



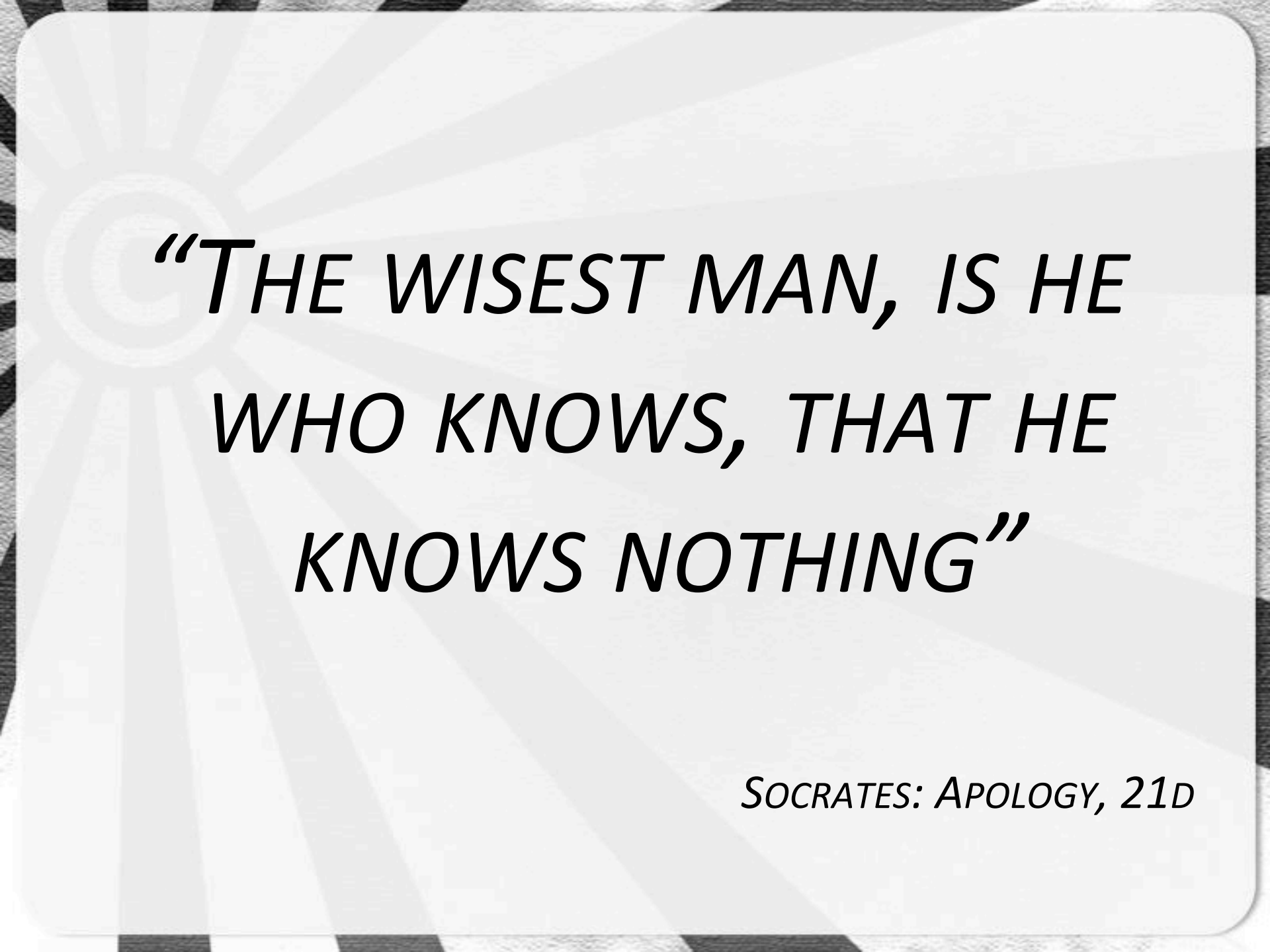
Defense by Numb3r5

*Making problems for script k1dd13s
and scanner monkeys*

@ChrisJohnRiley

dirtysec
podcaster
pentester
blogger
bad-programmer
researcher

twitter
Metasploit
Not-an-Expert
c22
eurotrash
scared
con-junkie
PTES



*“THE WISEST MAN, IS HE
WHO KNOWS, THAT HE
KNOWS NOTHING”*

SOCRATES: APOLOGY, 21D



edge case

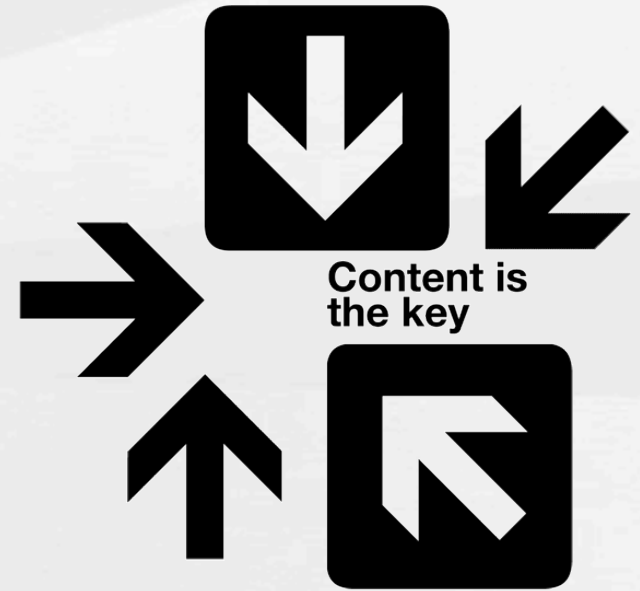


TL;DR

Goals for this talk

Describe the defensive uses of HTTP status codes

- 1) What**
- 2) Why**
- 3) How**
- 4) Goals**
- 5) Bringing it together**
- 6) Review**





#1

[WHAT?]

A large, faint copyright symbol (©) is positioned on the left side of the slide. From the right side of the symbol, a series of light gray rays radiate outwards across the background, creating a sunburst effect. The background is a light gray color with a subtle, repeating pattern of the copyright symbol.

HTTP STATUS CODES

HTTP/1.1 {206 Partial content}

Date: Wed, 15 Nov 1995 06:25:24 GMT

Last-Modified: Wed, 15 Nov 1995 04:58:

Content-Range: bytes 21010-47021/47022

Content-Length: 26012

Content-Type: image/gif



Seems like such a
Small detail



... *small* detail,

big impact





This talk contains:

- Numbers
- Bad Jokes
- Traces of peanuts
- Did I mention numbers?

HTTP Status Codes

- Majority part of RFC 2616 (HTTP/1.1)
- 5 *main* classes of response
 - 1XX *informational*
 - 2XX *success*
 - 3XX *redirection*
 - 4XX *client error*
 - 5XX *server error*

HTTP Status Codes

- Proposed RFC* for 7XX codes
- Examples:
 - 701 *Meh*
 - 719 *I am not a teapot*
 - 721 *Known unknowns*
 - 722 *Unknown unknowns*
 - 732 *Fucking Unicode*

* <https://github.com/joho/7XX-rfc>



#1.1

BASICS

AKA: THE BORING **THEORY** BIT

1XX Informational

- Indicates response received
- Processing is not yet completed
 - 100 Continue
 - 101 Switching Protocols
 - 102 Processing ([WebDAV RFC 2518](#))

2XX Success

- Indicates response received
- Processed and understood
 - 200 OK
 - 201 Created
 - 202 Accepted
 - 203 Non-Authoritative Information
 - 204 No Content

2XX Success (cont.)

- 205 Reset Content
- 206 Partial Content
- 207 Multi-Status ([WebDAV RFC 4918](#))

Codes not supported by Apache

- 208 Already Reported
- 226 IM Used
- 250 Low on Storage Space

3XX Redirection

- Action required to complete request
 - 300 Multiple Choices
 - 301 Moved Permanently
 - 302 Found / Moved Temporarily
 - 303 See Other
 - 304 Not Modified

3XX Redirection (cont.)

- 305 Use Proxy
- 306 Switch Proxy
- 307 Temporary Redirect

Codes not supported by Apache

- 308 Permanent Redirect

4XX Client Error

- Client caused an error
 - 400 Bad Request
 - 401 Unauthorized
 - 402 Payment Required
 - 403 Forbidden
 - 404 Not Found
 - 405 Method Not Allowed

4XX Client Error (cont.)

- 406 Not Accessible
- 407 Proxy Authentication Required
- 408 Request Timeout
- 409 Conflict
- 410 Gone
- 411 Length Required

4XX Client Error (cont.)

- 412 Precondition Failed
- 413 Request Entity Too Large
- 414 Request-URI Too Long
- 415 Unsupported Media Type
- 416 Request Range Not Satisfiable
- 417 Expectation Failed
- 418 I'm a Teapot ([WebDAV RFC 2324](#))

4XX Client Error (cont.)

- 419 / 420 / 421 Unused
- 422 Unprocessable Entity (RFC 4918)
- 423 Locked (RFC 4918)
- 424 Failed Dependency (RFC 4918)
- 425 No Code / Unordered Collection
- 426 Upgrade Required (RFC 2817)

4XX Client Error (cont.)

Codes not supported by Apache

- 428 Precondition Required
- 429 Too Many Requests
- 431 Request Header Fields Too Large
- 444 No Response (NGINX)
- 449 Retry With (Microsoft)
- 450 Blocked by Win. Parental Controls
- 451 Unavailable For Legal Reasons
- 494 Request Header Too Large (NGINX)
- 495 Cert Error (NGINX)
- 496 No Cert (NGINX)
- 497 HTTP to HTTPS (NGINX)
- 499 Client Closed Request (NGINX)

5XX Server Error

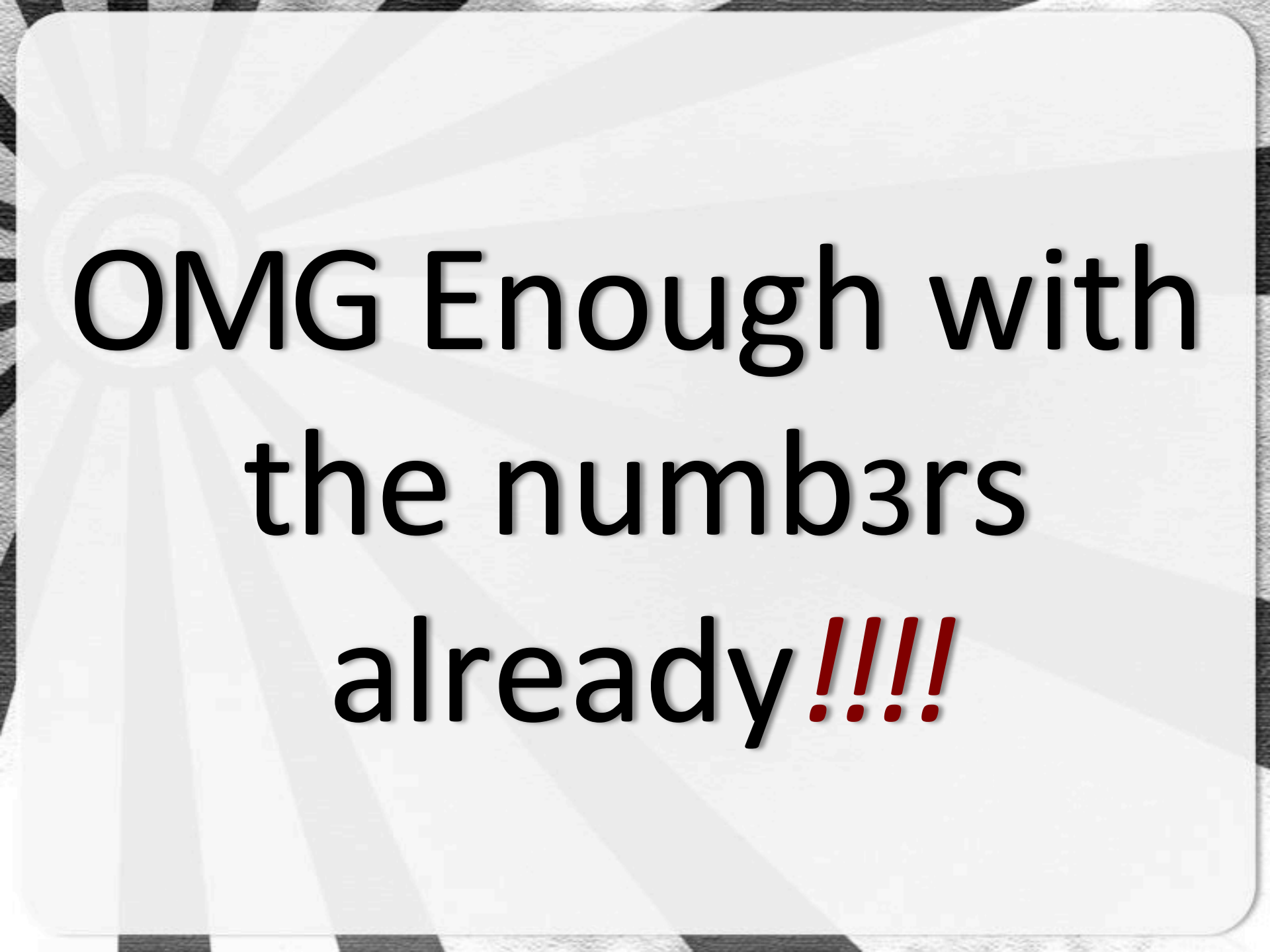
- Server error occurred
 - 500 Internal Server Error
 - 501 Not Implemented
 - 502 Bad Gateway
 - 503 Service Unavailable
 - 504 Gateway Timeout
 - 505 Method Not Allowed

5XX Server Error (cont.)

