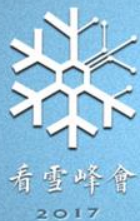




# 看雪 2017 安全开发者峰会

Kanxue 2017 Security Developer Summit

2000-2017



## Windows Subsystem for Linux

lu0@上海高重

# 关于我

lu0

- 之前 !@#\$%^&\*
- 现在 上海高重信息科技有限公司CIO





# 大纲

从另一个角度看待主机安全的攻与防

一、Windows下跑Linux应用的发展变迁

二、WSL漏洞的成因



# 在Windows下跑Linux/Unix应用的历史

## □ SFU

- 全称: Windows Service For Unix
- 最后的版本: 3.5
- 用于提供大量的UNIX工具集: 例如grep/vi等
- 能用gcc 3.3

## □ SUA

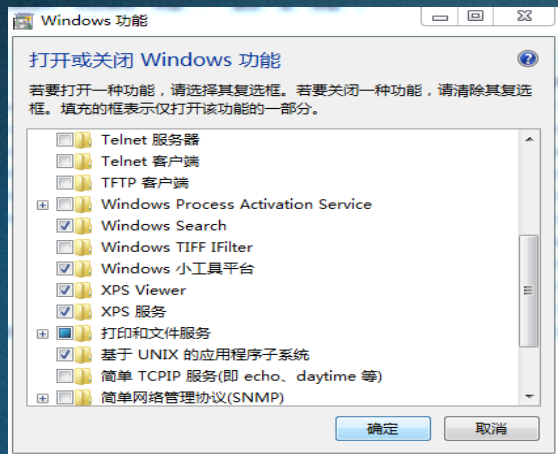
- 全称: Subsystem for UNIX-based Applications
- 适用Vista~Win7

## □ WSL

- 全称: Windows SubSystem for Linux
- Win10
- Beta状态



# Win7下的SUA



Choose the download you want

<input type="checkbox"/> File Name	Size
<input type="checkbox"/> Utilities and SDK for UNIX-based Applications_X86.exe	473.0 MB
<input checked="" type="checkbox"/> Utilities and SDK for UNIX-based Applications_AMD64.exe	475.2 MB
<input type="checkbox"/> Utilities and SDK for UNIX-based Applications_IA64.exe	484.1 MB

```
C Shell
Welcome to the SUA utilities.

DISPLAY=localhost:0.0
% pwd
/dev/fs/C/Users/user
% id
uid=197608(user) gid=197121(None) groups=197121(None), 65792(<Everyone>), 66930(<本地帐户和管理员组成员>), 131616(<Administrators>), 131617(<Users>), 66820(<INTERACTIVE>), 66049(<控制台登录>), 66827(<Authenticated Users>), 66831(<This Organization>), 66929(<本地帐户>), 4095(<CurrentSession>), 66048(<Local>), 262154(<NT AUTHORITY\NTLM Authentication>), 401408(<Mandatory Label+Medium Mandatory Level>)
%
```

- ❑ 安装完毕后，所有相关组件位于C:\WINDOWS\SUA
- ❑ su root相当于administrator
- ❑ 可执行文件为MZ格式
- ❑ 通过/dev/fs/C 来访问C盘所有数据
- ❑ SUA系统可设置为大小写敏感



