# 逆向分析技术

南京邮电大学
计算机学院信安系
2020-2021-2

## 课程信息

- 课程性质：专业限选课
- 学时：32（12学时实验）
- 考核：30%平时+70%期末
- 任课教师：朱枫

## OBE教学理念

——Outcome based education

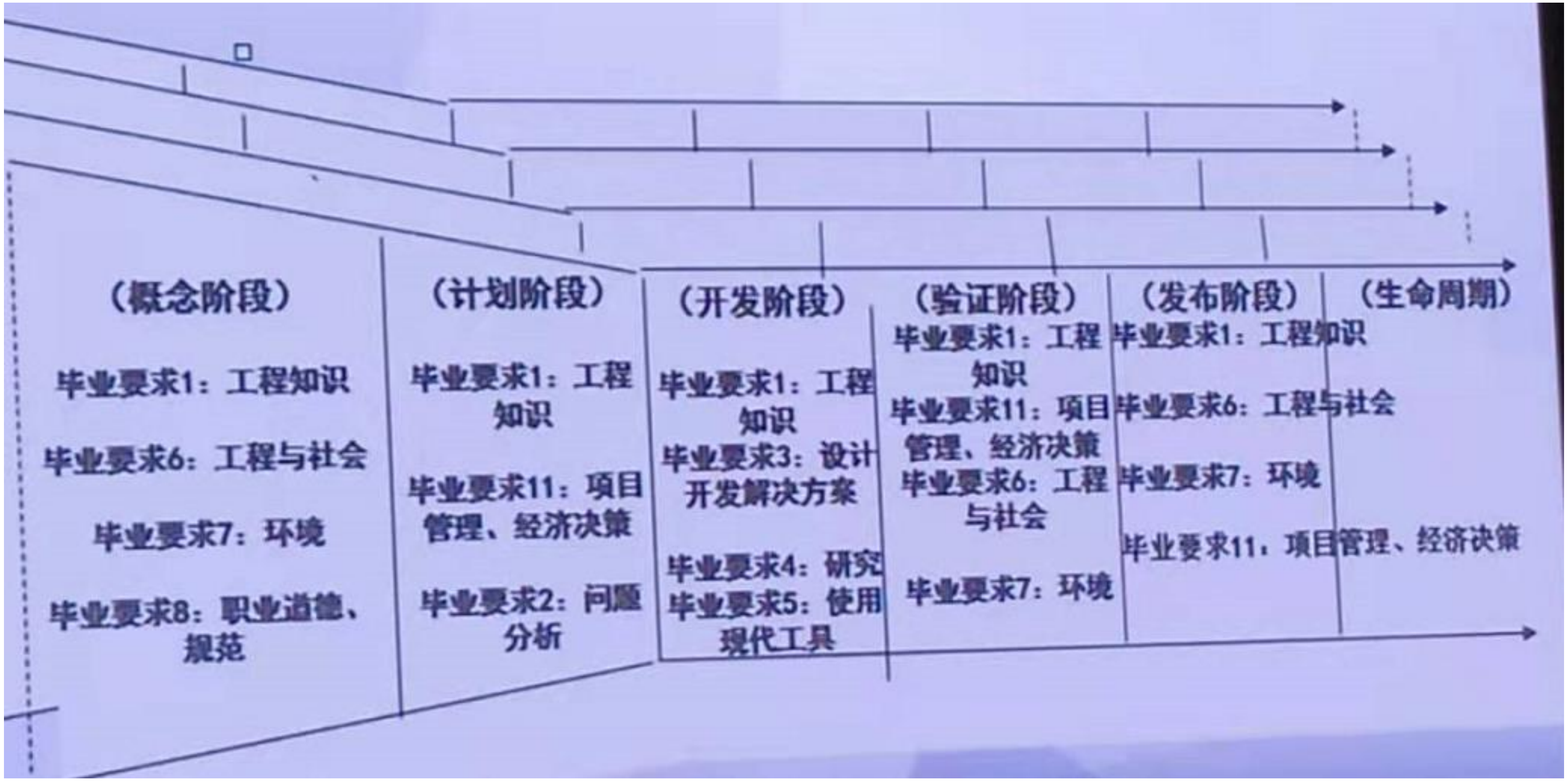
# 工程认证课程目标

- 课程目标3.1：使学生具有一定的自学能力和信息获取能力
- 课程目标3.2：使学生具有进行程序逆向分析，阅读复杂程序的能力
- 课程目标3.3：使学生具有对实际逆向分析问题研究分析、设计解决方案的能力

# 课程目标与OBE毕业要求

| 毕业要求 | 指标点 | 课程目标 |
|---|---|---|
| **4、研究** | 4-3-M 针对设计或开发的解决方案，能够基于信息安全领域科学原理对其进行研究，并能够通过理论证明、实验仿真或者系统实现等多种科学方案说明其有效性、合理性，并对解决方案的实施质量进行分析，通过信息综合得到合理有效的结论 | 3.1 |
| **5、使用现代工具** | 5-3-M 能够分析比较所使用的技术、资源和工具的优势和不足，并理解与表述问题解决方案的局限性。 | 3.2 |
| **3、设计/开发解决方案** | 3-3-M 充分理解信息安全领域软硬件系统的基础上，能够设计或开发满足特定需求和约束条件的信息安全系统、模块或算法流程，并能够进行系统级优化。 | 3.3 |

# 毕业要求与产品生命周期



（概念阶段）

毕业要求1：工程知识

毕业要求6：工程与社会

毕业要求7：环境

毕业要求8：职业道德、规范

（计划阶段）

毕业要求1：工程知识

毕业要求11：项目管理、经济决策

毕业要求2：问题分析

（开发阶段）

毕业要求1：工程知识

毕业要求3：设计开发解决方案

毕业要求4：研究

毕业要求5：使用现代工具

（验证阶段）

毕业要求1：工程知识

毕业要求11：项目管理、经济决策

毕业要求6：工程与社会

毕业要求7：环境

（发布阶段）

毕业要求1：工程知识

毕业要求6：工程与社会

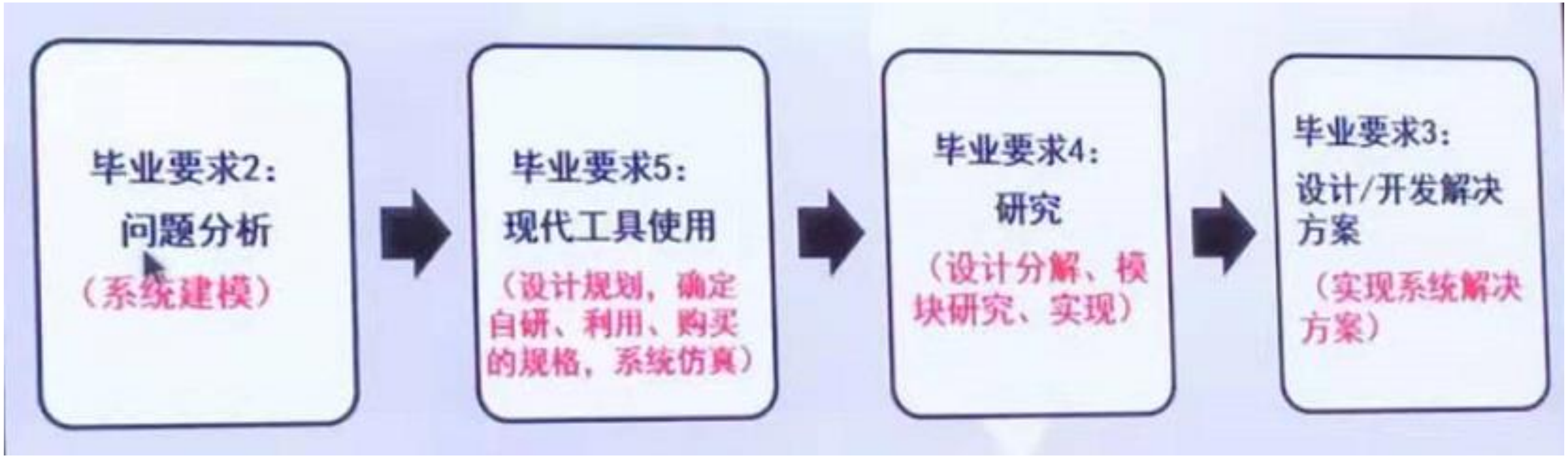
毕业要求7：环境

毕业要求11：项目管理、经济决策

（生命周期）

# 毕业要求与产品生命周期

- 毕业要求2、3、4、5的能力协同



## 逆向分析技术基本概念与应用背景

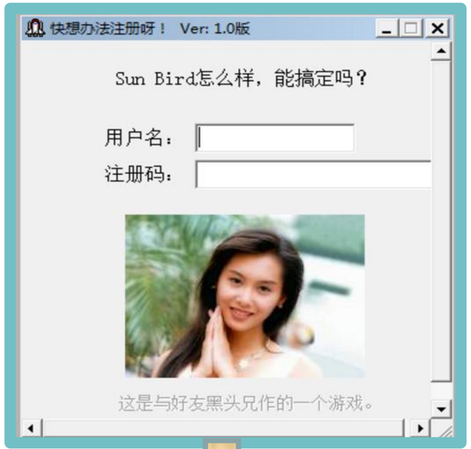
## 本次课程支撑的毕业要求指标点

- 毕业要求4-3：

针对设计或开发的解决方案，能够基于信息安全领域科学原理对其进行研究，并能够通过理论证明、实验仿真或者系统实现等多种科学方案说明其有效性、合理性，并对解决方案的实施质量进行分析，通过信息综合得到合理有效的结论

# 二进制安全应用背景

- Crack (程序破解)

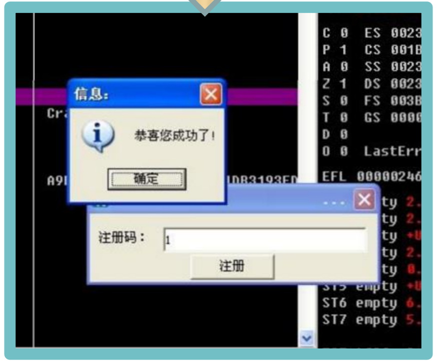

快想办法注册呀！ Ver: 1.0版

Sun Bird怎么样，能搞定吗？

用户名：

注册码：

这是与好友黑头兄作的一个游戏。


```
C 0  ES 0023
P 1  CS 001B
A 0  SS 0023
Z 1  DS 0023
S 0  FS 003B
T 0  GS 0000
D 0
O 0  LastErr

EFL 00000246

ST6 empty 6.
ST7 empty 5.
```