- 506 Variant Also Negotiates (RFC 2295)
- 507 Insufficient Storage (WebDAV RFC 4918)
- 508 Loop Detected (WebDAV RFC 5842)
- 509 Bandwidth Limit Exceeded (apache ext.)
- 510 Not Extended (RFC 2274)

Codes not supported by Apache

- 511 Network Authentication Required (RFC 6585)
- 550 Permission Denied
- 598 Network Read Timeout Error (Microsoft Proxy)
- 599 Network Connection Timeout Error (Microsoft Proxy)



OMG Enough with
the numb3rs
already!!!!



A large, faint sunburst graphic is centered in the background, with rays extending from a central point in the upper left towards the bottom right. The rays are light gray and create a sense of depth and focus.

#2

[WHY?]

It started as a simple idea...



MODSECURITY HANDBOOK

The Complete Guide to the Popular
Open Source Web Application Firewall



Ivan Ristić



the Tangled Web

*A Guide to Securing Modern
Web Applications*



Michał Zalewski





... and started to think






**SCREW WITH
SCANNERS**

A large, faint watermark of a copyright symbol (©) is visible in the upper left corner of the slide, with radiating lines extending from it across the background.

... AND SCRIPT

K1DD13S

A stylized sun with rays in the background. The sun is on the left side, with rays extending across the entire frame. The text is centered in the middle of the image.

**THAT SOUNDS
LIKE FUN!**



the grugq

@thegrugq

 **Follow**

Stop dismissing “obscurity” as a security feature, because “unpredictability” in your defences works to your advantage.

 Reply  Retweet  Favorite  More

28
RETWEETS

10
FAVORITES



10:19 AM - 26 Feb 13

@thegrugq 26 Feb 2013



the grugq

@thegrugq

 Follow

[@dhw](#) unpredictability is about increasing attacker costs, delaying their operation and increasing their potential for errors.

 Reply  Retweet  Favorite  More

10:26 AM - 26 Feb 13

@thegrugq 26 Feb 2013



INCREASE

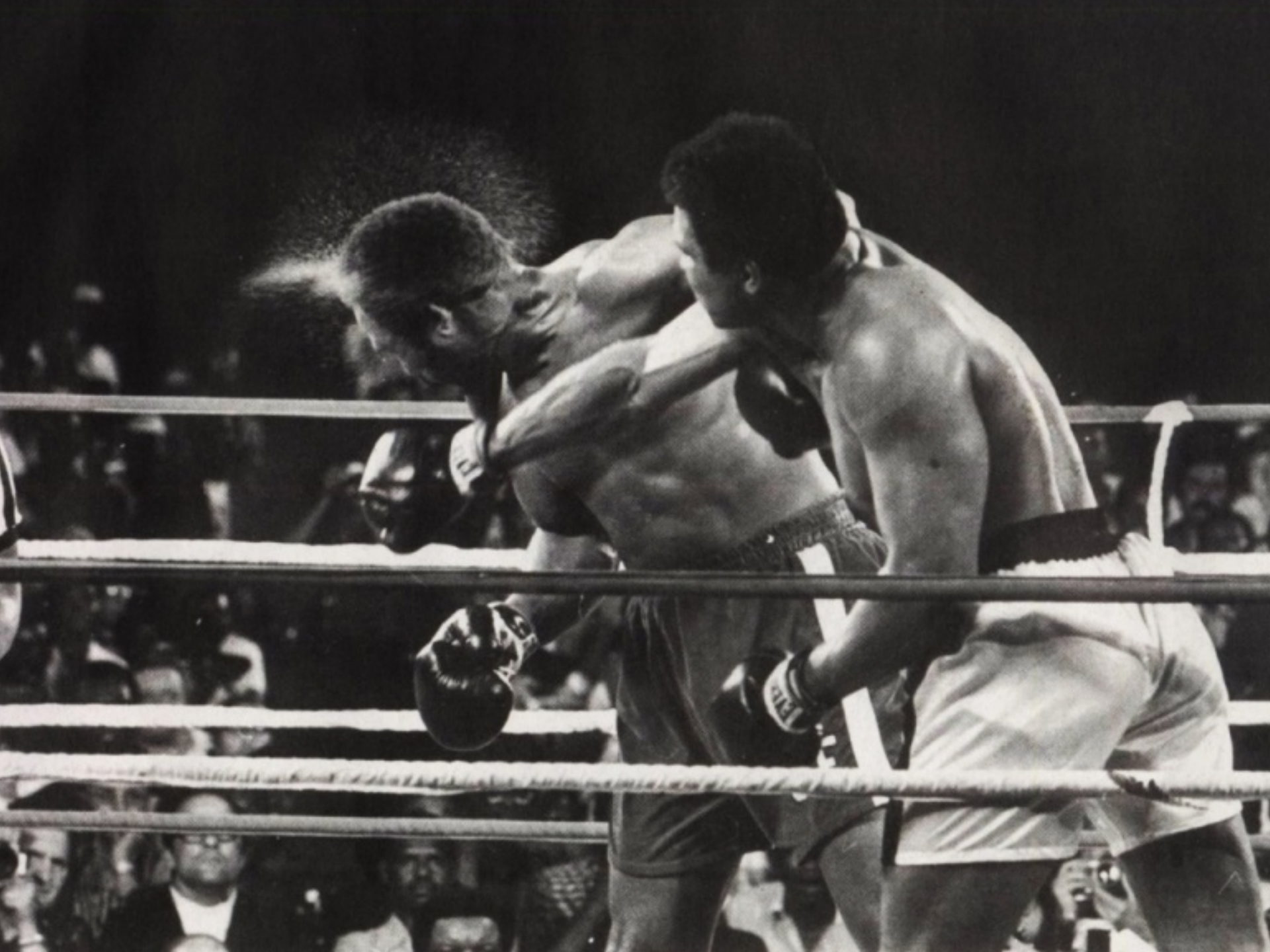
ATTACKER COSTS ^{\$} ^{\$} _{\$}



WASTE

ATTACKER TIME





Prior Art

- **When the tables turn (2004)**
 - *Roelof Temmingh, Haroon Meer, Charl van der Walt*
 - <http://slideshare.net/sensepost/strikeback>
- **Stopping Automated Attack Tools (2006)**
 - *Gunter Ollmann*
 - <http://www.technicalinfo.net/papers/StoppingAutomatedAttackTools.html>

Prior Art

- mod-security mailing list (2006)
 - Status Code 503 together with Retry-After header
 - *Ryan Barnett*
 - <http://bb10.com/apache-mod-security-user/2006-12/msg00042.html>

```
SecFilterDefaultAction "deny,log,status:503"  
SecFilter ".*"  
Header set Retry-After "120"
```

A large, faint sunburst graphic with rays emanating from the top-left corner, set against a light gray background with rounded corners.

#3

[HOW?]

A light gray sunburst graphic with a central circular core and radiating lines, set against a white background with rounded corners.

**BROWSERS HAVE
TO BE FLEXIBLE**

THIS LEADS TO INTERPRETATION

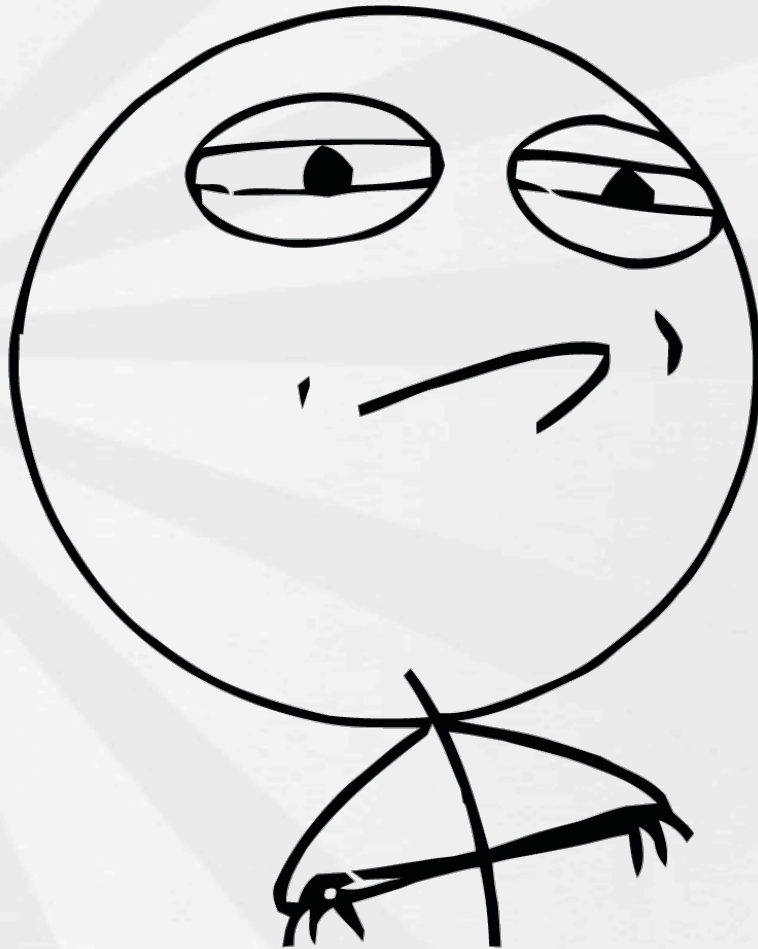
... which leads to the dark-side





RFCs...

**THEY'RE MORE OF A
GUIDELINE REALLY**



**WHAT COULD POSSIBLY GO
WRONG!**

A large, light gray sunburst graphic with a central circle and radiating lines, positioned on the left side of the slide. The central circle is black and contains the text "#3.1".

#3.1

TESTING

THE **HOW** OF THE THING!

- Restricted research to the **big 3**
 - Internet Explorer
 - Chrome / Chromium
 - Firefox



**NO... SAFARI ISN'T
IN THE TOP 10 3**



OPERA JUMPED...

...or was it pushed!





