Capability Exploration Lab

Computer Security

Capability Fundamentals

- Capabilities exist on a per-thread basis
- Capabilities assigned to a thread in the following ways:
  - Capabilities loaded from the file system
    - eg: Process loaded from file systems
  - Capabilities inherited from parent process
    - eg: using execve() after fork

Capability Fundamentals

- Capabilities are set on a **per-thread** basis
- For all ‘privileged’ operations, the kernel checks if the thread has the capability in its “Effective” set (Explained later)
- Capabilities are gained by a process/thread when that process/thread is launched from the file system with the capabilities attached to the file, i.e the file system contains the list of capabilities attached to the file.

Capability Fundamentals

- Each thread has 3 sets of capabilities:
  - **Permitted:** Superset of capabilities. If it does not exist in this set, the thread can never acquire the capability (unless it has the CAP_SETPCAP capability)
  - **Inheritable:** Capability set maintained after running execve(). This set becomes the Permitted set of the new process after execve()
  - **Effective:** This is the set of capabilities used by the kernel to perform permission checks on the thread.
- fork() makes a child get copies of parents' capability set

Capability Fundamentals

- Each file has 3 sets of capabilities:
  - **Permitted:** Automatically permitted to the thread regardless of the threads' inheritable capabilities
  - **Inheritable:** This set is ANDed with the threds inheritable set to determine the resultant inheritable set (assigned to thread after execve())
  - **Effective:** This, unlike thread capabilities, is a single bit. After execve() –
    - If the bit is set, effective set = (effective set U permitted set).
    - If unset, no new permitted capability is in the new effective set.

Capability related commands

- setcap
  - Used to assign capabilities to files
- getcap
  - Used to get the capabilities carried by a file
- getpcaps
  - Used to get capabilities carried by a process
- Run “man setcap”, “man getcap” and “man getpcap”
- Also visit to see the syntax of http://manpages.ubuntu.com/manpages/natty/man3/cap_from_text.3.html

Notes about the Tasks

- It is your job to figure out the syntax of the commands
- The libcap directory is located under /home/seed/temp/
- Refer to page 4 of the lab description for an explanation on how capabilities are set
- The functions to be added, as well as the user_cap.c required for Task 2 are at http://www.cis.syr.edu/~wedu/seed/Labs/Capability_Exploration/

Some hints for the tasks

- cap_dac_read_search
  - Try to open a file owned by root as the normal user
- cap_dac_override
  - try to write to a file owned by root, using a normal user
- cap_chown
  - “chown”
- cap_setuid
  - setuid()
- cap_kill
  - “kill”
- cap_net_raw
  - “ping”