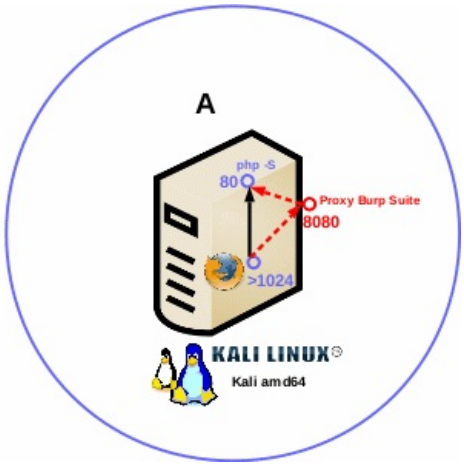
Práctica Seguridade Informática: Métodos GET, POST + Proxy Burp Suite

ESCENARIO

Máquina virtual A:
Rede Interna; ≦ 2048MB RAM; ≤ 2CPU; PAE/NX habilitado
Rede: 192.168.120.0/24
Servidor Web (php): $ sudo php -S localhost:80 -t /home/kali/web
Cliente Web: Navegador
ISO: Kali amd64
IP/MS: 192.168.120.100/24
Proxy burpsuite: 127.0.0.1:8080



LIMITACIÓN DE RESPONSABILIDADE O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

NOTAS:

- **Métodos GET e POST**
- **PortSwigger**
- **Burp Suite Community Edition**

1. **Configuración de rede**
   Na contorna gráfica abrir un terminal e executar:
   kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kali:~$ ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, as tarxetas de rede: loopback(lo) e interna(eth0).
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflicto con este demo.
   root@kali:~# /etc/init.d/network-manager stop || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) para poder configurar de forma manual a configuración de rede e non ter conflicto con este xestor.
   root@kali:~# ip addr add 192.168.120.100/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.100 e máscara de subrede: 255.255.255.0.
   root@kali:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
2. **Crear formulario: arquivo index.php**
   kali@kali:~$ mkdir /home/kali/web #Crear cartafol /home/kali/web, o cal será o DocumenRoot do noso servidor web, é dicir, será o cartafol por defecto que publicará o noso servidor web.
   kali@kali:~$ cat > /home/kali/web/index.php <<EOF #Crear arquivo que será visitado por defecto no noso servidor web. Este arquivo conteń o formulario co cal imos a probar os métodos GET e POST.

```
<?php
if(isset(\$_POST['user'],\$_POST['password'])):
  \$user=\$_POST['user'];
  \$password=\$_POST['password'];
  echo "<div class='metodo'>";
  echo "<h1>Método POST</h1>";
  echo "<h2>Usuario: ".\$user."<br>";
  echo "<h2>Contrasinal: ".\$password."<br>";
  echo "</div>";
elseif(isset(\$_GET['user'],\$_GET['password'])):
  \$user=\$_GET['user'];
  \$password=\$_GET['password'];
  echo "<div class='metodo'>";
  echo "<h1>Método GET</h1>";
  echo "<h2>Usuario: ".\$user."<br>";
  echo "<h2>Contrasinal: ".\$password."<br>";
  echo "</div>";
endif;
?>
<!DOCTYPE html>
<html>
  <head>
    <title>Formulario-GET-POST-Proxy</title>
    <style>
      .mtop0{margin-top:0px;}
      .mtop10{margin-top:10px;}
      .center{position: absolute;left: 50%;top: 50%;}
      .estilo,.metodo{border:1px dashed black;border-radius:10px; padding: 14px 14px;
      background-color:lightcyan;}
      .metodo{background-color:yellow;}
    </style>
  </head>
  <body>
    <div class='center estilo'>
      <h2 class='mtop0'>Autentificación</h2>
      <form>
      <!--<form method='GET'>-->
      <!--<form method='POST'>-->
        <input type="text" name="user" placeholder="Usuario" autofocus required>
        <br>
        <input class='mtop10' type="password" name="password" placeholder="Contrasinal" required>
        <br>
        <input class='mtop10' type="submit" value="Entrar">
      </form>
    </div>
  </body>
</html>
EOF
```

3. **Lanzar e acceder ao servidor web (localhost)**
   kali@kali:~$ sudo php -S localhost:80 -t /home/kali/web & #Executar en segundo plano (&) cos permisos de root(administrador) o comando *php -S localhost:80 -t /home/kali/web*, o cal activa no porto TCP 80 un servidor web php sendo /home/kali/web o DocumentRoot publicado.
   kali@kali:~$ firefox http://localhost:80 #Lanzar o navegador firefox na URL http://localhost no porto TCP 80, realizando a execución en primer plano, é dicir, acceder ao servidor web php do paso anterior.

