Codegate 2014 オンライン予選 Write Up
Chrono (Logical) 300

ソース： www.blue-lotus.net/2014-02-25-codegate-ctf-quals-2014-chrono-writeup/
和訳：田中ザック

説明：

```
ssh guest@58.229.183.16 / ExtremelyDangerousGuest
```

ssh guest@58.229.183.16 （PW：ExtremelyDangerousGuest）でログインし、少し見てみると、

```
guest@codegate:~$ ls -al
total 20
dr-xr-x--- 2 guest guest 4096 Feb 19 22:49 .
drwxr-xr-x 4 root  root  4096 Feb 22 00:07 ..
lrwxrwxrwx 1 root  root     9 Feb 19 22:22 .bash_history -> /dev/null
-rw-r--r-- 1 root  root   220 Feb 19 22:21 .bash_logout
-rw-r--r-- 1 guest guest 3637 Mar 31  2013 .bashrc
-rw-r--r-- 1 guest guest  675 Mar 31  2013 .profile

guest@codegate:~$ lsattr -a
----i--------e-- ./.profile
----i--------e-- ./.bash_logout
----------e-- ./..
----i--------e-- ./.
lsattr: Operation not supported While reading flags on ./.bash_history
----i--------e-- ./.bashrc

guest@codegate:/home/chrono$ ls -al
total 924
dr-xr-xr-x 2 chrono chrono   4096 Feb 19 22:24 .
drwxr-xr-x 4 root   root     4096 Feb 22 00:07 ..
lrwxrwxrwx 1 root   root        9 Feb 19 22:24 .bash_history -> /dev/null
-rw-r--r-- 1 chrono chrono    220 Mar 31  2013 .bash_logout
-rw-r--r-- 1 chrono chrono   3637 Mar 31  2013 .bashrc
-rwsr-xr-x 1 chrono chrono 921576 Feb 19 22:13 chrono
```

```
-r-------- 1 chrono chrono     28 Feb 22 00:07 key
-rw-r--r-- 1 chrono chrono    675 Mar 31  2013 .profile

guest@codegate:/home/chrono$ lsattr -a
lsattr: Permission denied While reading flags on ./key
----i--------e-- ./.profile
----i--------e-- ./.bash_logout
-------------e-- ./..
----i--------e-- ./.
lsattr: Operation not supported While reading flags on ./.bash_history
----i--------e-- ./chrono
----i--------e-- ./.bashrc
```

セキュリティ硬いね！

```
guest@codegate:/home/chrono$ file chrono chrono: setuid ELF 64-bit LSB executable, x86-64,
version 1 (SYSV), statically linked, for GNU/Linux 2.6.24,
BuildID[sha1]=0x8c0628afc74aa0a346020da6d9bbd44bd90709a0, stripped
```

ということはストリップされた 64 ビットのバイナリです。

/usr/lib/x86_64-linux-gnu から libc.a と libm.a の静的ライブラリを
ダウンロードして、IDA の FLAIR にインポートして、FLIRT シグネチャを
作ってバイナリに適用します。これでリバース・エンジニアリングする時に
静的ライブラリを解析しなくても良いです。

注意：IDA で timeval の struct のサイズが間違っています。
tv_sec と tv_usec はクアッドワードです。

IDA でよく解析すると、
出力が select している間の経過時間（何ミリ秒）によって変わる様です。

Linux の fs/select.c を読むと

```
SYSCALL_DEFINE5(select, int, n, fd_set __user *, inp, fd_set __user *, outp,
                fd_set __user *, exp, struct timeval __user *, tvp)
{
    struct timespec end_time, *to = NULL;
    struct timeval tv;
    int ret;

    if (tvp) {
        if (copy_from_user(&tv, tvp, sizeof(tv)))
            return -EFAULT;

        to = &end_time;
        if (poll_select_set_timeout(to,
                tv.tv_sec + (tv.tv_usec / USEC_PER_SEC),
                (tv.tv_usec % USEC_PER_SEC) * NSEC_PER_USEC))
            return -EINVAL;
    }

    ret = core_sys_select(n, inp, outp, exp, to);
    ret = poll_select_copy_remaining(&end_time, tvp, 1, ret);

    return ret;
}
```

timeval の struct が poll_select_copy_remaining で更新されます。

poll_select_copy_remaining :

```
static int poll_select_copy_remaining(struct timespec *end_time, void __user *p, int
timeval, int ret)
{
    struct timespec rts;
    struct timeval rtv;

    if (!p)
        return ret;

    if (current->personality & STICKY_TIMEOUTS)
        goto sticky;
    ...
    ...
    /*
     * If an application puts its timeval in read-only memory, we
     * don't want the Linux-specific update to the timeval to
     * cause a fault after the select has completed
     * successfully. However, because we're not updating the
     * timeval, we can't restart the system call.
     */
sticky:
    if (ret == -ERESTARTNOHAND)
        ret = -EINTR;
    return ret;
}
```

