Диаген.

Метод динамического контроля выполнения программы

Е.В. Маньков
ООО «Газинформсервис»
в соавторстве с Компанийцем Р.И. и Ковалевым В.В.

31.03.2011 Солнечногорск, МО

Малоосвещаемые вопросы ИБ функционирования ПО

Непредумышленные:

- дефекты программирования и используемого ПО
- ошибки бизнес логики
- недокументированные возможности (НДВ)

Предумышленные:

- модификации исполняемого кода (в памяти)
- атаки на исполняемый код (в памяти)

Технология ИРИДА

1. Дизассемблирование исполняемых модулей
2. «Склейка»: формирование общего адресного пространства, восстановление межмодульных связей (таблицы экспорта-импорта)
3. Построение модели программы (УГП всех подпрограмм)
4. Анализ статических маршрутов в подпрограмме, выявление структуры (управляющие конструкции) и нарушений структурированности
5. Формирование множества (доверенных) маршрутов – установка контрольных точек:
   - на вызовы подпрограмм (все, выборочные) – задача контроля/тестирования;
   - на передачи управления (например, в областях с нарушением структурированности) – задача контроля/тестирования;
   - на интересующие (критические по ИБ) участки кода – задача защиты от модификаций/атак;

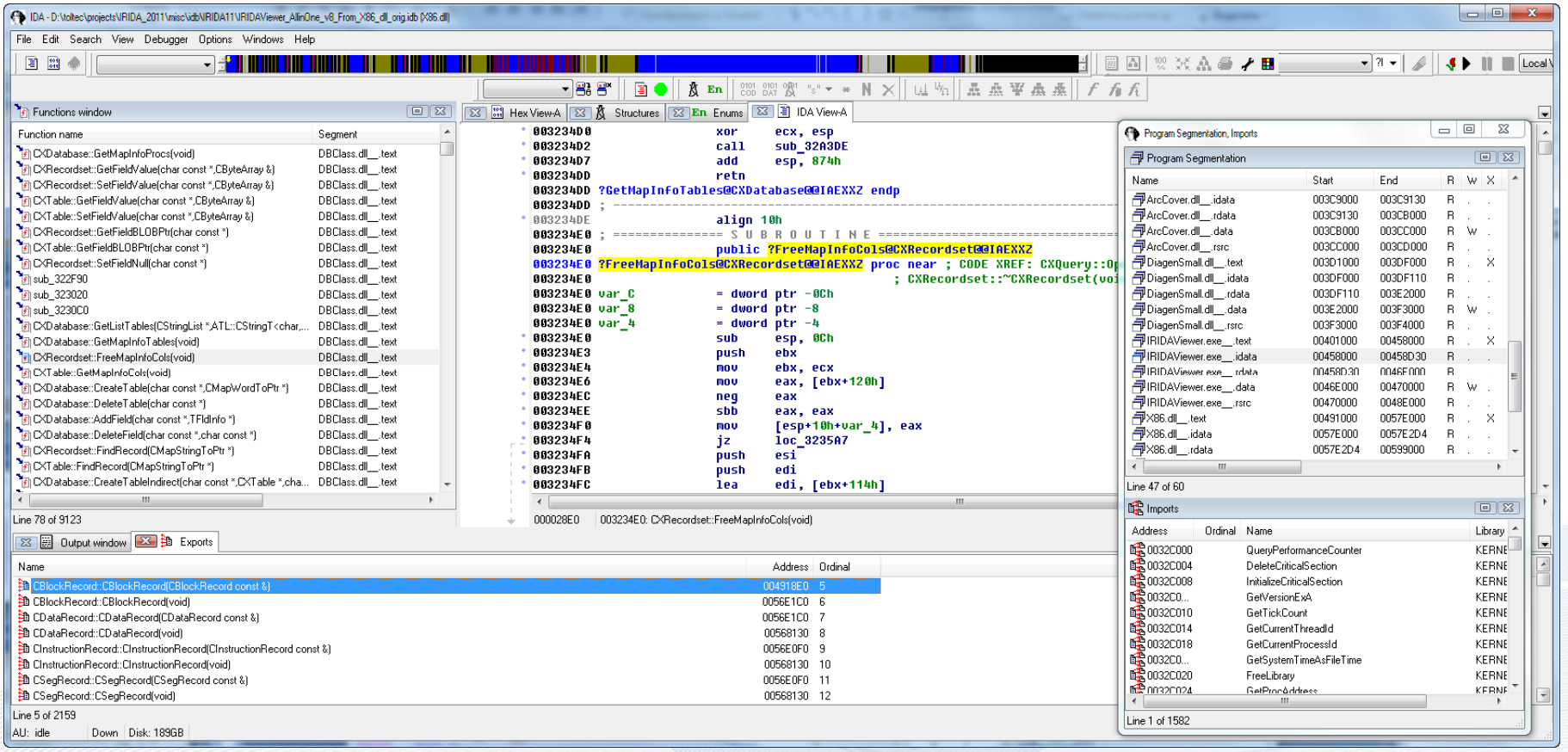
Технология ИРИДА

6. Создание грамматики описания множества (доверенных) маршрутов
7. Внедрение контрольных точек в контролируемую/защищаемую программу
8. **Генерация паспорта динамического контроля для исполняемого кода программы**
9. Динамический анализ маршрутов выполнения – задача контроля/тестирования
10. Сопоставление результатов СА и ДА, восстановление косвенных передач управления.
11. Уточнение и изменение набора контрольных точек и выполнение шагов 6-7