信息：

恭喜您成功了！

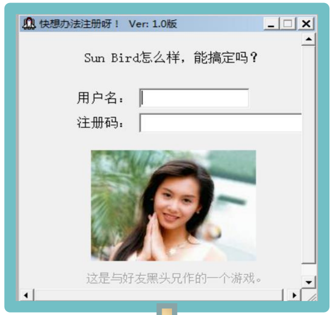
确定

注册码： 1

注册

# 二进制安全应用背景

- Reverse(程序逆向)

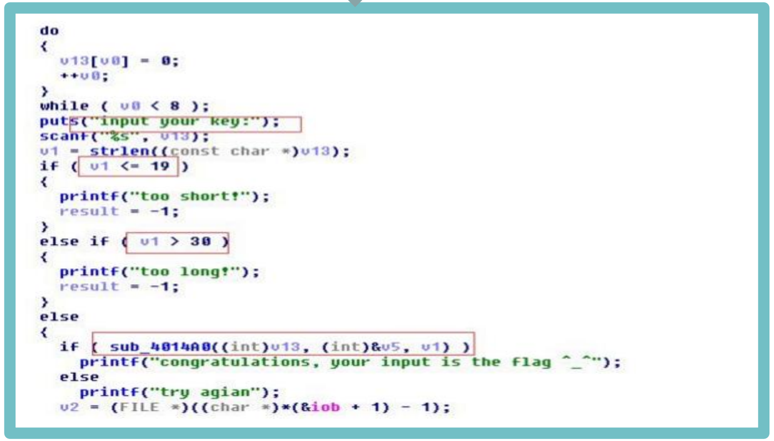


Sun Bird怎么样，能搞定吗？

用户名：

注册码：

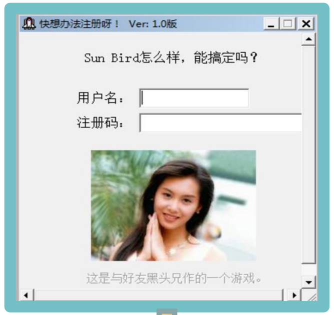
这是与好友黑头兄作的一个游戏。



```
do
{
  v13[v0] = 0;
  ++v0;
}
while ( v0 < 8 );
puts("input your key:");
scanf("%s", v13);
v1 = strlen((const char *)v13);
if ( v1 <= 19 )
{
  printf("too short!");
  result = -1;
}
else if ( v1 > 30 )
{
  printf("too long!");
  result = -1;
}
else
{
  if ( sub_4014A0((int)v13, (int)&v5, v1) )
    printf("congratulations, your input is the flag ^_^");
  else
    printf("try agian");
  v2 = (FILE *)((char *)*(&iob + 1) - 1);
```

# 二进制安全应用背景

- Unpack(程序脱壳)

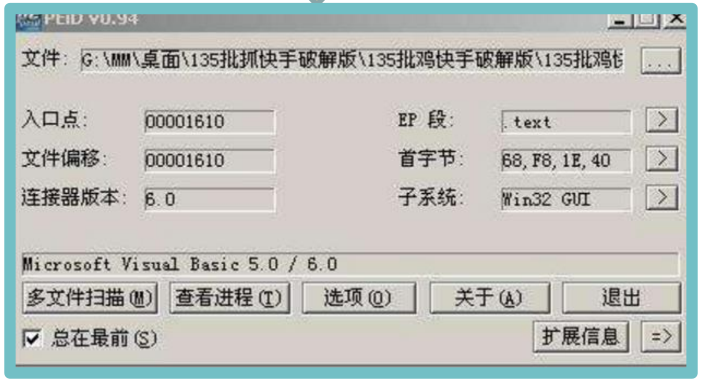

快想办法注册呀！ Ver: 1.0版

Sun Bird怎么样，能搞定吗？

用户名：

注册码：

这是与好友黑头兄作的一个游戏。


PEiD v0.94

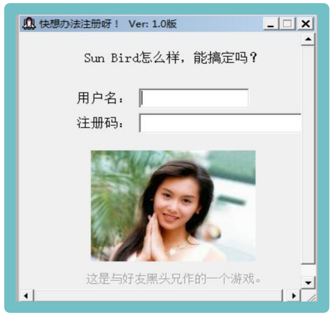
文件：G:\MM\桌面\135批抓快手破解版\135批鸡快手破解版\135批鸡

| | | | |
|---|---|---|---|
| 入口点： | 00001610 | EP 段： | .text |
| 文件偏移： | 00001610 | 首字节： | 68, F8, 1E, 40 |
| 连接器版本： | 6.0 | 子系统： | Win32 GUI |

Microsoft Visual Basic 5.0 / 6.0

# 二进制安全应用背景

- Pwn



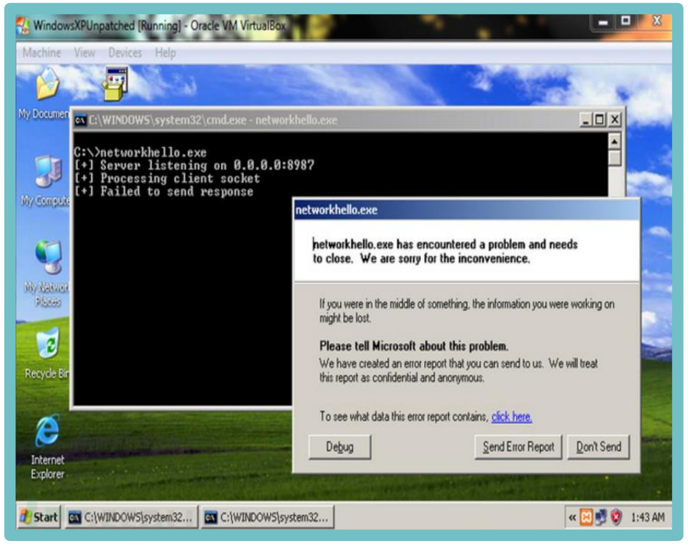
Sun Bird怎么样，能搞定吗？

用户名：

注册码：

这是与好友黑头兄作的一个游戏。

stack-overflow



```
C:\>networkhello.exe
[+] Server listening on 0.0.0.0:8987
[+] Processing client socket
[+] Failed to send response
```

networkhello.exe

**networkhello.exe has encountered a problem and needs to close. We are sorry for the inconvenience.**

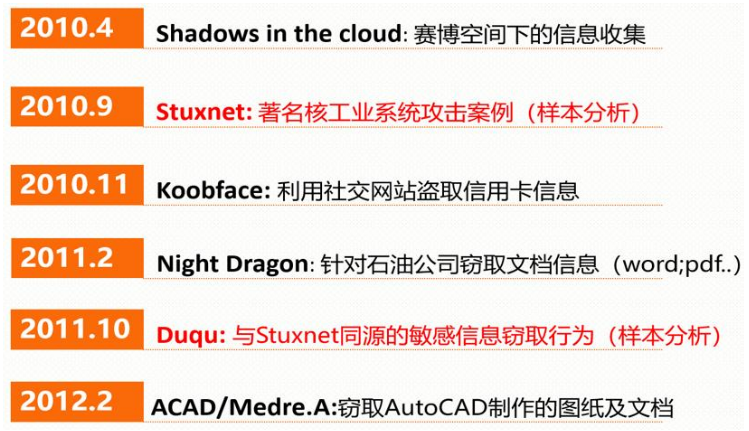
If you were in the middle of something, the information you were working on might be lost.

**Please tell Microsoft about this problem.**

We have created an error report that you can send to us. We will treat this report as confidential and anonymous.

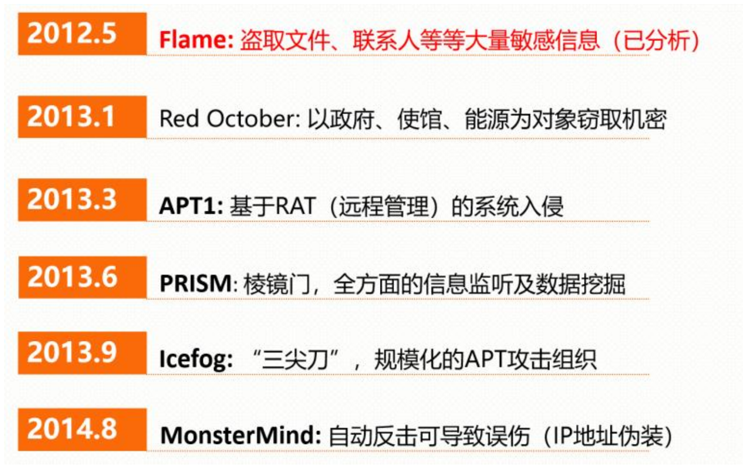
To see what data this error report contains, click here.

# 二进制安全应用背景
# 现实案例(1)



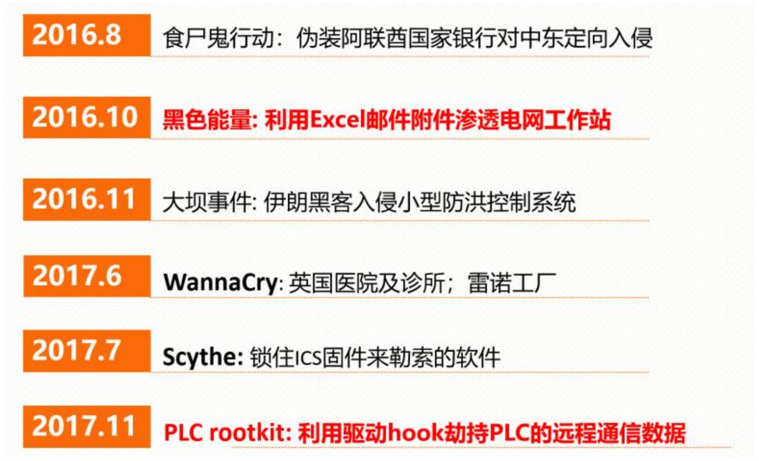
| | |
|---|---|
| 2010.4 | Shadows in the cloud: 赛博空间下的信息收集 |
| 2010.9 | Stuxnet: 著名核工业系统攻击案例（样本分析） |
| 2010.11 | Koobface: 利用社交网站盗取信用卡信息 |
| 2011.2 | Night Dragon: 针对石油公司窃取文档信息（word;pdf..） |
| 2011.10 | Duqu: 与Stuxnet同源的敏感信息窃取行为（样本分析） |
| 2012.2 | ACAD/Medre.A:窃取AutoCAD制作的图纸及文档 |

