

Bit, Byte ve Integer

BIL-304: Bilgisayar Mimarisi

Dersi veren öğretim üyesi:

Yrd. Doç. Dr. Fatih Gökçe

Ders kitabına ait sunum dosyalarından adapte edilmiştir: <http://csapp.cs.cmu.edu/>

Adapted from slides of the textbook: <http://csapp.cs.cmu.edu/>

Kod Güvenlik Örneği

```
/* Kullanıcının erişebileceği verileri tutan Kernel hafıza bölgesi*/
#define KSIZE 1024
char kbuf[KSIZE];

/* En fazla maxlen kadar byte'ı kernel bölgesinden kullanıcının
istediği yere kopyala*/
int copy_from_kernel(void *user_dest, int maxlen) {
    /* Byte sayısını gösteren len, buffer boyutu ve maxlen'den
    küçük olanın değerini alır*/
    int len = KSIZE < maxlen ? KSIZE : maxlen;
    memcpy(user_dest, kbuf, len);
    return len;
}
```

- Yukarıdaki kod yazılım hatası içeren FreeBSD'nin getpeername uygulamasına benzer bir koddur.
- Programlardaki açıkları bulmaya çalışan kişiler her zaman bulunur.

Tipik Kullanım

```
/* Kullanıcının erişebileceği verileri tutan Kernel hafıza bölgesi*/
#define KSIZE 1024
char kbuf[KSIZE];

/* En fazla maxlen kadar byte'ı kernel bölgesinden kullanıcının
istediği yere kopyala*/
int copy_from_kernel(void *user_dest, int maxlen) {
    /* Byte sayısını gösteren len, buffer boyutu ve maxlen'den
    küçük olanın değerini alır*/
    int len = KSIZE < maxlen ? KSIZE : maxlen;
    memcpy(user_dest, kbuf, len);
    return len;
}
```

```
#define MSIZE 528

void getstuff() {
    char mybuf[MSIZE];
    copy_from_kernel(mybuf, MSIZE);
    printf("%s\n", mybuf);
}
```

Kötü Niyetli Kullanım

```
/* memcpy kütüphane fonksiyonunun tanımı */  
void *memcpy(void *dest, void *src, size_t n);
```

```
/* Kullanıcının erişebileceği verileri tutan Kernel hafıza bölgesi*/  
#define KSIZE 1024  
char kbuf[KSIZE];  
  
/* En fazla maxlen kadar byte'ı kernel bölgesinden kullanıcının  
istediği yere kopyala*/  
int copy_from_kernel(void *user_dest, int maxlen) {  
    /* Byte sayısını gösteren len, buffer boyutu ve maxlen'den  
    küçük olanın değerini alır*/  
    int len = KSIZE < maxlen ? KSIZE : maxlen;  
    memcpy(user_dest, kbuf, len);  
    return len;  
}
```

```
#define MSIZE 528  
  
void getstuff() {  
    char mybuf[MSIZE];  
    copy_from_kernel(mybuf, -MSIZE);  
    . . .  
}
```

Ne zaman Unsigned Kullanmalıyım?

- ***Modüler aritmetiği işletirken kullanabiliriz.***
 - 1 bit daha hassas aritmetik imkanı sağlar
- ***Bitleri küme halinde kullanacaksak kullanabiliriz.***
 - Lojik sağa kaydırma, işaret biti çoğaltılmaz.

C'de Integer Bulmacaları

foo ve bar
herhangi bir
değer dönen iki
fonksiyon

```
int x = foo();
int y = bar();
unsigned ux = x;
unsigned uy = y;
```

- $(x > 0) \ || \ (x-1 < 0)$
- $(x \ \& \ 7) \ != \ 7 \ || \ (x \ll 29 < 0)$
- $(x * x) \ >= \ 0$
- $x < 0 \ || \ -x \ <= \ 0$
- $x > 0 \ || \ -x \ >= \ 0$
- $x + y == uy + ux$
- $x \sim y + uy * ux == -x$

Verilen ifadeler her zaman
DOĞRU mu döner, **YANLIŞ**
dönebilecekleri durum varsa bu
hangi durumdur?