Проблемы и пути решения практических задач анализа зависимостей между инструкциями при автоматизации динамического анализа программного кода

Тихонов А.Ю.
fireboo@mail.ru

Анализ потока данных в процессе динамического анализа

- Для чего используется
  - Выделение алгоритма (в т.ч. для последующего поиска уязвимостей с помощью решателя и поиска утечек чувствительных данных
  - Выявление фактов эксплуатации уязвимостей (dyn. taint)
- Типовые варианты применения

задача	способ решения
Dynamic tainting	Анализ зависимостей в прямом направлении
Выделение реализации алгоритма с помощью слайсинга (получение срезов)	**Анализ зависимостей в обратном направлении**

Чего хотим добиться

- Обеспечение полноты и отсутствие избыточности при динамическом анализе потока данных, т.е.
- исключение ошибок 1 и 2 рода, т.е.
  - исключение разрывов в потоке данных, приводящим к «недопометкам» (undertainting)
  - исключение из рассмотрения «лишних» зависимостей, приводящих к «перепометкам» (overtainting)

- Для этого
  - Выявить причины данных ошибок
  - Оценить масштаб проблемы
  - Рассмотреть способы решения

Три основных вида зависимостей по данным

- Функциональные (FD)

  выход одной инструкции является входом другой

- Адресные (AD)

  выход одной инструкции является адресом входной или выходной ячейки другой инструкции

- Зависимости по управлению (CD)

  от результата выполнения одной инструкции зависит факт и количество выполнений другой инструкции

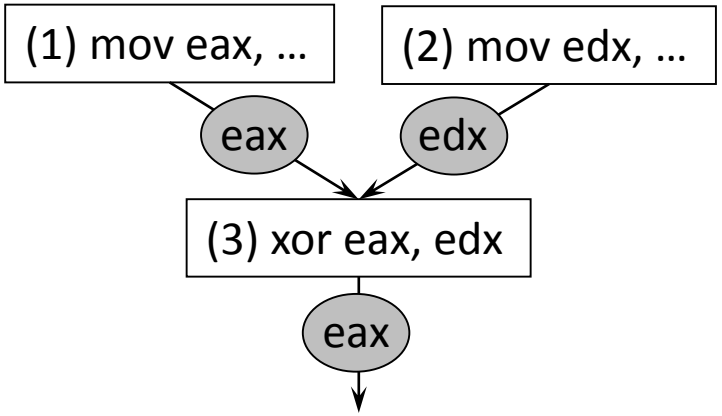
Функциональные зависимости

Выход одной инструкции является входом другой

(1) mov eax, …

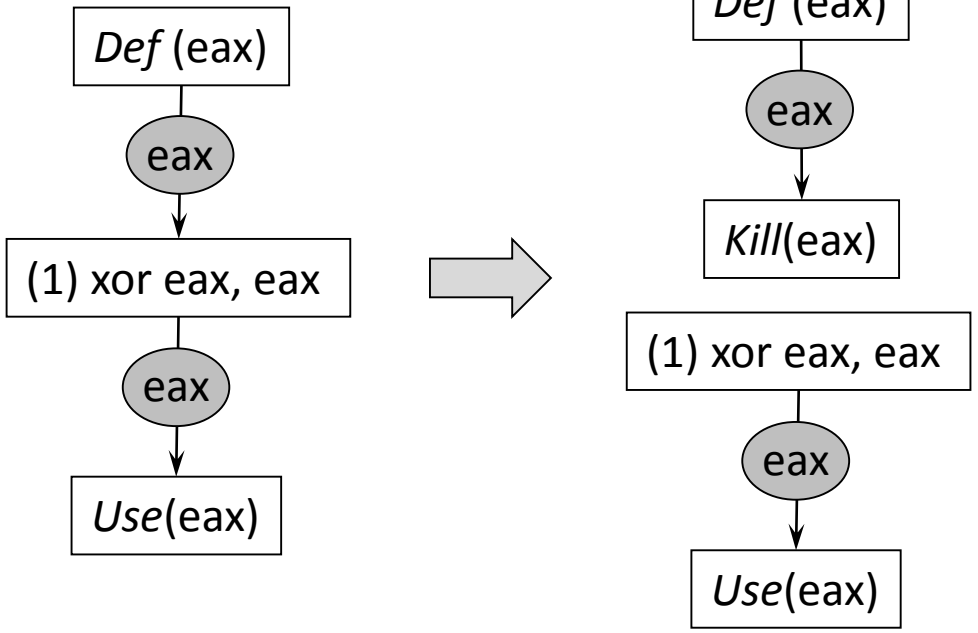
(2) mov edx, …

(3) xor eax, edx



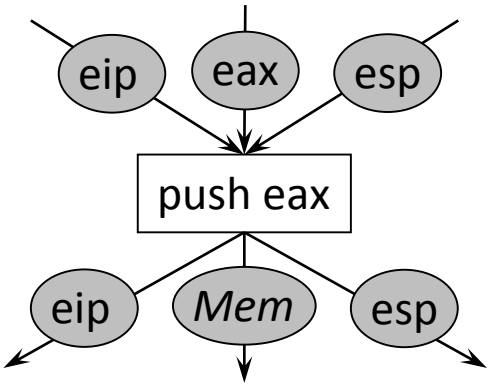
Проблема 1: зависимости могут «убиваться» вследствие математических свойств реализуемого алгоритма

(1) xor eax, eax



Следствие: перепометки

Проблема 2: между входами и выходами инструкции нет зависимости «все со всеми». Некоторые входы влияют на некоторые выходы.



Следствие: перепометки

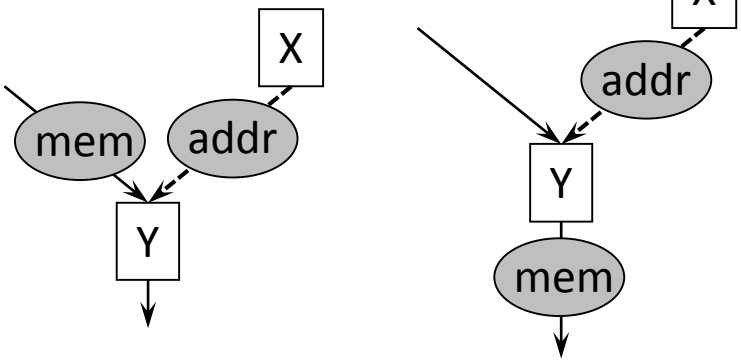
Проблема 3: в некоторых системах команд (x86) наличие множества сайд-эффектов, приводящих в том числе и к зависимостям по управлению. Пример – генерация исключений при выполнении инструкций

Следствие: недопометки

```
if(ValidAddr(in_esp))
    out_eip: FD(in_eip)
    out_Mem: FD(in_eax), AD(esp)
    out_esp: FD(in_esp)
else
    out_eip: в зависимости от вида ошибки FD(IDT(#GP)), FD(IDT(#PF)), FD(IDT(#GP)), FD(IDT(#SS)), FD(IDT(#AC)), …
```

Адресные зависимости

Выход одной инструкции является адресом входной или выходной ячейки другой инструкции

