



# 恶意软件分析

## 第3章：软件分析基础技术



# 主要内容

- 3.1 静态分析基础技术
- 3.2 在虚拟机中分析恶意代码
- 3.3 动态分析基础技术



## 3.1 静态分析基础技术

# 技术



- 反病毒引擎**扫描**，确认程序样本的恶意性
- 哈希值，使用哈希识别恶意代码
- 文件的字符串、函数和文件头，从中发掘有用信息

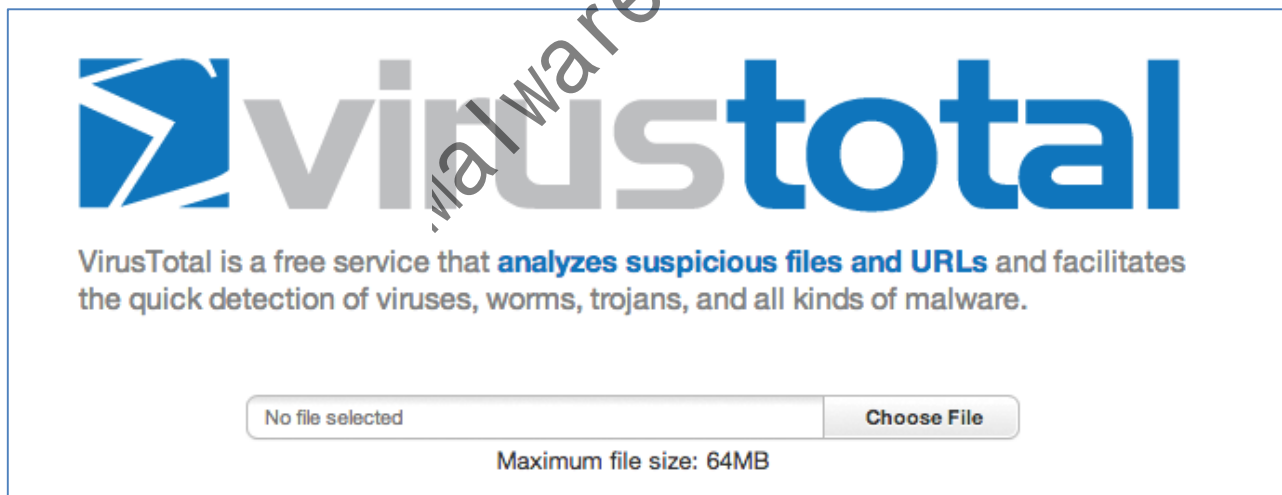


# 反病毒引擎扫描



# 只是第一步

- 恶意代码可以很容易地改变其特征，欺骗反病毒引擎
- 使用VirusTotal很方便，但可能使攻击者意识到他们已经被发现





# 哈希值

恶意代码的指纹



# 哈希值

- MD5 或者 SHA-1
- 将一个任意大小的文件转换成固定长度的指纹
- 实践中很有效，能够唯一地标识一个文件
  - 有MD5碰撞，但并不常见
  - 碰撞：两个不同的文件具有相同哈希值



# 哈希值计算



**H HashCalc**

Data Format: **File** Data: **C:\Users\student\Desktop\p3.pcap** ...

☐ HMAC Key Format: **Text string** Key:

---

☒ MD5 **52583b5e2c99d19c046915181fd7b29b**

☐ MD4

☒ SHA1 **991d4e880832dd6aaebadb8040798a6b9f163194**

☐ SHA256



# 哈希值用途

- 作为恶意代码的标签
- 与其他分析师分享哈希值，帮助他们识别恶意代码
- 在线搜索这段哈希值，看看这个文件是否已经被识别

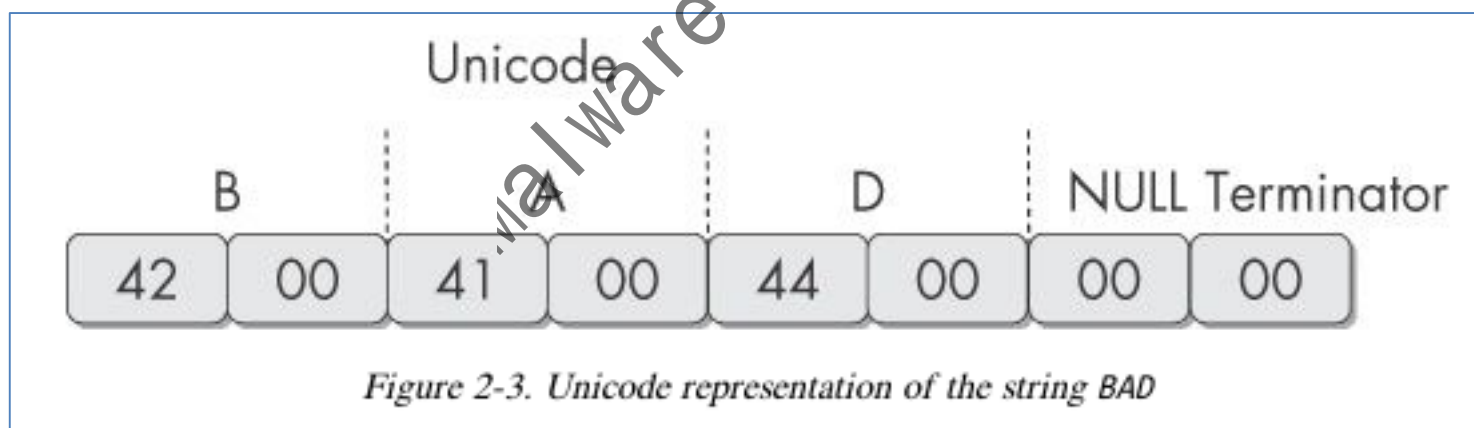
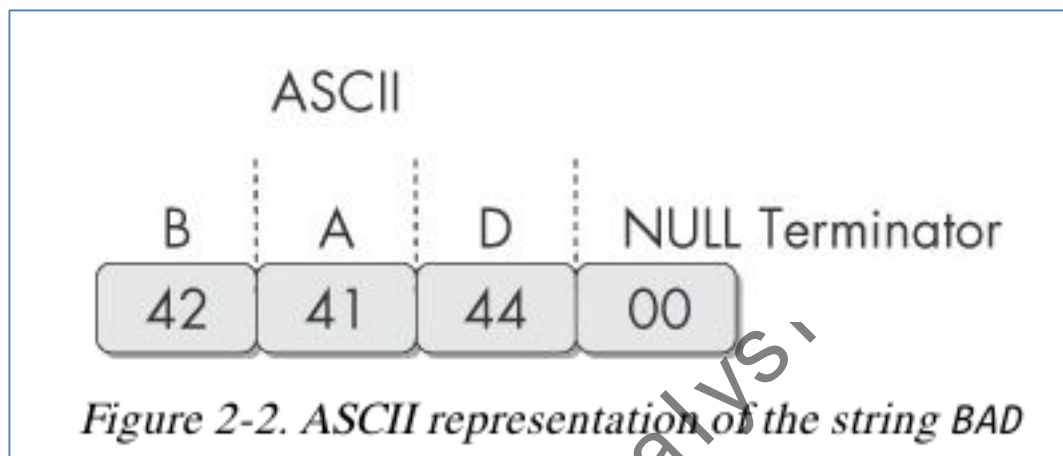


# 查找字符串



# 字符串

- 字符串是任何可打印的字符序列
- 字符串以空字符 (0x00)结尾
- ASCII字符长度为8位
  - ASCII扩展码称为ANSI
- Unicode字符长度为16位
  - 微软称他们“宽字符”





# 命令字符串

- 在Linux或Windows系统平台上
- 找出文件中所有的字符串或更长的字符串长度

Malware Analysis



# 命令字符串

- 加粗的字符串都可以忽略
- **GetLayout**和**SetLayout**是Windows函数
- **GDI32.DLL**
- 是动态链接库

```
C:>strings bp6.ex_  
VP3  
VW3  
t$@  
D$4  
99.124.22.1 4  
e-@  
GetLayout 1  
GDI32.DLL 3  
SetLayout 2  
M}C  
Mail system DLL is invalid.!Send Mail failed to  
send message. 5
```



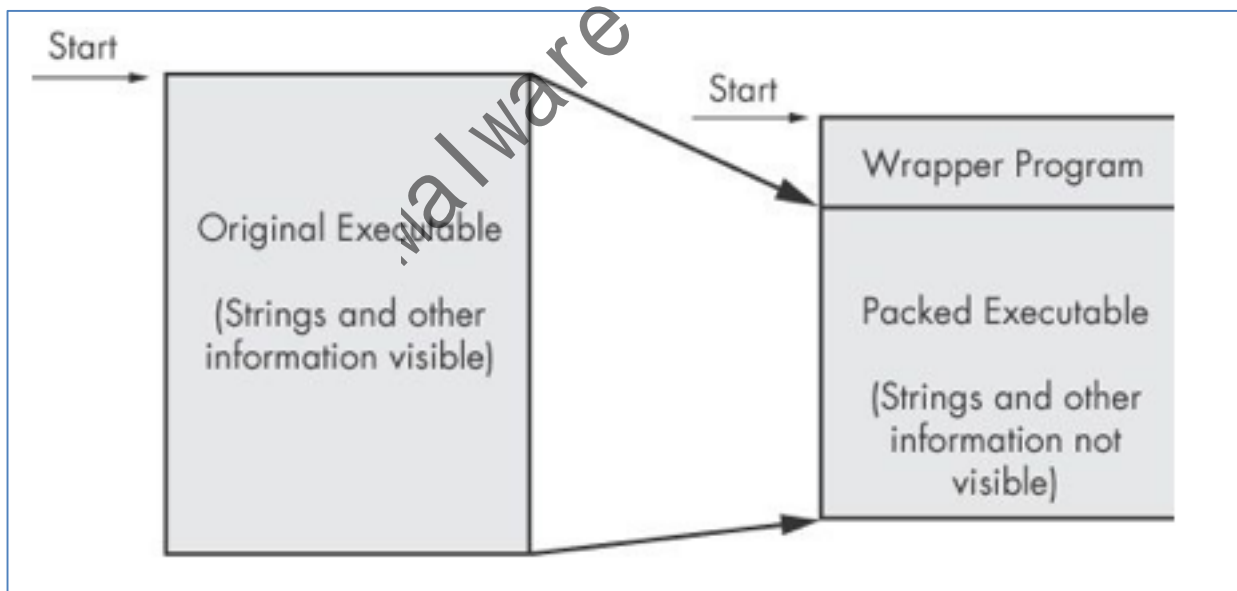
# 加壳与混淆恶意代码





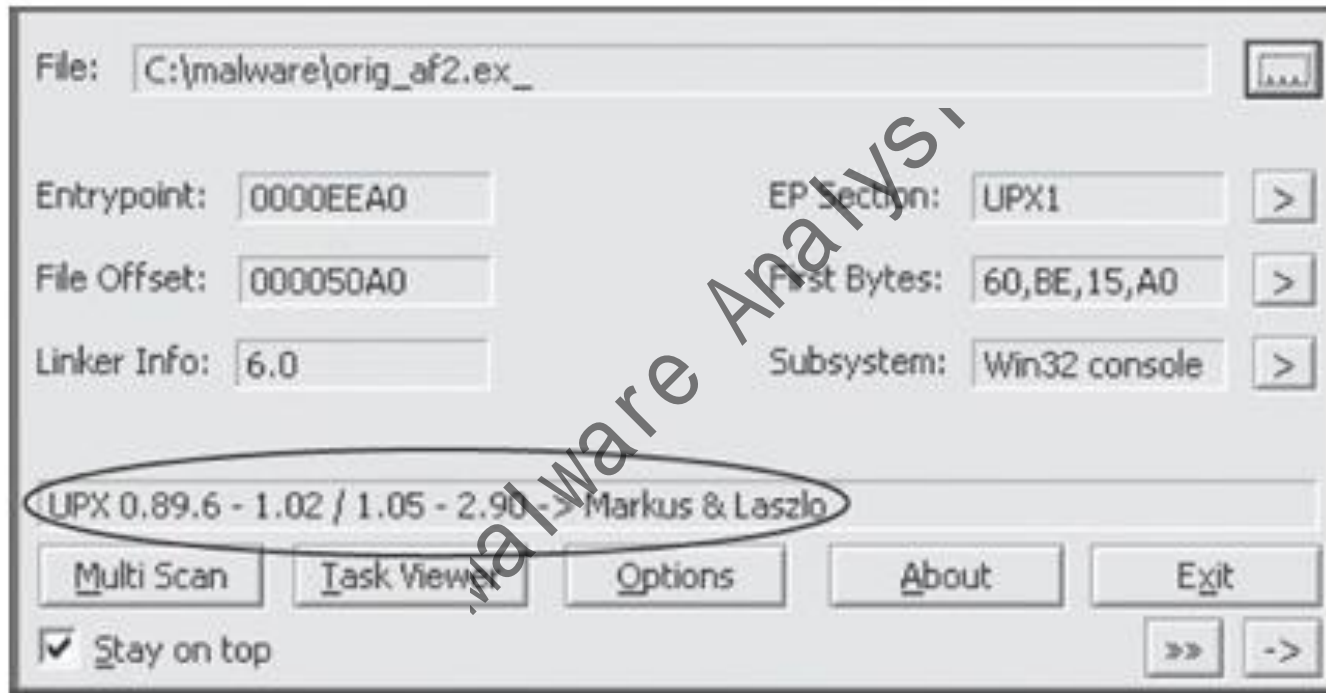
# 文件加壳

- 代码像Zip文件一样被压缩
- 使字符串和指令不可读
- 只有一小段脱壳代码可以被解析





# 使用PEiD检测加壳



*Figure 2-5. The PEiD program*

# Demo : UPX



```
root@kali: ~/126
File Edit View Search Terminal Help
root@kali:~/126# cat chatty.c
#include <stdio.h>
main()
{
char name[10];
printf("This program contains readable strings\n");
printf("Enter your name: ");
scanf("%s", name);
printf("Hello %s\n", name);
}

root@kali:~/126# gcc -static chatty.c -o chatty
root@kali:~/126# upx -o chatty-packed chatty
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2011
UPX 3.08 Markus Oberhumer, Laszlo Molnar & John Reiser Dec 12th 2011

File size      Ratio      Format      Name
-----
592800 -> 272588 45.98% linux/elf386 chatty-packed

Packed 1 file.
root@kali:~/126# ls -l
total 852
-rwxr-xr-x 1 root root 592800 Aug 16 20:34 chatty
-rw-r--r-- 1 root root 174 Aug 16 20:27 chatty.c
-rwxr-xr-x 1 root root 272588 Aug 16 20:34 chatty-packed
root@kali:~/126#
```

