NSS ASSIGNMENT - 1

NAME : SATATYA DE

ROLL : MT25084

STREAM : M.TECH CSE

YEAR : 1st YEAR

TOPIC : Access Control Semantics in UNIX-Like OSes(FreeBSD)

GitHub Link : https://github.com/satatya/NSS_assignment1/tree/main

1. Final Submission Package

The submission directory contains exactly the following items (source + build configuration):

- accheck.c — metadata- and ACL-based reasoning engine (predictor).
- accheck-helper.c — setuid-root validator that switches identity to the target user and performs the requested operation.
- accheck-test-read.c — setuid-root test that drops privileges to the invoking user and then attempts to read.
- accheck-test-write.c — setuid-root test that drops privileges to the invoking user and then attempts to append.
- accheck-test-exec.c — setuid-root test that drops privileges to the invoking user and then attempts to execute/search.
- Makefile — builds all binaries on FreeBSD without modification.
- README.md — quick build/run notes and scenario summaries.

2. Build and Installation Instructions

-> Compile

Inside the project directory:

make clean

make

-> SetUID configuration (run as root)

The validator and test suite must run as setuid-root to perform controlled identity switching. After compiling, configure ownership and the SetUID bit:

chown root:wheel accheck-helper accheck-test-read accheck-test-write

accheck-test-exec

chmod 4755 accheck-helper accheck-test-read accheck-test-write

accheck-test-exec

3. Three test scenarios.

- **NFSv4 ACL override on a 640 file (secret.txt)**
  A file /srv/testlab/secret.txt that was owned by alice:labgroup with mode 640, which would normally deny acces to bob via mode bits. An NFSv4 ACL entry (u:bob:r:allow) would be added to grant Bob read access without giving write/execute. The predictor (accheck) reports **ALLOWED** for read (shows the matching ALLOW ACE) and **DENIED** for write/exec; the validator (accheck-helper) confirms the kernel enforces the same result.

- **Directory traversal restriction (notraverse/inside.txt)**
  A directory /srv/testlab/notraverse was set to 700 (only owner can traverse). Even though the file inside (inside.txt) had an ACL that would allow Bob to read it, Bob cannot *reach* the file because he lacks execute/search permission on the directory. accheck shows traversal checks on each parent directory and returns **DENIED** with the reason "directory traversal denied"; accheck-helper confirms the kernel returns **DENIED**.

- **Setuid privilege-drop test (runme.sh)**
  An executable script /srv/testlab/runme.sh was created with 755. The program accheck-test-exec is installed setuid-root but explicitly drops privileges to the real invoking user (bob) before attempting execution. When run as bob, accheck-test-exec returns **ALLOWED**, demonstrating the "drop early" secure pattern: after dropping privilege, results reflect bob's real permissions, not root.

4. Three error cases that were handled

- **Unknown user name**
  If the target user doesn't exist (e.g., typo in username), accheck detects this using getpwnam() and exits gracefully with an error instead of crashing out or guessing a UID.

- **Invalid path / stat failure**
  If the path does not exist or cannot be accessed for metadata lookup, accheck fails safely (checks stat() return value), prints the OS error (via errno/strerror), and exits without producing a misleading ALLOW/DENY.

- **Invalid operation argument**
  If the user passes an invalid operation (anything other than read, write, execute), what accheck does is prints a clear usage message and exits with a non-zero status, preventing undefined behavior or incorrect decisions.

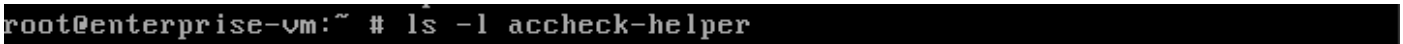
5. Screenshots

Below are the screenshots of the various scenarios I encountered during compilation of the assignment.