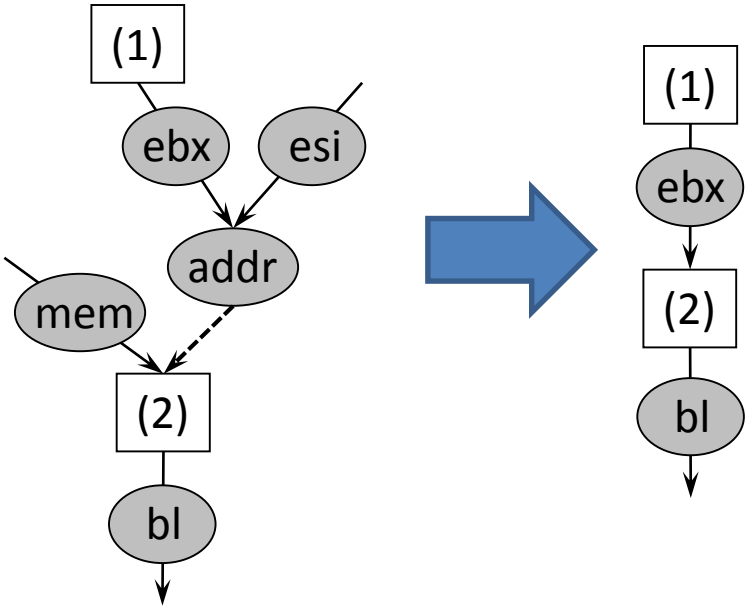


Проблема: функц. зависимость может быть выражена через адресную: табличное преобразование

(1) movzx ebx, BYTE PTR…

(2) mov bl, BYTE PTR [ebx+esi]

- esi указывает на константную таблицу



Следствие: недопометки

Зависимости по управлению

От результата выполнения одной инструкции зависит факт и количество выполнений другой инструкции

```
int i;
char *buf=somestring;
Отслеживается somestring.
for(i=0; buf[i]!=0; i++);
Далее i ипользуется в вычислениях.
```

Проблема: функциональная зависимость может быть выражена через зависимость по управлению:

копирование через табличное преобразование

Следствие: недопометки

```
char *alphabet="abcd…";
char *buf=somestring;
char stringcopy[64];
Отслеживается somestring.
for(i=0; buf[i]!=0; i++)
  for(int j=0; j<strlen(alphabet); j++)
    if(buf[i]== alphabet[j])
      { stringcopy[i]=j; break;}
Далее stringcopy используется в вычислениях.
```

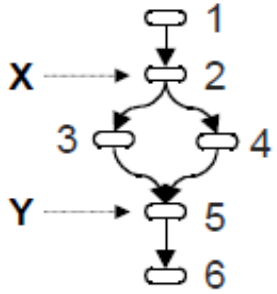
проблема	неучет	Попытка учета	
ФЗ 1: *kill* вследствие свойств алгоритма	перепометки	В частном случае - «паттерны» В общем - неразрешимая	
ФЗ 2: входы и выходы инструкции: нет зависимости «все со всеми»	перепометки	Введение микроинструкций – решает проблему	
ФЗ 3: сайд-эффекты и сложная логика	недопометки	Решение проблемы при сильном усложнении представления уровня микроинструкций	
Наличие адресных зависимостей	недопометки	Решение проблемы недопометок. Перепометки – **при обратном проходе неприемлемо много**	
Наличие зависимостей по управлению	недопометки	Решение проблемы недопометок. Перепометки - много	не учитываются в пошаговых отладчиках и известных системах динамического анализа помеченных данных

Статико-динамическая модель для анализа зависимостей по управлению

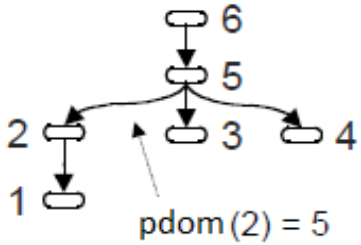
Код программы

Доминатор:
X dom Y: все пути из start в Y проходят через X

Граф потока управления
Control Flow Graph - CFG



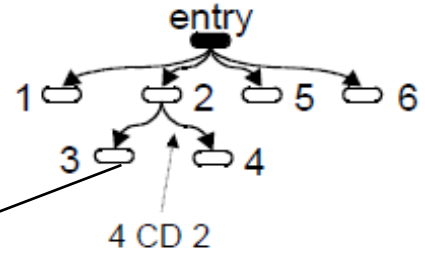
Дерево постдоминаторов



Постдоминатор:
Y pdom X: все пути из X в stop проходят через Y

Граф зависимостей по данным
Data Dependence Graph - DDG

Граф зависимостей по управлению
Control Dependence Graph - CDG



Граф зависимостей программы (Program Dependence Graph - PDG)

Зависимость по управлению:
если инструкция X влияет на возможность выполнения инструкции Y, то Y зависит по управлению от X

Y CD X:
1. Существует путь X->Y, на котором Y pdom каждой вершины пути, отличной от X и Y
2. Y не pdom X

Пример: выделение алгоритма получения кода символа по коду нажатой клавиши

Чем хорош пример:

- Поток данных проходит через разные процессы, потоки и адресные пространства. Чрезвычайно сложен для пошагового анализа
- Наличие двух копирований через адресные зависимости и одного – через зависимости по управлению
- В известных публикациях – учет зависимости по управлению либо не решается, либо частное решение, неприемлемое в общем случае
- Относительно прост для демонстрации: референсный слайс – 84 инструкции

- Референсный слайс получен склейкой нескольких слайсов, полученных путем отслеживания только функциональных зависимотей, в точках разрыва входных данных

