Angelo Prado
Neal Harris
Yoel Gluck

SSL, GONE IN 30 SECONDS

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

PREVIOUSLY…

CRIME

Presented at ekoparty 2012

Juliano Rizzo
Thai Duong

Target

Secrets in HTTP headers

Requirements

TLS compression
MITM
A browser

COMPRESSION OVERVIEW

- ✓ DELATE:
  - LZ77: reducing bits by reducing redundancy
    - Googling the googles -> Googling the g(-13,4)s
  - Huffman coding: reducing bits by employing an *entropy encoding algorithm*
    - *aka. replace common bytes with shorter codes*

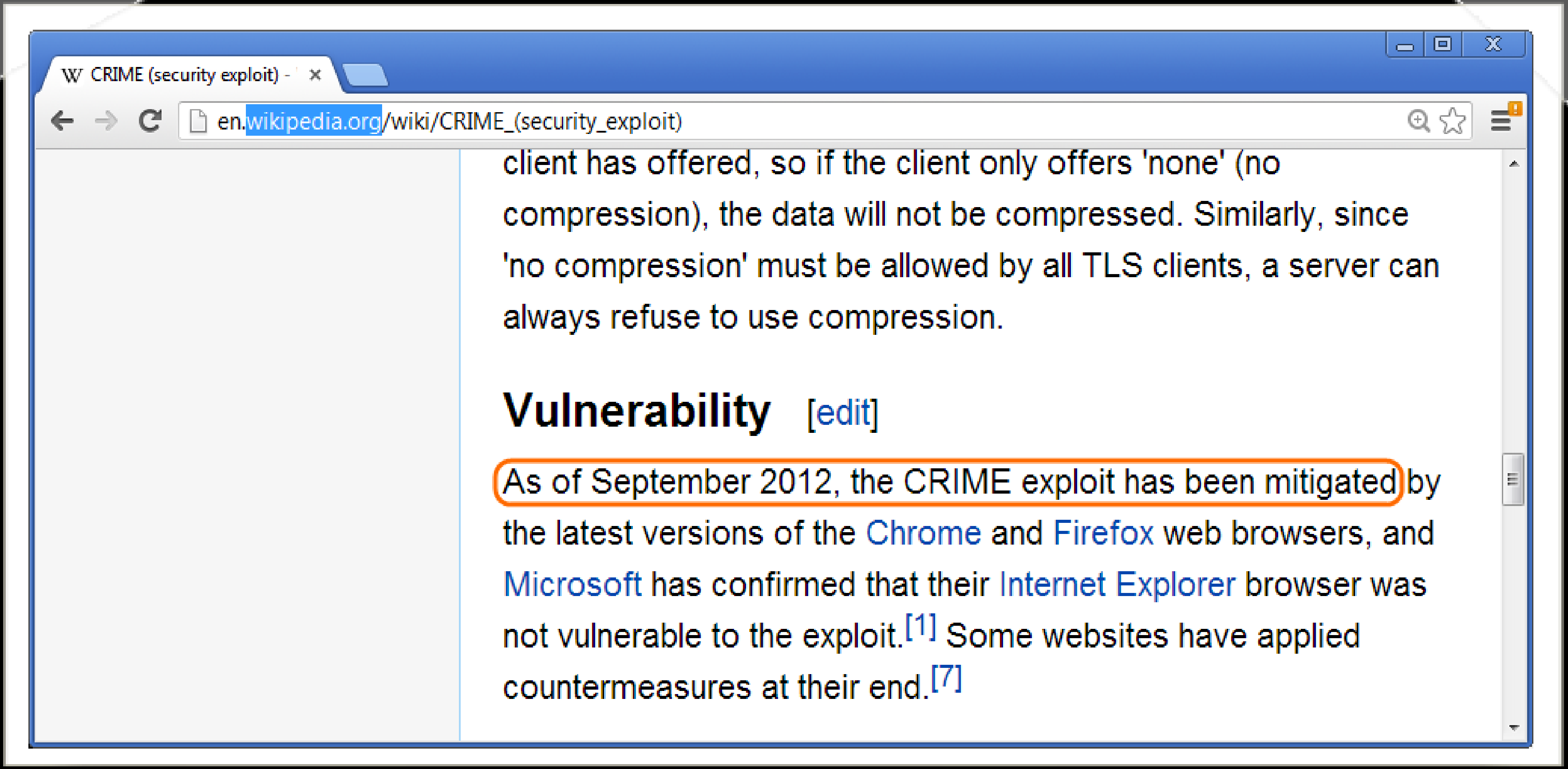
SO ABOUT **CRIME…**

The Compression Oracle:

- ✓ SSL doesn't hide **length**
- ✓ TLS/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

IT'S FIXED!

- TLS Compression Disabled



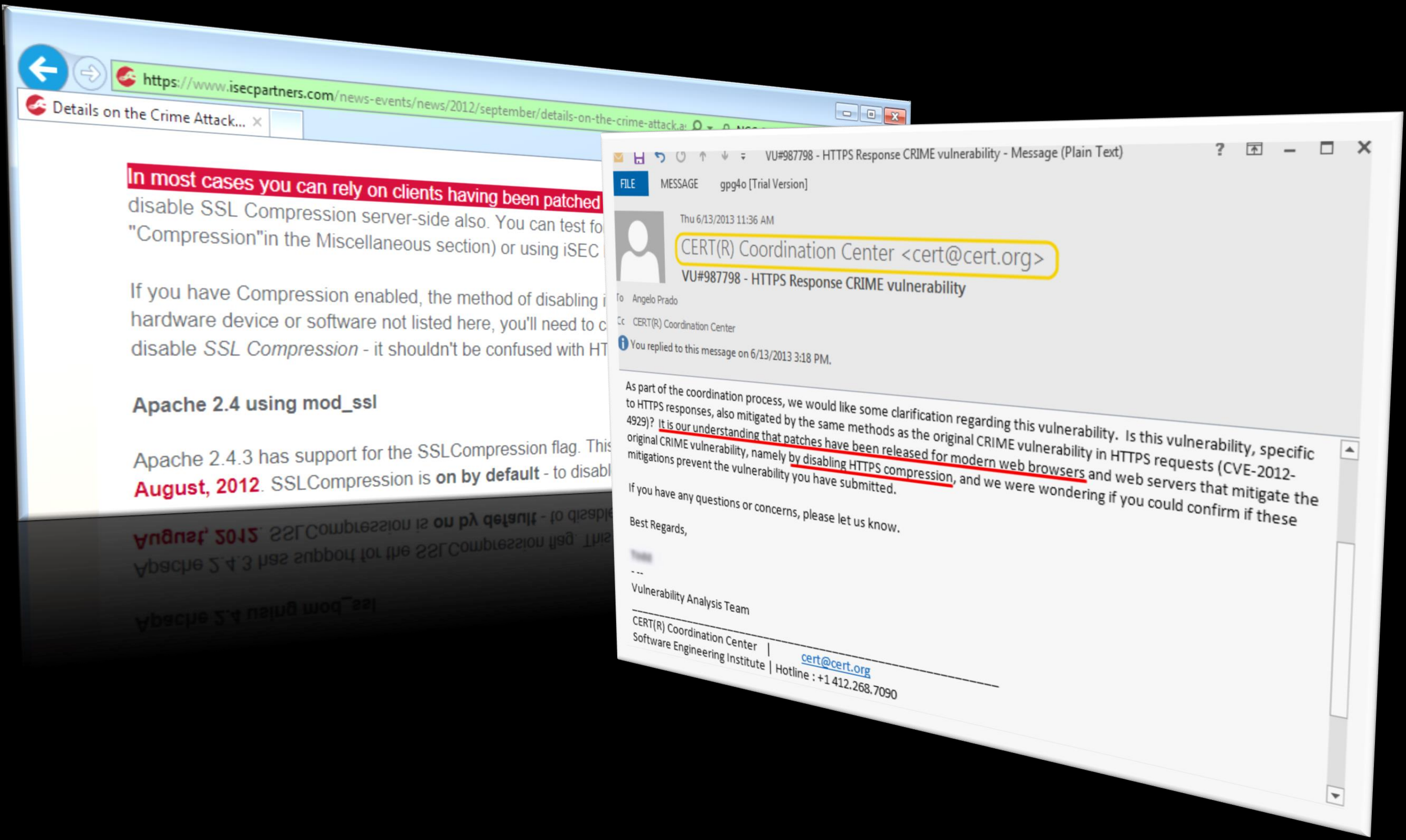
W CRIME (security exploit) - ×
en.wikipedia.org/wiki/CRIME_(security_exploit)