    или результирующий паспорт: шаг 8
12. **Контроль выполнения программы**

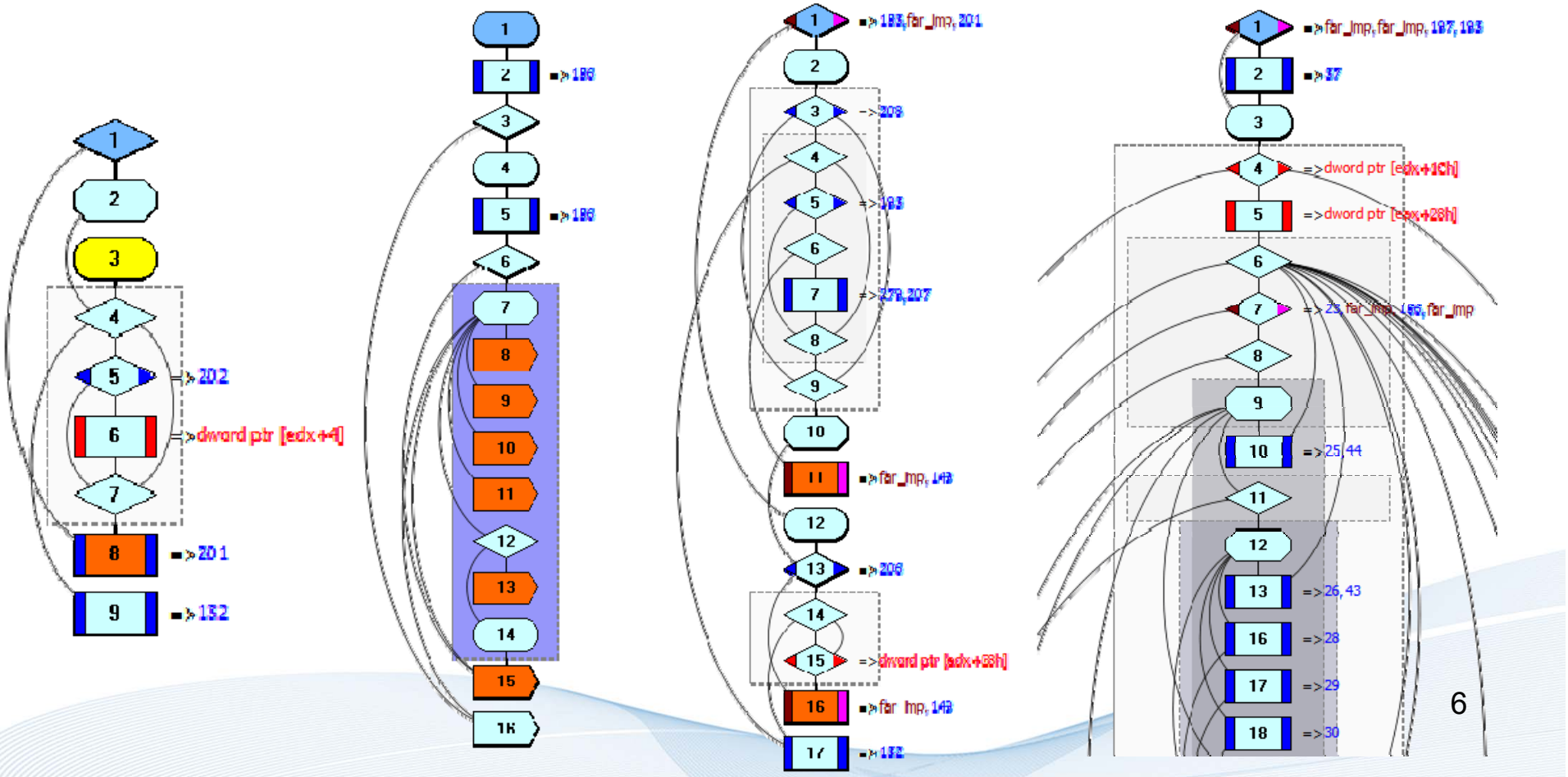
Дизассемблирование, «Склейка»

1. Дизассемблирование исполняемых модулей (IDA Pro).
2. «Склейка»: формирование общего адресного пространства, восстановление межмодульных связей (таблицы экспорта-импорта) –разработанный плагин AllinOne под IDA Pro формирования единой idb.



УГП - модели, управляющие структуры

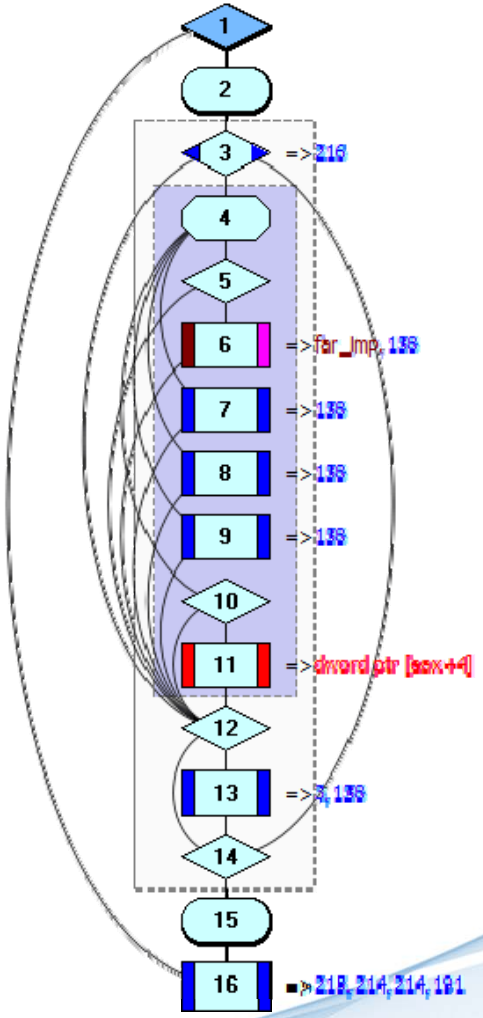
3. Построение модели программы (УГП всех подпрограмм).
4. Анализ статических маршрутов в подпрограмме, выявление структуры (управляющие конструкции) и нарушений структурированности (4-й граф).



