УДК 519.67

МОДЕЛЬ МАШИННОГО КОДА, СПЕЦИАЛИЗИРОВАННАЯ ДЛЯ ПОИСКА УЯЗВИМОСТЕЙ

М. В. Буйневич, К.Е. Израилов, О.В. Щербаков

Описаны предпосылки использования модели машинного кода в методе алгоритмизации. Заданы ее основные элементы и требования. Произведено сравнение с «классической» моделью. Рассмотрен процесс моделирования.

Ключевые слова: *безопасность, программное обеспечение, уязвимость, моделирование, машинный код, алгоритмизация.*

Введение. Актуальность проблемы безопасности программного обеспечения (далее – ПО) [1] заключается в наличии уязвимостей (как случайных, так и умышленных) в коде ПО при отсутствии эффективных средств их выявления. Ситуация в разы осложняется, когда исходный код отсутствует, а для анализа в наличии только машинный код (далее – МК), что наблюдается сплошь и рядом, например, в проприетарном ПО устройств телекоммуникационных сетей (в том числе в ключевых системах информационной инфраструктуры). С одним из немногих методов решения задачи поиска уязвимостей в МК можно ознакомиться в авторской статье [2].

Суть метода (далее – метод) заключается в восстановлении алгоритмов машинного кода и представлении их в виде, подходящем для ручного анализа с целью поиска уязвимостей. При этом основная часть метода может быть реализована с помощью программного средства. Такое совокупное использование как ручного, так и автоматизированного способов позволяет добиться высокой скорости работы метода при большом проценте обнаруживаемых уязвимостей. Возможность же итеративного выполнения его фаз с необходимыми корректировками предоставляет полный контроль над процессом поиска.

Буйневич М.В. д-р тех. наук, проф., профессор кафедры общетехнических и специальных дисциплин, Санкт-Петербургский университет ГПС МЧС России, Россия, г. Санкт-Петербург,
Израилов К.Е., аспирант,
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, Россия, г. Санкт-Петербург,
Щербаков О.В. д-р тех. наук, проф., профессор кафедры прикладной математики и информационных технологий, Санкт-Петербургский университет ГПС МЧС России, Россия, г. Санкт-Петербург,

Построение любого метода «поверх» подходящей модели является рациональным, а вследствие особенностей самого метода необходимым решением. Хотя его входными данными и является МК с четко заданным синтаксисом, основные результаты и конечная цель представляют собой такие зачастую неоднозначные и субъективные понятия, как алгоритм и уязвимость.

Это приводит к необходимости применения модели, которая бы позволила отразить основные понятия метода с помощью более формальных объектов. Так как основная часть метода автоматизирована, предлагаемое для этой цели специализированное программного средство (так называемая утилита [3]) – и вовсе не может быть реализовано без каких-либо математических (в основном программных) моделей. Этот вывод следует, например, из применяемого в утилите внутреннего представлении, основанного на графах и деревьях [4].

Выбору и описанию подходящей модели для задачи поиска уязвимостей в МК и посвящена настоящая статья.

Требования к модели. Определим вначале элементы модели (далее – элементы), которые будут отражать особенности самого метода.

Во-первых, объектом для применения метода является МК, который должен быть связан с большинством других элементов предметной области.

Во-вторых, назначение метода заключается в восстановлении алгоритмов кода – значит, некое их представление должно быть четко отражено в модели. В-третьих, целью метода служит поиск уязвимостей, а, следовательно, введение в саму модель такого элемента сильно упростило и расширило бы его применение.

Корректности ради заметим, что в модели достаточна поддержка не столько окончательной и верной информации об уязвимостях (что практически невозможно), сколько вероятности (или «подозрительности») их наличия. И, наконец, четвертым элементом должны быть метаданные кода (например, сигнатуры функций и глобальные переменные), частично присутствующие в МК и восстанавливаемые методом.

Понятия «алгоритм кода» и «уязвимость» в рамках метода являются чрезмерно широкими и для более точного отражения реального объекта описания (то есть, МК) требуют «дробления».

Для этого разделим понятие алгоритма кода на два: архитектура кода и алгоритм отдельной функции. Первое логично будет соответствовать схеме взаимодействия функций, а второе – работе самих функций. Затем произведем типизацию уязвимостей путем их иерархической структуризации (стратификации), что отражено в следующей таблице (строки таблицы расположены в порядке увеличения масштаба страты).