Malware Analysis(<http://scs.uacs.ac.cn>)



# 加壳会混淆字符串

```
root@kali:~/126# strings chatty | wc
1962    4498    33817
root@kali:~/126# strings chatty-packed | wc
3950    4290    23623
root@kali:~/126#
```



# 注意

- PEiD的许多插件会在不告知的情况下执行恶意代码！（看第三章学习怎么搭建运行恶意代码的安全环境）
- 像其他软件一样恶意代码分析软件也会存在漏洞。例如0.92版本的PEiD存在溢出漏洞允许攻击者执行任意代码。聪明的恶意代码编写者使用该漏洞，可以写一个程序用于控制分析恶意代码的计算机。
- 确保PEiD是最新版本。



# PE文件格式

EXE 文件



# PE 文件

- Windows可执行文件、对象代码和DLL使用的文件格式
- 一种数据结构，包含Windows系统加载文件所必需的信息。
- 几乎Windows系统中运行的每一个文件都是PE格式的



# PE 头

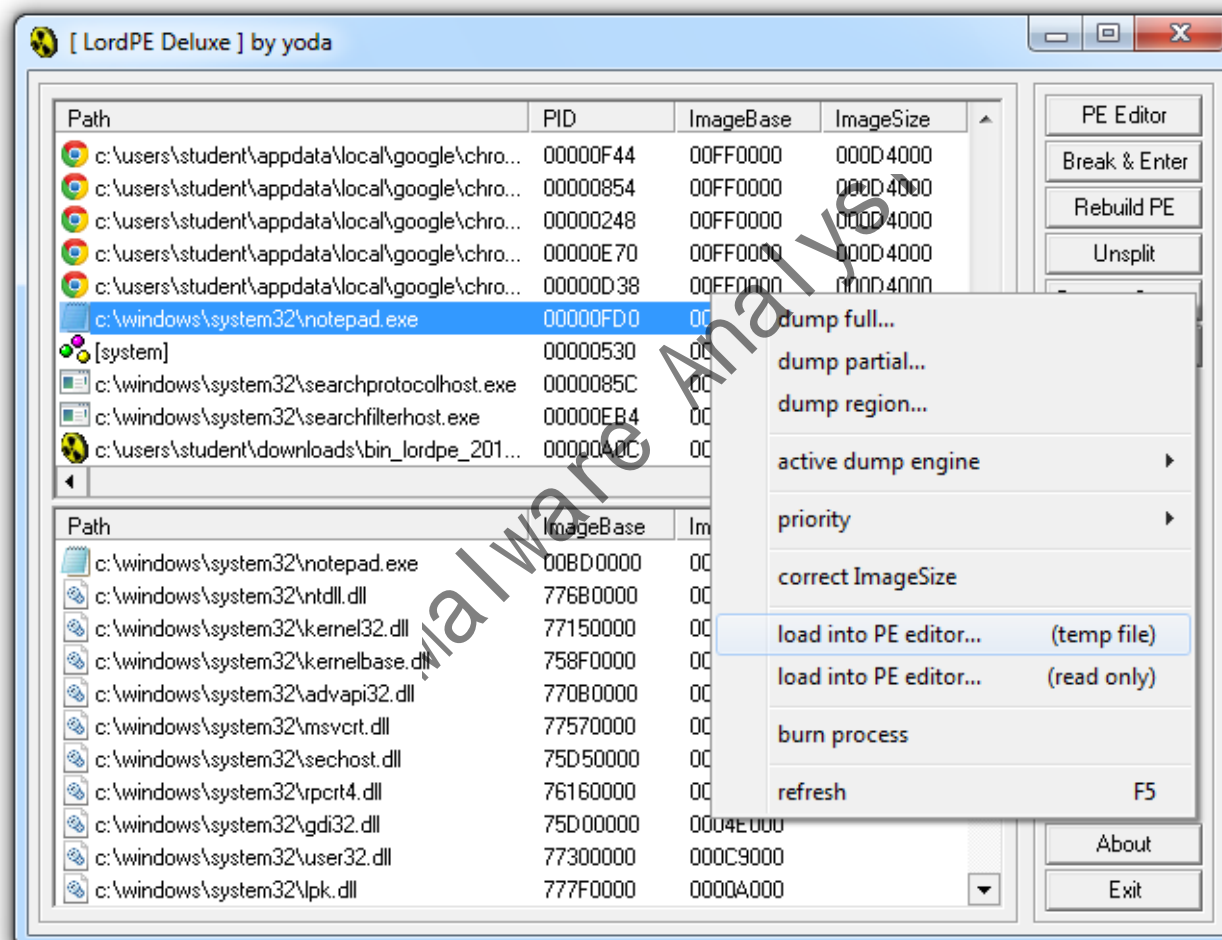
- 代码信息
- 应用程序类型
- 所需的库函数
- 空间需求

Malware Analysis





# LordPE Demo





# 头及节表信息

[ LordPE Deluxe ] by voda

[ PE Editor ] - c:\windows\system32\notepad.exe [READ ONLY]

Basic PE Header Information

EntryPoint:	00003689	Subsystem:	0002
ImageBase:	01000000	NumberOfSections:	0004
SizeOfImage:	00030000	TimeDateStamp:	4A5BC60F
BaseOfCode:	00001000	SizeOfHeaders:	00000400
BaseOfData:	0000C000	Characteristics:	0102
SectionAlignment:	00001000	Checksum:	00039741
FileAlignment:	00000200	SizeOfOptionalHeader:	00E0
Magic:	010B	NumOfRvaAndSizes:	00000010

OK  
Save  
Sections  
Directories  
FLC  
TDSC  
Compare  
L

[ Section Table ]

Name	VOffset	VSize	ROffset	RSize	Flags
.text	00001000	0000A68C	00000400	0000A800	60000020
.data	0000C000	00002164	00004C00	00001000	C0000040
.rsrc	0000F000	0001F160	00008C00	0001F200	40000040
.reloc	0002F000	00000E34	0002AE00	00001000	42000040

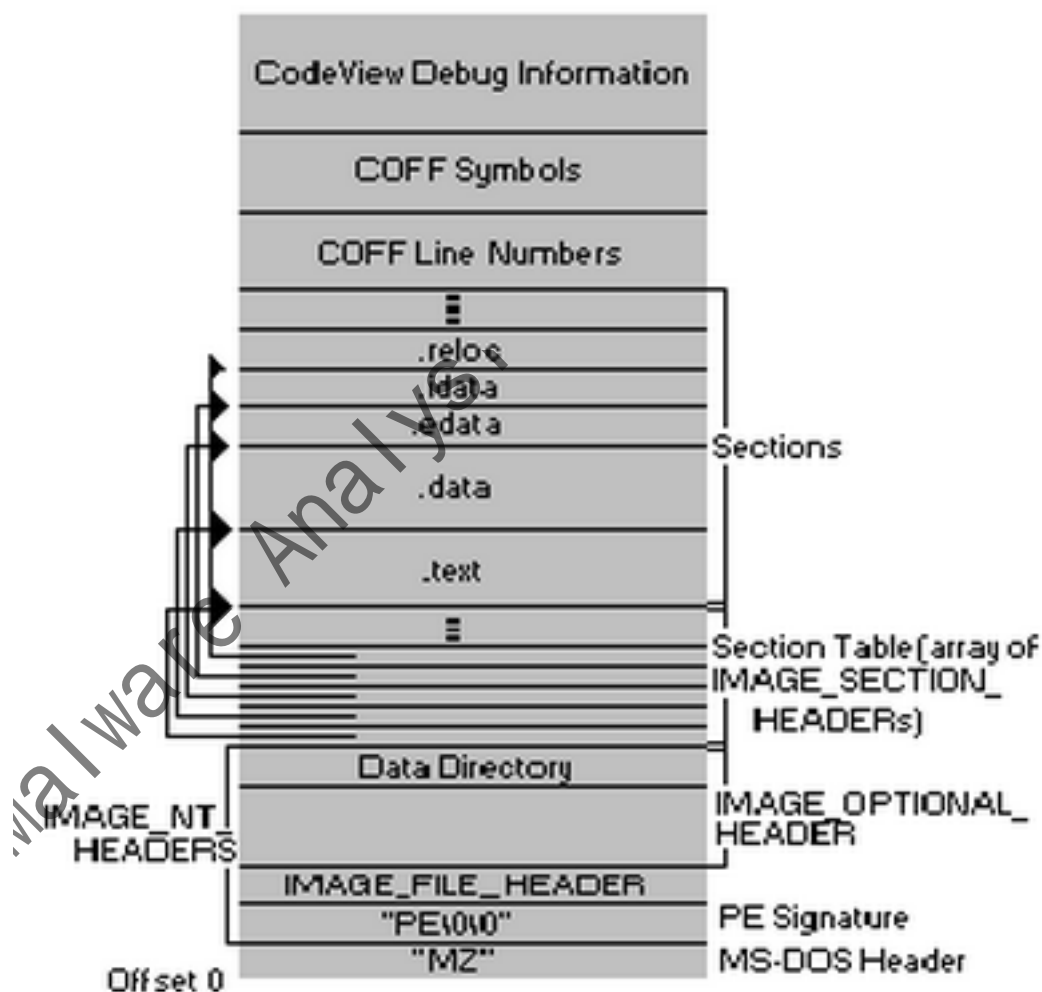
About  
Exit

c:\windows\system32\lpk.dll 777F0000 0000A000

Malware Analysis(<http://scs.ucas.ac.cn>)

# 这个PE文件 有很多节

- 目前关注主要的节信息就足够了



**Figure 1. The PE file format**



# 链接库与函数



# 导入表

- 程序使用的函数存在另一个程序中，如函数库
- 通过链接连接到主程序中
- 有三种链接方式
  - 静态链接
  - 运行时链接
  - 动态链接



# 静态链接

- Windows中很少用
- Unix 和Linux中比较常见
- 库中的所有代码都会被复制到可执行程序中
- 使可执行程序增大



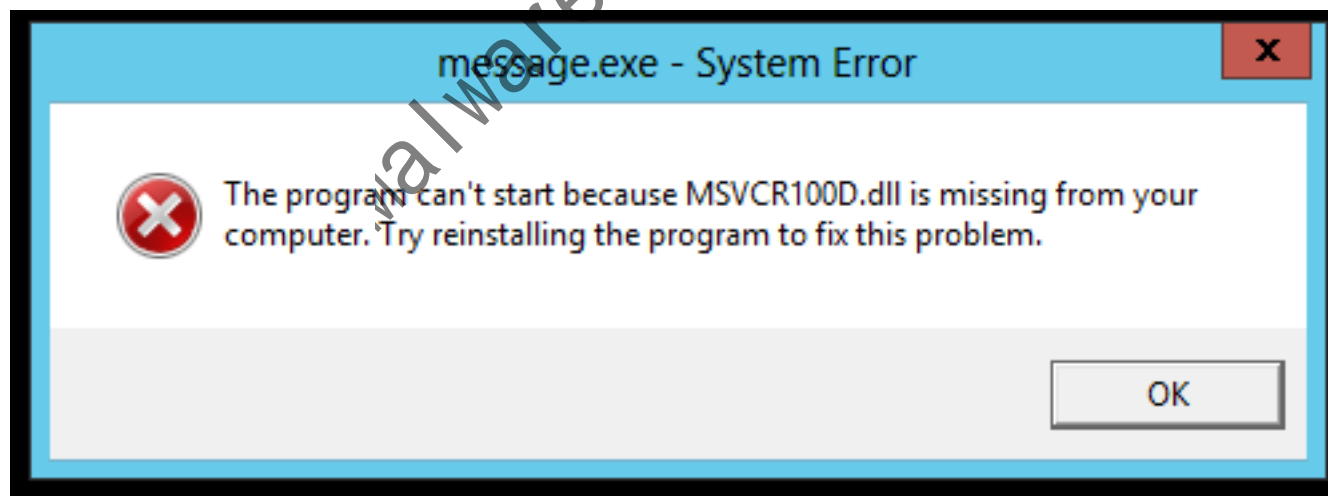
# 运行时链接

- 在合法**程序**中并不流行
- 在恶意代码中是常用的，特别是恶意代码被加壳或混淆的时候
- 只在**需要时**加载库，而不是在程序开始时就加载
- 最常见的由LoadLibrary 和 GetProcAddress 函数实现