УГП - модели, управляющие структуры

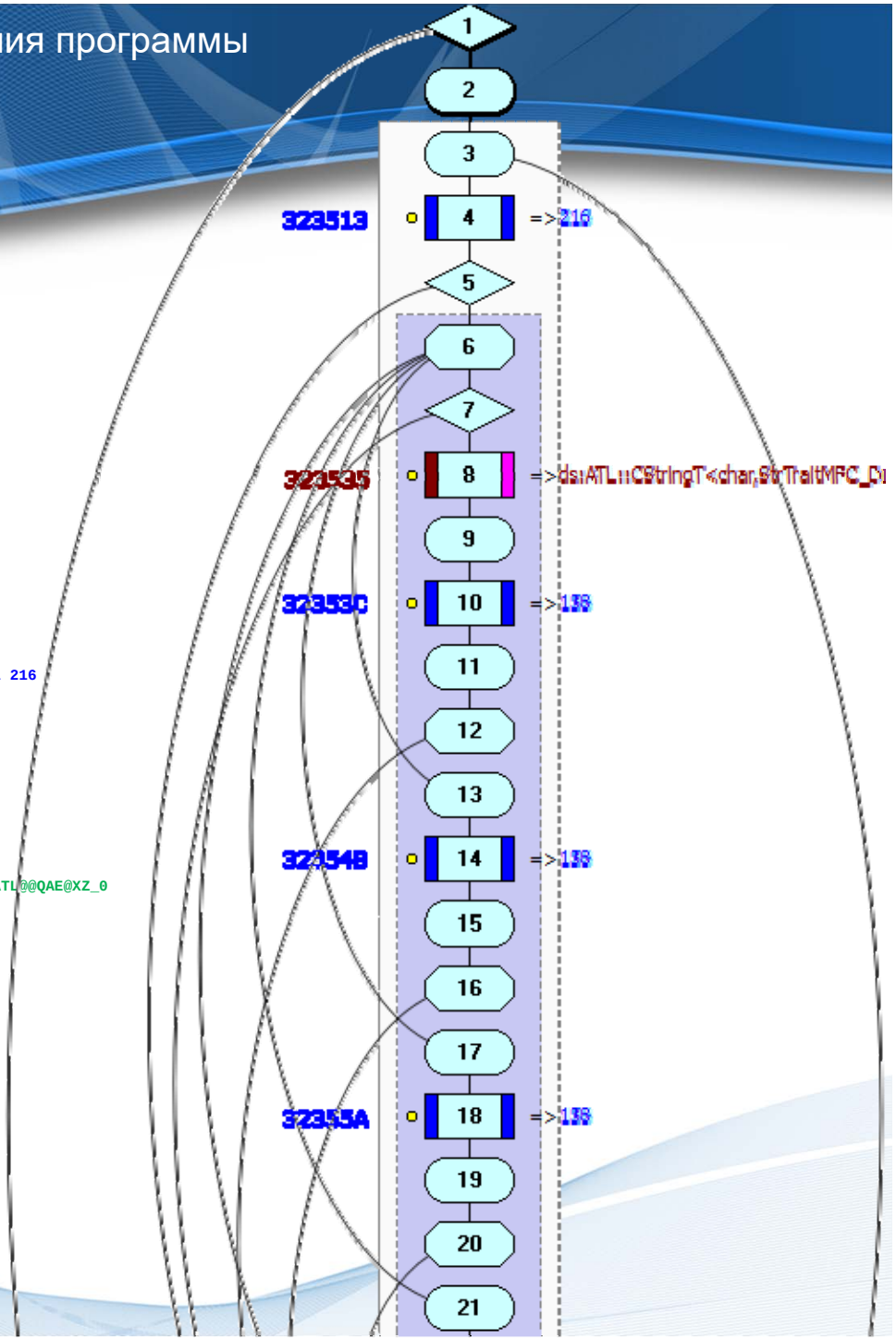
Пример подпрограммы на C++, дизассемблированный код и модель IDA

```
void CXRecordset::FreeMapInfoCols()
{
    POSITION CurPos;
    void * pVoid, * pValue;
    for(CurPos=m_oMapMemElements.GetStartPosition();CurPos!=NULL;)
    {
          m_oMapMemElements.GetNextAssoc(CurPos,pVoid,pValue);
          switch(((TFldInfo *)pVoid)->nFldType)
          {
                case FTString:
                case FTMemo:
                      delete (CString*)pValue;
                      break;
                case FTInteger:
                      delete (int*)pValue;
                      break;
                case FTLong:
                case FTCounter:
                      delete (long int*)pValue;
                      break;
                case FTSingle:
                      delete (float*)pValue;
                      break;
                case FTDouble:
                      delete (double*)pValue;
                      break;
                case FTBool:
                      delete (BOOL*)pValue;
                      break;
                case FTBLOB:
                      delete (CByteArray*)pValue;
          };
          delete (TFldInfo *)pVoid;
     };
     m_oMapMemElements.RemoveAll();
     oInfoColList.RemoveAll();
     oValueColList.RemoveAll();
     oColList.RemoveAll();
}
```



