

## Kali Update & Hardening



Written By Jeffery Waldron  
@VoidXDcypher

## Block 1 – Project Title

Kali Linux 2025.2 Update & UFW Hardening

## Block 2 – Objective

The objective of this project is to update all Kali Linux 2025.2 packages and tools to their latest versions and to install and configure the Uncomplicated Firewall (UFW) for baseline network hardening. This ensures the virtual machine is not only up-to-date and stable but also protected by a simple, effective firewall configuration before further penetration testing or lab activities are performed.

## Block 3 – Asset/Environment Details

**Host Machine:** Acer Nitro 5

**Host OS:** Windows 11

**Virtualization Platform:** VMware Workstation Player 16

**Guest OS:** Kali Linux 2025.2 (x64)

**Baseline Image:** Verified in BAR-1

**Firewall Tool:** Uncomplicated Firewall (UFW)

## Block 4 – Build Procedure (Summary)

At 17:04, I began updating the Kali Linux 2025.2 virtual machine by running:

`sudo apt update && sudo apt upgrade.` (See Fig. 1)

The update process identified 419 packages requiring upgrades. (See Fig. 2) I chose not to use the `-y` flag because I wanted to manually confirm any prompts during the installation. By the time the download phase reached approximately 50%, I realized that the total download size could be significant, which made me concerned about exceeding my 25 GB mobile hotspot limit. At that point, I decided to allow the process to continue rather than risk breaking dependencies by stopping midway. (See Fig. 3)

While monitoring the update, I noticed that the virtual machine's screen timeout could still trigger during inactivity. I attempted to adjust the screen blanking time within Kali by using Settings Manager → Power Manager and setting the timeout to 60 minutes. However, I later realized that these settings did not override VMware's behavior. The screen continued to

blank at the default interval, but I decided not to interfere with the process until the upgrade was complete. This issue will be addressed by adjusting the VM-level settings directly from VMware before future updates.

The GET phase of the update completed at 18:00, and the unpacking and installation of packages began. (See Fig. 4) This stage ran faster but still required close monitoring. Given that this process took place after an 11-hour driving shift, I was aware that it was cutting into my limited rest window. I allowed the installation to continue to avoid leaving the system in a partially upgraded state. The full update was successfully completed at 18:15 with all 419 packages upgraded. (See Fig. 5)

When I performed a similar update on this VM in April using a standard home Wi-Fi connection, the entire process (download and installation) completed in roughly 30 minutes. In contrast, using a mobile hotspot with limited bandwidth and higher latency significantly increased the total time and highlighted how inefficient mobile hotspots can be for large package updates.

Before proceeding, I questioned whether UFW would operate as expected in a virtual machine environment, given that Kali's power settings did not influence VMware's suspend and blanking behavior. I confirmed through targeted AI assistance that UFW operates at the guest operating system level and will filter traffic within Kali, independent of VMware's controls. This verification step is shown in the reference screenshot (Fig. 6).

Following the update, I installed UFW (Uncomplicated Firewall) to establish baseline network filtering inside the Kali Linux virtual machine. The installation process was performed using:

```
sudo apt install ufw
```

UFW was successfully installed and configured with its default rule sets, which include new configuration files for IPv4 and IPv6 rules.

I then enabled the firewall using:

```
sudo ufw enable
```

This set UFW to active mode and ensured it will start automatically with the system.

Finally, I verified the current firewall state with:

```
sudo ufw status verbose
```

The output confirmed that UFW was active, with default policies set to deny incoming traffic and allow outgoing traffic. These defaults are a strong starting point for hardening because they block all unsolicited inbound connections, which reduces the attack surface, while still allowing the VM to make necessary outbound connections for updates, downloads, and tool usage. (See Fig. 7)

As part of this review, I confirmed that these default UFW rules do not interfere with OpenVPN or TryHackMe connections, since outbound VPN traffic is allowed by default. If more restrictive outbound rules are added in the future, I can explicitly allow OpenVPN traffic on its default port (1194/UDP) to maintain VPN connectivity.

With these updates and configurations complete, the Kali Linux VM is now fully upgraded and protected with a baseline firewall configuration, ready for future penetration testing and lab activities.

## **Block 5 – Issues Encountered**

### **Hotspot Bandwidth and Speed**

Using a mobile hotspot instead of standard Wi-Fi caused the update process to run significantly slower than expected. A similar update performed in April on standard Wi-Fi completed in about 30 minutes, while this update took over an hour due to limited bandwidth and high latency. The large number of packages (419) and unknown total size caused concern about exceeding the 25 GB hotspot limit midway through the process. I allowed the upgrade to continue, but this highlighted the need for verifying update sizes and using a more stable network when possible.