- Задача – подключить учет адресных зависимостей и зависимостей по управлению для получения похожего результата без ручной склейки

```
                0008:806F68BE        IN          AL, DX
  i8042prt.sys  0008:F790744B        MOV         BYTE PTR [EDI], AL  ; [F6794D23]
                                     ...
    win32k.sys  0008:BF88AB26        MOV         CL, BYTE PTR [ESI]  ; [F78969D0]
    win32k.sys  0008:BF88AB34        MOVZX       ECX, CL

                                     разрыв по адресной зависимости (табличное преобразование)

    win32k.sys  0008:BF88AB37        MOV         BX, WORD PTR [EAX + ECX * 2]  ; [E173002A]
    win32k.sys  0008:BF88AB63        MOV         WORD PTR [ESI + 02h], BX  ; [F78969D2]
                                     ...
    win32k.sys  0008:BF822070        MOV         DWORD PTR [EAX + 08h], ECX  ; [F67A4D20]
    win32k.sys  0008:BF819E84    REP MOVSD       DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [0119FFB4], [F67A4D20]
    win32k.sys  0008:BF848A2B    REP MOVSD       DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [F67A4D28], [0119FFB4]
    win32k.sys  0008:BF84891C        MOVZX       EAX, WORD PTR [ESI + 08h]  ; [F67A4D28]
    win32k.sys  0008:BF848939        PUSH        EAX  ; [f67a4ca8]  <-----------------------------
                                     ...                                                        |
    win32k.sys  0008:BF848C8B        CMP         AL, BYTE PTR SS:[EBP + 08h]  ; [F67A4CA8]      |
    win32k.sys  0008:BF848C8E        JZ          0BF848CB5h  ; -> BF848CB5                      |
    win32k.sys  0008:BF848C90        MOVZX       EAX, BYTE PTR [EBX + 05h]  ; [E1730399]        |
    win32k.sys  0008:BF848C94        ADD         EDI, EAX    #44 win32k.sys!_xxxInternalToUnicode@32 <--
    win32k.sys  0008:BF848C96        JMP         0BF848C85h  ; -> BF848C85                      |     |
    win32k.sys  0008:BF848C85        MOV         AL, BYTE PTR [EDI]  ; [E1730202]               |     |
    win32k.sys  0008:BF848C87        TEST        AL, AL                                         |     |
    win32k.sys  0008:BF848C89        JZ          0BF848C98h  ; -> BF848C98                      |     |
    win32k.sys  0008:BF848C8B        CMP         AL, BYTE PTR SS:[EBP + 08h]  ; [F67A4CA8]  ------      |
    win32k.sys  0008:BF848C8E        JZ          0BF848CB5h  ; -> BF848CB5                      |
                                     ...                                                        |
                                     разрыв по зависимости по управлению (табличное преобразование)    |
                                                                                                |
    win32k.sys  0008:BF848D46        MOV         AX, WORD PTR [ESI + EDI + 02h]  ; [E1730204]  ---------
    win32k.sys  0008:BF848D4E        MOV         WORD PTR [ECX], AX  ; [F67A4CE0]
                                     ...
    winsrv.dll  001B:75B865B8        PUSH        EAX
     ntdll.dll  001B:7C9271B5        MOVZX       EBX, WORD PTR [EAX + 1Ch]  ; [0119F988]

                                     разрыв по адресной зависимости (табличное преобразование)

     ntdll.dll  001B:7C9271B9        MOV         BL, BYTE PTR [EBX + ESI]  ; [7FFC1499]
     ntdll.dll  001B:7C9271BC        MOV         BYTE PTR [ECX + 0Eh], BL  ; [0119F97B]
                                     ...
getch.exe!$LN8  001B:00401241        MOV         EAX, DWORD PTR SS:[EBP - 1Ch]  ; [0012FF58]
      getch.exe 001B:00401015        PUSH        EAX
```

- Формальное применение обратного слайсинга с учетом адресных зависимостей и зависимостей по управлению: разочаровывающий результат – число «лишних» инструкций на 2 порядка превышает число «полезных»

- Что делать?

