Angelo Prado
Neal Harris
Yoel Gluck

SSL, GONE IN 30 SECONDS

A **BREACH** beyond **CRIME**

AGENDA

Proceed with caution:

- ✓ Review of **CRIME**
- ✓ Introducing **BREACH**
- ✓ In the **weeds**
- ✓ **Demo** time!
- ✓ **Mitigations**

PREVIOUSLY…

CRIME

Presented at ekoparty 2012

Juliano Rizzo
Thai Duong

Target

Secrets in HTTP headers

Requirements

TLS compression
MITM
A browser

SO ABOUT **CRIME…**

The Compression Oracle:

- ✓ SSL doesn't hide **length**
- ✓ SSL/SPDY **compress headers**
- ✓ **CRIME** issues requests with every possible character, and measures the ciphertext **length**
- ✓ Looks for the **plaintext which compresses the most** – guesses the secret byte by byte
- ✓ Requires small **bootstrapping** sequence

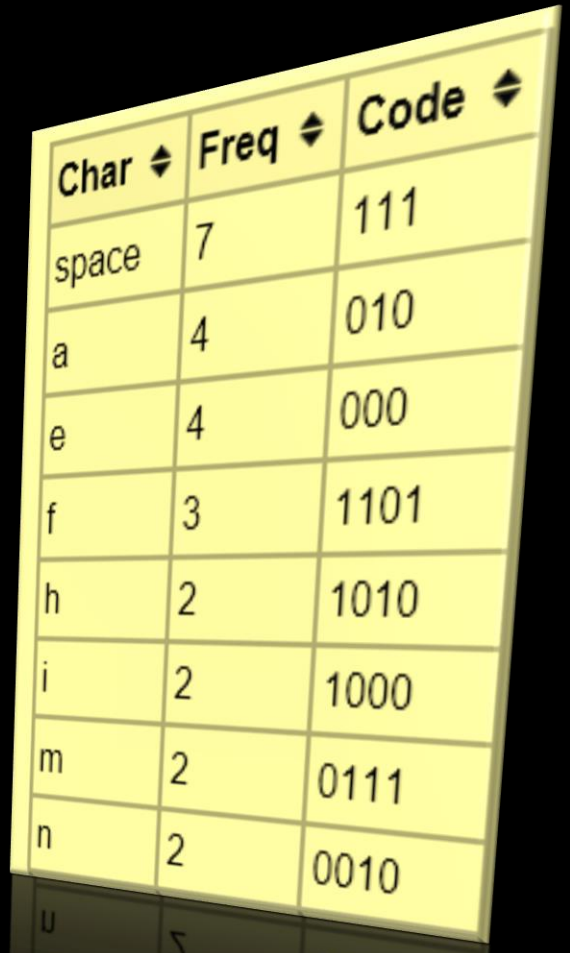
  knownKeyPrefix=secretCookieValue

COMPRESSION OVERVIEW

- ✓ DEFLATE / GZIP
  - LZ77: reducing redundancy

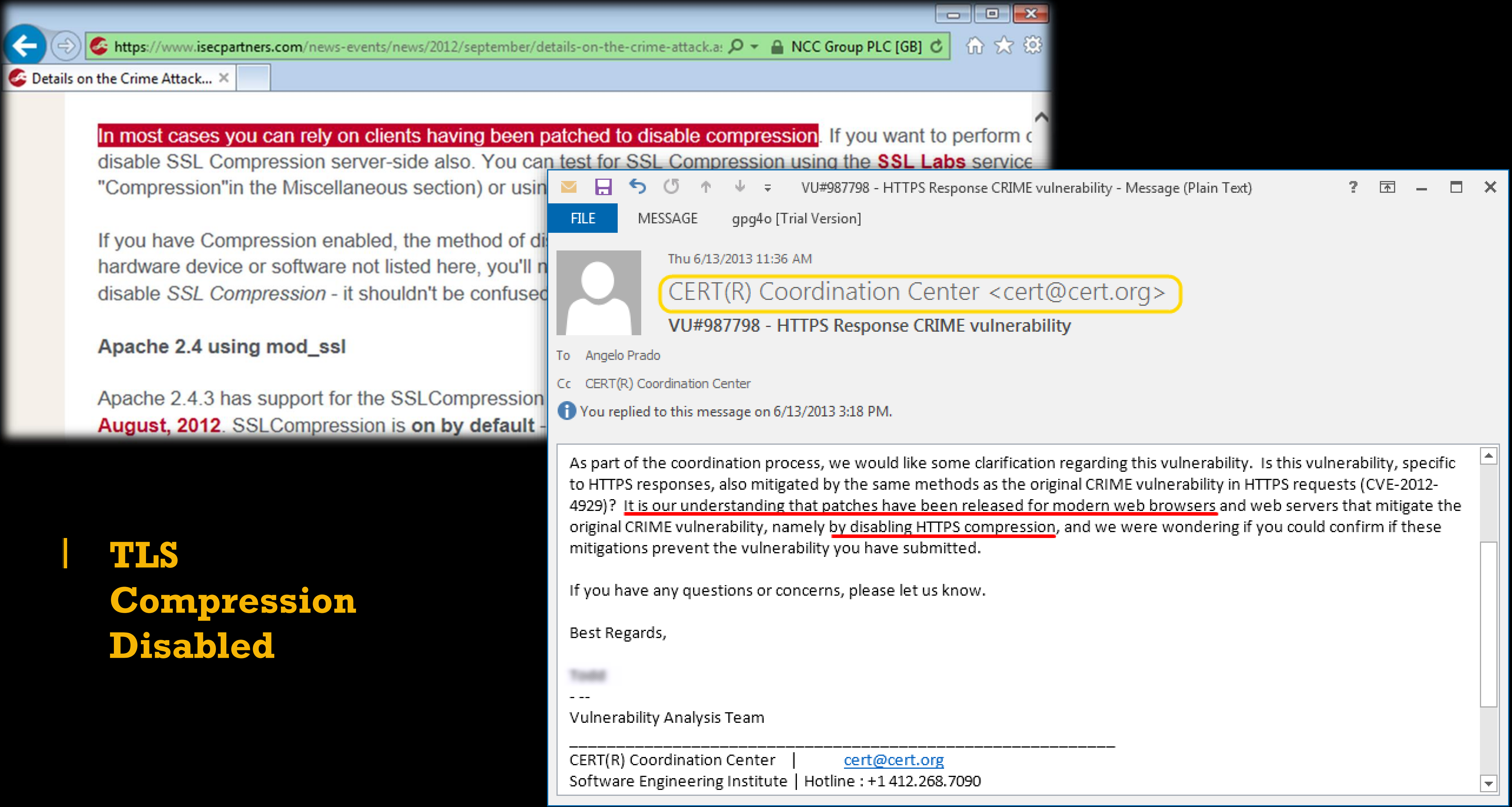
    Googling the googles -> Googling the g(-13,4)s

  - Huffman coding: *replace common bytes with shorter codes*



Char	Freq	Code
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010
i	2	1000
m	2	0111
n	2	0010

“ IT’S FIXED!



https://www.isecpartners.com/news-events/news/2012/september/details-on-the-crime-attack.a
NCC Group PLC [GB]
Details on the Crime Attack...

In most cases you can rely on clients having been patched to disable compression. If you want to perform c
disable SSL Compression server-side also. You can test for SSL Compression using the **SSL Labs** service
"Compression"in the Miscellaneous section) or usin

If you have Compression enabled, the method of di
hardware device or software not listed here, you'll n
disable *SSL Compression* - it shouldn't be confused

Apache 2.4 using mod_ssl

Apache 2.4.3 has support for the SSLCompression
August, 2012. SSLCompression is **on by default** -

VU#987798 - HTTPS Response CRIME vulnerability - Message (Plain Text)
FILE
MESSAGE
gpg4o [Trial Version]

Thu 6/13/2013 11:36 AM

CERT(R) Coordination Center <cert@cert.org>

VU#987798 - HTTPS Response CRIME vulnerability

To Angelo Prado

Cc CERT(R) Coordination Center

You replied to this message on 6/13/2013 3:18 PM.

As part of the coordination process, we would like some clarification regarding this vulnerability. Is this vulnerability, specific to HTTPS responses, also mitigated by the same methods as the original CRIME vulnerability in HTTPS requests (CVE-2012-4929)? It is our understanding that patches have been released for modern web browsers and web servers that mitigate the original CRIME vulnerability, namely by disabling HTTPS compression, and we were wondering if you could confirm if these mitigations prevent the vulnerability you have submitted.

If you have any questions or concerns, please let us know.

