Research on Same Source Feature Measuring Technology of Software

Liu Xin, Hu Song

Computer Department,Peking University

Topic

- **Research Background of Software Same Source Feature Measuring Problem**
- **Summary of Current Software Same Source Feature Measuring Technology**
- **Same Source Feature Measuring Technology Based on Character Definition of Executable Code**
- **Implementation of Software Same Source Feature Measuring System**

Research Background of Software Same Source Feature Measuring Problem

- To determine if one software is distortion of another software or the software is developed by the same author, by which to establish the relationship between the suspect and the case.

- Software Right Protection: Thought, process, operation method and mathematics concept are not included in software right protection.

The Current Same Source Feature Measuring Technology of Software

- Same Source Feature Analysis on Source Code Level
  - Based on comparability of Text
  - Based on Programming Style Analysis

The Current Same Source Feature Measuring Technology of Software

- **Same Source Feature Measuring Technology of Source Code Level based on Comparability of Text**

Which aims at solving partial replication problem of text, namely to determine if there exist a edit serial by which code segment A can be transformed into code segment B. (Entire copy is few used in plagiarism)

The Current Same Source Feature Measuring Technology of Software

üSame Source Feature Measuring Technology of Source Code Level based on Comparability of Text

1. Full Copy
2. Notation Change
3. Blank Change and Reedit
4. Identifier Rename
5. Code Segment Reordering
6. Sentence Sequence Change in Code Segment
7. Operator Sequence Change in Expression
8. Data Type Change
9. Redundancy Sentence and Variable Increment
10. Equivalent Control Structure Replacement

Plagiarism Switch List

The Current Same Source Feature Measuring Technology of Software

- Same Source Feature Measuring Technology of Source Code Level based on Comparability of Text

- Method based on substring match: Which is brought forward by H.T. Jankowitz.

To find the similarity of codes using Karp-Rabin algorithm based on fast substring match. The main point of the method is to select some character string named fingerprint, then map the fingerprint to Hash Table with one fingerprint corresponding one figure. The number or ratio of the same fingerprint can weigh the similarity of text.

The Current Same Source Feature Measuring Technology of Software

ü Method based on substring match:

There are numerous decision-making functions to compute the text similarity. The followings are the two simplest functions:

Let $F(A)$ is the fingerprint Set of document A, $F(B)$ is the fingerprint Set of document B and $S(A,B)$ is the similarity degree, then the first kind of decision-making function is $S_1(A,B) = \frac{F(A) \text{ I } F(B)}{F(A) \text{ Y } F(B)}$

and the second kind of decision-making function is

$$S_2(A,B) = F(A) \text{ I } F(B)$$

Apparently S(A， B) = S(B， A) is guaranteed in the two functions。

The Current Same Source Feature Measuring Technology of Software

- **Same Source Feature Measuring Technology of Source Code Level based on Comparability of Text：**

Method Based on Parameterized Match：

Which is presented by Brenda S. Baker and solve the problem that identifier name replacement make full match invalidate well.

The Current Same Source Feature Measuring Technology of Software

ü Same Source Feature Measuring Technology of Source Code Level based on Comparability of Text：

l Word Frequency Statistics Method： which is oriented of vector space model of information search technology. In the method the appearance number is counted first, then the word frequency construct the feature vector of the document. Doc matrix, cosine or other similar computing method is utilized to compute the feature vectors of the two documents to measure the similarity of the documents.

The Current Same Source Feature Measuring Technology of Software

- **Same Source Feature Measuring Technology based on Programming Style Analysis of Source Code Level:**

Which is also named software forensics. The methods aim at determining if the documents own the same author, so the character serial of codes reviewed can be fully different,even which can be written for different function.

Analysis based on code style can discover some situation of text replication too.

The Current Same Source Feature Measuring Technology of Software

ü Same Source Feature Measuring Technology based on Programming Style Analysis of Source Code Level:

Programming Style Classification include

- **Retract Style**
- **Code Style**
- **Program Style**

Same source feature measurement based on programming style analysis need plenty of programming experience and a mass of manual work.Quiet a number of criterion of it ,such as coherence determination of notation and code and software quality, is difficult to be measured, so the auto-analysis by computer is inconvenient.

The Current Same Source Feature Measuring Technology of Software

Same Source Feature Measuring Technology of Executable Code Level

- Dynamic Same Source Feature Analysis

  1. **Dynamic analysis based on System Transfer**

  2. **Dynamic analysis based on program exterior** action

- Static State Same Source Feature Analysis:

The Current Same Source Feature Measuring Technology of Software

Same Source Feature Measuring Technology of Executable Code Level

üDynamic Same Source Feature Analysis

Dynamic analysis based on System Transfer： which is usually used in IDS. Different system transfer analysis is used in different type of IDS. When adopting Misuse Detection technology, IDS distill intrusion feature form the transfer serial of the intrusion program. When Anomaly Detection Technology is used, IDS establishes model for normal user action and when distinction between user program action and the model exceed the threshold an intrusion is alarmed.