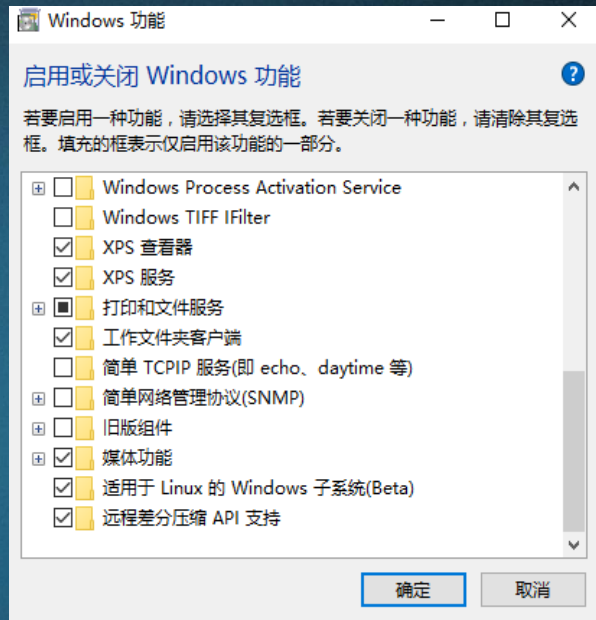
# Windows 10的进化WSL

## 使用开发人员功能

这些设置只用于开发。

[了解更多信息](#)

- ☐ Windows 应用商店应用  
仅安装 Windows 应用商店的应用。
- ☐ 旁加载应用  
从你所信任的其他源(例如工作区)安装应用。
- ☒ 开发人员模式  
安装任何签名的应用和使用高级开发功能。



- ❑ 通过开发人员模式才能安装
- ❑ Beta状态





# Windows 10的进化: WSL1

- ❑ 对linux的应用的支持有了革命性的转变
- ❑ 安装完毕后, 所有相关组件位于%user%\AppData\Local\lxss
- ❑ 安装后内容为Ubuntu 14.04.5 LTS的一个子集

```
...:/etc$ cat issue
Ubuntu 14.04.5 LTS \n \l
```

- ❑ 可执行文件为ELF格式

```
...:/bin$ readelf -a mkdir | more
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:       ELF64
  Data:        2's complement, little endian
  Version:     1 (current)
  OS/ABI:      UNIX - System V
  ABI Version: 0
  Type:        EXEC (Executable file)
  Machine:     Advanced Micro Devices X86-64
  Version:     0x1
  Entry point address: 0x401ea3
  Start of program headers: 64 (bytes into file)
  Start of section headers: 50144 (bytes into file)
  Flags:       0x0
  Size of this header:   64 (bytes)
  Size of program headers: 56 (bytes)
  Number of program headers: 9
  Size of section headers: 64 (bytes)
  Number of section headers: 28
  Section header string table index: 27
```

```
C:\
├── cache
├── data
├── home
│   └── lu0
├── mnt
│   ├── c
│   └── d
├── root
├── rootfs
│   ├── acct
│   ├── bin
│   ├── boot
│   │   └── grub
│   ├── cache
│   ├── data
│   ├── dev
│   │   ├── block
│   │   ├── disk
│   │   │   ├── by-label
│   │   │   └── by-uuid
│   │   ├── mapper
│   │   └── pts
│   └── etc
│       ├── acpi
│       │   └── events
│       ├── alternatives
│       ├── apm
│       │   └── event.d
│       ├── apparmor
│       │   └── init
│       │       └── network-interface-security
│       ├── apparmor.d
│       │   ├── abstractions
│       │   │   ├── apparmor_api
│       │   │   └── ubuntu-browsers.d
│       │   ├── cache
│       │   ├── disable
│       │   ├── force-complain
│       │   ├── local
│       │   ├── snap
│       │   │   ├── abstractions
│       │   │   │   ├── apparmor_api
│       │   │   │   └── ubuntu-browsers.d
│       │   └── tunables
│       │       ├── home.d
│       │       ├── multiarch.d
│       │       └── xdg-user-dirs.d
│       ├── apport
│       │   └── blacklist.d
│       ├── apt
│       │   ├── apt.conf.d
│       │   ├── preferences.d
│       │   ├── sources.list.d
│       │   └── trusted.gpg.d
│       ├── bash_completion.d
│       ├── byobu
│       ├── ca-certificates
│       │   └── update.d
│       └── calendar
-- More --
```





