Embeddable AntiVirus engine in tiny granularity ---- ------

our understanding and dream

seak

Seak@antiy.net

X'CON 2004

Syllabus

- Challenges to AV dialectics
- Processing in tiny granularity
- Embeddable Engine

XFOCUS TEAM
from internet,for internet

Challenges to AV dialectics

X'CON 2004

AV Dialectics

- AV is not simply technological confrontation.The overall AV system takes in many logical and legal

  factors ,as well as project planning ones,which shares some basic principles in common
- Objectively, these common principles are summarized from the practice, and then applied to guide the design of AV engine and tools
- In 1995 we had summarized the basic common principles in 44 items, informally named AV dialectics.

Some Important Items

- Computer virus is a kind of program in the final analysis
- Feature code of the computer virus

  is the only identifier to classify the viruses
- The crucial criterion of computer virus should be the feature code or some characteristics of the content
- The only reason that feature code should be distilled is being harmful or subjectively harmful
- Whether a certain program should be detected or not should be based on clear criteria

- **The clean up of a virus is the reverse of its infection**
- **User's rights to the AV software：**

  Right to define: Users can customize the function of the AV software besides using the default configuration

  Right to know：Users can know what the AV software has done in the system Right to backup：Users should be provided with means to backup the infected files
- **Detect the virus in the packages, can clean the viruses without deleting the package if the algorithm is authorized**
- **Precaution principle：virus monitoring should prevent the infected files from running (controlling the system)**

AV Dialectics

- Along with the development of both application environment and virus technics，many of our summarizations contradict each other
- The fundamental reason of these contradictions is the complication of information system

Challenge 1：Puzzling Criterion

- item： The crucial criterion of computer virus should be the feature code or some characteristics of the content
- exception： CMD backdoor left by Code Red
- Question： The traditional AV technologies is dealing with “Yes or No” problems， the only criterion is the content of the program.But under some circumstances， the boundary between harmful and harmless becomes vague.

Challenge 2：Paradox of Range

- Item：Whether a certain program should be detected or not should be based on clear criteria
- Exception： psexec tool used in Worm.Dvldr 。
- Question：The emergence of x-file conception is another puzzle in criterion.How far should the AV software reach? What is the criterion on earth? So far, many AV products add in adware detecting , is this reasonable or legal?

Challenge 3：Package Enigma

- Item： Detect the virus in the packages, can clean the viruses without deleting the package if the algorithm is authorized
- Exception：DIY worm(such as password worm)、worms using or saving in zip format(such as some variation of netsky)
- Question：The basic assumption of the traditional AV software is that the package file is normal file, but it may contain virus. DIY worm is a self-extraction package. Some worms made many zipped backup copies on the disk , AV software can not remove them

Challenge 4: Trashy Files

- Item： The only reason that feature code should be distilled is being harmful or subjectively harmful
- Exception： crisis caused by non-official evaluating
- Problem： If one company detects some trivial files, the other companies will just follow because of the competition in winning higher marks in the evaluating.However is that worthwhile? How to balance the efficiency and quantity of virus detection?

Challenge 5: Responsibility problem

- Item： The clean up of a virus is the reverse of its infection
- Case： backdoor left/The worm returns
- Question： Is AV software responsible for recovering all the modification of the system made by the virus? And how to deal with the leaks?Is this work endless?

Challenge 6：The time of action

- Item：Precaution principle：virus monitoring should prevent the infected files from running (controlling the system)
- Case：Arguments on the file evaluation
- Question：Since it is difficult to detect unknown PE viruses 、 trojans or backdoors，should the lag report based on behavior judgement be promoted?

Challenge 7：Active Protection

- Item：User’s rights to the AV software…
- Case：scanning worms changed the image of victims
- Question：Viruses has changes from “infected from whom” to “infecting whom” ,then should some department remove such viruses without the user’s permission? What means are acceptable? Technological question or legal question?