Dynamic Analysis based on system transfer analysis: a method to distinguish the UNIX program by dynamic analysis of system transfer (Stephanie Forrest) :

1。 To achieve the running feature database of the program by tracing the program executing process. To establish the program running feature database, the author design a moving window sized k+1 to scan the system transfer record of the whole program and record the occurrence of a system transfer before another system transfer.

For example, select k=3 and get the feature database for the following system

Open，read， mmap， mmap， open， getrlitmit， mmap，close

Let the moving window scans from the first operator open, read is record as the its second transfer and mmap is the third and the fourth. Then to record the second and third transfer after recording operator read by the same way. There is the following transfer relationship fig.

call	position 1	position 2	position 3
open	read, getrlimit	mmap	mmap, close
read	mmap	mmap	open
mmap	mmap, open, close	open, getrlimit	getrlimit, mmap
getrlimit	mmap	close	
close			

System transfer feature database of a UNIX program

Dynamic Analysis based on system transfer analysis: a method to distinguish the UNIX program by dynamic analysis of system transfer (Stephanie Forrest) :

2. After getting the system transfer feature database of a program ,to another transfer serial of another program

Open， read， mmap ， open ， open ， getrlitmit ， mmap ， close

The following method is used to compute the distinct between the two program:

To computer the unmatched number: there are 4 times of unmatch.

The third transfer after open in the feature database can not be open.

The second transfer after read in the feature database can not be open.

The first transfer after open in the feature database can not be open.

The second transfer after open in the feature databse is not getrlimit

Dividing the number of all possible unmatching with practical unmatching number, the biggest unmatched number to system transfer serial sized L and when the window size is k is:

$$k(L-k)+(k-1)+(k-2)+...+1=k(L-(k+1)/2)$$

In the example L=8 ， k=3，so the biggest unmatched number is 18. The unmatched ratio of the two system transfer serial is 4/18 = 22%。

The Current Same Source Feature Measuring Technology of Software

q Same Source Feature Measuring Technology of Executable Code Level

ü Dynamic Same Source Feature Analysis

2. Dynamic analysis based on program exterior action

: which is usually used in IDS. Different system transfer analysis is used in different type of IDS. When adopting Misuse Detection technology, IDS distill intrusion feature form the transfer serial of the intrusion program. When Anomaly Detection Technology is used, IDS establishes model for normal user action and when distinction between user program action and the model exceed the threshold an intrusion is alarmed.

The Current Same Source Feature Measuring Technology of Software

2Dynamic analysis based on program exterior action： Which is widely applied in antivirus real time monitor. For example, Norton Antivirus of symantec cord discovers the suspected program by monitoring the file access of application program at real time.

The exterior action include the following three types mainly

Ø file system operation： to read, write,amend and delete some important files or system files

Ø database operation： to inquire, append, amend and delete items of local or remote database

Ø Network Operation:to listen to port,connect or remote computer or receive and send data package

The Current Same Source Feature Measuring Technology of Software

- **Same Source Feature Measuring Technology of Executable Code Level**

ü **Static Same Source Feature Analysis:** **The widest application of the method is antivirus engine. The earlier file virus usually writes a segment of feature code in the fixed location of the affected files, so antiverus engine can find virus by searching the feature code segment. Fast string match method is adopted usually, which is similar with the technology of static analysis based on text comparability of source code level.**

The Current Same Source Feature Measuring Technology of Software

Static Same Source Feature Analysis of executable code level:

1. **Feature code technology**
2. **Extensive feature scan technology**
3. **Static analysis technology based on structure**
4. **Static analysis technology based on system transfer**

Same Source Feature Measuring Technology Based on Feature Definition of Executable Code

Same Source Feature Measuring Technology: a kind of computer forensic technology in software right protection

Background:

National ten-five science and technology program
"electronic evidence identifier technology research"

National 863 program **"Vulnerability Detection Technology based on attack and forensics of information system"**

Same Source Feature Measuring Technology

Based on Feature Definition of Executable Code

Ⅰ Same Source Feature Measurement based on static feature of code

- ØInstruction statistics compare
- ØKey code transfer compare
- ØIdentity compare based on the concept of equivalent code

Ⅰ Same Source Feature Measurement based on Static feature of code

Same Source Feature Measuring Technology

Based on Feature Definition of Executable Code

Vector computing definition

Let $v_1=(x_1,x_2,\cdots,x_n)$, $v_2=(y_1,y_2,\cdots,y_n)$ are two n-dimension vector, then $\|v_1\|=\sqrt{x_1^2+x_2^2+\cdots+x_n^2}$

$v_2-v_1=(y_1-x_1,y_2-x_2,..,y_n-x_n)$. $|v_1,v_2|=\|v_2-v_1\|/(\|v_1\|+\|v_2\|)$ is the distance of the two vector.

