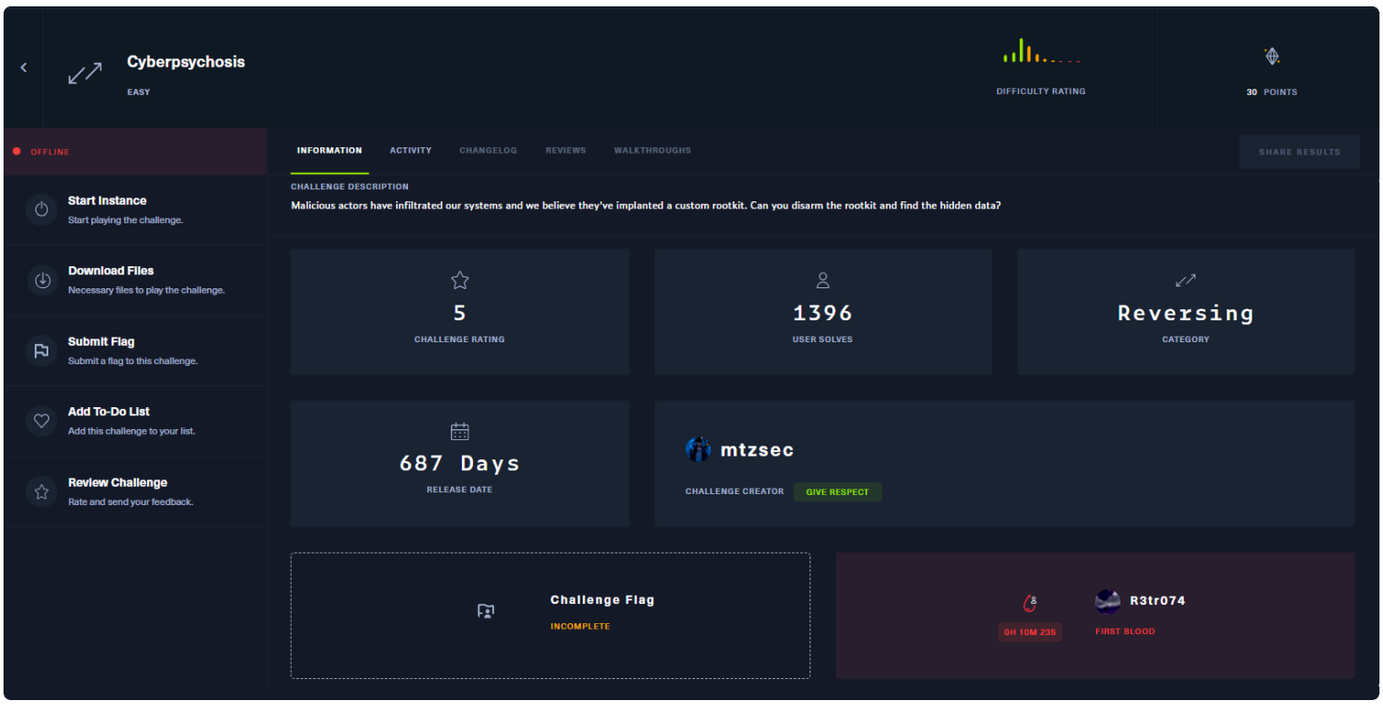
HTB-Cyberpsychosis



Attached File(s): `diamorphine.ko`

Static Analysis — Recon in Ghidra

Disassembling `diamorphine.ko` reveals a heavily customized `hacked_kill()` function, acting as a backdoor signal dispatcher. This is where the rootkit listens for specific `kill` signals to toggle functionality.