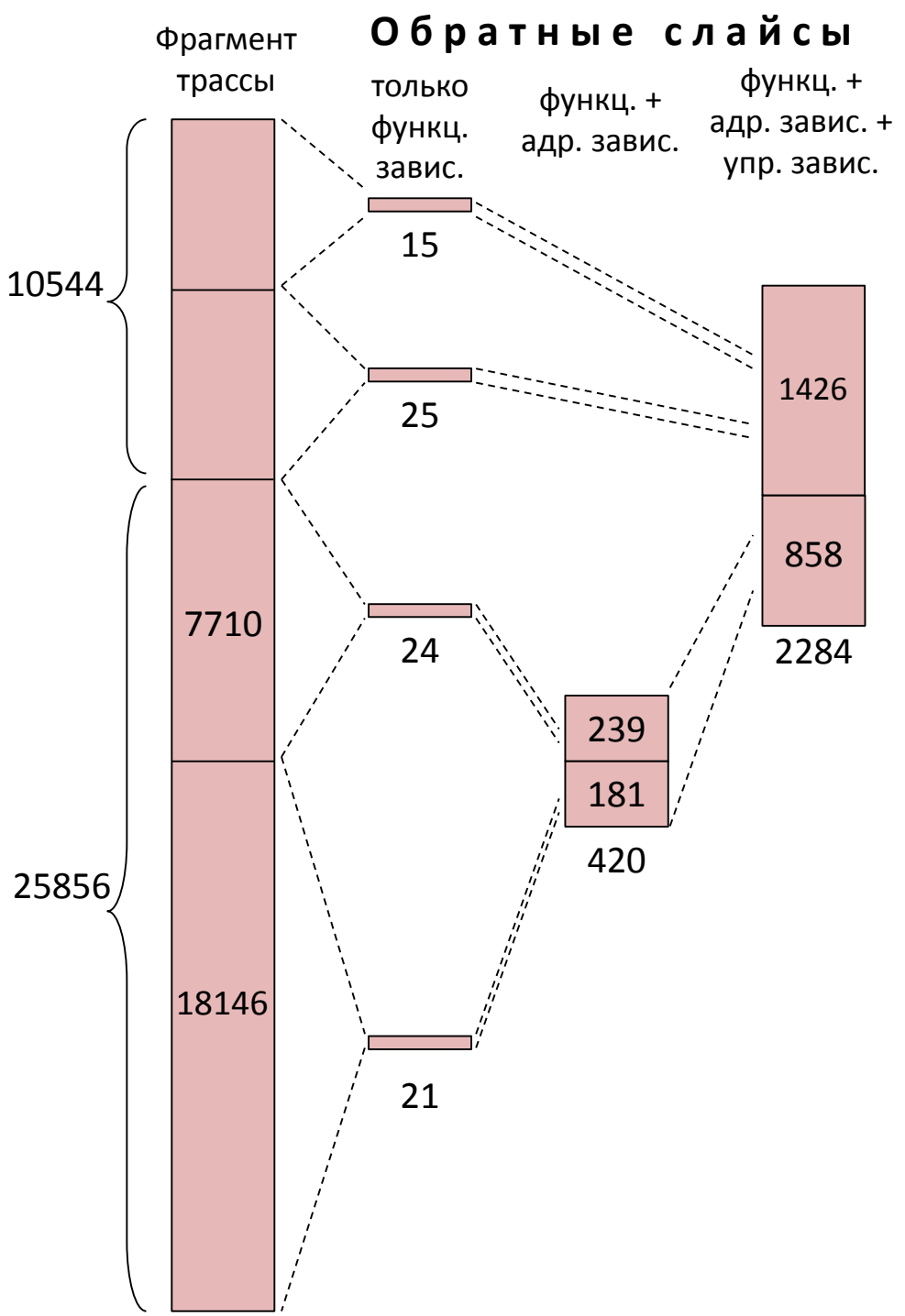
```
                0008:806F68BE            IN          AL, DX
    i8042prt.sys 0008:F790744B           MOV         BYTE PTR [EDI], AL  ; [F6794D23]
    i8042prt.sys 0008:F79074FD           MOV         EAX, DWORD PTR SS:[EBP - 1Dh]  ; [F6794D23]
    i8042prt.sys 0008:F7907789           MOVZX       CX, AL
    i8042prt.sys 0008:F790778D           MOV         WORD PTR [ESI + 00000202h], CX  ; [8668C2DA]
    i8042prt.sys 0008:F790768C           MOVSD       DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [866DB428], [8668C2D8]
    ntoskrnl.exe 0008:804D9E94           MOV         EAX, DWORD PTR [ESI + ECX * 4 - 0Ch]  ; [866DB428]
    ntoskrnl.exe 0008:804D9E98           MOV         DWORD PTR [EDI + ECX * 4 - 0Ch], EAX  ; [865E9120]
    ntoskrnl.exe 0008:804ECA2B      REP MOVSD        DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [E1725628], [865E9120]
      win32k.sys 0008:BF88A7E2           MOV         DL, BYTE PTR [EBX + 02h]  ; [E172562A]
      win32k.sys 0008:BF88A7E5           AND         DL, 7Fh
      win32k.sys 0008:BF88A7EB           MOV         BYTE PTR SS:[EBP - 1Ch], DL  ; [F78969D0]
      win32k.sys 0008:BF88AAFD           AND         BYTE PTR [ESI], 7Fh  ; [F78969D0]
      win32k.sys 0008:BF88AB26           MOV         CL, BYTE PTR [ESI]  ; [F78969D0]
      win32k.sys 0008:BF88AB34           MOVZX       ECX, CL

      win32k.sys 0008:BF88AB37           MOV         BX, WORD PTR [EAX + ECX * 2]  ; [E173002A]
      win32k.sys 0008:BF88AB63           MOV         WORD PTR [ESI + 02h], BX  ; [F78969D2]
      win32k.sys 0008:BF84882C           MOV         AX, WORD PTR [ESI + 02h]  ; [F78969D2]
      win32k.sys 0008:BF848830           PUSH        EAX
      win32k.sys 0008:BF847EC8           MOV         EDX, DWORD PTR SS:[EBP + 08h]  ; [F7896980]
      win32k.sys 0008:BF847F0D           MOV         BYTE PTR SS:[EBP - 04h], DL  ; [F7896974]
      win32k.sys 0008:BF847FBC           MOVZX       EAX, BYTE PTR SS:[EBP - 04h]  ; [F7896974]
      win32k.sys 0008:BF847FC2           MOV         DWORD PTR SS:[EBP - 14h], EAX  ; [F7896964]
      win32k.sys 0008:BF84806F           MOV         EAX, DWORD PTR SS:[EBP - 14h]  ; [F7896964]
      win32k.sys 0008:BF848076           MOV         DWORD PTR SS:[EBP + 20h], EAX  ; [F7896998]
      win32k.sys 0008:BF8480F5           PUSH        DWORD PTR SS:[EBP + 20h]  ; [F7896998]
      win32k.sys 0008:BF84764C           PUSH        DWORD PTR SS:[EBP + 14h]  ; [F789691C]
      win32k.sys 0008:BF8022B5           MOV         EAX, DWORD PTR SS:[EBP + 14h]  ; [F78968E8]
      win32k.sys 0008:BF8022B8           MOV         DWORD PTR [ECX + 10h], EAX  ; [E174FD68]
      win32k.sys 0008:BF820ADB      REP MOVSD        DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [F67A4C48], [E174FD68]
      win32k.sys 0008:BF821E4D           MOV         ECX, DWORD PTR SS:[EBP - 50h]  ; [F67A4C48]
      win32k.sys 0008:BF821E53           MOV         DWORD PTR SS:[EBP - 04h], ECX  ; [F67A4C94]
      win32k.sys 0008:BF82206D           MOV         ECX, DWORD PTR SS:[EBP - 04h]  ; [F67A4C94]
      win32k.sys 0008:BF822070           MOV         DWORD PTR [EAX + 08h], ECX  ; [F67A4D20]
      win32k.sys 0008:BF819E84      REP MOVSD        DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [0119FFB4], [F67A4D20]
      win32k.sys 0008:BF848A2B      REP MOVSD        DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [F67A4D28], [0119FFB4]
      win32k.sys 0008:BF84891C           MOVZX       EAX, WORD PTR [ESI + 08h]  ; [F67A4D28]
      win32k.sys 0008:BF848939           PUSH        EAX

                                         ; инкрементирование EDI в цикле
      win32k.sys 0008:BF848C8B           CMP         AL, BYTE PTR SS:[EBP + 08h]  ; [F67A4CA8]
      win32k.sys 0008:BF848C8E           JZ          0BF848CB5h  ; -> BF848CB5

      win32k.sys 0008:BF848D46           MOV         AX, WORD PTR [ESI + EDI + 02h]  ; [E1730204]
      win32k.sys 0008:BF848D4E           MOV         WORD PTR [ECX], AX  ; [F67A4CE0]
      win32k.sys 0008:BF8489BB           MOVZX       EAX, WORD PTR [ESI]  ; [F67A4CE0]
      win32k.sys 0008:BF8489BE           PUSH        EAX
      win32k.sys 0008:BF80885A           PUSH        DWORD PTR SS:[EBP + 10h]  ; [F67A4CC0]
      win32k.sys 0008:BF8022B5           MOV         EAX, DWORD PTR SS:[EBP + 14h]  ; [F67A4C80]
      win32k.sys 0008:BF8022B8           MOV         DWORD PTR [ECX + 10h], EAX  ; [E174FD68]
      win32k.sys 0008:BF8026B8      REP MOVSD        DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [F67A4D20], [E174FD68]
      win32k.sys 0008:BF819E84      REP MOVSD        DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [0119FFB4], [F67A4D20]
      user32.dll 001B:7E4189B8           PUSH        DWORD PTR [EDI]  ; [0119FFB4]
      user32.dll 001B:7E418807           PUSH        DWORD PTR SS:[EBP + 18h]  ; [0119FF20]
      user32.dll 001B:7E41871B           PUSH        DWORD PTR SS:[EBP + 14h]  ; [0119FEB4]
      winsrv.dll 001B:75B61D6E           MOV         EBX, DWORD PTR SS:[EBP + 10h]  ; [0119FE84]
      user32.dll 001B:7E4185DA           PUSH        EBX
      user32.dll 001B:7E41860D           POP         EBX
      winsrv.dll 001B:75B6AD77           PUSH        EBX
      winsrv.dll 001B:75B6C1EA           MOV         AX, WORD PTR SS:[EBP + 14h]  ; [0119FC2C]
      winsrv.dll 001B:75B6C1EE           MOV         WORD PTR SS:[EBP - 00000126h], AX  ; [0119FAF2]
       ntdll.dll 001B:7C90221C           MOV         EAX, DWORD PTR [ESI + ECX * 4 - 08h]  ; [0119FAF0]
       ntdll.dll 001B:7C902220           MOV         DWORD PTR [EDI + ECX * 4 - 08h], EAX  ; [004F265C]
      winsrv.dll 001B:75B7FE8E      REP MOVSD        DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [0016A954], [004F265C]
      winsrv.dll 001B:75B865B5           MOV         AX, WORD PTR [ESI]  ; [0016A956]
      winsrv.dll 001B:75B865B8           PUSH        EAX
       ntdll.dll 001B:7C9271B5           MOVZX       EBX, WORD PTR [EAX + 1Ch]  ; [0119F988]


       ntdll.dll 001B:7C9271B9           MOV         BL, BYTE PTR [EBX + ESI]  ; [7FFC1499]
       ntdll.dll 001B:7C9271BC           MOV         BYTE PTR [ECX + 0Eh], BL  ; [0119F97B]
      winsrv.dll 001B:75B76BA1           MOV         AL, BYTE PTR SS:[EBP - 01h]  ; [0119F97B]
      winsrv.dll 001B:75B865C7           MOV         BYTE PTR [ESI], AL  ; [0016A956]
    ntoskrnl.exe 0008:8056B789      REP MOVSD        DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [E1948C1C], [0016A954]
    ntoskrnl.exe 0008:8056B789      REP MOVSD        DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [0012FE58], [E1948C1C]
    kernel32.dll 001B:7C874411      REP MOVSD        DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [0012FF28], [0012FE58]
       getch.exe 001B:0040118F           MOVZX       EDI, BYTE PTR SS:[EBP - 0Eh]  ; [0012FF2A]
l!_SetConsoleMode@8 001B:7C81AF10        MOV         EDI, EDI
ClientCallServer@16 001B:7C912D71        MOV         EDI, EDI
       ntdll.dll 001B:7C912DAF           PUSH        EDI
       ntdll.dll 001B:7C912DD0           POP         EDI
       getch.exe 001B:004011C9           MOV         EAX, EDI
       getch.exe 001B:004011F9           MOV         EDI, EAX
.exe!__putch_nolock 001B:00401253        MOV         EDI, EDI
       getch.exe 001B:0040125B           PUSH        EDI
       getch.exe 001B:004012C3           POP         EDI
       getch.exe 001B:0040120C           MOV         EAX, EDI
       getch.exe 001B:00401232           MOV         DWORD PTR SS:[EBP - 1Ch], EAX  ; [0012FF58]
  getch.exe!$LN8 001B:00401241           MOV         EAX, DWORD PTR SS:[EBP - 1Ch]  ; [0012FF58]
       getch.exe 001B:00401015           PUSH        EAX
```



