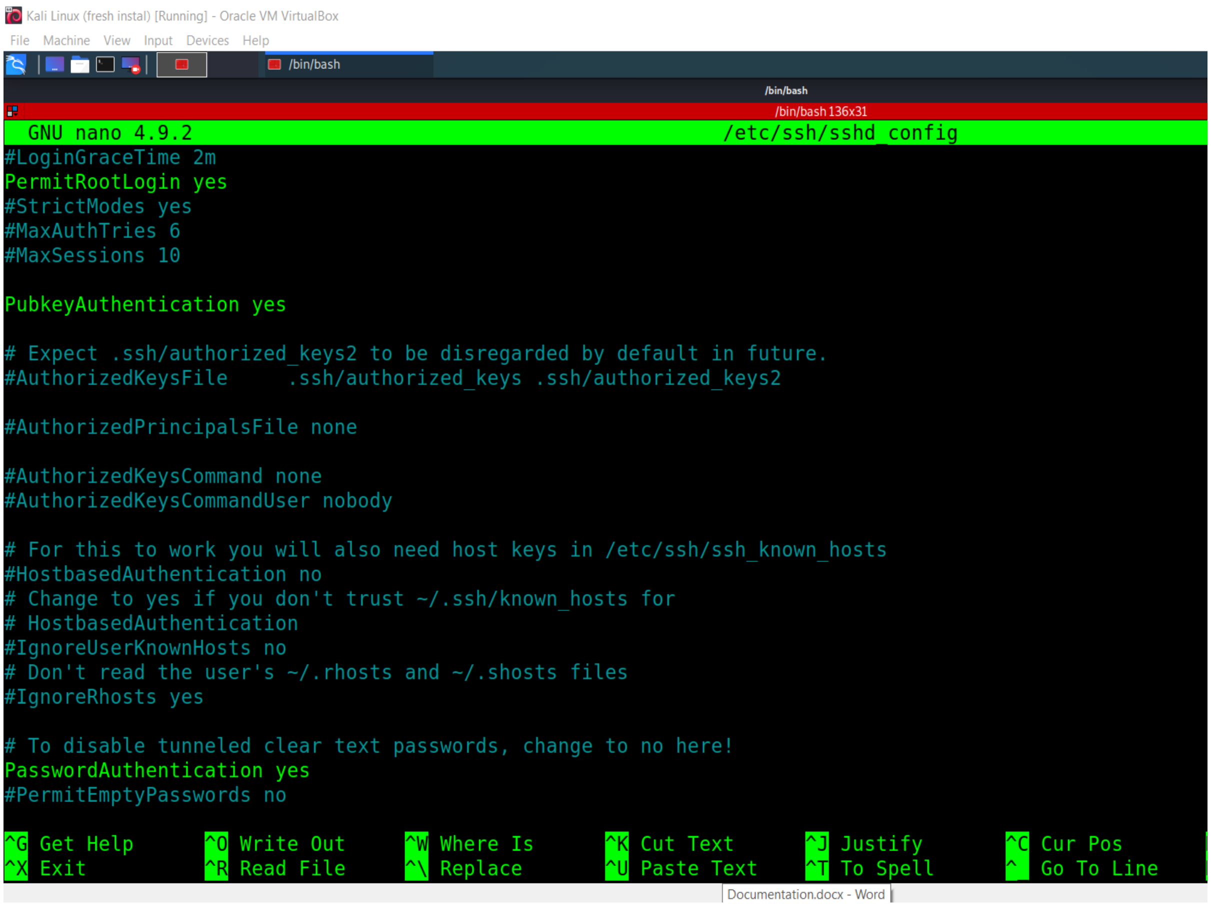
Fig. 1 Shows file stored on desktop



Kali Linux (fresh instal) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/bin/bash
/bin/bash
/bin/bash 136x31

