

Исследование методов генерации защищенного машинного кода

Диссертация на соискание степени магистра

Выполнил студент гр. 63601/2:

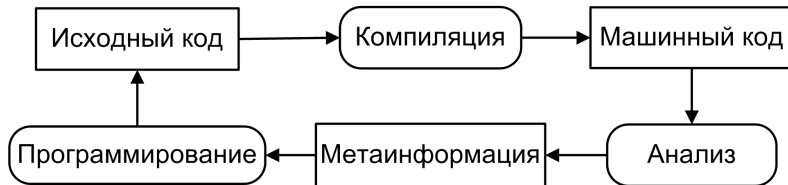
Научный руководитель:

А.Б. Терентьев

В.Ю. Аранов

26 июня 2014

Проблема обратного проектирования программ



Метаинформация из скомпилированного кода может быть
Извлечена → Проанализирована → Модифицирована

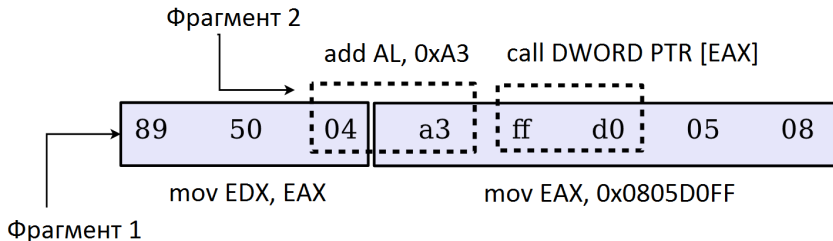
«Взломоустойчивость — число условных «человеко-часов», которые необходимо потратить для извлечения метаинформации достаточной для организации атаки¹»

- Устойчивость к различным методам обратного проектирования программы
- Степень потери производительности программы по времени выполнения и использованной динамической памяти
- Временная трудоемкость метода построения защиты

¹C. Collberg, *A Taxonomy of Obfuscating Transformations*, 1997

Код с перекрывающимися инструкциями

«Перспективным и заманчивым представляется также создание перекрывающихся инструкций²»



Байты по одному и тому же адресу используются в нескольких инструкциях

²С.И. Алейников, *Защита программ от дизассемблирования*, 2006

- Разработка подходов к генерации кода с перекрывающимися инструкциями
- Разработка критерия оценки качества кода, сгенерированного с помощью этих подходов
- Реализация подходов для платформы x86
- Оценка качества генерируемого защищенного кода

Гаджет — последовательность инструкций, содержащая ровно одну инструкцию передачи управления, находящуюся в конце

```
f7 c7 07 00 00 00  
0f 95 45 c3
```

```
test EDI, 0x00000007  
setnz EBP
```

```
c7 07 00 00 00 0f  
95  
45  
c3
```

```
mov EDI, 0x0F000000  
xchg EBP, EAX  
inc EBP  
ret
```

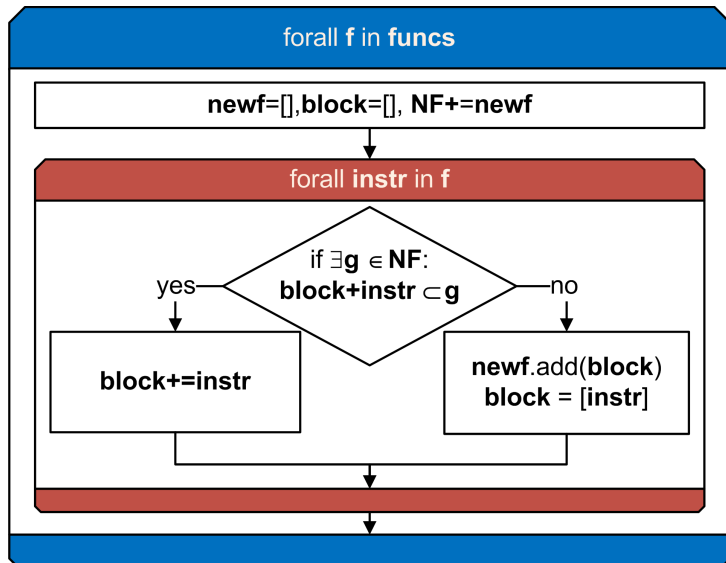
Алгоритм

Вход:

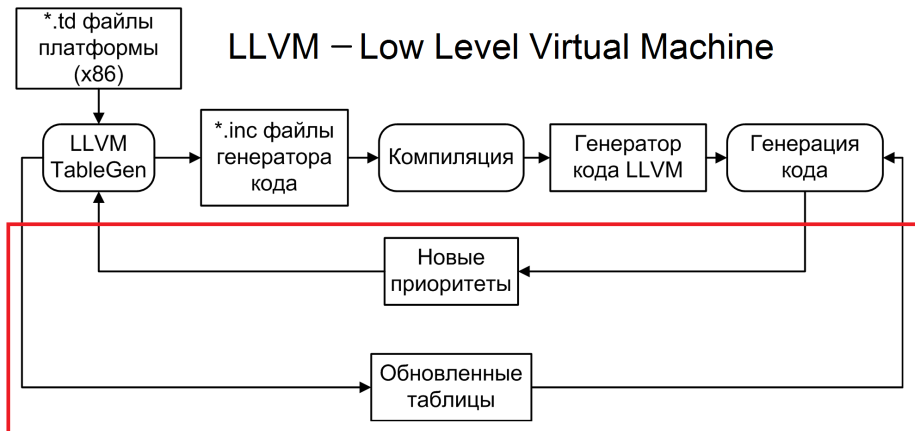
funcs

Выход:

NF



LLVM – Low Level Virtual Machine



Временная сложность в худшем случае — $O(n^2)$

n - размер программы в байтах

- Дано:

- ▶ Код программы P (немодифицированный генератор кода LLVM3.3), размер n
- ▶ Код программы Q (модифицированный генератор кода LLVM), размер m , $x_i, i = 1..m$ - значения перекрываемости байт (количество инструкций, элементом которых является i -ый байт)
- ▶ $W_P(Q)$ - искомая функция

- Требования:

- ▶ $W_P(P) \equiv 1$
- ▶ Учитывать изменение размера:
 - ★ $\forall Q : m = n, x_i = 1, i = 1..m : W_P(Q) \equiv 1;$
 - ★ $\forall Q : m < n, x_i = 1, i = 1..m : W_P(Q) > 1;$
 - ★ $\forall Q : m > n, x_i = 1, i = 1..m : W_P(Q) < 1$
- ▶ $\forall Q : m = n, x_i = N, i = 1..m : W_P(Q) \equiv N$

- Цель: $W_P(Q) > 1$

- Средняя перекрываемость $A\{x_1..x_k\}$: $V(A) = \frac{1}{k} \sum_{i=1}^k x_i$

- Удельная средняя перекрываемость:

$$V_{spec}(A) = \frac{V(A)}{k} = \frac{1}{k} \sum_{i=1}^k x_i / k$$

- f для программ A, B : $f(A, B) = \frac{V_{spec}(A)}{V_{spec}(B)}$

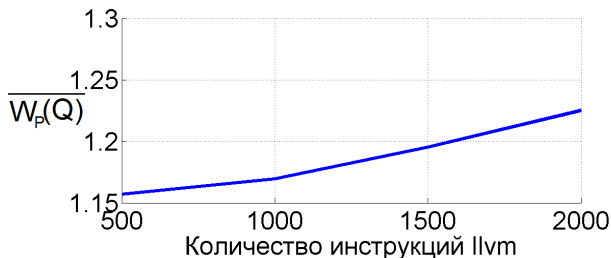
- Для программы P : $x_i \equiv 1$, $V_{spec}(P) = \frac{1}{n}$:

$$W_P(Q) = f(Q, P) = \frac{\left(\sum_{i=1}^m x_i\right) * n}{m^2}$$

Оценка качества кодогенерации

llvm-stress — генерация случайных .ll файлов.

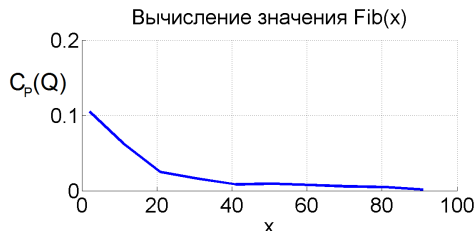
50 файлов на тест, 2000 инструкций ~ 6КБ(размер секций .text)



Размер	$\overline{W_p(Q)}$	σ	MIN	MAX
500	1.116	0.0156	1.017	1.753
1000	1.169	0.0113	1.006	1.614
1500	1.195	0.0046	1.084	1.376
2000	1.226	0.0085	1.113	1.450

Исследование производительности

- $C_P(Q) = \frac{C(Q) - C(P)}{C(P)}$, $C(A)$ — количество тактов процессора, необходимое для запуска A
- $T_P(Q) = \frac{T(Q) - T(P)}{T(P)}$, $T(A)$ — время, необходимое для запуска A
- 100000 вызовов каждой функции, CPU — Core i7 2600K



Тест	$T(P)$, с	$T(Q)$, с	$T_P(Q) * 100, \%$	$C_P(Q) * 100, \%$
Sin	36.149	36.044	-0.29%	+0.07%
Fact	3.334	3.477	+4.2%	+4.5%
Fib	3.211	3.301	+2.8%	+2.7%

- Затрудняет анализ и модификацию машинного кода
- Затрудняет использование стандартных программ для анализа — IDA, Hex-Rays, OllyDbg
- Может использоваться в комбинации с другими методами защиты машинного кода
- Иногда:
 - ▶ Модифицированная программа меньше оригинальной
 - ▶ Модифицированная программа быстрее оригинальной

- Разработаны новые подходы к генерации машинного кода с перекрывающимися инструкциями
- Подходы реализованы на основе генератора кода LLVM
- Разработана методика оценки результатов работы
- На основе предложенного критерия и результатов исследований производительности защищенного кода сделаны выводы о сравнительной эффективности разработанных алгоритмов

Доклад + статья на Spring/Summer Young Researchers' Colloquium on Software Engineering, 2014.

Направление дальнейших исследований

- Алгоритм Витерби

	ASSEMBLY	OPCODE	ASCII
1	<pre>push %esp push \$20657265 imul %esi,20(%ebx), \$616D2061 push \$6F jb short \$22</pre>	<pre>54 68 65726520 6973 20 61206D61 6A 6F 72 20</pre>	There is a major
2	<pre>push \$20736120 push %ebx je short \$63 jb short \$22</pre>	<pre>68 20617320 53 74 61 72 20</pre>	h as Star
1	Skip	2	Skip
There is a major center of economic activity, such as Star Trek, including The Ed			
Skip	3	Skip	
Sullivan Show. The former Soviet Union. International organization participation			

Средняя перекрываемость байт

Размер	$\overline{V(Q)}$	MAX	MIN
500	1.24	1.31	1.18
1000	1.28	1.41	1.21
1500	1.30	1.41	1.24
2000	1.33	1.49	1.25

Процент байт с перекрываемостью > 1

Размер	AVG	MAX	MIN
500	12.9	17.2	8.9
1000	13.6	16.5	9.8
1500	12.8	15.7	10.2
2000	13.1	15.6	11.6

Low Level Virtual Machine (LLVM)

Универсальная система анализа, трансформации и оптимизации программ, реализующая виртуальную машину

