RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: HTA-W02

POISON PIXELS: COMBATTING IMAGE STEGANOGRAPHY IN CYBERCRIME

Dr Simon R Wiseman

CTO
Deep Secure
@srw_deepsecure

Poison Pixels

- Steganography in cyber attacks
- What is it?
- Why is it a problem?
- How does it work?
- Where might it be?
- What can we do about it?

Resources online : https://rsa2018.deep-secure.com

STEGANOGRAPHY

What is it?

Steganography is…

The word *steganography* combines the Greek words *steganos*, meaning "covered, concealed, or protected" and *graphein* meaning "writing".

- Steg = Concealed writing

How well concealed does a message have to be to count as Steganography?

Steganography is…

- You can see the symbols and they make sense
- But there's another meaning that isn't obvious

Looks like an anodyne statement

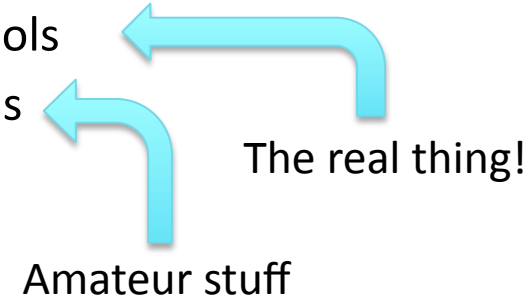
John has a long moustache

But is there a hidden meaning?

(the invasion starts tomorrow)

How Un-Obvious?

- Steganography is data with double meaning
  - One overt
  - One hidden
- Cannot discover presence of hidden meaning with normal tools
- Two grades of steganography
  - Can't be detected, even with specialist tools
  - Discoverable using specialist analysis tools

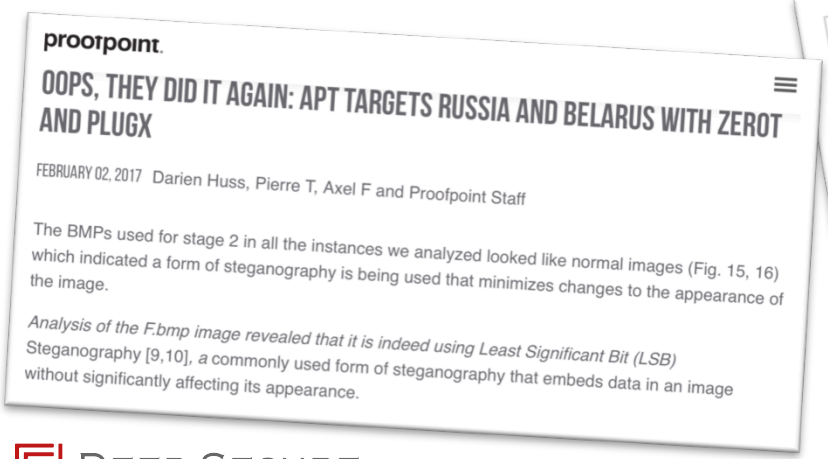


STEGANOGRAPHY

Why is it a problem?

Stegware

- Attackers use Steg to evade detection
  - Hiding dangerous looking code
  - Hiding command and control
  - Hiding exfiltrated sensitive data



proofpoint.

OOPS, THEY DID IT AGAIN: APT TARGETS RUSSIA AND BELARUS WITH ZEROT AND PLUGX

FEBRUARY 02, 2017 Darien Huss, Pierre T, Axel F and Proofpoint Staff

The BMPs used for stage 2 in all the instances we analyzed looked like normal images (Fig. 15, 16) which indicated a form of steganography is being used that minimizes changes to the appearance of the image.

Analysis of the F.bmp image revealed that it is indeed using Least Significant Bit (LSB) Steganography [9,10], *a* commonly used form of steganography that embeds data in an image without significantly affecting its appearance.

Stegano exploit kit poisoning pixels

BY PETER STANCIK POSTED 6 DEC 2016 - 12:00PM

ESET researchers have discovered a new exploit kit spreading via malicious ads on a number of reputable news websites, each with millions of visitors daily. Since at least the beginning of October 2016, the bad guys have been targeting users of Internet Explorer and scanning their computers for vulnerabilities in Flash Player. Exploiting these flaws in the code, they have been attempting to download and execute various types of malware.

