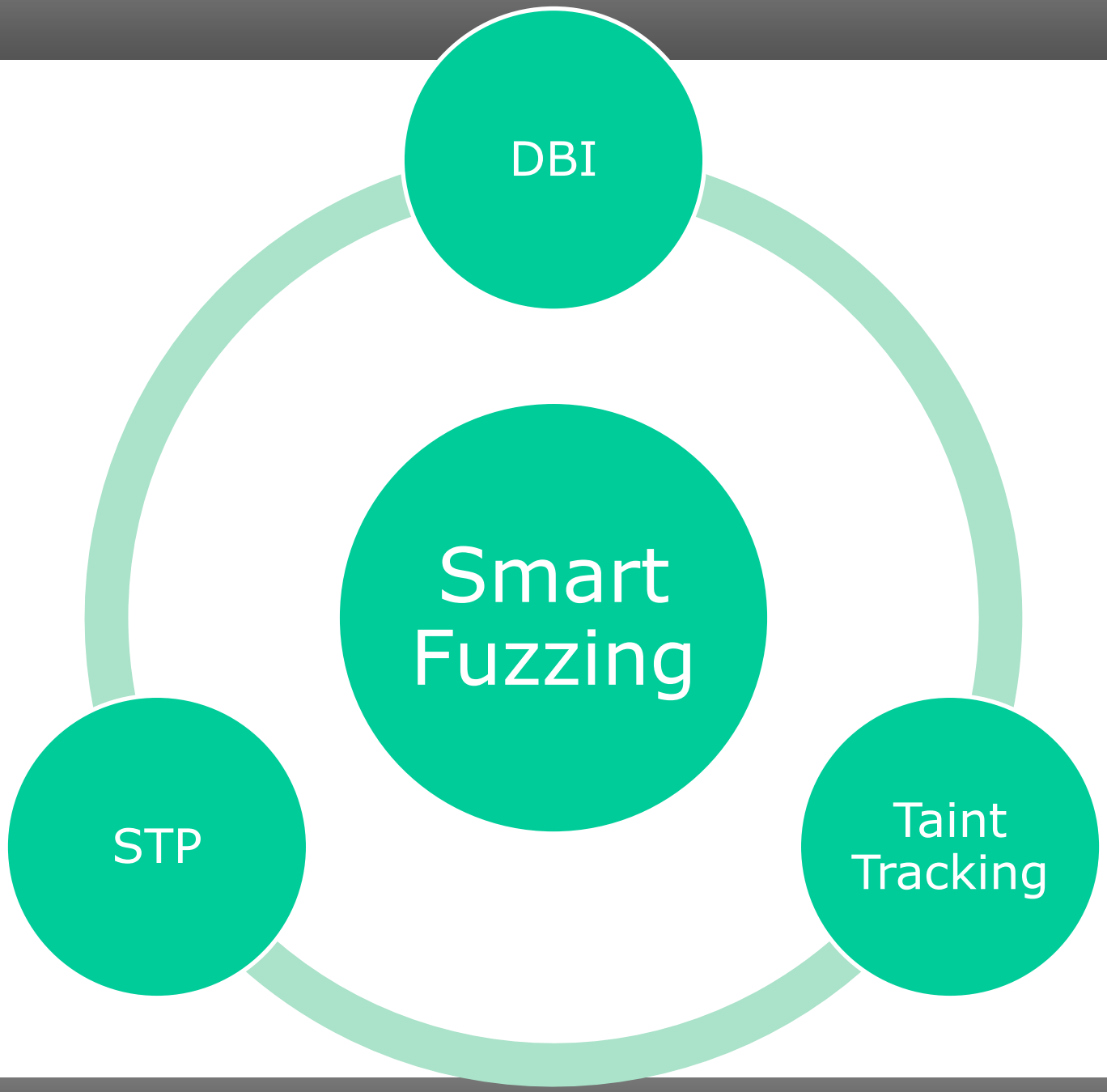
TÌM KIẾM LỖ HỔNG PHẦN MỀM BẰNG KĨ THUẬT FUZZING THÔNG MINH

suto[at]vnsecurity[dot]net

suto[at]bkitsec[dot]vn

```
$ id
uid=69(suto),666(bkitsec),1337(vnsec)
```


DBI
Smart Fuzzing
STP
Taint Tracking

Săn lỗi ứng dụng

Phân tích tĩnh

Fuzzing

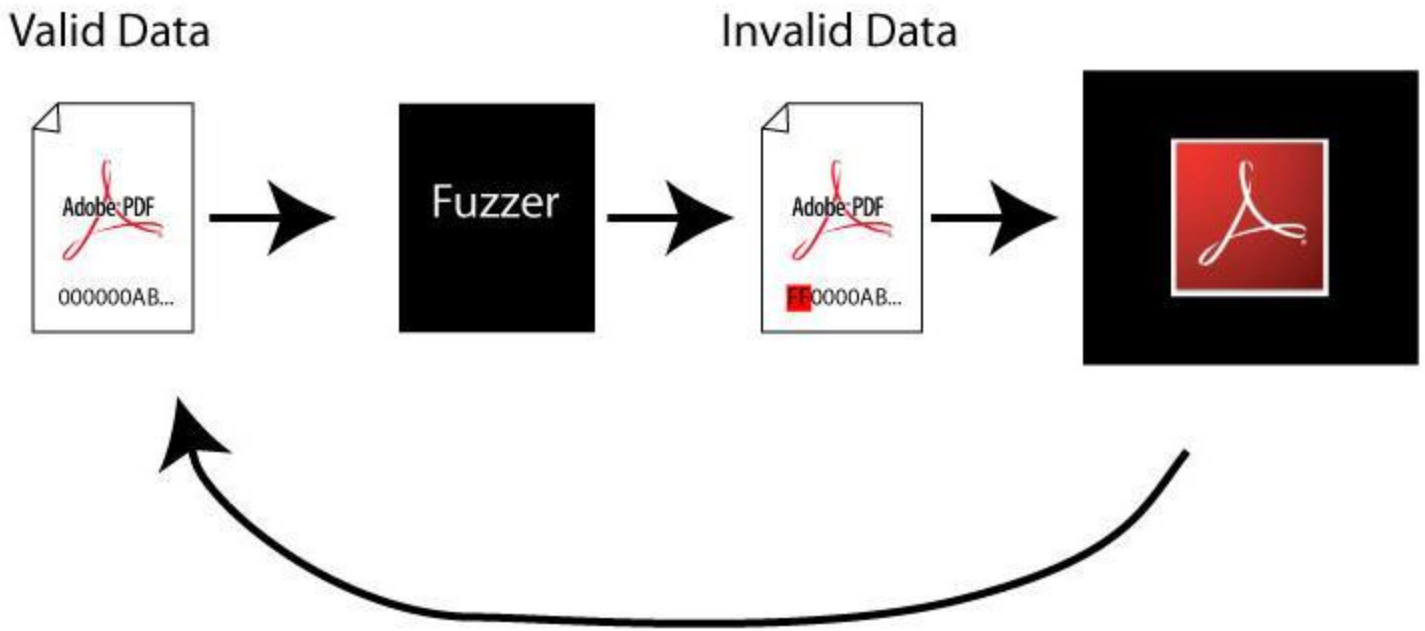
Phân tích tĩnh

Điều kiện

Mã nguồn

Mã máy

Fuzzing


Valid Data
Adobe PDF
000000AB...
Fuzzer
Invalid Data
Adobe PDF
FF0000AB...

ví dụ

Đầu vào:

Một con vật tùy thích

Chương trình:

Ve_con_vat_trong_hinh(hinh)

KẾT QUẢ HI VỌNG

Fuzzing

Đầu vào :

.....   

Lặp I = 1 -> N :

Ve_con_vat_trong_hinh(hinh)

Định dạng con mèo

```
<ConMeo>
<Dau><Tai><Dai>2</Dai><Rong>4</Rong></Tai>
…
</Dau>
<Minh>…..</Minh>
<Chan>
    <Chantruoc>…</Chantruoc>
    …
</Chan>
</ConMeo>
```

Phương thức vẽ con vật

```
 <snip>…
giay = Chuan_bi_giay(cat.dau.tai.dodai * cat.dau.tai.dorong)
copy_tai_vao_giay( tai_meo, giay)
…
<snip>
```