Same Source Feature Measuring Technology

Based on Feature Definition of Executable Code

Ⅰ Same Source Feature Measurement based on static feature of code:

ØInstruction statistics compare

Let $M_1, M_2, \cdots, M_n$ are the assembler instruction counted. For $1 \le i \le n$, let $x_i = \|(x_{i1}, x_{i2}, \cdots, x_{im})\|$ be the excursion of M_i in Executable Code S_1, and the vector $v_1 = (x_1, x_2, \cdots, x_n)$ is excursion vector of the instructions $M_1, M_2, \cdots, M_n$ in executable code S_2. Let the excursion vector of the n assembler instructions in the executable code is $v_2 = (y_1, y_2, \cdots, y_n)$。 Then distance between vector v_1 and v_2 is

$$|v_1, v_2| = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + \cdots + (y_n - x_n)^2} \Big/ \left(\sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} + \sqrt{y_1^2 + y_2^2 + \cdots + y_n^2}\right)$$

Same Source Feature Measuring Technology Based on Feature Definition of Executable Code

Ⅰ Same Source Feature Measurement based on static feature of code:

ØKey code transfer compare

Here we explain the method through an example of program in Windows system.

At first the two executable codes S_1 and S_2 are analyzed statically and the system DLLs and system functions in the DLLs are distrilled. Let the system functions are $f_1, f_2, \cdots\cdots, f_n$ Then the transfer locations are statistically analyzed. The transfer location is divided into K zones. For $1 \leqq j \leqq K$ and $1 \leqq i \leqq 2$, let the excursion vector of system functions $f_1, f_2, \cdots\cdots, f_n$ in the jth zone in program S_i is $v_{i,j}$, then the distance of program S_1, S_2 based on the key code transfer is

$$|S_1, S_2|_{kc} = \sqrt{\sum_{j=1,\cdots,k} |v_{1,j}, v_{2,j}|^2} \Big/ \sum_{j=1,\cdots,k} |v_{1,j}, v_{2,j}|$$

The distance can be used to weigh the similar degree between program S_1, S_2 based on the key code transfer.

Same Source Feature Measuring Technology

Based on Feature Definition of Executable Code

Same Source Feature Measurement based on static feature of code:

- Identity compare based on the concept of equivalent code

Definition If given the same input serial the same output serial can be generated, the two programs are named equivalent code.

The generating engine of equivalent code:

- Program evolvement
- Auto-distortion.

Same Source Feature Measuring Technology

Based on Feature Definition of Executable Code

Ⅰ Same Source Feature Measurement based on static feature of code:

ØIdentity compare based on the concept of equivalent code

Typical program evolvement technology include

（1）Equivalent instruction replacement
（2）Equivalent instruction serial replacement
（3）Instruction reordering
（4）Variable replacement
（5）Increment or deleting jump instruction
（6）Increment of deleting transfer instruction
（7）Insert garbage instruction
（8）Instruction encryption

Same Source Feature Measuring Technology

Based on Feature Definition of Executable Code

Same Source Feature Measurement based on static feature of code：

Identity compare based on the concept of equivalent code

Supposed the current system include the instruction set S= {$I_1, I_2, \cdots, I_n$} and $S_1, S_2, \cdots, S_n$ are all subset of S. The mapping relationship $I_i \to S_i$ is given, which indicates instruction I_i is equivalent with any $\forall I \subset S_i$. Define right multiplication as

$$(Ij \to Sj) * Ii = \begin{cases} Sj & 当i=j时 \\ Ii & 当i \neq j时 \end{cases}$$

and matrix multiplication

$$\begin{pmatrix} I_1 \to S_1 \\ \vdots \\ I_n \to S_n \end{pmatrix} (A1 \cdots\cdots Am) = \begin{pmatrix} B_{11} & \cdots & B_{1m} \\ \vdots & \ddots & \vdots \\ B_{n1} & \cdots & B_{nm} \end{pmatrix}$$

Same Source Feature Measuring Technology Based on Feature Definition of Executable Code

Same Source Feature Measurement based on dynamic feature of code：

For the monitor result for the two software to local disk,register chart and network, we present the following definition

- If the same operate to the same key value is executed, then the distance to the item between the two software is 1
- If the vary operate to the same key value is executed, then the distance to the item between the two software is 0.5
- If only one software operate the key value, then the distinction to the item between the two software is 0

To software's operation to disk and network the similar definition is presented.After one time of virtual running, if the sum of all the distance is D and the sum of all the disk, network and register chart operation is N, then the distance of the two software on the aspect of dynamic feature is D/N, which can weigh the similar degree of the two software about the dynamic feature.