### **Screen Timeout and VM Control**

I initially attempted to adjust Kali's screen blanking settings through Settings Manager → Power Manager, increasing the timeout to 60 minutes. However, I discovered that these changes did not affect VMware's control over the virtual display. The screen continued to blank at the default interval, which caused uncertainty about whether the update would be interrupted. This will need to be fixed by adjusting VMware-level power settings before future upgrades or installations.

### **Tool Overload and Lack of Familiarity**

During the upgrade process, I realized how many tools and packages I do not yet understand. Watching the upgrade output scroll by reinforced how broad Kali's default toolset is. This creates a psychological challenge, as I feel uncomfortable trusting and installing software I don't fully recognize. I plan to address this by documenting new tools as I encounter them and building familiarity through focused, tool-specific projects.

### **Time Constraints and Fatigue**

The update process took place after an 11-hour driving shift, which meant the operation was cutting into my rest schedule. Balancing the need to complete the upgrade with the need for proper rest created additional stress and required careful time management.

Future updates will be planned for times when I can monitor the process without sacrificing sleep.

## Block 6 – Key Evidence

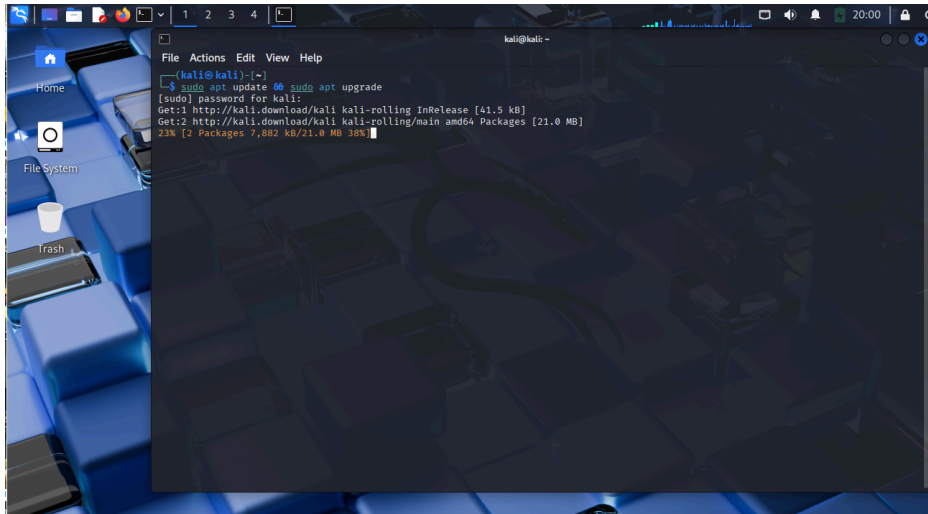


Fig. 1

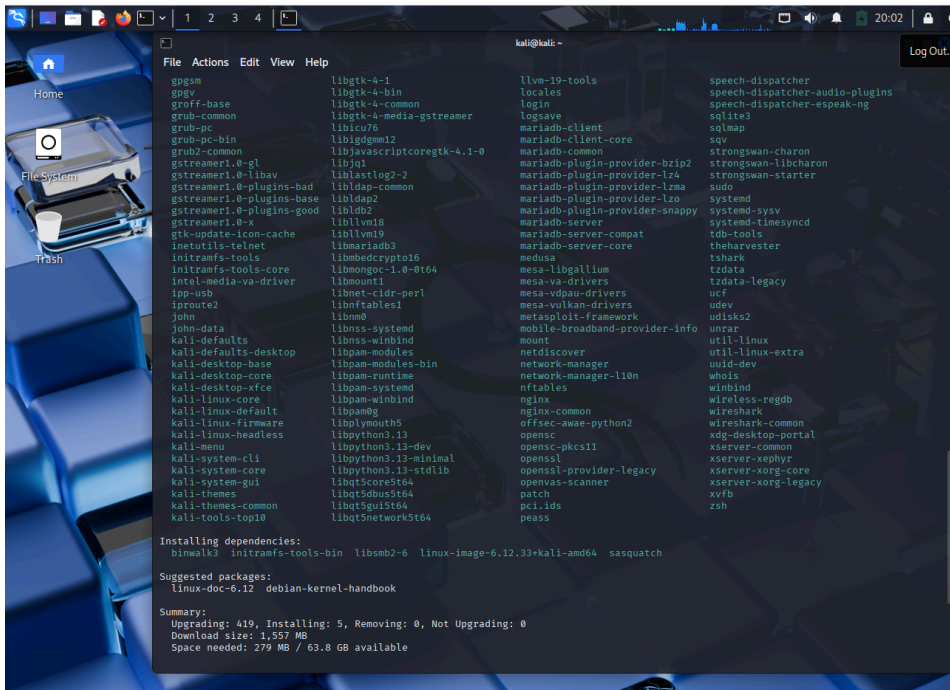


Fig 2.





