УДК 004.93

МЯГКАЯ МОДЕЛЬ ИДЕНТИФИКАЦИИ ОБЪЕКТОВ ИСКУССТВЕННОГО ПРОИСХОЖДЕНИЯ

Инкин П.А.

Аспирант 1 года обучения,

Академия Экономики и Управления,

ЧОУ ВО ЮУ (ИУБиП),

e-mail: v_gudenko@yandex.ru

Гребенюк Е.В.

Аспирант 2 года обучения

Сургутский государственный университет (СГУ)

Аннотация: Рассматриваются вопросы нечеткой идентификации программных закладок как объектов искусственного происхождения, в отличии от сбоев и помех, вызванных стихийными бедствиями или ошибочными действиями операторов. Предложена методика обнаружения и нейтрализации этих оообъектов.
Ключевые слова: квантовая декомпозиция, вредоносный код, семантический анализ, нормосхема.

SOFT MODEL OF IDENTIFICATION OF OBJECTS OF ARTIFICIAL ORIGIN

Inkin P.A.

Graduate student of 1 year of study,

Academy of Economics and Management,

PEI HE Southern University (IMBL),),

e-mail: v_gudenko@yandex.ru

Grebenyuk E.V.

Graduate student 2 years of study

Surgut State University (SSU)

Abstract: The issues of fuzzy identification of software bookmarks as objects of artificial origin are considered, in contrast to failures and interference caused by natural disasters or erroneous actions of operators. A technique for detecting and neutralizing these objects is proposed.
Keywords: quantum decomposition, malicious code, semantic analysis, normal scheme.

Анализ программного средства на предмет наличия в нем элементов разрушающих функций может быть представлен в виде следующей схемы:

Данная схема является внутренней структурой блока предварительной обработки (БПО) системы распознавания программных закладок [1-3], выполняющего основную функцию определения разрушающих функций в программе. На вход БПО поступает множество признаков разрушающих функций $P=\{p_1,..,p_k\}$, обнаруженных блоком анализатора ПС и образ программы в виде ее нормосхемы, разложенной до квантов второй степени или простых квантов.

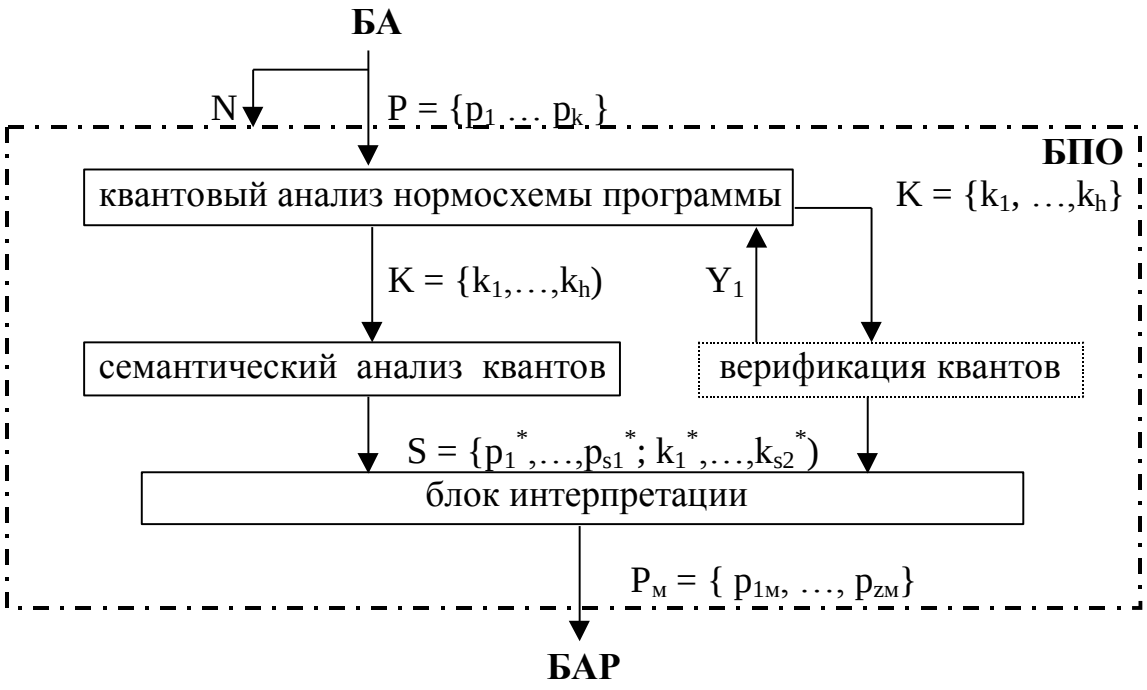

Рис.1. Схема блока предварительной обработки

Задача БПО – сортировка множества P по критерию «конфликта» с информацией блока АИ и формирование множества макропризнаков $p_{1м}$, …, $p_{zм}$, необходимых для функционирования алгоритма распознавания программных закладок [4].