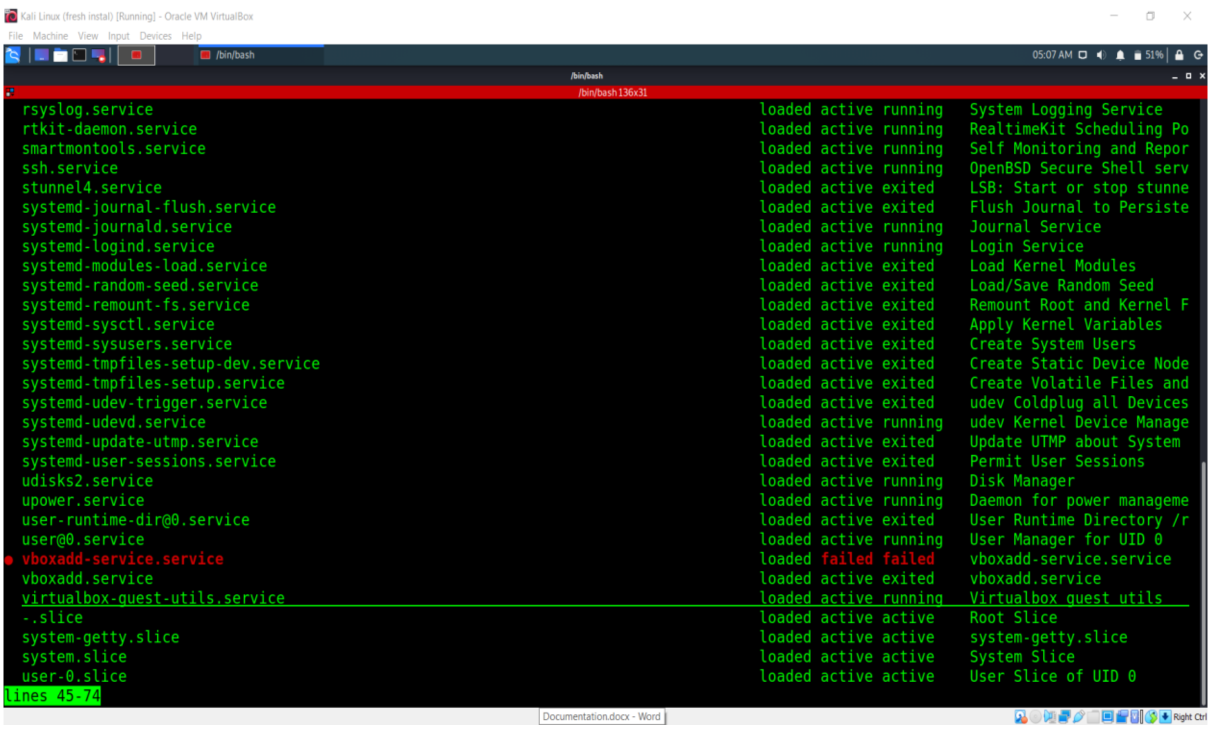
```
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: disabled)
     Active: active (running) since Thu 2020-07-09 05:01:34 EDT; 2min 8s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 568 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 583 (sshd)
      Tasks: 1 (limit: 4677)
     Memory: 4.0M
     CGroup: /system.slice/ssh.service
             └─583 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Jul 09 05:01:33 kali systemd[1]: Starting OpenBSD Secure Shell server...
Jul 09 05:01:34 kali sshd[583]: Server listening on 0.0.0.0 port 22.
Jul 09 05:01:34 kali sshd[583]: Server listening on :: port 22.
Jul 09 05:01:34 kali systemd[1]: Started OpenBSD Secure Shell server.
~
~
```

Fig.2 Shows status of the ssh server


Kali Linux (fresh instal) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/bin/bash
/bin/bash 136x31
Documentation.docx - Word

```
  GNU nano 4.9.2                                                /etc/ssh/sshd_config
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit         ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell     ^_ Go To Line
```

Fig. 3 Lines in fluorescent green changed as required


Kali Linux (fresh instal) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/bin/bash
05:07 AM
51%
/bin/bash 136x31
Documentation.docx - Word
Right Ctrl

```
  rsyslog.service                                                     loaded active running   System Logging Service
  rtkit-daemon.service                                                loaded active running   RealtimeKit Scheduling Po
  smartmontools.service                                               loaded active running   Self Monitoring and Repor
  ssh.service                                                         loaded active running   OpenBSD Secure Shell serv
  stunnel4.service                                                    loaded active exited    LSB: Start or stop stunne
  systemd-journal-flush.service                                       loaded active exited    Flush Journal to Persiste
  systemd-journald.service                                            loaded active running   Journal Service
  systemd-logind.service                                              loaded active running   Login Service
  systemd-modules-load.service                                        loaded active exited    Load Kernel Modules
  systemd-random-seed.service                                         loaded active exited    Load/Save Random Seed
  systemd-remount-fs.service                                          loaded active exited    Remount Root and Kernel F
  systemd-sysctl.service                                              loaded active exited    Apply Kernel Variables
  systemd-sysusers.service                                            loaded active exited    Create System Users
  systemd-tmpfiles-setup-dev.service                                  loaded active exited    Create Static Device Node
  systemd-tmpfiles-setup.service                                      loaded active exited    Create Volatile Files and
  systemd-udev-trigger.service                                        loaded active exited    udev Coldplug all Devices
  systemd-udevd.service                                               loaded active running   udev Kernel Device Manage
  systemd-update-utmp.service                                         loaded active exited    Update UTMP about System
  systemd-user-sessions.service                                       loaded active exited    Permit User Sessions
  udisks2.service                                                     loaded active running   Disk Manager
  upower.service                                                      loaded active running   Daemon for power manageme
  user-runtime-dir@0.service                                          loaded active exited    User Runtime Directory /r
  user@0.service                                                      loaded active running   User Manager for UID 0
● vboxadd-service.service                                             loaded failed failed    vboxadd-service.service
  vboxadd.service                                                     loaded active exited    vboxadd.service
  virtualbox-guest-utils.service                                      loaded active running   Virtualbox guest utils
  -.slice                                                             loaded active active    Root Slice
  system-getty.slice                                                  loaded active active    system-getty.slice
  system.slice                                                        loaded active active    System Slice
  user-0.slice                                                        loaded active active    User Slice of UID 0
lines 45-74
```

Fig.4 ssh service added to start up programs

Happy Hacking!!!!