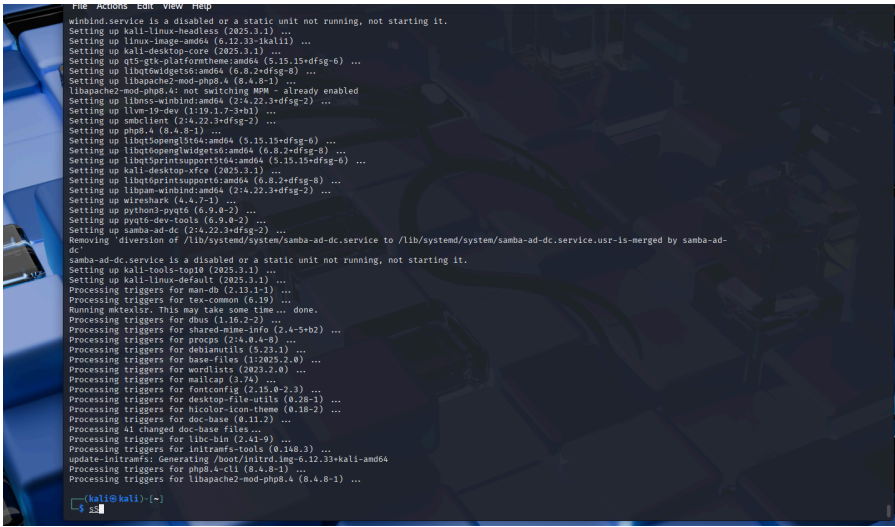


Fig. 5

question. the ufw. does it work if its a VM ? i mean if the power settings dont, how can we be sure this will?

Great question — and it's smart to ask. The short answer is: **Yes, UFW works inside a VM**, but its scope is limited to the **guest OS (Kali)** rather than the **host (Windows)** or VMware itself.

Why UFW Works in a VM

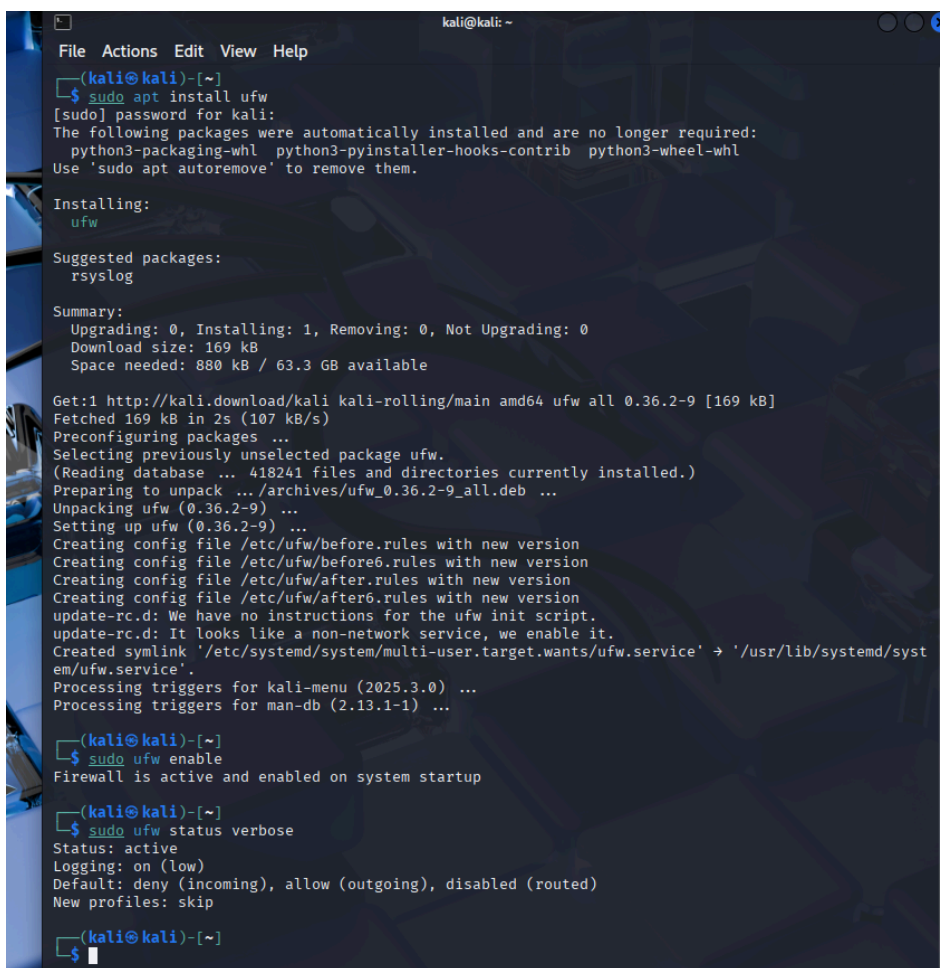
- **Kali is its own operating system:** The VM has its own network stack. When UFW is enabled inside Kali, it filters **traffic entering/leaving that VM**, just like it would on a physical machine.
- **VMware just provides the virtual hardware:** It passes traffic through the VM's virtual NIC, but the **guest OS firewall rules (UFW)** still apply.