Best Regards,

[redacted]

- --
Vulnerability Analysis Team

____
CERT(R) Coordination Center | cert@cert.org
Software Engineering Institute | Hotline : +1 412.268.7090

- **TLS Compression Disabled**

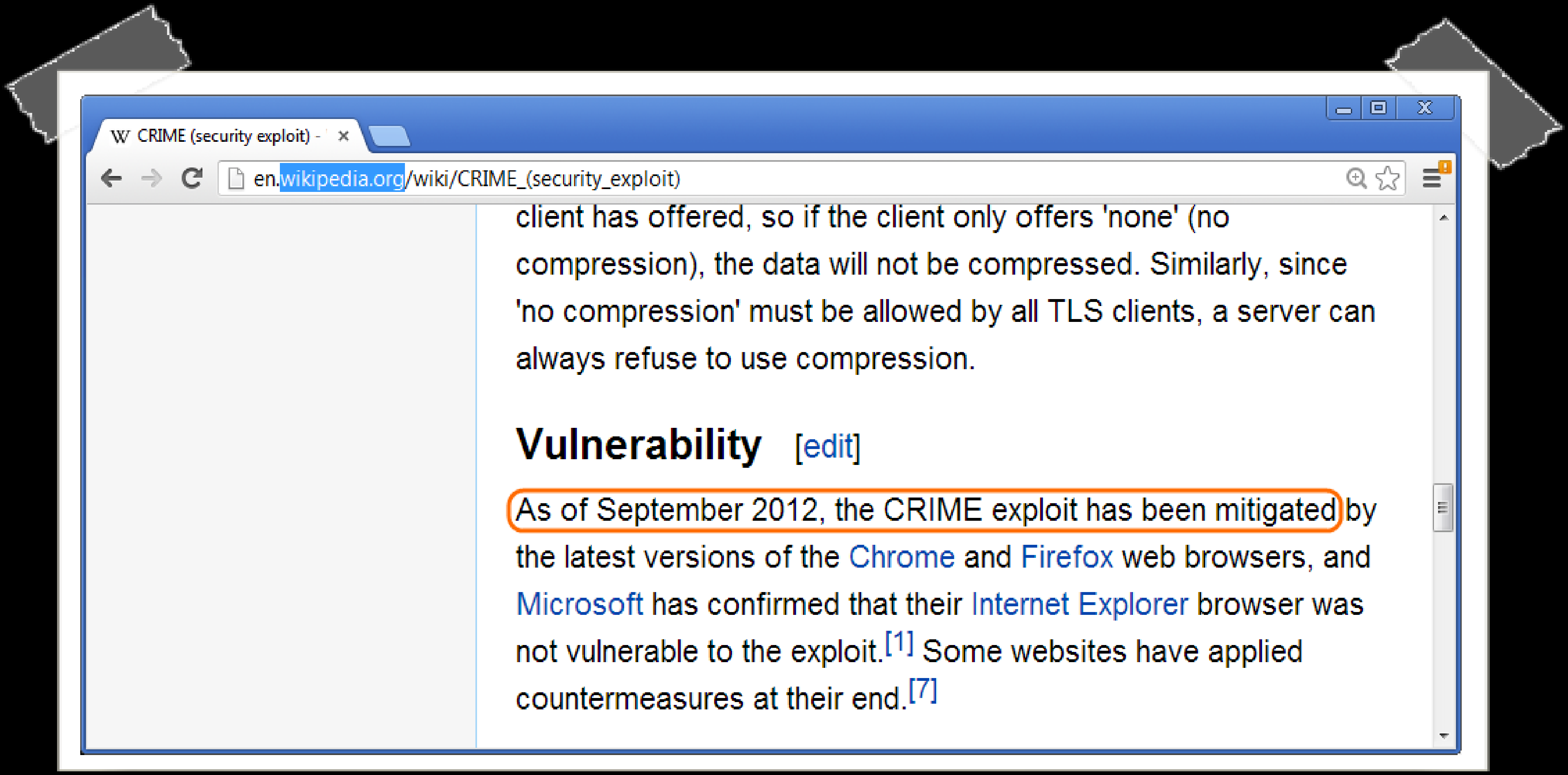
DO NOT PANIC:

TUBES SECURE

Or are they?

[let's bring it back to life]

FIRST THINGS FIRST: FIX WIKIPEDIA


CRIME (security exploit) -
en.wikipedia.org/wiki/CRIME_(security_exploit)

client has offered, so if the client only offers 'none' (no compression), the data will not be compressed. Similarly, since 'no compression' must be allowed by all TLS clients, a server can always refuse to use compression.

Vulnerability [edit]

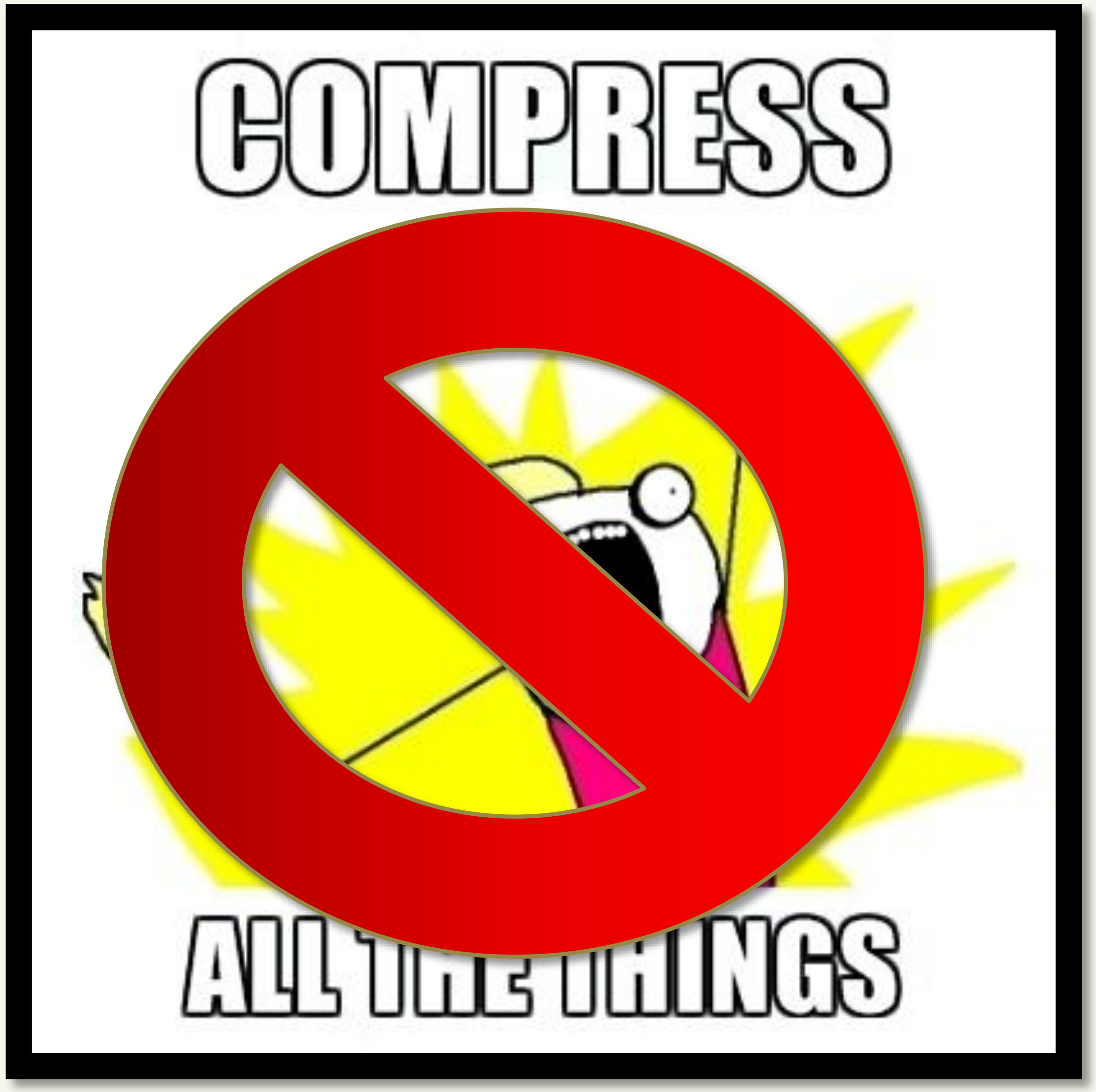
As of September 2012, the CRIME exploit has been mitigated by the latest versions of the Chrome and Firefox web browsers, and Microsoft has confirmed that their Internet Explorer browser was not vulnerable to the exploit.[1] Some websites have applied countermeasures at their end.[7]

NEW!