# Windows 10的进化: WSL2

## ❑ proc FS也能使用

```
total 0
dr-x----- 1 lu0 lu0 0 Jun 26 16:21 ./
dr-xr-xr-x 1 lu0 lu0 0 Jun 26 16:21 ../
lrwx----- 1 lu0 lu0 0 Jun 26 16:22 0 -> /dev/tty1
lrwx----- 1 lu0 lu0 0 Jun 26 16:22 1 -> /dev/tty1
lrwx----- 1 lu0 lu0 0 Jun 26 16:22 2 -> /dev/tty1
lrwx----- 1 lu0 lu0 0 Jun 26 16:22 255 -> /dev/tty1
/proc/2/fd$ pwd
/proc/2/fd$
```

## ❑ 寻找C盘D盘...

❑ WSL与SFU不同, 不使用devfs映射windows卷。

❑ 通过mnt目录下c,d等目录映射C:\,D:\

❑ 虽然表面上看是mount来的, 其实不是mount的, mount命令无法看到ntfs的盘存在

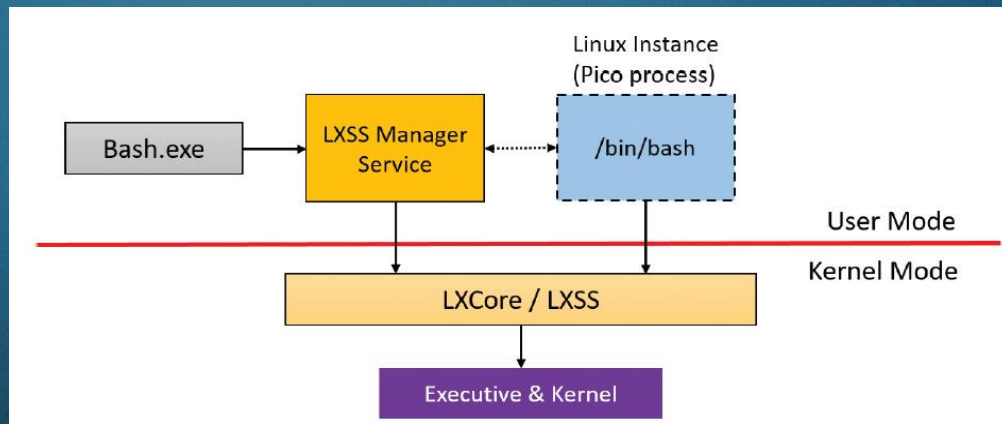
❑ 网络支持问题较多,随版本问题数量大幅减少

```
/mnt/c# pwd
/mnt/c
/mnt/c# ls
ls: cannot access hiberfil.sys: Permission denied
ls: cannot access pagefile.sys: Permission denied
ls: cannot access swapfile.sys: Permission denied
boot
bootmgr
BOOTNXT
Config.Msi
drivers
hiberfil.sys
pagefile.sys
PerfLogs
Program Files
Program Files (x86)
Recovery
System
System Volume Information
Users
Windows
```