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]

IT'S FIXED!



https://www.isecpartners.com/news-events/news/2012/september/details-on-the-crime-attack.a
Details on the Crime Attack...

In most cases you can rely on clients having been patched
disable SSL Compression server-side also. You can test fo
"Compression"in the Miscellaneous section) or using iSEC

If you have Compression enabled, the method of disabling i
hardware device or software not listed here, you'll need to c
disable *SSL Compression* - it shouldn't be confused with HT

Apache 2.4 using mod_ssl

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

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

DO NOT PANIC »

« IT'S FIXED

[let's bring it back to life]

INTRODUCING BREACH

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

BREACH / the ingredients

| GZIP

- Very **prevalent**
- Highly **impractical** to turn off
- **Any** browser, **any** web server

| Fairly stable pages

- It only takes **one**
- **Less than 30 seconds** for simple pages
- Minutes to hours for more complicated dynamic bodies

| MITM / traffic visibility

- No tampering / SSL downgrade

| SSL / TLS [*any* version]

- Could be turned off ;)

| A secret in the response body

- CSRF, SIDs, PII, ViewState…
- and **much more**

| Attacker-supplied data

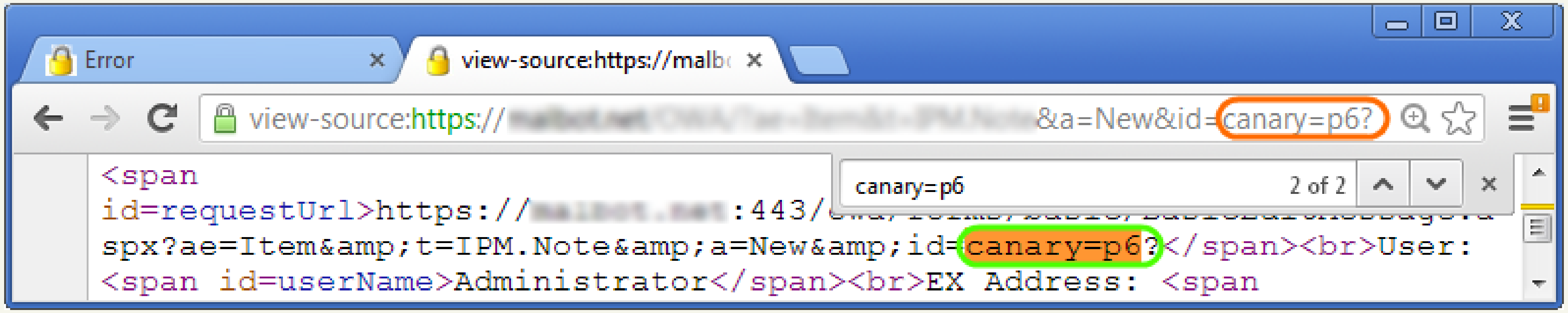
- Guess (in response body)

| Three-characters prefix

- To **bootstrap compression**

[PREFIX / sample bootstrap]

