X'CON 2003

病毒检测技术的取证应用--
外延化的广谱检测引擎和技术体制

感谢安全焦点的兄弟们长久以来对Antiy Labs的帮助和支持

作者：_江海客__

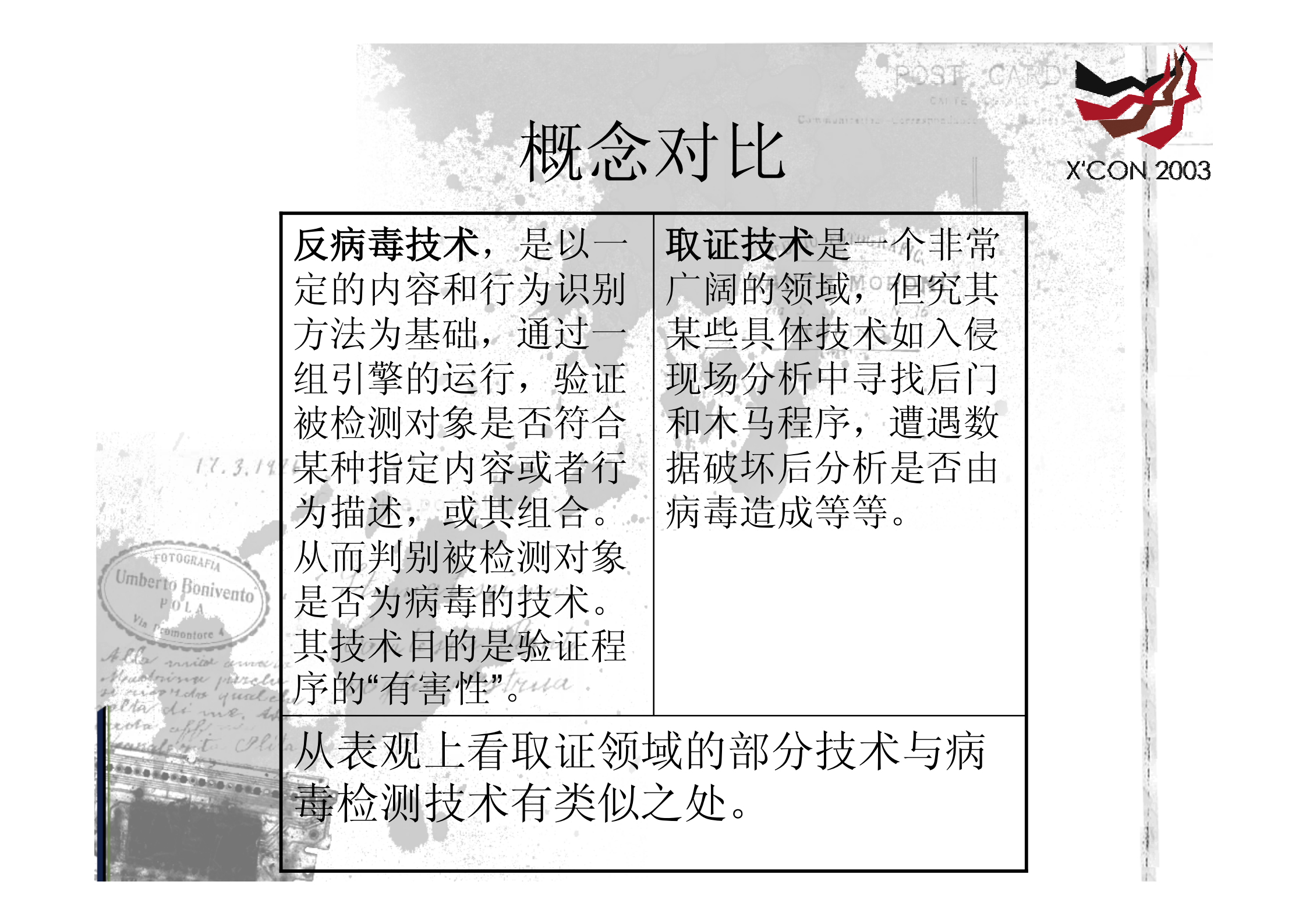
日期：_20031220

纲要

- 技术对比
- 走向取证的自发性发展
- 样本的分析体制与新探索
- 技术现状

一、技术对比

概念对比



反病毒技术，是以一定的内容和行为识别方法为基础，通过一组引擎的运行，验证被检测对象是否符合某种指定内容或者行为描述，或其组合。从而判别被检测对象是否为病毒的技术。其技术目的是验证程序的“有害性”。

取证技术是一个非常广阔的领域，但究其某些具体技术如入侵现场分析中寻找后门和木马程序，遭遇数据破坏后分析是否由病毒造成等等。

从表观上看取证领域的部分技术与病毒检测技术有类似之处。

技术相似

- 首先，相关的取证技术的部分环节与反病产品的性质一样，需要在大量的数据中，确定其中符合某种定义的信息的存在。
- 其次，从广义的理论领域来说，都可以视为信息指纹技术的相关分支。