## 二进制安全应用背景
## 现实案例(2)



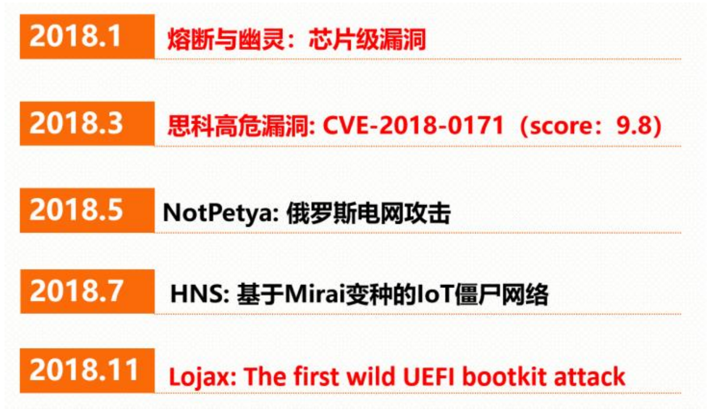
| | |
|---|---|
| 2012.5 | **Flame:** 盗取文件、联系人等等大量敏感信息（已分析） |
| 2013.1 | Red October: 以政府、使馆、能源为对象窃取机密 |
| 2013.3 | **APT1:** 基于RAT（远程管理）的系统入侵 |
| 2013.6 | **PRISM**: 棱镜门，全方面的信息监听及数据挖掘 |
| 2013.9 | **Icefog:** “三尖刀”，规模化的APT攻击组织 |
| 2014.8 | **MonsterMind:** 自动反击可导致误伤（IP地址伪装） |

# 二进制安全应用背景
# 现实案例(3)



| | |
|---|---|
| 2016.8 | 食尸鬼行动：伪装阿联酋国家银行对中东定向入侵 |
| 2016.10 | **黑色能量: 利用Excel邮件附件渗透电网工作站** |
| 2016.11 | 大坝事件: 伊朗黑客入侵小型防洪控制系统 |
| 2017.6 | **WannaCry**: 英国医院及诊所；雷诺工厂 |
| 2017.7 | **Scythe:** 锁住ICS固件来勒索的软件 |
| 2017.11 | **PLC rootkit: 利用驱动hook劫持PLC的远程通信数据** |

# 二进制安全应用背景
# 现实案例(4)



| | |
|---|---|
| 2018.1 | 熔断与幽灵：芯片级漏洞 |
| 2018.3 | 思科高危漏洞: CVE-2018-0171（score：9.8) |
| 2018.5 | NotPetya: 俄罗斯电网攻击 |
| 2018.7 | HNS: 基于Mirai变种的IoT僵尸网络 |
| 2018.11 | Lojax: The first wild UEFI bootkit attack |

## 相关基础 （1）

- 语言基础
  - 汇编：逆向之源，漏洞之本
  - C语言：算法实现，目标恢复
  - python：脚本处理，高效轻量

## 相关基础 （2）

- 操作系统
  - Windows
  - Linux
  - Android
  - Others

## 相关工具及平台（1）

Reverse Engineer
IDA; OllyDbg; Windbg;
gdb; Vmware; Winhex...

# 相关工具及平台（2）

Penetration test

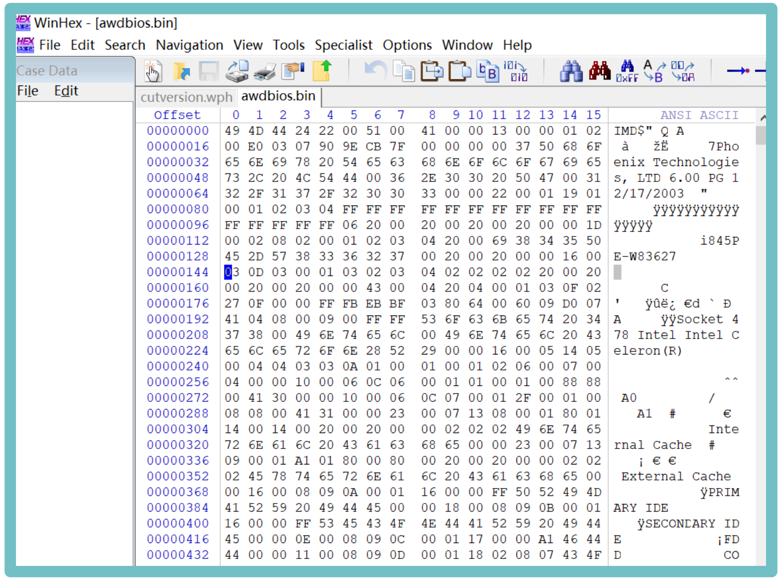
Metasploit; W3AF;

NMAP...

## 相关工具及平台（3）

# 二进制查看工具

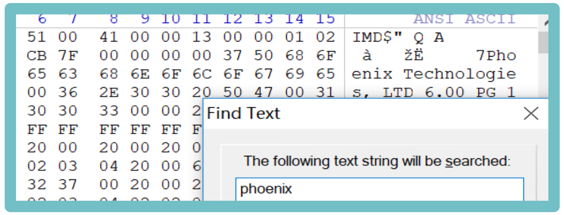
- Winhex/Ultraedit/…


WinHex - [awdbios.bin]
File Edit Search Navigation View Tools Specialist Options Window Help
Case Data
File Edit
cutversion.wph
awdbios.bin

```
 Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15     ANSI ASCII
00000000   49 4D 44 24 22 00 51 00  41 00 00 13 00 00 01 02  IMD$" Q A
00000016   00 E0 03 07 90 9E CB 7F  00 00 00 00 37 50 68 6F   à   žË     7Pho
00000032   65 6E 69 78 20 54 65 63  68 6E 6F 6C 6F 67 69 65  enix Technologie
00000048   73 2C 20 4C 54 44 00 36  2E 30 30 20 50 47 00 31  s, LTD 6.00 PG 1
00000064   32 2F 31 37 2F 32 30 30  33 00 00 22 00 01 19 01  2/17/2003  "
00000080   00 01 02 03 04 FF FF FF  FF FF FF FF FF FF FF FF       ÿÿÿÿÿÿÿÿÿÿÿ
00000096   FF FF FF FF FF 06 20 00  20 00 20 00 20 00 00 1D  ÿÿÿÿÿ
00000112   00 02 08 02 00 01 02 03  04 20 00 69 38 34 35 50             i845P
00000128   45 2D 57 38 33 36 32 37  00 20 00 20 00 00 16 00  E-W83627
00000144   03 0D 03 00 01 03 02 03  04 02 02 02 02 20 00 20
00000160   00 20 00 20 00 00 43 00  04 20 04 00 01 03 0F 02        C
00000176   27 0F 00 00 FF FB EB BF  03 80 64 00 60 09 D0 07  '   ÿûë¿ €d ` Ð
00000192   41 04 08 00 09 00 FF FF  53 6F 63 6B 65 74 20 34  A     ÿÿSocket 4
00000208   37 38 00 49 6E 74 65 6C  00 49 6E 74 65 6C 20 43  78 Intel Intel C
00000224   65 6C 65 72 6F 6E 28 52  29 00 00 16 00 05 14 05  eleron(R)
00000240   00 04 04 03 03 0A 01 00  01 00 01 02 06 00 07 00
00000256   04 00 00 10 00 06 0C 06  00 01 01 00 01 00 88 88                ^^
00000272   00 41 30 00 00 10 00 06  0C 07 00 01 2F 00 01 00   A0          /
00000288   08 08 00 41 31 00 00 23  00 07 13 08 00 01 80 01    A1   #       €
00000304   14 00 14 00 20 00 20 00  00 02 02 02 49 6E 74 65              Inte
00000320   72 6E 61 6C 20 43 61 63  68 65 00 00 23 00 07 13  rnal Cache  #
00000336   09 00 01 A1 01 80 00 80  00 20 00 20 00 00 02 02     ¡ € €
00000352   02 45 78 74 65 72 6E 61  6C 20 43 61 63 68 65 00   External Cache
00000368   00 16 00 08 09 0A 00 01  16 00 00 FF 50 52 49 4D             ÿPRIM
00000384   41 52 59 20 49 44 45 00  00 18 00 08 09 0B 00 01  ARY IDE
00000400   16 00 00 FF 53 45 43 4F  4E 44 41 52 59 20 49 44     ÿSECONDARY ID
00000416   45 00 00 0E 00 08 09 0C  00 01 17 00 00 A1 46 44  E            ¡FD
00000432   44 00 00 11 00 08 09 0D  00 01 18 02 08 07 43 4F  D             CO
```

# 二进制查看工具

- Winhex/Ultraedit/…

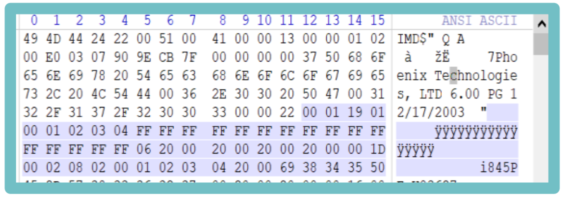


```
 6  7   8  9 10 11 12 13 14 15   ANSI ASCII
51 00  41 00 00 13 00 00 01 02  IMD$" Q A
CB 7F  00 00 00 00 37 50 68 6F   à   žË      7Pho
65 63  68 6E 6F 6C 6F 67 69 65  enix Technologie
00 36  2E 30 30 20 50 47 00 31  s, LTD 6.00 PG 1
30 30  33 00 00 2
FF FF  FF FF FF F
20 00  20 00 20 0
02 03  04 20 00 6
32 37  00 20 00 2
```

