



Pembuatan Piranti Lunak Perlu Perhatikan Masalah Keamanan

Piranti lunak (*software*) sudah menjadi bagian dari kehidupan kita sehari-hari, mulai dalam bentuk program yang dijalankan komputer biasa sampai ke program yang ada di embedded system seperti handphone. Pada mulanya software sangat sederhana dan tidak terlalu kompleks sehingga mudah dipahami dan diperiksa jika ada kesalahan. Sesuai dengan perkembangan jaman, kompleksitas dari software pun meningkat sehingga potensi timbulnya kesalahan menjadi lebih besar.

Kesalahan atau kegagalan yang ditimbulkan oleh software bisa beragam, mulai dari ketidaknyamanan, timbulnya lubang keamanan (*security hole*), kerugian finansial, dan bahkan dapat membahayakan nyawa. Tulisan ini memfokuskan kepada timbulnya lubang keamanan yang diakibatkan oleh pembuatan software yang tidak baik.

Secara umum, pengembang aplikasi (*programmer*) belum memiliki pemahaman tentang masalah keamanan. Keamanan tidak diperhatikan mulai dari desain, implementasi, sampai ke operasional. Masalah keamanan baru diperhatikan jika sudah timbul masalah yang mengakibatkan kerugian finansial. Padahal memperbaiki software yang sudah dijual dan digunakan oleh umum jauh lebih mahal daripada waktu desain atau implementasi. Untuk itu usaha membekali para programmer dengan ilmu keamanan merupakan langkah yang penting.

Masalah keamanan dalam pengembangan software yang sering muncul antara lain adalah *buffer overflow*, pemrosesan format string yang tidak divalidasi, dan input yang tidak divalidasi.

Mari kita bahas salah satunya, yaitu *buffer overflow*.

Buffer merupakan tempat di memori untuk menyimpan variabel dan program. Ketika kita mendeklarasikan sebuah variabel "A", dengan perintah "char A[16]" di dalam bahasa C/C++ misalnya, maka kita mengalokasikan 16 byte di memori untuk variabel A tersebut. Buffer overflow terjadi jika kita memberikan data lebih banyak daripada tempat yang disediakan. Perhatikan kode C/C++ di bawah ini.

```
#include <iostream>
using namespace std;
// contoh pemrograman yang buruk
// diadopsi dari tulisan Aleph One
// simpan berkas ini dengan nama jelek.cc
```

```
// dan rakit dengan g++

void fungsijelek(char *str)
{
    // fungsi ini meng-copy karakter dari str ke
    sementara
    // masalah ada di strcpy yang tidak dibatasi
    char sementara[16];
    strcpy(sementara, str);
}

int main()
{
    char stringpanjang[256];
    int i;
    for (i=0 ; i < 255 ; i++)
        stringpanjang[i] = 'A';
    fungsijelek(stringpanjang);
}
```

...pengembang aplikasi (programmer) belum memiliki pemahaman tentang masalah keamanan.

Dalam contoh di atas kita memiliki sebuah variabel yang bernama "stringpanjang" yang panjangnya 256 bytes. String ini kemudian kita isi dengan karakter 'A' sebanyak 256 buah. Kemudian kita memanggil fungsi yang bernama "fungsijelek" yang tugasnya adalah membuat duplikat (copy) dari string yang dikirimkan ke fungsi ini, yaitu "stringpanjang".

Sayangnya "fungsijelek" hanya mengalokasikan buffer sementara sebanyak 16 bytes akan tetapi dia tidak membatasi proses peng-copyan dengan menggunakan fungsi "strcpy". Akibatnya, ketika program dijalankan, program akan terhenti seperti contoh di bawah ini.

```
$ g++ jelek.cc
$ ./a.out
Segmentation fault (core dumped)
```

Akibat yang terjadi bisa bermacam-macam, mulai dari program yang terhenti seperti di atas, *server crash*, sampai user bisa masuk tanpa password. Untuk itu wawasan akan masalah keamanan sangat penting bagi pengembang software. 🐞