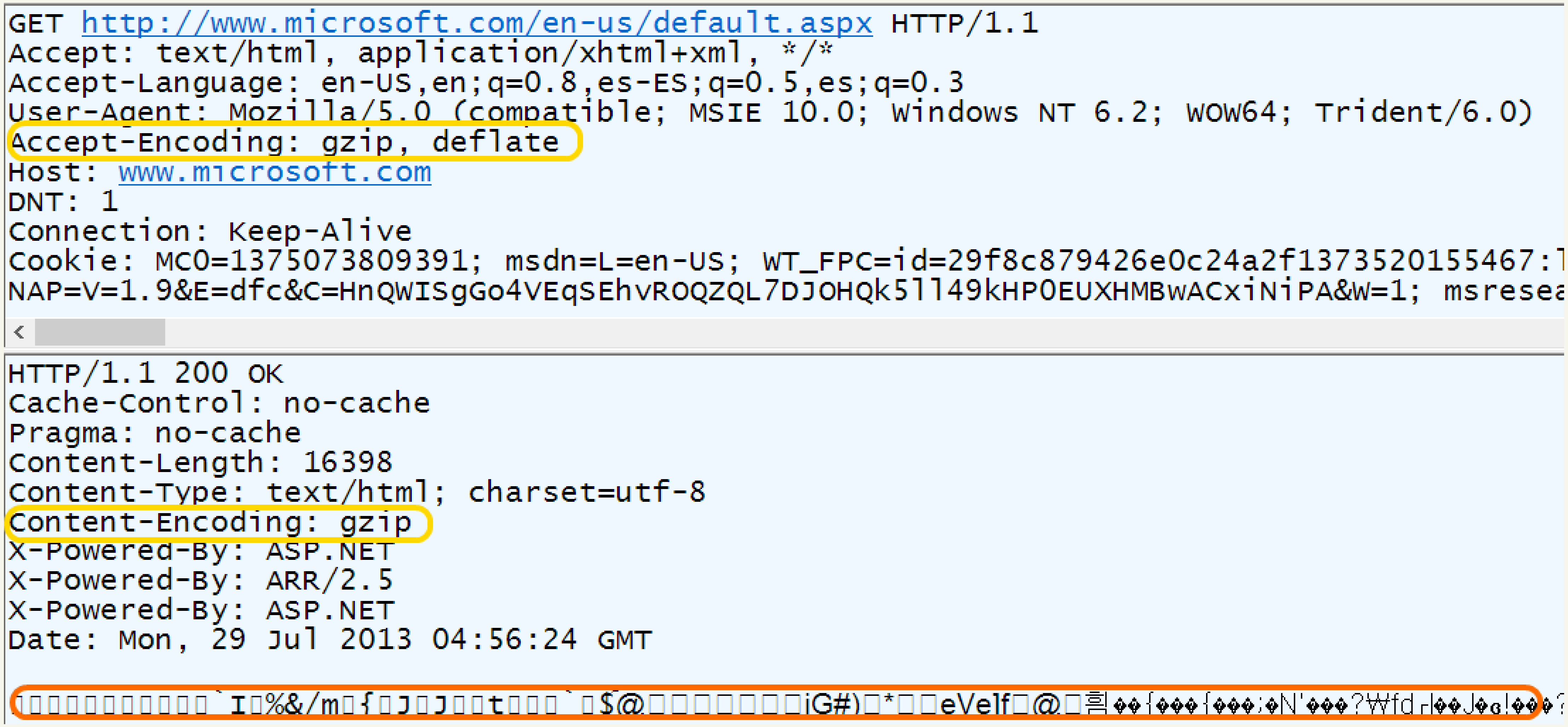
INTRODUCING BREACH

Browser **R**econnaissance & **E**xfiltration via **A**daptive **C**ompression of **H**ypertext

A CRIME AGAINST THE RESPONSE BODY



(sample traffic)



```
GET http://www.microsoft.com/en-us/default.aspx HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US,en;q=0.8,es-ES;q=0.5,es;q=0.3
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0)
Accept-Encoding: gzip, deflate
Host: www.microsoft.com
DNT: 1
Connection: Keep-Alive
Cookie: MC0=1375073809391; msdn=L=en-US; WT_FPC=id=29f8c879426e0c24a2f1373520155467:1
NAP=V=1.9&E=dfc&C=HnQWISgGo4VEqSEhvROQZQL7DJOHQk5l149kHPOEUXHMBwACxiNiPA&W=1; msresea
```

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 16398
Content-Type: text/html; charset=utf-8
Content-Encoding: gzip
X-Powered-By: ASP.NET
X-Powered-By: ARR/2.5
X-Powered-By: ASP.NET
Date: Mon, 29 Jul 2013 04:56:24 GMT

□□□□□□□□□□□□`I□%&/m□{□J□J□□t□□□`□$@□□□□□□□□□iG#)□*□□eVelf□@□흼��{���{���;�N'���?₩fd┌l��J�ɞ!���?
```

BREACH / the ingredients

| GZIP

· Very **prevalent,** any **browser**

| Fairly stable pages

· **Less than 30 seconds** for simple pages

| MITM / traffic visibility

· No SSL tampering / downgrade

| A secret in the response body

· CSRF, PII, ViewState… anything!

| Attacker-supplied guess

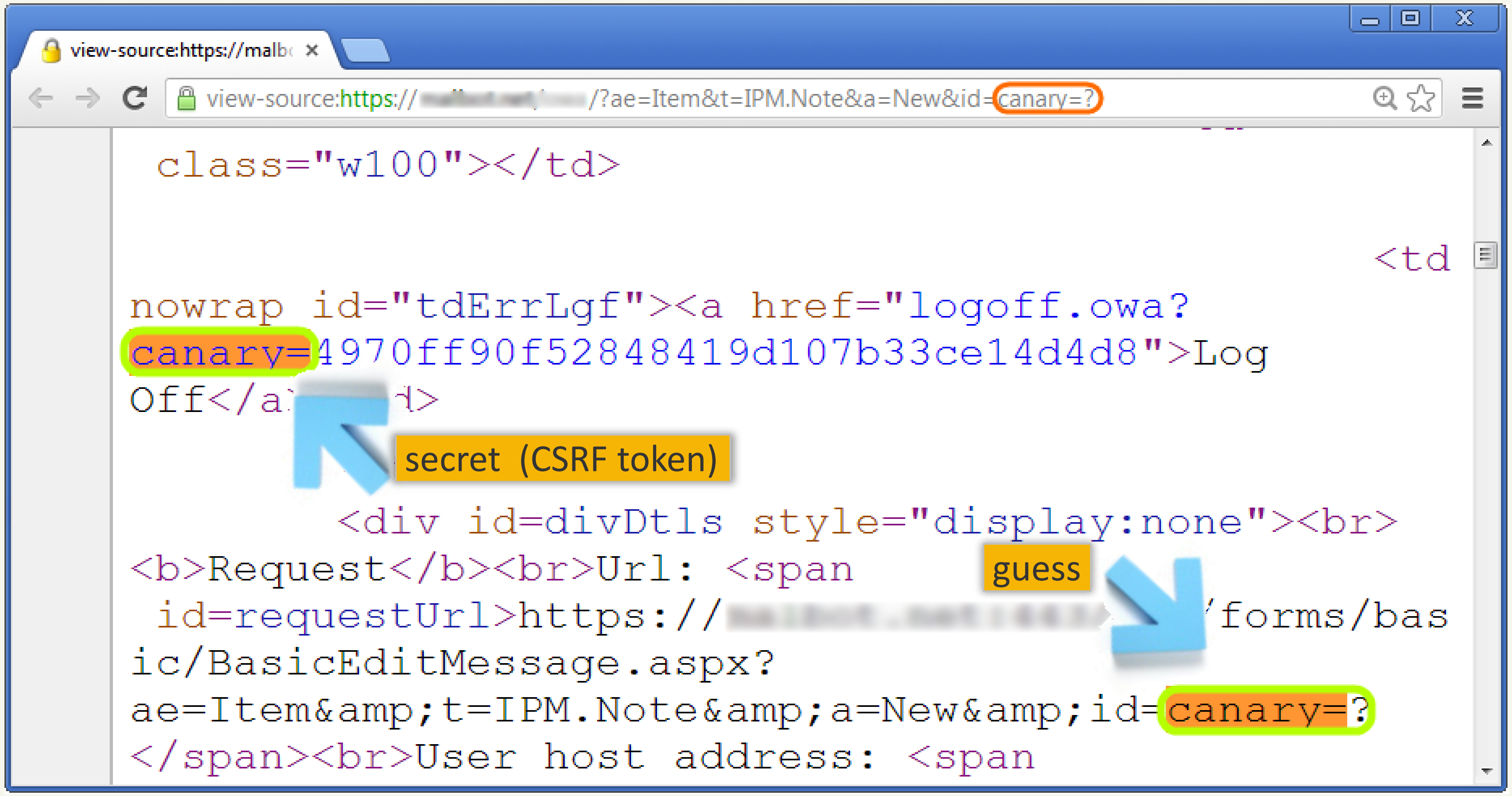
· In **response body**

| Three-character prefix

· To **bootstrap compression**

| Any version of SSL / TLS

[PREFIX / sample bootstrap]



view-source:https://malb

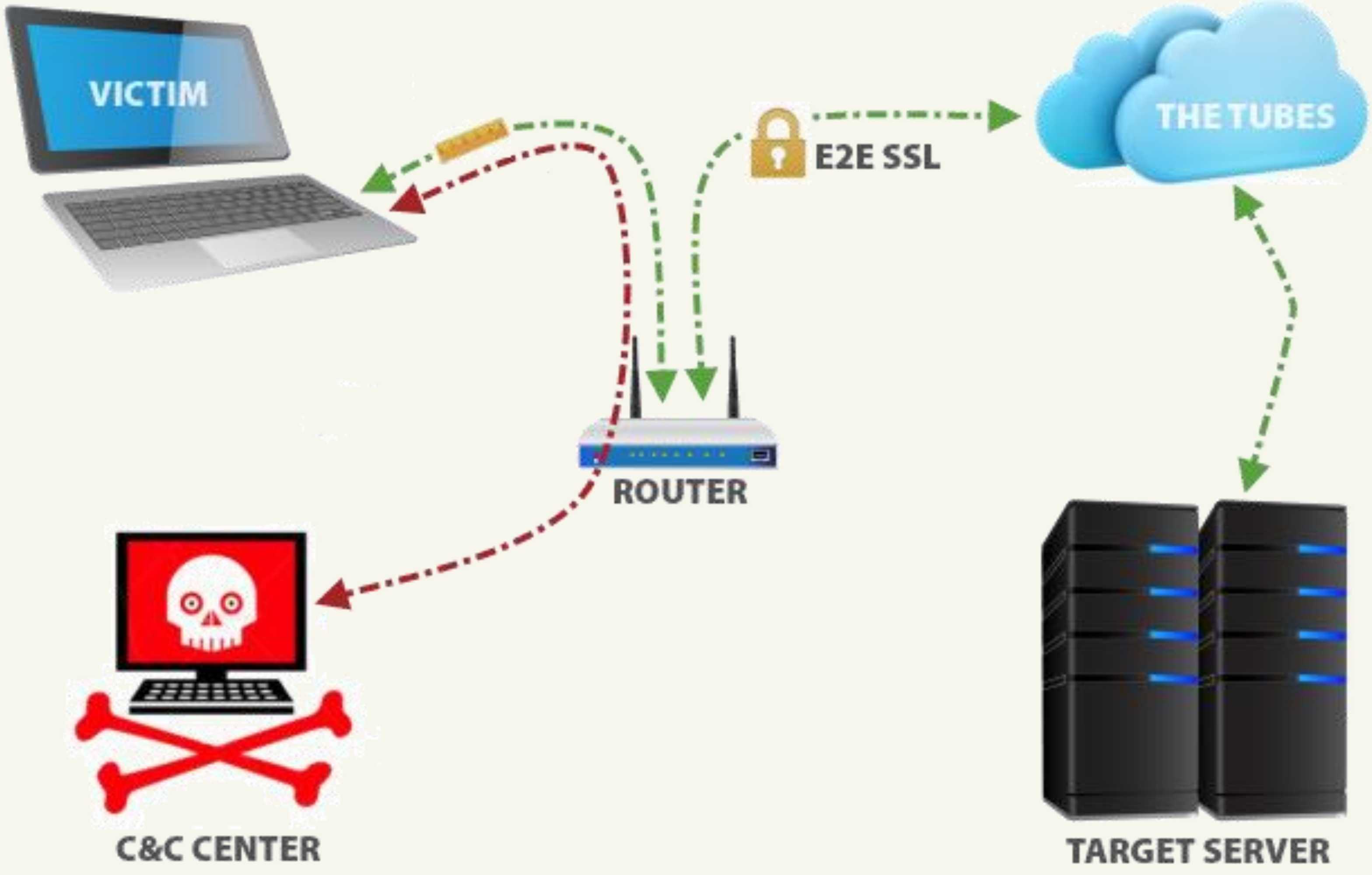
view-source:https://[redacted]/?ae=Item&t=IPM.Note&a=New&id=canary=?