STICKY_TIMEOUTS は setuid で実行しても STICKY_TIMEOUTS ビットはクリアされないし、使えそうですね。

include/uapi/linux/personality.h:

```
/*
 * Security-relevant compatibility flags that must be
 * cleared upon setuid or setgid exec:
 */
#define PER_CLEAR_ON_SETID (READ_IMPLIES_EXEC  | ADDR_NO_RANDOMIZE  | ¥
                            ADDR_COMPAT_LAYOUT | MMAP_PAGE_ZERO)
```

では、setarch を使って personality を変えます。

```
Usage: setarch <arch> [options] [program [program arguments]]

Options:
 -h, --help               displays this help text
 -v, --verbose            says what options are being switched on
 -R, --addr-no-randomize  disables randomization of the virtual address space
 -F, --fdpic-funcptrs     makes function pointers point to descriptors
 -Z, --mmap-page-zero     turns on MMAP_PAGE_ZERO
 -L, --addr-compat-layout changes the way virtual memory is allocated
 -X, --read-implies-exec  turns on READ_IMPLIES_EXEC
 -B, --32bit              turns on ADDR_LIMIT_32BIT
 -I, --short-inode        turns on SHORT_INODE
 -S, --whole-seconds      turns on WHOLE_SECONDS
 -T, --sticky-timeouts    turns on STICKY_TIMEOUTS
 -3, --3gb                limits the used address space to a maximum of 3 GB
     --4gb                ignored (for backward compatibility only)
     --uname-2.6          turns on UNAME26

For more information see setarch(8).
```

やってみると、フラグが出力されます。

```
guest@codegate:/home/chrono$ setarch x86_64 -T ./chrono
cat key

He said :
    import zlib
    zlib.compress(space)

 o
|+|
 !
[ ]

voila!
dIfF3rENT_L3VEL_s4me_aNsW3r
```

チャレンジを C にリバースしたソースコード：

```
/*
 * CodeGate 2014 - chrono 300 Point Logical
 *
 * Reverse Engineered by libmaru (libmaru#gmail.com)
 *
 * Build Environment:
 *   Linux codegate 3.11.0-15-generic #25-Ubuntu SMP
 *     Thu Jan 30 17:22:01 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
 *   gcc (Ubuntu/Linaro 4.8.1-10ubuntu9) 4.8.1
 *
 * Build Instruction:
 *   gcc -static -fno-stack-protector -o chrono chrono.c -lm
 *   strip chrono
 *
 * Produce identical binary except Build ID
 *
 */

#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <math.h>
#include <sys/select.h>

#define TIMEOUT     299792458
#define PARAM       0.00300000000000000000625L
#define IDEAL_SUM   6.6260690000000002087L
#define PI          3.1415920000000001622L

int main()
{
    long double remain = 0;
    long double param = 0;
    long double item = 0;
    long double sum = 0;
    struct timeval timeout;
    fd_set readfds;
```

```
char buf[256] = {};
int x,y;
bool flag = 1;

timeout.tv_sec  = TIMEOUT / 1000000;
timeout.tv_usec = TIMEOUT % 1000000;

FD_ZERO( &readfds );
FD_SET( 0, &readfds );

if( select( 1, &readfds, NULL, NULL, &timeout ) <= 0 )
{
    puts( "no hack." );
    exit( 0 );
}

remain = timeout.tv_sec * 1000000 + timeout.tv_usec;
puts( "¥nHe said : ¥n¥timport zlib¥n¥tzlib.compress(space)" );
param = ( TIMEOUT - remain ) * PARAM;

for( y = 10; y >= 0; --y )
{
    for( x = 0; x <= 45; ++x )
    {
        item = sin( x/param + 4 ) + 5;
        sum += item;

        if( flag && x > 4 && x == rint( param * PI ) && rint( item ) == y-1 )
            flag = 0, putchar('*');
        else if( flag && x == 45 && y-1 == rint( item ) )
            putchar('*');
        else if( x > 4 && x <= 45 && rint( item ) == y )
            putchar('#');
        else if( x == 1 && y == 7 )
            putchar('o');
        else if( x == 1 && y == 6 )
            putchar('+');
        else if( x == 0 && y == 6 )
```

```
                putchar('|');
            else if( x == 2 && y == 6 )
                putchar('|');
            else if( x == 1 && y == 5 )
                putchar('!');
            else if( x == 0 && y == 4 )
                putchar('[');
            else if( x == 2 && y == 4 )
                putchar(']');
            else
                putchar(' ');
        }
        putchar('¥n');
    }

    if( FD_ISSET( 0, &readfds ) )
    {
        fgets( buf, sizeof buf - 1, stdin );
        if( !( sum > IDEAL_SUM ) && !( sum < IDEAL_SUM ) )
        {
            puts( "voila!" );
            system( buf );
        }
    }

    return 0;
}
```