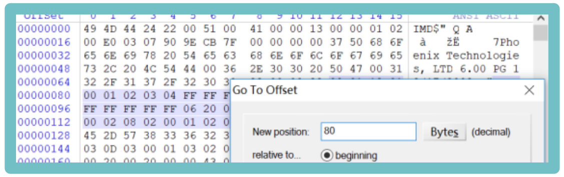
Find Text

The following text string will be searched:

phoenix



```
 0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15   ANSI ASCII
49 4D 44 24 22 00 51 00  41 00 00 13 00 00 01 02  IMD$" Q A
00 E0 03 07 90 9E CB 7F  00 00 00 00 37 50 68 6F   à   žË      7Pho
65 6E 69 78 20 54 65 63  68 6E 6F 6C 6F 67 69 65  enix Technologie
73 2C 20 4C 54 44 00 36  2E 30 30 20 50 47 00 31  s, LTD 6.00 PG 1
32 2F 31 37 2F 32 30 30  33 00 00 22 00 01 19 01  2/17/2003  "
00 01 02 03 04 FF FF FF  FF FF FF FF FF FF FF FF       ÿÿÿÿÿÿÿÿÿÿÿ
FF FF FF FF FF 06 20 00  20 00 20 00 20 00 00 1D  ÿÿÿÿÿ
00 02 08 02 00 01 02 03  04 20 00 69 38 34 35 50              i845P
```



```
Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15   ANSI ASCII
00000000  49 4D 44 24 22 00 51 00  41 00 00 13 00 00 01 02  IMD$" Q A
00000016  00 E0 03 07 90 9E CB 7F  00 00 00 00 37 50 68 6F   à   žË      7Pho
00000032  65 6E 69 78 20 54 65 63  68 6E 6F 6C 6F 67 69 65  enix Technologie
00000048  73 2C 20 4C 54 44 00 36  2E 30 30 20 50 47 00 31  s, LTD 6.00 PG 1
00000064  32 2F 31 37 2F 32 30 3
00000080  00 01 02 03 04 FF FF F
00000096  FF FF FF FF FF 06 20 0
00000112  00 02 08 02 00 01 02 0
00000128  45 2D 57 38 33 36 32 3
00000144  03 0D 03 00 01 03 02 0
```

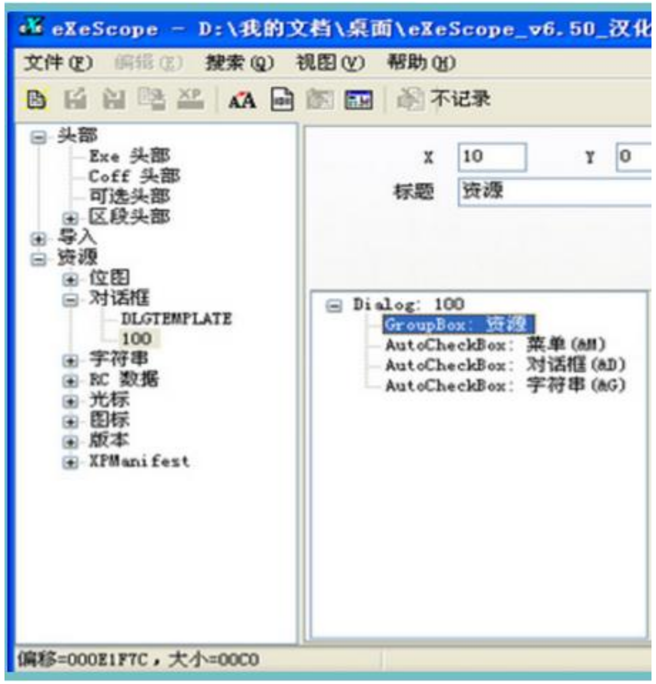
Go To Offset

New position: 80 Bytes (decimal)

relative to... [✓ beginning]

# 二进制查看工具

- exeScope
  – 查看文件结构框架


eXeScope - D:\我的文档\桌面\eXeScope_v6.50_汉化
文件(F) 编辑(E) 搜索(Q) 视图(V) 帮助(H)
不记录
X
Y
标题

```
⊟ 头部
    Exe 头部
    Coff 头部
    可选头部
  ⊞ 区段头部
⊞ 导入
⊟ 资源
  ⊞ 位图
  ⊟ 对话框
      DLGTEMPLATE
      100
  ⊞ 字符串
  ⊞ RC 数据
  ⊞ 光标
  ⊞ 图标
  ⊞ 版本
  ⊞ XPManifest
```

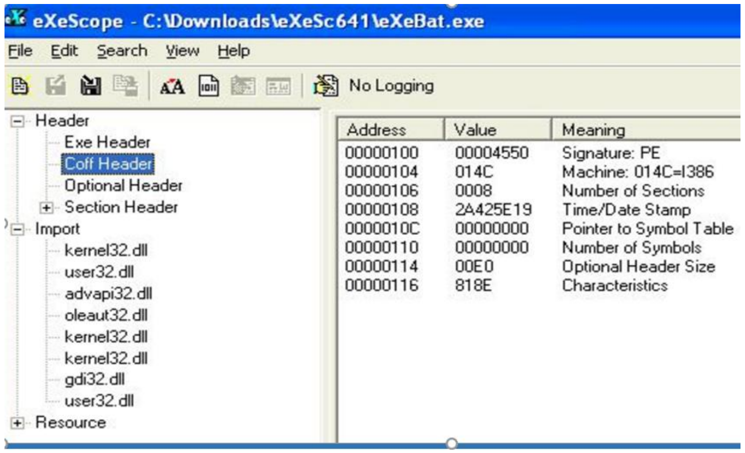
X: 10 Y: 0
标题: 资源

```
⊟ Dialog: 100
    GroupBox: 资源
    AutoCheckBox: 菜单(&M)
    AutoCheckBox: 对话框(&D)
    AutoCheckBox: 字符串(&G)
```

偏移=000E1F7C，大小=00C0

# 二进制查看工具

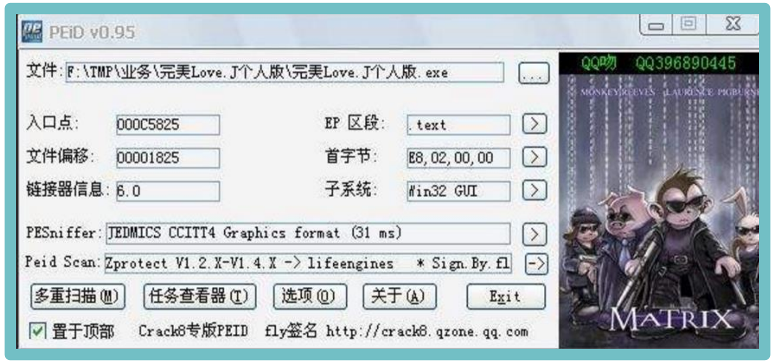
- exeScope
  - 文件结构解析



| Address | Value | Meaning |
|---|---|---|
| 00000100 | 00004550 | Signature: PE |
| 00000104 | 014C | Machine: 014C=I386 |
| 00000106 | 0008 | Number of Sections |
| 00000108 | 2A425E19 | Time/Date Stamp |
| 0000010C | 00000000 | Pointer to Symbol Table |
| 00000110 | 00000000 | Number of Symbols |
| 00000114 | 00E0 | Optional Header Size |
| 00000116 | 818E | Characteristics |

# 二进制查看工具
# --查壳工具

- PEiD
  - 识别zprotect



PEiD v0.95

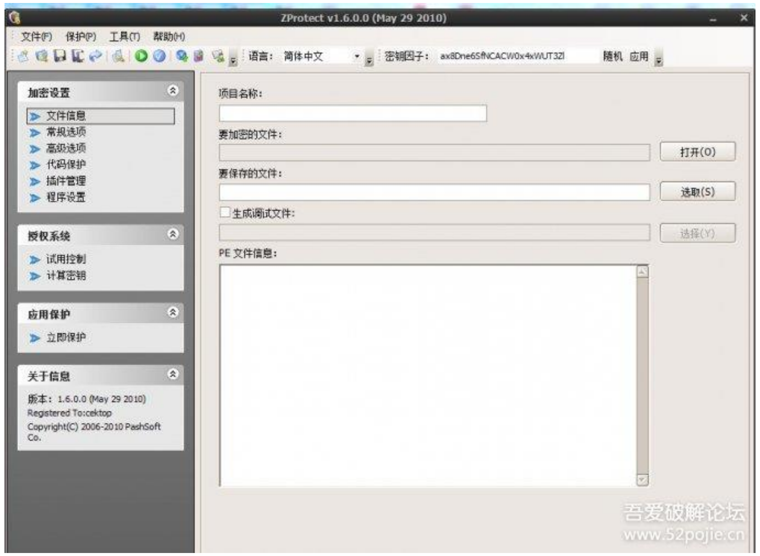
文件: F:\TMP\业务\完美Love.J个人版\完美Love.J个人版.exe

入口点: 000C5825 | EP 区段: .text
文件偏移: 00001825 | 首字节: E8,02,00,00
链接器信息: 6.0 | 子系统: Win32 GUI

PESniffer: JEDMICS CCITT4 Graphics format (31 ms)
Peid Scan: Zprotect V1.2.X-V1.4.X -> lifeengines * Sign.By.fl

# 二进制查看工具
# --查壳工具

- zprotect加壳
  - 虚拟机加密
  - 代码乱序
  - 动态代码结构.

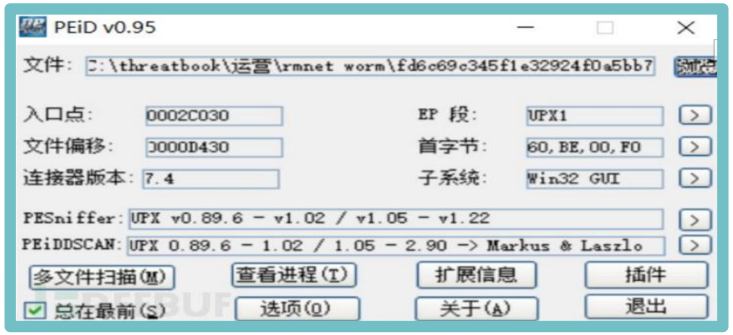


语言: 简体中文
密钥因子: ax8Dne6SfNCACW0x4xWUT3Zl

版本: 1.6.0.0 (May 29 2010)
Registered To:cektop
Copyright(C) 2006-2010 PashSoft Co.