Putting forward and solving the problems

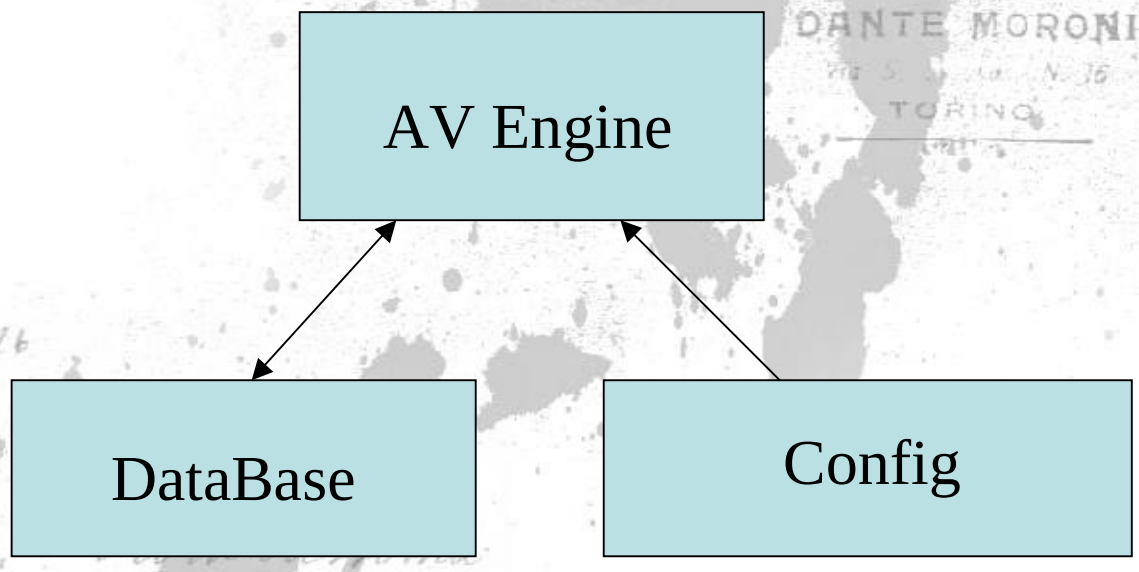
- None of them is difficult to be solved technologically
- Some of them are concerning with style and moral ,however “Puzzling Criterion”、“Package Enigma”、“Responsibility Problem ” are reaction to the traditional traditional system and framework of the AV engine.
- We need more adaptive and reasonable engine framework instead of expediency in programming

XFOCUS TEAM
from internet,for internet

Processing in Tiny Granularity

X'CON 2004

Three elements of Engine



Three elements of Engine

The three elements of engine are: Engine、Datebase、Configuation，the Engine relies on Database to detect, and the definitions in the Configuration to work.

Before, we put much emphasis on the Engine, now, we need to pay more attention to the configuration to see what it can offer for us.

Also to re-evaluate Database which is supposed to be labor work, to see whether the potential of creativity still exists.

The Traditional Database

	Type 1	Type 2	Type 3	Type 4
Number	√	√	√	√
Mod num	√	√	√	√
Virus name	√	√	√	√
First word of Feature code			√	√
Offset1+Sign 1			√	√
Offset2+Sign 2			√	√
File type flag				√
Process arg	√		√	√
Processing module name			√	√

The Traditional Database

- Working based on the Database, detecting 95% of viruses via records on type 3 and type 4 (feature code detecting).Detecting the rest 5% special viruses via type1 and type2 （independent module detecting）。
- Processing over 80% viruses via argument，the rest 20% via processing module。

Basic characteristics of the traditional Database

- Object Control：what to detect
- Behavior Control：how to process
- Effectiveness Control：intensity of detecting

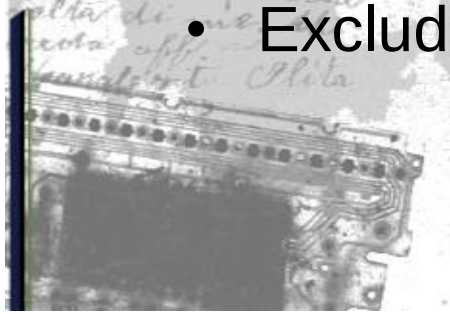
Traditional Configuration Means

- Flow control(Program)
- Debug Switch(Developer)
- INI control(User)

X'con 2004

Object Control

- Memory=Yes; check the mm
- Sectors=Yes ; check the booting sector
- Files=Yes ; check file system
- Packed=Yes ; check packages
- Archives=Yes ; check archives
- MailBases=Yes ; check mails
- MailPlain=Yes ; check coded files
- FileMask=2 ; check the extended names
- UserMask= ? ; user defined extension
- Exclude=No ; not to check customized extensions
- ExcludeMask= ; not to check definition of extensions



Behavior Control

- InfectedAction=0 ; remove viruses
- InfectedCopy=No ; back up viruses
- InfectedFolder=Infected ; back up folders
- SuspiciousCopy=No ; back up suspicious files
- SuspiciousFolder=Suspicious ; back up folders
- Report=Yes ; generate logs
- ReportFileName=Report.txt ; name of log file

Effectiveness Control

- Warnings=Yes ; Open the warning
- CodeAnalyzer=Yes ; Open the code analyzer
- RedundantScan=Yes ; Open Redundant scanning

That's enough?

