ИСП РАН

Поиск ошибок выхода за границы буфера в бинарном коде программ

Вадим Каушан

2 декабря 2016 г.

План

1. Метод поиска ошибок выхода за границы буфера
2. Проблемы
3. Анализ циклов
4. Расширение покрытия кода

Метод поиска ошибок выхода за границы буфера

- Анализируется трасса выполнения, соответствующая одному запуску программы
- Инструкции трассы транслируются в промежуточное представление, а затем в уравнения для SMT решателя
- Каждому выделяемому буферу или буферу с данными ставится в соответствие символьная длина
- При каждом обращении к памяти проверяются условия выхода за границы буфера
  - Отдельные инструкции
  - Функции работы с буферами (`memcpy`, `strcpy`, ...)

Символьная интерпретация длин буферов

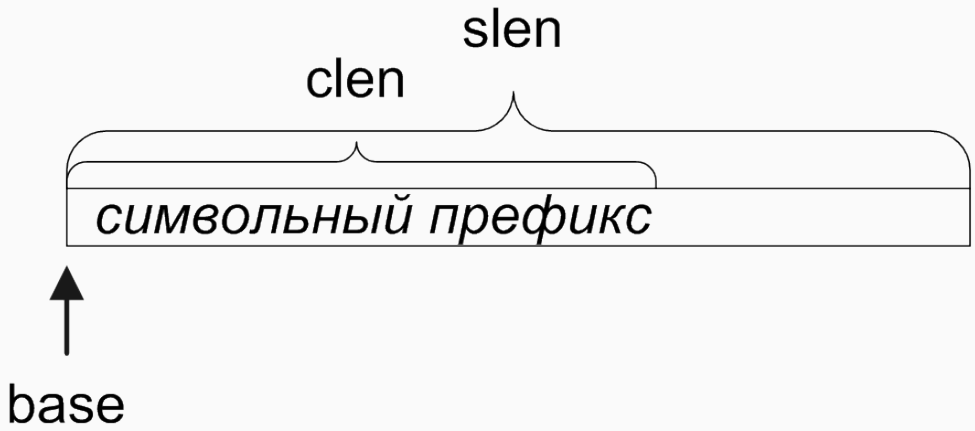
L-буфер представляет собой тройку

$$L = \{base, clen, slen\}$$

base – адрес базы,

clen – конкретная длина,

slen – выражение над символьными переменными, описывающее абстрактную длину буфера



Символьная интерпретация длин буферов

$$\frac{\langle \text{memcpy}, \{\text{dest.addr}, \text{src.addr}, \text{n}\}, \{\}\rangle}{\begin{array}{c} i_{src} = T\big(I, M(src.addr)\big), i_{dst} = T\big(I, M(dst.addr)\big), \\ i' = [\![M(dst.addr), M(n), i_{src}\Delta(n)]\!], \\ Assert\big(i_{src}.slen - (src.addr - i_{src}.base) < \Delta(n)\big) \\ a_{src} = T\big(A, M(src.addr)\big),\ Assert(i_{src} \sqsubset a_{src}) \\ a = T\big(A, M(dst.addr)\big), Assert(i' \sqsubset a) \end{array}}$$

$$\Delta, I \vdash \Delta\left[m_{i'.base}, .., m_{i'.base+i'.clen-1} \leftarrow m_{M(src.addr)}, .., m_{M(src.addr)+i'.clen-1}\right], I \cup i'$$

Проблемы

Циклы, работающие с данными, накладывают ограничения на их размер

Эти ограничения приводят к потенциальным пропускам ошибок. Поэтому необходимо анализировать циклы целиком.

Анализируется только один запуск программы

В рамках одного запуска анализируется ограниченный набор участков программы. Для улучшения качества анализа необходимо анализировать несколько запусков программы с существенно большим покрытием кода.

Анализ циклов

Анализ отдельных инструкций цикла приводит к добавлению неявных ограничений на длину строк:

```
REPNZ SCASB AL, [EDI] ; [8001F35F]!=0
REPNZ SCASB AL, [EDI] ; [8001F360]!=0
REPNZ SCASB AL, [EDI] ; [8001F361]!=0
REPNZ SCASB AL, [EDI] ; [8001F362]!=0
REPNZ SCASB AL, [EDI] ; [8001F363]==0
```

Добавляется ограничение $str.slen == 4$, которое приводит к отсутствию срабатывания на проверках вида $str.slen > 1024$.

Анализ циклов

- Рассматриваются два вида циклов
  - Циклы копирования строк
  - Циклы вычисления длины строк
- Каждый из найденных циклов описывается как единая сущность с параметрами
- В процессе анализа циклы обрабатываются целиком по аналогии с вызовами функций `strcpy`, `strlen`

Восстановление циклов работы со строками

Алгоритм поиска циклов работы со строками

- Определение индуктивных переменных цикла
- Определение буферов, к которым осуществлялся последовательный доступ
- Определение эффекта циклов по обрабатываемым буферам и характеру доступа, определение параметров обработки
- Сохранение результатов

Расширение покрытия кода

- Используется критерий покрытия по базовым блокам
- Перебор путей с помощью динамического символьного выполнения
  - Для каждого пути сохраняется набор входных данных, а также полученное покрытие
- Минимизация покрывающего набора входных данных
  - Жадный алгоритм для задачи покрытия множества
- Реализация на основе S2E

Расширение покрытия кода

Программа	Пути		Прирост покрытия
	Всего	Отобранные	
iwconfig	1276	3	25.4%
mkfs.jfs	9391	11	33.3%
grep	1845	9	16.3%
pkill	257	15	371.9%
objdump	292	35	183.7%
echo	8	5	1.3%

Спасибо за внимание!