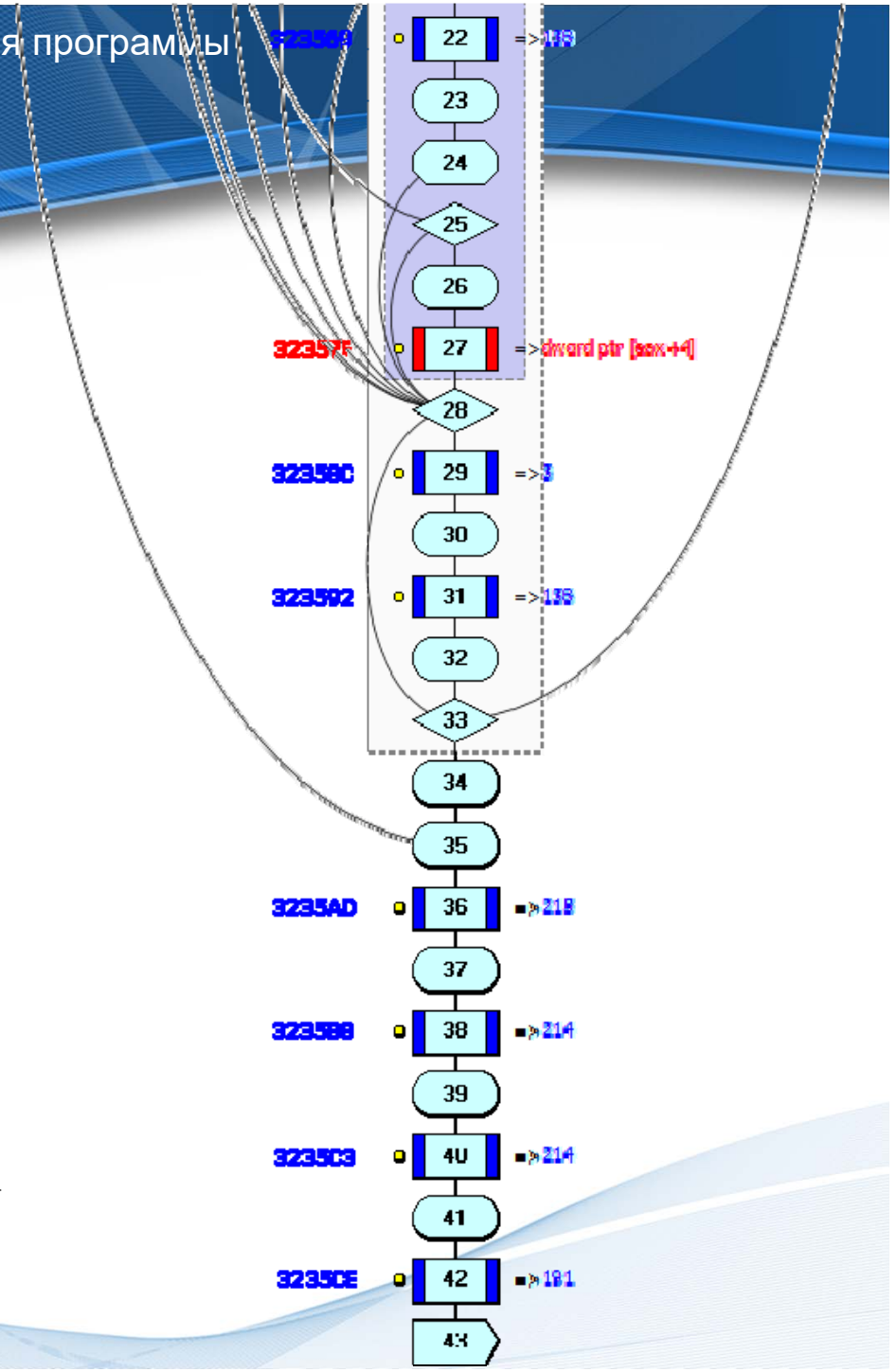
5. Установка контрольных точек

```
003234E0 ; ============== S U B R O U T I N E =======================================
003234E0 ; --BLOCK--1----------------------------------------------------------------
003234E0                 public ?FreeMapInfoCols@CXRecordset@@IAEXXZ
003234E0 ?FreeMapInfoCols@CXRecordset@@IAEXXZ proc near
003234E0 var_C           = dword ptr -0Ch
003234E0 var_8           = dword ptr -8
003234E0 var_4           = dword ptr -4
003234E0                 sub     esp, 0Ch
003234E3                 push    ebx
003234E4                 mov     ebx, ecx
003234E6                 mov     eax, [ebx+120h]
003234EC                 neg     eax
003234EE                 sbb     eax, eax
003234F0                 mov     [esp+10h+var_4], eax
003234F4                 jz      loc_3235A7
003234FA ; --BLOCK--2----------------------------------------------------------------
003234FA                 push    esi
003234FB                 push    edi
003234FC                 lea     edi, [ebx+114h]
00323502 ; --BLOCK--3----------------------------------------------------------------
00323502 loc_323502:     lea     eax, [esp+18h+var_C]
00323506                 push    eax
00323507                 lea     ecx, [esp+1Ch+var_8]
0032350B                 push    ecx
0032350C                 lea     edx, [esp+20h+var_4]
00323510                 push    edx
00323511                 mov     ecx, edi
00323513 ; --BLOCK--4----------------------------------------------------------------
00323513                 call    ?GetNextAssoc@CMapPtrToPtr@@QBEXAAPAU__POSITION@@AAPAX1@Z ; call 216
00323518 ; --BLOCK--5----------------------------------------------------------------
00323518                 mov     eax, [esp+18h+var_8]
0032351C                 mov     eax, [eax+4]
0032351F                 cmp     eax, 8
00323522                 ja      short loc_323582
00323524 ; --BLOCK--6----------------------------------------------------------------
00323524                 jmp     ds:off_3235D8[eax*4]
0032352B ; --BLOCK--7----------------------------------------------------------------
0032352B loc_32352B:     mov     ecx, [esp+18h+var_C]
0032352F                 test    ecx, ecx
00323531                 mov     esi, ecx
00323533                 jz      short loc_323582
00323535 ; --BLOCK--8----------------------------------------------------------------
00323535                 call    ds:??1?$CStringT@DV?$StrTraitMFC_DLL@DV?$ChTraitsCRT@D@ATL@@@@@ATL@@QAE@XZ_0
0032353B ; --BLOCK--9----------------------------------------------------------------
0032353B                 push    esi
0032353C ; --BLOCK--10---------------------------------------------------------------
0032353C                 call    ??3@YAXPAX@Z_0                          ; call 138
00323541 ; --BLOCK--11---------------------------------------------------------------
00323541                 add     esp, 4
00323544 ; --BLOCK--12---------------------------------------------------------------
00323544                 jmp     short loc_323582
00323546 ; --BLOCK--13---------------------------------------------------------------
00323546 loc_323546:     mov     ecx, [esp+18h+var_C]
0032354A                 push    ecx
0032354B ; --BLOCK--14---------------------------------------------------------------
0032354B                 call    ??3@YAXPAX@Z_0                          ; call 138
00323550 ; --BLOCK--15---------------------------------------------------------------
00323550                 add     esp, 4
00323553 ; --BLOCK--16---------------------------------------------------------------
00323553                 jmp     short loc_323582
00323555 ; --BLOCK--17---------------------------------------------------------------
00323555 loc_323555:     mov     edx, [esp+18h+var_C]
00323559                 push    edx
0032355A ; --BLOCK--18---------------------------------------------------------------
0032355A                 call    ??3@YAXPAX@Z_0                          ; call 138
0032355F ; --BLOCK--19---------------------------------------------------------------
0032355F                 add     esp, 4
00323562 ; --BLOCK--20---------------------------------------------------------------
00323562                 jmp     short loc_323582
00323564 ; --BLOCK--21---------------------------------------------------------------
00323564 loc_323564:     mov     eax, [esp+18h+var_C]
00323568                 push    eax
```