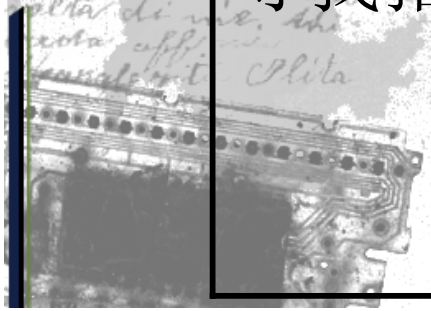
不同点

- 取证系统应最少的影响现场，不应影响文件的最后访问时间，而这在反病毒系统中则不需要考虑。反病毒软件不仅要检测病毒，也要能够清除和遏制病毒，而取证系统则不需要清除处理。
- 取证产品的检测范围要比反病毒软件更广泛，反病毒软件在报警上比较慎重，因此对类似商用spyware、远程控制工具等等，都不作检测，而取证产品则需要找到更多的信息。
- 同时，取证系统需要更深入的关联分析，取证技术最终是要确定某个事件背后的逻辑关系，如确定一个程序是否为有害程序，对反病毒产品来说，这本身既是一个目的，而在取证技术来说，这只是一个手段。

取证需求分析

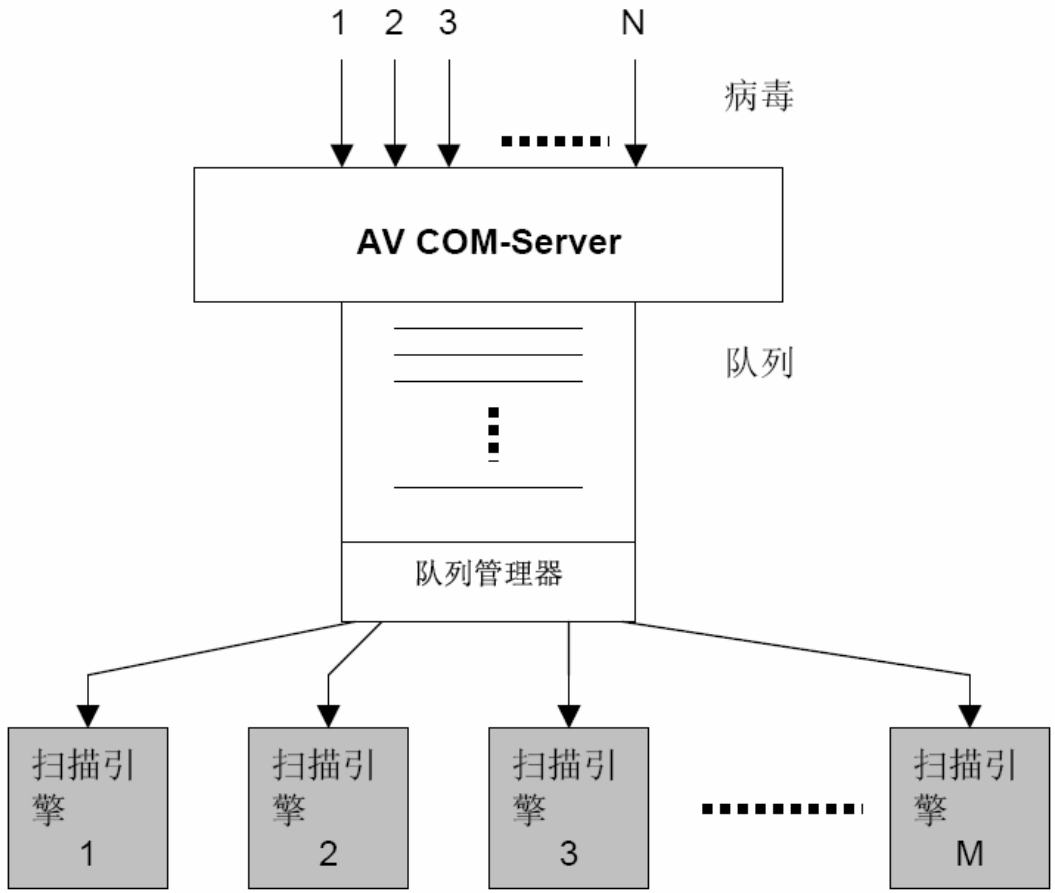
- 寻找指定格式文件
- 寻找指定功能文件
- 寻找指定内容文件

需求和对应技术



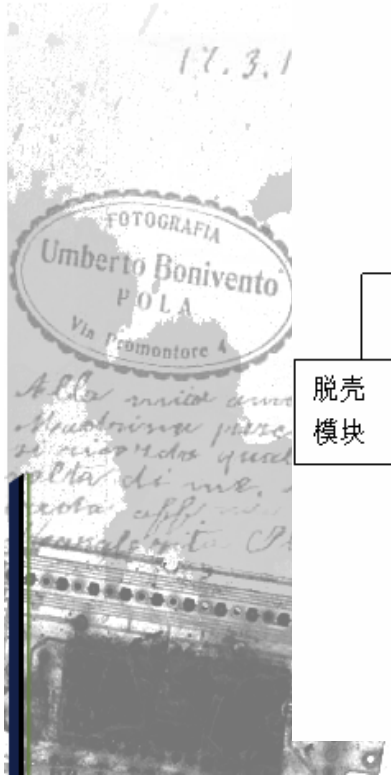
相关需求	对应技术
寻找指定格式文件	格式识别技术
寻找指定功能文件	特征码识别技术，高速匹配扫描技术
寻找指定内容文件	格式还原技术，内容抽取技术、高速匹配技术