Таблица

Тип уязвимости	Структурный уровень	Местоположение в коде
Вычислительный	Низкий (НУ)	Реализация вычислений, типов и структур данных
Алгоритмический	Средний (СУ)	Логика работы алгоритмов функций
Архитектурный	Высокий (СУ)	Схема взаимодействия функций

Примечание. Далее будем использовать сокращенные обозначения типов уязвимостей по их страте, а именно – НУ, СУ и ВУ.

Для лучшего представления страт приведем реальные примеры уязвимостей по всем трем типам. Типичным примером НУ является ошибка реализации функции при работе с массивами, такая как выход за пределы их допустимого диапазона. К ошибке СУ может быть отнесена неверная реализация схемы алгоритма шифрования DES [5] (при корректной реализации отдельных ее шагов). И наконец, пожалуй, самой тяжело формализуемой уязвимостью, а именно ВУ, может являться такое построение архитектуры, при котором модули обработки конфиденциальной информации передают данные без использования требуемого механизма защиты (например, пароли для раскрытия информации расположены в области памяти, к которой возможен несанкционированный доступ). С критичностью наличия последнего типа можно ознакомиться в авторской статье [6].

Необходимо отметить, что при такой типизации уязвимостей метод применим исключительно к СУ и ВУ. Однако это является лишь его ограничением, а не недостатком; методы и программные средства для поиска НУ в той или иной степени существуют и постоянно развиваются, чего нельзя сказать про уязвимости более высоких страт – на их поиске собственно и позиционируется метод.

Помимо определения элементов модели к ней предъявляются «общенаучные» требования, которые в рамках метода определяются как:

- Адекватность – модель должна соответствовать представляемому ею МК, в котором возможно наличие разнотиповых уязвимостей и который должен быть описан в алгоритмизированном виде.
- Анализируемость – получаемые с ее помощью результаты, а именно алгоритмы, должны подходить для последующего анализа экспертом по безопасности кода (далее – эксперт).
- Универсальность – модель по возможности должна мало зависеть от процессора МК и различных модификаций языка конечного описания его алгоритмов.
- Целесообразность – ресурсы, затрачиваемые на использование модели, должны подтверждать заявленное преимущество метода как низкотрудоемкого и высокоэффективного по сравнению с аналогами.

Анализ сферы моделирования МК показал отсутствие каких-либо моделей в принципе. Если быть более точным, то существует лишь «классическая» модель, явно нигде не упоминаемая и подразумеваемая de-facto. Ее описание приведено далее.

«Классическая» модель. Данная модель позволяет представить практически любой МК с позиции процессорных инструкций. Применением ее является как разработка и отладка самого кода, так и реализация всевозможных методов его анализа. К ним, в частности, относится и поиск уязвимостей. Модель неявно применяется в таких программных средствах, как компиляторы, ассемблеры, дизассемблеры и антивирусы.

Схема модели, адаптированная для использования в методе (путем внесения в нее всех требуемых элементов – МК, уязвимостей, алгоритмов, метаданных), представлена на рис. 1.

Основными элементами «классической» модели являются инструкции машинного кода, поделенные на блоки (в лучшем случае – на функции), что неизбежно приводит к затруднительному представлению более высокоуровневых абстрактных элементов (алгоритмы функций, архитектура). Так, например, если НУ и хорошо представляются моделью, то СУ «размыты» по всем элементам кода, что не дает эффективно применять алгоритмы их поиска. То же самое относится и к алгоритмам функций, нераздельно связанным в модели с МК, который (как неоднократно отмечалось в авторских статьях) абсолютно не подходит для анализа Экспертом. Остаточные же метаданные в МК, также

присутствующие в модели, не предназначены для решения задачи метода; ко всему прочему они, как и алгоритм, являются неотделимой частью МК.

С учетом вышеизложенного напрашивается вывод, что использование метода (а в особенности автоматизирующих его программных средств) на базе данной (пусть адаптированной, но все же «классической») модели будет крайне затруднительным.