Hiding Dangerous Code

- Code appended to an image file

A small boring image

With a hidden message

blue.gif

00	47	49	46	38	39	61	0F	00	10	00	80	00	00	00	5A	FF	00	00	00	2C	00	00	00	00	0F	00	10	00	00	02	0D	84
20	8F	A9	CB	ED	0F	A3	9C	B4	DA	8B	73	2B	00	3B	27	22	0D	0A	41	64	64	2D	54	79	70	65	20	2D	41	73	73	65
40	6D	62	6C	79	4E	61	6D	65	20	50	72	65	73	65	6E	74	61	74	69	6F	6E	43	6F	72	65	2C	50	72	65	73	65	6E
60	74	61	74	69	6F	6E	46	72	61	6D	65	77	6F	72	6B	0A	5B	53	79	73	74	65	6D	2E	57	69	6E	64	6F	77	73	2E
80	4D	65	73	73	61	67	65	42	6F	78	5D	3A	3A	53	68	6F	77	28	20	27	50	6F	69	73	6F	6E	20	50	69	78	65	6C
A0	73	21	27	20	29	0D	0A																									

GIF89a Ä Z˘ , Ñ
è©ÀÌ £ú¥/ãs+ ;'" Add-Type -Asse
mblyName PresentationCore,Presen
tationFramework [System.Windows.
MessageBox]::Show('Poison Pixel
s!')

Anti-Virus scanning: Sees harmless image

Covert Command & Control

Attacker back at base

- Hide command in an image
- Compose a Tweet including image
- Add agreed hash tag to Tweet
- Send Tweet

Attack in target system

- Poll Twitter for agreed hash tag
- Fetch Tweets, extract attached image
- Extract command from image
- Execute command

Network monitoring: Sees Tweets fetched from Twitter

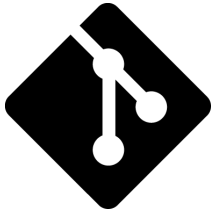
Covert Egress

Attacker back at base

- Send URL of a drop box to the target
  - Using Steg in a Tweet
  - Different and varied destinations
- Wait for data to be uploaded to URL
- Decode stolen data from image

Attack in target system

- Receive URL via a Tweet
- Put stolen data in an image using Steg
- Upload image to URL



Network monitoring: Sees Tweets and occasional harmless uploads

STEGANOGRAPHY

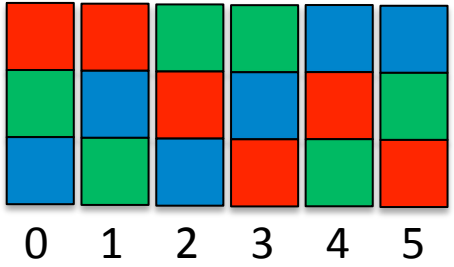
How does it work?

Encoding Information in Redundant Data

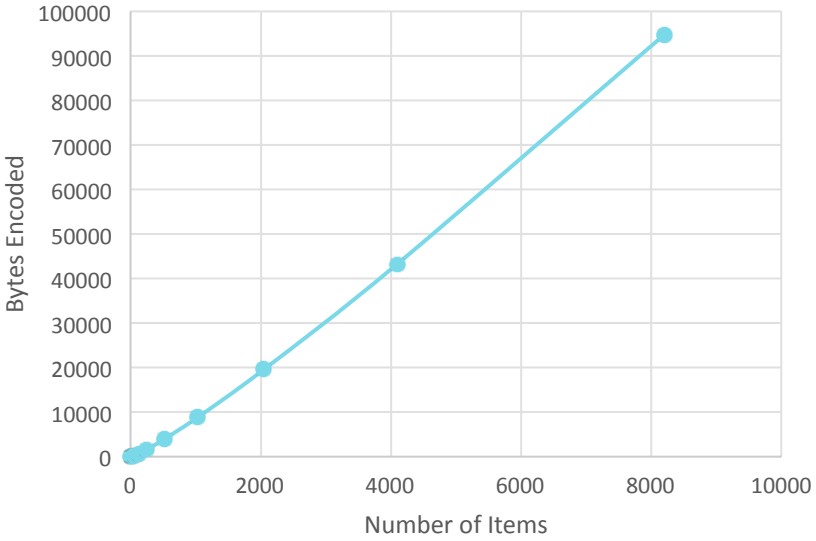
- Irrelevant data
  - Data appended to a file, e.g the GIF we've just seen
- The order of lists that implement sets
  - Order of colours in a palette, e.g. a GIF
- Redundant encodings
  - Duplicate colours in a palette, e.g. a GIF
- The low order bits of an audio / visual signal
  - Low order bits of a True Colour image, e.g. a PNG