Возможные подходы к устранению «лишних» зависимостей

1. Исключение из рассмотрения адресных зависимостей по стековым регистрам ESP и EBP
   – проблема (недопометки) если эти регистры будут задействованы в логике алгоритма

2. Трансформация слайса – исключение части инструкции, не задействованной в зависимостях, оптимизация - исключение парных инструкций (push/pop)

3. Прямой слайсинг по результатам обратного

Результаты применения разных подходов

- Наилучший результат дает комбинация подходов 1+3 или 3+2

Размер фрагмента трассы (размер референсного слайса), шагов	**Обратный слайс с AD, CD, kills**		**Обратный слайс с AD, CD, kills**		
	без стековых зависимостей (адресные по EBP,ESP)		со стековыми зависимостями		
	-	**2 проход: прямой слайс с AD**	**-**	**2 проход: прямой слайс с AD**	**3 проход: частичная оптимизация**
1	2	3	4	5	6
25914 (45)	951	47	10195	167	119

- Обратный слайсинг со всеми зависимостями кроме стековых + прямой слайсинг: 100% совпадение с референсным слайсом

- Обратный слайсинг со всеми зависимостями без исключения + прямой слайсинг: почти на 2 порядка сокращение числа «лишних» инструкций, основная часть оставшихся «лишних» инструкций – работа со стеком (относительно легко устраняется?)

```
29      user32.dll 001B:7E41860D X          POP       EBX  ; [0119FC08]
30      winsrv.dll 001B:75B6AD77 X          PUSH      EBX  ; [0119FC2C]
31      winsrv.dll 001B:75B6AD9F            PUSH      EBX  ; [0119FAB8]
32      winsrv.dll 001B:75B6AC8C            PUSH      EBX  ; [0119FAA0]
33    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
34    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
35    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
36    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
37    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
38    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
39    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
40    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
41    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
42    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
43    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
44    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
45    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
46    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
47    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
48    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
49    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
50    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
51      winsrv.dll 001B:75B6AD2A            POP       EBX  ; [0119FAA0]
52      winsrv.dll 001B:75B6C219            PUSH      ESI  ; [0119FAA0]
53    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
54    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
55    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
56      winsrv.dll 001B:75B6AC8C            PUSH      EBX  ; [0119FAA0]
57    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
58    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
59    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
60    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
61    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
62    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
63    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
64    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
65    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
66    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
67    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
68    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
69    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
70    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
71    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
72    ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
73    ntoskrnl.exe 0008:804DE723            MOV       EBX, DWORD PTR [0FFDFF01Ch]  ; [FFDFF01C]
74    ntoskrnl.exe 0008:804DE8CF            POP       EBX  ; [F67A4DC0]
75      winsrv.dll 001B:75B6AD2A            POP       EBX  ; [0119FAA0]
```

```
76          winsrv.dll 001B:75B6C1EA X          MOV       AX, WORD PTR SS:[EBP + 14h]  ; [0119FC2C]
77          winsrv.dll 001B:75B6C1EE X          MOV       WORD PTR SS:[EBP - 00000126h], AX  ; [0119FAF2]
78          winsrv.dll 001B:75B6AF9D            LEA       EAX, SS:[EBP - 00000134h]  ; [0119FAE4]
79          winsrv.dll 001B:75B6AFA8            PUSH      EDI  ; [0119FAA0]
80          winsrv.dll 001B:75B649B4            MOV       EBX, DWORD PTR SS:[EBP + 10h]  ; [0119FA94]
81           ntdll.dll 001B:7C90221C X          MOV       EAX, DWORD PTR [ESI + ECX * 4 - 08h]  ; [0119FAF0]
82           ntdll.dll 001B:7C902220 X          MOV       DWORD PTR [EDI + ECX * 4 - 08h], EAX  ; [004F265C]
83  <l.dll!_NtSetEvent@8 001B:7C90DBF0          MOV       EAX, 000000DBh
84         ntoskrnl.exe 0008:804DE720            PUSH      EBX  ; [F67A4DC0]
85         ntoskrnl.exe 0008:80564BF0            PUSH      ESI  ; [F67A4CE0]
86         ntoskrnl.exe 0008:80564B8C            PUSH      EBX  ; [F67A4CB8]
87          winsrv.dll 001B:75B7FE8E X      REP MOVSD     DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [0016A954], [004F265C]
88         ntoskrnl.exe 0008:80564BF0            PUSH      ESI  ; [F67A4D20]
89          winsrv.dll 001B:75B865B5 X          MOV       AX, WORD PTR [ESI]  ; [0016A956]
90          winsrv.dll 001B:75B865B8 X          PUSH      EAX  ; [0119F988]
91          winsrv.dll 001B:75B865BC            PUSH      DWORD PTR [EAX + 000001A4h]  ; [004F24F4], [0119F984]
92          winsrv.dll 001B:75B76B7C            PUSH      EAX  ; [0119F96C]
93          winsrv.dll 001B:75B76B7F            LEA       EAX, SS:[EBP - 01h]  ; [0119F97B]
94           ntdll.dll 001B:7C9271B5 X          MOVZX     EBX, WORD PTR [EAX + 1Ch]  ; [0119F988]
95           ntdll.dll 001B:7C9271B9 X          MOV       BL, BYTE PTR [EBX + ESI]  ; [7FFC1499]
96           ntdll.dll 001B:7C9271BC X          MOV       BYTE PTR [ECX + 0Eh], BL  ; [0119F97B]
97           ntdll.dll 001B:7C9271CA            POP       EBX  ; [0119F950]
98          winsrv.dll 001B:75B76BA1 X          MOV       AL, BYTE PTR SS:[EBP - 01h]  ; [0119F97B]
99          winsrv.dll 001B:75B865C7 X          MOV       BYTE PTR [ESI], AL  ; [0016A956]
100          ntdll.dll 001B:7C90E8B6            PUSH      EAX  ; [0119F990]
101          ntdll.dll 001B:7C90E8B7            MOV       EAX, DWORD PTR SS:[ESP + 10h]  ; [0119F9A0]
102        ntoskrnl.exe 0008:8056B789 X      REP MOVSD     DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [E1948C1C], [0016A954]
103         winsrv.dll 001B:75B61E6A            PUSH      DWORD PTR SS:[EBP - 00000164h]  ; [0119FD10], [0119FC2C]
104         winsrv.dll 001B:75B6C6CD            CALL      DWORD PTR [75B6101Ch]  ; [75B6101C], [0119FC08]
105        ntoskrnl.exe 0008:8056B789 X      REP MOVSD     DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [0012FE58], [E1948C1C]
106        kernel32.dll 001B:7C874411 X      REP MOVSD     DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [0012FF28], [0012FE58]
107          getch.exe 001B:0040118F X          MOVZX     EDI, BYTE PTR SS:[EBP - 0Eh]  ; [0012FF2A]
108  <l!_SetConsoleMode@8 001B:7C81AF10 X       MOV       EDI, EDI
109        kernel32.dll 001B:7C81AF38            PUSH      0000000Ch  ; [0012FE58]
110  <ClientCallServer@16 001B:7C912D71 X       MOV       EDI, EDI
111          ntdll.dll 001B:7C912DAF X          PUSH      EDI  ; [0012FE38]
112        ntoskrnl.exe 0008:804DE722            PUSH      EDI  ; [F69DFDB8]
113        ntoskrnl.exe 0008:804DE781            MOV       EDI, EAX
114        ntoskrnl.exe 0008:804DE8CD            POP       EDI  ; [F69DFDB8]
115          ntdll.dll 001B:7C912DD0 X          POP       EDI  ; [0012FE38]
116          getch.exe 001B:004011C9 X          MOV       EAX, EDI
117          getch.exe 001B:004011F9 X          MOV       EDI, EAX
118          getch.exe 001B:00401200            PUSH      EDI  ; [0012FF3C]
119  <.exe!__putch_nolock 001B:00401253 X       MOV       EDI, EDI
120          getch.exe 001B:0040125A            PUSH      ESI  ; [0012FF28]
```

