Pseudo Random Number Generation Lab

Liangyu W

Task 1: Generate Encryption Key in a Wrong Way

Compile and run the following program:

random.c (~/Desktop) - gedit

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define KEYSIZE 16

void main()
{
        int i;
        char key[KEYSIZE];

        printf("%lld\n", (long long) time(NULL));
        srand (time(NULL));

        for (i = 0; i< KEYSIZE; i++) {
                key[i] = rand()%256;
                printf("%.2x", (unsigned char)key[i]);
        }
        printf("\n");
}
```

Terminal

```
length_ext  random.c
[12/19/19]seed@liangyu:~/Desktop$ gcc
random.c -o random
[12/19/19]seed@liangyu:~/Desktop$ ./ra
ndom
1576792115
d234c00b3516a23bfb0071c55b3ae567
[12/19/19]seed@liangyu:~/Desktop$ ./ra
ndom
1576792117
961be9c2b032414d4d38b6c407eb219d
[12/19/19]seed@liangyu:~/Desktop$ ./ra
ndom
1576792122
67952e2175932422bbc64944c9eb8eb1
[12/19/19]seed@liangyu:~/Desktop$
```

*random.c (~/Desktop) - gedit

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define KEYSIZE 16

void main()
{
        int i;
        char key[KEYSIZE];

        printf("%lld\n", (long long) time(NULL));
        //srand (time(NULL));

        for (i = 0; i< KEYSIZE; i++) {
                key[i] = rand()%256;
                printf("%.2x", (unsigned char)key[i]);
        }
        printf("\n");
}
```

Terminal

```
[12/19/19]seed@liangyu:~/Desktop$ gcc random.c
 -o random
[12/19/19]seed@liangyu:~/Desktop$ ./random
1576793942
67c6697351ff4aec29cdbaabf2fbe346
[12/19/19]seed@liangyu:~/Desktop$ ./random
1576793943
67c6697351ff4aec29cdbaabf2fbe346
[12/19/19]seed@liangyu:~/Desktop$ ./random
1576793943
67c6697351ff4aec29cdbaabf2fbe346
[12/19/19]seed@liangyu:~/Desktop$ ./random
1576793944
67c6697351ff4aec29cdbaabf2fbe346
[12/19/19]seed@liangyu:~/Desktop$ ./random
1576793945
67c6697351ff4aec29cdbaabf2fbe346
[12/19/19]seed@liangyu:~/Desktop$
```

Time() function returns the current system time. srand() function sets the seed for the random number generator. In this case, the current system time is used as seed. If srand() is function is not used, then the default seed of 1 is used for rand(), this causes the same number to be generated every time the program runs.

Task 2: Guessing the Key

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <openssl/aes.h>
#include <string.h>

#define KEYSIZE 16

void main() {
    unsigned char plaintxt[] = "\x25\x50\x44\x46\x2d\x31\x2e\x35\x0a\x25\xd0\xd4\xc5\xd8\x0a\x34";
    unsigned char ciphertext[] = "\xd0\x6b\xf9\xd0\xda\xb8\xe8\xef\x8B\x06\x60\xd2\xaf\x65\xaa\x82";
    int i;
    int j;
    char cipher[128];
    char key[KEYSIZE];
    AES_KEY aeskeyEnc;
    int k;

    FILE *fo;
    fo = fopen("keys.txt", "w");

    for (i = 1524013729L; i<= 1524020929L; i++) {

        srand(i);
        for (j = 0; j < KEYSIZE; j++) {
            key[j] = rand()%256;
            fprintf(fo, "%.2x", (unsigned char)key[j]);
        }
        fprintf(fo, "\n");


        unsigned char iv[]  = "\x09\x00\x07\x06\x05\x04\x03\x02\x01\x00\xA2\xB2\xC2\xD2\xE2\xF2";
        AES_set_encrypt_key(key, 128, &aeskeyEnc);
        AES_cbc_encrypt(plaintxt, cipher, 33, &aeskeyEnc, iv, AES_ENCRYPT);


        if (memcmp(ciphertext, cipher, 16) == 0) {

            printf("\nMatch Found\nSeed: %d \nKey : ", i);

            for (k = 0; k < KEYSIZE; k++) {
                printf("%.2x", (unsigned char)key[k]);
            }
            printf("\nCipherText: ");
            for (k = 0; k < KEYSIZE; k++) {
                printf("%.2x", (unsigned char)cipher[k]);
            }
            printf("\n\n");
        }