LYNX

THE UNREALISTIC OPTION

- MITMproxy / MITMdump
 - Python-based
 - Simple to setup proxy / reverse proxy
 - Script-based actions

```
def response(context, flow):  
    if flow.response.code != respcode:  
        # alter response code and message  
        flow.response.code = respcode  
        flow.response.msg = respmsg
```

```
respcode = 200  
respmsg = "OK"
```

- PHP
 - Ability to set response code
 - Must be at the top of the PHP code
 - Can be added to php.ini
 - auto-prepend-file = /full/path
 - Limited by web-server (apache)

```
# set response code  
Header($_server["SERVER_PROTOCOL"]. " $status_code");
```

- Testing browsers automatically
 - Created PHP file to set status code
 - - <http://c22.cc/POC/resrcode.php?code=XXX>

Test Results

Requested Response Code .: 426

Actual Response Code .: 426

Headers .:

HTTP/1.1 426 Upgrade Required

Date: Sun, 31 Mar 2013 13:57:57 GMT

Content-Encoding: gzip

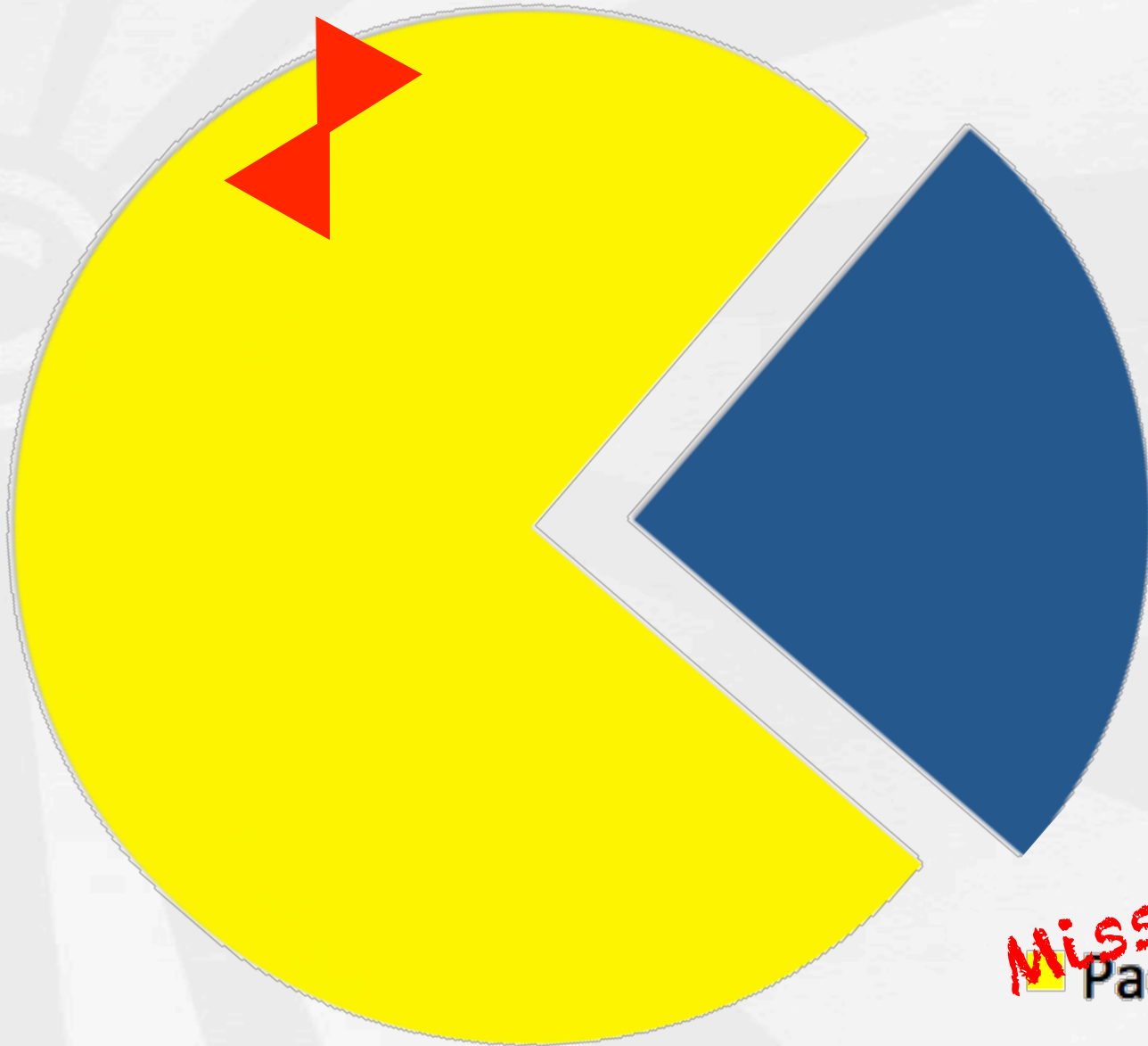
Server: ./msfcli auxiliary/server/capture/http set SRVPORT=80

A large, light gray sunburst graphic with a central circle and radiating lines, serving as a background for the slide.

#3.2

BROWSERS

... AND THEIR **STATUS CODE** HABITS



MISS
■ Pac-Man

■ Not Pac-Man



Status Code	HTML	iFrame	JavaScript	HTML	iFrame	JavaScript	HTML	iFrame	JavaScript
300			✗						
301			✗				✗	✗	✗
302			✗				✗	✗	✗
303			✗				✗	✗	✗
304	✗	✗	✗	✗	✗	✗	✗	✗	✗
305			✗						
306			✗						
307			✗				✗	✗	✗



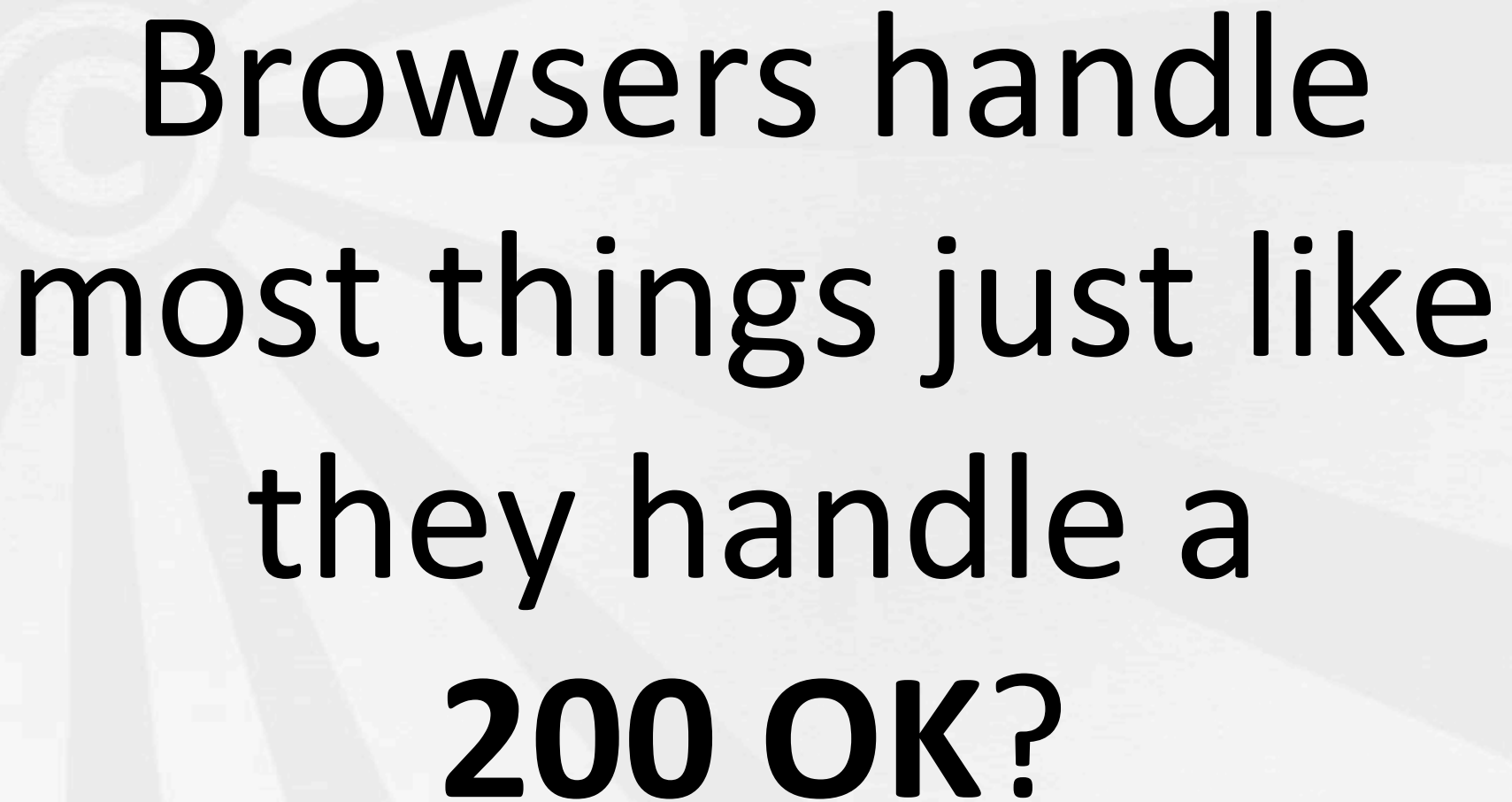
Status Code	HTML	iFrame	JavaScript	HTML	iFrame	JavaScript	HTML	iFrame	JavaScript
400			✘			✘		✘	✘
401			✘			✘			✘
402			✘			✘			✘
403			✘			✘		✘	✘
404			✘			✘		✘	✘
405			✘			✘		✘	✘
406			✘			✘		✘	✘
407			✘	Proxy	Proxy	Proxy			✘
408	✘	✘	✘			✘		✘	✘
409			✘			✘		✘	✘
410			✘			✘		✘	✘
411			✘			✘			✘
↓			✘			✘			✘
426			✘			✘			✘



Status Code	HTML	iFrame	JavaScript	HTML	iFrame	JavaScript	HTML	iFrame	JavaScript
500			✘			✘		✘	✘
501			✘			✘		✘	✘
502			✘			✘			✘
503			✘			✘			✘
504			✘			✘			✘
505			✘			✘		✘	✘
506			✘			✘			✘
507			✘			✘			✘
508			✘			✘			✘
509			✘			✘			✘
510			✘			✘			✘

Loading... Please Wait





Browsers handle
most things just like
they handle a
200 OK?



YEP...

MOSTLY

A decorative background featuring a stylized sunburst or starburst pattern in shades of gray, centered on the left side of the slide. The rays emanate from a central point, creating a sense of light and energy.

- HTML Responses

- Almost all response codes are rendered by the browser correctly

- iFrames

- Some special cases for IE, but other browsers handle this the same as HTML

- JavaScript/CSS
 - Limited accepted status codes
 - Limited 3XX support
 - Chrome is the exception here
 - No support for 4XX/5XX codes



JavaScript

So we know
what browsers
interpret
differently



What do
browsers have
in *common*?

- 1XX code handling
 - Retries
 - Confusion
 - Chrome / IE6 try to **download** the page!
 - Fun on Android... (never ending download)
 - Timeouts
 - Eventually

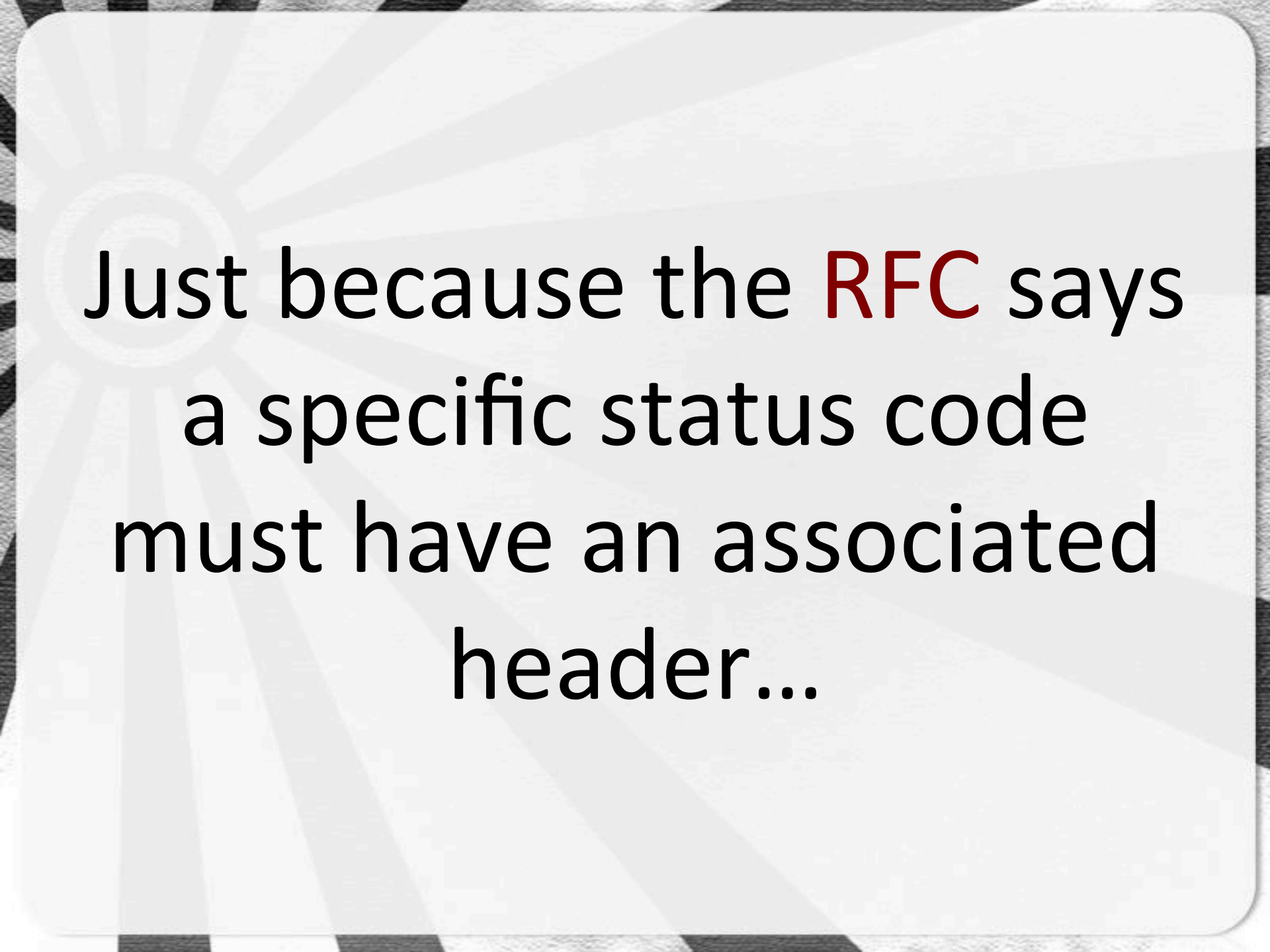
- 204 No Content
 - Um, no content!
- 304 Not Modified
 - Again, no content returned



A large, light gray sunburst graphic with many rays emanating from a central point on the left side of the slide. The rays spread out across the entire background.