Same Source Feature Measuring Technology

Based on Feature Definition of Executable Code

Let the two executable codes are S_1,S_2 and A is the matrix of instruction equivalence transfer:

if the digital summary of S_1 = the digital summary of S_2

then S_1 and S_2 are the same;

else if $A^*S_1 \supset S_2$

then S_1 and ,S_2 are the same;

else

{

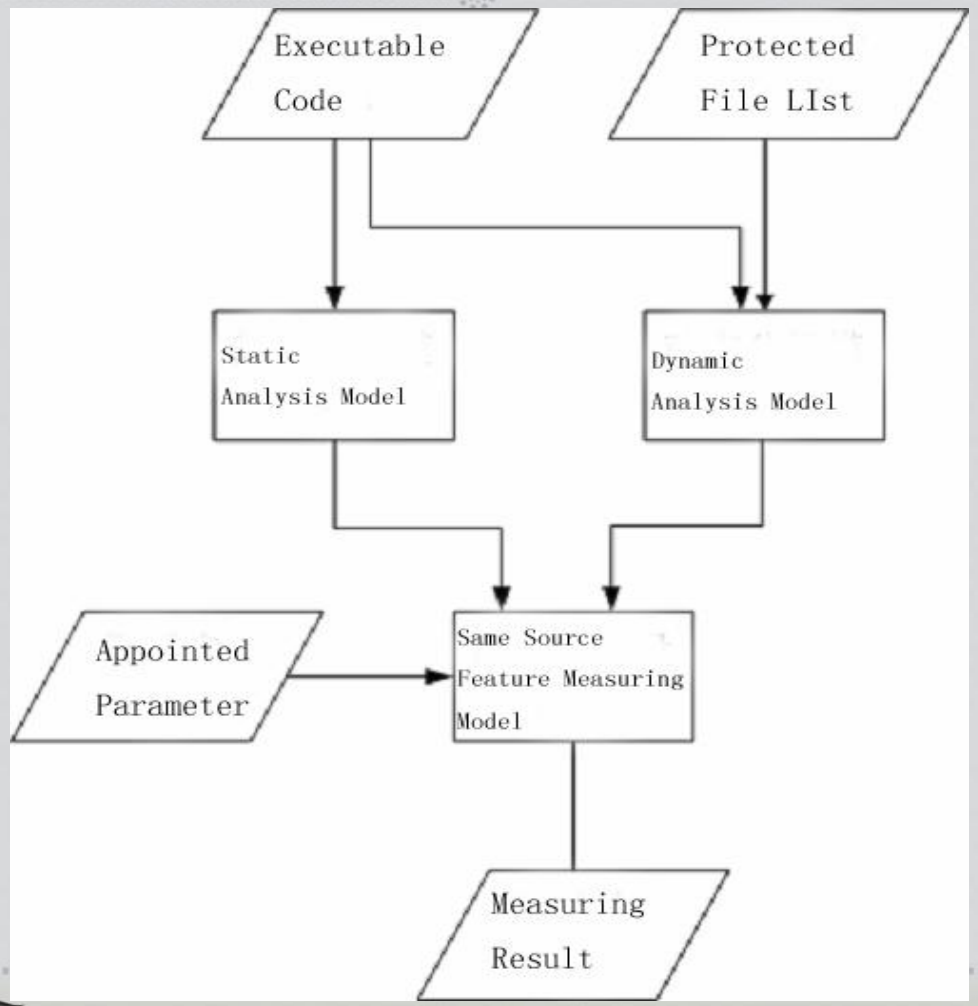
compute the distance between S_1 and S_2 based on instruction statistics, key code transfer and dynamic feature and let the three value are $|S_1, S_2|_s$, $|S_1, S_2|_{kc}$, $|S_1, S_2|_{df}$;

$$D = (1-\beta)*((1-\alpha)|S_1, S_2|_s + \alpha|S_1, S_2|_{kc}) + \beta|S_1, S_2|_{df} \quad (4)$$

