Format String Bugs

- Wuftpct 2.6.0
  - void loguser (char * user){ char buf [size]; Snprintf(buf, size, user); }
  - User = “rob” -> simple
  - User = “%d”, treats the string as a format argument going to look for an argument, doesn’t find it, so puts garbage on the stack.
    - %d will print first 4 bytes, %s which will cause a crash, or %c for a single byte

- Another example
  - printf (%s%n. iser. %count)
  - user = %n, write to memory location
  - What do we want to control:
    - Where we write
    - What we write
  - First controlling where to write
    - we can put “<0x1234568>%n” to change %n
    - Or end of buf overriding RA
  - Controlling what to write
  - “0000”<0xd1234568>,%8054321d,%n<shell code>
  - This will write the shell code in user location

Integer Over Flow

```
int n;

n = read (network);

if (n > 100)exit( ); blocking only upper, not lower too

buf = malloc(n);
```

Another example:

```
int Sys_read(int fd, void * buf, int len)

if (n > NFILES)

return – 1; another area in memory by a negative number

else f = file table [n]
```

Another example
int n = read (network);

if (n  0) exit(0);

send in a max possible integer and warp around to be a zero

b = malloc (n + 1);

mfm (b,0,n); put huge number of memory to zero

Command Injection Bugs

- SQL injection bug

  void getUser ( string username )
  SELECT * from user WHERE username = +username;
do execute (query)

- Username = “abcd!” or 1==1-abcdd”; DROP..

- How to stop this :
  o User prepared statements.
  o Don’t allow “;” and “\”