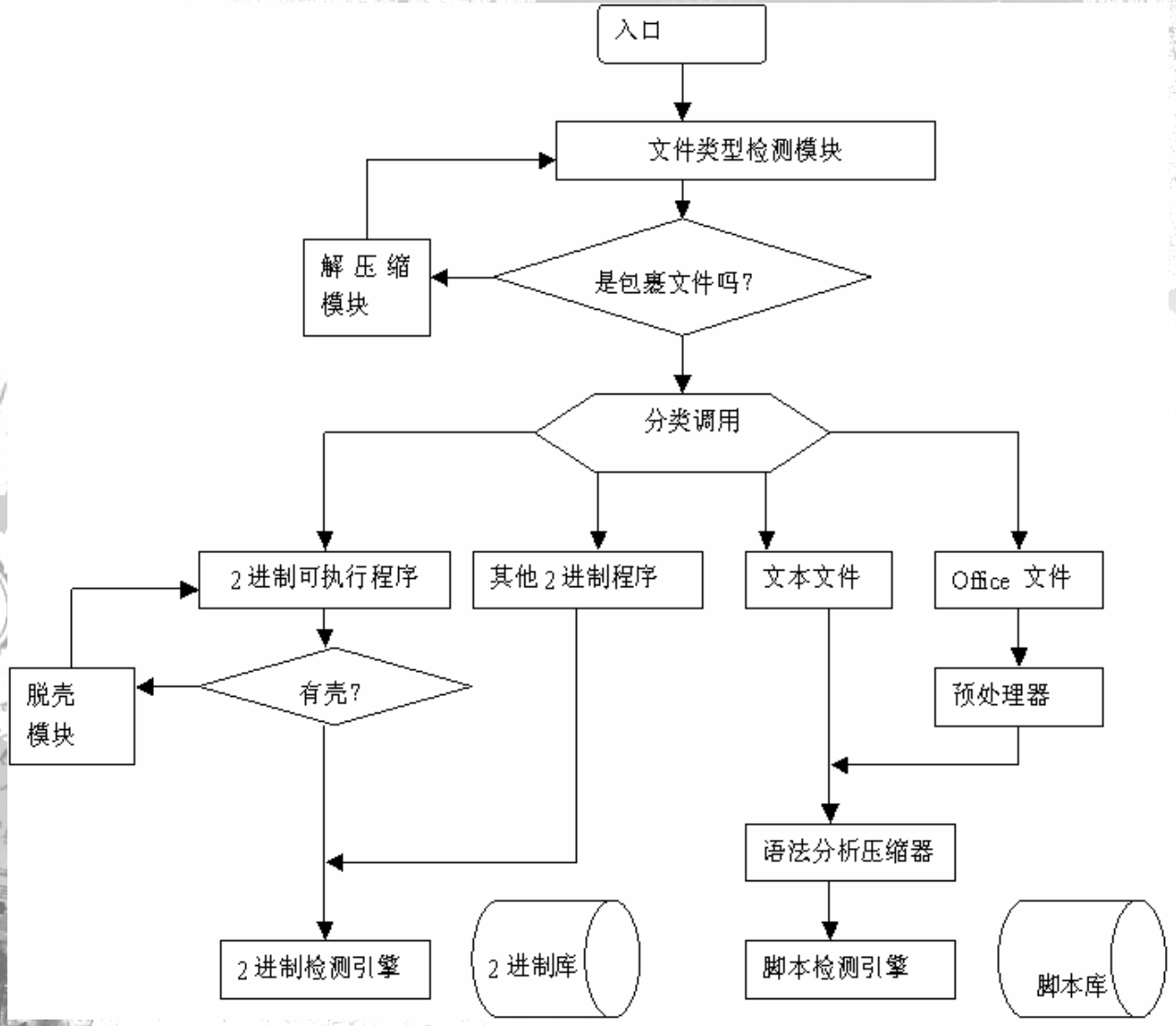
反病毒引擎已经解决上面问题



图片引自Kaspersky Labs

引擎工作示例


FOTOGRAFIA
Umberto Bonivento
POLA

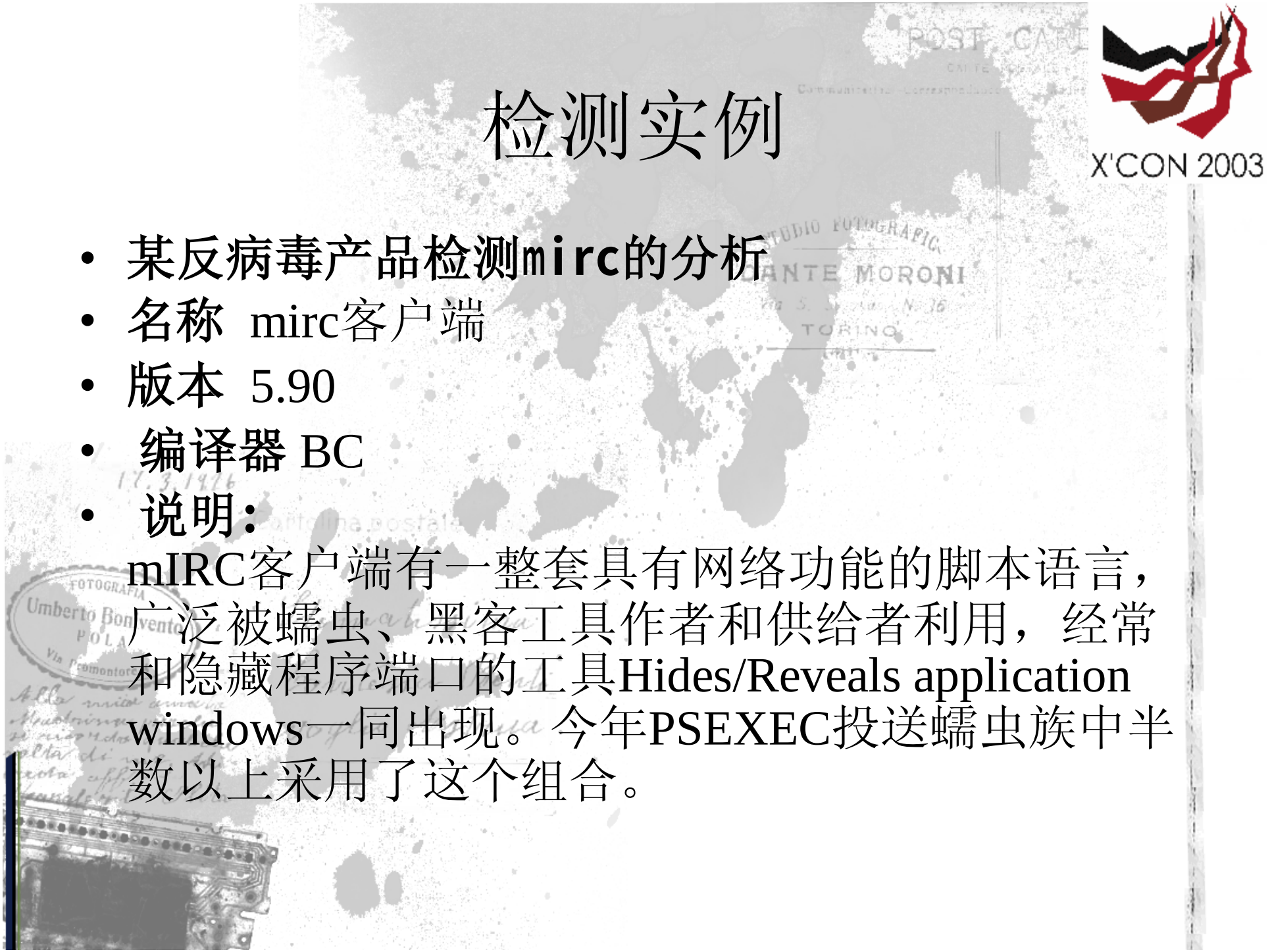

入口
文件类型检测模块
是包裹文件吗?
解压缩模块
分类调用
2进制可执行程序
其他2进制程序