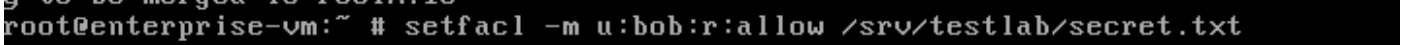
```
root@enterprise-vm:~ # mkdir -p /srv/testlab
root@enterprise-vm:~ # echo "Kernel Secret" > /srv/testlab/secret.txt
root@enterprise-vm:~ # chown alice:labgroup /srv/testlab/secret.txt
root@enterprise-vm:~ # chmod 640 /srv/testlab/secret.txt
root@enterprise-vm:~ # ./accheck alice read /srv/testlab/secret.txt
Reasoning for user : alice (UID : 1002)
File Owner UID: 1002 ¦ File Group GID : 1005
Traditional Mode : 640
PREDICTION: ALLOW
REASON: Match in Owner bits
root@enterprise-vm:~ #
```


```
root@enterprise-vm:~ # ./accheck charlie read /srv/testlab/secret.txt
Reasoning for user : charlie (UID : 1004)
File Owner UID: 1002 ¦ File Group GID : 1005
Traditional Mode : 640
PREDICTION: DENY
REASON: Denied by default logic
root@enterprise-vm:~ #
```


```
root@enterprise-vm:~ # ./accheck root read /srv/testlab/secret.txt
Reasoning for user : root (UID : 0)
File Owner UID: 1002 ¦ File Group GID : 1005
Traditional Mode : 640
PREDICTION: ALLOW
REASON: User is root (super-user)
root@enterprise-vm:~ #
```




```
root@enterprise-vm:~ # ls -l accheck-helper
-rwsr-xr-x  1 root wheel 11080 Feb 10 18:08 accheck-helper
root@enterprise-vm:~ #
```




```
root@enterprise-vm:~ # ls -l accheck-helper
-rwsr-xr-x  1 root wheel 11080 Feb 10 18:08 accheck-helper
root@enterprise-vm:~ # setfacl -m u:bob:rwx /srv/testlab/secret.txt
setfacl: /srv/testlab/secret.txt: branding mismatch; existing ACL is NFSv4, entr
y to be merged is POSIX.1e
root@enterprise-vm:~ # setfacl -m u:bob:r:allow /srv/testlab/secret.txt
root@enterprise-vm:~ # ./accheck-helper bob read /srv/testlab/secret.txt
KERNEL RESULT: ALLOW
root@enterprise-vm:~ # su - charlie -c "/root/accheck-test-read /srv/testlab/sec
ret.txt"
Using vt(4) on a laptop? Try this sh(1) function. It provides an "h"
command that prints the last 22 commands executed, the time, remaining
battery life, and current working directory:

h() { fc -l -22; printf "%s\n" "`date +%H:%M` -- `apm -l`% -- `pwd`"; }

                -- Alexander Ziaee <ziaee@FreeBSD.org>
-su: /root/accheck-test-read: Permission denied
root@enterprise-vm:~ #
```

```
root@enterprise-vm:~ # getfacl /srv/testlab/secret.txt
# file: /srv/testlab/secret.txt
# owner: alice
# group: labgroup
          user:bob:r-------------:-------:allow
            owner@:rw-p--aARWcCos:-------:allow
            group@:r-----a-R-c--s:-------:allow
         everyone@:------a-R-c--s:-------:allow
root@enterprise-vm:~ #
```


```
root@enterprise-vm:~ # ./accheck-helper bob read /srv/testlab/secret.txt
KERNEL RESULT: ALLOW
root@enterprise-vm:~ # ./accheck-helper bob write /srv/testlab/secret.txt
KERNEL RESULT: DENY
root@enterprise-vm:~ #
```


```
root@enterprise-vm:~ # ./accheck bob read /srv/testlab/secret.txt 2>&1
Checking access for user=bob uid=1003 path=/srv/testlab/secret.txt op=read
Target object: /srv/testlab/secret.txt
mode: -rw-r-----
ACL brand: NFSv4 (allow/deny order)
NFSv4 ACE #1 matched and ALLOWED overlap=0x8
ALLOWED
root@enterprise-vm:~ #
```

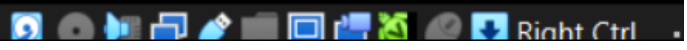

```
root@enterprise-vm:~ # ./accheck bob read /srv/testlab/secret.txt 2>&1
Checking access for user=bob uid=1003 path=/srv/testlab/secret.txt op=read
Target object: /srv/testlab/secret.txt
mode: -rw-r-----
ACL brand: NFSv4 (allow/deny order)
NFSv4 ACE #1 matched and ALLOWED overlap=0x8
ALLOWED
root@enterprise-vm:~ # ./accheck bob write /srv/testlab/secret.txt 2>&1
Checking access for user=bob uid=1003 path=/srv/testlab/secret.txt op=write
Target object: /srv/testlab/secret.txt
mode: -rw-r-----
ACL brand: NFSv4 (allow/deny order)
NFSv4 ACL: no ACEs satisfied remaining=0x10 -> DENY
DENIED
root@enterprise-vm:~ # ./accheck bob execute /srv/testlab/secret.txt 2>&1
Checking access for user=bob uid=1003 path=/srv/testlab/secret.txt op=execute
Target object: /srv/testlab/secret.txt
mode: -rw-r-----
NFSv4 note: file has no execute bits set (0111==0) -> DENY
DENIED
root@enterprise-vm:~ #
```

```
root@enterprise-vm:~ # ./accheck-test-read /srv/testlab/secret.txt
ALLOWED
root@enterprise-vm:~ # ./accheck-test-write /srv/testlab/secret.txt
ALLOWED
root@enterprise-vm:~ # ./accheck-test-exec /srv/testlab/secret.txt
DENIED
root@enterprise-vm:~ #
```

```
root@enterprise-vm:~ # sh -c 'printf "#!/bin/sh\necho RUN_OK\n" > /srv/testlab/r
unme.sh'
root@enterprise-vm:~ # chmod 755 /srv/testlab/runme.sh
root@enterprise-vm:~ # chown alice:labgroup /srv/testlab/runme.sh
root@enterprise-vm:~ # ls -l /srv/testlab/runme.sh
-rwxr-xr-x  1 alice labgroup 22 Feb 10 20:12 /srv/testlab/runme.sh
root@enterprise-vm:~ #
```


```
bob@enterprise-vm:/usr/local/src/NSS_assignment1 $ ./accheck-test-exec /srv/test
lab/runme.sh
ALLOWED
bob@enterprise-vm:/usr/local/src/NSS_assignment1 $
```