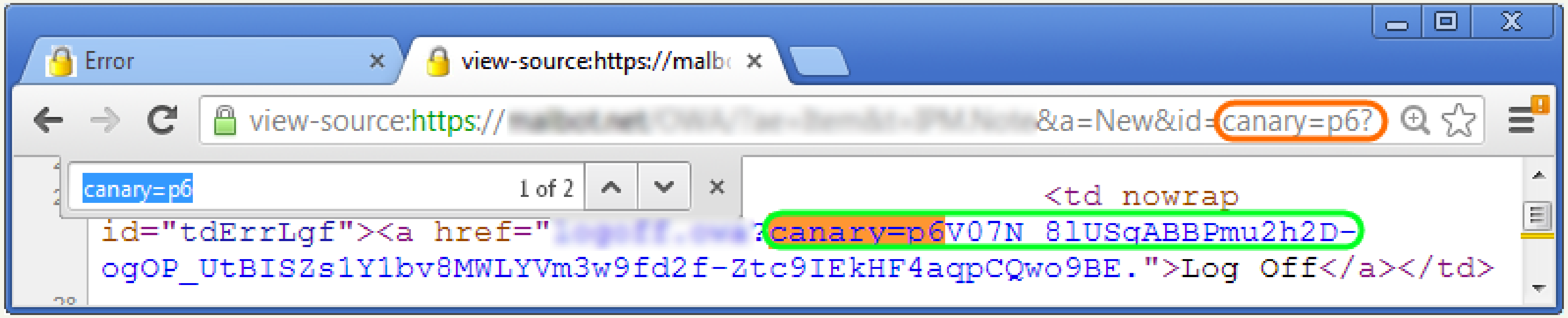
Guess (in response body)



```
view-source:https://[redacted]&a=New&id=canary=p6?
```

```
<span
id=requestUrl>https://[redacted]:443/
spx?ae=Item&amp;t=IPM.Note&amp;a=New&amp;id=canary=p6?</span><br>User:
<span id=userName>Administrator</span><br>EX Address: <span
```

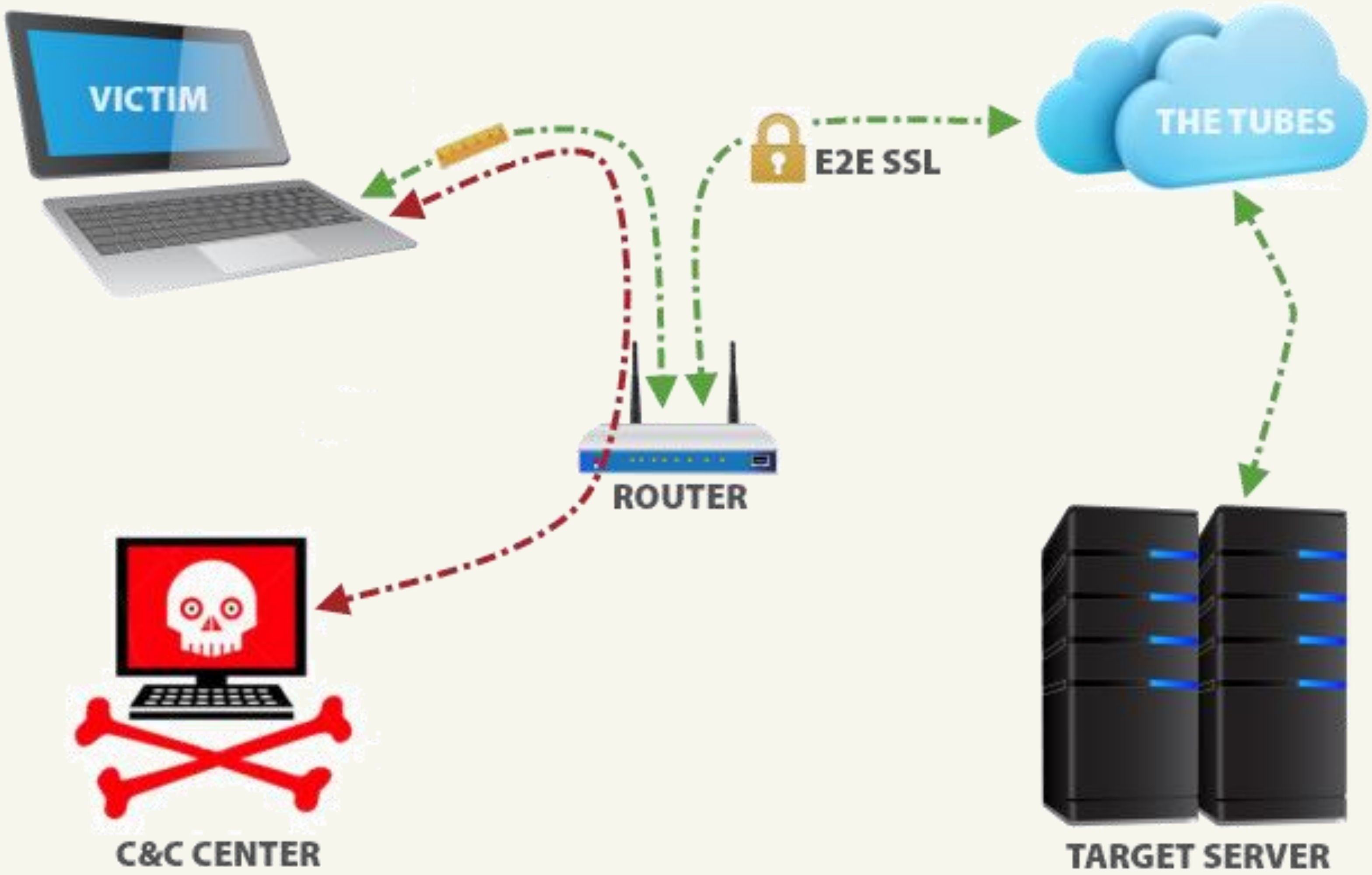
Target secret (CSRF token)