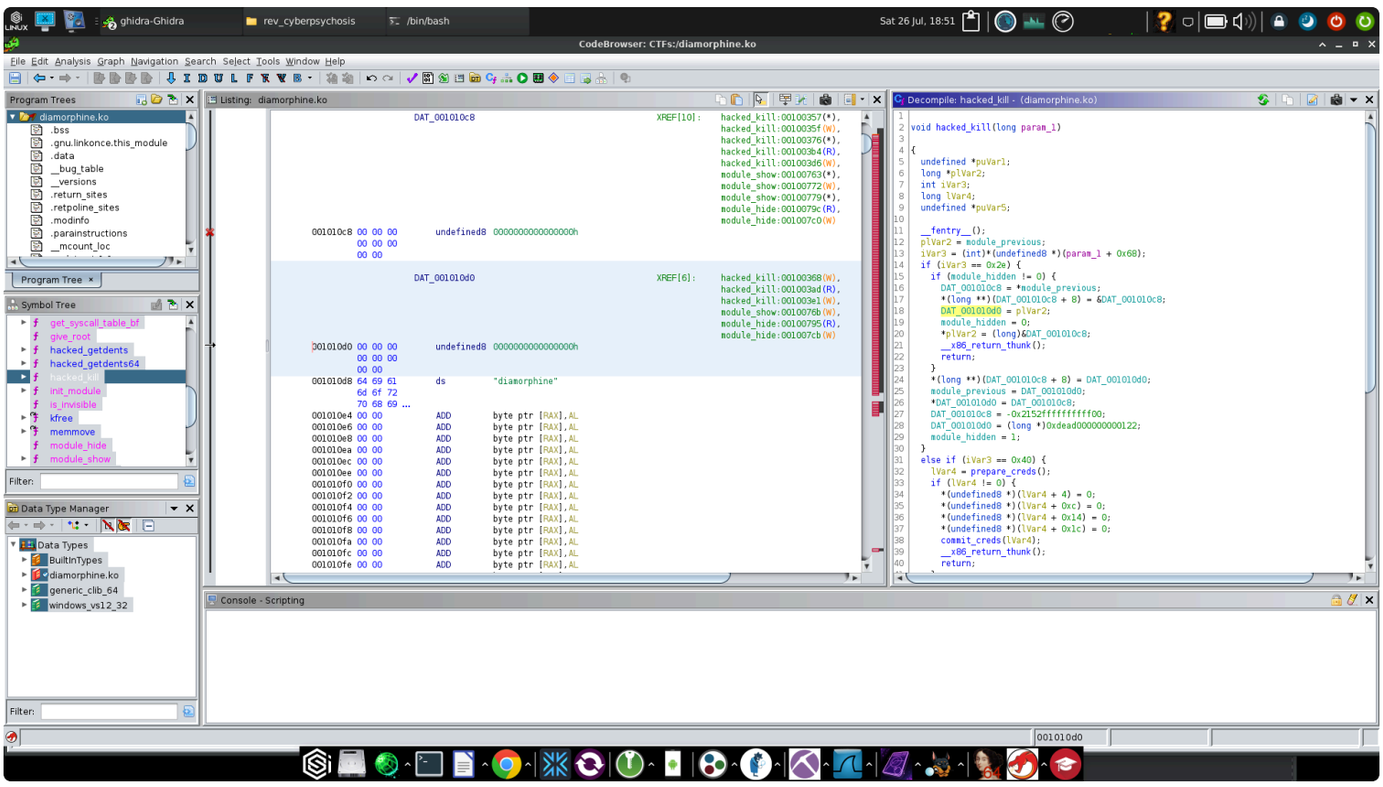
Signal-based logic in `hacked_kill`:

Signal	Hex	Purpose
64	`0x40`	Escalates current process to root
46	`0x2e`	Toggles module visibility (`/proc/modules`)
31	`0x1f`	Flips a process-hiding bit flag (unused here)

From the disassembly and pseudocode:

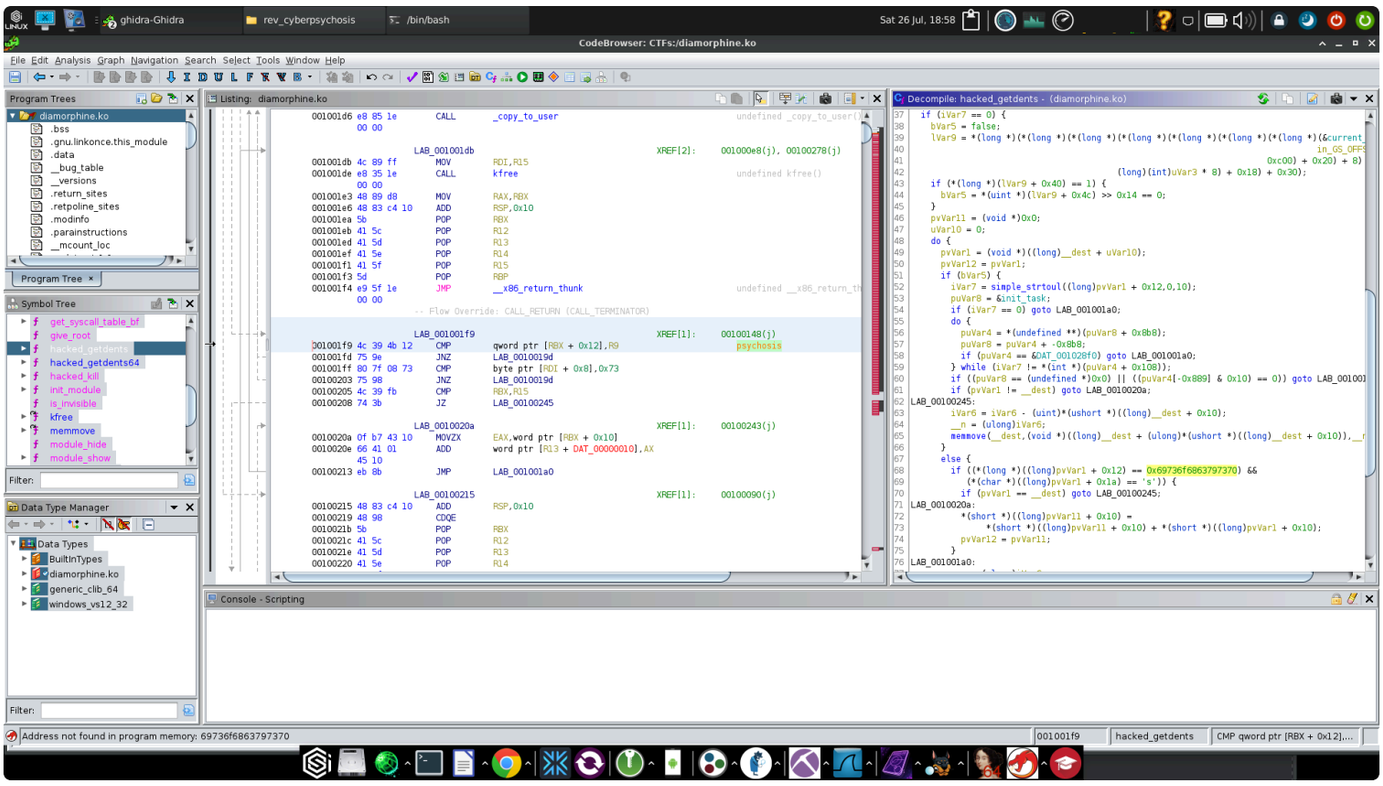
```
if (signal == 64)
    commit_creds(prepare_creds());  // privilege escalation

if (signal == 46)
    module_hidden ^= 1;             // hide/unhide kernel module
```

The module uses `module_hidden`, `module_previous`, and `DAT_001010c8/d0` to manually unlink and relink itself from the kernel's module list.



Hooks and Stealth – `hacked_getdents`

The rootkit's stealth comes from another hooked syscall: `getdents64()`, reimplemented as `hacked_getdents`. This function intercepts directory listings and filters out suspicious entries—specifically those matching the name **"psychosis"**.



So even if a file or folder exists—like a flag inside `/opt/psychosis`—you won't see it with `ls`, `find`, or `readdir()`-based tools. The system will pretend it's not there unless the module is removed or the hook is bypassed.

Connecting to the instance confirms the expected behavior:

```
whoami            → unknown uid 1000
ls /opt           → shows nothing
cat /proc/modules → empty
```

Step 1: Root Access via Signal

```
kill -64 $$
whoami → root
```

We use `kill` to send signal 64 to the current shell's PID (`$$`), and the module uses `prepare_creds()` + `commit_creds()` to elevate privileges. You're root now—but the hooks remain.

Step 2: Make the Rootkit Visible

Even with root:

```
cat /proc/modules → still blank
```

So we toggle visibility:

```
kill -46 0
```

Here, `kill -46 0` sends signal 46 to the entire process group. The rootkit's `hacked_kill()` function flips the `module_hidden` flag, relinking the module into the list.

```
cat /proc/modules
→ diamorphine 16384 0 - Live 0xffffffff...
```

Now we can remove it.

Step 3: Remove the Module

```
rmmod diamorphine
```

This undoes all the syscall hooking. The system is returned to a normal, unfiltered state. From here, the real filesystem reveals itself.

```
ls /opt
→ psychosis
```

Step 4: Grab the Flag

```
drwxr-xr-x   13 root     root           280 Jul 27 00:14 ..
~ # cat /proc/modules
cat /proc/modules
diamorphine 16384 0 - Live 0xffffffffc0123000 (OE)
~ # rmmod diamorphine
rmmod diamorphine
~ # cat /proc/modules
cat /proc/modules
~ # ls /home
ls /home
~ # ls /opt
ls /opt
psychosis
~ # find / -type f -name "*.txt"
find / -type f -name "*.txt"
/opt/psychosis/flag.txt
~ # cat /opt/psychosis/flag.txt
cat /opt/psychosis/flag.txt
HTB{N0w_Y0u_C4n_S33_m3_4nd_th3_r00tk1t_h4s_b33n_sUcc3ssfully_d3f34t3d!!}
~ # ^C
```

`HTB{N0w_Y0u_C4n_S33_m3_4nd_th3_r00tk1t_h4s_b33n_sUcc3ssfully_d3f34t3d!!}`