Order of Lists that Implement Sets

- Order of colours in a palette



3 x 2 x 1 = 6 possible orderings
equivalent to $\log^2(6) = 2.58$ bits



8’ palette has 256! possible orderings
equivalent to $\log^2(256!) = 1683.99$ bits = 210bytes

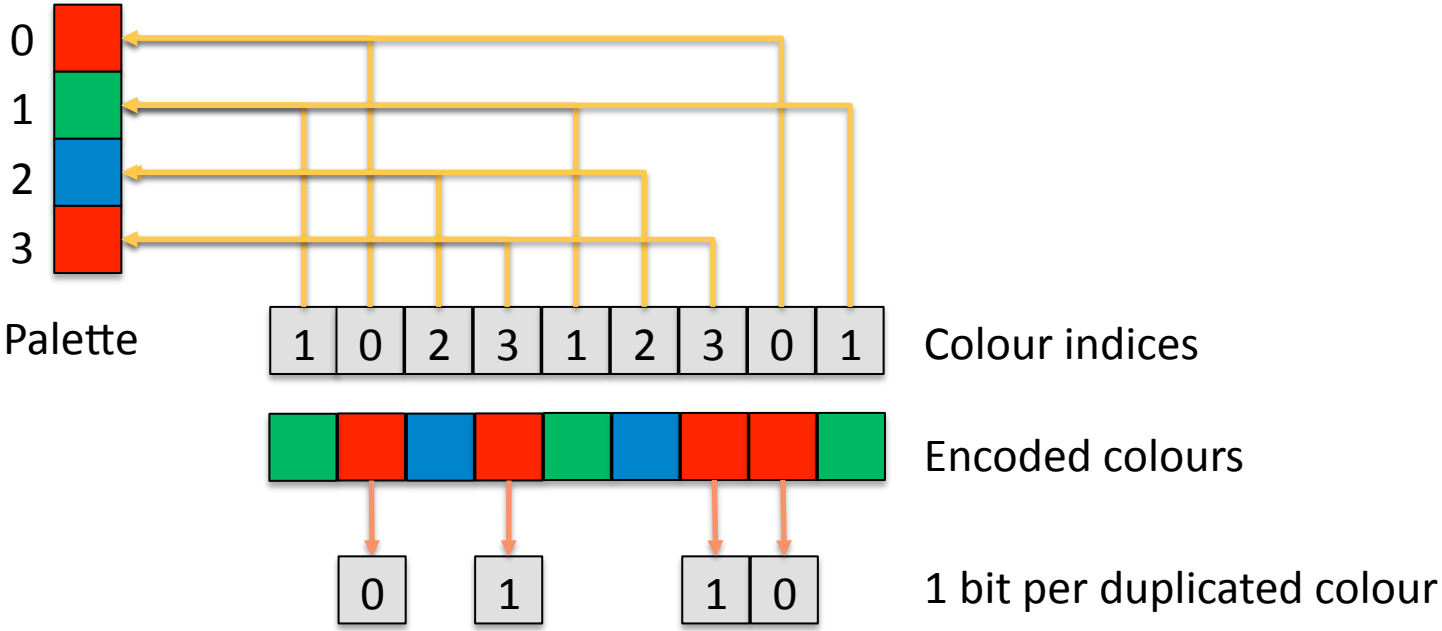
Exponential growth in capacity

Using Palette Ordering

- 256-colour BMP, GIF & PNG
  - Encoding/decoding algorithm complicated, but given in Knuth
  - Offsets to palette and pixel data (can be) fixed
  - Encoding requires pixel data to be adjusted – no problem for infiltration

