УДК 681.3.05

Статистический анализ программных модулей

Д. Ю. Борнин, А. А. Курочкин, Д. Б. Николаев

Рассмотрена проблема обеспечения безопасности и достоверности передачи информации во взаимосвязи с характеристиками передаваемых данных, обладающих внутренними зависимостями. Представлены результаты анализа исполняемых модулей программного обеспечения микропроцессорных устройств различных аппаратных и программных платформ.

Данная статья посвящена работам, проводимым в области преобразования и обработки информации. Особое значение имеет исследование информационного представления особенностей программ в машинном коде. При этом избыточность исходного текста может существенно влиять на возможности дальнейшего преобразования информации [1].

В статье приведены основные результаты по статистическому и вероятностному анализу языков программирования низкого уровня для процессоров Intel 8086, Intel 80286 – 80486, Intel Pentium, К580ВМ80, КМ1816ВЕ51 применительно к вычислительным системам и системам обеспечения информационной безопасности. Анализ включает:

- исследование машинных кодов команд на предмет избыточности;
- исследование формата команд;
- статистический анализ особенностей языков программирования низкого уровня.

В процессе работы создано программное обеспечение для проведения статистического анализа машинных кодов программ. Результаты анализа для вышеперечисленных процессоров приведены ниже. Кроме того, сформулированы рекомендации по устранению избыточности языка, которая влияет на стойкость защищенного исполняемого модуля.

Анализ языка программирования Assembler для процессоров Intel 8086 – 80286 [2] показал наличие информационной избыточности в машинном коде, которая существенно облегчает анализ при обработке информации. Анализ показал, что существуют три вида избыточности машинного кода:

- избыточность кода операции;
- избыточность формата команды;
- избыточность операндов.

Избыточность кода операции заключается в том, что команды с некоторыми кодами отсутствуют в таблице кодировки. Например, для процессора Intel 8086 неиспользуемыми кодами являются: 64, 65, 66, 67, D6.

Избыточность формата команды возникает вследствие ограничений, налагаемых на значения префикса и перекрытия сегмента. Избыточность формата команды практически не влияет на стойкость преобразованного машинного кода, так как байты префикса и перекрытия сегмента могут отсутствовать, а по кодировке они идентичны соответствующим командам языка.

Избыточность операнда определяется ограничениями, накладываемыми на операнд машинной команды структурой языка и конфигурацией системы. Избыточность операнда является наиболее существенной и может значительно облегчить анализ.

Операнд R/M при использовании в командах содержит незначительную избыточность, за исключением тех случаев, когда он содержит код операции (команды AR OP1, AR OP2, Shft OP, GRP1, GRP2, GRP3).

Непосредственное значение (imm8, imm16) содержит информационную избыточность при использовании в командах перехода RET, CALL, JMP, ENTER. В этих случаях значение imm может определяться конфигурацией системы. Конфигурацией системы также определяется избыточность команд, содержащих в качестве операнда непосредственный адрес памяти (mem). Наибольшую информационную избыточность несут в себе операнды, определяющие регистр (reg) или порт (port).

Структура системы команд процессора Intel 80386 в основном аналогична системе команд Intel 8086. Основные отличия содержатся в использовании 32-битных операндов и в расширенном формате команды. Кроме того, многие команды процессора 80386 имеют двухбайтный код операции.

Избыточность кода операции процессора 80386 минимальна: не используется код D6h. Формат команды отличается наличием префиксов размера операнда и адреса, а также SIB-байта. В результате анализ машинного кода несколько усложняется (особенно при использовании защищенного режима работы процессора). Наибольшая избыточность возникает при использовании 32-битных операндов. При этом большие ограничения на операнд могут накладываться конфигурацией конкретной микропроцессорной системы.

Анализ системы команд процессора К580ВМ80 [3] выдал следующие результаты:

– избыточность кода операции (в системе команд не используются следующие коды: 10h, 20h, 30h, 08h, 18h, 28h, 38h, D9h, CBh, DDh, EDh, FDh);

– формат команды процессора К580ВМ80 не содержит информационной избыточности ввиду отсутствия префиксов и служебных байтов;

– избыточность операнда определяется в основном конфигурацией конкретной микропроцессорной системы.

Результаты анализа системы команд процессора КМ1816ВЕ51 [4] показали, что:

– существует избыточность кода операции (в системе команд не используется код A4h);

– формат команды процессора КМ1816ВЕ51 не содержит информационной избыточности ввиду отсутствия префиксов и служебных байтов;

– избыточность операнда определяется в основном конфигурацией конкретной микропроцессорной системы.

Создание методов обеспечения безопасности, сохранности и достоверности информации тесно связано со статистическим и вероятностным анализом передаваемых данных, описываемых каким-либо языком, обладающим, как правило, внутренними зависимостями, которые ведут к возникновению существенной избыточности информации, являющейся слабым местом в системе обеспечения информационной безопасности.

В процессе работы создано соответствующее программное обеспечение, с помощью которого проведен статистический анализ исполняемых модулей операционных систем DOS и Windows.