```
 class="w100"></td>

                                                    <td
nowrap id="tdErrLgf"><a href="logoff.owa?
canary=4970ff90f52848419d107b33ce14d4d8">Log
Off</a       d>

        <div id=divDtls style="display:none"><br>
<b>Request</b><br>Url: <span
 id=requestUrl>https://[redacted]/forms/bas
ic/BasicEditMessage.aspx?
ae=Item&amp;t=IPM.Note&amp;a=New&amp;id=canary=?
</span><br>User host address: <span
```

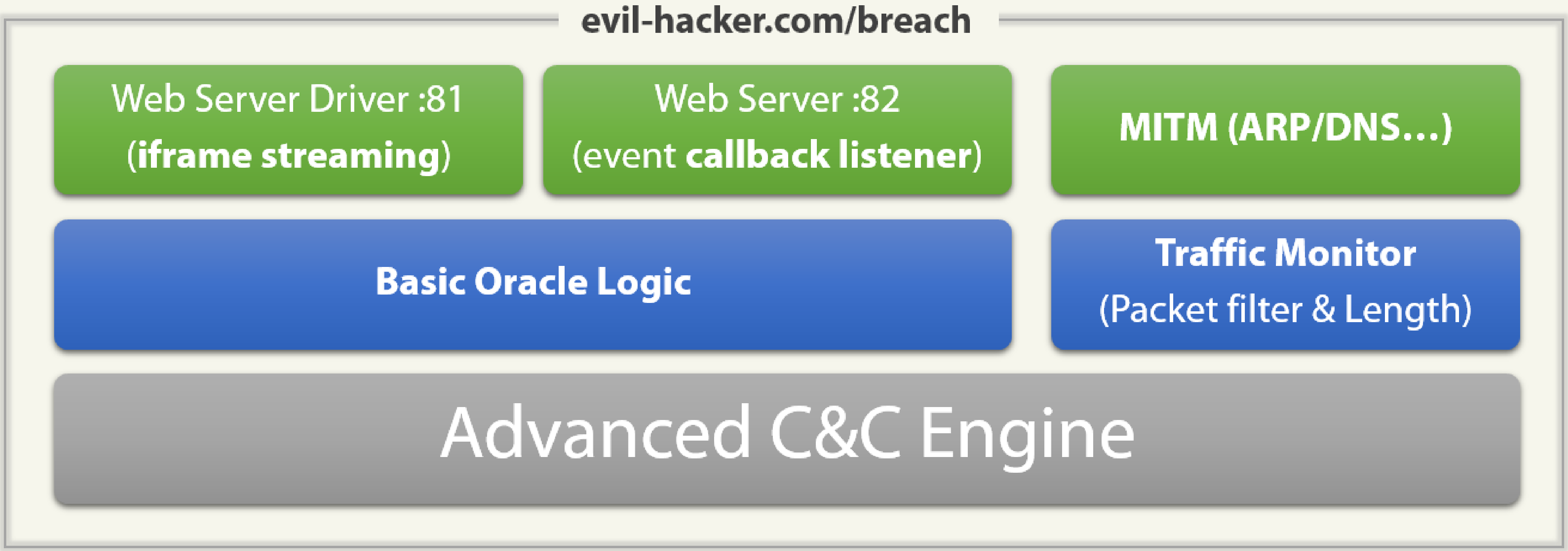
secret (CSRF token)

guess

BREACH / architecture



BREACH / command & control



evil-hacker.com/breach

Web Server Driver :81
(**iframe streaming**)

Web Server :82
(event **callback listener**)

MITM (ARP/DNS…)

Basic Oracle Logic

Traffic Monitor
(Packet filter & Length)

Advanced C&C Engine

C&C/ logic

- ✓ Traffic Monitor
  - Transparent relay **SSL proxy**

MITM: ARP spoofing, DNS, DHCP, WPAD…

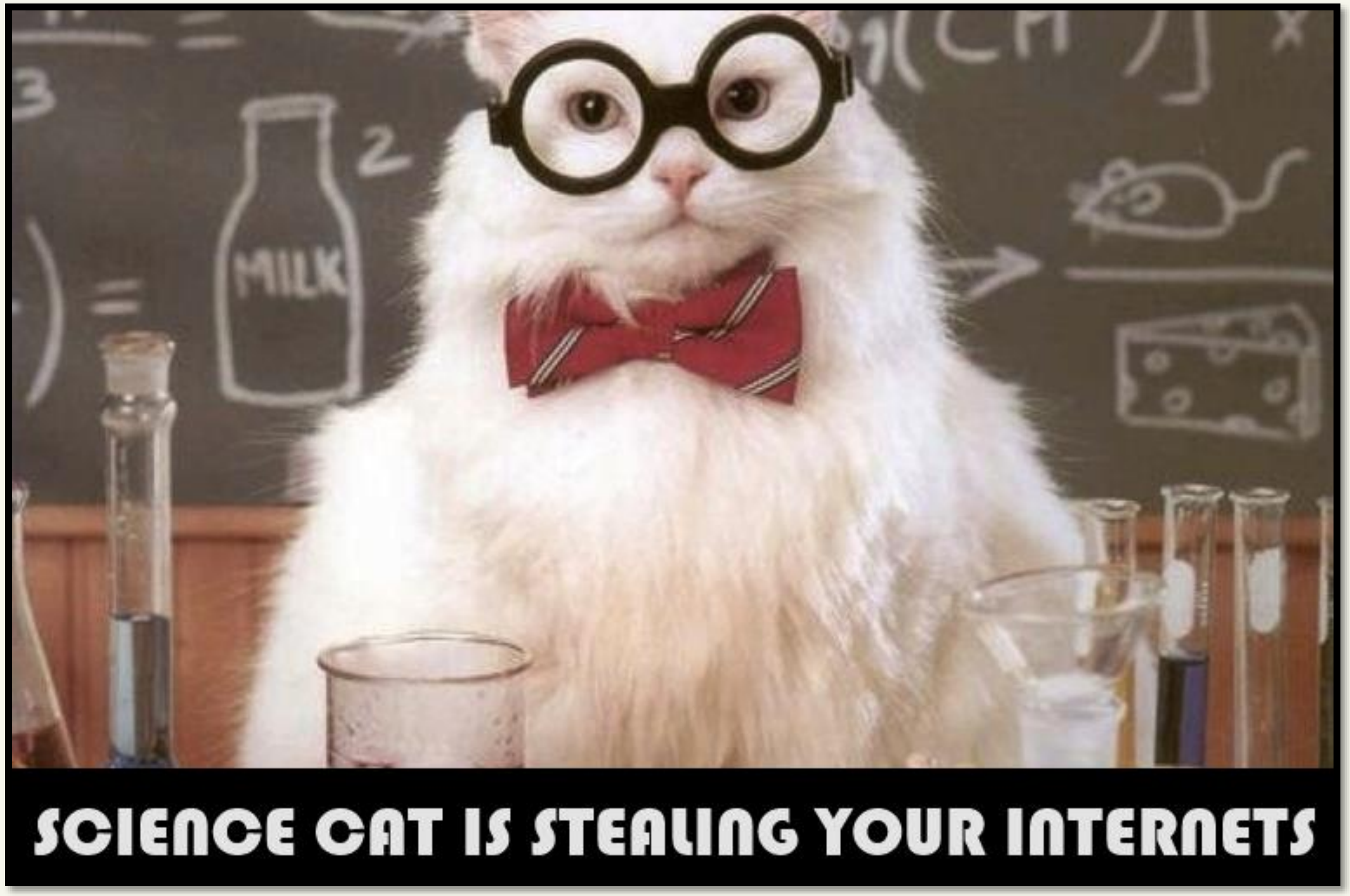
- ✓ HTML/JS Controller
  I. Dynamically generated for specific target server