#3.3

WHAT ABOUT HEADERS?



Just because the **RFC** says
a specific status code
must have an associated
header...



...doesn't mean
it *HAS* to


- Redirection codes (301-304, 307)
 - No Location header, no redirect
- 401 Unauthorized
 - No WWW-Authenticate header, no authentication prompt
- 407 Proxy Authentication Required
 - No Proxy-Authenticate header, no prompt

Just because the **RFC** says
a specific status code
shouldn't have an
associated header...

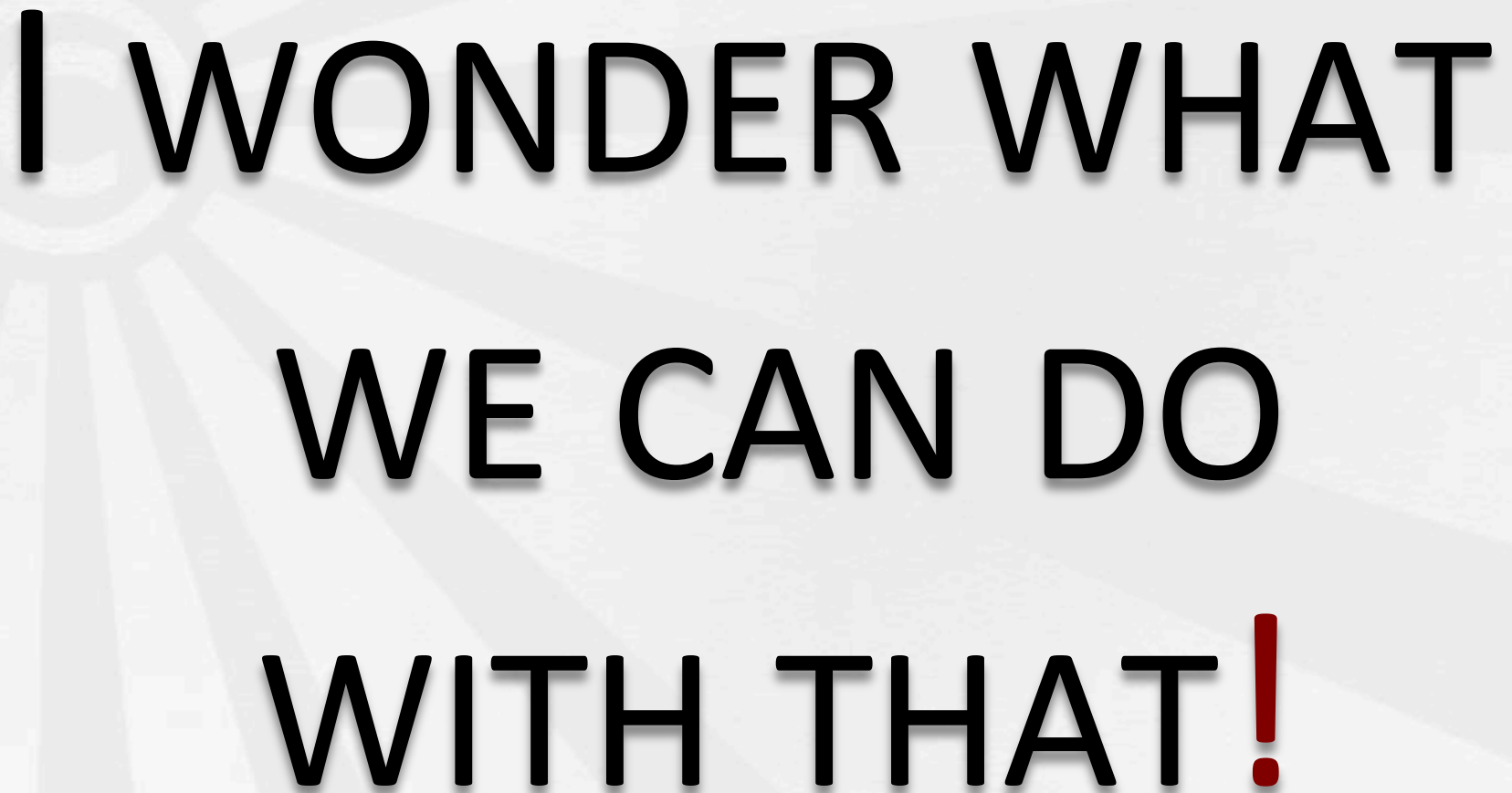


...doesn't mean
it *can't*

- 300 Multiple Choices w/ Location Header
 - Firefox / IE6 follows the redirect
 - Chrome doesn't
- More research needed in this direction
 - Most are uninteresting

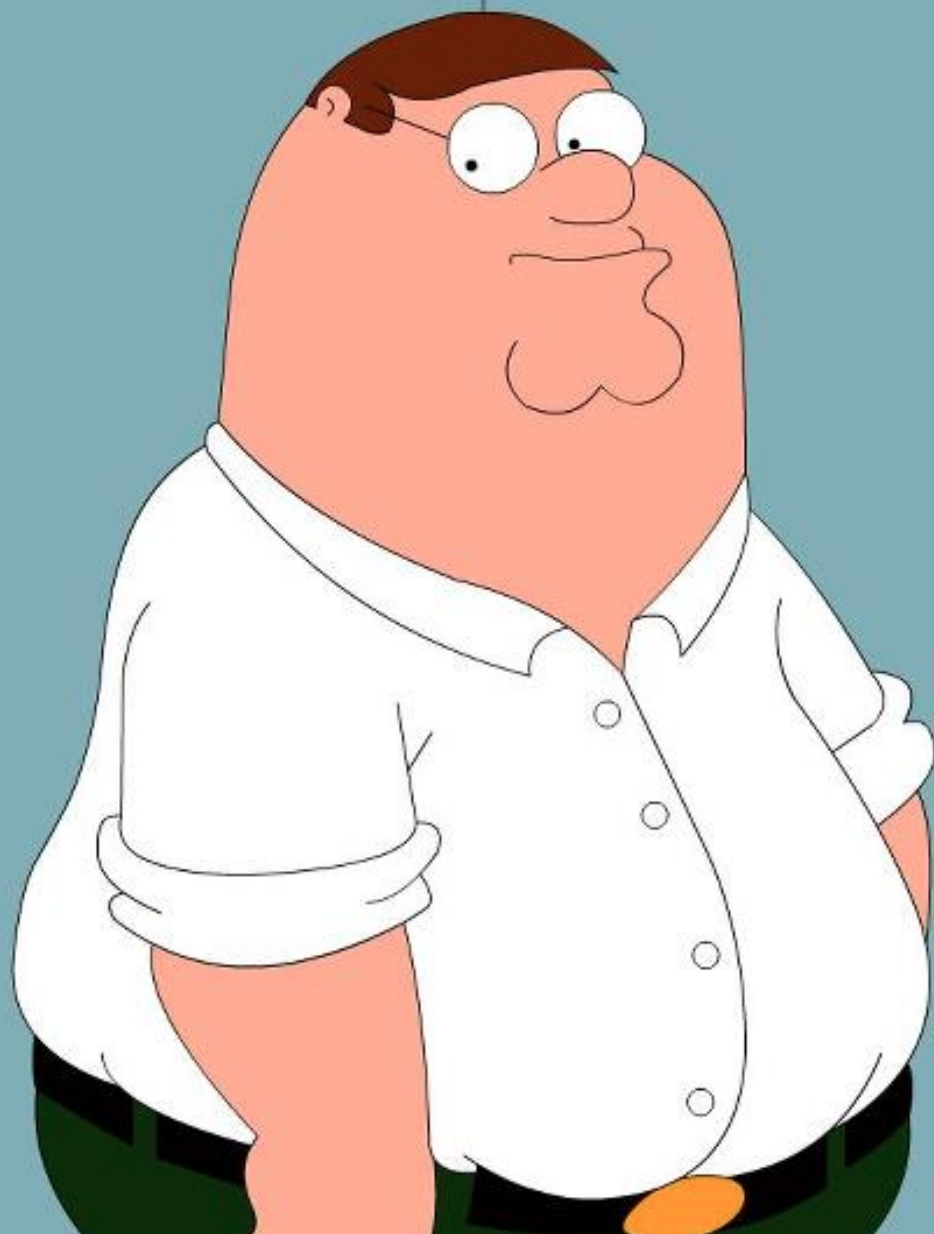


**EACH BROWSER
HANDLES THINGS A
LITTLE DIFFERENTLY**

A stylized sun with rays in the background. The sun is on the left side, with rays extending across the frame. The background is a light gray with a subtle pattern of rays.

**I WONDER WHAT
WE CAN DO
WITH THAT!**

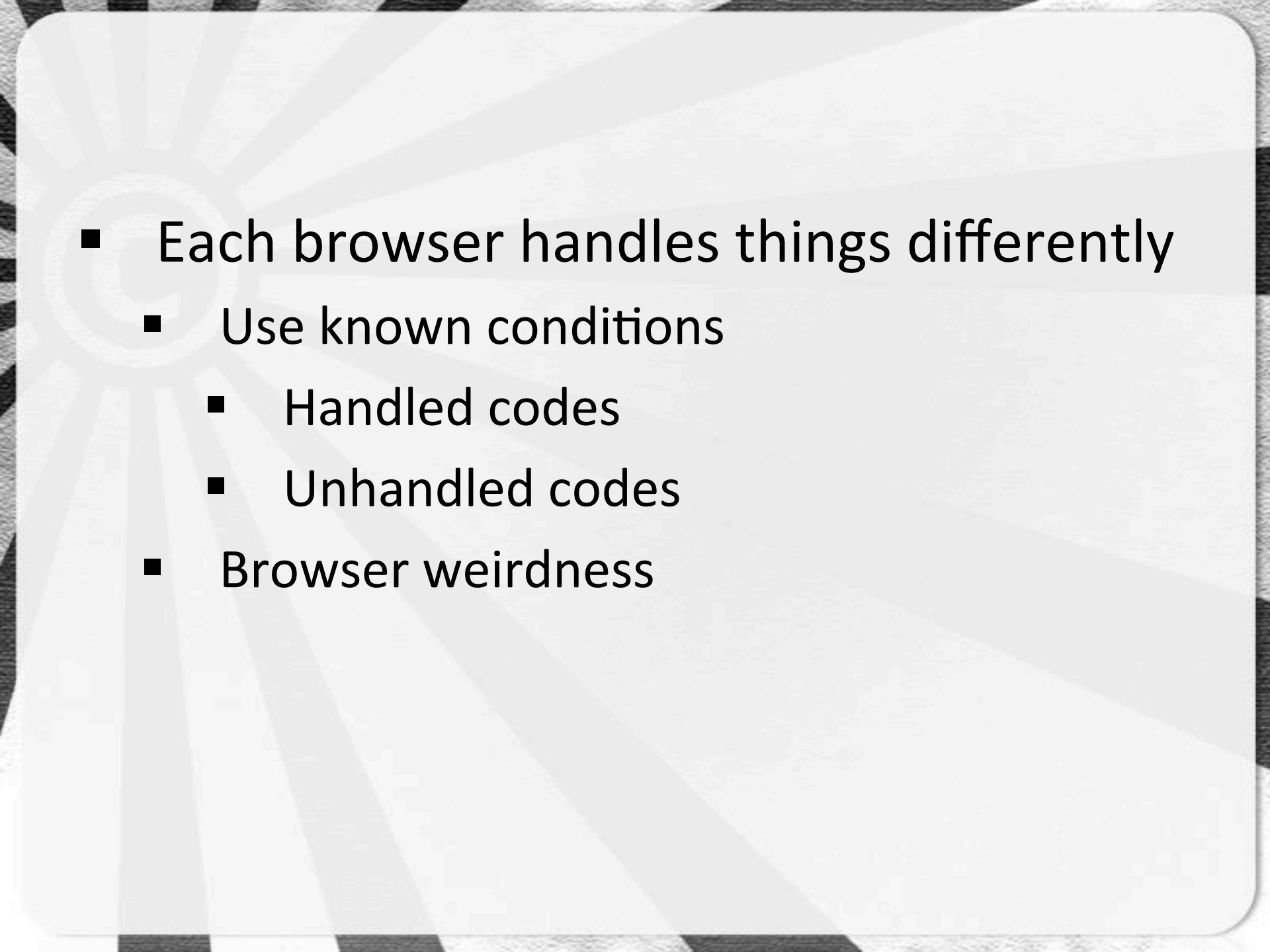
**DO NOT
PUSH
BUTTON**



A light gray sunburst pattern with rays emanating from the top left corner, set against a white background with rounded corners.