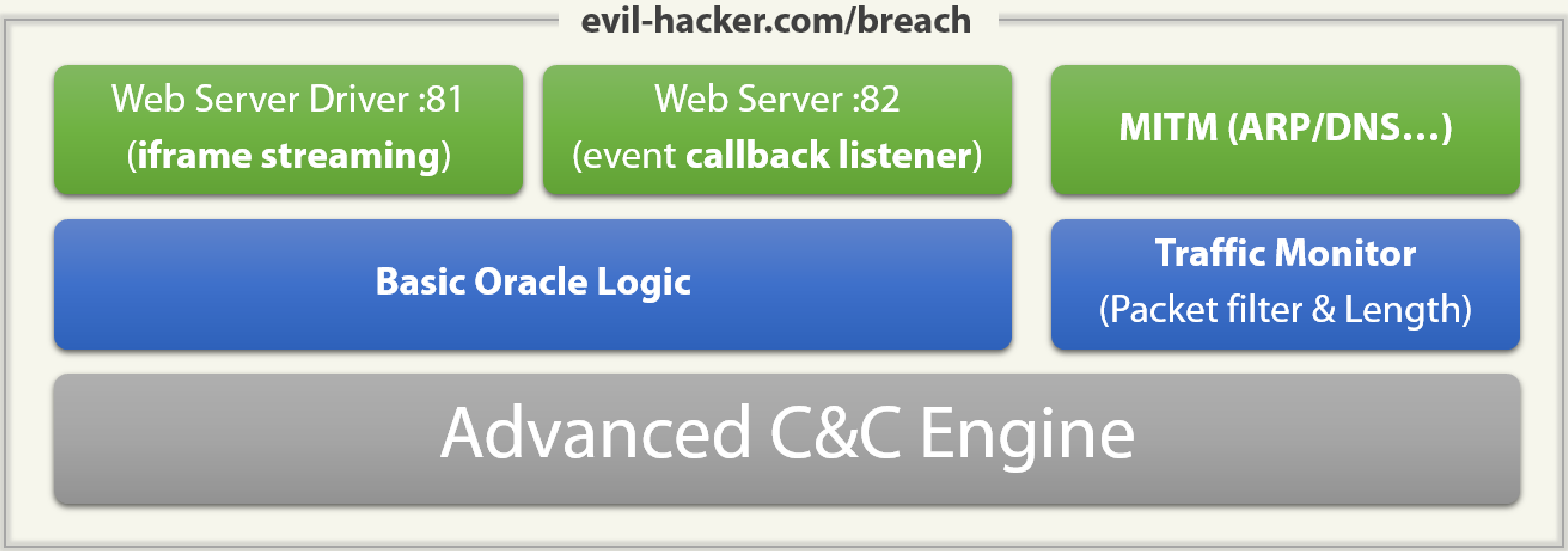
```
view-source:https://[redacted]&a=New&id=canary=p6?
```

```
                                              <td nowrap
id="tdErrLgf"><a href="[redacted]?canary=p6V07N_81USqABBPmu2h2D-
ogOP_UtBISZs1Y1bv8MWLYVm3w9fd2f-Ztc9IEkHF4aqpCQwo9BE.">Log Off</a></td>
```