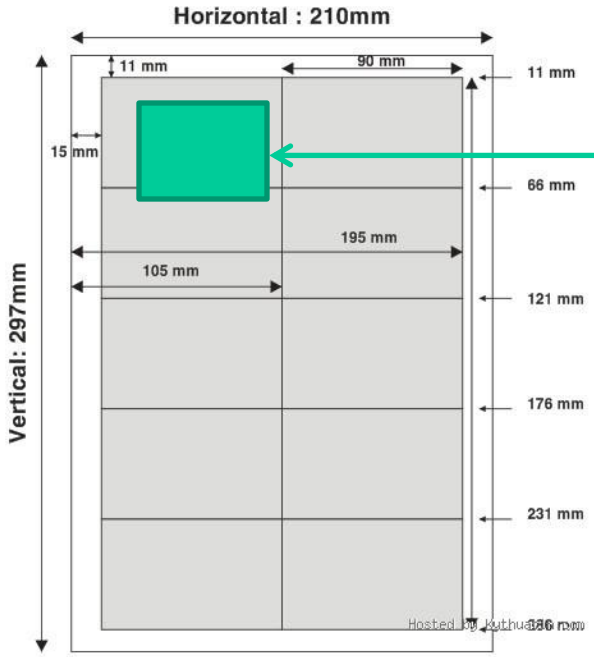
Kết quả

…<tai><dai>0xFFFFFFFE</dai><rong>0x2</rong>

chuan_bi_giay(0xFFFFFFFE*0x2)

copy_tai_vao_giay()

????????



Và trong thực tế

```
<image>
   <width>A</width>
   <high>B</high>
</image>
```

Trong quá trình fuzz thực sự A và B sẽ luôn thay đổi……

Trong mã nguồn

```
char* buffer  = malloc(AxB)
strcpy(buffer, image)
```

BUFFER OVERFLOW

Adobe Reader Image Data Buffer Allocation Integer Overflow Remote Code Execution Vulnerability