# 动态链接

- 最常见的方式
- 在程序加载时，主机操作系统搜索必要的库







# 函数库中的线索

- **PE**文件头列出了每个将被加载的库和函数
- 它们的名字能够揭示程序会做什么
- **URLDownloadToFile** 表明程序会下载东西



# Dependency Walker

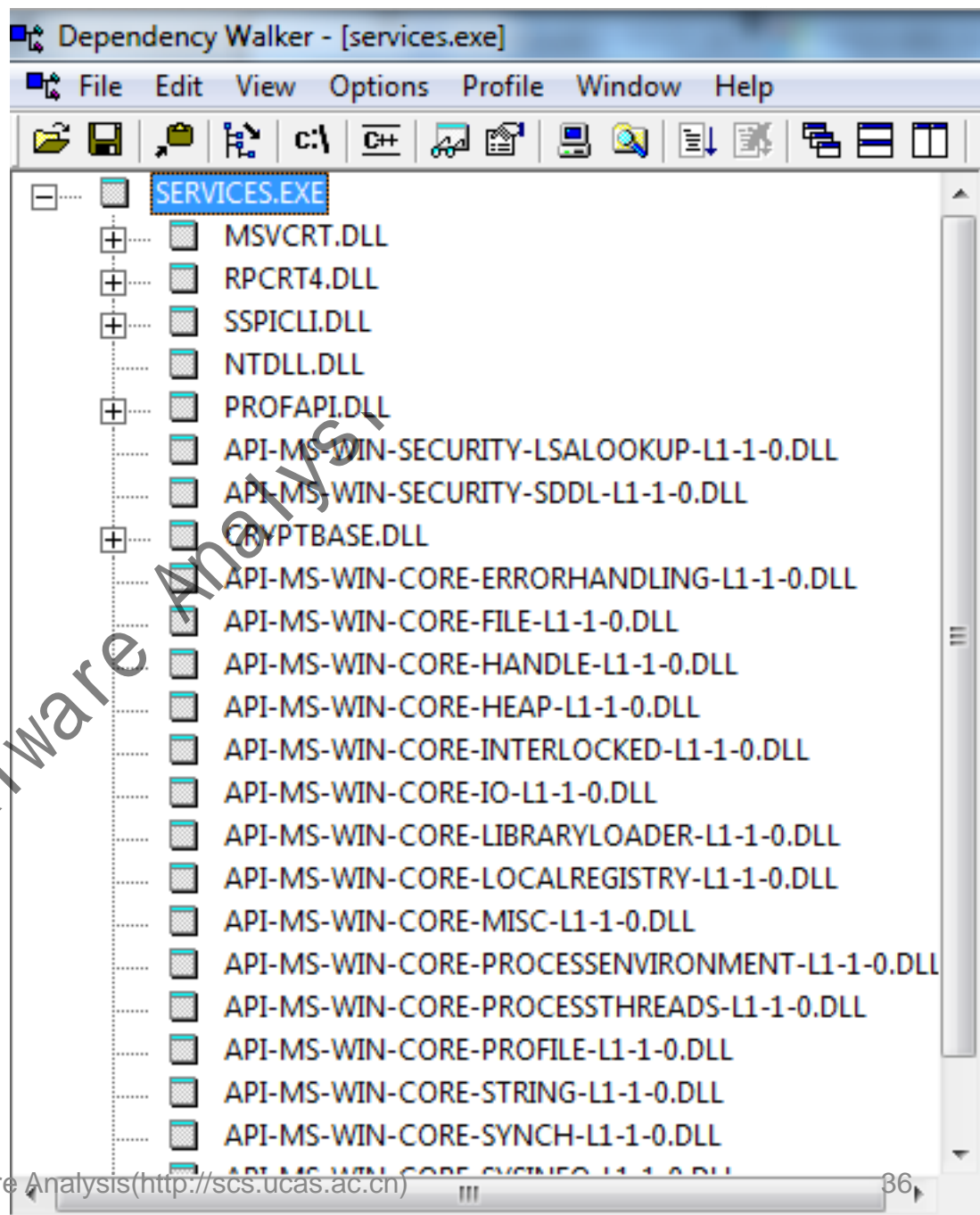


# 显示动态链接函数

- 正常的程序有很多的DLL
- 恶意代码常常只有很少的DLL

Malware Analysis

# Services.exe



# Services.ex\_ (malware)

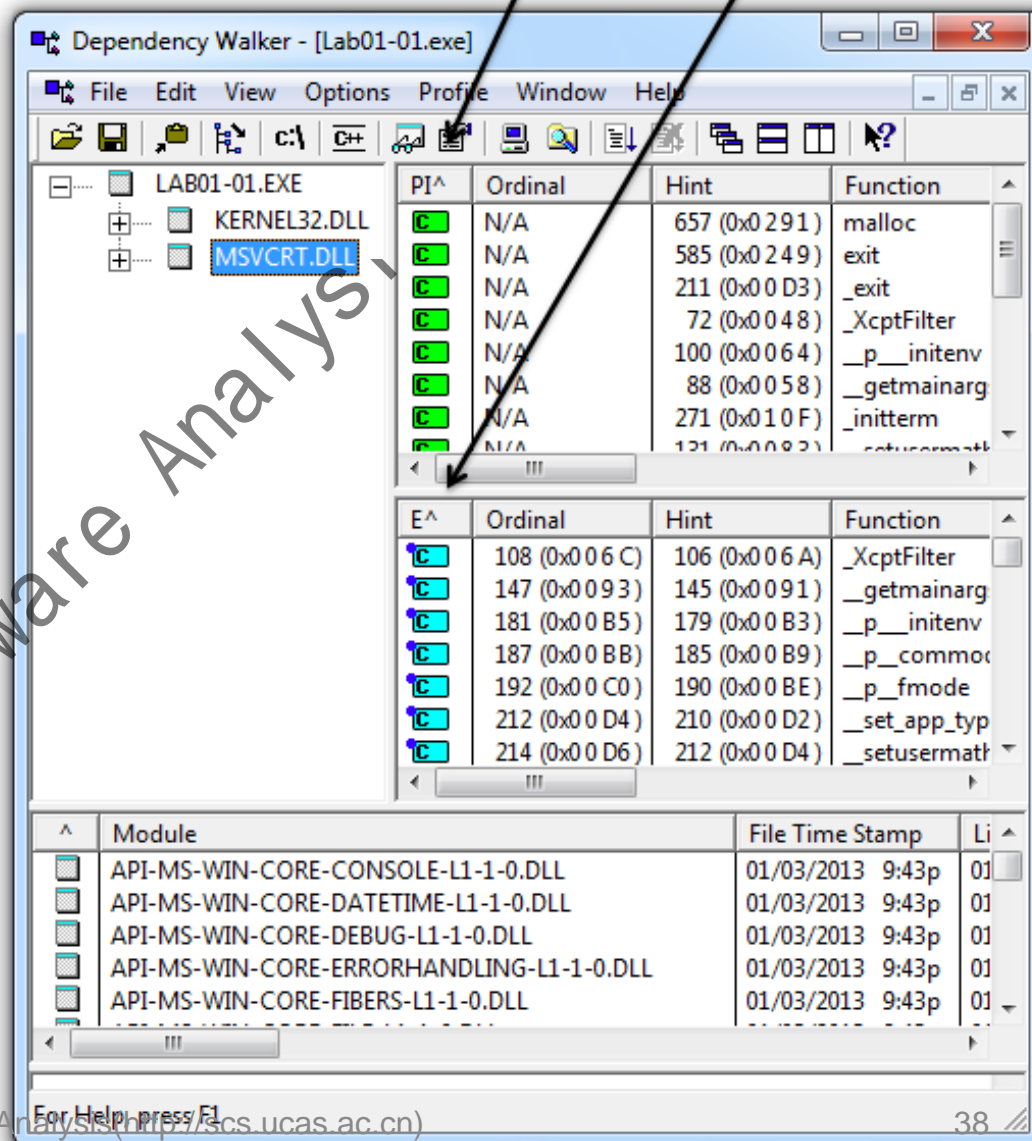


# Dependency

中显示动态  
链接库文件  
的导入导出  
表信息

Imports (PI)

Exports (E)





# 常见的DLL

- Kernel32.dll 这是一个很常见的DLL，它包含核心系统功能，如访问和操作内存、文件和硬件等
- Advapi32.dll 这个DLL提供了对核心Windows组件的访问，比如服务管理器和注册表
- User32.dll 这个DLL中包含了所有用户界面组件，如按钮、滚动条以及控制和响应用户操作的组件
- Gdi32.dll 这个DLL中包含了图形显示和操作的函数



# 常见的DLL

- Ntdll.dll 这个DLL是Windows内核的接口。可执行文件通常不直接导入这个文件，而是由Kernal32.dll间接导入，如果一个可执行文件导入了这个文件，这意味着作者企图使用那些不是正常提供给Windows程序使用的函数。一些如隐藏和操作进程等任务会使用这个接口
- Wsock32.dll和Ws2\_32.dll 这两个是联网DLL，访问其中任一个DLL的程序非常可能连接网络，或执行网络相关任务
- Winginet.dll 这个DLL包含了更高层次的网络函数，实现了如FTP、HTTP和NTP等协议





# 导出表

- DLL导出函数
- EXE导入函数
- 导出表和导入表都列在PE头中
- 这本书说导出表很少出现在EXE中，但我看到很多无辜的EXE有导出表



# 实例：键盘记录器

- 导入User32.dll并使用 SetWindowsHookEx 函数，是键盘记录器获取键盘输入最流行的方法
- 导出 LowLevelKeyboardProc 和 LowLevelMouseProc 函数向别处发送数据
- 使用RegisterHotKey函数来定义一个特殊的按键如Ctrl + Shift + P收获收集到的数据



# 实例：一个加了壳的程序

- 很少的函数
- 所看到的只有脱壳程序

*Table 2-3. DLLs and Functions Imported from PackedProgram.exe*

Kernel32.dll	User32.dll
GetModuleHandleA	MessageBoxA
LoadLibraryA	
GetProcAddress	
ExitProcess	
VirtualAlloc	
VirtualFree	



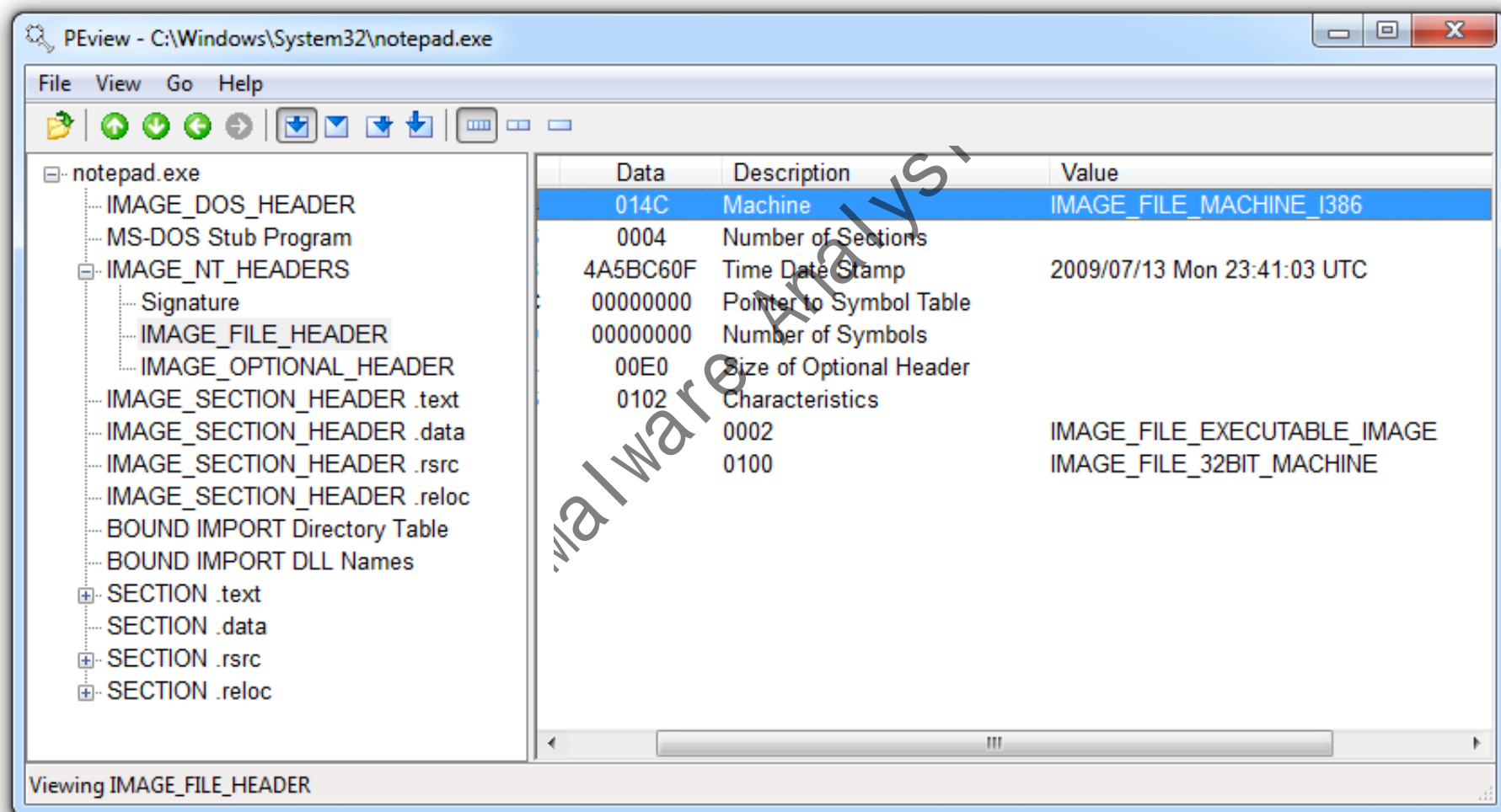
# PE 文件的头与分节



# PE文件主要的节

- .text – CPU执行指令
- .rdata – 导入表和导出表
- .data – 全局数据
- .rsrc – 字符串、图标、图片、菜单等

# PEView





# 日期时间戳

- 显示这个可执行文件是何时编译的
- 老的程序更有可能被杀毒程序识别
- 但有时日期是错误的
  - 所有的Delphi程序显示1992年6月19日
  - 日期也可以被伪造

# IMAGE\_SECTION\_HEADER



- 虚拟大小 – 内存
- 原始数据大小 – 硬盘
- 对于 **.text** 节，通常虚拟大小与原始数据大小相等或接近相等
- 对于 **.text** 节，加壳的可执行文件虚拟大小比原始大小大的多



# 未加壳程序



PEview - C:\Windows\System32\notepad.exe

File View Go Help

notepad.exe

- IMAGE\_DOS\_HEADER
- MS-DOS Stub Program
- IMAGE\_NT\_HEADERS
  - IMAGE\_SECTION\_HEADER .text
  - IMAGE\_SECTION\_HEADER .data
  - IMAGE\_SECTION\_HEADER .rsrc
  - IMAGE\_SECTION\_HEADER .reloc
- BOUND\_IMPORT Directory Table
- BOUND\_IMPORT DLL Names
- SECTION .text
- SECTION .data
- SECTION .rsrc
- SECTION .reloc

pFile	Data	Description
000001D8	2E 74 65 78	Name
000001DC	74 00 00 00	
000001E0	0000A68C	Virtual Size
000001E4	00001000	RVA
000001E8	0000A800	Size of Raw Data
000001EC	00000400	Pointer to Raw Data
000001F0	00000000	Pointer to Relocations
000001F4	00000000	Pointer to Line Numbers
000001F8	0000	Number of Relocations
000001FA	0000	Number of Line Numbers
000001FC	60000020	Characteristics
	00000020	
	20000000	
	40000000	