- In the traditional AV environment, it is enough to control in such granularity, however problems occur when it comes to the more complicated environment.

Application Case 1

- Consider what different features will the engine have when working as AV software for a single computer or one module in the mail server?
- I-Worm.Nimda.e is a infecting worm, when processed locally，it should be regarded as a PE infected file，but for a mail server, it should be simply discarded。
- Win95.CIH is a infecting virus, when detected, whether it is local or on mail server, it should be processed as an infected virus and the original file should be recovered.
- The essential difference is that Win95.CIH does't mail itself, if it occurs on the Internet, it should be an executable program mailed by the user, while Nimda behaves contrarily。
- This situation requires different processes for different kinds of viruses in various environments, it is beyond the capacity of traditional engine control.

Application Case 2

- One network viruse detecting equipment contains several responding modules
- What policy should these responding modules work with ?
- Some mail worms create addressers randomly,what will happen if sending feedbacks?

Some mail worms use bot to create addressers,what will happen if sending supplemental notices?

- Message Noticing
- Supplemental mail Noticing
- Feeding back mail Noticing
- reset connection…

Requirements from Engine User

	Smtp detecting				POP3 detecting			
	Faked Receiver	**Not Faked Receiver**	**True sender**	**False sender**	**Faked receiver**	**Not Faked receiver**	**True sender**	**False sender**
Feeding back noticing			**effective**	**ineffective**			**effective**	**ineffective**
Supplemental noticing	**ineffective**	**effective**			**effective**	**ineffective**		

Application Case 3

- Response is an effective method for bypass equipment responding.
- OPSEC、TOPSEC

- Different processing for scanning worm or mail worm.
- Is there a proxy server in the network?

That's not enough!

- New demand goes beyond the capacity of the traditional engine
- How to solve the problem?