Рис. 1. Адаптированная «классическая» модель машинного кода

Завершая анализ применимости данной модели, кратко укажем выполнение ею выдвинутых требований. Модель имеет недостаточную адекватность к реальной задаче по алгоритмизации кода, а анализируемость получаемых на ней результатов предсказывается как низкая. Хотя модель и имеет достаточную универсальность для представления кода, абсолютная нецелесообразность ее использования сводит на нет это единственное преимущество. Таким образом, логичным является создание собственной модели, построенной на заданных элементах и удовлетворяющей всем требованиям. В дополнение она сможет хотя бы частично заполнить пустующую сферу моделирования МК, заменив лишь негласно подразумеваемую «классическую» модель на вполне определенную – названную авторами «структурной».

«Структурная» модель. Идея предлагаемой модели заключается в создании системы следующих псевдоортогональных плоскостей: вложенных структур абстракций программного кода и линейного МК, обладающих следующими свойствами. Во-первых, из-за псевдоортогональности все абстракции связаны со своими представлениями в МК, что позволит избежать потери какой-либо существенной низкоуровневой информации (эта часть схожа с «классической» моделью). А, во-вторых, алгоритмы метода могут решать его подзадачи на элементах более высоких уровней абстракции (например, проводить анализ алгоритмов функций и построение связей между ними), поскольку последние имеют выделенную плоскость представления.

Таким образом, без потери самого МК, модель заключает в себе основные иерархические элементы соответствующего ему исходного кода. В интересах же решения основной задачи метода, состав ее элементов должен быть дополнен СУ и ВУ.

Для автоматизированного применения любого метода модель должна содержать данные и интерфейсы, которые могут применяться при моделировании с помощью программных средств.

Схема предлагаемой модели (далее – структурная модель) представлена на рис. 2.

Как хорошо видно, структурная модель содержит все необходимые элементы (при этом ни один из них не является «размытым» по предметной области), поддерживается их четкая иерархия, заложены условия для практического применения.

Также структурная модель соответствует всем необходимым требованиям: она содержит инструкции МК с уязвимостями, прогнозируемые на ней результаты (алгоритмы и уязвимости) подходят для анализа экспертом, основная часть модели не зависит от характеристик процессора МК, а наличие интерфейсов и данных позволяет добиться высокой степени автоматизации.

Объектом исследований на модели является МК, предметом исследования – его алгоритмическое представление, а конечной целью – поиск СУ и ВУ (далее под термином «уязвимость» будем подразумевать только эти два типа). Для применения в методе наиболее подходящим формальным видом модели является совокупность аналитического и информационного (включая формально-логический); собственно моделированием же может быть математическое на базе компьютерного. Границы применимости модели распространяются на любой МК, но при наличии исходного кода ее

эффективность (впрочем, как и метода в целом) будет ниже альтернативных – область поиска уязвимостей по исходному коду имеет более развитую (как теоретическую, так и практическую) базу.

