NEU CY 5770 Software Vulnerabilities and Security

Instructor: Dr. Ziming Zhao

Race Conditions

- Occurs when multiple operations access shared state concurrently
- Outcome depends on timing/interleaving
- Leads to non-deterministic behavior

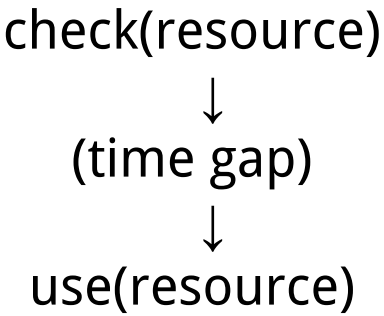
Breaks assumptions about order. Enables:

- privilege escalation
- data corruption
- bypassing checks

Hard to detect:

- may not reproduce consistently

Classic Pattern: Time-of-Check to Time-of-Use (TOCTOU)



Exploitation Strategy

- Identify:
  - check
  - use
- Create timing window (sleep, load, etc.)
- Repeatedly:
  - swap resource (file, pointer, etc.)
- Win when timing aligns

rc1

```
int main() {
    const char *path = "/tmp/raceflag";
    struct stat st;

    printf("Checking %s...\n", path);

    if (lstat(path, &st) < 0) {
        perror("lstat");
        return 1;}

    if (S_ISLNK(st.st_mode)) {
        puts("symlink not allowed");
        return 1;}

    puts("Looks safe. Processing...");
    sleep(20);  // 20s race window

    int fd = open(path, O_RDONLY);
    if (fd < 0) {
        perror("open");
        return 1;}

    char buf[64] = {0};
    read(fd, buf, sizeof(buf) - 1);
    close(fd);

    printf("Read: %s\n", buf);

    return 0;}
```

https://github.com/xairy/linux-kernel-exploitation/tree/master