УГП – модель п/п

```
00323568 ; --BLOCK--22----------------------------------------------------------
00323569                 call    ??3@YAXPAX@Z_0                          ; call 138
0032356E ; --BLOCK--23----------------------------------------------------------
0032356E                 add     esp, 4
00323571 ; --BLOCK--24----------------------------------------------------------
00323571                 jmp     short loc_323582
00323573 ; --BLOCK--25----------------------------------------------------------
00323573 loc_323573:     mov     ecx, [esp+18h+var_C]
00323577                 test    ecx, ecx
00323579                 jz      short loc_323582
00323579 ; --BLOCK--26----------------------------------------------------------
0032357B                 mov     eax, [ecx]
0032357D                 push    1
0032357F ; --BLOCK--27----------------------------------------------------------
0032357F                 call    dword ptr [eax+4]
00323582 ; --BLOCK--28----------------------------------------------------------
00323582 loc_323582:     mov     ecx, [esp+18h+var_8]
00323586                 test    ecx, ecx
00323588                 mov     esi, ecx
0032358A                 jz      short loc_32359A
0032358C ; --BLOCK--29----------------------------------------------------------
0032358C                 call    ??1TFldInfo@@QAE@XZ                     ; call 3
00323591 ; --BLOCK--30----------------------------------------------------------
00323591                 push    esi
00323592 ; --BLOCK--31----------------------------------------------------------
00323592                 call    ??3@YAXPAX@Z_0                          ; call 138
00323597 ; --BLOCK--32----------------------------------------------------------
00323597                 add     esp, 4
0032359A ; --BLOCK--33----------------------------------------------------------
0032359A loc_32359A:     cmp     [esp+18h+var_4], 0
0032359F                 jnz     loc_323502
003235A5 ; --BLOCK--34----------------------------------------------------------
003235A5                 pop     edi
003235A6                 pop     esi
003235A7 ; --BLOCK--35----------------------------------------------------------
003235A7 loc_3235A7:     lea     ecx, [ebx+114h]
003235AD ; --BLOCK--36----------------------------------------------------------
003235AD                 call    ?RemoveAll@CMapPtrToPtr@@QAEXXZ
003235B2 ; --BLOCK--37----------------------------------------------------------
003235B2                 lea     ecx, [ebx+14Ch]
003235B8 ; --BLOCK--38----------------------------------------------------------
003235B8                 call    ?RemoveAll@CMapStringToPtr@@QAEXXZ
003235B8 ; --BLOCK--39----------------------------------------------------------
003235BD                 lea     ecx, [ebx+130h]
003235B8 ; --BLOCK--40----------------------------------------------------------
003235C3                 call    ?RemoveAll@CMapStringToPtr@@QAEXXZ
003235B8 ; --BLOCK--41----------------------------------------------------------
003235C8                 lea     ecx, [ebx+168h]
003235B8 ; --BLOCK--42----------------------------------------------------------
003235CE                 call    ?RemoveAll@CStringList@@QAEXXZ
003235B8 ; --BLOCK--43----------------------------------------------------------
003235D3                 pop     ebx
003235D4                 add     esp, 0Ch
003235D7                 retn
003235D7 ?FreeMapInfoCols@CXRecordset@@IAEXXZ endp
003235D8 ; --D A T A    B L O C K   1   (S W I T C H   1    MAP) ---------------
003235D8 off_3235D8      dd offset loc_32352B    ; DATA XREF: CXRecordset::FreeMapInfoCols(void)+44r
003235D8                 dd offset loc_32352B
003235D8                 dd offset loc_323546
003235D8                 dd offset loc_323555
003235D8                 dd offset loc_323555
003235D8                 dd offset loc_323564
003235D8                 dd offset loc_323546
003235D8                 dd offset loc_323555
003235D8                 dd offset loc_323573
003235FC                 align 10h
```