文本文件
Office 文件
有壳?
脱壳模块
预处理器
语法分析压缩器
2进制检测引擎
2进制库
脚本检测引擎
脚本库

二、取证应用的自发性发展

问题与挑战

- 检测范围的自然扩大
- AV辩证法面临的挑战

检测实例



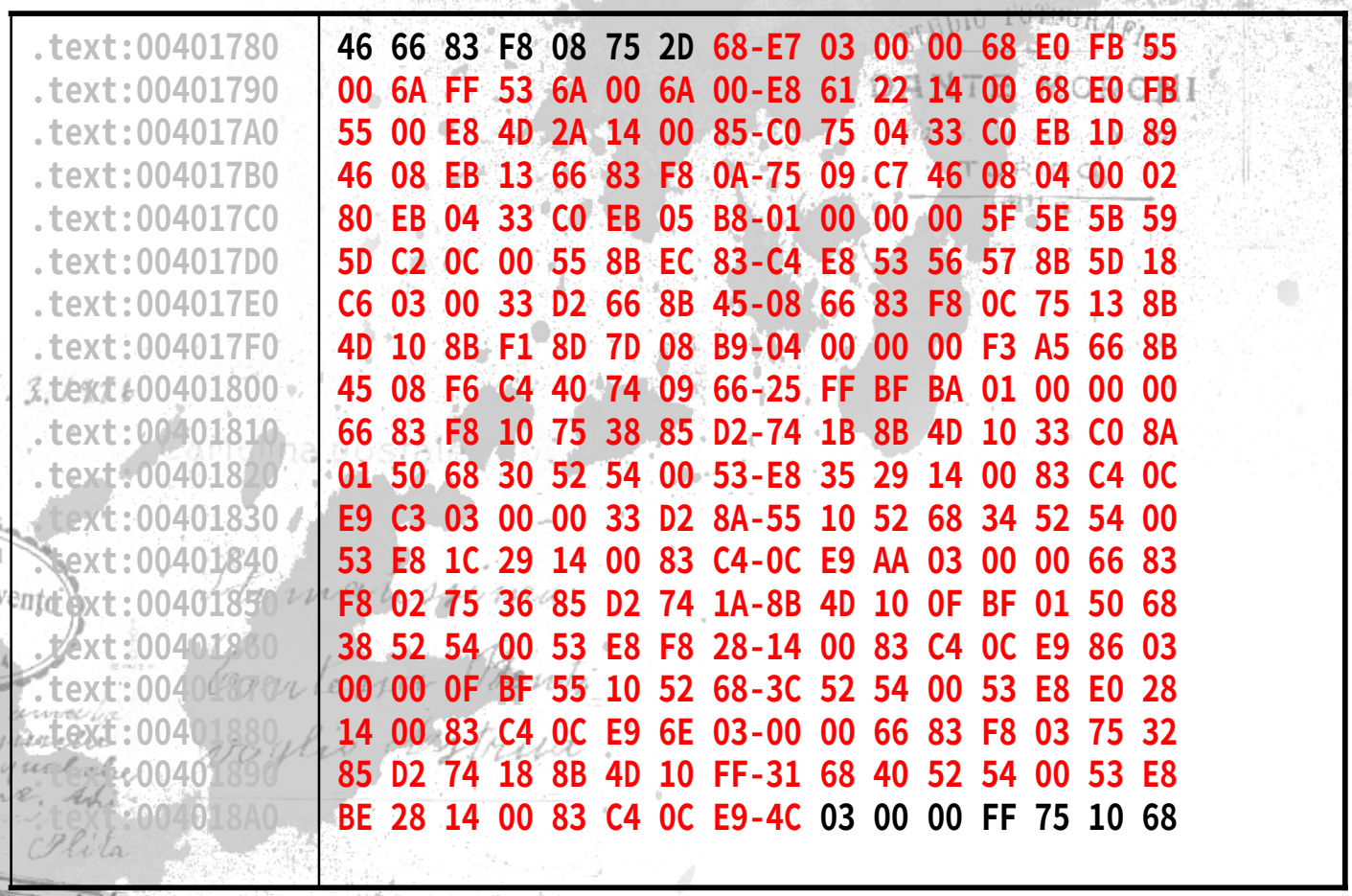
- 某反病毒产品检测mirc的分析
- 名称 mirc客户端
- 版本 5.90
- 编译器 BC
- 说明：