Все команды по вероятности появления можно разделить на следующие группы:

– команды с высокой вероятностью появления;

– команды со средней вероятностью появления;

– команды с низкой вероятностью появления;

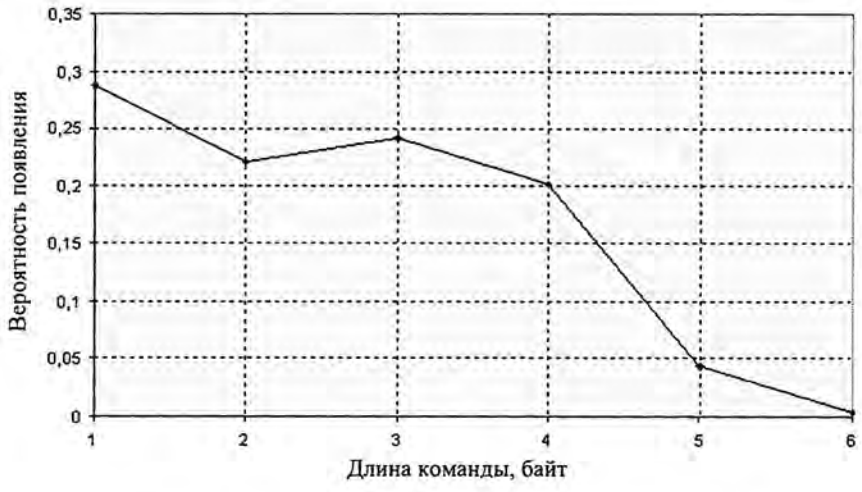
– команды с очень низкой вероятностью появления.

К командам с высокой вероятностью появления относятся команды со следующими кодами операций: 26, 8E, 8B, 75, 74, FF, 83. Эти коды принадлежат ряду команд перемещения MOV, ко-

мандам передачи управления JMP и командам, содержащим код операции во втором байте. Среднюю вероятность появления имеют команды групп 0X, 5X, 8X. В основном это команды работы с регистрами процессора (арифметические и логические операции, команды перемещения и обмена данными) и команды работы со стеком PUSH и POP, необходимые для обращения к подпрограммам.

К командам с очень низкой вероятностью появления относятся: 12, 14, E1, F1, D4, D5, D6, E6, 6E, 65, A6, A7, DA, 9B, BD, DD, ED, CE, 6F, AF, DF. Это либо коды несуществующих операций, либо такие малоиспользуемые команды, как LOOP, LOCK, STD, ESC, SCASD. Остальные команды имеют примерно одинаковую низкую вероятность появления.

Вероятности появления команд в зависимости от их длины приведены на рисунке и в таблице. Легко видеть, что наибольшие вероятности появления имеют одно-, двух- и трехбайтные команды, так как они составляют наибольшую часть системы команд.



Зависимость вероятности появления команд от длины

Вероятности появления команд

Длина команды, байт	1	2	3	4	5	6
Вероятность появления команды	0,2868	0,22067	0,24185	0,20147	0,04389	0,00341

Очевидно, что кроме структуры языка и типа программы, существенное влияние на статистическое распределение команд оказывает компилятор, с помощью которого был получен исполняемый код. Логично предположить, что именно системой генерации кода можно объяснить существование малой группы команд с высокой вероятностью появления. Для исследования влияния компилятора на исполняемый код написана программа, вычисляющая все возможные перестановки из заданных чисел на языках Borland C и Borland Pascal. С помощью указанных компиляторов получены исполняемые коды и проведен их статистический анализ. В целом вероятности появления команд в кодах, полученных с помощью разных компиляторов примерно одинаковы. Отличия компилятора Borland Pascal заключаются в более частом использовании команд групп 2X (арифметические операции с регистрами), 5X (команды работы со стеком), AX.

В перспективе планируется исследовать другие компиляторы и провести статистический анализ взаимного расположения команд в исполняемом коде.

Очевидно, что при преобразовании машинных кодов программ необходимо избавиться от существующей информационной избыточности. С целью защиты от анализа и открытия программ необходимо разрабатывать индивидуальный метод устранения избыточности для каждой конкретной группы команд. Реализация такого метода является относительно трудоемкой ввиду индивидуальности подхода к каждой группе команд и необходимости модификации алгоритма в зависимости от конфигурации системы. Вместе с тем метод дает очень хорошие результаты противодействия анализу с целью раскрытия.

Список литературы

1. Шеннон К. Работы по теории информации и кибернетике. М.: Изд-во иностр. лит., 1963.

2. Чекатков А. А. Использование Turbo Assembler при разработке программ. Киев: Диалектика, 1995.

3. Соучек Б. Микропроцессоры и микро-ЭВМ: Пер. с англ. / Под ред. А. И. Петренко. М.: Сов. радио, 1979.

4. Сташин В. В., Урусов А. В., Мологонцева О. Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах. М.: Энергоатомиздат, 1990.

Statistic Analysis of Software Modules

D. Yu. Bornin, A. A. Kurochkina, D. B. Nikolayev

The problem of safe and reliable transmission of data, featuring inherent dependability, as related to characteristics of the data transmitted, is considered. The results of analysis of software modules, supporting micro-processor devices based on different hardware and software platforms are presented.