

```

from qiling import Qiling
from qiling.const import QL_VERBOSE

seen_syscalls = set()

def try_read_str(ql, ptr):
    try:
        return ql.mem.string(ptr)
    except:
        return f"(unreadable @ {ptr:#x})"

def hook_all_syscalls(ql, syscall_id, args):
    name = ql.os.resolve_syscall_name(syscall_id)
    if name not in seen_syscalls:
        seen_syscalls.add(name)
    resolved = []
    for arg in args:
        if isinstance(arg, int):
            s = try_read_str(ql, arg)
            resolved.append(s if "/" in str(s) else f"{arg:#x}")
        else:
            resolved.append(str(arg))
    print(f"[SYSCALL] {name} ({', '.join(resolved)})")

def main():
    elf = "e2d97525cffd1220bc5427f25dcc24137a0bcc6fd9a54e60fe94b15eba32eaad.elf"
    rootfs = "/home/sg/qiling_rootfs/arm_linux"

    ql = Qiling([elf], rootfs, verbose=QL_VERBOSE.DEBUG)

    ql.os.set_syscall("*", hook_all_syscalls)

    ql.add_fs_mapper("/lib", "/home/sg/qiling_rootfs/arm_linux/lib")
    ql.add_fs_mapper("/usr/lib", "/usr/lib")

    ql.add_fs_mapper("/usr/arm-linux-gnueabi/lib", "/usr/arm-linux-gnueabi/lib")

    print(f"[INFO] Starting emulation of {elf}")
    try:
        ql.run()
    except Exception as e:
        print(f"[CRASH] Emulation stopped due to error: {e}")

if __name__ == "__main__":

```

```
main()
```