## 二进制查看工具
## --查壳工具

- PEiD
  - 识别UPX



PEiD v0.95

文件: C:\threatbook\运营\rmnet worm\fd6c69c345f1e32924f0a5bb7

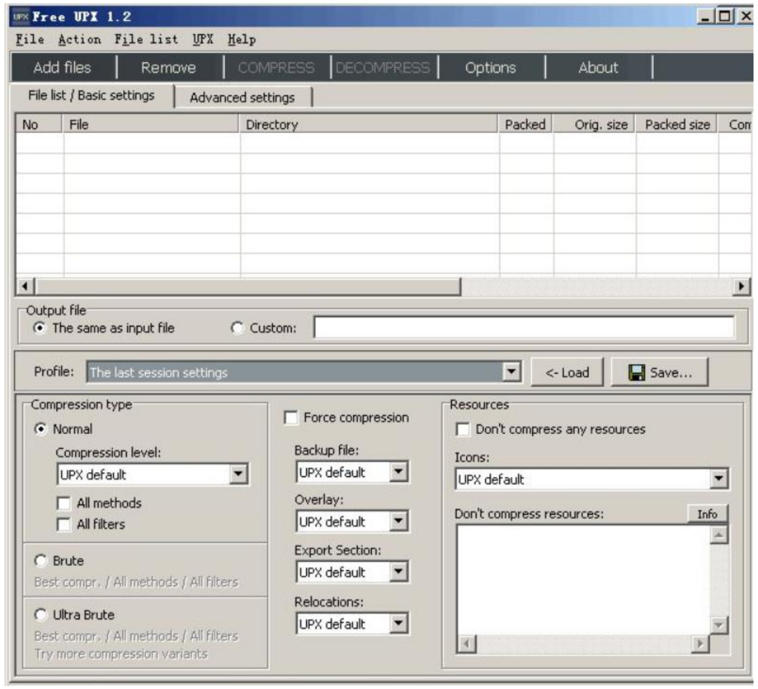
| | | | |
|---|---|---|---|
| 入口点: | 0002C030 | EP 段: | UPX1 |
| 文件偏移: | 0000D430 | 首字节: | 60, BE, 00, F0 |
| 连接器版本: | 7.4 | 子系统: | Win32 GUI |

PESniffer: UPX v0.89.6 - v1.02 / v1.05 - v1.22

PEiDDSCAN: UPX 0.89.6 - 1.02 / 1.05 - 2.90 -> Markus & Laszlo

## 二进制查看工具
## --查壳工具

- UPX加壳
  - 压缩壳



Output file
[✓ The same as input file] Custom:

Profile: The last session settings

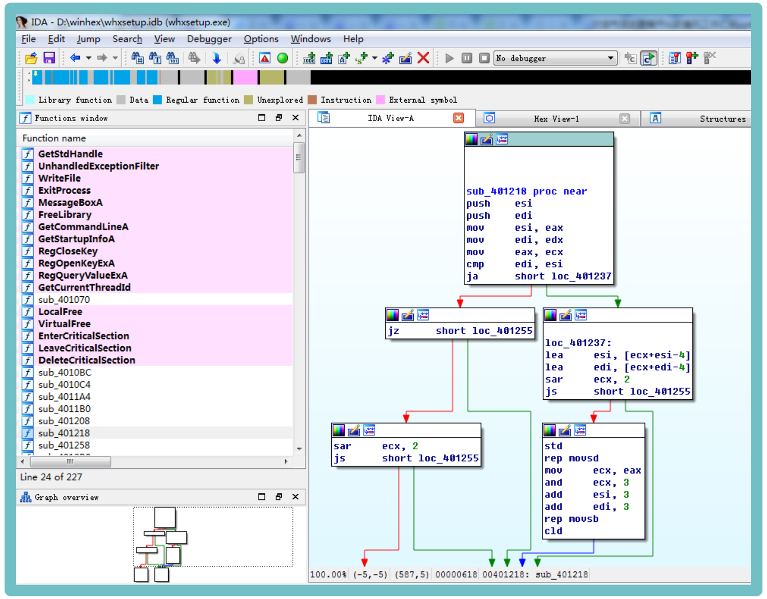
Compression type
[✓ Normal]
Compression level: UPX default
All methods
All filters
Brute
Best compr. / All methods / All filters
Ultra Brute
Best compr. / All methods / All filters
Try more compression variants

Force compression
Backup file: UPX default
Overlay: UPX default
Export Section: UPX default
Relocations: UPX default

Resources
Don't compress any resources
Icons: UPX default
Don't compress resources:

# 静态分析工具
# IDA

- DataRescue出品
- 交互式
- 多处理器


IDA - D:\winhex\whxsetup.idb (whxsetup.exe)
File Edit Jump Search View Debugger Options Windows Help
No debugger
Library function Data Regular function Unexplored Instruction External symbol
Functions window
IDA View-A
Hex View-1
Structures
Graph overview

Function name
GetStdHandle
UnhandledExceptionFilter
WriteFile
ExitProcess
MessageBoxA
FreeLibrary
GetCommandLineA
GetStartupInfoA
RegCloseKey
RegOpenKeyExA
RegQueryValueExA
GetCurrentThreadId
sub_401070
LocalFree
VirtualFree
EnterCriticalSection
LeaveCriticalSection
DeleteCriticalSection
sub_4010BC
sub_4010C4
sub_4011A4
sub_4011B0
sub_401208
sub_401218
sub_401258

Line 24 of 227

```
sub_401218 proc near
push    esi
push    edi
mov     esi, eax
mov     edi, edx
mov     eax, ecx
cmp     edi, esi
ja      short loc_401237
```

```
jz      short loc_401255
```

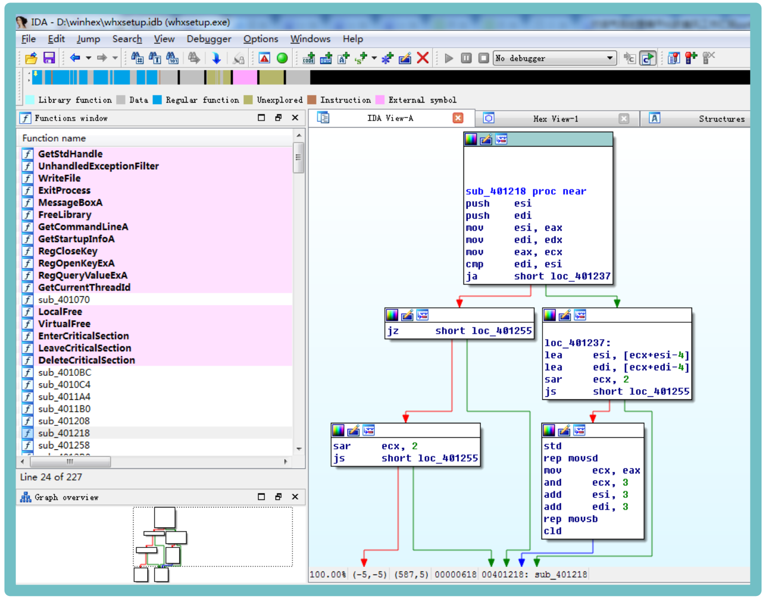
```
loc_401237:
lea     esi, [ecx+esi-4]
lea     edi, [ecx+edi-4]
sar     ecx, 2
js      short loc_401255
```

```
sar     ecx, 2
js      short loc_401255
```

```
std
rep movsd
mov     ecx, eax
and     ecx, 3
add     esi, 3
add     edi, 3
rep movsb
cld
```

```
100.00% (-5,-5) (587,5) 00000618 00401218: sub_401218
```

## 静态分析工具
## IDA



Function name
GetStdHandle
UnhandledExceptionFilter
WriteFile
ExitProcess
MessageBoxA
FreeLibrary
GetCommandLineA
GetStartupInfoA
RegCloseKey
RegOpenKeyExA
RegQueryValueExA
GetCurrentThreadId
sub_401070
LocalFree
VirtualFree
EnterCriticalSection
LeaveCriticalSection
DeleteCriticalSection
sub_4010BC
sub_4010C4
sub_4011A4
sub_4011B0
sub_401208
sub_401218
sub_401258

Line 24 of 227

```
sub_401218 proc near
push    esi
push    edi
mov     esi, eax
mov     edi, edx
mov     eax, ecx
cmp     edi, esi
ja      short loc_401237
```

```
jz      short loc_401255
```

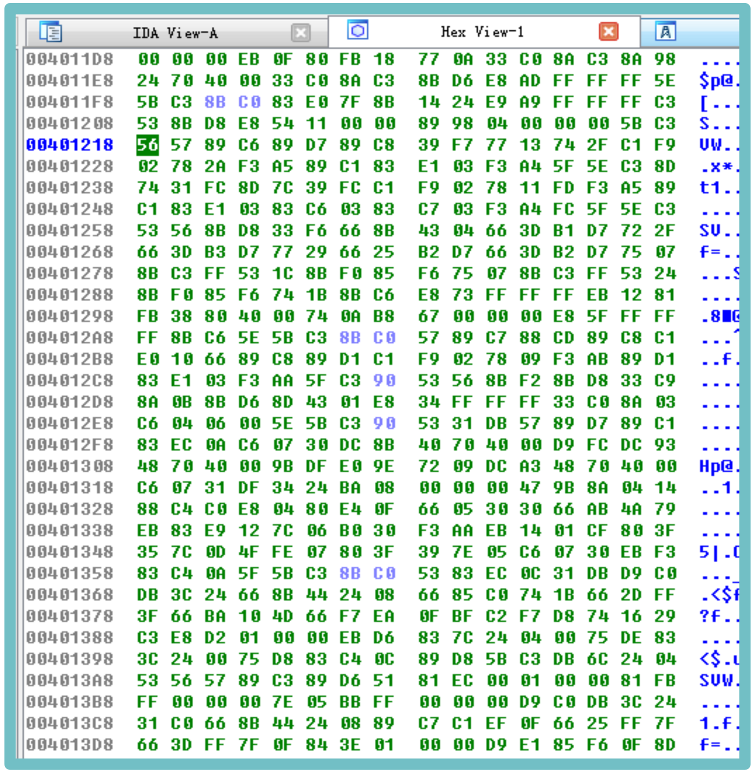
```
loc_401237:
lea     esi, [ecx+esi-4]
lea     edi, [ecx+edi-4]
sar     ecx, 2
js      short loc_401255
```

```
sar     ecx, 2
js      short loc_401255
```

```
std
rep movsd
mov     ecx, eax
and     ecx, 3
add     esi, 3
add     edi, 3
rep movsb
cld
```