Redundant Encodings

- Duplicate colours in a palette

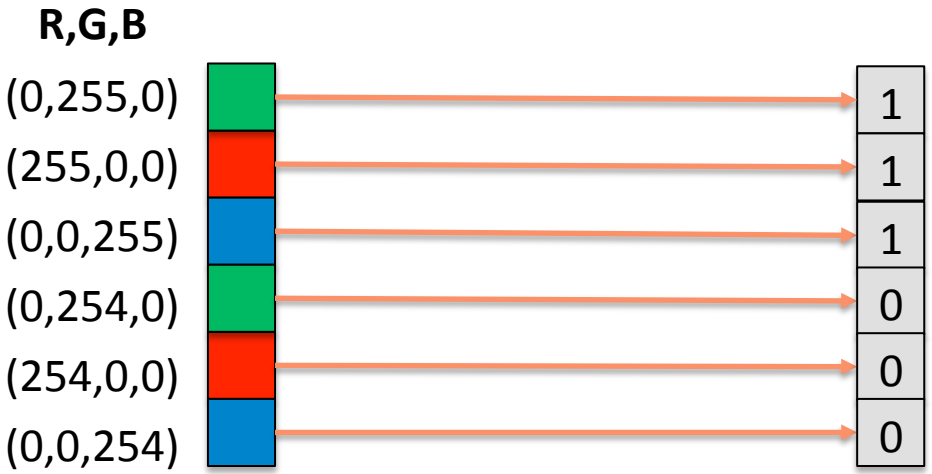


Using Duplicate Colours

- 256-colour BMP
  - Easy encoding/decoding algorithm
  - Palette with duplicates in fixed index positions can be agreed
  - Offset to pixel data can be fixed
- 256-colour GIF & PNG
  - Harder to encode/decode as pixel data is compressed

Low Order Bits in Audio / Visual Signal

- Low order bits of a True Colour image



Encoded colours

4032 × 3024 pixels
Hiding 725kBytes of text
"Pride and Prejudice"

1 bit per pixel per colour component
= 3 bits per pixel

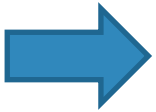
Using Low Order Bits

- 24’ BMP
  - Easy encoding/decoding algorithm
- 24’ PNG & JPEG
  - Difficult encoding/decoding algorithm as pixel data is compressed
    - But image libraries could be used

JPEG Compression

- JPEG compression uses a DCT transformation then Huffman encoding

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$



$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

8x8 block of pixel values, one colour component

Not very compressible

DCT coefficients

Highly compressible

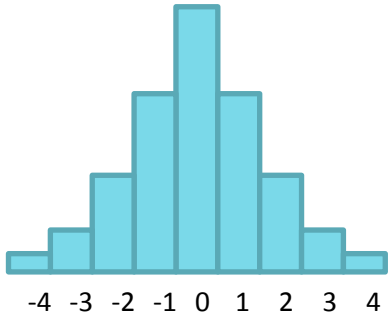
JPEG Least Significant Bit Steganography

- Exact values of DCT coefficients are not very important visually
  - Information can be encoded in LSB of coefficients

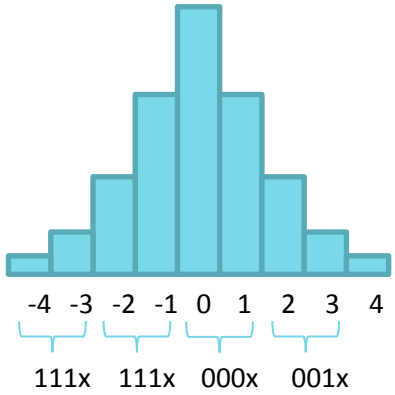
$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -26 & -3 & -6 & 3 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