XFOCUS TEAM
from internet,for internet

Embeddable AV engine in tiny granularity

X'CON 2004

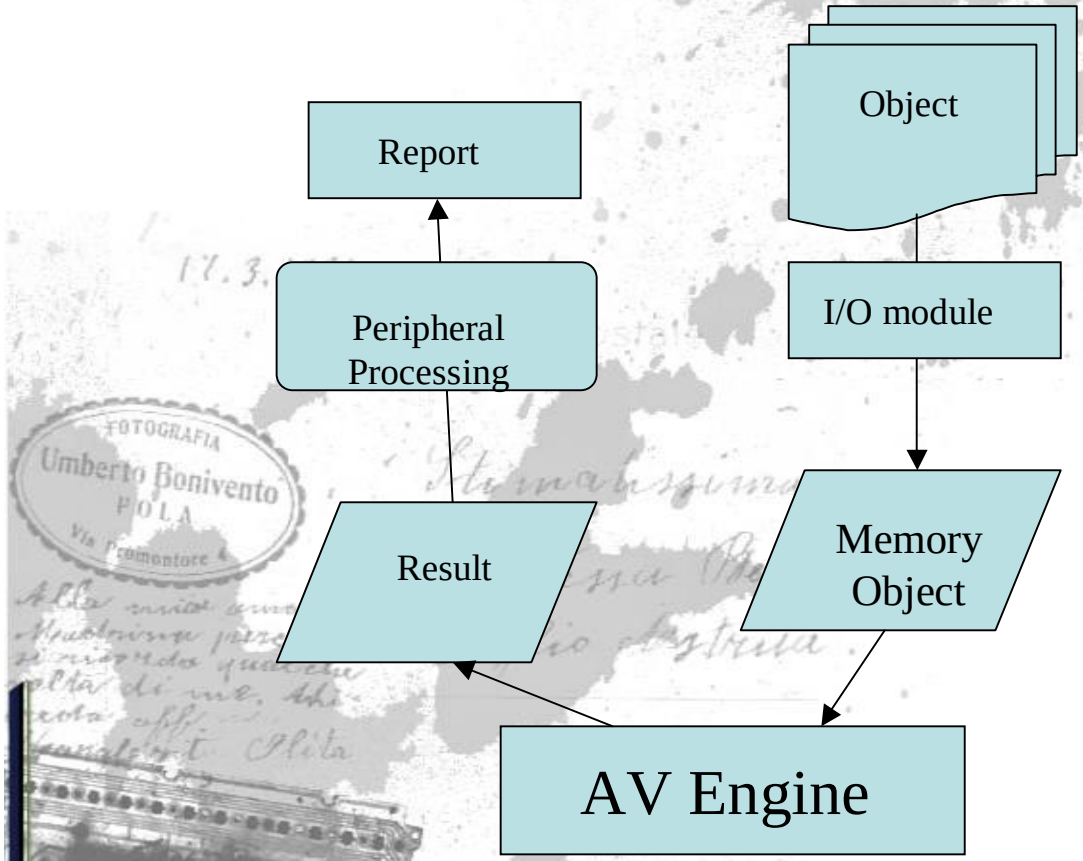
Putting Forward the Question

- The downward moving of the network security level implies the virus filter mechanism will extend to equipments of different levels
- Above discussion prepares for AV engine adapting to more complicated environments
- Embedded equipment or AV engine in other environments are designed for tiny granularity

Application of Embedded Engine

Application Form	Details
AV module in Firewall	Construct linear velocity based virus filtering module for package filtering firewall by network engine. Construct file stream based virus filter for application proxy, transparent proxy or stream filtering firewall by file engine.
AV module in router	Add virus filtering ability to routing equipment by high speed package level scanning
AV module in switcher	Add virus filtering ability to exchange/share equipment by high speed packet scanning.
Virus detecting plug-ins in IDS	Extend the Network engine to provide the IDS with network virus detecting ability
AV module in GAP	Extend the GAP equipment with virus filtering ability
Virus protecting in mail system	Embed virus detecting ability into mail server
Independent AV software	User need only to program the peripheral interface to develop their own AV software