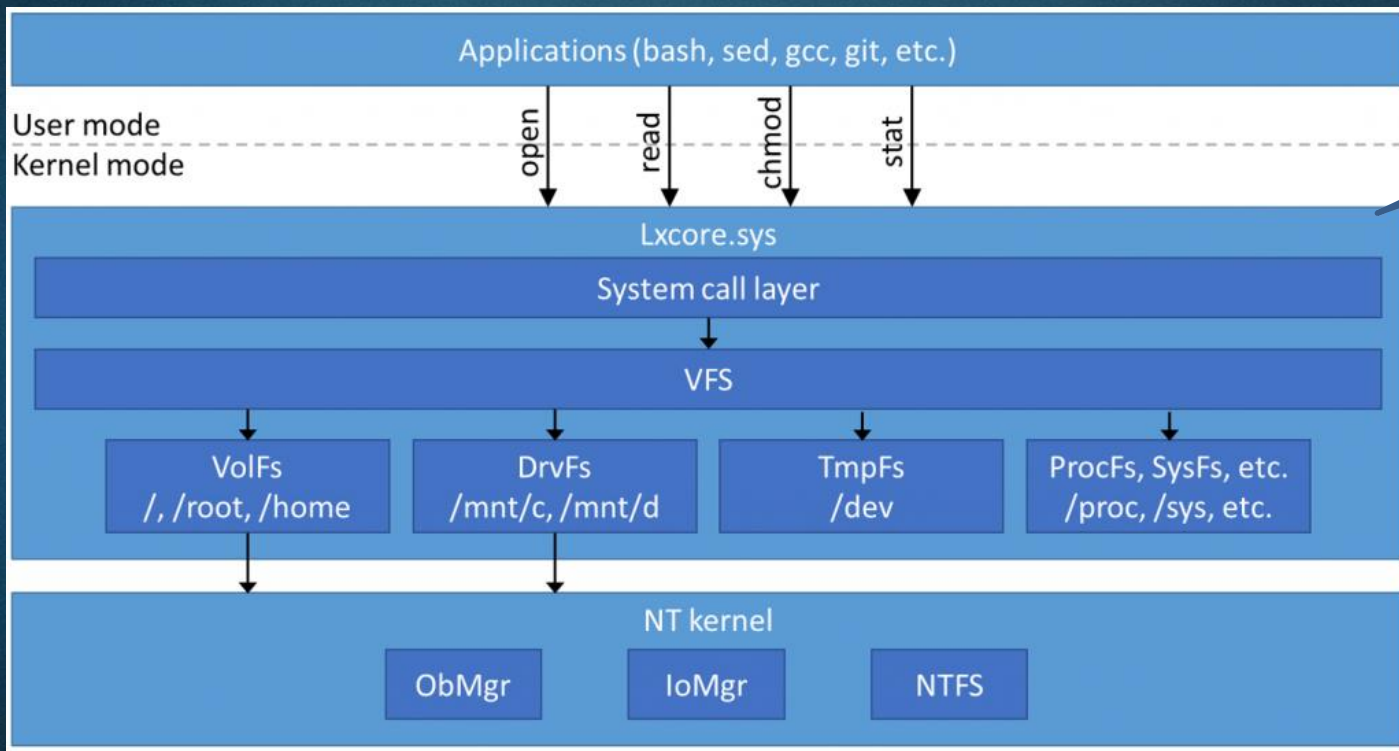


# WSL internals (1)

- ❑ WSL通过Windows内建的环境子系统来实现
- ❑ 曾经存在过的环境子系统包括：OS/2,POSIX等
- ❑ 为支持ELF文件的执行，Windows 10内核进行了较大的改造
  - ❑ 一部分通过利用现存的功能，改造并重新实现了一大批接口。
  - ❑ 通过pico provider完整实现了linux的VFS，/sys，/dev等关键组件



## WSL internals (2)



# 带来的安全挑战

- 料敌先机
  - ■安全软件将无法监控到进程派生关系
  - ■安全软件的网络监控功能，于下一版Win10的Insider版本发布时会受到破坏








# 带来的安全挑战

- 料敌先机
  - 安全软件将无法监控到网络数据
- 对主机安全软件的挑战
  - 恶意代码经gcc编译，绕过现有杀毒软件
- OS自身各模块的挑战
  - tmpfs导致BSOD，可引发DoS



# 根因分析

- 对主机安全软件的挑战
  - 进程监控的挑战：进程创建监视点发生了重大变化
    - 主机安全类软件主要依赖的进程创建通知不足以发现WSL子进程的创建

	1742 (0x06CE)	1737 (0x06C9)	PsSetCreateProcessNotifyRoutine	0x005AD080
	1743 (0x06CF)	1738 (0x06CA)	PsSetCreateProcessNotifyRoutineEx	0x005ACE80
	1744 (0x06D0)	1739 (0x06CB)	PsSetCreateProcessNotifyRoutineEx2	0x005ACF40

