DESIGN OF SOFTWARE TRUSTED TOOL BASED ON SEMANTIC ANALYSIS

Guofengli

Faculty of Information Technology, Beijing University of Technology, Beijing, China

ABSTRACT

At present, the research on software trustworthiness mainly focuses on two parts: behavioral trustworthiness and trusted computing. The research status of trusted computing is in the stage of active immune of trusted 3.0. Behavioral trustworthiness mainly focuses on the detection and monitoring of software behavior trajectory. Abnormal behaviors are found through scene and hierarchical monitoring program call sequence, Restrict sensitive and dangerous software behavior.

At present, the research of behavior trust mainly uses XML language to configure behavior statement, which constrains sensitive and dangerous software behaviors. These researches are mainly applied to software trust testing methods. The research of XML behavior statement file mainly uses the method of obtaining sensitive behavior set and defining behavior path to manually configure. It mainly focuses on the formulation of behavior statements and the generation of behavior statement test cases. There are few researches on behavior semantics trustworthiness. Behavior statements are all based on behavior set configuration XML format declaration files. There are complicated and time-consuming problems in manual configuration, including incomplete behavior sets. This paper uses the trusted tool of semantic analysis technology to solve the problem of behavior set integrity And can generate credible statement file efficiently

The main idea of this paper is to use semantic analysis technology to model requirements, including dynamic semantic analysis and static semantic analysis. This paper uses UML model to automatically generate XML language code, behavioral semantic analysis and modeling, and formal modeling of non functional requirements, so as to ensure the credibility of the developed software trusted tools and the automatically generated XML files. It is mainly based on the formal construction of non functional requirements Model research, semantic analysis of the state diagram and function layer in the research process, generation of XML language trusted behavior declaration file by activity diagram established by model driven method, and finally generation of functional semantic set and functional semantic tree set by semantic analysis to ensure the integrity of the software. Behavior set generates behavior declaration file in XML format by the design of trusted tools Trusted computing is used to verify the credibility of trusted tools.

KEYWORDS

behavior declaration, behavior semantic analysis, trusted tool design, functional semantic set.

Natarajan Meghanathan et al. (Eds): SIPP, BIGML, DaKM, SOEN, AISC - 2020
pp. 83-90, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100207

1. INTRODUCTION

Behavior declaration refers to the collection of application software descriptions for its sensitive behaviors. In this set, sensitive behaviors include behaviors that may infringe the user's wn rights, behaviors that may affect the normal operation of the application software, and behaviors that may cause unexpected configuration changes of the software and hardware environment.

The generation of trusted behavior statement is implemented in the implementation phase of software. Through the definition of trusted behavior declaration completed in the software design phase, XML In the software requirement analysis stage, the software credible requirement is acquired; in the software design stage, the definition and design of the trusted behavior statement is carried out; in the software implementation stage, the preparation of the trusted behavior statement is carried out; in the software testing stage, the behavior statement is carried out Verification and improvement.

The research of software behavior in semantic analysis is at the forefront stage. Qu Yanwen put forward the concept of software behavior semantics. From the perspective of behavior semantics, the software function is complete. The main body traverses the whole behavior tree to ensure the integrity of function connection and prevent the lack of software function. Yang Xiaohui and others used API call analysis and instruction execution analysis to conduct behavior semantic analysis, and established a corresponding relationship between network data and malware behavior. It can be seen that simply analyzing the program or behavior sequence, ignoring the internal relations, will cause the lack of behavior expression information, can not meet the development of behavior analysis, modern modeling towards the direction of functional semantic analysis. Fu Jianming successfully parsed the system object from the system call parameters, giving the meaning of state semantics. Because semantics itself has its inherent logic, and is closer to the actual operation of users, and can find more hidden application layer attacks, so semantic analysis has gradually become the focus of current software behavior and network behavior researchers.