Viewing IMAGE\_SECTION\_HEADER .text

Malware Analysis(<http://scs.uca.ac.cn>)



*Table 2-6. Section Information for PackedProgram.exe*

Name	Virtual size	Size of raw data
.text	A000	0000
.data	3000	0000
.rdata	4000	0000
.rsrc	19000	3400
Dijfpds	20000	0000
.sdfuok	34000	3313F
Kijijl	1000	0200

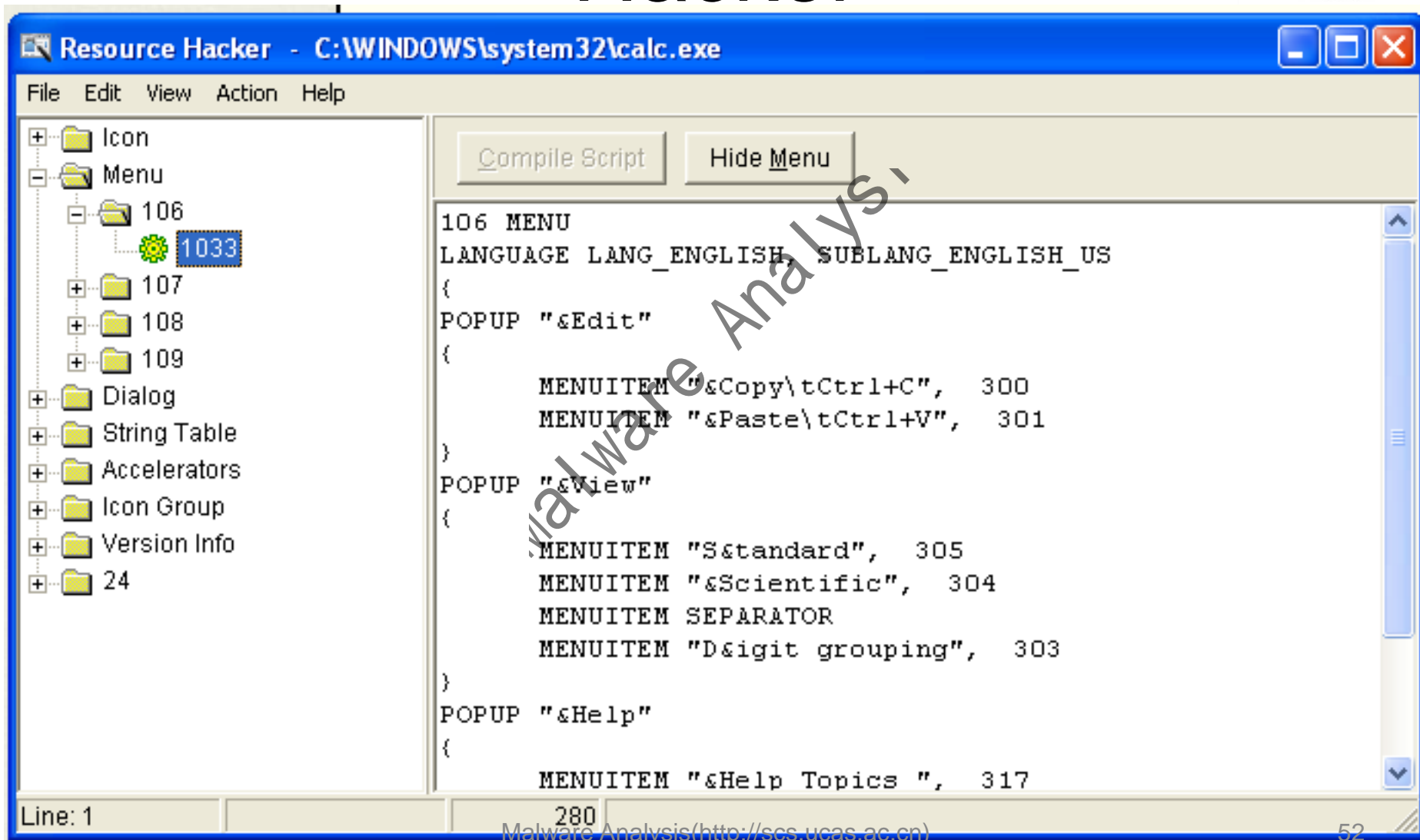


# Resource Hacker

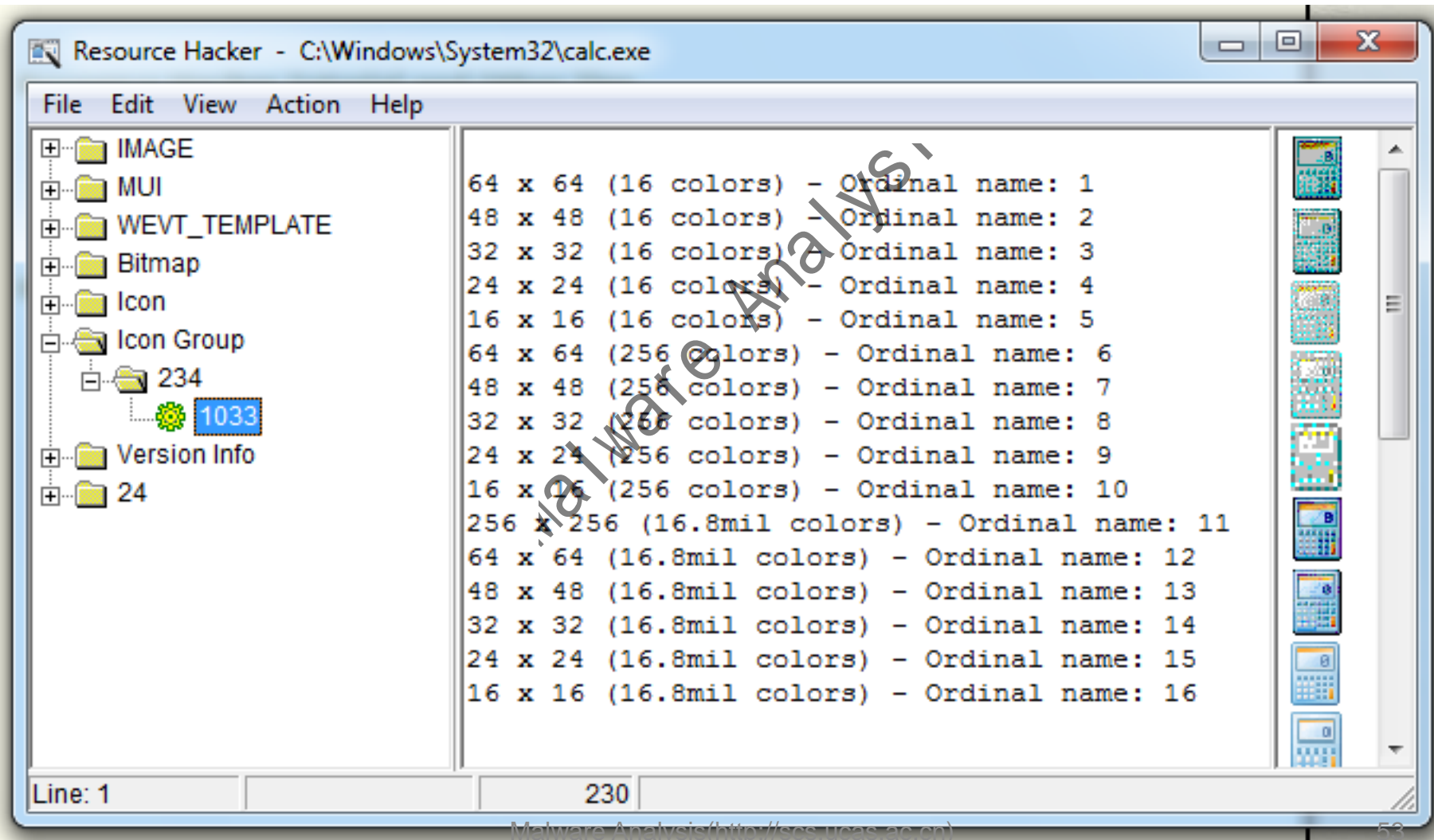
- 可以浏览 .rsrc节
- 字符串、图标和菜单

Malware Analysis

# Windows XP 中的Resource Hacker



# Windows 7中的Resource Hacker





## 3.2 在虚拟机中分析恶意代码



# 动态分析

- 运行并监测恶意代码
- 需要一个安全环境
- 必须阻止恶意代码蔓延到网络中的其他主机上
- 可以使用物理主机及隔离网络—与互联网或其它任何网都应断开。



# 物理主机

- 缺点
  - 不连接互联网，恶意代码中的部分功能可能运行。
  - 恶意代码很难被移除，需要恢复主机镜像
- 优点
  - 一些恶意代码检测到虚拟机便不**正常**执行了





# 虚拟机

- 最常见的方法
- 我们将这样做
- 保护宿主机免受恶意代码的侵害
  - 在一些非常罕见的情况下，恶意代码能逃脱虚拟机并感染宿主机



# VMware Player

- 免费但有限制
- 不能创建快照
- VMware Workstation 或者 Fusion 会更好些但需要花钱买
- 也可以用VirtualBox、Hyper-V、Parallels 或者Xen。



# Windows XP

- 我们分析的恶意代码大多是针对XP的
- 在课堂上分发的DVD中包含一个XP SP3虚拟机



# 虚拟机的结构

- 可以通过移除虚拟网络适配器的方式来禁用网络
- 主机模式网络允许连接宿主机而不是互联网

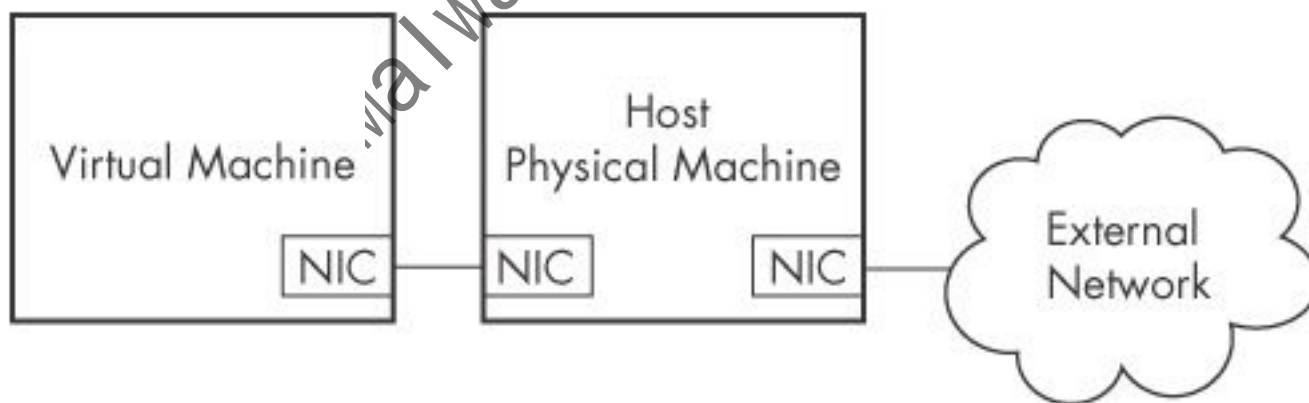


Figure 3-3. Host-only networking in VMware

Malware Analysis(<http://scs.ucas.ac.cn>)

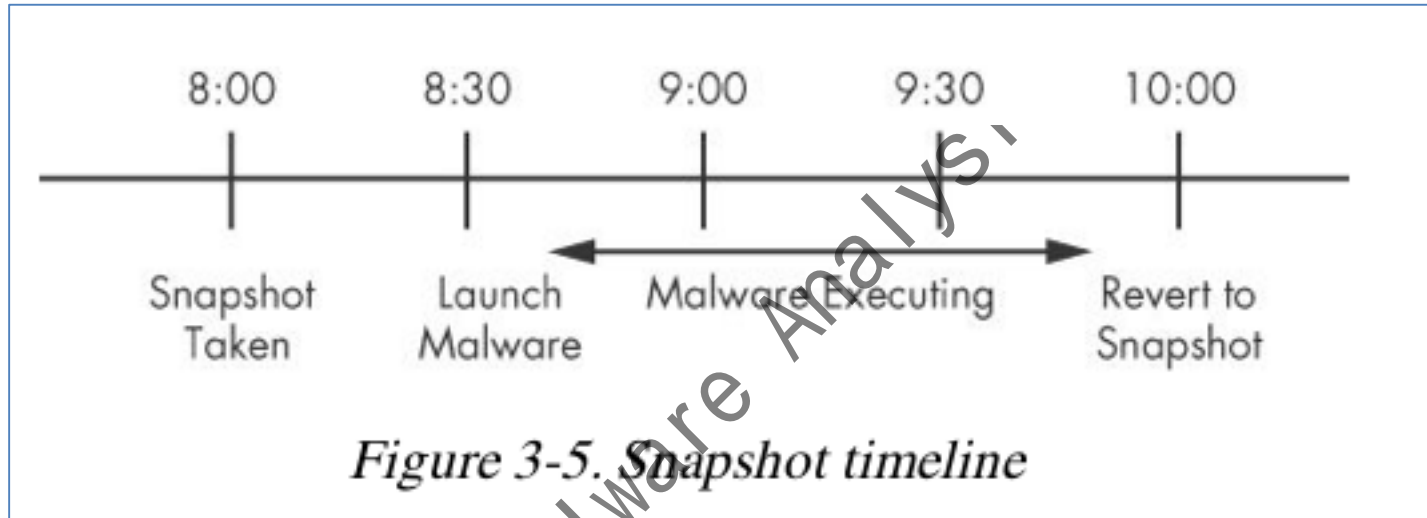


# 让恶意代码连接互联网

- 地址转换（**NAT**）模式让虚拟机之间可以访问并且能够连接互联网，但是在虚拟机和互联网之间放置了虚拟路由器
- **桥接**模式允许**虚拟机**与物理主机一样连接到相同的物理网卡上
- 是否允许恶意代码造成一些伤害或蔓延 — 存在争议
- 可以发送垃圾邮件或参与DDoS攻击