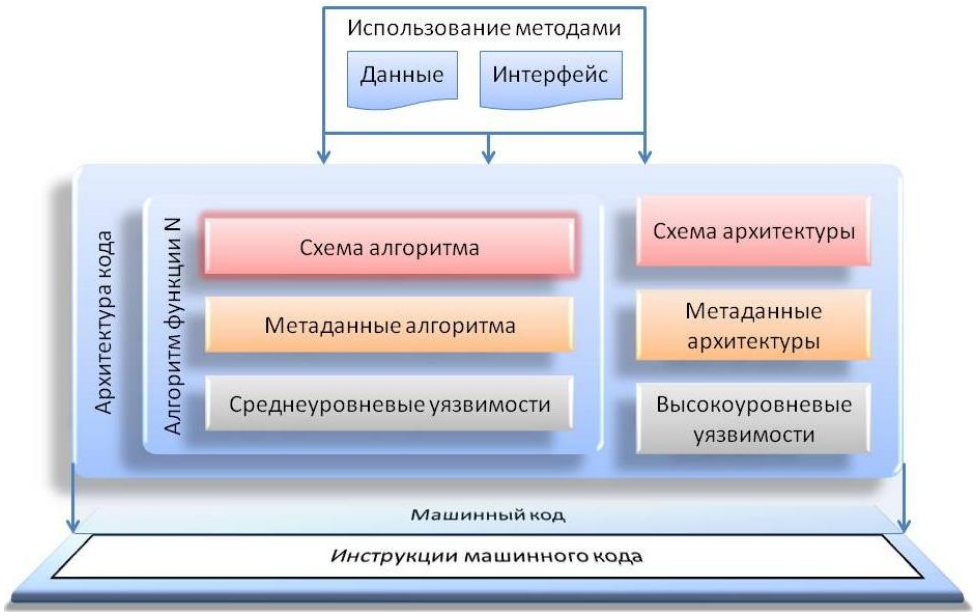

Рис. 2. «Структурная» модель машинного кода

Моделирование. При внимательном рассмотрении схемы метода [7] можно заметить, что она в точности описывает процесс моделирования МК (далее – моделирование), примененный к каждому конкретному экземпляру кода. То есть, алгоритмизацию МК с целью поиска уязвимостей можно считать созданием его точной модели с целью проведения необходимых экспериментов по воссозданию алгоритмов, их анализу и «переносу» найденных уязвимостей на код. Рассмотрим такое отожествление более подробно.

Акторы процесса моделирования соответствуют следующим сущностям метода: познаваемым объектом является конкретный вариант исследуемого МК, познающим субъектом – эсперт, а инструментом познания – собственно предложенная «структурная» модель, создаваемая для каждого экземпляра МК.

Хотя взаимно-однозначное соответствие фаз метода и моделирования и отсутствует, их совокупности совпадают: каждая фаза метода имеет свое отражение и в процессе моделирования,– что показано на рис. 3.

Важно отметить, что согласно рис. 3, наиболее «тяжелая» часть моделирования (фазы 2–4) полностью соответствует основной части Метода (фаза 2) и, следовательно, может быть реализована с помощь программного средства – утилиты.

Таким образом, в контексте моделирования авторский метод можно определить как: *процесс математического моделирования экземпляра МК на базе «структурной» модели с целью получения алгоритмов его работы, итерационно проводимый экспертом с использованием утилиты.*

Заключение. Как было показано выше, пока не существует подходящих моделей, на которых мог бы строиться метод поиска уязвимостей в МК. Удовлетворительным разрешением этой проблемной ситуации может служить предлагаемая авторами «структурная» модель. При этом она вкупе с авторским методом тождественна каноническому процессу моделирования, что является хоть и косвенным, но доказательством ее достоверности. Прямым доказательством может служить только ее работоспособность.

В завершении заметим, что, на первый взгляд, может показаться невозможность реального применения как самой модели, так и процесса моделирования предметной области. Причина такого суждения вытекает из очевидной необходимости проведения крайне не тривиального преобразования МК из его начальной формы (текстового ассемблера) к той, которая является основной для модели (некая трехмерная система взаимосвязанных параметров кода, таких как инструкции кода, графы алгоритмов, множество метаданных, предположения об уязвимостях). В ином случае как построить модель для отдельного экземпляра МК, так и выделить в нем алгоритмы, будет попросту невозможно.

Однако все эти преобразования согласно фазам моделирования могут быть решены с помощью утилиты, существование и работоспособность которой (пускай пока и имеющей вид прототипа) подтверждается опубликованными примерами ее работы [3, 4], а также реальной возможностью проведения натурной алгоритмизации в виде Web-приложения [8].

Рис. 3. Соответствие фаз метода и моделирования

Также остались за кадром какие-либо детали собственно самого способа построения модели, проведения на ней экспериментов и формирование оценок эффективности. Такого рода информация (по мере получения результатов авторами) будет представлена в последующих публикациях.

Библиографический список

1. **Израилов, К. Е.** Алгоритмизация машинного кода телекоммуникационных устройств как стратегическое средство обеспечения информационной безопасности / К. Е. Израилов // Национальная безопасность и стратегическое планирование. – 2013. – № 2 (2). – С. 28–36.

2. **Буйневич, М. В., Израилов, К. Е.** Метод алгоритмизации машинного кода телекоммуникационных устройств / М. В. Буйневич, К. Е. Израилов // Телекоммуникации. – 2012. – № 12. – С. 2–6.

References

1. **Izrailov, K. E.** Algoritmizatsiya mashin-nogo koda telekommunikatsionnyih ustroystv kak strategicheskoe sredstvo obespecheniya informatsionnoy bezopasnosti / K. E. Izrailov // Natsionalnaya bezopasnost i strategicheskoe planirovanie. – 2013. – № 2 (2). – С. 28–36.