#4

[GOALS]

- 
- Each browser handles things differently
 - Use known conditions
 - Handled codes
 - Unhandled codes
 - Browser weirdness

A large, light gray sunburst graphic with multiple rays emanating from a central point on the left side of the slide. The rays extend across the entire width and height of the slide, creating a background pattern.

#4.1

BROWSER FINGERPRINTING

Firefox

- Doesn't load JavaScript returned with a 300 'Multiple Choices' status code
 - Other browsers tested **DO** (IE/Chrome)
- Request JavaScript from server
- Response Status: **300 Multiple Choices**
- If JavaScript doesn't run in the browser
 - Firefox



Chrome

- Loads JavaScript returned with a 307 'Temporary Redirect' status code
 - Other browsers tested **DON'T** (IE/FF)
- Request JavaScript from server
- Response Status: 307 Temporary Redirect
- If JavaScript runs in the browser
 - Chrome




Internet Explorer

- Loads JavaScript returned with a 205 'Reset Content' status code
 - Other browsers tested **DON'T** (FF/Chrome)
- Request JavaScript from server
- Response Status: 205 Reset Content
- If JavaScript runs in the browser
 - Internet Explorer



- Other options to fingerprint browsers
 - 300 Redirect (Chrome)
 - 305 / 306 JavaScript (Firefox)
 - 400 iFrame (Internet Explorer)
 - ...



**BROWSER
FINGERPRINTING
DEMO**

A background graphic of a sun with rays, rendered in a light gray color, positioned on the left side of the slide. The sun's rays extend across the slide, creating a sense of light and focus.

USER-AGENTS

CAN BE
SPOOFED



BROWSER

TRAITS *CAN'T*

A large, faint sunburst graphic is centered in the background of the slide. It consists of a central circle with numerous rays extending outwards, creating a radial pattern across the entire page.

#4.2

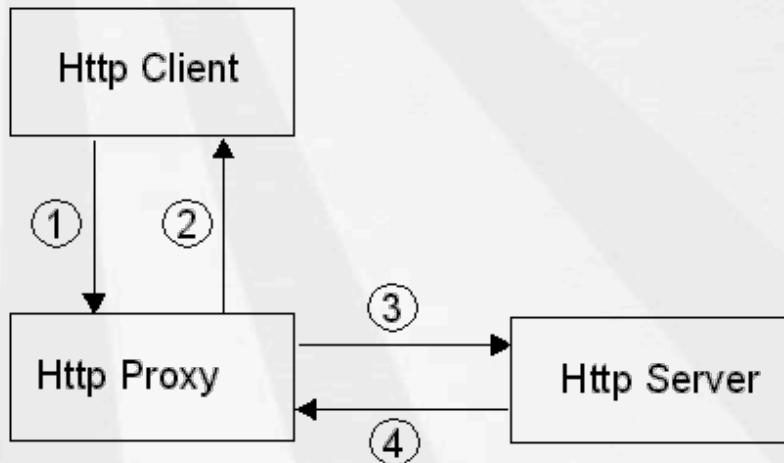
PROXY DETECTION

Chrome Proxy Detection

- Chrome handles proxy configuration differently to other browsers
 - 407 status code isn't rendered
 - Unless an **HTTP** proxy is set!
- Allows us to detect if an HTTP proxy is set
- Just not which proxy
 - Can only detect HTTP proxies ;(

Chrome Proxy Detection

- Request page from server
- Response Status: 407 Proxy Authentication
 - w/o Proxy-Authenticate header
- If Chrome responds HTTP proxy is set



BONUS

STAGE

Side-Effect: Owning Proxies

- Privoxy 3.0.20 ([CVE-2013-2503](#))
 - 407 Proxy Authentication Required
 - w/ Proxy-Authenticate header
 - User prompted for user/pass
 - Prompt appears to be from Privoxy
 - Privoxy passes user/pass to remote site
 - Profit???

Side-Effect: Owning Proxies

- Not just Privoxy that's effected
 - Any transparent proxy
 - e.g. Burp, ZAP, ...
 - Not really a vuln for most
 - Works as designed!

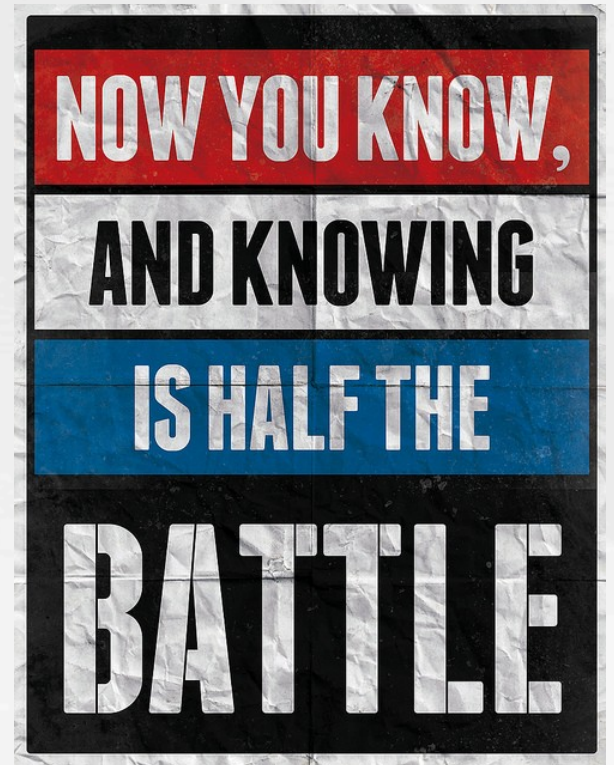
A light gray sunburst graphic with rays emanating from a central circle in the top-left corner. The central circle is black and contains the text "#5".

#5

BRINGING **IT** ALL
TO **O**GETHER

What we have

- Status codes all browsers treat as **content**
- Status codes all browsers can't handle
 - **1XX**, etc..
- Lots of browser quirks



What can we do

- F*ck with things
- Screw with scanner monkeys
- Make RFC lovers cry into their beer
- Break things in general

A red banner with the word "START" in white, bold, capital letters. The banner is hanging from a string and has a slightly wrinkled texture. The background of the slide features a faint sunburst pattern and a large, light-colored copyright symbol (©) on the left side.

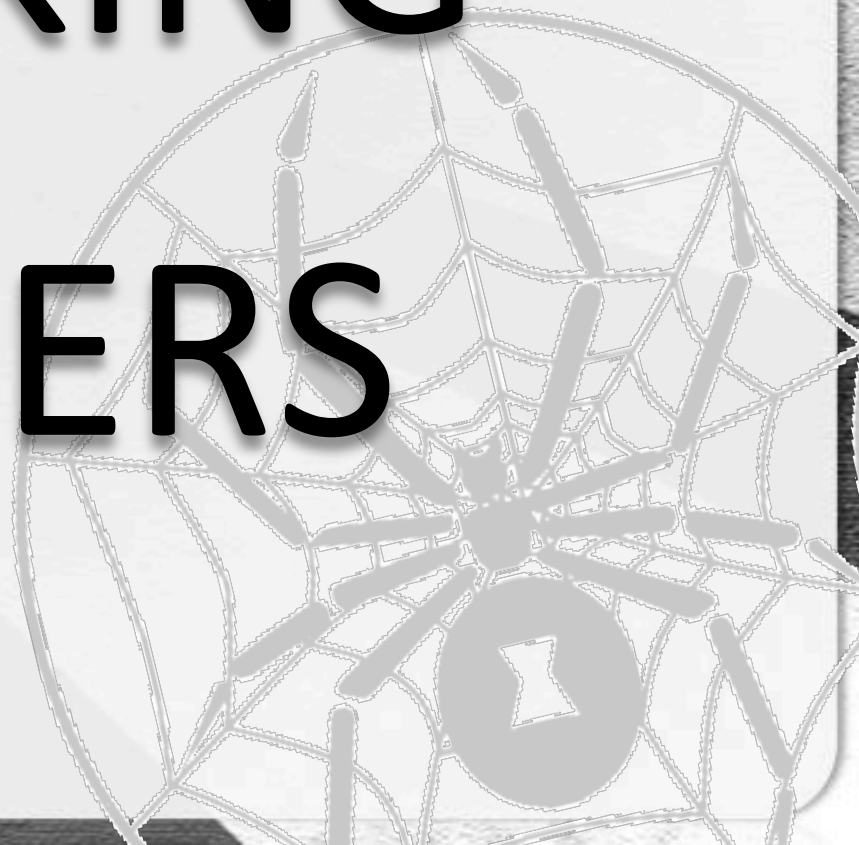
START

Let's try to...

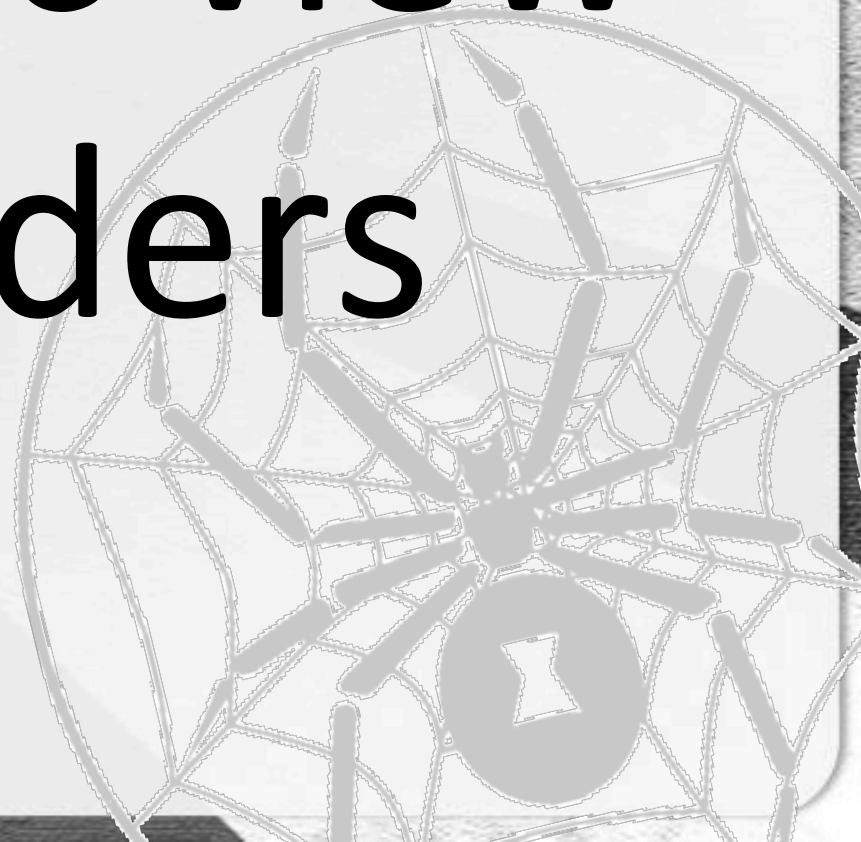
- Use what we've discovered to...
 - Break spidering tools
 - Cause false positives / negatives
 - Slow down attackers
 - The **fun** way!
 - Blocking successful exploitation