mIRC客户端有一整套具有网络功能的脚本语言，广泛被蠕虫、黑客工具作者和供给者利用，经常和隐藏程序端口的工具Hides/Reveals application windows一同出现。今年PSEXEC投送蠕虫族中半数以上采用了这个组合。

主要检测机理

- 得到一个扫描文件对象后，首先通过一个格式检测模块（smart）判定文件类型，如为PE文件，则文件类型返回“08”，将文件交给PE预处理引擎，预处理引擎将对PE文件进一步进行脱壳和还原处理，最终形成一个1K的动态的查毒缓冲区。之后，引擎根据病毒库中的一个word和两组相对偏移量的指定长度的内容的数字签名值对文件进行验证。

特征码



```
.text:00401780  46 66 83 F8 08 75 2D 68-E7 03 00 00 68 E0 FB 55
.text:00401790  00 6A FF 53 6A 00 6A 00-E8 61 22 14 00 68 E0 FB
.text:004017A0  55 00 E8 4D 2A 14 00 85-C0 75 04 33 C0 EB 1D 89
.text:004017B0  46 08 EB 13 66 83 F8 0A-75 09 C7 46 08 04 00 02
.text:004017C0  80 EB 04 33 C0 EB 05 B8-01 00 00 00 5F 5E 5B 59
.text:004017D0  5D C2 0C 00 55 8B EC 83-C4 E8 53 56 57 8B 5D 18
.text:004017E0  C6 03 00 33 D2 66 8B 45-08 66 83 F8 0C 75 13 8B
.text:004017F0  4D 10 8B F1 8D 7D 08 B9-04 00 00 00 F3 A5 66 8B
.text:00401800  45 08 F6 C4 40 74 09 66-25 FF BF BA 01 00 00 00
.text:00401810  66 83 F8 10 75 38 85 D2-74 1B 8B 4D 10 33 C0 8A
.text:00401820  01 50 68 30 52 54 00 53-E8 35 29 14 00 83 C4 0C
.text:00401830  E9 C3 03 00 00 33 D2 8A-55 10 52 68 34 52 54 00
.text:00401840  53 E8 1C 29 14 00 83 C4-0C E9 AA 03 00 00 66 83
.text:00401850  F8 02 75 36 85 D2 74 1A-8B 4D 10 0F BF 01 50 68
.text:00401860  38 52 54 00 53 E8 F8 28-14 00 83 C4 0C E9 86 03
.text:00401870  00 00 0F BF 55 10 52 68-3C 52 54 00 53 E8 E0 28
.text:00401880  14 00 83 C4 0C E9 6E 03-00 00 66 83 F8 03 75 32
.text:00401890  85 D2 74 18 8B 4D 10 FF-31 68 40 52 54 00 53 E8
.text:004018A0  BE 28 14 00 83 C4 0C E9-4C 03 00 00 FF 75 10 68
```

对应代码

```
push 3E7h ; cchWideChar
push offset WideCharStr ; lpWideCharStr
push 0FFFFFFFFh ; cbMultiByte
push ebx ; lpMultiByteStr
push 0 ; dwFlags
push 0 ; CodePage
call MultiByteToWideChar
push offset WideCharStr ; OLECHAR *
call SysAllocString
... ...
loc_4018AC: ; CODE XREF: sub_4017D4+BE j
push [ebp+lpWideCharStr]
push offset aLd_0 ; "%ld"
push ebx
call wsprintfA
add esp, 0Ch
```

二次定位

① 1、对文件名前若干个字符基于一个词表（约20个）做一个比较，如 expl* ; syst*;wind*.*;rund*.*;ms*.*; 如果在表内，则认为是Mirc类恶意程序（置AX）并返回，如果不在表内，则继续执行2。