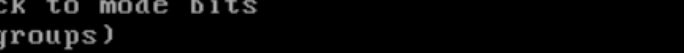

```
root@enterprise-vm:~ # chmod 700 /srv/testlab/
/srv/testlab/notraverse/ /srv/testlab/runme.sh    /srv/testlab/secret.txt
root@enterprise-vm:~ # chmod 700 /srv/testlab/notraverse/
root@enterprise-vm:~ # setfacl -m u:bob:r:allow /srv/testlab/notraverse/inside.t
xt
root@enterprise-vm:~ # getfacl /srv/testlab/notraverse
# file: /srv/testlab/notraverse
# owner: alice
# group: labgroup
            owner@:rwxp--aARWcCos:-------:allow
            group@:------a-R-c--s:-------:allow
         everyone@:------a-R-c--s:-------:allow
root@enterprise-vm:~ # getfacl /srv/testlab/notraverse/inside.txt
# file: /srv/testlab/notraverse/inside.txt
# owner: alice
# group: labgroup
           user:bob:r-------------:-------:allow
            owner@:rw-p--aARWcCos:-------:allow
            group@:r-----a-R-c--s:-------:allow
         everyone@:r-----a-R-c--s:-------:allow
root@enterprise-vm:~ #
```




```
root@enterprise-vm:~ # pw groupadd projgrp
root@enterprise-vm:~ # pw groupmod projgrp -m bob
root@enterprise-vm:~ # id bob
uid=1003(bob) gid=1003(bob) groups=1003(bob),1005(labgroup),1006(projgrp)
root@enterprise-vm:~ #
```


```
root@enterprise-vm:~ # chown alice:projgrp /srv/testlab/groupfile.txt
root@enterprise-vm:~ # chmod 640 /srv/testlab/groupfile.txt
root@enterprise-vm:~ # ls -l /srv/testlab/groupfile.txt
-rw-r-----  1 alice projgrp 10 Feb 10 22:56 /srv/testlab/groupfile.txt
root@enterprise-vm:~ #
```


```
root@enterprise-vm:~ # ./accheck bob read /srv/testlab/groupfile.txt 2>&1
Checking access for user=bob uid=1003 path=/srv/testlab/groupfile.txt op=read
Target object: /srv/testlab/groupfile.txt
mode: -rw-r-----
ACL brand: NFSv4 but trivial -> fall back to mode bits
mode class: group (matched one of user groups)
mode requires: r -> ALLOW
ALLOWED
root@enterprise-vm:~ # ./accheck-helper bob read /srv/testlab/groupfile.txt
ALLOWED
root@enterprise-vm:~ #
```



```
root@enterprise-vm:~ # ./accheck-helper bob read /srv/testlab/groupfile.txt
ALLOWED
root@enterprise-vm:~ #
```


```
root@enterprise-vm:~ # ./accheck bob read /srv/testlab/notraverse/inside.txt 2>&
1
Checking access for user=bob uid=1003 path=/srv/testlab/notraverse/inside.txt op
=read
Traversal check: /srv
mode: drwxr-xr-x
ACL brand: NFSv4 but trivial -> fall back to mode bits
mode class: other
mode requires: x -> ALLOW
Traversal check: /srv/testlab
mode: drwxr-xr-x
ACL brand: NFSv4 but trivial -> fall back to mode bits
mode class: other
mode requires: x -> ALLOW
Traversal check: /srv/testlab/notraverse
mode: drwx------
ACL brand: NFSv4 but trivial -> fall back to mode bits
mode class: group (matched one of user groups)
mode requires: x -> DENY
Result reason: directory traversal denied
DENIED
root@enterprise-vm:~ # ./accheck-helper bob read /srv/testlab/notraverse/inside.
txt
DENIED
root@enterprise-vm:~ #
```