- WSL的内核数据结构未公开



# 根因分析

- 对主机安全软件的挑战
  - 进程监控的挑战：进程创建监视点发生了重大变化
    - `fork/exec`模式下的一个pid在不同阶段对应不同可执行文件

```
child_pid = fork(); ①
if(child_pid == 0) {
    /* This is done by the child process. */
    do_something();
    execvp(argv[0], argv); ②

    /* If execvp returns, it must have failed. */

    printf("Unknown command\n");
    exit(0);
}
else {
    /* This is run by the parent.  Wait for the child
       to terminate. */

    do {
        pid_t tpid = wait(&child_status);
        if(tpid != child_pid) process_terminated(tpid);
    } while(tpid != child_pid);
}
```





# 根因分析

- 对主机安全软件的挑战
  - 可执行文件种类继续增加：除EXE/COM/BAT/PS/VBS/PL/PY...外，新增ELF/ELF64/!#模式
  - 网络模块的变化，内核socket的TDI Bypass提上日程





# 根因分析

## 对主机安全软件的挑战

- NTFS文件名的大小写敏感的特性，愈发带来挑战，处理不当导致打开错误文件。

```
lu@lu: /mnt/c/Windows/Security > dir a
2017/11/14 19:39          4 A
2017/11/02 13:35          4 a
                2 个文件          8 字节
                0 个目录 207,428,587,520 可用字节

lu@lu: /mnt/c/Windows/Security > type a
AAA
AAA

lu@lu: /mnt/c/Windows/Security > type A
AAA
AAA
```

```
lu@Granddatasafe:~$ ll
total 20
drwxr-xr-x 0 root root 512 Nov 14 19:39 ./
drwxr-xr-x 0 root root 512 Nov 9 22:30 ../
-rw-rw-rw- 1 root root 4 Nov 2 13:35 a
-rw-rw-rw- 1 root root 4 Nov 14 19:39 A
lrwxrwxrwx 1 root root 19 Nov 2 13:43 b -> /mnt/c/test/wsl.txt*
-rw-rw-rw- 1 root root 934 Jul 13 17:26 .bash_history
-rw-r--r-- 1 root root 220 Jun 25 22:52 .bash_logout
-rw-r--r-- 1 root root 3659 Jun 26 12:22 .bashrc
drwxr-xr-x 0 root root 512 Nov 9 22:37 .cache/
-rw-r--r-- 1 root root 675 Jun 25 22:52 .profile
drwx----- 0 root root 512 Jul 13 17:26 .ssh/
-rw-r--r-- 1 root root 1884 Oct 25 13:32 .viminfo
lu@Granddatasafe:~$ echo AAA > A
lu@Granddatasafe:~$ ll
total 20
drwxr-xr-x 0 root root 512 Nov 14 19:39 ./
drwxr-xr-x 0 root root 512 Nov 9 22:30 ../
-rw-rw-rw- 1 root root 4 Nov 2 13:35 a
-rw-rw-rw- 1 root root 4 Nov 14 19:39 A
lrwxrwxrwx 1 root root 19 Nov 2 13:43 b -> /mnt/c/test/wsl.txt*
-rw-rw-rw- 1 root root 934 Jul 13 17:26 .bash_history
-rw-r--r-- 1 root root 220 Jun 25 22:52 .bash_logout
-rw-r--r-- 1 root root 3659 Jun 26 12:22 .bashrc
drwxr-xr-x 0 root root 512 Nov 9 22:37 .cache/
-rw-r--r-- 1 root root 675 Jun 25 22:52 .profile
drwx----- 0 root root 512 Jul 13 17:26 .ssh/
-rw-r--r-- 1 root root 1884 Oct 25 13:32 .viminfo
lu@Granddatasafe:~$ cat A
AAA
lu@Granddatasafe:~$ cat a
aaa
```



# 谢谢