# Snapshots



# 使用VMware 进行恶意代码分析的风险



- 恶意代码会在虚拟环境中有着不同的执行过程
- **VMware** 也有漏洞：利用漏洞恶意代码会导致宿主机系统崩溃，甚至运行在宿主机中
- 恶意代码会传播或影响宿主机— 不要使用易受伤害的宿主机
- 所有书中的样本都是无害的



## 3.3 动态分析基础技术





# 为什么执行动态分析？

- 无法进行静态分析，由于
  - 混淆
  - 加壳
  - 分析者已经穷尽了可用的静态分析技术
- 动态分析是一种有效的方法并能让你看到恶意代码的真实功能



# 沙箱：简便但粗糙的方法



# 沙箱

- 用于基本动态分析的一体化软件
- 一个模拟网络服务的虚拟环境
- 如：Norman Sandbox，GFI Sandbox，Anubis，Joe Sandbox，ThreatExpert，BitBlaze，Comodo Instant Malware Analysis
- 昂贵，但易于使用
- 生成一个漂亮的PDF格式分析报告



# 运行恶意代码



# 加载DLL文件

- EXE 文件能直接执行，DLLs 文件却不能
- 使用 Rundll32.exe (Windows 自带)  
rundll32.exe *DLLname, Export arguments*
- *Export* 值是一个导出函数，可以使用  
Dependency Walker、Pevview 或者 PE Explorer  
找出 DLL 文件的导出函数



# 加载 DLL文件

- 例如：
  - rip.dll 有 **Install** 和 **Uninstall**两个导出函数  
rundll32.exe rip.dll , Install
- 一些函数用序号代替函数名  
rundll32.exe xyzzy.dll , #5
- 也可以修改 PE 文件头将一个DLL文件转变成一个EXE文件



# 进程监视器



# 进程监视器

- 监控注册表、文件系统、网络、进程和线程行为
- 所有记录的事件均被保存，经过滤显示更容易找到感兴趣的东西
- 不要运行太久，会填满所有内存并使机器崩溃



# 加载 Calc.exe



Process Monitor - Sysinternals: www.sysinternals.com

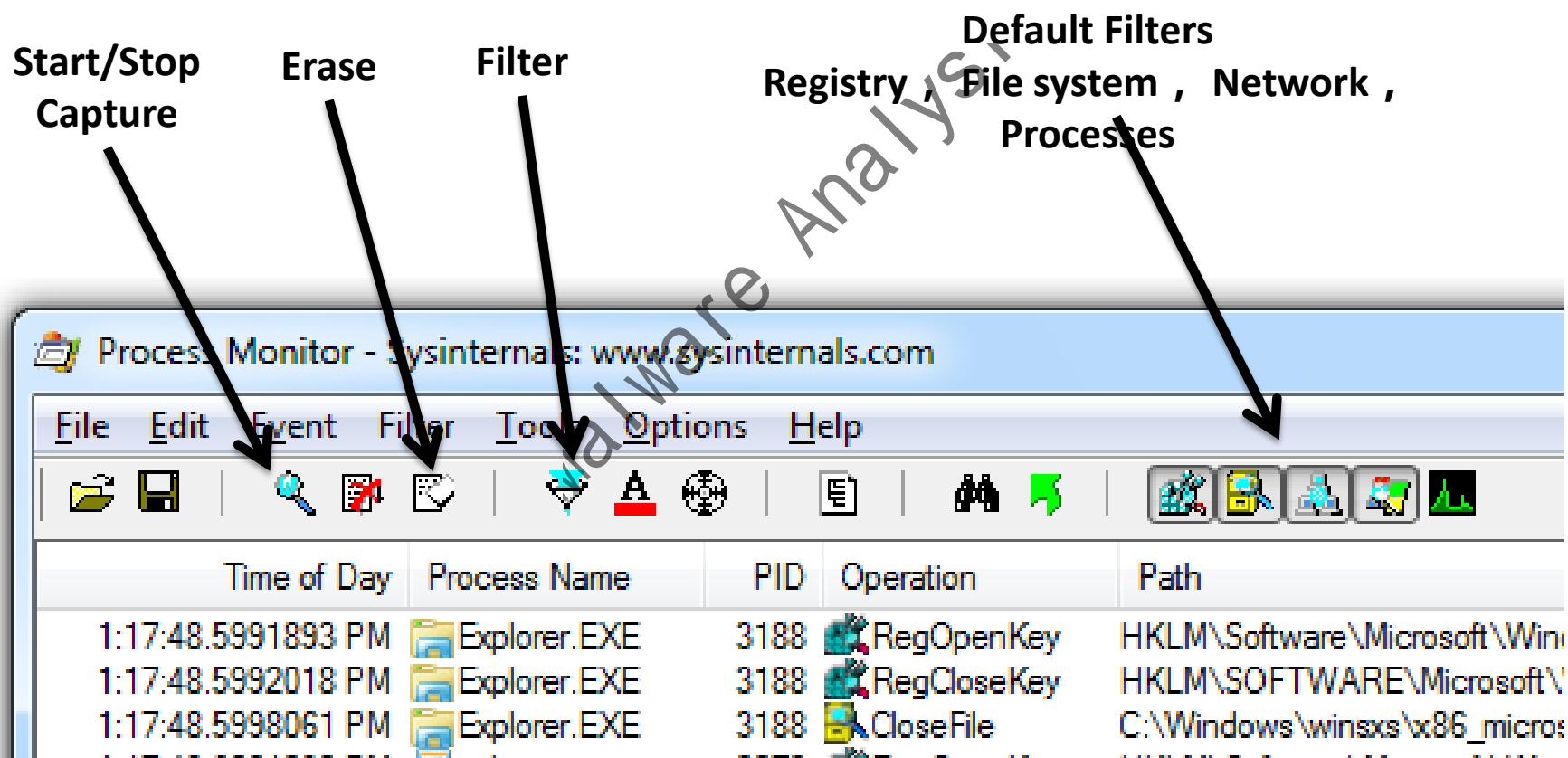
File Edit Event Filter Tools Options Help

Time of Day	Process Name	PID	Operation	Path	Result	Detail
1:17:48.5991893 PM	Explorer.EXE	3188	RegOpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\...	SUCCESS	Desired Access: Query Value
1:17:48.5992018 PM	Explorer.EXE	3188	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersi...	SUCCESS	
1:17:48.5998061 PM	Explorer.EXE	3188	CloseFile	C:\Windows\winsxs\x86_microsoft.windows.common-...	SUCCESS	
1:17:48.6001092 PM	calc.exe	2072	RegOpenKey	HKLM\Software\Microsoft\Windows\Windows Error ...	SUCCESS	Desired Access: Query Value
1:17:48.6001273 PM	calc.exe	2072	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows\Windows Err...	SUCCESS	Type: REG_DWORD, Length: 4, ...
1:17:48.6001350 PM	calc.exe	2072	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows\Windows Err...	SUCCESS	
1:17:48.6001722 PM	calc.exe	2072	ReadFile	C:\Windows\System32\calc.exe	SUCCESS	Offset: 103,424, Length: 32,768, l...
1:17:48.6011060 PM	calc.exe	2072	CreateFile	C:\Windows\System32\WindowsCodecs.dll	SUCCESS	Desired Access: Read Attributes, ...
1:17:48.6011278 PM	calc.exe	2072	QueryBasicInfor...	C:\Windows\System32\WindowsCodecs.dll	SUCCESS	CreationTime: 7/13/2009 4:29:14 ...
1:17:48.6011337 PM	calc.exe	2072	CloseFile	C:\Windows\System32\WindowsCodecs.dll	SUCCESS	
1:17:48.6012132 PM	calc.exe	2072	CreateFile	C:\Windows\System32\WindowsCodecs.dll	SUCCESS	Desired Access: Read Data/List ...
1:17:48.6012344 PM	calc.exe	2072	CreateFileMapp...	C:\Windows\System32\WindowsCodecs.dll	FILE LOCKED WI...	SyncType: SyncTypeCreateSecti...
1:17:48.6012901 PM	calc.exe	2072	CreateFileMapp...	C:\Windows\System32\WindowsCodecs.dll	SUCCESS	SyncType: SyncTypeOther
1:17:48.6013372 PM	calc.exe	2072	Load Image	C:\Windows\System32\WindowsCodecs.dll	SUCCESS	Image Base: 0x73aa0000, Image ...
1:17:48.6013796 PM	calc.exe	2072	CloseFile	C:\Windows\System32\WindowsCodecs.dll	SUCCESS	
1:17:48.6015378 PM	calc.exe	2072	RegOpenKey	HKCU\Software\Classes	SUCCESS	Desired Access: Maximum Allowe...
1:17:48.6015591 PM	calc.exe	2072	RegQueryValue	HKCU\Software\Classes	SUCCESS	Query: Name
1:17:48.6015697 PM	calc.exe	2072	RegOpenKey	HKCU\Software\Classes\CLSID\{FAE3D380-FEA4-4...	NAME NOT FOUND	Desired Access: Read
1:17:48.6015797 PM	calc.exe	2072	RegOpenKey	HKCR\CLSID\{FAE3D380-FEA4-4623-8C75-C6B6111...	SUCCESS	Desired Access: Read
1:17:48.6015937 PM	calc.exe	2072	RegQueryKey	HKCU\Software\Classes	SUCCESS	Query: Name
1:17:48.6016002 PM	calc.exe	2072	RegOpenKey	HKCU\Software\Classes\CLSID\{FAE3D380-FEA4-4...	NAME NOT FOUND	Desired Access: Read
1:17:48.6016130 PM	calc.exe	2072	RegOpenKey	HKCR\CLSID\{FAE3D380-FEA4-4623-8C75-C6B6111...	NAME NOT FOUND	Desired Access: Read

Showing 128,723 of 253,268 events (50%)      Backed by virtual memory



# 进程监视器的工具栏





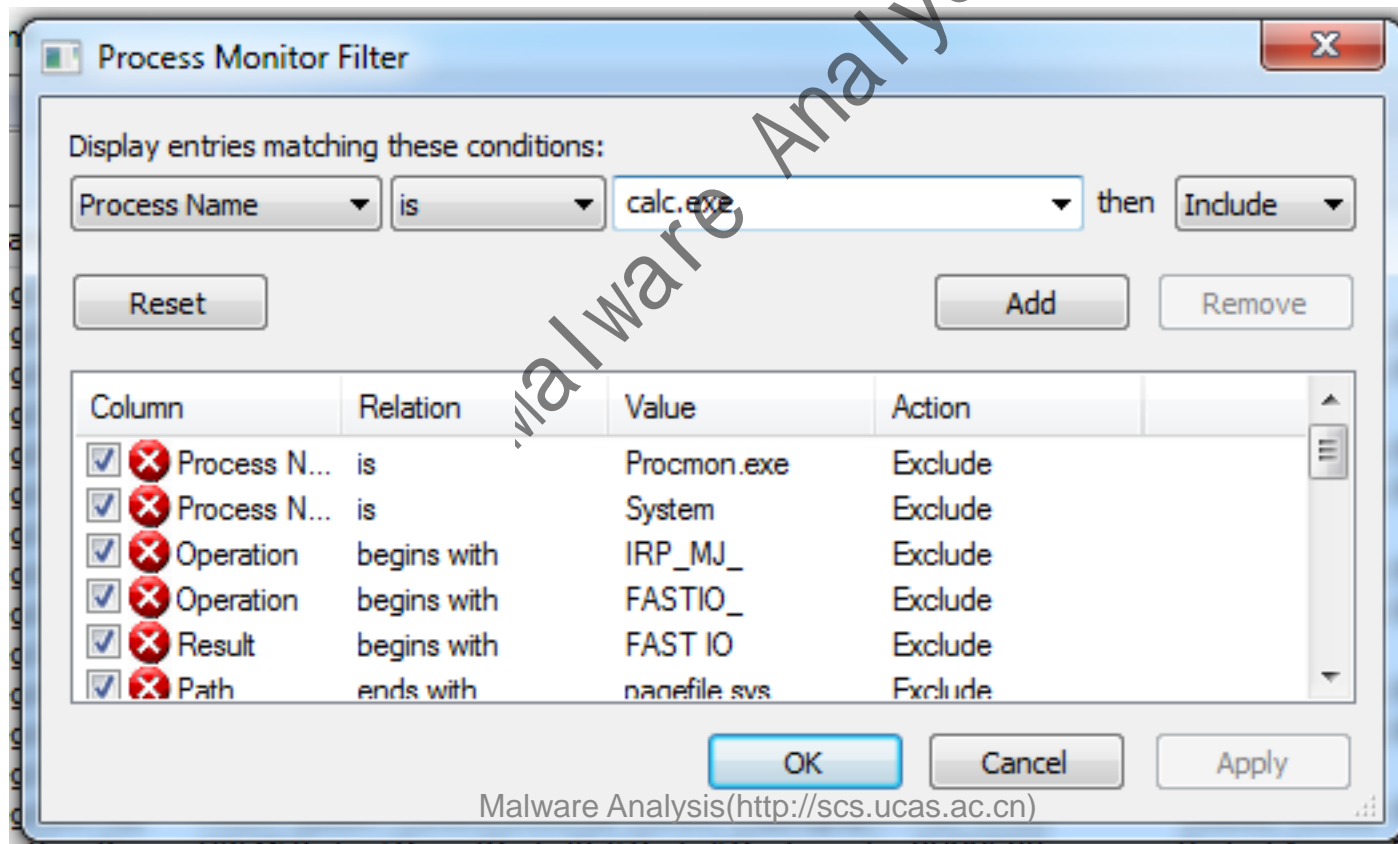
# 使用Exclude过滤

- 有一种技术：将恶意代码隐藏在正常行为之后加载
- 右键点击进程名并点击**Exclude**



# 使用Include过滤

- 最有用的过滤方法：进程名称、操作和一些细节





# 使用Process Explorer查看进程

Malware Analysis

Process Explorer - Sysinternals: www.sysinternals.com [W7\student]