Key Difference vs. Power Settings

- **Power settings** are more complicated because VMware controls the "pause/suspend" behavior. Kali's settings can't stop VMware from blanking or suspending the VM.
- **Firewall rules, however, are software-level inside the OS.** VMware doesn't override or interfere with UFW rules — unless you misconfigure network mode (e.g. NAT vs. Bridged), but that's a separate issue.



Fig. 6

A terminal window titled 'kali@kali' showing the installation of UFW. The user runs 'sudo apt install ufw'. The terminal output shows that several python3 packages were automatically installed and are no longer required. It then proceeds to install ufw, showing the download size (169 kB) and space needed (880 kB). The installation process includes creating config files, setting up the service, and processing triggers. Finally, the user runs 'sudo ufw enable', which activates the firewall, and 'sudo ufw status verbose', which shows the firewall is active with logging on (low) and default deny policies.

```
(kali@kali)-[~]
$ sudo apt install ufw
[sudo] password for kali:
The following packages were automatically installed and are no longer required:
python3-packaging-whl python3-pyinstaller-hooks-contrib python3-wheel-whl
Use 'sudo apt autoremove' to remove them.

Installing:
ufw

Suggested packages:
rsyslog

Summary:
Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 0
Download size: 169 kB
Space needed: 880 kB / 63.3 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 ufw all 0.36.2-9 [169 kB]
Fetched 169 kB in 2s (107 kB/s)
Preconfiguring packages ...
Selecting previously unselected package ufw.
(Reading database ... 418241 files and directories currently installed.)
Preparing to unpack .../archives/ufw_0.36.2-9_all.deb ...
Unpacking ufw (0.36.2-9) ...
Setting up ufw (0.36.2-9) ...
Creating config file /etc/ufw/before.rules with new version
Creating config file /etc/ufw/before6.rules with new version
Creating config file /etc/ufw/after.rules with new version
Creating config file /etc/ufw/after6.rules with new version
update-rc.d: We have no instructions for the ufw init script.
update-rc.d: It looks like a non-network service, we enable it.
Created symlink '/etc/systemd/system/multi-user.target.wants/ufw.service' -> '/usr/lib/systemd/system/ufw.service'.
Processing triggers for kali-menu (2025.3.0) ...
Processing triggers for man-db (2.13.1-1) ...

(kali@kali)-[~]
$ sudo ufw enable
Firewall is active and enabled on system startup

(kali@kali)-[~]
$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

(kali@kali)-[~]
$
```

Fig. 7

## Block 7 – Results

The Kali Linux 2025.2 virtual machine is now fully updated and configured with a baseline firewall using UFW. All 419 packages were successfully upgraded, and UFW is active with default policies set to deny all incoming traffic and allow all outgoing traffic. These changes have reduced the system's attack surface while maintaining necessary connectivity for updates and lab activities such as OpenVPN and TryHackMe.

The process of performing this update highlighted key operational challenges, including network inefficiency when using a mobile hotspot, the need to adjust VMware-level settings for screen management, and the importance of becoming familiar with Kali's preinstalled toolset. Despite these issues, the system upgrade and firewall configuration were completed without errors or service interruptions.

The VM is now stable, current, and ready to serve as the baseline platform for future penetration testing projects, hardening exercises, and workflow development.



## Block 8 – Next Steps

1. Install and configure OpenVPN on the Kali Linux VM.
2. Connect to TryHackMe from the VM to verify VPN compatibility and begin lab-based exercises.
3. Explore and document key Kali tools that are relevant to upcoming TryHackMe rooms, focusing on practical usage rather than the full toolset.
4. Plan BAR-3 to cover additional firewall customization or new hardening steps as needed.