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

ORACLE

ONE CHARACTER AT A TIME

· Guessing byte-by-byte

AIRBAGS

· Random amount of padding

COLLISIONS

· Attempt recovery for multiple winners

· Detect & roll-back from wrong path

TWO TRIES

· Issue two HTTPs requests per guess

```
https://target-server.com/page.php?blah=blah2...
          &secret=4bf7{}{}(...){}{}{}{}{}
          &secret=4bf{}{}(...){}{}{}{}{} 7
```

ORACLE / logic (II)

- ✓ Guess Swap
  - Swap last two characters in the guess
  - Measure overall size increase

```
https://target-server.com/page.php?blah=blah2...
        &secret=4bf7
        &secret=4b7f
```

- ✓ Character set pool (to eliminate Huffman tree changes between guesses)
  - Add all characters to all guesses, shifting the guessed character into position

```
https://target-server.com/page.php?blah=blah2...
  &secret=4bf7 {}{}(...){}{}{}{}{}---a-b-c-d-...-5-6-8-9-...
  &secret=4bf8 {}{}(...){}{}{}{}{}---a-b-c-d-...-5-6-7-9-...
```

C&C/ logic

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

MITM: ARP spoofing, DNS, DHCP, WPAD…

- ✓ HTML/JS Controller
  1. Dynamically generated for specific target server
  2. Injects & listens to **iframe streamer** from **c&c:81** that dictates the new HTTP requests to be performed (**img.src=…**)
  3. Issues the **outbound HTTP requests** to the target site via the victim's browser, session-riding a valid SSL channel
  4. 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 website
  - Listens to **JS callbacks** upon **request completion**
  - Oracle **measures** -inbound- packets **length**
  - Has built-in intelligence for **conflict resolution** and **recovery**

ROADBLOCKS

- ✓ 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

- ✓ Conflict resolution & recovery mechanisms (I)

  (In case of conflict / no winners)

  1. Dynamic **airbags**
  2. **Look-ahead** (2+ characters) – more reliable, but more expensive
     - Best value
     - Averages

ROADBLOCKS

- ✓ Conflict resolution & recovery mechanisms (II)
  - Rollback (in-memory path, **last-known conflict**)
  - Detect substrings in secret/guess
    - Check **compression ratio** of guess string

- ✓ Page URL / HTML entity encoding
  - Can interfere with collision **bootstrapping** and **secret key-space**

MORE ROADBLOCKS

✓ Circumventing cache

- For **targets & callback** – random timestamp

✓ Block mode vs. stream cipher mode

- Align response to a **tipping point** and overflow into the next block
- Guess Window (**keeping response aligned**) – as we add characters to the guess, we remove others