File Options View Process Find Users Help

Process	PID	CPU	Private Bytes	Working Set	Description	Company Name
System Idle Process	0	96.81	0 K	24 K		
System	4	0.09	48 K	560 K		
Interrupts	n/a	0.88	0 K	0 K	Hardware Interrupts and DPCs	
smss.exe	260		224 K	748 K	Windows Session Manager	Microsoft Corporation
csrss.exe	348	< 0.01	1,252 K	3,164 K	Client Server Runtime Process	Microsoft Corporation
wininit.exe	400		892 K	3,084 K	Windows Start-Up Application	Microsoft Corporation
services.exe	504	0.01	3,972 K	6,640 K	Services and Controller app	Microsoft Corporation
svchost.exe	652		2,700 K	6,024 K	Host Process for Windows S...	Microsoft Corporation
dllhost.exe	1716		6,176 K	4,804 K	COM Surrogate	Microsoft Corporation
WmiPrvSE.exe	740		1,804 K	4,736 K	WMI Provider Host	Microsoft Corporation
svchost.exe	724	< 0.01	2,972 K	6,012 K	Host Process for Windows S...	Microsoft Corporation
svchost.exe	772		19,776 K	11,760 K	Host Process for Windows S...	Microsoft Corporation
audiodg.exe	3200		14,960 K	13,972 K	Windows Audio Device Grap...	Microsoft Corporation
svchost.exe	912		37,940 K	42,292 K	Host Process for Windows S...	Microsoft Corporation
dwm.exe	3248	0.74	61,892 K	27,976 K	Desktop Window Manager	Microsoft Corporation
svchost.exe	936	0.02	20,836 K	29,900 K	Host Process for Windows S...	Microsoft Corporation
svchost.exe	1116	0.03	5,136 K	8,340 K	Host Process for Windows S...	Microsoft Corporation
svchost.exe	1260	0.06	10,840 K	11,960 K	Host Process for Windows S...	Microsoft Corporation
spoolsv.exe	1352		5,392 K	7,436 K	Spooler SubSystem App	Microsoft Corporation
svchost.exe	1388		6,752 K	8,720 K	Host Process for Windows S...	Microsoft Corporation
svchost.exe	1500		2,472 K	4,712 K	Host Process for Windows S...	Microsoft Corporation
gogoc.exe	1592	< 0.01	1,216 K	3,920 K	gogoCLIENT	gogo6, Inc.
vmtoolsd.exe	1728	0.07	7,260 K	10,368 K	VMware Tools Core Service	VMware, Inc.
svchost.exe						

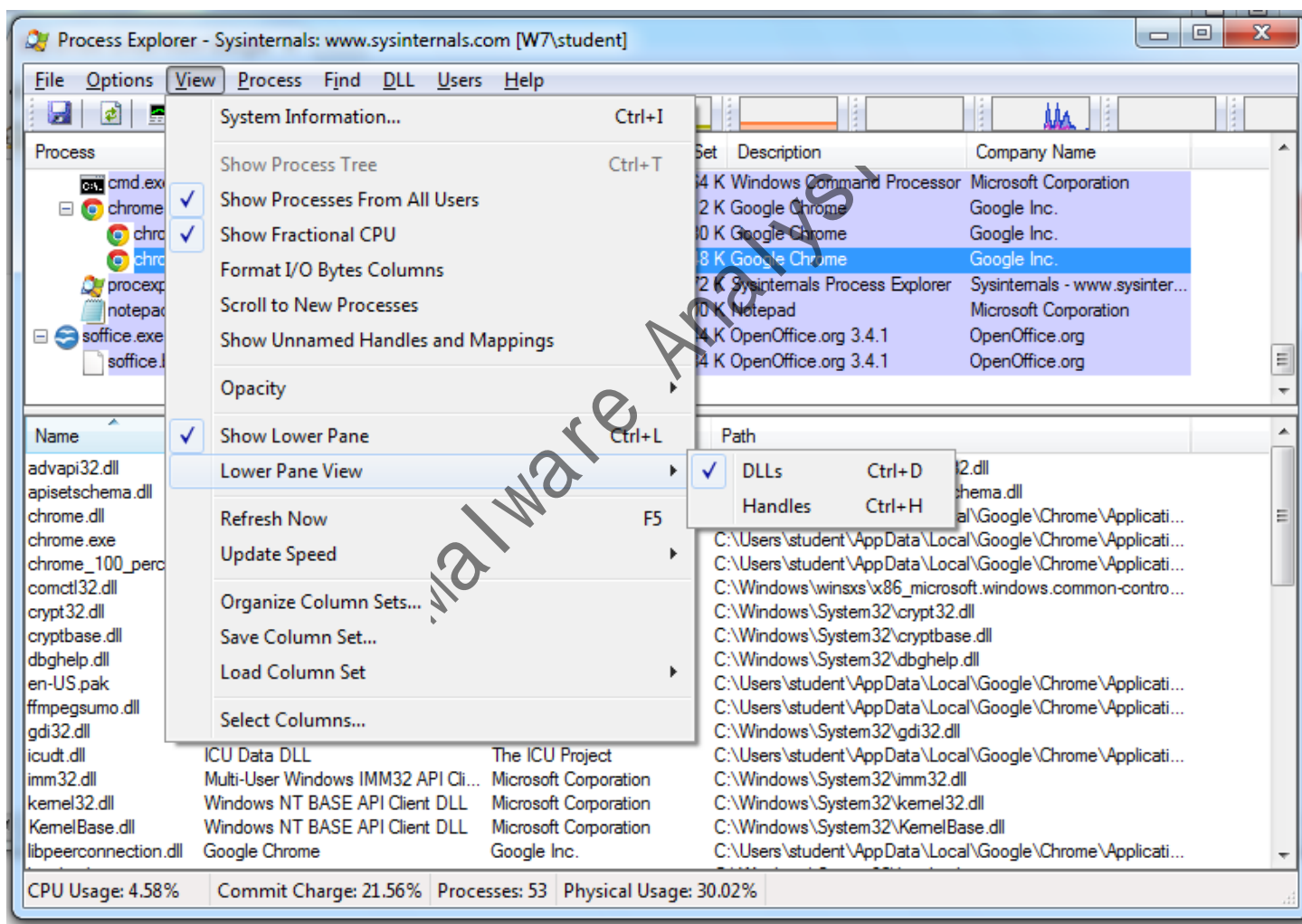
CPU Usage: 3.19% Commit Charge: 21.92% Processes: 57 Physical Usage: 30.24%



# 颜色

- 服务显示为粉红色
- 进程显示为蓝色
- 新进程显示为绿色
- 终止的进程显示为红色

# DLL 模式

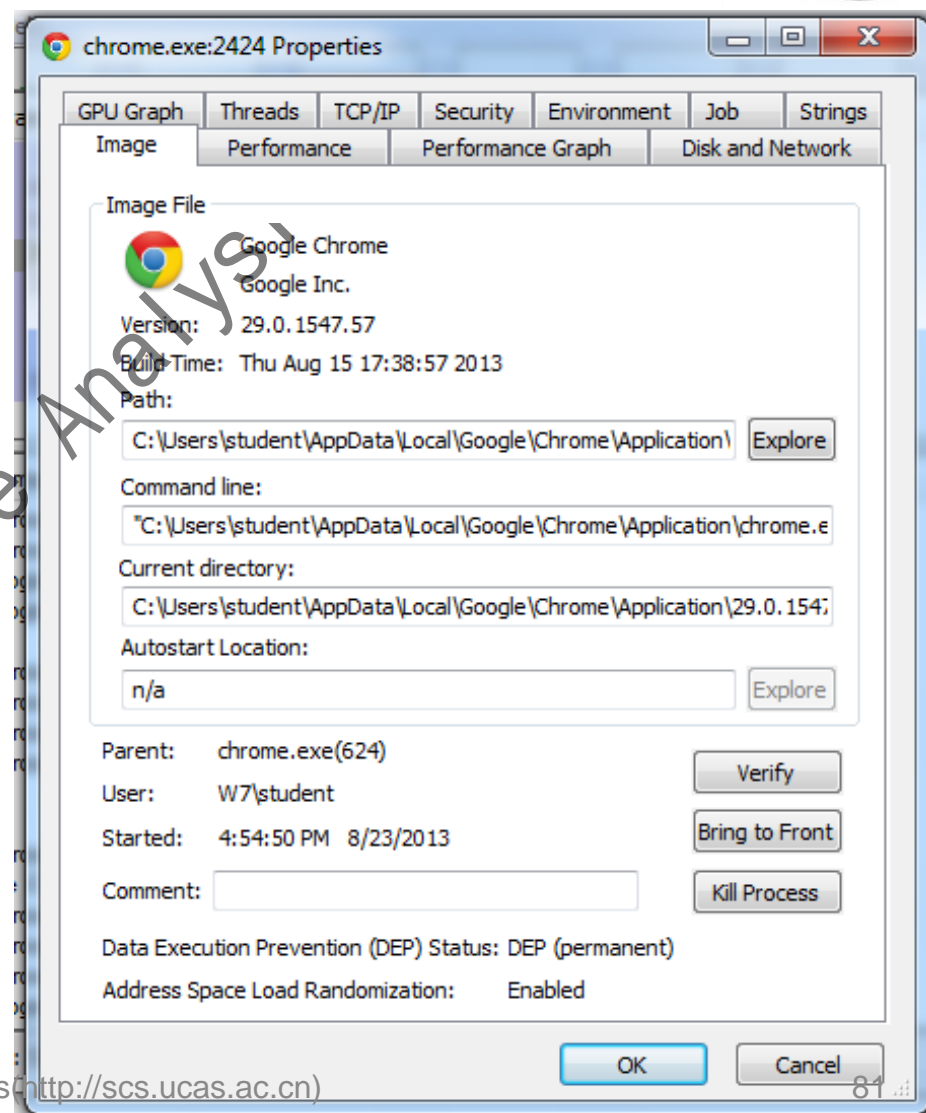






# 属性

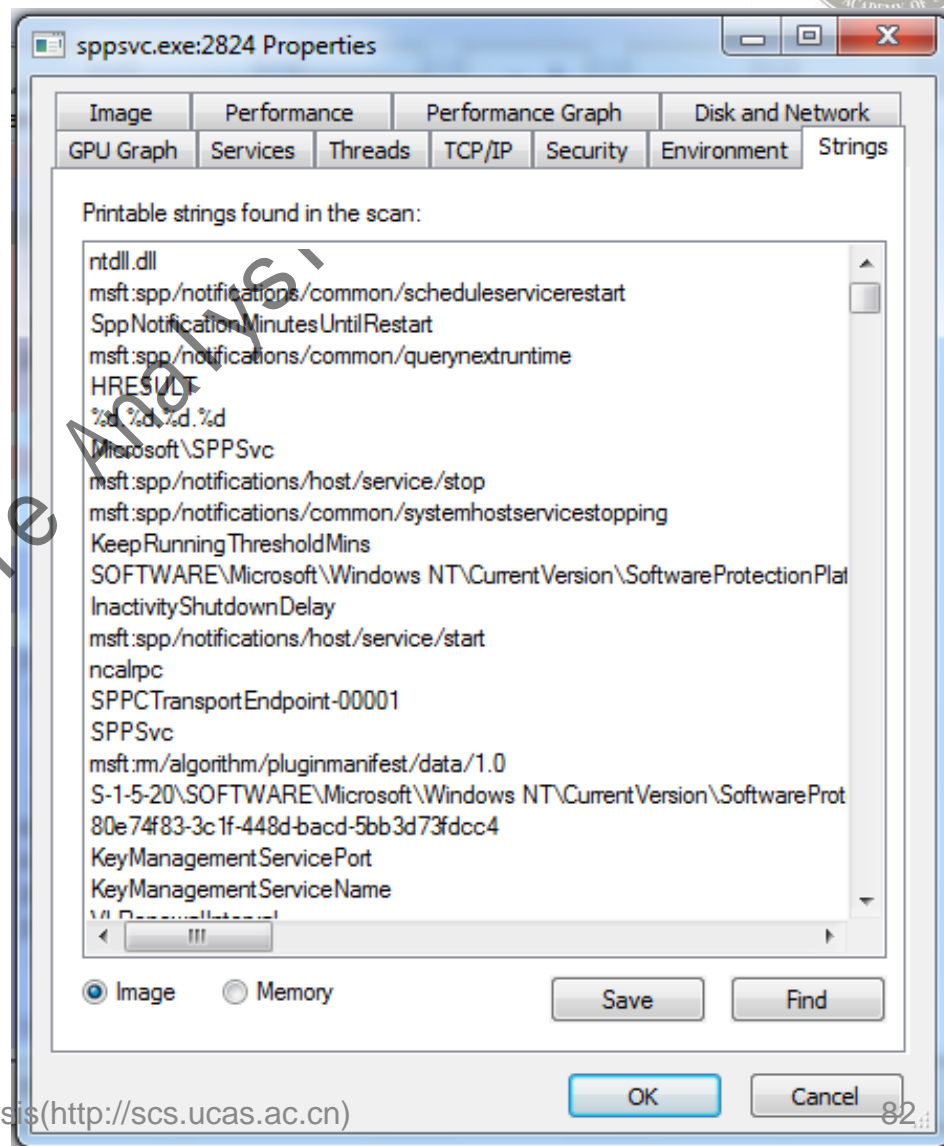
- 显示 DEP 和 ASLR 状态
- Verify 按钮检查磁盘文件的Windows签名
  - 不是检测内存镜像，所以不能检测进程替换