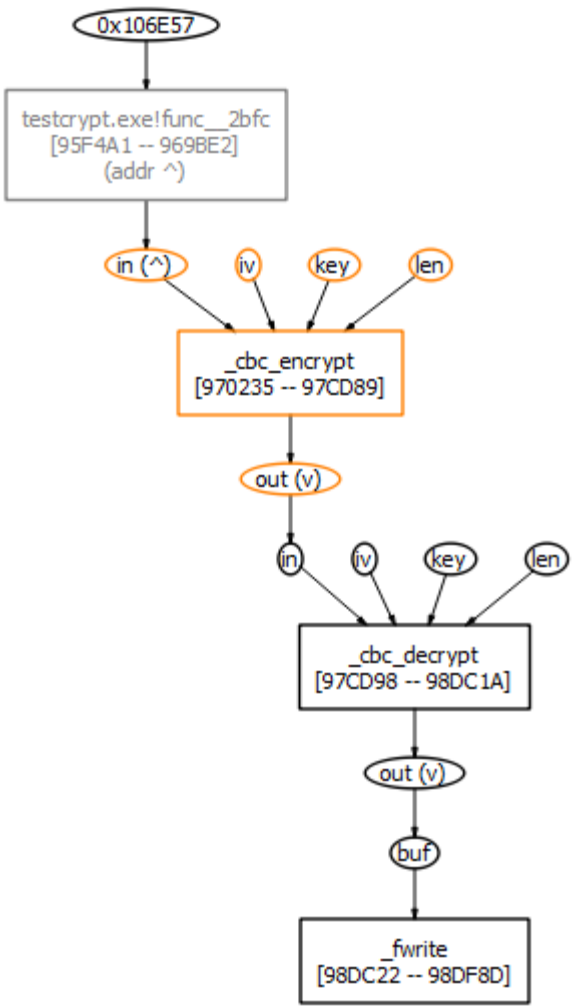
```
121        getch.exe 001B:0040125B X          PUSH      EDI  ; [0012FF24]
122        getch.exe 001B:0040226E            PUSH      EDI  ; [0012FF10]
123        getch.exe 001B:00402282            CALL      EAX  ; -> 7C8097D0 ; [0012FF08]
124      kernel32.dll 001B:7C8097D5           MOV       EAX, DWORD PTR FS:[00000018h]  ; [7FFDD018]
125        getch.exe 001B:004022D8            PUSH      EDI  ; [0012FF0C]
126        getch.exe 001B:004022DF            POP       EDI  ; [0012FF10]
127        getch.exe 001B:00401268            MOV       AL, BYTE PTR SS:[EBP + 08h]  ; [0012FF3C]
128        getch.exe 001B:0040126B            LEA       EDI, [ESI + 4Ch]  ; [00341EDC]
129        getch.exe 001B:00401277            MOV       BYTE PTR [EDI], AL  ; [00341EDC]
130        getch.exe 001B:00401283            PUSH      EAX  ; [0012FF20]
131        getch.exe 001B:00402011            PUSH      DWORD PTR SS:[EBP + 08h]  ; [0012FF20], [0012FF10]
132        getch.exe 001B:00402014            CALL      00401FD2h  ; -> 00401FD2 ; [0012FF0C]
133        getch.exe 001B:00402282            CALL      EAX  ; -> 7C8097D0 ; [0012FECC]
134        getch.exe 001B:00402019            POP       ECX  ; [0012FF10]
135        getch.exe 001B:0040201A            POP       ECX  ; [0012FF14]
136        getch.exe 001B:00401289            POP       ECX  ; [0012FF20]
137        getch.exe 001B:00401298            PUSH      EAX  ; [0012FF20]
138        getch.exe 001B:00401299            LEA       EAX, SS:[EBP - 04h]  ; [0012FF30]
139        getch.exe 001B:00401FBA            PUSH      EBP  ; [0012FF10]
140        getch.exe 001B:00401FBF            PUSH      DWORD PTR SS:[EBP + 10h]  ; [0012FF20], [0012FF08]
141        getch.exe 001B:00401EE9            MOVZX     CX, BYTE PTR [ESI]  ; [00341EDC]
142        getch.exe 001B:00401EED            MOV       WORD PTR [EAX], CX  ; [0012FF30]
143        getch.exe 001B:004012AB            PUSH      DWORD PTR SS:[EBP - 04h]  ; [0012FF30], [0012FF20]
144        getch.exe 001B:00401E1C            PUSH      ECX  ; [0012FEFC]
145        getch.exe 001B:00401E1F            LEA       ECX, SS:[EBP + 08h]  ; [0012FF20]
146      kernel32.dll 001B:7C8354A3           PUSH      DWORD PTR SS:[EBP + 14h]  ; [0012FEFC], [0012FEE0]
147      kernel32.dll 001B:7C81CBC5       REP MOVSB     BYTE PTR ES:[EDI], BYTE PTR [ESI]  ; [0012FE40], [0012FF20]
148      ntoskrnl.exe 0008:804DE722           PUSH      EDI  ; [F69DFDB8]
149      ntoskrnl.exe 0008:8056B789       REP MOVSD     DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [E1948C08], [0012FE38]
150      ntoskrnl.exe 0008:8056B789       REP MOVSD     DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [E1948C10], [0012FE40]
151      ntoskrnl.exe 0008:8056B789       REP MOVSD     DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [E1948C1C], [0012FE4C]
152      ntoskrnl.exe 0008:8056B789       REP MOVSD     DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [006AFF14], [E1948C08]
153      ntoskrnl.exe 0008:8056B789       REP MOVSD     DWORD PTR ES:[EDI], DWORD PTR [ESI]  ; [006AFF1C], [E1948C10]
154        winsrv.dll 001B:75B63598           PUSH      DWORD PTR [ESI + 28h]  ; [006AFF14], [006AFEC0]
155        winsrv.dll 001B:75B635CE           PUSH      40000000h  ; [006AFEC0]
156        winsrv.dll 001B:75B6450F           CMP       WORD PTR [EAX], 0020h  ; [006AFF1C]
157        getch.exe 001B:004012B3            POP       ECX  ; [0012FF20]
158        getch.exe 001B:004012C3 X          POP       EDI  ; [0012FF24]
159        getch.exe 001B:00401206            POP       ECX  ; [0012FF3C]
160        getch.exe 001B:0040120C X          MOV       EAX, EDI
161        getch.exe 001B:0040120E            POP       EDI  ; [0012FF40]
162        getch.exe 001B:00401232 X          MOV       DWORD PTR SS:[EBP - 1Ch], EAX  ; [0012FF58]
163        getch.exe 001B:0040124C            CALL      00401ADEh  ; -> 00401ADE ; [0012FF3C]
164        getch.exe 001B:00401AE3            MOV       EAX, DWORD PTR SS:[EBP + 08h]  ; [0012FF40]
165        getch.exe 001B:00401AED            CALL      DWORD PTR [0040803Ch]  ; [0040803C], [0012FF30]
166    getch.exe!$LN8 001B:00401241 X         MOV       EAX, DWORD PTR SS:[EBP - 1Ch]  ; [0012FF58]
167        getch.exe 001B:00401015 X          PUSH      EAX  ; [0012FF78]
```