Предложенная структура включает как стандартные компоненты анализа безопасности программ – семантический анализ, верификацию, так и новые – квантовый анализ потока управления нормосхемы программы, блок интерпретации признаков и квантов (компактов) программного средства [5,6].

Рассмотрим подробно назначение каждого компонента схемы.

1. **Квантовый анализ нормосхемы программы [7]**

Выполняет сопоставительный анализ первичных признаков $P = \{p_{1,}\ldots p_k\}$и нормосхемы с целью выделения тех квантов, которые содержат в своем составе первичные признаки - $K = \{k_1, \ldots, k_h\}$. При необходимости производит композицию нескольких квантов в компакт.

2. Семантический анализ квантов[8]

Выполняет функцию «отбора» квантов (компактов) нормосхемы программного средства. Критерий – конфликт программных функций с политикой целостности и безопасности среды применения, оперирующей с понятием нелегитимного доступа.

3. Верификация квантов [9]

Выполняет аналогичную функцию «отбора» квантов (компактов) нормосхемы программного средства. Критерий – конфликт программных функций со следующими документами: спецификацией программы; техническим заданием; описанием программы; пояснительной запиской программы.

4. Блок интерпретации [10]

Формирует макропризнаки разрушающих функций в лингвистическом виде $P_м = \{ p_{1м}, \ldots, p_{zм}\}$ методом анализа возможных сочетаний первичных признаков $p_1^*,\ldots,p_{s1}^*$. Выполняет функцию отображения квантов (компактов) программы $k_1^*,\ldots,k_{s2}^*$, содержащих первичные признаки, в их реальное представление в программе – например, в подпрограмму, объект, функцию, цикл и т.д. Основное отличие части блока интерпретации, выполняющей семантическое связывание первичных признаков, от метода поиска программных закладок вирусного типа по сигнатурам, состоит в отсутствии привязки к конкретной структуре и функционированию закладки (штаммы вирусов определенных версий).

Функционирование блока предварительной обработки в целом можно представить в виде следующей последовательности шагов [7]:

1. Идентификация квантов второй степени нормосхемы программного средства, содержащих первичные признаки $p_{1,}…p_k$. Предполагает определение тех квантов, в которые «попадает» хотя бы один первичный признак. В результате формируется множество квантов, которые необходимо подвергнуть анализу на предмет наличия элементов разрушающих функций – k_1, …, k_h;

2. Семантический анализ квантов k_1, …, k_h. На основе принятой операционной модели семантического анализа программного средства проводится выбор тех квантов, которые не удовлетворяют требованиям модели;

3. Верификация квантов (компактов) программного средства. Проводится поиск тех квантов (компактов) нормосхемы программы, которые не отвечают спецификациям, представленным в сопроводительных документах. В случае отсутствия информации по отдельному верифицируемому кванту k_i проводится преобразование его в компакт – операция композиции Y_1 ;

4. Формирование множества квантов программы, предположительно нарушающих целостность и безопасность вычислительной системы – $k_1^*,…,k_{s2}^*$. На основании выходной информации блоков семантического анализа и верификации принимается решение о формировании множества $k_1^*,…,k_{s2}^*$:

- приоритет отбора квантов для множества $k_1^*,…,k_{s2}^*$ принадлежит блоку семантического анализа, т.е. в случае обнаружения несоответствия кванта семантической модели он становится элементом указанного множества;
- если квант отвечает модели семантического анализа, но не проходит проверку в блоке верификации программы, то он также становится элементом множества $k_1^*,…,k_{s2}^*$.

В соответствии с полученной совокупностью квантов $k_1^*,…,k_{s2}^*$ изменяется множество первичных признаков $p_{1,}…p_k$. Множество $p_1^*,…,p_{s1}^*$

является подмножеством $p_{1,}…p_k$ и содержит все признаки, встречающиеся в выбранных квантах.