# 字符串

- 比较包含在磁盘上的可执行文件的字符串与内存中同一个执行文件的字符串，如果发现他们直接有很大不同，那么可能发生了内存替换



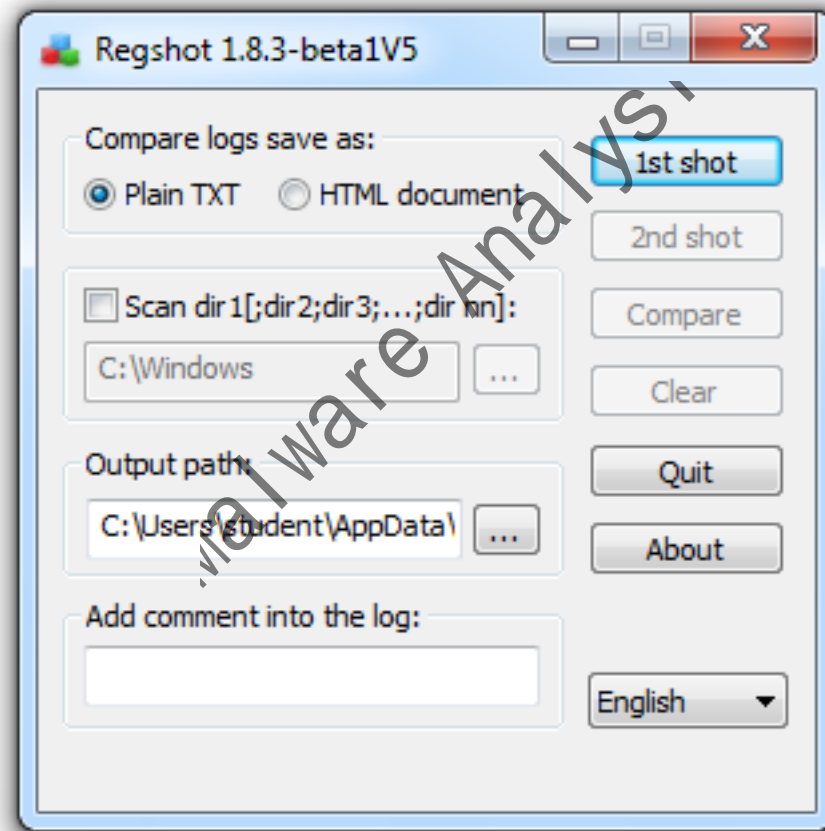


# 分析恶意文档

- 在系统中用存在漏洞的应用打开文档（如PDF）
- 通过**Process Explorer**观察它是否加载了一个进程
- 进程属性页中的**Image**标签能够显示恶意代码在磁盘上的位置



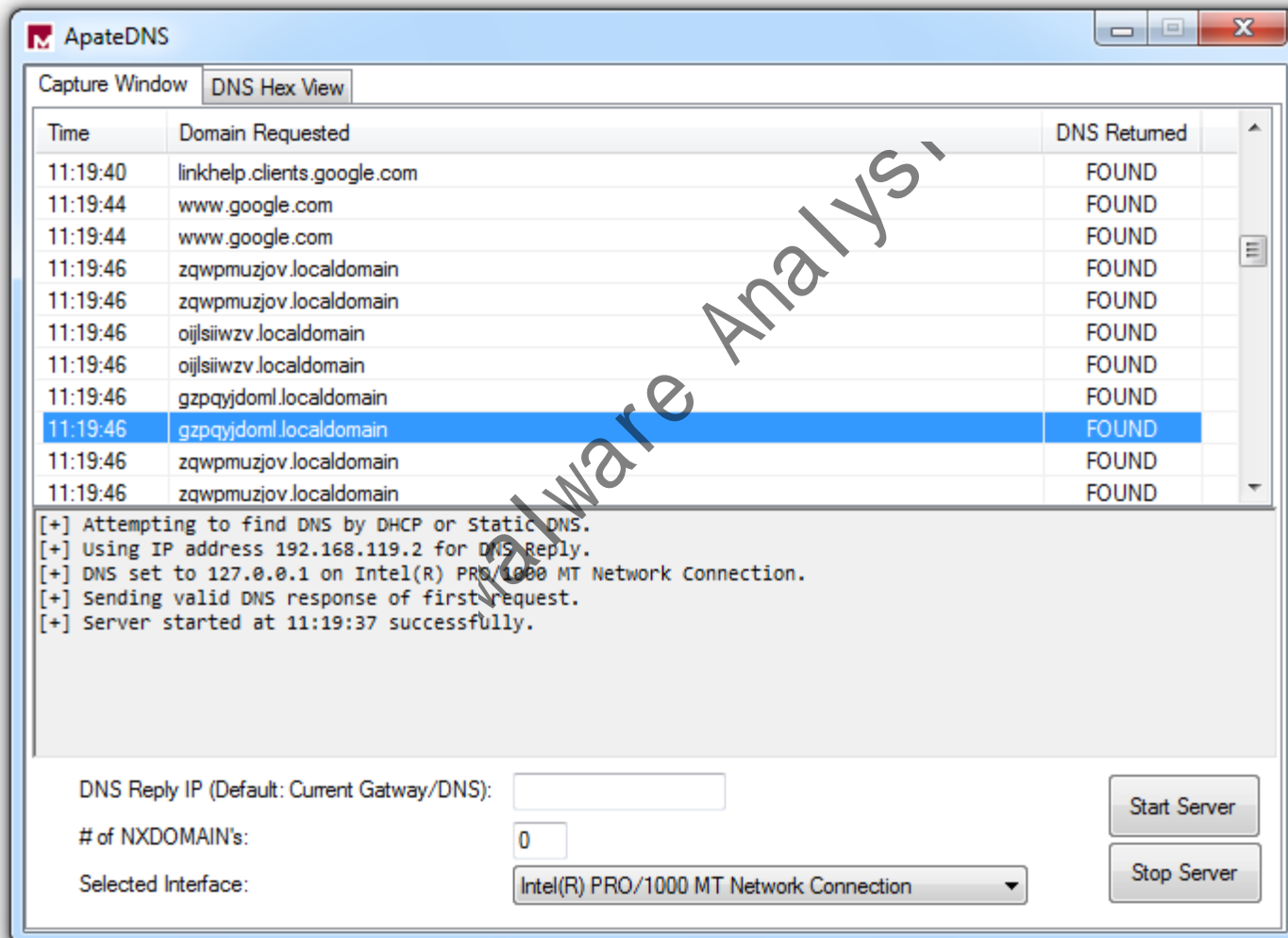
# 使用Regshot比较注册快照





# 模拟网络

# 使用ApateDNS 重定向DNS





# 若ApateDNS不起作用

- 在Win XP 或 Win7中不能重定向网络流量
- nslookup 可以用，但不能看到浏览器或ping中的内容
- 可以用INetSim 代替ApateDNS

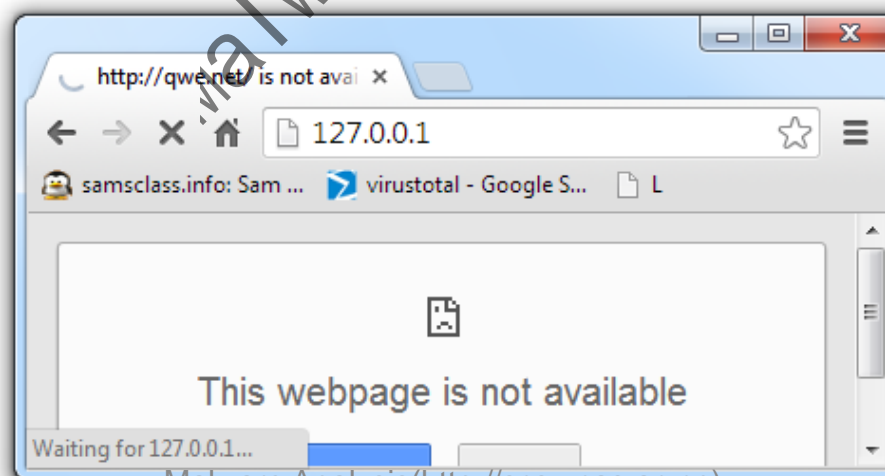




# 使用Ncat进行监视

```
Administrator: cmd - Shortcut (2) - ncat -l 80

C:\Windows\System32>ncat -l 80
GET / HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.57 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
```



# 使用Wireshark进行数据包监听



Capturing from Intel(R) PRO/1000 MT Network Connection

Filter: http

No.	Time	Source	Destination	Protocol	Info
1101	7.515707	192.168.119.154	23.65.1.224	HTTP	GET /f.gif?_id=137745723/561
1106	7.537336	18.181.0.31	192.168.119.154	HTTP	HTTP/1.1 200 OK (PNG)
1108	7.557449	93.184.216.139	192.168.119.154	HTTP	[TCP Retransmission] Cont
1110	7.590291	23.65.1.224	192.168.119.154	HTTP	HTTP/1.1 200 OK (GIF89a)
1111	7.691258	23.65.1.224	192.168.119.154	HTTP	[TCP Retransmission] HTTP/
1189	36.858744	192.168.119.154	199.16.156.21	HTTP	GET /widgets/timelines/pag
1193	36.881799	192.168.119.154	199.16.156.21	HTTP	GET /widgets/timelines/pag
1196	36.954204	199.16.156.21	192.168.119.154	HTTP	HTTP/1.1 200 OK (applicat
1199	37.045979	199.16.156.21	192.168.119.154	HTTP	HTTP/1.1 200 OK (applicat
1369	96.750725	192.168.119.154	199.16.156.21	HTTP	GET /widgets/timelines/pag
1373	96.772892	192.168.119.154	199.16.156.21	HTTP	GET /widgets/timelines/pag
1376	96.846439	199.16.156.21	192.168.119.154	HTTP	HTTP/1.1 200 OK (applicat
1381	96.944497	199.16.156.21	192.168.119.154	HTTP	HTTP/1.1 200 OK (applicat

Frame 48: 437 bytes on wire (3496 bits), 437 bytes captured (3496 bits)

Ethernet II, Src: Vmware\_52:34:92 (00:0c:29:52:34:92), Dst: Vmware\_e3:22:f1 (00:50:56:00:00:00)

Internet Protocol Version 4, Src: 192.168.119.154 (192.168.119.154), Dst: 141.101.11

0000 00 50 56 e3 22 f1 00 0c 29 52 34 92 08 00 45 00 .PV."... )R4...E.  
0010 01 a7 10 25 40 80 06 00 00 c0 a8 77 9a 8d 65 ....@... ..w...e  
0020 75 98 05 a9 00 50 0c 80 cd 2e dc ff 73 93 50 18 u....P... ..S.P.  
0030 fa f0 3c da 00 00 47 45 54 20 2f 20 48 54 54 50 ..<...GE T / HTTP  
0040 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 73 61 6d 73 /1.1..Ho st: sams  
0050 63 6c 61 73 73 2e 69 6e 66 6f 0d 0a 43 6f 6e 6e class.in fo..Conn  
0060 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 ection: keep-ali  
0070 76 65 0d 0a 41 63 63 65 70 74 3a 20 74 65 78 74 ve..Acce pt: text  
0080 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f /html,ap plicatio  
0090 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c n/xhtml1+ xml,appl  
00a0 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 2e ication/ xml:o=0.

Intel(R) PRO/1000 MT Network Connection: <live capture in pro... Packets: 1398 Dis... Profile: Default

samsclass.info: Sam Bowne x

samsclass.info

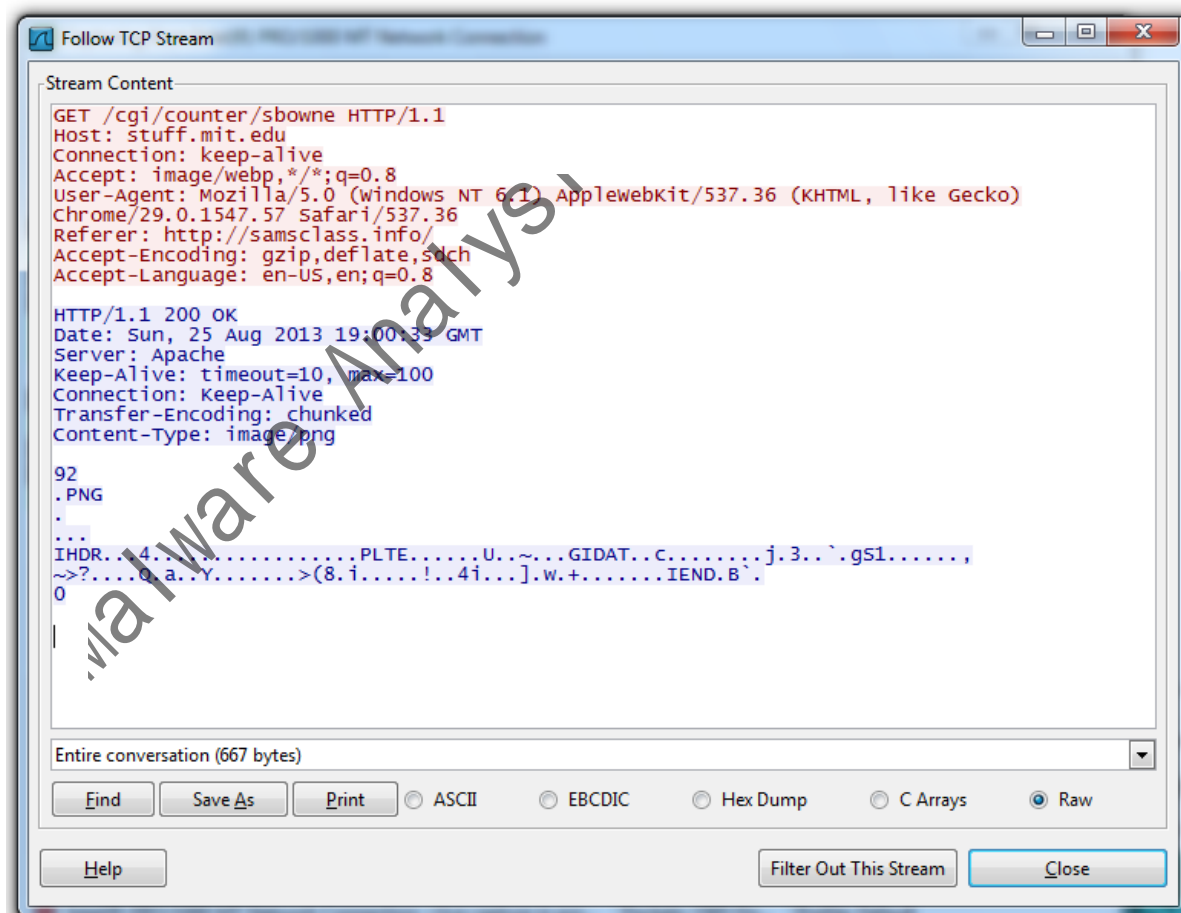
samsclass.info: Sam ... virustotal - Goc

