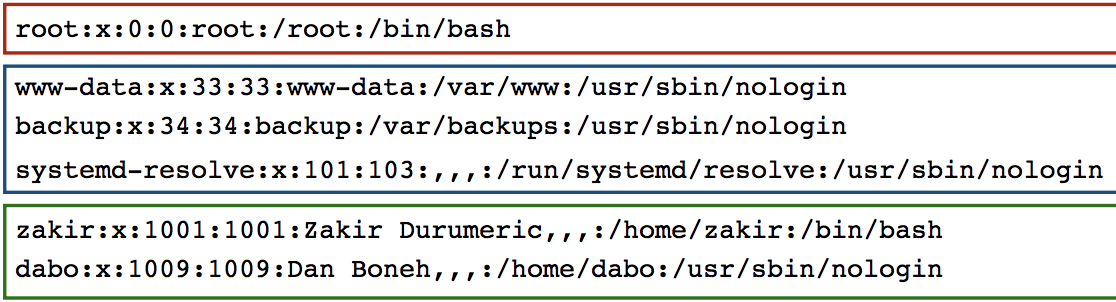
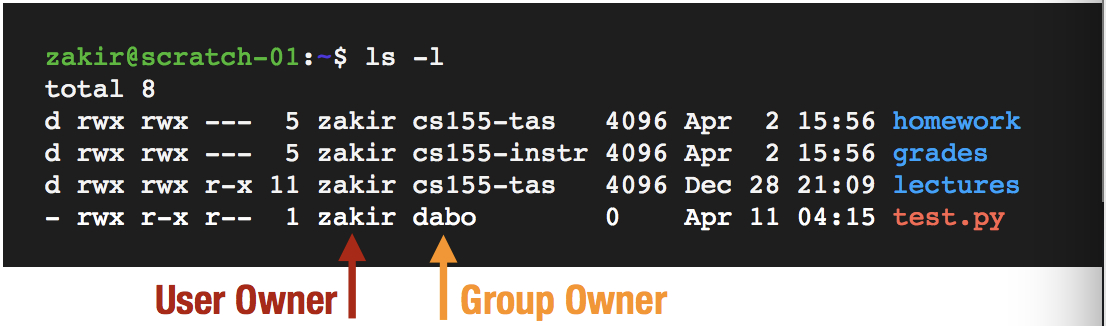
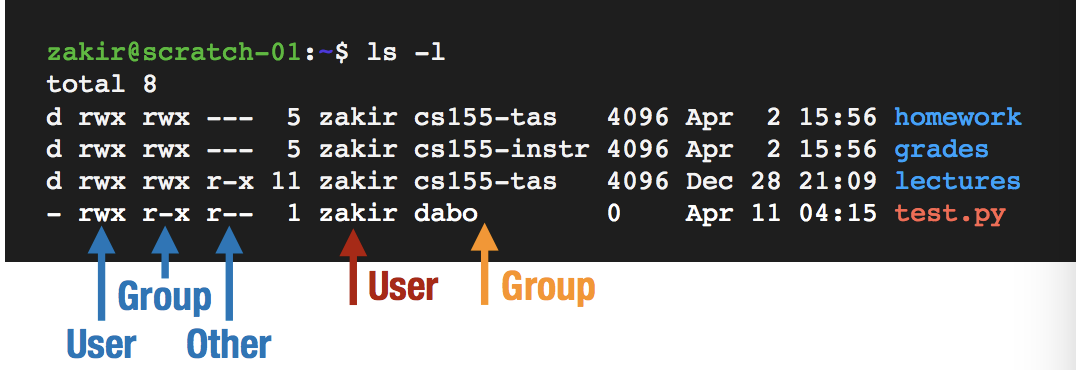
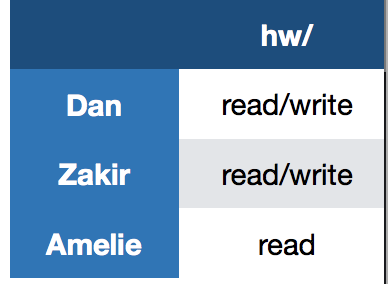
CAS CS 357

InClass Note 22

1. Vulnerabilities are inevitable
2. Any single buffer overflow, use-after-free, or null pointer dereference might allow an attacker to run malicious code
3. We’re getting better at finding and preventing bugs, but vulnerabilities are still common
4. Defense in Depth
5. Systems should be built with security protections at multiple layers
6. Principles of Secure systems
7. Defense in depth
8. Principle of least privilege
9. Privilege separation
10. Open design (Kerckhoffs’s principle)
11. Keep it simple
12. Least privilege
13. Users should only have access to the data and resources needed to perform routine, authorized tasks
14. Real world examples:
15. Faculty can only change grades for classes they teach
16. Only employees with background checks have access to classified documents
17. Privilege Separation
18. Least privilege requires dividing a system into parts to which we can limit access
19. Known as privilege separation
20. Segmenting a system into components with the least privilege needed can prevent an attacker from taking over the entire system
21. Security policies
22. Subject: acting system principals
23. Object: protected resources
24. Operation: how subjects operate on objects
25. Unix Security model
26. Subject: users, processes
27. Objects: files (sockets, pipes, hardware devices, kernel objects, process data), directories
28. Access operations: Read, write, execute
29. Users:
30. UNIX systems have many accounts

* Service accounts: used to run background processes (e.g., web server)
* User accounts: typically tied to a specific human

1. Every user has a unique integer ID – User ID – UID
2. UID 0 is reserved for special user root that has access to everything
3. 
4. Groups
5. UNIX has also groups – collections of users who can share files and other system resources
6. Every group has a group ID (GID) and name
7. 
8. File ownership
9. All Linux resources – sockets, devices, files – are managed as files
10. All files and directories have a single user owner and group owner
11. 
12. Access control
13. Three subjects have access to file: user owner, group owner, other
14. Subjects can have three operations: read, write, execute
15. Owner can change permissions and group. Root can change user ownership
16. 
17. Access control lists (ACLs)
18. UNIX’s permission model is a simple implementation of a generic access control strategy known as Access Control Lists (ACLs)
19. Every object has an ACL that identifies what operations subjects can perform
20. Each access to an object is checked against the object’s ACL
21. 
22. Role Based Access Control (RBAC)
23. Access control matrices can grow complex as number of subjects, objects, and possible operations grow
24. Observation: users change more often than roles
25. 