5. Формирование макропризнаков в блоке интерпретации [11]. Основа работы блока – база знаний, содержащая наиболее общие структуры участков программ, характерных для специальных программных закладок. Примеры таких участков программ, написанных на языке программирования Ассемблер, приведены ниже:

С помощью этой базы знаний проводится анализ полученной на предыдущих этапах совокупности первичных признаков $p_1^*,…,p_{s1}^*$. Метод проведения анализа состоит в автоматическом подборе возможных сочетаний первичных признаков с целью получения аналогичных структур, заложенных в базе знаний и постановки им в соответствие макропризнака в лингвистическом виде. Например, п. 3 - структура соответствует макропризнаку «установка резидентного модуля». Таким образом в результате формируется множество макропризнаков $P_м = \{ p_{1м}, …, p_{zм}\}$.

По результатам анализа БПО можно предварительно идентифицировать участки программы [4,11], нарушающие целостность и безопасность вычислительной системы.

СПИСОК ЛИТЕРАТУРЫ

1. Храмов В.В. Принцип интеллатентности и его использование в задачах распознавания // Тематический научно-технический сборник. – Пущино, 1994. — С. 62–66. — URL: https://elibrary.ru/item.asp?id=32838003. (дата обращения: 04.02.2020).
2. Храмов В.В. Способ агрегирования нескольких источников нечеткой информации // Известия ТРТУ. — 2001. — № 3(21). — С. 52–53 — URL: https://elibrary.ru/item.asp?id=12886331. (дата обращения: 04.02.2020).
3. Храмов В.В. Особенности мажоритарной обработки нечеткой информации // Спектральные методы обработки информации в научных исследованиях: Доклады I Всероссийской конференции (Спектр-2000). РФФИ, Институт математических проблем биологии РАН. – 2000. – С. 136-138. – URL:https://elibrary.ru/item.asp?id=32656899(дата обращения: 18.01.2020).
4. Храмов В.В. и др. Защита информации в вычислительных системах// Учебное пособие для вузов. – М., 2002. – 192с. – URL:https://elibrary.ru/item.asp?id=32762286(дата обращения: 18.01.2020).
5. Информационная безопасность и защита информации в цифровой экономике. Элементы теории и тестовые задания: Учебное пособие. – / И.Д. Алекперов, В.В.

Храмов, А.А. Горбачева, Д.П. Фомичев; ЮУ (ИУБиП). – Ростов-на Дону, 2020. – 114 с.
6. Храмов В.В., Трубников А.Н. Модель элементарной защиты программного средства // Информационные технологии и проблемы микроэлектроники: Сборник научных статей /под ред. докт. тех. наук, проф. В.А. Бархоткина. – М., 1999. – С. 192-197. –URL:https://elibrary.ru/item.asp?id=327112590(дата обращения: 18.01.2020).
7. Храмов В.В.Основы методологии синтеза средств защиты информации // Проблемы обеспечения эффективности и устойчивости функционирования сложных технических систем: Материалы XXI Межведомственной научно-технической конференции. – 2002. – С. 115-120. – URL: https://elibrary.ru/item.asp?id=32877301(дата обращения: 18.01.2020).
8. Храмов В.В., Хадыка А.С. Разработка принципов, методов и средств самоорганизации систем защиты информации // Транспорт-2006: Труды Всероссийской научно-практической конференции. – РГУПС, 2006. – С. 230-232. – URL:https://elibrary.ru/item.asp?id=32632220(дата обращения: 18.01.2020).
9. Голубенко Е.В., Ковтун О.Г. и др. Информационная безопасность и защита информации на транспорте: Тестовые задания по дисциплине. – Ростов-на-Дону, 2015. –URL: https://elibrary.ru/item.asp?id=36311930(дата обращения: 18.01.2020).
10. Khramov V.V., Trubnikov A.N.Analysis of the aggressiveness of a software product //Automatic Control and Computer Sciences. – 1999. – Т. 33, № 2. – С. 28-34. – URL:https://elibrary.ru/item.asp?id=13328155(датаобращения: 18.01.2020).
11. Губарев О.К. и др. Способ повышения безопасности программных средств и пути его реализации // Тематический научно-технический сборник. – Пущино, 1994. – С. 56-61. –URL:https://elibrary.ru/item.asp?id=32837963(дата обращения: 18.01.2020).