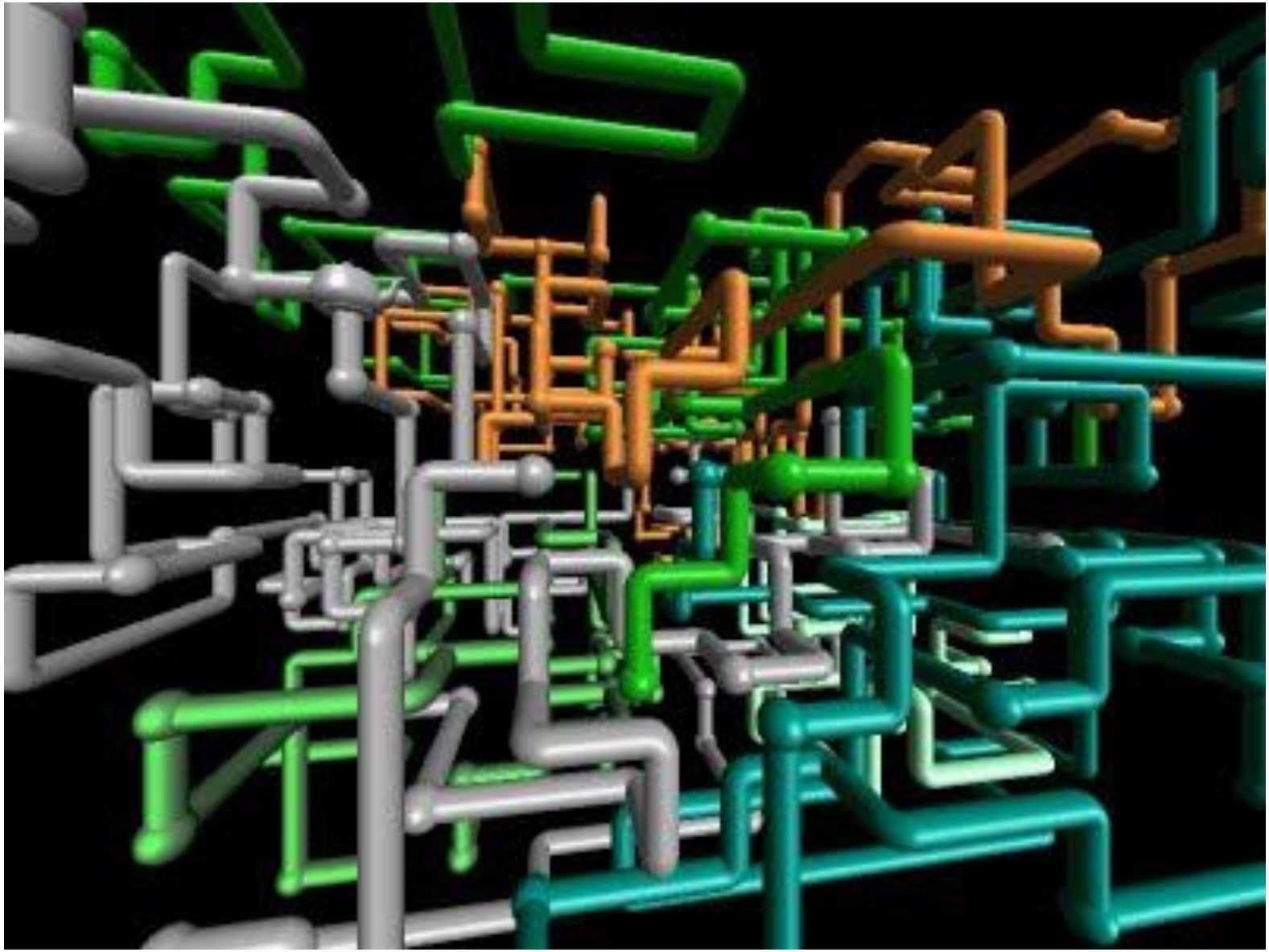
ZDI-11-283: October 13th, 2011

The specific flaw exists within the Adobe Image parsing library. When Adobe Reader tries to parse an malformed .BMP image with bitfields encoded image data an integer overflow can occur while calculation the size of the image data. This can result in a heap buffer overflow with remote code execution under the context of the current user.

Và còn phức tạp hơn thế nữa ☹

Một chương trình sẽ thao tác trên nhiều loại tập tin khác nhau

Một chương trình còn cho phép nhúng các loại tập tin khác nhau lên nó.

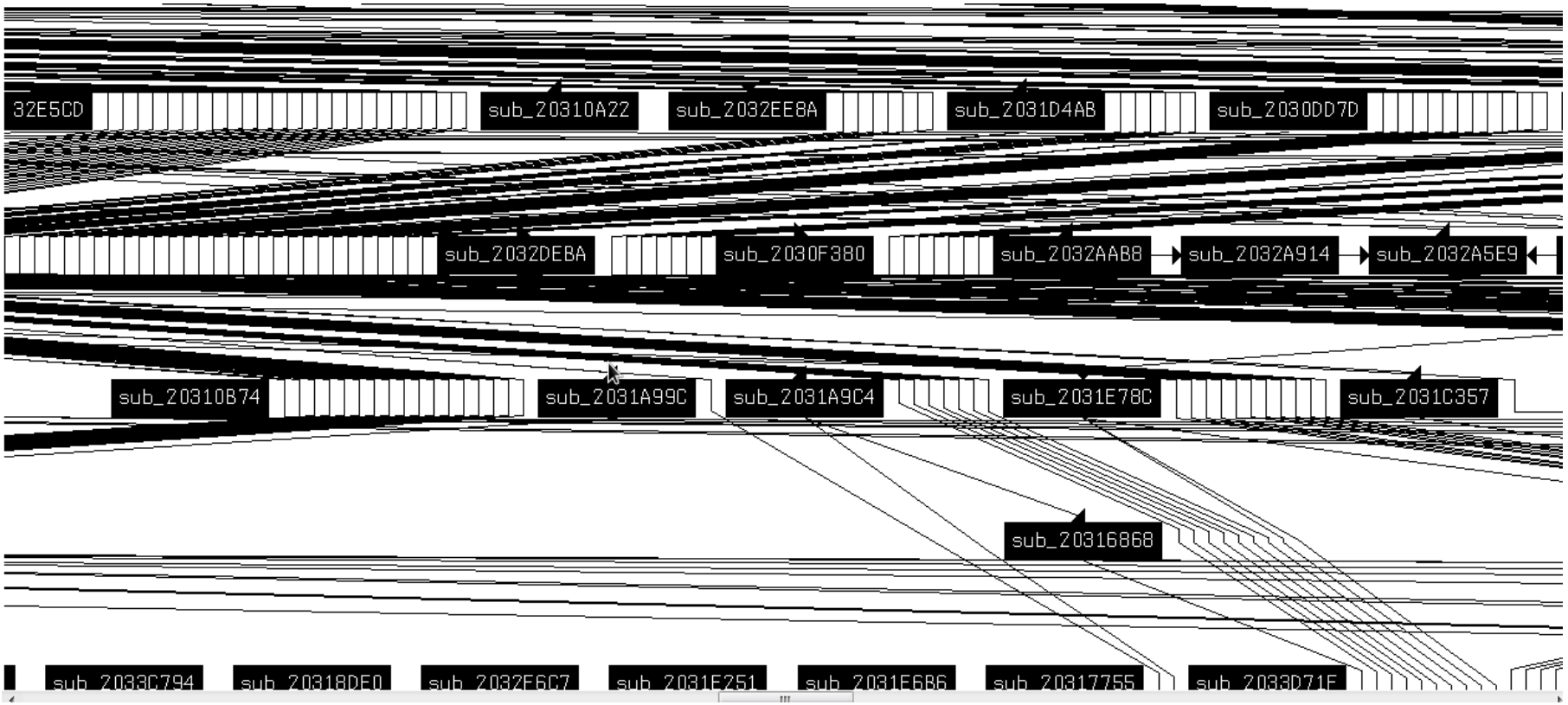


Ví dụ

Nếu là thỏ:
    Cho xem tai
    Vecontho(contro)