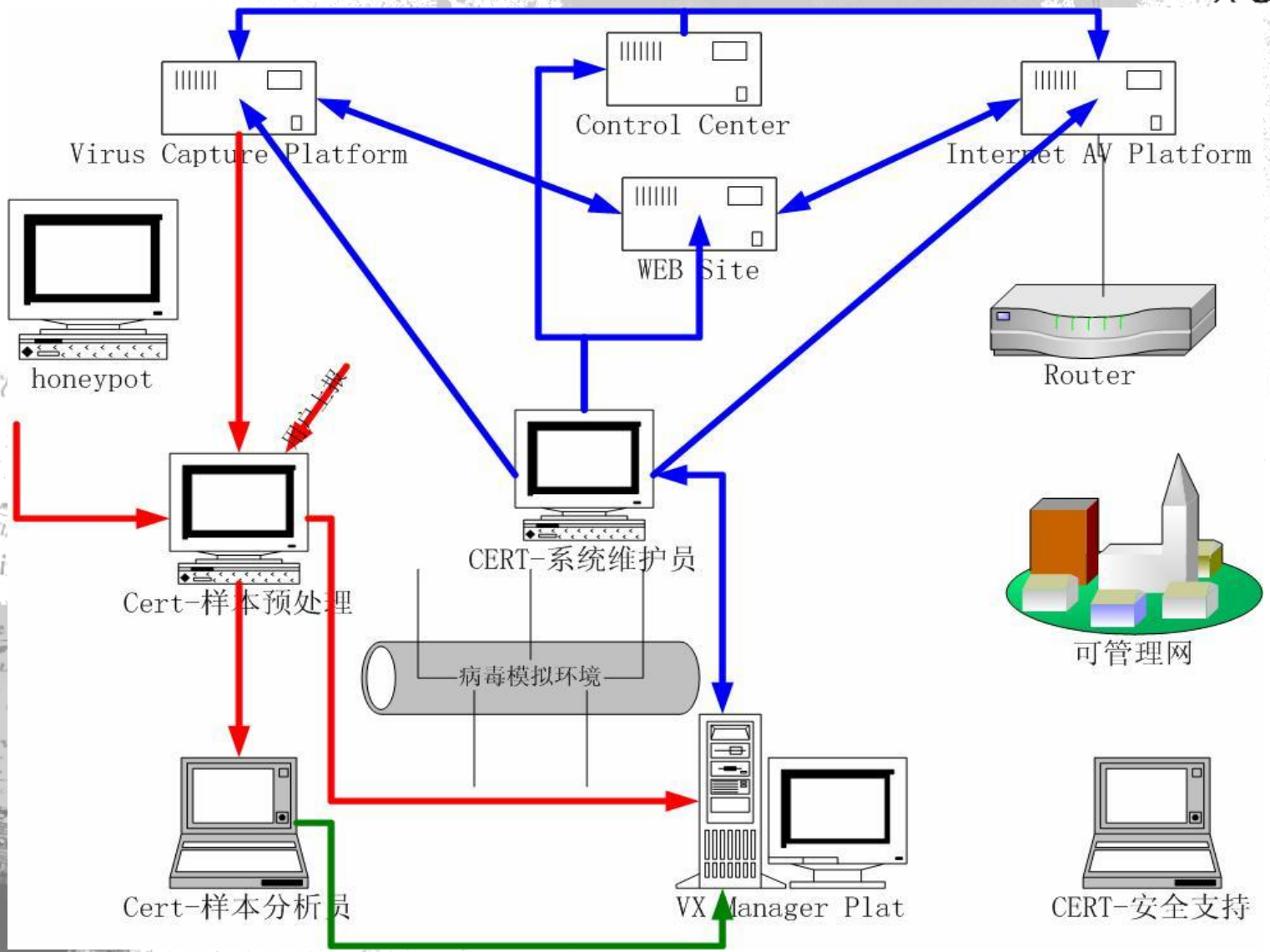
②**2**、从文件名左起寻找第一个".",如果没有找到，则认为是正常程序返回（清AX）。如果找到则继续执行3。

3、判断"."之前的字符数是否大于8个，如果大于（长文件名），则认为是正常程序返回，如果少于（含等于）8个字符，则判断文件名的字符是否均为数字，如果是则认为是恶意程序，如果不是则作为正常程序返回。

