Rootkit Features:

Directory Hiding

- *ls /tmp and ls /dev/shm do not show secret. Other entries are unaffected*
- *Implemented via a kretprobe on getdents64, filtering out any entries of protected paths (modify the prev d_reclen to prev d_reclen + hidden d_reclen)*

Process hiding

- *Designated processes are invisible to ps, top, ls /proc and anything reading /proc.*
- *All processes with GID 1337 are blocked and other processes can be blocked via PID*
- *Implemented via a kretprobe on getdents64, filtering out reads from /proc that match a protected pid or a process with the protected GID*

Path protection with traversal handling

- *Prevents opens of any file within /tmp/secret and /dev/shm/secret*
- *Implemented via an ftrace hook on sys_openat2 that checks for the hidden dirs using strstr*

Operator Bypass

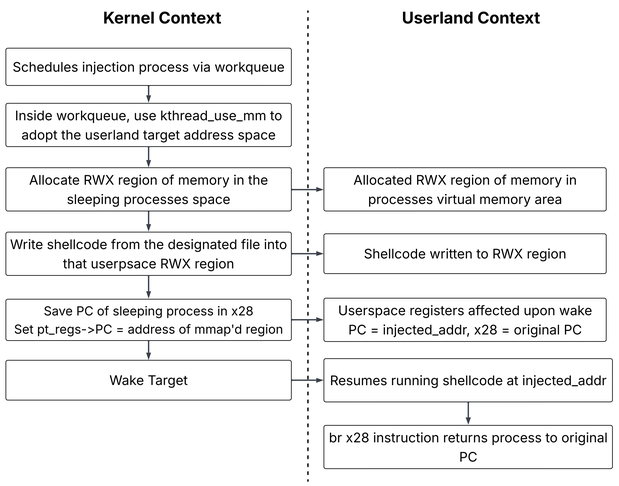
- *Any process with GID 1337 can bypass viewing hidden files, directories, and processes*
- *Implemented via has_magic_gid helper, checking if the current calling process has GID 1337*

Reverse Shell Spawning

- *Spawns a reverse shell back to our C2 operator on port 4445*
- *Implemented via a call_usermodehelper that simply runs a bash command that redirects stdin and stdout to /dev/tcp/C2_IP/4445.*
- *Also includes PTY upgrade trick that allows terminal to run clear, ^C, etc.*

Kernel to User-space Shellcode injection

- *Can inject shellcode into a sleeping process given their PID, keeping the process alive*