#5.1

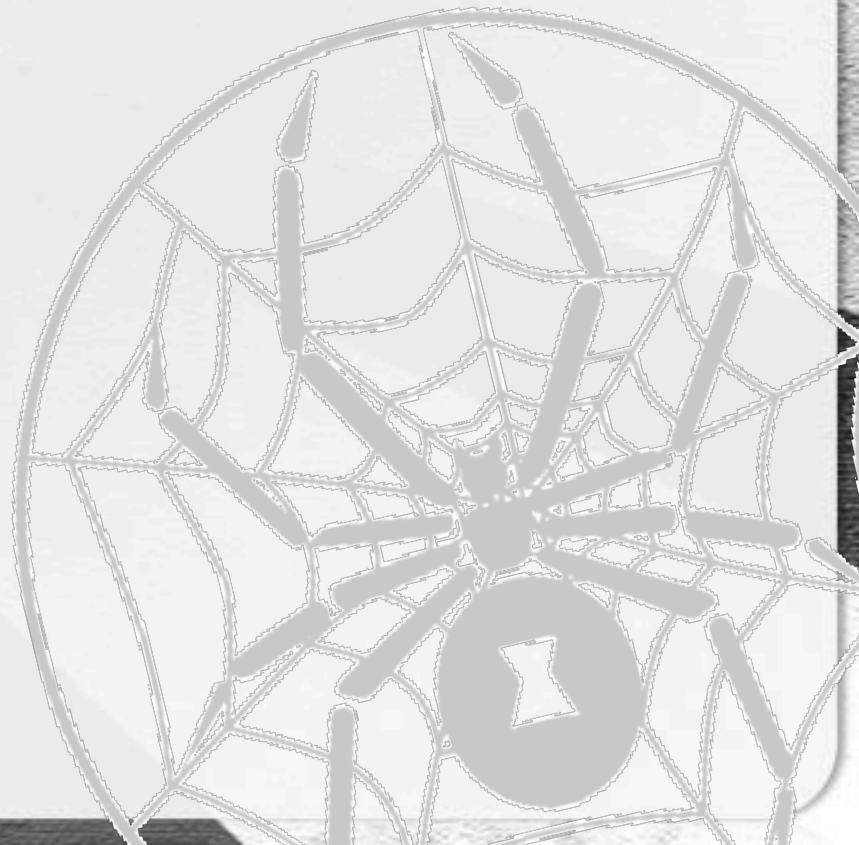
BREAKING SPIDERS



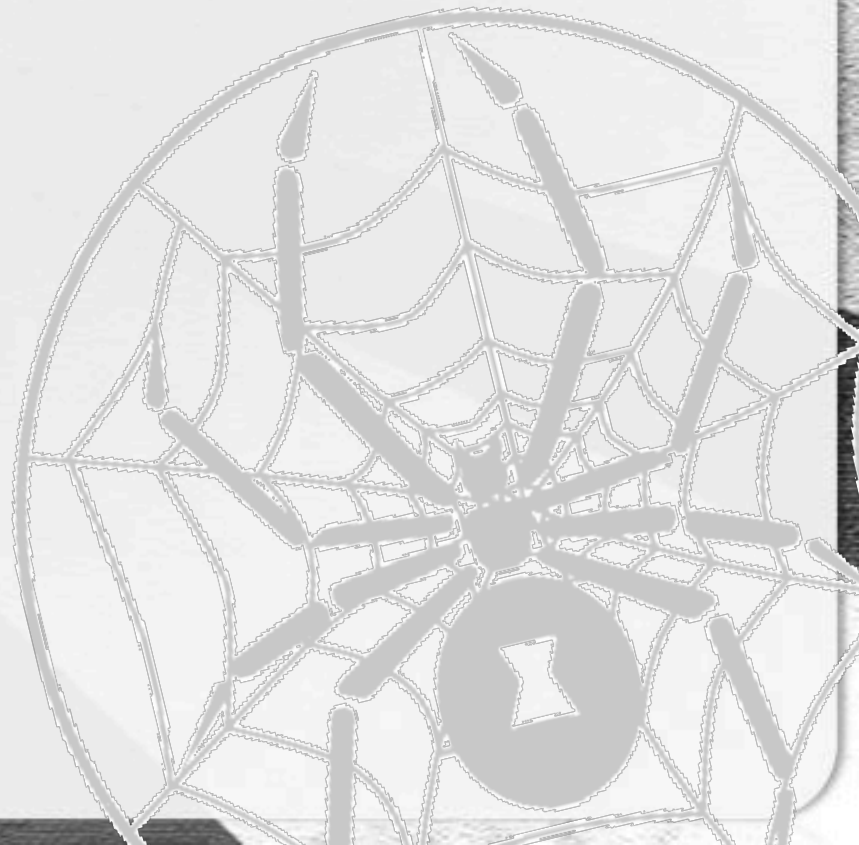
Simplistic view of spiders



- Access target URL
- Read links / functions
- Test them out
- If true: continue
 - What is **TRUE**?



- What happens **if**:
 - Every response is
 - 200
 - 404
 - 500




200 OK

- IF 200 == True:
 - Problems!
 - Never-ending spider









404 Not Found

- IF 404 == False:
 - What website?



Acunetix Threat Level 2
One or more medium-severity type vulnerabilities have been by the scanner. You should investigate each of these vulnerabilities to ensure they will not escalate to more severe problems.

 Alerts summary	3 alerts	
 Statistics	868 requests	
 Target information	http://192.168.0.150:80/	
Port scanner	Finished	
Crawler	Finished	
Files found	0	
Directories found	0	
Variations found	0	
Scripting	Finished	

500 Internal Server Error

- Skipfish != happy fish

```
skipfish version 2.09b by lcamtuf@google.com
- default.testapache.local -
Scan statistics:
  Scan time : 0:20:08.162
  HTTP requests : 22339 (18.6/s), 63885 kB in, 7526 kB out (59.1 kB/s)
  Compression : 56992 kB in, 1010083 kB out (89.3% gain)
  HTTP faults : 38 net errors, 0 proto errors, 0 retried, 0 drops
  TCP handshakes : 50 total (466.8 req/conn)
  TCP faults : 0 failures, 38 timeouts, 2 purged
  External links : 21724 skipped
  Reqs pending : 1001
Database statistics:
  Pivots : 2461 total, 2174 done (88.34%)
  In progress : 136 pending, 99 init, 37 attacks, 15 dict
  Missing nodes : 5 spotted
  Node types : 1 serv, 242 dir, 4 file, 0 pinfo, 90 unkn, 87 par, 2037 val
  Issues found : 2421 info, 15 warn, 2095 low, 2107 medium, 3 high impact
  Dict size : 52 words (52 new), 4 extensions, 256 candidates
  Signatures : 75 total
Killed
root@bt:~/pentest/web/skipfish#
```

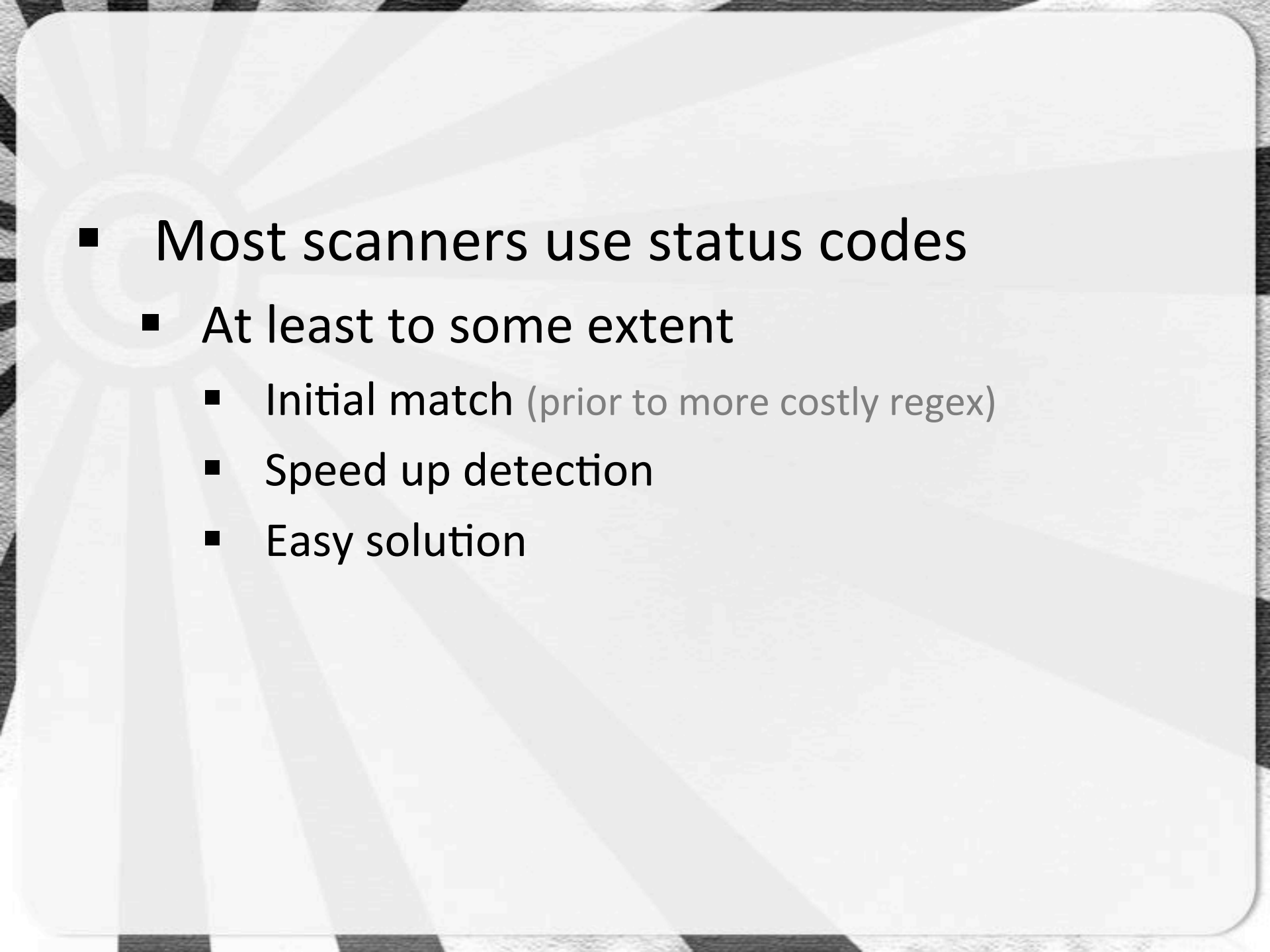
A large, light gray sunburst graphic is centered on the page, with rays extending outwards. A black circle with a white border is positioned at the top left of the sunburst, containing the text "#5.2".

#5.2

False

Positives /

Negatives

- 
- Most scanners use status codes
 - At least to some extent
 - Initial match (prior to more costly regex)
 - Speed up detection
 - Easy solution

- What happens **if**:
 - Every response is
 - 200
 - 404
 - 500
 - raNd0M*



* Using codes that are accepted by all browsers as content

Vulnerability Baseline

- w3af
 - Information Points → 79
 - Vulnerabilities → 65
 - Shells → 0 shells 😞
 - Scan time → 1h37m23s

Every response 200 OK

- No change in discoveries
 - All points discovered - per baseline
 - 79 Information Points
 - 65 Vulnerabilities
 - 0 Shells
 - Scan time → 9h56m55s
 - Lots more to check ;)

Every response 404 Not Found

- Less to scan == Less to find
 - False negatives
 - 44 Information Points (-35)
 - 37 Vulnerabilities (-28)
- Scan time → 7m13s
 - Much quicker scan
 - Less paths traversed



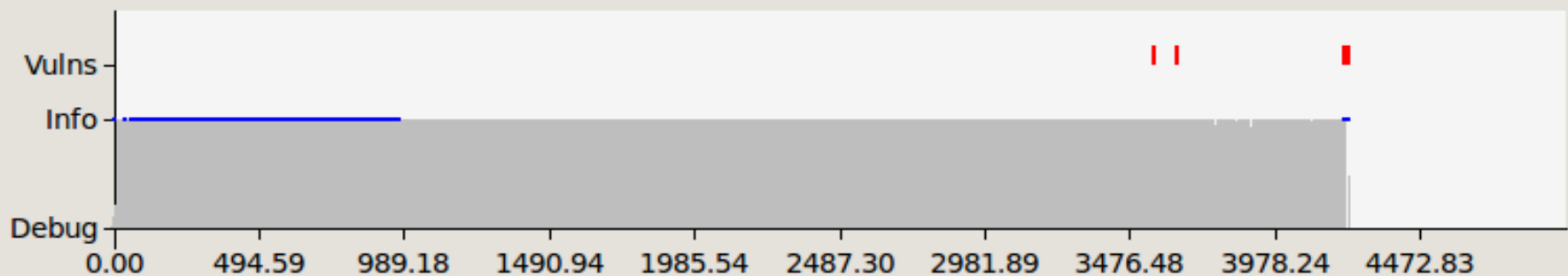
Every response 500

- Server Error == OMG VULN!
 - False positives+++
 - 9540 Information points (+9461)
 - 9526 Vulnerabilities (+9461)

[Sat 06 Apr 2013 04:53:24 PM CEST] Scan finished in 1 hour 10 minutes 29 seconds.