    }
    fclose(fo);

}
```

```
Terminal
[12/23/19]seed@liangyu:~/Desktop$ gcc random.
c -o random -lcrypto
[12/23/19]seed@liangyu:~/Desktop$ ./random

Match Found
Seed: 1524017695
Key : 95fa2030e73ed3f8da761b4eb805dfd7
CipherText: d06bf9d0dab8e8ef880660d2af65aa82

[12/23/19]seed@liangyu:~/Desktop$
```

Task 3: Measure the Entropy of Kernel

```
Terminal
[12/23/19]seed@liangyu:~$ cat /proc/sys/ke
rnel/random/entropy_avail
3596
[12/23/19]seed@liangyu:~$
```

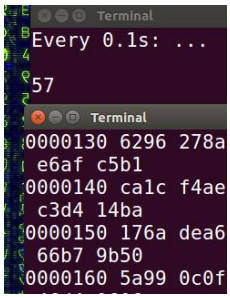
```
Terminal
Every 0.1s: c...

3429
```

Moving mouse around randomly generates significant amount of entropy.

Task 4: Get Pseudo Random Numbers from/dev/random

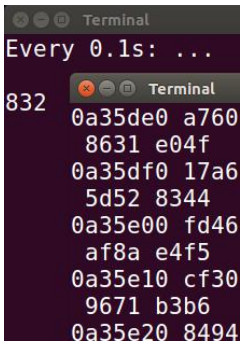
Run command “cat /dev/random | hexdump”:



Not moving mouse or type anything generates little to no entropy, while moving mouse around generates a lot of entropy. /dev/random require a lot of entropy to generate a pseudo random number.

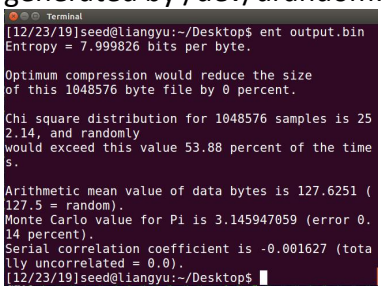
If a server uses /dev/random to generate the random session key with a client, an attacker can launch a DOS attack by keep requesting random number from /dev/random until the entropy of the randomness pool drops to zero. When this happens, /dev/random will block and the server will no longer be able to generate random session keys for other users.

Task 5: Get Random Numbers from/dev/urandom



/dev/urandom keeps generating pseudo random number even when there is no user interaction with the system. Moving the mouse around increases the amount of entropy in the kernel, but doesn’t seem to effect how quickly /dev/urandom generates pseudo random numbers.

Ent applies various statistical tests to the sequence of pseudo random numbers generated by /dev/urandom:



code to generate a 256-bit encryption key using /dev/urandom:

```c
#include <stdio.h>
#include <stdlib.h>

#define LEN 32  // 256 bits

int main() {

    unsigned char*key = (unsigned char*) malloc(sizeof(unsigned char)*LEN);
    FILE*random = fopen("/dev/urandom", "r");
    fread(key, sizeof(unsigned char)*LEN, 1, random);

    for (int i = 0; i < LEN; i++) {
        printf("%.2x", key[i]);
    }
    printf("\n");
    fclose(random);

}
```

Terminal

```
[12/23/19]seed@liangyu:~/Desktop$ gcc enckey.c -o
 enckey
[12/23/19]seed@liangyu:~/Desktop$ ./enckey
cf31a8c43709b3ae41c441c3c431a26fcc3b220daa9ba29c9
fe5a6b3aa20ea89
[12/23/19]seed@liangyu:~/Desktop$ ./enckey
bd5371f86339c43fd7fcd9a4b75823cea554370f94affc062
8b3cda9b105c7b2
[12/23/19]seed@liangyu:~/Desktop$ ./enckey
229dff126ab435c0125a1b0ab7bfbd929508159bb64a18a16
96a1599392b90f4
[12/23/19]seed@liangyu:~/Desktop$ ./enckey
d6715367e9c227abe8c16f0ae0664a7a4d1756ef1cd5d5234
65cab38c432e09a
[12/23/19]seed@liangyu:~/Desktop$
```