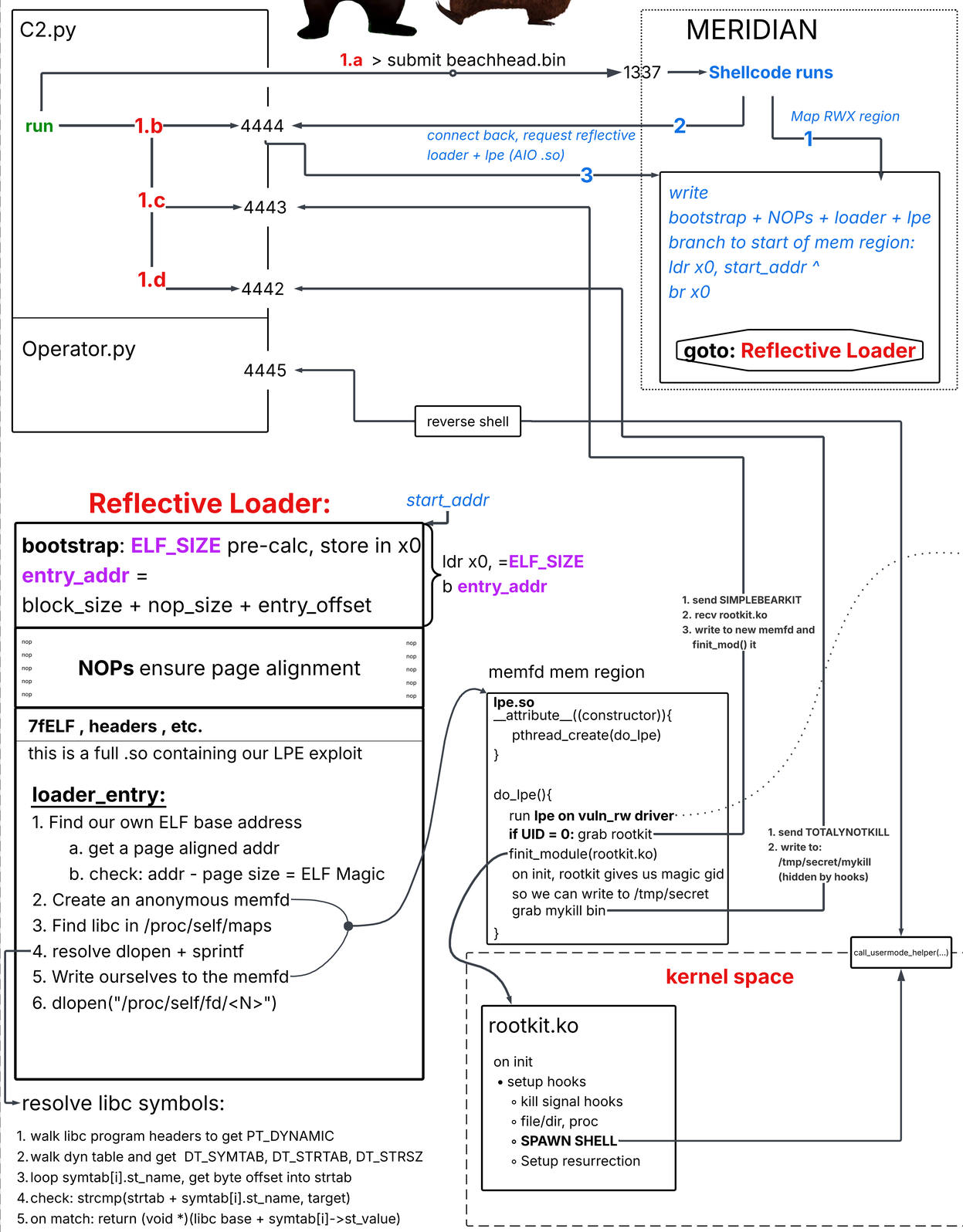


Module Hiding

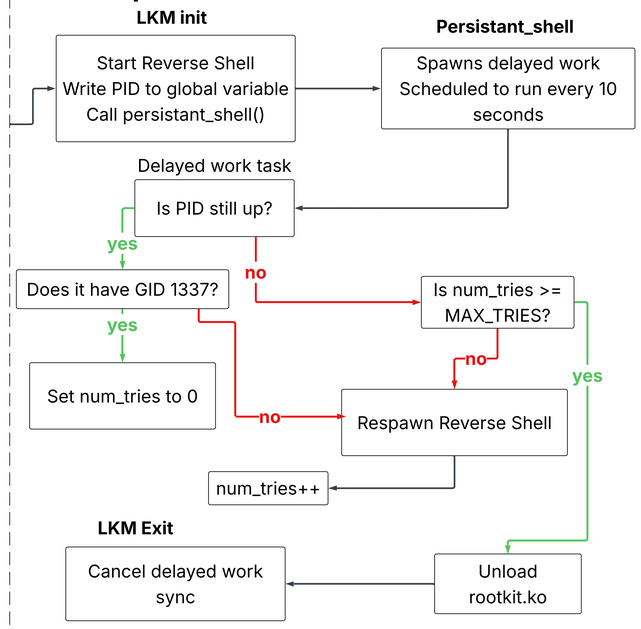
- *Rootkit does not appear in lsmod or /proc/modules*
- *Implemented by unlinking from the kernel module doubly-linked list*

Cleanup

- *rmmod deletes the directories /tmp/secret and /dev/shm/secret from the disk before removing itself*



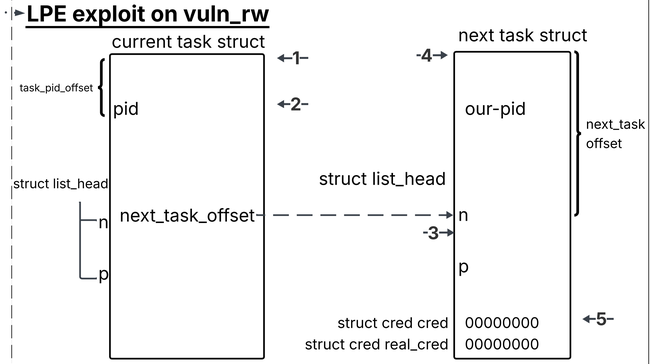
Special Feature
Operator Resurrection



Covert C2 Communication

- *Wraps the kill syscall to issue commands to the rootkit without a network connection*
- *Supports:*
  - *Query status*
  - *Toggle directory hiding*
  - *Toggle file access blocking*
  - *Toggle process hiding*
  - *Mark a PID as hidden*
  - *Assign GID 1337 to a process by PID*
  - *Trigger shellcode injection given a binary*
  - *Toggle module hiding*
  - *Spawn a reverse shell*

LPE exploit on vuln_rw



1. current = first task in doubly linked list
   do
   {
2. pid = cur + pid_offset
   is it our pid?
3. next_task = cur + next_task_offset
4. cur = next_task - next_task_offset
   } while (cur != Init_task)

5. Once we have the task, 0 out cred_structs cred, real_cred using offsets

Limitations

Module Hiding - incomplete:

- What works: unlinking from the kernel module doubly-linked list hides the rootkit from lsmod and /proc/modules
- What doesn't: the kobject entry in sysfs is not removed, so /sys/module/rootkit/ still exists and is visible
- Why: In order to get this working we would have to delete the kobject from the kernel's bookkeeping, dealing with parents and references that come from that

SIMPLE BEAR rootkit presented by: Alp Berrak, Arjun Uppal and Ethan Friedman!