Audit progress: 0.0 % - ETA: 00d 00h 00m 00s

Not running.



i 9540 ⚠ 9526 🟢 0

Random Status Codes

- Multiple test runs
 - All tests produced False positives++
 - avg. 619 Information points (+540)
 - avg. 550 Vulnerabilities (+485)
- Avg. scan time → 11m37s
 - Often much quicker scans
 - Lots of variation in scan times

Random Status Codes

- Skipfish + \$random_status = chaos
 - False Positives + False Negatives
 - Scan jobs killed (due to lack of scanner resources)
- Scan times
 - 1st scan time → 10h3m35s
 - 2nd scan time → 0h0m4s
 - 3rd scan time → 16h47m41s



#5.3

Slowing
attackers
down!

What does
your **WAF**
really do?




No entry

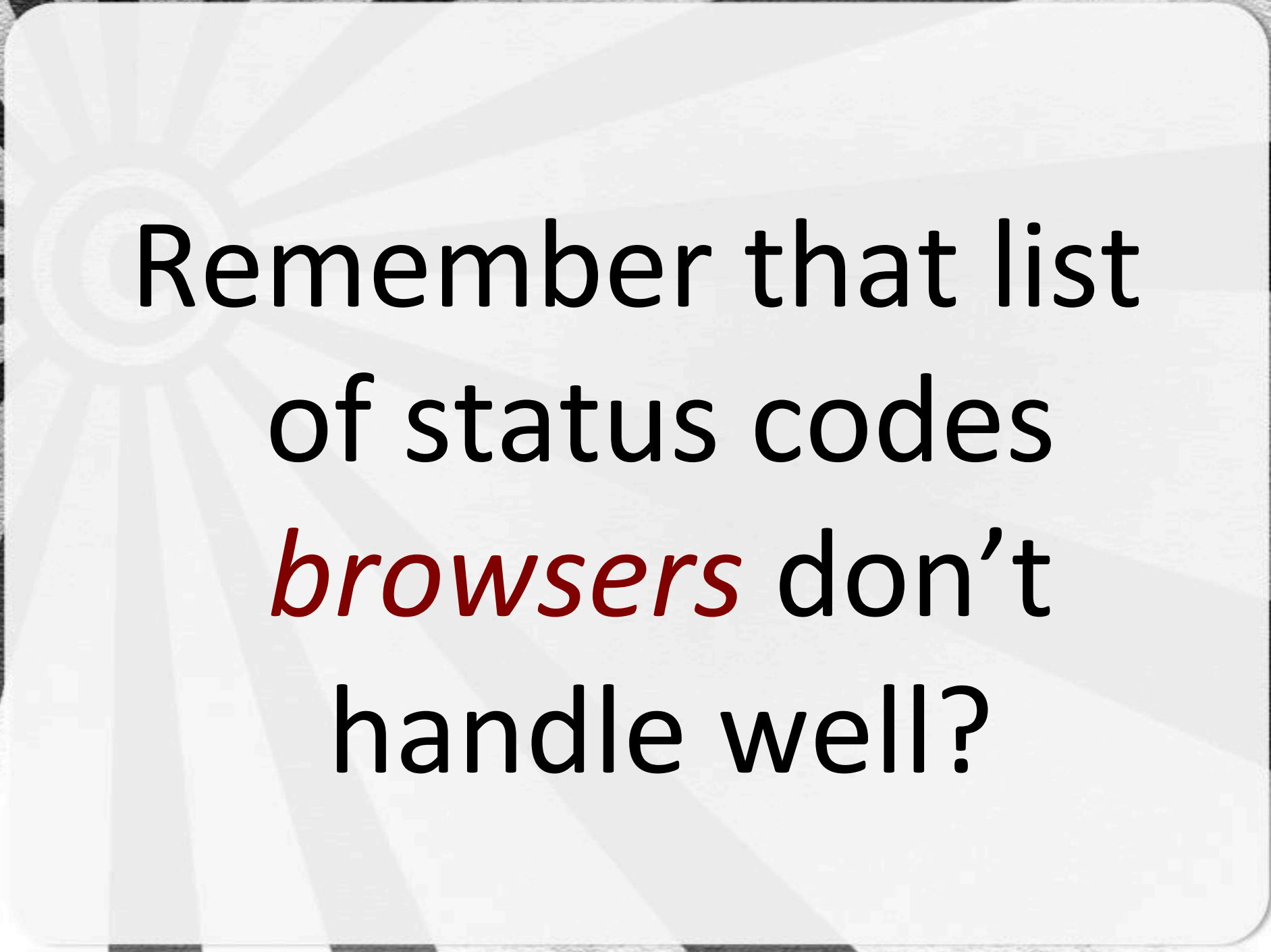
- **OMG Attack**
- **Block / Return error**
 - 403,500, ...
- **Profit???**



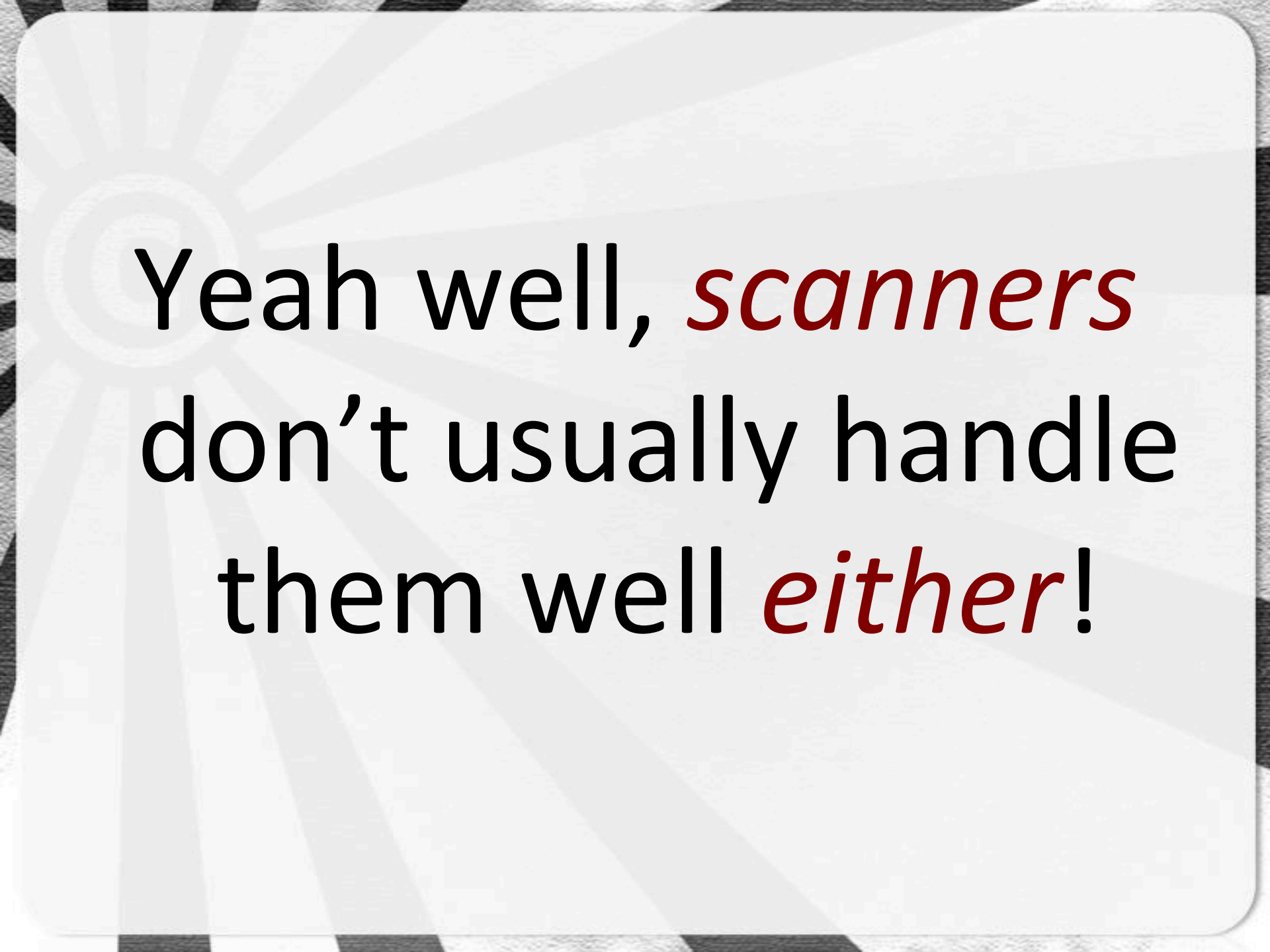
No entry



why?



Remember that list
of status codes
browsers don't
handle well?



Yeah well, *scanners*
don't usually handle
them well *either!*



Especially the
1XX codes

- Remember LaBrea tarpit?
 - Tim Liston 2001 *
 - Designed to **slow** spread of Code Red
 - Slows down scans / attackers

* <http://labrea.sourceforge.net>



IDEA



How about an
HTTP Tarpit!

CASE IN POINT

by Tom Fishburne

WHERE DID ALL THIS
T.A.R. COME FROM?



HTTP Tarpit Scenario

- WAF detects scan / attack
- Adds source IP to “naughty” list
- Rewrite all responses from the server
 - 100 | 101 | 102 status codes only (random)
 - 204 | 304 might also be useful (no content)

Let's do
some
science!*



* Science not included

NIKTO

vs. the HTTP TARPIT



Baseline

HTTP Tarpit

Scan time

2m 18s

14h 33m 2s

Findings

18

10

W3AF

vs. the HTTP TARPIT



Baseline

HTTP Tarpit

Scan time

1h 37m 23s

18m 10s

Findings

65

0

SKIPFISH

vs. the HTTP TARPIT

Google code

skipfish

Baseline

HTTP Tarpit

Scan time

18m 10s

05s

Findings

Low: 2519

Low: 0

Med: 2522

Med: 0

High: 12

High: 3

ACUNETIX

vs. the HTTP TARPIT



Baseline

HTTP Tarpit

Scan time

1h 19m

33m

Findings

Info: 1104

Info: 3

Low: 30

Low: 3

Med: 32

Med: 1

High: 24

High: 0

HTTP Tarpit

- HTTP Tarpit Results *
- Slow down scans (nikto)
 - 340x as long
 - Other give up quicker ;)
- Unreliable / aborted scans
 - Up to 100% less findings

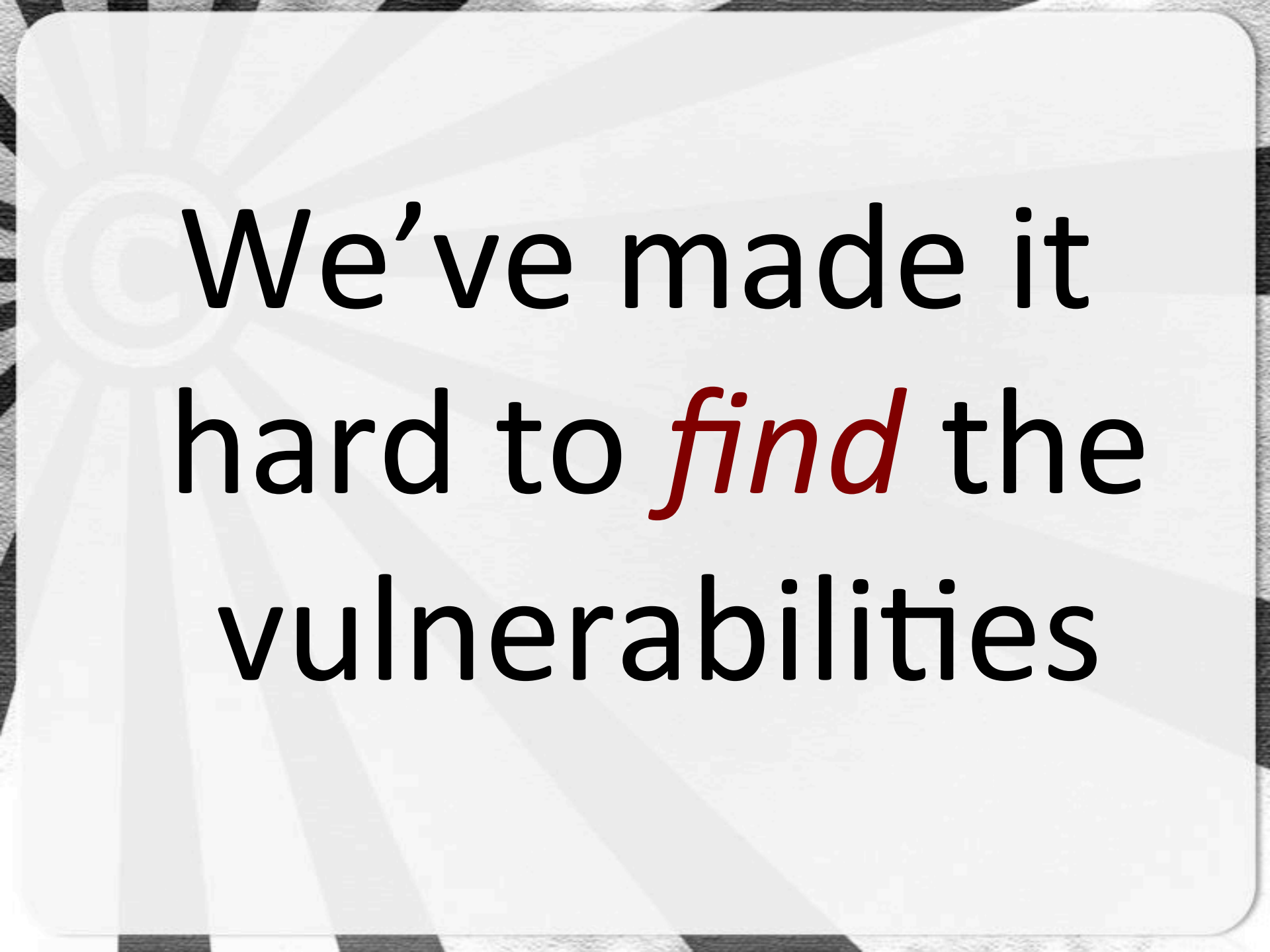


* Not scientifically sound ;)