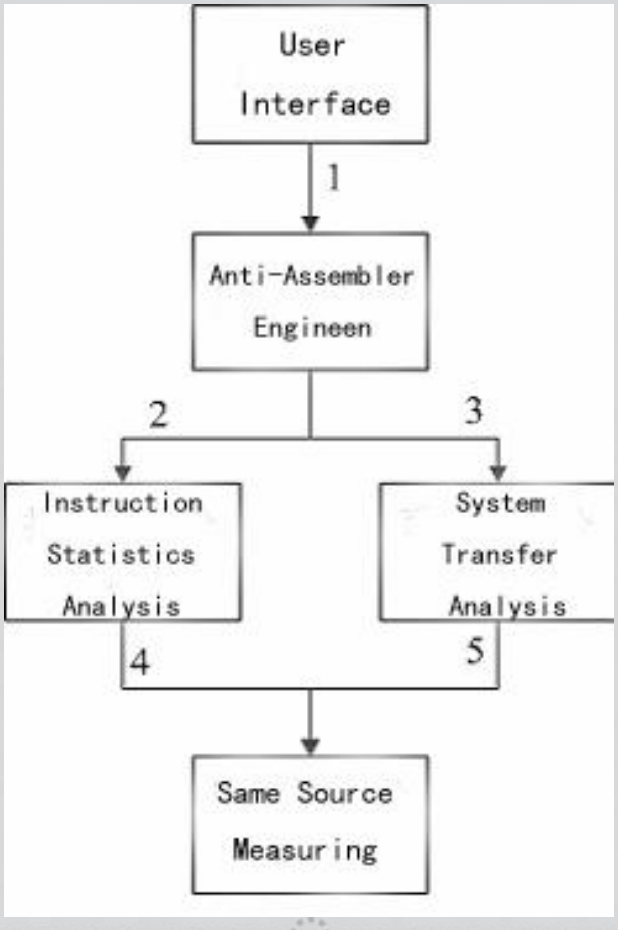
}

$0 \le \alpha, \beta \le 1$, the value of them can be computed from numerical executable code analysis experiment. D can be used to measure the same source feature of S_1 and S_2. We can say S_1 and S_2 have the same source at the ratio of D.

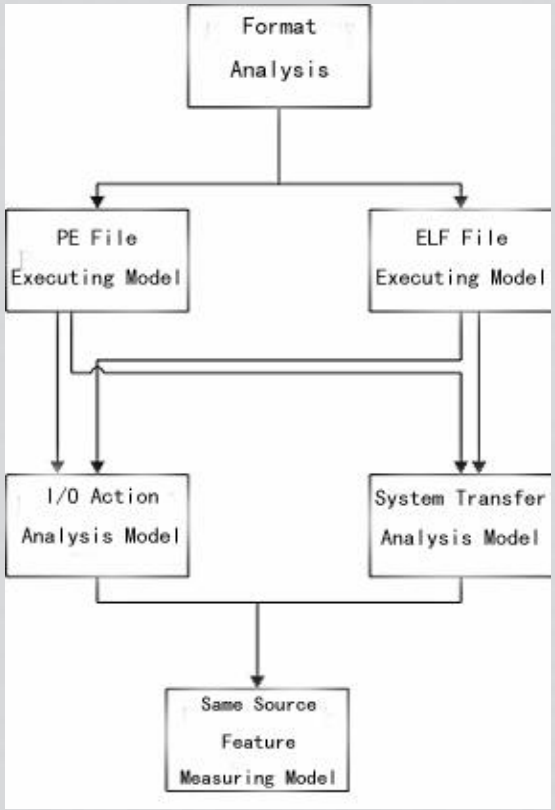
Structure of Executable Code Same Source Feature Measurement system



