Методика восстановления формата данных

Гетьман А.И., Падарян В.А.
{thorin, vartan}@ispras.ru
Институт системного программирования РАН
http://www.ispras.ru

РусКрипто'2010, 3 апреля 2010 г.

Задача восстановления формата

- Рассматривается обмен сообщениями между двумя процессами в рамках некоторого протокола
- Обмен может включать работу по сети или обработку данных в файле; файл, в этом случае, рассматривается как совокупность сообщений
- Сообщение представляет собой байтовый буфер известного размера

Основные подходы

- Анализ вводимых (выводимых) данных с выделением повторяющихся паттернов

  Недостаток: сложность определения семантики полей
- Анализ кода клиента или сервера

  - статический

  - **динамический** (отслеживается процесс разбора или создания сообщения)

Цели работы

- Предоставление информации об иерархической структуре сообщения
- Анализ реализаций протоколов
- Применение восстановленного формата для автоматического разбора сообщений

Формат сообщения

- Границы отдельных полей сообщения, содержащих базовые типы данных
- Группировка полей в структуры и последовательности
- Вариации сообщений одного типа (например наличие или отсутствие некоторого «плавающего» поля)
- Семантика полей (например поля длины, задающие размер последовательностей)
- Информация о значениях полей, входящих в сообщение.

Методика восстановления

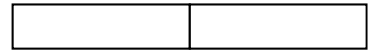
- Выделение в трассе кода обработки сообщения
  прямой слайсинг
- Восстановление полей по размерам операндов инструкций доступа к данным буфера, включая разрешение коллизий
- Группировка полей в структуры и последовательности на основе анализа CFG
- Анализ семантики полей на основе анализа инструкций сравнения и условных переходов
  - поиск полей длины выделенных последовательностей
  - поиск полей-разделителей выделенных последовательностей
  - поиск полей-указателей
  - поиск полей, содержащих ключевые значения
  - поиск полей-флагов
- Построение и обобщение дерева формата, выявление вариативности в структуре сообщения

Определение границ полей

- Анализ размеров операндов инструкций доступа к буферу (в битах)

Преодоление неоднозначности

- Инструкции типа *mov* не рассматриваются
- Решение на основе оптимального по весу покрытия. Вес поля – количество инструкций доступа.




```
mov eax, byte ptr [addr]
mov eax, byte ptr [addr+1]
…
mov ebx, word ptr [addr]
```

Пример неоднозначности разбиения на поля

Выделение структур и последовательностей

Эвристические предположения

- Последовательности обрабатываются в циклах
- Если цикл состоит из структур, то каждая структура обрабатывается на отдельной итерации цикла

Способы задания длины последовательности

- С помощью поля-разделителя (например, нуль-терминированная строка).
- С помощью поля длины, хранящего значение, на основе которого можно вычислить длину последовательности
- Длина фиксирована в протоколе

Особенности подхода

- Битовая гранулярность
- Источник данных – трасса
  CFG восстанавливается по трассе

Ограничения

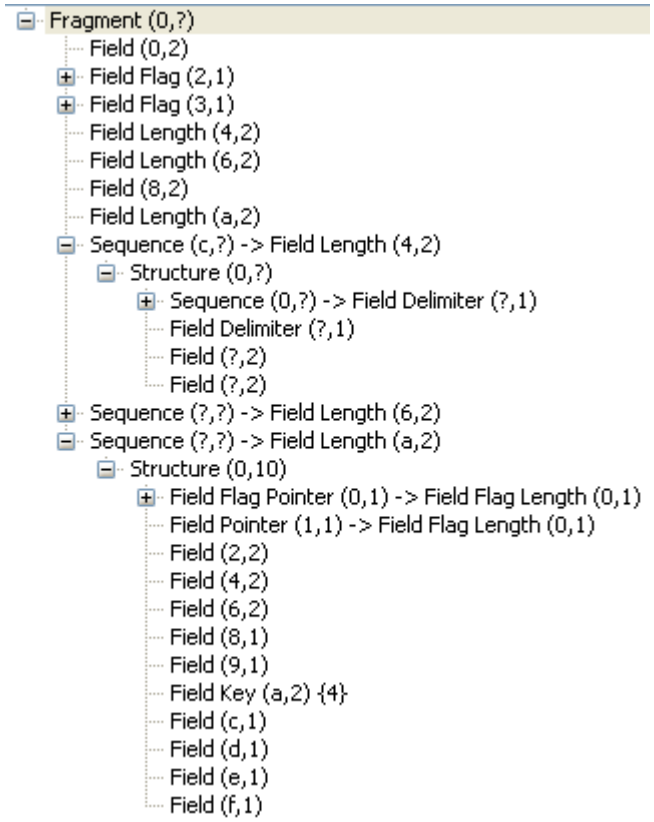
- Анализу доступны только экземпляры сообщения попавшие в трассу
- Наличие оптимизаций разворота циклов

Методы преодоления

- Уточнение формата на основе анализа нескольких трасс
- Автоматический поиск развёрнутых циклов в трассе

Пример восстановления DNS сообщения

Спецификация DNS по RFC-1035



Фрагмент восстановленной спецификации