Sam Bowne



# Follow TCP Stream窗口

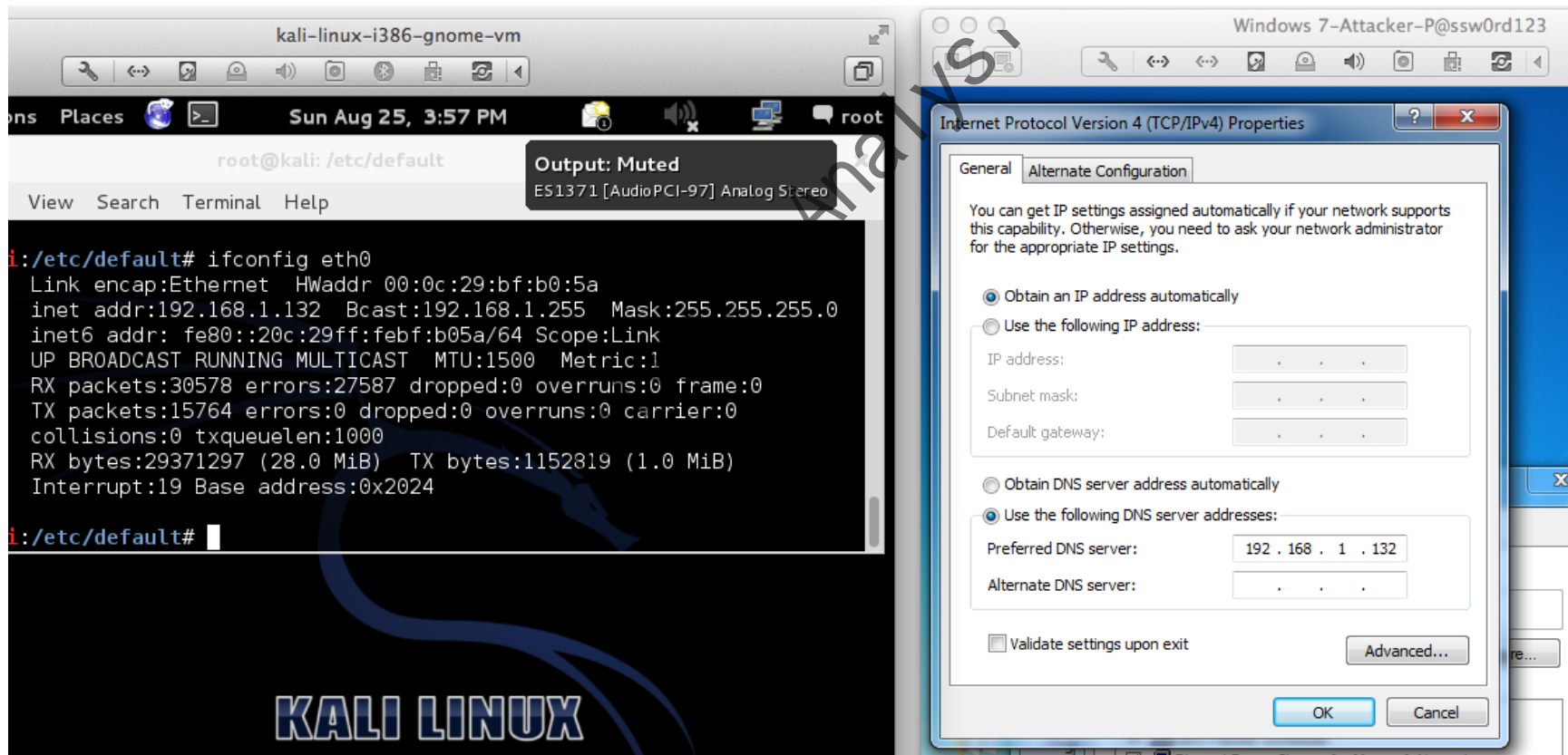
- Wireshark存在许多安全漏洞，一定要在一个安全的环境里运行它



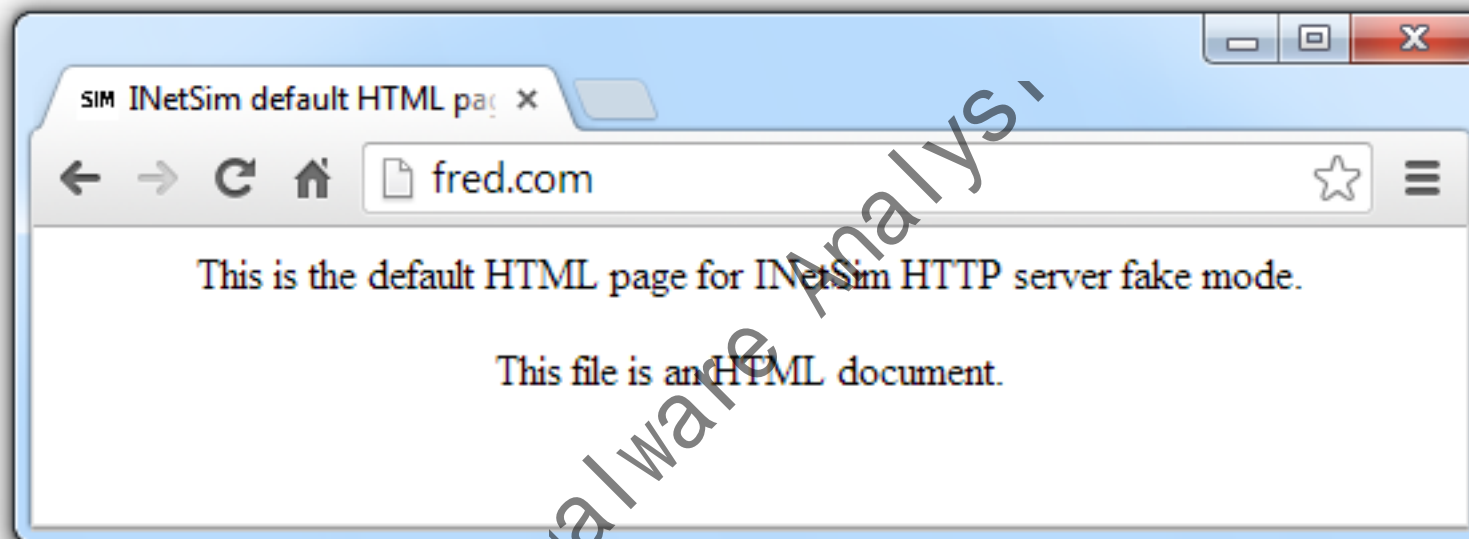


# Using INetSim

# 使用Inetsim



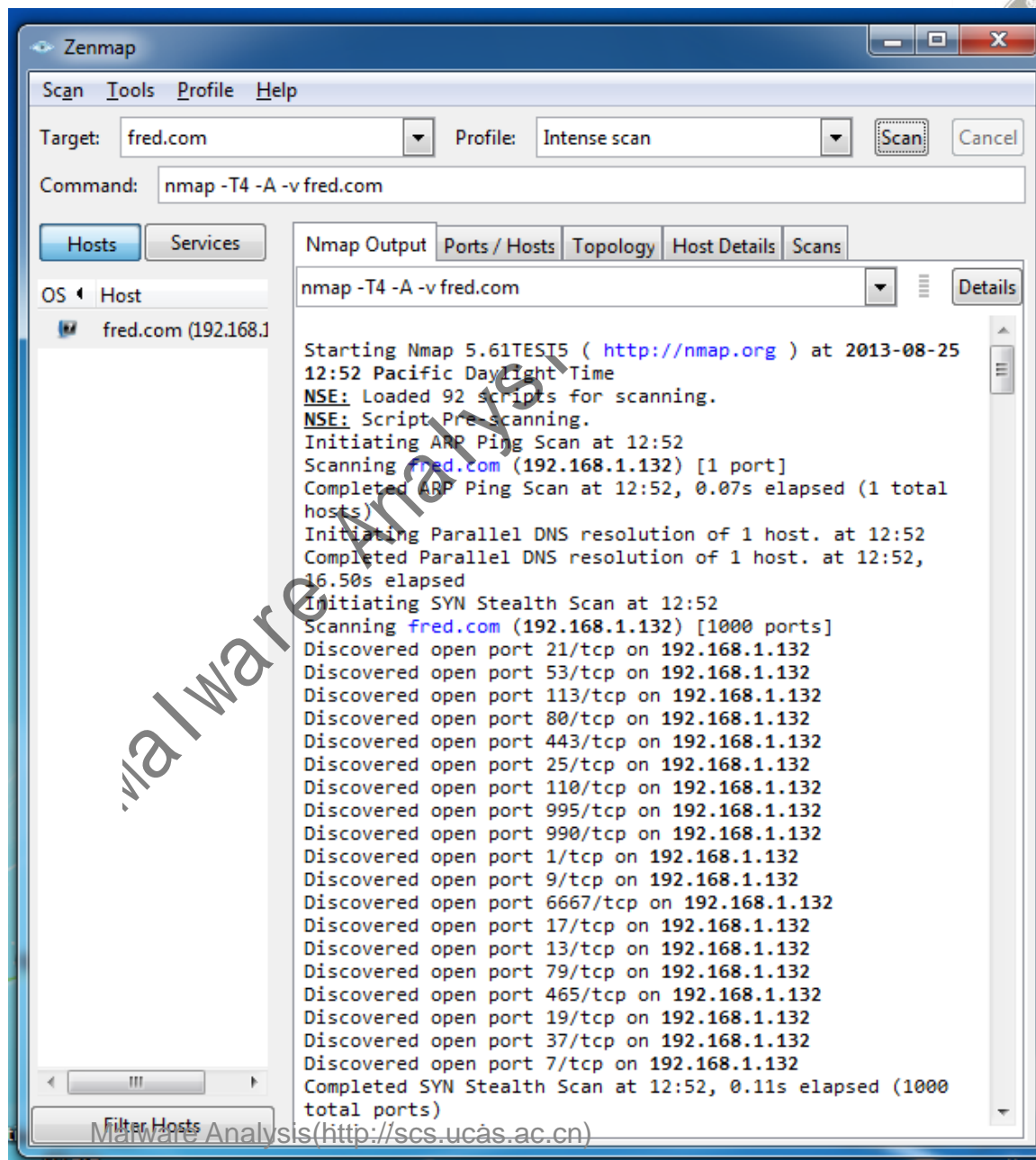
# InetSim 欺骗浏览器



# INetSim

## 欺骗

## Nmap





# 基础动态分析工具实践

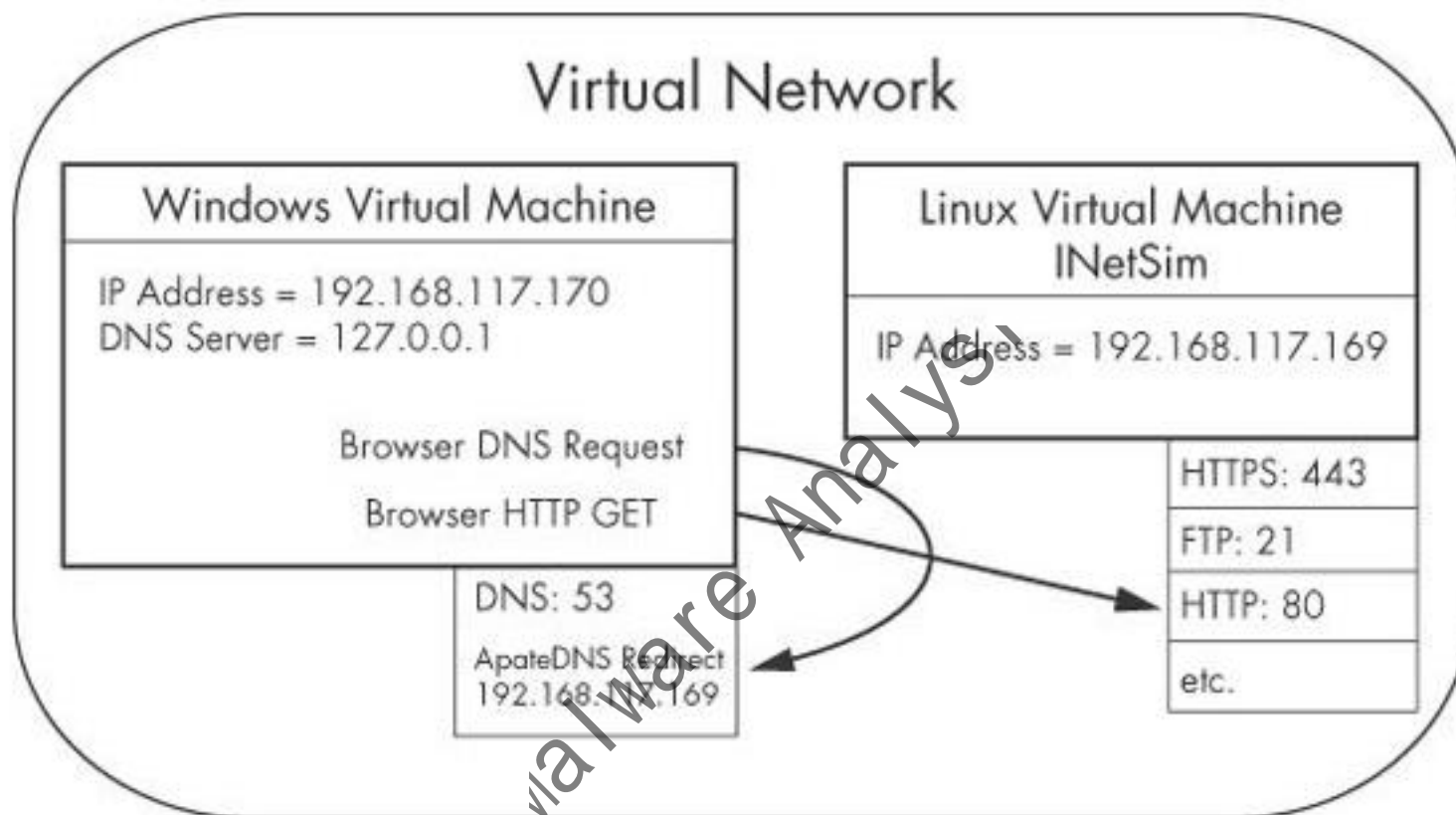


# 回顾本章讨论的所有工具



- Procmon
- Process Explorer
- Regshot
- INetSim
- Wireshark

Malware Analysis



*Figure 4-12. Example of a virtual network*