#5.4

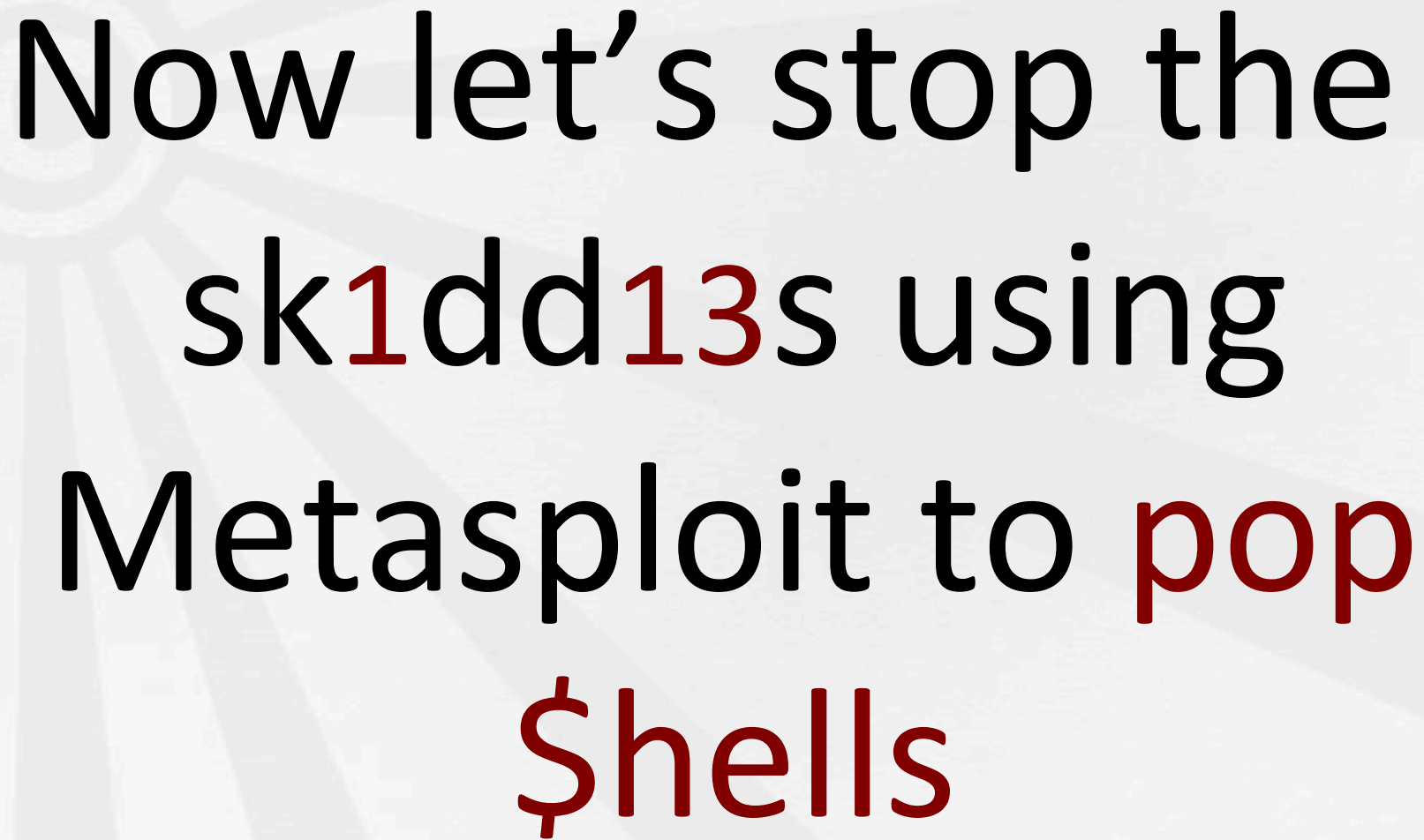
Blocking successful exploitation



We've made it
hard to *find* the
vulnerabilities



We've made it
time consuming
for attackers




Now let's stop the
sk1dd13s using
Metasploit to pop
\$hells

Q: How often does Metasploit reference status codes?

```
rgrep -E 'res[p|ponse]?\.code' *
```

→ 958 *

* Not scientifically sound ;)



Lots of
dependency on
status codes*

* yep, even the stuff I wrote

```
if (res.code < 200 or res.code >= 300)
```

```
  case res.code
```

```
  when 401
```

```
    print_warning("Warning: The web site  
asked for authentication: #{res.headers  
['WWW-Authenticate']} || res.headers  
['Authentication']}")
```

```
  end
```

```
  fail_with(Exploit::Failure::Unknown,
```

```
    "Upload failed on #{path_tmp}  
[#{res.code} #{res.message}]")
```

```
end
```

No match,

No shell*

* exploit dependent

A large, faint sunburst graphic is centered in the background, with rays extending outwards. In the top-left corner, there is a black circle containing the text "#6".

#6

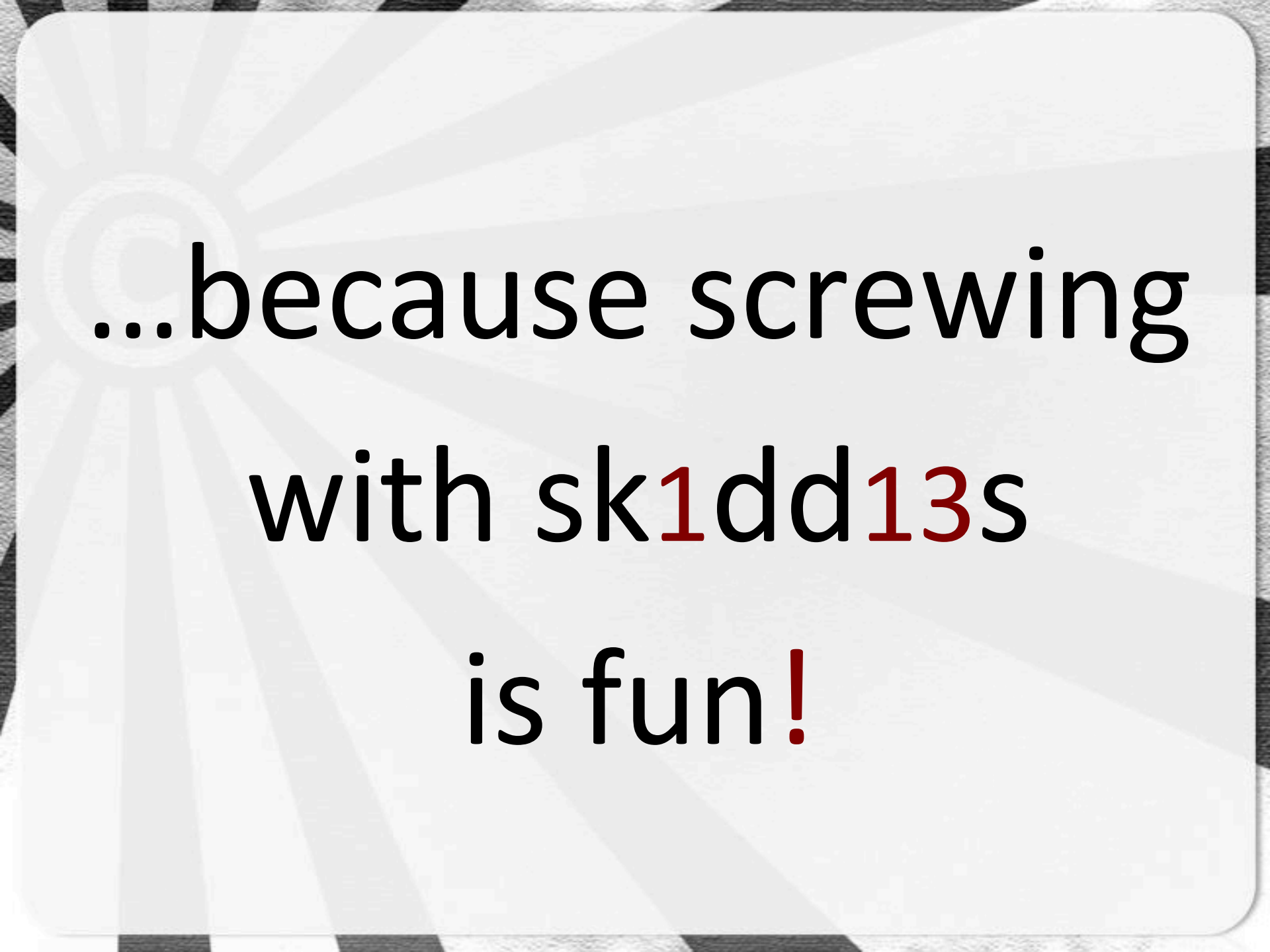
[REVIEW]

- Using status codes to our benefit is fun
 - ... and **useful!**
- Browsers can be quirky
- Scanners / attack toolkits are sometimes set in their ways
 - Take the easy route
 - Easy to fool

- WAFs need to get more offensive about their defense
 - More than just blocking a request
 - Even if you use a snazzy message
 - Hacking back is **bad**
 - **Slowing** down known attacks is good
 - Make life harder for skiddies is pricele**\$\$**

- Current tools are much the same as APT
 - APT (**Adequate** Persistent Threat)
 - Only as advanced as they NEED to be





...because screwing
with sk1dd13s
is fun!

A large, light gray sunburst graphic is centered on the page, with rays extending outwards. In the top-left corner, there is a black circle containing the text "#6.1".

#6.1

Implementation

CRUISE



- Getto implementation
 - PHP (the lowest common denominator)
 - auto-prepend-file
 - Limited to resources PHP handles
 - MITMdump
 - MITMproxy == memory hog
 - Reverse proxy mode



APPROVED



- Usable implementation
 - Nginx as reverse proxy
 - Requires: ngx_lua
 - ngx.status = **XXX**
 - Bugs in non-git version
 - 203, 305, 306, 414, 505, 506 return *nil*

STAY IN LANE

A green rectangular sign with a white border. The text "STAY IN LANE" is written in white, bold, sans-serif capital letters. Below the text, there are two white arrows: one pointing to the left and one pointing to the right.

FUTURE

A green rectangular sign with a white border. The text "FUTURE" is written in white, bold, sans-serif capital letters. Below the text, there are two white arrows pointing downwards.

PAST

A green rectangular sign with a white border. The text "PAST" is written in white, bold, sans-serif capital letters. Below the text, there are two white arrows pointing downwards.

- Ease adoption
 - Implement into mod-security
 - Not a simple task
 - Already been discussed many times
 - Help wanted ;)

A large, light gray sunburst graphic is centered on the page, with rays extending outwards. In the upper left corner, there is a black circle containing the text "#6.2" in white and red.

#6.2

Countering this research

- Less reliance on status codes
- More reliance on content / headers
 - Pros
 - Better matching / intelligence
 - Cons
 - Slower? (regex matching)
 - More resource intensive





Questions?



CODE / SCRIPTS AVAILABLE

[HTTP://GITHUB.COM/CHRISJOHNRILEY/RANDOM_CODE](http://github.com/ChrisJohnRiley/Random_Code)

What doesn't kill you,
makes you smaller!





Thanks for coming

<http://c22.cc>

[@ChrisJohnRiley](#) | contact@c22.cc