  II. Injects & listens to **iframe streamer** from **c&c:81** that dictates the new HTTP requests to be performed (**img.src=…**)
  III. Issues the **outbound HTTP requests** to the target site via the victim's browser, session-riding a valid SSL channel
  IV. Upon synchronous completion of every request (***onerror***), performs a unique callback to **c&c:82** for the Traffic Monitor to **measure encrypted response size**

C&C/ logic

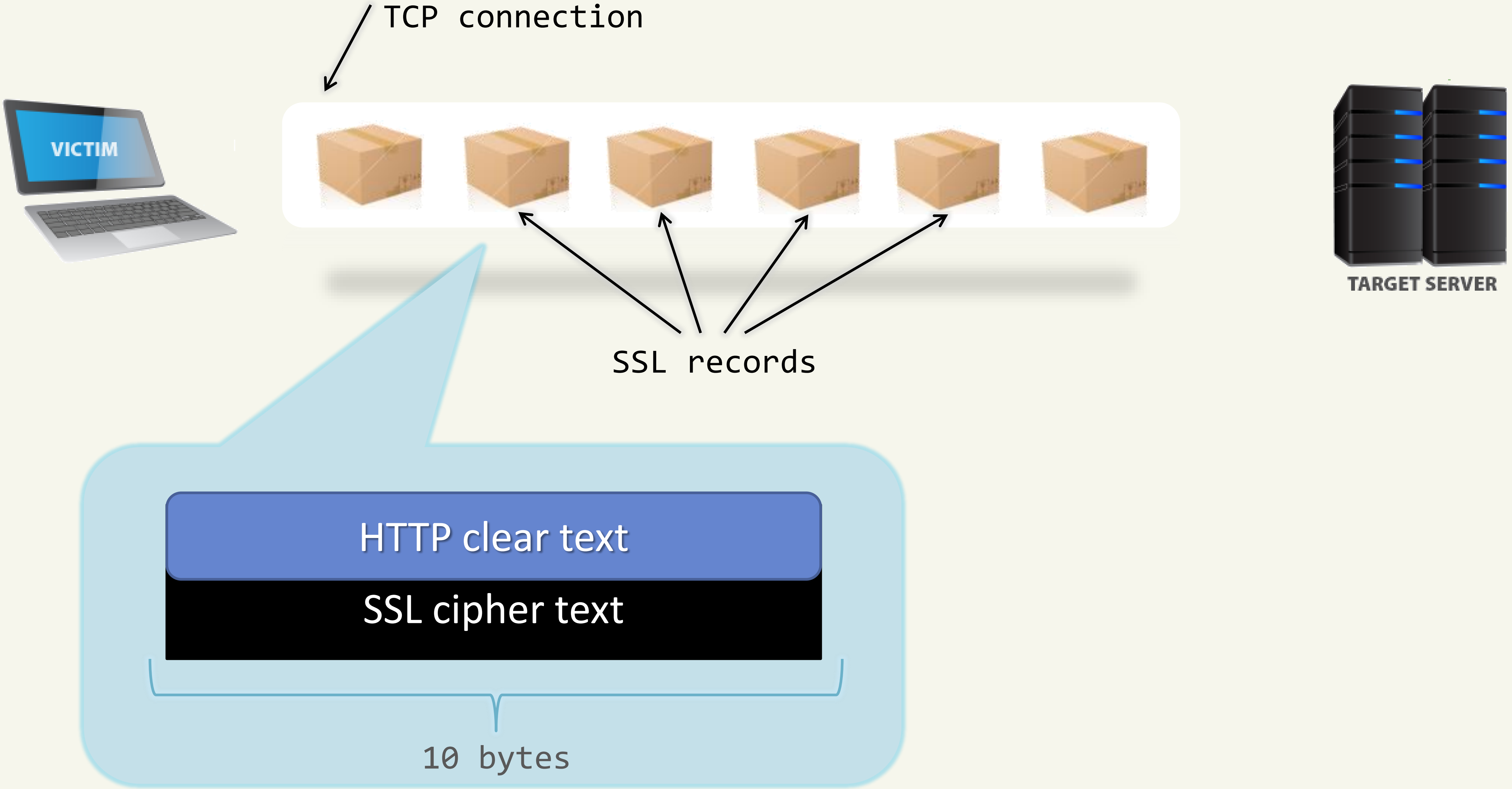
- ✓ Main C&C Driver
  - Coordinates **character guessing**
  - Adaptively **issues requests** to target site
  - Listens to **JS callbacks** upon **request completion**
  - **Measures** -inbound- packets **length**
  - Has built-in intelligence for **compression oracle runtime recovery**

THE ORACLE

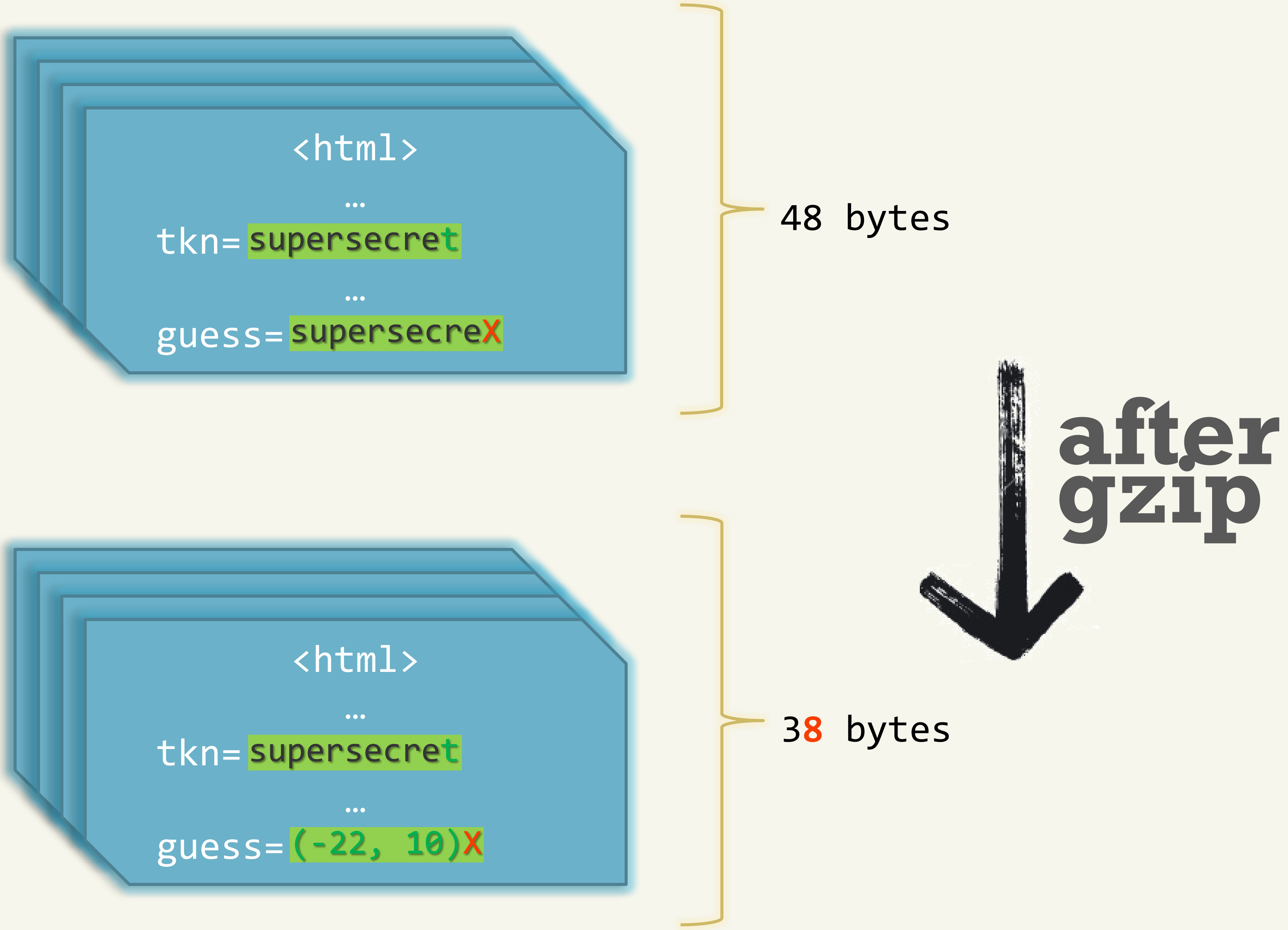
- **MEASURE**
  SIZE DELTA
- **GUESSING**
  BYTE-BY-BYTE
- **ERROR**
  RECOVERY



SSL REVEALS LENGTH



COMPRESSION ORACLE (I)



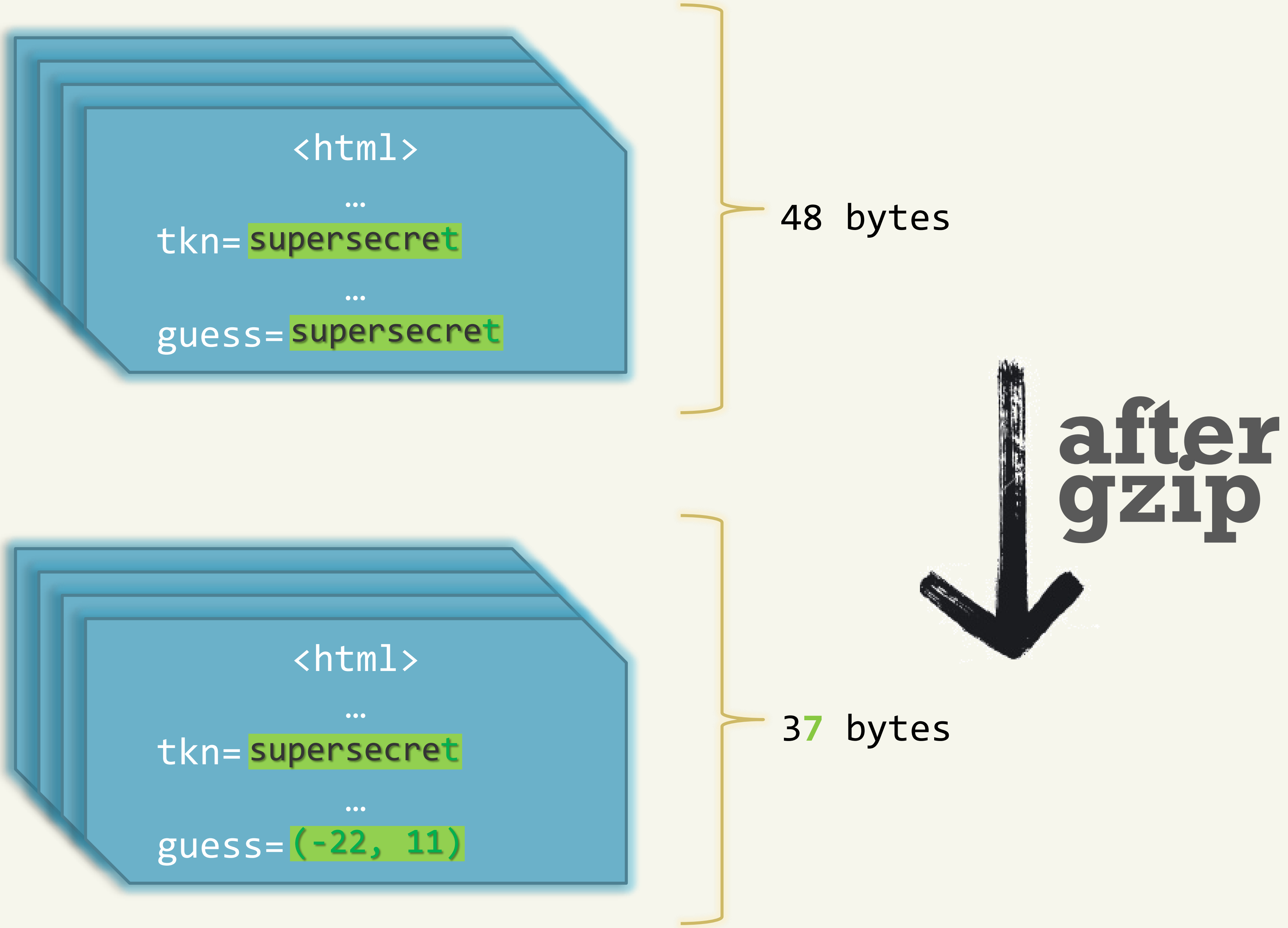
```
<html>
…
tkn=supersecret
…
guess=supersecreX
```

48 bytes