The research content of this paper uses the method of semantic analysis of software behavior to constrain the functional requirements. The UML use case diagram, activity diagram and state diagram generated by the demand modeling are used to produce the functional semantic tree, and then the functional semantic set is generated. The behavioral statement file is generated by the trusted operation behavior set, so as to avoid the lack or attack of the developed software function and ensure the reliability of the software.

2. SCHEME DESIGN:

Design ideas of trusted tools

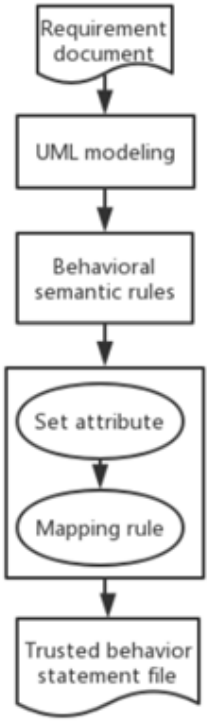

Figure 2-1Schematic diagram of trusted tool development process

2.1. Trusted Requirement Stage

According to the traditional requirement analysis process. Acquisition of non functional requirements. It is also necessary to obtain credible requirements for credibility.

2.2. Design of Behavior Statement Document

Statement of credible behavior

Statement file of trusted behavior: 1. Describe expected behavior of application software; 2. Implementation protocol of trusted software; 3. Evidence of trusted verification; 4. Auxiliary other documents

2.3. Template Design of Trusted Behavior Statement

This paper uses XML based format as the description and expression of trusted behavior statement, and its structure is consistent with the structure of trusted behavior statement file described above. The general definition of behavior statement is as follows:

< behavior declarationlist> / / behavior list
< behavior name one > / / represents a behavior
< behavior ID >< behavior ID > / / behavior number
< behavior item one > * * * < / behavior item one > / / behavior sub item 1
< behavior item two > * * * < / behavior item two > / / behavior sub item 2
... .
< credit level > dangerous < / credit level > / / credit level
</Behavior Name One>
< Behavior Name Two >
...
Other behaviors
< /Behavior Declaration List >

In the above example, < behavior declaration list > represents a list of all behaviors contained in the behavior declaration file. It contains multiple behavior items < behavior name XXX >, and multiple behavior sub items < behavior item XXX > constitute the specific operation and flow of the behavior item. Each behavior item must have corresponding trust level (such as danger, suspicion, trust, etc.), and the < security level > is used in the trusted behavior statement.

As an important parameter of trusted statement, the trustworthiness level clearly defines the trustworthiness of this behavior item, which provides an extremely important basis for trusted design, implementation and testing.

2.4. Credibility Detection of Behavioral Semantic Rules

If each function has credibility, the validity is tested. First, record the function trace, and then check whether it deviates from the function tree. If the function trace is < connect, verify, disconnect >, which is a path in the semantic tree, then the behavior has validity; if the function trace is < connect, communicate, disconnect, bypass the verification state, then there is an exception. The function level detection is based on the state level detection for software function semantics. Each function behavior should conform to the definition of function semantics set, and the function transformation should not exceed the function semantics tree. In this paper, combining the characteristics of software behavior, the function semantics rules are formulated to detect software behavior.

3. DESIGN AND IMPLEMENTATION OF TRUSTED TOOLS

Through the description of the first two chapters, the UML activity diagram is generated from the trusted requirement acquisition, and then the semantic analysis traverses the activity diagram to obtain the functional semantic set and the functional semantic tree. The software function defects are determined by rules, the software function semantic set and function semantic tree are divided, the system call sequence is used to ensure the ergodic integrity of the software call, and finally the software function integrity is guaranteed, and the software function credibility and complete semantic set data are used to develop the credible software behavior statement file.
Through a GUI tool generation, UML activity diagram behavior sets and attributes generate XML language form files,