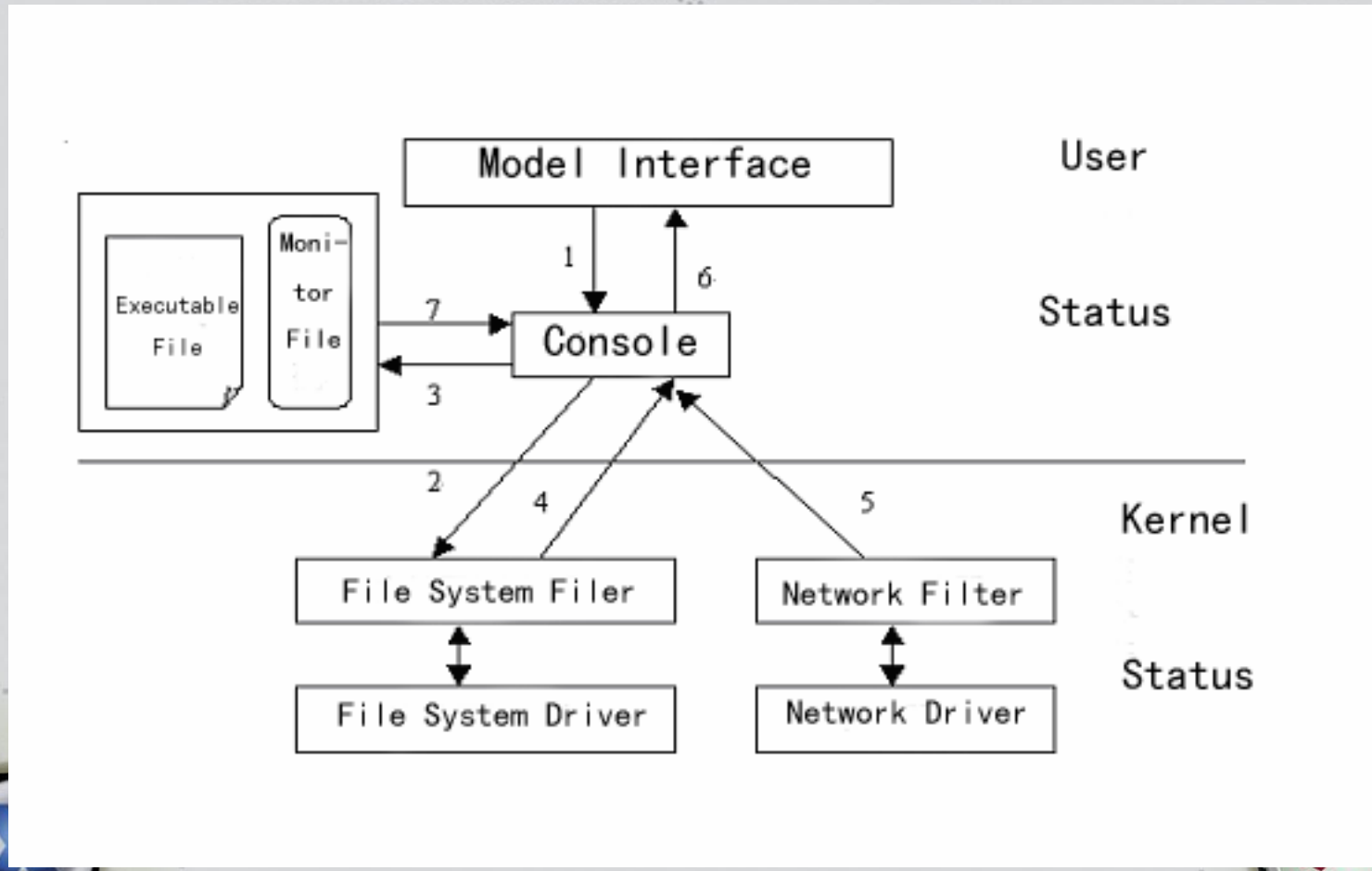
Operating Flow of Static Analysis Model



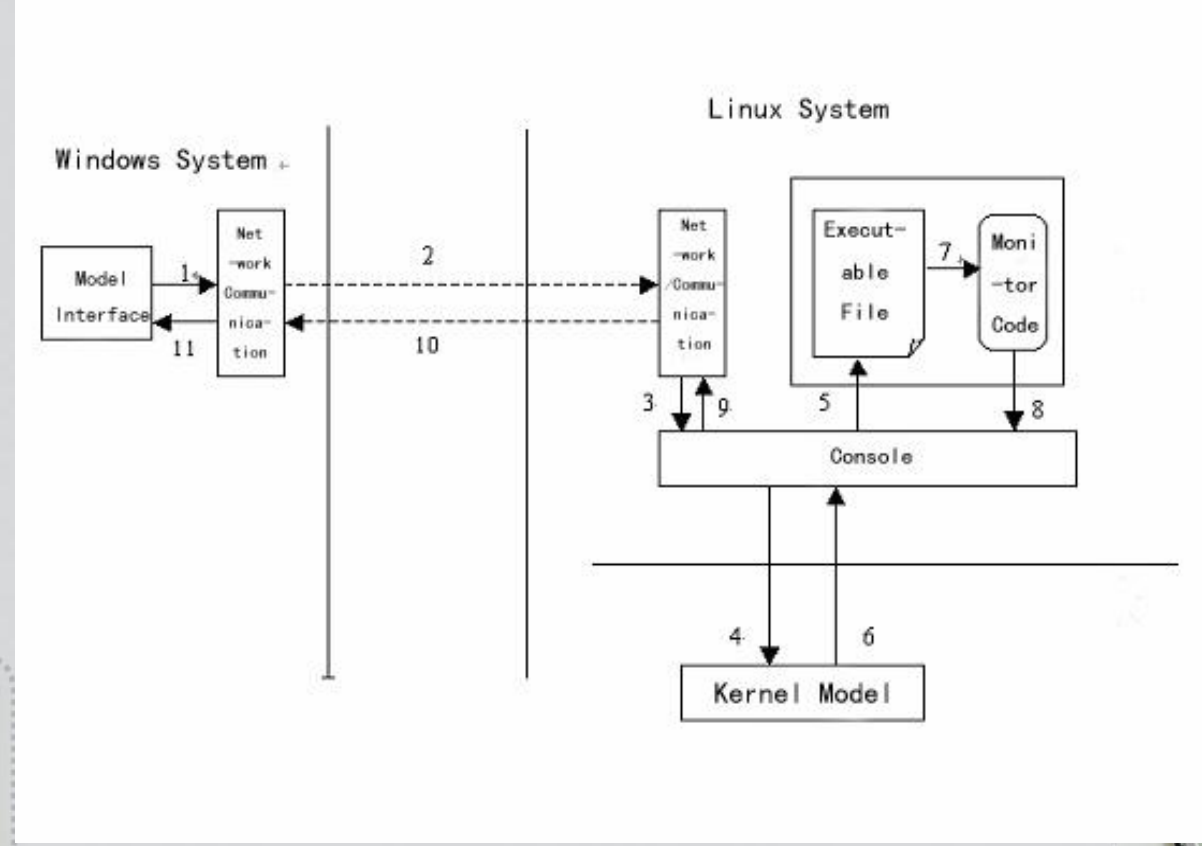
Operating Flow of Dynamic Analysis Model