```
<html>
…
tkn=supersecret
…
guess=(-22, 10)X
```

38 bytes

COMPRESSION ORACLE (II)



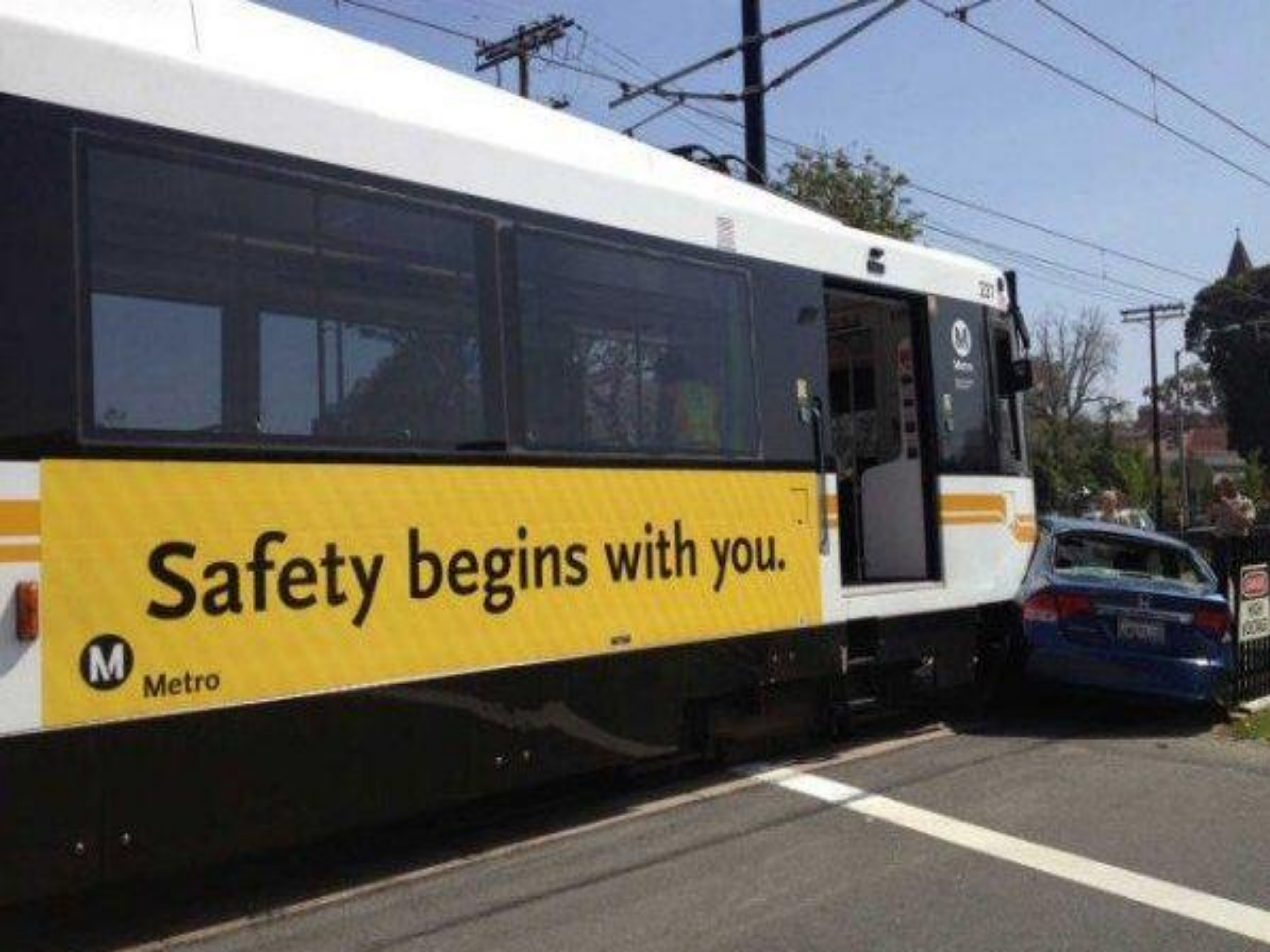
<html>
…
tkn=supersecret
…
guess=supersecret

48 bytes

after gzip

<html>
…
tkn=supersecret
…
guess=(-22, 11)

37 bytes



Safety begins with you.
Metro

THE ORACLE

Huffman Coding Nightmares

- ✓ Correct Guess

  https://target-server.com/page.php?blah=blah2...
  &secret=4bfb (response: 1358 bytes)

- ✓ Incorrect Guess

  https://target-server.com/page.php?blah=blah2...
  &secret=4bfa (response: 1358 bytes)

THE ORACLE

Fighting Huffman Coding

✓ Two Tries + random [*dynamic*] padding

https://target-server.com/page.php?blah=blah2...
&secret=4bf 7{}{}(...){}{}{}{}{}
&secret=4bf{}{}(...){}{}{}{}{} 7

✓ Character set pool + random padding

https://target-server.com/page.php?blah=blah2...
&secret=4bf 7{}{}(...){}{}{}{}{}---a-b-c-d-...-5-6-8-9-...
&secret=4bf 8{}{}(...){}{}{}{}{}---a-b-c-d-...-5-6-7-9-...

THE ORACLE

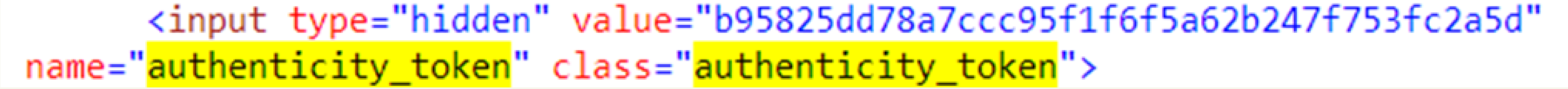
Two Tries Reality

- ✓ Less than ideal conditions:
  - In theory, **two-tries** allows for short-circuiting once winner is found
  - In practice, still need to **evaluate all candidates**
  - **Huffman encoding** causes collisions

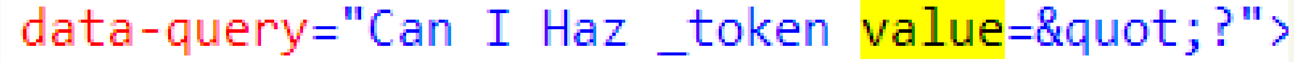
ROADBLOCKS

- ✓ Conflict & Recovery mechanisms
  (no winners / too many winners)
  - **Look-ahead** (2+ characters) – reliable, but expensive
    - Best value / averages
  - **Rollback** (last-known conflict)
  - Check **compression ratio** of guess string

- ✓ Page URL / HTML entity encoding
  - Can interfere with **bootstrapping**