Nếu là nai:
    Cho xem gạc
    Veconnai(contro)

switch/case

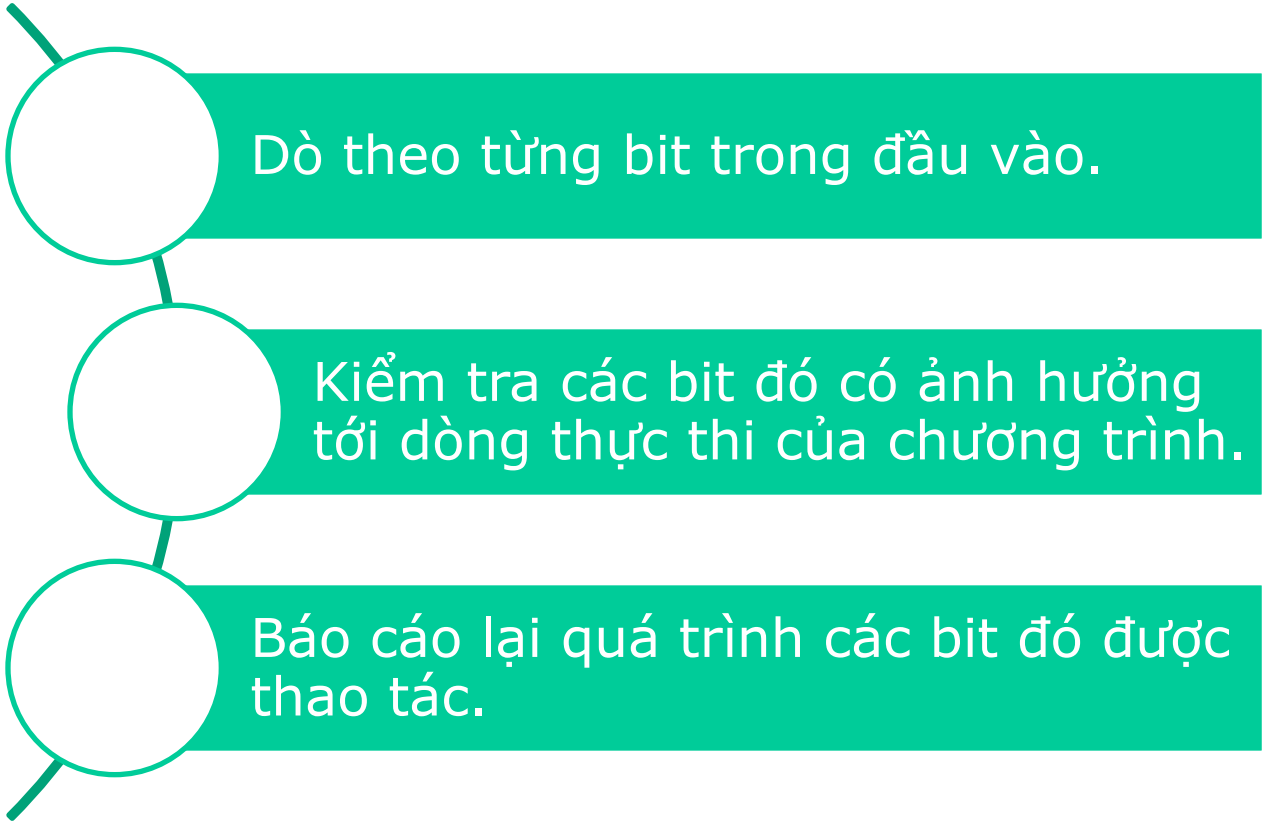


Vấn đề đặt ra

Một chương trình không chỉ đơn giản chỉ có 1 hay vài phép nhảy có điều kiện.

Cho mẫu thử chỉ là 1 trong các loại tệp hoặc chỉ có 1 loại thẻ mà chương trình xử lí được, làm sao để tạo ra những mẫu khác mà chương trình chấp nhận/chứa những thẻ khác nhau.

