A0216695U

Task 1

```python
log_file = 'task1.log'

accessed_files = {}

with open(log_file, 'r') as f:
    for row in f.readlines():
        if 'malicious-prog' in row.lower():
            prog = row.split()[-1:][0]
            if prog in accessed_files.keys():
                accessed_files[prog] += 1
            else:
                accessed_files[prog] = 1

sorted_accessed_files = sorted(accessed_files.items(),
        key=lambda x: (-x[1], x[0])
        )

# Get the top 10 path accesses
top_10_paths = sorted_accessed_files[:10]

# Print the top 10 path accesses and their counts
for path, count in top_10_paths:
    print(f'{path} {count}')
```

The program first reads each line of the log file and checks if the malicious-prog was in each line. If it is, then take the accessed file, add it to the key of the dictionary accessed_files and add 1 to the counter which is the value of the key. Then sort the dictionary in descending using the counter first then ascending alphabetically by using the file name within each count. Then cut the list to only the first 10 items.

Task 2

My lsm hook attaches to file_open. My eBPF program only prevents malicious-prog by first getting the current task running. Then accessing the task_struct with &task->comm will get the filename of the current executing task. I then compare that against the string “malicious-prog”. Then only if the current task is “malicious-prog”, the comparison of filename will happen to prevent the sensitive files from being read.

```c
task = (struct task_struct *)bpf_get_current_task();
bpf_core_read(&taskname, sizeof(taskname), &task->comm);
if (STRNCMP(taskname, prgm_to_prevent, sizeof(prgm_to_prevent)) == 0) {
        // Get the filename
        dname = BPF_CORE_READ(file, f_path.dentry, d_name);
        bpf_probe_read_kernel(buf, sizeof(buf), dname.name);

        if (STRNCMP(buf, sensitive_file1, sizeof(sensitive_file1)) == 0 || STRNCMP(buf, sensitive_file2, sizeof(sensitive_file2)) == 0) {
                bpf_printk("Access to sensitive file %s is detected.\n", buf);
                // Return -EACCES to deny access
                return -EACCES;
        }
}
```

```
Successfully started! Please run `sudo cat /sys/kernel/debug/tracing/trace_pipe` to see output of the BPF programs.
....
```

```
copied "/usr/include/linux/if_packet.h" to "./output/105-if_packet.h"
copied "/usr/include/linux/atmlec.h" to "./output/106-atmlec.h"
copied "/usr/include/linux/netfilter_bridge/ebt_802_3.h" to "./output/107-ebt_802_3.h"
copied "/usr/include/linux/zorro_ids.h" to "./output/108-zorro_ids.h"
copied "/usr/include/linux/reiserfs_xattr.h" to "./output/109-reiserfs_xattr.h"
filesystem error: filesystem error: cannot copy: Permission denied [/usr/include/linux/if.h] [./o
utput/if.h]
copied "/usr/include/linux/kcm.h" to "./output/110-kcm.h"
copied "/usr/include/linux/tc_act/tc_nat.h" to "./output/111-tc_nat.h"
copied "/usr/include/linux/wmi.h" to "./output/112-wmi.h"
copied "/usr/include/linux/ppp_defs.h" to "./output/113-ppp_defs.h"
copied "/usr/include/linux/tcp_metrics.h" to "./output/114-tcp_metrics.h"
copied "/usr/include/linux/netfilter/xt_rateest.h" to "./output/115-xt_rateest.h"
copied "/usr/include/linux/msg.h" to "./output/116-msg.h"
```

```
copied "/usr/include/linux/virtio_pmem.h" to "./output/175-virtio_pmem.h"
copied "/usr/include/linux/if_eql.h" to "./output/176-if_eql.h"
copied "/usr/include/linux/cryptouser.h" to "./output/177-cryptouser.h"
filesystem error: filesystem error: cannot copy: Permission denied [/usr/include/linux/un.h] [./o
utput/un.h]
copied "/usr/include/linux/if_infiniband.h" to "./output/178-if_infiniband.h"
copied "/usr/include/linux/inet_diag.h" to "./output/179-inet_diag.h"
copied "/usr/include/linux/netfilter/xt_addrtype.h" to "./output/180-xt_addrtype.h"
copied "/usr/include/linux/tc_act/tc_bpf.h" to "./output/181-tc_bpf.h"
copied "/usr/include/linux/jffs2.h" to "./output/182-jffs2.h"
copied "/usr/include/linux/stm.h" to "./output/183-stm.h"
copied "/usr/include/linux/usb/audio.h" to "./output/184-audio.h"
copied "/usr/include/linux/netfilter/xt_MARK.h" to "./output/185-xt_MARK.h"
```

```
  malicious-prog-5377    [002] d...1 12921.876602: bpf_trace_printk: Access to sensitive file un.
h is detected.

  malicious-prog-5377    [002] d...1 12922.722819: bpf_trace_printk: Access to sensitive file if.
h is detected.

  malicious-prog-5377    [002] d...1 12928.476572: bpf_trace_printk: Access to sensitive file if.
h is detected.

  malicious-prog-5377    [002] d...1 12931.183763: bpf_trace_printk: Access to sensitive file un.
h is detected.

  malicious-prog-5377    [002] d...1 12934.167843: bpf_trace_printk: Access to sensitive file if.
h is detected.

  malicious-prog-5377    [002] d...1 12939.858485: bpf_trace_printk: Access to sensitive file if.
h is detected.

  malicious-prog-5377    [002] d...1 12940.374487: bpf_trace_printk: Access to sensitive file un.
h is detected.
```