6. Формирование грамматики

Грамматика в терминах блоков

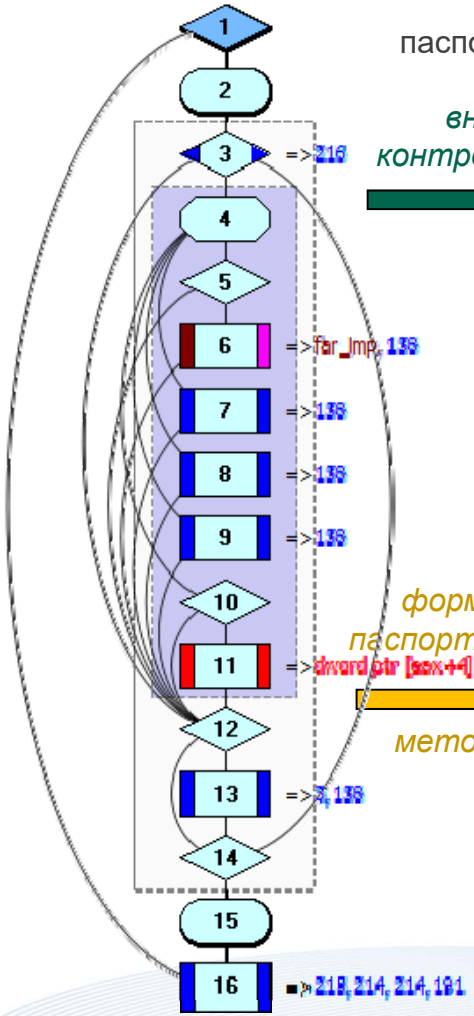
```
Route_78 :
( 4
      (
            8 10
          | 14
          | 18
          | 22
          | 27
      )?
      (29 31)
)*
(36 38 40 42) ;
```

Грамматика в терминах КТ (паспорт)

```
Route_78 :
(KT_323513 m4 KT_323513
      (
            KT_323535 m8 KT_323535
            KT_32353C m10 KT_32353C
          | KT_32354B m14 KT_32354B
          | KT_32355A m18 KT_32355A
          | KT_323569 m22 KT_323569
          | KT_32357F m27 KT_32357F
      )?
      (KT_32358C m29 KT_32358C
       KT_323592 m31 KT_323592)
)*
(KT_3235AD m36 KT_3235AD
KT_3235B8 m38 KT_3235B8
KT_3235C3 m40 KT_3235C3
KT_3235CE m42 KT_3235CE) ;
// адрес вызываемой ближней п/п-мы
m4 : {NEAR_CALL(0x00328513)} ;
// адрес вызываемой дальней п/п-мы
m8 : {FAR_CALL(00407000h)} ;
...
```

Механизм работы метода

Стадия формирование паспорта программы



внедрение контрольных точек

формирование паспорта программы

метод ДИАГЕН

Стадия выполнения паспортизированной программы

Исполняемый файл программы с внедренными контрольными точками

Прерывание по КТ

Возврат управления

Прерывание по КТ

Возврат управления

Прерывание по КТ

Останов

Библиотека паспорта программы

трасса

Попытка выполнения

недокументированный (недоверенный) маршрут

Диаген. Особенности

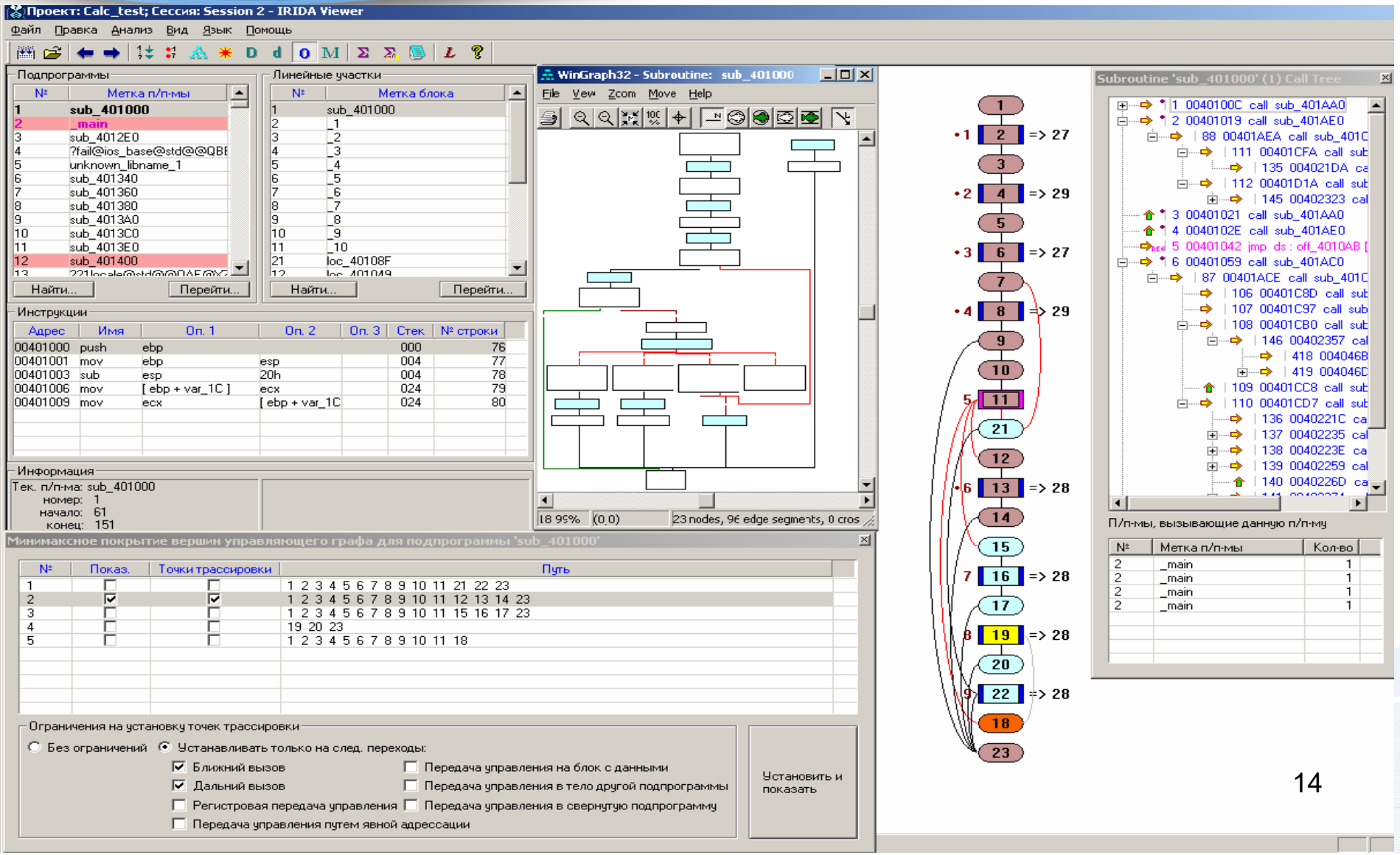
- Контролируемая программа «не знает» куда она передает управления в точках контроля!
- В точках контроля программа всегда вызывает одну и ту же функцию Hook без параметров, экспортируемую паспортом
- Функция Hook получает с вершины стека вызовов процедур адрес возврата – адрес контрольной точки и передает это значение на вход конечного автомата (КА) паспорта
- КА определяет по своему текущему состоянию допустимость нахождения контролируемой программы в данной точке в данный момент времени ее выполнения; в случае допустимости КА осуществляет вызов уже реальной подпрограммы и переходит в следующее свое состояние, либо прерывает выполнение контролируемой программы