```
        <input type="hidden" value="b95825dd78a7ccc95f1f6f5a62b247f753fc2a5d"
 name="authenticity_token" class="authenticity_token">
```

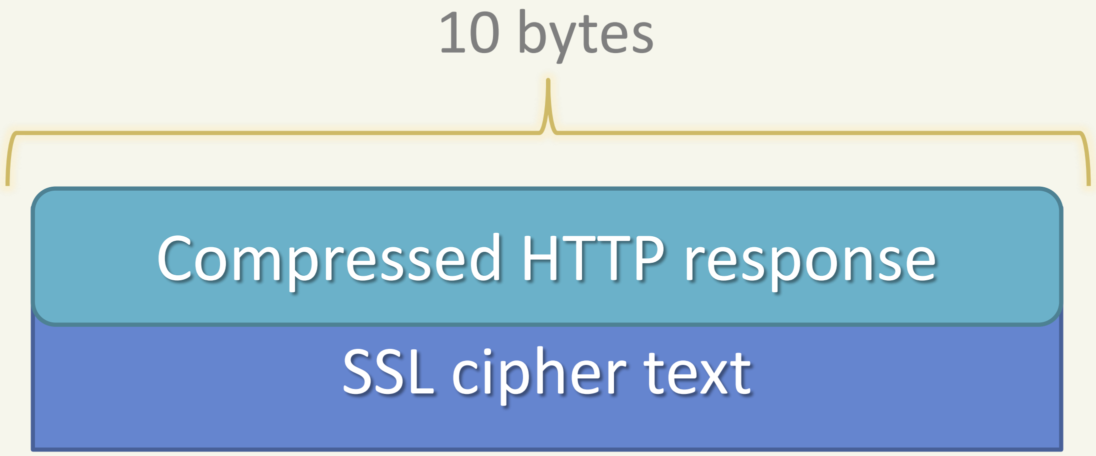


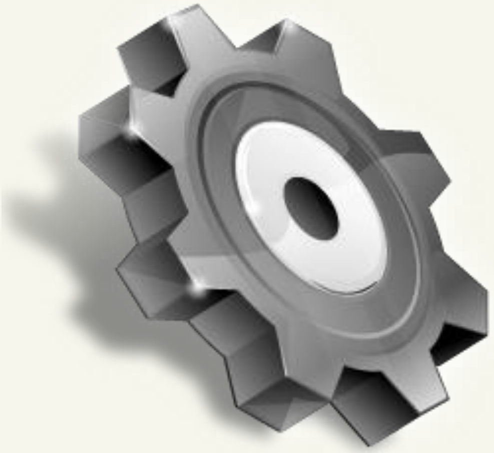
```
data-query="Can I Haz _token value=&quot;?">
```

MORE ROADBLOCKS

- ✓ Stream cipher vs. block cipher

- Stream cipher **reveals** exact plain text length

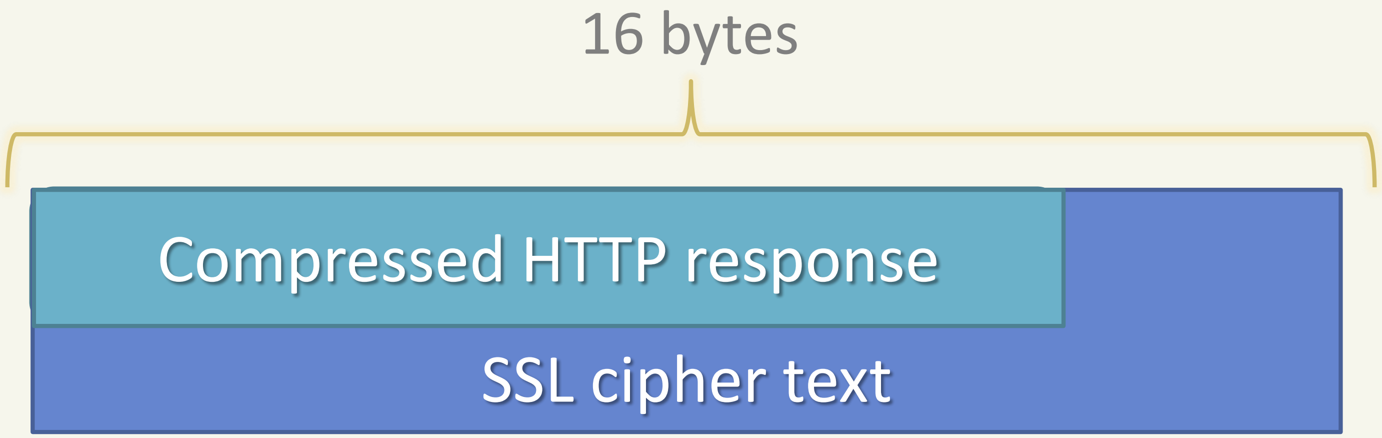




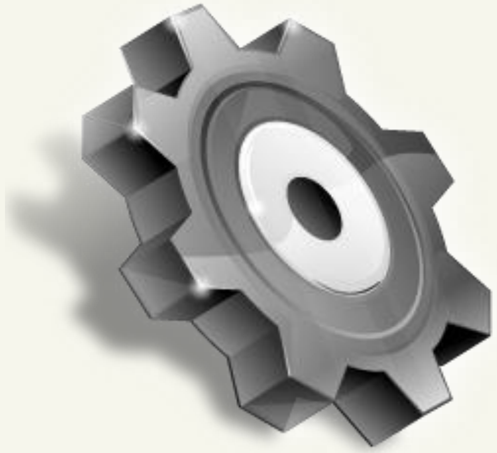
MORE ROADBLOCKS

✓ Stream cipher vs. block cipher

- Block cipher **hides** exact plain text length



- Align response to a tipping point
- Guess Window (keeping response aligned)



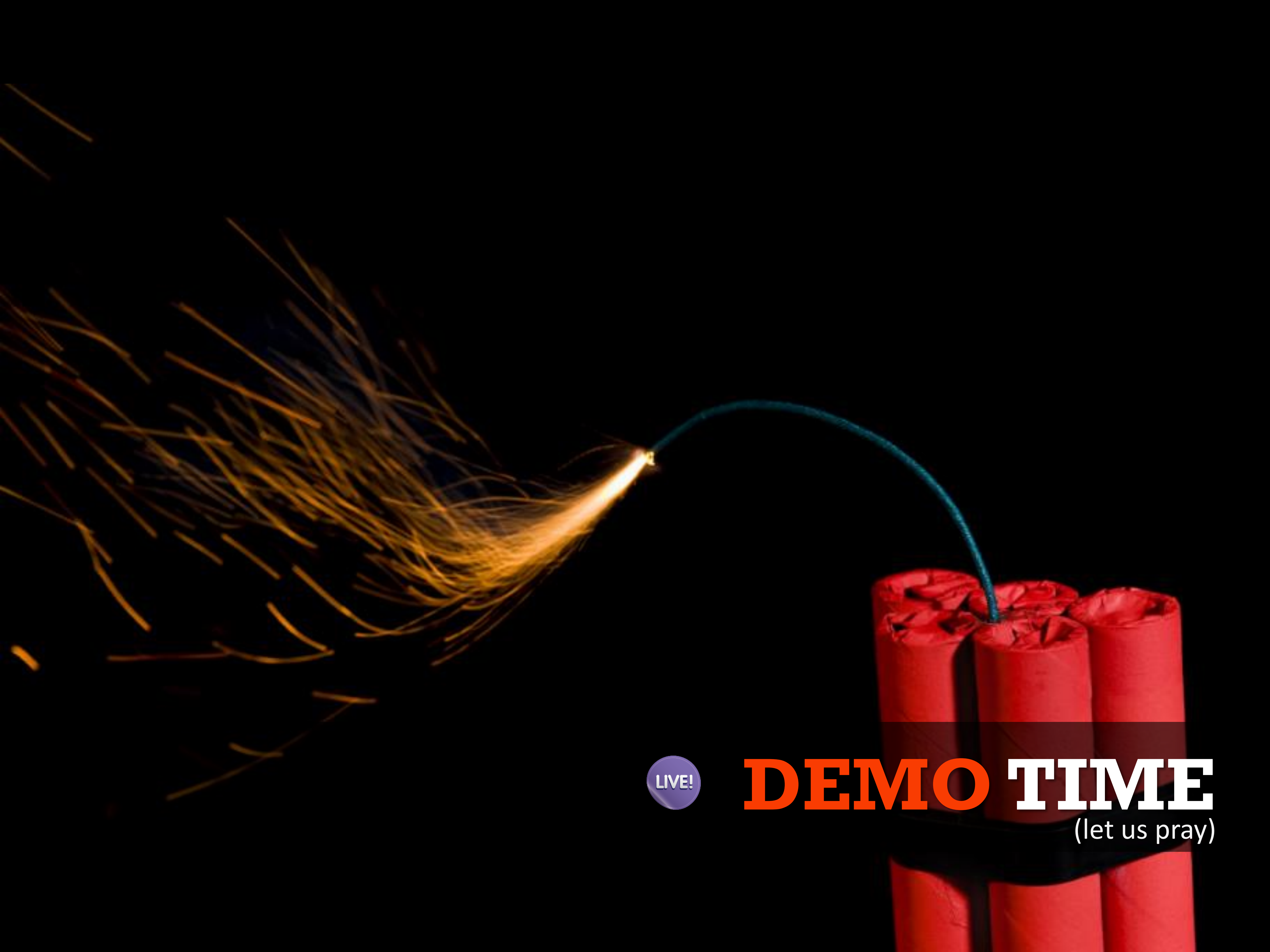
EVEN MORE ROADBLOCKS

- ✓ Keep-Alive (a premature death)
  - **Image** requests vs. **scripts** vs. **CORS** requests
- ✓ Browser synchronicity limits (1x)
  - Hard to correlate **HTTP requests** to **TCP segments**
- ✓ Filtering out noise
  - Active application?
  - Background polling?

YET MORE ROADBLOCKS

- ✓ **‘Unstable’ pages** (w/ *random* DOM blocks)
  - Averaging & outlier removal
- ✓ The war against **Huffman coding**
  - Weight (symbol) normalization
- ✓ Circumventing cache
  - Random timestamp