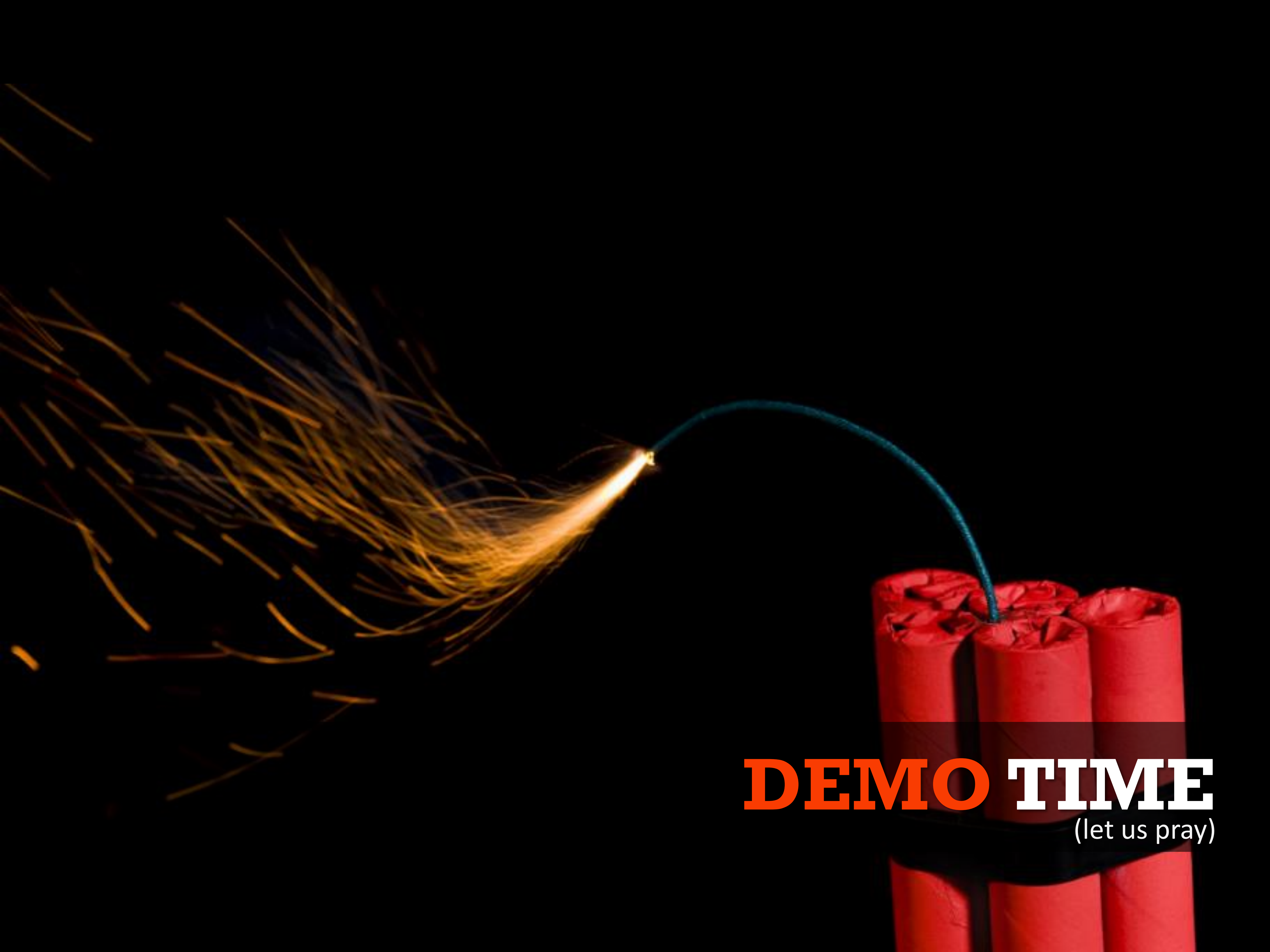
EVEN MORE ROADBLOCKS

- 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 – statistical outlier removal and detection
- ✓ Collateral effects of Huffman tree
  - Weight (symbol) normalization
- ✓ Other Misc. 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