Проблема применения комбинации обратный-прямой слайс

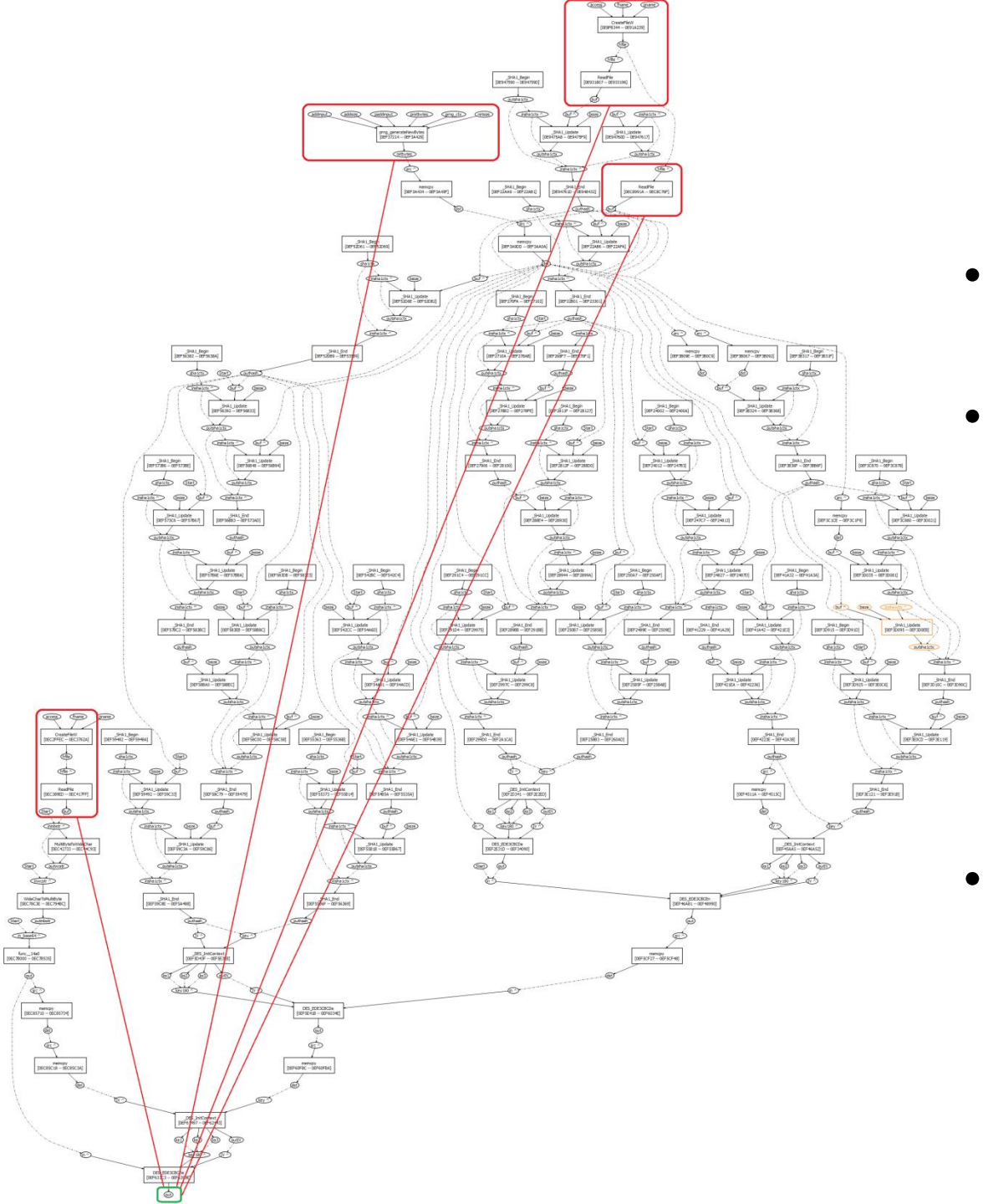
- Точка расположения выходных данных (шаг и ячейки) как правило известна
- Проблема - нужно знать точку расположения входных данных
  - В некоторых случаях (использование документированных API-функций или аппаратных интерфейсов) – такая точка легко находится
  - С точки зрения применение прямого слайсинга к обратному – легко идентифицируемых точек ввода входных данных как правило несколько. Как искать?