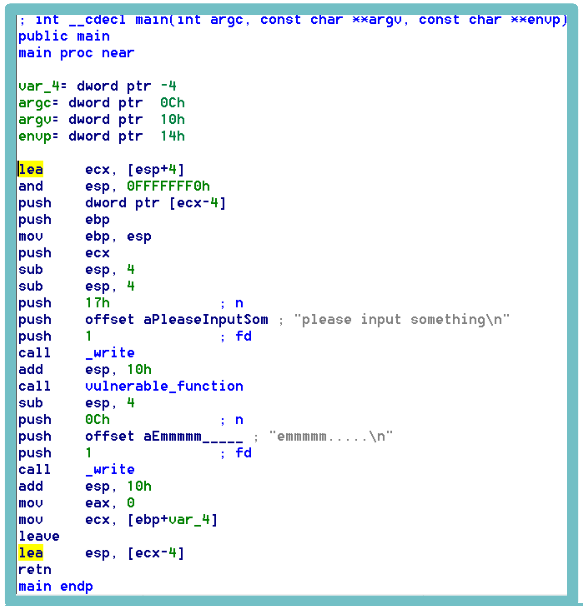
```
100.00% (-5,-5) (587,5) 00000618 00401218: sub_401218
```

# 静态分析工具
## IDA

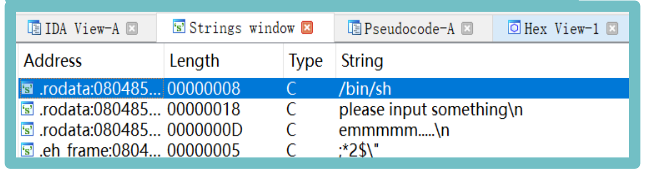


```
004011D8  00 00 00 EB 0F 80 FB 18  77 0A 33 C0 8A C3 8A 98  ....
004011E8  24 70 40 00 33 C0 8A C3  8B D6 E8 AD FF FF FF 5E  $p@.
004011F8  5B C3 8B C0 83 E0 7F 8B  14 24 E9 A9 FF FF FF C3  [...
00401208  53 8B D8 E8 54 11 00 00  89 98 04 00 00 00 5B C3  S...
00401218  56 57 89 C6 89 D7 89 C8  39 F7 77 13 74 2F C1 F9  VW..
00401228  02 78 2A F3 A5 89 C1 83  E1 03 F3 A4 5F 5E C3 8D  .x*.
00401238  74 31 FC 8D 7C 39 FC C1  F9 02 78 11 FD F3 A5 89  t1..
00401248  C1 83 E1 03 83 C6 03 83  C7 03 F3 A4 FC 5F 5E C3  ....
00401258  53 56 8B D8 33 F6 66 8B  43 04 66 3D B1 D7 72 2F  SV..
00401268  66 3D B3 D7 77 29 66 25  B2 D7 66 3D B2 D7 75 07  f=..
00401278  8B C3 FF 53 1C 8B F0 85  F6 75 07 8B C3 FF 53 24  ...S
00401288  8B F0 85 F6 74 1B 8B C6  E8 73 FF FF FF EB 12 81  ....
00401298  FB 38 80 40 00 74 0A B8  67 00 00 00 E8 5F FF FF  .8█@
004012A8  FF 8B C6 5E 5B C3 8B C0  57 89 C7 88 CD 89 C8 C1  ...^
004012B8  E0 10 66 89 C8 89 D1 C1  F9 02 78 09 F3 AB 89 D1  ..f.
004012C8  83 E1 03 F3 AA 5F C3 90  53 56 8B F2 8B D8 33 C9  ....
004012D8  8A 0B 8B D6 8D 43 01 E8  34 FF FF FF 33 C0 8A 03  ....
004012E8  C6 04 06 00 5E 5B C3 90  53 31 DB 57 89 D7 89 C1  ....
004012F8  83 EC 0A C6 07 30 DC 8B  40 70 40 00 D9 FC DC 93  ....
00401308  48 70 40 00 9B DF E0 9E  72 09 DC A3 48 70 40 00  Hp@.
00401318  C6 07 31 DF 34 24 BA 08  00 00 00 47 9B 8A 04 14  ..1.
00401328  88 C4 C0 E8 04 80 E4 0F  66 05 30 30 66 AB 4A 79  ....
00401338  EB 83 E9 12 7C 06 B0 30  F3 AA EB 14 01 CF 80 3F  ....
00401348  35 7C 0D 4F FE 07 80 3F  39 7E 05 C6 07 30 EB F3  5|.O
00401358  83 C4 0A 5F 5B C3 8B C0  53 83 EC 0C 31 DB D9 C0  ..._
00401368  DB 3C 24 66 8B 44 24 08  66 85 C0 74 1B 66 2D FF  .<$f
00401378  3F 66 BA 10 4D 66 F7 EA  0F BF C2 F7 D8 74 16 29  ?f..
00401388  C3 E8 D2 01 00 00 EB D6  83 7C 24 04 00 75 DE 83  ....
00401398  3C 24 00 75 D8 83 C4 0C  89 D8 5B C3 DB 6C 24 04  <$.u
004013A8  53 56 57 89 C3 89 D6 51  81 EC 00 01 00 00 81 FB  SVW.
004013B8  FF 00 00 00 7E 05 BB FF  00 00 00 D9 C0 DB 3C 24  ....
004013C8  31 C0 66 8B 44 24 08 89  C7 C1 EF 0F 66 25 FF 7F  1.f.
004013D8  66 3D FF 7F 0F 84 3E 01  00 00 D9 E1 85 F6 0F 8D  f=..
```

# 静态分析工具
# IDA

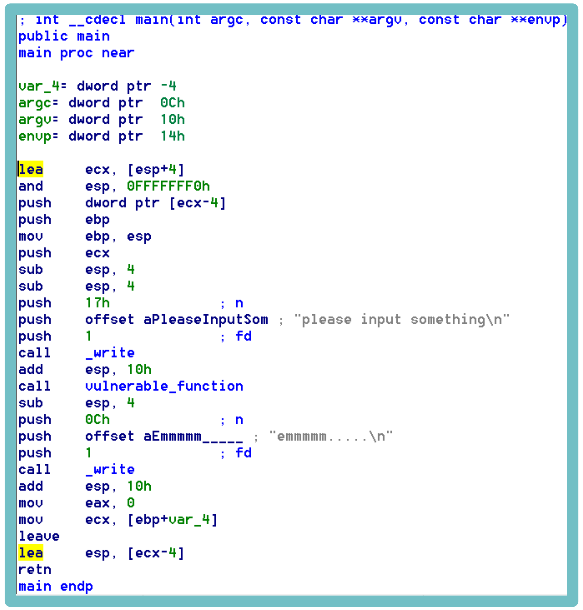


```
; int __cdecl main(int argc, const char **argv, const char **envp)
public main
main proc near

var_4= dword ptr -4
argc= dword ptr  0Ch
argv= dword ptr  10h
envp= dword ptr  14h

lea     ecx, [esp+4]
and     esp, 0FFFFFFF0h
push    dword ptr [ecx-4]
push    ebp
mov     ebp, esp
push    ecx
sub     esp, 4
sub     esp, 4
push    17h             ; n
push    offset aPleaseInputSom ; "please input something\n"
push    1               ; fd
call    _write
add     esp, 10h
call    vulnerable_function
sub     esp, 4
push    0Ch             ; n
push    offset aEmmmmm_____ ; "emmmmm.....\n"
push    1               ; fd
call    _write
add     esp, 10h
mov     eax, 0
mov     ecx, [ebp+var_4]
leave
lea     esp, [ecx-4]
retn
main endp
```

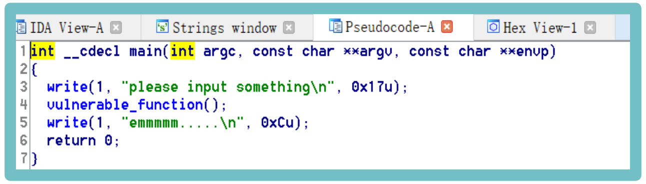


| Address | Length | Type | String |
|---|---|---|---|
| .rodata:080485... | 00000008 | C | /bin/sh |
| .rodata:080485... | 00000018 | C | please input something\n |
| .rodata:080485... | 0000000D | C | emmmmm.....\n |
| .eh_frame:0804... | 00000005 | C | ;*2$\" |

# 静态分析工具
# IDA

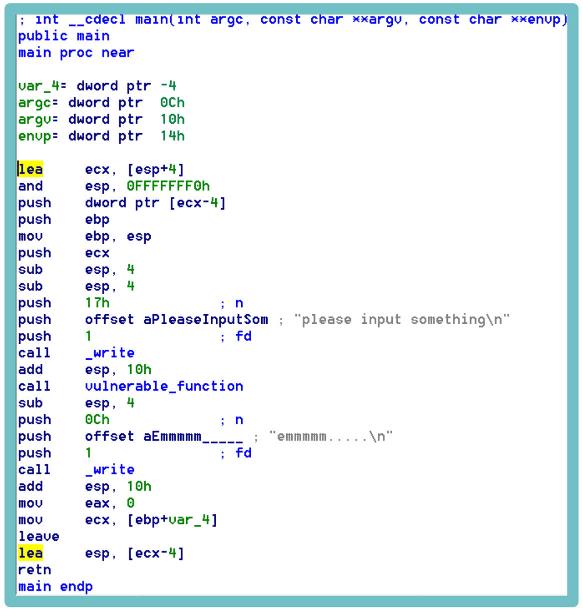


```
; int __cdecl main(int argc, const char **argv, const char **envp)
public main
main proc near

var_4= dword ptr -4
argc= dword ptr  0Ch
argv= dword ptr  10h
envp= dword ptr  14h

lea     ecx, [esp+4]
and     esp, 0FFFFFFF0h
push    dword ptr [ecx-4]
push    ebp
mov     ebp, esp
push    ecx
sub     esp, 4
sub     esp, 4
push    17h             ; n
push    offset aPleaseInputSom ; "please input something\n"
push    1               ; fd
call    _write
add     esp, 10h
call    vulnerable_function
sub     esp, 4
push    0Ch             ; n
push    offset aEmmmmm_____ ; "emmmmm.....\n"
push    1               ; fd
call    _write
add     esp, 10h
mov     eax, 0
mov     ecx, [ebp+var_4]
leave
lea     esp, [ecx-4]
retn
main endp
```

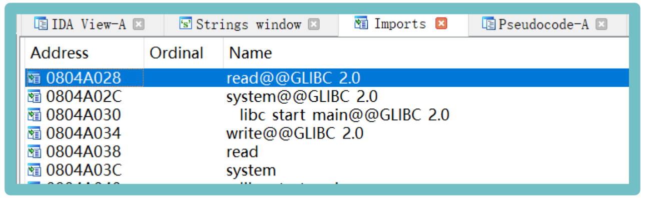


```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  write(1, "please input something\n", 0x17u);
  vulnerable_function();
  write(1, "emmmmm.....\n", 0xCu);
  return 0;
}
```