Chúng ta cần gì



Cần một người hùng

và...

Cuối cùng mục tiêu chính là làm sao tạo ra các dữ liệu phù hợp để điều khiển dòng thực thi đi hết vào các ngóc ngách của chương trình.

Tuổi thơ……

3 điều ước

I
- Dynamic binary instrumentation

II
- Dynamic taint tracking

III
- Decision Procedure for Bit-Vectors and Arrays

Thao tác chương trình trực tiếp

Chèn thêm mã thực thi vào mã máy của chương trình.

Phân tích hoạt động của chương trình trong lúc thực thi.

Có đầy đủ thông tin lúc chương trình thực thi.

Các vấn đề được giải quyết

- Chương trình lấy dữ liệu từ những đâu(từ tập tin mẫu, từ mạng, từ người dùng…..)
- Chương trình đã làm những gì trên những dữ liệu đó(tính toán, so sánh, kiểm tra, thêm, bớt …).

Các framework hỗ trợ hiện nay

Valgrind

The DR. is in.

Ví dụ sử dụng pin

```
// The running count of instructions is kept here
// make it static to help the compiler optimize docount
static UINT64 icount = 0;

// This function is called before every instruction is executed
VOID docount() { icount++; }

// Pin calls this function every time a new instruction is encountered
VOID Instruction(INS ins, VOID *v)
{
    // Insert a call to docount before every instruction, no arguments are passed
    INS_InsertCall(ins, IPOINT_BEFORE, (AFUNPTR)docount, IARG_END);
}

KNOB<string> KnobOutputFile(KNOB_MODE_WRITEONCE, "pintool",
    "o", "inscount.out", "specify output file name");

// This function is called when the application exits
VOID Fini(INT32 code, VOID *v)
{
    // Write to a file since cout and cerr maybe closed by the application
    ofstream OutFile;
    OutFile.open(KnobOutputFile.Value().c_str());
    OutFile.setf(ios::showbase);
    OutFile << "Count " << icount << endl;
    OutFile.close();
}
```

Và kết quả

bash$ pin -t inscount0.so -- /bin/ls

bash$ cat inscount.out

Count 422838

Dò theo dữ liệu đầu vào

Mục tiêu

Kiểm tra xem dòng chảy của chương trình có bị ảnh hưởng bởi dữ liệu đầu vào không.

Dữ liệu này đã được tính toán như thế nào cho tới lúc nó được kiểm tra.

DÒ THEO GIÁ TRỊ ĐẦU VÀO

Ví dụ:

```
int main(int argc,char* argv[]){
char* buffer = (char*)malloc(256);
readFile("user_file.tst",buffer);
printf("%s\n",buffer);
if(strncmp(buffer,"1",1)==0)
    printf("Case 1\n");
if(strncmp(buffer,"2",1)==0)
    printf("Case 2\n");
```

```
.text:004010F0 _main            proc near               ; CODE XREF: __tmainCRTStartup+10A↓p
.text:004010F0
.text:004010F0 argc             = dword ptr  4
.text:004010F0 argv             = dword ptr  8
.text:004010F0 envp             = dword ptr  0Ch
.text:004010F0
.text:004010F0                  push    ebx
.text:004010F1                  push    esi
.text:004010F2                  push    edi
.text:004010F3                  push    100h            ; fContent
.text:004010F8                  call    ds:__imp__malloc
.text:004010FE                  mov     esi, eax
.text:00401100                  push    esi             ; fName
.text:00401101                  call    ?readFile@@YAIPAD0@Z ; readFile(char *,char *)
.text:00401106                  mov     ebx, ds:__imp__printf
.text:0040110C                  push    esi
.text:0040110D                  push    offset Format   ; "%s\n"
.text:00401112                  call    ebx ; __imp__printf
.text:00401114                  mov     edi, ds:__imp__strncmp
.text:0040111A                  push    1               ; MaxCount
.text:0040111C                  push    offset Str2     ; "1"
.text:00401121                  push    esi             ; Str1
.text:00401122                  call    edi ; __imp__strncmp
.text:00401124                  add     esp, 1Ch
.text:00401127                  test    eax, eax
.text:00401129                  jnz     short loc_401135
.text:0040112B                  push    offset aCase1   ; "Case 1\n"
.text:00401130                  call    ebx ; __imp__printf
.text:00401132                  add     esp, 4
.text:00401135
.text:00401135 loc_401135:                              ; CODE XREF: _main+39↑j
.text:00401135                  push    1               ; MaxCount
.text:00401137                  push    offset a2       ; "2"
.text:0040113C                  push    esi             ; Str1
.text:0040113D                  call    edi ; __imp__strncmp
.text:0040113F                  add     esp, 0Ch
.text:00401142                  test    eax, eax
.text:00401144                  jnz     short loc_401150
.text:00401146                  push    offset aCase2   ; "Case 2\n"
.text:0040114B                  call    ebx ; __imp__printf
.text:0040114D                  add     esp, 4
.text:00401150
.text:00401150 loc_401150:                              ; CODE XREF: _main+54↑j
.text:00401150                  pop     edi
.text:00401151                  pop     esi
.text:00401152                  xor     eax, eax
.text:00401154                  pop     ebx
.text:00401155                  retn
.text:00401155 _main            endp
.text:00401155
.text:00401156
.text:00401156 ; =============== S U B R O U T I N E =======================================
.text:00401156
```