Диаген. Особенности

- Таким образом, куда передавать управление сообщает программе ее паспорт на основании состояния своего КА
- Взламывать и/или атаковать (с целью изменения потока управления) целесообразно прежде всего модуль паспорта, а не контролируемую программу
- Для взлома паспорта необходимо решить задачу восстановления грамматики конечного автомата в терминах установленных контрольных точек
- Для усложнения взлома конечный автомат может быть обфускирован, зашифрован и привязан к токену
- Паспорт может выполняться на доверенной системе, в отличие от контролируемой программы

Диаген. Метод динамического контроля выполнения программы

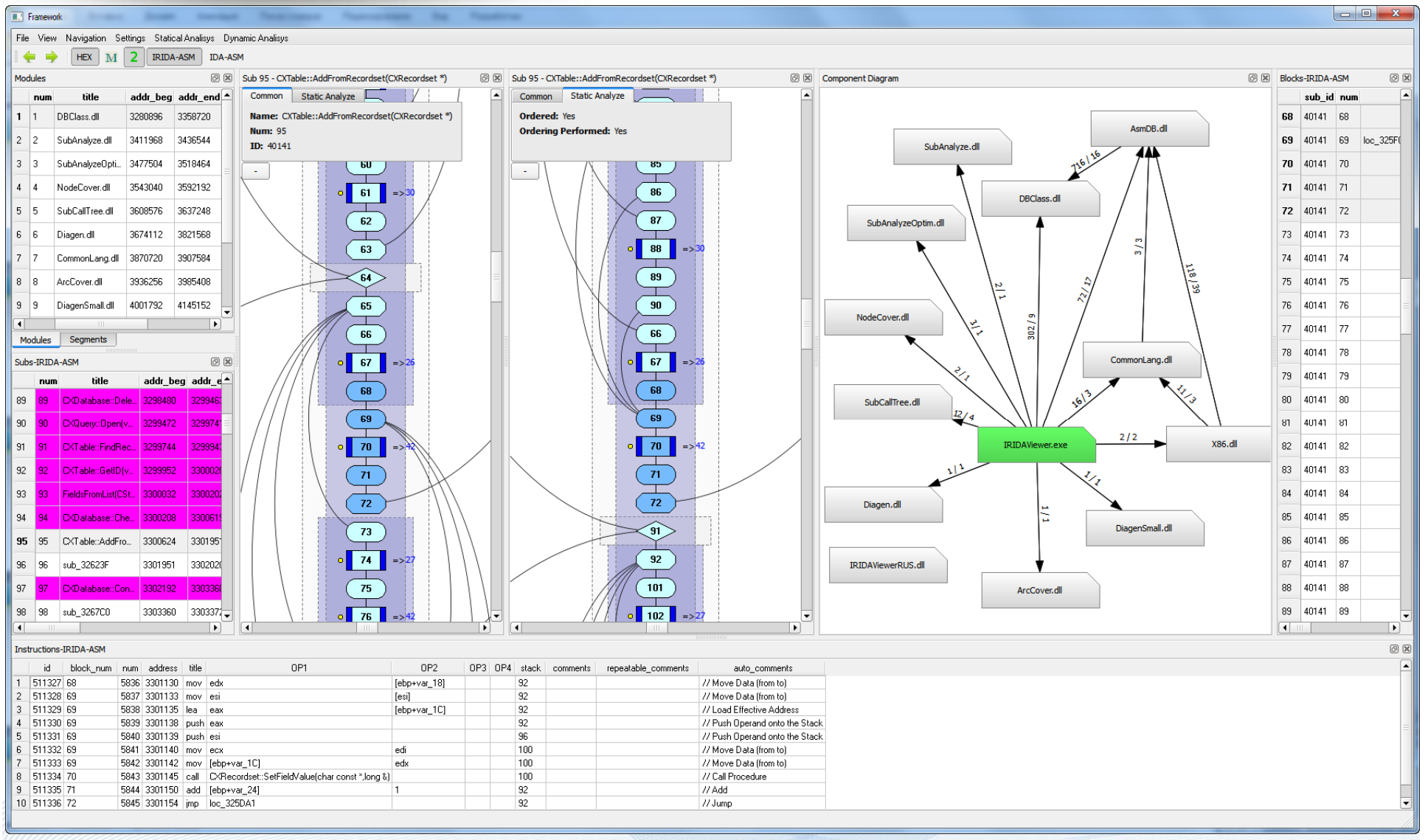
IRIDA 1.1



IRIDA 1.1

- Синтаксический разбор и анализ ассемблерных кодов и дизассемблированных листингов, и построение модели программы в БД MS Access/ MySQL
- Поддержка ассемблеров платформ Intel x86 и MIPS
- Структурирование и упорядочивание моделей подпрограмм; выявление нарушений структурированности программного кода
- Построение дерева вызова подпрограмм
- Формирование описания управляющего графа в виде грамматики в терминах контрольных точек дальних и ближних вызовов подпрограмм
- Создание паспортов потоков управления в исполняемых модулях (метод «Диаген»)
- Встраивание в контролируемую/исследуемую программу контрольных точек и подключение к ней паспорта
- Контроль выполнения программы посредством паспорта программы
- Получение трассы выполнения и ее наложение на модельное представление; как результат решение следующих задач:
  - покрытия кода;
  - получение статистики по вызовам и передачам управления;
  - выявление невызываемых подпрограмм.

IRIDA 2.0. state of the art



IRIDA 2.0. Изменения

- Полный цикл работы с многомодульным ПО, начиная с этапа формирования БД в среде дизассемблера (собственный плагин под дизассемблер IDA Pro) и заканчивая ДА и паспортом программы
- Плагинная архитектура, собственный SDK
- Тесная интеграция с дизассемблерами (HexRays IDA Pro), поддержка его встроенных отладчиков
- Поддержка различных форматов исполняемых файлов (PE, ELF, MIPS, etc.)
- Трассировка и динамический анализ внешними средствами

IRIDA 2.0. Изменения

- Сопоставление результатов СА и ДА
- Уточнение модели ПО по результатам сопоставления результатов СА и ДА
- Выявление участков модифицируемого кода
- Выявление нарушений структурированности
- IRIDA Sources – интегрированный в IRIDA модуль (исходные тексты программ)
- Поддержка пользовательских изменений модели ПО
- Поддержка различных СУБД: Oracle, MySQL, PostgreSQL
- Кроссплатформенность: Windows/Linux/MacOS

To Do

1. Внедрение точек на условные передачи управления
2. Переход с грамматического аппарата типа LL(k) на LL(*) для разрешения проблемы глубины поиска альтернатив в конечном автомате