Basic Requirement 1——Memory Engine



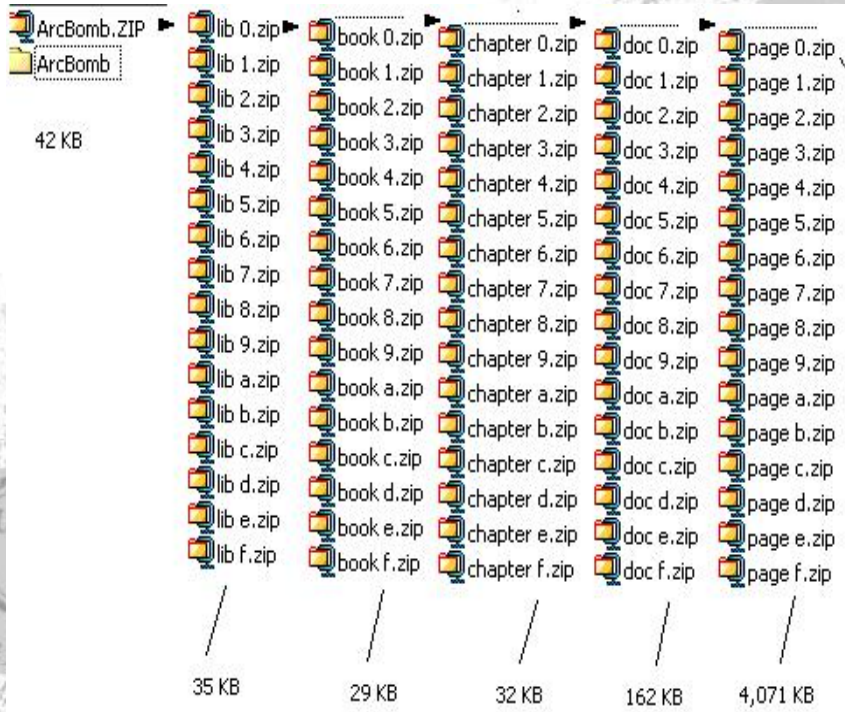
Memory Engine Interface Demo

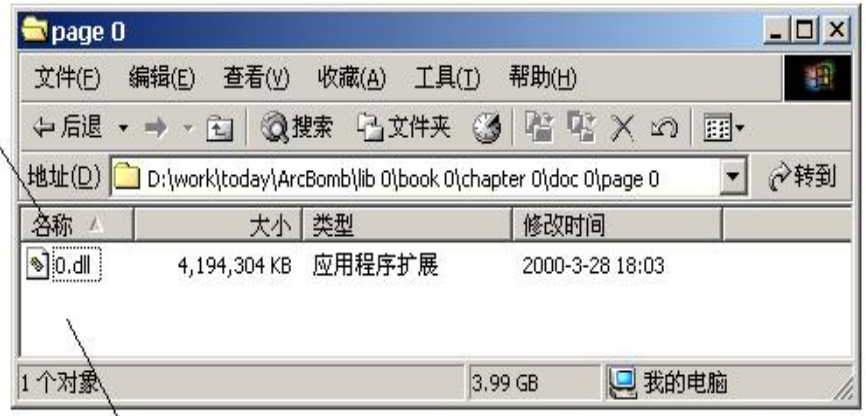
```
/*scanning parameter structure */
typedef struct _AVLF_SDK_SCAN_PARA
{
char * pBuffer;                 /* pointer  to buffer */
unsigned long ulSize;           /*size of the buffer */
const char * pDescription;      /* description information */
int   bUnpack;                  /* whether to unpack*/
int  bKill;                     /* whether to  kill the virus */
int  bKilled;                   /* whether succeed to kill the virus */
} AVLF_SDK_SCAN_PARA,*PAVLF_SDK_SCAN_PARA;

/* set the receiver */
AVLEACHSDK_API int AVLF_SDK_SetReciver(IReportReciver *pReciver);

/*scanning：return 0 if no virus detected，return 1 if virus found，detailed information is received by
    the receiver class*/

AVLEACHSDK_API int AVLF_SDK_Scan(PAVLF_SDK_SCAN_PARA pParamter);
```

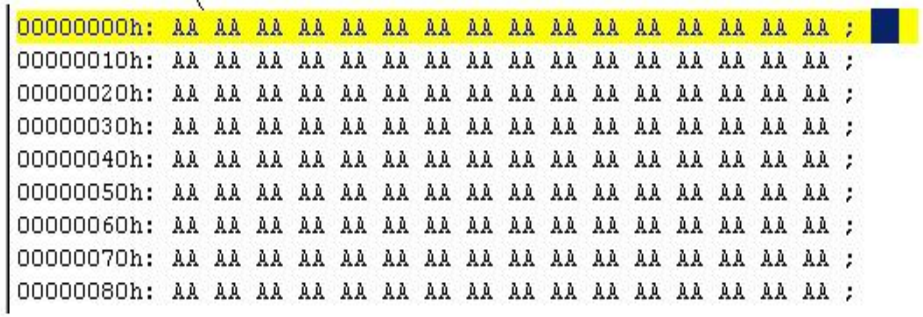
Basic Requirement 2——Recursive Engine

- Modern AV engine have evolved from branched engine leaded by modules format recognition to recursive engine
- In recursive engine, scanned object could bear multiple flags, which can be detected by corresponding modules
- Mcafee's bug in detecting SFX
- archbomb.zip