Điều cuối cùng....

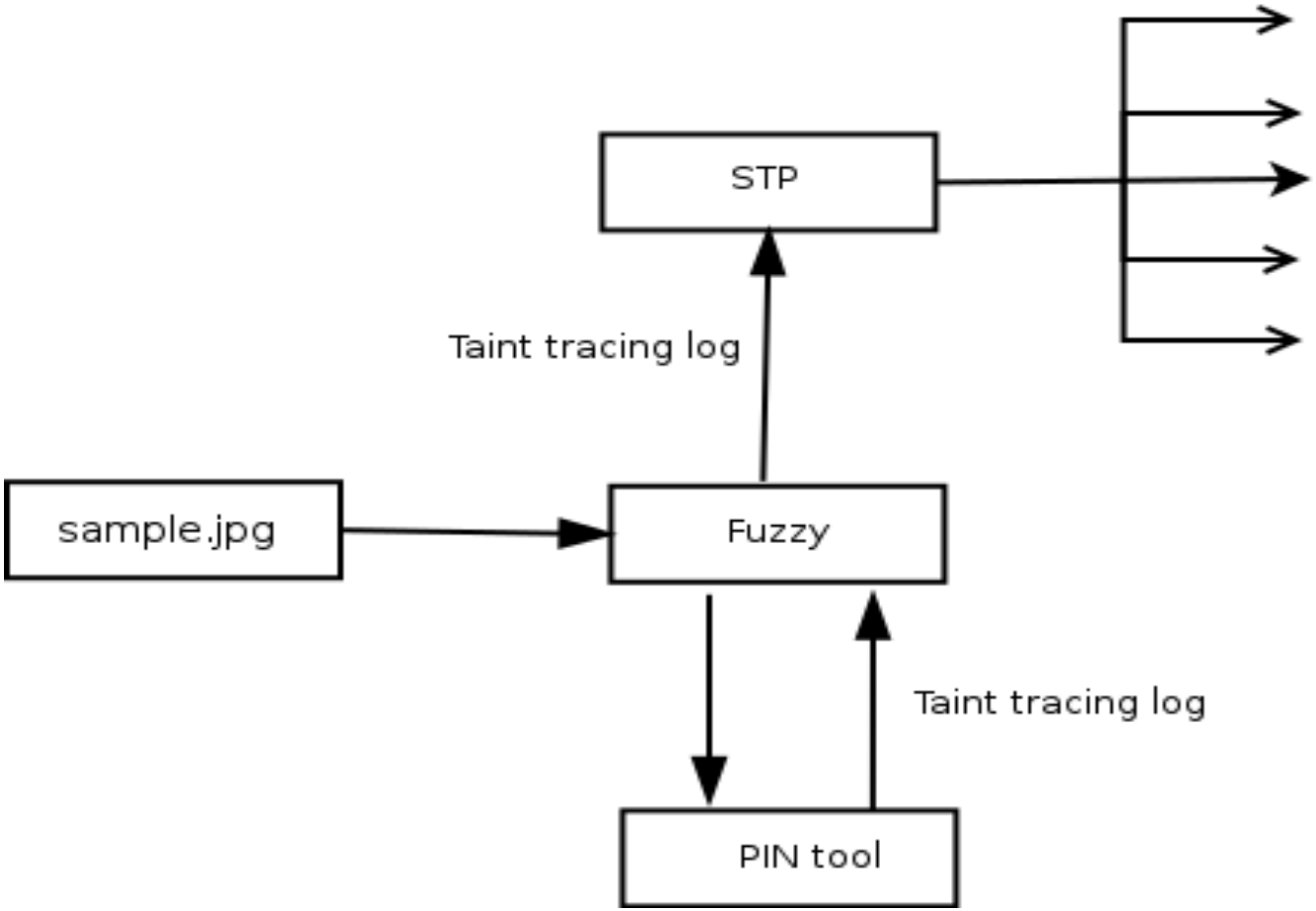
Dựa vào những thông tin đã có, tính toán ra được các giá trị dữ liệu ban đầu phù hợp để có thể đi được vào hết các ngõ ngách của ứng dụng.