Structure of PE File Executing Model



Structure of ELF File Executing Model

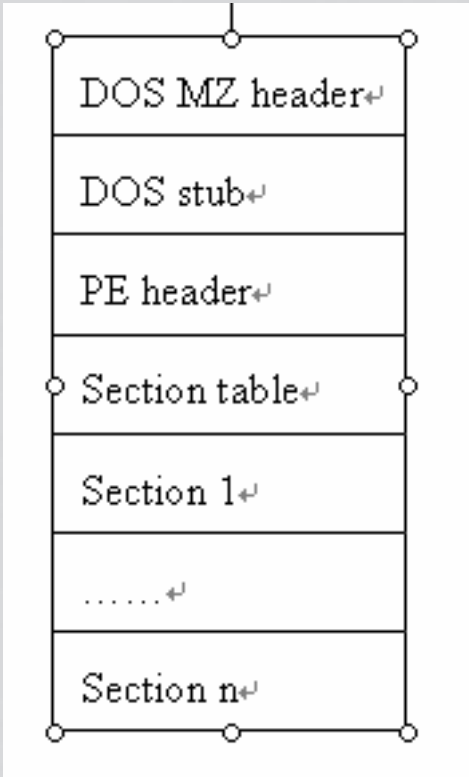


Key Technology in the Implementation of Executable Code Same Source Feature Measuring System

- Anti-Assembler Engine
- System Transfer Tracing
- I/O Monitor Technology

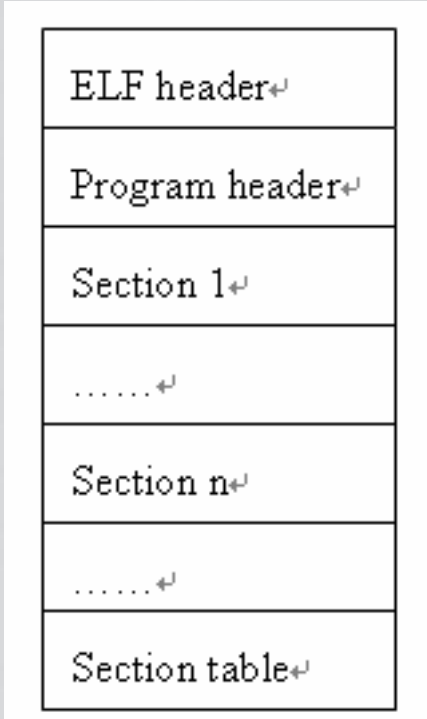
Anti-Assembler Engine

qStructure of PE File



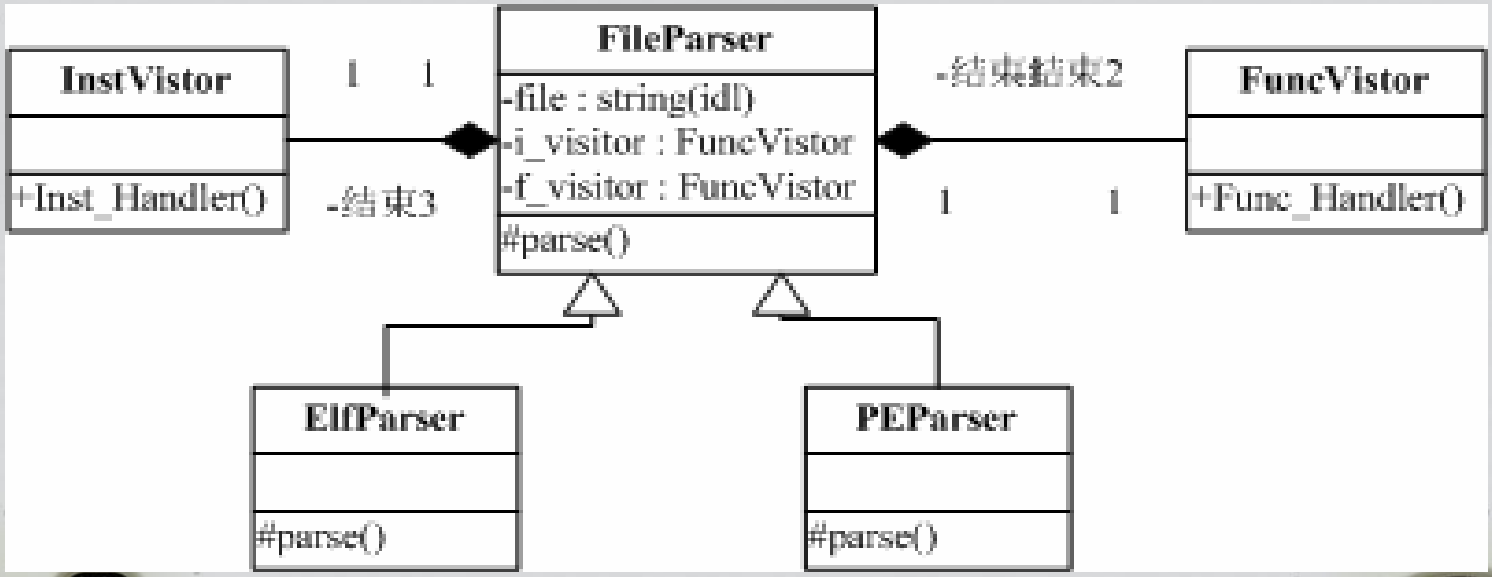
Anti-Assembler Engine

qStructure of ELF File



Anti-Assembler Engine

qClass structure of anti-assembler model



System Transfer Tracing

- **Key transfer tracing： to complete system transfer capture and record by revising the memory mapping of executable file. The process is as following**

1、 to write monitor code.