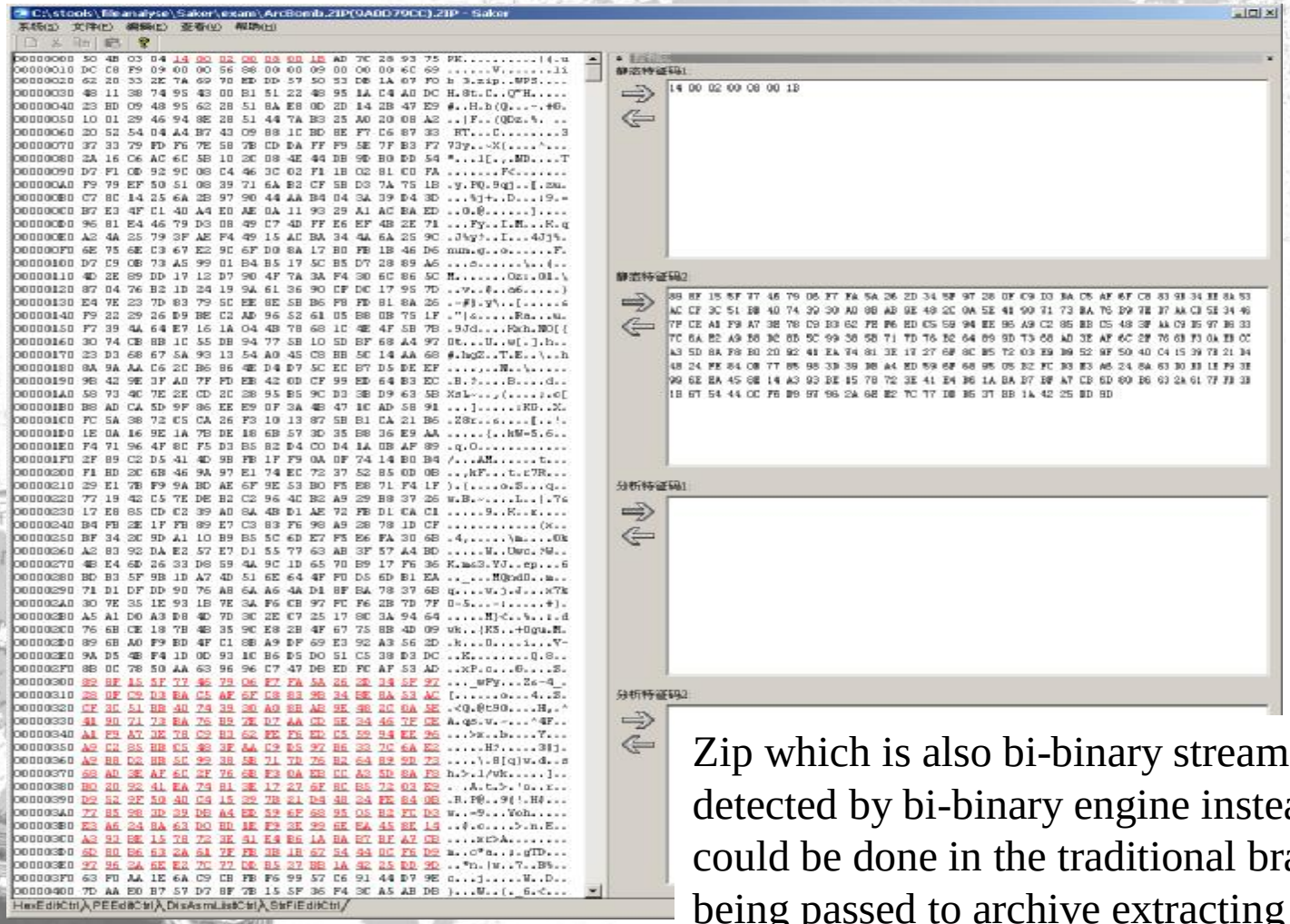
Archbomb.zip DoS AVware







How to detect archbomb.zip



sign1

Offset: 4h

Length: 7h

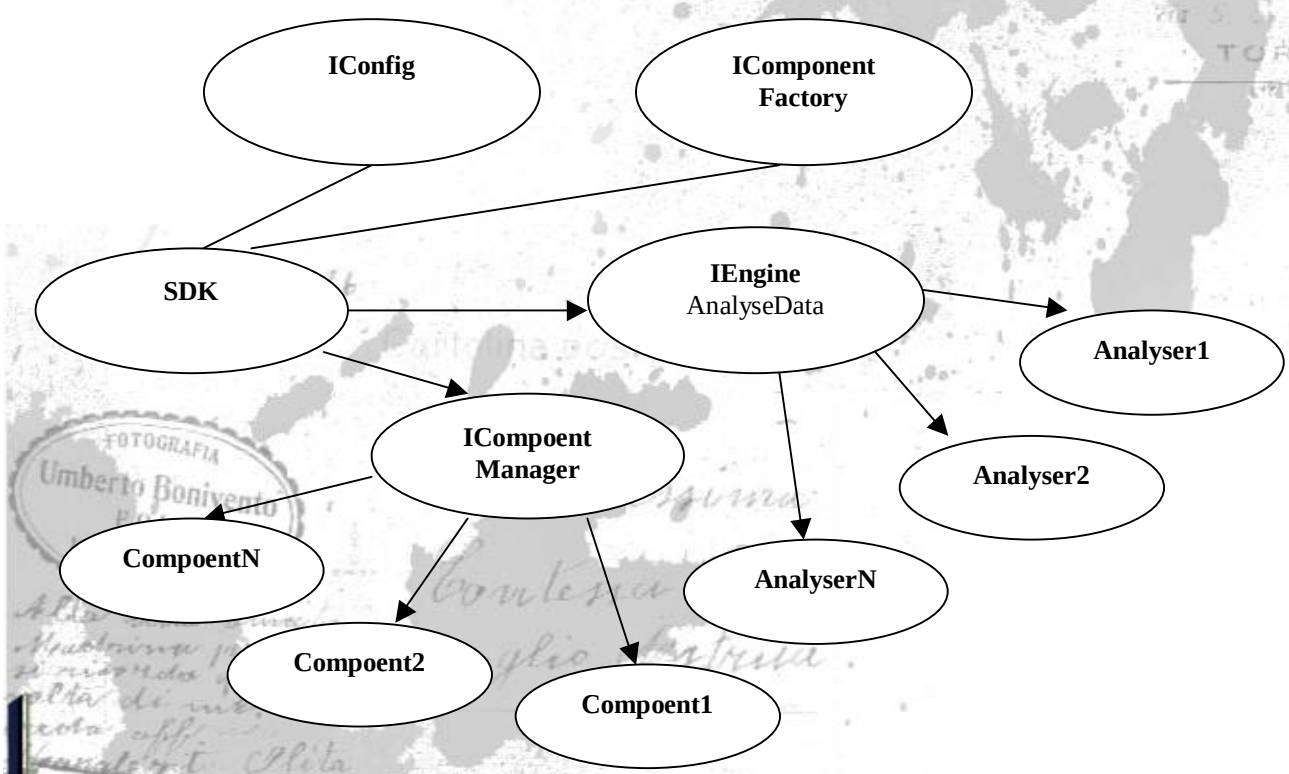
Sign 2

Offset:300h

Length:F0h

Zip which is also bi-binary stream could be detected by bi-binary engine instead of what could be done in the traditional branched engine: being passed to archive extracting module by format recognition module.

Eample——Recursive Engine



1、Analyzers are parallel in structure, no one is pre-required.

2、Results from the analyzers differ in priority rating, detecting viruses listed as the highest while needing further pre-process as the lowest

3、In principle, analyzers work serially, whose results have higher priority being forwarded.

Basic Requirement 3——portable engine

- Working environment could be x86 architecture , or others like PPC
- Those modules written in x86 assembly language become barrier for porting.

Basic Requirement 4——Highly Controllable Engine

- What is the essential requirements in tiny granularity?
- Virus processing in different environments does not only rely on infecting feature but also the “speciality” of the virus.
- The granularity of control needs to reach the individual virus, the Database needs to provide more information.
- Branching is not controlled by speciality instead of program.

Control Level

- Flow control(Program)
- Debug Switch(Developer)
- INI control(User)

- Flow control(Program)
- Virus attribute
- Debug Switch(Developer)
- Stencil(Condition)
- INI control(User)

Traditional Database V.S. Tiny Granularity Database

```
struct vxdb
{
char name[255];
char fword[4];
char offset1[4];
char crc1[8];
char offset2[4];
char crc2[8];
…
};
```

```
struct tgvxdb
{
char name[255];
char fword[4];
char offset1[4];
char crc1[8];
char offset2[4];
char crc2[8];
…
int vxattribute ;
};
```