   Unha vez que accedamos á páxina paramos a execución do comando anterior premendo Ctrl+C no terminal onde executamos o comando firefox

4. **Lanzar e acceder ao servidor web (192.168.120.100)**
   kali@kali:~$ sudo php -S 192.168.120.100:80 -t /home/kali/web & #Executar en segundo plano (&) cos permisos de root(administrador) o comando *php -S 192.168.120.100:80 -t /home/kali/web*, o cal activa no porto TCP 80 un servidor web php sendo /home/kali/web o DocumentRoot publicado.
   kali@kali:~$ firefox http://192.168.120.100:80 & #Lanzar o navegador firefox na URL http://192.168.120.100 no porto TCP 80, realizando a execución en segundo plano (&), é dicir, acceder ao servidor web php do paso anterior.

5. **Formulario: Método GET**
   A. Introducir credenciais (usuario/contrasinal) no formulario, por exemplo usuario *kali* e contrasinal *abc123.*
   B. Premer no botón Enviar

      Podemos observar no código do arquivo index.php que na etiqueta <form> non está definido ningún método o que equivale ao método GET, é dicir, se non se indica método por defecto o envío do formulario farase mediante o método GET. Que acontece coa URL? Agora aparecen os parámetros na propia URL, é dicir, podemos ver os valores das variables user e password na propia URL separadas mediante o caracter &: `http://192.168.120.100/?user=kali&password=abc123.`

   C. Modificar o código do arquivo index.php para que o método a empregar no formulario sexa o método GET, é dicir, comentar a liña 36 e descomentar a liña 37:

```
<!--<form-->
<form method='GET'>
<!--<form method='POST'>-->
```

   D. Realizar de novo os pasos 5A e 5B

      Podemos observar que agora no código do arquivo index.php na etiqueta *form* está definido o método GET. Que acontece coa URL? Agora aparecen os parámetros na propia URL, é dicir, podemos ver os valores das variables user e password na propia URL separadas mediante o caracter &: `http://192.168.120.100/?user=kali&password=abc123.`

   E. Modificar a URL como segue: `http://192.168.120.100/?user=alumnado&password=12345678`
   F. Premer Enter na URL. Que acontece?

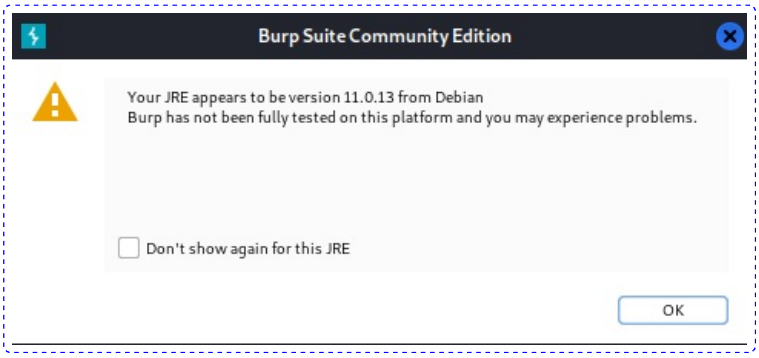
6. **Formulario: Método POST**
   A. Modificar o código do arquivo index.php para que o método a empregar no formulario sexa o método POST, é dicir, comentar a liña 37 e descomentar a liña 38:

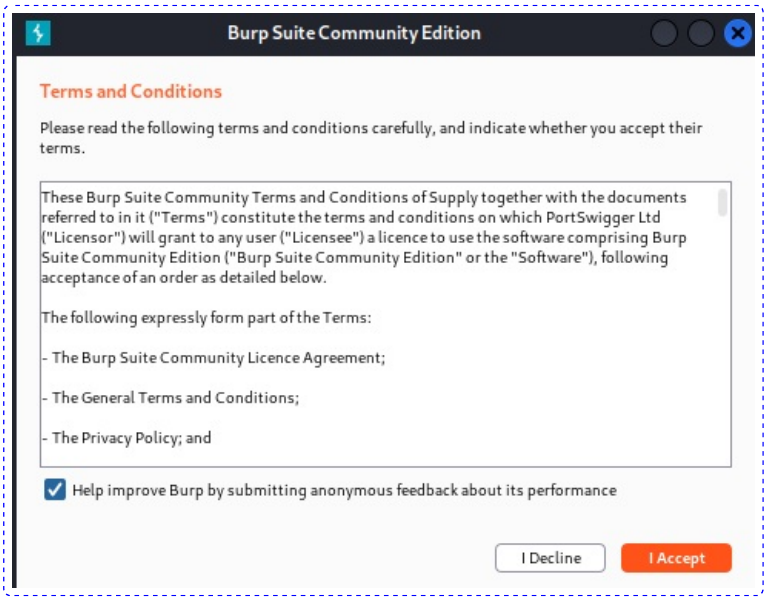
```
<!--<form-->
<!--form method='GET'>-->
<form method='POST'>
```

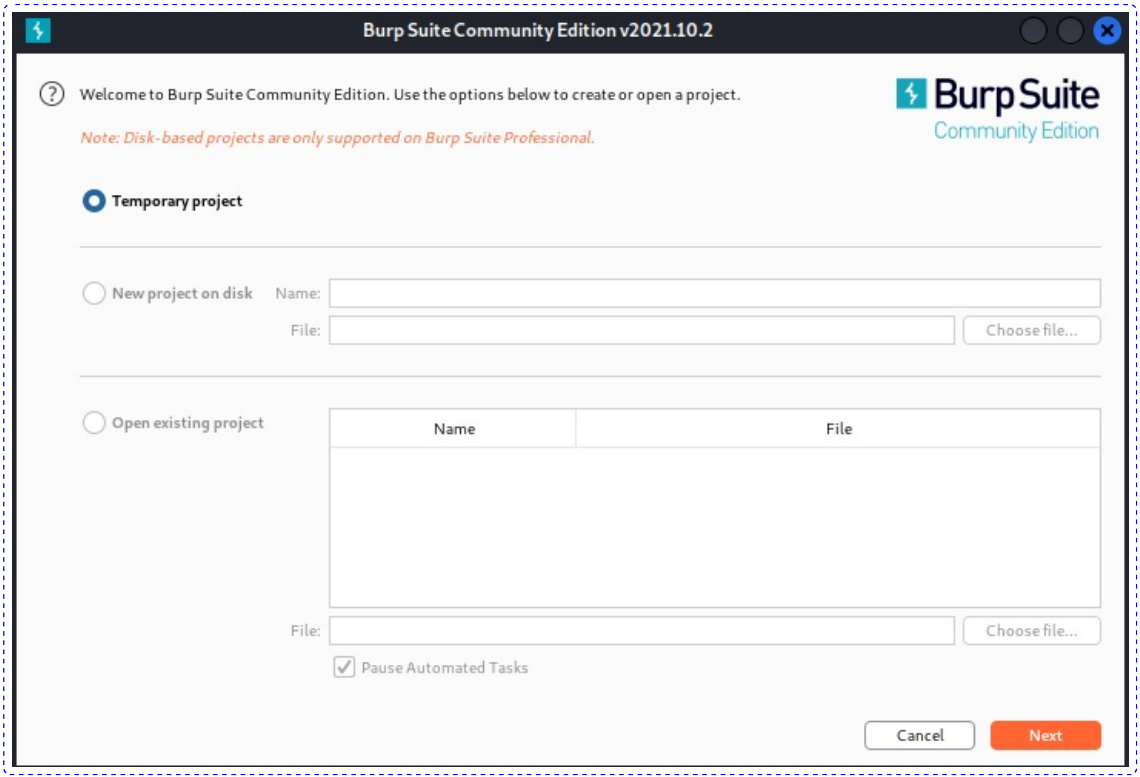
   B. Abrir unha nova lapela e acceder á URL http://192.168.120.100:80
   C. Realizar de novo os pasos 5A e 5B