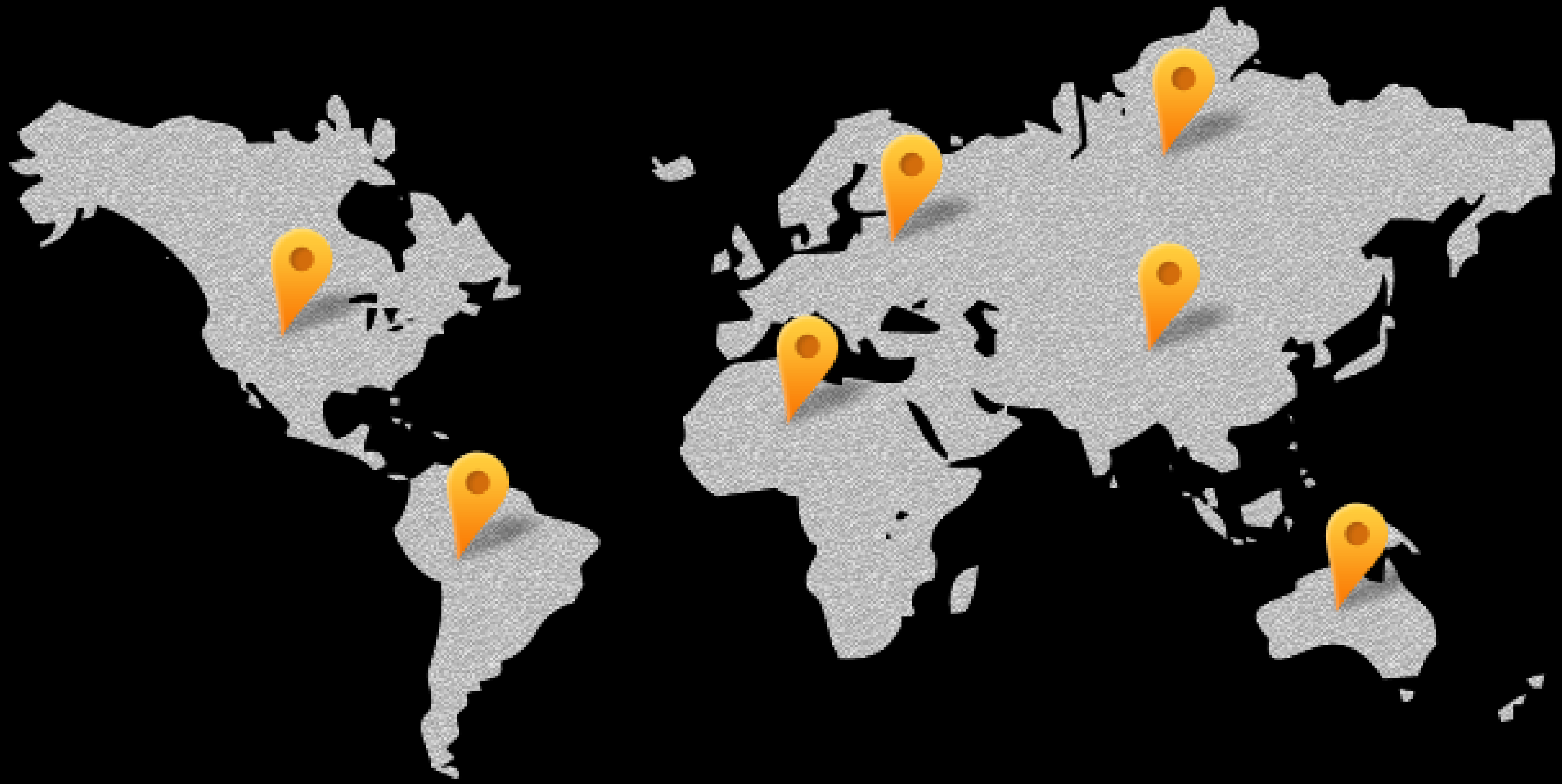
THROTTLING & MONITORING

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?



BreachAttack.com

PAPER | PRESENTATION | POC TOOL

THANK YOU EVERYBODY !

Angelo Prado

angelpm@gmail.com

@PradoAngelo

Neal Harris

neal.harris@gmail.com

@IAmTheNeal

Yoel Gluck

yoel.gluck2@gmail.com

✓ Don’t forget to fill out* the questionnaire if you liked it

** ignore otherwise*

BreachAttack.com