Basic Requirement 3——Precisely Processing Engine

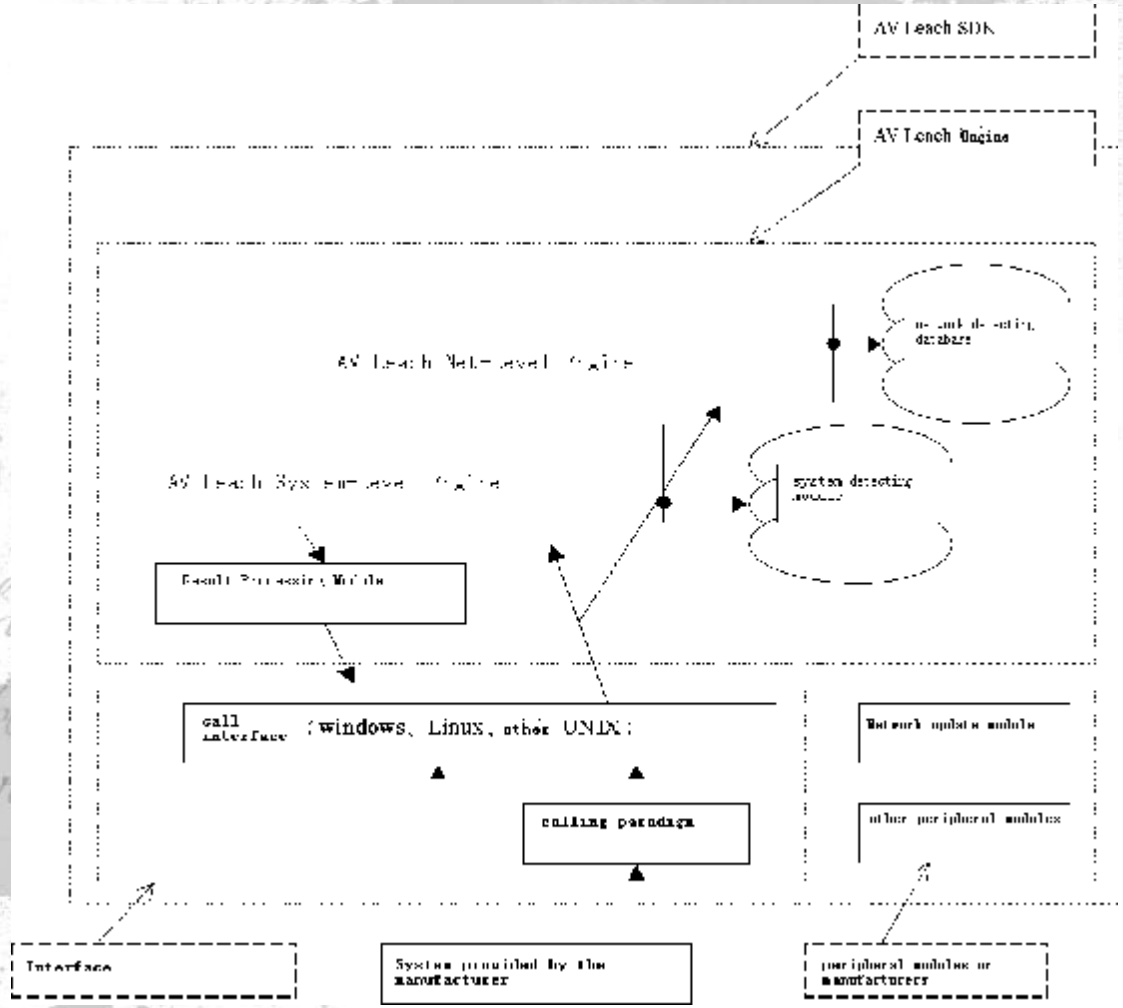
- Perfect reverse process is a dream。
- Tiny Granularity Engine ends the era in which the AV company needs not to “analyze” the virus.

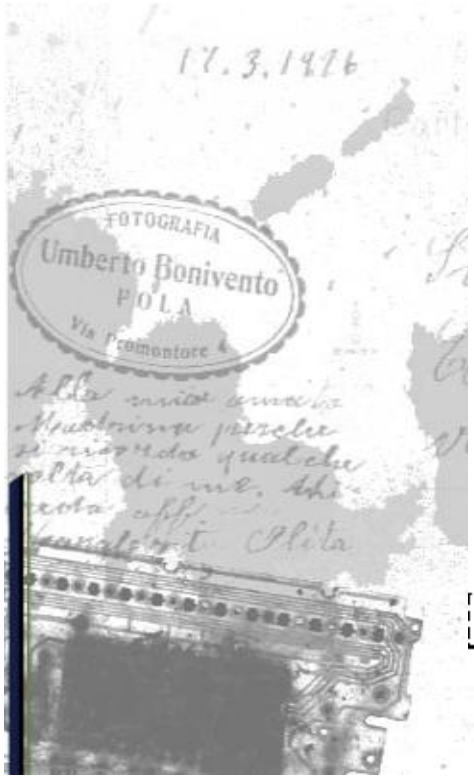
How to Process

- Clean com tail
- Clean com head
- Clean exe tail
- Clean ne tail
- Clean pe tail
- Remove file
- Copy data block
- Move data block
- Insert data block
- Modify data block
- Delete data block
- Fill in data block
- Truncate data tail
- Truncate data head

- Left side is the cleaning parameters set which is widely accepted by many companies, on the contrary, we have not been patient to non-infecting viruses before.
- We need the same detailed processing script
- Is this work endless?

Our Model





Virus Analyzer Machine Demo

Our Understanding and Dream

- AV dialectics is not invariable,instead, it is the evolving dynamic principle.It requires not only summarizing but also supplementing and breaking through.
- We believe in our understanding , we persist on our dream.
- Thank xfocus!
- Thank you!