3. Интеграция метода «Диаген» с методом семантической верификации на базе теории размерностей и использование атрибутных денотационных грамматик

To Do

4. Реализация общего паспорта многомодульного ПО: N исполняемых модулей (в рамках 1 адресного пространства) – 1 библиотека паспорта

5. Реализация выполнения автомата в потоке другого ядра (для многоядерной/многопроцессорной архитектуры); в общем оптимизация по скорости выполнения

6. Защита исполняемого кода паспорта программы посредством шифрования, обфускации, использования токенов, контроля запуска отладчиков в процессе выполнения паспорта

To Do

7. Реализация внедрения контрольных точек не на уровне исполняемых кодов, а на этапе создания исполняемого кода компилятором (этапы компиляции и линковки) – профилирование программы в терминах КТ

Причины необходимости такой реализации

1. Недостаточность на уровне исходных текстов
2. Техническая сложность внедрения на некоторые передачи управления (условные передачи управления) на уровне исполняемых кодов
3. Современные антивирусы идентифицируют технологию Диаген как хакерскую

To Do

7. Реализация внедрения контрольных точек не на уровне исполняемых кодов, а на этапе создания исполняемого кода компилятором (этапы компиляции и линковки) – профилирование программы в терминах КТ

Особенности такой реализации

1. Необходимость отдельной реализации для различных компиляторов под различные языки программирования
2. Как следствие, необходимость поддержки с выходом новых версий компиляторов
3. **Возможность внедрения КТ на любой участок контролируемого кода**

Выводы

- НДВ всегда присутствуют в ПО
- Контроль отсутствия/наличия НДВ в ПО требует специализированных методик и инструментов
- Создание условий невозможности проявления НДВ (запрета выполнения недоверенного кода) реализуемо программно на практике
- Паспортизация ПО – один из путей решения проблемы защиты ПО как от скрытых НДВ, так и от атак на ПО в памяти

Выводы

- ПО с исходными текстами целесообразно (начинать) контролировать еще на этапе разработки (но недостаточно только лишь на этом этапе)
- ПО без исходных текстов можно паспортизировать на уровне исполняемых кодов с помощью метода «Диаген»
- Для решения задачи комплексной защиты ПО в процессе его функционирования необходима интеграция метода как с существующими (шифрование, обфускация, tokens), так и с перспективными (семантический контроль информационных потоков) методами защиты ПО

Спасибо за внимание

ООО «Газинформсервис»
www.gaz-is.ru
Санкт-Петербург

Компаниец Радион Иванович *kompaniec-r@gaz-is.ru*
Ковалев Виктор Васильевич *kovalev-v@gaz-is.ru*
Маньков Евгений Викторович *mankov-e@gaz-is.ru*