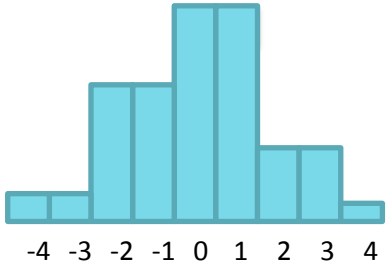
JPEG Steganalysis

- LSB steganography in DCT is detectable to a degree



Natural images have normal distribution of coefficients



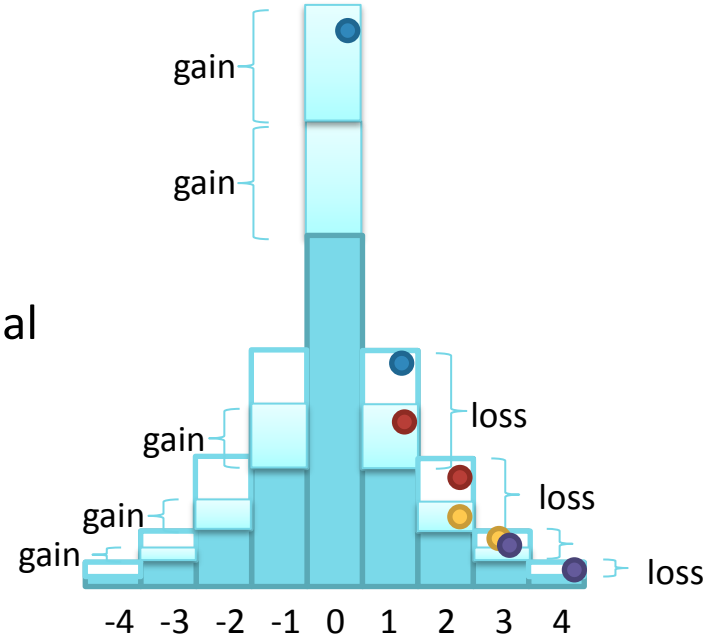


Randomising least significant bits flattens out the histogram

Chi^2 test measures flattening

F3 Steganography Algorithm

- F3 decrements absolute value
  - Cannot encode values in zero coefficients
  - Preserves symmetry of histogram
  - Defeats Chi^2 test
- Greatly increases zeroes
  - Detectable because slope gradient is unusual



Coefficient Swapping

- Information encoded in relative ordering of coefficients
  - Does not change coefficient values, so preserves first order statistics
  - Reduced capacity but undetectable

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & 2 & -1 & 0 & 0 & 0 \\ -3 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