# 静态分析工具
# IDA

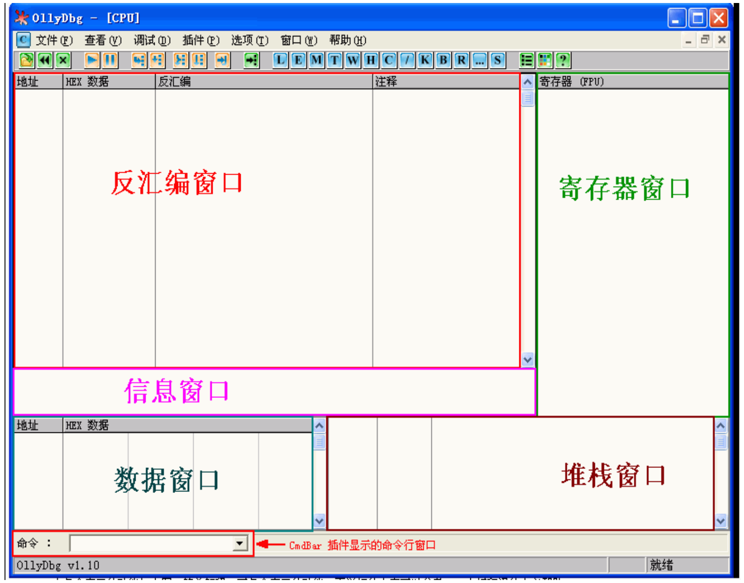


```
; int __cdecl main(int argc, const char **argv, const char **envp)
public main
main proc near

var_4= dword ptr -4
argc= dword ptr  0Ch
argv= dword ptr  10h
envp= dword ptr  14h

lea     ecx, [esp+4]
and     esp, 0FFFFFFF0h
push    dword ptr [ecx-4]
push    ebp
mov     ebp, esp
push    ecx
sub     esp, 4
sub     esp, 4
push    17h             ; n
push    offset aPleaseInputSom ; "please input something\n"
push    1               ; fd
call    _write
add     esp, 10h
call    vulnerable_function
sub     esp, 4
push    0Ch             ; n
push    offset aEmmmmm_____ ; "emmmmm.....\n"
push    1               ; fd
call    _write
add     esp, 10h
mov     eax, 0
mov     ecx, [ebp+var_4]
leave
lea     esp, [ecx-4]
retn
main endp
```

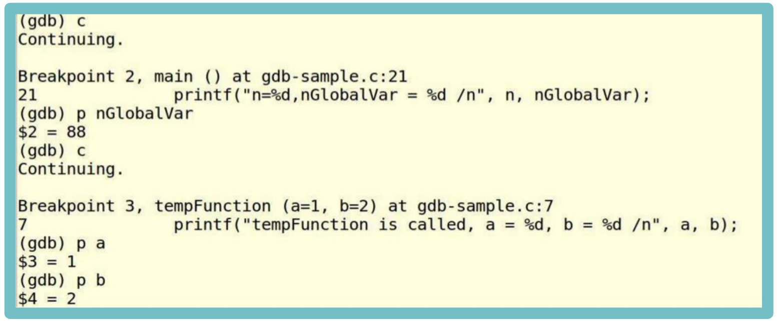


| Address | Ordinal | Name |
|---|---|---|
| 0804A028 | | read@@GLIBC 2.0 |
| 0804A02C | | system@@GLIBC 2.0 |
| 0804A030 | | libc start main@@GLIBC 2.0 |
| 0804A034 | | write@@GLIBC 2.0 |
| 0804A038 | | read |
| 0804A03C | | system |

# 动态分析工具
# OllyDbg

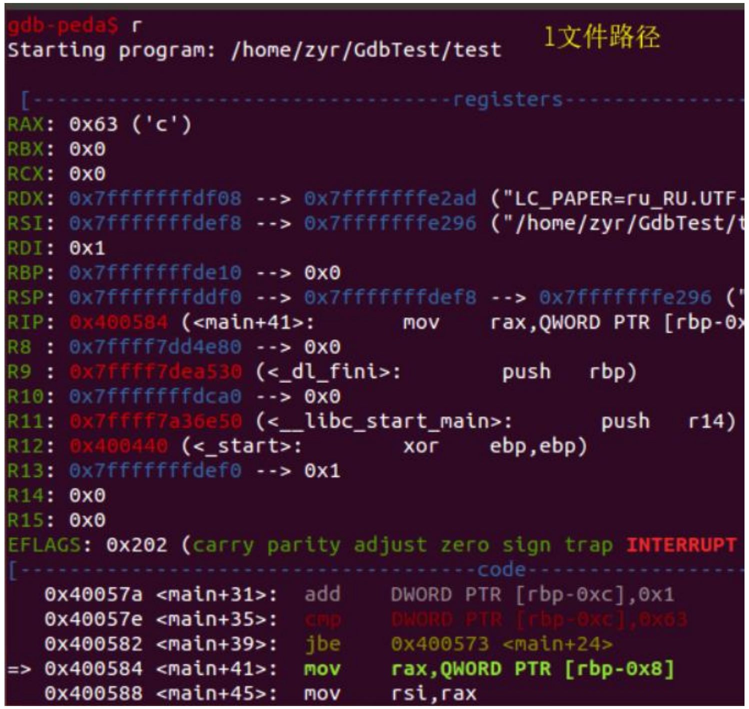


## Linux平台动态分析
## gdb



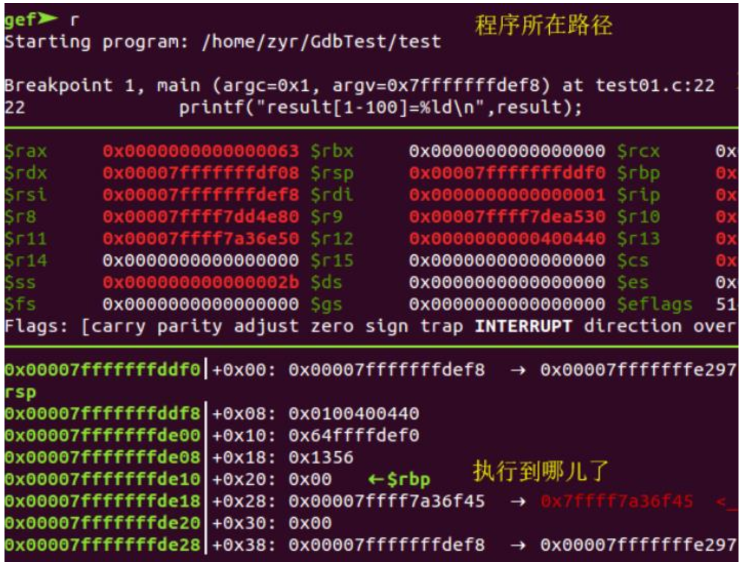
```
(gdb) c
Continuing.

Breakpoint 2, main () at gdb-sample.c:21
21              printf("n=%d,nGlobalVar = %d /n", n, nGlobalVar);
(gdb) p nGlobalVar
$2 = 88
(gdb) c
Continuing.

Breakpoint 3, tempFunction (a=1, b=2) at gdb-sample.c:7
7               printf("tempFunction is called, a = %d, b = %d /n", a, b);
(gdb) p a
$3 = 1
(gdb) p b
$4 = 2
```

# Linux平台动态分析
# gdb插件peda



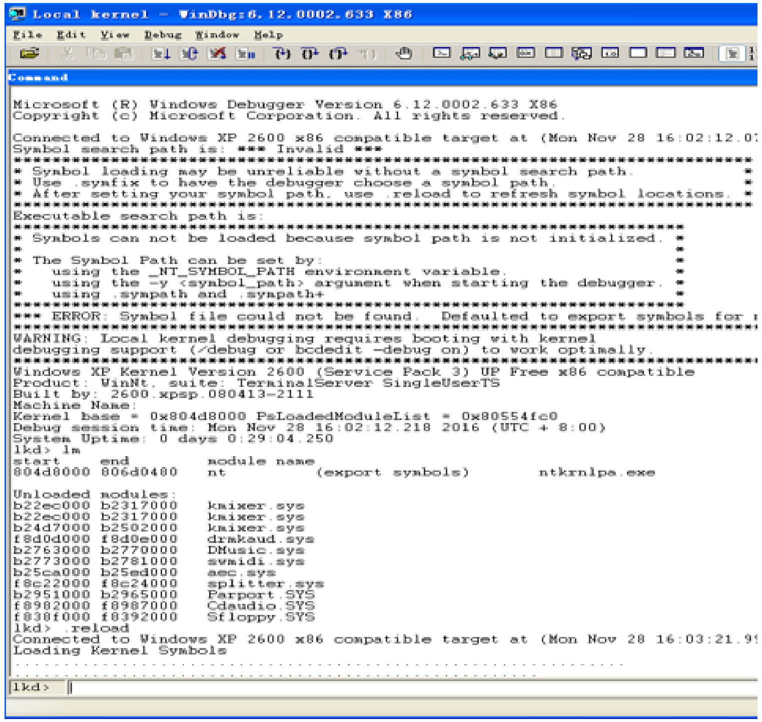
```
gdb-peda$ r
Starting program: /home/zyr/GdbTest/test
[----------------------------------registers---------------
RAX: 0x63 ('c')
RBX: 0x0
RCX: 0x0
RDX: 0x7fffffffdf08 --> 0x7fffffffe2ad ("LC_PAPER=ru_RU.UTF-
RSI: 0x7fffffffdef8 --> 0x7fffffffe296 ("/home/zyr/GdbTest/t
RDI: 0x1
RBP: 0x7fffffffde10 --> 0x0
RSP: 0x7fffffffddf0 --> 0x7fffffffdef8 --> 0x7fffffffe296 ("
RIP: 0x400584 (<main+41>:        mov    rax,QWORD PTR [rbp-0x
R8 : 0x7ffff7dd4e80 --> 0x0
R9 : 0x7ffff7dea530 (<_dl_fini>:        push   rbp)
R10: 0x7fffffffdca0 --> 0x0
R11: 0x7ffff7a36e50 (<__libc_start_main>:        push   r14)
R12: 0x400440 (<_start>:        xor    ebp,ebp)
R13: 0x7fffffffdef0 --> 0x1
R14: 0x0
R15: 0x0
EFLAGS: 0x202 (carry parity adjust zero sign trap INTERRUPT
[-------------------------------------code-----------------
   0x40057a <main+31>:  add    DWORD PTR [rbp-0xc],0x1
   0x40057e <main+35>:  cmp    DWORD PTR [rbp-0xc],0x63
   0x400582 <main+39>:  jbe    0x400573 <main+24>
=> 0x400584 <main+41>:  mov    rax,QWORD PTR [rbp-0x8]
   0x400588 <main+45>:  mov    rsi,rax
```

