

Random Testing Using Problem-Specific Features

Did you notice that, for any x ,

```
ntohl (ntohl (x)) == x ?
```

We can use

- a **random number generator** (`rand`)
- to make up **1,000,000 values of x**
- (more than 5,000!)
- and **test each one**
- with **no need for a human**.

Is that good enough?

25

How About One More Human-Designed Test?

`rand` generates **non-negative values** (32-bit numbers in the range $[0, 2^{31} - 1]$).

As a graduate student, my desktop's IP* address was 128.32.36.37.

Let's try that: 0x80202425.

Otherwise, we may not have a negative case...

*Internet Protocol; specifically, IPv4.

26

Shift and Copy One Byte at a Time

```
int32_t ntohl (int32_t arg)
{
    int32_t res;
    res = (arg << 24);
    res |= ((arg & 0xFF00) << 8);
    res |= ((arg >> 8) & 0xFF00);
    res |= (arg >> 24);
    return res;
}
arg: 0x80202425
res: 0x25000000
```

So far, so good!

27

Shift and Copy One Byte at a Time

```
int32_t ntohl (int32_t arg)
{
    int32_t res;
    res = (arg << 24);
    res |= ((arg & 0xFF00) << 8);
    res |= ((arg >> 8) & 0xFF00);
    res |= (arg >> 24);
    return res;
}
arg: 0x80202425
res: 0x25240000
```

So far, so good!

28