2、to compile the monitor code into DLL and educe monitor function from it.

3、to write the withdll aimed at startup the executable file, which receive the executable code and monitor code DLL file as its parameter. When starting the executable program WriteProcessMerory is used to inject the monitor code to its process space.

4、to start the program and revise all the system transfer function head to make the monitored process jump into the corresponding monitor function. Revise trampoline function on the same time. Copy instruction of original system transfer into trampoline function head.

5、withdll transfers control right to monitored process. The monitor code write the name of system transfer to name pipe and the analysis process read the system transfer serial in the name pipe.

System Transfer Tracing

q Principle of Monitor Code

```
;; Target Function
SystemFunc:
    push    ebp   [1 byte]
    mov     ebp,esp   [2 bytes]
    push    ebx   [1 bytes]
    push    esi   [1 byte]
    push    edi
    ....?
```

```
;; Trampoline Function
TrampolineSystemFunc:
    Nop
    Nop
    Nop
    Nop
    Nop
    jmp     SystemFunc
;; Detour Function
MonitorSystemFunc:
    doSomeMonitorJob
    jmp     Trampoline
```

```
;; Target Function
SystemFunc:
    jmp     MonitorSystemFunc[5 bytes]
    push    edi
    ....?
```

```
;; Trampoline Function
TrampolineSystemFunc:
    push    ebp
    mov     ebp,esp
    push    ebx
    push    esi
    jmp SystemFunc
;; Detour Function
MonitorSystemFunc:
    doSomeMonitorJob
    jmp     Trampoline+5
```

I/O Monitor Technology

- **I/O architecture of Windows NT：** an IRP include
  - One head field: including some bookkeeping information.
  - One or more parameter zone, which is named I/O stack location.

I/O Monitor Technology

- **Filter driver:the rules to complete a filter driver are**
  - A filter driver must fit the lower driver and guarantee system work normal after it is adopted. Even the filter driver need to repair a mistake of lower driver the condition must be satisfied also.
  - There should be a filter driver acknowledging the work principle of the device which the filter attached.
  - A filter driver must appear in any higher layer drivers and be as close to the original device as possible.
  - A filter driver must cooperate well with other filter drivers.

I/O Monitor Technology

q Redirectional function:

For example the monitored program wants to write a key system file C:\winnt\system32\kernel32.dll. If the key file is destroyed the system will face the danger of breakdown.So for running the program normally, our filter driver must do some work to the key file. If it is the first time to read or write on the key file, then KERNEL32.dll is copied to appointed backup directory beforehand first, then the read or write request is redirect to the backup directory. At the time the filter driver should transfer two IRP request to the lower driver: one is to backup the key file and the other is completing read or write operation to the backup file. If the key file has been backup, the work is only to redirect the read or write operating request to the backup file.(which means we need to maintain a protected file list to record the key file status in the kernel filter driver).If a delete operate is requested a success answer is returned directly.

Same Source Feature Measuring Technology Based on Feature Definition of Executable Code

The test principle of the presented same source feature measuring method of executable code: The used test cases are all hand worked, so the true same source feature of them is known by the author, which can be used to determine the confidential of the similarity value computed by our method.

The test result indicate the measurement computing value is press close to the true same source feature.

At the same time the system running result reveal that the compiling environment affect the computing destination severely to the program developed by high-level language especially to the code destination based on key code transfer. So the code transfer architecture of compelling environment should be included in the design of future measuring method research.

Same Source Feature Measuring Technology Based on Feature Definition of Executable Code

q Unsolved problem

In our model the measurement of static feature and dynamic feature are divided and a weighed computing is finished at last.

Ø Can static analysis result provide information in favor of dynamic analysis?

Ø How to enhance the cooperation of the static analysis and dynamic analysis?

Ø How to optimize the method based on subtler feature description?

Same Source Feature Measuring Technology of Software

Summary

The justice practice advanced insistent request to the same source feature measurement of software. In the era with software technology developed rapidly the method based on handworked interface compare and bit flow compare to verify the same source feature of the software can not satisfy the demanding of software right protection and strike with cyber crime. The feature of executable code is defined from static and dynamic angle for the first time in our research, based on which a same source feature measuring method is brought forward. The running result of the demo system reflects the same source feature of software well.

Thanks!