# Linux平台动态分析
# gdb插件gef



```
gef➤ r
Starting program: /home/zyr/GdbTest/test

Breakpoint 1, main (argc=0x1, argv=0x7fffffffdef8) at test01.c:22
22              printf("result[1-100]=%ld\n",result);

$rax      0x0000000000000063 $rbx      0x0000000000000000 $rcx      0x
$rdx      0x00007fffffffdf08 $rsp      0x00007fffffffddf0 $rbp      0x
$rsi      0x00007fffffffdef8 $rdi      0x0000000000000001 $rip      0x
$r8       0x00007ffff7dd4e80 $r9       0x00007ffff7dea530 $r10      0x
$r11      0x00007ffff7a36e50 $r12      0x0000000000400440 $r13      0x
$r14      0x0000000000000000 $r15      0x0000000000000000 $cs       0x
$ss       0x000000000000002b $ds       0x0000000000000000 $es       0x
$fs       0x0000000000000000 $gs       0x0000000000000000 $eflags   51
Flags: [carry parity adjust zero sign trap INTERRUPT direction over

0x00007fffffffddf0|+0x00: 0x00007fffffffdef8  → 0x00007fffffffe297
rsp
0x00007fffffffddf8|+0x08: 0x0100400440
0x00007fffffffde00|+0x10: 0x64ffffdef0
0x00007fffffffde08|+0x18: 0x1356
0x00007fffffffde10|+0x20: 0x00    ←$rbp
0x00007fffffffde18|+0x28: 0x00007ffff7a36f45  → 0x7ffff7a36f45  <_
0x00007fffffffde20|+0x30: 0x00
0x00007fffffffde28|+0x38: 0x00007fffffffdef8  → 0x00007fffffffe297
```

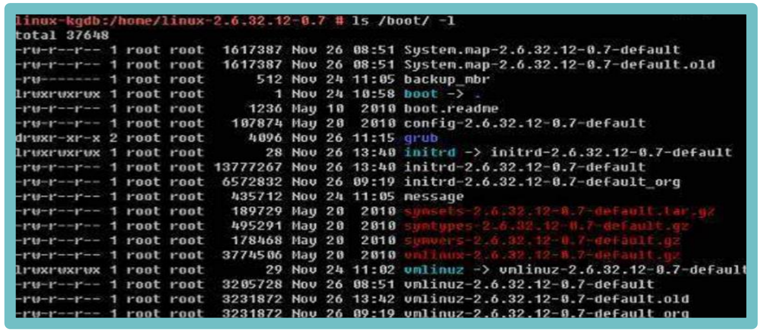
# Window平台内核调试—WinDbg


Local kernel - WinDbg:6.12.0002.633 X86
File Edit View Debug Window Help
Command

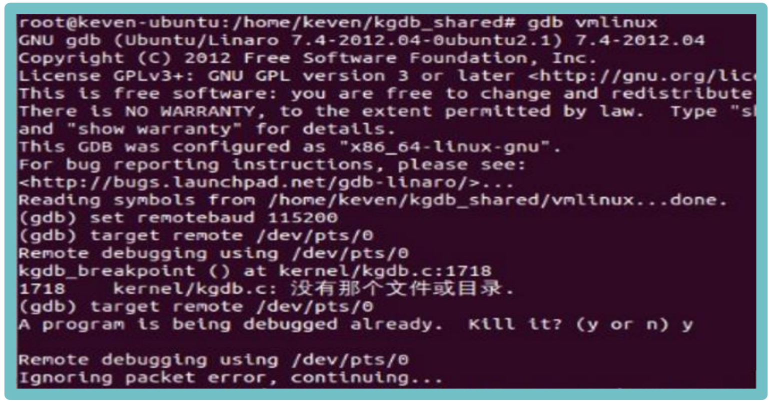
```
Microsoft (R) Windows Debugger Version 6.12.0002.633 X86
Copyright (c) Microsoft Corporation. All rights reserved.

Connected to Windows XP 2600 x86 compatible target at (Mon Nov 28 16:02:12.07
Symbol search path is: *** Invalid ***
****************************************************************************
* Symbol loading may be unreliable without a symbol search path.           *
* Use .symfix to have the debugger choose a symbol path.                   *
* After setting your symbol path, use .reload to refresh symbol locations. *
****************************************************************************
Executable search path is:
*********************************************************************
* Symbols can not be loaded because symbol path is not initialized. *
*                                                                   *
* The Symbol Path can be set by:                                    *
*   using the _NT_SYMBOL_PATH environment variable.                 *
*   using the -y <symbol_path> argument when starting the debugger. *
*   using .sympath and .sympath+                                    *
*********************************************************************
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for
*****************************************************************************
WARNING: Local kernel debugging requires booting with kernel
debugging support (/debug or bcdedit -debug on) to work optimally.
*****************************************************************************
Windows XP Kernel Version 2600 (Service Pack 3) UP Free x86 compatible
Product: WinNt, suite: TerminalServer SingleUserTS
Built by: 2600.xpsp.080413-2111
Machine Name:
Kernel base = 0x804d8000 PsLoadedModuleList = 0x80554fc0
Debug session time: Mon Nov 28 16:02:12.218 2016 (UTC + 8:00)
System Uptime: 0 days 0:29:04.250
lkd> lm
start    end        module name
804d8000 806d0480   nt           (export symbols)       ntkrnlpa.exe

Unloaded modules:
b22ec000 b2317000   kmixer.sys
b22ec000 b2317000   kmixer.sys
b24d7000 b2502000   kmixer.sys
f8d0d000 f8d0e000   drmkaud.sys
b2763000 b2770000   DMusic.sys
b2773000 b2781000   swmidi.sys
b25ca000 b25ed000   aec.sys
f8c22000 f8c24000   splitter.sys
b2951000 b2965000   Parport.SYS
f8982000 f8987000   Cdaudio.SYS
f838f000 f8392000   Sfloppy.SYS
lkd> .reload
Connected to Windows XP 2600 x86 compatible target at (Mon Nov 28 16:03:21.9
Loading Kernel Symbols
...............................................................
.......................................................

lkd>
```

# Linux平台内核调试
# — kgdb

- 需要重新编译内核打开kgdb选项



```
linux-kgdb:/home/linux-2.6.32.12-0.7 # ls /boot/ -l
total 37648
-rw-r--r-- 1 root root  1617387 Nov 26 08:51 System.map-2.6.32.12-0.7-default
-rw-r--r-- 1 root root  1617387 Nov 26 08:51 System.map-2.6.32.12-0.7-default.old
-rw------- 1 root root      512 Nov 24 11:05 backup_mbr
lrwxrwxrwx 1 root root        1 Nov 24 10:58 boot -> .
-rw-r--r-- 1 root root     1236 May 10  2010 boot.readme
-rw-r--r-- 1 root root   107874 May 20  2010 config-2.6.32.12-0.7-default
drwxr-xr-x 2 root root     4096 Nov 26 11:15 grub
lrwxrwxrwx 1 root root       28 Nov 26 13:40 initrd -> initrd-2.6.32.12-0.7-default
-rw-r--r-- 1 root root 13777267 Nov 26 13:40 initrd-2.6.32.12-0.7-default
-rw-r--r-- 1 root root  6572832 Nov 26 09:19 initrd-2.6.32.12-0.7-default_org
-rw-r--r-- 1 root root   435712 Nov 24 11:05 message
-rw-r--r-- 1 root root   189729 May 20  2010 symsets-2.6.32.12-0.7-default.tar.gz
-rw-r--r-- 1 root root   495291 May 20  2010 symtypes-2.6.32.12-0.7-default.gz
-rw-r--r-- 1 root root   178468 May 20  2010 symvers-2.6.32.12-0.7-default.gz
-rw-r--r-- 1 root root  3774506 May 20  2010 vmlinux-2.6.32.12-0.7-default.gz
lrwxrwxrwx 1 root root       29 Nov 24 11:02 vmlinuz -> vmlinuz-2.6.32.12-0.7-default
-rw-r--r-- 1 root root  3205728 Nov 26 08:51 vmlinuz-2.6.32.12-0.7-default
-rw-r--r-- 1 root root  3231872 Nov 26 13:42 vmlinuz-2.6.32.12-0.7-default.old
-rw-r--r-- 1 root root  3231872 Nov 26 09:19 vmlinuz-2.6.32.12-0.7-default_org
```



```
root@keven-ubuntu:/home/keven/kgdb_shared# gdb vmlinux
GNU gdb (Ubuntu/Linaro 7.4-2012.04-0ubuntu2.1) 7.4-2012.04
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/lic
This is free software: you are free to change and redistribute
There is NO WARRANTY, to the extent permitted by law.  Type "s
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
For bug reporting instructions, please see:
<http://bugs.launchpad.net/gdb-linaro/>...
Reading symbols from /home/keven/kgdb_shared/vmlinux...done.
(gdb) set remotebaud 115200
(gdb) target remote /dev/pts/0
Remote debugging using /dev/pts/0
kgdb_breakpoint () at kernel/kgdb.c:1718
1718    kernel/kgdb.c: 没有那个文件或目录.
(gdb) target remote /dev/pts/0
A program is being debugged already.  Kill it? (y or n) y

Remote debugging using /dev/pts/0
Ignoring packet error, continuing...
```

## 逆向平台搭建（1）

静态分析

- Winhex, UltraEdit
- PEID, LoadPE, DIE
- IDA and plugins

动态分析

- R3: Ollydbg, gdb
- R0: windbg, kgdb, ..
- VM: vmware, Qemu..

## 逆向平台搭建（2）

非主流工具

- 符号执行：Angr；Z3

- 污点跟踪：Pin；Valgrind；TraintDroid

- 模糊测试：FileFuzz；AFL；Trinity；Peach

# 逆向平台搭建（3）

平台专用工具

- Android专用：apktool；SMALI/BAKSMALI；Dex2JAR；JD-GUI；

- 固件专用：Binwalk；firmware-mod-kit；Qemu；

感谢大家！

南京郵電大學
Nanjing University of Posts and Telecommunications