Trusted tool design

GUI design mainly uses Tkinter module of Python third-party library to design gui. According to the template rules, the UML activity diagram is mapped to behavior set and attribute, and the XML file of behavior statement is generated

trusted statement tool

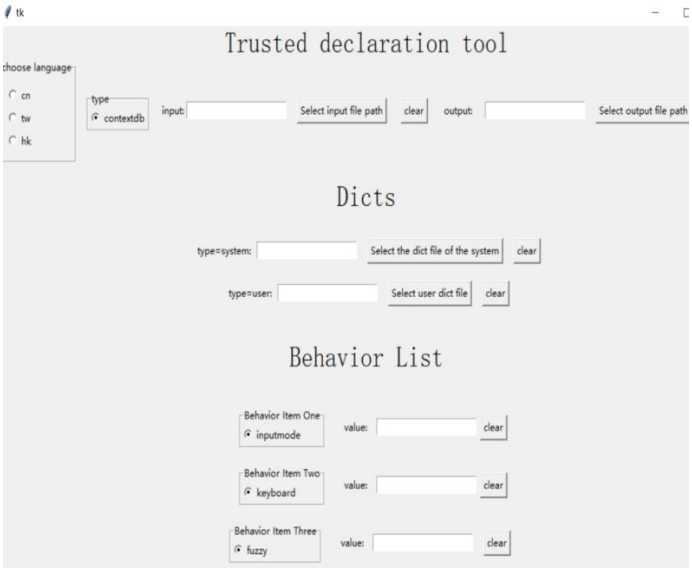

Figure 3-1schematic diagram of trusted declaration tool

4. GENERATE XML TRUSTED BEHAVIOR DECLARATION FILE

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- TODO: xml schema definition -->
  <login_List>
   <Behavior Name_One>
    <Behavior Id>10001</Behavior_Id>
    <Behavior Item_One>Operation</Behavior_Item_One>
       <Behavior Item_Two>login</Behavior_Item_Two>
       <Behavior Item_Three>username</Behavior_Item_Three>
       <Behavior Item_Four>password</Behavior_Item_Four>
     <Credibility Level>Dangerous</ Credibility_Level >
</Behavior_Name_One>
<_Behavior_Name_Two_>
<Behavior Id>10002</Behavior_Id>
    <Behavior Item_One>Operation</Behavior_Item_One>
       <Behavior Item_Two>logout</Behavior_Item_Two>
       <Behavior Item_Three>username</Behavior_Item_Three>
       <Behavior Item_Four>password</Behavior_Item_Four>
     <Credibility Level>credit</ Credibility_Level >
</Behavior_Name_Two>
</login_List >
```

Figure 3-2 Result Display of XML Behavior Declaration File Generation

5. EXPERIMENT SIMULATION AND RESULT ANALYSIS

4-1 table of experimental environment

development tool	IntelliJIDEA	development language	Java
operating system	CentOS 6.5	Tomcat	8.0
JDK	1.8	Mysql	5.7

Experimental case data: two kinds of data of file function (normal and illegal)

The definition of trusted level shows that it is divided into trusted danger

Experiment on software applications with and without trusted behavior statements
Through the design of functional behavior, normal login, registration, exit, upload, download and other behavior sets

Untrustworthy behavior: illegal login, illegal registration, illegal exit, upload dangerous files, illegal download behavior

Credibility calculation and analysis

Tool credibility

In order to verify the credibility of the tool, the concept of credibility is proposed. If a is a credible behavior, then t (a, b) is the credibility of B relative to a. In fact, it is to compare the indicator data of two behaviors a and B with the sample data defined in the behavior declaration file. According to the credibility of a single test data point to the sample data set, we turn the problem into the average of the sum of the credibility of each data point in B to the data set a, which is expressed as:

To achieve the effect of trusted behavior screening, through the trusted / untrusted operation of the application, the results are as follows:

$$T(\mathrm{A},\mathrm{B}) = \sum_{n=1}^{N} \frac{t(\mathrm{A},\mathrm{B}_n)}{n}$$

Calculate and analyze the reliability of the experiment

Table 5-1 configuration behavior statement trusted experiment group

Action	Test1	Test2	Test3	Test4	Test5
Sign in	0.98	0.96	0.97	0.92	0.96
register	0.94	0.98	0.94	0.89	0.99
Sign out	0.98	0.91	0.89	0.98	0.97
upload	0.93	0.92	0.90	0.93	0.93
download	0.97	0.99	0.91	0.92	0.95

Table 5-2unconfigured behavior statement untrusted experimental group

Action	Test1	Test2	Test3	Test4	Test5
Sign in	0.18	0.16	0.17	0.12	0.06
register	0.04	0.08	0.14	0.09	0.29
Sign out	0.28	0.21	0.09	0.08	0.07
upload	0.13	0.02	0.10	0.03	0.03
download	0.07	0.19	0.11	0.02	0.05

Analysis of experimental results