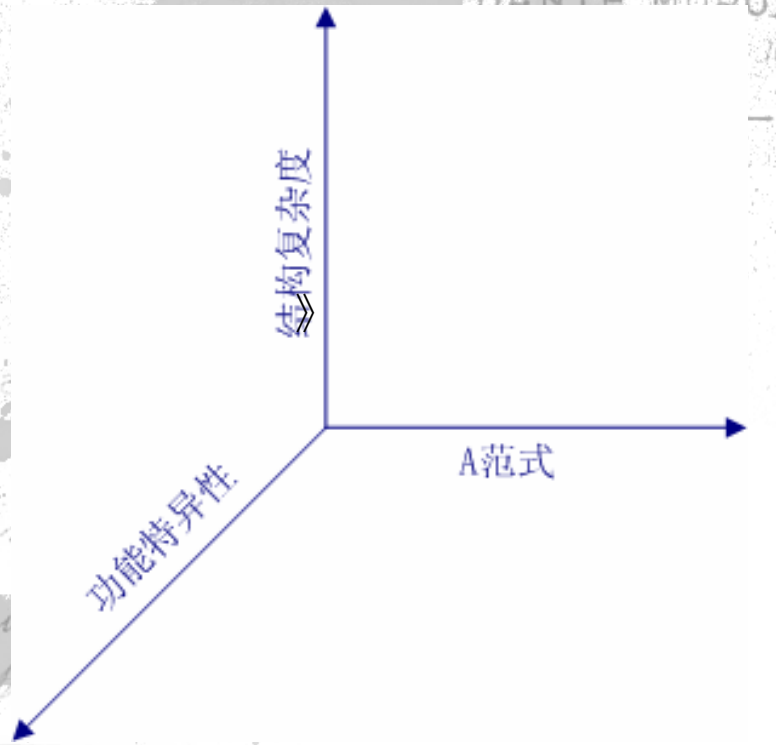
```
...
cmp eax, 646E7572h ;是否为rund
ja short loc_D5
jz loc_284
cmp eax, 646E6977h ;是否为wind
jz loc_284
jmp loc_252
....
cmp ecx, 736Dh ;是否为ms
jnz short loc_289
...
loc_284: ; CODE XREF: _bnm+22|j
; _bnm+4B|j ...
mov ax, 2 ;  置标
retn
...
loc_289: ; CODE XREF: _bnm+A|j
; _bnm+282|j
xor ax, ax ;  清标
retn
_bnm endp

align 4
_text ends
```

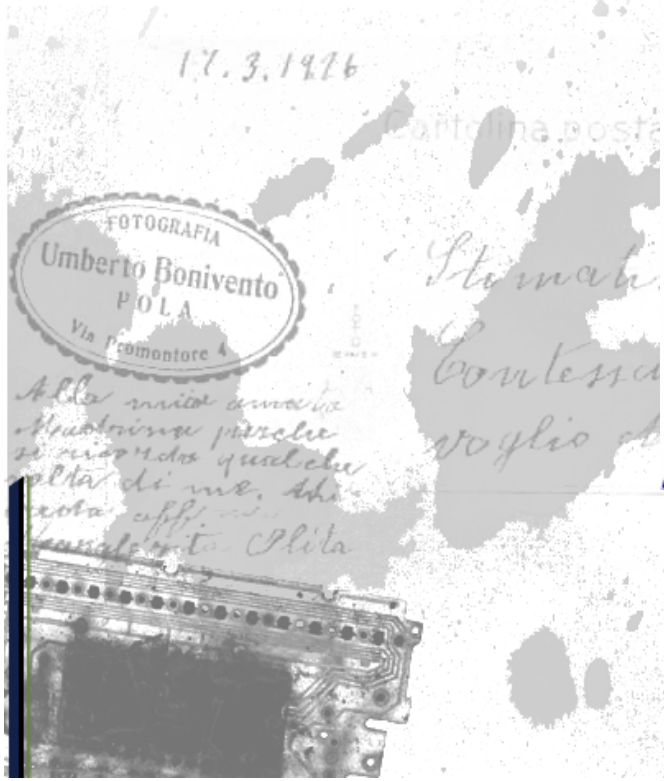
三、样本分析体制的新探索

后端体制


Virus Capture Platform
Control Center
Internet AV Platform
WEB Site
honeypot
Router
Cert-样本预处理
CERT-系统维护员
病毒模拟环境
可管理网
Cert-样本分析员
VX Manager Plat
CERT-安全支持

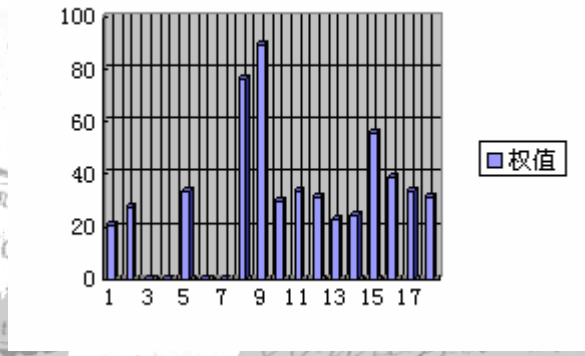
特征码的质量评估

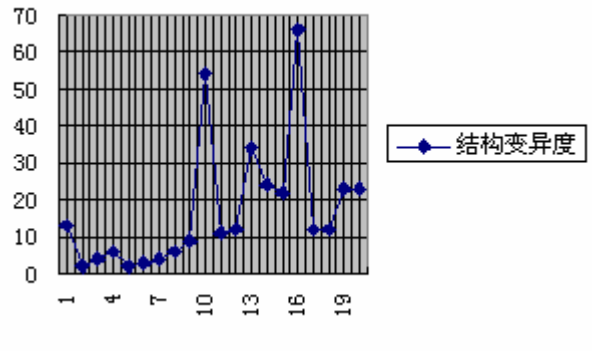




X'CON 2003

特征码的自动挖掘


100
80
60
40
20
0
1 3 5 7 9 11 13 15 17
权值


70
60
50
40
30
20
10
0
1 4 7 10 13 16 19
结构变异度

分析机（专用调试器）

- 保证特征码质量的前提是获取稳定（静态）代码流，而先进病毒检测引擎中的预处理器可以将静态代码流送入查毒缓冲区。
- 分析机由同样的预处理引擎和反汇编器组成。

如果没有预处理器

- UPX1:00407041 push ebp
  UPX1:00407042 mov ebp, esp
  UPX1:00407044 push 0FFFFFFFFh
  UPX1:00407046 push 683350A8h
  UPX1:0040704B xchg eax, esp
  UPX1:0040704C and al, 4
  UPX1:0040704E mov edi, 647BBDFDh
  UPX1:00407053 mov eax, ds:64500000h
  UPX1:00407058 mov ds:10EC8307h, esp
  UPX1:0040705E push eax
  UPX1:0040705F push edi
  UPX1:00407060 mov [ebp-18h], esp
  UPX1:00407063 dd 0FFFCDEFFh, 210415DDh, 0D48AD233h, 69001589h, 0C88B0040h
  UPX1:00407063 dd 89FFE181h, 7568FC0Dh, 0DDFB67Fh, 308E1C1h, 0E8F80ACAh
  UPX1:00407063 dd 7F4A310h, 0F7FDFB6Ah, 5B1394FFh, 75C08559h, 721C6A08h