R > G => 0

R < G => 1

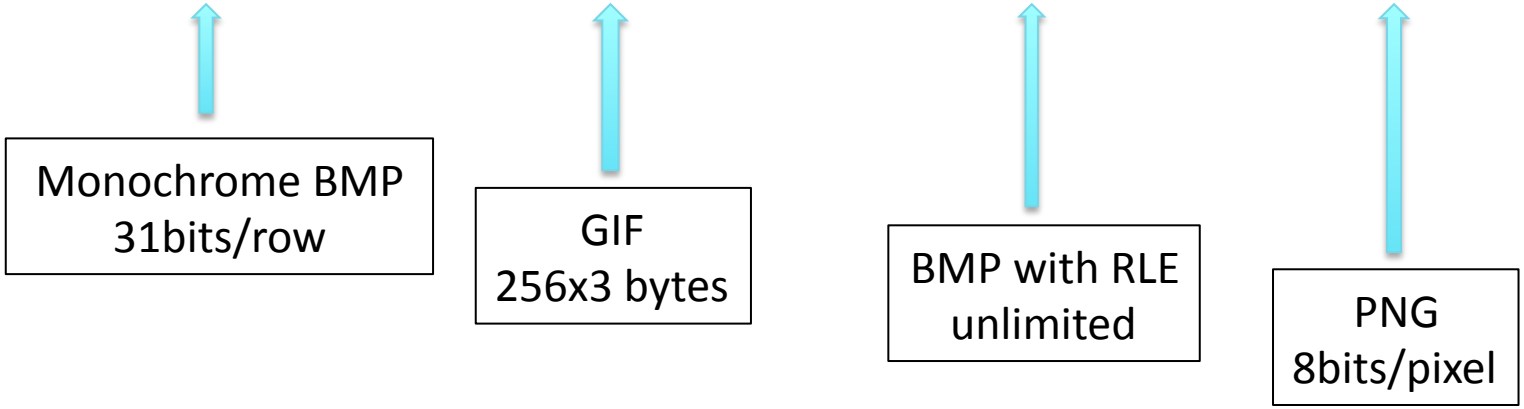
10 pairs => 3 bytes/64 pixels

STEGANOGRAPHY

Where might it be?

Opportunities for Hiding

- There's a Steg opportunity in every file format…
- Images
  - Colour palette ordering, Pixel Least Significant Bit, Coefficient ordering
  - Scan line padding, Redundant palettes, Pixel overwriting, Transparency



Opportunities for Hiding

- There's a Steg opportunity in every file format
- JSON, XML, Office
  - Attribute ordering

```
<w:bottom w:val="single" w:sz="8" w:space="4" w:color="4F81BD" />
<w:bottom w:color="4F81BD" w:space="4" w:sz="8" w:val="single" />
```

4 attributes => $\log_2(4!) = 24$ => 4 bits

  - Whitespace
    - Space, tab, linefeed, carriage return => 4 codes = 2 bits per character

Opportunities for Hiding

- There's a Steg opportunity in every file format
- Plain Text
  - Unicode combining diacritics vs. precomposed characters

y + ˘ = ў

Cyrillic y + breve = ў

(U+0443) (U+02D8) (U+045E)

STEGANOGRAPHY

What can we do about it?

Detection

- Detectable steganography
  - Well yes, it's detectable
  - But can be difficult to avoid false positives
- Undetectable steganography
  - Er well, it's not detectable
  - At best, only with considerable false positives and false negatives
  - At worst, completely invisible to the eye and to analysis
  - No more difficult to encode/decode than detectable steg

Detection isn't going to defeat Stegware

Annihilation

- Don’t try to detect it
- Eliminate the places it hides
- Remove redundant data
- Replace redundant data

Normalisation

Only use one way to encode information

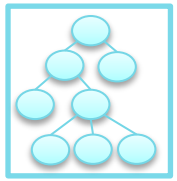
Content Threat Removal

- CTR is a method of defeating attacks in digital content
- Does not rely on detection
- Transforms the way information is represented
- Annihilates steg as a by-product

Extraction not Detection

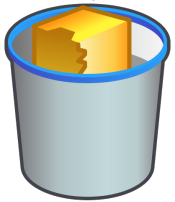
1 Data arrives

2 Information extracted



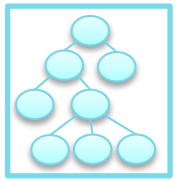
3 Original data always discarded

(whether it is safe or not)



Build New not Fix Up Old

4 Information extracted



5 New data built for delivery

Business information delivered
Any attacks discarded

Stegware Removal

- CTR Transformation process naturally normalises data
  - Eliminates irrelevant data
  - Orders structures in a fixed way
  - Always uses same encoding

  } Annihilates discoverable steganography

- Easy addition to remove audio/visual redundancy
  - Reduce resolution to match usage
  - Replace redundant bits with zero

  } Annihilates undetectable steganography

WRAPPING UP

Summary

- Steganography is being used by cyber attackers
- Detection strategies are flawed
  - You can't detect steganography when it is done properly
- Look for strategies that annihilate steganography
  - Not just re-writing images to eliminate cross-site attacks

When you get back to the office…

- If you are accepting images from the public
  - Check your servers are washing the images properly
- If you allow social media
  - Keep it away from sensitive data and systems
- Start thinking that detection is not the answer

THANK YOU FOR COMING ALONG

Follow up at the Deep Secure stand #4522 in North Expo

Questions?

simon.wiseman@deep-secure.com