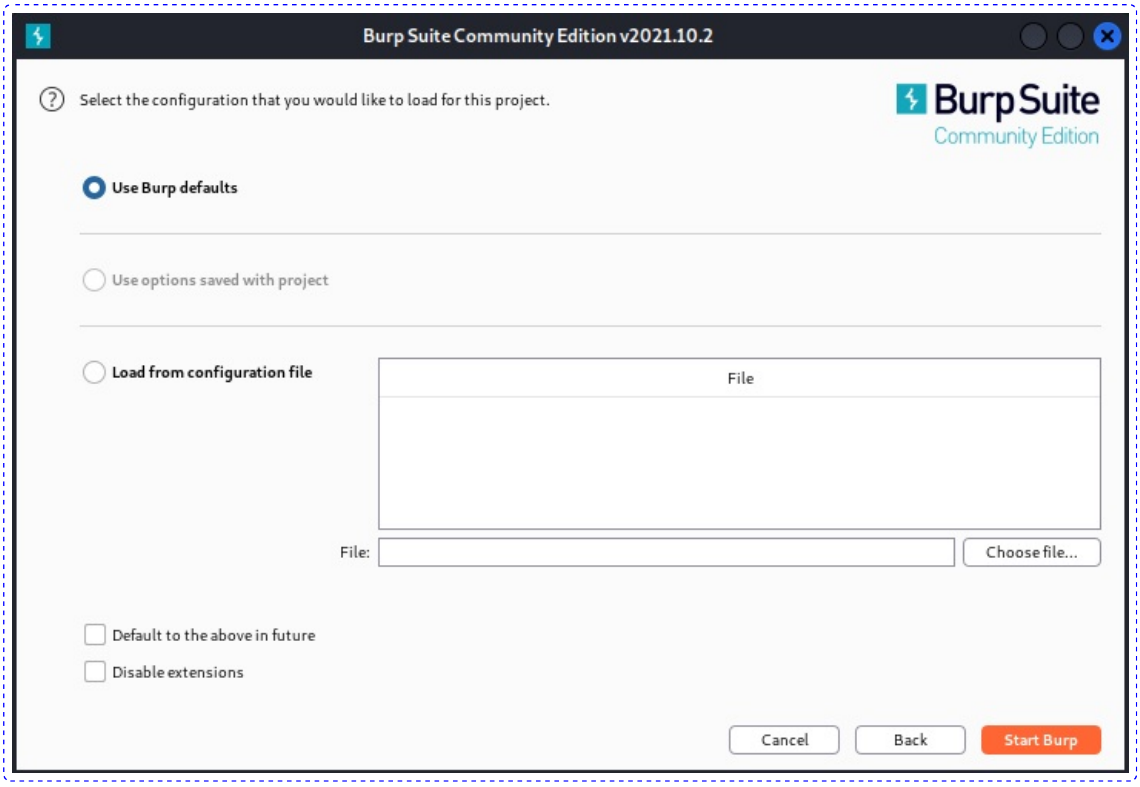
      Podemos observar que agora no código do arquivo index.php na etiqueta *form* está definido o método POST. Que acontece coa URL? Agora non aparecen os parámetros na propia URL, é dicir, non podemos ver os valores das variables user e password na propia URL separadas mediante o caracter &

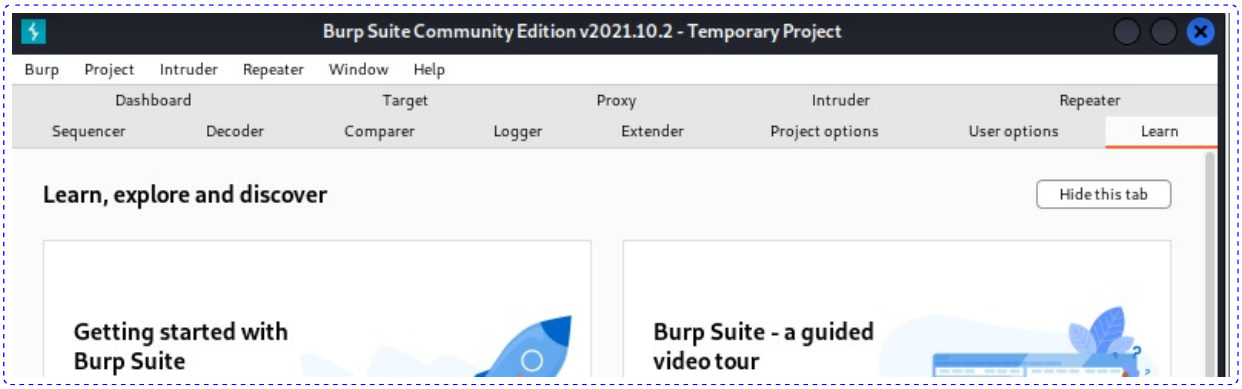
7. **Proxy burpsuite**
   A. Imos visualizar as variables introducidas co método POST mediante o Proxy burpsuite. Así, abrir un novo terminal e executar:
      kali@kali:~$ burpsuite & #Executar o proxy burpsuite en segundo plano (&).
   B. Aceptar Todo