带有预处理的分析机工作


```
F:\>cd work
F:\work> csa helloupx.exe -f -p

    ----------------------------
    Code Stream analyser V1.0

    CopyRight By Antiy Labs
    ----------------------------
    Shell found UPX Ver 1.24
    unpacking.......
    unpacking ok!

    Start.........

    ------------------
```

```
.text:00401000 68 30604000 PUSH hello.00406030
.text:00401005 E8 06000000 CALL hello.00401010
.text:0040100A 83C4 04 ADD ESP,4
.text:0040100D 33C0 XOR EAX,EAX
.text:0040100F C3 RETN
.text:00401010 53 PUSH EBX
.text:00401011 56 PUSH ESI
.text:00401012 BE 70604000 MOV ESI,hello.00406070
.text:00401017 57 PUSH EDI
.text:00401018 56 PUSH ESI
.text:00401019 E8 4B010000 CALL hello.00401169
.text:0040101E 8BF8 MOV EDI,EAX
.text:00401020 8D4424 18 LEA EAX,DWORD PTR SS:[ESP+18]
.text:00401024 50 PUSH EAX
.text:00401025 FF7424 18 PUSH DWORD PTR SS:[ESP+18]
.text:00401029 56 PUSH ESI
.text:0040102A E8 04020000 CALL hello.00401233
.text:0040102F 56 PUSH ESI
.text:00401030 57 PUSH EDI
.text:00401031 8BD8 MOV EBX,EAX
.text:00401033 E8 BE010000 CALL hello.004011F6
.text:00401038 83C4 18 ADD ESP,18
.text:0040103B 8BC3 MOV EAX,EBX
.text:0040103D 5F POP EDI
.text:0040103E 5E POP ESI
.text:0040103F 5B POP EBX
.text:00401040 C3 RETN
.text:00401041 55 PUSH EBP
```


X'CON 2003
STUDIO FOTOGRAFICO
DANTE MORONI
TORINO
FOTOGRAFIA
Umberto Bonivento
POLA

四、发展现状

X-File

Like VX
- ├ LogoPicture
- ├ AdvWare
- ├ FalseAlarm
- └ Joke

CopyRight
- ├ Cracker
- ├ KeyGen
- └ Simulator

PornWare
- ├ PornWare.Dialer
- ├ PornWare.Downloader
- └ PornWare.Tool

NetTool
- ├ NetTool.Scanner
- ├ NetTool.Sniffer
- ├ NetTool.Spoofer
- └ Other

Tool
- ├ Tool.DOS
- ├ Tool.Win16
- └ Tool.Win32

RiskWare
- ├ RiskWare.Dialer
- ├ RiskWare.FTP
- ├ RiskWare.Monitor
- ├ RiskWare.PSWTool
- ├ RiskWare.Proxy
- ├ RiskWare.RemoteAdmin
- ├ RiskWare.Tool
- ├ RiskWare.mIRC
- ├ RiskWare.IRC
- └ RiskWare.WebServer

新的技术领域

- 更广泛的文件格式判断与解析
- 扇区分析技术
- 网络定性诊断技术

独立检测SDK

AGBE SDK(改造自Antiy Ghostbuster)

Sector Scope SDK

Ghost Info

通过一个简单的接口，使应用程序很容易的具备对目标对象扫描检测的能力。

接口描述

1. int InitFileEngine(const char * pszLibName)

/* 文件引擎初始化(提供数据库名称,返回:true成功,false失败)*/

2. int ProcessFile(const char * pszFileName , char * pszName , int nMaxLength);

/* 文件处理(提供全路径文件名,返回:true发现目标,false未发现；名称存放于pszName)*/

3. int InitSectorEngine(const char * pszLibName)

/* 扇区引擎初始化(提供数据库名称,返回:true成功,false失败)*/

4. int ProcessSector(SectorProviderInterface * spi, char * pszName , int nMaxLength)

/* 扇区处理(提供扇区提供者,返回:true发现目标,,false未发现；名称存放于pszName)*/

5. int InitEncyclopediaEngine(const char * pszLibName)

/* 检测对象信息引擎初始化(提供数据库名称,返回:true成功,false失败)*/

6. int ReadEncyclopedia(const char * pszName , char * pszDescription , int nMaxLength)

/* 检测对象信息(提供对象名称,返回:true成功，false失败)；信息存放于pszDescription)*/

7. int ProcessMemory(DWORD dwPID, char * pszName , int nMaxLength)

/* 处理内存进程(提供PID,返回:true发现目标，false未发现)；名称存放于pszName)*/

Seak@antiy.net

Thanks !

X'CON 2003