- Вариант решения – интерактивное построение схемы алгоритма

Построение схемы алгоритма

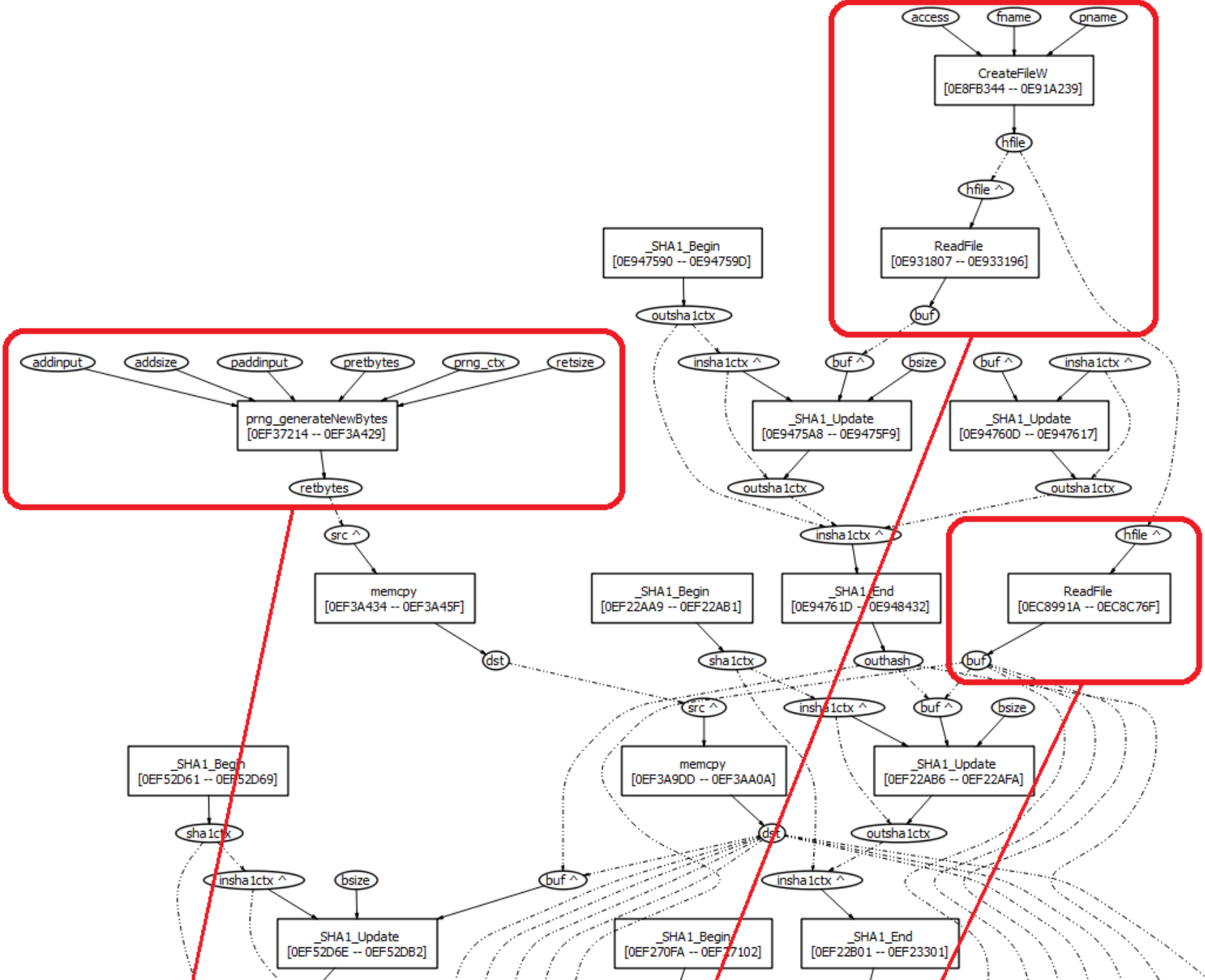


- ***Схема алгоритма*** – граф, который строится на основе трассы программы и позволяет в наглядном виде представить преобразования *некоторых* данных, которые осуществляет исследуемая программа.
- Граф включает в себя множество вершин, соответствующих вызовам функций с известной (и описанной) семантикой, и отображает потоки данных между входными и выходными параметрами этих функций.
- Для формального описания семантики функций используются так называемые ***модели*** функций, которые задают множество входных и выходных параметров и возможные связи между ними.
- Построение схемы алгоритма является итеративный и интерактивный процесс, поскольку данные входных и выходных параметров могут формироваться и использоваться в функциях, еще не имеющих моделей.



Пример схемы алгоритма

- Размер фрагмента трассы – 6,7 млн шагов
- Время автоматизированного построения с учетом необходимости задания моделей аналитиком – около 2 часов.
- Время «ручного» построения (отладчик + дизассемблер) – несколько недель (оптимистично)


access
fname
pname
CreateFileW
[0E8FB344 -- 0E91A239]
hfile
hfile ^
ReadFile
[0E931807 -- 0E933196]
buf
_SHA1_Begin
[0E947590 -- 0E94759D]
outsha1ctx
addinput
addsize
paddinput
pretbytes
prng_ctx
retsize
prng_generateNewBytes
[0EF37214 -- 0EF3A429]
retbytes
insha1ctx ^
buf ^
bsize
_SHA1_Update
[0E9475A8 -- 0E9475F9]
outsha1ctx
buf ^
insha1ctx ^
_SHA1_Update
[0E94760D -- 0E947617]
outsha1ctx
src ^
memcpy
[0EF3A434 -- 0EF3A45F]
dst
insha1ctx ^
_SHA1_Begin
[0EF22AA9 -- 0EF22AB1]
sha1ctx
_SHA1_End
[0E94761D -- 0E948432]
outhash
hfile ^
ReadFile
[0EC8991A -- 0EC8C76F]
buf
src ^
insha1ctx ^
buf ^
bsize
_SHA1_Begin
[0EF52D61 -- 0EF52D69]
sha1ctx
memcpy
[0EF3A9DD -- 0EF3AA0A]
dst
_SHA1_Update
[0EF22AB6 -- 0EF22AFA]
outsha1ctx
insha1ctx ^
bsize
buf ^
_SHA1_Update
[0EF52D6E -- 0EF52DB2]
insha1ctx ^
_SHA1_Begin
[0EF270FA -- 0EF27102]
_SHA1_End
[0EF22B01 -- 0EF23301]

Выводы

- Формальное применение обратного слайсинга для выделения алгоритма приводит к большому числу «лишних» инструкций
- Для исключения перепометок требуется:
  - промежуточное представление, позволяющее учитывать связи между входами/выходами конкретных инструкций
  - учет эффекта уничтожения зависимостей вследствие математических свойств алгоритма
  - применение прямого слайсинга по результатам обратного. Позволяет значительно (до двух порядков) сократить число «лишних» инструкций с хорошими перспективами по устранению оставшихся «лишних» инструкций с помощью оптимизирующих преобразований
- Для исключения недопометок требуется:
  - полносистемная трасса или usermode-трасса с заданием связей по входу/выходу в точках вызовов системных сервисов
  - модель взаимодействия с устройствами в/в
  - учет адресных зависимостей и зависимостей по управлению
- Даже свободный от «лишних» инструкций слайс в практических случаях непригоден для восприятия человеком. Выход - декомпозиция слайса на блоки инструкций и построение схемы алгоритма.