Through trusted computing, it can be seen clearly that the behavior statement file can detect illegal software behavior. When the file system is operated normally, the credibility is about 0.9 The software can identify dangerous / suspicious operations, and the reliability is less than 0.2. The application can deny access.

6. CONCLUSION

In this paper, based on the early stage of software requirement analysis, the developed software functions are divided into use case diagrams and activity diagrams, and the activity diagrams obtained from the modeling are extracted. Through the formal modeling of software functional requirements, the UML activity diagrams extract the software functional lines as a set, conduct behavioral semantic Analysis on the functional layer, form the functional semantic tree, and generate the functional semantics Set, reduce functional error logic, conduct behavior trust detection constraints on functional semantic set, generate XML behavior statement through trusted statement generation tool according to the pre design behavior statement template, and use trusted computing method to verify the functional data set, which meets the expected results.

In this experiment, we use the experimental data from functional semantic set to divide the attribute and behavior level to generate set template data. The behavior statements generated by trusted tools are verified to be credible by trusted computing. Simulation experimental data and experimental simulation results show that the software configures trusted behavior statements by using untrusted function behaviors and trusted function behaviors as test cases, which can effectively deny access to untrusted behaviors. The behavior statements generated by trusted computing methods meet the explicit indicators and behavioral reliability standards.

REFERENCES

[1] Tian L Q, Lin C, Ni Y. Evaluation of User behavior trust in cloud computing[C]. Computer Application and system Modeling. 2010(7):567-572.

[2] Song H, Kim B W, Mukherjee B. Multi-thread polling: A dynamic bandwidth distribution scheme in long-reach PON[J]. Selected Areas in Communications, IEEE Journal on, 2009, 27(2): 134-142.

[3] Wei H, Chen X Y, Wang C. User Behavior analysis based on network data stream scenario. IEEE 14th International Conference on Communication Conference. 2012:1017-1021.

[4] Gan T, Lin F H, Chen C J. User Behavior Analysis in Website Identification Registration[C]. China Communications.2013(3):76-81

[5] Su D, Li J, Wang ZY. A method of dynamic trusted researching of software behavior and its trusted elements.Network Security Technology & Application, 2013, (4):14–17.

[6] TIAN J, HAN J. Trustiness Evaluation Model Based on Software Behavior[J]. Energy Procedia, 2011, 13：7991-8002.

[9] Clercq R D, Keulenaer R D, Coppens B, et al. SOFIA: Software and control flow integrity architecture[C]. Design, Automation& Test in Europe Conference & Exhibition. 2016:1172-1177.

[10] Gao D, Reiter M K, Song D. Behavioral distance for intrusion detection[C]//Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, 2006: 63-81.

[11] Lin C, Tian L Q, Wang Y Z. Trusted network user behavior in the credible research [J].Research and development of the computer.2008,45(2):2033-2043.

[12] Paul C.Jorgensen. Software Testing [M]. CRC Press.2002.6.

[13] TIAN J, WANG Y. Dynamic credibility detection model base on scene mining for software behavior[J]. Energy Procedia, 2011, 13：577-584.