2. **Buynevich, M. V., Izrailov, K. E.** Metod algoritmizatsii mashinnogo koda telekommunikatsionnyih ustroystv / M. V. Buynevich, K. E. Izrailov // Telekommunikatsii. – 2012. – № 12.

3. **Буйневич, М. В., Израилов К. Е.** Автоматизированное средство алгоритмизации машинного кода телекоммуникационных устройств / М. В. Буйневич, К. Е. Израилов // Телекоммуникации. – 2013. – № 6. – С. 2–9.

4. **Израилов, К.Е.** Внутреннее представление прототипа утилиты для восстановления кода / К. Е. Израилов // Фундаментальные и прикладные исследования в современном мире: матер. II Междунар. научно-практ. конф. – СПбГПУ, 2013. – С. 79–90.

5. **Data Encryption Standard** (DES) // Википедия: энциклопедия. URL: http://wikipedia.org/wiki/Data_Encryption_Standard (15.03.2014).

6. **Израилов, К.Е.** Архитектурные уязвимости программного обеспечения / К. Е. Израилов // «ИНЖЭКОН-2013»: тез. докл. VI науч. конгресса студентов и аспирантов. СПбГИЭУ. 18-19 апреля 2013 г. – СПб, 2013. – С. 35.

7. **Buinevich, M., Izrailov, K.** Method and utility for recovering code algorithms of telecommunication devices for vulnerability search / M. Buinevich, K. Izrailov // The 16th International Conference on Advanced Communications Technology (ICACT-2014), Bongpyeong-myeon, Korea (South), on February 16-19, 2014.– P. 172-176.

8. **On-line версия программы восстановления машинного кода** [личный сайт Израилова К.Е.]. URL: http://demono.ru/onlineDemono.aspx (16.03.2014).

– С. 2–6.

3. **Buynevich, M. V., Izrailov K. E**. Avtomatizirovannoe sredstvo algoritmizatsii mashinnogo koda telekommunikatsionnyih ustroystv / M. V. Buynevich, K. E. Izrailov // Telekommunikatsii. – 2013. – № 6. – C. 2–9.

4. **Izrailov, K.E.** Vnutrennee predstavle-nie prototipa utilityi dlya vosstanovleniya koda / K. E. Izrailov // Fundamentalnyie i prikladnyie issledovaniya v sovremennom mire: mater. II Mezhdunar. nauchno-prakt. konf. – SPbGPU, 2013. – S. 79–90.

5. **Data Encryption Standard** (DES) // Vikipediya: entsiklopediya. URL: http://wikipedia.org/wiki/Data_Encryption_Standard (15.03.2014).

6. **Izrailov, K.E.** Arhitekturnyie uyazvimosti programmnogo obespecheniya / K. E. Izrailov // «INZhEKON-2013»: tez. dokl. VI nauch. kongressa studentov i aspirantov. SPbGIEU. 18-19 aprelya 2013 g. – SPb, 2013. – C. 35.

7. **Buinevich, M., Izrailov, K.** Method and utility for recovering code algorithms of telecommunication devices for vulnerability search / M. Buinevich, K. Izrailov // The 16th International Conference on Advanced Communications Technology (ICACT-2014), Bongpyeongmyeon, Korea (South), on February 16-19, 2014.

8. **On-line versiya programmyi vosstanovleniya mashinnogo koda** [lichnyiy sayt Izrailova K.E.]. URL: http://demono.ru/onlineDemono.aspx (16.03.2014).

MODEL OF MACHINE CODE SPECIALIZED FOR VULNERABILITIES SEARCH

M.V. Buinevich,
doctor of technical sciences, professor,
Saint-Petersburg university of State fire service of EMECOM of Russia
Russia, Saint-Petersburg;
K.E. Izrailov, adjunct,
Saint-Petersburg state university of telecommunication named after prof. M.A. Bonch-Bruevich,
Russia, Saint-Petersburg;
O.V. Scherbakov, doctor of technical sciences, professor,
Saint-Petersburg university of State fire service of EMECOM of Russia,
Russia, Saint-Petersburg.

Described the background to the using of model of machine code in the method algorithm presentation. Given elements and requirements of this model. Compared with the «classical» model. Described the process of modeling.

Keywords: *security, software, vulnerability, modeling, machine code, algorithm presentation.*