- ✓ Other Oracles
  - *Patent-pending!*

“ OVERWHELMED?



DEMO TIME

(let us pray)

THE TOOL

MITIGATIONS

RANDOMIZING THE LENGTH

- variable padding
- fighting against math
- /FAIL

DYNAMIC SECRETS

- dynamic CSRF tokens per request

MASKING THE SECRET

- random **XOR** – easy, dirty, practical path
- downstream enough

SEPARATING SECRETS

- deliver secrets in input-less servlets
- chunked secret separation (lib patch)

CSRF-PROTECT EVERYTHING

- unrealistic

THROTTLING & MONITORING

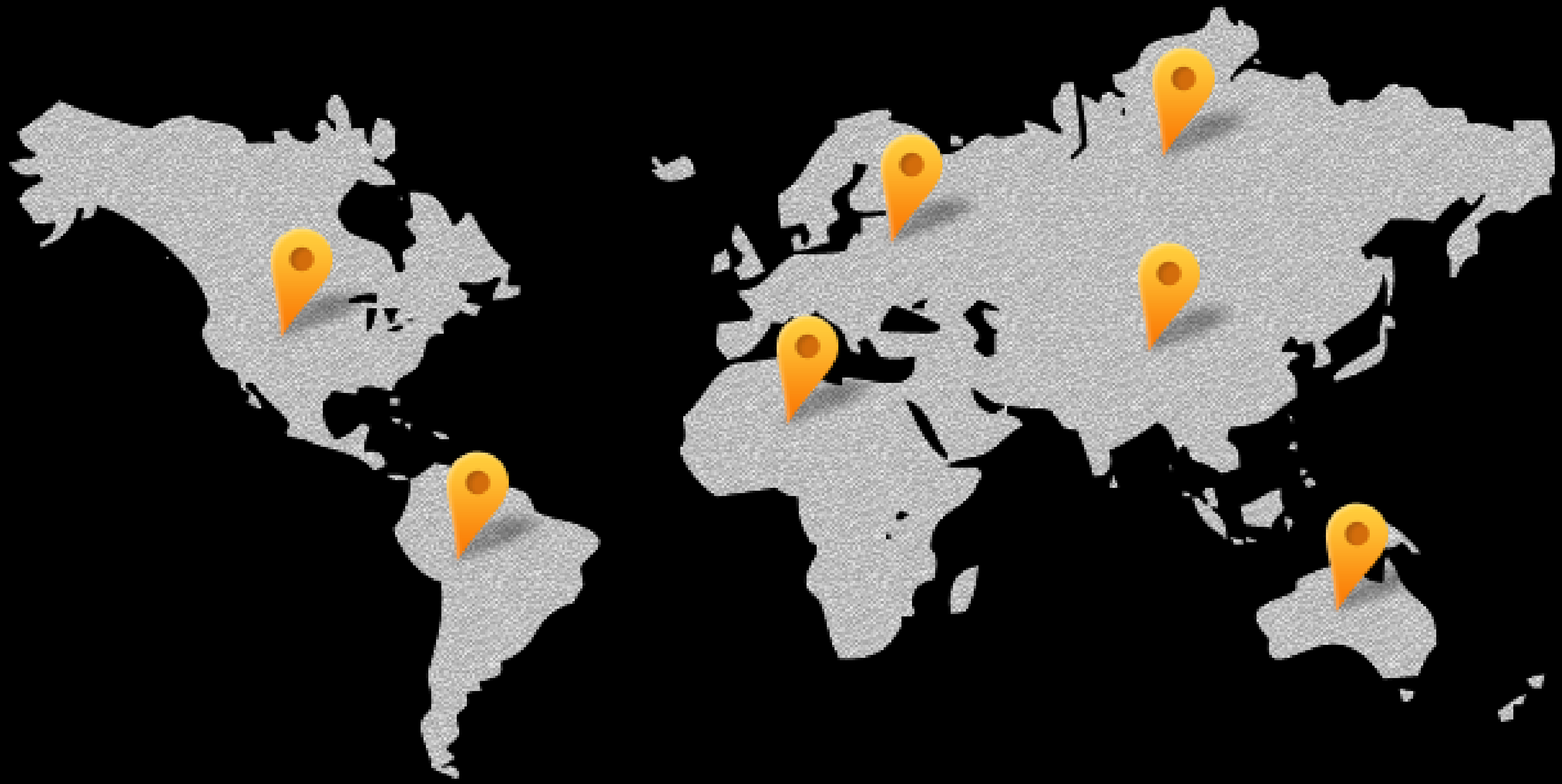
DISABLING GZIP FOR DYNAMIC PAGES

FUTURE WORK

- ✓ Better understanding of DEFLATE / GZIP
- ✓ Beyond HTTPS
  - Very generic side-channel
  - Other protocols, contexts?
- ✓ Stay tuned for the next BREACH

WANT MORE?

AGENTS STANDING BY



BreachAttack.com

PAPER | PRESENTATION | POC TOOL

THANK YOU EVERYBODY !

BREACHATTACK.COM

Angelo Prado

angelpm@gmail.com

🐦 @PradoAngelo

Neal Harris

neal.harris@gmail.com

🐦 @IAmTheNeal

Yoel Gluck

yoel.gluck2@gmail.com

✓ If you liked the talk*, don't forget to scan your badge for the evaluation survey

** ignore otherwise*

black hat
USA 2013

JULY 27 - AUGUST 1, 2013
CAESARS PALACE | LAS VEGAS, NV
WWW.BLACKHAT.COM