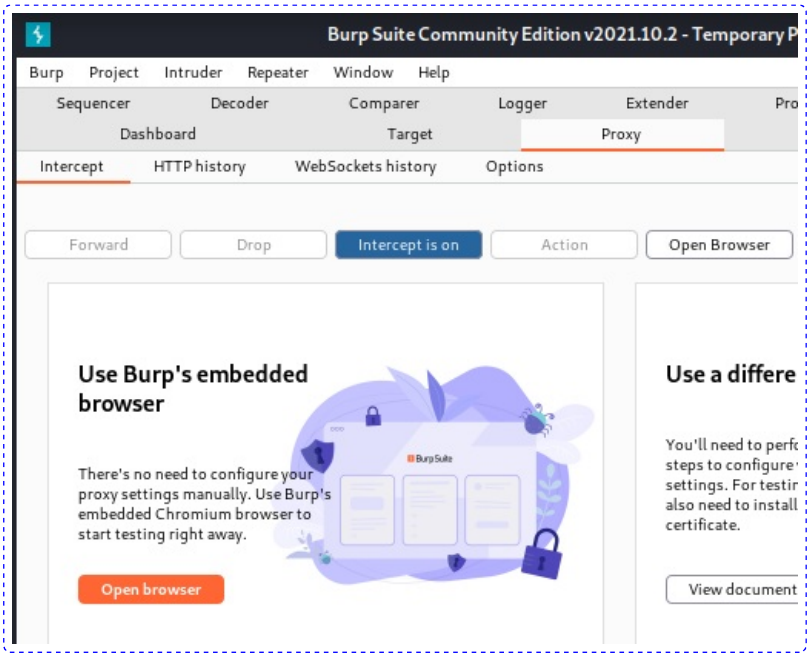


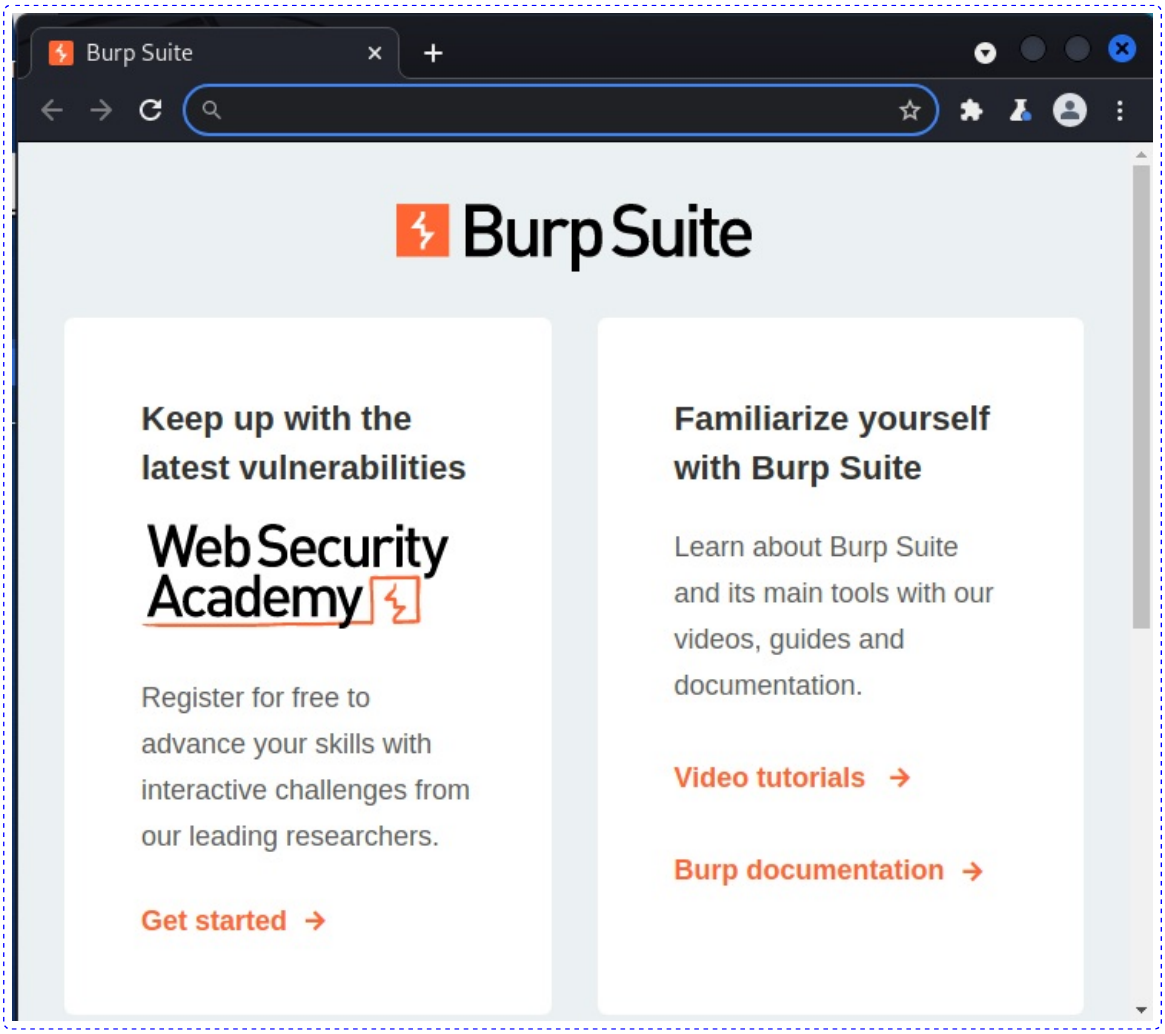




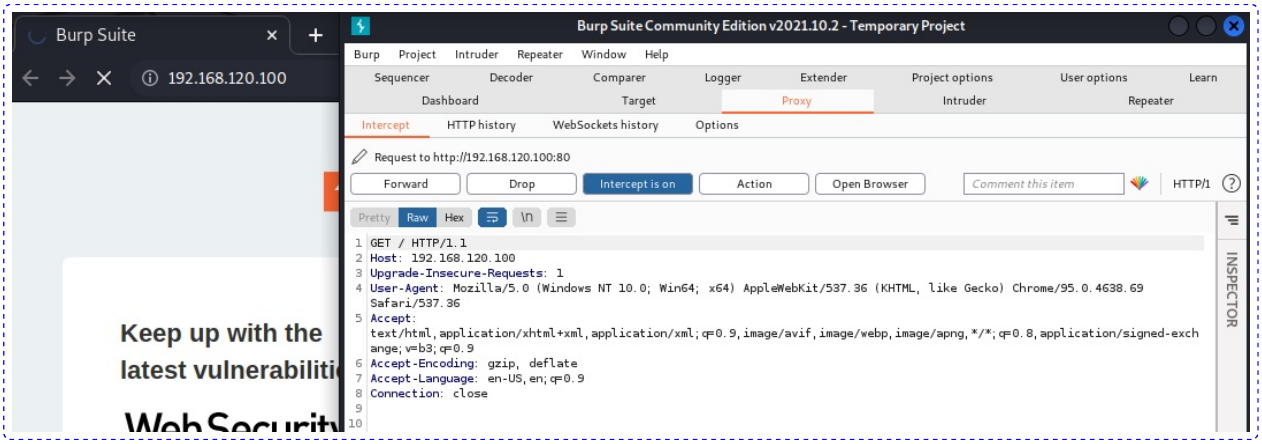


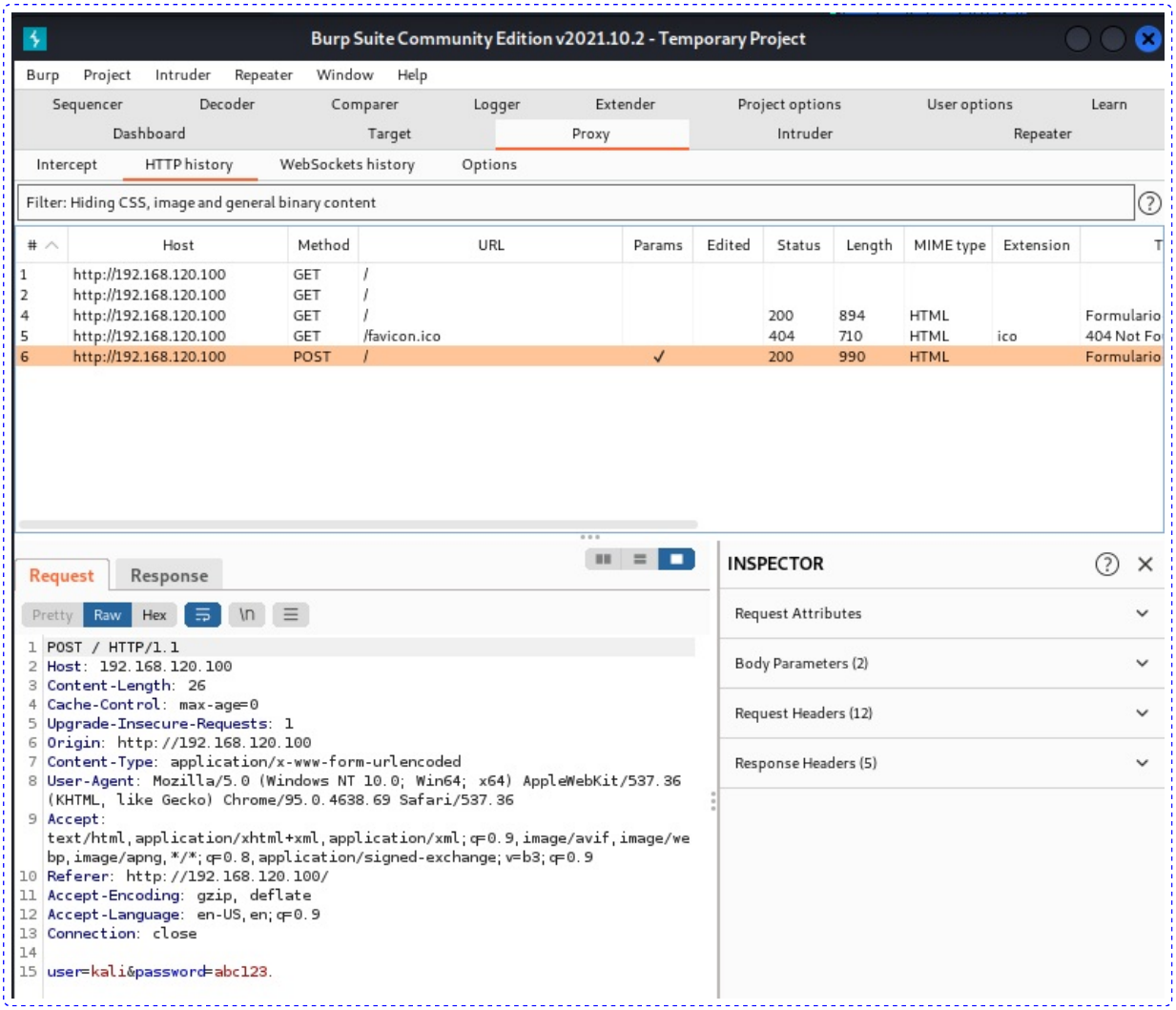
C. Seleccionar a lapela Proxy e executar o navegador embebido (Open browser)





D. Realizar de novo os pasos 6B e 6C





- O proxy burpsuite está activo por defecto na interface loopback na IP 127.0.0.1 e no porto TCP 8080.
- O navegador embebido xa está configurado para facer peticións a través do proxy burpsuite (127.0.0.1:8080).
- Todas as peticións do cliente quedaran gardadas na lapela HTTP History.
- Por defecto temos no Proxy → Intercept On, polo cal debemos premer en Forward para permitir que calquera petición do cliente pase a través do proxy e poida comunicarse co servidor web. Se non quixeramos que tivera lugar a petición podemos descartala premendo en Drop. Tamén poderiamos pór Intercept a Off para permitir todas as peticións do cliente para revisalas logo na lapela HTTP History.
- Unha vez introducidas as credenciais podemos observar en HTTP History as peticións realizadas, escoller a comunicación POST e visualizar as variables introducidas. A maiores se tiveramos Intercept On tamén poderiamos modificar esas variables antes de envialas ao servidor na petición POST do cliente.

Ricardo Feijoo Costa