STP

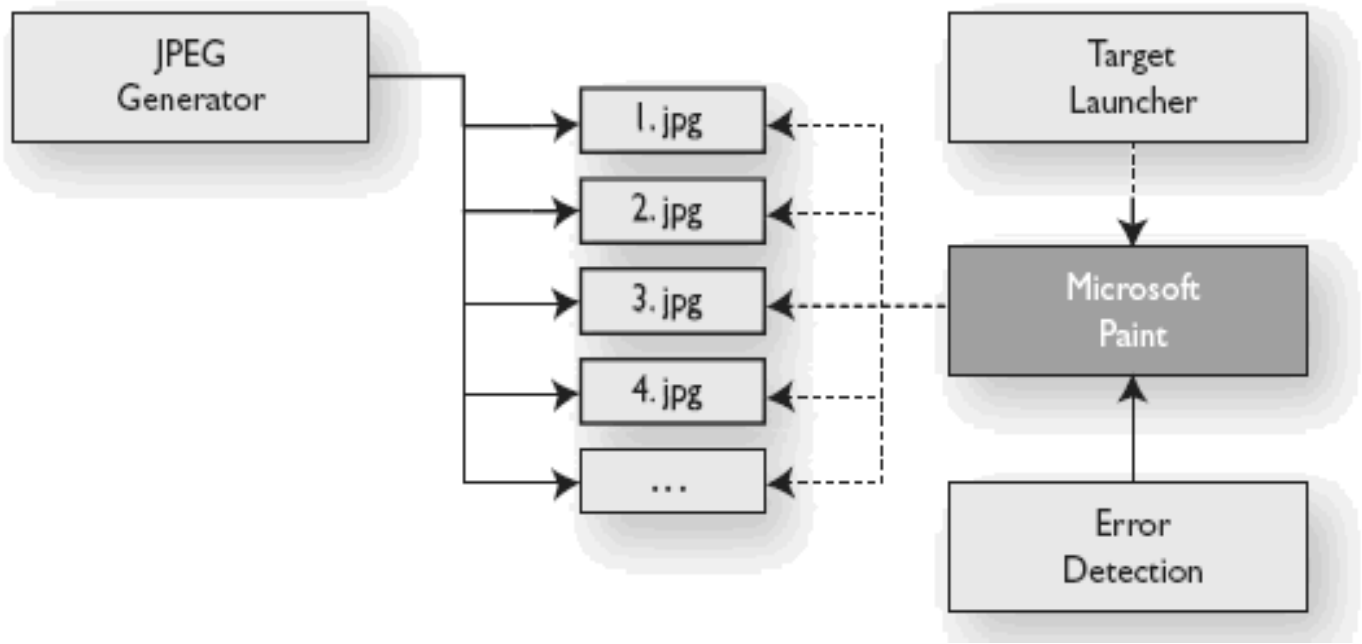
STP sẽ nhận input là các phép toán.

Lần lượt kiểm tra nếu các QUERY có thể thỏa mãn được, nếu có sẽ tính toán ra các giá trị đầu tiên và tạo ra một tập tin mới với những giá trị đó.

Từ Mẫu đầu tiên



Fuzz thông qua các MẫU con



Tổng kết

Dynamic taint tracking sẽ ghi lại tất cả các hoạt động tính toán trên các bit của dữ liệu đầu vào và các phép kiểm tra, các lệnh nhảy đồng thời ghi dữ liệu này với định dạng đầu vào cho STP.

STP sẽ dựa vào dữ liệu bên trên để kiểm tra tại các điều nhảy xem có dữ liệu đầu vào nào thỏa mãn không, nếu có sẽ sinh ra các tập tin mới.

Các tập tin mới này sẽ được đưa vào chương trình một lần nữa để xác nhận là nó đã đi vào những điểm khác so với dữ liệu gốc ban đầu.

Các tập tin mới sẽ được đưa vào fuzzer để trực tiếp fuzz.

Cảm ơn !

Q & A

http://vnsecurity.net
http://bkitsec.vn